User's Manual

# RZ/A3M Group

## User's Manual: Hardware

Renesas Microprocessor
RZ Family / RZ/A Series

arm

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

Renesas Electronics
www.renesas.com

Rev.1.10 Mar 2025

# 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 or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving 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, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. 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 shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended 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; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.

   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or 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 damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (“Vulnerability Issues”). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, 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 and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. Renesas Electronics products and technologies shall 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 shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.

(Note2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

© 2025 Renesas Electronics Corporation. All rights reserved.

## Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan

www.renesas.com

## Contact information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit: www.renesas.com/contact/.

## Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

For the “Cortex” notation, it is used as follows;
- Arm® Cortex®-A55
Note that after this page, they may be noted as Cortex-A55 respectively.

Examples of trademark or registered trademark used in the RZ/A series, User’s Manual: Hardware.

CoreLink™: CoreLink is a trademark of Arm Limited.
CoreSight™: CoreSight is a trademark of Arm Limited.
Neon™: Neon is a trademark of Arm Limited.
MIPI®: MIPI is a registered trademark of MIPI Alliance, Inc.
AMBA®: AMBA is a registered trademark of Arm Limited.
DynamIQ™: DynamIQ is a trademark of Arm Limited.

Note that in each section of the Manual, trademark notation of ® and TM may be omitted.
All other trademarks and registered trademarks are the property of their respective owners.

© 2025 Renesas Electronics Corporation. All rights reserved.

# General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit 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. Precaution against Electrostatic Discharge (ESD)
   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.
2. Processing at power-on
   The state of the product is undefined at the time 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 time 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 time 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 time when power is supplied until the power reaches the level at which resetting is specified.
3. Input of signal during power-off state
   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.
4. Handling of unused pins
   Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the 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.
5. Clock signals
   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is 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. Additionally, 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.
6. Voltage application waveform at input pin
   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between $V_{IL}$ (Max.) and $V_{IH}$ (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between $V_{IL}$ (Max.) and $V_{IH}$ (Min.).
7. Prohibition of access to reserved addresses
   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.
8. Differences between products
   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of 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.

# Table of Contents


1. Overview ..... 1-1
1.1 Introduction ..... 1-1
1.2 List of Specifications ..... 1-2
1.2.1 CPU Core ..... 1-2
1.2.2 CPU Peripheral ..... 1-3
1.2.3 Internal Memory ..... 1-3
1.2.4 External Memory Interface ..... 1-4
1.2.5 Video Processing Unit ..... 1-4
1.2.6 Display Interface ..... 1-5
1.2.7 Sound Interface ..... 1-5
1.2.8 Storage and Network ..... 1-6
1.2.9 Timer ..... 1-6
1.2.10 Peripheral Module ..... 1-7
1.2.11 Others ..... 1-7
1.2.12 Power Supply Voltage ..... 1-8
1.2.13 Temperature Range ..... 1-8
1.2.14 Quality level ..... 1-8
1.2.15 Package ..... 1-8
1.3 Product Lineup ..... 1-8
1.4 Pin ..... 1-9
1.4.1 Pin Assignment ..... 1-9
1.4.2 External Pins ..... 1-9
2. System CPU Cortex-A55 ..... 2-1
2.1 Features ..... 2-1
2.2 Resets ..... 2-2
2.2.1 Cold Reset ..... 2-2
2.2.2 Core Warm Reset ..... 2-3
2.2.3 Debug Reset ..... 2-3
2.3 Low Power Consumption Mode ..... 2-4
2.3.1 Cortex-A55 Sleep Mode ..... 2-4
2.4 Function Reference ..... 2-5
4. Boot Mode ..... 4-1
4.1 Overview ..... 4-1
4.2 Operation ..... 4-2
4.2.1 Boot Mode 3 (1.8-V Single or Quad Serial NOR Flash Memory) ..... 4-2
4.2.1.1 External Connections ..... 4-3
4.2.1.2 Overview of Operation ..... 4-4
4.2.1.3 Operation of Booting from 1.8-V Serial NOR Flash Memory ..... 4-4
4.2.1.4 Allocation of the Loader Program in the Serial NOR Flash Memory ..... 4-5
4.2.2 Boot Mode 4 (3.3-V Single or Quad Serial NOR Flash Memory) ..... 4-7
4.2.2.1 External Connections ..... 4-8
4.2.2.2 Overview of Operation ..... 4-9
4.2.2.3 Operation of Booting from 3.3-V Serial NOR Flash Memory ..... 4-9
4.2.2.4 Note ..... 4-9
4.2.3 Boot Mode 5 (SCIF Downloading) ..... 4-10
4.2.3.1 External Connections ..... 4-10
4.2.3.2 Overview of Operation ..... 4-11
4.2.3.3 Operation of Booting through SCIF Downloading ..... 4-12
4.2.3.4 Allocation of the Loader Program for SCIF Downloading ..... 4-13

4.2.3.5 Note ..... 4-13
4.2.4 Boot Mode 6 (1.8-V Single or Quad Serial NAND Flash Memory) ..... 4-14
4.2.4.1 External Connections ..... 4-15
4.2.4.2 Overview of Operation ..... 4-16
4.2.4.3 Operation of Booting from 1.8-V Serial NAND Flash Memory ..... 4-16
4.2.4.4 Allocation of the Loader Program in the Serial NAND Flash Memory ..... 4-18
4.2.4.5 Padding in SHA-256 Function ..... 4-19
4.2.5 Boot Mode 7 (3.3-V Single or Quad Serial NAND Flash Memory) ..... 4-20
4.2.5.1 External Connections ..... 4-21
4.2.5.2 Overview of Operation ..... 4-22
4.2.5.3 Operation of Booting from 3.3-V Serial NAND Flash Memory ..... 4-22
5. LSI Internal Bus ..... 5-1
5.1 Overview ..... 5-1
5.1.1 Features ..... 5-1
5.1.2 Block Diagram of LSI Internal Bus ..... 5-1
5.2 Area Maps ..... 5-3
5.2.1 Overall Address Space ..... 5-3
5.3 Accessible Areas ..... 5-7
5.4 Bus System Control ..... 5-9
5.4.1 Bus Error Interrupt Generation ..... 5-9
6. System Controller (SYSC) ..... 6-1
6.1 Overview ..... 6-1
6.1.1 Features ..... 6-1
6.1.2 Block Diagram of SYSC ..... 6-2
6.2 Register Configuration ..... 6-3
6.3 Register Descriptions ..... 6-4
6.3.1 ECCRAM0 ECC Setting Register (SYS_RAM0_ECC) ..... 6-4
6.3.2 ECCRAM0 Access Control Register (SYS_RAM0_EN) ..... 6-4
6.3.3 ECCRAM1 ECC Setting Register (SYS_RAM1_ECC) ..... 6-5
6.3.4 ECCRAM1 Access Control Register (SYS_RAM1_EN) ..... 6-5
6.3.5 WDT0 Control Register (SYS_WDT0_CTRL) ..... 6-6
6.3.6 I2C0 Config Register (SYS_I2C0_CFG) ..... 6-7
6.3.7 I2C1 Config Register (SYS_I2C1_CFG) ..... 6-7
6.3.8 CA55 Core0 Reset Vector Address Low Configuration Register (SYS_CA55_CFG_RVAL0) ..... 6-8
6.3.9 CA55 Core0 Reset Vector Address High Configuration Register (SYS_CA55_CFG_RVAH0) ..... 6-8
6.3.10 LSI Mode Signal Register (SYS_LSI_MODE) ..... 6-9
6.3.11 LSI Device ID Register (SYS_LSI_DEVID) ..... 6-10
6.3.12 LSI Product Register (SYS_LSI_PRR) ..... 6-11
6.3.13 SYS_LP_CTL2 ..... 6-12
6.3.14 SYS_LP_CTL5 ..... 6-13
6.3.15 SYS_LP_CTL6 ..... 6-14
6.3.16 SYS_LP_CA55CK_CTL1 ..... 6-15
6.3.17 SYS_LP_CA55CK_CTL2 ..... 6-16
6.3.18 SYS_LP_CA55CK_CTL3 ..... 6-18
6.3.19 General Register0 (SYS_GPREG_0) ..... 6-20
6.3.20 General Register1 (SYS_GPREG_1) ..... 6-20
6.3.21 General Register2 (SYS_GPREG_2) ..... 6-21
6.3.22 General Register3 (SYS_GPREG_3) ..... 6-21
6.4 Operation ..... 6-22
6.4.1 External Terminal State Capture Function ..... 6-22
6.4.2 WDT Stop Control Function ..... 6-22
7. Clock Pulse Generator (CPG) ..... 7-1
7.1 Features ..... 7-1

7.1.1 Overview ..... 7-1
7.1.2 CPG configuration ..... 7-2
7.1.3 Pin description ..... 7-3
7.1.4 Clock ..... 7-4
7.1.4.1 Clock System Diagram ..... 7-4
7.1.4.2 Clock List ..... 7-6
7.2 Register description ..... 7-7
7.2.1 Register attributes ..... 7-7
7.2.2 Register classification ..... 7-7
7.2.3 Register configuration ..... 7-8
7.2.4 Register Descriptions ..... 7-12
7.2.4.1 PLLn (SSCG) Standby Control Register (CPG_SAMPLLn_STBY) (n = 1 or 4) ..... 7-12
7.2.4.2 PLLn (SSCG) Output Clock Setting Register 1 (CPG_SAMPLLn_CLK1) (n = 1 or 4) ..... 7-14
7.2.4.3 PLLn (SSCG) Output Clock Setting Register 2 (CPG_SAMPLLn_CLK2) (n = 1 or 4) ..... 7-16
7.2.4.4 PLLn (SSCG) Monitor Register (CPG_SAMPLLn_MON) (n = 1 or 4) ..... 7-19
7.2.4.5 PLL1_SEL_SETTING Register (CPG_PLL1_SETTING) ..... 7-20
7.2.4.6 CPG_OTPPLL0_MON Register (CPG_OTPPLL0_MON) ..... 7-21
7.2.4.7 CPG_OTPPLL1_MON Register (CPG_OTPPLL1_MON) ..... 7-22
7.2.4.8 CPG_OTPPLL2_MON Register (CPG_OTPPLL2_MON) ..... 7-23
7.2.4.9 PLL5 (SSCG) Standby Control Register (CPG_SIPLL5_STBY) ..... 7-24
7.2.4.10 PLL5 (SSCG) Output Clock Setting Register 1 (CPG_SIPLL5_CLK1) ..... 7-26
7.2.4.11 PLL5 (SSCG) Output Clock Setting Register 2 (CPG_SIPLL5_CLK2) ..... 7-27
7.2.4.12 PLL5 (SSCG) Output Clock Setting Register 3 (CPG_SIPLL5_CLK3) ..... 7-29
7.2.4.13 PLL5 (SSCG) Output Clock Setting Register 4 (CPG_SIPLL5_CLK4) ..... 7-30
7.2.4.14 PLL5 (SSCG) Output Clock Setting Register 5 (CPG_SIPLL5_CLK5) ..... 7-31
7.2.4.15 PLLn (SSCG) Monitor Register (CPG_SIPLLn_MON) (n = 2, 3, or 5) ..... 7-33
7.2.4.16 Division Ratio Setting (PLL1) Register (CPG_PL1_DDIV) ..... 7-34
7.2.4.17 Division Ratio Setting (PLL2) Register (CPG_PL2_DDIV) ..... 7-35
7.2.4.18 Division Ratio Setting (PLL3A) Register (CPG_PL3A_DDIV) ..... 7-37
7.2.4.19 Division Ratio Setting (PLL3B) Register (CPG_PL3B_DDIV) ..... 7-39
7.2.4.20 Source Clock Setting (SDHI) Register (CPG_PL2SDHI_DSEL) ..... 7-40
7.2.4.21 Source Clock Setting (DDR) Register (CPG_PL4_DSEL) ..... 7-41
7.2.4.22 Clock Status Monitor Register (CPG_CLKSTATUS) ..... 7-42
7.2.4.23 Source Clock Setting Register (CPG_PL3_SSEL) ..... 7-44
7.2.4.24 Division Ratio Setting (PLL5) Register (CPG_PL5_SDIV) ..... 7-45
7.2.4.25 Clock Control Register Cortex-A55 (CPG_CLKON_CA55) ..... 7-47
7.2.4.26 Clock Control Register ACPU (CPG_CLKON_SRAM_ACPU) ..... 7-49
7.2.4.27 Clock Control Register MCPU (CPG_CLKON_SRAM_MCPU) ..... 7-50
7.2.4.28 Clock Control Register GIC600 (CPG_CLKON_GIC600) ..... 7-51
7.2.4.29 Clock Control Register IA55 (CPG_CLKON_IA55) ..... 7-52
7.2.4.30 Clock Control Register CST (CPG_CLKON_CST) ..... 7-53
7.2.4.31 Clock Control Register SYC (CPG_CLKON_SYC) ..... 7-55
7.2.4.32 Clock Control Register DMAC_REG (CPG_CLKON_DMAC_REG) ..... 7-56
7.2.4.33 Clock Control Register GTM (CPG_CLKON_GTM) ..... 7-57
7.2.4.34 Clock Control Register MTU (CPG_CLKON_MTU) ..... 7-58
7.2.4.35 Clock Control Register POE3 (CPG_CLKON_POE3) ..... 7-59
7.2.4.36 Clock Control Register WDT (CPG_CLKON_WDT) ..... 7-60
7.2.4.37 Clock Control Register DDR (CPG_CLKON_DDR) ..... 7-62
7.2.4.38 Clock Control Register SPI_MULTI (CPG_CLKON_SPI_MULTI) ..... 7-63
7.2.4.39 Clock Control Register SDHI (CPG_CLKON_SDHI) ..... 7-64
7.2.4.40 Clock Control Register MIPI_DSI (CPG_CLKON_MIPI_DSI) ..... 7-66
7.2.4.41 Clock Control Register LCDC (CPG_CLKON_LCDC) ..... 7-68
7.2.4.42 Clock Control Register SSI (CPG_CLKON_SSI) ..... 7-69
7.2.4.44 Clock Control Register USB (CPG_CLKON_USB) ..... 7-70
7.2.4.45 Clock Control Register I2C (CPG_CLKON_I2C) ..... 7-71
7.2.4.46 Clock Control Register SCIF (CPG_CLKON_SCIF) ..... 7-72
7.2.4.47 Clock Control Register SCI (CPG_CLKON_SCI) ..... 7-74
7.2.4.48 Clock Control Register IRDA (CPG_CLKON_IRDA) ..... 7-75
7.2.4.49 Clock Control Register RSPI (CPG_CLKON_RSPI) ..... 7-76

7.2.4.50 Clock Control Register GPIO (CPG_CLKON_GPIO) ..... 7-77
7.2.4.51 Clock Control Register TSU (CPG_CLKON_TSU) ..... 7-78
7.2.4.52 Clock Control Register DRW (CPG_CLKON_DRW) ..... 7-79
7.2.4.53 Clock Monitor Register Cortex-A55 (CPG_CLKMON_CA55) ..... 7-80
7.2.4.54 Clock Monitor Register SRAM_ACPU (CPG_CLKMON_SRAM_ACPU) ..... 7-81
7.2.4.55 Clock Monitor Register SRAM_MCPU (CPG_CLKMON_SRAM_MCPU) ..... 7-82
7.2.4.56 Clock Monitor Register GIC600 (CPG_CLKMON_GIC600) ..... 7-83
7.2.4.57 Clock Monitor Register IA55 (CPG_CLKMON_IA55) ..... 7-84
7.2.4.58 Clock Monitor Register CST (CPG_CLKMON_CST) ..... 7-85
7.2.4.59 Clock Monitor Register SYC (CPG_CLKMON_SYC) ..... 7-86
7.2.4.60 Clock Monitor Register DMAC_REG (CPG_CLKMON_DMAC_REG) ..... 7-87
7.2.4.61 Clock Monitor Register GTM (CPG_CLKMON_GTM) ..... 7-88
7.2.4.62 Clock Monitor Register MTU (CPG_CLKMON_MTU) ..... 7-89
7.2.4.63 Clock Monitor Register POE3 (CPG_CLKMON_POE3) ..... 7-90
7.2.4.64 Clock Monitor Register WDT (CPG_CLKMON_WDT) ..... 7-91
7.2.4.65 Clock Monitor Register DDR (CPG_CLKMON_DDR) ..... 7-92
7.2.4.66 Clock Monitor Register SPI_MULTI (CPG_CLKMON_SPI_MULTI) ..... 7-93
7.2.4.67 Clock Monitor Register SDHI (CPG_CLKMON_SDHI) ..... 7-94
7.2.4.68 Clock Monitor Register MIPI_DSI (CPG_CLKMON_MIPI_DSI) ..... 7-95
7.2.4.69 Clock Monitor Register LCDC (CPG_CLKMON_LCDC) ..... 7-96
7.2.4.70 Clock Monitor Register SSI (CPG_CLKMON_SSI) ..... 7-97
7.2.4.71 Clock Monitor Register USB (CPG_CLKMON_USB) ..... 7-98
7.2.4.72 Clock Monitor Register I2C (CPG_CLKMON_I2C) ..... 7-99
7.2.4.73 Clock Monitor Register SCIF (CPG_CLKMON_SCIF) ..... 7-100
7.2.4.74 Clock Monitor Register SCI (CPG_CLKMON_SCI) ..... 7-101
7.2.4.75 Clock Monitor Register IRDA (CPG_CLKMON_IRDA) ..... 7-102
7.2.4.76 Clock Monitor Register RSPI (CPG_CLKMON_RSPI) ..... 7-103
7.2.4.77 Clock Monitor Register GPIO (CPG_CLKMON_GPIO) ..... 7-104
7.2.4.78 Clock Monitor Register TSU (CPG_CLKMON_TSU) ..... 7-105
7.2.4.79 Clock Monitor Register DRW (CPG_CLKMON_DRW) ..... 7-106
7.2.4.80 Reset Control Register Cortex-A55 (CPG_RST_CA55) ..... 7-107
7.2.4.81 Reset Control Register SRAM_ACPU (CPG_RST_SRAM_ACPU) ..... 7-110
7.2.4.82 Reset Control Register SRAM_MCPU (CPG_RST_SRAM_MCPU) ..... 7-111
7.2.4.83 Reset Control Register GIC600 (CPG_RST_GIC600) ..... 7-112
7.2.4.84 Reset Control Register IA55 (CPG_RST_IA55) ..... 7-113
7.2.4.85 Reset Control Register SYC (CPG_RST_SYC) ..... 7-114
7.2.4.86 Reset Control Register DMAC (CPG_RST_DMAC) ..... 7-115
7.2.4.87 Reset Control Register GTM (CPG_RST_GTM) ..... 7-116
7.2.4.88 Reset Control Register MTU (CPG_RST_MTU) ..... 7-117
7.2.4.89 Reset Control Register POE3 (CPG_RST_POE3) ..... 7-118
7.2.4.90 Reset Control Register WDT (CPG_RST_WDT) ..... 7-119
7.2.4.91 Reset Control Register DDR (CPG_RST_DDR) ..... 7-120
7.2.4.92 Reset Control Register SPI (CPG_RST_SPI) ..... 7-122
7.2.4.93 Reset Control Register SDHI (CPG_RST_SDHI) ..... 7-123
7.2.4.94 Reset Control Register MIPI_DSI (CPG_RST_MIPI_DSI) ..... 7-124
7.2.4.95 Reset Control Register LCDC (CPG_RST_LCDC) ..... 7-125
7.2.4.96 Reset Control Register SSIF (CPG_RST_SSIF) ..... 7-126
7.2.4.97 Reset Control Register USB (CPG_RST_USB) ..... 7-127
7.2.4.98 Reset Control Register I2C (CPG_RST_I2C) ..... 7-128
7.2.4.99 Reset Control Register SCIF (CPG_RST_SCIF) ..... 7-129
7.2.4.100 Reset Control Register SCI (CPG_RST_SCI) ..... 7-131
7.2.4.101 Reset Control Register IRDA (CPG_RST_IRDA) ..... 7-132
7.2.4.102 Reset Control Register RSPI (CPG_RST_RSPI) ..... 7-133
7.2.4.103 Reset Control Register GPIO (CPG_RST_GPIO) ..... 7-134
7.2.4.104 Reset Control Register TSU (CPG_RST_TSU) ..... 7-135
7.2.4.105 Reset Control Register DRW (CPG_RST_DRW) ..... 7-136
7.2.4.106 Reset Monitor Register Cortex-A55 (CPG_RSTMON_CA55) ..... 7-137
7.2.4.107 Reset Monitor Register SRAM_ACPU (CPG_RSTMON_SRAM_ACPU) ..... 7-139
7.2.4.108 Reset Monitor Register SRAM_MCPU (CPG_RSTMON_SRAM_MCPU) ..... 7-140

7.2.4.109 Reset Monitor Register GIC600 (CPG_RSTMON_GIC600) ..... 7-141
7.2.4.110 Reset Monitor Register IA55 (CPG_RSTMON_IA55) ..... 7-142
7.2.4.111 Reset Monitor Register SYC (CPG_RSTMON_SYC) ..... 7-143
7.2.4.112 Reset Monitor Register DMAC (CPG_RSTMON_DMAC) ..... 7-144
7.2.4.113 Reset Monitor Register GTM (CPG_RSTMON_GTM) ..... 7-145
7.2.4.114 Reset Monitor Register MTU (CPG_RSTMON_MTU) ..... 7-146
7.2.4.115 Reset Monitor Register POE3 (CPG_RSTMON_POE3) ..... 7-147
7.2.4.116 Reset Monitor Register WDT (CPG_RSTMON_WDT) ..... 7-148
7.2.4.117 Reset Monitor Register DDR (CPG_RSTMON_DDR) ..... 7-149
7.2.4.118 Reset Monitor Register SPI (CPG_RSTMON_SPI) ..... 7-150
7.2.4.119 Reset Monitor Register SDHI (CPG_RSTMON_SDHI) ..... 7-151
7.2.4.120 Reset Monitor Register MIPI_DSI (CPG_RSTMON_MIPI_DSI) ..... 7-152
7.2.4.121 Reset Monitor Register LCDC (CPG_RSTMON_LCDC) ..... 7-153
7.2.4.122 Reset Monitor Register SSIF (CPG_RSTMON_SSIF) ..... 7-154
7.2.4.123 Reset Monitor Register USB (CPG_RSTMON_USB) ..... 7-155
7.2.4.124 Reset Monitor Register I2C (CPG_RSTMON_I2C) ..... 7-156
7.2.4.125 Reset Monitor Register SCIF (CPG_RSTMON_SCIF) ..... 7-157
7.2.4.126 Reset Monitor Register SCI (CPG_RSTMON_SCI) ..... 7-158
7.2.4.127 Reset Monitor Register IRDA (CPG_RSTMON_IRDA) ..... 7-159
7.2.4.128 Reset Monitor Register RSPI (CPG_RSTMON_RSPI) ..... 7-160
7.2.4.129 Reset Monitor Register GPIO (CPG_RSTMON_GPIO) ..... 7-161
7.2.4.130 Reset Monitor Register TSU (CPG_RSTMON_TSU) ..... 7-162
7.2.4.131 Reset Monitor Register DRW (CPG_RSTMON_DRW) ..... 7-163
7.2.4.132 WDT Overflow System Reset Register (CPG_WDTOVF_RST) ..... 7-164
7.2.4.133 WDT Reset Selector Register (CPG_WDTRST_SEL) ..... 7-165
7.2.4.134 MSTOP Register ACPU (CPG_BUS_ACPU_MSTOP) ..... 7-167
7.2.4.135 MSTOP Register MCPU1 (CPG_BUS_MCPU1_MSTOP) ..... 7-168
7.2.4.136 MSTOP Register MCPU2 (CPG_BUS_MCPU2_MSTOP) ..... 7-170
7.2.4.137 MSTOP Register PERI_COM (CPG_BUS_PERI_COM_MSTOP) ..... 7-173
7.2.4.138 MSTOP Register PERI_CPU (CPG_BUS_PERI_CPU_MSTOP) ..... 7-175
7.2.4.139 MSTOP Register PERI_DDR (CPG_BUS_PERI_DDR_MSTOP) ..... 7-177
7.2.4.140 MSTOP Register PERI_VIDEO (CPG_BUS_PERI_VIDEO_MSTOP) ..... 7-178
7.2.4.141 MSTOP Register REG0 (CPG_BUS_REG0_MSTOP) ..... 7-180
7.2.4.142 MSTOP Register REG1 (CPG_BUS_REG1_MSTOP) ..... 7-182
7.2.4.143 Other Function Register 1 (CPG_OTHERFUNC1_REG) ..... 7-184
7.2.4.144 Other Function Register 2 (CPG_OTHERFUNC2_REG) ..... 7-185
7.2.4.145 Clock Control Register MBUS (CPG_CLKON_MBUS) ..... 7-186
7.2.4.146 Clock Monitor Register MBUS (CPG_CLKMON_MBUS) ..... 7-187
7.3 Functions of the CPG ..... 7-188
7.3.1 Clock Control ..... 7-188
7.3.1.1 PLL Control, Setting, and Monitoring Functions ..... 7-188
7.3.1.2 Clock Frequency Dividers and Selectors ..... 7-190
7.3.1.3 Fixed Frequency Dividers ..... 7-190
7.3.1.4 PLL Clock Monitoring ..... 7-191
7.3.2 Clock Generation and Control Functions ..... 7-192
7.3.2.1 Range of Reset Application ..... 7-192
7.3.2.2 System Reset (External Pin) ..... 7-193
7.3.2.3 WDT System Reset ..... 7-193
7.3.2.4 Module Reset Application Control ..... 7-197
7.4 Operating Procedures ..... 7-198
7.4.1 Procedures for Supplying and Stopping Module Clocks ..... 7-198
7.4.2 Procedures for Supplying and Stopping Reset Signals ..... 7-199
7.4.3 Procedure for Activating Modules ..... 7-200
7.4.4 Procedure for Activating the Modules Related to PLL5 ..... 7-201
7.4.5 Procedures for PLL Setup ..... 7-203
7.4.5.1 Procedure for Setting PLL Normal Mode (Changing the Output Clock and SSCG Mode) ..... 7-203
7.4.5.2 Procedure for Setting PLL Standby Mode ..... 7-205
7.4.5.3 PLLs with Restrictions ..... 7-205

7.4.6 Procedure for Switching the Division Ratio of the Dynamic Switching Frequency Dividers ..... 7-206
7.4.7 Procedure for Switching Clocks by the Dynamic Switching Frequency Selectors ..... 7-207
7.4.8 Procedure for Switching Clocks by the Static Switching Frequency Dividers and Selectors (Procedure for Switching Source Clocks) ..... 7-208
7.4.9 Boot Sequence (Normal Operation) ..... 7-208
7.4.10 Boot Sequence (Debug Mode) ..... 7-208
7.4.11 General Operating Procedures ..... 7-209
7.4.11.1 Procedure for Writing to Registers ..... 7-209
8. Interrupt Controller ..... 8-1
8.1 Features ..... 8-1
8.1.1 External Signal Pins ..... 8-2
8.1.2 Block Diagram ..... 8-3
8.2 Interrupt Mapping ..... 8-4
8.3 GIC-600 Register Configuration ..... 8-17
8.4 GIC-600 Register Descriptions ..... 8-17
8.5 IA55 Register Configuration ..... 8-18
8.6 IA55 Register Descriptions ..... 8-19
8.6.1 NMI Status Control Register (NSCR) ..... 8-19
8.6.2 NMI Interrupt Type Selection Register (NITSR) ..... 8-20
8.6.3 IRQ Status Control Register (ISCR) ..... 8-21
8.6.4 IRQ Interrupt Type Selection Register (IITSR) ..... 8-22
8.6.5 TINT Status Control Register (TSCR) ..... 8-23
8.6.6 TINT Interrupt Type Selection Register0 (TITSR0) ..... 8-24
8.6.7 TINT Interrupt Type Selection Register1 (TITSR1) ..... 8-24
8.6.8 TINT Source Selection Register0 (TSSR0) ..... 8-25
8.6.9 TINT Source Selection Register1 (TSSR1) ..... 8-26
8.6.10 TINT Source Selection Register2 (TSSR2) ..... 8-27
8.6.11 TINT Source Selection Register3 (TSSR3) ..... 8-28
8.6.12 TINT Source Selection Register4 (TSSR4) ..... 8-29
8.6.13 TINT Source Selection Register5 (TSSR5) ..... 8-30
8.6.14 TINT Source Selection Register6 (TSSR6) ..... 8-31
8.6.15 TINT Source Selection Register7 (TSSR7) ..... 8-32
8.6.16 Bus Error Interrupt Status Control Register0 (BEISR0) ..... 8-33
8.6.17 Bus Error Interrupt Status Control Register1 (BEISR1) ..... 8-34
8.6.18 ECCRAM Error Interrupt Status Control Register0 (EREISR0) ..... 8-35
8.6.19 ECCRAM Error Interrupt Status Control Register1 (EREISR1) ..... 8-37
8.7 Operation ..... 8-39
8.7.1 NMI Interrupt ..... 8-39
8.7.2 IRQ Interrupt ..... 8-39
8.7.3 GPIO Interrupt (TINT) ..... 8-40
8.7.4 Internal Interrupt ..... 8-42
8.7.4.1 Bus Error Interrupt ..... 8-42
8.7.4.2 ECCRAM Error Interrupt ..... 8-44
8.8 Usage Note ..... 8-46
8.8.1 Precaution when use the peripheral modules which can initiate DMA Controller. ..... 8-46
8.8.2 Clear Timing of Interrupt Cause ..... 8-46
8.8.3 Precaution when Changing Interrupt Settings ..... 8-46
9. DDR3L SDRAM Memory Controller (MEMC) ..... 9-1

9.1 Features ..... 9-1
9.2 Interface ..... 9-3
9.2.1 Power, Ground ..... 9-3
9.2.2 External Pins ..... 9-3
9.3 Usage Note ..... 9-4
9.3.1 Power On ..... 9-4
10. On-chip RAM ..... 10-1
10.1 Features ..... 10-1
11. Error Correcting Code (ECC) for On-Chip RAM ..... 11-1
11.1 Overview ..... 11-1
11.1.1 Features ..... 11-1
11.2 Register Configuration ..... 11-2
11.3 Register Descriptions ..... 11-5
11.3.1 ECC Control Registers_n (<REG_base>CTL_n) (n = 0 to 7) ..... 11-5
11.3.2 ECC Error Address Registers m_n (<REG_base>EADm_n) (m = 0 to 7) (n = 0 to 7) ..... 111-8
11.4 Initializing the ECC Function ..... 11-9
11.5 Initializing the Detection-Usage Areas of the On-Chip RAM ..... 11-9
11.6 Usage Notes ..... 11-10
11.6.1 Notes on Setting the Registers of This Module ..... 11-10
11.6.2 8- or 16-Bit Access ..... 11-10
11.6.3 Numbers of Cycles for Reading from or Writing to a RAM Area with the ECC Function Enabled ..... 111-10
11.6.4 ECC Error Address Register Overwrite Conditions ..... 11-11
14. Direct Memory Access Controller ..... 14-1
14.1 Features ..... 14-1
14.2 Input/Output Pins ..... 14-1
14.3 Register Configuration ..... 14-2
14.4 Register Descriptions ..... 14-4
14.4.1 Next Source Address Register n (N0SA_n, N1SA_n) ..... 14-12
14.4.2 Next Destination Address Register n (N0DA_n, N1DA_n) ..... 14-12
14.4.3 Next Transaction Byte Register n (N0TB_n, N1TB_n) ..... 14-13
14.4.4 Current Source Address Register n (CRSA_n) ..... 14-14
14.4.5 Current Destination Address Register n (CRDA_n) ..... 14-15
14.4.6 Current Transaction Byte Register n (CRTB_n) ..... 14-16
14.4.7 Channel Status Register n (CHSTAT_n) ..... 14-17
14.4.8 Channel Control Register n (CHCTRL_n) ..... 14-21
14.4.9 Channel Configuration Register n (CHCFG_n) ..... 14-23
14.4.10 Channel Interval Register n (CHITVL_n) ..... 14-26
14.4.11 Channel Extension Register n (CHEXT_n) ..... 14-27
14.4.12 Next Link Address Register n (NXLA_n) ..... 14-28
14.4.13 Current Link Address Register n (CRLA_n) ..... 14-28
14.4.14 DMA Control Register (DCTRL_0_7, DCTRL_8_15) ..... 14-29
14.4.15 DMA Status EN Register (DSTAT_EN_0_7) ..... 14-30
14.4.16 DMA Status EN Register (DSTAT_EN_8_15) ..... 14-31
14.4.17 DMA Status ER Register (DSTAT_ER_0_7) ..... 14-32
14.4.18 DMA Status ER Register (DSTAT_ER_8_15) ..... 14-33

14.4.19 DMA Status END Register (DSTAT_END_0_7) ..... 14-34
14.4.20 DMA Status END Register (DSTAT_END_8_15) ..... 14-35
14.4.21 DMA Status TC Register (DSTAT_TC_0_7) ..... 14-36
14.4.22 DMA Status TC Register (DSTAT_TC_8_15) ..... 14-37
14.4.23 DMA Status SUS Register (DSTAT_SUS_0_7) ..... 14-38
14.4.24 DMA Status SUS Register (DSTAT_SUS_8_15) ..... 14-39
14.4.25 DMA Extension Resource Selectors 0 to 7 (DMARS0 to DMARS7) ..... 14-40
14.5 Operation ..... 14-43
14.5.1 Transfer Flow ..... 14-43
14.5.2 DMA Transfer Requests ..... 14-44
14.6 DMA Mode ..... 14-49
14.6.1 Mode Setting ..... 14-49
14.6.2 Register Mode ..... 14-50
14.6.3 Link Mode ..... 14-56
14.7 DMA Transfer ..... 14-65
14.7.1 Transfer Mode ..... 14-65
14.7.2 Priority Control for DMA Channels ..... 14-66
14.7.3 DMA Transfer Request ..... 14-69
14.7.4 DMA Transfer End Interrupt ..... 14-69
14.7.5 DMA Error Interrupt ..... 14-70
14.7.6 Interval Count Function ..... 14-70
14.7.7 Difference in Operation Due to the Transfer Size ..... 14-71
14.7.8 Transfer Status ..... 14-73
14.8 DMA Setting Examples ..... 14-77
14.8.1 Setting Example 1 (Register Mode/Hardware Request) ..... 14-77
14.8.2 Setting Example 2 (Register Mode/Software Request) ..... 14-79
14.8.3 Setting Example 3 (Register Mode/Continuous Execution) ..... 14-81
14.8.4 Setting Example 4 (Link Mode) ..... 14-83
14.8.5 Next Register Set Continuous Execution Setting ..... 14-86
14.9 Points for Caution ..... 14-88
14.9.1 Problem of the Interrupt Signal Overtaking DMA Transfer ..... 14-88
14.9.1.1 Overview ..... 14-88
14.9.1.2 Workaround ..... 14-89
15. System Counter (SYC) ..... 15-1
15.1 Overview ..... 15-1
15.1.1 Features ..... 15-1
15.1.2 Block Diagram of SYC ..... 15-2
15.2 Address Space ..... 15-3
15.3 Register Descriptions ..... 15-3
15.4 Operation ..... 15-4
15.4.1 System Bus Access Control ..... 15-4
15.4.2 Halt on Debug Function ..... 15-4
16. Multi-Function Timer Pulse Unit 3 (MTU3a) ..... 16-1
16.1 Overview ..... 16-1
16.2 Register Descriptions ..... 16-8
16.2.1 Timer Control Register (TCR) ..... 16-13
16.2.2 Timer Control Register 2 (TCR2) ..... 16-15
16.2.3 Timer Mode Register 1 (TMDR1) ..... 16-20
16.2.4 Timer Mode Registers 2 (TMDR2A and TMDR2B) ..... 16-23

16.2.5 Timer Mode Register 3 (TMDR3).....16-24
16.2.6 Timer I/O Control Register (TIOR).....16-26
16.2.7 Timer Compare Match Clear Register (TCNTCMPCLR).....16-45
16.2.8 Timer Interrupt Enable Register (TIER).....16-46
16.2.9 Timer Status Register (TSR).....16-50
16.2.10 Timer Buffer Operation Transfer Mode Register (TBTM).....16-51
16.2.11 Timer Input Capture Control Register (TICCR).....16-53
16.2.12 Timer Synchronous Clear Register (TSYCR).....16-54
16.2.13 Timer Counter (TCNT).....16-55
16.2.14 Timer Longword Counter (TCNTLW).....16-56
16.2.15 Timer General Register (TGR).....16-57
16.2.16 Timer Longword General Registers (TGRALW and TGRBLW).....16-59
16.2.17 Timer Start Registers (TSTRA, TSTRB, and TSTR).....16-60
16.2.18 Timer Synchronous Registers (TSYRA and TSYRB).....16-63
16.2.19 Timer Counter Synchronous Start Register (TCSYSTR).....16-65
16.2.20 Timer Read/Write Enable Registers (TRWERA and TRWERB).....16-67
16.2.21 Timer Output Master Enable Registers (TOERA and TOERB).....16-68
16.2.22 Timer Output Control Registers 1 (TOCR1A and TOCR1B).....16-70
16.2.23 Timer Output Control Registers 2 (TOCR2A and TOCR2B).....16-72
16.2.24 Timer Output Level Buffer Registers (TOLBRA and TOLBRB).....16-75
16.2.25 Timer Gate Control Register A (TGCRA).....16-76
16.2.26 Timer Subcounters (TCNTSA and TCNTSB).....16-78
16.2.27 Timer Cycle Data Registers (TCDRA and TCDRB).....16-78
16.2.28 Timer Cycle Buffer Registers (TCBRA and TCBRB).....16-79
16.2.29 Timer Dead Time Data Registers (TDDRA and TDDRB).....16-79
16.2.30 Timer Dead Time Enable Registers (TDERA and TDERB).....16-80
16.2.31 Timer Buffer Transfer Set Registers (TBTERA and TBTERB).....16-81
16.2.32 Timer Waveform Control Registers (TWCRA and TWCRB).....16-82
16.2.33 Noise Filter Control Register n (NFCRn) (n = 0 to 4, 6, 7, 8, C).....16-84
16.2.34 Noise Filter Control Register 5 (NFCR5).....16-88
16.2.35 Timer A/D Converter Start Request Control Register (TADCR).....16-89
16.2.36 Timer A/D Converter Start Request Cycle Set Registers (TADCORA and TADCORB).....1116-93
16.2.37 Timer A/D Converter Start Request Cycle Set Buffer Registers (TADCOBRA and TADCOBRB).....16-94
16.2.38 Timer Interrupt Skipping Mode Registers (TITMRA and TITMRB).....16-94
16.2.39 Timer Interrupt Skipping Set Registers 1 (TITCR1A and TITCR1B).....16-95
16.2.40 Timer Interrupt Skipping Counters 1 (TITCNT1A and TITCNT1B).....16-98
16.2.41 Timer Interrupt Skipping Set Registers 2 (TITCR2A and TITCR2B).....16-100
16.2.42 Timer Interrupt Skipping Counters 2 (TITCNT2A and TITCNT2B).....16-102
16.2.43 Bus Master Interface.....16-104
16.3 Operation.....16-105
16.3.1 Basic Functions.....16-105
16.3.2 Synchronous Operation.....16-111
16.3.3 Buffer Operation.....16-113
16.3.4 Cascaded Operation.....16-118
16.3.5 PWM Modes.....16-123
16.3.6 Phase Counting Mode.....16-128
16.3.6.1 16-Bit Phase Counting Mode.....16-129
16.3.6.2 Cascade Connection 32-Bit Phase Counting Mode.....16-140
16.3.7 Reset-Synchronized PWM Mode.....16-143
16.3.8 Complementary PWM Mode.....16-146
16.3.9 A/D Converter Start Request Delaying Function.....16-194
16.3.10 Synchronous Operation of MTU0 to MTU4, MTU6, and MTU7.....16-202
16.3.11 External Pulse Width Measurement.....16-206
16.3.12 Dead Time Compensation.....16-207

16.3.13 TCNTU, TCNTV, and TCNTW Capture at Crest and/or Trough in Complementary PWM Mode ..... 16-209
16.3.14 Noise Filter Function ..... 16-210
16.4 Interrupt Sources ..... 16-211
16.4.1 Interrupt Sources and Priorities ..... 16-211
16.4.2 DMAC Activation ..... 16-212
16.4.3 A/D Converter Activation ..... 16-213
16.5 Operation Timing ..... 16-215
16.5.1 Input/Output Timing ..... 16-215
16.5.2 Interrupt Signal Timing ..... 16-221
16.6 Usage Note ..... 16-224
16.6.1 Count Clock Restrictions ..... 16-224
16.6.2 Note on Cycle Setting ..... 16-224
16.6.3 Contention between TCNT Write and Clear Operations ..... 16-225
16.6.4 Contention between TCNT Write and Increment Operations ..... 16-225
16.6.5 Contention between TGR Write Operation and Compare Match ..... 16-226
16.6.6 Contention between Buffer Register Write Operation and Compare Match ..... 16-226
16.6.7 Contention between Buffer Register Write and TCNT Clear Operations ..... 16-227
16.6.8 Contention between TGR Read Operation and Input Capture ..... 16-227
16.6.9 Contention between TGR Write Operation and Input Capture ..... 16-228
16.6.10 Contention between Buffer Register Write Operation and Input Capture ..... 16-229
16.6.11 Contention between MTU2.TCNT Write Operation and Overflow/Underflow in Cascaded Operation ..... 16-230
16.6.12 Counter Value When Count Operation is Stopped in Complementary PWM Mode ..... 16-231
16.6.13 Buffer Operation Setting in Complementary PWM Mode ..... 16-231
16.6.14 Buffer Operation and Compare Match in Reset-Synchronized PWM Mode ..... 16-232
16.6.15 Overflow in Reset-Synchronized PWM Mode ..... 16-233
16.6.16 Contention between Overflow/Underflow and Counter Clearing ..... 16-234
16.6.17 Contention between TCNT Write Operation and Overflow/Underflow ..... 16-235
16.6.18 Note on Transition from Normal Mode or PWM Mode 1 to Reset-Synchronized PWM Mode ..... 16-235
16.6.19 Output Level in Complementary PWM Mode and Reset-Synchronized PWM Mode ..... 116-235
16.6.20 Simultaneous Input Capture in MTU1.TCNT and MTU2.TCNT in Cascade Connection ..... 16-236
16.6.21 Interrupt-Skipping Function 2 ..... 16-236
16.6.22 Notes When Complementary PWM Mode Output Protection Function is Not Used ..... 116-236
16.6.23 Notes Regarding Timer Counter (MTU5.TCNT) and Timer General Register (MTU5.TGR) ..... 16-237
16.6.24 Notes to Prevent Malfunctions in Synchronous Clearing for Complementary PWM Mode ..... 16-238
16.6.25 Continuous Output of Interrupt Signal in Response to a Compare Match ..... 16-240
16.6.26 Notes on A/D Converter Start Request Delaying Function in Complementary PWM Mode ..... 16-241
16.7 MTU Output Pin Initialization ..... 16-243
16.7.1 Operating Modes ..... 16-243
16.7.2 Operation in Case of Re-Setting Due to Error during Operation ..... 16-243
16.7.3 Overview of Initialization Procedures and Mode Transitions in Case of Error during Operation ..... 16-244
17. Port Output Enable 3 (POE3) ..... 17-1
17.1 Overview ..... 17-1

17.2 Register Descriptions ..... 17-4
17.2.1 Input Level Control/Status Register 1 (ICSR1) ..... 17-5
17.2.2 Input Level Control/Status Register 2 (ICSR2) ..... 17-6
17.2.3 Input Level Control/Status Register 3 (ICSR3) ..... 17-7
17.2.4 Input Level Control/Status Register 4 (ICSR4) ..... 17-9
17.2.5 Output Level Control/Status Register 1 (OCSR1) ..... 17-11
17.2.6 Output Level Control/Status Register 2 (OCSR2) ..... 17-12
17.2.7 Software Port Output Enable Register (SPOER) ..... 17-13
17.2.8 Port Output Enable Control Register 1 (POECR1) ..... 17-15
17.2.9 Port Output Enable Control Register 2 (POECR2) ..... 17-16
17.2.10 Port Output Enable Control Register 4 (POECR4) ..... 17-18
17.2.11 Port Output Enable Control Register 5 (POECR5) ..... 17-20
17.3 Operation ..... 17-21
17.3.1 MTU Pin Selection ..... 17-24
17.3.2 Input Level Detection Operation ..... 17-26
17.3.3 Output Level Compare Operation ..... 17-28
17.3.4 High-Impedance Control Using Registers ..... 17-28
17.3.5 Additional Functions for Controlling High-Impedance States ..... 17-28
17.3.6 Release from High-Impedance State ..... 17-29
17.4 POE3 Setting Procedure ..... 17-30
17.5 Interrupts ..... 17-31
17.6 Usage Notes ..... 17-32
17.6.1 High-Impedance Control when MTU6 and MTU7 are not Used ..... 17-32
20. General Timer (GTM) ..... 20-1
20.1 Functional Overview ..... 20-1
20.1.1 Features of GTM ..... 20-1
20.2 Registers ..... 20-3
20.2.1 Registers Overview ..... 20-3
20.2.2 Details of OSTM Registers ..... 20-4
20.2.2.1 OSTMnCMP — OSTM Compare Register ..... 20-4
20.2.2.2 OSTMnCNT — OSTM Counter Register ..... 20-5
20.2.2.3 OSTMnTE — OSTM Count Enable Status Register ..... 20-6
20.2.2.4 OSTMnTS — OSTM Count Start Trigger Register ..... 20-7
20.2.2.5 OSTMnTT — OSTM Count Stop Trigger Register ..... 20-7
20.2.2.6 OSTMnCTL — OSTM Control Register ..... 20-8
20.3 Functional Description ..... 20-9
20.3.1 Block Diagram ..... 20-9
20.3.2 Count Clock ..... 20-10
20.3.3 Generation of Interrupt Request ..... 20-10
20.3.4 Starting and Stopping the Timer ..... 20-11
20.3.5 Interval Timer Mode ..... 20-12
20.3.5.1 Basic Operation in Interval Timer Mode ..... 20-12
20.3.5.2 Operation when OSTMnCMP = H'0000 0000 ..... 20-14
20.3.6 Free-Running Comparison Mode ..... 20-15
20.3.6.1 Basic Operation in Free-Running Comparison Mode ..... 20-15
20.3.6.2 Operation when OSTMnCMP = H'0000 0000 ..... 20-17
21. Watchdog Timer (WDT) ..... 21-1
21.1 Features ..... 21-2
21.1.1 Normal Watchdog Timer Function ..... 21-2
21.1.2 Reset Request Function due to CPU Parity Error ..... 21-2
21.1.3 Internal Block Diagram ..... 21-3

21.1.4 Input/Output Pin ..... 21-4
21.2 Register Configuration ..... 21-5
21.3 Register Descriptions ..... 21-6
21.3.1 WDT Control Register_n (WDTCNT_n) (n = 0) ..... 21-6
21.3.2 WDT Period Setting Register_n (WDTSET_n) (n = 0) ..... 21-7
21.3.3 WDT Elapsed Time Register_n (WDTTIM_n (n = 0)) ..... 21-8
21.3.4 WDT Interrupt Control Register_n (WDTINT_n (n = 0)) ..... 21-9
21.3.5 Parity Error Control Register_n (PECR_n (n = 0)) ..... 21-10
21.3.6 Parity Error Forced Enable Register_n (PEEN_n (n = 0)) ..... 21-11
21.3.7 Parity Error Status Register_n (PESR_n (n = 0)) ..... 21-12
21.3.8 Parity Error Enable Register_n (PEER_n (n = 0)) ..... 21-13
21.3.9 Parity Error Polarity Setting Register_n (PEPO_n (n = 0)) ..... 21-14
21.3.10 Register Setting Order for Each WDT Channel ..... 21-15
21.4 Operation ..... 21-16
21.4.1 WDT_CORE Operaton Timing ..... 21-16
21.4.2 WDT Interrupt Control Register (WDTINT) Specification for Write Access Interval ..... 121-17
21.4.3 Watchdog Timer Operation Reflection Timing of Register Setting Value ..... 21-18
21.4.4 Timing of Reflecting WDT Counter Value to WDTTIM Register ..... 21-20
21.4.5 Conditions Under which WDTRSTB and PERROUT Operate ..... 21-21
21.5 Reset Control ..... 21-25
21.5.1 WDTRST Control of Each WDT Channels ..... 21-25
21.6 Usage Note ..... 21-26
22. Serial Communications Interface with FIFO (SCIFA) ..... 22-1
22.1 Overview ..... 22-1
22.2 Register Descriptions ..... 22-3
22.2.1 Receive Shift Register (RSR) ..... 22-4
22.2.2 Receive FIFO Data Register (FRDR) ..... 22-4
22.2.3 Transmit Shift Register (TSR) ..... 22-4
22.2.4 Transmit FIFO Data Register (FTDR) ..... 22-5
22.2.5 Serial Mode Register (SMR) ..... 22-6
22.2.6 Serial Control Register (SCR) ..... 22-8
22.2.7 Serial Status Register (FSR) ..... 22-10
22.2.8 Bit Rate Register (BRR) ..... 22-14
22.2.9 Modulation Duty Register (MDDR) ..... 22-19
22.2.10 FIFO Control Register (FCR) ..... 22-22
22.2.11 FIFO Data Count Register (FDR) ..... 22-24
22.2.12 Serial Port Register (SPTR) ..... 22-25
22.2.13 Line Status Register (LSR) ..... 22-28
22.2.14 Serial Extended Mode Register (SEMR) ..... 22-29
22.2.15 FIFO Trigger Control Register (FTCR) ..... 22-31
22.3 Operation ..... 22-32
22.3.1 Overview ..... 22-32
22.3.2 Operation in Asynchronous Mode ..... 22-34
22.3.3 Operation in Clock Synchronous Mode ..... 22-45
22.4 Bit Rate Modulation ..... 22-54
22.5 Interrupt Sources ..... 22-55
22.6 Serial Port Register (SPTR) and SCIFA-Related Pins ..... 22-56
22.7 Noise Cancellation ..... 22-58

22.8 Usage Notes ....22-59
22.8.1 FTDR Register Writing and TDFE Flag....22-59
22.8.2 FRDR Register Reading and RDF Flag ....22-59
22.8.3 Break Detection and Processing....22-59
22.8.4 Writing to the SPTR Register....22-59
22.8.5 Break Signal Transmission....22-60
22.8.6 Receive Data Sampling Timing and Receive Margin in Asynchronous Mode....22-61
22.8.7 Note on FER Flag and PER Flag in Serial Status Register (FSR)....22-62
22.8.8 Notes on External Clock Input in Clock Synchronous Mode....22-62
22.8.9 Module Standby Mode Setting....22-62
22.8.10 Notes on Operation for Reception when an Internal Clock is Selected in Clock Synchronous Mode ....22-62
23. Serial Communications Interface (SCIg) ....23-1
23.1 Overview....23-1
23.2 Register Descriptions ....23-4
23.2.1 Receive Shift Register (RSR)....23-5
23.2.2 Receive Data Register (RDR) ....23-5
23.2.3 Receive Data Register H, L, HL (RDRH, RDRL, RDRHL) ....23-6
23.2.4 Transmit Data Register (TDR) ....23-7
23.2.5 Transmit Data Register H, L, HL (TDRH, TDRL, TDRHL) ....23-8
23.2.6 Transmit Shift Register (TSR)....23-8
23.2.7 Serial Mode Register (SMR)....23-9
23.2.8 Serial Control Register (SCR)....23-13
23.2.9 Serial Status Register (SSR)....23-18
23.2.10 Smart Card Mode Register (SCMR) ....23-24
23.2.11 Bit Rate Register (BRR)....23-26
23.2.12 Modulation Duty Register (MDDR) ....23-30
23.2.13 Serial Extended Mode Register (SEMR) ....23-31
23.2.14 Noise Filter Setting Register (SNFR)....23-33
23.2.15 Extended function control register (SECR)....23-34
23.3 Operation in Asynchronous Mode ....23-35
23.3.1 Serial Data Transfer Format ....23-36
23.3.2 Receive Data Sampling Timing and Reception Margin in Asynchronous Mode ..... ....23-37
23.3.3 Clock ....23-38
23.3.4 Double-Speed Mode....23-38
23.3.5 CTS and RTS Functions....23-39
23.3.6 SCI Initialization (Asynchronous Mode) ....23-40
23.3.7 Serial Data Transmission (Asynchronous Mode)....23-41
23.3.8 Serial Data Reception (Asynchronous Mode)....23-45
23.4 Multi-Processor Communications Function ....23-49
23.4.1 Multi-Processor Serial Data Transmission ....23-50
23.4.2 Multi-Processor Serial Data Reception ....23-51
23.5 Operation in Clock Synchronous Mode....23-54
23.5.1 Clock ....23-55
23.5.2 CTS and RTS Functions....23-55
23.5.3 SCI Initialization (Clock Synchronous Mode) ....23-56
23.5.4 Serial Data Transmission (Clock Synchronous Mode)....23-57
23.5.5 Serial Data Reception (Clock Synchronous Mode)....23-62
23.5.6 Simultaneous Serial Data Transmission and Reception (Clock Synchronous Mode) ..... 23-65
23.6 Operation in Smart Card Interface Mode....23-67

23.6.1 Sample Connection ....23-67
23.6.2 Data Format (Except in Block Transfer Mode) ....23-68
23.6.3 Block Transfer Mode ....23-70
23.6.4 Receive Data Sampling Timing and Reception Margin ....23-71
23.6.5 SCI Initialization (Smart Card Interface Mode) ....23-72
23.6.6 Serial Data Transmission (Except in Block Transfer Mode) ....23-73
23.6.7 Serial Data Reception (Except in Block Transfer Mode) ....23-76
23.6.8 Clock Output Control ....23-78
23.7 Noise Cancellation Function ....23-79
23.8 Interrupt Sources ....23-80
23.8.1 Buffer Operations for TXI and RXI Interrupts ....23-80
23.8.2 Interrupts in Asynchronous Mode and Clock Synchronous Mode ....23-80
23.8.3 Interrupts in Smart Card Interface Mode ....23-81
23.9 Usage Notes ....23-82
23.9.1 Break Detection and Processing ....23-82
23.9.2 Mark State and Production of Breaks ....23-82
23.9.3 Receive Error Flags and Transmit Operations (Clock Synchronous Mode) ....23-82
23.9.4 Writing Data to TDR ....23-82
23.9.5 Restrictions on Clock Synchronous Transmission (Clock Synchronous Mode) ....23-83
23.9.6 Restrictions on Using DMAC ....23-85
23.9.7 Notes on Starting Transfer ....23-85
23.9.8 External Clock Input in Clock Synchronous Mode ....23-85
23.9.9 Note on Transmit Enable Bit (TE bit) ....23-85
23.10 IrDA Communications ....23-86
23.11 Description of the IrDA Register ....23-87
23.11.1 IrDA Control Register (IRCR) ....23-88
23.12 IrDA Operation ....23-89
23.12.1 Flow of Settings for IrDA Operation ....23-89
23.12.2 Transmission ....23-89
23.12.3 Reception ....23-90
23.12.4 Relationship between Bit Rate and High-Level Pulse Width ....23-90
23.13 Note on IrDA Usage ....23-91
23.13.1 Minimum Pulse Width in Reception ....23-91
23.13.2 Base Clock in Asynchronous Mode for the Serial Communications Interface ....23-91
24. Renesas Serial Peripheral Interface ....24-1
24.1.1 Features ....24-1
24.2 Input/Output Pins ....24-4
24.3 Register Descriptions ....24-5
24.3.1 Control Register (SPCR) ....24-6
24.3.2 Slave Select Polarity Register (SSLP) ....24-7
24.3.3 Pin Control Register (SPPCR) ....24-8
24.3.4 Status Register (SPSR) ....24-9
24.3.5 Data Register (SPDR) ....24-11
24.3.6 Sequence Control Register (SPSCR) ....24-12
24.3.7 Sequence Status Register (SPSSR) ....24-13
24.3.8 Bit Rate Register (SPBR) ....24-14
24.3.9 Data Control Register (SPDCR) ....24-15
24.3.10 Clock Delay Register (SPCKD) ....24-16
24.3.11 Slave Select Negation Delay Register (SSLND) ....24-17
24.3.12 Next-Access Delay Register (SPND) ....24-18
24.3.13 Command Register (SPCMD) ....24-19

24.3.14 Buffer Control Register (SPBFCR) ..... 24-21
24.3.15 Buffer Data Count Setting Register (SPBFDR) ..... 24-23
24.4 Operation ..... 24-24
24.4.1 Overview of Operations ..... 24-24
24.4.2 Pin Control ..... 24-25
24.4.3 System Configuration Example ..... 24-26
24.4.4 Transfer Format ..... 24-28
24.4.5 Data Format ..... 24-30
24.4.6 Error Detection ..... 24-36
24.4.7 Initialization ..... 24-39
24.4.8 SPI Operation ..... 24-40
24.4.9 Error Handling ..... 24-50
24.4.10 Loopback Mode ..... 24-51
24.4.11 Interrupt Sources ..... 24-51
25. SPI Multi I/O Bus Controller ..... 25-1
25.1 Features ..... 25-1
25.1.1 Serial Flash Memory Interface ..... 25-1
25.1.2 External Address Space Read Mode ..... 25-1
25.1.3 Manual Mode ..... 25-1
25.2 Block Diagram ..... 25-2
25.3 Input/Output Pins ..... 25-3
25.4 Register Descriptions ..... 25-4
25.4.1 Common Control Register (CMNCR) ..... 25-5
25.4.2 SSL Delay Register (SSLDR) ..... 25-7
25.4.3 Data Read Control Register (DRCR) ..... 25-9
25.4.4 Data Read Command Setting Register (DRCMR) ..... 25-11
25.4.5 Data Read Extended Address Setting Register (DREAR) ..... 25-12
25.4.6 Data Read Option Setting Register (DROPR) ..... 25-13
25.4.7 Data Read Enable Setting Register (DRENR) ..... 25-14
25.4.8 Manual Mode Control Register (SMCR) ..... 25-16
25.4.9 Manual Mode Command Setting Register (SMCMR) ..... 25-17
25.4.10 Manual Mode Address Setting Register (SMADR) ..... 25-18
25.4.11 Manual Mode Option Setting Register (SMOPR) ..... 25-19
25.4.12 Manual Mode Enable Setting Register (SMENR) ..... 25-20
25.4.13 Manual Mode Read Data Register 0 (SMRDR0) ..... 25-22
25.4.14 Manual Mode Write Data Register 0 (SMWDR0) ..... 25-23
25.4.15 Common Status Register (CMNSR) ..... 25-24
25.4.16 Data Read Dummy Cycle Setting Register (DRDMCR) ..... 25-25
25.4.17 Data Read DDR Enable Register (DRDRENR) ..... 25-26
25.4.18 Manual Mode Dummy Cycle Setting Register (SMDMCR) ..... 25-27
25.4.19 Manual Mode DDR Enable Register (SMDRENR) ..... 25-28
25.4.20 PHY Control Register (PHYCNT) ..... 25-29
25.4.21 PHY Offset Register 1 (PHYOFFSET1) ..... 25-31
25.4.22 PHY Offset Register 2 (PHYOFFSET2) ..... 25-33
25.4.23 PHY Adjustment Register 1 (PHYADJ1) ..... 25-34
25.4.24 PHY Adjustment Register 2 (PHYADJ2) ..... 25-35
25.5 Operation ..... 25-36
25.5.1 System Configuration ..... 25-36
25.5.2 Address Map ..... 25-37
25.5.3 32-bit Serial Flash Addresses ..... 25-38
25.5.4 Operating Modes ..... 25-39
25.5.5 External Address Space Read Mode ..... 25-39
25.5.6 Read Cache ..... 25-44

25.5.7 Manual mode....25-45
25.5.8 Command Sequence....25-49
25.5.9 Data Pin Control....25-53
25.5.10 QSPIn_SSL Pin Control....25-54
25.5.11 Flags....25-55
25.5.12 Write Buffer Operation....25-56
25.5.13 Supported Protocol for Serial Flash Memory....25-58
25.5.14 Timing Adjustment....25-59
25.5.15 Data Alignment....25-62
25.6 Usage Notes....25-63
25.6.1 Transfer to read data while the signal on the QSPIn_SSL pin is de-asserted....25-63
25.6.2 Transfer to read data while the signal on the QSPIn_SSL pin is asserted....25-63
25.6.3 Notes on Starting Transfer from the QSPIn_SSL Retained State in Manual Mode....25-63
26. I²C Bus Interface....26-1
26.1 Features....26-1
26.1.1 Channels....26-1
26.1.2 Register Base Addresses....26-1
26.1.3 External I/O Signals....26-2
26.2 Overview....26-3
26.2.1 Functional Overview....26-3
26.2.2 Block Diagram....26-5
26.3 Registers....26-7
26.3.1 RIICnCR1 — I²C Bus Control Register 1....26-8
26.3.2 RIICnCR2 — I²C Bus Control Register 2....26-11
26.3.3 RIICnMR1 — I²C Bus Mode Register 1....26-15
26.3.4 RIICnMR2 — I²C Bus Mode Register 2....26-17
26.3.5 RIICnMR3 — I²C Bus Mode Register 3....26-19
26.3.6 RIICnFER — I²C Bus Function Enable Register....26-22
26.3.7 RIICnSER — I²C Bus Status Enable Register....26-25
26.3.8 RIICnIER — I²C Bus Interrupt Enable Register....26-27
26.3.9 RIICnSR1 — I²C Bus Status Register 1....26-29
26.3.10 RIICnSR2 — I²C Bus Status Register 2....26-32
26.3.11 RIICnSARy — I²C Slave Address Register y (y = 0 to 2)....26-37
26.3.12 RIICnBRL — I²C Bus Bit Rate Low-Level Register....26-39
26.3.13 RIICnBRH — I²C Bus Bit Rate High-Level Register....26-40
26.3.14 RIICnDRT — I²C Bus Transmit Data Register....26-43
26.3.15 RIICnDRR — I²C Bus Receive Data Register....26-44
26.3.16 RIICnDRS — I²C Bus Shift Register....26-45
26.4 Interrupt Sources....26-46
26.5 Operation....26-47
26.5.1 Communication Data Format....26-47
26.5.2 Initial Settings....26-49
26.5.3 Master Transmit Operation....26-50
26.5.4 Master Receive Operation....26-54
26.5.5 Slave Transmit Operation....26-60
26.5.6 Slave Receive Operation....26-63
26.6 SCL Synchronization Circuit....26-66
26.7 Facility for Delaying SDA Output....26-67
26.8 Digital Noise-Filter Circuits....26-68
26.9 Address Match Detection....26-69

26.9.1 Slave-Address Match Detection ..... 26-69
26.9.2 Detection of the General Call Address ..... 26-72
26.9.3 Device-ID Address Detection ..... 26-73
26.9.4 Host Address Detection ..... 26-75
26.10 Automatically Low-Hold Function for SCL ..... 26-76
26.10.1 Function to Prevent Wrong Transmission of Transmit Data ..... 26-76
26.10.2 NACK Reception Transfer Suspension Function ..... 26-77
26.10.3 Function to Prevent Failure to Receive Data ..... 26-78
26.11 Arbitration-Lost Detection Functions ..... 26-80
26.11.1 Master Arbitration-Lost Detection (MALE Bit) ..... 26-80
26.11.2 Function to Detect Loss of Arbitration during NACK Transmission (NALE Bit) ..... 26-83
26.11.3 Slave Arbitration-Lost Detection (SALE Bit) ..... 26-84
26.12 Start Condition/Restart Condition/Stop Condition Issuing Function ..... 26-85
26.12.1 Issuing a Start Condition ..... 26-85
26.12.2 Issuing a Restart Condition ..... 26-86
26.12.3 Issuing a Stop Condition ..... 26-87
26.13 Bus Hanging ..... 26-88
26.13.1 Timeout Function ..... 26-88
26.13.2 Extra SCL Clock Cycle Output Function ..... 26-90
26.13.3 RIIC Reset and Internal Reset ..... 26-91
26.14 SMBus Operation ..... 26-92
26.14.1 SMBus Timeout Measurement ..... 26-92
26.14.2 SMBus Host Notification Protocol/Notify ARP Master ..... 26-93
26.15 Reset Function of RIIC ..... 26-94
27. Serial Sound Interface (SSIF-2) ..... 27-1
27.1 Overview ..... 27-1
27.1.1 Features ..... 27-1
27.1.2 Block Diagram ..... 27-3
27.2 Input/Output Pins ..... 27-5
27.3 List of Registers ..... 27-6
27.4 Function Details ..... 27-7
27.4.1 Register Descriptions ..... 27-7
27.4.1.1 Control Register (SSICR) ..... 27-7
27.4.1.2 Status Register (SSISR) ..... 27-20
27.4.1.3 FIFO Control Register (SSIFCR) ..... 27-33
27.4.1.4 IFO Status Register (SSIFSR) ..... 27-42
27.4.1.5 Transmit FIFO Data Register (SSIFTDR) ..... 27-46
27.4.1.6 Receive FIFO Data Register (SSIFRDR) ..... 27-49
27.4.1.7 Audio Format Register (SSIOFR) ..... 27-51
27.4.1.8 Status Control Register (SSISCR) ..... 27-55
27.4.2 Communication Formats ..... 27-56
27.4.2.1 $I^2S$ Format ..... 27-57
27.4.2.2 Monaural Format ..... 27-58
27.4.2.3 TDM Format ..... 27-60
27.4.3 Communication Modes ..... 27-61
27.4.3.1 Slave-mode Communication ..... 27-62
27.4.3.2 Master-mode Communication ..... 27-62
27.4.3.3 Transmission ..... 27-62
27.4.3.4 Reception ..... 27-62
27.4.3.5 Transmission and Reception ..... 27-62
27.5 Operation ..... 27-63

27.5.1 Operational State ..... 27-63
27.5.1.1 Idle State ..... 27-63
27.5.1.2 Communication States ..... 27-66
27.5.2 Communication Operation ..... 27-70
27.5.2.1 Start Communication ..... 27-71
27.5.2.2 Transmission ..... 27-72
27.5.2.3 Reception ..... 27-74
27.5.2.4 Transmission and Reception ..... 27-75
27.5.2.5 Halt Communication ..... 27-76
27.5.2.6 Error Handling ..... 27-77
27.5.2.7 Resume Communication ..... 27-79
27.5.3 Interrupt Sources ..... 27-80
27.5.3.1 INT_ssif_int_req Interrupt ..... 27-80
27.5.3.2 INT_ssif_dma_tx Interrupt [full-duplex communication] ..... 27-82
27.5.3.3 INT_ssif_dma_rx Interrupt [full-duplex communication] ..... 27-82
27.5.3.4 INT_ssif_dma_rt Interrupt [half-duplex communication] ..... 27-83
27.5.4 Software Resets ..... 27-84
27.5.4.1 Software Reset Procedure ..... 27-84
27.6 Notes ..... 27-86
27.6.1 Notes ..... 27-86
27.6.1.1 Attention by communication using DMA ..... 27-86
27.6.1.2 Notes for Slave-mode Communication ..... 27-86
27.6.1.3 Notes for Master-mode Communication ..... 27-87
27.6.1.4 Notes for Communication Flow ..... 27-87
27.6.1.5 Write Access Restriction ..... 27-89
32. USB2.0 ..... 32-1
32.1 Features ..... 32-1
32.1.1 Ch0: USB2.0 (Host/Function) interface ..... 32-1
32.1.2 Block Diagram ..... 32-2
32.1.3 External signal pins ..... 32-3
32.1.4 Power and GND pins ..... 32-3
32.1.5 Interrupt ..... 32-4
32.2 Register Configuration ..... 32-5
32.2.1 Host controller Register ..... 32-5
32.2.2 Function controller Register ..... 32-5
32.2.3 USBPHY Control Register ..... 32-5
32.3 Register Descriptions ..... 32-6
32.3.1 Reset register (RESET) ..... 32-6
32.3.2 Clock control register (UCLKCTL) ..... 32-7
32.3.3 Direct power down setting register (UDIRPD) ..... 32-8
32.3.4 VBUSEN control register (VBENCTL) ..... 32-9
32.3.5 Connection control register (CON_CTRL) ..... 32-9
32.3.6 Clock status register (CLK_STAT) ..... 32-11
32.4 Operation ..... 32-12
32.4.1 USB/PHY Initialization ..... 32-12
32.4.2 Handling of permanently unused pin ..... 32-13
32.4.2.1 USB/PHY related pins ..... 32-13
32.4.2.2 Handling of USB controller related pins ..... 32-14
32.4.3 Direct power down (DIRPD) ..... 32-14
32.4.3.1 Enter to Direct power down mode ..... 32-14
32.4.3.2 Resume from Direct power down mode ..... 32-15
32.5 Usage Notes ..... 32-16
32.5.1 Precautions in power down mode ..... 32-16
32A. USB 2.0 Host Module ..... 32A-1

32A.1 Overview ..... 32A-1
32A.1.1 Overview ..... 32A-1
32A.1.2 Features ..... 32A-1
32A.1.2.1 EHCI v1.1 functions ..... 32A-2
32A.1.2.2 Link Power Management (LPM) function ..... 32A-3
32A.1.2.3 OTG function ..... 32A-4
32A.1.2.4 Battery-charging function ..... 32A-4
32A.1.2.5 Suspend extension function ..... 32A-4
32A.1.3 Support of USB-Related Specifications ..... 32A-5
32A.2 Register Descriptions ..... 32A-6
32A.2.1 Register Attributes ..... 32A-6
32A.2.2 Base Address ..... 32A-6
32A.2.3 Register Overview ..... 32A-7
32A.2.4 Description of Registers ..... 32A-9
32A.2.4.1 OHCI Operational Register ..... 32A-9
32A.2.4.2 EHCI Controller Capability Register ..... 32A-41
32A.2.4.3 HCI Operational Register ..... 32A-46
32A.2.4.4 AHB Bridge Register ..... 32A-63
32A.2.4.5 UCOM Register ..... 32A-76
32A.3 Clock Signals ..... 32A-83
32A.3.1 Clock Gating Specifications ..... 32A-83
32A.3.1.1 Overview of clock gating ..... 32A-83
32A.3.1.2 Specifications of NONUSE_CLK_MSK operation ..... 32A-83
32A.3.1.3 Specifications of PERI_CLK_MSK and HOST_CLK_MSK operations ..... 32A-84
32A.4 Interrupt Sources ..... 32A-85
32A.4.1 Interrupt Signals ..... 32A-85
32A.4.2 Interrupt Sources and Control ..... 32A-86
32A.4.2.1 U2H_INT assertion source and control ..... 32A-86
32A.4.2.2 U2H_OHCI_INT assertion source and control ..... 32A-87
32A.4.2.3 U2H_EHCI_INT assertion source and control ..... 32A-88
32A.4.2.4 U2H_WAKEON_INT assertion source and control ..... 32A-89
32A.4.2.5 U2H_OBINT assertion source and control ..... 32A-90
32A.4.3 Timing of De-asserting Interrupt Signals ..... 32A-91
32A.5 Power-Saving Function ..... 32A-92
32A.5.1 Controlling the SUSPENDM and SLEEPM Pins of the USBPHY ..... 32A-92
32A.5.2 Controlling the Clock-Gating Function ..... 32A-92
32A.6 Battery Charging ..... 32A-93
32A.6.1 Support of charging port ..... 32A-93
32A.6.2 Support of portable device ..... 32A-93
32A.7 Bus Master ..... 32A-94
32A.7.1 Functional specifications of the bus master ..... 32A-94
32A.7.1.1 Supported bus master functions ..... 32A-94
32A.7.1.2 Issuing requests for different types of bus transfer ..... 32A-94
32A.7.1.3 Supported responses ..... 32A-94
32A.7.1.4 Protection control information ..... 32A-94
32A.7.1.5 Maximum burst length ..... 32A-95
32A.7.1.6 Boundary of transfer data ..... 32A-95
32A.7.1.7 Start address of fixed-length INCR burst transfer ..... 32A-96
32A.8 Overcurrent Control and VBUS Control ..... 32A-97
32A.8.1 OVRCUR/VBUSEN pin ..... 32A-97
32A.8.2 Overcurrent detection timer setting ..... 32A-97
32A.8.3 Port Power (VBUS) control specifications ..... 32A-98
32A.8.4 Timing Chart for Overcurrent Detection and Recovery ..... 32A-99

32A.9 Procedure for Setting this Module ..... 32A-100
32A.9.1 Host/Peripheral Common Setting Sequence ..... 32A-100
32A.9.2 Initialization Sequence ..... 32A-101
32A.9.3 Flow of Error Handling ..... 32A-102
32B. USB 2.0 Function Module ..... 32B-1
32B.1 Overview ..... 32B-1
32B.1.1 Overview ..... 32B-1
32B.1.2 Features ..... 32B-1
32B.1.2.1 Peripheral controller supporting high-speed USB ..... 32B-1
32B.1.2.2 Support of all types of USB transfer ..... 32B-1
32B.1.2.3 Bus interface ..... 32B-1
32B.1.2.4 Pipe configuration ..... 32B-2
32B.1.2.5 Features of peripheral functions ..... 32B-2
32B.1.2.6 Features of DMA transfer ..... 32B-2
32B.1.2.7 Other functions ..... 32B-3
32B.1.3 Overview of Functions ..... 32B-4
32B.1.3.1 Automatic recognition of USB transfer speed ..... 32B-4
32B.1.3.2 USB event ..... 32B-4
32B.1.3.3 USB data transfer ..... 32B-4
32B.1.3.4 SOF pulse output function ..... 32B-4
32B.1.4 Restriction matter and Notes ..... 32B-5
32B.1.4.1 Restriction matter ..... 32B-5
32B.1.4.2 Notes ..... 32B-5
32B.2 Registers ..... 32B-6
32B.2.1 Base Address ..... 32B-7
32B.2.2 List of Registers ..... 32B-7
32B.2.3 System Configuration Control Registers ..... 32B-10
32B.2.3.1 System Configuration Control Register 0 [SYSCFG0] <Address: H'000> ..... 32B-10
32B.2.3.2 System Configuration Control Register 1 [SYSCFG1] <Address: H'002> ..... 32B-13
32B.2.4 System Configuration Status ..... 32B-14
32B.2.4.1 System Configuration Status Register (SYSSTS0) <Address: H'004> ..... 32B-14
32B.2.5 USB Signal Control Registers ..... 32B-15
32B.2.5.1 Device State Control Register 0 [DVSTCTR0] <Address: H'008> ..... 32B-15
32B.2.6 Test Mode Register ..... 32B-17
32B.2.6.1 USB Test Mode Register [TESTMODE] <Address: H'00C> ..... 32B-17
32B.2.7 FIFO Port Registers ..... 32B-18
32B.2.7.1 CFIFO Port Register [CFIFO] <Address: H'014> ..... 32B-18
32B.2.7.2 CFIFO Port Register [CFIFO] <Address: H'016> ..... 32B-18
32B.2.7.3 CFIFO Port Select Register [CFIFOSEL] <Address: H'020> ..... 32B-20
32B.2.7.4 D0FIFO Port Select Register [D0FIFOSEL] <Address:H'028> D1FIFO Port Select Register [D1FIFOSEL] <Address:H'02C> ..... 32B-23
32B.2.7.5 CFIFO Port Control Register [CFIFOCTR] <Address: H'022> D0FIFO Port Control Register [D0FIFOCTR] <Address: H'02A> D1FIFO Port Control Register [D1FIFOCTR] <Address: H'02E> ..... 132B-25
32B.2.8 Interrupt Enable Registers (INTENBx, BRDYENB, NRDYENB, BEMPENB) ..... 32B-27
32B.2.8.1 Interrupt Enable Register 0 [INTENB0] <Address: H'030> ..... 32B-27
32B.2.8.2 BRDY Interrupt Enable Register [BRDYENB] <Address: H'036> ..... 32B-28
32B.2.8.3 NRDY Interrupt Enable Register [NRDYENB] <Address: H'038> ..... 32B-29
32B.2.8.4 BEMP Interrupt Enable Register [BEMPENB] <Address: H'03A> ..... 32B-30
32B.2.9 SOF Control Register ..... 32B-31
32B.2.9.1 SOF Pin Configuration Register [SOFCFG] <Address: H'03C> ..... 32B-31
32B.2.10 Interrupt Status ..... 32B-32
32B.2.10.1 Interrupt Status Register 0 [INTSTS0] <Address: H'040> ..... 32B-32
32B.2.10.2 BRDY Interrupt Status Register [BRDYSTS] <Address: H'046> ..... 32B-35
32B.2.10.3 NRDY Interrupt Status Register [NRDYSTS] <Address: H'048> ..... 32B-38

32B.2.10.4 BEMP Interrupt Status Register [BEMPSTS] <Address: H'04A> ..... 32B-40
32B.2.11 Frame Number Registers (FRMNUM, UFRMNUM) ..... 32B-41
32B.2.11.1 Frame Number Register [FRMNUM] <Address: H'04C> ..... 32B-41
32B.2.11.2 Micro Frame Number Register [UFRMNUM] <Address: H'04E> ..... 32B-43
32B.2.12 USB Address ..... 32B-44
32B.2.12.1 USB Address Register [USBADDR] <Address: H'050> ..... 32B-44
32B.2.13 USB Request Registers ..... 32B-45
32B.2.13.1 USB Request Type Register [USBREQ] <Address: H'054> ..... 32B-45
32B.2.13.2 USB Request Value Register [USBVAL] <Address: H'056> ..... 32B-46
32B.2.13.3 USB Request Index Register [USBINDX] <Address: H'058> ..... 32B-46
32B.2.13.4 USB Request Length Register [USBLENG] <Address: H'05A> ..... 32B-47
32B.2.14 DCP Configuration ..... 32B-48
32B.2.14.1 DCP Configuration Register [DCPCFG] <Address: H'05C> ..... 32B-48
32B.2.14.2 DCP Max. Packet Size Register [DCPMAXP] <Address: H'05E> ..... 32B-49
32B.2.14.3 DCP Control Register [DCPCTR] <Address: H'060> ..... 32B-50
32B.2.15 Pipe Configuration Registers (PIPESEL, PIPECFG, PIPEBUF, PIPEMAXP, PIPEPERI) ..... 32B-53
32B.2.15.1 Pipe Window Select Register [PIPESEL] <Address: H'064> ..... 32B-53
32B.2.15.2 Pipe Configuration Register [PIPECFG] <Address: H'068> ..... 32B-54
32B.2.15.3 Pipe Buffer Setting Register [PIPEBUF] <Address: H'06A> ..... 32B-60
32B.2.15.4 Pipe Maximum Packet Size Register [PIPEMAXP] <Address: H'06C> ..... 32B-62
32B.2.15.5 Pipe Cycle Control Register [PIPEPERI] <Address: H'06E> ..... 32B-63
32B.2.16 Pipe Control Registers (PIPExCTR) ..... 32B-66
32B.2.16.1 PIPE1 Control Register [PIPE1CTR] <Address: H'070> PIPE2 Control Register [PIPE2CTR] <Address: H'072> PIPE3 Control Register [PIPE3CTR] <Address: H'074> PIPE4 Control Register [PIPE4CTR] <Address: H'076> PIPE5 Control Register [PIPE5CTR] <Address: H'078> 32B-66
32B.2.16.2 PIPE6 Control Register [PIPE6CTR] <Address: H'07A> PIPE7 Control Register [PIPE7CTR] <Address: H'07C> PIPE8 Control Register [PIPE8CTR] <Address: H'07E> PIPE9 Control Register [PIPE9CTR] <Address: H'080> ..... 32B-72
32B.2.17 Transaction Counters (PIPExTRE) ..... 32B-74
32B.2.17.1 PIPE1 Transaction Counter Enable Register [PIPE1TRE] <Address: H'090> PIPE2 Transaction Counter Enable Register [PIPE2TRE] <Address: H'094> PIPE3 Transaction Counter Enable Register [PIPE3TRE] <Address: H'098> PIPE4 Transaction Counter Enable Register [PIPE4TRE] <Address: H'09C> PIPE5 Transaction Counter Enable Register [PIPE5TRE] <Address: H'0A0> . 32B-74
32B.2.17.2 PIPE1 Transaction Counter Register [PIPE1TRN] <Address: H'092> PIPE2 Transaction Counter Register [PIPE2TRN] <Address: H'096> PIPE3 Transaction Counter Register [PIPE3TRN] <Address: H'09A> PIPE4 Transaction Counter Register [PIPE4TRN] <Address: H'09E> PIPE5 Transaction Counter Register [PIPE5TRN] <Address: H'0A2> ..... 32B-76
32B.2.18 Low Power Control Register ..... 32B-78
32B.2.18.1 Low Power Control Register [LPCTRL] <Address: H'100> ..... 32B-78
32B.2.19 Low Power Status Register ..... 32B-79
32B.2.19.1 Low Power Status Register [LPSTS] <Address: H'102> ..... 32B-79
32B.2.20 PHY Function Control Register ..... 32B-80
32B.2.20.1 PHY Function Control Register [PHYFUNCTR] <Address: H'104> ..... 32B-80
32B.2.21 PHY_OTG Control Register (PHYOTGCTR) ..... 32B-81
32B.2.21.1 PHY_OTG Control Register [PHYOTGCTR] <Address: H'10A> ..... 32B-81
32B.2.22 Peripheral L1 Control Register 1 (PL1CTRL) ..... 32B-82
32B.2.22.1 Peripheral L1 Control Register 1 [PL1CTRL1] <Address: H'144> ..... 32B-82
32B.2.23 Peripheral L1 Control Register 2 ..... 32B-85
32B.2.23.1 Peripheral L1 Control Register 2 [PL1CTRL2] <Address: H'146> ..... 32B-85
32B.3 Next Register Set ..... 32B-86
32B.3.1 Next Source Address Register n ..... 32B-86
32B.3.1.1 Next0 Source Address Register ch0 [N0SA_0] <Address: H'400> Next1 Source Address Register ch0 [N1SA_0] <Address: H'40C> Next0 Source Address Register ch1 [N0SA_1] <Address: H'440> Next1 Source Address Register ch1 [N1SA_1] <Address: H'44C> ..... 132B-86
32B.3.2 Next Destination Address Register n ..... 32B-87
32B.3.2.1 Next0 Destination Address Register ch0 [N0DA_0] <Address: H'404> Next1 Destination Address Register ch0 [N1DA_0] <Address: H'410> Next0 Destination Address Register ch1 [N0DA_1] <Address: H'444> Next1 Destination Address Register ch1 [N1DA_1] <Address: H'450> ..... 32B-87

32B.3.3 Next Transaction Byte Register n ..... 32B-88
32B.3.3.1 Next0 Transaction Byte Register ch0 [N0TB_0] <Address: H'408> Next1 Transaction Byte Register ch0 [N1TB_0] <Address: H'414> Next0 Transaction Byte Register ch1 [N0TB_1] <Address: H'448> Next1 Transaction Byte Register ch1 [N1TB_1] <Address: H'454> ..... 32B-88
32B.4 Current Register Set ..... 32B-89
32B.4.1 Current Source Address Register ..... 32B-89
32B.4.1.1 Current Source Address Register ch0 [CRSA_0] <Address: H'418> Current Source Address Register ch1 [CRSA_1] <Address: H'458> ..... 32B-89
32B.4.2 Current Destination Address Register ..... 32B-90
32B.4.2.1 Current Destination Address Register ch0 [CRDA_0] <Address: H'41C> Current Destination Address Register ch1 [CRDA_1] <Address: H'45C> ..... 32B-90
32B.4.3 Current Transaction Byte Register ..... 32B-91
32B.4.3.1 Current Transaction Byte Register ch0 [CRTB_0] <Address: H'420> Current Transaction Byte Register ch1 [CRTB_1] <Address: H'460> ..... 32B-91
32B.5 Channel Register Set ..... 32B-92
32B.5.1 Channel Status Register n ..... 32B-92
32B.5.1.1 Channel Status Register ch0 [CHSTAT_0] <Address: H'424> Channel Status Register ch1 [CHSTAT_1] <Address: H'464> ..... 32B-92
32B.5.2 Channel Control Register n ..... 32B-97
32B.5.2.1 Channel Control Register ch0 [CHCTRL_0] <Address: H'428> Channel Control Register ch1 [CHCTRL_1] <Address: H'468> ..... 32B-97
32B.5.3 Channel Configuration Register ..... 32B-100
32B.5.3.1 Channel Configuration Register ch0 [CHCFG_0] <Address: H'42C> Channel Configuration Register ch1 [CHCFG_1] <Address: H'46C> ..... 32B-100
32B.5.4 Channel Interval Register n ..... 32B-104
32B.5.4.1 Channel Interval Register ch0 [CHITVL_0] <Address: H'430> Channel Interval Register ch1 [CHITVL_1] <Address: H'470> ..... 32B-104
32B.5.5 Channel Extension Register n ..... 32B-105
32B.5.5.1 Channel Extension Register ch0 [CHEXT_0] <Address: H'434> Channel Extension Register ch1 [CHEXT_1] <Address: H'474> ..... 32B-105
32B.6 Link Register Set ..... 32B-106
32B.6.1 Next Link Address Register n (NXLA_n) ..... 32B-106
32B.6.1.1 Next Link Address Register ch0 [NXLA_0] <Address: H'438> Next Link Address Register ch1 [NXLA_1] <Address: H'478> ..... 32B-106
32B.6.2 Current Link Address Register n (CRLA_n) ..... 32B-107
32B.6.2.1 Current Link Address Register ch0 [CRLA_0] <Address: H'43C> Current Link Address Register ch1 [CRLA_1] <Address: H'47C> ..... 32B-107
32B.7 Skip Register Set ..... 32B-108
32B.7.1 Source Continuous Register n ..... 32B-108
32B.7.1.1 Source Continuous Register ch0 [SCNT_0] <Address: H'600> Source Continuous Register ch1 [SCNT_1] <Address: H'620> ..... 32B-108
32B.7.2 Source Skip Register n ..... 32B-109
32B.7.2.1 Source Skip Register ch0 [SSKP_0] <Address: H'604> Source Skip Register ch1 [SSKP_1] <Address: H'624> ..... 32B-109
32B.7.3 Destination Continuous Register n ..... 32B-111
32B.7.3.1 Destination Continuous Register ch0 [DCNT_0] <Address: H'608> Destination Continuous Register ch1 [DCNT_1] <Address: H'628> ..... 32B-111
32B.7.4 Destination Skip Register n ..... 32B-112
32B.7.4.1 Destination Skip Register ch0 [DSKP_0] <Address: H'60C> Destination Skip Register ch1 [DSKP_1] <Address: H'62C> ..... 32B-112
32B.8 DMA Register Set ..... 32B-114
32B.8.1 DMA Control Register ..... 32B-114
32B.8.1.1 DMA Control Register [DCTRL] <Address: H'700> ..... 32B-114
32B.8.2 Descriptor Interval Register n ..... 32B-115
32B.8.2.1 Descriptor Interval Register [DSCITVL] <Address: H'704> ..... 32B-115
32B.8.3 DMA Control Register ..... 32B-115
32B.8.3.1 DMA Status EN Register [DSTAT_EN] <Address: H'710> ..... 32B-115

32B.8.4 DMA Status ER Register ... 32B-116
32B.8.4.1 DMA Status ER Register [DSTAT_ER] <Address: H'714> ... 32B-116
32B.8.5 DMA Status END Register ... 32B-116
32B.8.5.1 DMA Status END Register [DSTAT_END] <Address: H'718> ... 32B-116
32B.8.6 DMA Status TC Register ... 32B-117
32B.8.6.1 DMA Status TC Register [DSTAT_TC] <Address: H'71C> ... 32B-117
32B.8.7 DMA Status SUS Register ... 32B-117
32B.8.7.1 DMA Status SUS Register [DSTAT_SUS] <Address: H'720> ... 32B-117
32B.9 Functions ... 32B-118
32B.9.1 System Control and Oscillation Control ... 32B-118
32B.9.1.1 USB data bus resistor control ... 32B-118
32B.9.2 Interrupt Function ... 32B-119
32B.9.2.1 Overview of Interrupt Function (other than DMA Master) ... 32B-119
32B.9.2.2 Device State Transition Interrupts ... 32B-121
32B.9.2.3 Control Transfer Stage Transition Interrupts ... 32B-122
32B.9.2.4 Interrupts Relating to DMA Master ... 32B-124
32B.9.3 Pipe Control ... 32B-125
32B.9.3.1 Maximum packet size setting ... 32B-126
32B.9.3.2 Response PID ... 32B-126
32B.9.3.3 Pipe control register switching procedure ... 32B-127
32B.9.3.4 Data PID sequence bit ... 32B-128
32B.9.4 FIFO Buffer ... 32B-129
32B.9.4.1 FIFO buffer allocation ... 32B-129
32B.9.4.2 Clearing FIFO buffers ... 32B-130
32B.9.5 FIFO Port Functions ... 32B-131
32B.9.5.1 FIFO port selection ... 32B-132
32B.9.5.2 DxFIFO automatic clear mode (DxFIFO port read direction) ... 32B-133
32B.9.5.3 BRDY interrupt timing selection function ... 32B-133
32B.9.6 Control Transfer (DCP) ... 32B-134
32B.9.6.1 Control transfer ... 32B-134
32B.9.7 Bulk Transfer (Pipes 1 to 5) ... 32B-136
32B.9.7.1 NYET handshake control ... 32B-136
32B.9.8 Interrupt Transfer (Pipes 6 to 9) ... 32B-137
32B.9.9 Isochronous Transfer (Pipes 1 and 2) ... 32B-138
32B.9.9.1 Isochronous transfer error detection ... 32B-138
32B.9.9.2 DATA-PID ... 32B-139
32B.9.9.3 Interval counter ... 32B-140
32B.9.9.4 Send data setup for isochronous transfer ... 32B-141
32B.9.9.5 Transmit buffer flush for isochronous transfer ... 32B-142
32B.9.10 SOF Interpolation Function ... 32B-144
32B.9.11 Link Power Management Processing ... 32B-145
32B.9.11.1 Descriptor ... 32B-145
32B.9.11.2 Basic processing ... 32B-146
32B.9.11.3 HIRD value negotiation ... 32B-146
32B.9.12 DMA Mode ... 32B-147
32B.9.12.1 Register mode/link mode ... 32B-147
32B.9.12.2 Write only mode ... 32B-164
32B.9.13 DMA Transfer ... 32B-165
32B.9.13.1 Transfer modes ... 32B-165
32B.9.13.2 DMA channel priority control ... 32B-166
32B.9.13.3 Forced sweeping request ... 32B-168
32B.9.13.4 DMA transfer completion interrupt (USBFDMAmn) ... 32B-169
32B.9.13.5 DMA error interrupt (USBFDMAERRm) ... 32B-169
32B.9.13.6 Interval Count Function ... 32B-170
32B.9.13.7 Operational difference depending on the transfer size ... 32B-171
32B.9.13.8 Transfer state ... 32B-173
32B.9.14 Access Type ... 32B-176
32B.9.14.1 DMA master transfer combination list ... 32B-176
32B.9.14.2 DMA master descriptor combination list ... 32B-179
32B.9.15 Arbitration between DMACs ... 32B-180

32B.9.16 Notes ..... 32B-181
32B.9.16.1 Access ..... 32B-181
32B.9.16.2 Level Interrupt bit ..... 32B-181
33. LCDC ..... 33-1
33.1 Overview ..... 33-1
33.1.1 Features ..... 33-1
33.1.2 Block Diagram ..... 33-2
33.1.2.1 FCPVD ..... 33-4
33.1.2.2 VSPD ..... 33-4
33.1.2.3 DU ..... 33-13
33.1.3 External Pins ..... 33-14
33.1.4 Clock and Interrupt signal ..... 33-15
33.2 Register Configuration ..... 33-16
33.2.1 FCPVD Register Configuration ..... 33-16
33.2.2 VCPD Register Configuration ..... 33-17
33.2.3 DU Register Configuration ..... 33-20
33.3 Register Descriptions ..... 33-21
33.3.1 FCPVD Registers ..... 33-21
33.3.1.1 FCPVD version Control Register (FCP_VCR) ..... 33-21
33.3.2 VCPD Registers ..... 33-22
33.3.2.1 Notational Conventions for Registers and Bit Fields ..... 33-22
33.3.2.2 Register Classification ..... 33-22
33.3.2.3 Restrictions on Access to Registers and Lookup Tables ..... 33-23
33.3.2.4 General Control Registers ..... 33-24
33.3.2.5 Display List Control Registers ..... 33-41
33.3.2.6 RPF Control Registers ..... 33-49
33.3.2.7 WPF Control Registers ..... 33-87
33.3.2.8 DPR Control Registers ..... 33-105
33.3.2.9 LUT Control Register ..... 33-109
33.3.2.10 BRS Control Registers ..... 33-113
33.3.2.11 LIF Control Registers ..... 33-123
33.3.3 DU Registers ..... 33-128
33.3.3.1 DU Module Control Register 0 (DU_MCR0) ..... 33-128
33.3.3.2 DU Module Status Register 0 (DU_MSR0) ..... 33-129
33.3.3.3 DU Module Status Register 1 (DU_MSR1) ..... 33-130
33.3.3.4 DU Interrupt Mask Register 0 (DU_IMR0) ..... 33-130
33.3.3.5 DU Display I/F Timing Register 0 (DU_DITR0) ..... 33-131
33.3.3.6 DU Display I/F Timing Register 1 (DU_DITR1) ..... 33-132
33.3.3.7 DU Display I/F Timing Register 2 (DU_DITR2) ..... 33-134
33.3.3.8 DU Display I/F Timing Register 3 (DU_DITR3) ..... 33-134
33.3.3.9 DU Display I/F Timing Register 4 (DU_DITR4) ..... 33-135
33.3.3.10 DU Module Control Register 1 (DU_MCR1) ..... 33-136
33.3.3.11 DU PBUF Control Register 0 (DU_PBCR0) ..... 33-137
33.3.3.12 DU PBUF Control Register 1 (DU_PBCR1) ..... 33-138
33.3.3.13 DU PBUF Control Register 2 (DU_PBCR2) ..... 33-139
33.4 Operation ..... 33-140
33.4.1 Operation Control Setting ..... 33-140
33.4.2 Reset Operation ..... 33-140
33.4.2.1 Power-on Reset, Hardware reset ..... 33-140
33.4.3 Registers to set fixed value ..... 33-140
33.4.4 Concept of VSPD Operation Starting and Stopping ..... 33-141
33.4.5 Display List ..... 33-142
33.4.5.1 Functional Description ..... 33-142
33.4.5.2 Normal Display List Mode ..... 33-144
33.4.5.3 Header-less Display List Mode ..... 33-149
33.4.5.4 Restrictions on Display List Usage ..... 33-149
33.4.6 Interrupt Processing ..... 33-150

33.4.7 Lookup Table Settings....33-151
33.4.7.1 LUT....33-151
33.4.8 Linked with DU....33-152
33.4.8.1 Operation flow of VSPD and DU....33-152
33.4.8.2 Controlling Two Register Planes Using Display Lists....33-154
33.5 Usage Note....33-156
33.5.1 Assignment in Memory Space....33-156
33.5.2 Limitations on Software Reset and Module Standby....33-156
33.5.3 Input Image Size....33-156
33.5.4 Output Image Size....33-157
33.5.5 Restriction....33-159
34. MIPI DSI....34-1
34.1 Overview....34-1
34.1.1 Features....34-1
34.1.2 Block Diagram....34-4
34.1.3 Clock....34-5
34.1.4 External Pins....34-6
34.2 Register Configuration....34-7
34.2.1 Registers in LINK....34-7
34.2.2 Registers in D-PHY....34-10
34.3 Register Descriptions....34-11
34.3.1 Resisters in LINK....34-11
34.3.1.1 Interrupt Status Register (ISR)....34-12
34.3.1.2 Link Status Register (LINKSR)....34-13
34.3.1.3 Tx Set Register (TXSETR)....34-14
34.3.1.4 HS Clock Set Register (HSCLKSETR)....34-15
34.3.1.5 ULPS Set Register (ULPSSETR)....34-16
34.3.1.6 ULPS Control Register (ULPSCR)....34-17
34.3.1.7 Reset Control Register (RSTCR)....34-18
34.3.1.8 Reset Status Register (RSTSR)....34-19
34.3.1.9 DSI Set Register (DSISETR)....34-20
34.3.1.10 Receive Buffer Size Register (RXBUFSZR)....34-21
34.3.1.11 Tx Packet Payload Data 0 Register (TXPPD0R)....34-22
34.3.1.12 Tx Packet Payload Data 1 Register (TXPPD1R)....34-23
34.3.1.13 Tx Packet Payload Data 2 Register (TXPPD2R)....34-24
34.3.1.14 Tx Packet Payload Data 3 Register (TXPPD3R)....34-25
34.3.1.15 Rx Status Register (RXSR)....34-26
34.3.1.16 Rx Status Clear Register (RXSCR)....34-28
34.3.1.17 Rx Interrupt Enable Register (RXIER)....34-30
34.3.1.18 Peripheral Response Timeout BTA Set Register (PRESPTOBTASETR)....34-32
34.3.1.19 Peripheral Response Timeout LP Set Register (PRESPTOLPSETR)....34-33
34.3.1.20 Peripheral Response Timeout HS Set Register (PRESPTOHSSETR)....34-34
34.3.1.21 Acknowledge and Error Report Packet Parameter Latest Info Register (AKEPLATIR)34-35
34.3.1.22 Acknowledge and Error Report Packet Parameter Accumulate Status Register (AKEPACMSR)....134-36
34.3.1.23 Acknowledge and Error Report Packet Parameter Status Clear Register (AKEPSCR)....134-37
34.3.1.24 Rx Result Saved Status Register (RXRSSR)....34-38
34.3.1.25 Rx Result Saved Status Clear Register (RXRSSCR)....34-39
34.3.1.26 Rx Result Info Overwrite Status Register (RXRINFOOWSR)....34-40
34.3.1.27 Rx Result Info Overwrite Status Clear Register (RXRINFOOWSCR)....34-41
34.3.1.28 Rx Result Save Slot x Register (RXRSSxR)....34-42
34.3.1.29 Rx Packet Payload Data 0 Register (RXPPD0R)....34-44
34.3.1.30 Rx Packet Payload Data 1 Register (RXPPD1R)....34-45
34.3.1.31 Rx Packet Payload Data 2 Register (RXPPD2R)....34-46
34.3.1.32 Rx Packet Payload Data 3 Register (RXPPD3R)....34-47
34.3.1.33 HSTX Timeout Set Register (HSTXTOSETR)....34-48
34.3.1.34 LRX-H Timeout Set Register (LRXHTOSETR)....34-49

34.3.1.35 TA Timeout Set Register (TATOSETR) ..... 34-50
34.3.1.36 Fatal Error Status Register (FERRSR) ..... 34-51
34.3.1.37 Fatal Error Status Clear Register (FERRSCR) ..... 34-52
34.3.1.38 Fatal Error Interrupt Enable Register (FERRIER) ..... 34-53
34.3.1.39 Clock Lane Stop Time Set Register (CLSTPTSETR) ..... 34-54
34.3.1.40 LP Transition Time Set Register (LPTRNSTSETR) ..... 34-56
34.3.1.41 Physical Lane Status Register (PLSR) ..... 34-57
34.3.1.42 Physical Lane Status Clear Register (PLSCR) ..... 34-59
34.3.1.43 Physical Lane Interrupt Enable Register (PLIER) ..... 34-60
34.3.1.44 Video-Input Channel 1 Set 0 Register (VICH1SET0R) ..... 34-61
34.3.1.45 Video-Input Channel 1 Set 1 Register (VICH1SET1R) ..... 34-63
34.3.1.46 Video-Input Channel 1 Status Register (VICH1SR) ..... 34-64
34.3.1.47 Video-Input Channel 1 Status Clear Register (VICH1SCR) ..... 34-65
34.3.1.48 Video-Input Channel 1 Interrupt Enable Register (VICH1IER) ..... 34-66
34.3.1.49 Video-Input Channel 1 Pixel Packet Set Register (VICH1PPSETR) ..... 34-67
34.3.1.50 Video-Input Channel 1 Vertical Size Set Register (VICH1VSSETR) ..... 34-68
34.3.1.51 Video-Input Channel 1 Vertical Porch Set Register (VICH1VPSETR) ..... 34-69
34.3.1.52 Video-Input Channel 1 Horizontal Size Set Register (VICH1HSSETR) ..... 34-70
34.3.1.53 Video-Input Channel 1 Horizontal Porch Set Register (VICH1HPSETR) ..... 34-71
34.3.1.54 Sequence Channel x Set 0 Register (SQCHxSET0R) ..... 34-72
34.3.1.55 Sequence Channel x Set 1 Register (SQCHxSET1R) ..... 34-73
34.3.1.56 Sequence Channel x Status Register (SQCHxSR) ..... 34-74
34.3.1.57 Sequence Channel x Status Clear Register (SQCHxSCR) ..... 34-76
34.3.1.58 Sequence Channel x Interrupt Enable Register (SQCHxIER) ..... 34-78
34.3.1.59 Sequence Channel x Descriptor y-A Register (SQCHxDSCyAR) ..... 34-80
34.3.1.60 Sequence Channel x Descriptor y-B Register (SQCHxDSCyBR) ..... 34-82
34.3.1.61 Sequence Channel x Descriptor y-C Register (SQCHxDSCyCR) ..... 34-83
34.3.1.62 Sequence Channel x Descriptor y-D Register (SQCHxDSCyDR) ..... 34-84
34.3.2 Resisters in D-PHY ..... 34-85
34.3.2.1 DSI D-PHY Control Register 0 (DSIDPHYCTRL0) ..... 34-85
34.3.2.2 DSI D-PHY Timing Register 0 (DSIDPHYTIM0) ..... 34-86
34.3.2.3 DSI D-PHY Timing Register 1 (DSIDPHYTIM1) ..... 34-87
34.3.2.4 DSI D-PHY Timing Register 2 (DSIDPHYTIM2) ..... 34-88
34.3.2.5 DSI D-PHY Timing Register 3 (DSIDPHYTIM3) ..... 34-89
34.4 Operation ..... 34-90
34.4.1 Interrupts ..... 34-90
34.4.2 Operation Flow ..... 34-93
34.4.2.1 Reset ..... 34-93
34.4.2.2 Start/Stop of HS clock ..... 34-97
34.4.2.3 Sequence Operation ..... 34-100
34.4.2.4 Video-Input Operation ..... 34-105
34.4.2.5 Enter/Exit of ULPS ..... 34-110
34.4.2.6 Error Handling ..... 34-114
34.4.3 Interface ..... 34-121
34.4.3.1 Video-Input Operation ..... 34-121
34.4.3.2 Video Input Start, End, Change Setting ..... 34-125
34.4.3.3 Clock Frequencies ..... 34-125
34.5 Usage Note ..... 34-126
34.5.1 Restriction and Notes ..... 34-126
39. 2D Drawing Engine (DRW) ..... 39-1
39.1 Overview ..... 39-1
39.2 Register Descriptions ..... 39-4
39.2.1 Geometry Control Register (CONTROL) ..... 39-6
39.2.2 Surface Control Register (CONTROL2) ..... 39-8
39.2.3 Interrupt Control Register (IRQCTL) ..... 39-12
39.2.4 Cache Control Register (CACHECTL) ..... 39-13
39.2.5 Status Control Register (STATUS) ..... 39-14
39.2.6 Hardware Version and Feature Set ID Register (HWREVISION) ..... 39-16

39.2.7 Base Color Register (COLOR1) ..... 39-17
39.2.8 Secondary Color Register (COLOR2) ..... 39-18
39.2.9 Pattern Register (PATTERN) ..... 39-19
39.2.10 Limiter N Start Value Register (LnSTART) ..... 39-20
39.2.11 Limiter N X-Axis Increment Register (LnXADD) ..... 39-21
39.2.12 Limiter N Y-Axis Increment Register (LnYADD) ..... 39-22
39.2.13 Limiter M Bandwidth Parameter Register (LmBAND) ..... 39-23
39.2.14 Texture Base Address Register (TEXORIGIN) ..... 39-24
39.2.15 Texels Per Texture Line Register (TEXPITCH) ..... 39-25
39.2.16 Texture Size or Texture Address Mask Register (TEXMASK) ..... 39-26
39.2.17 U Limiter Start Value Register (LUSTART) ..... 39-27
39.2.18 U Limiter X-Axis Increment Register (LUXADD) ..... 39-28
39.2.19 U Limiter Y-Axis Increment Register (LUYADD) ..... 39-29
39.2.20 V Limiter Start Value Integer Part Register (LVSTARTI) ..... 39-30
39.2.21 V Limiter Start Value Fractional Part Register (LVSTARTF) ..... 39-31
39.2.22 V Limiter X-Axis Increment Integer Part Register (LVXADDI) ..... 39-32
39.2.23 V Limiter Y-Axis Increment Integer Part Register (LVYADDI) ..... 39-33
39.2.24 V Limiter Increment Fractional Parts Register (LVYXADDF) ..... 39-34
39.2.25 CLUT Start Address Register (TEXCLADDR) ..... 39-35
39.2.26 CLUT Data Register (TEXCLDATA) ..... 39-36
39.2.27 CLUT Offset Register (TEXCLOFFSET) ..... 39-37
39.2.28 Color Key Register (COLKEY) ..... 39-38
39.2.29 Bounding Box Dimension Register (SIZE) ..... 39-39
39.2.30 Framebuffer Pitch And Spanstore Delay Register (PITCH) ..... 39-40
39.2.31 Framebuffer Base Address Register (ORIGIN) ..... 39-41
39.2.32 Display List Start Address Register (DLISTSTART) ..... 39-42
39.2.33 Performance Counters Control Register (PERFTRIGGER) ..... 39-43
39.2.34 Performance Counter k (PERFCOUNTk) (k = 1, 2) ..... 39-44

39.3 Drawing Features ..... 39-45

39.3.1 Drawing Features Summary ..... 39-45
39.3.1.1 Color Formats ..... 39-45
39.3.1.2 BitBLT Features ..... 39-45
39.3.1.3 Vector Drawing Features ..... 39-46
39.3.2 Vector Drawing ..... 39-47
39.3.3 BitBLT ..... 39-48
39.3.3.1 Fill ..... 39-48
39.3.3.2 Copy ..... 39-48
39.3.3.3 Stretch BitBLT ..... 39-48
39.3.3.4 Rotate and Scale ..... 39-48
39.3.3.5 Alpha Blending ..... 39-48
39.3.3.6 Bilinear Filtering ..... 39-49
39.3.3.7 Color Conversion ..... 39-49

39.4 Input and Output Data Formats ..... 39-50

39.4.1 Source and Destination Data ..... 39-50
39.4.2 Framebuffer Color Formats ..... 39-50
39.4.3 Texture Color Formats ..... 39-52

39.5 Texture Data Processing ..... 39-55

39.5.1 Texture Color Format ..... 39-55
39.5.2 Run Length Encoded (RLE) Unit ..... 39-56
39.5.2.1 RLE Texel Formats ..... 39-57
39.5.2.2 Targa RLE Format ..... 39-58
39.5.3 Color Lookup Table (CLUT) ..... 39-59
39.5.3.1 CLUT/I Pixel Data Formats ..... 39-59
39.5.4 Color Keying ..... 39-60

39.6 Rendering Pipeline ..... 39-61

39.6.1 Coordinate Transformation ..... 39-61

39.6.2 Rasterization....39-61
39.6.2.1 Edge Setup Linear Case....39-63
39.6.2.2 Edge Setup Quadratic Case....39-67
39.6.2.3 Band Filter....39-71
39.6.2.4 Clamping Unit....39-72
39.6.2.5 Combiner Unit....39-73
39.6.2.6 Rasterization Optimization....39-74
39.6.3 Texturing....39-77
39.6.3.1 Mathematical Background....39-77
39.6.3.2 Limiter Operation....39-81
39.6.4 Colorization....39-82
39.6.5 Blending....39-83
39.6.5.1 Color Channel Blending....39-83
39.6.5.2 Alpha Channel Blending....39-85
39.7 Rendering Modes....39-87
39.7.1 Register Mode....39-87
39.7.2 Display List Mode....39-87
39.7.3 Stopping the Render Process....39-91
39.8 Interrupts....39-92
39.8.1 Interrupt Sources....39-92
39.8.2 Interrupt Control....39-93
39.9 Performance Counters....39-94
39.10 Stopping the 2D Drawing Engine Render Process....39-95
40. SD Host Interface....40-1
40.1 Overview....40-1
40.1.1 Features....40-1
40.1.2 External Pins....40-3
40.1.3 Register Configuration....40-4
40.2 Register Description....40-6
40.2.1 Command Type Register (SD_CMD)....40-6
40.2.2 Command Argument Register (SD_ARG)....40-7
40.2.3 Data Stop Register (SD_STOP)....40-8
40.2.4 Block Count Register (SD_SECCNT)....40-10
40.2.5 SD Card Response Registers (SD_RSP)....40-11
40.2.6 SD Card Interrupt Flag Register (SD_INFO1)....40-13
40.2.7 SD Card Interrupt Flag Register (SD_INFO2)....40-16
40.2.8 SD_INFO1 Interrupt Mask Register (SD_INFO1_MASK)....40-20
40.2.9 SD_INFO2 Interrupt Mask Register (SD_INFO2_MASK)....40-21
40.2.10 SD Clock Control Register (SD_CLK_CTRL)....40-22
40.2.11 Transfer Data Length Register (SD_SIZE)....40-24
40.2.12 SD Card Access Control Option Register (SD_OPTION)....40-25
40.2.13 SD Error Status Register 1 (SD_ERR_STS1)....40-26
40.2.14 SD Error Status Register 2 (SD_ERR_STS2)....40-27
40.2.15 SD Buffer Read/Write Register (SD_BUF0)....40-27
40.2.16 SDIO Mode Control Register (SDIO_MODE)....40-28
40.2.17 SDIO Interrupt Flag Register (SDIO_INFO1)....40-30
40.2.18 SDIO_INFO1 Interrupt Mask Register (SDIO_INFO1_MASK)....40-31
40.2.19 DMA Mode Enable Register (CC_EXT_MODE)....40-32
40.2.20 Software Reset Register (SOFT_RST)....40-32
40.2.21 Version Register (VERSION)....40-33
40.2.22 Host Interface Mode Setting Register (HOST_MODE)....40-33
40.2.23 SD Status Register (SD_STATUS)....40-34
40.2.24 DMAC Transfer Mode Register (DM_CM_DTRAN_MODE)....40-35
40.2.25 DMAC Transfer Control Register (DM_CM_DTRAN_CTRL)....40-35

40.2.26 DMAC Reset Register (DM_CM_RST) ..... 40-36
40.2.27 DMAC Interrupt Register 1 (DM_CM_INFO1) ..... 40-37
40.2.28 DM_CM_INFO1 Interrupt Mask Register (DM_CM_INFO1_MASK) ..... 40-38
40.2.29 DMAC Interrupt Register 2 (DM_CM_INFO2) ..... 40-39
40.2.30 DM_CM_INFO2 Interrupt Mask Register (DM_CM_INFO2_MASK) ..... 40-40
40.2.31 DMAC Transfer Address Register (DM_DTRAN_ADDR) ..... 40-40
40.3 Operation ..... 40-41
40.3.1 SD Interface ..... 40-41
40.3.2 Card Detect/Write Protect ..... 40-45
40.3.3 Interrupt Request ..... 40-47
40.3.4 Communications Errors and Timeouts ..... 40-48
40.4 Usage Example ..... 40-50
40.4.1 Command without Data Transfer ..... 40-50
40.4.2 Single Block Read ..... 40-53
40.4.3 Single Block Write ..... 40-55
40.4.4 Multiple Block Read ..... 40-57
40.4.5 Multiple Block Write (when Using Internal Timer) ..... 40-59
40.4.6 Multiple Block Write (when Using External Timer) ..... 40-61
40.4.7 IO_RW_DIRECT Commend (CMD52) ..... 40-63
40.4.8 IO_RW_EXTENDED (CMD53/Multiple Block Read) ..... 40-64
40.4.9 IO_RW_EXTENDED (CMD53/Multiple Block Write) ..... 40-66
40.4.10 DMA Transfer ..... 40-68
40.4.11 High-Priority Interrupt (without Data Transfer) ..... 40-70
40.4.12 High-Priority Interrupt (at Single Block Write) ..... 40-72
40.4.13 High-Priority Interrupt (at Multiple Block Write) ..... 40-74
40.4.14 Example of SD_CMD Register Setting ..... 40-78
40.5 Usage Notes ..... 40-80
40.6 Sampling Clock Controller (SCC) ..... 40-83
40.6.1 Features ..... 40-83
40.6.2 SCC Block Diagram ..... 40-83
40.6.3 SCC Register Configuration ..... 40-84
40.7 SCC Register Descriptions ..... 40-85
40.7.1 Initial Setting Register (SCC_DTCNTL) ..... 40-85
40.7.2 Sampling Clock Position Setting Register (SCC_TAPSET) ..... 40-85
40.7.3 Sampling Clock Selection Register (SCC_CKSEL) ..... 40-85
40.7.4 Sampling Clock Position Correction Register (SCC_RVSCNTL) ..... 40-86
40.7.5 Sampling Clock Position Correction Request Register (SCC_RVSREQ) ..... 40-86
40.7.6 Hardware Adjustment Register 1 (SCC_DT2FF) ..... 40-87
40.7.7 Sampling data comparison register (SCC_SMPCMP) ..... 40-87
40.7.8 Hardware adjustment register 2 (SCC_TMPPORT) ..... 40-88
40.8 Usage Example of SCC ..... 40-89
40.8.1 Tuning ..... 40-89
40.8.2 Sampling Clock Position Correction after Tuning ..... 40-92
40.8.3 Change point of the input data ..... 40-94
41. General Purpose Input Output Port (GPIO) ..... 41-1
41.1 Features ..... 41-1
41.1.1 General Purpose Input Output Port (GPIO) ..... 41-1
41.1.2 Port Function Control ..... 41-1
41.1.3 Special Purpose Port Function Control ..... 41-2
41.1.4 General Purpose Input Output Port Configuration ..... 41-3
41.1.5 Special purpose port Configuration ..... 41-4
41.2 Register Configuration ..... 41-9
41.2.1 Port Register (Pn) ..... 41-9

41.2.2 Port Mode Register (PMn) ....41-9
41.2.3 Port Mode Control Register (PMCn) ....41-10
41.2.4 Port Function Control Register (PFCm) ....41-10
41.2.5 Port Input Register (PINn) ....41-11
41.2.6 Interrupt Enable Control Register (ISEL) ....41-11
41.2.7 Driving Ability Control Register (IOLH) ....41-12
41.2.8 Slew Rate Switching Register (SR) ....41-13
41.2.9 Pull Up/Pull down Switching Register (PUPD) ....41-14
41.2.10 Write protected register (PWPR) ....41-14
41.2.11 Digital Noise Filter Switching Register (FILONOFF) ....41-14
41.2.12 Digital Noise Filter Number Register (FILNUM) ....41-15
41.2.13 Digital Noise Filter Clock Selection Register (FILCLKSEL) ....41-16
41.2.14 SD ch0 IO Voltage Mode Control Register (SD_ch0) ....41-16
41.2.15 QSPI IO Voltage Mode Control Register (QSPI) ....41-17
41.2.16 Input Enable Control Register (IEN) ....41-17
41.3 Register Descriptions ....41-18
41.3.1 Port Register (Pn) ....41-18
41.3.2 Port Mode Register (PMn) ....41-19
41.3.3 Port Mode Control Register (PMCn) ....41-20
41.3.4 Port Function Control Register (PFCm) ....41-22
41.3.5 Port Input Register (PINn) ....41-24
41.3.6 Interrupt Enable Control Register (ISEL) ....41-25
41.3.7 Driving Ability Control Register (IOLH) ....41-27
41.3.8 Slew Rate Switching Register (SR) ....41-29
41.3.9 Pull Up/Pull down Switching Register (PUPD) ....41-31
41.3.10 Write protected register (PWPR) ....41-33
41.3.11 Digital Noise Filter Switching Register (FILONOFF) ....41-34
41.3.12 Digital Noise Filter Number Register (FILNUM) ....41-36
41.3.13 Digital Noise Filter Clock Selection Register (FILCLKSEL) ....41-38
41.3.14 SD ch0 IO Voltage Mode Control Register (SD_ch0) ....41-43
41.3.15 QSPI IO Voltage Mode Control Register (QSPI) ....41-44
41.3.16 Input Enable Control Register (IEN) ....41-45
41.4 Operation ....41-46
41.4.1 Operation for GPIO Function ....41-46
41.4.2 Operation for Peripheral Function ....41-46
42. Low Power Mode ....42-1
42.1 Overview ....42-1
42.2 Module Standby Mode ....42-2
42.2.1 Overview ....42-2
42.2.2 Operation ....42-4
42.2.2.1 Setting of Clock control and MSTOP register ....42-5
42.3 Cortex-A55 Sleep Mode ....42-7
42.3.1 Overview ....42-7
43. Debug Interface ....43-1
43.1 Overview ....43-1
43.1.1 Features ....43-1
43.1.2 Block Diagram ....43-2
43.1.3 External Pins ....43-4
43.1.4 Treatment of Unused Pins ....43-6
43.2 Address Space ....43-7
43.3 Register Descriptions ....43-9
43.3.1 ID Register (BSID) ....43-9
43.3.2 Other Registers ....43-9
43.4 Debug Connection ....43-10
43.4.1 Debug Connection Mode ....43-10

43.5 Debug Access Range....43-10
43.6 Operation....43-11
43.6.1 Debug Related Reset....43-11
43.6.2 DAP....43-11
43.6.3 Trace....43-11
43.6.3.1 Data Flow....43-11
43.6.3.2 Timestamp....43-11
43.6.4 Cross Trigger....43-12
43.6.4.1 CoreSight Cross Trigger Connectiion....43-12
43.6.4.2 Cortex-A55 Cross Trigger Connectiion....43-13
43.6.5 WDT Counter Stop Control....43-14
43.6.6 GTM Counter Stop Control....43-14
43.6.7 SYC Counter Stop Control....43-14
44. Thermal Sensor Unit (TSU)....44-1
44.1 Overview....44-1
44.1.1 Features....44-1
44.1.2 Block Diagram....44-2
44.1.2.1 APB_REG....44-3
44.1.2.2 Thermal Sensor....44-3
44.1.2.3 DIV40_TS and DIV10_TS....44-3
44.1.2.4 SOC_TS_GEN....44-3
44.1.2.5 EOC_SY_TS....44-3
44.1.2.6 Read Buffer (RB)....44-3
44.1.3 External Pins....44-4
44.1.4 Interrupts....44-4
44.2 Register Configuration....44-5
44.2.1 List of Register....44-5
44.3 Register Descriptions....44-6
44.3.1 Sensor Mode Register (TSU_SM)....44-6
44.3.2 Sensor Trigger Register (TSU_ST)....44-7
44.3.3 Sensor ADC Data Register (TSU_SAD)....44-8
44.3.4 Sensor Status Register (TSU_SS)....44-9
44.3.5 TSU OTP Calibration Register 0 (OTPTSUTRIM0_REG)....44-10
44.3.6 TSU OTP Calibration Register 1 (OTPTSUTRIM1_REG)....44-11
44.4 Operation....44-12
44.4.1 Procedure for Starting the A/D Conversion....44-12
44.4.2 Procedure for Stopping the A/D Conversion....44-12
44.4.3 Confirming the State of the A/D Conversion by Monitoring the CONV Bit....44-13
44.4.4 Procedure for Measuring the Temperature....44-13
44.5 Usage Note....44-14
48. Electrical Characteristics....48-1
48.1 Absolute Maximum Ratings....48-1
48.2 Power Supply....48-2
48.3 Power-On/Power-Off Sequence....48-3
48.4 DC Characteristics....48-5
48.5 AC Characteristics....48-10
48.5.1 Clock Timing....48-10
48.5.2 SDHI Access Timing....48-12
48.5.2.1 SDHI Access Timing (SDR 3.3-V)....48-12
48.5.3 LCDC Access Timing....48-13
48.5.4 USB 2.0 Host/Function Module Access Timing....48-14
48.5.4.1 USB 2.0 Low-Speed Access Timing....48-14
48.5.4.2 USB 2.0Full-Speed Access Timing....48-15
48.5.4.3 USB 2.0 Hi-Speed Access Timing....48-16
48.5.5 JTAG Debugger Interface Access Timing....48-17
48.5.6 SPI Multi I/O Bus Controller Access Timing....48-19

48.5.7 Control Signal Access Timing....48-22
48.5.8 Serial Sound Interface (SSIF-2) Access Timing....48-24
48.5.9 Multi-Function Timer Pulse Unit 3 (MTU3a) Access Timing....48-26
48.5.10 Port Output Enable 3 (POE3) Access Timing....48-27
48.5.11 $I^2C$ Bus Interface Access Timing....48-28
48.5.12 Serial Communications Interface with FIFO (SCIFA) Access Timing....48-30
48.5.13 Serial Communications Interface (SCIg) Access Timing....48-32
48.5.14 Renesas Serial Peripheral Interface (RSPI) Access Timing....48-34
48.5.15 Watchdog Timer Access Timing....48-37
APPENDIX A PACKAGE DIMENSIONS.... APPENDIX A-1
A.1 Package Dimensions....A-1
REVISION HISTORY....REVISION HISTORY-1

# 1. Overview

## 1.1 Introduction

The RZ/A3M includes:

- 1.0 GHz Arm® Cortex®-A55 Single MPCore core
- Memory controller for DDR3L-1600 with 16 bits (DRAM chip is embedded)
- 2D drawing engine
- 1 channel MIPI DSI interface or 1 channel parallel output interface selectable
- USB2.0 host / function interface
- SD host interface
- Sound interface

NOTE

Arm and Cortex are registered trademark of Arm Limited. All other brands or product names are the property of their respective holders.

## 1.2 List of Specifications

### 1.2.1 CPU Core

| Item | Description |
|---|---|
| System CPU Cortex-A55 | • Arm Cortex-A55 Single MPCore 1.0 GHz<br>• L1 I-cache 32 Kbytes (Parity) / D-cache 32 Kbytes (ECC)<br>• L2 cache 0 Kbyte<br>• L3 cache 256 Kbytes (ECC)<br>• Arm® NEON™ / FPU supported<br>• Arm® v8.2-A architecture |
| Boot | • 5 boot modes<br>Boot Mode 3: Booting from a serial NOR flash memory (Single / Quad) connected to the SPI Multi I/O bus space (1.8 V)<br>Boot Mode 4: Booting from a serial NOR flash memory (Single / Quad) connected to the SPI Multi I/O bus space (3.3 V)<br>Boot Mode 5: Booting from SCIF download<br>Boot Mode 6: Booting from a serial NAND flash memory (Single / Quad) Connected to the SPI Multi I/O bus space (1.8V)<br>Boot Mode 7: Booting from a serial NAND flash memory (Single / Quad) Connected to the SPI Multi I/O bus space (3.3V) |
| Debug Interface | • Arm® CoreSight™ architecture<br>• JTAG / SWD interface supported<br>• ETF 16 Kbytes for program flow trace (each cluster) |

### 1.2.2 CPU Peripheral

| Item | Description |
|---|---|
| Clock Pulse Generator (CPG) | • Generates the clocks from external clock (EXCLK 24 MHz).<br>Maximum Arm Cortex-A55 clock: 1.0 GHz<br>Maximum DDR clock: 800 MHz (DDR3L-1600)<br>Maximum DRW clock: 200 MHz<br>Maximum AXI-bus clock: 200 MHz<br>Maximum APB-bus clock: 100 MHz<br>• SSC (Spread Spectrum Clock) supported |
| Direct Memory Access Controller (DMAC) | • 2 modules, 16 channels per module<br>• Transfer request: On-chip peripheral request / auto request (software trigger)<br>• A specific DMA transfer interval can be specified to adjust the bus occupancy.<br>• LINK mode (DMA transfer under descriptor control) supported<br>• Transfer information can be automatically reloaded |
| Interrupt Controller | • Arm® CoreLink™ Generic Interrupt Controller (GIC-600) for Arm Cortex-A55<br>• External Interrupt pins (NMI, IRQ7 to IRQ0, TINT31-0)<br>• On-chip peripheral Interrupts: Priority level set for each module |
| General-purpose I/O (GPIO) | • General-purpose I/O ports |
| Thermal Sensor Unit (TSU) | • 1 channel |

### 1.2.3 Internal Memory

| Item | Description |
|---|---|
| On-chip RAM | • RAM of 128 Kbytes (ECC) |

### 1.2.4 External Memory Interface

| Item | Description |
|---|---|
| External Bus Controller for DDR3L SDRAM (DDR) *1 | • Support DDR3L-1600<br>• Bus Width: 16-bit<br>• Memory Size: 128M Bytes<br>• Auto Refresh supported |
| SPI Multi I/O Bus Controller | • 1 channel (4-bit Double data rate)<br>• 1 serial flash memory with multiple I/O bus sizes (single / quad) can be connected<br>• External address space read mode (built-in read cache)<br>• SPI operation mode<br>• Maximum Clock Frequency:<br>– 50 MHz (Quad-SPI DDR)<br>– 66 MHz (Quad-SPI SDR) |
| SD Card Host Interface(SD) | • 1 channel<br>• Channel 0 supports SDHI<br>• SD memory I/O card interface (1-bit / 4-bit SD bus)<br>• SD, SDHC and SDXC SD memory card access supported<br>• Compliant with SD 3.0<br>• Default, high-speed, UHS-I/SDR50, SDR104 transfer modes supported<br>• Error check function: CRC7 (Command), CRC16 (Data)<br>• Card detection function, write protect supported |

Note 1. This LSI is a SiP (system in package) with build-in DDR3L SDRAM, so customers cannot use DDR interface.

### 1.2.5 Video Processing Unit

| Item | Description |
|---|---|
| 2D drawing engine (DRW) | • Support almost any object geometry, rather than being bound to only a few specific geometries such as lines, triangles, or circles. The edges of every object can be independently blurred or anti-aliased<br>• Color Formats<br>Frame buffer formats<br>8-bit: a (8)<br>16-bit: RGB (565), aRGB (4444), aRGB (1555)<br>32-bit: aRGB (8888)<br>Texture formats<br>1-bit: CLUT (1) / I (1)<br>2-bit: CLUT (2) / I (2)<br>4-bit: CLUT (4) / I (4)<br>16-bit: aRGB (4444), aRGB (1555), RGB (565)<br>24-bit: RGB (888) (run length encoded (RLE) unit)<br>32-bit: aRGB (8888).<br>CLUT formats use a 256-entry color lookup table. |

### 1.2.6 Display Interface

| Item | Description |
|---|---|
| LCD Controller | ● 1 channel (Parallel output)<br>● 2 planes blending (can blend 2 different size images)<br>● Support Image Processing:<br>– Dither processing (RGB666)<br>– Clipping<br>– RGB Gamma Correction LUT<br>● Support Input Data Format:<br>– RGB565 / RGB666 / RGB888<br>– ARGB1555 / ARGB4444 / ARGB8888<br>– YcbCr444 8-bit / YcbCr422 8-bit / YcbCr420 8-bit |
| MIPI DSI Interface | ● 1 channel<br>● The number of Lane: 4-lane<br>● Support up to WXGA (1280 x 800), 60 fps (RGB888)<br>● Maximum Bandwidth: 1.5 Gbps per lane<br>● Support Output Data Format:<br>- RGB666 / RGB888 |
| Parallel Output Interface | ● 1 channel<br>● Support WXGA (1280 × 800), 60 fps<br>● Support Output Data Format:<br>– RGB666 / RGB888<br>● CLK / HD / VD timing signal supported |

### 1.2.7 Sound Interface

| Item | Description |
|---|---|
| Serial Sound Interface (SSI) | ● 1 channel bidirectional serial transfer<br>● 2 external clock sources available<br>● Duplex communication<br>● Support of I2S / Monaural / TDM audio formats<br>● Support of master and slave functions<br>● Generation of programmable word clock and bit clock<br>● Multi-channel formats<br>● Support of 8, 16, 18, 20, 22, 24, and 32-bit data formats<br>● Support of 32-stage FIFO for transmission and reception<br>● Support of LR-clock continue function in which the LR-clock signal is not stopped |

### 1.2.8 Storage and Network

| Item | Description |
|---|---|
| USB2.0 Host / Function (USB) | • 1 channel (Host-Function)<br>• Compliance with USB2.0<br>• Supports Battery Charging Function<br>• Internal dedicated DMA |

### 1.2.9 Timer

| Item | Description |
|---|---|
| Multi-function Timer Pulse Unit 3 (MTU3a) | • 9 channels (16 bits × 8 channels, 32 bits × 1 channel)<br>• Module clock frequency (P0ϕ): 100 MHz<br>• Maximum 28 lines of pulse inputs/outputs and 3 lines of pulse inputs<br>• 14 types of count clocks selectable<br>• Input capture function<br>• 39 outputs compare and input capture registers<br>• Counter clear operation (Simultaneous counter clearing by Compare match or Input capture is available)<br>• Simultaneous writing to multiple timer counters (TCNT)<br>• Synchronous input/output of each register due to synchronous operation of the counter<br>• Buffered operation<br>• Cascade-connected operation<br>• 43 types of interrupt sources<br>• Automatic transfer of register data<br>• Pulse output modes<br>Toggle, PWM, complementary PWM, and reset-synchronized PWM modes<br>• Synchronization of multiple counters<br>• Phase counting mode<br>– 16-bit mode (channel 1 and 2)<br>– 32-bit mode (channel 1 and 2)<br>• Counter function of dead time compensation<br>• Digital filter functions for the input capture and external count clock pin |
| Port Output Enable 3 (POE3) | • Control of the high-impedance state of the MTU3a waveform output pins<br>• Activation with four input pins<br>• Activation on detection of short-circuited outputs<br>• Activation by register write<br>• Additional programming of output control target pins is possible. |
| Watchdog Timer (WDT) | • 1 channel<br>• A counter overflow can reset the LSI<br>• CPU parity error can reset the LSI |
| General Timer (GTM) | • 32 bits × 3 channels<br>• Two operating modes<br>– Interval timer mode<br>– Free-running comparison mode |

### 1.2.10 Peripheral Module

| Item | Description |
|---|---|
| I2C Bus Interface (I2C) | • 2 channels (ch0 = Dedicated pin, ch1 = Multiplexed pin)<br>• Master mode and slave mode supported<br>• Support for 7-bit and 10-bit slave address formats<br>• Support for multi-master operation<br>• Timeout detection |
| Serial Communication Interface with FIFO (SCIFA) | • 5 channels<br>• Clock synchronous mode or asynchronous mode selectable<br>• Simultaneous transmission and reception (full-duplex communication) supported<br>• Dedicated baud rate generator<br>• Separate 16-byte FIFO registers for transmission and reception<br>• Modem control function (channel 0, 1, and 2 in asynchronous mode) |
| Serial Communication Interface (SCIg) | • 2 channels<br>• Clock synchronous mode, asynchronous mode, or smart card interface mode is selectable<br>• Simultaneous transmission and reception (full-duplex communication) supported<br>• Dedicated baud rate generator<br>• LSB first / MSB first selectable<br>• Modem control function<br>• Encoding and decoding of IrDA communications waveforms in accord with version 1.0 of the IrDA standard (on channel 0) |
| Renesas Serial Peripheral Interface (RSPI) | • 2 channels<br>• SPI operation<br>• Master mode and slave mode supported<br>• Programmable bit length, clock polarity, clock phase can be selected<br>• Consecutive transfers<br>• LSB first / MSB first selectable |

### 1.2.11 Others

| Item | Description |
|---|---|
| Boundary Scan | • Boundary scan based on IEEE 1149.1 via JTAG interface is supported.<br>Note that some module pins are not available on this boundary scan. |

### 1.2.12 Power Supply Voltage

| Item | Description |
|---|---|
| Power supply voltage | • VDD, PLLn_DVDD11 (n = 23, 5): 1.05 to 1.15 V<br>• DDR_VDDQn (n = 1,2): 1.283 to 1.45 V (DDR3L)<br>• VDD18, JTAG_PVDD, PLLn_AVDD18 (n = 1, 23, 4, 5): 1.62 to 1.98 V<br>• OTP_VDD18, USB_VDD18, DSI_VDD18: 1.65 to 1.95 V<br>• PVDD: 2.97 to 3.63 V<br>• USB_VDD33: 3.00 to 3.60 V<br>• SDn_PVDD (n = 0), SPI_PVDD: 2.97 to 3.63 V / 1.70 to 1.95 V |

### 1.2.13 Temperature Range

| Item | Description |
|---|---|
| Temperature range | • $T_a$: −40°C to +85°C*1<br>• $T_j$: −40°C to +110°C |

Note 1. If wider temp is required than this range, use case has to be investigated.

### 1.2.14 Quality level

| Item | Description |
|---|---|
| Quality level | • Industrial usage, etc. |

### 1.2.15 Package

| Item | Description |
|---|---|
| Package | • 244-pin LFBGA, 17-mm square, 0.8-mm pitch |

## 1.3 Product Lineup

Table 1.1 Product Lineup

| Group | RAM | Security | Part Number |
|---|---|---|---|
| RZ/A3M | 128MByte | No-Security | R9A07G066M04GBG |

## 1.4 Pin

This section describes the pins of this LSI.

### 1.4.1 Pin Assignment

Refer to attached excel file for the “ball view” about pin assignment of this LSI.

(Please double-click the icon on the right side) 

### 1.4.2 External Pins and Multiplexed Functional Pins

Refer to attached excel file for the “pin function list” about information of external pins and multiplexed functional pins of this LSI (Please double-click the icon on the right side). 

# 2. System CPU Cortex-A55

The Cortex-A55 system CPU is a core block equipped with single Cortex-A55 core. For details on the functions including interrupts of the Cortex-A55, see the Arm® Cortex®-A55 Core Technical Reference Manual and related documents in **Section 2.4**. For details of interrupt type and control, see the chapter on Interrupt Controller.

## 2.1 Features

The Cortex-A55 incorporates an extensively redesigned microarchitecture system that improves performance across the board while being very competitive in area and power efficiency.

Table 2.1 Function Summary

| | Description |
|---|---|
| CPU | • 1.0 GHz Single Cortex-A55 r2p0 (Armv8-A) core |
| Cache memory | • L1$ (I/D) = 32 KB(Parity) / 32 KB (ECC)<br>• L2$ = None<br>• L3$ = 256 KB (ECC) |

Table 2.2 Core Configuration

| | Value | Description |
|---|---|---|
| L1 instruction cache size | 32 KB | L1 instruction cache = 32 KB (Parity) |
| L1 data cache size | 32 KB | L1 data cache = 32 KB (ECC) |
| L3 cache size | 256 KB | L3 cache = 256KB (ECC) |
| ECC or parity core cache protection | Included | Support for core cache ECC |
| Advanced SIMD and floating-point support (including dot product instruction support) | Included | Support for NEON™ FPU engine |
| Big endian | N/A | Not supported |

## 2.2 Resets

**Table 2.3** shows the supported types of reset and the areas that are reset by the respective types. For the procedures of handling the individual resets, see **Section 2.2.1** and the subsequent parts. See the Arm® DynamIQ™ Shared Unit Technical Reference Manual and Arm® Cortex®-A55 Core Technical Reference Manual.

Table 2.3 Areas to be Reset

| Reset Function | Areas to be Reset |
|---|---|
| Cold reset | Entire area |
| Core warm reset | Entire area of the target core except for the debug, RAS, and ETM registers |
| Debug reset | DebugBlock |

Reset address vector can be set by below Reset Vector Address Configuration Registers. For details, see the chapter of System Controller.

CA55 Core0 Reset Vector Address High/Low Configuration Register:
SYS_CA55_CFG_RVAH0 / SYS_CA55_CFG_RVAL0

To change reset vector address, follow the procedure below.

1. Store codes to new reset address.
2. Write the new reset address to CA55 Reset Vector Address Configuration Registers.
3. Issue reset to target core.

### 2.2.1 Cold Reset

A cold reset is applied at any of the following cases.

(a) The sequence of a power-on reset by the CPG is executed.

(b) The WDT counter has overflowed or a non-correctable error has been detected.

For details on (a) and (b) reset, see the chapters on the clock pulse generator (CPG) and watchdog timer (WDT), respectively.

### 2.2.2 Core Warm Reset

A core warm reset requires software control by the Cortex-A55 system CPU itself.

To apply a core warm reset to core 0, follow the procedure below.

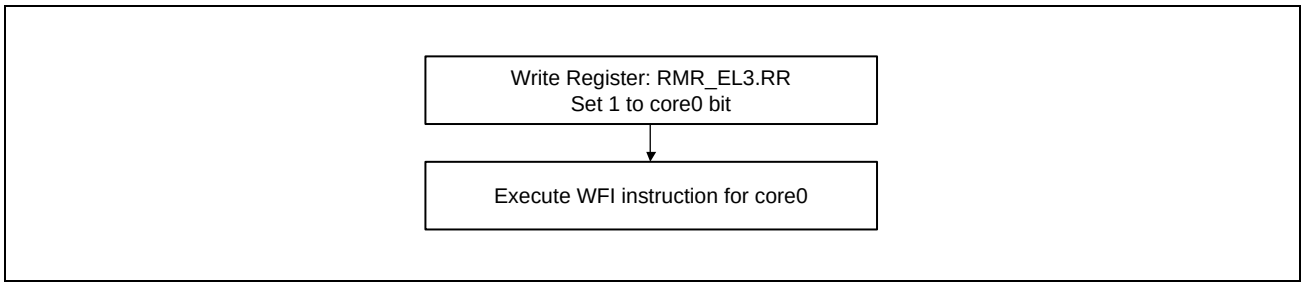



Figure 2.1 Core Warm Reset flow

Release from a core warm reset is automatic following application of the reset. No specific processing is required. For details, see the Arm® DynamIQ™ Shared Unit Technical Reference Manual and Arm® Cortex®-A55 Core Technical Reference Manual.

### 2.2.3 Debug Reset

Debug resets can only be applied under software control.

To apply a debug reset, follow the procedure below.

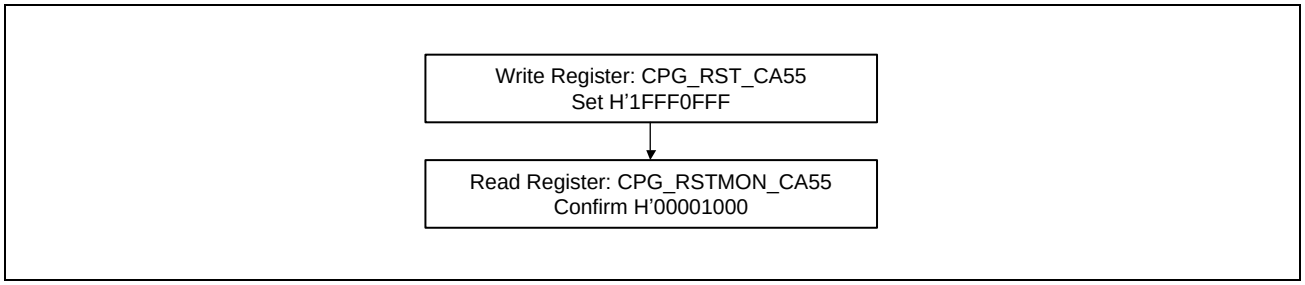



Figure 2.2 Debug Reset flow

To release from the debug reset state, follow the procedure below.

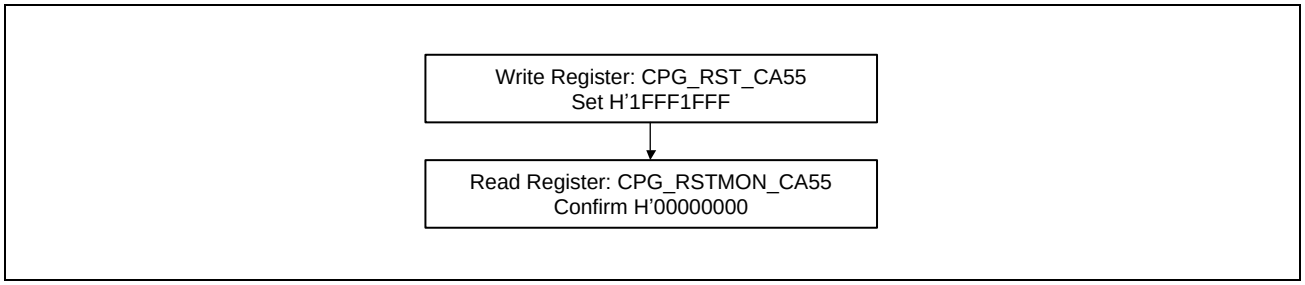



Figure 2.3 Release from Debug Reset state

## 2.3 Low Power Consumption Mode

### 2.3.1 Cortex-A55 Sleep Mode

Cortex-A55 Sleep Mode of this LSI is a low power consumption mode obtained by having a core execute the WFI instruction. For details on the WFI instructions, see the Arm® Cortex®-A55 Core Technical Reference Manual and Arm® DynamIQ Shared Unit Technical Reference Manual.

The following procedure is the example sequence of a transition to Cortex-A55 Sleep Mode (core0).

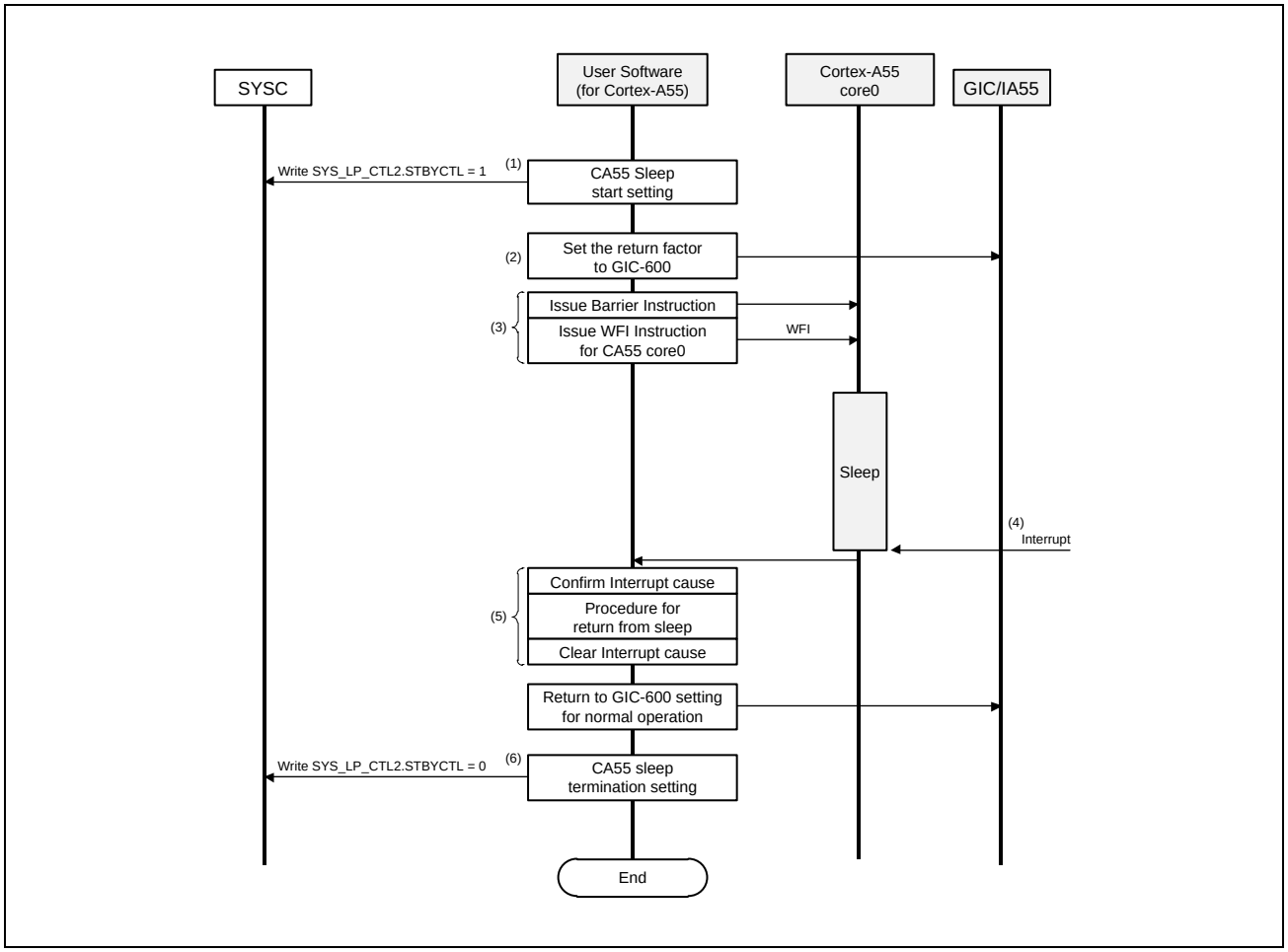



Figure 2.4 Sequence of a transition to Cortex-A55 Sleep Mode

(1) Set the STBYCTL bit of the SYS_LP_CTL2 register to 1 to start the Cortex-A55 Sleep Mode.

(2) Set the return factor to GIC. Enables only the interrupts used for return from the sleep mode and disables the interrupts used for the normal operation.

(3) Issue Barrier instruction.
Issue the WFI instruction for Cortex-A55 core0. The Cortex-A55 core0 is put into the sleep mode.

(4) Return from the sleep mode when the event occurs or the interrupt set in GIC occurs.

(5) Confirm the interrupt cause and perform the processing required for returning from the sleep mode.
And then clear the interrupt cause.
Set the end of interrupt to GIC.
Set the interrupt causes for the normal operation to GIC.

(6) Clear the STBYCTL bit of the SYS_LP_CTL2 register.

## 2.4 Function Reference

For details on the functions of the DynamIQ™ Shared Unit and Cortex-A55, see the documents listed below.

- Arm® DynamIQ Shared Unit Technical Reference Manual
- Arm® Cortex®-A55 Core Technical Reference Manual
- Arm® Cortex®-A55 Core Advanced SIMD and Floating-point Support Technical Reference Manual
- Arm® Cortex®-A55 Core Cryptographic Extension Technical Reference Manual

For details on clock and the controls, see the chapter on the clock pulse generator (CPG).

# 4. Boot Mode

## 4.1 Overview

This LSI has five boot modes.

Boot mode 3: Booting from the 1.8-V Single or Quad serial NOR flash memory connected to the SPI multi I/O bus space

Boot mode 4: Booting from the 3.3-V Single or Quad serial NOR flash memory connected to the SPI multi I/O bus space

Boot mode 5: Booting from the program downloaded through the serial communications with FIFO (SCIF)

Boot mode 6: Booting from the 1.8-V Single or Quad serial NAND flash memory connected to the SPI multi I/O bus space

Boot mode 7: Booting from the 3.3-V Single or Quad serial NAND flash memory connected to the SPI multi I/O bus space

Once this LSI is released from the system reset state, the clock pulse generator (CPG) executes a specified sequence, and then Cortex-A55 Core 0 is started first to boot from the device selected according to the settings through the MD_BOOT2 to MD_BOOT0 pins. The values of the MD_BOOT2 to MD_BOOT0 pins are read once the LSI is released from the system reset state.

Table 4.1 Selection of Boot Mode

| MD_BOOT2 to MD_BOOT0 | | | Boot Mode | Interface Module | Connected Device | Reference Section |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | Reserved | — | Reserved | — |
| 0 | 0 | 1 | Reserved | — | Reserved | — |
| 0 | 1 | 0 | Reserved | — | Reserved | — |
| 0 | 1 | 1 | Boot mode 3 | SPIBSC | 1.8-V Single or Quad serial NOR flash memory | **4.2.1** |
| 1 | 0 | 0 | Boot mode 4 | SPIBSC | 3.3-V Single or Quad serial NOR flash memory | **4.2.2** |
| 1 | 0 | 1 | Boot mode 5 | SCIF0 | Downloading through SCIF | **4.2.3** |
| 1 | 1 | 0 | Boot mode 6 | SPIBSC | 1.8-V Single or Quad serial NAND flash memory | **4.2.4** |
| 1 | 1 | 1 | Boot mode 7 | SPIBSC | 3.3-V Single or Quad serial NAND flash memory | **4.2.5** |

## 4.2 Operation

### 4.2.1 Boot Mode 3 (1.8-V Single or Quad Serial NOR Flash Memory)

**Table 4.2** shows the interface signals used for connection with the external device in boot mode 3 (1.8-V serial NOR flash memory).

Table 4.2 External Interface Signals Used in Boot Mode 3

| Interface Module | Pin Name | I/O | Function | Pin Type |
|---|---|---|---|---|
| QSPI0 | QSPI0_SPCLK | Input/output | QSPI clock | Dedicated pins (1.8 V) |
| | QSPI0_IO0 | Input/output | QSPI data 0 | |
| | QSPI0_IO1 | Input/output | QSPI data 1 | |
| | QSPI0_IO2 | Input/output | QSPI data 2 | |
| | QSPI0_IO3 | Input/output | QSPI data 3 | |
| | QSPI0_SSL | Output | Slave select | |

#### 4.2.1.1 External Connections

**Figure 4.1** and **Figure 4.2** show the connections with the Single SPI flash memory and Quad SPI flash memory, respectively.

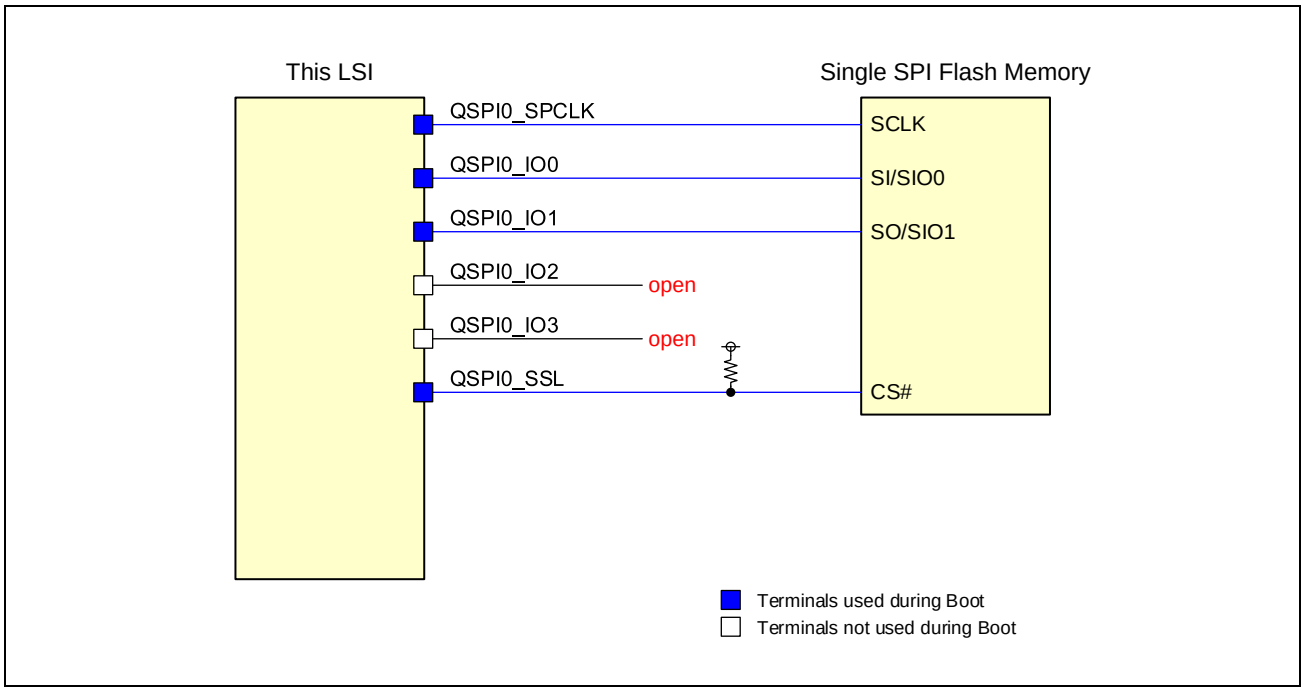



Figure 4.1 Connections with Single SPI Flash Memory in Boot Mode 3

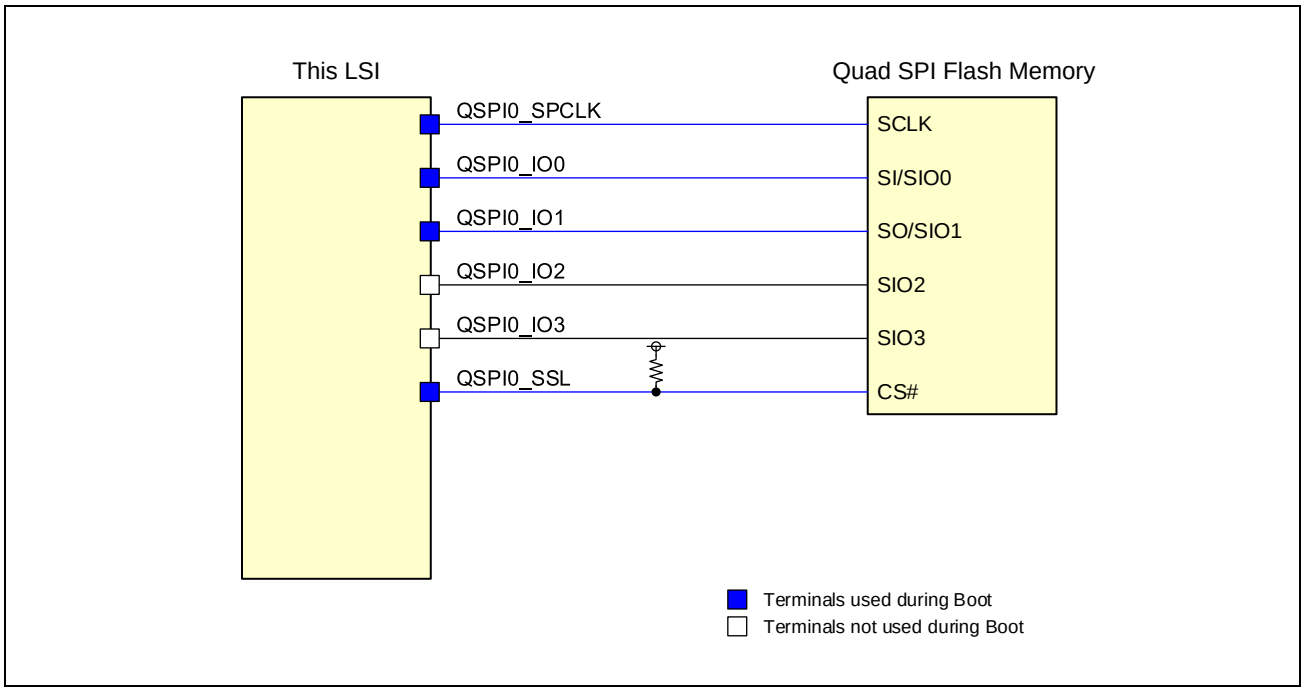



Figure 4.2 Connections with Quad SPI Flash Memory in Boot Mode 3

#### 4.2.1.2 Overview of Operation

This mode is used to boot this LSI from the user program stored in the Single or Quad flash memory device.

The serial flash memory mounted in advance on the board is used in this mode. The target device operates at 1.8 V and the I/O power of this LSI is also fixed to 1.8 V in this mode. The clock frequency at the startup is set to 16.6 MHz. The LSI is booted up through an SPI handshake with the Single or Quad flash memory device.

Upon detecting boot mode 3 according to the value read from the SYSC, the boot program makes the necessary settings of the SPI multi I/O bus controller to access memory. The boot program handles the following steps to control the booting process.

1. The CPG is set up to supply clocks, and the SPI multi I/O bus controller receives the operating clock and is released from the reset state.
2. The necessary information to control the I/O buffers is read from the registers in the SYSC.
   The given information is copied from the OTPC to the SYSC registers in advance.
3. The properties such as the driving ability of the I/O buffers of the signals to be used are set up.
4. After the above setup, the SPI multi I/O bus controller performs a handshake process with the target memory device.
5. When a Quad SPI handshake is required, the user program should control the process.

#### 4.2.1.3 Operation of Booting from 1.8-V Serial NOR Flash Memory

In boot mode 3 (booting from the serial flash memory), this LSI is booted from the loader program in the 1.8-V serial flash memory connected to the SPI multi I/O bus controller (SPIBSC).

The boot program executes the following processing to access the SPI multi I/O bus space.

1. Releasing the SPIBSC from module standby mode
2. Setting up the registers in the necessary peripheral modules (SPIBSC and GPIO)
3. Obtaining the size of the loader program data stored in the area starting from "the start address of the SPI multi I/O bus space (H’0_2000_0000) + offset" through the SPIBSC

After obtaining the size of the loader program data, the SPIBSC transfers the loader program stored in the area starting from "the start address of the SPI multi I/O bus space (H’0_2000_0000) + offset" to the addresses H’0_0001_2000 to H’0_0002_EFFF of the on-chip RAM for the obtained data size.

Then, execution branches to the start address (H’0_0001_2000) of the loader program transferred to the on-chip RAM to execute the loader program that was stored in the serial flash memory.

If the boot program has failed to read data, the boot program enters fail-safe mode (SCIF downloading mode). If the fail-safe processing has failed, execution enters an infinite loop in the on-chip ROM and the boot processing is terminated.

#### 4.2.1.4 Allocation of the Loader Program in the Serial NOR Flash Memory

**Figure 4.3** shows the allocation of the loader program size block and loader program in the serial flash memory.

In the loader program size block, the loader program size is stored in the first four bytes and the signature H'AA55 is stored in the last two bytes. **Figure 4.4** shows the structure of the loader program size data. Be sure to store the loader program size in 4-byte little endian.

The loader program is transferred from the area starting from the address "H'0_2000_0000 + H'0200 (offset)" to the addresses H'0_0001_2000 to H'0_0002_EFFF of the on-chip RAM for the obtained loader program size.

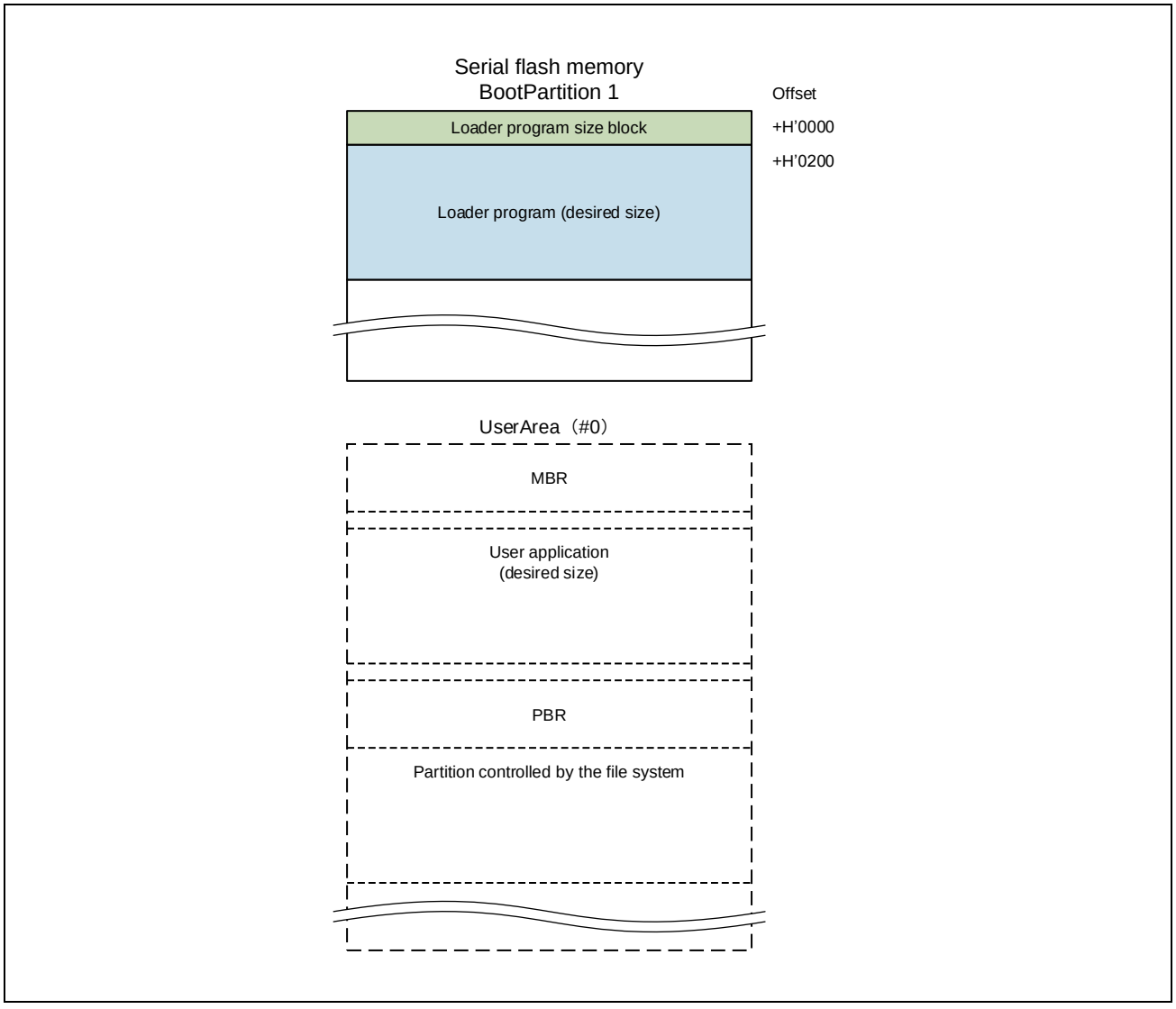



Figure 4.3 Allocation of Loader Program in the Serial Flash Memory Device

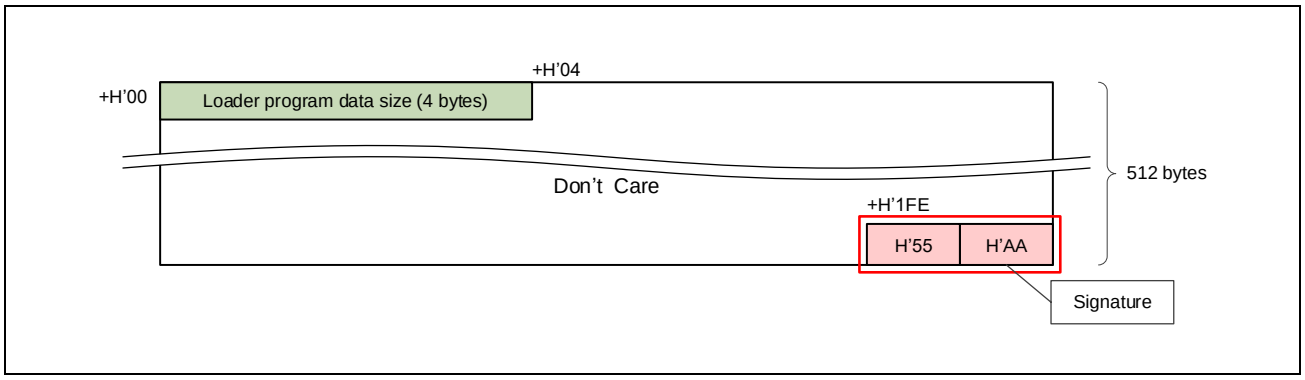



Figure 4.4 Structure of the Loader Program Size Block

### 4.2.2 Boot Mode 4 (3.3-V Single or Quad Serial NOR Flash Memory)

**Table 4.3** shows the interface signals used for connection with the external device in boot mode 4 (3.3-V serial NOR flash memory).

Table 4.3 External Interface Signals Used in Boot Mode 4

| Interface Module | Pin Name | I/O | Function | Pin Type |
|---|---|---|---|---|
| QSPI0 | QSPI0_SPCLK | Input/output | QSPI clock | Dedicated pins (3.3 V) |
| | QSPI0_IO0 | Input/output | QSPI data 0 | |
| | QSPI0_IO1 | Input/output | QSPI data 1 | |
| | QSPI0_IO2 | Input/output | QSPI data 2 | |
| | QSPI0_IO3 | Input/output | QSPI data 3 | |
| | QSPI0_SSL | Output | Slave select | |

### 4.2.2.1 External Connections

**Figure 4.5** and **Figure 4.6** show the connections with the 3.3-V Single SPI flash memory and 3.3-V Quad SPI flash memory, respectively.

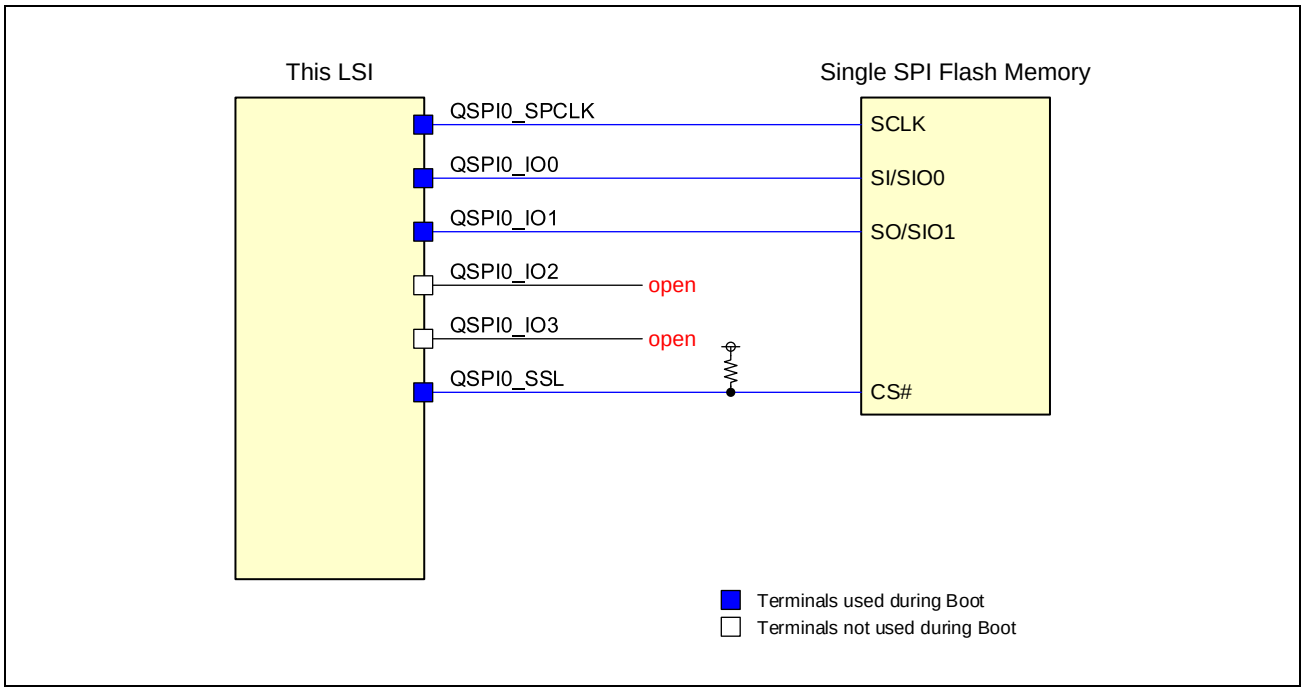



Figure 4.5 Connections with Single SPI Flash Memory in Boot Mode 4

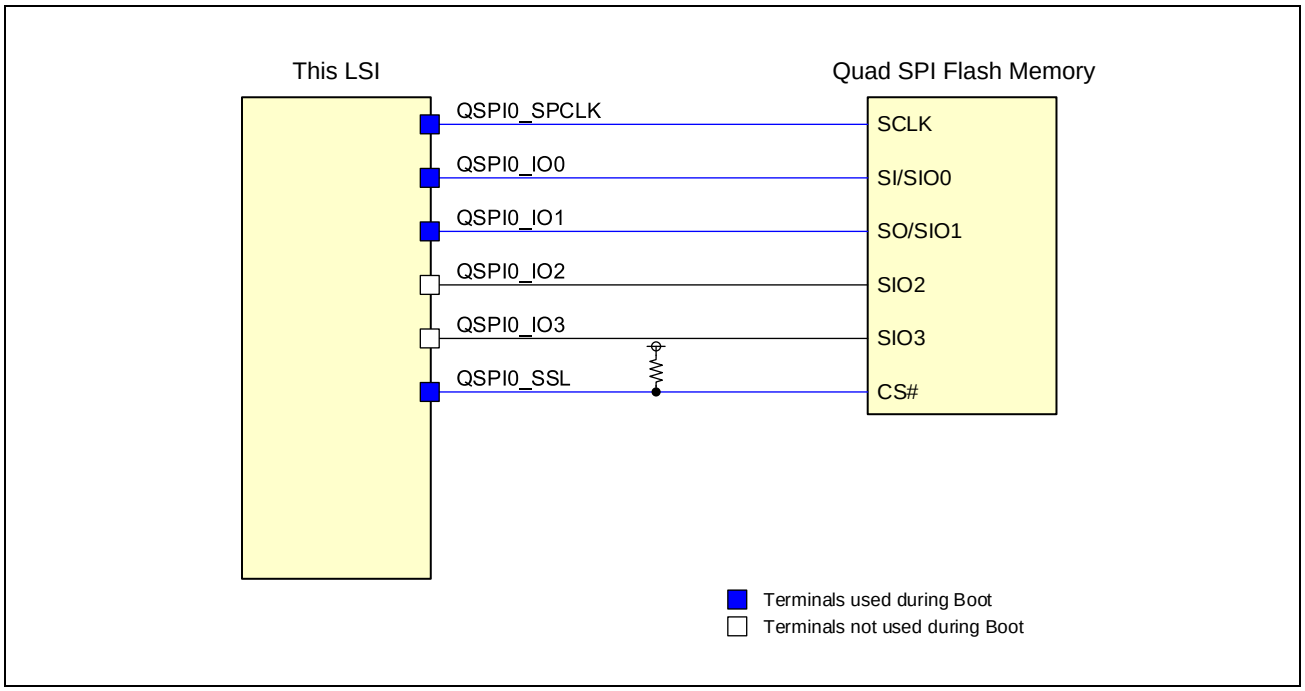



Figure 4.6 Connections with Quad SPI Flash Memory in Boot Mode 4

#### 4.2.2.2 Overview of Operation

This mode is used to boot this LSI from the user program stored in the Single or Quad SPI flash memory device. The serial flash memory mounted in advance on the board is used in this mode. The target device operates at 3.3 V and the I/O power of this LSI is also fixed to 3.3 V in this mode. The clock frequency at the startup is set to 16.6 MHz.

Upon detecting boot mode 4 according to the value read from the SYSC, the boot program makes the necessary settings of the SPI multi I/O bus controller to access memory. The boot program handles the following steps to control the booting process.

4. The CPG is set up to supply clocks, and the SPI multi I/O bus controller receives the operating clock and is released from the reset state.
5. The necessary information to control the I/O buffers is read from the registers in the SYSC.
   The information is copied from the OTPC to the SYSC registers in advance.
6. The properties such as the driving ability of the I/O buffers of the signals to be used are set up.
7. After the above setup, the SPI multi I/O bus controller performs a handshake process with the target memory device.
8. When a Quad SPI handshake is required, the user program should control the process.

#### 4.2.2.3 Operation of Booting from 3.3-V Serial NOR Flash Memory

Boot mode 4 (booting from 3.3-V serial flash memory) only differs from boot mode 3 (booting from 1.8-V serial flash memory) in the setting of supplied voltage as shown in **Table 4.4**.

Table 4.4 Resources Used by the Boot Program for 3.3-V Serial Flash Memory (Difference from 1.8-V Serial Flash Memory)

| Register Name | Address | Initial Value | Value after Booting | Remarks |
|---|---|---|---|---|
| GPIO_QSPI | H'0_1103_3008 | H'0000_0000 | H'0000_0000 | QSPI_PVDD: 0 = 3.3 V<br>In boot mode 3 or 4, this is automatically set by hardware. The user does not need to set this register. |

#### 4.2.2.4 Note

When interface 0 of the SPI multi I/O bus controller is connected with the SPI or QSPI memory, interface 1 can only be used to connect the same type of memory.

### 4.2.3 Boot Mode 5 (SCIF Downloading)

**Table 4.5** shows the interface signals used for connection with the external device in boot mode 5 (SCIF downloading).

Table 4.5 External Interface Signals Used in Boot Mode 5

| Interface Module | Pin Name | I/O | Function | Pin Type |
|---|---|---|---|---|
| SCIF0 | SCIF0_SCK*1 | Input/output | SCIF0 serial clock | Multiplexed pins (3.3 V) |
| | SCIF0_RXD | Input | SCIF0 receive data | |
| | SCIF0_TXD | Output | SCIF0 transmit data | |
| | SCIF0_CTS#*1 | Input/output | SCIF0 transmission enable | |
| | SCIF0_RTS#*1 | Output | SCIF0 transmission request | |

Note 1. This terminal is not used at boot time.

#### 4.2.3.1 External Connections

**Figure 4.7** shows the external connections for downloading through the SCIF.

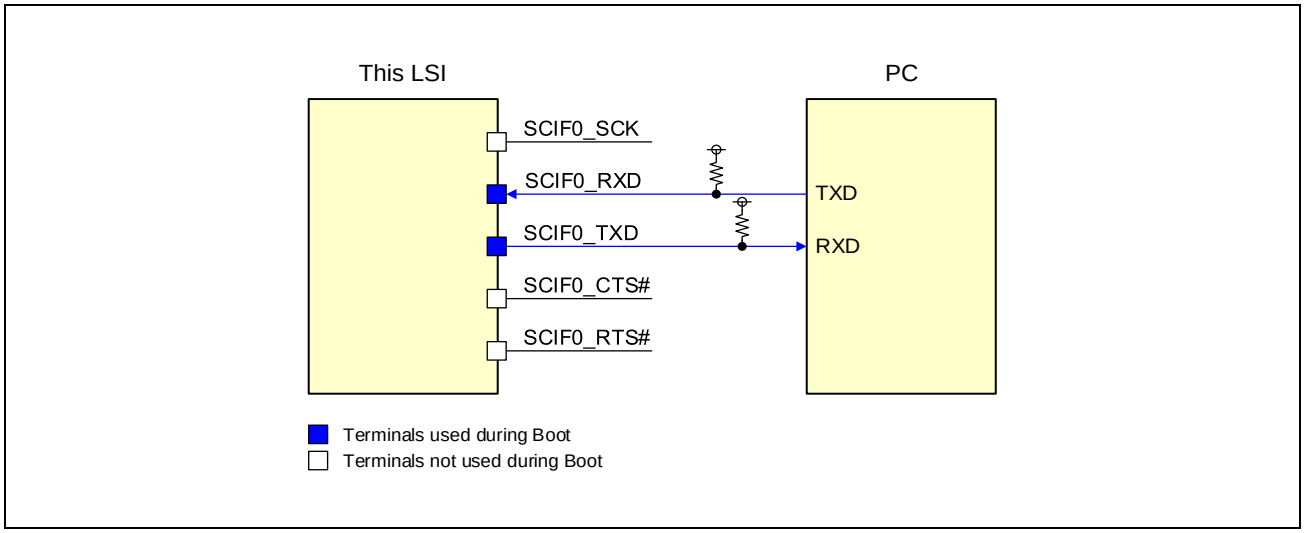



Figure 4.7 Connections for Downloading through the SCIF in Boot Mode 5

#### 4.2.3.2 Overview of Operation

This mode is used to download the user program from the host PC. Upon detecting boot mode 5 according to the value read from the SYSC, the boot program makes the necessary settings of the SCIF0 module to access memory. The boot program handles the following steps to control the booting process.

1. The SCIF0 module is set up to handle communications as shown below. Asynchronous communications proceed in this mode. The SCIF0_RXD and SCIF0_TXD pins are used.
   Baud rate: 115200 bps
   Data length: 8 bits
   Stop bit: 1 bit
   Flow control: None
   Data format: Motorola S-record
2. The sequence shown in **Figure 4.8** is used to download the user program through the SCIF.

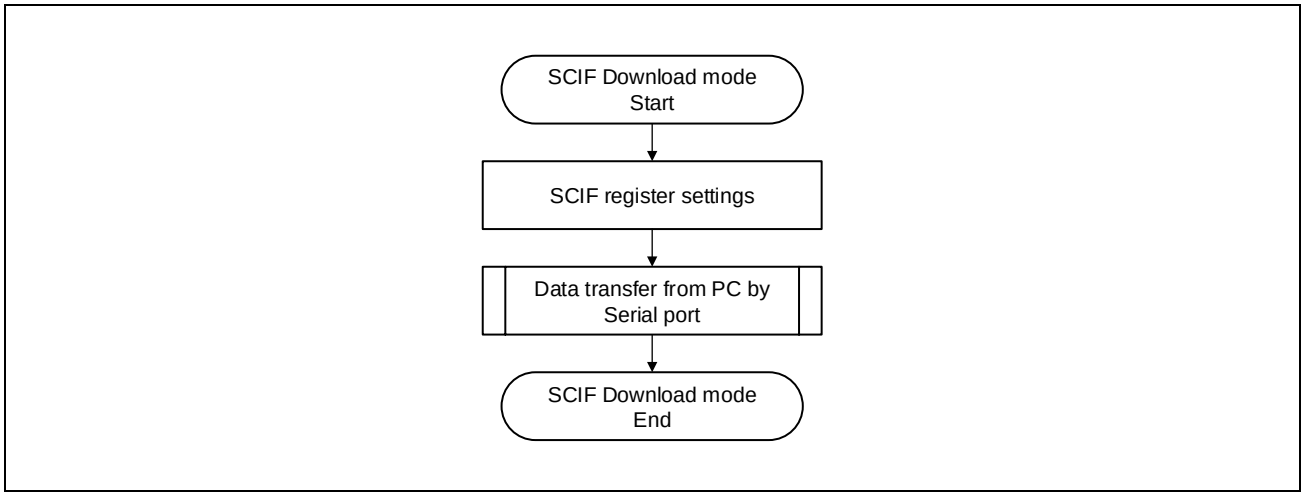



Figure 4.8 Sequence of Downloading through the SCIF0

#### 4.2.3.3 Operation of Booting through SCIF Downloading

In boot mode 5 (booting through SCIF downloading), this LSI is booted from the loader program downloaded from the external host PC through the serial communications interface with FIFO (SCIFA).

The boot program executes the following processing.

1. Setting up the necessary peripheral modules (SCIFA channel 0 and GPIO) and specifying the parameters for communications.

   When the LSI becomes ready for data reception, the boot program outputs the following message to the SCIFA.
   "SCIF Download mode"

   If execution enters boot mode 5 due to the fail-safe processing, the boot program outputs the following message to the SCIFA.
   "SCIF Download mode due to parameter error"

2. Upon receiving the Motorola S0 record sent from the external host PC through the SCIFA, the boot program converts the S3 record to binary data and copies the data to the addresses H'0_0001_2000 to H'0_0002_EFFE (up to 116 Kbytes) of the on-chip RAM.

   The boot program outputs the following message to the SCIFA.
   "-- Load Program to System RAM ----------"

   **Note:** The S0 record must always be generated because the boot program requires an S0 record as a trigger for the reception processing.

3. Upon receiving the S7 record, the boot program outputs the following message to the SCIFA and terminates copying to the on-chip RAM. After that, execution branches to the address H'0_0001_2000 of the on-chip RAM.
   "-- Start Boot Program on System RAM ---------"

   If the data in the Motorola S records sent from the external host PC satisfies any of the following error conditions, the corresponding error message is output to the SCIFA and copying to the on-chip RAM is aborted.

   — An S1 or S2 record is received: "Invalid Record Type Error!!! "

   — An address outside the on-chip RAM area is specified: "Address Error!!! "

   — An illegal character code is found: "Invalid Character Error!!! "

   — An illegal byte count is found: "Invalid Byte Count Error!!! "

The data transmission and reception operation in SCIF communications (asynchronous) is shown in **Figure 4.9**.

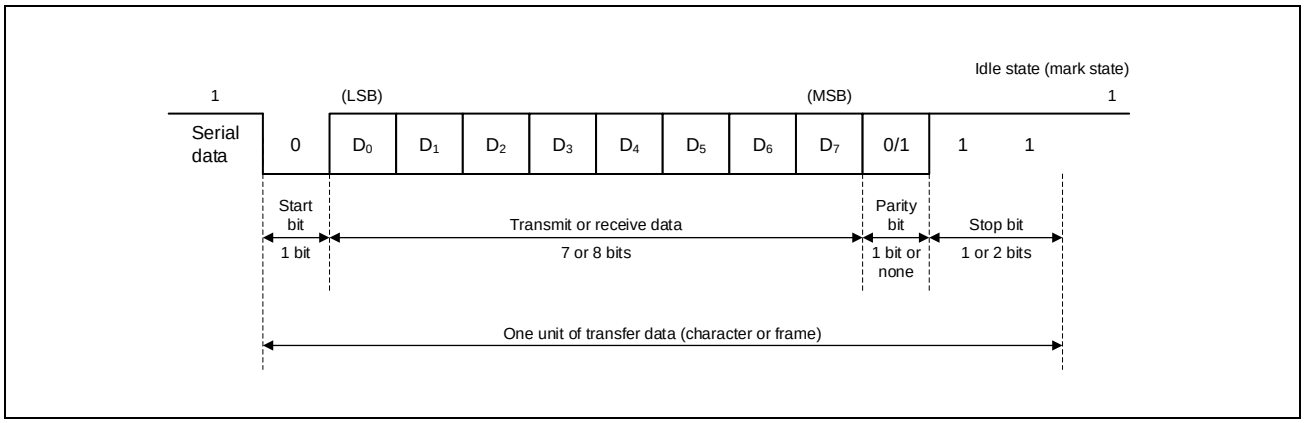



Figure 4.9 Data Transmission and Reception Operation in SCIF Communications

### 4.2.3.4 Allocation of the Loader Program for SCIF Downloading

**Figure 4.10** shows the allocation of the loader program copied to the on-chip RAM.

In the loader program size block (512 bytes), the loader program size is stored in the first four bytes and the signature H'AA55 is stored in the last two bytes. Be sure to store the loader program size in 4-byte little endian.

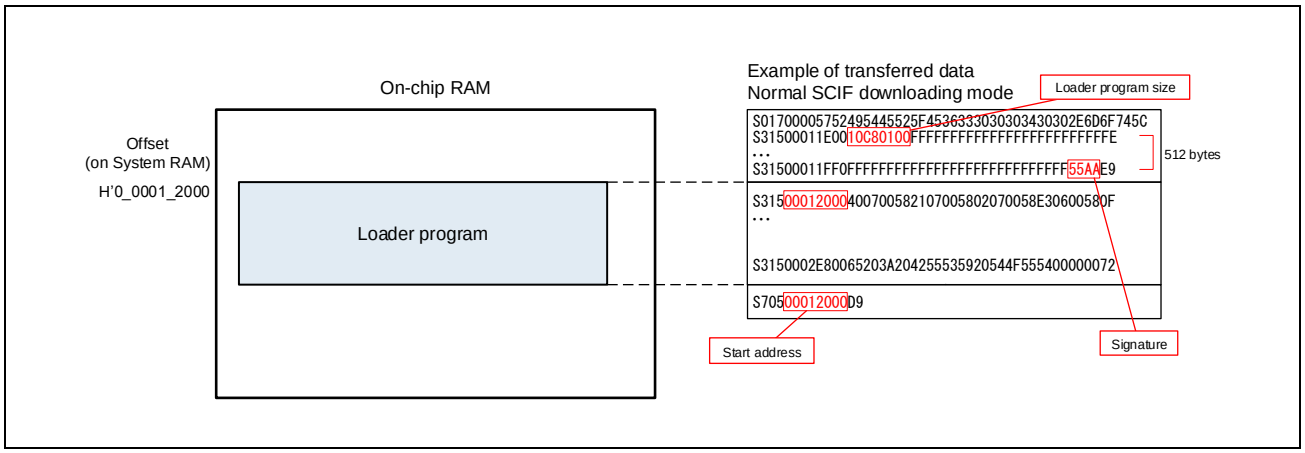



Figure 4.10 Allocation of the Loader Program for SCIF Downloading Copied to the On-chip RAM

### 4.2.3.5 Note

The pins assigned to the SCIF0_SCK, SCIF0_CTS, and SCIF0_RTS signals, which are not used in this mode, must not be changed from the initial settings (GPIO pins). If such a pin is set to operate as an SCIF0 pin, it becomes an input pin or an input/output pin. In this case, when the pin is externally placed in the Hi-Z state, this LSI cannot communicate with the host PC correctly.

To use the SCIFA for normal serial communications after booting, the SCIF0_SCK, SCIF0_CTS, and SCIF0_RTS pins should be pulled up or down in accordance with the communication mode and the driver software specifications.

### 4.2.4 Boot Mode 6 (1.8-V Single or Quad Serial NAND Flash Memory)

**Table 4.6** shows the interface signals used for connection with the external device in boot mode 6 (1.8-V serial NAND flash memory).

Table 4.6 External Interface Signals Used in Boot Mode 6

| Interface Module | Pin Name | I/O | Function | Pin Type |
|---|---|---|---|---|
| QSPI0 | QSPI0_SPCLK | Input/output | QSPI clock | Dedicated pins (1.8 V) |
| | QSPI0_IO0 | Input/output | QSPI data 0 | |
| | QSPI0_IO1 | Input/output | QSPI data 1 | |
| | QSPI0_IO2 | Input/output | QSPI data 2 | |
| | QSPI0_IO3 | Input/output | QSPI data 3 | |
| | QSPI0_SSL | Output | Slave select | |

### 4.2.4.1 External Connections

**Figure 4.11** and **Figure 4.12** show the connections with the Single SPI flash memory and Quad SPI flash memory, respectively.

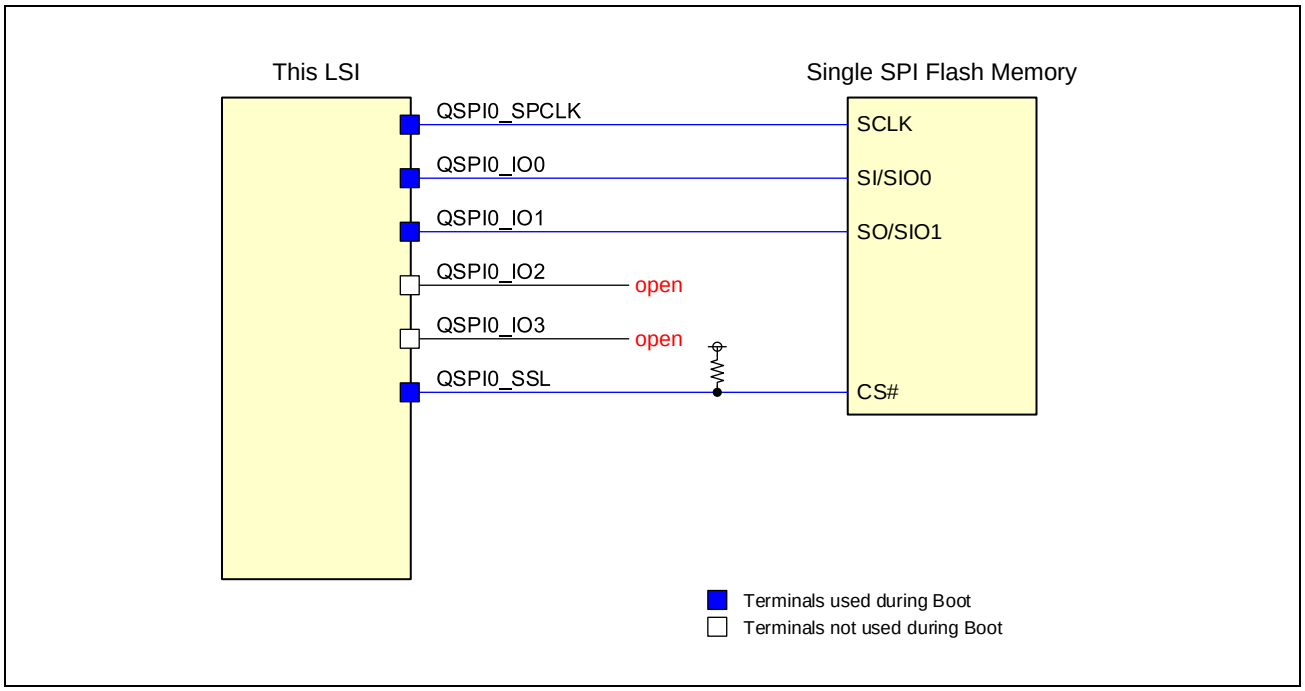



Figure 4.11 Connections with Single SPI Flash Memory in Boot Mode 6

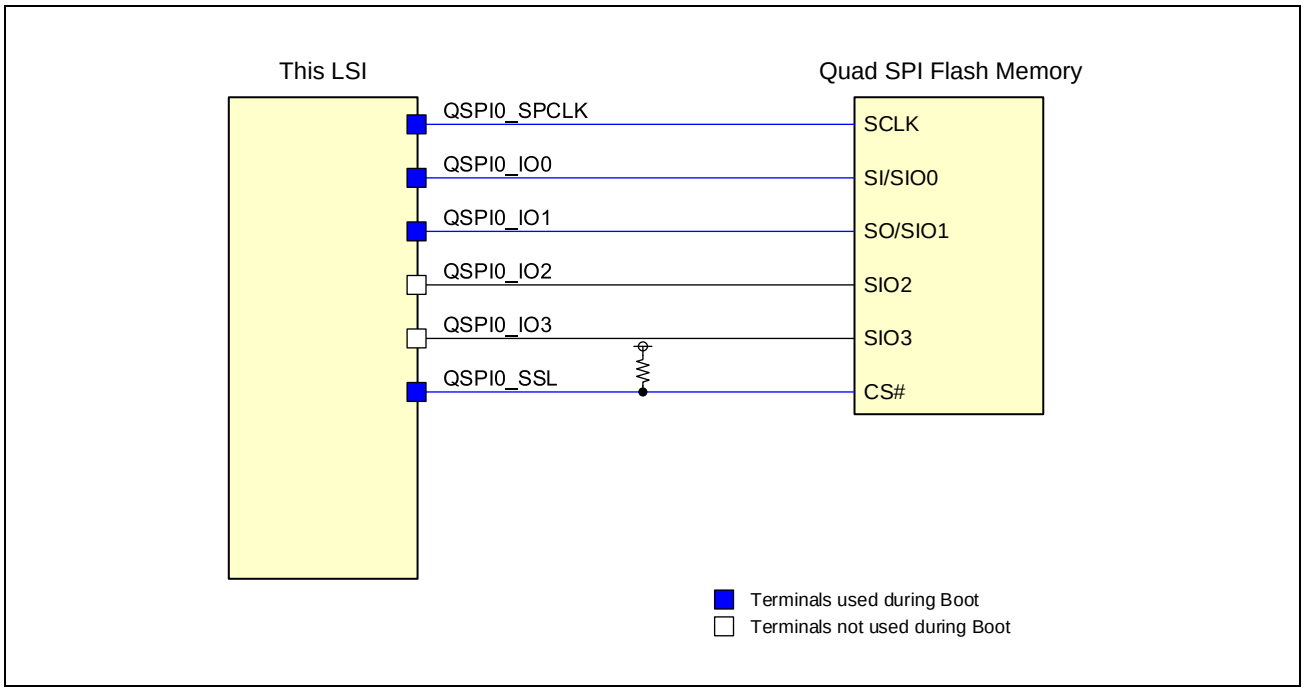



Figure 4.12 Connections with Quad SPI Flash Memory in Boot Mode 6

#### 4.2.4.2 Overview of Operation

This mode is used to boot this LSI from the user program stored in the Single or Quad flash memory device.

The serial flash memory mounted in advance on the board is used in this mode. The target device operates at 1.8 V and the I/O power of this LSI is also fixed to 1.8 V in this mode. The clock frequency at the startup is set to 16.6 MHz. The LSI is booted up through an SPI handshake with the Single or Quad flash memory device.

Upon detecting boot mode 6 according to the value read from the SYSC, the boot program makes the necessary settings of the SPI multi I/O bus controller to access memory. The boot program handles the following steps to control the booting process.

1. The CPG is set up to supply clocks, and the SPI multi I/O bus controller receives the operating clock and is released from the reset state.
2. The necessary information to control the I/O buffers is read from the registers in the SYSC.
   The given information is copied from the OTPC to the SYSC registers in advance.
3. The properties such as the driving ability of the I/O buffers of the signals to be used are set up.
4. After the above setup, the SPI multi I/O bus controller performs a handshake process with the target memory device.
5. When a Quad SPI handshake is required, the user program should control the process.

#### 4.2.4.3 Operation of Booting from 1.8-V Serial NAND Flash Memory

Boot mode 6 (Serial NAND Flash boot) boots from a loader program multiplexed into each Block on Serial NAND Flash memory (1.8V) connected to an SPI Multi I/O Bus Controller (SPIBSC).

For information on the deployment configuration of the loader program, refer to Section 4.2.4.4.

1) Release module standby of SPIBSC

2) Set the startup capability value at 1.8V

3) Get Serial NAND Flash device information

Table 4.7 Serial NAND Flash Device Information

| Parameter Name | Value | Description |
|---|---|---|
| bytes_page | 2kBytes | Bytes per Page |
| pages_block | 64pages | Pages per Block |
| loader_search_start_block | 0block | Starting Block Number of the Loader Program |
| loader_search_count | 4blocks | Number of blocks of loader program multiplexing |

4) Load the boot parameters from the first page of the Start Block into the SRAM programsize block destination (H'0_0001_1000). At this time, check the following contents to make sure that the boot parameters are valid.

- The exclusive OR between the first and second boot parameters and their respective inverted values is 1 for all bits.
- The first boot parameter and the second boot parameter are equal
- The loader size must be within the range (greater than zero and rounded up to the page size within the SRAM range).
- Signature must be H'AA55

5) Read the size of the page from the second page to the loader program destination of SRAM (H'0_0001_2000) and calculate the Hash value of the loader. Verify that it matches the Hash value stored in the boot parameters.

6) If the loader program check is successful, the load success flag and the loaded block number and loader program size are stored in the BootInfo area, the loading process is terminated, and the loader program is executed by branching to the first address of the loader program (H'0_0001_2000).

7) If the load fails, or the boot parameters or loader program check fails, load the loader for the next block. If all blocks of multiplexing fail to load, set an error flag in the BootInfo area as a boot failure and move to failsafe mode (SCIF download mode). If the failsafe process fails, an infinite loop occurs in the internal ROM and the boot process ends.

### 4.2.4.4 Allocation of the Loader Program in the Serial NAND Flash Memory

The boot parameters are written to the internal RAM H'0_0001_1000~H'0_0001_FFFF for the size (2 kB) according to the device used, and the loader program is written to the H'0_0001_2000~H' 0_0002_EFFF of the internal RAM for the size of the loader program.

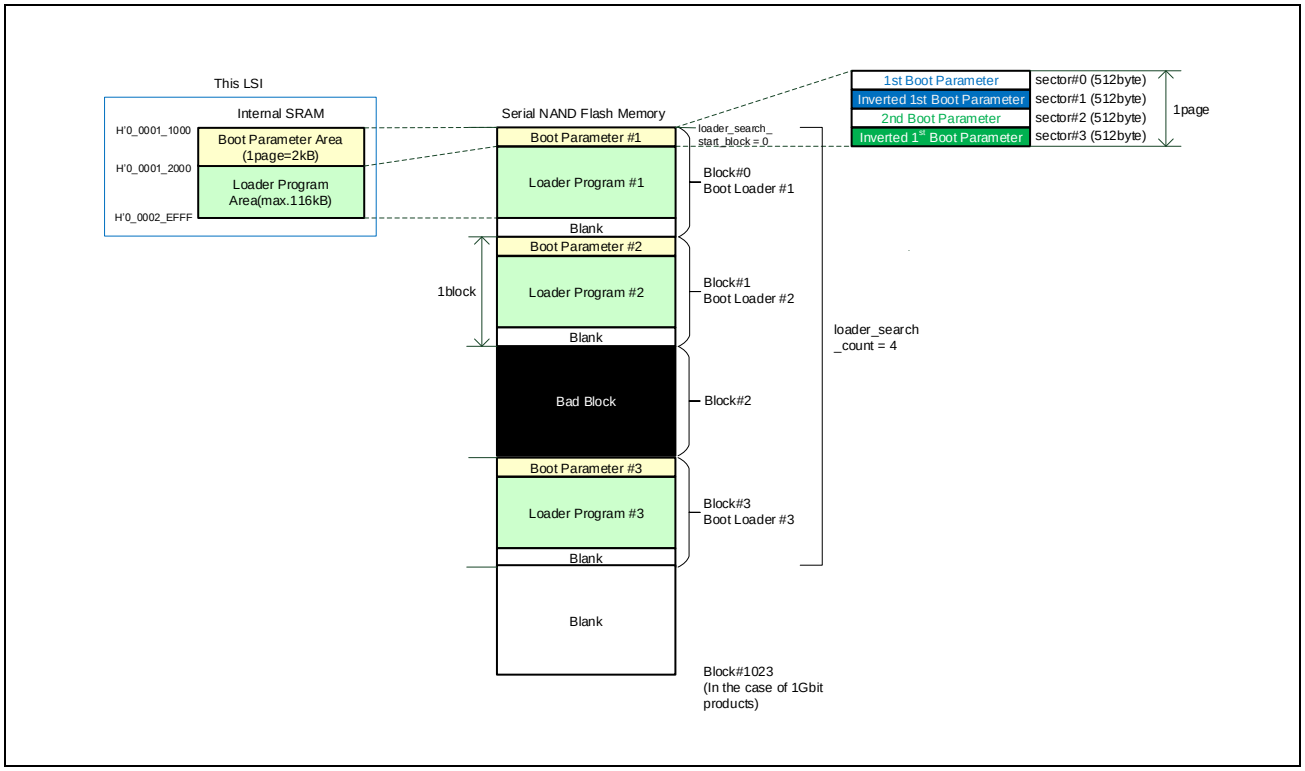



Figure 4.13 Allocation of Loader Program in the Serial NAND Flash Memory Device

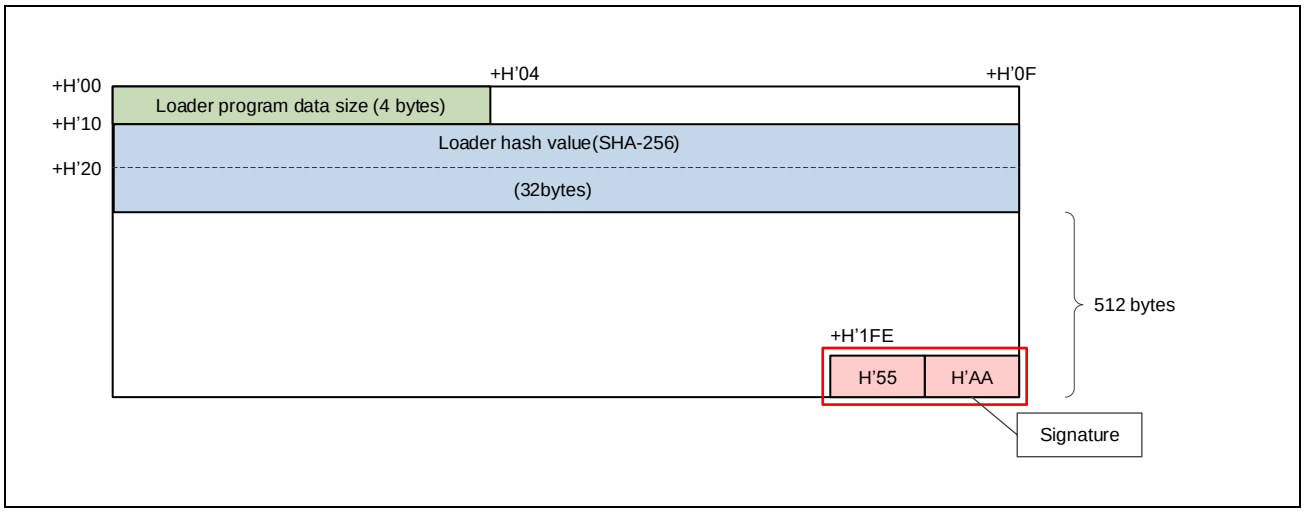



Figure 4.14 Structure of the Loader Program Size Block

### 4.2.4.5 Padding in SHA-256 Function

Serial NAND Flash boot uses the SHA-256 algorithm to calculate the hash value of the loader program.
When calculating, padding the target data is performed according to the following procedure.

1) When the data size is an integer multiple of the block size (256 bytes)
   - Added padding for 256byte
   - Store H'80 from the beginning of the padding (the next byte of the data) to the fourth byte
   - Data size stored at 57th byte and 61th~64th byte of padding
   - Zero except for the above
2) When the data size is not an integer multiple of the block size (256 bytes)
   - Add padding to increments of 256bytes
   - Stores the H'80 in the first byte of padding (the next byte of data)
   - When the fraction of the data size is 56 bytes or less, the data size is stored in the 57th byte and 61th~64th byte
   - If the data size fraction is 56 bytes or more, add another 256byte padding to store the data size at 57th byte and 61th~64th byte
   - Zero except for the above

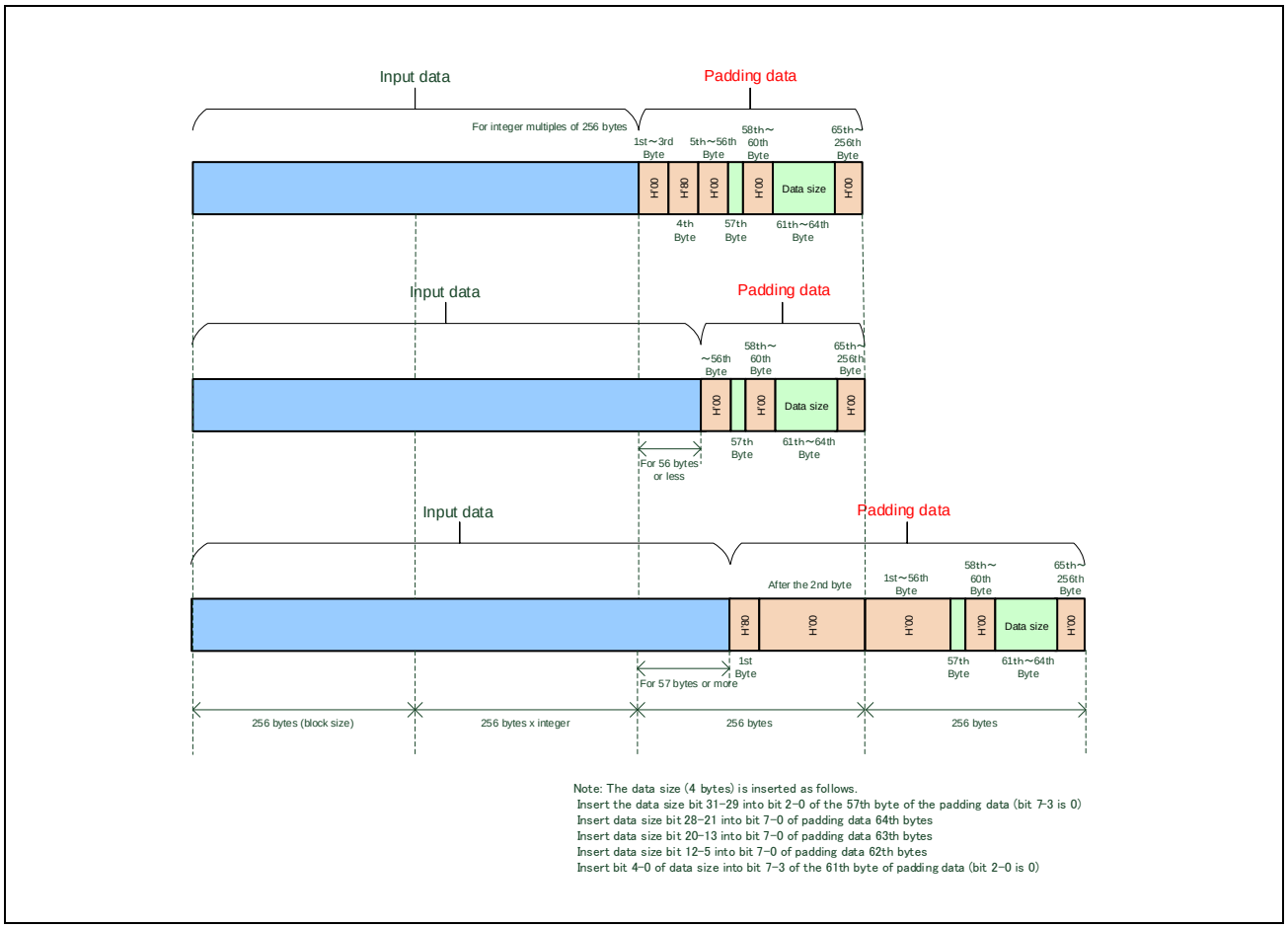



Figure 4.15 Image of padding processing in SHA-256

### 4.2.5 Boot Mode 7 (3.3-V Single or Quad Serial NAND Flash Memory)

**Table 4.8** shows the interface signals used for connection with the external device in boot mode 7 (3.3-V serial NAND flash memory).

Table 4.8 External Interface Signals Used in Boot Mode 7

| Interface Module | Pin Name | I/O | Function | Pin Type |
|---|---|---|---|---|
| QSPI0 | QSPI0_SPCLK | Input/output | QSPI clock | Dedicated pins (3.3 V) |
| | QSPI0_IO0 | Input/output | QSPI data 0 | |
| | QSPI0_IO1 | Input/output | QSPI data 1 | |
| | QSPI0_IO2 | Input/output | QSPI data 2 | |
| | QSPI0_IO3 | Input/output | QSPI data 3 | |
| | QSPI0_SSL | Output | Slave select | |

### 4.2.5.1 External Connections

**Figure 4.16** and **Figure 4.17** show the connections with the 3.3-V Single SPI flash memory and 3.3-V Quad SPI flash memory, respectively.

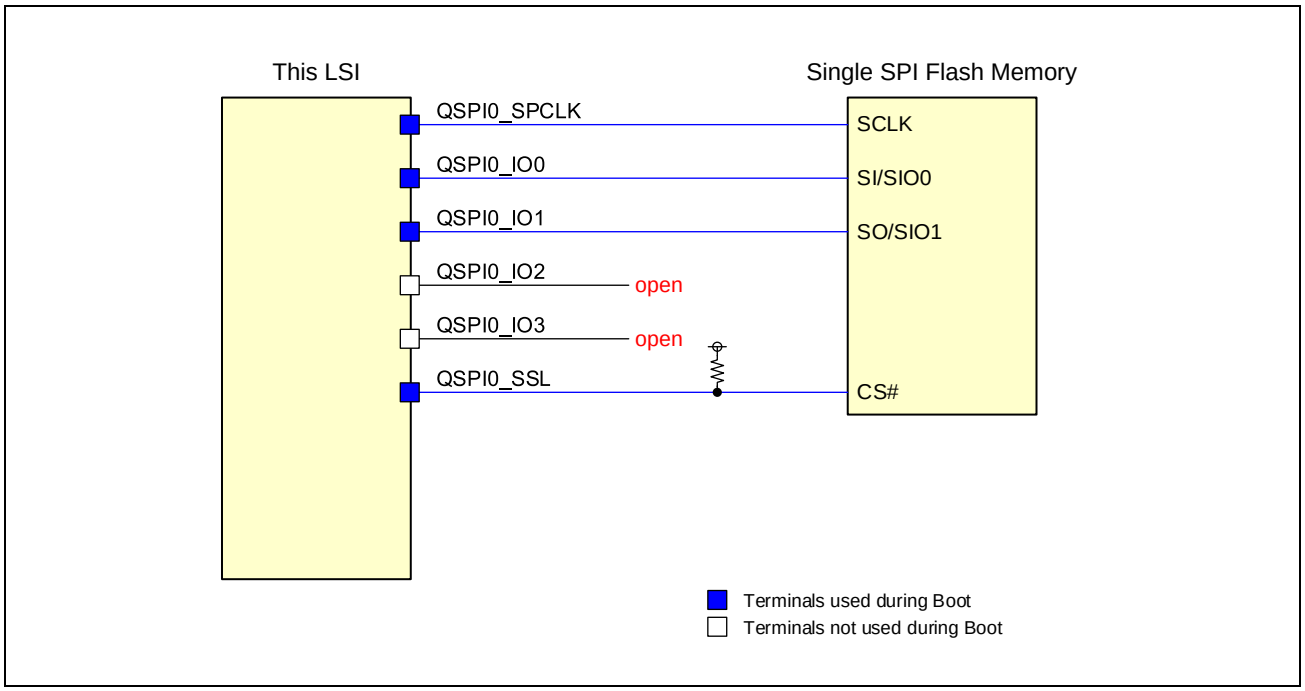



Figure 4.16 Connections with Single SPI Flash Memory in Boot Mode 7

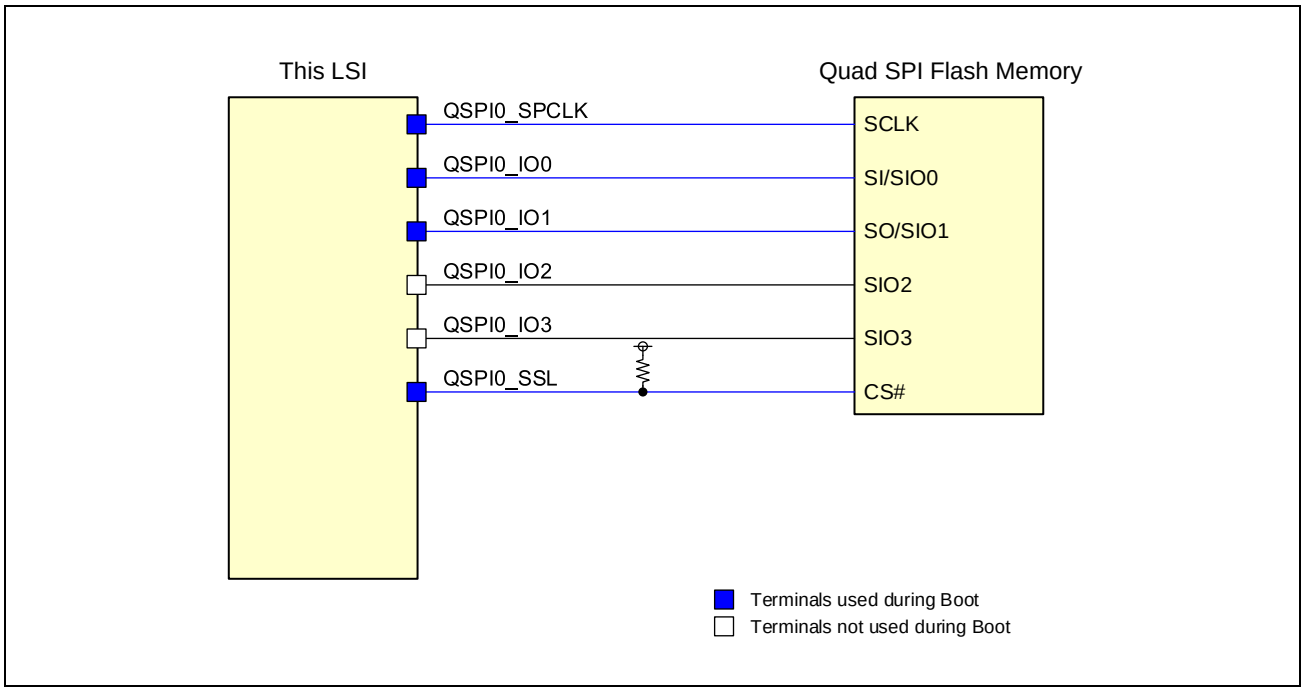



Figure 4.17 Connections with Quad SPI Flash Memory in Boot Mode 7

#### 4.2.5.2 Overview of Operation

This mode is used to boot this LSI from the user program stored in the Single or Quad SPI flash memory device. The serial flash memory mounted in advance on the board is used in this mode. The target device operates at 3.3 V and the I/O power of this LSI is also fixed to 3.3 V in this mode. The clock frequency at the startup is set to 16.6 MHz.

Upon detecting boot mode 7 according to the value read from the SYSC, the boot program makes the necessary settings of the SPI multi I/O bus controller to access memory. The boot program handles the following steps to control the booting process.

1. The CPG is set up to supply clocks, and the SPI multi I/O bus controller receives the operating clock and is released from the reset state.
2. The necessary information to control the I/O buffers is read from the registers in the SYSC.
   The information is copied from the OTPC to the SYSC registers in advance.
3. The properties such as the driving ability of the I/O buffers of the signals to be used are set up.
4. After the above setup, the SPI multi I/O bus controller performs a handshake process with the target memory device.
5. When a Quad SPI handshake is required, the user program should control the process.

#### 4.2.5.3 Operation of Booting from 3.3-V Serial NAND Flash Memory

Boot mode 7 (booting from 3.3-V serial NAND flash memory) only differs from boot mode 6 (booting from 1.8-V serial NAND flash memory) in the setting of supplied voltage as shown in **Table 4.9**

Table 4.9 Resources Used by the Boot Program for 3.3-V Serial NAND Flash Memory (Difference from 1.8-V Serial NAND Flash Memory)

| Register Name | Address | Initial Value | Value after Booting | Remarks |
|---|---|---|---|---|
| GPIO_QSPI | H'0_1103_3008 | H'0000_0000 | H'0000_0000 | QSPI_PVDD: 0 = 3.3 V |

# 5. LSI Internal Bus

## 5.1 Overview

### 5.1.1 Features

The bus system of this LSI provides a physical address space of 16 Gbytes (address bus width of 34 bits). The LSI internal bus of this LSI incorporates Arm CoreLink NIC-400, etc., and controls the following bus functions.

**Interrupt generation:**

AXI bus error interrupt generation

**Unit state detection:**

Slave unit stop state detection

### 5.1.2 Block Diagram of LSI Internal Bus

The LSI internal bus of this LSI consists of the ACPU bus, MCPU bus, and system bus. **Figure 5.1** shows the configuration of the buses.

**ACPU bus:**

A bus connected to Cortex-A55, DDR memory controllers, image processing units, and Storage and Network

**MCPU bus:**

A bus connected to serial interface units

**System bus:**

A bus connected to the control registers of each unit

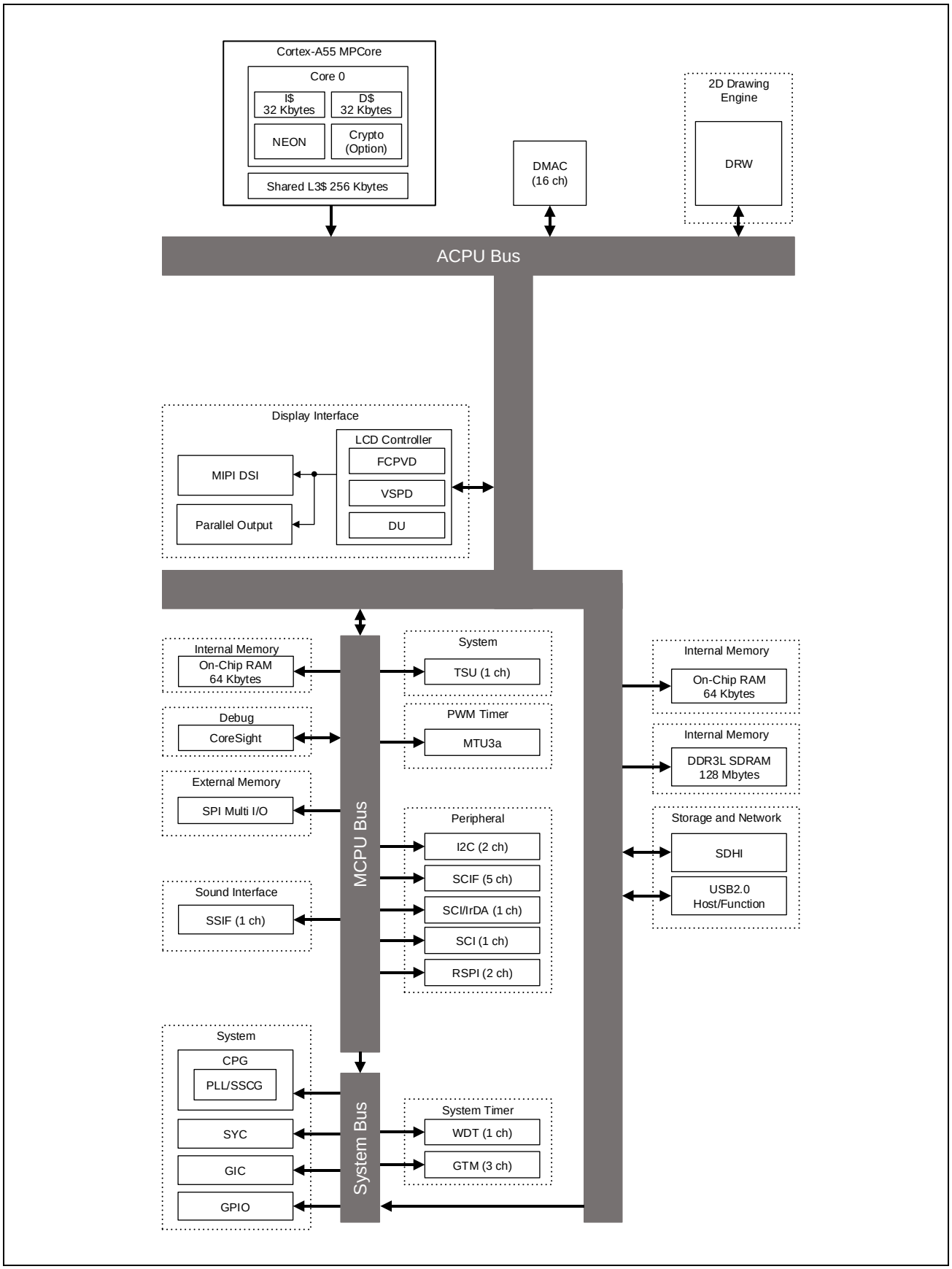



Figure 5.1 Configuration of LSI Internal Bus

## 5.2 Area Maps

### 5.2.1 Overall Address Space

**Figure 5.2** shows the overall address space of this LSI and **Table 5.1** shows the detailed address space.

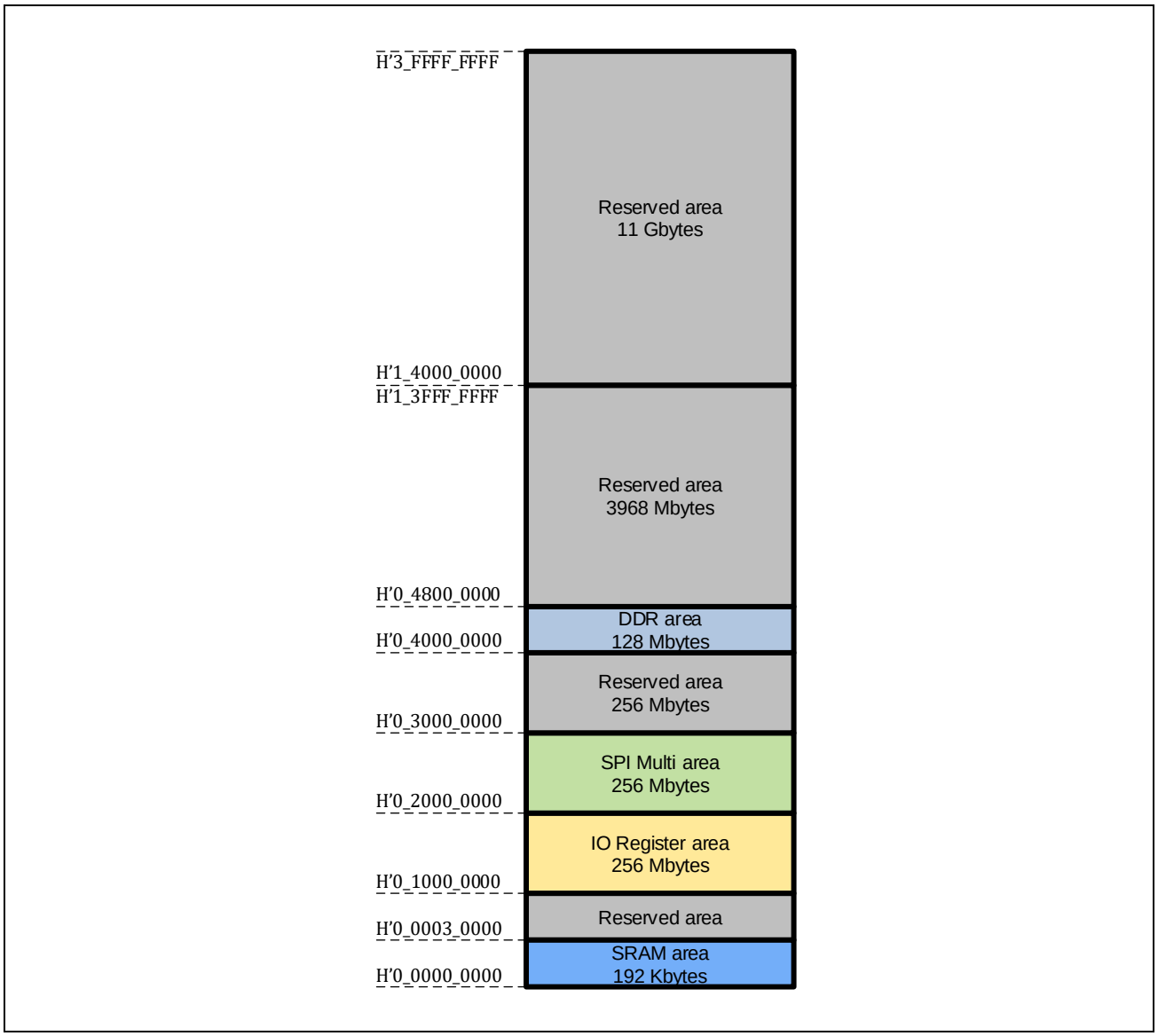



Figure 5.2 Overall Address Space

Table 5.1 Detailed Address Space (1/3)

| Start Address | End Address | Size | Space | Remarks |
|---|---|---|---|---|
| H'1_4000_0000 | H'3_FFFF_FFFF | 11 Gbytes | Reserved | *1 |
| H'0_4800_0000 | H'1_3FFF_FFFF | 3968 Mbytes | Reserved | *1 |
| H'0_4000_0000 | H'0_47FF_FFFF | 128 Mbytes | DDR (Memory) | *2 |
| H'0_3000_0000 | H'0_3FFF_FFFF | 256 Mbytes | Reserved | *1 |
| H'0_2000_0000 | H'0_2FFF_FFFF | 256 Mbytes | SPI Multi I/O (Memory) | |
| H'0_1500_0000 | H'0_1FFF_FFFF | 176 Mbytes | Reserved | *1 |
| H'0_14F0_0000 | H'0_14FF_FFFF | 1 Mbyte | Reserved | *1 |
| H'0_14C0_0000 | H'0_14EF_FFFF | 3 Mbytes | Reserved | *1 |
| H'0_14B0_0000 | H'0_14BF_FFFF | 1 Mbyte | Reserved | *1 |
| H'0_1400_0000 | H'0_14AF_FFFF | 11 Mbytes | Reserved | *1 |
| H'0_1300_0000 | H'0_13FF_FFFF | 16 Mbytes | Reserved | *1 |
| H'0_12C0_0000 | H'0_12FF_FFFF | 4 Mbytes | Reserved | *1 |
| H'0_12B0_0000 | H'0_12BF_FFFF | 1 Mbyte | Reserved | *1 |
| H'0_12A0_0000 | H'0_12AF_FFFF | 1 Mbyte | Reserved | *1 |
| H'0_1280_1C00 | H'0_129F_FFFF | 2041 Kbytes | Reserved | *1 |
| H'0_1280_1800 | H'0_1280_1BFF | 1 Kbyte | GTM ch2 | |
| H'0_1280_1400 | H'0_1280_17FF | 1 Kbyte | GTM ch1 | |
| H'0_1280_1000 | H'0_1280_13FF | 1 Kbyte | GTM ch0 | |
| H'0_1280_0C00 | H'0_1280_0FFF | 1 Kbyte | Reserved | *1 |
| H'0_1280_0800 | H'0_1280_0BFF | 1 Kbyte | WDT CA55 core 0 | |
| H'0_1280_0400 | H'0_1280_07FF | 1 Kbyte | Reserved | *1 |
| H'0_1280_0000 | H'0_1280_03FF | 1 Kbyte | Reserved | *1 |
| H'0_1270_0000 | H'0_127F_FFFF | 1 Mbyte | Reserved | *1 |
| H'0_1240_0000 | H'0_126F_FFFF | 3 Mbytes | Reserved | *1 |
| H'0_1230_0000 | H'0_123F_FFFF | 1 Mbyte | Reserved | *1 |
| H'0_1200_0000 | H'0_122F_FFFF | 3 Mbytes | Reserved | *1 |
| H'0_11F0_0000 | H'0_11FF_FFFF | 1 Mbyte | Reserved | *1 |
| H'0_11C9_0000 | H'0_11EF_FFFF | 2496 Kbytes | Reserved | *1 |
| H'0_11C8_0000 | H'0_11C8_FFFF | 64 Kbytes | DRW | |
| H'0_11C7_0000 | H'0_11C7_FFFF | 64 Kbytes | Reserved | *1 |
| H'0_11C6_0000 | H'0_11C6_FFFF | 64 Kbytes | USB ch0 (OTG-Func) | |
| H'0_11C5_0000 | H'0_11C5_FFFF | 64 Kbytes | USB ch0 (OTG-Host) | |
| H'0_11C4_0000 | H'0_11C4_FFFF | 64 Kbytes | USBPHY Control | |
| H'0_11C3_0000 | H'0_11C3_FFFF | 64 Kbytes | Reserved | *1 |
| H'0_11C2_0000 | H'0_11C2_FFFF | 64 Kbytes | Reserved | *1 |
| H'0_11C1_0000 | H'0_11C1_FFFF | 64 Kbytes | Reserved | *1 |
| H'0_11C0_0000 | H'0_11C0_FFFF | 64 Kbytes | SD ch0 | |
| H'0_11B0_0000 | H'0_11BF_FFFF | 1 Mbyte | Reserved | *1 |
| H'0_11A0_0000 | H'0_11AF_FFFF | 1 Mbyte | Reserved | *1 |
| H'0_1190_0000 | H'0_119F_FFFF | 1 Mbyte | GIC | |
| H'0_1187_0000 | H'0_118F_FFFF | 1 Mbyte | Reserved | *1 |
| H'0_1186_0000 | H'0_1186_FFFF | 64 Kbytes | Reserved | *1 |
| H'0_1185_0000 | H'0_1185_FFFF | 64 Kbytes | Reserved | *1 |
| H'0_1184_0000 | H'0_1184_FFFF | 64 Kbytes | Reserved | *1 |
| H'0_1183_0000 | H'0_1183_FFFF | 64 Kbytes | Non Secure DMAC (DMAC_NS) APB | |

Table 5.1 Detailed Address Space (2/3)

| Start Address | End Address | Size | Space | Remarks |
|---|---|---|---|---|
| H'0_1182_0000 | H'0_1182_FFFF | 64 Kbytes | Non Secure DMAC (DMAC_NS) AXI | |
| H'0_1181_0000 | H'0_1181_FFFF | 64 Kbytes | Reserved | *1 |
| H'0_1180_0000 | H'0_1180_FFFF | 64 Kbytes | Reserved | *1 |
| H'0_1170_0000 | H'0_117F_FFFF | 1 Mbyte | Reserved | *1 |
| H'0_1143_0000 | H'0_116F_FFFF | 2880 Kbytes | Reserved | *1 |
| H'0_1142_0000 | H'0_1142_FFFF | 64 Kbyes | Reserved | *1 |
| H'0_1141_0000 | H'0_1141_FFFF | 64 Kbytes | DDR (Control Reg) | |
| H'0_1140_0000 | H'0_1140_FFFF | 64 Kbytes | DDR (PHY) | |
| H'0_1130_0000 | H'0_113F_FFFF | 1 Mbyte | Reserved | *1 |
| H'0_110C_0000 | H'0_112F_FFFF | 2304 Kbytes | Reserved | *1 |
| H'0_110B_0000 | H'0_110B_FFFF | 64 Kbytes | Reserved | *1 |
| H'0_110A_0000 | H'0_110A_FFFF | 64 Kbytes | IA55 (Interrupt controller) | |
| H'0_1109_0000 | H'0_1109_FFFF | 64 Kbytes | SRAM MCPU (Reg) | |
| H'0_1108_0000 | H'0_1108_FFFF | 64 Kbytes | SRAM ACPU (Reg) | |
| H'0_1107_0000 | H'0_1107_FFFF | 64 Kbytes | Reserved | *1 |
| H'0_1106_0000 | H'0_1106_FFFF | 64 Kbytes | Reserved | *1 |
| H'0_1105_0000 | H'0_1105_FFFF | 64 Kbytes | Reserved | *1 |
| H'0_1104_0000 | H'0_1104_FFFF | 64 Kbytes | Reserved | *1 |
| H'0_1103_0000 | H'0_1103_FFFF | 64 Kbytes | GPIO | |
| H'0_1102_0000 | H'0_1102_FFFF | 64 Kbytes | SYSC | |
| H'0_1101_0000 | H'0_1101_FFFF | 64 Kbytes | CPG | |
| H'0_1100_0000 | H'0_1100_FFFF | 64 Kbytes | SYC | |
| H'0_10C0_0000 | H'0_10FF_FFFF | 4 Mbytes | CST (CoreSight) | |
| H'0_10B0_0000 | H'0_10BF_FFFF | 1 Mbyte | Reserved | *1 |
| H'0_108A_0000 | H'0_10AF_FFFF | 2432 Kbytes | Reserved | *1 |
| H'0_1089_0000 | H'0_1089_FFFF | 64 Kbytes | DU | |
| H'0_1088_0000 | H'0_1088_FFFF | 64 Kbytes | FCPVD | |
| H'0_1087_0000 | H'0_1087_FFFF | 64 Kbytes | VSPD | |
| H'0_1086_0000 | H'0_1086_FFFF | 64 Kbytes | DSI (Controller) | |
| H'0_1085_0000 | H'0_1085_FFFF | 64 Kbytes | DSI (PHY) | |
| H'0_1084_0000 | H'0_1084_FFFF | 64 Kbytes | Reserved | *1 |
| H'0_1083_0000 | H'0_1083_FFFF | 64 Kbytes | Reserved | *1 |
| H'0_1082_0000 | H'0_1082_FFFF | 64 Kbytes | Reserved | *1 |
| H'0_1081_0000 | H'0_1081_FFFF | 64 Kbytes | Reserved | *1 |
| H'0_1080_0000 | H'0_1080_FFFF | 64 Kbytes | Reserved | *1 |
| H'0_1070_0000 | H'0_107F_FFFF | 1 Mbyte | Reserved | *1 |
| H'0_1060_0000 | H'0_106F_FFFF | 1 Mbyte | Reserved | *1 |
| H'0_1041_0000 | H'0_105F_FFFF | 1984 Kbytes | Reserved | *1 |
| H'0_1040_0000 | H'0_1040_FFFF | 64 Kbytes | Reserved | *1 |
| H'0_1030_0000 | H'0_103F_FFFF | 1 Mbyte | Reserved | *1 |
| H'0_1020_0000 | H'0_102F_FFFF | 1 Mbyte | Reserved | *1 |
| H'0_1009_0000 | H'0_101F_FFFF | 1472 Kbytes | Reserved | *1 |
| H'0_1008_0000 | H'0_1008_FFFF | 64 Kbytes | Reserved | *1 |
| H'0_1007_0000 | H'0_1007_FFFF | 64 Kbytes | SPI Multi I/O (Write Buf) | |
| H'0_1006_0000 | H'0_1006_FFFF | 64 Kbytes | SPI Multi I/O (Reg) | |

Table 5.1 Detailed Address Space (3/3)

| Start Address | End Address | Size | Space | Remarks |
|---|---|---|---|---|
| H'0_1005_9800 | H'0_1005_FFFF | 26 Kbytes | Reserved | *1 |
| H'0_1005_9400 | H'0_1005_97FF | 1 Kbyte | TSU | |
| H'0_1005_9000 | H'0_1005_93FF | 1 Kbyte | Reserved | *1 |
| H'0_1005_8C00 | H'0_1005_8FFF | 1 Kbyte | Reserved | *1 |
| H'0_1005_8800 | H'0_1005_8BFF | 1 Kbyte | Reserved | *1 |
| H'0_1005_8400 | H'0_1005_87FF | 1 Kbyte | I2C ch1 | |
| H'0_1005_8000 | H'0_1005_83FF | 1 Kbyte | I2C ch0 | |
| H'0_1005_0000 | H'0_1005_7FFF | 32 Kbytes | Reserved | *1 |
| H'0_1004_D800 | H'0_1004_FFFF | 10 Kbytes | Reserved | *1 |
| H'0_1004_D400 | H'0_1004_D7FF | 1 Kbyte | SCI ch1 | |
| H'0_1004_D000 | H'0_1004_D3FF | 1 Kbyte | SCI ch0 | |
| H'0_1004_CC00 | H'0_1004_CFFF | 1 Kbyte | IrDA (SCI) | |
| H'0_1004_C800 | H'0_1004_CBFF | 1 Kbyte | SCIF ch4 | |
| H'0_1004_C400 | H'0_1004_C7FF | 1 Kbyte | SCIF ch3 | |
| H'0_1004_C000 | H'0_1004_C3FF | 1 Kbyte | SCIF ch2 | |
| H'0_1004_BC00 | H'0_1004_BFFF | 1 Kbyte | SCIF ch1 | |
| H'0_1004_B800 | H'0_1004_BBFF | 1 Kbyte | SCIF ch0 | |
| H'0_1004_B400 | H'0_1004_B7FF | 1 Kbyte | Reserved | *1 |
| H'0_1004_B000 | H'0_1004_B3FF | 1 Kbyte | RSPI ch1 | |
| H'0_1004_AC00 | H'0_1004_AFFF | 1 Kbyte | RSPI ch0 | |
| H'0_1004_A800 | H'0_1004_ABFF | 1 Kbyte | Reserved | *1 |
| H'0_1004_A400 | H'0_1004_A7FF | 1 Kbyte | Reserved | *1 |
| H'0_1004_A000 | H'0_1004_A3FF | 1 Kbyte | Reserved | *1 |
| H'0_1004_9C00 | H'0_1004_9FFF | 1 Kbyte | SSIF ch0 | |
| H'0_1004_9800 | H'0_1004_9BFF | 1 Kbyte | POE3 | |
| H'0_1004_9400 | H'0_1004_97FF | 1 Kbyte | Reserved | *1 |
| H'0_1004_9000 | H'0_1004_93FF | 1 Kbyte | Reserved | *1 |
| H'0_1004_8C00 | H'0_1004_8FFF | 1 Kbyte | Reserved | *1 |
| H'0_1004_8800 | H'0_1004_8BFF | 1 Kbyte | Reserved | *1 |
| H'0_1004_8000 | H'0_1004_87FF | 2 Kbytes | Reserved | *1 |
| H'0_1004_7000 | H'0_1004_7FFF | 4 Kbytes | Reserved | *1 |
| H'0_1004_0000 | H'0_1004_6FFF | 28 Kbytes | Reserved | *1 |
| H'0_1000_0000 | H'0_1003_FFFF | 256 Kbytes | MTU3a | |
| H'0_0003_0000 | H'0_0FFF_FFFF | 261952 Kbytes | Reserved | *1 |
| H'0_0002_0000 | H'0_0002_FFFF | 64 Kbytes | SRAM ACPU (Memory) | |
| H'0_0001_0000 | H'0_0001_FFFF | 64 Kbytes | SRAM MCPU (Memory) | |
| H'0_0000_0000 | H'0_0000_FFFF | 64 Kbytes | Reserved | *1 |

Note 1. Access to the reserved areas is prohibited.
If access to the reserved area is attempted, incorrect operation may occur.

Note 2. Access to the addresses H'4000_0000 to H'4000_001F (for DDR3L) in the DDR area is prohibited because these are used for DDR training. For details, see **Section 9**, **DDR3L SDRAM Memory Controller (MEMC)**.

## 5.3 Accessible Areas

In the bus system of this LSI, each bus master unit can only access the areas that are used for register access or data transfer. **Table 5.2** shows the areas that can be accessed from each master.

Table 5.2 Accessible Areas (1/2)

| Slave Unit | Master Unit | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Cortex-A55 | DMAC_NS | CoreSight [AXI-AP] | CoreSight [ETR] | SDHI (ch0) | USB2.0 (ch0) | DRW | FCPVD | DSI (Controller) |
| DDR (Memory) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| SRAM ACPU (Memory) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| SRAM MCPU (Memory) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| SPI Multi I/O (Memory/Reg) | ✓ | ✓ | ✓ | × | × | × | × | × | × |
| I2C (ch0, ch1) | ✓ | ✓ | ✓ | × | × | × | × | × | × |
| TSU | ✓ | ✓ | ✓ | × | × | × | × | × | × |
| POE3 | ✓ | ✓ | ✓ | × | × | × | × | × | × |
| MTU3a | ✓ | ✓ | ✓ | × | × | × | × | × | × |
| SSIF (ch0) | ✓ | ✓ | ✓ | × | × | × | × | × | × |
| RSPI (ch0, ch1) | ✓ | ✓ | ✓ | × | × | × | × | × | × |
| SCIF (ch0 to ch4) | ✓ | ✓ | ✓ | × | × | × | × | × | × |
| IrDA (SCI) | ✓ | ✓ | ✓ | × | × | × | × | × | × |
| SCI (ch0, ch1) | ✓ | ✓ | ✓ | × | × | × | × | × | × |
| WDT (ch0) | ✓ | ✓ | ✓ | × | × | × | × | × | × |
| GTM (ch0 to ch2) | ✓ | ✓ | ✓ | × | × | × | × | × | × |
| DMAC_NS | ✓ | ✓ | ✓ | × | × | × | × | × | × |
| SDHI (ch0) | ✓ | ✓ | ✓ | × | × | × | × | × | × |
| USBPHY Control | ✓ | ✓ | ✓ | × | × | × | × | × | × |
| USB ch0 (OTG-Host) | ✓ | ✓ | ✓ | × | × | × | × | × | × |
| USB ch0 (OTG-Func) | ✓ | ✓ | ✓ | × | × | × | × | × | × |
| DRW | ✓ | ✓ | ✓ | × | × | × | × | × | × |
| FCPVD | ✓ | ✓ | ✓ | × | × | × | × | × | × |
| VSPD | ✓ | ✓ | ✓ | × | × | × | × | × | × |
| DU | ✓ | ✓ | ✓ | × | × | × | × | × | × |
| DSI (Controller) | ✓ | ✓ | ✓ | × | × | × | × | × | × |
| DSI (PHY) | ✓ | ✓ | ✓ | × | × | × | × | × | × |
| CoreSight | ✓ | ✓ | ✓ | × | × | × | × | × | × |
| SYSC | ✓ | ✓ | ✓ | × | × | × | × | × | × |
| SRAM ACPU (Reg) | ✓ | ✓ | ✓ | × | × | × | × | × | × |
| SRAM MCPU (Reg) | ✓ | ✓ | ✓ | × | × | × | × | × | × |
| SYC | ✓ | ✓ | ✓ | × | × | × | × | × | × |

Table 5.2 Accessible Areas (2/2)

| Slave Unit | Master Unit | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Cortex-A55 | DMAC_NS | CoreSight [AXI-AP] | CoreSight [ETR] | SDHI (ch0) | USB2.0 (ch0) | DRW | FCPVD | DSI (Controller) |
| GPIO | ✓ | ✓ | ✓ | × | × | × | × | × | × |
| CPG | ✓ | ✓ | ✓ | × | × | × | × | × | × |
| IA55 (Interrupt controller) | ✓ | ✓ | ✓ | × | × | × | × | × | × |
| GIC | ✓ | ✓ | ✓ | × | × | × | × | × | × |
| DDR (Controller) | ✓ | ✓ | ✓ | × | × | × | × | × | × |
| DDR (PHY) | ✓ | ✓ | ✓ | × | × | × | × | × | × |

**Remarks:** ✓: Accessible path, ×: Prohibited path

**Note:** Usage of paths marked with × in the table is prohibited.
If access to such a path is attempted, incorrect operation may occur.

## 5.4 Bus System Control

### 5.4.1 Bus Error Interrupt Generation

This LSI generates a bus error interrupt when any of the following conditions is met.

- A bus error is generated when access to a slave unit in the module stop state (MSTOP) is attempted.
- A bus error is generated by a slave unit.

This function is supported for all bus master units included in this LSI.

Bus error interrupts are controlled (retained, cleared, and gathered into a single signal) by the interrupt synchronization circuits (IA55). For details, including the types of the bus errors, see **Section 8, Interrupt Controller**.

# 6. System Controller (SYSC)

## 6.1 Overview

### 6.1.1 Features

SYSC is a unit that performs system control of this LSI and has the following functions.

- Product information, external terminal state capture function
  - Management of product information of this LSI, and external terminal status
- Low power consumption control (see **Section 42, Low Power Mode** for details)
  - Control the transition of this LSI to each mode of low power consumption
  - Setting cancellation conditions and confirming factors at the time of cancellation
- WDT stop control
  - Control WDT stop from CoreSight
  - Stop control of specified channel of WDT from CPU

### 6.1.2 Block Diagram of SYSC

The block diagram of SYSC is shown in **Figure 6.1**.

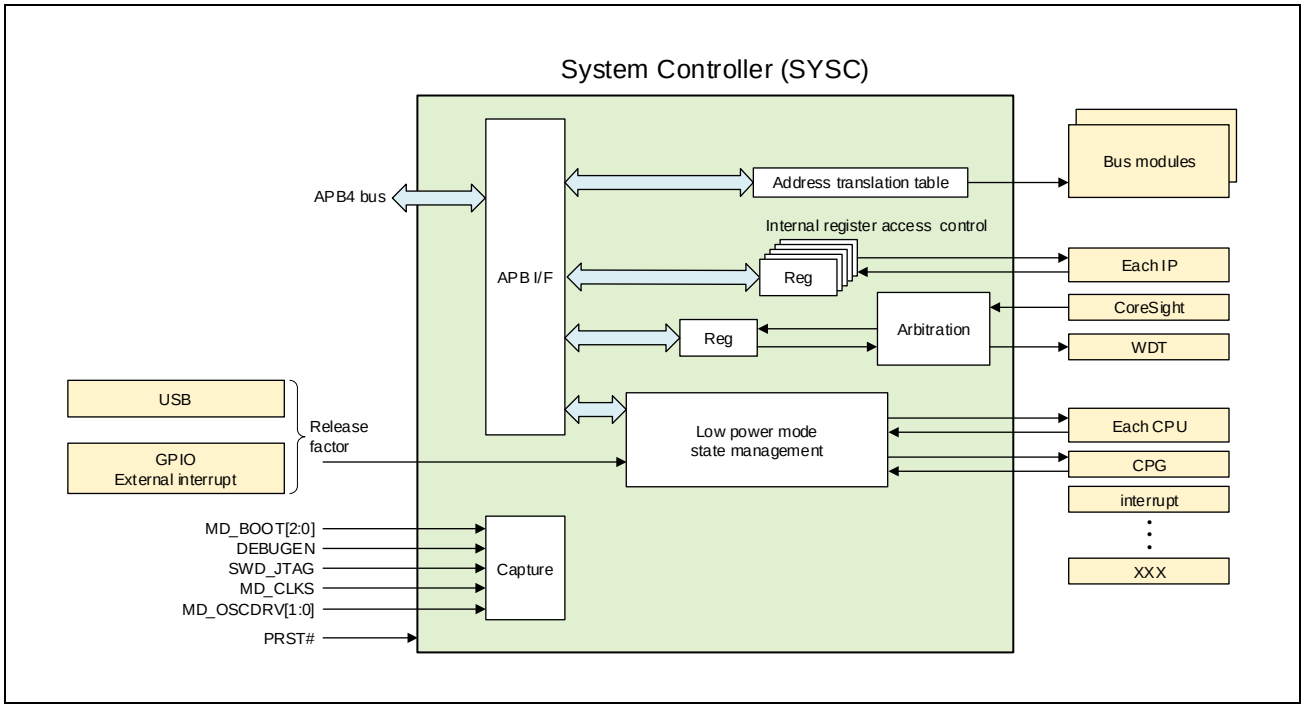



Figure 6.1 SYSC Block Diagram

## 6.2 Register Configuration

**Table 6.1** shows the register configuration of SYSC. The address of the SYSC register is represented by the offset address from the base address. SYSC base address is as follows:

SYSC base address: H'0_1102_0000 (Overall Address Space)

Table 6.1 SYSC Register Configuration

| Register Name | Abbreviation | R/W | Initial Value | Offset Address | Access Size |
|---|---|---|---|---|---|
| ECCRAM0 ECC Setting Register | SYS_RAM0_ECC | R/W | H'0000_0000 | H'0200 | 32 |
| ECCRAM0 Access Control Register | SYS_RAM0_EN | R/W | H'0000_0003 | H'0204 | 32 |
| ECCRAM1 ECC Setting Register | SYS_RAM1_ECC | R/W | H'0000_0000 | H'0210 | 32 |
| ECCRAM1 Access Control Register | SYS_RAM1_EN | R/W | H'0000_0003 | H'0214 | 32 |
| WDT0 Control Register | SYS_WDT0_CTRL | R/W | H'0000_0000 | H'0220 | 32 |
| I2C0 Config Register | SYS_I2C0_CFG | R/W | H'0000_0000 | H'0400 | 32 |
| I2C1 Config Register | SYS_I2C1_CFG | R/W | H'0000_0000 | H'0410 | 32 |
| CA55 Core0 Reset Vector Address Low Configuration Register | SYS_CA55_CFG_RVAL0 | R/W | H'0000_0000 | H'0858 | 32 |
| CA55 Core0 Reset Vector Address High Configuration Register | SYS_CA55_CFG_RVAH0 | R/W | H'0000_0000 | H'085C | 32 |
| LSI Mode Signal Register | SYS_LSI_MODE | R | H'000x_xx0x | H'0A00 | 32 |
| LSI Device ID Register | SYS_LSI_DEVID | R | H'x867_A447 | H'0A04 | 32 |
| LSI Product Register | SYS_LSI_PRR | R | H'00xx_xxxx | H'0A08 | 32 |
| Lowpower Sequence Control Register2 | SYS_LP_CTL2 | R/W | H'0000_0000 | H'0D08 | 32 |
| Lowpower Sequence Control Register5 | SYS_LP_CTL5 | R/W | H'0000_0000 | H'0D14 | 32 |
| Lowpower Sequence Control Register6 | SYS_LP_CTL6 | R/W | H'0000_0000 | H'0D18 | 32 |
| CA55 Clock Control Register1 | SYS_LP_CA55CK_CTL1 | R | H'0000_0x0x | H'0D38 | 32 |
| CA55 Clock Control Register2 | SYS_LP_CA55CK_CTL2 | R/W | H'0000_0F06 | H'0D3C | 32 |
| CA55 Clock Control Register3 | SYS_LP_CA55CK_CTL3 | R | H'0x0x_0x0x | H'0D40 | 32 |
| General Register0 | SYS_GPREG_0 | R/W | H'0000_0000 | H'0E00 | 32 |
| General Register1 | SYS_GPREG_1 | R/W | H'0000_0000 | H'0E04 | 32 |
| General Register2 | SYS_GPREG_2 | R/W | H'0000_0000 | H'0E08 | 32 |
| General Register3 | SYS_GPREG_3 | R/W | H'0000_0000 | H'0E0C | 32 |

# 6.3 Register Descriptions

## 6.3.1 ECCRAM0 ECC Setting Register (SYS_RAM0_ECC)

This register is a register that enables and disables the ECC function of ECCRAM0 (SRAM ACPU).

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | VECCEN |
| 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<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 0 | VECCEN | 0 | R/W | Set enable or disable error correction for ECCRAM0 (SRAM ACPU)<br>0: ECC function disabled<br>1: ECC function enabled |

## 6.3.2 ECCRAM0 Access Control Register (SYS_RAM0_EN)

This register is a register that controls access to ECCRAM0 (SRAM ACPU).

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | VLWEN | VCEN |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 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 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 1 | VLWEN | 1 | R/W | Set enable or disable writes to ECRAM0 (SRAM ACPU)<br>0: Write disabled<br>1: Write enabled |
| 0 | VCEN | 1 | R/W | Set enable or disable access to ECRAM0 (SRAM ACPU)<br>0: Access disabled<br>1: Access enabled |

### 6.3.3 ECCRAM1 ECC Setting Register (SYS_RAM1_ECC)

This register is a register that enables and disables the ECC function of ECCRAM1 (SRAM MCPU).

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | VECCEN |
| 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<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 0 | VECCEN | 0 | R/W | Set enable or disable error correction for ECCRAM1 (SRAM MCPU)<br>0: ECC function disabled<br>1: ECC function enabled |

### 6.3.4 ECCRAM1 Access Control Register (SYS_RAM1_EN)

This register is the register that controls access to ECCRAM1 (SRAM M 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | VLWEN | VCEN |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 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 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 1 | VLWEN | 1 | R/W | Set enable or disable writes to ECCRAM1 (SRAM MCPU)<br>0: Write disabled<br>1: Write enabled |
| 0 | VCEN | 1 | R/W | Set enable or disable access to ECRAM1 (SRAM MCPU)<br>0: Access disabled<br>1: Access enabled |

### 6.3.5 WDT0 Control Register (SYS_WDT0_CTRL)

This register is the register that set the count control of WDT0.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | WDTSTOPMASK |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | WDTSTOP |
| 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<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 16 | WDTSTOPMASK | 0 | R/W | Controls the stop function of WDT0 (for Cortex-A55 Core0) from CoreSight during debugging.<br>0: Stops WDT0 counting during debugging.<br>1: Not stop WDT0 counting during debugging. |
| 15 to 1 | — | All 0 | R | Reserved<br>When read, the initial value is read. The written value will be ignored. |
| 0 | WDTSTOP | 0 | R/W | Set whether the WDT0 (for Cortex-A55 Core0) counter is forced to stop<br>0: Continue counting<br>1: Stop counting and hold the counter value |

### 6.3.6 I2C0 Config Register (SYS_I2C0_CFG)

This register is a register that set the bypass mode of the analog filter of I2C ch0.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | af_bypass |
| 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<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 0 | af_bypass | 0 | R/W | Set whether to bypass the analog filter for noise removal for the input terminals (RIIC0_SDA, RIIC0_SCL) of I2C ch0.<br>0: Noise reduction is performed using an analog filter for the input signal.<br>1: The input signal is bypassed an analog filter and noise reduction is not performed. |

### 6.3.7 I2C1 Config Register (SYS_I2C1_CFG)

This register is a register that set the bypass mode of the analog filter of I2C ch1.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | af_bypass |
| 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<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 0 | af_bypass | 0 | R/W | Set whether to bypass the analog filter for noise removal for the input terminals (RIIC1_SDA, RIIC1_SCL) of I2C ch1.<br>0: Noise reduction is performed using an analog filter for the input signal.<br>1: The input signal is bypassed an analog filter and noise reduction is not performed. |

### 6.3.8 CA55 Core0 Reset Vector Address Low Configuration Register (SYS_CA55_CFG_RVAL0)

This register is a register that set the reset vector address of Cortex-A55 core0.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RVBARADDRL0[31:2] | | | | | | | | | | | | | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RVBARADDRL0[31:2] | | | | | | | | | | | | | | — | — |
| 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 | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 2 | RVBARADDRL0[31: 2] | H'0000_0000 | R/W | Represent bottom bit 31 to bit 2 of the reset vector address of Cortex-A55 Core0.<br>When you change a vector address, you must set the address value and then release the reset (Cold reset/Cluster Warm reset/Core Warm reset). See **Section 2, System CPU Cortex-A55** for the sequence of each reset. |
| 1, 0 | — | 00b | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |

### 6.3.9 CA55 Core0 Reset Vector Address High Configuration Register (SYS_CA55_CFG_RVAH0)

This register is a register that set the reset vector address of Cortex-A55 core0.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | RVBARADDRH0[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<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 7 to 0 | RVBARADDRH0[7:0] | H'00 | R/W | Represent upper bit 39 to bit 32 of the reset vector address of Cortex-A55 Core0.<br>When you change a vector address, you must set the address value and then release the reset (Cold reset/Cluster Warm reset/Core Warm reset). See **Section 2, System CPU Cortex-A55** for the sequence of each reset. |

### 6.3.10 LSI Mode Signal Register (SYS_LSI_MODE)

This register indicates the state of the MD_BOOT terminal, the DEBUGEN terminal, the MD_CLKS terminal, and the MD_OSCDRV terminal.

| 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 | — |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | STAT_MD_OSCDRV[1:0] | | — | STAT_MD_CLKS | — | STAT_SWD_JTAG | STAT_DEBUGEN | — | — | — | — | — | — | STAT_MD_BOOT[2:0] | | |
| 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 17 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 16 | — | — | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 15, 14 | STAT_MD_OSCDRV[1:0] | — | R | Indicates the terminal state of the external terminal MD_OSCDRV[1:0].<br>The value is updated at the rising edge of PRST#.<br>If a reset occurs inside the LSI, this register is not updated. |
| 13 | — | 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 12 | STAT_MD_CLKS | — | R | Indicates the terminal state of the external terminal MD_CLKS.<br>The value is updated at the rising edge of PRST#.<br>If a reset occurs inside the LSI, this register is not updated. |
| 11 | — | 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 10 | STAT_SWD_JTAG | — | R | Indicates the terminal state of the external terminal SWD_JTAG.<br>The value is updated at the rising edge of PRST#.<br>If a reset occurs inside the LSI, this register is not updated. |
| 9 | STAT_DEBUGEN | — | R | Indicates the terminal state of the external terminal DEBUGEN.<br>The value is updated at the rising edge of PRST#.<br>If a reset occurs inside the LSI, this register is not updated. |
| 8 | — | 0 | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 7 to 3 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 2 to 0 | STAT_MD_BOOT[2:0] | — | R | Indicates the terminal state of the external terminal MD_BOOT[2:0].<br>The value is updated at the rising edge of PRST#.<br>If a reset occurs inside the LSI, this register is not updated. |

### 6.3.11 LSI Device ID Register (SYS_LSI_DEVID)

This register indicates the product specific device ID.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | DEV_ID[31:28] | | | | DEV_ID[27:0] | | | | | | | | | | | |
| Initial Value | — | — | — | — | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | DEV_ID[27:0] | | | | | | | | | | | | | | | |
| Initial Value | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 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 28 | DEV_ID [31:28] | — | R | Indicates the product revision. |
| 27 to 0 | DEV_ID[27:0] | H’867_A447 | R | Indicates the product specific fixed value. |

### 6.3.12 LSI Product Register (SYS_LSI_PRR)

This register indicates information about product options.

| 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 | — |
| 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 | — |
| 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 21 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 20 | — | — | R | Reserved<br>When read, the initial value is read. The written value will be ignored. |
| 19 to 17 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 16 | — | — | R | Reserved<br>When read, the initial value is read. The written value will be ignored. |
| 15 to 13 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 12 | — | — | R | Reserved<br>When read, the initial value is read. The written value will be ignored. |
| 11 to 9 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 8 | — | — | R | Reserved<br>When read, the initial value is read. The written value will be ignored. |
| 7 to 5 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 4 | — | — | R | Reserved<br>When read, the initial value is read. The written value will be ignored. |
| 3 to 1 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 0 | — | — | R | Reserved<br>When read, the initial value is read. The written value will be ignored. |

### 6.3.13 SYS_LP_CTL2

This register is set before and after Cortex-A55 Sleep 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CA55_STBYCTL |
| 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<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 0 | CA55_STBY CTL | 0b | R/W | Registers to be set before and after cortex-A55 Sleep Mode.<br>0: Cortex-A55 Sleep Mode operation complete.<br>When using in other modes, be sure to 0.<br>1: Cortex-A55 Sleep Mode start |

### 6.3.14 SYS_LP_CTL5

This register controls the low power 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | AMCLKQDENY_F | ASCLKQDENY_F | — |
| 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 |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 21 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 20 | — | 0b | R | Reserved<br>When read, the initial value is read. The written value will be ignored. |
| 19 to 17 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 16 | — | 0b | R | Reserved<br>When read, the initial value is read. The written value will be ignored. |
| 15 to 11 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 10 to 8 | — | 000b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value.<br>Operation is not guaranteed if a value other than the initial value is written. |
| 7 to 5 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 4 | — | 0b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value.<br>Operation is not guaranteed if a value other than the initial value is written. |
| 3 | — | 0b | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 2 | AMCLKQDENY_F | 0b | R/W | Cortex-A55 ACE Asynchronous Bridge Master Interface QDENY Factor AMCLKQDENY Flag<br>0: No Deny response (clear with 0 write)<br>1: Deny response (SYS_CA55_DENY interrupt occurs) |
| 1 | ASCLKQDENY_F | 0b | R/W | Cortex-A55 ACE Asynchronous Bridge Slave Interface QDENY Factor ASCLKQDENY Flag<br>0: No Deny response (clear with 0 write)<br>1: Deny response (SYS_CA55_DENY interrupt occurs) |
| 0 | — | 0b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value.<br>Operation is not guaranteed if a value other than the initial value is written. |

### 6.3.15 SYS_LP_CTL6

This register controls the low power 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | AMCLKQDENY_E | ASCLKQDENY_E | — |
| 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 |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 11 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 10 to 8 | — | 000b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 7 | — | 0b | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 6 | — | 0b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 5 | — | 0b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 4 | — | 0b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 3 | — | 0b | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 2 | AMCLKQDENY_E | 0b | R/W | Cortex-A55 ACE Asynchronous Bridge Master Interface QDENY Configuration Factor Mask<br>0: Masks the configuration factors AMCLKQDENY_F for cortex-A55 ACE asynchronous bridge master interface QDENY<br>1: Do not mask the configuration factor AMCLKQDENY_F for Cortex-A55 ACE Asynchronous Bridge Master Interface QDENY |
| 1 | ASCLKQDENY_E | 0b | R/W | Cortex-A55 ACE Asynchronous Bridge Slave Interface QDENY Configuration Factor Mask<br>0: Masks the configuration factors ASCLKQDENY_F for cortex-A55 ACE asynchronous bridge slave interface QDENY<br>1: Do not mask the configuration factor ASCLKQDENY_F for Cortex-A55 ACE Asynchronous bridge slave Interface QDENY |
| 0 | — | 0b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |

### 6.3.16 SYS_LP_CA55CK_CTL1

This register controls the low power mode of Cortex-A55.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | PDBGCLKQACTIVE | GICCLKQACTIVE | ATCLKQACTIVE | PCLKQACTIVE | — | — | — | — | — | AMCLKQACTIVE | ASCLKQACTIVE | — |
| Initial Value | 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 12 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 11 | PDBGCLKQACTIVE | — | R | Represents the status of a QACTIVE signal from the PDBGCLK domain of Cortex-A55 cluster.*1 |
| 10 | GICCLKQACTIVE | — | R | Represents the status of a QACTIVE signal from the GICCLK domain of Cortex-A55 cluster.*1 |
| 9 | ATCLKQACTIVE | — | R | Represents the status of a QACTIVE signal from the ATCLK domain of Cortex-A55 cluster.*1 |
| 8 | PCLKQACTIVE | — | R | Represents the status of a QACTIVE signal from the PCLK domain of Cortex-A55 cluster.*1 |
| 7 to 3 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 2 | AMCLKQACTIVE | — | R | Represents the state of the QACTIVE signal from cortex-A55 ACE asynchronous bridge Master Interface.*1 |
| 1 | ASCLKQACTIVE | — | R | Represents the state of the QACTIVE signal from the Cortex-A55 ACE asynchronous bridge Slave Interface.*1 |
| 0 | — | — | R | Reserved<br>When read, the initial value is read. The written value will be ignored. |

Note 1. The state of this register is used to determine whether subsequent Q-channel controls will start clock stop operations. For more information about QACTIVE signal specifications, see AMBA® Low Power Interface ARM® Q-Channel and P-Channel Interfaces.

### 6.3.17 SYS_LP_CA55CK_CTL2

This register controls the low power mode of Cortex-A55.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | PDBGCLKQREQn | GICCLKQREQn | ATCLKQREQn | PCLKQREQn | — | — | — | — | — | AMCLKQREQn | ASCLKQREQn | — |
| Initial Value | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 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 |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 12 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 11 | PDBGCLKQREQn | 1b | R/W | Set the level of the QREQn signal to the PDBGCLK domain of the Cortex-A55 cluster. When this register is read, the value set at the time of the write is read.<br>This register represents the state of the Q-channel in combination with PDBGCLKQACCEPTn and PDBGCLKQDENY.<br>For more information on these combinations, see AMBA® Low Power Interface Arm® Q-Channel and P-Channel Interfaces. |
| 10 | GICCLKQREQn | 1b | R/W | Set the level of the QREQn signal to the GICCLK domain of the Cortex-A55 cluster. When this register is read, the value set at the time of the write is read.<br>This register represents the state of the Q-channel in combination with GICCLKQACCEPTn and GICCLKQDENY.<br>For more information on these combinations, see AMBA® Low Power Interface Arm® Q-Channel and P-Channel Interfaces. |
| 9 | ATCLKQREQn | 1b | R/W | Set the level of the QREQn signal to the ATCLK domain of the Cortex-A55 cluster. When this register is read, the value set at the time of the write is read.<br>This register represents the state of the Q-channel in combination with ATCLKQACCEPTn and ATCLKQDENY.<br>For more information on these combinations, see AMBA® Low Power Interface Arm® Q-Channel and P-Channel Interfaces. |
| 8 | PCLKQREQn | 1b | R/W | Set the level of the QREQn signal to the PCLK domain of the Cortex-A55 cluster. When this register is read, the value set at the time of the write is read.<br>This register represents the state of the Q-channel in combination with PCLKQACCEPTn and PCLKQDENY.<br>For more information on these combinations, see AMBA® Low Power Interface Arm® Q-Channel and P-Channel Interfaces. |
| 7 to 3 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 2 | AMCLKQREQn | 1b | R/W | Set the level of the QREQn signal to the ACE asynchronous bridge Master Interface on cortex-A55. When this register is read, the value set at the time of the write is read.<br>This register represents the state of the Q-channel in combination with AMCLKQACCEPTn and AMCLKQDENY.<br>For more information on these combinations, see AMBA® Low Power Interface Arm® Q-Channel and P-Channel Interfaces. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 1 | ASCLKQREQn | 1b | R/W | Set the level of the QREQn signal to the ACE asynchronous bridge Slave Interface on cortex-A55. When this register is read, the value set at the time of the write is read.<br>This register represents the state of the Q-channel in combination with ASCLKQACCEPTn and ASCLKQDENY.<br>For more information on these combinations, see AMBA® Low Power Interface Arm® Q-Channel and P-Channel Interfaces. |
| 0 | — | 0b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |

## 6.3.18 SYS_LP_CA55CK_CTL3

This register controls the low power mode of Cortex-A55.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | PDBGCLKQDENY | GICCLKQDENY | ATCLKQDENY | PCLKQDENY | — | — | — | — | — | AMCLKQDENY | ASCLKQDENY | — |
| Initial Value | 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 |

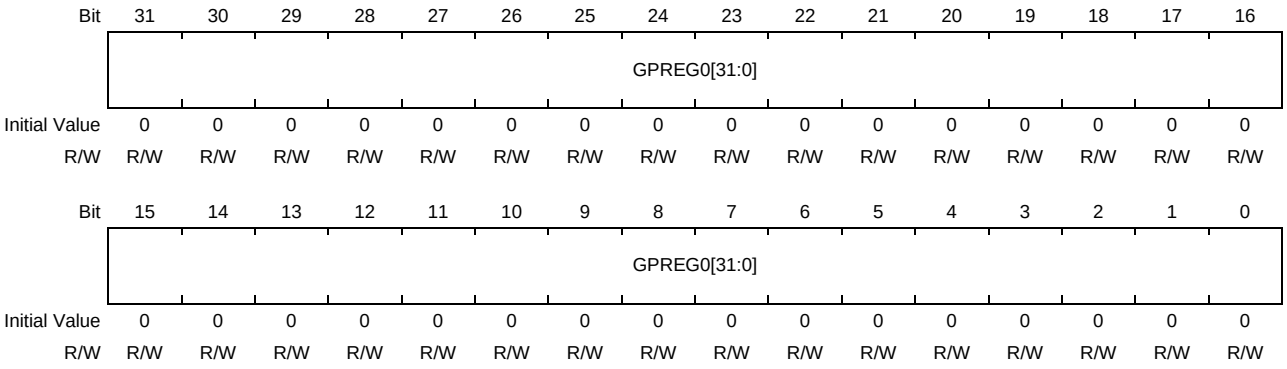
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | PDBGCLKQACCEPTn | GICCLKQACCEPTn | ATCLKQACCEPTn | PCLKQACCEPTn | — | — | — | — | — | AMCLKQACCEPTn | ASCLKQACCEPTn | CA55_CORERINSTRRUN[0] |
| Initial Value | 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 | — | 0000b | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 27 | PDBGCLKQDENY | — | R | Represents the status of the QDENY signal from the PDBGCLK domain of a Cortex-A55 cluster.<br>This register represents the state of the Q-channel in combination with PDBGCLKQREQn and PDBGCLKQACCEPTn.<br>For more information on these combinations, see AMBA® Low Power Interface Arm® Q-Channel and P-Channel Interfaces. |
| 26 | GICCLKQDENY | — | R | Represents the status of the QDENY signal from the GICCLK domain of a Cortex-A55 cluster.<br>This register represents the state of the Q-channel in combination with GICCLKQREQn and GICCLKQACCEPTn.<br>For more information on these combinations, see AMBA® Low Power Interface Arm® Q-Channel and P-Channel Interfaces. |
| 25 | ATCLKQDENY | — | R | Represents the status of the QDENY signal from the ATCLK domain of a Cortex-A55 cluster.<br>This register represents the state of the Q-channel in combination with ATCLKQREQn and ATCLKQACCEPTn.<br>For more information on these combinations, see AMBA® Low Power Interface Arm® Q-Channel and P-Channel Interfaces. |
| 24 | PCLKQDENY | — | R | Represents the status of the QDENY signal from the PCLK domain of a Cortex-A55 cluster.<br>This register represents the state of the Q-channel in combination with PCLKQREQn and PCLKQACCEPTn.<br>For more information on these combinations, see AMBA® Low Power Interface Arm® Q-Channel and P-Channel Interfaces. |
| 23 to 19 | — | 00000b | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 18 | AMCLKQDENY | — | R | Represents the status of the QDENY signal from the ACE asynchronous bridge Master Interface on cortex-A55.<br>This register represents the state of the Q-channel in combination with AMCLKQREQn and AMCLKQACCEPTn.<br>For more information on these combinations, see AMBA® Low Power Interface Arm® Q-Channel and P-Channel Interfaces. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 17 | ASCLKQDENY | — | R | Represents the status of the QDENY signal from the ACE asynchronous bridge Slave Interface on cortex-A55.<br>This register represents the state of the Q-channel in combination with ASCLKQREQn and ASCLKQACCEPTn.<br>For more information on these combinations, see AMBA® Low Power Interface Arm® Q-Channel and P-Channel Interfaces. |
| 16 | — | — | R | Reserved<br>When read, the initial value is read. The written value will be ignored. |
| 15 to 12 | — | 0000b | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 11 | PDBGCLKQACCEPTn | — | R | Represents the state of the QACCEPTn signal from the PDBGCLK domain of a Cortex-A55 cluster.<br>This register represents the state of the Q-channel in combination with PDBGCLKQREQn and PDBGCLKQDENY.<br>For more information on these combinations, see AMBA® Low Power Interface Arm® Q-Channel and P-Channel Interfaces. |
| 10 | GICCLKQACCEPTn | — | R | Represents the state of the QACCEPTn signal from the GICCLK domain of a Cortex-A55 cluster.<br>This register represents the state of the Q-channel in combination with GICCLKQREQn and GICCLKQDENY.<br>For more information on these combinations, see AMBA® Low Power Interface Arm® Q-Channel and P-Channel Interfaces. |
| 9 | ATCLKQACCEPTn | — | R | Represents the state of the QACCEPTn signal from the ATCLK domain of a Cortex-A55 cluster.<br>This register represents the state of the Q-channel in combination with ATCLKQREQn and ATCLKQDENY.<br>For more information on these combinations, see AMBA® Low Power Interface Arm® Q-Channel and P-Channel Interfaces. |
| 8 | PCLKQACCEPTn | — | R | Represents the state of the QACCEPTn signal from the PCLK domain of a Cortex-A55 cluster.<br>This register represents the state of the Q-channel in combination with PCLKQREQn and PCLKQDENY.<br>For more information on these combinations, see AMBA® Low Power Interface Arm® Q-Channel and P-Channel Interfaces. |
| 7 to 3 | — | 00000b | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 2 | AMCLKQACCEPTn | — | R | Represents the state of the QACCEPTn signal from the ACE asynchronous bridge Master Interface on cortex-A55.<br>This register represents the state of the Q-channel in combination with AMCLKQREQn and AMCLKQDENY.<br>For more information on these combinations, see AMBA® Low Power Interface Arm® Q-Channel and P-Channel Interfaces. |
| 1 | ASCLKQACCEPTn | — | R | Represents the state of the QACCEPTn signal from the ACE asynchronous bridge Slave Interface on cortex-A55.<br>This register represents the state of the Q-channel in combination with ASCLKQREQn and ASCLKQDENY.<br>For more information on these combinations, see AMBA® Low Power Interface Arm® Q-Channel and P-Channel Interfaces. |
| 0 | CA55_COREINSTRRUN[0] | — | R | Indicates that Cortex-A55 Core0 has transitioned to the WFI/WFE state.<br>0: Transitioned or OFF state<br>1: Normal operation |

## 6.3.19 General Register0 (SYS_GPREG_0)

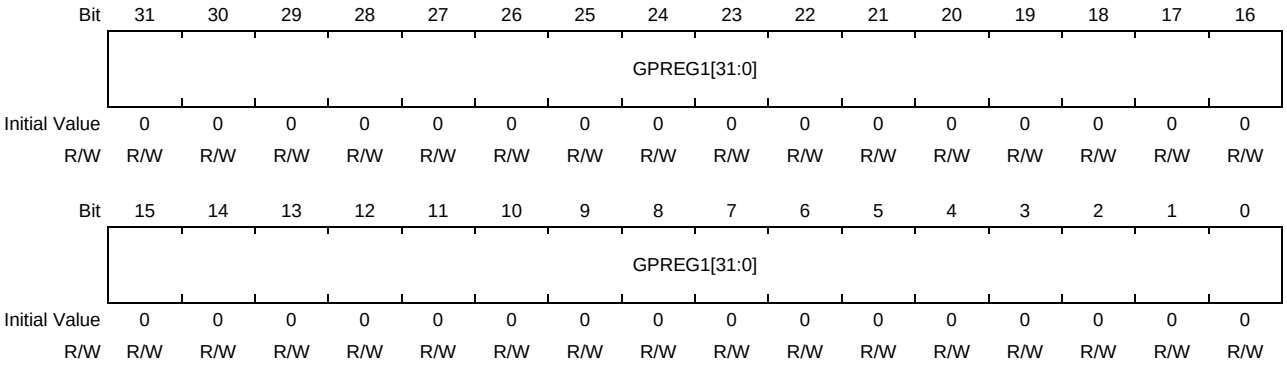
This register is a general-purpose register.




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 0 | GPREG0 [31:0] | H'0000_0000 | R/W | General-purpose register 0<br>It is a 32-bit register.<br>It is possible to write any value, read the written value. |

## 6.3.20 General Register1 (SYS_GPREG_1)

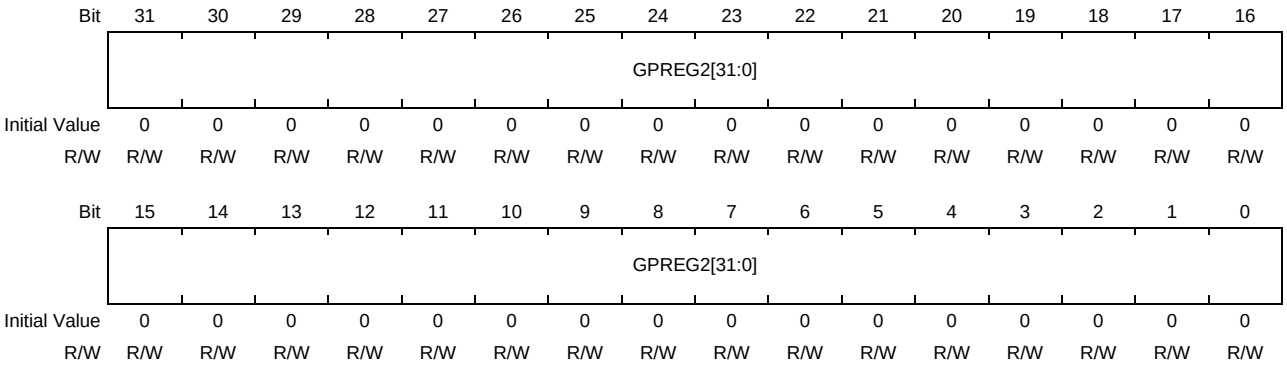
This register is a general-purpose register.




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 0 | GPREG1 [31:0] | H'0000_0000 | R/W | General-purpose register 1<br>It is a 32-bit register.<br>It is possible to write any value, read the written value. |

## 6.3.21 General Register2 (SYS_GPREG_2)

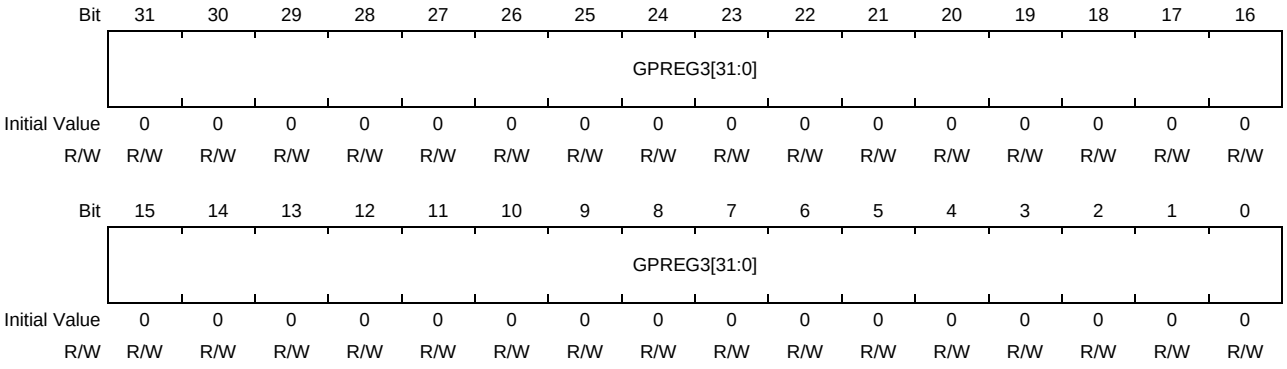
This register is a general-purpose register.




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 0 | GPREG2 [31:0] | H'0000_0000 | R/W | General-purpose register 2<br>It is a 32-bit register.<br>It is possible to write any value, read the written value. |

## 6.3.22 General Register3 (SYS_GPREG_3)

This register is a general-purpose register.




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 0 | GPREG3 [31:0] | H'0000_0000 | R/W | General-purpose register 3<br>It is a 32-bit register.<br>It is possible to write any value, read the written value. |

# 6.4 Operation

## 6.4.1 External Terminal State Capture Function

Captures the state of the following five external terminals at the rising edge of PRST# terminal and holds them into the SYS_LSI_MODE register. The value is not updated by a reset that occurs inside the LSI.

- MD_BOOT[2:0]
- DEBUGEN
- SWD_JTAG
- MD_CLKS
- MD_OSCDRV[1:0]

## 6.4.2 WDT Stop Control Function

By writing 1 to the WDTSTOP bit of the SYS_WDTn_CTRL register (n: 0), the following WDT corresponding to n is forcibly stopped and the count value is retained.

Also, by writing 1 to the WDTSTOPMASK bit of the SYS_WDTn_CTRL register (n: 0), the following WDT count corresponding to n can be continued even during debugging. If 0 is written, the WDT count will stop during debugging.

| n | Stop WDT |
|---|---|
| 0 | WDT0 (for Cortex-A55 core0) |

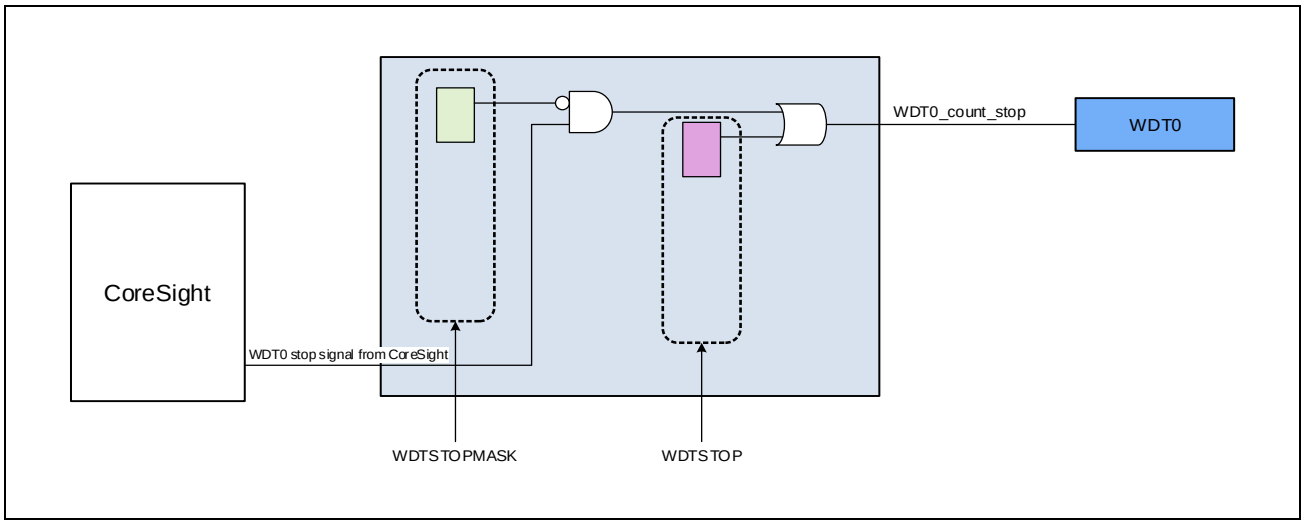



Figure 6.2 Overview of WDT Stop Function

# 7. Clock Pulse Generator (CPG)

This chapter describes the functions of the clock pulse generator module (CPG).

## 7.1 Features

### 7.1.1 Overview

**Table 7.1** provides a list of CPG features.

Table 7.1 List of functions

| Features | Specification |
|---|---|
| PLL control | ● SSCG control, PLL multiplication factor setting, and ON/OFF control are performed by register setting.<br>● This LSI is equipped with 5 PLLs. The roles of each are as follows:<br>– PLL1 (SSCG-PLL: Default SSCG ON): Cortex-A55 only<br>– PLL2 (SSCG-PLL: Default SSCG OFF): For SYSTEM-BUS & Module (no SSCG)<br>– PLL3 (SSCG-PLL: Default SSCG ON): For SYSTEM-BUS & Module (SSCG Yes)<br>– PLL4 (SSCG-PLL: Default SSCG ON): DDR-SDRAM Only<br>– PLL5 (SSCG-PLL: Default SSCG ON): Video-Out (DSI/LCDC)<br>● Initial value of SSCG for each PLL (PLL3 is fixed):<br>– Fmod: 31.25kHz<br>– Spread: Down spread<br>– Modulation Depth: 2.25% (PLL1 and PLL4)<br>2.20% (PLL3 and PLL5) |
| Clock generation and control | The clock to be supplied to each module is generated from the external clock input (EXCLK), crystal oscillator input / output (XIN / XOUT), or PLL output clock.<br>● Selection of the peri-ratio by register setting<br>● Clock path selection by clock selector by register setting<br>● Clock supply ON/OFF control is performed by register setting. (Module Standby Mode control) |
| Reset generation and control | A reset is generated from the reset factor in (). The types of reset are as follows.<br>● System reset (external pin)<br>● Software reset (system reboot, reset control of each module)<br>● Watchdog reset (WDT module)<br>● Cortex-A55 Warm reset control (software control)<br>● ON/OFF control for each unit clock (software control) |
| Monitor | ● PLL clock monitor<br>– The function to monitor the operation of the output clocks from PLL1 to PLL5<br>● Clock monitor for each module<br>– Function to monitor that the clock monitor is operating<br>● Reset monitor<br>– Function to monitor the reset status of each module |

### 7.1.2 CPG configuration

**Figure 7.1** shows the CPG configuration.

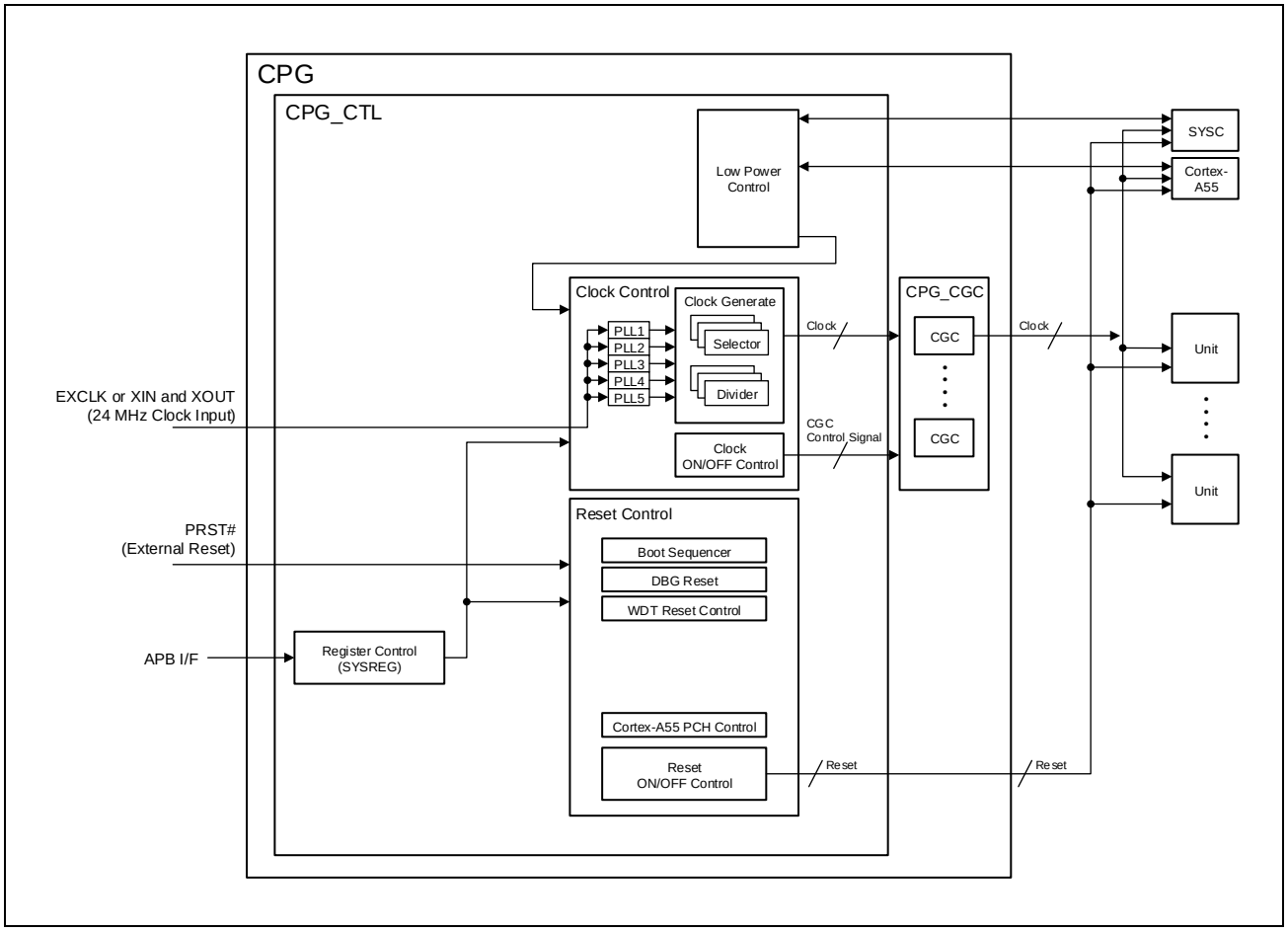



Figure 7.1 CPG Configuration

### 7.1.3 Pin description

The table below shows a list of CPG pins.

Table 7.2 Pin List

| Name | Pin Name | IO | Functions |
|---|---|---|---|
| Mode control pin | MD_BOOT0 | I | Set the operating mode.<br>Do not change during PRST# pin asserts and after negating until the operating mode is confirmed. |
| | MD_BOOT1 | I | |
| | MD_BOOT2 | I | |
| | MD_CLKS | I | Set ON/OFF of SSCG circuit operation of PLL3.<br>Do not change during PRST# terminal asserts and after negating until the operating mode is confirmed. |
| External clock Input pin*2 | EXCLK | I | This is the pin to input the external clock. |
| Crystal oscillator input/output pin*1 | XIN*3 | I | This is the pin to connect the crystal oscillator. |
| | XOUT*3 | O | |
| Power on reset input pin | PRST# | I | When this pin becomes low level, it will be in the power on reset state. |
| OSC drive capacity setting pin 0 | MD_OSCDRV0 | I | This is a pin for adjusting the drive capacity of the OSC buffer. |
| OSC drive capacity setting pin 1 | MD_OSCDRV1 | I | This is a pin for adjusting the drive capacity of the OSC buffer. |

Note 1. When using a crystal oscillator, fix the pins of MD_OSCDRV0 and MD_OSCDRV1 to Low Level. Also, refer to the C/R constraints and implementation precautions described in the PCB design checklist.

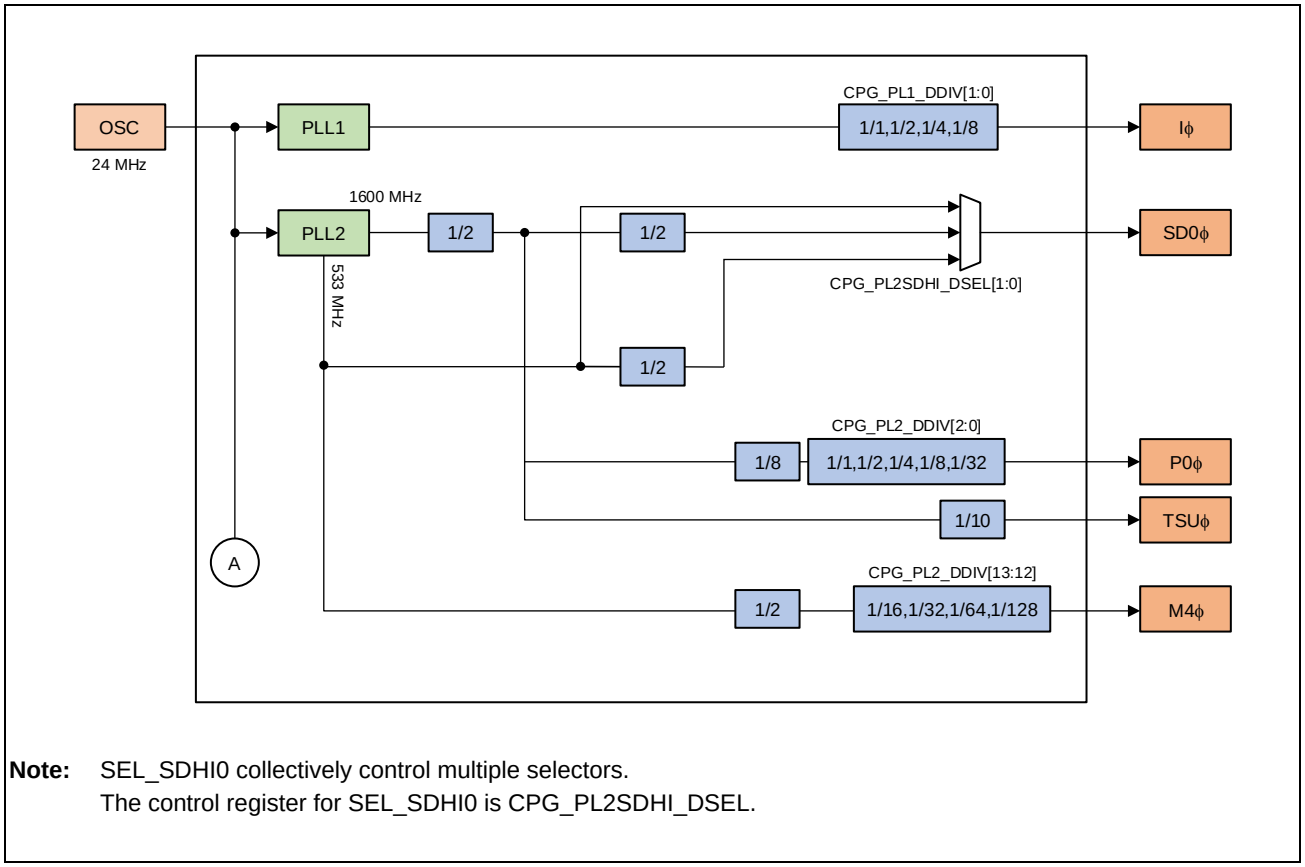
Note 2. When using crystal oscillation, fix the EXCLK pin externally to Low Level.

Note 3. When using the clock from the outside without using the crystal oscillator, set the XIN pin to the external low level fixing process and the XOUT pin to Open.

## 7.1.4 Clock

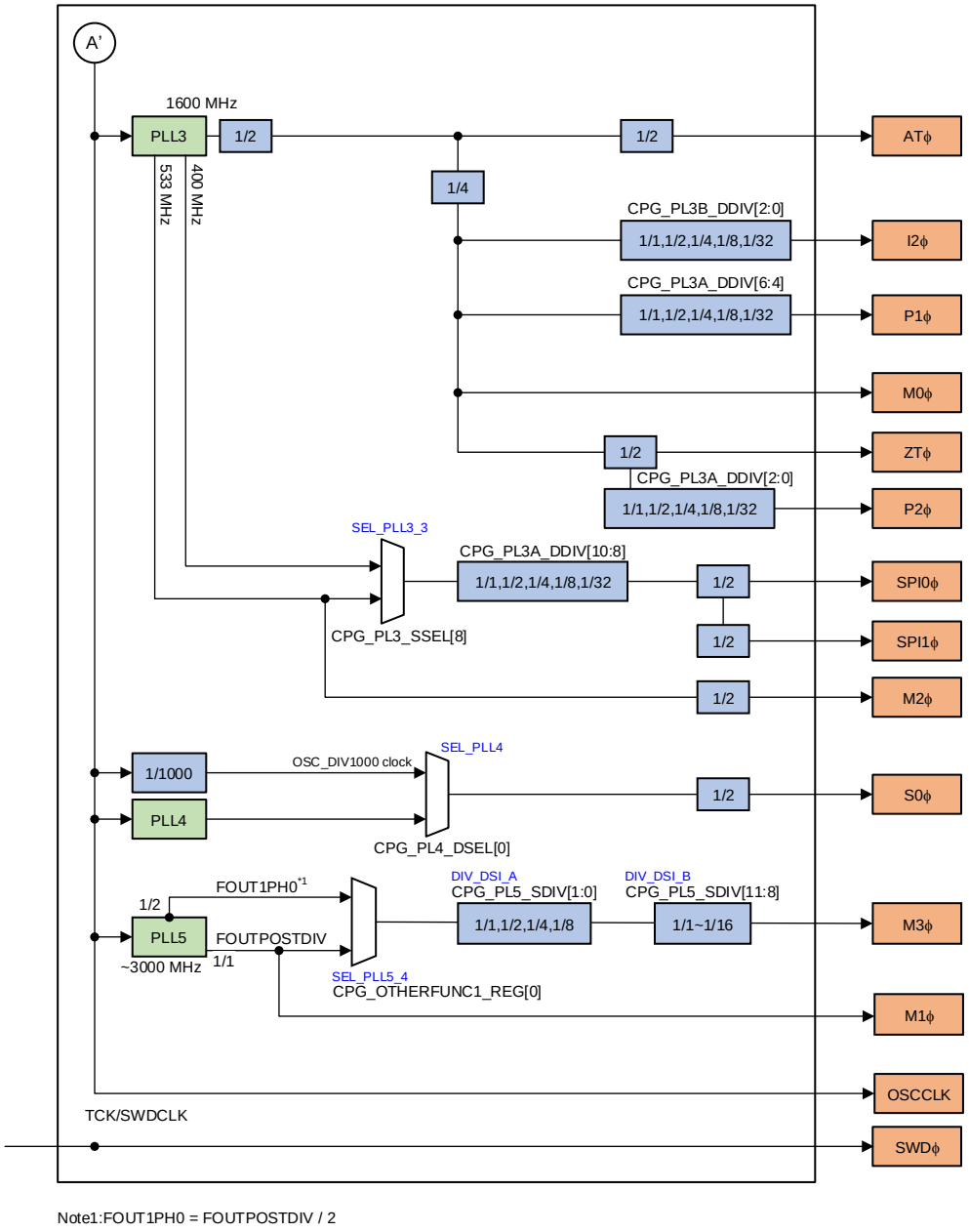
### 7.1.4.1 Clock System Diagram

**Figure 7.2** and **Figure 7.3** show the clock system diagram.




**Note:** SEL_SDHI0 collectively control multiple selectors.
The control register for SEL_SDHI0 is CPG_PL2SDHI_DSEL.

Figure 7.2 Clock System Diagram (1)




**Note:** The control registers of each selector are as follows.

- SEL_PLL3_3: CPG_PL3_SSEL
- SEL_PLL4: CPG_PL4_DSEL
- SEL_PLL5_4: CPG_OTHERFUNC1_REG

Figure 7.3 Clock System Diagram (2)

#### 7.1.4.2 Clock List

Please refer to another file for the clock list.

(Please double-click the icon on the right side) 

## 7.2 Register description

### 7.2.1 Register attributes

The following table shows the register attributes described in Register Details.

Table 7.3 Register attributes

| Register attributes | Description |
|---|---|
| RW | A register that can be read and written. |
| RW0 | A register that can read and 0b write. 1b Write is invalid. |
| RW1 | A register that can read and 1b write. 0b Write is invalid. |
| R0W | It is a register that can be written with 0b Read at all times. |
| R1W | It is a register that can be written with 1b Read at all times. |
| R | A register that can read. Write is invalid. |
| R0W0 | It is a register that can always read 0b and write 0b. 1b Write is invalid. |
| R0W1 | It is a register that can always read 0b and write 1b. 0b Write is invalid. |
| R1W0 | It is a register that can always read 1b and write 0b. 1b Write is invalid. |
| R1W1 | It is a register that can always read 1b and write 1b. 0b Write is invalid. |
| RCW0 | A register that can be cleared by Read and 0b Write is possible. 1b Write is invalid. |
| RCW1 | A register that can be cleared by Read and 1b Write is possible. 0b Write is invalid. |

**Note:** This LSI does not guarantee the access result to the Reserved bit area and the undefined bit area.
At the same time, the Reserved bit area and undefined area may have values other than the initial value “0”. We do not guarantee the result of changing these values.

### 7.2.2 Register classification

The table below shows the general classification of the register space in this module.

The address of the CPG register is represented by an offset address from the CPG-based address.

The CPG-based address is H’0_1101_0000.

Table 7.4 Register Address space

| Classification | Offset Address |
|---|---|
| SSCG PLL Control Registers (PLL1, 4) | H’000 - H’0FF |
| Fast PLL Control Registers (PLL2, 3, 5) | H’100 - H’17F |
| Reserved | H’180 - H’1FF |
| Frequency Dynamic Change Control Registers | H’200 - H’3FF |
| Frequency Static Change Control Registers | H’400 - H’4FF |
| Clock Control Registers | H’500 - H’67F |
| Clock Monitor Registers | H’680 - H’7FF |
| Reset Control Registers | H’800 - H’97F |
| Reset Monitor Registers | H’980 - H’AFF |
| Other Registers | H’B00 - H’FFF |

### 7.2.3 Register configuration

Table 7.5 Register list (1/4)

| Register Name | Abbreviation | R/W | Initial value*1 | Address*2 | Access Size |
|---|---|---|---|---|---|
| PLL1 (SSCG) Standby Control Register | CPG_SAMPLL1_STBY | RW | H'0000 0005 | H'000 | 32 |
| PLL4 (SSCG) Standby Control Register | CPG_SAMPLL4_STBY | RW | H'0000 0004 | H'010 | 32 |
| PLL1 (SSCG) Output Clock Setting Register 1 | CPG_SAMPLL1_CLK1 | RW | H'0000 3E83 | H'004 | 32 |
| PLL4 (SSCG) Output Clock Setting Register 1 | CPG_SAMPLL4_CLK1 | RW | H'0000 3203 | H'014 | 32 |
| PLL1 (SSCG) Output Clock Setting Register 2 | CPG_SAMPLL1_CLK2 | RW | H'0008 2D01 | H'008 | 32 |
| PLL4 (SSCG) Output Clock Setting Register 2 | CPG_SAMPLL4_CLK2 | RW | H'0008 2400 | H'018 | 32 |
| PLL1 (SSCG) Monitor Register | CPG_SAMPLL1_MON | R | H'0000 0011 | H'00C | 32 |
| PLL4 (SSCG) Monitor Register | CPG_SAMPLL4_MON | R | H'0000 0000 | H'01C | 32 |
| PLL1_SEL_SETTING Register | CPG_PLL1_SETTING | RW | H'0000 0001 | H'040 | 32 |
| CPG_OTPPLL0_MON Register | CPG_OTPPLL0_MON | R | H'xxxx xxxx | H'044 | 32 |
| CPG_OTPPLL1_MON Register | CPG_OTPPLL1_MON | R | H'xxxx xxxx | H'048 | 32 |
| CPG_OTPPLL2_MON Register | CPG_OTPPLL2_MON | R | H'xxxx xxxx | H'04C | 32 |
| PLL5 (SSCG) Standby Control Register | CPG_SIPLL5_STBY | RW | H'0000 0014 | H'140 | 32 |
| PLL5 (SSCG) Output Clock Setting Register 1 | CPG_SIPLL5_CLK1 | RW | H'0000 0111 | H'144 | 32 |
| PLL5 (SSCG) Output Clock Setting Register 2 | CPG_SIPLL5_CLK2 | RW | H'0000 0100 | H'148 | 32 |
| PLL5 (SSCG) Output Clock Setting Register 3 | CPG_SIPLL5_CLK3 | RW | H'0000 0006 | H'14C | 32 |
| PLL5 (SSCG) Output Clock Setting Register 4 | CPG_SIPLL5_CLK4 | RW | H'007D 00FF | H'150 | 32 |
| PLL5 (SSCG) Output Clock Setting Register 5 | CPG_SIPLL5_CLK5 | RW | H'0000 0016 | H'154 | 32 |
| PLL2(SSCG) Monitor Register | CPG_SIPLL2_MON | R | H'0000 0011 | H'11C | 32 |
| PLL3(SSCG) Monitor Register | CPG_SIPLL3_MON | R | H'0000 0011 | H'13C | 32 |
| PLL5(SSCG) Monitor Register | CPG_SIPLL5_MON | R | H'0000 0000 | H'15C | 32 |
| Division Ratio Setting (PLL1) Register | CPG_PL1_DDIV | RW | H'0000 000x | H'200 | 32 |
| Division Ratio Setting (PLL2) Register | CPG_PL2_DDIV | RW | H'0000 0000 | H'204 | 32 |
| Division Ratio Setting (PLL3A) Register | CPG_PL3A_DDIV | RW | H'0000 0000 | H'208 | 32 |
| Division Ratio Setting (PLL3B) Register | CPG_PL3B_DDIV | RW | H'0000 0000 | H'20C | 32 |
| Source Clock Setting (SDHI) Register | CPG_PL2SDHI_DSEL | RW | H'0000 0011 | H'218 | 32 |
| Source Clock Setting (DDR) Register | CPG_PL4_DSEL | RW | H'0000 0000 | H'21C | 32 |
| Clock Status Monitor Register | CPG_CLKSTATUS | R | H'0000 0000 | H'280 | 32 |
| Source Clock Setting Register | CPG_PL3_SSEL | RW | H'0000 1111 | H'408 | 32 |
| Division Ratio Setting (PLL5) Register | CPG_PL5_SDIV | RW | H'0000 0001 | H'420 | 32 |
| Clock Control Register Cortex-A55 | CPG_CLKON_CA55 | RW | H'0000 003F | H'500 | 32 |
| Clock Control Register ACPU | CPG_CLKON_SRAM_ACPU | RW | H'0000 0001 | H'508 | 32 |
| Clock Control Register MCPU | CPG_CLKON_SRAM_MCPU | RW | H'0000 0001 | H'50C | 32 |
| Clock Control Register GIC600 | CPG_CLKON_GIC600 | RW | H'0000 0001 | H'514 | 32 |
| Clock Control Register IA55 | CPG_CLKON_IA55 | RW | H'0000 0003 | H'518 | 32 |
| Clock Control Register CST | CPG_CLKON_CST | RW | H'0000 07FF | H'524 | 32 |
| Clock Control Register SYC | CPG_CLKON_SYC | RW | H'0000 0000 | H'528 | 32 |
| Clock Control Register DMAC_REG | CPG_CLKON_DAMC_REG | RW | H'0000 0000 | H'52C | 32 |
| Clock Control Register GTM | CPG_CLKON_GTM | RW | H'0000 0001 | H'534 | 32 |

Table 7.5 Register list (2/4)

| Register Name | Abbreviation | R/W | Initial value*1 | Address*2 | Access Size |
|---|---|---|---|---|---|
| Clock Control Register MTU | CPG_CLKON_MTU | RW | H'0000 0000 | H'538 | 32 |
| Clock Control Register POE3 | CPG_CLKON_POE3 | RW | H'0000 0000 | H'53C | 32 |
| Clock Control Register WDT | CPG_CLKON_WDT | RW | H'0000 00C3 | H'548 | 32 |
| Clock Control Register DDR | CPG_CLKON_DDR | RW | H'0000 0000 | H'54C | 32 |
| Clock Control Register SPI_MULTI | CPG_CLKON_SPI_MULTI | RW | H'0000 0000 | H'550 | 32 |
| Clock Control Register SDHI | CPG_CLKON_SDHI | RW | H'0000 0000 | H'554 | 32 |
| Clock Control Register MIPI_DSI | CPG_CLKON_MIPI_DSI | RW | H'0000 0000 | H'568 | 32 |
| Clock Control Register LCDC | CPG_CLKON_LCDC | RW | H'0000 0000 | H'56C | 32 |
| Clock Control Register SSI | CPG_CLKON_SSI | RW | H'0000 0000 | H'570 | 32 |
| Clock Control Register USB | CPG_CLKON_USB | RW | H'0000 0000 | H'578 | 32 |
| Clock Control Register I2C | CPG_CLKON_I2C | RW | H'0000 0000 | H'580 | 32 |
| Clock Control Register SCIF | CPG_CLKON_SCIF | RW | H'0000 0000 | H'584 | 32 |
| Clock Control Register SCI | CPG_CLKON_SCI | RW | H'0000 0000 | H'588 | 32 |
| Clock Control Register IRDA | CPG_CLKON_IRDA | RW | H'0000 0000 | H'58C | 32 |
| Clock Control Register RSPI | CPG_CLKON_RSPI | RW | H'0000 0000 | H'590 | 32 |
| Clock Control Register GPIO | CPG_CLKON_GPIO | RW | H'0000 0000 | H'598 | 32 |
| Clock Control Register TSU | CPG_CLKON_TSU | RW | H'0000 0000 | H'5AC | 32 |
| Clock Control Register DRW | CPG_CLKON_DRW | RW | H'0000 0000 | H'600 | 32 |
| Clock Monitor Register Cortex-A55 | CPG_CLKMON_CA55 | R | H'0000 003F | H'680 | 32 |
| Clock Monitor Register SRAM_ACPU | CPG_CLKMON_SRAM_ACPU | R | H'0000 0001 | H'688 | 32 |
| Clock Monitor Register SRAM_MCPU | CPG_CLKMON_SRAM_MCPU | R | H'0000 0001 | H'68C | 32 |
| Clock Monitor Register GIC600 | CPG_CLKMON_GIC600 | R | H'0000 0001 | H'694 | 32 |
| Clock Monitor Register IA55 | CPG_CLKMON_IA55 | R | H'0000 0003 | H'698 | 32 |
| Clock Monitor Register CST | CPG_CLKMON_CST | R | H'0000 07FF | H'6A4 | 32 |
| Clock Monitor Register SYC | CPG_CLKMON_SYC | R | H'0000 0000 | H'6A8 | 32 |
| Clock Monitor Register DMAC_REG | CPG_CLKMON_DMAC_REG | R | H'0000 0000 | H'6AC | 32 |
| Clock Monitor Register GTM | CPG_CLKMON_GTM | R | H'0000 0001 | H'6B4 | 32 |
| Clock Monitor Register MTU | CPG_CLKMON_MTU | R | H'0000 0000 | H'6B8 | 32 |
| Clock Monitor Register POE3 | CPG_CLKMON_POE3 | R | H'0000 0000 | H'6BC | 32 |
| Clock Monitor Register WDT | CPG_CLKMON_WDT | R | H'0000 00C3 | H'6C8 | 32 |
| Clock Monitor Register DDR | CPG_CLKMON_DDR | R | H'0000 0000 | H'6CC | 32 |
| Clock Monitor Register SPI_MULTI | CPG_CLKMON_SPI_MULTI | R | H'0000 0000 | H'6D0 | 32 |
| Clock Monitor Register SDHI | CPG_CLKMON_SDHI | R | H'0000 0000 | H'6D4 | 32 |
| Clock Monitor Register MIPI_DSI | CPG_CLKMON_MIPI_DSI | R | H'0000 0000 | H'6E8 | 32 |
| Clock Monitor Register LCDC | CPG_CLKMON_LCDC | R | H'0000 0000 | H'6EC | 32 |
| Clock Monitor Register SSI | CPG_CLKMON_SSI | R | H'0000 0000 | H'6F0 | 32 |
| Clock Monitor Register USB | CPG_CLKMON_USB | R | H'0000 0000 | H'6F8 | 32 |
| Clock Monitor Register I2C | CPG_CLKMON_I2C | R | H'0000 0000 | H'700 | 32 |
| Clock Monitor Register SCIF | CPG_CLKMON_SCIF | R | H'0000 0000 | H'704 | 32 |
| Clock Monitor Register SCI | CPG_CLKMON_SCI | R | H'0000 0000 | H'708 | 32 |
| Clock Monitor Register IRDA | CPG_CLKMON_IRDA | R | H'0000 0000 | H'70C | 32 |
| Clock Monitor Register RSPI | CPG_CLKMON_RSPI | R | H'0000 0000 | H'710 | 32 |

Table 7.5 Register list (3/4)

| Register Name | Abbreviation | R/W | Initial value*1 | Address*2 | Access Size |
|---|---|---|---|---|---|
| Clock Monitor Register GPIO | CPG_CLKMON_GPIO | R | H'0000 0000 | H'718 | 32 |
| Clock Monitor Register TSU | CPG_CLKMON_TSU | R | H'0000 0000 | H'72C | 32 |
| Clock Monitor Register DRW | CPG_CLKMON_DRW | R | H'0000 0000 | H'780 | 32 |
| Reset Control Register Cortex-A55 | CPG_RST_CA55 | RW | H'00001FFF | H'800 | 32 |
| Reset Control Register SRAM_ACPU | CPG_RST_SRAM_ACPU | RW | H'0000 0001 | H'808 | 32 |
| Reset Control Register SRAM_MCPU | CPG_RST_SRAM_MCPU | RW | H'0000 0001 | H'80C | 32 |
| Reset Control Register GIC600 | CPG_RST_GIC600 | RW | H'0000 0003 | H'814 | 32 |
| Reset Control Register IA55 | CPG_RST_IA55 | RW | H'0000 0001 | H'818 | 32 |
| Reset Control Register SYC | CPG_RST_SYC | RW | H'0000 0000 | H'828 | 32 |
| Reset Control Register DMAC | CPG_RST_DMAC | RW | H'0000 0000 | H'82C | 32 |
| Reset Control Register GTM | CPG_RST_GTM | RW | H'0000 0001 | H'834 | 32 |
| Reset Control Register MTU | CPG_RST_MTU | RW | H'0000 0000 | H'838 | 32 |
| Reset Control Register POE3 | CPG_RST_POE3 | RW | H'0000 0000 | H'83C | 32 |
| Reset Control Register WDT | CPG_RST_WDT | RW | H'0000 0009 | H'848 | 32 |
| Reset Control Register DDR | CPG_RST_DDR | RW | H'0000 0000 | H'84C | 32 |
| Reset Control Register SPI | CPG_RST_SPI | RW | H'0000 0000 | H'850 | 32 |
| Reset Control Register SDHI | CPG_RST_SDHI | RW | H'0000 0000 | H'854 | 32 |
| Reset Control Register MIPI_DSI | CPG_RST_MIPI_DSI | RW | H'0000 0000 | H'868 | 32 |
| Reset Control Register LCDC | CPG_RST_LCDC | RW | H'0000 0000 | H'86C | 32 |
| Reset Control Register SSIF | CPG_RST_SSIF | RW | H'0000 0000 | H'870 | 32 |
| Reset Control Register USB | CPG_RST_USB | RW | H'0000 0000 | H'878 | 32 |
| Reset Control Register I2C | CPG_RST_I2C | RW | H'0000 0000 | H'880 | 32 |
| Reset Control Register SCIF | CPG_RST_SCIF | RW | H'0000 0000 | H'884 | 32 |
| Reset Control Register SCI | CPG_RST_SCI | RW | H'0000 0000 | H'888 | 32 |
| Reset Control Register IRDA | CPG_RST_IRDA | RW | H'0000 0000 | H'88C | 32 |
| Reset Control Register RSPI | CPG_RST_RSPI | RW | H'0000 0000 | H'890 | 32 |
| Reset Control Register GPIO | CPG_RST_GPIO | RW | H'0000 0000 | H'898 | 32 |
| Reset Control Register TSU | CPG_RST_TSU | RW | H'0000 0000 | H'8AC | 32 |
| Reset Control Register DRW | CPG_RST_DRW | RW | H'0000 0000 | H'900 | 32 |
| Reset Monitor Register Cortex-A55 | CPG_RSTMON_CA55 | R | H'0000 0000 | H'980 | 32 |
| Reset Monitor Register SRAM_ACPU | CPG_RSTMON_SRAM_ACPU | R | H'0000 0000 | H'988 | 32 |
| Reset Monitor Register SRAM_MCPU | CPG_RSTMON_SRAM_MCPU | R | H'0000 0000 | H'98C | 32 |
| Reset Monitor Register GIC600 | CPG_RSTMON_GIC600 | R | H'0000 0000 | H'994 | 32 |
| Reset Monitor Register IA55 | CPG_RSTMON_IA55 | R | H'0000 0000 | H'998 | 32 |
| Reset Monitor Register SYC | CPG_RSTMON_SYC | R | H'0000 0001 | H'9A8 | 32 |
| Reset Monitor Register DMAC | CPG_RSTMON_DMAC | R | H'0000 0003 | H'9AC | 32 |
| Reset Monitor Register GTM | CPG_RSTMON_GTM | R | H'0000 0006 | H'9B4 | 32 |
| Reset Monitor Register MTU | CPG_RSTMON_MTU | R | H'0000 0001 | H'9B8 | 32 |
| Reset Monitor Register POE3 | CPG_RSTMON_POE3 | R | H'0000 0001 | H'9BC | 32 |
| Reset Monitor Register WDT | CPG_RSTMON_WDT | R | H'0000 0006 | H'9C8 | 32 |
| Reset Monitor Register DDR | CPG_RSTMON_DDR | R | H'0000 007F | H'9CC | 32 |
| Reset Monitor Register SPI | CPG_RSTMON_SPI | R | H'0000 0001 | H'9D0 | 32 |

Table 7.5 Register list (4/4)

| Register Name | Abbreviation | R/W | Initial value*1 | Address*2 | Access Size |
|---|---|---|---|---|---|
| Reset Monitor Register SDHI | CPG_RSTMON_SDHI | R | H'0000 0003 | H'9D4 | 32 |
| Reset Monitor Register MIPI_DSI | CPG_RSTMON_MIPI_DSI | R | H'0000 0007 | H'9E8 | 32 |
| Reset Monitor Register LCDC | CPG_RSTMON_LCDC | R | H'0000 0001 | H'9EC | 32 |
| Reset Monitor Register SSIF | CPG_RSTMON_SSIF | R | H'0000 000F | H'9F0 | 32 |
| Reset Monitor Register USB | CPG_RSTMON_USB | R | H'0000 000F | H'9F8 | 32 |
| Reset Monitor Register I2C | CPG_RSTMON_I2C | R | H'0000 000F | H'A00 | 32 |
| Reset Monitor Register SCIF | CPG_RSTMON_SCIF | R | H'0000 001F | H'A04 | 32 |
| Reset Monitor Register SCI | CPG_RSTMON_SCI | R | H'0000 0003 | H'A08 | 32 |
| Reset Monitor Register IRDA | CPG_RSTMON_IRDA | R | H'0000 0001 | H'A0C | 32 |
| Reset Monitor Register RSPI | CPG_RSTMON_RSPI | R | H'0000 0007 | H'A10 | 32 |
| Reset Monitor Register GPIO | CPG_RSTMON_GPIO | R | H'0000 0007 | H'A18 | 32 |
| Reset Monitor Register TSU | CPG_RSTMON_TSU | R | H'0000 0001 | H'A2C | 32 |
| Reset Monitor Register DRW | CPG_RSTMON_DRW | R | H'0000 0001 | H'A80 | 32 |
| WDT Overflow System Reset Register | CPG_WDTOVF_RST | RW | H'0000 0000 | H'B10 | 32 |
| WDT Reset Selector Register | CPG_WDTRST_SEL | RW | H'0000 0088 | H'B14 | 32 |
| MSTOP Register ACPU | CPG_BUS_ACPU_MSTOP | RW | H'0000 0000 | H'B60 | 32 |
| MSTOP Register MCPU1 | CPG_BUS_MCPU1_MSTOP | RW | H'0000 0000 | H'B64 | 32 |
| MSTOP Register MCPU2 | CPG_BUS_MCPU2_MSTOP | RW | H'0000 0000 | H'B68 | 32 |
| MSTOP Register PERI_COM | CPG_BUS_PERI_COM_MSTOP | RW | H'0000 0000 | H'B6C | 32 |
| MSTOP Register PERI_CPU | CPG_BUS_PERI_CPU_MSTOP | RW | H'0000 0000 | H'B70 | 32 |
| MSTOP Register PERI_DDR | CPG_BUS_PERI_DDR_MSTOP | RW | H'0000 0000 | H'B74 | 32 |
| MSTOP Register PERI_VIDEO | CPG_BUS_PERI_VIDEO_MSTOP | RW | H'0000 0000 | H'B78 | 32 |
| MSTOP Register REG0 | CPG_BUS_REG0_MSTOP | RW | H'0000 0000 | H'B7C | 32 |
| MSTOP Register REG1 | CPG_BUS_REG1_MSTOP | RW | H'0000 0000 | H'B80 | 32 |
| Other Function Register 1 | CPG_OTHERFUNC1_REG | RW | H'0000 0000 | H'BE8 | 32 |
| Other Function Register 2 | CPG_OTHERFUNC2_REG | RW | H'0000 0000 | H'BEC | 32 |
| Clock Control Register MBUS | CPG_CLKON_MBUS | RW | H'0000 0FFF | H'5B8 | 32 |
| Clock Monitor Register MBUS | CPG_CLKMON_MBUS | R | H'0000 0FFF | H'738 | 32 |

Note 1. **Initial value:** Release of system reset The reset is released according to the sequence, and the value immediately after the last reset release in the sequence.

Note 2. **Address:** The offset address from the base address.

## 7.2.4 Register Descriptions

### 7.2.4.1 PLLn (SSCG) Standby Control Register (CPG_SAMPLLn_STBY) (n = 1 or 4)

This register is used to control the power-saving mode and standby state and enable or disable the SSCG.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | SSCG_MODE_WEN | — | SSCG_EN_WEN | — | RESETB_WEN |
| Initial Value n = 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Initial Value n = 4 | 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 | R0W1 | R | R0W1 | R | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | SSCG_MODE[1] | SSCG_MODE[0] | — | SSCG_EN | — | RESETB |
| Initial Value n = 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 |
| Initial Value n = 4 | 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 | RW | RW | R | RW | R | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 21 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 20 | SSCG_MODE_WEN | 0b | R0W1 | Flag for enabling the writing to the SSCG_MODE[1:0] bits (bits 5 and 4)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 19 | — | 0b | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 18 | SSCG_EN_WEN | 0b | R0W1 | Flag for enabling the writing to the SSCG_EN bit (bit 2)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 17 | — | 0b | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 16 | RESETB_WEN | 0b | R0W1 | Flag for enabling the writing to the RESETB bit (bit 0)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 6 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 5, 4 | SSCG_MODE[1:0] | 00b | RW | SSCG center spread or down spread setting (see the mode setting table below.*1)<br>When writing to these bits, set the SSCG_MODE_WEN bit to 1 at the same time.<br>00b: Down spread<br>01b: Setting prohibited<br>10b: Setting prohibited<br>11b: Setting prohibited |
| 3 | — | 0b | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 2 | SSCG_EN | n=1,4: 1b | RW | SSCG enable or disable setting (see the mode setting table below.*1)<br>When writing to this bit, set the SSCG_EN_WEN bit to 1 at the same time.<br>0: SSCG is disabled.<br>1: SSCG is enabled. |
| 1 | — | 0b | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 0 | RESETB | n=1: 1b<br>n=4: 0b | RW | PLL reset setting (see the mode setting table below.*1)<br>When writing to this bit, set the RESETB_WEN bit to 1 at the same time.<br>0: Reset state<br>1: Active state (PLL is released from the reset state.) |

Note 1. Mode Setting Table

| Mode | | Bit Name | | |
|---|---|---|---|---|
| | | RESETB | SSCG_EN | SSCG_MODE[1:0] |
| Reset state | | 0 | — | — |
| SSCG disabled | | 1 | 0 | — |
| SSCG enabled | Down spread | | 1 | 00b |
| Setting prohibited | | | | 11b |

### 7.2.4.2 PLLn (SSCG) Output Clock Setting Register 1 (CPG_SAMPLLn_CLK1) (n = 1 or 4)

This register is used to specify the frequency division values k, m, and p for SSCG PLLn (n = 1 or 4).

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | DIV_K | | | | | | | | | | | | | | | |
| Initial Value n = 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Initial Value n = 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | DIV_M | | | | | | | | | | DIV_P | | | | | |
| Initial Value n = 1 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| Initial Value n = 4 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| R/W | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 16 | DIV_K | All 0 | RW | Delta-sigma modulator (DSM) setting*4<br>Specify a 2's complement of a decimal value from -32,768 to 32,767 that satisfies the restrictions*1.<br>1000_0000_0000_0000b: k = -32,768<br>1000_0000_0000_0001b: k = -32,767<br>1000_0000_0000_0010b: k = -32,766<br>:<br>0111_1111_1111_1101b: k = +32,765<br>0111_1111_1111_1110b: k = +32,766<br>0111_1111_1111_1111b: k = +32,767 |
| 15 to 6 | DIV_M | n = 1: 00_1111_1010b<br>n = 4: 00_1100_1000b | RW | Main divider setting*4<br>Specify an unsigned binary value from 64 to 533 that satisfies the restrictions*1 *2.<br>11_1111_1111b to 10_0001_0110b (Division value m = 534): Setting prohibited<br>10_0001_0101b: Division value m = 533<br>10_0001_0100b: Division value m = 532<br>:<br>00_0100_0001b: Division value m = 65<br>00_0100_0000b: Division value m = 64<br>00_0011_1111b to 00_0000_0000b: Setting prohibited |
| 5 to 0 | DIV_P | 'n = 1: 00_0011b<br>'n = 4: 00_0011b | RW | Pre-divider setting*4<br>Specify a value from 1 to 8 that satisfies the restrictions*1 *2.<br>11_1111b to 00_00_0101b: Setting prohibited<br>00_0100b: Division value p = 4 (FFREF = 6 MHz)<br>00_0011:b Division value p = 3 (FFREF = 8 MHz)<br>00_0010b: Division value p = 2 (FFREF = 12 MHz)<br>00_0001b: Division value p = 1 (FFREF = 24 MHz)<br>00_0000b: Setting prohibited |

Note 1. The PLL oscillation frequency ($F_{FVCO}$) and output frequency ($F_{FOUT}$) can be calculated by the following equations.

- Equations

$$F_{FVCO} = \frac{((m + k/65536) \times F_{Fin})}{p}$$

$$F_{Fout} = \frac{((m + k/65536) \times F_{Fin})^{*3}}{(p \times 2^{s})}$$

- Restrictions
  $F_{Fin}$: 24 MHz
  6 MHz ≤ $F_{FREF}$ = $F_{FIN}$/p ≤ 30 MHz
  1,125 MHz ≤ $F_{FVCO}$ ≤ 2,500 MHz

Note 2. Specify DIV_K, DIV_M, DIV_P, and DIV_S*3 within the allowable ranges of $F_{FVCO}$ and $F_{Fout}$ for each PLL shown below.

| SSCG PLL | FFout Min. | Default FFout Setting | FFout Max. |
|---|---|---|---|
| PLL1 | 1,000 MHz | 1,000 MHz | 1,000 MHz |
| PLL4 | 800 MHz | 1,600 MHz | 1,600 MHz |

Note 3. For DIV_S (value “s”), see **Section 7.2.4.3, PLLn (SSCG) Output Clock Setting Register 2 (CPG_SAMPLLn_CLK2) (n = 1 or 4)**.

Note 4. The PLL should be reset after the PLL settings are modified. For details, refer to **Section 7.4.5, Procedures for PLL Setup**.

### 7.2.4.3 PLLn (SSCG) Output Clock Setting Register 2 (CPG_SAMPLLn_CLK2) (n = 1 or 4)

This register is used to specify the frequency division value s and SSCG modulation values mfr and mrr for SSCG PLLn (n = 1 or 4).

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | MFR | | | | | | | |
| Initial Value n = 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| Initial Value n = 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | RW | RW | RW | RW | RW | RW | RW | RW |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | MRR | | | | | | — | — | — | — | — | DIV_S | | |
| Initial Value n = 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| Initial Value n = 4 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | RW | RW | RW | RW | RW | RW | R | R | R | R | R | RW | RW | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 24 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 23 to 16 | MFR | n = 1: H'08<br>n = 4: H'08 | RW | Modulation frequency setting<br>The PLL should be reset after this setting is modified. For details, refer to **Section 7.4.5, Procedures for PLL Setup**.<br>Specify a value from 0 to 255 (decimal).*1<br>1111_1111b: mfr = 255<br>1111_1110b: mfr = 254<br>:<br>0000_0001b: mfr = 1<br>0000_0000b: mfr = 0 |
| 15, 14 | — | 00b | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 13 to 8 | MRR | 'n = 1: 10_1101b<br>'n = 4: 10_0100b | RW | Modulation ratio setting<br>The PLL should be reset after this setting is modified. For details, refer to **Section 7.4.5, Procedures for PLL Setup**.<br>Specify a value from 1 to 63 (decimal).*1<br>11_1111b: mrr = 63<br>11_1110b: mrr = 62<br>:<br>00_0010b: mrr = 2<br>00_0001b: mrr = 1<br>00_0000b: Setting prohibited |
| 7 to 3 | — | 00000b | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 2 to 0 | DIV_S | ‘n = 1: 001b<br>‘n = 4: 000b | RW | Divider s setting*2<br>Specify a value from 0 to 6 (decimal).<br>111b: Setting prohibited<br>110b: Division value s = 6 (Division ratio = 64)<br>101b: Division value s = 5 (Division ratio = 32)<br>100b: Division value s = 4 (Division ratio = 16)<br>011b: Division value s = 3 (Division ratio = 8)<br>010b: Division value s = 2 (Division ratio = 4)<br>001b: Division value s = 1 (Division ratio = 2)<br>000b: Division value s = 0 (Division ratio = 1) |

Note 1. The modulation frequency (MF) and modulation ratio (pk-pk) (MR) can be calculated by the following equations.

- Equations

$$MF = \frac{(F_{Fin})}{p \times mfr \times (2^5)}$$

$$MR = \frac{mf \times mrr}{m \times (2^6)} \times 100[\%]$$

- Restrictions

  $F_{Fin}$: 24 MHz

  0000_0000b ≤ MFR[7:0] ≤ 1111_1111b

  00_0001b ≤ MRR[5:0] ≤ 11_1111b

  0 ≤ mrr × mfr ≤ 512

Note 2. For DIV_M (value “m”) and DIV_P (value “p”), see **Section 7.2.4.2, PLLn (SSCG) Output Clock Setting Register 1 (CPG_SAMPLLn_CLK1) (n = 1 or 4)**.

Note 3. Change the SSC setting of PLL4 from the default setting to the following value.

- Modulation Depth [%]: −1.0%
- Fmod [kHz]: About 30 kHz

**Important:**

For the procedures for setting up the PLL, see **Section 7.4.5, Procedures for PLL Setup**.

**Initial Values:**

The initial values of the registers for PLL1 and PLL4 and output clocks are shown below.

- Initial values of registers

| | CPG_SAMPLLn_STBY (n = 1 or 4) | | |
|---|---|---|---|
| | SSCG_MODE | SSCG_EN | RESETB |
| PLL | Bits 5 and 4 | Bit 2 | Bit 0 |
| PLL1 | H'0 (Down spread) | 1 (SSCG is enabled) | 1 (Released from the reset state) |
| PLL4 | H'0 (Down spread) | 1 (SSCG is enabled) | 0 (Reset state) |

| | CPG_SAMPLLn_CLK1 (n = 1 or 4) | | | CPG_SAMPLLn_CLK2 (n = 1 or 4) | | |
|---|---|---|---|---|---|---|
| | DIV_K | DIV_M | DIV_P | DIV_S | MFR | MRR |
| PLL | Bits 31 to 16 | Bits 15 to 6 | Bits 5 to 0 | Bits 2 to 0 | Bits 23 to 16 | Bits 13 to 8 |
| PLL1 | H'0000 (k = 0) | H'0FA (m = 250) | H'3 (p = 3) | H'1 (s = 1) | H'08 (mfr = 8) | H'2D (mrr = 45) |
| PLL4 | H'0000 (k = 0) | H'0C8 (m = 200) | H'3 (p = 3) | H'0 (s = 0) | H'08 (mfr = 8) | H'24 (mrr = 36) |

- Output clocks (initial values)

| PLL | PLL Operating Mode | Frequency ($F_{Fout}$) [MHz] | SSCG | Fmod [kHz] | Spread | Modulation Depth [%] |
|---|---|---|---|---|---|---|
| PLL1 | Active state | 1,000 | Enabled | 31.25 | Down spread | 2.25 |
| PLL4 | Reset state | 1,600 | Enabled | 31.25 | Down spread | 2.25 |

### 7.2.4.4 PLLn (SSCG) Monitor Register (CPG_SAMPLLn_MON) (n = 1 or 4)

This register is used to monitor the state of the SSCG PLLn (n = 1 or 4).

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Initial Value n = 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Initial Value n = 4 | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | PLLn_LOCK | — | — | — | PLLn_RESETB |
| Initial Value n = 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| Initial Value n = 4 | 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<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 4 | PLLn_LOCK | n = 1: 1b<br>n = 4: 0b | R | SSCG PLL lock state monitoring<br>0: PLL is not locked.<br>1: PLL is locked. |
| 3 to 1 | — | 000b | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 0 | PLLn_RESETB | n = 1: 1b<br>n = 4: 0b | R | SSCG PLL operating mode monitoring<br>1: Normal mode<br>0: Reset state (standby mode) |

### 7.2.4.5 PLL1_SEL_SETTING Register (CPG_PLL1_SETTING)

This register is used to specify the frequency of PLL1.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | SEL_PLL1_WEN |
| 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 | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | SEL_PLL1 |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 17 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 16 | SEL_PLL1_WEN | 0b | R0W1 | Flag for enabling the writing to the SEL_PLL1 bit. This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 1 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 0 | SEL_PLL1 | 1b | RW | The PLL multiplication factor is specified.<br>0: PLL1 operates according to the settings in the PLL1 SYSREG of the CPG.<br>1: When bit 0 of OTP_OTPPLL0 is 1, PLL1 operates according to the settings of OTP_OTPPLL0 to OTP_OTPPLL2.<br>When bit 0 of OTP_OTPPLL0 is 0, PLL1 operates according to the settings in the PLL1 SYSREG of the CPG. |

### 7.2.4.6 CPG_OTPPLL0_MON Register (CPG_OTPPLL0_MON)

This register indicates the state of the OTP_OTPPLL0[31:0] pins of the CPG.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | OTP31_MON | OTP30_MON | OTP29_MON | OTP28_MON | OTP27_MON | OTP26_MON | OTP25_MON | OTP24_MON | OTP23_MON | OTP22_MON | OTP21_MON | OTP20_MON | OTP19_MON | OTP18_MON | OTP17_MON | OTP16_MON |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | OTP15_MON | OTP14_MON | OTP13_MON | OTP12_MON | OTP11_MON | OTP10_MON | OTP9_MON | OTP8_MON | OTP7_MON | OTP6_MON | OTP5_MON | OTP4_MON | OTP3_MON | OTP2_MON | OTP1_MON | OTP0_MON |
| 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 0 | OTPn_MON [n = 31 to 0] | — | R | These bits indicate the state of the OTP_OTPPLL0[31:0] pins of the CPG.<br>*Note*: The values specified in the OTP are set in these bits as the initial values. |

### 7.2.4.7 CPG_OTPPLL1_MON Register (CPG_OTPPLL1_MON)

This register indicates the state of the OTP_OTPPLL1[31:0] pins of the CPG.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | OTP31_MON | OTP30_MON | OTP29_MON | OTP28_MON | OTP27_MON | OTP26_MON | OTP25_MON | OTP24_MON | OTP23_MON | OTP22_MON | OTP21_MON | OTP20_MON | OTP19_MON | OTP18_MON | OTP17_MON | OTP16_MON |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | OTP15_MON | OTP14_MON | OTP13_MON | OTP12_MON | OTP11_MON | OTP10_MON | OTP9_MON | OTP8_MON | OTP7_MON | OTP6_MON | OTP5_MON | OTP4_MON | OTP3_MON | OTP2_MON | OTP1_MON | OTP0_MON |
| 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 0 | OTPn_MON [n = 31 to 0] | — | R | These bits indicate the state of the OTP_OTPPLL1[31:0] pins of the CPG.<br>*Note*: The values specified in the OTP are set in these bits as the initial values. |

### 7.2.4.8 CPG_OTPPLL2_MON Register (CPG_OTPPLL2_MON)

This register indicates the state of the OTP_OTPPLL2[31:0] pins of the CPG.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | OTP31_MON | OTP30_MON | OTP29_MON | OTP28_MON | OTP27_MON | OTP26_MON | OTP25_MON | OTP24_MON | OTP23_MON | OTP22_MON | OTP21_MON | OTP20_MON | OTP19_MON | OTP18_MON | OTP17_MON | OTP16_MON |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | OTP15_MON | OTP14_MON | OTP13_MON | OTP12_MON | OTP11_MON | OTP10_MON | OTP9_MON | OTP8_MON | OTP7_MON | OTP6_MON | OTP5_MON | OTP4_MON | OTP3_MON | OTP2_MON | OTP1_MON | OTP0_MON |
| 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 0 | OTPn_MON [n = 31 to 0] | — | R | These bits indicate the state of the OTP_OTPPLL2[31:0] pins of the CPG.<br>*Note*: The values specified in the OTP are set in these bits as the initial values. |

### 7.2.4.9 PLL5 (SSCG) Standby Control Register (CPG_SIPLL5_STBY)

This register is used to control the power-saving mode and standby state and enable or disable the SSC in PLL5 that operates at 3 GHz or a higher frequency. When enable PLL5, change to SSCG_EN=0.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | DOWNSPREAD_WEN | — | SSCG_EN_WEN | — | RESETB_WEN |
| 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 | R0W1 | R | R0W1 | R | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | DOWNSPREAD | — | SSCG_EN | — | RESETB |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | RW | R | RW | R | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 21 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 20 | DOWNSPREAD_WEN | 0b | R0W1 | Flag for enabling the writing to the DOWNSPREAD bit (bit 4)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 19 | — | 0b | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 18 | SSCG_EN_WEN | 0b | R0W1 | Flag for enabling the writing to the SSCG_EN bit (bit 2)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 17 | — | 0b | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 16 | RESETB_WEN | 0b | R0W1 | Flag for enabling the writing to the RESETB bit (bit 0)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 5 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 4 | DOWNSPREAD | 1b | RW | SSC center spread or down spread setting<br>When writing to this bit, set the DOWNSPREAD_WEN bit to 1 at the same time.<br>0: Setting prohibited<br>1: Down spread |
| 3 | — | 0b | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 2 | SSCG_EN*2 | 1b | RW | SSC enable or disable setting (see the mode setting table below.*1)<br>When writing to this bit, set the SSCG_EN_WEN bit to 1 at the same time.<br>0: SSC is disabled.<br>1: SSC is enabled. |
| 1 | — | 0b | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 0 | RESETB | 0b | RW | PLL reset setting (see the mode setting table below.*1)<br>When writing to this bit, set the RESETB_WEN bit to 1 at the same time.<br>0: Reset state<br>1: Active state (PLL is released from the reset state.) |

Note 1. Mode Setting Table

Note 2. Set the SSCG_EN bit to 0, when operating PLL5.

| Mode | | Bit Name | | |
|---|---|---|---|---|
| | | RESETB | SSCG_EN | DOWNSPREAD |
| Reset state | | 0 | — | — |
| SSC disabled | | 1 | 0 | — |
| SSC enabled | Down spread | | 1 | 1b |

### 7.2.4.10 PLL5 (SSCG) Output Clock Setting Register 1 (CPG_SIPLL5_CLK1)

This register is used to set up PLL5.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | REFDIV | | | | | | — | POSTDIV2 | | | — | POSTDIV1 | | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| R/W | R | R | RW | RW | RW | RW | RW | RW | R | RW | RW | RW | R | RW | RW | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 14 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 13 to 8 | REFDIV | 00_0001b | RW | REFDIV setting for PLL<br>Specify a value from 1 to 63. |
| 7 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 6 to 4 | POSTDIV2 | 001b | RW | POSTDIV2 setting for PLL<br>Specify a value from 1 to 7.<br>The POSTDIV value is obtained by POSTDIV1 × POSTDIV2. |
| 3 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 2 to 0 | POSTDIV1 | 001b | RW | POSTDIV1 setting for PLL<br>Specify a value from 1 to 7.<br>The POSTDIV value is obtained by POSTDIV1 × POSTDIV2. |

### 7.2.4.11 PLL5 (SSCG) Output Clock Setting Register 2 (CPG_SIPLL5_CLK2)

This register is used to set up PLL5.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | FOUTVCOPD_WEN | — | FOUTPOSTDIVPD_WEN | — | FOUT4PHASEPD_WEN | — | DSMPD_WEN | — | DACPD_WEN |
| 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 | R0W1 | R | R0W1 | R | R0W1 | R | R0W1 | R | R0W1 |

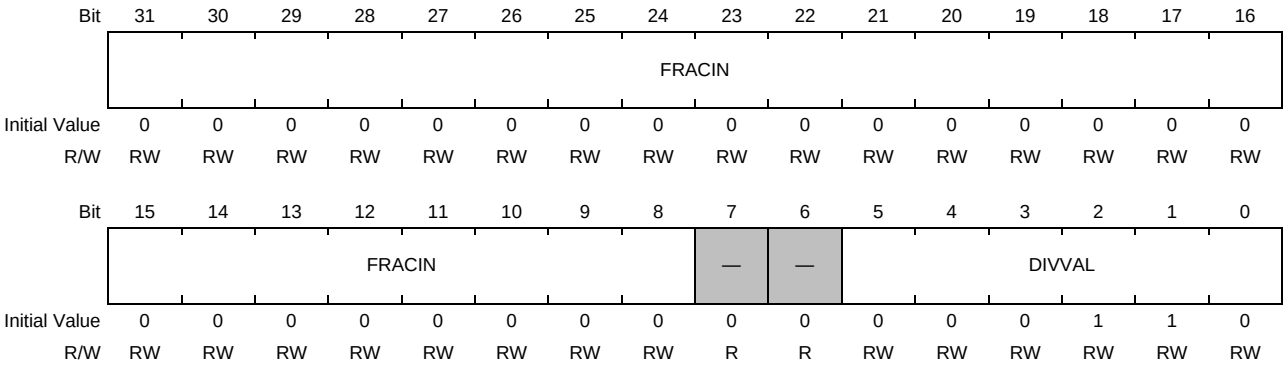
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | FOUTVCOPD | — | FOUTPOSTDIVPD | — | FOUT4PHASEPD | — | DSMPD | — | DACPD |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | RW | R | RW | R | RW | R | RW | R | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 25 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 24 | FOUTVCOPD_WEN | 0b | R0W1 | Flag for enabling the writing to the FOUTVCOPD bit (bit 8)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 23 | — | 0b | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 22 | FOUTPOSTDIVPD_WEN | 0b | R0W1 | Flag for enabling the writing to the FOUTPOSTDIVPD bit (bit 6)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 21 | — | 0b | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 20 | FOUT4PHASEPD_WEN | 0b | R0W1 | Flag for enabling the writing to the FOUT4PHASEPD bit (bit 4)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 19 | — | 0b | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 18 | DSMPD_WEN | 0b | R0W1 | Flag for enabling the writing to the DSMPD bit (bit 2)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 17 | — | 0b | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 16 | DACPD_WEN | 0b | R0W1 | Flag for enabling the writing to the DACPD bit (bit 0)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 9 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 8 | FOUTVCOPD | 1b | RW | FOUTVCOPD setting for the PLL<br>Power-down of FOUTVCO is specified.<br>0: FOUTVCO is active.<br>1: FOUTVCO power is turned down. |
| 7 | — | 0b | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 6 | FOUTPOSTDIVPD | 0b | RW | FOUTPOSTDIVPD setting for the PLL<br>Power-down of FOUTPOSTDIVPD is specified.<br>(Target: FOUTPOSTDIVPD, FOUT1PH0, FOUT2, FOUT3, and FOUT4)<br>0: FOUTPOSTDIVPD is active.<br>1: FOUTPOSTDIVPD power is turned down. |
| 5 | — | 0b | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 4 | FOUT4PHASEPD | 0b | RW | FOUT4PHASEPD setting for the PLL<br>Power-down of FOUT4PHASEPD is specified.<br>(Target: FOUT1PH0, FOUT2, FOUT3, and FOUT4)<br>0: FOUT4PHASEPD is active.<br>1: FOUT4PHASEPD power is turned down. |
| 3 | — | 0b | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 2 | DSMPD | 0b | RW | Delta-sigma modulator (DSM) power-down setting<br>0: DSM is active.<br>1: DSM power is turned down. |
| 1 | — | 0b | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 0 | DACPD | 0b | RW | Noise-cancelling DAC power-down setting in the FRAC mode<br>0: DAC is active (default mode).<br>1: DAC power is turned down (for test mode only). |

### 7.2.4.12 PLL5 (SSCG) Output Clock Setting Register 3 (CPG_SIPLL5_CLK3)

This register is used to set up the SSCG for PLL5.




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 8 | FRACIN | H'00_0000 | RW | Division value setting for frequency division |
| 7, 6 | — | 00b | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 5 to 0 | DIVVAL | 00_0110b | RW | Modulation frequency setting<br>When DIVVAL = 6'd0, the division value is set to 1.<br>When DIVVAL > 0, the DIVVAL value is used as the division value. |

### 7.2.4.13 PLL5 (SSCG) Output Clock Setting Register 4 (CPG_SIPLL5_CLK4)

This register is used to set up the SSCG for PLL5.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | INTIN | | | | | | | | | | | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| R/W | R | R | R | R | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW |

| 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 | 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 28 | — | All 0 | R | Reserved<br>When read, the initial value is read. The written value will be ignored. |
| 27 to 16 | INTIN | H’07D | RW | Setting of integer part of division value. |
| 15 to 8 | — | All 0 | R | Reserved<br>When read, the initial value is read. The written value will be ignored. |
| 7 to 0 | — | H’FF | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |

### 7.2.4.14 PLL5 (SSCG) Output Clock Setting Register 5 (CPG_SIPLL5_CLK5)

This register is used to set up the SSCG for PLL5.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | SPREAD | | | | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | RW | RW | RW | RW | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 5 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 4 to 0 | SPREAD | 1_0110b | RW | Spread depth setting<br>00000b: Setting prohibited<br>00001b: 0.1%<br>00010b: 0.2%<br>:<br>10000b: 1.6%<br>:<br>11111b: 3.1% |

**Important:**

For the procedures for setting up the PLL, see **Section 7.4.5, Procedures for PLL Setup**.

**Initial Values:**

The initial values of the registers for PLL5 and output clock are shown below.

- Initial values of registers

| | CPG_SIPLL5_STBY | | |
|---|---|---|---|
| | DOWNSPREAD | SSCG_EN | RESETB |
| PLL | Bit 4 | Bit 2 | Bit 0 |
| PLL5 | H'1 (Down spread) | 1 (SSCG is enabled) | 0 (Reset state) |

| | CPG_SIPLL5_CLK1 | | |
|---|---|---|---|
| | REFDIV | POSTDIV2 | POSTDIV1 |
| PLL | Bits 13 to 8 | Bits 6 to 4 | Bits 2 to 0 |
| PLL5 | H'01 (REFDIV = 1) | H'1 (POSTDIV2 = 1) | H'1 (POSTDIV1 = 1) |

| | CPG_SIPLL5_CLK3 | |
|---|---|---|
| | FRACIN | DIVVAL |
| PLL | Bits 31 to 8 | Bits 5 to 0 |
| PLL5 | H'00_0000 (FRACIN = 0) | H'6 (DIVVAL = 6) |

| | CPG_SIPLL5_CLK4 | CPG_SIPLL5_CLK5 |
|---|---|---|
| | INTIN | SPREAD |
| PLL | Bits 27 to 16 | Bits 4 to 0 |
| PLL5 | H'07D (INTIN = 125) | H'16 (SPREAD = 22) |

- Output clock (initial values)

| PLL | PLL Operating Mode | Frequency ($F_{Fout}$) [MHz] | SSCG | Fmod [kHz] | Spread | Modulation Depth [%] |
|---|---|---|---|---|---|---|
| PLL5 | Reset state | 3,000 | Enabled | 31.25 | Down spread | 2.20 |

FOUTVCO = FREF/REFDIV × (INTIN + FRACIN / 2^24)
FOUTPOSTDIV = FOUTVCO / (POSTDIV1 × POSTDIV2)

The specifiable ranges of parameters for PLL5 are shown below.

| Parameter | Restrictions | Specifiable Range |
|---|---|---|
| FREF | Fixed to the OSC frequency (24) | 24 |
| REFDIV | 10 ≤ FREF/REFDIV ≤ FOUTVCO/20 (Frac) | 1 to 2 |
| INTIN*1 | 20 to 320 (Frac)<br>and<br>800 ≤ FOUTVCO ≤ 3000 (PLL5) | Values that satisfy the restriction shown to the left |
| FRACIN*1 | 800 ≤ FOUTVCO ≤ 3000 (PLL5) | Values that satisfy the restriction shown to the left |
| POSTDIV1 | — | 1 to 7 |
| POSTDIV2 | — | 1 to 7 |

Note 1. The restrictions when the SSCG is disabled are shown. When the SSCG is enabled, the restrictions should be satisfied with the modulation depth taken into account.

### 7.2.4.15 PLLn (SSCG) Monitor Register (CPG_SIPLLn_MON) (n = 2, 3, or 5)

This register is used to monitor the state of the SSCG PLLn (n = 2, 3, or 5).

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Initial Value n = 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Initial Value n = 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Initial Value n = 5 | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | PLLn_LOCK | — | — | — | PLLn_RESETB |
| Initial Value n = 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| Initial Value n = 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| Initial Value n = 5 | 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<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 4 | PLLn_LOCK [n = 2, 3, or 5] | n = 2: 1b<br>n = 3: 1b<br>n = 5: 0b | R | SSCG PLL lock state monitoring<br>0: PLL is not locked.<br>1: PLL is locked. |
| 3 to 1 | — | 000b | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 0 | PLLn_RESETB [n = 2, 3, or 5] | n = 2: 1b<br>n = 3: 1b<br>n = 5: 0b | R | SSCG PLL operating mode monitoring<br>1: Normal mode<br>0: Reset state (standby mode) |

### 7.2.4.16 Division Ratio Setting (PLL1) Register (CPG_PL1_DDIV)

This register is used to set the division ratio of the clock for the Cortex-A55.

The PLL1 (1,000 MHz) output is used as the source clock.

The setting of this register can be dynamically modified while the clock is operating.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | DIV_PLL1SET_WEN |
| 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 | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | DIVPL1_SET | |
| 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 | RW | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 17 | — | All 0 | R | Reserved<br>These bits are always read as 0. Writing to these bits is ignored. |
| 16 | DIV_PLL1SET_WEN | 0b | R0W1 | Flag for enabling the writing to the DIVPL1_SET bits<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 2 | — | All 0 | R | Reserved<br>These bits are always read as 0. Writing to these bits is ignored. |
| 1, 0 | DIVPL1_SET | 00b | RW | Iϕ division ratio setting<br>00b: 1/1 (1,000MHz)<br>01b: 1/2 (500MHz)<br>10b: 1/4 (250MHz)<br>11b: 1/8 (125MHz) |

### 7.2.4.17 Division Ratio Setting (PLL2) Register (CPG_PL2_DDIV)

This register is used to set the division ratios of the clocks for the system bus and the IP modules that cannot use SSCG clocks.

The PLL2 (up to 3,200 MHz) output is used as the source clock.

The setting of this register can be dynamically modified while the clock is operating.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | DIV_DSI_LPCLK_WEN | — | — | — | — | — | — | — | — | — | — | — | DIV_PL L2_A_W EN |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R0W1 | R | R | R | R | R | R | R | R | R | R | R | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | DIVDSILPCLK_SET | | — | — | — | — | — | — | — | — | — | DIVPL2A_SET | | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | RW | RW | R | R | R | R | R | R | R | R | R | RW | RW | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 29 | — | 000b | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 28 | DIV_DSI_LPCLK_WEN | 0b | R0W1 | Flag for enabling the writing to the DIVDSILPCLK_SET bits<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 27 to 17 | — | 000b | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 16 | DIV_PLL2_A_WEN | 0b | R0W1 | Flag for enabling the writing to the DIVPL2A_SET bits<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15, 14 | — | 00b | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 13, 12 | DIVDSILPCLK_SET | 00b | RW | M4ϕ division ratio setting<br>000b: 1/16 (16.656MHz)<br>001b: 1/32 (8.328MHz)<br>010b: 1/64 (4.164MHz)<br>011b: 1/128 (2.082MHz) |
| 11 | — | 0b | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 10 to 8 | — | 000b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 7 | — | 0b | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 6 to 4 | — | 000b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 3 | — | 0b | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 2 to 0 | DIVPL2A_SET | 000b | RW | Pϕ division ratio setting<br>000b: 1/1 (100MHz)<br>001b: 1/2 (50MHz)<br>010b: 1/4 (25MHz)<br>011b: 1/8 (12.5MHz)<br>100b: 1/32 (3.125MHz)<br>Others: Setting prohibited |

### 7.2.4.18 Division Ratio Setting (PLL3A) Register (CPG_PL3A_DDIV)

This register is used to set the division ratios of the clocks for the system bus and the IP modules that can use SSCG clocks.

The clock obtained by dividing the output of PLL3 is used as the source clock.

The setting of this register can be dynamically modified while the clock is operating.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | DIV_PLL3_C_WEN | — | — | — | DIV_PLL3_B_WEN | — | — | — | DIV_PLL3_A_WEN |
| 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 | R0W1 | R | R | R | R0W1 | R | R | R | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | DIVPL3C_SET | | | — | DIVPL3B_SET | | | — | DIVPL3A_SET | | |
| 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 | RW | RW | RW | R | RW | RW | RW | R | RW | RW | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 29 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 28 | — | 0b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 27 to 25 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 24 | DIV_PLL3_C_WEN | 0b | R0W1 | Flag for enabling the writing to the DIVPL3C_SET bits<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 23 to 21 | — | 000b | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 20 | DIV_PLL3_B_WEN | 0b | R0W1 | Flag for enabling the writing to the DIVPL3B_SET bits<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 19 to 17 | — | 000b | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 16 | DIV_PLL3_A_WEN | 0b | R0W1 | Flag for enabling the writing to the DIVPL3A_SET bits<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 14 to 12 | — | 000b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 11 | — | 0b | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 10 to 8 | DIVPL3C_SET | 000b | RW | DIV_PLL3_C division ratio setting<br>000b: 1/1 (400/533MHz)*1<br>001b: 1/2 (200/266.5MHz)*1<br>010b: 1/4 (100/133.25MHz)*1<br>011b: 1/8 (50/66.625MHz)*1<br>100b: 1/32 (12.5/16.656MHz)*1<br>Others: Setting prohibited<br>*Note 1.* (xx/yyMHz)<br>The source clock for xx is 400MHz.<br>The source clock for yy is 533MHz.<br>Please select 400 MHz or 533MHz with CPG_PL3_SSEL Register.<br>In the latter stage, SPI0ϕ is divided into 1/2 and SPI1ϕ is divided into 1/4 |
| 7 | — | 0b | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 6 to 4 | DIVPL3B_SET | 000b | RW | P1ϕ division ratio setting<br>The source clock uses a clock (200MHz) that is 1/8 of the output of PLL3 (1,600MHz).<br>000b: 1/1 (200MHz)<br>001b: 1/2 (100MHz)<br>010b: 1/4 (50MHz)<br>011b: 1/8 (25MHz)<br>100b: 1/32 (6.25MHz)<br>Others: Setting prohibited |
| 3 | — | 0b | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 2 to 0 | DIVPL3A_SET | 000b | RW | P2ϕ division ratio setting<br>The source clock uses a clock (100MHz) that is 1/16 of the output of PLL3 (1,600MHz).<br>000b: 1/1 (100MHz)<br>001b: 1/2 (50MHz)<br>010b: 1/4 (25MHz)<br>011b: 1/8 (12.5MHz)<br>100b: 1/32 (3.125MHz)<br>Others: Setting prohibited |

### 7.2.4.19 Division Ratio Setting (PLL3B) Register (CPG_PL3B_DDIV)

The source clock uses a clock (200MHz) that is 1/8 of the output of PLL3 (1,600MHz).

The setting of this register can be dynamically modified while the clock is operating.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | DIV_PLL3_CLK200FIX_WEN |
| 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 | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | DIVPL3CLK200FIX_SET | | |
| 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 | RW | RW | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 17 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 16 | DIV_PLL3_CLK200FIX_WEN | 0b | R0W1 | Flag for enabling the writing to the DIVPL3CLK200FIX_SET bits<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 3 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 2 to 0 | DIVPL3CLK200FIX_SET | 000b | RW | I2ϕ division ratio setting<br>000b: 1/1 (200MHz)<br>001b: 1/2 (100MHz)<br>010b: 1/4 (50MHz)<br>011b: 1/8 (25MHz)<br>100b: 1/32 (6.25MHz)<br>Others: Setting prohibited<br>*Note*: Before writing to this field, read the value at first.It is possible to write to this field only if the reading value is different from the writing value. |

### 7.2.4.20 Source Clock Setting (SDHI) Register (CPG_PL2SDHI_DSEL)

This register is used to switch the clocks for the SDHI.

The setting of this register can be dynamically modified while the clock is operating.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | SEL_SDHI0_WEN |
| 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 | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | SEL_SDHI0_SET | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | RW | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 21 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 20 | — | 0b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 19 to 17 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 16 | SEL_SDHI0_WEN | 0b | R0W1 | Flag for enabling the writing to the SEL_SDHI0_SET bits<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 6 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 5, 4 | — | 01b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 3, 2 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 1, 0 | SEL_SDHI0_SET | 01b | RW | SEL_SDHI0 clock switching*1<br>00b: Setting prohibited<br>01b: 533 MHz<br>10b: 400 MHz<br>11b: 266 MHz |

**Note:** When 1 is written to the SEL_SDHI0_WEN bit placed in the upper 16 bits of this register, the clock switching control begins even if the settings in the lower bits are not changed. Check the status monitor register and wait until the switching is completed. The clock will temporarily stop at the timing of switching.

Note 1. To change the setting from 01b (533 MHz) to 10b (400 MHz) or vice versa, do not directly switch to the target clock. Switch to 11b (266 MHz) first, and then switch to the target setting (01b (533 MHz) or 10b (400 MHz)). Switching the clock directly between the above frequencies can temporarily generate pulses faster than 533MHz, which can cause malfunctions.

### 7.2.4.21 Source Clock Setting (DDR) Register (CPG_PL4_DSEL)

This register is used to switch the source clock for the DDR.

When switching the source clock setting (DDR) register (CPG_PL4_DSEL), wait at least 800 us after Cortex-A55 starts in the boot sequence.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | SEL_PLL4_WEN |
| 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 | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | SEL_PLL4_SET |
| 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 | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 17 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 16 | SEL_PLL4_WEN | 0b | R0W1 | Flag for enabling the writing to the SEL_PLL4_SET bit (bit 0)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 1 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 0 | SEL_PLL4_SET | 0b | RW | Source clock setting<br>0: OSC_DIV1000 clock (0.024MHz)<br>1: PLL4 clock (1600MHz)<br>*Note*: The clock selected by this register is halved in the latter stage.<br>0.024MHz / 2 = 0.012MHz<br>1600MHz / 2 = 800MHz<br>S0ϕ = 0.012MHz or 800MHz |

### 7.2.4.22 Clock Status Monitor Register (CPG_CLKSTATUS)

This register is used to monitor whether clock switching is completed in the dynamically modifiable frequency dividers and selectors.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | SELPLL4_STS | — | — | SELSDHI0_STS | — | — | — | — | — | — | — | — | — | — | — | — |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | DIVPL3CLK200FIX_STS | DIVPL3C_STS | DIVPL3B_STS | DIVPL3A_STS | — | — | — | DIVPL2A_STS | — | — | — | DIVPL1_STS |
| 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 | SELPLL4_STS | 0b | R | SELPLL4 clock switch completion state<br>0: Switching is completed.<br>1: Switching is not completed (busy). |
| 30 | — | 0b | R | Reserved<br>When read, the initial value is read. The written value will be ignored. |
| 29 | — | 0b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 28 | SELSDHI0_STS | 0b | R | SELSDHI0 clock switch completion state<br>0: Switching is completed.<br>1: Switching is not completed (busy). |
| 27 to 13 | — | All 0 | R | Reserved<br>When read, the initial value is read. The written value will be ignored. |
| 12 | — | 0b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 11 | DIVPL3CLK200FIX_STS | 0b | R | DIVPL3CLK200FIX clock switch completion state<br>0: Switching is completed.<br>1: Switching is not completed (busy). |
| 10 | DIVPL3C_STS | 0b | R | DIVPL3C clock switch completion state<br>0: Switching is completed.<br>1: Switching is not completed (busy). |
| 9 | DIVPL3B_STS | 0b | R | DIVPL3B clock switch completion state<br>0: Switching is completed.<br>1: Switching is not completed (busy). |
| 8 | DIVPL3A_STS | 0b | R | DIVPL3A clock switch completion state<br>0: Switching is completed.<br>1: Switching is not completed (busy). |
| 7 | — | 0b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 6,5 | — | 00b | R | Reserved<br>When read, the initial value is read. The written value will be ignored. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 4 | DIVPL2A_STS | 0b | R | DIVPL2A clock switch completion state<br>0: Switching is completed.<br>1: Switching is not completed (busy). |
| 3 to 1 | — | 000b | R | Reserved<br>When read, the initial value is read. The written value will be ignored. |
| 0 | DIVPL1_STS | 0b | R | DIVPL1 clock switch completion state<br>0: Switching is completed.<br>1: Switching is not completed (busy). |

### 7.2.4.23 Source Clock Setting Register (CPG_PL3_SSEL)

This register is used to switch the source clocks for the system bus and the IPs that can use SSCG clocks.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | SEL_PLL3_3_WEN | — | — | — | — | — | — | — | — |
| 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 | RW | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | SEL_PLL3_3_SET | — | — | — | — | — | — | — | — |
| Initial Value | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
| R/W | R | R | R | R | R | R | R | RW | R | R | R | R | R | R | R | R |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 29 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 28 | — | 0b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 27 to 25 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 24 | SEL_PLL3_3_WEN | 0b | RW | Flag for enabling the writing to the SEL_PLL3_3_SET bit (bit 8)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 23 to 13 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 12 | — | 1b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 11 to 9 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 8 | SEL_PLL3_3_SET | 1b | RW | Source clock setting<br>0: 533MHz<br>1: 400MHz |
| 7 to 5 | — | 000b | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 4 | — | 1b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 3 to 1 | — | 000b | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 0 | — | 1b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |

### 7.2.4.24 Division Ratio Setting (PLL5) Register (CPG_PL5_SDIV)

This register is used to set the division ratio of the clock for the LCDC/MIPI-DSI.

The clock obtained by dividing the PLL5 (Up to 3000 MHz or 1500MHz) output is used as the source clock.

Select Up to 3000 MHz or 1500 MHz with SEL_PLL5_4 in the previous stage.

This register does not support dynamic modification.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | DIV_DSI_B_WEN | — | — | — | — | — | — | — | DIV_DSI_A_WEN |
| 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 | R0W1 | R | R | R | R | R | R | R | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | DIVDSIB_SET | | | | — | — | — | — | — | — | DIVDSIA_SET | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| R/W | R | R | R | R | RW | RW | RW | RW | R | R | R | R | R | R | RW | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 25 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 24 | DIV_DSI_B_WEN | 0b | R0W1 | Flag for enabling the writing to the DIVDSIB_SET bits (bits 11 to 8)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 23 to 17 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 16 | DIV_DSI_A_WEN | 0b | R0W1 | Flag for enabling the writing to the DIVDSIA_SET bits (bits 1 and 0)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 12 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 11 to 8 | DIVDSIB_SET | 0000b | RW | Division ratio setting for the DPI clock<br>The division ratio is obtained by DIV_DSI_A × DIV_DSI_B and these bits specify DIV_DSI_B. The clock output range is 5.08 to 148.5MHz.<br>0000b: 1/1<br>0001b: 1/2<br>:<br>1110b: 1/15<br>1111b: 1/16 |
| 7 to 2 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 1, 0 | DIVDSIA_SET | 01b | RW | Division ratio setting for the DPI clock<br>The division ratio is obtained by DIV_DSI_A × DIV_DSI_B and these bits specify DIV_DSI_A.<br>00b: 1/1<br>01b: 1/2<br>10b: 1/4<br>11b: 1/8 |

**Note:** The DIV_DSI_A and DIV_DSI_B settings should satisfy the following conditions.

| Mode | DIV_DSI_A | | | | DIV_DSI_B |
|---|---|---|---|---|---|
| | Input Clock (SEL_PLL5_4 Route Setting) Selection by CPG_OTHERFUNC1_REG (Bit 0) 0: FOUTPOSTDIV 1: FOUT1PH0 | Maximum Input Frequency | Division Setting | Maximum Output Frequency | Division Setting |
| DSI (PLLCLK_IN/vclk1 = even value) | 1: FOUT1PH0 | 1.5 GHz | 1/1, 1/2, 1/4, or 1/8 | 750 MHz | 1/1 to 1/16 |
| DSI (PLLCLK_IN/vclk1 = odd value) | 0: FOUTPOSTDIV | 750 MHz | 1 | 750 MHz | Odd value |
| DPI | 0: FOUTPOSTDIV | 167 MHz | 1/2, 1/4, or 1/8 | 83.5 MHz | 1/1 |

**Note 1.** For MIPI_DSI_VCLK, refer to the separate clock list.

### 7.2.4.25 Clock Control Register Cortex-A55 (CPG_CLKON_CA55)

This register is used to supply or stop clocks for individual modules.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | CLK5_ONWEN | CLK4_ONWEN | CLK3_ONWEN | CLK2_ONWEN | CLK1_ONWEN | CLK0_ONWEN |
| 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 | R0W1 | R0W1 | R0W1 | R0W1 | R0W1 | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | CLK5_ON | CLK4_ON | CLK3_ON | CLK2_ON | CLK1_ON | CLK0_ON |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | RW | RW | RW | RW | RW | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 22 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 21 | CLK5_ONWEN | 0b | R0W1 | Flag for enabling the writing to the CLK5_ON bit (bit 5)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 20 | CLK4_ONWEN | 0b | R0W1 | Flag for enabling the writing to the CLK4_ON bit (bit 4)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 19 | CLK3_ONWEN | 0b | R0W1 | Flag for enabling the writing to the CLK3_ON bit (bit 3)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 18 | CLK2_ONWEN | 0b | R0W1 | Flag for enabling the writing to the CLK2_ON bit (bit 2)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 17 | CLK1_ONWEN | 0b | R0W1 | Flag for enabling the writing to the CLK1_ON bit (bit 1)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 16 | CLK0_ONWEN | 0b | R0W1 | Flag for enabling the writing to the CLK0_ON bit (bit 0)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 6 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 5 | CLK5_ON | 1b | RW | The CA55_TSCLK clock is supplied or stopped.<br>0: Clock is stopped.<br>1: Clock is supplied. |
| 4 | CLK4_ON | 1b | RW | The CA55_ACLK clock is supplied or stopped.<br>0: Clock is stopped.<br>1: Clock is supplied. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 3 | CLK3_ON | 1b | RW | The CA55_GICCLK clock is supplied or stopped.<br>0: Clock is stopped.<br>1: Clock is supplied. |
| 2 | CLK2_ON | 1b | RW | The CA55_ATCLK clock is supplied or stopped.<br>0: Clock is stopped.<br>1: Clock is supplied. |
| 1 | CLK1_ON | 1b | RW | The CA55_PCLK clock is supplied or stopped.<br>0: Clock is stopped.<br>1: Clock is supplied. |
| 0 | CLK0_ON | 1b | RW | The CA55_SCLK/PERICLK clock is supplied or stopped.<br>0: Clock is stopped.<br>1: Clock is supplied. |

### 7.2.4.26 Clock Control Register ACPU (CPG_CLKON_SRAM_ACPU)

This register is used to supply or stop clocks for individual modules.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CLK0_ONWEN |
| 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 | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CLK0_ON |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 17 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 16 | CLK0_ONWEN | 0b | R0W1 | Flag for enabling the writing to the CLK0_ON bit (bit 0)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 1 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 0 | CLK0_ON | 1b | RW | The SRAM_ACPU_ACLK clock is supplied or stopped.<br>0: Clock is stopped.<br>1: Clock is supplied. |

### 7.2.4.27 Clock Control Register MCPU (CPG_CLKON_SRAM_MCPU)

This register is used to supply or stop clocks for individual modules.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CLK0_ONWEN |
| 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 | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CLK0_ON |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 17 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 16 | CLK0_ONWEN | 0b | R0W1 | Flag for enabling the writing to the CLK0_ON bit (bit 0)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 1 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 0 | CLK0_ON | 1b | RW | The SRAM_MCPU_ACLK clock is supplied or stopped.<br>0: Clock is stopped.<br>1: Clock is supplied. |

### 7.2.4.28 Clock Control Register GIC600 (CPG_CLKON_GIC600)

This register is used to supply or stop clocks for individual modules.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CLK0_ONWEN |
| 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 | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CLK0_ON |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 17 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 16 | CLK0_ONWEN | 0b | R0W1 | Flag for enabling the writing to the CLK0_ON bit (bit 0)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 1 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 0 | CLK0_ON | 1b | RW | The GIC600_GICCLK clock is supplied or stopped.<br>0: Clock is stopped.<br>1: Clock is supplied. |

### 7.2.4.29 Clock Control Register IA55 (CPG_CLKON_IA55)

This register is used to supply or stop clocks for individual modules.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CLK1_ONWEN | CLK0_ONWEN |
| 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 | R0W1 | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CLK1_ON | CLK0_ON |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | RW | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 18 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 17 | CLK1_ONWEN | 0b | R0W1 | Flag for enabling the writing to the CLK1_ON bit (bit 1)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 16 | CLK0_ONWEN | 0b | R0W1 | Flag for enabling the writing to the CLK0_ON bit (bit 0)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 2 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 1 | CLK1_ON | 1b | RW | The IA55_CLK clock is supplied or stopped.<br>0: Clock is stopped.<br>1: Clock is supplied. |
| 0 | CLK0_ON | 1b | RW | The IA55_PCLK clock is supplied or stopped.<br>0: Clock is stopped.<br>1: Clock is supplied. |

### 7.2.4.30 Clock Control Register CST (CPG_CLKON_CST)

This register is used to supply or stop clocks for individual modules.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | CLK10_ONWEN | CLK9_ONWEN | — | CLK7_ONWEN | CLK6_ONWEN | — | CLK4_ONWEN | — | CLK2_ONWEN | CLK1_ONWEN | CLK0_ONWEN |
| 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 | R0W1 | R0W1 | R | R0W1 | R0W1 | R | R0W1 | R | R0W1 | R0W1 | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | CLK10_ON | CLK9_ON | — | CLK7_ON | CLK6_ON | — | CLK4_ON | — | CLK2_ON | CLK1_ON | CLK0_ON |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| R/W | R | R | R | R | R | RW | RW | R | RW | RW | R | RW | R | RW | RW | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 27 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 26 | CLK10_ONWEN | 0b | R0W1 | Flag for enabling the writing to the CLK10_ON bit (bit 10)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 25 | CLK9_ONWEN | 0b | R0W1 | Flag for enabling the writing to the CLK9_ON bit (bit 9)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 24 | — | 0b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value.<br>Operation is not guaranteed if a value other than the initial value is written. |
| 23 | CLK7_ONWEN | 0b | R0W1 | Flag for enabling the writing to the CLK7_ON bit (bit 7)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 22 | CLK6_ONWEN | 0b | R0W1 | Flag for enabling the writing to the CLK6_ON bit (bit 6)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 21 | — | 0b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value.<br>Operation is not guaranteed if a value other than the initial value is written. |
| 20 | CLK4_ONWEN | 0b | R0W1 | Flag for enabling the writing to the CLK4_ON bit (bit 4)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 19 | — | 0b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value.<br>Operation is not guaranteed if a value other than the initial value is written. |
| 18 | CLK2_ONWEN | 0b | R0W1 | Flag for enabling the writing to the CLK2_ON bit (bit 2)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 17 | CLK1_ONWEN | 0b | R0W1 | Flag for enabling the writing to the CLK1_ON bit (bit 1)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 16 | CLK0_ONWEN | 0b | R0W1 | Flag for enabling the writing to the CLK0_ON bit (bit 0)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 11 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 10 | CLK10_ON | 1b | RW | The CST_AXI_ETR_CLK clock is supplied or stopped.<br>0: Clock is stopped.<br>1: Clock is supplied. |
| 9 | CLK9_ON | 1b | RW | The CST_AXI_SB_CLK clock is supplied or stopped.<br>0: Clock is stopped.<br>1: Clock is supplied. |
| 8 | — | 1b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value.<br>Operation is not guaranteed if a value other than the initial value is written. |
| 7 | CLK7_ON | 1b | RW | The CST_ATB_CA55_CLK clock is supplied or stopped.<br>0: Clock is stopped.<br>1: Clock is supplied. |
| 6 | CLK6_ON | 1b | RW | The CST_AHB_ATH_CLK clock is supplied or stopped.<br>0: Clock is stopped.<br>1: Clock is supplied. |
| 5 | — | 1b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value.<br>Operation is not guaranteed if a value other than the initial value is written. |
| 4 | CLK4_ON | 1b | RW | The CST_APB_CA55_CLK clock is supplied or stopped.<br>0: Clock is stopped.<br>1: Clock is supplied. |
| 3 | — | 1b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value.<br>Operation is not guaranteed if a value other than the initial value is written. |
| 2 | CLK2_ON | 1b | RW | The CST_APB_SB_CLK clock is supplied or stopped.<br>0: Clock is stopped.<br>1: Clock is supplied. |
| 1 | CLK1_ON | 1b | RW | The CST_TS_CLK clock is supplied or stopped.<br>0: Clock is stopped.<br>1: Clock is supplied. |
| 0 | CLK0_ON | 1b | RW | The CST_CS_CLK clock is supplied or stopped.<br>0: Clock is stopped.<br>1: Clock is supplied. |

### 7.2.4.31 Clock Control Register SYC (CPG_CLKON_SYC)

This register is used to supply or stop clocks for individual modules.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CLK0_ONWEN |
| 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 | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CLK0_ON |
| 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 | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 17 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 16 | CLK0_ONWEN | 0b | R0W1 | Flag for enabling the writing to the CLK0_ON bit (bit 0)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 1 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 0 | CLK0_ON | 0b | RW | The SYC_CNT_CLK clock is supplied or stopped.<br>0: Clock is stopped.<br>1: Clock is supplied. |

### 7.2.4.32 Clock Control Register DMAC_REG (CPG_CLKON_DMAC_REG)

This register is used to supply or stop clocks for individual modules.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CLK1_ONWEN | CLK0_ONWEN |
| 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 | R0W1 | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CLK1_ON | CLK0_ON |
| 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 | RW | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 18 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 17 | CLK1_ONWEN | 0b | R0W1 | Flag for enabling the writing to the CLK1_ON bit (bit 1)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 16 | CLK0_ONWEN | 0b | R0W1 | Flag for enabling the writing to the CLK0_ON bit (bit 0)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 2 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 1 | CLK1_ON | 0b | RW | The DMAC_PCLK clock is supplied or stopped.<br>0: Clock is stopped.<br>1: Clock is supplied. |
| 0 | CLK0_ON | 0b | RW | The DMAC_ACLK clock is supplied or stopped.<br>0: Clock is stopped.<br>1: Clock is supplied. |

### 7.2.4.33 Clock Control Register GTM (CPG_CLKON_GTM)

This register is used to supply or stop clocks for individual modules.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | CLK2_ONWEN | CLK1_ONWEN | CLK0_ONWEN |
| 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 | R0W1 | R0W1 | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | CLK2_ON | CLK1_ON | CLK0_ON |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | RW | RW | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 19 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 18 | CLK2_ONWEN | 0b | R0W1 | Flag for enabling the writing to the CLK2_ON bit (bit 2)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 17 | CLK1_ONWEN | 0b | R0W1 | Flag for enabling the writing to the CLK1_ON bit (bit 1)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 16 | CLK0_ONWEN | 0b | R0W1 | Flag for enabling the writing to the CLK0_ON bit (bit 0)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 3 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 2 | CLK2_ON | 0b | RW | The OSTM2_PCLK clock is supplied or stopped.<br>0: Clock is stopped.<br>1: Clock is supplied. |
| 1 | CLK1_ON | 0b | RW | The OSTM1_PCLK clock is supplied or stopped.<br>0: Clock is stopped.<br>1: Clock is supplied. |
| 0 | CLK0_ON | 1b | RW | The OSTM0_PCLK clock is supplied or stopped.<br>0: Clock is stopped.<br>1: Clock is supplied. |

### 7.2.4.34 Clock Control Register MTU (CPG_CLKON_MTU)

This register is used to supply or stop clocks for individual modules.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CLK0_ONWEN |
| 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 | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CLK0_ON |
| 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 | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 17 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 16 | CLK0_ONWEN | 0b | R0W1 | Flag for enabling the writing to the CLK0_ON bit (bit 0)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 1 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 0 | CLK0_ON | 0b | RW | The MTU_X_MCK_MTU3 clock is supplied or stopped.<br>0: Clock is stopped.<br>1: Clock is supplied. |

### 7.2.4.35 Clock Control Register POE3 (CPG_CLKON_POE3)

This register is used to supply or stop clocks for individual modules.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CLK0_ONWEN |
| 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 | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CLK0_ON |
| 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 | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 17 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 16 | CLK0_ONWEN | 0b | R0W1 | Flag for enabling the writing to the CLK0_ON bit (bit 0)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 1 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 0 | CLK0_ON | 0b | RW | The POE3_CLKM_POE clock is supplied or stopped.<br>0: Clock is stopped.<br>1: Clock is supplied. |

### 7.2.4.36 Clock Control Register WDT (CPG_CLKON_WDT)

This register is used to supply or stop clocks for individual modules.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CLK1_ONWEN | CLK0_ONWEN |
| 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 | R0W1 | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CLK1_ON | CLK0_ON |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | RW | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 22 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 21 | — | 0b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 20 | — | 0b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 19, 18 | — | 00b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 17 | CLK1_ONWEN | 0b | R0W1 | Flag for enabling the writing to the CLK1_ON bit (bit 1)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 16 | CLK0_ONWEN | 0b | R0W1 | Flag for enabling the writing to the CLK0_ON bit (bit 0)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 8 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 7 | — | 1b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 6 | — | 1b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 5 | — | 0b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 4 | — | 0b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 3, 2 | — | 00b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 1 | CLK1_ON | 1b | RW | The WDT0_CLK clock is supplied or stopped.<br>0: Clock is stopped.<br>1: Clock is supplied. |
| 0 | CLK0_ON | 1b | RW | The WDT0_PCLK clock is supplied or stopped.<br>0: Clock is stopped.<br>1: Clock is supplied. |

### 7.2.4.37 Clock Control Register DDR (CPG_CLKON_DDR)

This register is used to supply or stop clocks for individual modules.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CLK1_ONWEN | CLK0_ONWEN |
| 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 | R0W1 | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CLK1_ON | CLK0_ON |
| 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 | RW | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 18 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 17 | CLK1_ONWEN | 0b | R0W1 | Flag for enabling the writing to the CLK1_ON bit (bit 1)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 16 | CLK0_ONWEN | 0b | R0W1 | Flag for enabling the writing to the CLK0_ON bit (bit 0)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 2 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 1 | CLK1_ON | 0b | RW | The DDR_REG_ACLK clock is supplied or stopped.<br>0: Clock is stopped.<br>1: Clock is supplied. |
| 0 | CLK0_ON | 0b | RW | The DDR_PCLK clock is supplied or stopped.<br>0: Clock is stopped.<br>1: Clock is supplied. |

### 7.2.4.38 Clock Control Register SPI_MULTI (CPG_CLKON_SPI_MULTI)

This register is used to supply or stop clocks for individual modules.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CLK1_ONWEN | CLK0_ONWEN |
| 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 | R0W1 | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CLK1_ON | CLK0_ON |
| 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 | RW | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 18 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 17 | CLK1_ONWEN | 0b | R0W1 | Flag for enabling the writing to the CLK1_ON bit (bit 1)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 16 | CLK0_ONWEN | 0b | R0W1 | Flag for enabling the writing to the CLK0_ON bit (bit 0)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 2 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 1 | CLK1_ON | 0b | RW | The SPI_CLK clock is supplied or stopped.<br>0: Clock is stopped.<br>1: Clock is supplied. |
| 0 | CLK0_ON | 0b | RW | The SPI_CLK2 clock is supplied or stopped.<br>0: Clock is stopped.<br>1: Clock is supplied. |

### 7.2.4.39 Clock Control Register SDHI (CPG_CLKON_SDHI)

This register is used to supply or stop clocks for individual modules.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | CLK3_ONWEN | CLK2_ONWEN | CLK1_ONWEN | CLK0_ONWEN |
| 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 | R0W1 | R0W1 | R0W1 | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | CLK3_ON | CLK2_ON | CLK1_ON | CLK0_ON |
| 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 | RW | RW | RW | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 24 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 23 to 20 | — | All 0 | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 19 | CLK3_ONWEN | 0b | R0W1 | Flag for enabling the writing to the CLK3_ON bit (bit 3)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 18 | CLK2_ONWEN | 0b | R0W1 | Flag for enabling the writing to the CLK2_ON bit (bit 2)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 17 | CLK1_ONWEN | 0b | R0W1 | Flag for enabling the writing to the CLK1_ON bit (bit 1)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 16 | CLK0_ONWEN | 0b | R0W1 | Flag for enabling the writing to the CLK0_ON bit (bit 0)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 8 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 7 to 4 | — | All 0 | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 3 | CLK3_ON | 0b | RW | The SDHI0_ACLK clock is supplied or stopped.<br>0: Clock is stopped.<br>1: Clock is supplied. |
| 2 | CLK2_ON | 0b | RW | The SDHI0_CLK_HS clock is supplied or stopped.<br>0: Clock is stopped.<br>1: Clock is supplied. |
| 1 | CLK1_ON | 0b | RW | The SDHI0_IMCLK2 clock is supplied or stopped.<br>0: Clock is stopped.<br>1: Clock is supplied. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 0 | CLK0_ON | 0b | RW | The SDHI0_IMCLK clock is supplied or stopped.<br>0: Clock is stopped.<br>1: Clock is supplied. |

### 7.2.4.40 Clock Control Register MIPI_DSI (CPG_CLKON_MIPI_DSI)

This register is used to supply or stop clocks for individual modules.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | CLK5_ONWEN | CLK4_ONWEN | CLK3_ONWEN | CLK2_ONWEN | CLK1_ONWEN | CLK0_ONWEN |
| 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 | R0W1 | R0W1 | R0W1 | R0W1 | R0W1 | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | CLK5_ON | CLK4_ON | CLK3_ON | CLK2_ON | CLK1_ON | CLK0_ON |
| 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 | RW | RW | RW | RW | RW | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 22 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 21 | CLK5_ONWEN | 0b | R0W1 | Flag for enabling the writing to the CLK5_ON bit (bit 5)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 20 | CLK4_ONWEN | 0b | R0W1 | Flag for enabling the writing to the CLK4_ON bit (bit 4)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 19 | CLK3_ONWEN | 0b | R0W1 | Flag for enabling the writing to the CLK3_ON bit (bit 3)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 18 | CLK2_ONWEN | 0b | R0W1 | Flag for enabling the writing to the CLK2_ON bit (bit 2)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 17 | CLK1_ONWEN | 0b | R0W1 | Flag for enabling the writing to the CLK1_ON bit (bit 1)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 16 | CLK0_ONWEN | 0b | R0W1 | Flag for enabling the writing to the CLK0_ON bit (bit 0)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 6 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 5 | CLK5_ON | 0b | RW | The MIPI_DSI_LPCLK clock is supplied or stopped.<br>0: Clock is stopped.<br>1: Clock is supplied. |
| 4 | CLK4_ON | 0b | RW | The MIPI_DSI_VCLK clock is supplied or stopped.<br>0: Clock is stopped.<br>1: Clock is supplied. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 3 | CLK3_ON | 0b | RW | The MIPI_DSI_PCLK clock is supplied or stopped.<br>0: Clock is stopped.<br>1: Clock is supplied. |
| 2 | CLK2_ON | 0b | RW | The MIPI_DSI_ACLK clock is supplied or stopped.<br>0: Clock is stopped.<br>1: Clock is supplied. |
| 1 | CLK1_ON | 0b | RW | The MIPI_DSI_SYSCLK clock is supplied or stopped.<br>0: Clock is stopped.<br>1: Clock is supplied. |
| 0 | CLK0_ON | 0b | RW | The MIPI_DSI_PLLCLK clock is supplied or stopped.<br>0: Clock is stopped.<br>1: Clock is supplied. |

### 7.2.4.41 Clock Control Register LCDC (CPG_CLKON_LCDC)

This register is used to supply or stop clocks for individual modules.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CLK1_ONWEN | CLK0_ONWEN |
| 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 | R0W1 | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CLK1_ON | CLK0_ON |
| 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 | RW | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 18 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 17 | CLK1_ONWEN | 0b | R0W1 | Flag for enabling the writing to the CLK1_ON bit (bit 1)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 16 | CLK0_ONWEN | 0b | R0W1 | Flag for enabling the writing to the CLK0_ON bit (bit 0)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 2 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 1 | CLK1_ON | 0b | RW | The LCDC_CLK_D clock is supplied or stopped.<br>0: Clock is stopped.<br>1: Clock is supplied. |
| 0 | CLK0_ON | 0b | RW | The LCDC_CLK_A and LCDC_CLK_P clocks are supplied or stopped.<br>0: Clocks are stopped.<br>1: Clocks are supplied. |

### 7.2.4.42 Clock Control Register SSI (CPG_CLKON_SSI)

This register is used to supply or stop clocks for individual modules.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CLK1_ONWEN | CLK0_ONWEN |
| 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 | R0W1 | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CLK1_ON | CLK0_ON |
| 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 | RW | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 24 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 23 to 18 | — | All 0 | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 17 | CLK1_ONWEN | 0b | R0W1 | Flag for enabling the writing to the CLK1_ON bit (bit 1)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 16 | CLK0_ONWEN | 0b | R0W1 | Flag for enabling the writing to the CLK0_ON bit (bit 0)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 8 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 7 to 2 | — | All 0 | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 1 | CLK1_ON | 0b | RW | The SSI0_PCLK_SFR clock is supplied or stopped.<br>0: Clock is stopped.<br>1: Clock is supplied. |
| 0 | CLK0_ON | 0b | RW | The SSI0_PCLK2 clock is supplied or stopped.<br>0: Clock is stopped.<br>1: Clock is supplied. |

### 7.2.4.43 Clock Control Register USB (CPG_CLKON_USB)

This register is used to supply or stop clocks for individual modules.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | CLK3_ONWEN | CLK2_ONWEN | — | CLK0_ONWEN |
| 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 | R0W1 | R0W1 | R | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | CLK3_ON | CLK2_ON | — | CLK0_ON |
| 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 | RW | RW | R | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 20 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 19 | CLK3_ONWEN | 0b | R0W1 | Flag for enabling the writing to the CLK3_ON bit (bit 3)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 18 | CLK2_ONWEN | 0b | R0W1 | Flag for enabling the writing to the CLK2_ON bit (bit 2)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 17 | — | 0b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 16 | CLK0_ONWEN | 0b | R0W1 | Flag for enabling the writing to the CLK0_ON bit (bit 0)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 4 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 3 | CLK3_ON | 0b | RW | The USB_PCLK clock is supplied or stopped.<br>0: Clock is stopped.<br>1: Clock is supplied. |
| 2 | CLK2_ON | 0b | RW | The USB_U2P_EXR_CPUCLK clock is supplied or stopped.<br>0: Clock is stopped.<br>1: Clock is supplied. |
| 1 | — | 0b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 0 | CLK0_ON | 0b | RW | The USB_U2H0_HCLK clock is supplied or stopped.<br>0: Clock is stopped.<br>1: Clock is supplied. |

### 7.2.4.44 Clock Control Register I2C (CPG_CLKON_I2C)

This register is used to supply or stop clocks for individual modules.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CLK1_ONWEN | CLK0_ONWEN |
| 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 | R0W1 | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CLK1_ON | CLK0_ON |
| 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 | RW | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 20 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 19,18 | — | 00b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 17 | CLK1_ONWEN | 0b | R0W1 | Flag for enabling the writing to the CLK1_ON bit (bit 1)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 16 | CLK0_ONWEN | 0b | R0W1 | Flag for enabling the writing to the CLK0_ON bit (bit 0)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 4 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 3,2 | — | 00b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 1 | CLK1_ON | 0b | RW | The I2C1_PCLK clock is supplied or stopped.<br>0: Clock is stopped.<br>1: Clock is supplied. |
| 0 | CLK0_ON | 0b | RW | The I2C0_PCLK clock is supplied or stopped.<br>0: Clock is stopped.<br>1: Clock is supplied. |

### 7.2.4.45 Clock Control Register SCIF (CPG_CLKON_SCIF)

This register is used to supply or stop clocks for individual modules.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | CLK4_ONWEN | CLK3_ONWEN | CLK2_ONWEN | CLK1_ONWEN | CLK0_ONWEN |
| 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 | R0W1 | R0W1 | R0W1 | R0W1 | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | CLK4_ON | CLK3_ON | CLK2_ON | CLK1_ON | CLK0_ON |
| 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 | RW | RW | RW | RW | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 21 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 20 | CLK4_ONWEN | 0b | R0W1 | Flag for enabling the writing to the CLK4_ON bit (bit 4)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 19 | CLK3_ONWEN | 0b | R0W1 | Flag for enabling the writing to the CLK3_ON bit (bit 3)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 18 | CLK2_ONWEN | 0b | R0W1 | Flag for enabling the writing to the CLK2_ON bit (bit 2)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 17 | CLK1_ONWEN | 0b | R0W1 | Flag for enabling the writing to the CLK1_ON bit (bit 1)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 16 | CLK0_ONWEN | 0b | R0W1 | Flag for enabling the writing to the CLK0_ON bit (bit 0)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 5 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 4 | CLK4_ON | 0b | RW | The SCIF4_CLK_PCK clock is supplied or stopped.<br>0: Clock is stopped.<br>1: Clock is supplied. |
| 3 | CLK3_ON | 0b | RW | The SCIF3_CLK_PCK clock is supplied or stopped.<br>0: Clock is stopped.<br>1: Clock is supplied. |
| 2 | CLK2_ON | 0b | RW | The SCIF2_CLK_PCK clock is supplied or stopped.<br>0: Clock is stopped.<br>1: Clock is supplied. |
| 1 | CLK1_ON | 0b | RW | The SCIF1_CLK_PCK clock is supplied or stopped.<br>0: Clock is stopped.<br>1: Clock is supplied. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 0 | CLK0_ON | 0b | RW | The SCIF0_CLK_PCK clock is supplied or stopped.<br>0: Clock is stopped.<br>1: Clock is supplied. |

### 7.2.4.46 Clock Control Register SCI (CPG_CLKON_SCI)

This register is used to supply or stop clocks for individual modules.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CLK1_ONWEN | CLK0_ONWEN |
| 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 | R0W1 | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CLK1_ON | CLK0_ON |
| 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 | RW | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 18 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 17 | CLK1_ONWEN | 0b | R0W1 | Flag for enabling the writing to the CLK1_ON bit (bit 1)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 16 | CLK0_ONWEN | 0b | R0W1 | Flag for enabling the writing to the CLK0_ON bit (bit 0)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 2 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 1 | CLK1_ON | 0b | RW | The SCI1_CLKP clock is supplied or stopped.<br>0: Clock is stopped.<br>1: Clock is supplied. |
| 0 | CLK0_ON | 0b | RW | The SCI0_CLKP clock is supplied or stopped.<br>0: Clock is stopped.<br>1: Clock is supplied. |

### 7.2.4.47 Clock Control Register IRDA (CPG_CLKON_IRDA)

This register is used to supply or stop clocks for individual modules.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CLK0_ONWEN |
| 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 | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CLK0_ON |
| 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 | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 17 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 16 | CLK0_ONWEN | 0b | R0W1 | Flag for enabling the writing to the CLK0_ON bit (bit 0)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 1 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 0 | CLK0_ON | 0b | RW | The IRDA_CLKP clock is supplied or stopped.<br>0: Clock is stopped.<br>1: Clock is supplied. |

### 7.2.4.48 Clock Control Register RSPI (CPG_CLKON_RSPI)

This register is used to supply or stop clocks for individual modules.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CLK1_ONWEN | CLK0_ONWEN |
| 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 | R0W1 | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CLK1_ON | CLK0_ON |
| 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 | RW | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 19 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 18 | — | 0b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 17 | CLK1_ONWEN | 0b | R0W1 | Flag for enabling the writing to the CLK1_ON bit (bit 1)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 16 | CLK0_ONWEN | 0b | R0W1 | Flag for enabling the writing to the CLK0_ON bit (bit 0)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 3 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 2 | — | 0b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 1 | CLK1_ON | 0b | RW | The RSPI1_CLKB clock is supplied or stopped.<br>0: Clock is stopped.<br>1: Clock is supplied. |
| 0 | CLK0_ON | 0b | RW | The RSPI0_CLKB clock is supplied or stopped.<br>0: Clock is stopped.<br>1: Clock is supplied. |

### 7.2.4.49 Clock Control Register GPIO (CPG_CLKON_GPIO)

This register is used to supply or stop clocks for individual modules.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CLK0_ONWEN |
| 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 | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CLK0_ON |
| 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 | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 17 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 16 | CLK0_ONWEN | 0b | R0W1 | Flag for enabling the writing to the CLK0_ON bit (bit 0)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 1 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 0 | CLK0_ON | 0b | RW | The GPIO_HCLK clock is supplied or stopped.<br>0: Clock is stopped.<br>1: Clock is supplied. |

### 7.2.4.50 Clock Control Register TSU (CPG_CLKON_TSU)

This register is used to supply or stop clocks for individual modules.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CLK0_ONWEN |
| 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 | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CLK0_ON |
| 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 | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 17 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 16 | CLK0_ONWEN | 0b | R0W1 | Flag for enabling the writing to the CLK0_ON bit (bit 0)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 1 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 0 | CLK0_ON | 0b | RW | The TSU_PCLK clock is supplied or stopped.<br>0: Clock is stopped.<br>1: Clock is supplied. |

### 7.2.4.51 Clock Control Register DRW (CPG_CLKON_DRW)

This register is used to supply or stop clocks for individual modules.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CLK0_ONWEN |
| 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 | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CLK0_ON |
| 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 | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 17 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 16 | CLK0_ONWEN | 0b | R0W1 | Flag for enabling the writing to the CLK0_ON bit (bit 0)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 1 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 0 | CLK0_ON | 0b | RW | The DRW_CLK clock is supplied or stopped.<br>0: Clock is stopped.<br>1: Clock is supplied. |

### 7.2.4.52 Clock Monitor Register Cortex-A55 (CPG_CLKMON_CA55)

This register is used to monitor the state of the clocks supplied to individual modules.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | CLK5_MON | CLK4_MON | CLK3_MON | CLK2_MON | CLK1_MON | CLK0_MON |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 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 6 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 5 | CLK5_MON | 1b | R | The state of the CA55_TSCLK clock is monitored.<br>0: Clock is not supplied.<br>1: Clock is supplied. |
| 4 | CLK4_MON | 1b | R | The state of the CA55_ACLK clock is monitored.<br>0: Clock is not supplied.<br>1: Clock is supplied. |
| 3 | CLK3_MON | 1b | R | The state of the CA55_GICCLK clock is monitored.<br>0: Clock is not supplied.<br>1: Clock is supplied. |
| 2 | CLK2_MON | 1b | R | The state of the CA55_ATCLK clock is monitored.<br>0: Clock is not supplied.<br>1: Clock is supplied. |
| 1 | CLK1_MON | 1b | R | The state of the CA55_PCLK clock is monitored.<br>0: Clock is not supplied.<br>1: Clock is supplied. |
| 0 | CLK0_MON | 1b | R | The state of the CA55_SCLK/PERICLK clock is monitored.<br>0: Clock is not supplied.<br>1: Clock is supplied. |

### 7.2.4.53 Clock Monitor Register SRAM_ACPU (CPG_CLKMON_SRAM_ACPU)

This register is used to monitor the state of the clocks supplied to individual modules.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CLK0_MON |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 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 1 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 0 | CLK0_MON | 1b | R | The state of the SRAM_ACPU_ACLK clock is monitored.<br>0: Clock is not supplied.<br>1: Clock is supplied. |

### 7.2.4.54 Clock Monitor Register SRAM_MCPU (CPG_CLKMON_SRAM_MCPU)

This register is used to monitor the state of the clocks supplied to individual modules.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CLK0_MON |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 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 1 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 0 | CLK0_MON | 1b | R | The state of the SRAM_MCPU_ACLK clock is monitored.<br>0: Clock is not supplied.<br>1: Clock is supplied. |

### 7.2.4.55 Clock Monitor Register GIC600 (CPG_CLKMON_GIC600)

This register is used to monitor the state of the clocks supplied to individual modules.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CLK0_MON |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 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 1 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 0 | CLK0_MON | 1b | R | The state of the GIC600_GICCLK clock is monitored.<br>0: Clock is not supplied.<br>1: Clock is supplied. |

### 7.2.4.56 Clock Monitor Register IA55 (CPG_CLKMON_IA55)

This register is used to monitor the state of the clocks supplied to individual modules.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CLK1_MON | CLK0_MON |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 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 2 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 1 | CLK1_MON | 1b | R | The state of the IA55_CLK clock is monitored.<br>0: Clock is not supplied.<br>1: Clock is supplied. |
| 0 | CLK0_MON | 1b | R | The state of the IA55_PCLK clock is monitored.<br>0: Clock is not supplied.<br>1: Clock is supplied. |

### 7.2.4.57 Clock Monitor Register CST (CPG_CLKMON_CST)

This register is used to monitor the state of the clocks supplied to individual modules.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | CLK10_MON | CLK9_MON | — | CLK7_MON | CLK6_MON | — | CLK4_MON | — | CLK2_MON | CLK1_MON | CLK0_MON |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 1 | 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 11 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 10 | CLK10_MON | 1b | R | The state of the CST_AXI_ETR_CLK clock is monitored.<br>0: Clock is not supplied.<br>1: Clock is supplied. |
| 9 | CLK9_MON | 1b | R | The state of the CST_AXI_SB_CLK clock is monitored.<br>0: Clock is not supplied.<br>1: Clock is supplied. |
| 8 | — | 1b | R | Reserved<br>When read, the initial value is read. The written value will be ignored. |
| 7 | CLK7_MON | 1b | R | The state of the CST_ATB_CA55_CLK clock is monitored.<br>0: Clock is not supplied.<br>1: Clock is supplied. |
| 6 | CLK6_MON | 1b | R | The state of the CST_AHB_ATH_CLK clock is monitored.<br>0: Clock is not supplied.<br>1: Clock is supplied. |
| 5 | — | 1b | R | Reserved<br>When read, the initial value is read. The written value will be ignored. |
| 4 | CLK4_MON | 1b | R | The state of the CST_APB_CA55_CLK clock is monitored.<br>0: Clock is not supplied.<br>1: Clock is supplied. |
| 3 | — | 1b | R | Reserved<br>When read, the initial value is read. The written value will be ignored. |
| 2 | CLK2_MON | 1b | R | The state of the CST_APB_SB_CLK clock is monitored.<br>0: Clock is not supplied.<br>1: Clock is supplied. |
| 1 | CLK1_MON | 1b | R | The state of the CST_TS_CLK clock is monitored.<br>0: Clock is not supplied.<br>1: Clock is supplied. |
| 0 | CLK0_MON | 1b | R | The state of the CST_CS_CLK clock is monitored.<br>0: Clock is not supplied.<br>1: Clock is supplied. |

**Note:** No CGC is provided for CST_SWCLKTCK.

### 7.2.4.58 Clock Monitor Register SYC (CPG_CLKMON_SYC)

This register is used to monitor the state of the clocks supplied to individual modules.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CLK0_MON |
| 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 1 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 0 | CLK0_MON | 0b | R | The state of the SYC_CNT_CLK clock is monitored.<br>0: Clock is not supplied.<br>1: Clock is supplied. |

### 7.2.4.59 Clock Monitor Register DMAC_REG (CPG_CLKMON_DMAC_REG)

This register is used to monitor the state of the clocks supplied to individual modules.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CLK1_MON | CLK0_MON |
| 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 2 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 1 | CLK1_MON | 0b | R | The state of the DMAC_PCLK clock is monitored.<br>0: Clock is not supplied.<br>1: Clock is supplied. |
| 0 | CLK0_MON | 0b | R | The state of the DMAC_ACLK clock is monitored.<br>0: Clock is not supplied.<br>1: Clock is supplied. |

### 7.2.4.60 Clock Monitor Register GTM (CPG_CLKMON_GTM)

This register is used to monitor the state of the clocks supplied to individual modules.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | CLK2_MON | CLK1_MON | CLK0_MON |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 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 3 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 2 | CLK2_MON | 0b | R | The state of the OSTM2_PCLK clock is monitored.<br>0: Clock is not supplied.<br>1: Clock is supplied. |
| 1 | CLK1_MON | 0b | R | The state of the OSTM1_PCLK clock is monitored.<br>0: Clock is not supplied.<br>1: Clock is supplied. |
| 0 | CLK0_MON | 1b | R | The state of the OSTM0_PCLK clock is monitored.<br>0: Clock is not supplied.<br>1: Clock is supplied. |

### 7.2.4.61 Clock Monitor Register MTU (CPG_CLKMON_MTU)

This register is used to monitor the state of the clocks supplied to individual modules.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CLK0_MON |
| 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 1 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 0 | CLK0_MON | 0b | R | The state of the MTU_X_MCK_MTU3 clock is monitored.<br>0: Clock is not supplied.<br>1: Clock is supplied. |

### 7.2.4.62 Clock Monitor Register POE3 (CPG_CLKMON_POE3)

This register is used to monitor the state of the clocks supplied to individual modules.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CLK0_MON |
| 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 1 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 0 | CLK0_MON | 0b | R | The state of the POE3_CLKM_POE clock is monitored.<br>0: Clock is not supplied.<br>1: Clock is supplied. |

### 7.2.4.63 Clock Monitor Register WDT (CPG_CLKMON_WDT)

This register is used to monitor the state of the clocks supplied to individual modules.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CLK1_MON | CLK0_MON |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 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 8 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 7 | — | 1b | R | Reserved<br>When read, the initial value is read. The written value will be ignored. |
| 6 | — | 1b | R | Reserved<br>When read, the initial value is read. The written value will be ignored. |
| 5 | — | 0b | R | Reserved<br>When read, the initial value is read. The written value will be ignored. |
| 4 | — | 0b | R | Reserved<br>When read, the initial value is read. The written value will be ignored. |
| 3, 2 | — | 00b | R | Reserved<br>When read, the initial value is read. The written value will be ignored. |
| 1 | CLK1_MON | 1b | R | The state of the WDT0_CLK clock is monitored.<br>0: Clock is not supplied.<br>1: Clock is supplied. |
| 0 | CLK0_MON | 1b | R | The state of the WDT0_PCLK clock is monitored.<br>0: Clock is not supplied.<br>1: Clock is supplied. |

### 7.2.4.64 Clock Monitor Register DDR (CPG_CLKMON_DDR)

This register is used to monitor the state of the clocks supplied to individual modules.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CLK1_MON | CLK0_MON |
| 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 2 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 1 | CLK1_MON | 0b | R | The state of the DDR_REG_ACLK clock is monitored.<br>0: Clock is not supplied.<br>1: Clock is supplied. |
| 0 | CLK0_MON | 0b | R | The state of the DDR_PCLK clock is monitored.<br>0: Clock is not supplied.<br>1: Clock is supplied. |

### 7.2.4.65 Clock Monitor Register SPI_MULTI (CPG_CLKMON_SPI_MULTI)

This register is used to monitor the state of the clocks supplied to individual modules.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CLK1_MON | CLK0_MON |
| 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 2 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 1 | CLK1_MON | 0b | R | The state of the SPI_CLK clock is monitored.<br>0: Clock is not supplied.<br>1: Clock is supplied. |
| 0 | CLK0_MON | 0b | R | The state of the SPI_CLK2 clock is monitored.<br>0: Clock is not supplied.<br>1: Clock is supplied. |

### 7.2.4.66 Clock Monitor Register SDHI (CPG_CLKMON_SDHI)

This register is used to monitor the state of the clocks supplied to individual modules.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | CLK3_MON | CLK2_MON | CLK1_MON | CLK0_MON |
| 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 4 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 3 | CLK3_MON | 0b | R | The state of the SDHI0_ACLK clock is monitored.<br>0: Clock is not supplied.<br>1: Clock is supplied. |
| 2 | CLK2_MON | 0b | R | The state of the SDHI0_CLK_HS clock is monitored.<br>0: Clock is not supplied.<br>1: Clock is supplied. |
| 1 | CLK1_MON | 0b | R | The state of the SDHI0_IMCLK2 clock is monitored.<br>0: Clock is not supplied.<br>1: Clock is supplied. |
| 0 | CLK0_MON | 0b | R | The state of the SDHI0_IMCLK clock is monitored.<br>0: Clock is not supplied.<br>1: Clock is supplied. |

### 7.2.4.67 Clock Monitor Register MIPI_DSI (CPG_CLKMON_MIPI_DSI)

This register is used to monitor the state of the clocks supplied to individual modules.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | CLK5_MON | CLK4_MON | CLK3_MON | CLK2_MON | CLK1_MON | CLK0_MON |
| 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 6 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 5 | CLK5_MON | 0b | R | The state of the MIPI_DSI_LP.CLK clock is monitored.<br>0: Clock is not supplied.<br>1: Clock is supplied. |
| 4 | CLK4_MON | 0b | R | The state of the MIPI_DSI_VCLK clock is monitored.<br>0: Clock is not supplied.<br>1: Clock is supplied. |
| 3 | CLK3_MON | 0b | R | The state of the MIPI_DSI_PCLK clock is monitored.<br>0: Clock is not supplied.<br>1: Clock is supplied. |
| 2 | CLK2_MON | 0b | R | The state of the MIPI_DSI_ACLK clock is monitored.<br>0: Clock is not supplied.<br>1: Clock is supplied. |
| 1 | CLK1_MON | 0b | R | The state of the MIPI_DSI_SYSCLK is monitored.<br>0: Clock is not supplied.<br>1: Clock is supplied. |
| 0 | CLK0_MON | 0b | R | The state of the MIPI_DSI_PLLCLK clock is monitored.<br>0: Clock is not supplied.<br>1: Clock is supplied. |

### 7.2.4.68 Clock Monitor Register LCDC (CPG_CLKMON_LCDC)

This register is used to monitor the state of the clocks supplied to individual modules.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CLK1_MON | CLK0_MON |
| 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 2 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 1 | CLK1_MON | 0b | R | The state of the LCDC_CLK_D clock is monitored.<br>0: Clock is not supplied.<br>1: Clock is supplied. |
| 0 | CLK0_MON | 0b | R | The state of the LCDC_CLK_A/P clock is monitored.<br>0: Clock is not supplied.<br>1: Clock is supplied. |

### 7.2.4.69 Clock Monitor Register SSI (CPG_CLKMON_SSI)

This register is used to monitor the state of the clocks supplied to individual modules.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CLK1_MON | CLK0_MON |
| 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 2 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 1 | CLK1_MON | 0b | R | The state of the SSI0_PCLK_SFR clock is monitored.<br>0: Clock is not supplied.<br>1: Clock is supplied. |
| 0 | CLK0_MON | 0b | R | The state of the SSI0_PCLK2 clock is monitored.<br>0: Clock is not supplied.<br>1: Clock is supplied. |

### 7.2.4.70 Clock Monitor Register USB (CPG_CLKMON_USB)

This register is used to monitor the state of the clocks supplied to individual modules.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | CLK3_MON | CLK2_MON | — | CLK0_MON |
| 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 4 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 3 | CLK3_MON | 0b | R | The state of the USB_PCLK clock is monitored.<br>0: Clock is not supplied.<br>1: Clock is supplied. |
| 2 | CLK2_MON | 0b | R | The state of the USB_U2P_EXR_CPUCLK clock is monitored.<br>0: Clock is not supplied.<br>1: Clock is supplied. |
| 1 | — | 0b | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 0 | CLK0_MON | 0b | R | The state of the USB_U2H0_HCLK clock is monitored.<br>0: Clock is not supplied.<br>1: Clock is supplied. |

### 7.2.4.71 Clock Monitor Register I2C (CPG_CLKMON_I2C)

This register is used to monitor the state of the clocks supplied to individual modules.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CLK1_MON | CLK0_MON |
| 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 2 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 1 | CLK1_MON | 0b | R | The state of the I2C1_PCLK clock is monitored.<br>0: Clock is not supplied.<br>1: Clock is supplied. |
| 0 | CLK0_MON | 0b | R | The state of the I2C0_PCLK clock is monitored.<br>0: Clock is not supplied.<br>1: Clock is supplied. |

### 7.2.4.72 Clock Monitor Register SCIF (CPG_CLKMON_SCIF)

This register is used to monitor the state of the clocks supplied to individual modules.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | CLK4_MON | CLK3_MON | CLK2_MON | CLK1_MON | CLK0_MON |
| 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<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 4 | CLK4_MON | 0b | R | The state of the SCIF4_CLK_PCK clock is monitored.<br>0: Clock is not supplied.<br>1: Clock is supplied. |
| 3 | CLK3_MON | 0b | R | The state of the SCIF3_CLK_PCK clock is monitored.<br>0: Clock is not supplied.<br>1: Clock is supplied. |
| 2 | CLK2_MON | 0b | R | The state of the SCIF2_CLK_PCK clock is monitored.<br>0: Clock is not supplied.<br>1: Clock is supplied. |
| 1 | CLK1_MON | 0b | R | The state of the SCIF1_CLK_PCK clock is monitored.<br>0: Clock is not supplied.<br>1: Clock is supplied. |
| 0 | CLK0_MON | 0b | R | The state of the SCIF0_CLK_PCK clock is monitored.<br>0: Clock is not supplied.<br>1: Clock is supplied. |

### 7.2.4.73 Clock Monitor Register SCI (CPG_CLKMON_SCI)

This register is used to monitor the state of the clocks supplied to individual modules.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CLK1_MON | CLK0_MON |
| 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 2 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored.. |
| 1 | CLK1_MON | 0b | R | The state of the SCI1_CLKP clock is monitored.<br>0: Clock is not supplied.<br>1: Clock is supplied. |
| 0 | CLK0_MON | 0b | R | The state of the SCI0_CLKP clock is monitored.<br>0: Clock is not supplied.<br>1: Clock is supplied. |

### 7.2.4.74 Clock Monitor Register IRDA (CPG_CLKMON_IRDA)

This register is used to monitor the state of the clocks supplied to individual modules.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CLK0_MON |
| 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 1 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 0 | CLK0_MON | 0b | R | The state of the IRDA_CLKP clock is monitored.<br>0: Clock is not supplied.<br>1: Clock is supplied. |

### 7.2.4.75 Clock Monitor Register RSPI (CPG_CLKMON_RSPI)

This register is used to monitor the state of the clocks supplied to individual modules.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CLK1_MON | CLK0_MON |
| 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 2 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 1 | CLK1_MON | 0b | R | The state of the RSPI1_CLKB clock is monitored.<br>0: Clock is not supplied.<br>1: Clock is supplied. |
| 0 | CLK0_MON | 0b | R | The state of the RSPI0_CLKB clock is monitored.<br>0: Clock is not supplied.<br>1: Clock is supplied. |

### 7.2.4.76 Clock Monitor Register GPIO (CPG_CLKMON_GPIO)

This register is used to monitor the state of the clocks supplied to individual modules.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CLK0_MON |
| 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 1 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 0 | CLK0_MON | 0b | R | The state of the GPIO_HCLK clock is monitored.<br>0: Clock is not supplied.<br>1: Clock is supplied. |

### 7.2.4.77 Clock Monitor Register TSU (CPG_CLKMON_TSU)

This register is used to monitor the state of the clocks supplied to individual modules.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CLK0_MON |
| 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 1 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 0 | CLK0_MON | 0b | R | The state of the TSU_PCLK clock is monitored.<br>0: Clock is not supplied.<br>1: Clock is supplied. |

### 7.2.4.78 Clock Monitor Register DRW (CPG_CLKMON_DRW)

This register is used to monitor the state of the clocks supplied to individual modules.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CLK0_MON |
| 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 1 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 0 | CLK0_MON | 0b | R | The state of the DRW_CLK clock is monitored.<br>0: Clock is not supplied.<br>1: Clock is supplied. |

### 7.2.4.79 Reset Control Register Cortex-A55 (CPG_RST_CA55)

This register is used to control the reset signals for individual modules.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | UNIT12_RST WEN | UNIT11_RST WEN | UNIT10_RST WEN | UNIT9_RST WEN | UNIT8_RST WEN | UNIT7_RST WEN | UNIT6_RST WEN | UNIT5_RST WEN | UNIT4_RST WEN | — | UNIT2_RST WEN | — | UNIT0_RST WEN |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R0W1 | R0W1 | R0W1 | R0W1 | R0W1 | R0W1 | R0W1 | R0W1 | R0W1 | R | R0W1 | R | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | UNIT12_RSTB | UNIT11_RSTB | UNIT10_RSTB | UNIT9_RSTB | UNIT8_RSTB | UNIT7_RSTB | UNIT6_RSTB | UNIT5_RSTB | UNIT4_RSTB | — | UNIT2_RSTB | — | UNIT0_RSTB |
| Initial Value | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| R/W | R | R | R | RW | RW | RW | RW | RW | RW | RW | RW | RW | R | RW | R | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 29 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 28 | UNIT12_RST WEN | 0b | R0W1 | Flag for enabling the writing to the UNIT12_RSTB bit (bit 12)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 27 | UNIT11_RST WEN | 0b | R0W1 | Flag for enabling the writing to the UNIT11_RSTB bit (bit 11)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 26 | UNIT10_RST WEN | 0b | R0W1 | Flag for enabling the writing to the UNIT10_RSTB bit (bit 10)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 25 | UNIT9_RST WEN | 0b | R0W1 | Flag for enabling the writing to the UNIT9_RSTB bit (bit 9)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 24 | UNIT8_RST WEN | 0b | R0W1 | Flag for enabling the writing to the UNIT8_RSTB bit (bit 8)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 23 | UNIT7_RST WEN | 0b | R0W1 | Flag for enabling the writing to the UNIT7_RSTB bit (bit 7)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 22 | UNIT6_RST WEN | 0b | R0W1 | Flag for enabling the writing to the UNIT6_RSTB bit (bit 6)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 21 | UNIT5_RST WEN | 0b | R0W1 | Flag for enabling the writing to the UNIT5_RSTB bit (bit 5)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 20 | UNIT4_RSTWEN | 0b | R0W1 | Flag for enabling the writing to the UNIT4_RSTB bit (bit 4)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 19 | — | 0b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 18 | UNIT2_RSTWEN | 0b | R0W1 | Flag for enabling the writing to the UNIT2_RSTB bit (bit 2)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 17 | — | 0b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 16 | UNIT0_RSTWEN | 0b | R0W1 | Flag for enabling the writing to the UNIT0_RSTB bit (bit 0)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 13 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 12 | UNIT12_RSTB | 1b | RW | The CA55_RST12 reset terminal is controlled.<br>0: Reset signal is applied (reset state).<br>1: Reset signal is stopped (released from the reset state). |
| 11 | UNIT11_RSTB | 1b | RW | The CA55_RST11 reset terminal is controlled.<br>0: Reset signal is applied (reset state).<br>1: Reset signal is stopped (released from the reset state). |
| 10 | UNIT10_RSTB | 1b | RW | The CA55_RST10 reset terminal is controlled.<br>0: Reset signal is applied (reset state).<br>1: Reset signal is stopped (released from the reset state). |
| 9 | UNIT9_RSTB | 1b | RW | The CA55_RST9 reset terminal is controlled.<br>0: Reset signal is applied (reset state).<br>1: Reset signal is stopped (released from the reset state). |
| 8 | UNIT8_RSTB | 1b | RW | The CA55_RST8 reset terminal is controlled.<br>0: Reset signal is applied (reset state).<br>1: Reset signal is stopped (released from the reset state). |
| 7 | UNIT7_RSTB | 1b | RW | The CA55_RST7 reset terminal is controlled.<br>0: Reset signal is applied (reset state).<br>1: Reset signal is stopped (released from the reset state). |
| 6 | UNIT6_RSTB | 1b | RW | The CA55_RST6 reset terminal is controlled.<br>0: Reset signal is applied (reset state).<br>1: Reset signal is stopped (released from the reset state). |
| 5 | UNIT5_RSTB | 1b | RW | The CA55_RST5 reset terminal is controlled.<br>0: Reset signal is applied (reset state).<br>1: Reset signal is stopped (released from the reset state). |
| 4 | UNIT4_RSTB | 1b | RW | The CA55_RST4 reset terminal is controlled.<br>0: Reset signal is applied (reset state).<br>1: Reset signal is stopped (released from the reset state). |
| 3 | — | 1b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 2 | UNIT2_RSTB | 1b | RW | The CA55_RST3_0 reset terminal is controlled.<br>0: Reset signal is applied (reset state).<br>1: Reset signal is stopped (released from the reset state). |
| 1 | — | 1b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 0 | UNIT0_RSTB | 1b | RW | The CA55_RST1_0 reset terminal is controlled.<br>0: Reset signal is applied (reset state).<br>1: Reset signal is stopped (released from the reset state). |

### 7.2.4.80 Reset Control Register SRAM_ACPU (CPG_RST_SRAM_ACPU)

This register is used to control the reset signals for individual modules.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | UNIT0_RSTWEN |
| 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 | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | UNIT0_RSTB |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 17 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 16 | UNIT0_RSTWEN | 0b | R0W1 | Flag for enabling the writing to the UNIT0_RSTB bit (bit 0)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 1 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 0 | UNIT0_RSTB | 1b | RW | The SRAM_ACPU_ARESETN reset terminal is controlled.<br>0: Reset signal is applied (reset state).<br>1: Reset signal is stopped (released from the reset state). |

### 7.2.4.81 Reset Control Register SRAM_MCPU (CPG_RST_SRAM_MCPU)

This register is used to control the reset signals for individual modules.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | UNIT0_RSTWEN |
| 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 | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | UNIT0_RSTB |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 17 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 16 | UNIT0_RSTWEN | 0b | R0W1 | Flag for enabling the writing to the UNIT0_RSTB bit (bit 0)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 1 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 0 | UNIT0_RSTB | 1b | RW | The SRAM_MCPU_ARESETN reset terminal is controlled.<br>0: Reset signal is applied (reset state).<br>1: Reset signal is stopped (released from the reset state). |

### 7.2.4.82 Reset Control Register GIC600 (CPG_RST_GIC600)

This register is used to control the reset signals for individual modules.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | UNIT1_RSTWEN | UNIT0_RSTWEN |
| 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 | R0W1 | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | UNIT1_RSTB | UNIT0_RSTB |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | RW | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 18 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 17 | UNIT1_RSTWEN | 0b | R0W1 | Flag for enabling the writing to the UNIT1_RSTB bit (bit 1)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 16 | UNIT0_RSTWEN | 0b | R0W1 | Flag for enabling the writing to the UNIT0_RSTB bit (bit 0)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 2 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 1 | UNIT1_RSTB | 1b | RW | The GIC600_DBG_GICRESET_N reset terminal is controlled.<br>0: Reset signal is applied (reset state).<br>1: Reset signal is stopped (released from the reset state). |
| 0 | UNIT0_RSTB | 1b | RW | The GIC600_GICRESET_N reset terminal is controlled.<br>0: Reset signal is applied (reset state).<br>1: Reset signal is stopped (released from the reset state). |

### 7.2.4.83 Reset Control Register IA55 (CPG_RST_IA55)

This register is used to control the reset signals for individual modules.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | UNIT0_RSTWEN |
| 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 | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | UNIT0_RSTB |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 17 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 16 | UNIT0_RSTWEN | 0b | R0W1 | Flag for enabling the writing to the UNIT0_RSTB bit (bit 0)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 1 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 0 | UNIT0_RSTB | 1b | RW | The IA55_RESETN reset terminal is controlled.<br>0: Reset signal is applied (reset state).<br>1: Reset signal is stopped (released from the reset state). |

### 7.2.4.84 Reset Control Register SYC (CPG_RST_SYC)

This register is used to control the reset signals for individual modules.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | UNIT0_RSTWEN |
| 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 | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | UNIT0_RSTB |
| 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 | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 17 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 16 | UNIT0_RSTWEN | 0b | R0W1 | Flag for enabling the writing to the UNIT0_RSTB bit (bit 0)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 1 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 0 | UNIT0_RSTB | 0b | RW | The SYC_RESETN reset terminal is controlled.<br>0: Reset signal is applied (reset state).<br>1: Reset signal is stopped (released from the reset state). |

### 7.2.4.85 Reset Control Register DMAC (CPG_RST_DMAC)

This register is used to control the reset signals for individual modules.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | UNIT1_RSTWEN | UNIT0_RSTWEN |
| 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 | R0W1 | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | UNIT1_RSTB | UNIT0_RSTB |
| 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 | RW | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 18 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 17 | UNIT1_RSTWEN | 0b | R0W1 | Flag for enabling the writing to the UNIT1_RSTB bit (bit 1)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 16 | UNIT0_RSTWEN | 0b | R0W1 | Flag for enabling the writing to the UNIT0_RSTB bit (bit 0)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 2 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 1 | UNIT1_RSTB | 0b | RW | The DMAC_RST_ASYNC reset terminal is controlled.<br>0: Reset signal is applied (reset state).<br>1: Reset signal is stopped (released from the reset state). |
| 0 | UNIT0_RSTB | 0b | RW | The DMAC_ARESETN reset terminal is controlled.<br>0: Reset signal is applied (reset state).<br>1: Reset signal is stopped (released from the reset state). |

### 7.2.4.86 Reset Control Register GTM (CPG_RST_GTM)

This register is used to control the reset signals for individual modules.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | UNIT2_RST WEN | UNIT1_RST WEN | UNIT0_RST WEN |
| 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 | R0W1 | R0W1 | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | UNIT2_RSTB | UNIT1_RSTB | UNIT0_RSTB |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | RW | RW | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 19 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 18 | UNIT2_RST WEN | 0b | R0W1 | Flag for enabling the writing to the UNIT2_RSTB bit (bit 2)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 17 | UNIT1_RST WEN | 0b | R0W1 | Flag for enabling the writing to the UNIT1_RSTB bit (bit 1)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 16 | UNIT0_RST WEN | 0b | R0W1 | Flag for enabling the writing to the UNIT0_RSTB bit (bit 0)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 3 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 2 | UNIT2_RSTB | 0b | RW | The OSTM2_PRESETZ reset terminal is controlled.<br>0: Reset signal is applied (reset state).<br>1: Reset signal is stopped (released from the reset state). |
| 1 | UNIT1_RSTB | 0b | RW | The OSTM1_PRESETZ reset terminal is controlled.<br>0: Reset signal is applied (reset state).<br>1: Reset signal is stopped (released from the reset state). |
| 0 | UNIT0_RSTB | 1b | RW | The OSTM0_PRESETZ reset terminal is controlled.<br>0: Reset signal is applied (reset state).<br>1: Reset signal is stopped (released from the reset state). |

### 7.2.4.87 Reset Control Register MTU (CPG_RST_MTU)

This register is used to control the reset signals for individual modules.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | UNIT0_RSTWEN |
| 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 | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | UNIT0_RSTB |
| 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 | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 17 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 16 | UNIT0_RSTWEN | 0b | R0W1 | Flag for enabling the writing to the UNIT0_RSTB bit (bit 0)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 1 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 0 | UNIT0_RSTB | 0b | RW | The MTU_X_PRESET_MTU3 reset terminal is controlled.<br>0: Reset signal is applied (reset state).<br>1: Reset signal is stopped (released from the reset state). |

### 7.2.4.88 Reset Control Register POE3 (CPG_RST_POE3)

This register is used to control the reset signals for individual modules.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | UNIT0_RSTWEN |
| 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 | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | UNIT0_RSTB |
| 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 | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 17 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 16 | UNIT0_RSTWEN | 0b | R0W1 | Flag for enabling the writing to the UNIT0_RSTB bit (bit 0)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 1 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 0 | UNIT0_RSTB | 0b | RW | The POE3_RST_M_REG reset terminal is controlled.<br>0: Reset signal is applied (reset state).<br>1: Reset signal is stopped (released from the reset state). |

### 7.2.4.89 Reset Control Register WDT (CPG_RST_WDT)

This register is used to control the reset signals for individual modules.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | UNIT0_RSTWEN |
| 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 | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | UNIT0_RSTB |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 19 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 18 | — | 0b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value.<br>Operation is not guaranteed if a value other than the initial value is written. |
| 17 | — | 0b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value.<br>Operation is not guaranteed if a value other than the initial value is written. |
| 16 | UNIT0_RSTWEN | 0b | R0W1 | Flag for enabling the writing to the UNIT0_RSTB bit (bit 0)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 4 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 3 | — | 1b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value.<br>Operation is not guaranteed if a value other than the initial value is written. |
| 2 | — | 0b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value.<br>Operation is not guaranteed if a value other than the initial value is written. |
| 1 | — | 0b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value.<br>Operation is not guaranteed if a value other than the initial value is written. |
| 0 | UNIT0_RSTB | 1b | RW | The WDT0_PRESETN reset terminal is controlled.<br>0: Reset signal is applied (reset state).<br>1: Reset signal is stopped (released from the reset state). |

### 7.2.4.90 Reset Control Register DDR (CPG_RST_DDR)

This register is used to control the reset signals for individual modules.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | UNIT6_RST WEN | UNIT5_RST WEN | UNIT4_RST WEN | UNIT3_RST WEN | UNIT2_RST WEN | UNIT1_RST WEN | UNIT0_RST WEN |
| 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 | R0W1 | R0W1 | R0W1 | R0W1 | R0W1 | R0W1 | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | UNIT6_RSTB | UNIT5_RSTB | UNIT4_RSTB | UNIT3_RSTB | UNIT2_RSTB | UNIT1_RSTB | UNIT0_RSTB |
| 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 | RW | RW | RW | RW | RW | RW | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 23 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 22 | UNIT6_RST WEN | 0b | R0W1 | Flag for enabling the writing to the UNIT6_RSTB bit (bit 6)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 21 | UNIT5_RST WEN | 0b | R0W1 | Flag for enabling the writing to the UNIT5_RSTB bit (bit 5)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 20 | UNIT4_RST WEN | 0b | R0W1 | Flag for enabling the writing to the UNIT4_RSTB bit (bit 4)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 19 | UNIT3_RST WEN | 0b | R0W1 | Flag for enabling the writing to the UNIT3_RSTB bit (bit 3)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 18 | UNIT2_RST WEN | 0b | R0W1 | Flag for enabling the writing to the UNIT2_RSTB bit (bit 2)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 17 | UNIT1_RST WEN | 0b | R0W1 | Flag for enabling the writing to the UNIT1_RSTB bit (bit 1)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 16 | UNIT0_RST WEN | 0b | R0W1 | Flag for enabling the writing to the UNIT0_RSTB bit (bit 0)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 7 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 6 | UNIT6_RSTB | 0b | RW | The DDR_REG_ARESETN reset terminal is controlled.<br>0: Reset signal is applied (reset state).<br>1: Reset signal is stopped (released from the reset state). |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 5 | UNIT5_RSTB | 0b | RW | The DDR_AXI3_ARESETN reset terminal is controlled.<br>0: Reset signal is applied (reset state).<br>1: Reset signal is stopped (released from the reset state). |
| 4 | UNIT4_RSTB | 0b | RW | The DDR_AXI2_ARESETN reset terminal is controlled.<br>0: Reset signal is applied (reset state).<br>1: Reset signal is stopped (released from the reset state). |
| 3 | UNIT3_RSTB | 0b | RW | The DDR_AXI1_ARESETN reset terminal is controlled.<br>0: Reset signal is applied (reset state).<br>1: Reset signal is stopped (released from the reset state). |
| 2 | UNIT2_RSTB | 0b | RW | The DDR_AXI0_ARESETN reset terminal is controlled.<br>0: Reset signal is applied (reset state).<br>1: Reset signal is stopped (released from the reset state). |
| 1 | UNIT1_RSTB | 0b | RW | The DDR_PRESETN reset terminal is controlled.<br>0: Reset signal is applied (reset state).<br>1: Reset signal is stopped (released from the reset state). |
| 0 | UNIT0_RSTB | 0b | RW | The DDR_RESET_N reset terminal is controlled.<br>0: Reset signal is applied (reset state).<br>1: Reset signal is stopped (released from the reset state). |

### 7.2.4.91 Reset Control Register SPI (CPG_RST_SPI)

This register is used to control the reset signals for individual modules.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | UNIT0_RSTWEN |
| 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 | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | UNIT0_RSTB |
| 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 | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 17 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 16 | UNIT0_RSTWEN | 0b | R0W1 | Flag for enabling the writing to the UNIT0_RSTB bit (bit 0)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 1 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 0 | UNIT0_RSTB | 0b | RW | The SPI_RST reset terminal is controlled.<br>0: Reset signal is applied (reset state).<br>1: Reset signal is stopped (released from the reset state). |

### 7.2.4.92 Reset Control Register SDHI (CPG_RST_SDHI)

This register is used to control the reset signals for individual modules.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | UNIT0_RSTWEN |
| 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 | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | UNIT0_RSTB |
| 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 | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 18 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 17 | — | 0b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 16 | UNIT0_RSTWEN | 0b | R0W1 | Flag for enabling the writing to the UNIT0_RSTB bit (bit 0)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 2 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 1 | — | 0b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 0 | UNIT0_RSTB | 0b | RW | The SDHI0_IXRST reset terminal is controlled.<br>0: Reset signal is applied (reset state).<br>1: Reset signal is stopped (released from the reset state). |

### 7.2.4.93 Reset Control Register MIPI_DSI (CPG_RST_MIPI_DSI)

This register is used to control the reset signals for individual modules.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | UNIT2_RST WEN | UNIT1_RST WEN | UNIT0_RST WEN |
| 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 | R0W1 | R0W1 | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | UNIT2_RSTB | UNIT1_RSTB | UNIT0_RSTB |
| 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 | RW | RW | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 19 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 18 | UNIT2_RST WEN | 0b | R0W1 | Flag for enabling the writing to the UNIT2_RSTB bit (bit 2)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 17 | UNIT1_RST WEN | 0b | R0W1 | Flag for enabling the writing to the UNIT1_RSTB bit (bit 1)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 16 | UNIT0_RST WEN | 0b | R0W1 | Flag for enabling the writing to the UNIT0_RSTB bit (bit 0)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 3 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 2 | UNIT2_RSTB | 0b | RW | The MIPI_DSI_PRESET_N reset terminal is controlled.<br>0: Reset signal is applied (reset state).<br>1: Reset signal is stopped (released from the reset state). |
| 1 | UNIT1_RSTB | 0b | RW | The MIPI_DSI_ARESET_N reset terminal is controlled.<br>0: Reset signal is applied (reset state).<br>1: Reset signal is stopped (released from the reset state). |
| 0 | UNIT0_RSTB | 0b | RW | The MIPI_DSI_CMN_RSTB reset terminal is controlled.<br>0: Reset signal is applied (reset state).<br>1: Reset signal is stopped (released from the reset state). |

### 7.2.4.94 Reset Control Register LCDC (CPG_RST_LCDC)

This register is used to control the reset signals for individual modules.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | UNIT0_RSTWEN |
| 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 | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | UNIT0_RSTB |
| 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 | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 17 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 16 | UNIT0_RSTWEN | 0b | R0W1 | Flag for enabling the writing to the UNIT0_RSTB bit (bit 0)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 1 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 0 | UNIT0_RSTB | 0b | RW | The LCDC_RESET_N reset terminal is controlled.<br>0: Reset signal is applied (reset state).<br>1: Reset signal is stopped (released from the reset state). |

### 7.2.4.95 Reset Control Register SSIF (CPG_RST_SSIF)

This register is used to control the reset signals for individual modules.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | UNIT0_RSTWEN |
| 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 | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | UNIT0_RSTB |
| 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 | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 20 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 19 to 17 | — | 000b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 16 | UNIT0_RSTWEN | 0b | R0W1 | Flag for enabling the writing to the UNIT0_RSTB bit (bit 0)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 4 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 3 to 1 | — | 000b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 0 | UNIT0_RSTB | 0b | RW | The SSI0_RST_M2_REG reset terminal is controlled.<br>0: Reset signal is applied (reset state).<br>1: Reset signal is stopped (released from the reset state). |

### 7.2.4.96 Reset Control Register USB (CPG_RST_USB)

This register is used to control the reset signals for individual modules.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | UNIT3_RST WEN | UNIT2_RST WEN | — | UNIT0_RST WEN |
| 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 | R0W1 | R0W1 | R | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | UNIT3_RSTB | UNIT2_RSTB | — | UNIT0_RSTB |
| 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 | RW | RW | R | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 20 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 19 | UNIT3_RST WEN | 0b | R0W1 | Flag for enabling the writing to the UNIT3_RSTB bit (bit 3)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 18 | UNIT2_RST WEN | 0b | R0W1 | Flag for enabling the writing to the UNIT2_RSTB bit (bit 2)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 17 | — | 0b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 16 | UNIT0_RST WEN | 0b | R0W1 | Flag for enabling the writing to the UNIT0_RSTB bit (bit 0)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 4 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 3 | UNIT3_RSTB | 0b | RW | The USB_PRESETN reset terminal is controlled.<br>0: Reset signal is applied (reset state).<br>1: Reset signal is stopped (released from the reset state). |
| 2 | UNIT2_RSTB | 0b | RW | The USB_U2P_EXL_SYSRST reset terminal is controlled.<br>0: Reset signal is applied (reset state).<br>1: Reset signal is stopped (released from the reset state). |
| 1 | — | 0b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 0 | UNIT0_RSTB | 0b | RW | The USB_U2H0_HRESETN reset terminal is controlled.<br>0: Reset signal is applied (reset state).<br>1: Reset signal is stopped (released from the reset state). |

### 7.2.4.97 Reset Control Register I2C (CPG_RST_I2C)

This register is used to control the reset signals for individual modules.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | UNIT1_RST WEN | UNIT0_RST WEN |
| 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 | R0W1 | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | UNIT1_RSTB | UNIT0_RSTB |
| 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 | RW | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 20 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 19,18 | — | 00b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 17 | UNIT1_RST WEN | 0b | R0W1 | Flag for enabling the writing to the UNIT1_RSTB bit (bit 1)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 16 | UNIT0_RST WEN | 0b | R0W1 | Flag for enabling the writing to the UNIT0_RSTB bit (bit 0)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 4 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 3,2 | — | 00b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 1 | UNIT1_RSTB | 0b | RW | The I2C1_MRST reset terminal is controlled.<br>0: Reset signal is applied (reset state).<br>1: Reset signal is stopped (released from the reset state). |
| 0 | UNIT0_RSTB | 0b | RW | The I2C0_MRST reset terminal is controlled.<br>0: Reset signal is applied (reset state).<br>1: Reset signal is stopped (released from the reset state). |

### 7.2.4.98 Reset Control Register SCIF (CPG_RST_SCIF)

This register is used to control the reset signals for individual modules.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | UNIT4_RST WEN | UNIT3_RST WEN | UNIT2_RST WEN | UNIT1_RST WEN | UNIT0_RST WEN |
| 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 | R0W1 | R0W1 | R0W1 | R0W1 | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | UNIT4_RSTB | UNIT3_RSTB | UNIT2_RSTB | UNIT1_RSTB | UNIT0_RSTB |
| 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 | RW | RW | RW | RW | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 21 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 20 | UNIT4_RST WEN | 0b | R0W1 | Flag for enabling the writing to the UNIT4_RSTB bit (bit 4)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 19 | UNIT3_RST WEN | 0b | R0W1 | Flag for enabling the writing to the UNIT3_RSTB bit (bit 3)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 18 | UNIT2_RST WEN | 0b | R0W1 | Flag for enabling the writing to the UNIT2_RSTB bit (bit 2)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 17 | UNIT1_RST WEN | 0b | R0W1 | Flag for enabling the writing to the UNIT1_RSTB bit (bit 1)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 16 | UNIT0_RST WEN | 0b | R0W1 | Flag for enabling the writing to the UNIT0_RSTB bit (bit 0)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 5 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 4 | UNIT4_RSTB | 0b | RW | The SCIF4_RST_SYSTEM_N reset terminal is controlled.<br>0: Reset signal is applied (reset state).<br>1: Reset signal is stopped (released from the reset state). |
| 3 | UNIT3_RSTB | 0b | RW | The SCIF3_RST_SYSTEM_N reset terminal is controlled.<br>0: Reset signal is applied (reset state).<br>1: Reset signal is stopped (released from the reset state). |
| 2 | UNIT2_RSTB | 0b | RW | The SCIF2_RST_SYSTEM_N reset terminal is controlled.<br>0: Reset signal is applied (reset state).<br>1: Reset signal is stopped (released from the reset state). |
| 1 | UNIT1_RSTB | 0b | RW | The SCIF1_RST_SYSTEM_N reset terminal is controlled.<br>0: Reset signal is applied (reset state).<br>1: Reset signal is stopped (released from the reset state). |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 0 | UNIT0_RSTB | 0b | RW | The SCIF0_RST_SYSTEM_N reset terminal is controlled.<br>0: Reset signal is applied (reset state).<br>1: Reset signal is stopped (released from the reset state). |

### 7.2.4.99 Reset Control Register SCI (CPG_RST_SCI)

This register is used to control the reset signals for individual modules.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | UNIT1_RSTWEN | UNIT0_RSTWEN |
| 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 | R0W1 | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | UNIT1_RSTB | UNIT0_RSTB |
| 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 | RW | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 18 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 17 | UNIT1_RSTWEN | 0b | R0W1 | Flag for enabling the writing to the UNIT1_RSTB bit (bit 1)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 16 | UNIT0_RSTWEN | 0b | R0W1 | Flag for enabling the writing to the UNIT0_RSTB bit (bit 0)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 2 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 1 | UNIT1_RSTB | 0b | RW | The SCI1_RST reset terminal is controlled.<br>0: Reset signal is applied (reset state).<br>1: Reset signal is stopped (released from the reset state). |
| 0 | UNIT0_RSTB | 0b | RW | The SCI0_RST reset terminal is controlled.<br>0: Reset signal is applied (reset state).<br>1: Reset signal is stopped (released from the reset state). |

### 7.2.4.100 Reset Control Register IRDA (CPG_RST_IRDA)

This register is used to control the reset signals for individual modules.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | UNIT0_RSTWEN |
| 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 | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | UNIT0_RSTB |
| 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 | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 17 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 16 | UNIT0_RSTWEN | 0b | R0W1 | Flag for enabling the writing to the UNIT0_RSTB bit (bit 0)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 1 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 0 | UNIT0_RSTB | 0b | RW | The IRDA_RST reset terminal is controlled.<br>0: Reset signal is applied (reset state).<br>1: Reset signal is stopped (released from the reset state). |

### 7.2.4.101 Reset Control Register RSPI (CPG_RST_RSPI)

This register is used to control the reset signals for individual modules.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | UNIT1_RST WEN | UNIT0_RST WEN |
| 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 | R0W1 | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | UNIT1_RSTB | UNIT0_RSTB |
| 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 | RW | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 19 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 18 | — | 0b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 17 | UNIT1_RST WEN | 0b | R0W1 | Flag for enabling the writing to the UNIT1_RSTB bit (bit 1)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 16 | UNIT0_RST WEN | 0b | R0W1 | Flag for enabling the writing to the UNIT0_RSTB bit (bit 0)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 3 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 2 | — | 0b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 1 | UNIT1_RSTB | 0b | RW | The RSPI1_RST reset terminal is controlled.<br>0: Reset signal is applied (reset state).<br>1: Reset signal is stopped (released from the reset state). |
| 0 | UNIT0_RSTB | 0b | RW | The RSPI0_RST reset terminal is controlled.<br>0: Reset signal is applied (reset state).<br>1: Reset signal is stopped (released from the reset state). |

### 7.2.4.102 Reset Control Register GPIO (CPG_RST_GPIO)

This register is used to control the reset signals for individual modules.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | UNIT2_RST WEN | UNIT1_RST WEN | UNIT0_RST WEN |
| 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 | R0W1 | R0W1 | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | UNIT2_RSTB | UNIT1_RSTB | UNIT0_RSTB |
| 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 | RW | RW | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 19 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 18 | UNIT2_RST WEN | 0b | R0W1 | Flag for enabling the writing to the UNIT2_RSTB bit (bit 2)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 17 | UNIT1_RST WEN | 0b | R0W1 | Flag for enabling the writing to the UNIT1_RSTB bit (bit 1)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 16 | UNIT0_RST WEN | 0b | R0W1 | Flag for enabling the writing to the UNIT0_RSTB bit (bit 0)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 3 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 2 | UNIT2_RSTB | 0b | RW | The GPIO_SPARE_RESETN reset terminal is controlled.<br>0: Reset signal is applied (reset state).<br>1: Reset signal is stopped (released from the reset state). |
| 1 | UNIT1_RSTB | 0b | RW | The GPIO_PORT_RESETN reset terminal is controlled.<br>0: Reset signal is applied (reset state).<br>1: Reset signal is stopped (released from the reset state). |
| 0 | UNIT0_RSTB | 0b | RW | The GPIO_RSTN reset terminal is controlled.<br>0: Reset signal is applied (reset state).<br>1: Reset signal is stopped (released from the reset state). |

### 7.2.4.103 Reset Control Register TSU (CPG_RST_TSU)

This register is used to control the reset signals for individual modules.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | UNIT0_RSTWEN |
| 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 | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | UNIT0_RSTB |
| 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 | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 17 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 16 | UNIT0_RSTWEN | 0b | R0W1 | Flag for enabling the writing to the UNIT0_RSTB bit (bit 0)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 1 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 0 | UNIT0_RSTB | 0b | RW | The TSU_PRESETN reset terminal is controlled.<br>0: Reset signal is applied (reset state).<br>1: Reset signal is stopped (released from the reset state). |

### 7.2.4.104 Reset Control Register DRW (CPG_RST_DRW)

This register is used to control the reset signals for individual modules.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | UNIT0_RSTWEN |
| 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 | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | UNIT0_RSTB |
| 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 | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 17 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 16 | UNIT0_RSTWEN | 0b | R0W1 | Flag for enabling the writing to the UNIT0_RSTB bit (bit 0)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 1 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 0 | UNIT0_RSTB | 0b | RW | The DRW_RESETN reset terminal is controlled.<br>0: Reset signal is applied (reset state).<br>1: Reset signal is stopped (released from the reset state). |

### 7.2.4.105 Reset Monitor Register Cortex-A55 (CPG_RSTMON_CA55)

This register is used to monitor the reset signals of individual modules.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | RST12_MON | RST11_MON | RST10_MON | RST9_MON | RST8_MON | RST7_MON | RST6_MON | RST5_MON | RST4_MON | — | RST2_MON | — | RST0_MON |
| 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 13 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 12 | RST12_MON | 0b | R | The state of the CA55_RST12 reset signal is monitored.<br>0: Reset signal is not applied (released from the reset state).<br>1: Reset signal is applied (reset state). |
| 11 | RST11_MON | 0b | R | The state of the CA55_RST11 reset signal is monitored.<br>0: Reset signal is not applied (released from the reset state).<br>1: Reset signal is applied (reset state). |
| 10 | RST10_MON | 0b | R | The state of the CA55_RST10 reset signal is monitored.<br>0: Reset signal is not applied (released from the reset state).<br>1: Reset signal is applied (reset state). |
| 9 | RST9_MON | 0b | R | The state of the CA55_RST9 reset signal is monitored.<br>0: Reset signal is not applied (released from the reset state).<br>1: Reset signal is applied (reset state). |
| 8 | RST8_MON | 0b | R | The state of the CA55_RST8 reset signal is monitored.<br>0: Reset signal is not applied (released from the reset state).<br>1: Reset signal is applied (reset state). |
| 7 | RST7_MON | 0b | R | The state of the CA55_RST7 reset signal is monitored.<br>0: Reset signal is not applied (released from the reset state).<br>1: Reset signal is applied (reset state). |
| 6 | RST6_MON | 0b | R | The state of the CA55_RST6 reset signal is monitored.<br>0: Reset signal is not applied (released from the reset state).<br>1: Reset signal is applied (reset state). |
| 5 | RST5_MON | 0b | R | The state of the CA55_RST5 reset signal is monitored.<br>0: Reset signal is not applied (released from the reset state).<br>1: Reset signal is applied (reset state). |
| 4 | RST4_MON | 0b | R | The state of the CA55_RST4 reset signal is monitored.<br>0: Reset signal is not applied (released from the reset state).<br>1: Reset signal is applied (reset state). |
| 3 | — | 0b | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 2 | RST2_MON | 0b | R | The state of the CA55_RST3_0 reset signal is monitored.<br>0: Reset signal is not applied (released from the reset state).<br>1: Reset signal is applied (reset state). |
| 1 | — | 0b | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 0 | RST0_MON | 0b | R | The state of the CA55_RST1_0 reset signal is monitored.<br>0: Reset signal is not applied (released from the reset state).<br>1: Reset signal is applied (reset state). |

### 7.2.4.106 Reset Monitor Register SRAM_ACPU (CPG_RSTMON_SRAM_ACPU)

This register is used to monitor the reset signals of individual modules.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | RST0_MON |
| 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 1 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 0 | RST0_MON | 0b | R | The state of the SRAM_ACPU_ARESETN reset signal is monitored.<br>0: Reset signal is not applied (released from the reset state).<br>1: Reset signal is applied (reset state). |

### 7.2.4.107 Reset Monitor Register SRAM_MCPU (CPG_RSTMON_SRAM_MCPU)

This register is used to monitor the reset signals of individual modules.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | RST0_MON |
| 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 1 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 0 | RST0_MON | 0b | R | The state of the SRAM_MCPU_ARESETN reset signal is monitored.<br>0: Reset signal is not applied (released from the reset state).<br>1: Reset signal is applied (reset state). |

### 7.2.4.108 Reset Monitor Register GIC600 (CPG_RSTMON_GIC600)

This register is used to monitor the reset signals of individual modules.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | RST1_MON | RST0_MON |
| 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 2 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 1 | RST1_MON | 0b | R | The state of the GIC600_DBG_GICRESET_N reset signal is monitored.<br>0: Reset signal is not applied (released from the reset state).<br>1: Reset signal is applied (reset state). |
| 0 | RST0_MON | 0b | R | The state of the GIC600_GICRESET_N reset signal is monitored.<br>0: Reset signal is not applied (released from the reset state).<br>1: Reset signal is applied (reset state). |

### 7.2.4.109 Reset Monitor Register IA55 (CPG_RSTMON_IA55)

This register is used to monitor the reset signals of individual modules.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | RST0_MON |
| 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 1 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 0 | RST0_MON | 0b | R | The state of the IA55_RESETN reset signal is monitored.<br>0: Reset signal is not applied (released from the reset state).<br>1: Reset signal is applied (reset state). |

### 7.2.4.110 Reset Monitor Register SYC (CPG_RSTMON_SYC)

This register is used to monitor the reset signals of individual modules.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | RST0_MON |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 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 1 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 0 | RST0_MON | 1b | R | The state of the SYC_RESETN reset signal is monitored.<br>0: Reset signal is not applied (released from the reset state).<br>1: Reset signal is applied (reset state). |

### 7.2.4.111 Reset Monitor Register DMAC (CPG_RSTMON_DMAC)

This register is used to monitor the reset signals of individual modules.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | RST1_MON | RST0_MON |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 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 2 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 1 | RST1_MON | 1b | R | The state of the DMAC_RST_ASYNC reset signal is monitored.<br>0: Reset signal is not applied (released from the reset state).<br>1: Reset signal is applied (reset state). |
| 0 | RST0_MON | 1b | R | The state of the DMAC_ARESETN reset signal is monitored.<br>0: Reset signal is not applied (released from the reset state).<br>1: Reset signal is applied (reset state). |

### 7.2.4.112 Reset Monitor Register GTM (CPG_RSTMON_GTM)

This register is used to monitor the reset signals of individual modules.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | RST2_MON | RST1_MON | RST0_MON |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 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 3 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 2 | RST2_MON | 1b | R | The state of the OSTM2_PRESETZ reset signal is monitored.<br>0: Reset signal is not applied (released from the reset state).<br>1: Reset signal is applied (reset state). |
| 1 | RST1_MON | 1b | R | The state of the OSTM1_PRESETZ reset signal is monitored.<br>0: Reset signal is not applied (released from the reset state).<br>1: Reset signal is applied (reset state). |
| 0 | RST0_MON | 0b | R | The state of the OSTM0_PRESETZ reset signal is monitored.<br>0: Reset signal is not applied (released from the reset state).<br>1: Reset signal is applied (reset state). |

### 7.2.4.113 Reset Monitor Register MTU (CPG_RSTMON_MTU)

This register is used to monitor the reset signals of individual modules.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | RST0_MON |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 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 1 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 0 | RST0_MON | 1b | R | The state of the MTU_X_PRESET_MTU3 reset signal is monitored.<br>0: Reset signal is not applied (released from the reset state).<br>1: Reset signal is applied (reset state). |

### 7.2.4.114 Reset Monitor Register POE3 (CPG_RSTMON_POE3)

This register is used to monitor the reset signals of individual modules.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | RST0_MON |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 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 1 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 0 | RST0_MON | 1b | R | The state of the POE3_RST_M_REG reset signal is monitored.<br>0: Reset signal is not applied (released from the reset state).<br>1: Reset signal is applied (reset state). |

### 7.2.4.115 Reset Monitor Register WDT (CPG_RSTMON_WDT)

This register is used to monitor the reset signals of individual modules.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | RST0_MON |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 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 4 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 3 | — | 0b | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 2 | — | 1b | R | Reserved<br>When read, the initial value is read. The written value will be ignored. |
| 1 | — | 1b | R | Reserved<br>Whenever it is read, 1 is read. The written value will be ignored. |
| 0 | RST0_MON | 0b | R | The state of the WDT0_PRESETN reset signal is monitored.<br>0: Reset signal is not applied (released from the reset state).<br>1: Reset signal is applied (reset state). |

### 7.2.4.116 Reset Monitor Register DDR (CPG_RSTMON_DDR)

This register is used to monitor the reset signals of individual modules.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | RST6_MON | RST5_MON | RST4_MON | RST3_MON | RST2_MON | RST1_MON | RST0_MON |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 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 7 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 6 | RST6_MON | 1b | R | The state of the DDR_REG_ARESETN reset signal is monitored.<br>0: Reset signal is not applied (released from the reset state).<br>1: Reset signal is applied (reset state). |
| 5 | RST5_MON | 1b | R | The state of the DDR_AXI3_ARESETN reset signal is monitored.<br>0: Reset signal is not applied (released from the reset state).<br>1: Reset signal is applied (reset state). |
| 4 | RST4_MON | 1b | R | The state of the DDR_AXI2_ARESETN reset signal is monitored.<br>0: Reset signal is not applied (released from the reset state).<br>1: Reset signal is applied (reset state). |
| 3 | RST3_MON | 1b | R | The state of the DDR_AXI1_ARESETN reset signal is monitored.<br>0: Reset signal is not applied (released from the reset state).<br>1: Reset signal is applied (reset state). |
| 2 | RST2_MON | 1b | R | The state of the DDR_AXI0_ARESETN reset signal is monitored.<br>0: Reset signal is not applied (released from the reset state).<br>1: Reset signal is applied (reset state). |
| 1 | RST1_MON | 1b | R | The state of the DDR_PRESETN reset signal is monitored.<br>0: Reset signal is not applied (released from the reset state).<br>1: Reset signal is applied (reset state). |
| 0 | RST0_MON | 1b | R | The state of the DDR_RESET_N reset signal is monitored.<br>0: Reset signal is not applied (released from the reset state).<br>1: Reset signal is applied (reset state). |

### 7.2.4.117 Reset Monitor Register SPI (CPG_RSTMON_SPI)

This register is used to monitor the reset signals of individual modules.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | RST0_MON |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 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 1 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 0 | RST0_MON | 1b | R | The state of the SPI_RST reset signal is monitored.<br>0: Reset signal is not applied (released from the reset state).<br>1: Reset signal is applied (reset state). |

### 7.2.4.118 Reset Monitor Register SDHI (CPG_RSTMON_SDHI)

This register is used to monitor the reset signals of individual modules.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | RST0_MON |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 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 2 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 1 | — | 1b | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 0 | RST0_MON | 1b | R | The state of the SDHI0_IXRST reset signal is monitored.<br>0: Reset signal is not applied (released from the reset state).<br>1: Reset signal is applied (reset state). |

### 7.2.4.119 Reset Monitor Register MIPI_DSI (CPG_RSTMON_MIPI_DSI)

This register is used to monitor the reset signals of individual modules.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | RST2_MON | RST1_MON | RST0_MON |
| 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 | R | R | R | R | R | R | R | R | R | R |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 3 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 2 | RST2_MON | 1b | R | The state of the MIPI_DSI_PRESET_N reset signal is monitored.<br>0: Reset signal is not applied (released from the reset state).<br>1: Reset signal is applied (reset state). |
| 1 | RST1_MON | 1b | R | The state of the MIPI_DSI_ARESET_N reset signal is monitored.<br>0: Reset signal is not applied (released from the reset state).<br>1: Reset signal is applied (reset state). |
| 0 | RST0_MON | 1b | R | The state of the MIPI_DSI_CMN_RSTB reset signal is monitored.<br>0: Reset signal is not applied (released from the reset state).<br>1: Reset signal is applied (reset state). |

### 7.2.4.120 Reset Monitor Register LCDC (CPG_RSTMON_LCDC)

This register is used to monitor the reset signals of individual modules.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | RST0_MON |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 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 1 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 0 | RST0_MON | 1b | R | The state of the LCDC_RESET_N reset signal is monitored.<br>0: Reset signal is not applied (released from the reset state).<br>1: Reset signal is applied (reset state). |

### 7.2.4.121 Reset Monitor Register SSIF (CPG_RSTMON_SSIF)

This register is used to monitor the reset signals of individual modules.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | RST0_MON |
| 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 | R | R | R | R | R | R | R |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 4 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 3 to 1 | — | 111b | R | Reserved<br>Whenever it is read, 1 is read. The written value will be ignored. |
| 0 | RST0_MON | 1b | R | The state of the SSI0_RST_M2_REG reset signal is monitored.<br>0: Reset signal is not applied (released from the reset state).<br>1: Reset signal is applied (reset state). |

### 7.2.4.122 Reset Monitor Register USB (CPG_RSTMON_USB)

This register is used to monitor the reset signals of individual modules.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | RST3_MON | RST2_MON | — | RST0_MON |
| 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 | R | R | R | R | R | R | R |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 4 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 3 | RST3_MON | 1b | R | The state of the USB_PRESETN reset signal is monitored.<br>0: Reset signal is not applied (released from the reset state).<br>1: Reset signal is applied (reset state). |
| 2 | RST2_MON | 1b | R | The state of the USB_U2P_EXL_SYSRST reset signal is monitored.<br>0: Reset signal is not applied (released from the reset state).<br>1: Reset signal is applied (reset state). |
| 1 | — | 1b | R | Reserved<br>Whenever it is read, 1 is read. The written value will be ignored. |
| 0 | RST0_MON | 1b | R | The state of the USB_U2H0_HRESETN reset signal is monitored.<br>0: Reset signal is not applied (released from the reset state).<br>1: Reset signal is applied (reset state). |

### 7.2.4.123 Reset Monitor Register I2C (CPG_RSTMON_I2C)

This register is used to monitor the reset signals of individual modules.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | RST1_MON | RST0_MON |
| 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 | R | R | R | R | R | R | R |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 4 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 3,2 | — | 11b | R | Reserved<br>Whenever it is read, 1 is read. The written value will be ignored. |
| 1 | RST1_MON | 1b | R | The state of the I2C1_MRST reset signal is monitored.<br>0: Reset signal is not applied (released from the reset state).<br>1: Reset signal is applied (reset state). |
| 0 | RST0_MON | 1b | R | The state of the I2C0_MRST reset signal is monitored.<br>0: Reset signal is not applied (released from the reset state).<br>1: Reset signal is applied (reset state). |

### 7.2.4.124 Reset Monitor Register SCIF (CPG_RSTMON_SCIF)

This register is used to monitor the reset signals of individual modules.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | RST4_MON | RST3_MON | RST2_MON | RST1_MON | RST0_MON |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 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 5 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 4 | RST4_MON | 1b | R | The state of the SCIF4_RST_SYSTEM_N reset signal is monitored.<br>0: Reset signal is not applied (released from the reset state).<br>1: Reset signal is applied (reset state). |
| 3 | RST3_MON | 1b | R | The state of the SCIF3_RST_SYSTEM_N reset signal is monitored.<br>0: Reset signal is not applied (released from the reset state).<br>1: Reset signal is applied (reset state). |
| 2 | RST2_MON | 1b | R | The state of the SCIF2_RST_SYSTEM_N reset signal is monitored.<br>0: Reset signal is not applied (released from the reset state).<br>1: Reset signal is applied (reset state). |
| 1 | RST1_MON | 1b | R | The state of the SCIF1_RST_SYSTEM_N reset signal is monitored.<br>0: Reset signal is not applied (released from the reset state).<br>1: Reset signal is applied (reset state). |
| 0 | RST0_MON | 1b | R | The state of the SCIF0_RST_SYSTEM_N reset signal is monitored.<br>0: Reset signal is not applied (released from the reset state).<br>1: Reset signal is applied (reset state). |

### 7.2.4.125 Reset Monitor Register SCI (CPG_RSTMON_SCI)

This register is used to monitor the reset signals of individual modules.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | RST1_MON | RST0_MON |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 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 2 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 1 | RST1_MON | 1b | R | The state of the SCI1_RST reset signal is monitored.<br>0: Reset signal is not applied (released from the reset state).<br>1: Reset signal is applied (reset state). |
| 0 | RST0_MON | 1b | R | The state of the SCI0_RST reset signal is monitored.<br>0: Reset signal is not applied (released from the reset state).<br>1: Reset signal is applied (reset state). |

### 7.2.4.126 Reset Monitor Register IRDA (CPG_RSTMON_IRDA)

This register is used to monitor the reset signals of individual modules.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | RST0_MON |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 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 1 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 0 | RST0_MON | 1b | R | The state of the IRDA_RST reset signal is monitored.<br>0: Reset signal is not applied (released from the reset state).<br>1: Reset signal is applied (reset state). |

### 7.2.4.127 Reset Monitor Register RSPI (CPG_RSTMON_RSPI)

This register is used to monitor the reset signals of individual modules.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | RST1_MON | RST0_MON |
| 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 | R | R | R | R | R | R | R | R | R | R |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 3 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 2 | — | 1b | R | Reserved<br>Whenever it is read, 1 is read. The written value will be ignored. |
| 1 | RST1_MON | 1b | R | The state of the RSPI1_RST reset signal is monitored.<br>0: Reset signal is not applied (released from the reset state).<br>1: Reset signal is applied (reset state). |
| 0 | RST0_MON | 1b | R | The state of the RSPI0_RST reset signal is monitored.<br>0: Reset signal is not applied (released from the reset state).<br>1: Reset signal is applied (reset state). |

### 7.2.4.128 Reset Monitor Register GPIO (CPG_RSTMON_GPIO)

This register is used to monitor the reset signals of individual modules.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | RST2_MON | RST1_MON | RST0_MON |
| 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 | R | R | R | R | R | R | R | R | R | R |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 3 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 2 | RST2_MON | 1b | R | The state of the GPIO_SPARE_RESETN reset signal is monitored.<br>0: Reset signal is not applied (released from the reset state).<br>1: Reset signal is applied (reset state). |
| 1 | RST1_MON | 1b | R | The state of the GPIO_PORT_RESETN reset signal is monitored.<br>0: Reset signal is not applied (released from the reset state).<br>1: Reset signal is applied (reset state). |
| 0 | RST0_MON | 1b | R | The state of the GPIO_RSTN reset signal is monitored.<br>0: Reset signal is not applied (released from the reset state).<br>1: Reset signal is applied (reset state). |

### 7.2.4.129 Reset Monitor Register TSU (CPG_RSTMON_TSU)

This register is used to monitor the reset signals of individual modules.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | RST0_MON |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 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 1 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 0 | RST0_MON | 1b | R | The state of the TSU_PRESETN reset signal is monitored.<br>0: Reset signal is not applied (released from the reset state).<br>1: Reset signal is applied (reset state). |

### 7.2.4.130 Reset Monitor Register DRW (CPG_RSTMON_DRW)

This register is used to monitor the reset signals of individual modules.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | RST0_MON |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 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 1 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 0 | RST0_MON | 1b | R | The state of the DRW_PRESETN reset signal is monitored.<br>0: Reset signal is not applied (released from the reset state).<br>1: Reset signal is applied (reset state). |

### 7.2.4.131 WDT Overflow System Reset Register (CPG_WDTOVF_RST)

After the release from a system reset applied by the WDT reset circuit in response to a reset request such as a WDT overflow, this register can be used to identify the WDT channel that generated the reset request. If a system reset is not applied according to the WDTRSTSEL register setting, the WDT is not reset and the source of the reset request can be identified by checking the interrupt from the WDT and the status register.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | WDTOVF0_WEN |
| 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 | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | WDTOVF0 |
| 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 | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 19 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 18 | — | 0b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value.<br>Operation is not guaranteed if a value other than the initial value is written. |
| 17 | — | 0b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value.<br>Operation is not guaranteed if a value other than the initial value is written. |
| 16 | WDTOVF0_WEN | 0b | R0W1 | Flag for enabling the writing to the WDTOVF0 bit (bit 0)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 3 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 2 | — | 0b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value.<br>Operation is not guaranteed if a value other than the initial value is written. |
| 1 | — | 0b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value.<br>Operation is not guaranteed if a value other than the initial value is written. |
| 0 | WDTOVF0 | 0b | RW | The system reset generated WDT channel 0 for Cortex-A55 Core 0 is indicated.<br>0: System reset has not been generated by WDT channel 0 for Cortex-A55 Core 0.<br>1: System reset has been generated by WDT channel 0 for Cortex-A55 Core 0.<br>*Note*: This bit is cleared by writing 1. |

### 7.2.4.132 WDT Reset Selector Register (CPG_WDTRST_SEL)

This register is used to mask reset requests from the WDT.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | WDTRSTSEL8_WEN | — | — | — | WDTRSTSEL4_WEN | — | — | — | WDTRSTSEL0_WEN |
| 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 | R0W1 | R | R | R | R0W1 | R | R | R | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | WDTRSTSEL8 | — | — | — | WDTRSTSEL4 | — | — | — | WDTRSTSEL0 |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | RW | R | R | R | RW | R | R | R | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 27 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 26,25 | — | 00b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value.<br>Operation is not guaranteed if a value other than the initial value is written. |
| 24 | WDTRSTSEL8_WEN | 0b | R0W1 | Flag for enabling the writing to the WDTRSTSEL8 bit (bit 8)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 23 to 21 | — | 00b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value.<br>Operation is not guaranteed if a value other than the initial value is written. |
| 20 | WDTRSTSEL4_WEN | 0b | R0W1 | Flag for enabling the writing to the WDTRSTSEL4 bit (bit 4)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 19 to 17 | — | 0b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value.<br>Operation is not guaranteed if a value other than the initial value is written. |
| 16 | WDTRSTSEL0_WEN | 0b | R0W1 | Flag for enabling the writing to the WDTRSTSEL0 bit (bit 0)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 11 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 10,9 | — | 00b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value.<br>Operation is not guaranteed if a value other than the initial value is written. |
| 8 | WDTRSTSEL8 | 0b | RW | Whether to mask the reset request by WDT channel 0 is specified.<br>0: Cold reset of the Cortex-A55 is masked.<br>1: Cold reset of the Cortex-A55 is enabled. |
| 7 | — | 1b | R | Reserved<br>Whenever it is read, 1 is read. The written value will be ignored. |
| 6 | — | 0b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value.<br>Operation is not guaranteed if a value other than the initial value is written. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 5 | — | 0b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 4 | WDTRSTSEL4 | 0b | RW | Whether to mask the assertion of WDTOVFOUT due to a request from WDT channel 0 for Cortex-A55 Core 0 is specified.<br>0: Assertion of WDTOVFOUT is masked.<br>1: Assertion of WDTOVFOUT is enabled. |
| 3 | — | 1b | R | Reserved<br>Whenever it is read, 1 is read. The written value will be ignored. |
| 2 | — | 0b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value.<br>Operation is not guaranteed if a value other than the initial value is written. |
| 1 | — | 0b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 0 | WDTRSTSEL0 | 0b | RW | Whether to mask WDTSYSREST[0] due to a request from WDT channel 0 for Cortex-A55 Core 0 is specified.<br>0: System reset is masked.<br>1: System reset is enabled. |

**Note:** Do not make the settings WDTRSTSEL3 = 1 and WDTRSTSEL7 = 0. If these settings are attempted, a system reset initializes the WDTRSTSEL7 bit to 1 (initial value) and WDTOVFOUT is unexpectedly asserted.

### 7.2.4.133 MSTOP Register ACPU (CPG_BUS_ACPU_MSTOP)

This register indicates the stop state of individual modules.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | MSTOP0_ON_WEN |
| 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 | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | MSTOP0_ON |
| 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 | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 17 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 16 | MSTOP0_ON_WEN | 0b | R0W1 | Flag for enabling the writing to the MSTOP0_ON bit (bit 0)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 1 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 0 | MSTOP0_ON | 0b | RW | The state of BUS_ACPU_MSTOP_MXSRAM_A operation is indicated.<br>0: Normal operation<br>1: Module stop state (MSTP) |

### 7.2.4.134 MSTOP Register MCPU1 (CPG_BUS_MCPU1_MSTOP)

This register indicates the stop state of individual modules.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MSTOP15_ON_WEN | MSTOP14_ON_WEN | — | — | — | MSTOP10_ON_WEN | MSTOP9_ON_WEN | — | — | — | — | — | — | MSTOP2_ON_WEN | MSTOP1_ON_WEN | MSTOP0_ON_WEN |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R0W1 | R0W1 | R | R | R | R0W1 | R0W1 | R | R | R | R | R | R | R0W1 | R0W1 | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MSTOP15_ON | MSTOP14_ON | — | — | — | MSTOP10_ON | MSTOP9_ON | — | — | — | — | — | — | MSTOP2_ON | MSTOP1_ON | MSTOP0_ON |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | RW | RW | R | R | R | RW | RW | R | R | R | R | R | R | RW | RW | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 | MSTOP15_ON_WEN | 0b | R0W1 | Flag for enabling the writing to the MSTOP15_ON bit (bit 15)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 30 | MSTOP14_ON_WEN | 0b | R0W1 | Flag for enabling the writing to the MSTOP14_ON bit (bit 14)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 29 to 27 | — | All 0 | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 26 | MSTOP10_ON_WEN | 0b | R0W1 | Flag for enabling the writing to the MSTOP10_ON bit (bit 10)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 25 | MSTOP9_ON_WEN | 0b | R0W1 | Flag for enabling the writing to the MSTOP9_ON bit (bit 9)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 24 to 19 | — | All 0 | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 18 | MSTOP2_ON_WEN | 0b | R0W1 | Flag for enabling the writing to the MSTOP2_ON bit (bit 2)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 17 | MSTOP1_ON_WEN | 0b | R0W1 | Flag for enabling the writing to the MSTOP1_ON bit (bit 1)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 16 | MSTOP0_ON_WEN | 0b | R0W1 | Flag for enabling the writing to the MSTOP0_ON bit (bit 0)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 | MSTOP15_ON | 0b | RW | The state of BUS_MCPU_MSTOP13_MHRSPI_1 operation is indicated.<br>0: Normal operation<br>1: Module stop state (MSTP) |
| 14 | MSTOP14_ON | 0b | RW | The state of BUS_MCPU_MSTOP12_MHRSPI_0 operation is indicated.<br>0: Normal operation<br>1: Module stop state (MSTP) |
| 13 to 11 | — | All 0 | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 10 | MSTOP10_ON | 0b | RW | The state of BUS_MCPU_MSTOP8_MHSSIF_0 operation is indicated.<br>0: Normal operation<br>1: Module stop state (MSTP) |
| 9 | MSTOP9_ON | 0b | RW | The state of BUS_MCPU_MSTOP7_MHPOE3 operation is indicated.<br>0: Normal operation<br>1: Module stop state (MSTP) |
| 8 to 3 | — | All 0 | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 2 | MSTOP2_ON | 0b | RW | The state of BUS_MCPU_MSTOP0_MHMTU3A operation is indicated.<br>0: Normal operation<br>1: Module stop state (MSTP) |
| 1 | MSTOP1_ON | 0b | RW | The state of BUS_MCPU_MSTOP_MHSPI operation is indicated.<br>0: Normal operation<br>1: Module stop state (MSTP) |
| 0 | MSTOP0_ON | 0b | RW | The state of BUS_MCPU_MSTOP_MXSRAM_M operation is indicated.<br>0: Normal operation<br>1: Module stop state (MSTP) |

### 7.2.4.135 MSTOP Register MCPU2 (CPG_BUS_MCPU2_MSTOP)

This register indicates the stop state of individual modules.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MSTOP15_ON_WEN | — | — | — | MSTOP11_ON_WEN | MSTOP10_ON_WEN | — | MSTOP8_ON_WEN | MSTOP7_ON_WEN | MSTOP6_ON_WEN | MSTOP5_ON_WEN | MSTOP4_ON_WEN | MSTOP3_ON_WEN | MSTOP2_ON_WEN | MSTOP1_ON_WEN | — |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R0W1 | R | R | R | R0W1 | R0W1 | R | R0W1 | R0W1 | R0W1 | R0W1 | R0W1 | R0W1 | R0W1 | R0W1 | R |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MSTOP15_ON | — | — | — | MSTOP11_ON | MSTOP10_ON | — | MSTOP8_ON | MSTOP7_ON | MSTOP6_ON | MSTOP5_ON | MSTOP4_ON | MSTOP3_ON | MSTOP2_ON | MSTOP1_ON | — |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | RW | R | R | R | RW | RW | R | RW | RW | RW | RW | RW | RW | RW | RW | R |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 | MSTOP15_ON_WEN | 0b | R0W1 | Flag for enabling the writing to the MSTOP15_ON bit (bit 15)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 30 to 28 | — | All 0 | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 27 | MSTOP11_ON_WEN | 0b | R0W1 | Flag for enabling the writing to the MSTOP11_ON bit (bit 11)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 26 | MSTOP10_ON_WEN | 0b | R0W1 | Flag for enabling the writing to the MSTOP10_ON bit (bit 10)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 25 | — | 0b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 24 | MSTOP8_ON_WEN | 0b | R0W1 | Flag for enabling the writing to the MSTOP8_ON bit (bit 8)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 23 | MSTOP7_ON_WEN | 0b | R0W1 | Flag for enabling the writing to the MSTOP7_ON bit (bit 7)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 22 | MSTOP6_ON_WEN | 0b | R0W1 | Flag for enabling the writing to the MSTOP6_ON bit (bit 6)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 21 | MSTOP5_ON_WEN | 0b | R0W1 | Flag for enabling the writing to the MSTOP5_ON bit (bit 5)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 20 | MSTOP4_ON_WEN | 0b | R0W1 | Flag for enabling the writing to the MSTOP4_ON bit (bit 4)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 19 | MSTOP3_ON_WEN | 0b | R0W1 | Flag for enabling the writing to the MSTOP3_ON bit (bit 3)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 18 | MSTOP2_ON_WEN | 0b | R0W1 | Flag for enabling the writing to the MSTOP2_ON bit (bit 2)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 17 | MSTOP1_ON_WEN | 0b | R0W1 | Flag for enabling the writing to the MSTOP1_ON bit (bit 1)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 16 | — | 0b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 15 | MSTOP15_ON | 0b | RW | The state of BUS_MCPU_MSTOP29_MPTSU operation is indicated.<br>0: Normal operation<br>1: Module stop state (MSTP) |
| 14 to 12 | — | All 0 | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 11 | MSTOP11_ON | 0b | RW | The state of BUS_MCPU_MSTOP25_MPI2C_1 operation is indicated.<br>0: Normal operation<br>1: Module stop state (MSTP) |
| 10 | MSTOP10_ON | 0b | RW | The state of BUS_MCPU_MSTOP24_MPI2C_0 operation is indicated.<br>0: Normal operation<br>1: Module stop state (MSTP) |
| 9 | — | 0b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 8 | MSTOP8_ON | 0b | RW | The state of BUS_MCPU_MSTOP22_MHSCI_1 operation is indicated.<br>0: Normal operation<br>1: Module stop state (MSTP) |
| 7 | MSTOP7_ON | 0b | RW | The state of BUS_MCPU_MSTOP21_MHSCI_0 operation is indicated.<br>0: Normal operation<br>1: Module stop state (MSTP) |
| 6 | MSTOP6_ON | 0b | RW | The state of BUS_MCPU_MSTOP20_MHIRDA operation is indicated.<br>0: Normal operation<br>1: Module stop state (MSTP) |
| 5 | MSTOP5_ON | 0b | RW | The state of BUS_MCPU_MSTOP19_MHSCIF_4 operation is indicated.<br>0: Normal operation<br>1: Module stop state (MSTP) |
| 4 | MSTOP4_ON | 0b | RW | The state of BUS_MCPU_MSTOP18_MHSCIF_3 operation is indicated.<br>0: Normal operation<br>1: Module stop state (MSTP) |
| 3 | MSTOP3_ON | 0b | RW | The state of BUS_MCPU_MSTOP17_MHSCIF_2 operation is indicated.<br>0: Normal operation<br>1: Module stop state (MSTP) |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 2 | MSTOP2_ON | 0b | RW | The state of BUS_MCPU_MSTOP16_MHSCIF_1 operation is indicated.<br>0: Normal operation<br>1: Module stop state (MSTP) |
| 1 | MSTOP1_ON | 0b | RW | The state of BUS_MCPU_MSTOP15_MHSCIF_0 operation is indicated.<br>0: Normal operation<br>1: Module stop state (MSTP) |
| 0 | — | 0b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |

### 7.2.4.136 MSTOP Register PERI_COM (CPG_BUS_PERI_COM_MSTOP)

This register indicates the stop state of individual modules.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | MSTOP9_ON_WEN | MSTOP8_ON_WEN | — | MSTOP6_ON_WEN | MSTOP5_ON_WEN | MSTOP4_ON_WEN | — | — | — | MSTOP0_ON_WEN |
| 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 | R0W1 | R0W1 | R | R0W1 | R0W1 | R0W1 | R | R | R | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | MSTOP9_ON | MSTOP8_ON | — | MSTOP6_ON | MSTOP5_ON | MSTOP4_ON | — | — | — | MSTOP0_ON |
| 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 | RW | RW | R | RW | RW | RW | R | R | R | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 26 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 25 | MSTOP9_ON_WEN | 0b | R0W1 | Flag for enabling the writing to the MSTOP9_ON bit (bit 9)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 24 | MSTOP8_ON_WEN | 0b | R0W1 | Flag for enabling the writing to the MSTOP8_ON bit (bit 8)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 23 | — | 0b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 22 | MSTOP6_ON_WEN | 0b | R0W1 | Flag for enabling the writing to the MSTOP6_ON bit (bit 6)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 21 | MSTOP5_ON_WEN | 0b | R0W1 | Flag for enabling the writing to the MSTOP5_ON bit (bit 5)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 20 | MSTOP4_ON_WEN | 0b | R0W1 | Flag for enabling the writing to the MSTOP4_ON bit (bit 4)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 19 to 17 | — | All 0 | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 16 | MSTOP0_ON_WEN | 0b | R0W1 | Flag for enabling the writing to the MSTOP0_ON bit (bit 0)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 10 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 9 | MSTOP9_ON | 0b | RW | The state of BUS_PERI_COM_MSTOP8_MXCOM operation is indicated.<br>0: Normal operation<br>1: Module stop state (MSTP) |
| 8 | MSTOP8_ON | 0b | RW | The state of BUS_PERI_COM_MSTOP8_MXCOM operation is indicated.<br>0: Normal operation<br>1: Module stop state (MSTP) |
| 7 | — | 0b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 6 | MSTOP6_ON | 0b | RW | The state of BUS_PERI_COM_MSTOP6_MHUSB2_0_F operation is indicated.<br>0: Normal operation<br>1: Module stop state (MSTP) |
| 5 | MSTOP5_ON | 0b | RW | The state of BUS_PERI_COM_MSTOP5_MHUSB2_0_H operation is indicated.<br>0: Normal operation<br>1: Module stop state (MSTP) |
| 4 | MSTOP4_ON | 0b | RW | The state of BUS_PERI_COM_MSTOP4_MPUSBT operation is indicated.<br>0: Normal operation<br>1: Module stop state (MSTP) |
| 3 to 1 | — | All 0 | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 0 | MSTOP0_ON | 0b | RW | The state of BUS_PERI_COM_MSTOP0_MXSDHI_0 operation is indicated.<br>0: Normal operation<br>1: Module stop state (MSTP) |

### 7.2.4.137 MSTOP Register PERI_CPU (CPG_BUS_PERI_CPU_MSTOP)

This register indicates the stop state of individual modules.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MSTOP15_ON_WEN | — | MSTOP13_ON_WEN | MSTOP12_ON_WEN | MSTOP11_ON_WEN | — | — | — | — | MSTOP6_ON_WEN | — | MSTOP4_ON_WEN | MSTOP3_ON_WEN | MSTOP2_ON_WEN | MSTOP1_ON_WEN | MSTOP0_ON_WEN |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R0W1 | R | R0W1 | R0W1 | R0W1 | R | R | R | R | R0W1 | R | R0W1 | R0W1 | R0W1 | R0W1 | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MSTOP15_ON | — | MSTOP13_ON | MSTOP12_ON | MSTOP11_ON | — | — | — | — | MSTOP6_ON | — | MSTOP4_ON | MSTOP3_ON | MSTOP2_ON | MSTOP1_ON | MSTOP0_ON |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | RW | R | RW | RW | RW | R | R | R | R | RW | R | RW | RW | RW | RW | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 | MSTOP15_ON_WEN | 0b | R0W1 | Flag for enabling the writing to the MSTOP15_ON bit (bit 15)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 30 | — | 0b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value.<br>Operation is not guaranteed if a value other than the initial value is written. |
| 29 | MSTOP13_ON_WEN | 0b | R0W1 | Flag for enabling the writing to the MSTOP13_ON bit (bit 13)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 28 | MSTOP12_ON_WEN | 0b | R0W1 | Flag for enabling the writing to the MSTOP12_ON bit (bit 12)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 27 | MSTOP11_ON_WEN | 0b | R0W1 | Flag for enabling the writing to the MSTOP11_ON bit (bit 11)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 26 to 23 | — | All 0 | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value.<br>Operation is not guaranteed if a value other than the initial value is written. |
| 22 | MSTOP6_ON_WEN | 0b | R0W1 | Flag for enabling the writing to the MSTOP6_ON bit (bit 6)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 21 | — | 0b | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 20 | MSTOP4_ON_WEN | 0b | R0W1 | Flag for enabling the writing to the MSTOP4_ON bit (bit 4)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 19 | MSTOP3_ON_WEN | 0b | R0W1 | Flag for enabling the writing to the MSTOP3_ON bit (bit 3)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 18 | MSTOP2_ON_WEN | 0b | R0W1 | Flag for enabling the writing to the MSTOP2_ON bit (bit 2)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 17 | MSTOP1_ON_WEN | 0b | R0W1 | Flag for enabling the writing to the MSTOP1_ON bit (bit 1)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 16 | MSTOP0_ON_WEN | 0b | R0W1 | Flag for enabling the writing to the MSTOP0_ON bit (bit 0)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 | MSTOP15_ON | 0b | RW | The state of BUS_PERI_CPU_MSTOP15_MXREG0 operation is indicated.<br>0: Normal operation<br>1: Module stop state (MSTP) |
| 14 | — | 0b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value.<br>Operation is not guaranteed if a value other than the initial value is written. |
| 13 | MSTOP13_ON | 0b | RW | The state of BUS_PERI_CPU_MSTOP13_MPIA55 operation is indicated.<br>0: Normal operation<br>1: Module stop state (MSTP) |
| 12 | MSTOP12_ON | 0b | RW | The state of BUS_PERI_CPU_MSTOP12_MPSRAM_M operation is indicated.<br>0: Normal operation<br>1: Module stop state (MSTP) |
| 11 | MSTOP11_ON | 0b | RW | The state of BUS_PERI_CPU_MSTOP11_MPSRAM_A operation is indicated.<br>0: Normal operation<br>1: Module stop state (MSTP) |
| 10 to 7 | — | All 0 | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value.<br>Operation is not guaranteed if a value other than the initial value is written. |
| 6 | MSTOP6_ON | 0b | RW | The state of BUS_PERI_CPU_MSTOP6_MHGPIO operation is indicated.<br>0: Normal operation<br>1: Module stop state (MSTP) |
| 5 | — | 0b | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 4 | MSTOP4_ON | 0b | RW | The state of BUS_PERI_CPU_MSTOP4_MPCPG operation is indicated.<br>0: Normal operation<br>1: Module stop state (MSTP) |
| 3 | MSTOP3_ON | 0b | RW | The state of BUS_PERI_CPU_MSTOP3_MPSYC operation is indicated.<br>0: Normal operation<br>1: Module stop state (MSTP) |
| 2 | MSTOP2_ON | 0b | RW | The state of BUS_PERI_CPU_MSTOP2_MPCST operation is indicated.<br>0: Normal operation<br>1: Module stop state (MSTP) |
| 1 | MSTOP1_ON | 0b | RW | The state of BUS_PERI_CPU_MSTOP1_MXACPU operation is indicated.<br>0: Normal operation<br>1: Module stop state (MSTP) |
| 0 | MSTOP0_ON | 0b | RW | The state of BUS_PERI_CPU_MSTOP0_MXMCPU operation is indicated.<br>0: Normal operation<br>1: Module stop state (MSTP) |

### 7.2.4.138 MSTOP Register PERI_DDR (CPG_BUS_PERI_DDR_MSTOP)

This register indicates the stop state of individual modules.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | MSTOP1_ON_WEN | MSTOP0_ON_WEN |
| 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 | R0W1 | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | MSTOP1_ON | MSTOP0_ON |
| 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 | RW | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 18 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 17 | MSTOP1_ON_WEN | 0b | R0W1 | Flag for enabling the writing to the MSTOP1_ON bit (bit 1)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 16 | MSTOP0_ON_WEN | 0b | R0W1 | Flag for enabling the writing to the MSTOP0_ON bit (bit 0)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 2 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 1 | MSTOP1_ON | 0b | RW | The state of BUS_PERI_DDR_MSTOP1_MXMEMC_REG operation is indicated.<br>0: Normal operation<br>1: Module stop state (MSTP) |
| 0 | MSTOP0_ON | 0b | RW | The state of BUS_PERI_DDR_MSTOP0_MPPHY operation is indicated.<br>0: Normal operation<br>1: Module stop state (MSTP) |

### 7.2.4.139 MSTOP Register PERI_VIDEO (CPG_BUS_PERI_VIDEO_MSTOP)

This register indicates the stop state of individual modules.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | MSTOP10_ON_WEN | MSTOP9_ON_WEN | MSTOP8_ON_WEN | MSTOP7_ON_WEN | MSTOP6_ON_WEN | MSTOP5_ON_WEN | — | — | — | — | — |
| 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 | R0W1 | R0W1 | R0W1 | R0W1 | R0W1 | R0W1 | R | R | R | R | R |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | MSTOP10_ON | MSTOP9_ON | MSTOP8_ON | MSTOP7_ON | MSTOP6_ON | MSTOP5_ON | — | — | — | — | — |
| 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 | RW | RW | RW | RW | RW | RW | R | R | R | R | R |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 27 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 26 | MSTOP10_ON_WEN | 0b | R0W1 | Flag for enabling the writing to the MSTOP10_ON bit (bit 10)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 25 | MSTOP9_ON_WEN | 0b | R0W1 | Flag for enabling the writing to the MSTOP9_ON bit (bit 9)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 24 | MSTOP8_ON_WEN | 0b | R0W1 | Flag for enabling the writing to the MSTOP8_ON bit (bit 8)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 23 | MSTOP7_ON_WEN | 0b | R0W1 | Flag for enabling the writing to the MSTOP7_ON bit (bit 7)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 22 | MSTOP6_ON_WEN | 0b | R0W1 | Flag for enabling the writing to the MSTOP6_ON bit (bit 6)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 21 | MSTOP5_ON_WEN | 0b | R0W1 | Flag for enabling the writing to the MSTOP5_ON bit (bit 5)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 20 to 16 | — | All 0 | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 15 to 11 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 10 | MSTOP10_ON | 0b | RW | The state of BUS_PERI_VIDEO_MSTOP10_MXVIDEO operation is indicated.<br>0: Normal operation<br>1: Module stop state (MSTP) |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 9 | MSTOP9_ON | 0b | RW | The state of BUS_PERI_VIDEO_MSTOP9_MPDU operation is indicated.<br>0: Normal operation<br>1: Module stop state (MSTP) |
| 8 | MSTOP8_ON | 0b | RW | The state of BUS_PERI_VIDEO_MSTOP8_MPFCPVD operation is indicated.<br>0: Normal operation<br>1: Module stop state (MSTP) |
| 7 | MSTOP7_ON | 0b | RW | The state of BUS_PERI_VIDEO_MSTOP7_MPVSPD operation is indicated.<br>0: Normal operation<br>1: Module stop state (MSTP) |
| 6 | MSTOP6_ON | 0b | RW | The state of BUS_PERI_VIDEO_MSTOP6_MPDSIL operation is indicated.<br>0: Normal operation<br>1: Module stop state (MSTP) |
| 5 | MSTOP5_ON | 0b | RW | The state of BUS_PERI_VIDEO_MSTOP5_MPDSIPHY operation is indicated.<br>0: Normal operation<br>1: Module stop state (MSTP) |
| 4 to 0 | — | All 0 | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |

### 7.2.4.140 MSTOP Register REG0 (CPG_BUS_REG0_MSTOP)

This register indicates the stop state of individual modules.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | MSTOP6_ON_WEN | MSTOP5_ON_WEN | MSTOP4_ON_WEN | — | MSTOP2_ON_WEN | — | — |
| 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 | R0W1 | R0W1 | R0W1 | R | R0W1 | R | R |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | MSTOP6_ON | MSTOP5_ON | MSTOP4_ON | — | MSTOP2_ON | — | — |
| 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 | RW | RW | RW | R | RW | R | R |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 23 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 22 | MSTOP6_ON_WEN | 0b | R0W1 | Flag for enabling the writing to the MSTOP6_ON bit (bit 6)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 21 | MSTOP5_ON_WEN | 0b | R0W1 | Flag for enabling the writing to the MSTOP5_ON bit (bit 5)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 20 | MSTOP4_ON_WEN | 0b | R0W1 | Flag for enabling the writing to the MSTOP4_ON bit (bit 4)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 19 | — | 0b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value.<br>Operation is not guaranteed if a value other than the initial value is written. |
| 18 | MSTOP2_ON_WEN | 0b | R0W1 | Flag for enabling the writing to the MSTOP2_ON bit (bit 2)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 17 | — | 0b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value.<br>Operation is not guaranteed if a value other than the initial value is written. |
| 16 | — | 0b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value.<br>Operation is not guaranteed if a value other than the initial value is written. |
| 15 to 7 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 6 | MSTOP6_ON | 0b | RW | The state of BUS_REG0_MSTOP6_MPOSTM_2 operation is indicated.<br>0: Normal operation<br>1: Module stop state (MSTP) |
| 5 | MSTOP5_ON | 0b | RW | The state of BUS_REG0_MSTOP5_MPOSTM_1 operation is indicated.<br>0: Normal operation<br>1: Module stop state (MSTP) |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 4 | MSTOP4_ON | 0b | RW | The state of BUS_REG0_MSTOP4_MPOSTM_0 operation is indicated.<br>0: Normal operation<br>1: Module stop state (MSTP) |
| 3 | — | 0b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value.<br>Operation is not guaranteed if a value other than the initial value is written. |
| 2 | MSTOP2_ON | 0b | RW | The state of BUS_REG0_MSTOP2_MPWDT_0 operation is indicated.<br>0: Normal operation<br>1: Module stop state (MSTP) |
| 1 | — | 0b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value.<br>Operation is not guaranteed if a value other than the initial value is written. |
| 0 | — | 0b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value.<br>Operation is not guaranteed if a value other than the initial value is written. |

### 7.2.4.141 MSTOP Register REG1 (CPG_BUS_REG1_MSTOP)

This register indicates the stop state of individual modules.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | MSTOP7_ON_WEN | — | — | — | MSTOP3_ON_WEN | MSTOP2_ON_WEN | MSTOP1_ON_WEN | MSTOP0_ON_WEN |
| 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 | R0W1 | R | R | R | R0W1 | R0W1 | R0W1 | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | MSTOP7_ON | — | — | — | MSTOP3_ON | MSTOP2_ON | MSTOP1_ON | MSTOP0_ON |
| 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 | RW | R | R | RW | RW | RW | RW | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 24 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 23 | MSTOP7_ON_WEN | 0b | R0W1 | Flag for enabling the writing to the MSTOP7_ON bit (bit 7)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 22 to 20 | — | 000b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value.<br>Operation is not guaranteed if a value other than the initial value is written. |
| 19 | MSTOP3_ON_WEN | 0b | R0W1 | Flag for enabling the writing to the MSTOP3_ON bit (bit 3)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 18 | MSTOP2_ON_WEN | 0b | R0W1 | Flag for enabling the writing to the MSTOP2_ON bit (bit 2)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 17 | MSTOP1_ON_WEN | 0b | R0W1 | Flag for enabling the writing to the MSTOP1_ON bit (bit 1)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 16 | MSTOP0_ON_WEN | 0b | R0W1 | Flag for enabling the writing to the MSTOP0_ON bit (bit 0)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 8 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 7 | MSTOP7_ON | 0b | RW | The state of BUS_REG1_MSTOP7_MXGIC operation is indicated.<br>0: Normal operation<br>1: Module stop state (MSTP) |
| 6 to 4 | — | 000b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value.<br>Operation is not guaranteed if a value other than the initial value is written. |
| 3 | MSTOP3_ON | 0b | RW | The state of BUS_REG1_MSTOP3_MPDMAC_NS operation is indicated.<br>0: Normal operation<br>1: Module stop state (MSTP) |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 2 | MSTOP2_ON | 0b | RW | The state of BUS_REG1_MSTOP2_MXDMAC_NS operation is indicated.<br>0: Normal operation<br>1: Module stop state (MSTP) |
| 1 | MSTOP1_ON | 0b | RW | The state of BUS_REG1_MSTOP1_MPDMAC_S operation is indicated.<br>0: Normal operation<br>1: Module stop state (MSTP) |
| 0 | MSTOP0_ON | 0b | RW | The state of BUS_REG1_MSTOP0_MXDMAC_S operation is indicated.<br>0: Normal operation<br>1: Module stop state (MSTP) |

### 7.2.4.142 Other Function Register 1 (CPG_OTHERFUNC1_REG)

This register is used for miscellaneous functions.

Bit 0 is used to switch the source clock for the system bus and the IPs that can use SSCG clocks

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RES15_ON_WEN | RES14_ON_WEN | RES13_ON_WEN | RES12_ON_WEN | RES11_ON_WEN | RES10_ON_WEN | RES9_ON_WEN | RES8_ON_WEN | RES7_ON_WEN | RES6_ON_WEN | RES5_ON_WEN | RES4_ON_WEN | RES3_ON_WEN | RES2_ON_WEN | RES1_ON_WEN | RES0_ON_WEN |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R0W1 | R0W1 | R0W1 | R0W1 | R0W1 | R0W1 | R0W1 | R0W1 | R0W1 | R0W1 | R0W1 | R0W1 | R0W1 | R0W1 | R0W1 | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RES15_SET | RES14_SET | RES13_SET | RES12_SET | RES11_SET | RES10_SET | RES9_SET | RES8_SET | RES7_SET | RES6_SET | RES5_SET | RES4_SET | RES3_SET | RES2_SET | RES1_SET | RES0_SET |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 17 | RESn_ON_WEN<br>[n=15 to 1] | All 0 | R0W1 | Flags for enabling the writing to the RESn_SET bits. These bits are always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 16 | RES0_ON_WEN | B’0 | R0W1 | Flag for enabling the writing to the RES0_SET bit (bit 0) (SEL_PLL5_4).<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 1 | RESn_SET<br>[n = 15 to 1] | All 0 | RW | No function is assigned to these bits.<br>When the flags for enabling the writing are set to 1, these bits can be written to.<br>The written value is read from these bits. |
| 0 | RES0_SET | B’0 | RW | Selection of the clock to be supplied to SEL_PLL5_4<br>0: Up to 3,000MHz<br>1: 1500MHz |

### 7.2.4.143 Other Function Register 2 (CPG_OTHERFUNC2_REG)

This register is used for miscellaneous functions.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RES15_ON_WEN | RES14_ON_WEN | RES13_ON_WEN | RES12_ON_WEN | RES11_ON_WEN | RES10_ON_WEN | RES9_ON_WEN | RES8_ON_WEN | RES7_ON_WEN | RES6_ON_WEN | RES5_ON_WEN | RES4_ON_WEN | RES3_ON_WEN | RES2_ON_WEN | RES1_ON_WEN | RES0_ON_WEN |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R0W1 | R0W1 | R0W1 | R0W1 | R0W1 | R0W1 | R0W1 | R0W1 | R0W1 | R0W1 | R0W1 | R0W1 | R0W1 | R0W1 | R0W1 | R0W1 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RES15_SET | RES14_SET | RES13_SET | RES12_SET | RES11_SET | RES10_SET | RES9_SET | RES8_SET | RES7_SET | RES6_SET | RES5_SET | RES4_SET | RES3_SET | RES2_SET | RES1_SET | RES0_SET |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 17 | RESn_ON_WEN [n=15 to 1] | All 0 | R0W1 | Flags for enabling the writing to the RESn_SET bits. These bits are always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 16 | RES0_ON_WEN | 0b | R0W1 | Flag for enabling the writing to the RES0_SET bit (bit 0) (DDR_RST_N).<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 15 to 1 | RESn_SET [n = 15 to 1] | All 0 | RW | No function is assigned to these bits.<br>When the flags for enabling the writing are set to 1, these bits can be written to.<br>The written value is read from these bits. |
| 0 | RES0_SET | 0b | RW | The DDR_RST_N reset terminal is controlled.<br>0: Reset signal is applied (reset state).<br>1: Reset signal is stopped (released from the reset state). |

**Note:** The change to the setting in this register is immediately reflected in the reset signal. Therefore, no register is provided to monitor the result of the change.

### 7.2.4.144 Clock Control Register MBUS (CPG_CLKON_MBUS)

This register is used to supply or stop clocks for individual modules.

Use this register only when switching SEL_PLL3_3. Reserved bit is prohibited from setting anything other than the initial value.The clocks listed in this register are not listed in the separate clock list.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | CLK9_ON_WEN | — | — | — | — | — | — | — | — | — |
| 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 | R0W1 | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | CLK9_ON | — | — | — | — | — | — | — | — | — |
| Initial Value | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| R/W | R | R | R | R | R | R | RW | R | R | R | R | R | R | R | R | R |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 26 | — | All 0 | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 25 | CLK9_ONWEN | 0b | R0W1 | Flag for enabling the writing to the CLK9_ON bit (bit 9)<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 24 to 12 | — | All 0 | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 11,10 | — | 11b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 9 | CLK9_ON | 1b | RW | The MBUS_CLK9 clock is supplied or stopped.<br>0: Clock is stopped.<br>1: Clock is supplied. |
| 8 to 0 | — | All 1 | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |

### 7.2.4.145 Clock Monitor Register MBUS (CPG_CLKMON_MBUS)

This register is used to monitor the state of the clocks supplied to individual modules.

The clocks listed in this register are not listed in the separate clock list.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | CLK9_MON | — | — | — | — | — | — | — | — | — |
| Initial Value | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 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 12 | — | All 0 | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 11,10 | — | 11b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 9 | CLK9_MON | 1b | R | The state of the MBUS_CLK9 clock is monitored.<br>0: Clock is stopped.<br>1: Clock is supplied. |
| 8 to 0 | — | FF7h | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |

## 7.3 Functions of the CPG

This section describes the functions of the CPG.

### 7.3.1 Clock Control

The CPG has the following five types of clock control functions.

- PLL control, setting, and monitoring functions
- Clock frequency dividers and selectors
- Fixed frequency dividers
- Clock enabling function

#### 7.3.1.1 PLL Control, Setting, and Monitoring Functions

This section describes the functions for controlling, setting, and monitoring the PLL (PLL1 to PLL5).

The PLL is controlled through the registers in the CPG.

The PLL control module counts the PLL lock wait time to generate the PLL lock signal and it is used to control the PLL output clocks by the CGC so that invalid clocks are not output before the PLL is locked.

The reset state and lock state of the PLL can be monitored through registers in the CPG.

The following shows the PLL control, setting, and monitoring functions.

- PLL operating mode control
- Output clock setting
- PLL reset and lock state monitoring

These functions are described in detail in the following sections.

**(1) PLL Operating Mode Control**

The CPG registers are used to specify the SSCG PLL operating mode and make SSCG settings.

The normal mode or standby mode can be selected as the operating mode.

The following can be set up through CPG registers.

- Operating mode setting (normal mode or standby mode)
- SSCG enable or disable setting (valid in the normal mode)

The following table shows the registers to be used.

For details of the settings, refer to the functional description of the target register.

Table 7.6 Registers for Controlling the PLL Operating Mode

| Register Name | Abbreviation | Function |
|---|---|---|
| PLLn (SSCG) standby control registers | CPG_SAMPLLn_STBY (n = 1 or 4)<br>CPG_SIPLL5_STBY | Standby mode setting (reset state) and SSCG enable or disable setting |

### (2) Output Clock Setting

The CPG registers are used to specify the clocks output from the PLL.

The following values can be set up.

- Output clock frequency setting
- SSCG modulation value setting

Table 7.7 Registers for Setting PLL Output Clocks

| Register Name | Abbreviation | Function |
|---|---|---|
| PLLn (SSCG) output clock setting register 1 | CPG_SAMPLLn_CLK1 | Frequency setting |
| PLLn (SSCG) output clock setting register 2 | CPG_SAMPLLn_CLK2 | Frequency setting and SSCG modulation value setting |

**Note:** n = 1 or 4

Table 7.8 Registers for Setting PLL Output Clocks 2

| Register Name | Abbreviation | Function |
|---|---|---|
| PLLn (SSCG) output clock setting register 1 | CPG_SIPLL5_CLK1 | Frequency setting |
| PLLn (SSCG) output clock setting register 3 | CPG_SIPLL5_CLK3 | Frequency setting and SSCG modulation value setting |
| PLLn (SSCG) output clock setting register 4 | CPG_SIPLL5_CLK4 | Frequency setting |
| PLLn (SSCG) output clock setting register 5 | CPG_SIPLL5_CLK5 | SSCG modulation value setting |

### (3) PLL State Monitoring

The reset state and lock state of the PLL can be monitored through CPG registers.

The following table shows the registers to be used.

Table 7.9 Registers for Monitoring the PLL State

| Register Name | Abbreviation | Function |
|---|---|---|
| PLLn monitor registers | CPG_SAMPLLn_MON (n = 1 or 4)<br>CPG_SIPLLn_MON (n = 2, 3, or 5) | PLL state monitoring |

#### (4) SSCG Switching in PLL3

In PLL3, the SSCG function is enabled or disabled through the MD_CLKS signal from the SYSC.

The state of the MD_CLKS signal is determined when the LSI is released from the system reset (PRST#) state and the signal state does not change after the system is activated.

The SSCG enabled or disable state in PLL3 cannot be switched through register settings.

MD_CLKS = 0: SSCG is disabled.

MD_CLKS = 1: SSCG is enabled.

### 7.3.1.2 Clock Frequency Dividers and Selectors

The CPG uses several frequency dividers and selectors to provide appropriate clocks that meet the specifications of individual modules. The frequency dividers and selectors are classified into the dynamic switching type that can be switched without generating a glitch and the static switching type that should be switched after the clock for the target module is stopped because a glitch is generated if this type of divider or selector is switched while the clock is supplied.

The following frequency dividers and selectors are provided.

- Fixed frequency dividers
- Dynamic switching selectors
- Dynamic switching variable-frequency dividers
  (with a duty ratio of 50:50 when divided by an even number)
- Static switching selectors
- Static switching variable-frequency dividers
  (with a duty ratio of 50:50 when divided by an even number)

A specific procedure should be used to switch each frequency divider or selector. Refer to **Section 7.4.6, Procedure for Switching the Division Ratio of the Dynamic Switching Frequency Dividers**, **Section 7.4.7, Procedure for Switching Clocks by the Dynamic Switching Frequency Selectors**, and **Section 7.4.8, Procedure for Switching Clocks by the Static Switching Frequency Dividers and Selectors (Procedure for Switching Source Clocks)**.

### 7.3.1.3 Fixed Frequency Dividers

A fixed frequency divider uses a fixed division value. The CPG provides 1/2, 1/4, 1/8, 1/10, 1/16, and 1/32 dividers, and all fixed dividers generate clocks with a duty ratio of 50:50.

#### 7.3.1.4 PLL Clock Monitoring

This function monitors whether the PLL1 to PLL5 output clocks are operating.

The frequency in the CPG is divided to obtain the frequency that can be monitored through the external pin of this LSI and then output to the upper layer.

The following table shows the instance names and clock frequencies of the monitor points.

Table 7.10 Instance Names and Clock Frequencies of the Monitor Points

| PLL Name | Instance Name of Monitor Point | Maximum Frequency (MHz) | Divider Value | Output Clock Frequency (MHz) |
|---|---|---|---|---|
| PLL1 | CPG_CTL/PLL1/FOUT | 1200 | 32 | 38 |
| PLL2 | CPG_CTL/PLL2/FOUT1PH0 | 1600 (1/2 of 3200) | 32 | 50 |
| PLL2 | CPG_CTL/PLL2/FOUT3 | 533 (1/6 of 3200) | 16 | 34 |
| PLL3 | CPG_CTL/PLL3/FOUT1PH0 | 1600 (1/2 of 3200) | 32 | 50 |
| PLL3 | CPG_CTL/PLL3/FOUT3 | 533 (1/6 of 3200) | 16 | 34 |
| PLL3 | CPG_CTL/PLL3/FOUT4 | 400 (1/8 of 3200) | 8 | 50 |
| PLL4 | CPG_CTL/PLL4/FOUT | 1600 | 32 | 50 |
| PLL5 | CPG_CTL/PLL5/FOUTPOSTDIV | 3000 | 64 | 47 |
| PLL5 | CPG_CTL/PLL5/FOUT3 | 500 (1/6 of 3000) | 16 | 32 |
| PLL5 | CPG_CTL/PLL5/FOUT1PH0 | 1500 (1/2 of 3000) | 32 | 47 |

### 7.3.2 Clock Generation and Control Functions

The following types of reset are provided.

Table 7.11 Reset Types

| No. | Type | Method | Source | Range of Reset |
|---|---|---|---|---|
| 1 | System reset | Hardware reset | Reset by PRST# | All |
| 2 | | WDT system reset | WDT overflow, etc. | All<br>(The WDTOVF register is excluded.) |
| 3 | Module reset | WDT system reset | WDT overflow, etc. | CPU corresponding to each WDT |
| 4 | | Software reset | Control through CPG registers | individual on-chip modules |

#### 7.3.2.1 Range of Reset Application

The following shows the application (target) range of the reset generated by each reset source.

(1) Hardware sources (PRST#, TRST#, and DEBUGEN)

(2) WDT system reset sources

Table 7.12 Range of Reset Application (1/2)

| No. | Operating Mode DEBUGEN | Reset Source: Debug Reset TRST# | Reset Source: System Reset PRST# | Reset Source: WDT Reset | JTAG Interface (CST) | Range of Reset Application: Modules Related to Debugging: Cortex-A55 Debug Reset (nPDGRESET) | Range of Reset Application: Modules Related to Debugging: Coresight Reset | Range of Reset Application: WDT: Reset for WDTOVF Register | Range of Reset Application: All Modules*1 | Description of Reset |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | Reset | Reset | Reset | Reset | Reset | User mode,<br>Hardware system reset |
| 2 | 0 | 0 | 0 | 1 | Reset | Reset | Reset | Reset | Reset | User mode,<br>Hardware system reset |
| 3 | 0 | 0 | 1 | 0 | Reset | Reset | Reset | Released | Reset | User mode,<br>WDT system reset |
| 4 | 0 | 0 | 1 | 1 | Reset | Released | Reset | Released | Released | User mode,<br>Normal operation state |
| 5 | 0 | 1 | 0 | 0 | Setting prohibited | — | — | — | — | — |
| 6 | 0 | 1 | 0 | 1 | Setting prohibited | — | — | — | — | — |
| 7 | 0 | 1 | 1 | 0 | Setting prohibited | — | — | — | — | — |
| 8 | 0 | 1 | 1 | 1 | Setting prohibited | — | — | — | — | — |

Table 7.12 Range of Reset Application (2/2)

| No. | Operating Mode DEBUGEN | Reset Source: Debug Reset TRST# | Reset Source: System Reset PRST# | Reset Source: WDT Reset | JTAG Interface | Range of Reset Application: Modules Related to Debugging: Cortex-A55 Debug Reset (nPDGRESET) | Range of Reset Application: Modules Related to Debugging: Coresight Reset | Range of Reset Application: WDT: Reset for WDTOVF Register | Range of Reset Application: All Modules*1 | Description of Reset |
|---|---|---|---|---|---|---|---|---|---|---|
| 9 | 1 | 0 | 0 | 0 | Reset | Reset | Reset | Reset | Reset | Debug mode, Hardware system reset |
| 10 | 1 | 0 | 0 | 1 | Reset | Reset | Reset | Reset | Reset | Debug mode, Hardware system reset |
| 11 | 1 | 0 | 1 | 0 | Reset*2 | Reset*3 | Reset*2 | Released | Reset*2 | Debug mode, WDT system reset |
| 12 | 1 | 0 | 1 | 1 | Reset | Released | Released | Released | Released | Debug mode, Normal operation state |
| 13 | 1 | 1 | 0 | 0 | Reset | Reset | Reset | Released | Reset | Debug mode, Hardware system reset |
| 14 | 1 | 1 | 0 | 1 | Reset | Reset | Reset | Reset | Reset | Debug mode, Hardware system reset |
| 15 | 1 | 1 | 1 | 0 | Reset*2 | Reset*3 | Reset*2 | Released | Reset*2 | Debug mode, WDT system reset |
| 16 | 1 | 1 | 1 | 1 | Released | Released | Released | Released | Released | Debug mode, Normal operation state |

Note 1. Reset when the system reset is enabled by the WDTRSTSEL register.

Note 2. Reset when the Cortex-A55 reset is enabled by the WDTRSTSEL register.

### 7.3.2.2 System Reset (External Pin)

A system reset is applied when the reset signal from the external PRST# pin is asserted and it initializes the entire LSI.

### 7.3.2.3 WDT System Reset

This LSI has three WDT channels and the CPG issues a reset request when any of the following sources is generated.

| WDT Channel | Target of Monitoring | Function |
|---|---|---|
| Channel 0 | Cortex-A55 Core 0 | Reset due to a parity error in Cortex-A55 Core 0 |

### (1) WDT System Reset Circuit

The reset requests from three WDT channels can be assigned to the desired signals shown below by using the masking function specified through the settings of the WDTRSTSEL register implemented in the CPG.

If a bit in the WDTRSTSEL register is set to 1, the corresponding signal is asserted when the WDT issues a reset request.

| Target of Monitoring | Function |
|---|---|
| WDT system reset | The reset request signal WDTRST# is sent to the reset generation circuit to apply a system reset. |
| WDTOVF_PERROUT | The WDTOVF_PERR signal is asserted to notify the external devices of the application of a system reset. |
| Cortex-A55 cold reset | A cold reset request is issued when either WDT channel 0 detects an error. |

| Bit | Bit Name | Initial Value | R/W | Reset Source Module | Function |
|---|---|---|---|---|---|
| 0 | WDTRSTSEL0 | 0 | R/W | WDT (channel 0)<br>Cortex-A55 Core 0 | When the corresponding WDT issues a reset request due to an overflow, etc.<br>0: WDT system reset is masked.<br>1: WDT system reset is applied. |
| 1 | Reserved | 0 | R | — | — |
| 2 | Reserved | 0 | R | — | — |
| 3 | Reserved | 1 | R | — | — |
| 4 | WDTRSTSEL4 | 0 | R/W | WDT (channel 0)<br>Cortex-A55 Core 0 | When the corresponding WDT issues a reset request due to an overflow, etc.<br>0: Assertion of WDTOVFOUT is masked.<br>1: WDTOVFOUT is asserted. |
| 5 | Reserved | 0 | R | — | — |
| 6 | Reserved | 0 | R | — | — |
| 7 | Reserved | 1 | R | — | — |
| 8 | WDTRSTSEL8 | 0 | R/W | WDT (channel 0)<br>Cortex-A55 Core 0 | When either WDT channel 0 issues a reset request*1,*2<br>0: Cold reset of the Cortex-A55 is masked.<br>1: Cold reset of the Cortex-A55 is applied. |
| 9 | Reserved | 0 | R | — | — |
| 10 | Reserved | 0 | R | — | — |

Note 1. When an error occurs in Core 0 in the Cortex-A55, a cold reset is applied to the entire Cortex-A55.

Note 2. For the timing of the cold reset in the Cortex-A55, refer to **Section 7.3.2.3(4), Timing of Cortex-A55 Cold Reset due to the WDT Source**.

### (2) WDTOVF_PERROUT Signal

This is an external signal that notifies the system that the WDT has applied a WDT system reset to this LSI.

If another WDT system reset is requested before the output of this signal pulse is completed, it is ignored.

The GPIO settings of the WDTOVF_PERROUT# pin such as the driving capability are reset by the GPIO_RST_WDTOVFN signal for the CPG_WDTOVFRST[3:0] register bits.

The following shows the sequence of reset by the WDT.

(1) The WDT channel selected by the WDTRSTSEL[7:4] bits generates a reset request.

(2) The CPG applies a WDT system reset.

(3) 64 cycles of the 24-MHz clock are waited.
* This is the period until the necessary settings are reflected in the entire LSI.

(4) A low pulse is output from the WDTOVF_PERROUT pin for 64 cycles of the 24-MHz clock.

(5) 64 cycles of the 24-MHz clock are waited.

(6) The WDT system reset signal is deasserted and the boot sequence is performed.

### (3) WDT Overflow System Reset Register (CPG_WDTOVF_RST)

After the release from the WDT system reset applied by the WDT system reset circuit in response to a reset request such as a WDT overflow, this register is used to identify the WDT channel that generated the reset request.

This register is not reset by the WDT system reset generated by the WDT.

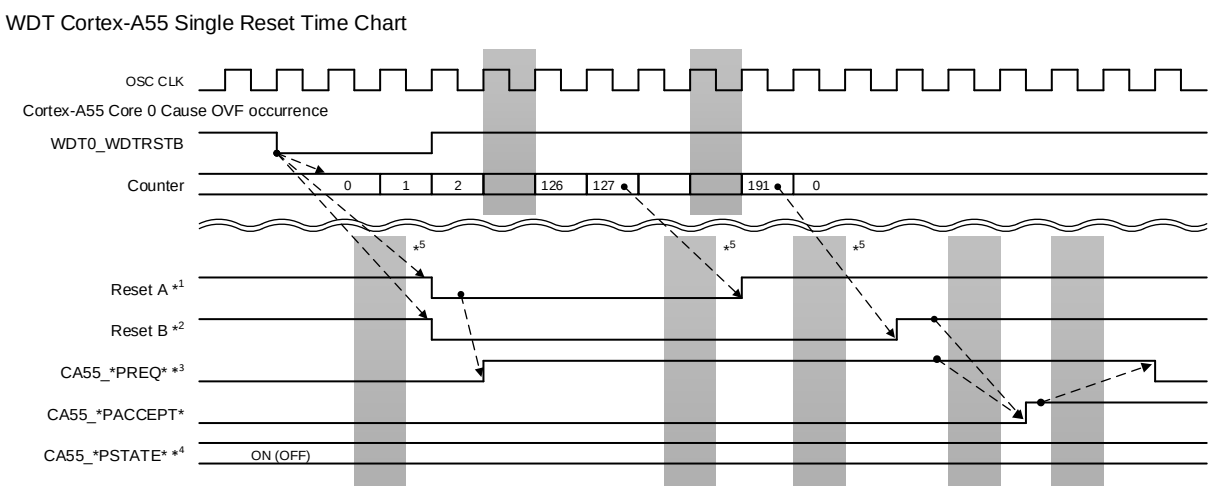
If a WDT system reset is not applied according to the WDTRSTSEL register setting, the WDT is not reset and the source of the reset request can be identified by checking the interrupt from the WDT and the status register in the WDT.

Table 7.13 WDT Overflow System Reset Register (CPG_WDTOVF_RST)

| Bit | Bit Name | Initial Value | R/W | Reset Source Module | Function |
|---|---|---|---|---|---|
| 0 | WDTOVF0 | 0 | R/W | WDT (channel 0)<br>Cortex-A55 Core 0 | When a WDT channel has applied a WDT system reset, the corresponding bit is set to 1.<br>These bits are cleared by writing 1. |
| 1 | Reserved | 0 | R | — | — |
| 2 | Reserved | 0 | R | — | — |
| 15 to 3 | Reserved | 0 | R | — | — |
| 16 | WDTOVF0_WEN | 0 | R0W1 | — | Writing to the WDTOVF0 bit (bit 0) is masked.<br>This bit is always read as 0.<br>0: Writing is disabled.<br>1: Writing is enabled. |
| 17 | Reserved | 0 | R | — | — |
| 18 | Reserved | 0 | R | — | — |
| 31 to 19 | Reserved | 0 | R | — | — |

### (4) Timing of Cortex-A55 Cold Reset due to the WDT Source

The following figure shows the timing of the cold reset for the Cortex-A55 alone due to the WDT source.




Note 1. CA55_RST10/CA55_RST11/CA55_RST12

Note 2. CA55_RST1[0]/CA55_RST3[0]/CA55_RST4/CA55_RST5/CA55_RST6/CA55_RST7/CA55_RST8/CA55_RST9

Note 3. The SYSREG setting must be set to ‘0’.

Note 4. The SYSREG setting is ‘ON’ for Core 0/Cluster.

Note 5. Cortex-A55 reset generation has a certain time lag (approximately 35CLK@24MHz).

Figure 7.4 Timing of Cold Reset for the Cortex-A55 Alone due to the WDT Source

#### 7.3.2.4 Module Reset Application Control

The application of a reset is controlled for individual modules.

In addition to a reset applied to individual modules according to register settings by software, some modules are reset in a certain reset mode.

For register setting specifications, refer to the descriptions of the reset control register *** (CPG_RST_***).

For a clock that is controlled by the corresponding CGC and is synchronized with a reset signal, the processing for stopping the clock is performed at the release of the reset state.

## 7.4 Operating Procedures

This section describes the procedures for operating the CPG.

### 7.4.1 Procedures for Supplying and Stopping Module Clocks

Use the following procedures to switch between the supply and stop of the target clock for a module.

* Sample procedures for the SRAM_MCPU clock (SRAM_MCPU_ACLK) are described here.

**Supplying a Clock:**

1. Setting the register for supplying the SRAM_MCPU clock (SRAM_MCPU_ACLK)

   Set up the following bits in the clock control register SRAM_MCPU (CPG_CLKON_SRAM_MCPU).
   — Setting for supplying the clock: Bit 0 (CLK0_ON) = 1 and bit 16 (CLK0_ONWEN) = 1

2. Confirming that the output of the SRAM_MCPU clock (SRAM_MCPU_ACLK) has started

   Read the following bit in the clock monitor register SRAM_MCPU (CPG_CLKMON_SRAM_MCPU) to confirm that the output of the clock has started.
   — Clock state: Bit 0 (UNIT0_CLK_MON), 1: Clock is supplied. (0: Clock is stopped.)

**Stopping a Clock:**

1. Setting the register for stopping the SRAM_MCPU clock (SRAM_MCPU_ACLK)

   Set up the following bits in the clock control register SRAM_MCPU (CPG_CLKON_SRAM_MCPU).
   — Setting for stopping the clock: Bit 0 (CLK0_ON) = 0 and bit 16 (CLK0_ONWEN) = 1

2. Confirming that the output of the SRAM_MCPU clock (SRAM_MCPU_ACLK) has been stopped

   Read the following bit in the clock monitor register SRAM_MCPU (CPG_CLKMON_SRAM_MCPU) to confirm that the output of the clock has been stopped.
   — Clock state: Bit 0 (UNIT0_CLK_MON), 0: Clock is stopped. (1: Clock is supplied.)

The following registers are used in the above procedures.

— 7.2.4.x Clock control registers ***** (CPG_CLKON_*****)

— 7.2.4.x Clock monitor registers ***** (CPG_CLKMON_*****)

For the procedures for supplying and stopping a clock signal, refer also to the section on the target module. If specific procedures for the module are described, use them.

### 7.4.2 Procedures for Supplying and Stopping Reset Signals

Use the following procedures to switch between the supply (reset state) and stop (released from the reset state) of the target reset signal for a module.

* Sample procedures for the SRAM_MCPU reset (SRAM_MCPU_ARESETN) are described here.

**Supplying a Reset Signal:**

1. Setting the register for supplying the SRAM_MCPU reset signal (SRAM_MCPU_ARESETN)

   Set up the following bits in the reset control register SRAM_MCPU (CPG_RST_SRAM_MCPU).
   — Setting for supplying the reset signal: Bit 0 (UNIT0_RSTB) = 0 and bit 16 (UNIT0_RST WEN) = 1

2. Confirming that the output of the SRAM_MCPU reset signal (SRAM_MCPU_ARESETN0) has been started

   Read the following bit in the reset monitor register SRAM_MCPU (CPG_RSTMON_SRAM_MCPU) to confirm that the output of the reset signal has been started.
   — Reset state: Bit 0 (UNIT0_RST_MON)
   0: Reset signal is stopped (released from the reset state)
   1: Reset signal is output.

**Stopping a Reset Signal:**

1. Setting the register for stopping the SRAM_MCPU reset signal (SRAM_MCPU_ARESETN)

   Set up the following bits in the reset control register SRAM_MCPU (CPG_RST_SRAM_MCPU).

   - Setting for stopping the reset signal: Bit 0 (UNIT0_RSTB) = 1 and bit 16 (UNIT0_RST WEN) = 1

2. Confirming that the output of the SRAM_MCPU reset signal (SRAM_MCPU_ARESETN) has been stopped

   Read the following bit in the reset monitor register SRAM_MCPU (CPG_RSTMON_SRAM_MCPU) to confirm that the output of the reset signal has been stopped.
   — Reset state: Bit 0 (UNIT0_RST_MON)
   0: Reset signal is stopped (released from the reset state)
   1: Reset signal is output.

The following registers are used in the above procedures.

— 7.2.4.x Reset control registers ***** (CPG_RST_*****)

— 7.2.4.x Reset monitor registers ***** (CPG_RSTMON_*****)

For the procedures for supplying and stopping a reset signal, refer also to the section on the target module. If specific procedures for the module are described, use them.

### 7.4.3 Procedure for Activating Modules

Use the following procedure to activate an inactive module after a system reset.

For the specific procedure for activating the target module, refer to the section on the module.

1. Set up the clock control register for the clock signal connected to the target module to start the supply of the clock. Note that the PLL for the clock should be started before the clock if the PLL is stopped.
2. Read the clock monitor register to confirm that the supply of the target clock has started.
3. Set up the reset control register for the reset signal connected to the target module to release the module from the reset state.
4. Read the reset monitor register to conform that the module has been released from the reset state.

### 7.4.4 Procedure for Activating the Modules Related to PLL5

The synchronizer is intentionally removed from some PLL5 clock paths. For the following modules that use PLL5 clocks, use the activation flow and software control procedures shown in **Figure 7.5** and described later to avoid malfunctions due to hazards.

Applicable module

- DSI
- LCDC

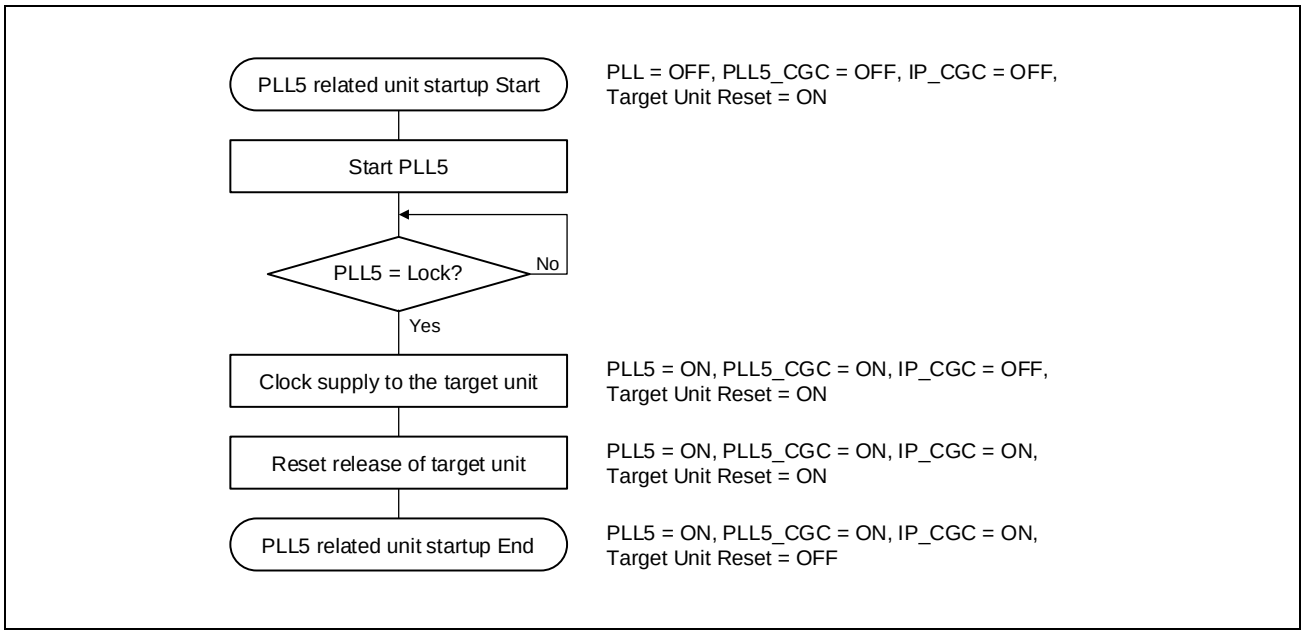



Figure 7.5 Flowchart of Activation of the Modules Related to PLL5

**[Activation Flow]**

<Initial state>

PLL5 = stopped, PLL5_CGC = activated, IP_CGC = activated,
and reset signals = applied (by hardware)

**1) From the activation of PLL5 to the wait for the lock of PLL5:**

PLL5 = activated (by software), PLL5_CGC = stopped, IP_CGC = stopped,
and reset signals for the target module = applied
In this period, the unstable clock before PLL5 is stopped by PLL5_CGC and does not propagate to the frequency divider or target module in the subsequent stages.

**2) From the lock of PLL5 to the supply of the clock to the frequency divider DIV_DSI_A/B:**

When PLL5 is locked, PLL5_CGC is automatically activated.

PLL5 = activated, PLL5_CGC = activated (by hardware), IP_CGC = stopped,
and reset signals for the applicable modules = applied

As soon as PLL5_CGC is activated, a clock hazard is momentarily present on the frequency divider but the incorrect output from the divider is stopped by the IP_CGC.
The frequency divider DIV_DSI_A will be stable in a short time.

The time until the divider becomes stable depends on the frequency setting in PLL5 but it is very short; there is no need to check whether it is stable by software.

**3) Supplying clocks to the target module**

Start the supply of the clocks to the target module by software.

PLL5 = activated, PLL5_CGC = activated, IP_CGC = activated (by software),
and reset signals for the target module = applied

Whether the IP_CGC is released from the reset state in synchronization with the clock differs between the PLLCLK_IN line and other lines.

For the PLLCLK_IN line, a synchronizer that synchronizes the maximum frequency (3 GHz) of PLLCLK_IN cannot be implemented by primitives. Therefore, the IP_CGC is released from the reset state asynchronously and a clock hazard will be present.

**4) Releasing the target module from the reset state**

Release the target module from the reset state by software.

PLL5 = activated, PLL5_CGC = activated, IP_CGC = activated,
and reset signals = stopped (by software)

Before and after the release from the reset state, the clocks with which the reset signals are synchronized are stopped in the target module for several tens of clock cycles to avoid contention or adjust the timing flexibly.

## 7.4.5 Procedures for PLL Setup

The following shows the procedures for PLL setup.

### 7.4.5.1 Procedure for Setting PLL Normal Mode (Changing the Output Clock and SSCG Mode)

Use the following procedure to shift the PLL from the standby mode to the normal mode (change the output clock and SSCG mode).

**[n = 1 or 4: PLL1 or PLL4]**

**1) Confirming the PLL operating mode (confirming that the PLL is in the standby mode)**

- Check the following bit in the PLLn (SSCG) monitor register (CPG_SAMPLLn_MON).
- Shift to the standby mode: Confirm that bit 0 (PLLn_RESETB) = 0 (standby mode).
- Check the following bit in the PLLn (SSCG) monitor register (CPG_SAMPLLn_MON).
- Unlock of PLL: Confirm that bit 4 (PLLn_LOCK) = 0 (PLL is not locked).

Wait until the above conditions are satisfied.

**2) Setting the output clock 1**

- Modify the following bits in PLLn (SSCG) output clock setting register 1 (CPG_SAMPLLn_CLK1).
- For details of the settings, refer to **Section 7.2.4, Register Descriptions**.
- Setting of output clock (desired setting): Bits 5 to 0 (DIV_P_5 to DIV_P_0), bits 15 to 6 (DIV_M_9 to DIV_M_0), and bits 31 to 16 (DIV_K_15 to DIV_K_0)

**3) Setting the output clock and SSCG modulation value 2**

- Modify the following bits in PLLn (SSCG) output clock setting register 2 (CPG_SAMPLLn_CLK2).
- For details of the settings, refer to **Section 7.2.4, Register Descriptions**.
- Setting of output clock (desired setting): Bits 2 to 0 (DIV_S_2 to DIV_S_0)
- Setting of SSCG modulation value (desired setting): Bits 13 to 8 (MRR_5 to MRR_0) and bits 23 to 16 (MFR_7 to MFR_0)

**4) Setting the PLL normal mode**

- Set up the following bits in the PLLn (SSCG) standby control register (CPG_SAMPLLn_STBY).
- Setting of operating mode: Bit 0 (RESETB) = 1 (normal mode)
- Setting of SSCG mode: Bit 2 (SSCG_EN) = Desired setting

**5) Confirming the shift to the PLL normal mode and the stable output of the clock**

- Check the following bit in the PLLn (SSCG) monitor register (CPG_SAMPLLn_MON).
- Shift to standby mode: Confirm that bit 0 (PLLn_RESETB) = 1 (normal mode).
- Check the following bit in the PLLn (SSCG) monitor register (CPG_SAMPLLn_MON).
- Lock of PLL: Confirm that bit 4 (PLLn_LOCK) = 1 (PLL is locked).

Wait until the above conditions are satisfied.

NOTE

If there is no change in a parameter in step (2) or (3), the corresponding register does not need to be modified.

**[PLL5]**

**1) PLL operation mode status check (confirmation that it is in standby mode)**

- Check the following bits of the PLL5 (SSCG) monitor register (CPG_SIPLL5_MON).
- Standby mode transition confirmation: bit0 (PLLn_RESETB) = 0 (standby mode).
- Check the following bits of the PLL5 (SSCG) monitor register (CPG_SIPLL5_MON).
- Confirmation of PLL lock release: bit4 (PLLn_LOCK) = 0 (PLL is not locked).

Wait until the above conditions are satisfied.

**2) Output clock setting 1**

- Change the following bits of PLL5 (SSCG) output clock setting register 1 (CPG_SIPLL5_CLK1).
- For details on the setting values, refer to **Section 7.2.4, Register Descriptions**.
- Output frequency setting (optional):bit2-0(POSTDIV1)、bit6-4(POSTDIV2)、bit13-8(REFDIV)

**3) Output clock setting, SSCG modulation value setting 3**

- Change the following bits of PLLn (SSCG) output clock setting register 3 (CPG_SIPLL5_CLK3).
- For details of the setting values, refer to **Section 7.2.4, Register Descriptions**.
- Output frequency setting (optional): bit31-8(FRACIN)
- SSCG modulation value setting (optional): bit5-0(DIVVAL)

**4) Output clock setting 4**

- Change the following bits of PLL5 (SSCG) output clock setting register 4 (CPG_SIPLL5_CLK4).
- For details of the setting values, refer to **Section 7.2.4, Register Descriptions**.
- Output frequency setting (optional): bit27-16(INTIN)

**5) SSCG modulation value setting 5**

- Change the following bits of PLL5 (SSCG) output clock setting register 5 (CPG_SIPLL5_CLK5).
- For details of the setting values, refer to **Section 7.2.4, Register Descriptions**.
- SSCG modulation value setting (optional): bit4-0(SPREAD)

**6) PLL normal mode setting**

- Set the following bits of the PLLn (SSCG) standby control register (CPG_SIPLL5_STBY).
- Operation mode setting: bit0 (RESETB) = 1 (normal mode)
- SSCG mode setting: bit2 (SSC_EN) = optional

**7) PLL normal mode transition, output clock stability check**

- Check the following bits of the PLL5 (SSCG) monitor register (CPG_SIPLL5_MON).

- Confirmation of transition to normal mode: bit0 (PLLn_RESETB) = 1 (normal mode).
- Check the following bits of the PLL5 (SSCG) monitor register (CPG_SIPLL5_MON).
- PLL Lock confirmation: Bit4 (PLLn_LOCK) = 1 (PLL Lock).

Wait until the above conditions are satisfied.

NOTE

If there is no change in a parameter in step (2), (3), (4) or (5), the corresponding register does not need to be modified.

### 7.4.5.2 Procedure for Setting PLL Standby Mode

Use the following procedure to shift the PLL from the normal standby mode to the standby mode.

**[n = 1 or 4: PLL1 or PLL4]**

**1) Setting the PLL standby mode**

- Set the following bits of the PLL5 (SSCG) standby control register (CPG_SAM / SIPLL5_STBY).
- Operation mode setting: bit0 (RESETB) = 0 (standby mode) (other bits are “0”).

**2) Confirming the shift to the PLL standby mode**

- Check the following bits of the PLLn (SSCG) monitor register (CPG_SAM / SIPLL5_MON).
- Standby mode transition confirmation: bit0 (PLLn_RESETB) = 0 (standby mode).
- Check the following bits of the PLLn (SSCG) monitor register (CPG_SAM / SIPLL5_MON).
- Confirmation of PLL lock release: bit4 (PLLn_LOCK) = 0 (PLL is not locked).

Wait until the above conditions are satisfied.

### 7.4.5.3 PLLs with Restrictions

PLL1 to PLL3 cannot be switched to desired clock settings by the user because these PLLs output clocks for the CPU and bus operation.

The PLL2 and PLL3 settings cannot be modified after booting because these PLLs output system clocks for bus operation.

PLL4 and PLL5 can be set up, started, or stopped by the user at the desired timing under the allowable conditions of use for the target module.

### 7.4.6 Procedure for Switching the Division Ratio of the Dynamic Switching Frequency Dividers

Use the following procedure to set the division ratio of the dynamic switching frequency dividers.

* A sample procedure for DIV_PLL2_A is described here.

#### 1) Confirming the DIV_PLL2_A state (checking that it is not busy)

- Check the following bit in the clock status monitor register (CPG_CLKSTATUS).
- DIV_PLL2_A state: Confirm that bit 0 (DIVPL2A_STS) = 0 (switching is completed).

Wait until the above condition is satisfied.

#### 2) Setting the DIV_PLL2_A division ratio

- Set up the following bits in the division ratio setting (PLL2) register (CPG_PL2_DDIV).
- Setting of division ratio: Bits 1 and 0 (DIVPL2A_SET) = desired value and bit 16 (DIV_PLL2_A_WEN) = 1

#### 3) Confirming the DIV_PLL2_A state (checking that it is not busy)

- Check the following bit in the clock status monitor register (CPG_CLKSTATUS).
- DIV_PLL2_A state: Confirm that bit 0 (DIVPL2A_STS) = 0 (switching is completed).

Wait until the above condition is satisfied.

The following registers are used in the above procedure.

— Division ratio setting (PLL1) register (CPG_PL1_DDIV)

— Division ratio setting (PLL2) register (CPG_PL2_DDIV)

— Division ratio setting (PLL3) register (CPG_PL3A_DDIV)

— Division ratio setting (PLL3) register (CPG_PL3B_DDIV)

The following status monitor register is used.

— Clock status monitor register (CPG_CLKSTATUS)

When 1 is written to a xx_WEN bit placed in the upper 16 bits of a division ratio setting register, the clock switching control begins even if the settings in the lower bits are not changed. Check the status monitor register (CPG_CLKSTATUS) and wait until the switching is completed. The clock will temporarily stop at the timing of switching.

### 7.4.7 Procedure for Switching Clocks by the Dynamic Switching Frequency Selectors

Use the following procedure to switch clocks by the dynamic switching frequency selectors.

* A sample procedure for SEL_PLL4 is described here.

#### 1) Confirming the SEL_PLL4 state (checking that it is not busy)

- Check the following bit in the clock status monitor register (CPG_CLKSTATUS).
- SEL_PLL4 state: Confirm that bit 31 (SELPLL4_STS) = 0 (switching is completed).

Wait until the above condition is satisfied.

#### 2) Setting the switching of the clock output from SEL_PLL4

- Set up the following bits in the source clock setting (DDR) register (CPG_PL4_DSEL).
- Setting of clocks switching (switch from PLL4 to OSC_DIV1000): Bits 31 to 0 = H’0001_0000
  (switch from OSC_DIV1000 to PLL4): Bits 31 to 0 = H’0001_0001

#### Confirming the SEL_PLL4 state (checking that it is not busy)

- Check the following bit in the clock status monitor register (CPG_CLKSTATUS).
- SEL_PLL4 state: Confirm that bit 31 (SELPLL4_STS) = 0 (switching is completed).

Wait until the above condition is satisfied.

The following registers are used in the above procedure.

— Source clock setting (DDR) register (CPG_PL4_DSEL)

— Source clock setting (SDHI) register (CPG_PL2SDHI_DSEL)

When 1 is written to a xx_WEN bit placed in the upper 16 bits of a source clock setting register, the clock switching control begins even if the settings in the lower bits are not changed. Check the status monitor register (CPG_CLKSTATUS) and wait until the switching is completed. The clock will temporarily stop at the timing of switching.

### 7.4.8 Procedure for Switching Clocks by the Static Switching Frequency Dividers and Selectors (Procedure for Switching Source Clocks)

Use the following procedure to set the division ratio of the static switching frequency dividers and to switch clocks by the static switching frequency selectors.

When a source clock is active: Stop the clock. → Make settings for switching clocks. → Start the clock.

When a source clock is stopped: Make settings for switching.

The following registers are used in the above procedure.

— Source clock setting register (CPG_PL3_SSEL)*1

— Division ratio setting (PLL5) register (CPG_PL5_SDIV)

— Bit 0 (RES0_SET) in other function register 1 (CPG_OTHERFUNC1_REG)

**Note 1.** When switching SEL_PLL3_3 in CPG_PL3_SSEL, please note the following points.

1) Do not access SPI-Multi during switching.
2) Before switching, stop the following clocks.
   - CPG_CLKON_SPI_MULTI (bit1, bit0)
   - CPG_CLKON_MBUS (bit11, bit9)
3) CPG_CLKON_SPI_MULTI and CPG_CLKON_MBUS are not in the order to stop the clock.

### 7.4.9 Boot Sequence (Normal Operation)

After the reset sequence ends, perform the following operations if necessary.

- Start PLLs.
- Set up clock selectors.
- Set up clock division ratios.
- Supply clocks to modules.
- Release modules from the reset state.

### 7.4.10 Boot Sequence (Debug Mode)

After the reset sequence (debug mode) ends, perform the following operations if necessary.

- Start PLLs.
- Set up clock selectors.
- Set up clock division ratios.
- Supply clocks to modules.
- Release modules from the reset state.

## 7.4.11 General Operating Procedures

The following describes general operating procedures.

### 7.4.11.1 Procedure for Writing to Registers

The registers in the CPG have write-enable flags in the upper 16 bits and normal data fields in the lower 16 bits except for the registers shown below. Therefore, only the desired bit can be modified without read-modify-write operation.

To modify the value of a bit, write 1 to the write-enable flag having the target bit number + 16 and write a desired value to the target bit.

(Examples: To modify bit 0 to 1b, write H'0001_0001. To modify it to 0b, write H'0001_0000. When H'0000_xxxx is written, the value of bits 15 to 0 does not change.)

NOTE

In some registers, a single write-enable flag controls multiple data bit. Refer to the description of each register.

Registers that have data fields in the upper bits instead of write-enable flags:

— PLLn (SSCG) output clock setting register 1 (CPG_SAMPLLn_CLK1): (n = 1 or 4)

— PLLn (SSCG) output clock setting register 2 (CPG_SAMPLLn_CLK2): (n = 1 or 4)

— PLLn (SSCG) output clock setting register 3 (CPG_SIPLLn_CLK3): (n = 5)

— PLLn (SSCG) output clock setting register 4 (CPG_SIPLLn_CLK4): (n = 5)

— PLLn (SSCG) output clock setting register 5 (CPG_SIPLLn_CLK5): (n = 5)

# 8. Interrupt Controller

The interrupt controller decides the priority of interrupt sources and controls interrupt requests to the CPU.

The interrupt controller registers set the order of priority of each interrupt, allowing the user to process interrupt requests according to the user-set priority.

## 8.1 Features

### ■ Equipped with Arm® CoreLink™ GIC-600 Generic Interrupt Controller*1 for Arm Cortex-A55.

- 32 levels of priority.
- SGI (Software Generated Interrupt): 16 interrupts
- PPI (Private Peripheral Interrupt): 16 interrupts
- SPI (Shared Peripheral Interrupt): 480 interrupts
- LPI (Locality-specific Peripheral Interrupt): Not supported

### ■ External Interrupts

- NMI, IRQ0-7, TINT0-31
- Noise filter function.*2

### ■ Internal Interrupts

- On-chip peripheral interrupts
- On-chip Bus error interrupts
- On-chip RAM ECC error interrupts

**Note 1.** Refer to Arm® CoreLink™ GIC-600 Generic Interrupt Controller Revision: r1p6 Technical Reference Manual.

**Note 2.** Refer to **Section 41, General Purpose Input Output Port (GPIO)**.

### 8.1.1 External Signal Pins

The following table shows external signal pins for Interrupt Controller.

Table 8.1 External Signal Pins

| Name | Width | I/O | Description |
|---|---|---|---|
| NMI | 1 | I | NMI (Non Maskable Interrupt) pin.<br>Refer to **Section 8.7.1**. |
| IRQ0 – IRQ7 | 8 | I | IRQ (Interrupt Request) pins.<br>GPIO pins can be used as IRQ pins.<br>Refer to **Section 8.7.2**. |
| P0_0 – P23_1 | 70 | I | GPIO pins.<br>GPIO pins can be used as external interrupt input pins (GPIOINT0-81).<br>Assign only 32 pins in GPIOINT0-81 to TINT0-31.<br>Refer to **Section 8.7.3**. |

### 8.1.2 Block Diagram

The block diagram of Interrupt Controller is as follows.

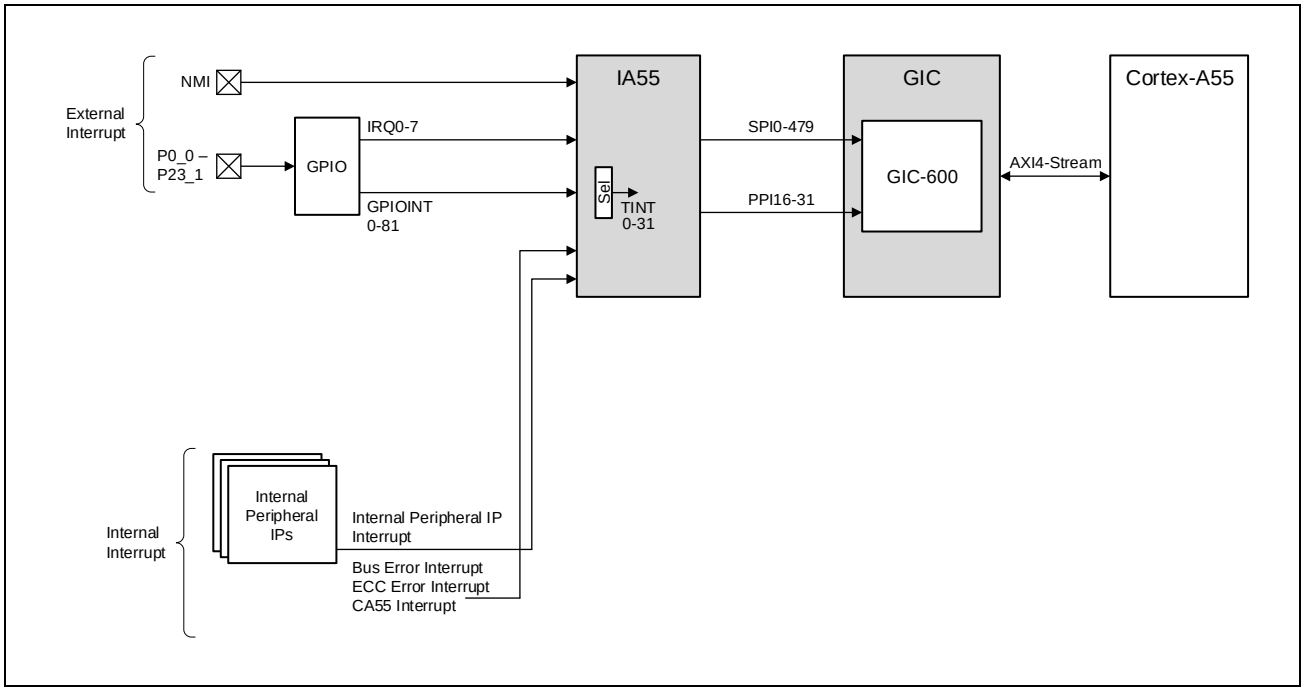



Figure 8.1 Interrupt Controller Block Diagram

#### GIC

GIC is equipped with Arm® CoreLink™ GIC-600 Generic Interrupt Controller for Cortex-A55.

#### IA55

IA55 performs various interrupt controls including synchronization for the external interrupts of NMI, IRQ, and GPIOINT and the interrupts of the built-in peripheral interrupts output by each IP. And it notifies the interrupt to the GIC.

- Select 32 TINT from 82 GPIOINT.
- Integration of bus error interrupts from system bus.
- Integration of ECC error interrupts from On-chip RAM.
- Indicate interrupt status. (NMI, IRQ, TINT, integrated bus error interrupt and integrated ECC error interrupt)
- Setting of interrupt detection method. (NMI, IRQ and TINT)

## 8.2 Interrupt Mapping

The following table indicates interrupt mapping for Cortex-A55.

Refer to **Section 14, Direct Memory Access Controller** with the interrupts through DMA Controller, and **Section 8.8.1**.

Table 8.2 Interrupt mapping (1/13)

| Interrupt Source*2 | Cause of Interrupt | Cortex-A55 | | Interrupt Type*1 | Note |
|---|---|---|---|---|---|
| | | Interrupt ID | SGI,PPI,SPI No. | | |
| GIC (Software Interrupt) | SGI ID0 | 0 | SGI 0 | — | |
| | SGI ID1 | 1 | SGI 1 | — | |
| | SGI ID2 | 2 | SGI 2 | — | |
| | SGI ID3 | 3 | SGI 3 | — | |
| | SGI ID4 | 4 | SGI 4 | — | |
| | SGI ID5 | 5 | SGI 5 | — | |
| | SGI ID6 | 6 | SGI 6 | — | |
| | SGI ID7 | 7 | SGI 7 | — | |
| | SGI ID8 | 8 | SGI 8 | — | |
| | SGI ID9 | 9 | SGI 9 | — | |
| | SGI ID10 | 10 | SGI 10 | — | |
| | SGI ID11 | 11 | SGI 11 | — | |
| | SGI ID12 | 12 | SGI 12 | — | |
| | SGI ID13 | 13 | SGI 13 | — | |
| | SGI ID14 | 14 | SGI 14 | — | |
| | SGI ID15 | 15 | SGI 15 | — | |
| Reserved | — | 16 | PPI 0 | — | |
| Reserved | — | 17 | PPI 1 | — | |
| Reserved | — | 18 | PPI 2 | — | |
| Reserved | — | 19 | PPI 3 | — | |
| Reserved | — | 20 | PPI 4 | — | |
| Reserved | — | 21 | PPI 5 | — | |
| Cortex-A55 | nCOMIRQ | 22 | PPI 6 | Level | |
| | nPMUIRQ | 23 | PPI 7 | Level | |
| | CTIIRQ | 24 | PPI 8 | Level | |
| | nVCPUMNTIRQ | 25 | PPI 9 | Level | |
| | nCNTHPIRQ | 26 | PPI 10 | Level | |
| | nCNTVIRQ | 27 | PPI 11 | Level | |
| | nCNTHVIRQ | 28 | PPI 12 | Level | |
| | nCNTPSIRQ | 29 | PPI 13 | Level | |
| | nCNTPNSIRQ | 30 | PPI 14 | Level | |
| Reserved | — | 31 | PPI 15 | — | |
| NMI pin | NMI | 32 | SPI 0 | Level | *3 |

Table 8.2 Interrupt mapping (2/13)

| Interrupt Source*2 | Cause of Interrupt | Cortex-A55 | | Interrupt Type*1 | Note |
|---|---|---|---|---|---|
| | | Interrupt ID | SGI,PPI,SPI No. | | |
| IRQ pins | IRQ0 | 33 | SPI 1 | Level | *4 |
| | IRQ1 | 34 | SPI 2 | Level | *4 |
| | IRQ2 | 35 | SPI 3 | Level | *4 |
| | IRQ3 | 36 | SPI 4 | Level | *4 |
| | IRQ4 | 37 | SPI 5 | Level | *4 |
| | IRQ5 | 38 | SPI 6 | Level | *4 |
| | IRQ6 | 39 | SPI 7 | Level | *4 |
| | IRQ7 | 40 | SPI 8 | Level | *4 |
| Cortex-A55 | nCOMMIRQ[1] | 41 | SPI 9 | Level | |
| | nCOMMIRQ[0] | 42 | SPI 10 | Level | |
| | nPMUIRQ[1] | 43 | SPI 11 | Level | |
| | nPMUIRQ[0] | 44 | SPI 12 | Level | |
| | nCLUSTERPMUIRQ | 45 | SPI 13 | Level | |
| Reserved | — | 46 | SPI 14 | — | |
| Cortex-A55 | nERRIRQ[1] | 47 | SPI 15 | Level | |
| | nERRIRQ[0] | 48 | SPI 16 | Level | |
| Reserved | — | 49 | SPI 17 | — | |
| Cortex-A55 | nFAULTIRQ[1] | 50 | SPI 18 | Level | |
| | nFAULTIRQ[0] | 51 | SPI 19 | Level | |
| Reserved | — | 52 | SPI 20 | — | |
| Reserved | — | 53 | SPI 21 | — | |
| GIC | fault_int | 54 | SPI 22 | Level | |
| | err_int | 55 | SPI 23 | Level | |
| | pmu_int | 56 | SPI 24 | Level | |
| System Bus | BUS_ERR_INT | 57 | SPI 25 | Edge | *6 |
| Reserved | — | 58 | SPI 26 | — | |
| Reserved | — | 59 | SPI 27 | — | |
| Reserved | — | 60 | SPI 28 | — | |
| Reserved | — | 61 | SPI 29 | — | |
| Reserved | — | 62 | SPI 30 | — | |
| Reserved | — | 63 | SPI 31 | — | |
| Reserved | — | 64 | SPI 32 | — | |
| Reserved | — | 65 | SPI 33 | — | |
| ECCRAM0 | EC7TIE1_0 | 66 | SPI 34 | Edge | *7 |
| | EC7TIE2_0 | 67 | SPI 35 | Edge | *7 |
| | EC7TIOVF_0 | 68 | SPI 36 | Edge | *7 |
| ECCRAM1 | EC7TIE1_1 | 69 | SPI 37 | Edge | *7 |
| | EC7TIE2_1 | 70 | SPI 38 | Edge | *7 |
| | EC7TIOVF_1 | 71 | SPI 39 | Edge | *7 |
| DDR3L/4 | controller_int | 72 | SPI 40 | Level | |
| Reserved | — | 73 | SPI 41 | — | |
| SYSC | SYS_LPM_INT | 74 | SPI 42 | Level | |
| | SYS_CA55STBYDONE_INT | 75 | SPI 43 | Level | |
| Reserved | — | 76 | SPI 44 | — | |

Table 8.2 Interrupt mapping (3/13)

| Interrupt Source*2 | Cause of Interrupt | Cortex-A55 | | Interrupt Type*1 | Note |
|---|---|---|---|---|---|
| | | Interrupt ID | SGI,PPI,SPI No. | | |
| SYSC | SYS_CA55_DENY | 77 | SPI 45 | Level | |
| GTM (ch0) | OSTM0TINT | 78 | SPI 46 | Edge | |
| GTM (ch1) | OSTM1TINT | 79 | SPI 47 | Edge | |
| GTM (ch2) | OSTM2TINT | 80 | SPI 48 | Edge | |
| WDT (CA55 core0) | WDTINT_A0 | 81 | SPI 49 | Level | |
| | PERROUT_A0 | 82 | SPI 50 | Level | |
| Reserved | — | 83 | SPI 51 | — | |
| Reserved | — | 84 | SPI 52 | — | |
| Reserved | — | 85 | SPI 53 | — | |
| Reserved | — | 86 | SPI 54 | — | |
| Reserved | — | 87 | SPI 55 | — | |
| Reserved | — | 88 | SPI 56 | — | |
| Reserved | — | 89 | SPI 57 | — | |
| Reserved | — | 90 | SPI 58 | — | |
| Reserved | — | 91 | SPI 59 | — | |
| Reserved | — | 92 | SPI 60 | — | |
| Reserved | — | 93 | SPI 61 | — | |
| Reserved | — | 94 | SPI 62 | — | |
| Reserved | — | 95 | SPI 63 | — | |
| Reserved | — | 96 | SPI 64 | — | |
| Reserved | — | 97 | SPI 65 | — | |
| Reserved | — | 98 | SPI 66 | — | |
| Reserved | — | 99 | SPI 67 | — | |
| Reserved | — | 100 | SPI 68 | — | |
| Reserved | — | 101 | SPI 69 | — | |
| Reserved | — | 102 | SPI 70 | — | |
| Reserved | — | 103 | SPI 71 | — | |
| Reserved | — | 104 | SPI 72 | — | |
| Reserved | — | 105 | SPI 73 | — | |
| Reserved | — | 106 | SPI 74 | — | |
| Reserved | — | 107 | SPI 75 | — | |
| Reserved | — | 108 | SPI 76 | — | |
| Reserved | — | 109 | SPI 77 | — | |
| Reserved | — | 110 | SPI 78 | — | |
| Reserved | — | 111 | SPI 79 | — | |
| Reserved | — | 112 | SPI 80 | — | |
| Reserved | — | 113 | SPI 81 | — | |
| Reserved | — | 114 | SPI 82 | — | |
| Reserved | — | 115 | SPI 83 | — | |
| Reserved | — | 116 | SPI 84 | – | |
| Reserved | — | 117 | SPI 85 | – | |
| Reserved | — | 118 | SPI 86 | – | |

Table 8.2 Interrupt mapping (4/13)

| Interrupt Source*2 | Cause of Interrupt | Cortex-A55 | | Interrupt Type*1 | Note |
|---|---|---|---|---|---|
| | | Interrupt ID | SGI,PPI,SPI No. | | |
| Reserved | — | 119 | SPI 87 | — | |
| Reserved | — | 120 | SPI 88 | — | |
| Reserved | — | 121 | SPI 89 | — | |
| USB2.0 (Host ch0) | U2H0_INT | 122 | SPI 90 | Level | |
| | U2H0_OHCI_INT | 123 | SPI 91 | Level | |
| | U2H0_EHCI_INT | 124 | SPI 92 | Level | |
| | U2H0_WAKEON_INT | 125 | SPI 93 | Level | |
| | U2H0_OBINT | 126 | SPI 94 | Level | |
| Reserved | — | 127 | SPI 95 | — | |
| Reserved | — | 128 | SPI 96 | — | |
| Reserved | — | 129 | SPI 97 | — | |
| Reserved | — | 130 | SPI 98 | — | |
| Reserved | — | 131 | SPI 99 | — | |
| USB2.0 (Function) | U2P_IXL_INT | 132 | SPI 100 | Edge | |
| | U2P_INT_DMA[0] | 133 | SPI 101 | Level | |
| | U2P_INT_DMA[1] | 134 | SPI 102 | Level | |
| | U2P_INT_DMAERR | 135 | SPI 103 | Level | |
| SDHI (ch0) | OXMNIRQ0 | 136 | SPI 104 | Level | |
| | OXASIOIRQ0 | 137 | SPI 105 | Level | |
| Reserved | — | 138 | SPI 106 | — | |
| Reserved | — | 139 | SPI 107 | — | |
| Reserved | — | 140 | SPI 108 | — | |
| Reserved | — | 141 | SPI 109 | — | |
| Reserved | — | 142 | SPI 110 | — | |
| Reserved | — | 143 | SPI 111 | — | |
| Reserved | — | 144 | SPI 112 | — | |
| Reserved | — | 145 | SPI 113 | — | |
| Reserved | — | 146 | SPI 114 | — | |
| Reserved | — | 147 | SPI 115 | — | |
| Reserved | — | 148 | SPI 116 | — | |
| Reserved | — | 149 | SPI 117 | — | |
| Reserved | — | 150 | SPI 118 | — | |
| Reserved | — | 151 | SPI 119 | — | |
| Reserved | — | 152 | SPI 120 | — | |
| Reserved | — | 153 | SPI 121 | — | |
| Reserved | — | 154 | SPI 122 | — | |
| Reserved | — | 155 | SPI 123 | — | |
| Reserved | — | 156 | SPI 124 | — | |

Table 8.2 Interrupt mapping (5/13)

| Interrupt Source*2 | Cause of Interrupt | Cortex-A55 | | Interrupt Type*1 | Note |
|---|---|---|---|---|---|
| | | Interrupt ID | SGI,PPI,SPI No. | | |
| DMAC (Non-Secure) | DMAINT0_NS | 157 | SPI 125 | Edge | |
| | DMAINT1_NS | 158 | SPI 126 | Edge | |
| | DMAINT2_NS | 159 | SPI 127 | Edge | |
| | DMAINT3_NS | 160 | SPI 128 | Edge | |
| | DMAINT4_NS | 161 | SPI 129 | Edge | |
| | DMAINT5_NS | 162 | SPI 130 | Edge | |
| | DMAINT6_NS | 163 | SPI 131 | Edge | |
| | DMAINT7_NS | 164 | SPI 132 | Edge | |
| | DMAINT8_NS | 165 | SPI 133 | Edge | |
| | DMAINT9_NS | 166 | SPI 134 | Edge | |
| | DMAINT10_NS | 167 | SPI 135 | Edge | |
| | DMAINT11_NS | 168 | SPI 136 | Edge | |
| | DMAINT12_NS | 169 | SPI 137 | Edge | |
| | DMAINT13_NS | 170 | SPI 138 | Edge | |
| | DMAINT14_NS | 171 | SPI 139 | Edge | |
| | DMAINT15_NS | 172 | SPI 140 | Edge | |
| | DMAERR_NS | 173 | SPI 141 | Edge | |
| MIPI-DSI | dsi_int_sq0 | 174 | SPI 142 | Level | |
| | dsi_int_sq1 | 175 | SPI 143 | Level | |
| | dsi_int_vin1 | 176 | SPI 144 | Level | |
| | dsi_int_rcv | 177 | SPI 145 | Level | |
| | dsi_int_ferr | 178 | SPI 146 | Level | |
| | dsi_int_ppi | 179 | SPI 147 | Level | |
| Reserved | — | 180 | SPI 148 | — | |
| LCDC | VSPD_INT | 181 | SPI 149 | Level | |
| Reserved | — | 182 | SPI 150 | — | |
| Reserved | — | 183 | SPI 151 | — | |
| LCDC | DU_INT | 184 | SPI 152 | Level | |
| Reserved | — | 185 | SPI 153 | — | |
| Reserved | — | 186 | SPI 154 | — | |
| Reserved | — | 187 | SPI 155 | — | |
| Reserved | — | 188 | SPI 156 | — | |
| Reserved | — | 189 | SPI 157 | — | |
| Reserved | — | 190 | SPI 158 | — | |
| Reserved | — | 191 | SPI 159 | — | |
| Reserved | — | 192 | SPI 160 | — | |
| Reserved | — | 193 | SPI 161 | — | |
| Reserved | — | 194 | SPI 162 | — | |
| Reserved | — | 195 | SPI 163 | — | |
| Reserved | — | 196 | SPI 164 | — | |
| Reserved | — | 197 | SPI 165 | — | |

Table 8.2 Interrupt mapping (6/13)

| Interrupt Source*2 | Cause of Interrupt | Cortex-A55 | | Interrupt Type*1 | Note |
|---|---|---|---|---|---|
| | | Interrupt ID | SGI,PPI,SPI No. | | |
| Reserved | — | 198 | SPI 166 | — | |
| Reserved | — | 199 | SPI 167 | — | |
| Reserved | — | 200 | SPI 168 | — | |
| DRW | DRW_IRQ | 201 | SPI 169 | Level | |
| MTU3a (ch0) | TGIA0 | 202 | SPI 170 | Edge | |
| | TGIB0 | 203 | SPI 171 | Edge | |
| | TGIC0 | 204 | SPI 172 | Edge | |
| | TGID0 | 205 | SPI 173 | Edge | |
| | TCIV0 | 206 | SPI 174 | Edge | |
| | TGIE0 | 207 | SPI 175 | Edge | |
| | TGIF0 | 208 | SPI 176 | Edge | |
| MTU3a (ch1) | TGIA1 | 209 | SPI 177 | Edge | |
| | TGIB1 | 210 | SPI 178 | Edge | |
| | TCIV1 | 211 | SPI 179 | Edge | |
| | TCIU1 | 212 | SPI 180 | Edge | |
| MTU3a (ch2) | TGIA2 | 213 | SPI 181 | Edge | |
| | TGIB2 | 214 | SPI 182 | Edge | |
| | TCIV2 | 215 | SPI 183 | Edge | |
| | TCIU2 | 216 | SPI 184 | Edge | |
| MTU3a (ch3) | TGIA3 | 217 | SPI 185 | Edge | |
| | TGIB3 | 218 | SPI 186 | Edge | |
| | TGIC3 | 219 | SPI 187 | Edge | |
| | TGID3 | 220 | SPI 188 | Edge | |
| | TCIV3 | 221 | SPI 189 | Edge | |
| MTU3a (ch4) | TGIA4 | 222 | SPI 190 | Edge | |
| | TGIB4 | 223 | SPI 191 | Edge | |
| | TGIC4 | 224 | SPI 192 | Edge | |
| | TGID4 | 225 | SPI 193 | Edge | |
| | TCIV4 | 226 | SPI 194 | Edge | |
| MTU3a (ch5) | TGIU5 | 227 | SPI 195 | Edge | |
| | TGIV5 | 228 | SPI 196 | Edge | |
| | TGIW5 | 229 | SPI 197 | Edge | |
| MTU3a (ch6) | TGIA6 | 230 | SPI 198 | Edge | |
| | TGIB6 | 231 | SPI 199 | Edge | |
| | TGIC6 | 232 | SPI 200 | Edge | |
| | TGID6 | 233 | SPI 201 | Edge | |
| | TCIV6 | 234 | SPI 202 | Edge | |
| MTU3a (ch7) | TGIA7 | 235 | SPI 203 | Edge | |
| | TGIB7 | 236 | SPI 204 | Edge | |
| | TGIC7 | 237 | SPI 205 | Edge | |
| | TGID7 | 238 | SPI 206 | Edge | |
| | TCIV7 | 239 | SPI 207 | Edge | |

Table 8.2 Interrupt mapping (7/13)

| Interrupt Source*2 | Cause of Interrupt | Cortex-A55 | | Interrupt Type*1 | Note |
|---|---|---|---|---|---|
| | | Interrupt ID | SGI,PPI,SPI No. | | |
| MTU3a (ch8) | TGIA8 | 240 | SPI 208 | Edge | |
| | TGIB8 | 241 | SPI 209 | Edge | |
| | TGIC8 | 242 | SPI 210 | Edge | |
| | TGID8 | 243 | SPI 211 | Edge | |
| | TCIV8 | 244 | SPI 212 | Edge | |
| Reserved | — | 245 | SPI 213 | — | |
| POE3 | OEI1 | 246 | SPI 214 | Level | |
| | OEI2 | 247 | SPI 215 | Level | |
| | OEI3 | 248 | SPI 216 | Level | |
| | OEI4 | 249 | SPI 217 | Level | |
| Reserved | — | 250 | SPI 218 | — | |
| Reserved | — | 251 | SPI 219 | — | |
| Reserved | — | 252 | SPI 220 | — | |
| Reserved | — | 253 | SPI 221 | — | |
| Reserved | — | 254 | SPI 222 | — | |
| Reserved | — | 255 | SPI 223 | — | |
| Reserved | — | 256 | SPI 224 | — | |
| Reserved | — | 257 | SPI 225 | — | |
| Reserved | — | 258 | SPI 226 | — | |
| Reserved | — | 259 | SPI 227 | — | |
| Reserved | — | 260 | SPI 228 | — | |
| Reserved | — | 261 | SPI 229 | — | |
| Reserved | — | 262 | SPI 230 | — | |
| Reserved | — | 263 | SPI 231 | — | |
| Reserved | — | 264 | SPI 232 | — | |
| Reserved | — | 265 | SPI 233 | — | |
| Reserved | — | 266 | SPI 234 | — | |
| Reserved | — | 267 | SPI 235 | — | |
| Reserved | — | 268 | SPI 236 | — | |
| Reserved | — | 269 | SPI 237 | — | |
| Reserved | — | 270 | SPI 238 | — | |
| Reserved | — | 271 | SPI 239 | — | |
| Reserved | — | 272 | SPI 240 | — | |
| Reserved | — | 273 | SPI 241 | — | |
| Reserved | — | 274 | SPI 242 | — | |
| Reserved | — | 275 | SPI 243 | — | |
| Reserved | — | 276 | SPI 244 | — | |
| Reserved | — | 277 | SPI 245 | — | |
| Reserved | — | 278 | SPI 246 | — | |
| Reserved | — | 279 | SPI 247 | — | |
| Reserved | — | 280 | SPI 248 | — | |
| Reserved | — | 281 | SPI 249 | — | |
| Reserved | — | 282 | SPI 250 | — | |
| Reserved | — | 283 | SPI 251 | — | |

Table 8.2 Interrupt mapping (8/13)

| Interrupt Source*2 | Cause of Interrupt | Cortex-A55 | | Interrupt Type*1 | Note |
|---|---|---|---|---|---|
| | | Interrupt ID | SGI,PPI,SPI No. | | |
| Reserved | — | 284 | SPI 252 | — | |
| Reserved | — | 285 | SPI 253 | — | |
| Reserved | — | 286 | SPI 254 | — | |
| Reserved | — | 287 | SPI 255 | — | |
| Reserved | — | 288 | SPI 256 | — | |
| Reserved | — | 289 | SPI 257 | — | |
| Reserved | — | 290 | SPI 258 | — | |
| Reserved | — | 291 | SPI 259 | — | |
| Reserved | — | 292 | SPI 260 | — | |
| Reserved | — | 293 | SPI 261 | — | |
| Reserved | — | 294 | SPI 262 | — | |
| Reserved | — | 295 | SPI 263 | — | |
| Reserved | — | 296 | SPI 264 | — | |
| Reserved | — | 297 | SPI 265 | — | |
| Reserved | — | 298 | SPI 266 | — | |
| Reserved | — | 299 | SPI 267 | — | |
| Reserved | — | 300 | SPI 268 | — | |
| Reserved | — | 301 | SPI 269 | — | |
| Reserved | — | 302 | SPI 270 | — | |
| Reserved | — | 303 | SPI 271 | — | |
| Reserved | — | 304 | SPI 272 | — | |
| Reserved | — | 305 | SPI 273 | — | |
| Reserved | — | 306 | SPI 274 | — | |
| Reserved | — | 307 | SPI 275 | — | |
| Reserved | — | 308 | SPI 276 | — | |
| Reserved | — | 309 | SPI 277 | — | |
| Reserved | — | 310 | SPI 278 | — | |
| Reserved | — | 311 | SPI 279 | — | |
| Reserved | — | 312 | SPI 280 | — | |
| Reserved | — | 313 | SPI 281 | — | |
| Reserved | — | 314 | SPI 282 | — | |
| Reserved | — | 315 | SPI 283 | — | |
| Reserved | — | 316 | SPI 284 | — | |
| Reserved | — | 317 | SPI 285 | — | |
| Reserved | — | 318 | SPI 286 | — | |
| Reserved | — | 319 | SPI 287 | — | |
| Reserved | — | 320 | SPI 288 | — | |
| Reserved | — | 321 | SPI 289 | — | |
| Reserved | — | 322 | SPI 290 | — | |
| Reserved | — | 323 | SPI 291 | — | |
| Reserved | — | 324 | SPI 292 | — | |
| Reserved | — | 325 | SPI 293 | — | |
| Reserved | — | 326 | SPI 294 | — | |
| Reserved | — | 327 | SPI 295 | — | |

Table 8.2 Interrupt mapping (9/13)

| Interrupt Source*2 | Cause of Interrupt | Cortex-A55 | | Interrupt Type*1 | Note |
|---|---|---|---|---|---|
| | | Interrupt ID | SGI,PPI,SPI No. | | |
| Reserved | — | 328 | SPI 296 | — | |
| Reserved | — | 329 | SPI 297 | — | |
| Reserved | — | 330 | SPI 298 | — | |
| Reserved | — | 331 | SPI 299 | — | |
| Reserved | — | 332 | SPI 300 | — | |
| Reserved | — | 333 | SPI 301 | — | |
| Reserved | — | 334 | SPI 302 | — | |
| Reserved | — | 335 | SPI 303 | — | |
| Reserved | — | 336 | SPI 304 | — | |
| Reserved | — | 337 | SPI 305 | — | |
| Reserved | — | 338 | SPI 306 | — | |
| Reserved | — | 339 | SPI 307 | — | |
| Reserved | — | 340 | SPI 308 | — | |
| Reserved | — | 341 | SPI 309 | — | |
| Reserved | — | 342 | SPI 310 | — | |
| Reserved | — | 343 | SPI 311 | — | |
| Reserved | — | 344 | SPI 312 | — | |
| Reserved | — | 345 | SPI 313 | — | |
| Reserved | — | 346 | SPI 314 | — | |
| Reserved | — | 347 | SPI 315 | — | |
| Reserved | — | 348 | SPI 316 | — | |
| Reserved | — | 349 | SPI 317 | — | |
| Reserved | — | 350 | SPI 318 | — | |
| Reserved | — | 351 | SPI 319 | — | |
| Reserved | — | 352 | SPI 320 | — | |
| Reserved | — | 353 | SPI 321 | — | |
| Reserved | — | 354 | SPI 322 | — | |
| Reserved | — | 355 | SPI 323 | — | |
| Reserved | — | 356 | SPI 324 | — | |
| Reserved | — | 357 | SPI 325 | — | |
| SSIF (ch0) | INT_ssif_int_req_0 | 358 | SPI 326 | Level | |
| | INT_ssif_dma_rx_0 | 359 | SPI 327 | Edge | |
| | INT_ssif_dma_tx_0 | 360 | SPI 328 | Edge | |
| Reserved | — | 361 | SPI 329 | — | |
| Reserved | — | 362 | SPI 330 | — | |
| Reserved | — | 363 | SPI 331 | — | |
| Reserved | — | 364 | SPI 332 | — | |
| Reserved | — | 365 | SPI 333 | — | |
| Reserved | — | 366 | SPI 334 | — | |
| Reserved | — | 367 | SPI 335 | — | |
| Reserved | — | 368 | SPI 336 | — | |
| Reserved | — | 369 | SPI 337 | — | |

Table 8.2 Interrupt mapping (10/13)

| Interrupt Source*2 | Cause of Interrupt | Cortex-A55 | | Interrupt Type*1 | Note |
|---|---|---|---|---|---|
| | | Interrupt ID | SGI,PPI,SPI No. | | |
| Reserved | — | 370 | SPI 338 | — | |
| Reserved | — | 371 | SPI 339 | — | |
| Reserved | — | 372 | SPI 340 | — | |
| Reserved | — | 373 | SPI 341 | — | |
| Reserved | — | 374 | SPI 342 | — | |
| | — | 375 | SPI 343 | — | |
| | — | 376 | SPI 344 | — | |
| | — | 377 | SPI 345 | — | |
| | — | 378 | SPI 346 | — | |
| Reserved | — | 379 | SPI 347 | — | |
| I2C (ch0) | INTRIICRI0 | 380 | SPI 348 | Edge | |
| | INTRIICTI0 | 381 | SPI 349 | Edge | |
| | INTRIICTEI0 | 382 | SPI 350 | Level | |
| | INTRIICNAKI0 | 383 | SPI 351 | Level | |
| | INTRIICSPI0 | 384 | SPI 352 | Level | |
| | INTRIICSTI0 | 385 | SPI 353 | Level | |
| | INTRIICALI0 | 386 | SPI 354 | Level | |
| | INTRIICTMOI0 | 387 | SPI 355 | Level | |
| I2C (ch1) | INTRIICRI1 | 388 | SPI 356 | Edge | |
| | INTRIICTI1 | 389 | SPI 357 | Edge | |
| | INTRIICTEI1 | 390 | SPI 358 | Level | |
| | INTRIICNAKI1 | 391 | SPI 359 | Level | |
| | INTRIICSPI1 | 392 | SPI 360 | Level | |
| | INTRIICSTI1 | 393 | SPI 361 | Level | |
| | INTRIICALI1 | 394 | SPI 362 | Level | |
| | INTRIICTMOI1 | 395 | SPI 363 | Level | |
| Reserved | — | 396 | SPI 364 | — | |
| Reserved | — | 397 | SPI 365 | — | |
| Reserved | — | 398 | SPI 366 | — | |
| Reserved | — | 399 | SPI 367 | — | |
| Reserved | — | 400 | SPI 368 | — | |
| Reserved | — | 401 | SPI 369 | — | |
| Reserved | — | 402 | SPI 370 | — | |
| Reserved | — | 403 | SPI 371 | — | |
| Reserved | — | 404 | SPI 372 | — | |
| Reserved | — | 405 | SPI 373 | — | |
| Reserved | — | 406 | SPI 374 | — | |
| Reserved | — | 407 | SPI 375 | — | |
| Reserved | — | 408 | SPI 376 | — | |
| Reserved | — | 409 | SPI 377 | — | |
| Reserved | — | 410 | SPI 378 | — | |
| Reserved | — | 411 | SPI 379 | — | |

Table 8.2 Interrupt mapping (11/13)

| Interrupt Source*2 | Cause of Interrupt | Cortex-A55 | | Interrupt Type*1 | Note |
|---|---|---|---|---|---|
| | | Interrupt ID | SGI,PPI,SPI No. | | |
| SCIFA (ch0) | ERI0 | 412 | SPI 380 | Level | |
| | BRI0 | 413 | SPI 381 | Level | |
| | RXI0 | 414 | SPI 382 | Level | |
| | TXI0 | 415 | SPI 383 | Level | |
| | TEI0_DRI0 | 416 | SPI 384 | Level | |
| SCIFA (ch1) | ERI1 | 417 | SPI 385 | Level | |
| | BRI1 | 418 | SPI 386 | Level | |
| | RXI1 | 419 | SPI 387 | Level | |
| | TXI1 | 420 | SPI 388 | Level | |
| | TEI1_DRI1 | 421 | SPI 389 | Level | |
| SCIFA (ch2) | ERI2 | 422 | SPI 390 | Level | |
| | BRI2 | 423 | SPI 391 | Level | |
| | RXI2 | 424 | SPI 392 | Level | |
| | TXI2 | 425 | SPI 393 | Level | |
| | TEI2_DRI2 | 426 | SPI 394 | Level | |
| SCIFA (ch3) | ERI3 | 427 | SPI 395 | Level | |
| | BRI3 | 428 | SPI 396 | Level | |
| | RXI3 | 429 | SPI 397 | Level | |
| | TXI3 | 430 | SPI 398 | Level | |
| | TEI3_DRI3 | 431 | SPI 399 | Level | |
| SCIFA (ch4) | ERI4 | 432 | SPI 400 | Level | |
| | BRI4 | 433 | SPI 401 | Level | |
| | RXI4 | 434 | SPI 402 | Level | |
| | TXI4 | 435 | SPI 403 | Level | |
| | TEI4_DRI4 | 436 | SPI 404 | Level | |
| SCIg (ch0) | ERI0 | 437 | SPI 405 | Level | |
| | RXI0 | 438 | SPI 406 | Edge | |
| | TXI0 | 439 | SPI 407 | Edge | |
| | TEI0 | 440 | SPI 408 | Level | |
| SCIg (ch1) | ERI1 | 441 | SPI 409 | Level | |
| | RXI1 | 442 | SPI 410 | Edge | |
| | TXI1 | 443 | SPI 411 | Edge | |
| | TEI1 | 444 | SPI 412 | Level | |
| RSPI (ch0) | SPRI0 | 445 | SPI 413 | Level | |
| | SPTI0 | 446 | SPI 414 | Level | |
| | SPEI0 | 447 | SPI 415 | Level | |
| RSPI (ch1) | SPRI1 | 448 | SPI 416 | Level | |
| | SPTI1 | 449 | SPI 417 | Level | |
| | SPEI1 | 450 | SPI 418 | Level | |
| Reserved | — | 451 | SPI 419 | — | |
| Reserved | — | 452 | SPI 420 | — | |
| Reserved | — | 453 | SPI 421 | — | |

Table 8.2 Interrupt mapping (12/13)

| Interrupt Source*2 | Cause of Interrupt | Cortex-A55 | | Interrupt Type*1 | Note |
|---|---|---|---|---|---|
| | | Interrupt ID | SGI,PPI,SPI No. | | |
| Reserved | — | 454 | SPI 422 | — | |
| Reserved | — | 455 | SPI 423 | — | |
| Reserved | — | 456 | SPI 424 | — | |
| Reserved | — | 457 | SPI 425 | — | |
| Reserved | — | 458 | SPI 426 | — | |
| Reserved | — | 459 | SPI 427 | — | |
| Reserved | — | 460 | SPI 428 | — | |
| Reserved | — | 461 | SPI 429 | — | |
| Reserved | — | 462 | SPI 430 | — | |
| Reserved | — | 463 | SPI 431 | — | |
| Reserved | — | 464 | SPI 432 | — | |
| Reserved | — | 465 | SPI 433 | — | |
| Reserved | — | 466 | SPI 434 | — | |
| Reserved | — | 467 | SPI 435 | — | |
| Reserved | — | 468 | SPI 436 | — | |
| Reserved | — | 469 | SPI 437 | — | |
| Reserved | — | 470 | SPI 438 | — | |
| Reserved | — | 471 | SPI 439 | — | |
| Reserved | — | 472 | SPI 440 | — | |
| Reserved | — | 473 | SPI 441 | — | |
| Reserved | — | 474 | SPI 442 | — | |
| Reserved | — | 475 | SPI 443 | — | |
| GPIO Interrupt | TINT0 | 476 | SPI 444 | Level | *5 |
| | TINT1 | 477 | SPI 445 | Level | *5 |
| | TINT2 | 478 | SPI 446 | Level | *5 |
| | TINT3 | 479 | SPI 447 | Level | *5 |
| | TINT4 | 480 | SPI 448 | Level | *5 |
| | TINT5 | 481 | SPI 449 | Level | *5 |
| | TINT6 | 482 | SPI 450 | Level | *5 |
| | TINT7 | 483 | SPI 451 | Level | *5 |
| | TINT8 | 484 | SPI 452 | Level | *5 |
| | TINT9 | 485 | SPI 453 | Level | *5 |
| | TINT10 | 486 | SPI 454 | Level | *5 |
| | TINT11 | 487 | SPI 455 | Level | *5 |

Table 8.2 Interrupt mapping (13/13)

| Interrupt Source*2 | Cause of Interrupt | Cortex-A55 | | Interrupt Type*1 | Note |
|---|---|---|---|---|---|
| | | Interrupt ID | SGI,PPI,SPI No. | | |
| GPIO Interrupt | TINT12 | 488 | SPI 456 | Level | *5 |
| | TINT13 | 489 | SPI 457 | Level | *5 |
| | TINT14 | 490 | SPI 458 | Level | *5 |
| | TINT15 | 491 | SPI 459 | Level | *5 |
| | TINT16 | 492 | SPI 460 | Level | *5 |
| | TINT17 | 493 | SPI 461 | Level | *5 |
| | TINT18 | 494 | SPI 462 | Level | *5 |
| | TINT19 | 495 | SPI 463 | Level | *5 |
| | TINT20 | 496 | SPI 464 | Level | *5 |
| | TINT21 | 497 | SPI 465 | Level | *5 |
| | TINT22 | 498 | SPI 466 | Level | *5 |
| | TINT23 | 499 | SPI 467 | Level | *5 |
| | TINT24 | 500 | SPI 468 | Level | *5 |
| | TINT25 | 501 | SPI 469 | Level | *5 |
| | TINT26 | 502 | SPI 470 | Level | *5 |
| | TINT27 | 503 | SPI 471 | Level | *5 |
| | TINT28 | 504 | SPI 472 | Level | *5 |
| | TINT29 | 505 | SPI 473 | Level | *5 |
| | TINT30 | 506 | SPI 474 | Level | *5 |
| | TINT31 | 507 | SPI 475 | Level | *5 |
| Reserved | — | 508 | SPI 476 | — | |
| Reserved | — | 509 | SPI 477 | — | |
| Reserved | — | 510 | SPI 478 | — | |
| Reserved | — | 511 | SPI 479 | — | |

Note 1. Set GICD_ICFGR<n> of the interrupt controller for Cortex-A55 (GIC-600) according to the description of Interrupt Type for each Interrupt ID.
Refer to “ARM CoreLink GIC-600 Generic Interrupt Controller Technical Reference Manual” for the specification of GIC-600 registers.

Note 2. Refer to the source unit section for the specification of each interrupt source.

Note 3. NMI is not treated as NMI exception.
NMI interrupt setting can be controlled by the registers in IA55. Refer to **Section 8.7.1** for details.

Note 4. IRQ0-7 interrupt settings can be controlled by the registers in IA55. Refer to **Section 8.7.2** for details.

Note 5. TINT0-31 interrupt settings can be controlled by the registers in IA55. Refer to **Section 8.7.3** for details.

Note 6. BUS_ERR_INT is the interrupt integrated multiple interrupts into one in IA55.
The interrupt settings can be controlled by the registers in IA55. Refer to **Section 8.7.4.1** for details.

Note 7. Each EC7TIE1, TC7TIE2 and EC7TIOVF is the interrupt integrated multiple interrupts into one in IA55.
The interrupt settings can be controlled by the registers in IA55. Refer to **Section 8.7.4.2** for details.

## 8.3 GIC-600 Register Configuration

Refer to Arm® CoreLink™ GIC-600 Generic Interrupt Controller Revision: r1p6 Technical Reference Manual.
ITS and LPI are not supported.

Base Address: H’0_1190_0000 (Cortex-A55 Address Space)

## 8.4 GIC-600 Register Descriptions

Refer to Arm® CoreLink™ GIC-600 Generic Interrupt Controller Revision: r1p6 Technical Reference Manual for detail.

## 8.5 IA55 Register Configuration

Base Address: H'0_110A_0000 (Cortex-A55 Address Space)

The following table shows IA55 Register list.
Prohibit to write undefined area.

Table 8.3 IA55 Register List

| Register Name | Abbreviation | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|
| NMI Status Control Register | NSCR | R/W | H'0000_0000 | H'0000 | 32 |
| NMI Interrupt Type Selection Register | NITSR | R/W | H'0000_0000 | H'0004 | 32 |
| IRQ Status Control Register | ISCR | R/W | H'0000_0000 | H'0010 | 32 |
| IRQ Interrupt Type Selection Register | IITSR | R/W | H'0000_0000 | H'0014 | 32 |
| TINT Status Control Register | TSCR | R/W | H'0000_0000 | H'0020 | 32 |
| TINT Interrupt Type Selection Register0 | TITSR0 | R/W | H'0000_0000 | H'0024 | 32 |
| TINT Interrupt Type Selection Register1 | TITSR1 | R/W | H'0000_0000 | H'0028 | 32 |
| TINT Source Selection Register0 | TSSR0 | R/W | H'0000_0000 | H'0030 | 32 |
| TINT Source Selection Register1 | TSSR1 | R/W | H'0000_0000 | H'0034 | 32 |
| TINT Source Selection Register2 | TSSR2 | R/W | H'0000_0000 | H'0038 | 32 |
| TINT Source Selection Register3 | TSSR3 | R/W | H'0000_0000 | H'003C | 32 |
| TINT Source Selection Register4 | TSSR4 | R/W | H'0000_0000 | H'0040 | 32 |
| TINT Source Selection Register5 | TSSR5 | R/W | H'0000_0000 | H'0044 | 32 |
| TINT Source Selection Register6 | TSSR6 | R/W | H'0000_0000 | H'0048 | 32 |
| TINT Source Selection Register7 | TSSR7 | R/W | H'0000_0000 | H'004C | 32 |
| Bus Error Interrupt Status Control Register0 | BEISR0 | R/W | H'0000_0000 | H'0050 | 32 |
| Bus Error Interrupt Status Control Register1 | BEISR1 | R/W | H'0000_0000 | H'0054 | 32 |
| ECCRAM Error Interrupt Status Control Register0 | EREISR0 | R/W | H'0000_0000 | H'0060 | 32 |
| ECCRAM Error Interrupt Status Control Register1 | EREISR1 | R/W | H'0000_0000 | H'0064 | 32 |

# 8.6 IA55 Register Descriptions

## 8.6.1 NMI Status Control Register (NSCR)

This register shows NMI status.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | NSMON |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | NSTAT |
| 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<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 16 | NSMON | 0 | R | This bit indicates NMI pin signal level.<br><br>[Read operation]<br>0: NMI pin is Low-level.<br>1: NMI pin is High-level.<br>[Write operation]<br>Invalid to write. |
| 15 to 1 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 0 | NSTAT | 0 | R/W | This bit indicates NMI interrupt status.<br>NMI interrupt status can be cleared writing 0 in case NSTAT is 1.<br><br>[Read operation]<br>0: NMI interrupt is not detected.<br>1: NMI interrupt is detected.<br>[Write operation]<br>• In case NSTAT is 1<br>0: NMI interrupt status is cleared.<br>1: Invalid to write.<br>• In case NSTAT is 0<br>Invalid to write. |

## 8.6.2 NMI Interrupt Type Selection Register (NITSR)

This register selects NMI detecting method.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | NTSEL |
| 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<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 0 | NTSEL | 0 | R/W | This bit is used to select NMI interrupt detection method.<br>0: Falling-edge detection.<br>1: Rising-edge detection. |

### 8.6.3 IRQ Status Control Register (ISCR)

This register shows IRQ status.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | ISTAT7 | ISTAT6 | ISTAT5 | ISTAT4 | ISTAT3 | ISTAT2 | ISTAT1 | ISTAT0 |
| 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<br>Whenever it is read, 0 is read. The written value will be ignored. |
| n | ISTATn | 0 | R/W | This bit indicates IRQn interrupt status.<br>When the interrupt detection method is set to an edge type detection in IITSR, IRQn interrupt status can be cleared writing 0 in case ISTATn is 1.<br><br>[Read operation]<br>0: IRQn interrupt is not detected.<br>1: IRQn interrupt is detected.<br>[Write operation]<br>When “Falling-edge detection”, “Rising-edge detection” or “Falling/Rising-edge detection” is set in ITSR.:<br>• In case ISTAT is 1<br>0: IRQn interrupt detection status is cleared.<br>1: Invalid to write.<br>• In case ISTAT is 0<br>Invalid to write.<br>When “Low-level detection” is set in IITSR.:<br>Invalid to write. |

**Note:** n = 7 to 0

### 8.6.4 IRQ Interrupt Type Selection Register (IITSR)

This register selects IRQ detecting method.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | IITSEL7 | | IITSEL6 | | IITSEL5 | | IITSEL4 | | IITSEL3 | | IITSEL2 | | IITSEL1 | | IITSEL0 | |
| 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<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 2n+1 to 2n | IITSEL | All 0 | R/W | This bit is used to select IRQn interrupt detection method.<br>00: Low-level detection.<br>01: Falling-edge detection.<br>10: Rising-edge detection.<br>11: Falling/Rising-edge detection. |

**Note:** n = 7 to 0

### 8.6.5 TINT Status Control Register (TSCR)

This register shows TINT status.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TSTAT31 | TSTAT30 | TSTAT29 | TSTAT28 | TSTAT27 | TSTAT26 | TSTAT25 | TSTAT24 | TSTAT23 | TSTAT22 | TSTAT21 | TSTAT20 | TSTAT19 | TSTAT18 | TSTAT17 | TSTAT16 |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TSTAT15 | TSTAT14 | TSTAT13 | TSTAT12 | TSTAT11 | TSTAT10 | TSTAT9 | TSTAT8 | TSTAT7 | TSTAT6 | TSTAT5 | TSTAT4 | TSTAT3 | TSTAT2 | TSTAT1 | TSTAT0 |
| 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 |
|---|---|---|---|---|
| n | TSTATn | All 0 | R/W | This bit indicates TINTn interrupt status.<br>When the interrupt detection method is set to edge type detection in TITSR0/1, TINTn interrupt status can be cleared writing 0 in case TSTATn is 1.<br><br>[Read operation]<br>0: TINTn interrupt is not detected.<br>1: TINTn interrupt is detected.<br>[Write operation]<br>When “Rising-edge detection” or “Falling-edge detection” is set in TITSR0/1.:<br>• In case TSTAT is 1<br>0: TINTn interrupt detection status is cleared.<br>1: Invalid to write.<br>• In case TSTAT is 0<br>Invalid to write.<br>When “Low-level detection” or ”High-Level detection” is set in TITSR0/1.:<br>Invalid to write. |

**Note:** n = 31 to 0

### 8.6.6 TINT Interrupt Type Selection Register0 (TITSR0)

This register selects detecting method of TINT15 to TINT0.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TITSEL15 | | TITSEL14 | | TITSEL13 | | TITSEL12 | | TITSEL11 | | TITSEL10 | | TITSEL9 | | TITSEL8 | |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TITSEL7 | | TITSEL6 | | TITSEL5 | | TITSEL4 | | TITSEL3 | | TITSEL2 | | TITSEL1 | | TITSEL0 | |
| 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 |
|---|---|---|---|---|
| 2n+1 to 2n | TITSELn | All 0 | R/W | This bit is used to select TINTn interrupt detection method.<br>00: Rising-edge detection.<br>01: Falling-edge detection.<br>10: High-level detection.<br>11: Low-level detection. |

**Note:** n = 15 to 0

### 8.6.7 TINT Interrupt Type Selection Register1 (TITSR1)

This register selects detecting method of TINT31 to TINT16.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TITSEL31 | | TITSEL30 | | TITSEL29 | | TITSEL28 | | TITSEL27 | | TITSEL26 | | TITSEL25 | | TITSEL24 | |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TITSEL23 | | TITSEL22 | | TITSEL21 | | TITSEL20 | | TITSEL19 | | TITSEL18 | | TITSEL17 | | TITSEL16 | |
| 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 |
|---|---|---|---|---|
| 2n+1 to 2n | TITSEL n+16 | All 0 | R/W | This bit is used to select TINTn+16 interrupt detection method.<br>00: Rising-edge detection.<br>01: Falling-edge detection.<br>10: High-level detection.<br>11: Low-level detection. |

**Note:** n = 15 to 0

## 8.6.8 TINT Source Selection Register0 (TSSR0)

This register selects the interrupt source of TINT3 to TINT0.

Refer to **Section 8.7.3** about mapping GPIOINT onto GPIO pin.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TIEN3 | TSSEL3 | | | | | | | TIEN2 | TSSEL2 | | | | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TIEN1 | TSSEL1 | | | | | | | TIEN0 | TSSEL0 | | | | | | |
| 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 |
|---|---|---|---|---|
| 8n+7 | TIENn | 0 | R/W | This bit enables TINTn.<br>1: Enable.<br>0: Invalid. |
| 8n+6 to 8n | TSSELn | All 0 | R/W | This bit selects a source of TINTn.<br>0000000: TINTn = GPIOINT0<br>0000001: TINTn = GPIOINT1<br>~<br>1010001: TINTn = GPIOINT81<br>1010010: Cannot be set.<br>~<br>1111011: Cannot be set.<br>1111100: Cannot be set.<br>1111101: Cannot be set.<br>1111110: Cannot be set.<br>1111111: Cannot be set.<br><br>Assign different sources to TINT0-31. |

**Note:** n = 3 to 0

### 8.6.9 TINT Source Selection Register1 (TSSR1)

This register selects the interrupt source of TINT7 to TINT4.

Refer to **Section 8.7.3** about mapping GPIOINT onto GPIO pin.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TIEN7 | TSSEL7 | | | | | | | TIEN6 | TSSEL6 | | | | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TIEN5 | TSSEL5 | | | | | | | TIEN4 | TSSEL4 | | | | | | |
| 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 |
|---|---|---|---|---|
| 8(n-4)+7 | TIENn | 0 | R/W | This bit enables TINTn.<br>1: Enable.<br>0: Invalid. |
| 8(n-4)+6 to 8(n-4) | TSSELn | All 0 | R/W | This bit selects a source of TINTn.<br>0000000: TINTn = GPIOINT0<br>0000001: TINTn = GPIOINT1<br>~<br>1010001: TINTn = GPIOINT81<br>1010010: Cannot be set.<br>~<br>1111011: Cannot be set.<br>1111100: Cannot be set.<br>1111101: Cannot be set.<br>1111110: Cannot be set.<br>1111111: Cannot be set.<br><br>Assign different sources to TINT0-31. |

**Note:** n = 7 to 4

## 8.6.10 TINT Source Selection Register2 (TSSR2)

This register selects the interrupt source of TINT11 to TINT8.

Refer to **Section 8.7.3** about mapping GPIOINT onto GPIO pin.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TIEN11 | TSSEL11 | | | | | | | TIEN10 | TSSEL10 | | | | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TIEN9 | TSSEL9 | | | | | | | TIEN8 | TSSEL8 | | | | | | |
| 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 |
|---|---|---|---|---|
| 8(n-8)+7 | TIENn | 0 | R/W | This bit enables TINTn.<br>1: Enable.<br>0: Invalid. |
| 8(n-8)+6 to 8(n-8) | TSSELn | All 0 | R/W | This bit selects a source of TINTn.<br>0000000: TINTn = GPIOINT0<br>0000001: TINTn = GPIOINT1<br>~<br>1010001: TINTn = GPIOINT81<br>1010010: Cannot be set.<br>~<br>1111011: Cannot be set.<br>1111100: Cannot be set.<br>1111101: Cannot be set.<br>1111110: Cannot be set.<br>1111111: Cannot be set.<br><br>Assign different sources to TINT0-31. |

**Note:** n = 11 to 8

## 8.6.11 TINT Source Selection Register3 (TSSR3)

This register selects the interrupt source of TINT15 to TINT12.

Refer to **Section 8.7.3** about mapping GPIOINT onto GPIO pin.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TIEN15 | TSSEL15 | | | | | | | TIEN14 | TSSEL14 | | | | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TIEN13 | TSSEL13 | | | | | | | TIEN12 | TSSEL12 | | | | | | |
| 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 |
|---|---|---|---|---|
| 8(n-12)+7 | TIENn | 0 | R/W | This bit enables TINTn.<br>1: Enable.<br>0: Invalid. |
| 8(n-12)+6 to 8(n-12) | TSSELn | All 0 | R/W | This bit selects a source of TINTn.<br>0000000: TINTn = GPIOINT0<br>0000001: TINTn = GPIOINT1<br>~<br>1010001: TINTn = GPIOINT81<br>1010010: Cannot be set.<br>~<br>1111011: Cannot be set.<br>1111100: Cannot be set.<br>1111101: Cannot be set.<br>1111110: Cannot be set.<br>1111111: Cannot be set.<br><br>Assign different sources to TINT0-31. |

**Note:** n = 15 to 12

## 8.6.12 TINT Source Selection Register4 (TSSR4)

This register selects the interrupt source of TINT19 to TINT16.

Refer to **Section 8.7.3** about mapping GPIOINT onto GPIO pin.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TIEN19 | TSSEL19 | | | | | | | TIEN18 | TSSEL18 | | | | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TIEN17 | TSSEL17 | | | | | | | TIEN16 | TSSEL16 | | | | | | |
| 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 |
|---|---|---|---|---|
| 8(n-16)+7 | TIENn | 0 | R/W | This bit enables TINTn.<br>1: Enable.<br>0: Invalid. |
| 8(n-16)+6 to 8(n-16) | TSSELn | All 0 | R/W | This bit selects a source of TINTn.<br>0000000: TINTn = GPIOINT0<br>0000001: TINTn = GPIOINT1<br>~<br>1010001: TINTn = GPIOINT81<br>1010010: Cannot be set.<br>~<br>1111011: Cannot be set.<br>1111100: Cannot be set.<br>1111101: Cannot be set.<br>1111110: Cannot be set.<br>1111111: Cannot be set.<br><br>Assign different sources to TINT0-31. |

**Note:** n = 19 to 16

### 8.6.13 TINT Source Selection Register5 (TSSR5)

This register selects the interrupt source of TINT23 to TINT20.

Refer to **Section 8.7.3** about mapping GPIOINT onto GPIO pin.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TIEN23 | TSSEL23 | | | | | | | TIEN22 | TSSEL22 | | | | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TIEN21 | TSSEL21 | | | | | | | TIEN20 | TSSEL20 | | | | | | |
| 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 |
|---|---|---|---|---|
| 8(n-20)+7 | TIENn | 0 | R/W | This bit enables TINTn.<br>1: Enable.<br>0: Invalid. |
| 8(n-20)+6 to 8(n-20) | TSSELn | All 0 | R/W | This bit selects a source of TINTn.<br>0000000: TINTn = GPIOINT0<br>0000001: TINTn = GPIOINT1<br>~<br>1010001: TINTn = GPIOINT81<br>1010010: Cannot be set.<br>~<br>1111011: Cannot be set.<br>1111100: Cannot be set.<br>1111101: Cannot be set.<br>1111110: Cannot be set.<br>1111111: Cannot be set.<br><br>Assign different sources to TINT0-31. |

**Note:** n = 23 to 20

## 8.6.14 TINT Source Selection Register6 (TSSR6)

This register selects the interrupt source of TINT27 to TINT24.

Refer to **Section 8.7.3** about mapping GPIOINT onto GPIO pin.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TIEN27 | TSSEL27 | | | | | | | TIEN26 | TSSEL26 | | | | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TIEN25 | TSSEL25 | | | | | | | TIEN24 | TSSEL24 | | | | | | |
| 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 |
|---|---|---|---|---|
| 8(n-24)+7 | TIENn | 0 | R/W | This bit enables TINTn.<br>1: Enable.<br>0: Invalid. |
| 8(n-24)+6 to 8(n-24) | TSSELn | All 0 | R/W | This bit selects a source of TINTn.<br>0000000: TINTn = GPIOINT0<br>0000001: TINTn = GPIOINT1<br>~<br>1010001: TINTn = GPIOINT81<br>1010010: Cannot be set.<br>~<br>1111011: Cannot be set.<br>1111100: Cannot be set.<br>1111101: Cannot be set.<br>1111110: Cannot be set.<br>1111111: Cannot be set.<br><br>Assign different sources to TINT0-31. |

**Note:** n = 27 to 24

## 8.6.15 TINT Source Selection Register7 (TSSR7)

This register selects the interrupt source of TINT31 to TINT28.

Refer to **Section 8.7.3** about mapping GPIOINT onto GPIO pin.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TIEN31 | TSSEL31 | | | | | | | TIEN30 | TSSEL30 | | | | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TIEN29 | TSSEL29 | | | | | | | TIEN28 | TSSEL28 | | | | | | |
| 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 |
|---|---|---|---|---|
| 8(n-28)+7 | TIENn | 0 | R/W | This bit enables TINTn.<br>1: Enable.<br>0: Invalid. |
| 8(n-28)+6 to 8(n-28) | TSSELn | All 0 | R/W | This bit selects a source of TINTn.<br>0000000: TINTn = GPIOINT0<br>0000001: TINTn = GPIOINT1<br>~<br>1010001: TINTn = GPIOINT81<br>1010010: Cannot be set.<br>~<br>1111011: Cannot be set.<br>1111100: Cannot be set.<br>1111101: Cannot be set.<br>1111110: Cannot be set.<br>1111111: Cannot be set.<br><br>Assign different sources to TINT0-31. |

**Note:** n = 31 to 28

## 8.6.16 Bus Error Interrupt Status Control Register0 (BEISR0)

This register shows Bus error status.

Refer to LSI Internal Bus section and **Section 8.7.4.1** for bus error interrupt cause.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | BESTAT31 | BESTAT30 | BESTAT29 | BESTAT28 | BESTAT27 | BESTAT26 | BESTAT25 | BESTAT24 | BESTAT23 | BESTAT22 | BESTAT21 | BESTAT20 | BESTAT19 | BESTAT18 | BESTAT17 | BESTAT16 |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | BESTAT15 | BESTAT14 | BESTAT13 | TSTAT12 | BESTAT11 | BESTAT10 | BESTAT9 | BESTAT8 | BESTAT7 | BESTAT6 | BESTAT5 | BESTAT4 | BESTAT3 | BESTAT2 | BESTAT1 | BESTAT0 |
| 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 |
|---|---|---|---|---|
| n | BESTATn | All 0 | R/W | This bit indicates BUSERR_INTn interrupt status.<br>BUSERR_INTn interrupt status can be cleared writing 0 in case BESTAT is 1.<br><br>[Read operation]<br>0: BUSERR_INTn is not detected.<br>1: BUSERR_INTn is detected.<br>[Write operation]<br>• In case BESTATn is 1<br>0: BUSERR_INTn interrupt status is cleared.<br>1: Invalid to write.<br>• In case BESTATn is 0<br>Invalid to write. |

**Note:** n = 31 to 0

## 8.6.17 Bus Error Interrupt Status Control Register1 (BEISR1)

This register shows Bus error status.

Refer to LSI Internal Bus section and **Section 8.7.4.1** for bus error interrupt cause.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | BESTAT44 | BESTAT43 | BESTAT42 | BESTAT41 | BESTAT40 | BESTAT39 | BESTAT38 | BESTAT37 | BESTAT36 | BESTAT35 | BESTAT34 | BESTAT33 | BESTAT32 |
| 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 |
|---|---|---|---|---|
| 31 to 13 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| n-32 | BESTATn | All 0 | R/W | This bit indicates BUSERR_INTn interrupt status.<br>BUSERR_INTn interrupt status can be cleared writing 0 in case BESTAT is 1.<br><br>[Read operation]<br>0: BUSERR_INTn is not detected.<br>1: BUSERR_INTn is detected.<br>[Write operation]<br>• In case BESTATn is 1<br>0: BUSERR_INTn interrupt status is cleared.<br>1: Invalid to write.<br>• In case BESTATn is 0<br>Invalid to write. |

**Note:** n = 44 to 32

## 8.6.18 ECCRAM Error Interrupt Status Control Register0 (EREISR0)

This register shows ECC Error status of ECCRAM0.

Refer to **Section 8.7.4.2** for ECCRAM error interrupt cause.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | OFSTAT7 | OFSTAT6 | OFSTAT5 | OFSTAT4 | OFSTAT3 | OFSTAT2 | OFSTAT1 | OFSTAT0 |
| 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 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | E2STAT7 | E2STAT6 | E2STAT5 | E2STAT4 | E2STAT3 | E2STAT2 | E2STAT1 | E2STAT0 | E1STAT7 | E1STAT6 | E1STAT5 | E1STAT4 | E1STAT3 | E1STAT2 | E1STAT1 | E1STAT0 |
| 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 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 23 to 16 | OFSTATn | All 0 | R/W | This bit indicates ECCRAM0_OFn interrupt status.<br>ECCRAM0_OFn interrupt status can be cleared writing 0 in case OFSTATn is 1.<br><br>[Read operation]<br>0: ECCRAM0_OFn is not detected.<br>1: ECCRAM0_OFn is detected.<br>[Write operation]<br>• In case OFSTATn is 1<br>0: ECCRAM0_OFn interrupt status is cleared.<br>1: Invalid to write.<br>• In case OFSTATn is 0<br>Invalid to write. |
| 15 to 8 | E2STATn | All 0 | R/W | This bit indicates ECCRAM0_2En interrupt status.<br>ECCRAM0_2En interrupt status can be cleared writing 0 in case E2STATn is 1.<br><br>[Read operation]<br>0: ECCRAM0_2En is not detected.<br>1: ECCRAM0_2En is detected.<br>Write operation:<br>• In case E2STATn is 1<br>0: ECCRAM0_2En interrupt status is cleared.<br>1: Invalid to write.<br>• In case E2STATn is 0<br>Invalid to write. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 to 0 | E1STATn | All 0 | R/W | This bit indicates ECCRAM0_1En interrupt status.<br>ECCRAM0_1En interrupt status can be cleared writing 0 in case E1STATn is 1.<br><br>[Read operation]<br>0: ECCRAM0_1En is not detected.<br>1: ECCRAM0_1En is detected.<br>[Write operation]<br>• In case E1STATn is 1<br>0: ECCRAM0_1En interrupt status is cleared.<br>1: Invalid to write.<br>• In case E1STATn is 0<br>Invalid to write. |

**Note:** n = 7 to 0

## 8.6.19 ECCRAM Error Interrupt Status Control Register1 (EREISR1)

This register shows ECC Error status of ECCRAM1.

Refer to **Section 8.7.4.2** for ECCRAM error interrupt cause.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | OFSTAT7 | OFSTAT6 | OFSTAT5 | OFSTAT4 | OFSTAT3 | OFSTAT2 | OFSTAT1 | OFSTAT0 |
| 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 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | E2STAT7 | E2STAT6 | E2STAT5 | E2STAT4 | E2STAT3 | E2STAT2 | E2STAT1 | E2STAT0 | E1STAT7 | E1STAT6 | E1STAT5 | E1STAT4 | E1STAT3 | E1STAT2 | E1STAT1 | E1STAT0 |
| 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 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 23 to 16 | OFSTATn | All 0 | R/W | This bit indicates ECCRAM1_OFn interrupt status.<br>ECCRAM1_OFn interrupt status can be cleared writing 0 in case OFSTATn is 1.<br><br>[Read operation]<br>0: ECCRAM1_OFn is not detected.<br>1: ECCRAM1_OFn is detected.<br>[Write operation]<br>• In case OFSTATn is 1<br>0: ECCRAM1_OFn interrupt status is cleared.<br>1: Invalid to write.<br>• In case OFSTATn is 0<br>Invalid to write. |
| 15 to 8 | E2STATn | All 0 | R/W | This bit indicates ECCRAM1_2En interrupt status.<br>ECCRAM1_2En interrupt status can be cleared writing 0 in case E2STATn is 1.<br><br>[Read operation]<br>0: ECCRAM1_2En is not detected.<br>1: ECCRAM1_2En is detected.<br>[Write operation]<br>• In case E2STATn is 1<br>0: ECCRAM1_2En interrupt status is cleared.<br>1: Invalid to write.<br>• In case E2STATn is 0<br>Invalid to write. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 to 0 | E1STATn | All 0 | R/W | This bit indicates ECCRAM1_1En interrupt status.<br>ECCRAM1_1En interrupt status can be cleared writing 0 in case E1STATn is 1.<br><br>[Read operation]<br>0: ECCRAM1_1En is not detected.<br>1: ECCRAM1_1En is detected.<br>[Write operation]<br>• In case E1STATn is 1<br>0: ECCRAM1_1En interrupt status is cleared.<br>1: Invalid to write.<br>• In case E1STATn is 0<br>Invalid to write. |

**Note:** n = 7 to 0

## 8.7 Operation

### 8.7.1 NMI Interrupt

NMI interrupt is the highest priority interrupt by default even though it can be masked.

NMI is not treated as NMI exception.

[Setting]

- A noise filter function of NMI pin can be enabled by GPIO register setting.
  Refer to **Section 41.4.1, Operation for GPIO Function** for detail.
- NMI pin input is detected by edge. A detection method is set by NTSEL bit of NMI Interrupt Type Selection Register (NITSR) in IA55. It can be selected “Rising-edge detection” or “Falling-edge detection”.

[Status Control]

- NMI interrupt request can be confirmed by reading NSTAT bit of NMI Status Control Register (NSCR) in IA55. The interrupt request can be cleared by writing 0 to NSTAT bit.
- A signal level (High or Low) of NMI pin can be confirmed by reading NSMON bit of NMI Status Control Register (NSCR) in IA55.

### 8.7.2 IRQ Interrupt

IRQ interrupt is the interrupt from IRQ0-7 input pins.

[Setting]

- Using IRQ pins as interrupt must be mapped GPIO pins onto IRQ pins by GPIO register setting.
  Refer to **Section 41.4.2, Operation for Peripheral Function** for detail.
- A noise filter function of IRQ pins can be enabled by GPIO register setting.
  Refer to **Section 41.4.1, Operation for GPIO Function** for detail.
- IRQ pin inputs are detected by a level or an edge. A detection method is set by ITSEL bit of IRQ Interrupt Type Selection Register (IITSR) in IA55. It can be selected “Low-level detection”, “Falling-edge detection”, “Rising-edge detection” or “Falling/Rising-edge detection”.

[Status Control]

- IRQ interrupt requests can be confirmed by reading ISTATn bit of IRQ Status Control Register (ISCR) in IA55.
- In case selected a level detection in IITSR, the interrupt requests are cleared when the interrupt sources de-assert the interrupts.
- In case selected an edge detection in IITSR, the interrupt requests can be cleared by writing 0 to ISTATn bit.

### 8.7.3 GPIO Interrupt (TINT)

GPIO interrupt is the interrupt using GPIO pins as external interrupt input pins.

[Setting]

- When using GPIO pins as interrupts, assign GPIO pins as external interrupt input pins (GPIOINT0-81) by GPIO register setting.
  Refer to **Section 41.4.1, Operation for GPIO Function** for details.
  Refer to **Table 8.4** for mapping GPIO pins onto GPIOINT0-81.
- The noise filter function of GPIO pins assigned to TINT can be enabled by GPIO register setting.
  Refer to **Section 41.4.1, Operation for GPIO Function** for details.
- Assign any 32 interrupt sources to TINT from GPIOINT0-81 by TINT Source Selection Register (TSSR0-7) in IA55.

  *Note*: Even if the TINT is assigned by TSSR0-7, no interrupt request is generated unless it is assigned to an external pin interrupt by GPIO registers.
- TINT inputs are detected by a level or an edge. A detection method is set by TTSEL bit of TINT Interrupt Type Selection Register (TITSR0/1) in IA55. It can be selected “Rising-edge detection”, “Falling-edge detection”, “High-level detection” or “Low-level detection”.

[Status Control]

- TINT interrupt requests can be confirmed by reading TSTATn bit of TINT Status Control Register (TSCR) in IA55.
- In case selected a level detection in TITSR0/1, the interrupt requests are cleared when the interrupt sources de-assert the interrupts.
- In case selected an edge detection in TITSR0/1, the interrupt requests can be cleared by writing 0 to TSTATn bit.

Table 8.4 GPIOINT Mapping

| No. | GPIO Pin | GPIOINT |
|---|---|---|
| 1 | P0_0 | GPIOINT0 |
| 2 | P0_1 | GPIOINT1 |
| 3 | P0_2 | GPIOINT2 |
| 4 | P0_3 | GPIOINT3 |
| 5 | P1_0 | GPIOINT4 |
| 6 | P1_1 | GPIOINT5 |
| 7 | P1_2 | GPIOINT6 |
| 8 | P1_3 | GPIOINT7 |
| 9 | P1_4 | GPIOINT8 |
| 10 | P2_0 | GPIOINT9 |
| 11 | P2_1 | GPIOINT10 |
| 12 | P2_2 | GPIOINT11 |
| 13 | P2_3 | GPIOINT12 |
| 14 | P3_0 | GPIOINT13 |
| 15 | P3_1 | GPIOINT14 |
| 16 | P3_2 | GPIOINT15 |
| 17 | P3_3 | GPIOINT16 |
| 18 | P4_0 | GPIOINT17 |
| 19 | P4_1 | GPIOINT18 |
| 20 | P4_2 | GPIOINT19 |
| 21 | P4_3 | GPIOINT20 |
| 22 | P4_4 | GPIOINT21 |
| 23 | P4_5 | GPIOINT22 |
| 24 | P5_0 | GPIOINT23 |
| 25 | P5_1 | GPIOINT24 |
| 26 | P5_2 | GPIOINT25 |
| 27 | P5_3 | GPIOINT26 |
| 28 | P5_4 | GPIOINT27 |
| 29 | P6_0 | GPIOINT28 |
| 30 | P6_1 | GPIOINT29 |
| 31 | P6_2 | GPIOINT30 |
| 32 | P6_3 | GPIOINT31 |
| 33 | P6_4 | GPIOINT32 |
| 34 | P7_0 | GPIOINT33 |
| 35 | P7_1 | GPIOINT34 |
| 36 | — | GPIOINT35 |
| 37 | — | GPIOINT36 |
| 38 | — | GPIOINT37 |
| 39 | P8_0 | GPIOINT38 |
| 40 | P8_1 | GPIOINT39 |
| 41 | P8_2 | GPIOINT40 |
| 42 | P8_3 | GPIOINT41 |
| 43 | P9_0 | GPIOINT42 |
| 44 | P9_1 | GPIOINT43 |
| 45 | P9_2 | GPIOINT44 |
| 46 | P9_3 | GPIOINT45 |
| 47 | P9_4 | GPIOINT46 |
| 48 | P10_0 | GPIOINT47 |
| 49 | P10_1 | GPIOINT48 |
| 50 | P10_2 | GPIOINT49 |
| 51 | P10_3 | GPIOINT50 |
| 52 | — | GPIOINT51 |
| 53 | P11_0 | GPIOINT52 |
| 54 | P11_1 | GPIOINT53 |
| 55 | P11_2 | GPIOINT54 |
| 56 | P11_3 | GPIOINT55 |
| 57 | P12_0 | GPIOINT56 |
| 58 | P12_1 | GPIOINT57 |
| 59 | P20_0 | GPIOINT58 |
| 60 | P20_1 | GPIOINT59 |
| 61 | P20_2 | GPIOINT60 |
| 62 | P20_3 | GPIOINT61 |
| 63 | P20_4 | GPIOINT62 |
| 64 | P20_5 | GPIOINT63 |
| 65 | P21_0 | GPIOINT64 |
| 66 | P21_1 | GPIOINT65 |
| 67 | P21_2 | GPIOINT66 |
| 68 | P22_0 | GPIOINT67 |
| 69 | P22_1 | GPIOINT68 |
| 70 | P22_2 | GPIOINT69 |
| 71 | P22_3 | GPIOINT70 |
| 72 | P22_4 | GPIOINT71 |
| 73 | P23_0 | GPIOINT72 |
| 74 | P23_1 | GPIOINT73 |
| 75 | — | GPIOINT74 |
| 76 | — | GPIOINT75 |
| 77 | — | GPIOINT76 |
| 78 | — | GPIOINT77 |
| 79 | — | GPIOINT78 |
| 80 | — | GPIOINT79 |
| 81 | — | GPIOINT80 |
| 82 | — | GPIOINT81 |

## 8.7.4 Internal Interrupt

### 8.7.4.1 Bus Error Interrupt

Bus error interrupts (BUSERR_INT0-41) generated by system bus are integrated into one interrupt (BUS_ERR_INT) by IA55 and the integrated interrupt is notified to CPU.

[Status Control]

- Bus error interrupt requests can be confirmed by reading BUS Error Interrupt Status Control Register (BEISR0/1) in IA55.
  Refer to **Table 8.5** and **Table 8.6** for mapping BESTAT0-41 onto bus error interrupt causes.
- The interrupt requests can be cleared by writing 0 to BESTAT0-41 bit.

Table 8.5 BEISR0 Bus Error Mapping

| BEISR0 | Interrupt | Description |
|---|---|---|
| BESTAT0 | BUSERR_INT0 | Bus Write Error Interrupt for CA55 |
| BESTAT1 | BUSERR_INT1 | Bus Read Error Interrupt for CA55 |
| BESTAT2 | BUSERR_INT2 | Reserved |
| BESTAT3 | BUSERR_INT3 | Reserved |
| BESTAT4 | BUSERR_INT4 | Reserved |
| BESTAT5 | BUSERR_INT5 | Reserved |
| BESTAT6 | BUSERR_INT6 | Bus Write Error Interrupt for DMAC_NS |
| BESTAT7 | BUSERR_INT7 | Bus Read Error Interrupt for DMAC_NS |
| BESTAT8 | BUSERR_INT8 | Bus Write Error Interrupt for CST_ETR |
| BESTAT9 | BUSERR_INT9 | Bus Read Error Interrupt for CST_ETR |
| BESTAT10 | BUSERR_INT10 | Bus Write Error Interrupt for CST_AP |
| BESTAT11 | BUSERR_INT11 | Bus Read Error Interrupt for CST_AP |
| BESTAT12 | BUSERR_INT12 | Reserved |
| BESTAT13 | BUSERR_INT13 | Reserved |
| BESTAT14 | BUSERR_INT14 | Reserved |
| BESTAT15 | BUSERR_INT15 | Reserved |
| BESTAT16 | BUSERR_INT16 | Bus Write Error Interrupt for SDHI Ch0 |
| BESTAT17 | BUSERR_INT17 | Bus Read Error Interrupt for SDHI Ch0 |
| BESTAT18 | BUSERR_INT18 | Reserved |
| BESTAT19 | BUSERR_INT19 | Reserved |
| BESTAT20 | BUSERR_INT20 | Bus Write Error Interrupt for DRW1 |
| BESTAT21 | BUSERR_INT21 | Bus Read Error Interrupt for DRW1 |
| BESTAT22 | BUSERR_INT22 | Bus Write Error Interrupt for DRW2 |
| BESTAT23 | BUSERR_INT23 | Bus Read Error Interrupt for DRW2 |
| BESTAT24 | BUSERR_INT24 | Bus Write Error Interrupt for USB Ch0 Host |
| BESTAT25 | BUSERR_INT25 | Bus Read Error Interrupt for USB Ch0 Host |
| BESTAT26 | BUSERR_INT26 | Reserved |
| BESTAT27 | BUSERR_INT27 | Reserved |
| BESTAT28 | BUSERR_INT28 | Bus Write Error Interrupt for USB Ch0 Function |
| BESTAT29 | BUSERR_INT29 | Bus Read Error Interrupt for USB Ch0 Function |
| BESTAT30 | BUSERR_INT30 | Reserved |
| BESTAT31 | BUSERR_INT31 | Reserved |

Table 8.6 BEISR1 Bus Error Mapping

| BEISR1 | Interrupt | Description |
|---|---|---|
| BESTAT32 | BUSERR_INT32 | Reserved |
| BESTAT33 | BUSERR_INT33 | Reserved |
| BESTAT34 | BUSERR_INT34 | Bus Write Error Interrupt for FCPVD |
| BESTAT35 | BUSERR_INT35 | Bus Read Error Interrupt for FCPVD |
| BESTAT36 | BUSERR_INT36 | Bus Write Error Interrupt for MIPI DSI LINK |
| BESTAT37 | BUSERR_INT37 | Bus Read Error Interrupt for MIPI DSI LINK |
| BESTAT38 | BUSERR_INT38 | Reserved |
| BESTAT39 | BUSERR_INT39 | Reserved |
| BESTAT40 | BUSERR_INT40 | Reserved |
| BESTAT41 | BUSERR_INT41 | Reserved |
| BESTAT42 | — | Reserved |
| BESTAT43 | — | Reserved |
| BESTAT44 | — | Reserved |

### 8.7.4.2 ECCRAM Error Interrupt

ECCRAM error interrupts generated by On-Chip RAM (ECCRAM0, ECCRAM1) are integrated into one interrupt from eight interrupts by IA55 and the integrated interrupt is notified to CPU.

Integrated 1bit error interrupt: EC7TIE1_0, EC7TIE1_1
Integrated 2bit error interrupt: EC7TIE2_0, EC7TIE2_1
Integrated overflow interrupt: EC7TIOVF_0, EC7TIOVF_1

[Status Control]

- ECCRAM error interrupt requests can be confirmed by reading ECCRAM Error Interrupt Status Control Register (EREISR0/1) in IA55.
  Refer to **Table 8.7** and **Table 8.8** for mapping E1STAT0-7, E2STAT0-7 and OFSTAT0-7 onto ECCRAM error interrupt causes.
- The interrupt requests can be cleared by writing 0 to E1STAT0-7, E2STAT0-7 and OFSTAT0-7 bit.

Table 8.7 ECCRAM0 ECC Error Mapping

| EREISR0 | Interrupt | Description | Integrated Interrupt |
|---|---|---|---|
| E1STAT0 | ECCRAM0_1E0 | ECC 1bit error interrupt bit[31:0] | EC7TIE1_0 |
| E1STAT1 | ECCRAM0_1E1 | ECC 1bit error interrupt bit[63:32] | |
| E1STAT2 | ECCRAM0_1E2 | ECC 1bit error interrupt bit[95:64] | |
| E1STAT3 | ECCRAM0_1E3 | ECC 1bit error interrupt bit[127:96] | |
| E1STAT4 | ECCRAM0_1E4 | ECC 1bit error interrupt bit[159:128] | |
| E1STAT5 | ECCRAM0_1E5 | ECC 1bit error interrupt bit[191:160] | |
| E1STAT6 | ECCRAM0_1E6 | ECC 1bit error interrupt bit[223:192] | |
| E1STAT7 | ECCRAM0_1E7 | ECC 1bit error interrupt bit[255:224] | |
| E2STAT0 | ECCRAM0_2E0 | ECC 2bit error interrupt bit[31:0] | EC7TIE2_0 |
| E2STAT1 | ECCRAM0_2E1 | ECC 2bit error interrupt bit[63:32] | |
| E2STAT2 | ECCRAM0_2E2 | ECC 2bit error interrupt bit[95:64] | |
| E2STAT3 | ECCRAM0_2E3 | ECC 2bit error interrupt bit[127:96] | |
| E2STAT4 | ECCRAM0_2E4 | ECC 2bit error interrupt bit[159:128] | |
| E2STAT5 | ECCRAM0_2E5 | ECC 2bit error interrupt bit[191:160] | |
| E2STAT6 | ECCRAM0_2E6 | ECC 2bit error interrupt bit[223:192] | |
| E2STAT7 | ECCRAM0_2E7 | ECC 2bit error interrupt bit[255:224] | |
| OFSTAT0 | ECCRAM0_OF0 | ECC error overflow interrupt bit[31:0] | EC7TIOVF_0 |
| OFSTAT1 | ECCRAM0_OF1 | ECC error overflow interrupt bit[63:32] | |
| OFSTAT2 | ECCRAM0_OF2 | ECC error overflow interrupt bit[95:64] | |
| OFSTAT3 | ECCRAM0_OF3 | ECC error overflow interrupt bit[127:96] | |
| OFSTAT4 | ECCRAM0_OF4 | ECC error overflow interrupt bit[159:128] | |
| OFSTAT5 | ECCRAM0_OF5 | ECC error overflow interrupt bit[191:160] | |
| OFSTAT6 | ECCRAM0_OF6 | ECC error overflow interrupt bit[223:192] | |
| OFSTAT7 | ECCRAM0_OF7 | ECC error overflow interrupt bit[255:224] | |

Table 8.8 ECCRAM1 ECC Error Mapping

| EREISR1 | Interrupt | Description | Integrated Interrupt |
|---|---|---|---|
| E1STAT0 | ECCRAM1_1E0 | ECC 1bit error interrupt bit[31:0] | EC7TIE1_1 |
| E1STAT1 | ECCRAM1_1E1 | ECC 1bit error interrupt bit[63:32] | |
| E1STAT2 | ECCRAM1_1E2 | ECC 1bit error interrupt bit[95:64] | |
| E1STAT3 | ECCRAM1_1E3 | ECC 1bit error interrupt bit[127:96] | |
| E1STAT4 | ECCRAM1_1E4 | ECC 1bit error interrupt bit[159:128] | |
| E1STAT5 | ECCRAM1_1E5 | ECC 1bit error interrupt bit[191:160] | |
| E1STAT6 | ECCRAM1_1E6 | ECC 1bit error interrupt bit[223:192] | |
| E1STAT7 | ECCRAM1_1E7 | ECC 1bit error interrupt bit[255:224] | |
| E2STAT0 | ECCRAM1_2E0 | ECC 2bit error interrupt bit[31:0] | EC7TIE2_1 |
| E2STAT1 | ECCRAM1_2E1 | ECC 2bit error interrupt bit[63:32] | |
| E2STAT2 | ECCRAM1_2E2 | ECC 2bit error interrupt bit[95:64] | |
| E2STAT3 | ECCRAM1_2E3 | ECC 2bit error interrupt bit[127:96] | |
| E2STAT4 | ECCRAM1_2E4 | ECC 2bit error interrupt bit[159:128] | |
| E2STAT5 | ECCRAM1_2E5 | ECC 2bit error interrupt bit[191:160] | |
| E2STAT6 | ECCRAM1_2E6 | ECC 2bit error interrupt bit[223:192] | |
| E2STAT7 | ECCRAM1_2E7 | ECC 2bit error interrupt bit[255:224] | |
| OFSTAT0 | ECCRAM1_OF0 | ECC error overflow interrupt bit[31:0] | EC7TIOVF_1 |
| OFSTAT1 | ECCRAM1_OF1 | ECC error overflow interrupt bit[63:32] | |
| OFSTAT2 | ECCRAM1_OF2 | ECC error overflow interrupt bit[95:64] | |
| OFSTAT3 | ECCRAM1_OF3 | ECC error overflow interrupt bit[127:96] | |
| OFSTAT4 | ECCRAM1_OF4 | ECC error overflow interrupt bit[159:128] | |
| OFSTAT5 | ECCRAM1_OF5 | ECC error overflow interrupt bit[191:160] | |
| OFSTAT6 | ECCRAM1_OF6 | ECC error overflow interrupt bit[223:192] | |
| OFSTAT7 | ECCRAM1_OF7 | ECC error overflow interrupt bit[255:224] | |

## 8.8 Usage Note

### 8.8.1 Precaution when use the peripheral modules which can initiate DMA Controller.

Some on-chip peripheral modules use the same signal both for an interrupt request and for a DMA transfer request. Refer to **Section 14.4.25, DMA Extension Resource Selectors 0 to 7** for detail.

If such a module is selected by a DMARSn register, the signal works as a DMA transfer request signal and interrupt requests to the interrupt controller are masked.

To enable the interrupt, clear the setting of DMARSn (set all MID[7:0] and RID[1:0] to 0).
In addition, need following register settings even if DMA controller is not used.

- Set CPG_CLKON_DMAC_REG register to supply a clock for DMA Controller.
  Refer to **Section 7.2.4, Clock Control Register DMAC_REG** for register detail.
- Set CPG_RST_DMAC register to release a reset for DMA Controller.
  Refer to **Section 7.2.4, Reset Control Register DMAC** for register detail.

### 8.8.2 Clear Timing of Interrupt Cause

Clear the interrupt cause flag in the interrupt exception service routine. It takes time from clearing the interrupt cause flag until the interrupt cause to the CPU is actually removed. Therefore, to prevent the interrupt cause that should have been cleared from being accidentally accepted again, the interrupt cause flag is read after clearing, and then the return instruction is executed.

### 8.8.3 Precaution when Changing Interrupt Settings

To change the NMI, IRQ, TINT interrupt settings, mask the interrupts and change the settings.

- When changing the noise filter settings.
- When switching the GPIO pins to IRQ or GPIOINT.
- When changing the source of TINT.
- When changing the interrupt detection method.

When changing the NMI, IRQ, TINT interrupt detection method to the edge type, perform the process of clearing the interrupt status after changing the setting.

- NMI: Write 0 to the NSTAT bit of NSCR.
- IRQ: Write 0 to the ISTATn bit of ISCR.
- TINT: Write 0 to the TSTATn bit of TSCR.

# 9. DDR3L SDRAM Memory Controller (MEMC)

DDR3L SDRAM Memory Controller (MEMC) is External Bus Controller for DDR3L SDRAM (DDR). This block supports up to DDR3L-1600 SDRAM. Interface bus width is 16 bits. In line ECC can be supported.
For setup and configuration to this block, please use and refer to the linux software package of this LSI including DDR setup codes.

## 9.1 Features

This unit consists of MC (Memory controller) and PHY. It supports the following specifications.

- DDR3L (JEDEC STANDARD JESD79-3F, JESD79-3-1A.01)

**Figure 9.1** shows features of this block.

Table 9.1 DDR3L SDRAM Memory Controller (MEMC) features

| Feature | Description |
|---|---|
| DRAM IF | • DDR3L: 800 to 1600 Mbps (400 to 800 MHz)<br>• Width: 16 bits (Full (16 bits) or Half (8 bits) data width mode)<br>• Rank: 1, 2 |
| MC (Memory controller) | • Fully pipelined command, read and write data interfaces to the controller.<br>• Advanced bank look-ahead features for high memory throughput.<br>• A programmable register interface to control memory parameters and protocols including auto pre-charge.<br>• Full initialization of memory on controller reset.<br>• ECC function for single bit and double bit error reporting, single bit error correction, and programmable removal of ECC storage. |
| PHY | • Write Leveling (Deskew CK vs Write-DQS)<br>• Bit Leveling (Deskew DQS vs DQ/DM, Read and Write)<br>• Vref Training (Read and Write)<br>• Self Calibration Logic (Deskew Read vs Internal clock)<br>• Command/Address Swizzling |

The diagrams of this unit are follows.

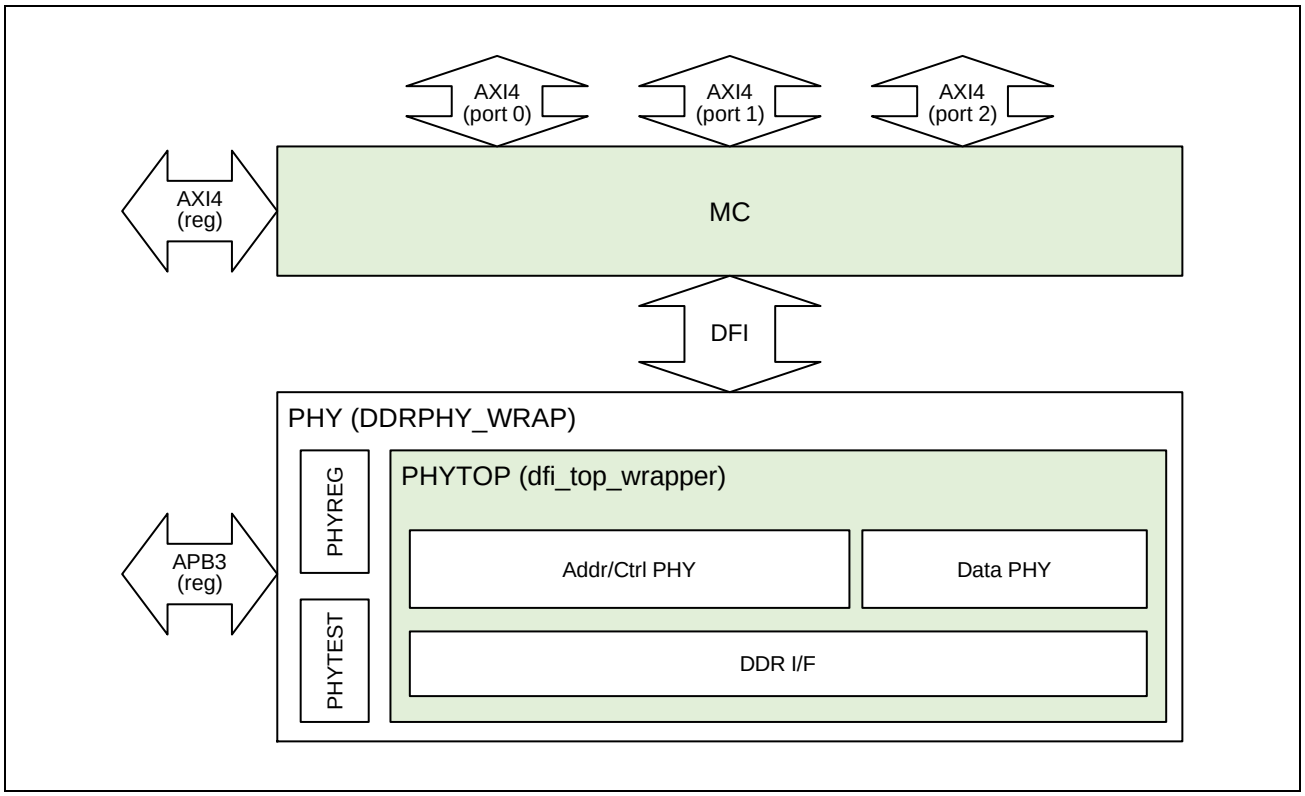



Figure 9.1 Block Diagram

This unit has the following restrictions.

**DDR3L**

- BC4 is not supported.
- Interleave is not supported.

## 9.2 Interface

### 9.2.1 Power, Ground

**Table 9.2** shows the power and ground pins of this unit.

Table 9.2 Power, Ground

| Name | I/O | Description |
|---|---|---|
| DDR_VDDQ1 | Supply | Power supply for I/O |
| DDR_VDDQ2 | Supply | Power supply for DDR |
| VDD | Supply | Power supply for core |
| VSS | Ground | Ground for core |

### 9.2.2 External Pins

**Table 9.3** shows the external pins for DDR interface.

These pins cannot be used for customers, only for SiP, except for DDR_CLK_P, DDR_CLK_N, and DDR_CALIBRATION pins.

Table 9.3 External pin list

| External pin | I/O | Description |
|---|---|---|
| DDR_CLK_P | O | DDR Clock (Pos)<br>Should be connected to an external resister and capacitor. |
| DDR_CLK_N | O | DDR Clock (Neg)<br>Should be connected to an external resister and capacitor. |
| DDR_RESET# | O | DDR Active Low Asynchronous Reset |
| DDR_CKE | O | DDR Clock Enable |
| DDR_CS0# | O | DDR Address or Command |
| DDR_CS1# | O | DDR Address or Command |
| DDR_RAS# | O | DDR Address or Command |
| DDR_CAS# | O | DDR Address or Command |
| DDR_WE# | O | DDR Address or Command |
| DDR_BA0 to DDR_BA2 | O | DDR Address or Command |
| DDR_ADDR0 to DDR_ADDR15 | O | DDR Address or Command |
| DDR_ODT0 | O | DDR On Die Termination |
| DDR_ODT1 | O | DDR On Die Termination |
| DDR_DQS0_P | IO | DDR Data Strobe (Pos) |
| DDR_DQS0_N | IO | DDR Data Strobe (Neg) |
| DDR_DQS1_P | IO | DDR Data Strobe (Pos) |
| DDR_DQS1_N | IO | DDR Data Strobe (Neg) |
| DDR_DQ0 to DDR_DQ15 | IO | DDR Data |
| DDR_DM0 | O | DDR Data Mask |
| DDR_DM1 | O | DDR Data Mask |
| DDR_CALIBRATION | IO | Should be connected to an external 240-ohm resistor. |

Note: Grayed terminals are internal pins for SiP package.

## 9.3 Usage Note

This unit has the following restrictions.

Table 9.4

| Item | Restriction |
|---|---|
| Address map | It has the area (address) that access is prohibited, because it is used by the periodic training. It is as below.<br><br>DRAM address:<br>BA = 0, ROW = H'0, COL = H'000 to H'00F<br>System address:<br>H'040000000 to H'04000001F (if DDR3L) |
| Period of DRAM inaccessibility | It has the period of DRAM inaccessibility to execute the periodic training. It depends only on the Bit Rate, and not on the kind of DRAM. It is belows:<br>1600Mbps: 5.0 µsec every 1000 msec<br>1333Mbps: 1.5 µsec every 1000 msec<br>1066Mbps: 1.875 µsec every 1000 msec<br>800Mbps: 2.5 µsec every 1000 msec |

### 9.3.1 Power On

The power on sequence for the DDR_VDDQ and VDD domains is such that either the 2 voltages should be ramped up together or VDD should be ramped up prior to DDR_VDDQ in order to prevent voltage stress in the DDR IOs.

# 10. On-chip RAM

This LSI has an on-chip RAM for work area. These memory units can be used to store instructions or data. The operation and write access to the on-chip RAM can be enabled or disabled through the RAM enable bit (VCEN in SYS_RAMn_EN register) (n = 0-1) and RAM write enable bit (VLWEN in SYS_RAMn_EN register) (n = 0-1).

For details on the register, refer to the **Section 6.3** in chapter SYSC.

## 10.1 Features

### ■ Memory size

128 Kbytes (SRAM ACPU: 64 Kbytes, SRAM MCPU: 64 Kbytes)

### ■ Ports

On-chip RAM has one read and write port and is connected to the AXI bus.

### ■ Method of arbitration

When the same port of the on-chip RAM is accessed from different masters simultaneously, the AXI bus performs arbitration in round-robin mode.

### ■ Number of access cycles

The number of cycles for access to read or write is one cycle of SRAM_ACPU_ACLK (P1ϕ).

The number of cycles for access to read or write is one cycle of SRAM_MCPU_ACLK (I2ϕ).

# 11. Error Correcting Code (ECC) for On-Chip RAM

## 11.1 Overview

This LSI chip has two 64-Kbyte areas of on-chip RAM for use as the working area.

Each of on-chip RAM incorporates an ECC function, which allows the detection and correction of 1-bit errors and the detection of 2-bit errors. When a 1-bit or 2-bit error occurs, the address where the error occurred can be stored in an ECC error address register.

The data bus width is 256 bits. This is divided into eight channels as shown in **Figure 11.1**, with detection handled in each 32-bit channel.

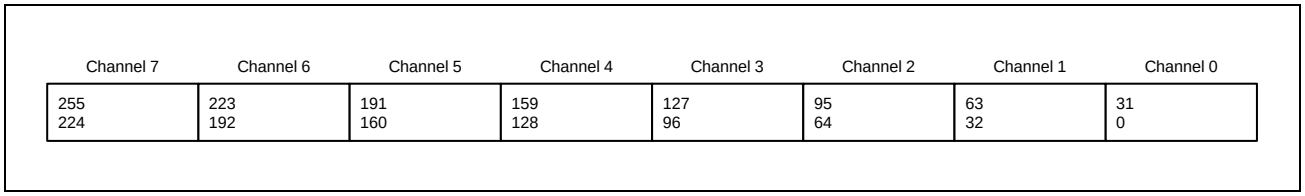



Figure 11.1 Division for ECC Processing

### 11.1.1 Features

- Error detection and correction
  - Detection and correction of 1-bit errors and detection of 2-bit errors are possible.
  - An error involving 3 or more bits cannot be correctly handled and may result in erroneous detection and correction.
- Control of error detection and correction
  - Error detection and correction, and correction of 1-bit errors can respectively be enabled or disabled.
- Capture of the addresses where errors occurred
  - When a 1-bit or 2-bit error is detected, the address where the error was encountered is stored in an ECC error address register. Each channel has eight ECC error address registers.
- Interrupt request
  - An interrupt request can be generated on the detection of a 1-bit error (selectable as enabled or disabled).
  - An interrupt request can be generated on the detection of a 2-bit error (selectable as enabled or disabled).
  - An overflow interrupt is generated from each channel on detection of an error when all ECC error address registers contain values.

## 11.2 Register Configuration

**Table 11.1** and **Table 11.2** respectively show the base register and base address. The addresses of the ECC registers are represented by offsets from the base address.

Table 11.1 Base Register

| Base Register Name | On-Chip RAM | Register Abbreviation |
|---|---|---|
| <REG_base> | SRAM ACPU | ECCRAM0 |
| | SRAM MCPU | ECCRAM1 |

Table 11.2 Base Register

| Base Register Name | On-Chip RAM | Register Abbreviation |
|---|---|---|
| <ADR_base> | SRAM ACPU | H'0_1108_0000 (Overall Address Space) |
| | SRAM MCPU | H'0_1109_0000 (Overall Address Space) |

**Table 11.3** lists the ECC registers.

Do not attempt access to the on-chip RAM areas while the registers listed in **Table 11.3** are being set.

Table 11.3 List of the ECC Registers (1/2)

| Channel | Register Name | Abbreviation | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|---|
| 0 | ECC control register_0 | <REG_base>CTL_0 | R/W | H'0000_0010 | <ADR_base> + H'0000 | 32 |
| | ECC error address register 0_0 | <REG_base>EAD0_0 | R | H'0000_0000 | <ADR_base> + H'0010 | 32 |
| | ECC error address register 1_0 | <REG_base>EAD1_0 | R | H'0000_0000 | <ADR_base> + H'0014 | 32 |
| | ECC error address register 2_0 | <REG_base>EAD2_0 | R | H'0000_0000 | <ADR_base> + H'0018 | 32 |
| | ECC error address register 3_0 | <REG_base>EAD3_0 | R | H'0000_0000 | <ADR_base> + H'001C | 32 |
| | ECC error address register 4_0 | <REG_base>EAD4_0 | R | H'0000_0000 | <ADR_base> + H'0020 | 32 |
| | ECC error address register 5_0 | <REG_base>EAD5_0 | R | H'0000_0000 | <ADR_base> + H'0024 | 32 |
| | ECC error address register 6_0 | <REG_base>EAD6_0 | R | H'0000_0000 | <ADR_base> + H'0028 | 32 |
| | ECC error address register 7_0 | <REG_base>EAD7_0 | R | H'0000_0000 | <ADR_base> + H'002C | 32 |
| 1 | ECC control register_1 | <REG_base>CTL_1 | R/W | H'0000_0010 | <ADR_base> + H'0040 | 32 |
| | ECC error address register 0_1 | <REG_base>EAD0_1 | R | H'0000_0000 | <ADR_base> + H'0050 | 32 |
| | ECC error address register 1_1 | <REG_base>EAD1_1 | R | H'0000_0000 | <ADR_base> + H'0054 | 32 |
| | ECC error address register 2_1 | <REG_base>EAD2_1 | R | H'0000_0000 | <ADR_base> + H'0058 | 32 |
| | ECC error address register 3_1 | <REG_base>EAD3_1 | R | H'0000_0000 | <ADR_base> + H'005C | 32 |
| | ECC error address register 4_1 | <REG_base>EAD4_1 | R | H'0000_0000 | <ADR_base> + H'0060 | 32 |
| | ECC error address register 5_1 | <REG_base>EAD5_1 | R | H'0000_0000 | <ADR_base> + H'0064 | 32 |
| | ECC error address register 6_1 | <REG_base>EAD6_1 | R | H'0000_0000 | <ADR_base> + H'0068 | 32 |
| | ECC error address register 7_1 | <REG_base>EAD7_1 | R | H'0000_0000 | <ADR_base> + H'006C | 32 |
| 2 | ECC control register_2 | <REG_base>CTL_2 | R/W | H'0000_0010 | <ADR_base> + H'0080 | 32 |
| | ECC error address register 0_2 | <REG_base>EAD0_2 | R | H'0000_0000 | <ADR_base> + H'0090 | 32 |
| | ECC error address register 1_2 | <REG_base>EAD1_2 | R | H'0000_0000 | <ADR_base> + H'0094 | 32 |
| | ECC error address register 2_2 | <REG_base>EAD2_2 | R | H'0000_0000 | <ADR_base> + H'0098 | 32 |
| | ECC error address register 3_2 | <REG_base>EAD3_2 | R | H'0000_0000 | <ADR_base> + H'009C | 32 |
| | ECC error address register 4_2 | <REG_base>EAD4_2 | R | H'0000_0000 | <ADR_base> + H'00A0 | 32 |
| | ECC error address register 5_2 | <REG_base>EAD5_2 | R | H'0000_0000 | <ADR_base> + H'00A4 | 32 |
| | ECC error address register 6_2 | <REG_base>EAD6_2 | R | H'0000_0000 | <ADR_base> + H'00A8 | 32 |
| | ECC error address register 7_2 | <REG_base>EAD7_2 | R | H'0000_0000 | <ADR_base> + H'00AC | 32 |
| 3 | ECC control register_3 | <REG_base>CTL_3 | R/W | H'0000_0010 | <ADR_base> + H'00C0 | 32 |
| | ECC error address register 0_3 | <REG_base>EAD0_3 | R | H'0000_0000 | <ADR_base> + H'00D0 | 32 |
| | ECC error address register 1_3 | <REG_base>EAD1_3 | R | H'0000_0000 | <ADR_base> + H'00D4 | 32 |
| | ECC error address register 2_3 | <REG_base>EAD2_3 | R | H'0000_0000 | <ADR_base> + H'00D8 | 32 |
| | ECC error address register 3_3 | <REG_base>EAD3_3 | R | H'0000_0000 | <ADR_base> + H'00DC | 32 |
| | ECC error address register 4_3 | <REG_base>EAD4_3 | R | H'0000_0000 | <ADR_base> + H'00E0 | 32 |
| | ECC error address register 5_3 | <REG_base>EAD5_3 | R | H'0000_0000 | <ADR_base> + H'00E4 | 32 |
| | ECC error address register 6_3 | <REG_base>EAD6_3 | R | H'0000_0000 | <ADR_base> + H'00E8 | 32 |
| | ECC error address register 7_3 | <REG_base>EAD7_3 | R | H'0000_0000 | <ADR_base> + H'00EC | 32 |

Table 11.3 List of the ECC Registers (2/2)

| Channel | Register Name | Abbreviation | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|---|
| 4 | ECC control register_4 | <REG_base>CTL_4 | R/W | H'0000_0010 | <ADR_base> + H'0100 | 32 |
| | ECC error address register 0_4 | <REG_base>EAD0_4 | R | H'0000_0000 | <ADR_base> + H'0110 | 32 |
| | ECC error address register 1_4 | <REG_base>EAD1_4 | R | H'0000_0000 | <ADR_base> + H'0114 | 32 |
| | ECC error address register 2_4 | <REG_base>EAD2_4 | R | H'0000_0000 | <ADR_base> + H'0118 | 32 |
| | ECC error address register 3_4 | <REG_base>EAD3_4 | R | H'0000_0000 | <ADR_base> + H'011C | 32 |
| | ECC error address register 4_4 | <REG_base>EAD4_4 | R | H'0000_0000 | <ADR_base> + H'0120 | 32 |
| | ECC error address register 5_4 | <REG_base>EAD5_4 | R | H'0000_0000 | <ADR_base> + H'0124 | 32 |
| | ECC error address register 6_4 | <REG_base>EAD6_4 | R | H'0000_0000 | <ADR_base> + H'0128 | 32 |
| | ECC error address register 7_4 | <REG_base>EAD7_4 | R | H'0000_0000 | <ADR_base> + H'012C | 32 |
| 5 | ECC control register_5 | <REG_base>CTL_5 | R/W | H'0000_0010 | <ADR_base> + H'0140 | 32 |
| | ECC error address register 0_5 | <REG_base>EAD0_5 | R | H'0000_0000 | <ADR_base> + H'0150 | 32 |
| | ECC error address register 1_5 | <REG_base>EAD1_5 | R | H'0000_0000 | <ADR_base> + H'0154 | 32 |
| | ECC error address register 2_5 | <REG_base>EAD2_5 | R | H'0000_0000 | <ADR_base> + H'0158 | 32 |
| | ECC error address register 3_5 | <REG_base>EAD3_5 | R | H'0000_0000 | <ADR_base> + H'015C | 32 |
| | ECC error address register 4_5 | <REG_base>EAD4_5 | R | H'0000_0000 | <ADR_base> + H'0160 | 32 |
| | ECC error address register 5_5 | <REG_base>EAD5_5 | R | H'0000_0000 | <ADR_base> + H'0164 | 32 |
| | ECC error address register 6_5 | <REG_base>EAD6_5 | R | H'0000_0000 | <ADR_base> + H'0168 | 32 |
| | ECC error address register 7_5 | <REG_base>EAD7_5 | R | H'0000_0000 | <ADR_base> + H'016C | 32 |
| 6 | ECC control register_6 | <REG_base>CTL_6 | R/W | H'0000_0010 | <ADR_base> + H'0180 | 32 |
| | ECC error address register 0_6 | <REG_base>EAD0_6 | R | H'0000_0000 | <ADR_base> + H'0190 | 32 |
| | ECC error address register 1_6 | <REG_base>EAD1_6 | R | H'0000_0000 | <ADR_base> + H'0194 | 32 |
| | ECC error address register 2_6 | <REG_base>EAD2_6 | R | H'0000_0000 | <ADR_base> + H'0198 | 32 |
| | ECC error address register 3_6 | <REG_base>EAD3_6 | R | H'0000_0000 | <ADR_base> + H'019C | 32 |
| | ECC error address register 4_6 | <REG_base>EAD4_6 | R | H'0000_0000 | <ADR_base> + H'01A0 | 32 |
| | ECC error address register 5_6 | <REG_base>EAD5_6 | R | H'0000_0000 | <ADR_base> + H'01A4 | 32 |
| | ECC error address register 6_6 | <REG_base>EAD6_6 | R | H'0000_0000 | <ADR_base> + H'01A8 | 32 |
| | ECC error address register 7_6 | <REG_base>EAD7_6 | R | H'0000_0000 | <ADR_base> + H'01AC | 32 |
| 7 | ECC control register_7 | <REG_base>CTL_7 | R/W | H'0000_0010 | <ADR_base> + H'01C0 | 32 |
| | ECC error address register 0_7 | <REG_base>EAD0_7 | R | H'0000_0000 | <ADR_base> + H'01D0 | 32 |
| | ECC error address register 1_7 | <REG_base>EAD1_7 | R | H'0000_0000 | <ADR_base> + H'01D4 | 32 |
| | ECC error address register 2_7 | <REG_base>EAD2_7 | R | H'0000_0000 | <ADR_base> + H'01D8 | 32 |
| | ECC error address register 3_7 | <REG_base>EAD3_7 | R | H'0000_0000 | <ADR_base> + H'01DC | 32 |
| | ECC error address register 4_7 | <REG_base>EAD4_7 | R | H'0000_0000 | <ADR_base> + H'01E0 | 32 |
| | ECC error address register 5_7 | <REG_base>EAD5_7 | R | H'0000_0000 | <ADR_base> + H'01E4 | 32 |
| | ECC error address register 6_7 | <REG_base>EAD6_7 | R | H'0000_0000 | <ADR_base> + H'01E8 | 32 |
| | ECC error address register 7_7 | <REG_base>EAD7_7 | R | H'0000_0000 | <ADR_base> + H'01EC | 32 |

## 11.3 Register Descriptions

### 11.3.1 ECC Control Registers_n (<REG_base>CTL_n) (n = 0 to 7)

Each ECC control register_n is used to control the ECC function in the corresponding channel n shown in **Figure 11.1**.

After writing the initial value to the entire usage area of the on-chip RAM, set the given ECERVF bits to 1 to enable error detection. Be sure to write 01b to the corresponding EMCA[1:0] bits to enable writing to an ECERVF bit.

For writing the initial value to the on-chip RAM areas, refer to **Section 11.5, Initializing the Detection-Usage Areas of the On-Chip RAM**.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ECDEDF7 | ECSEDF7 | ECDEDF6 | ECSEDF6 | ECDEDF5 | ECSEDF5 | ECDEDF4 | ECSEDF4 | ECDEDF3 | ECSEDF3 | ECDEDF2 | ECSEDF2 | ECDEDF1 | ECSEDF1 | ECDEDF0 | ECSEDF0 |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | EMCA[1:0] | | — | — | ECOVF | ECER2C | ECER1C | — | — | ECERVF | EC1ECP | EC2EDIC | EC1EDIC | ECER2F | ECER1F | — |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R | R | R | R/W | R/W | R | R | R/W | R/W | R/W | R/W | R | R | R |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 | ECDEDF7 | 0 | R | ECC 2-Bit Error Detection Flags ECDEDFm (m = 0 to 7) |
| 30 | ECSEDF7 | 0 | R | These bits indicate the occurrence of 2-bit errors in the data output from the RAM while error detection is enabled. The addresses of the errors will be stored in the <REG_base>EADm_n registers. |
| 29 | ECDEDF6 | 0 | R | |
| 28 | ECSEDF6 | 0 | R | |
| 27 | ECDEDF5 | 0 | R | ECC 1-Bit Error Detection Flags ECSEDFm (m = 0 to 7) |
| 26 | ECSEDF5 | 0 | R | These bits indicate the occurrence of 1-bit errors in the data output from the RAM while error detection is enabled. The addresses of the errors will be stored in the <REG_base>EADm_n registers. |
| 25 | ECDEDF4 | 0 | R | |
| 24 | ECSEDF4 | 0 | R | |
| 23 | ECDEDF3 | 0 | R | The ECDEDFm and ECSEDFm bits are cleared to 0 under any of the following conditions.<br>1. Power-on reset<br>2. Writing 1 to the corresponding ECER2C bit<br>3. Setting the ECC error detection enable bit to “disabled” (ECERVF = 0)<br>4. Reset by WDT overflow |
| 22 | ECSEDF3 | 0 | R | |
| 21 | ECDEDF2 | 0 | R | |
| 20 | ECSEDF2 | 0 | R | |
| 19 | ECDEDF1 | 0 | R | |
| 18 | ECSEDF1 | 0 | R | |
| 17 | ECDEDF0 | 0 | R | |
| 16 | ECSEDF0 | 0 | R | |
| 15, 14 | EMCA[1:0] | 00b | R/W | Enable Writing to ECERVF<br>01b: Enabled<br>Others: Disabled<br>These bits are always read as 00b. |
| 13, 12 | — | 00b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |

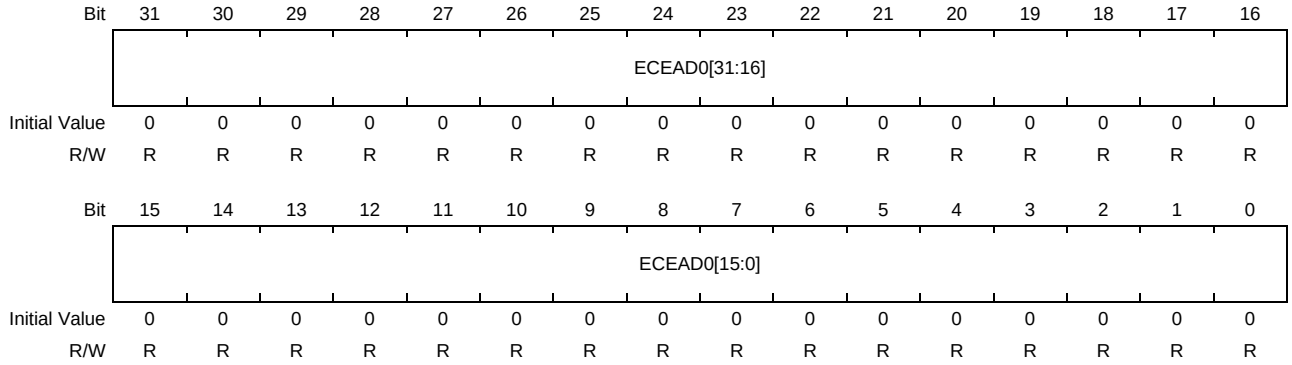
| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 11 | ECOVFF | 0 | R | ECC Overflow Detection Flag<br>This bit is set to 1 and an overflow interrupt is output on detection of an error when the ECC error address registers (<REG_base>EADm_n (m = 0 to 7)) all contain data. An overflow interrupt is output even if the error is detected while the setting of this bit is already 1.<br>This bit is cleared to 0 under any of the following conditions.<br>1. Power-on reset<br>2. Writing 1 to the corresponding ECER2C bit or ECER1C bit<br>3. Setting the ECC error detection enable bit to “disabled” (ECERVF = 0) |
| 10 | ECER2C | 0 | R/W | 2-Bit Error Detection Flag Clear<br>0: The internal state does not change.<br>1: The corresponding 2-bit error detection flag ECER2F is cleared.<br>This bit is always read as 0. |
| 9 | ECER1C | 0 | R/W | 1-Bit Error Detection Flag Clear<br>0: The internal state does not change.<br>1: The corresponding 1-bit error detection flag ECER1F is cleared.<br>This bit is always read as 0. |
| 8, 7 | — | 00b | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 6 | ECERVF | 0 | R/W | ECC Error Detection Enable<br>0: Disables error detection.<br>1: Enables error detection.<br>1-bit error correction (EC1ECP) and interrupt control (EC2EDIC and EC1EDIC) are only possible when error detection is enabled.<br>*Note*: To write to this bit, the corresponding EMCA[1:0] bits must be set to 01b. |
| 5 | EC1ECP | 0 | R/W | 1-Bit Error Correction Enable<br>0: The error is corrected when a 1-bit error is detected.<br>1: The error is not corrected when a 1-bit error is detected. |
| 4 | EC2EDIC | 1 | R/W | 2-Bit Error Detection Interrupt Control<br>0: An interrupt is not generated when a 2-bit error is detected.<br>1: An interrupt is generated when a 2-bit error is detected.<br>The interrupt is only generated when the corresponding 2-bit error detection flag (ECER2F) changes from 0 to 1. |
| 3 | EC1EDIC | 0 | R/W | 1-Bit Error Detection Interrupt Control<br>0: An interrupt is not generated when a 1-bit error is detected.<br>1: An interrupt is generated when a 1-bit error is detected.<br>The interrupt is only generated when the corresponding 1-bit error detection flag (ECER1F) changes from 0 to 1. |
| 2 | ECER2F | 0 | R | 2-Bit Error Detection Flag<br>0: A 2-bit error has not occurred.<br>1: A 2-bit error has occurred.<br>This bit is cleared to 0 under any of the following conditions.<br>1. Power-on reset<br>2. Writing 1 to the corresponding ECER2C bit<br>3. Setting the ECC error detection enable bit to "disabled" (ECERVF = 0) |
| 1 | ECER1F | 0 | R | 1-Bit Error Detection Flag<br>0: A 1-bit error has not occurred.<br>1: A 1-bit error has occurred.<br>This bit is cleared to 0 under any of the following conditions.<br>1. Power-on reset<br>2. Writing 1 to the corresponding ECER1C bit<br>3. Setting the ECC error detection enable bit to "disabled" (ECERVF = 0) |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 0 | — | 0 | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |

### 11.3.2 ECC Error Address Registers m_n (<REG_base>EADm_n) (m = 0 to 7) (n = 0 to 7)

When a 1- or 2-bit error occurs, up to eight error addresses are sequentially stored in the registers from <REG_base>EAD0_n to <REG_base>EAD7_n.

When a 2-bit error is detected while only 1-bit error addresses were stored up to <REG_base>EAD7_n, <REG_base>EAD7_n is overwritten by the address of the 2-bit error. Otherwise, it will not be overwritten. For details, refer to **Section 11.6.4, ECC Error Address Register Overwrite Conditions**.




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 0 | ECEAD0 [31:0] | All 0 | R | These bits hold the address of the (m + 1)th error to have been detected.<br>These bits are cleared to 0 under the following condition:<br>clearing ECER2F or ECER1F in the <REG_base>CTL_n register. |

## 11.4 Initializing the ECC Function

Initialize the ECC by following the procedure below to enable the ECC function.

1. Set the VECCEN bit in the ECCRAM setting register corresponding to the RAM area for which the ECC function is to be enabled to 1.
2. Write the initial value to the entire usage area of the RAM for which the ECC function is to be enabled.
3. Set the ECERVF bits in the ECC control registers_n to 1 to enable error detection. Then, set the respective bits to control 1-bit error correction and interrupts.

For notes on the settings of registers of this module, refer to **Section 11.6.1, Notes on Setting the Registers of This Module**.

## 11.5 Initializing the Detection-Usage Areas of the On-Chip RAM

Before using the ECC function, write the initial value to the entire usage area of the on-chip RAM for which the ECC function is to be enabled. The initial value can be any value. Since the detection of ECC errors is handled in 32-bit units in each channel, the initial value should also be written in 32-bit units.

Enabling error detection without initialization may cause an ECC error due to unintended reading.

### Examples of Operations that May Lead to Errors

1. When writing is to proceed in 8- or 16-bit units, reading the 32-bit value from the given address and then writing the data
2. If the locations are not initialized, an ECC error may occur when the 32-bit value is read. For details on writing in 8- or 16-bit units, refer to **section 11.6.2, 8- or 16-Bit Access**.
3. When the CPU cache is enabled, if a cache miss occurs in writing, the cache being refilled by a 1-line unit
   At this time if data are not initialized, an ECC error may occur because data are read in 1-line (32-byte) units.

## 11.6 Usage Notes

### 11.6.1 Notes on Setting the Registers of This Module

Ensure that none of the bus masters proceeds with access to the on-chip RAM areas while the registers listed in **Table 11.3** are being set up.

### 11.6.2 8- or 16-Bit Access

The ECC function for the on-chip RAM areas handles error detection in 32-bit units. Accordingly, when access is to be in 8- or 16-bit units, the operations proceed as follows.

#### (1) 8- or 16-bit reading

(a) The 32 bits of data are read and the ECC is used to determine whether or not an error is present.

(b) The 8 or 16 bits of data are then returned to the bus master that issued the read request.

#### (2) 8- or 16-bit writing

(a) The 32 bits of data from the address that contains the location for writing are read and the ECC is used to determine whether or not an error is present.

(b) The 32 bits of data that have been read are then overwritten with the 8- or 16-bit data.

(c) The ECC value for the overwritten 32-bit data in (b) is added by writing it to the RAM.

### 11.6.3 Numbers of Cycles for Reading from or Writing to a RAM Area with the ECC Function Enabled

The numbers of cycles for reading from or writing to an on-chip RAM area with the ECC function enabled are as follows.

#### (1) Reading

The number of cycles for reading in 8- or 16-bit and 32-bit units when the ECC function is enabled is one cycle of SRAM_ACPU_ACLK or SRAM_MCPU_ACLK longer than when the ECC function is disabled.

#### (2) Writing

The number of cycles for writing in 8- or 16-bit units when the ECC function is enabled is two cycles of SRAM_ACPU_ACLK or SRAM_MCPU_ACLK longer than when the ECC function is disabled. The number of cycles for writing in 32-bit units when the ECC function is enabled is the same as when the ECC function is disabled.

### 11.6.4 ECC Error Address Register Overwrite Conditions

If a new ECC error occurs with the ECC error address stored up to <REG_base> EAD7_n, <REG_base> EAD7_n may or may not be overwritten. The conditions to be overwritten are shown below.

| Error Address Storage Status Up to <REG_base> EAD7_n | New ECC Error | <REG_base>EAD7_n Overwrite | Overflow Interrupt |
|---|---|---|---|
| Only 1-bit error | 1-bit error | Not overwritten | Occur |
| Only 1-bit error | 2-bit error | Overwritten | Occur |
| Only 2-bit error | 1-bit error | Not overwritten | Occur |
| Only 2-bit error | 2-bit error | Not overwritten | Occur |
| 1-bit error and 2-bit error | 1-bit error | Not overwritten | Occur |
| 1-bit error and 2-bit error | 2-bit error | Not overwritten | Occur |

# 14. Direct Memory Access Controller

The direct memory access controller can be used in place of the CPU to perform high-speed transfers between external memory, on-chip memory, memory-mapped external devices, and on-chip peripheral modules.

This module has controllers that handles non-secure access. Do not assign the same DMA transfer request to non-secure access, respectively.

## 14.1 Features

- Number of channels selectable: 16 channels (CH0 to CH15).
- 4-Gbyte address space (according to the architecture)
- Transfer data size: Byte, 2 bytes, 4 bytes, 8 bytes, 16 bytes, 32 bytes, 64 bytes, and 128 bytes
- Maximum transfer count: $2^{32} - 1$ bytes
- Address mode: Dual address mode
- Transfer requests: Can be selected from the two types of on-chip peripheral module request, and auto request (software trigger)
- Transfer mode: Single transfer mode and block transfer mode are selectable.
- Priority: The channel priority levels within channels 0 to 7 and within channels 8 to 15 are selectable between fixed mode and round-robin mode (the channel priority level between the group of channels 0 to 7 and the group of channels 8 to 15 is round-robin mode).
- Interrupt request: An interrupt request can be sent to the CPU on completion of data transfer (DMA transfer end interrupt per channel) or on occurrence of a transfer error (DMA error interrupt).
- The DMA registers have a continuous execution function that allows the next DMA transfer to be executed continuously by making settings for the next DMA transfer during execution of the current DMA transfer. This continuous execution function can be enabled or disabled independently in each channel.
- Link mode: In this mode, the setting data (descriptor data) located in the memory by the CPU is automatically retrieved by the DMAC, and DMA transfer is performed according to those values.
- Buffer sweep: If an ongoing DMA transfer is forced to end, the data already retrieved into the buffer can be output before DMA transfer ends.
- Interval: A specific DMA transfer interval can be specified to adjust the bus occupancy.

## 14.2 Input/Output Pins

Requests from external pins are not supported.

## 14.3 Register Configuration

The register configuration is shown in the figure below.

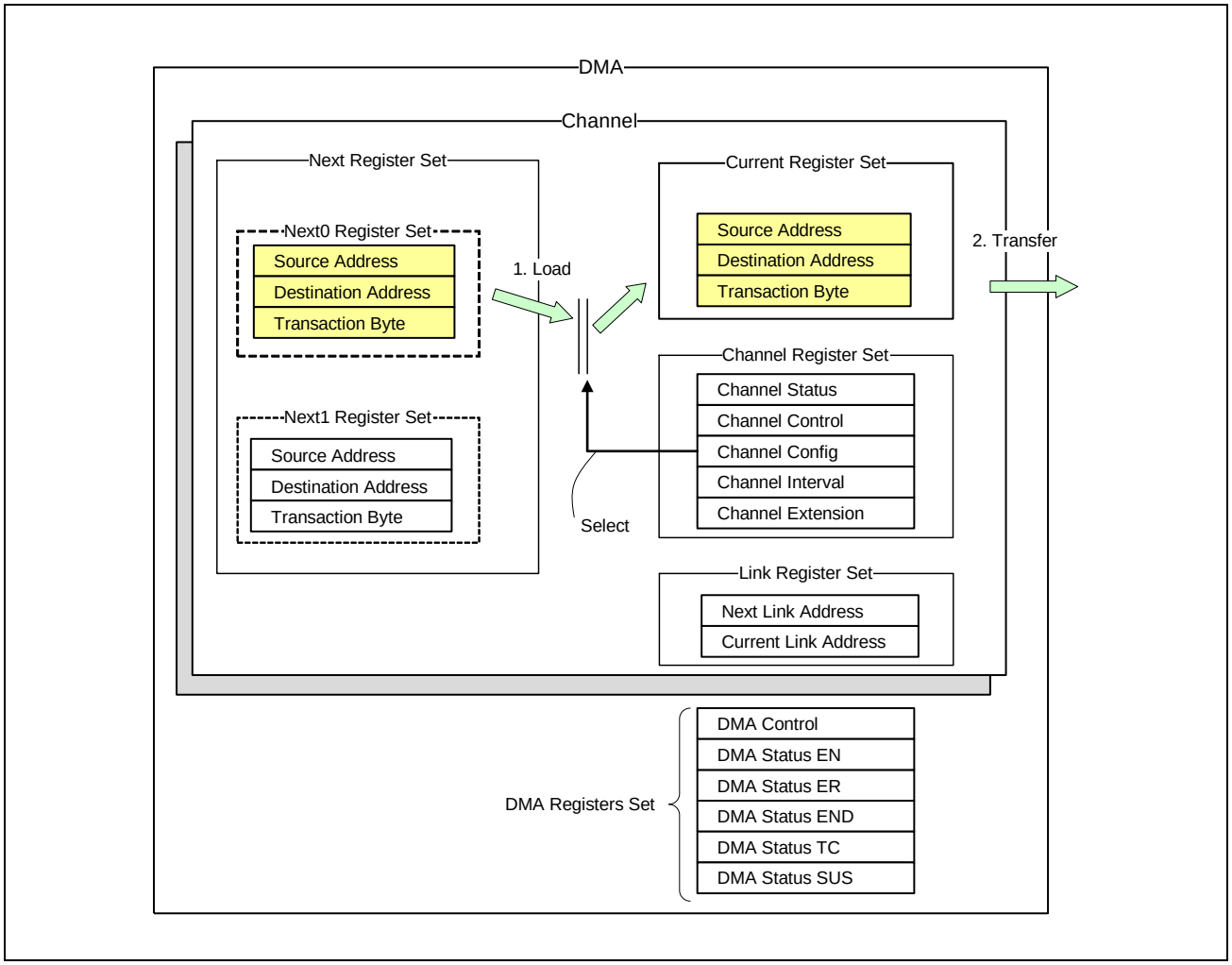



Figure 14.1 Register Configuration

### (a) Next Register Set

This register set is used to set the source address, destination address, and transfer byte count of the DMA transaction to be executed next.

It consists of the Next0 register set and the Next1 register set.

In register mode, set this register set by using software. In link mode, the descriptor read data is automatically set in the Next0 register set.

These registers set values are loaded to the Current Register Set and used for DMA transfer.

### (b) Current Register Set

This register set indicates the source address, destination address, and transfer byte count of the currently executed DMA transaction.

The values are loaded from the Next0/1 register set (register mode) or from the descriptor read data (link mode). The user cannot write directly to this register set.

The register set is automatically updated each time a DMA transaction is executed.

### (c) Channel Register Set

This register set is used to make the DMA transfer settings.
The settings to be made with this register set include channel status indication, channel control, DMA transaction setting, and DMA transaction interval.

### (d) Link Register Set

This register set consists of a register that sets the address of the descriptor to be loaded next in link mode (Next Link Address Register) and a register that indicates the address of the currently executed descriptor (Current Link Address Register).

The Current Link Address Register is automatically updated when a descriptor is read. The user cannot write directly to this register set.

### (e) DMA Register Set

This register set consists of a register that controls DMA as a whole and registers that indicate the status of the corresponding channels. It enables channel priority control as well as the monitoring of the channel status (EN, ER, END, TC, and SUS).

### (f) Extended Resource Selector Register Set

This register set is used to select the on-chip peripheral module to perform DMA transfer request.

## 14.4 Register Descriptions

**Table 14.1** lists the register configuration. There are eleven control registers and five status registers for each channel, and twelve common control registers are used by all channels. In addition, there is one extension resource selector per two channels.

The notation for the registers of each channel is as follows.

- Next 0 Source Address Register of channel 0 for non-secure access:
  Next 0 Source Address Register 0 (abbreviation: N0SA_0)

### (1) AXI I/F

| Base Address Name | Base Address |
|---|---|
| <Base_NS0> | H'0_1182_0000 (Cortex-A55 Address Space) |

Table 14.1 Register Configuration (1/7)

| Channel | Register Name | Abbreviation | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|---|
| 0 | Next0 Source Address Register 0 | N0SA_0 | RW | H'00000000 | <Base_NS0>+H'0000 | 32 |
| | Next0 Destination Address Register 0 | N0DA_0 | RW | H'00000000 | <Base_NS0>+H'0004 | 32 |
| | Next0 Transaction Byte Register 0 | N0TB_0 | RW | H'00000000 | <Base_NS0>+H'0008 | 32 |
| | Next1 Source Address Register 0 | N1SA_0 | RW | H'00000000 | <Base_NS0>+H'000C | 32 |
| | Next1 Destination Address Register 0 | N1DA_0 | RW | H'00000000 | <Base_NS0>+H'0010 | 32 |
| | Next1 Transaction Byte Register 0 | N1TB_0 | RW | H'00000000 | <Base_NS0>+H'0014 | 32 |
| | Current Source Address Register 0 | CRSA_0 | R | H'00000000 | <Base_NS0>+H'0018 | 32 |
| | Current Destination Address Register 0 | CRDA_0 | R | H'00000000 | <Base_NS0>+H'001C | 32 |
| | Current Transaction Byte Register 0 | CRTB_0 | R | H'00000000 | <Base_NS0>+H'0020 | 32 |
| | Channel Status Register 0 | CHSTAT_0 | R | H'00000000 | <Base_NS0>+H'0024 | 32 |
| | Channel Control Register 0 | CHCTRL_0 | RW | H'00000000 | <Base_NS0>+H'0028 | 32 |
| | Channel Configuration Register 0 | CHCFG_0 | RW | H'00000000 | <Base_NS0>+H'002C | 32 |
| | Channel Interval Register 0 | CHITVL_0 | RW | H'00000000 | <Base_NS0>+H'0030 | 32 |
| | Channel Extension Register 0 | CHEXT_0 | RW | H'00000000 | <Base_NS0>+H'0034 | 32 |
| | Next Link Address Register 0 | NXLA_0 | RW | H'00000000 | <Base_NS0>+H'0038 | 32 |
| | Current Link Address Register 0 | CRLA_0 | R | H'00000000 | <Base_NS0>+H'003C | 32 |
| 1 | Next0 Source Address Register 1 | N0SA_1 | RW | H'00000000 | <Base_NS0>+H'0040 | 32 |
| | Next0 Destination Address Register 1 | N0DA_1 | RW | H'00000000 | <Base_NS0>+H'0044 | 32 |
| | Next0 Transaction Byte Register 1 | N0TB_1 | RW | H'00000000 | <Base_NS0>+H'0048 | 32 |
| | Next1 Source Address Register 1 | N1SA_1 | RW | H'00000000 | <Base_NS0>+H'004C | 32 |
| | Next1 Destination Address Register 1 | N1DA_1 | RW | H'00000000 | <Base_NS0>+H'0050 | 32 |
| | Next1 Transaction Byte Register 1 | N1TB_1 | RW | H'00000000 | <Base_NS0>+H'0054 | 32 |
| | Current Source Address Register 1 | CRSA_1 | R | H'00000000 | <Base_NS0>+H'0058 | 32 |
| | Current Destination Address Register 1 | CRDA_1 | R | H'00000000 | <Base_NS0>+H'005C | 32 |
| | Current Transaction Byte Register 1 | CRTB_1 | R | H'00000000 | <Base_NS0>+H'0060 | 32 |
| | Channel Status Register 1 | CHSTAT_1 | R | H'00000000 | <Base_NS0>+H'0064 | 32 |
| | Channel Control Register 1 | CHCTRL_1 | RW | H'00000000 | <Base_NS0>+H'0068 | 32 |
| | Channel Configuration Register 1 | CHCFG_1 | RW | H'00000000 | <Base_NS0>+H'006C | 32 |
| | Channel Interval Register 1 | CHITVL_1 | RW | H'00000000 | <Base_NS0>+H'0070 | 32 |
| | Channel Extension Register 1 | CHEXT_1 | RW | H'00000000 | <Base_NS0>+H'0074 | 32 |
| | Next Link Address Register 1 | NXLA_1 | RW | H'00000000 | <Base_NS0>+H'0078 | 32 |

Table 14.1 Register Configuration (2/7)

| Channel | Register Name | Abbreviation | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|---|
| 1 | Current Link Address Register 1 | CRLA_1 | R | H'00000000 | <Base_NS0>+H'007C | 32 |
| 2 | Next0 Source Address Register 2 | N0SA_2 | RW | H'00000000 | <Base_NS0>+H'0080 | 32 |
| | Next0 Destination Address Register 2 | N0DA_2 | RW | H'00000000 | <Base_NS0>+H'0084 | 32 |
| | Next0 Transaction Byte Register 2 | N0TB_2 | RW | H'00000000 | <Base_NS0>+H'0088 | 32 |
| | Next1 Source Address Register 2 | N1SA_2 | RW | H'00000000 | <Base_NS0>+H'008C | 32 |
| | Next1 Destination Address Register 2 | N1DA_2 | RW | H'00000000 | <Base_NS0>+H'0090 | 32 |
| | Next1 Transaction Byte Register 2 | N1TB_2 | RW | H'00000000 | <Base_NS0>+H'0094 | 32 |
| | Current Source Address Register 2 | CRSA_2 | R | H'00000000 | <Base_NS0>+H'0098 | 32 |
| | Current Destination Address Register 2 | CRDA_2 | R | H'00000000 | <Base_NS0>+H'009C | 32 |
| | Current Transaction Byte Register 2 | CRTB_2 | R | H'00000000 | <Base_NS0>+H'00A0 | 32 |
| | Channel Status Register 2 | CHSTAT_2 | R | H'00000000 | <Base_NS0>+H'00A4 | 32 |
| | Channel Control Register 2 | CHCTRL_2 | RW | H'00000000 | <Base_NS0>+H'00A8 | 32 |
| | Channel Configuration Register 2 | CHCFG_2 | RW | H'00000000 | <Base_NS0>+H'00AC | 32 |
| | Channel Interval Register 2 | CHITVL_2 | RW | H'00000000 | <Base_NS0>+H'00B0 | 32 |
| | Channel Extension Register 2 | CHEXT_2 | RW | H'00000000 | <Base_NS0>+H'00B4 | 32 |
| | Next Link Address Register 2 | NXLA_2 | RW | H'00000000 | <Base_NS0>+H'00B8 | 32 |
| | Current Link Address Register 2 | CRLA_2 | R | H'00000000 | <Base_NS0>+H'00BC | 32 |
| 3 | Next0 Source Address Register 3 | N0SA_3 | RW | H'00000000 | <Base_NS0>+H'00C0 | 32 |
| | Next0 Destination Address Register 3 | N0DA_3 | RW | H'00000000 | <Base_NS0>+H'00C4 | 32 |
| | Next0 Transaction Byte Register 3 | N0TB_3 | RW | H'00000000 | <Base_NS0>+H'00C8 | 32 |
| | Next1 Source Address Register 3 | N1SA_3 | RW | H'00000000 | <Base_NS0>+H'00CC | 32 |
| | Next1 Destination Address Register 3 | N1DA_3 | RW | H'00000000 | <Base_NS0>+H'00D0 | 32 |
| | Next1 Transaction Byte Register 3 | N1TB_3 | RW | H'00000000 | <Base_NS0>+H'00D4 | 32 |
| | Current Source Address Register 3 | CRSA_3 | R | H'00000000 | <Base_NS0>+H'00D8 | 32 |
| | Current Destination Address Register 3 | CRDA_3 | R | H'00000000 | <Base_NS0>+H'00DC | 32 |
| | Current Transaction Byte Register 3 | CRTB_3 | R | H'00000000 | <Base_NS0>+H'00E0 | 32 |
| | Channel Status Register 3 | CHSTAT_3 | R | H'00000000 | <Base_NS0>+H'00E4 | 32 |
| | Channel Control Register 3 | CHCTRL_3 | RW | H'00000000 | <Base_NS0>+H'00E8 | 32 |
| | Channel Configuration Register 3 | CHCFG_3 | RW | H'00000000 | <Base_NS0>+H'00EC | 32 |
| | Channel Interval Register 3 | CHITVL_3 | RW | H'00000000 | <Base_NS0>+H'00F0 | 32 |
| | Channel Extension Register 3 | CHEXT_3 | RW | H'00000000 | <Base_NS0>+H'00F4 | 32 |
| | Next Link Address Register 3 | NXLA_3 | RW | H'00000000 | <Base_NS0>+H'00F8 | 32 |
| | Current Link Address Register 3 | CRLA_3 | R | H'00000000 | <Base_NS0>+H'00FC | 32 |
| 4 | Next0 Source Address Register 4 | N0SA_4 | RW | H'00000000 | <Base_NS0>+H'0100 | 32 |
| | Next0 Destination Address Register 4 | N0DA_4 | RW | H'00000000 | <Base_NS0>+H'0104 | 32 |
| | Next0 Transaction Byte Register 4 | N0TB_4 | RW | H'00000000 | <Base_NS0>+H'0108 | 32 |
| | Next1 Source Address Register 4 | N1SA_4 | RW | H'00000000 | <Base_NS0>+H'010C | 32 |
| | Next1 Destination Address Register 4 | N1DA_4 | RW | H'00000000 | <Base_NS0>+H'0110 | 32 |
| | Next1 Transaction Byte Register 4 | N1TB_4 | RW | H'00000000 | <Base_NS0>+H'0114 | 32 |
| | Current Source Address Register 4 | CRSA_4 | R | H'00000000 | <Base_NS0>+H'0118 | 32 |
| | Current Destination Address Register 4 | CRDA_4 | R | H'00000000 | <Base_NS0>+H'011C | 32 |
| | Current Transaction Byte Register 4 | CRTB_4 | R | H'00000000 | <Base_NS0>+H'0120 | 32 |
| | Channel Status Register 4 | CHSTAT_4 | R | H'00000000 | <Base_NS0>+H'0124 | 32 |
| | Channel Control Register 4 | CHCTRL_4 | RW | H'00000000 | <Base_NS0>+H'0128 | 32 |
| | Channel Configuration Register 4 | CHCFG_4 | RW | H'00000000 | <Base_NS0>+H'012C | 32 |

Table 14.1 Register Configuration (3/7)

| Channel | Register Name | Abbreviation | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|---|
| 4 | Channel Interval Register 4 | CHITVL_4 | RW | H'00000000 | <Base_NS0>+H'0130 | 32 |
| | Channel Extension Register 4 | CHEXT_4 | RW | H'00000000 | <Base_NS0>+H'0134 | 32 |
| | Next Link Address Register 4 | NXLA_4 | RW | H'00000000 | <Base_NS0>+H'0138 | 32 |
| | Current Link Address Register 4 | CRLA_4 | R | H'00000000 | <Base_NS0>+H'013C | 32 |
| 5 | Next0 Source Address Register 5 | N0SA_5 | RW | H'00000000 | <Base_NS0>+H'0140 | 32 |
| | Next0 Destination Address Register 5 | N0DA_5 | RW | H'00000000 | <Base_NS0>+H'0144 | 32 |
| | Next0 Transaction Byte Register 5 | N0TB_5 | RW | H'00000000 | <Base_NS0>+H'0148 | 32 |
| | Next1 Source Address Register 5 | N1SA_5 | RW | H'00000000 | <Base_NS0>+H'014C | 32 |
| | Next1 Destination Address Register 5 | N1DA_5 | RW | H'00000000 | <Base_NS0>+H'0150 | 32 |
| | Next1 Transaction Byte Register 5 | N1TB_5 | RW | H'00000000 | <Base_NS0>+H'0154 | 32 |
| | Current Source Address Register 5 | CRSA_5 | R | H'00000000 | <Base_NS0>+H'0158 | 32 |
| | Current Destination Address Register 5 | CRDA_5 | R | H'00000000 | <Base_NS0>+H'015C | 32 |
| | Current Transaction Byte Register 5 | CRTB_5 | R | H'00000000 | <Base_NS0>+H'0160 | 32 |
| | Channel Status Register 5 | CHSTAT_5 | R | H'00000000 | <Base_NS0>+H'0164 | 32 |
| | Channel Control Register 5 | CHCTRL_5 | RW | H'00000000 | <Base_NS0>+H'0168 | 32 |
| | Channel Configuration Register 5 | CHCFG_5 | RW | H'00000000 | <Base_NS0>+H'016C | 32 |
| | Channel Interval Register 5 | CHITVL_5 | RW | H'00000000 | <Base_NS0>+H'0170 | 32 |
| | Channel Extension Register 5 | CHEXT_5 | RW | H'00000000 | <Base_NS0>+H'0174 | 32 |
| | Next Link Address Register 5 | NXLA_5 | RW | H'00000000 | <Base_NS0>+H'0178 | 32 |
| | Current Link Address Register 5 | CRLA_5 | R | H'00000000 | <Base_NS0>+H'017C | 32 |
| 6 | Next0 Source Address Register 6 | N0SA_6 | RW | H'00000000 | <Base_NS0>+H'0180 | 32 |
| | Next0 Destination Address Register 6 | N0DA_6 | RW | H'00000000 | <Base_NS0>+H'0184 | 32 |
| | Next0 Transaction Byte Register 6 | N0TB_6 | RW | H'00000000 | <Base_NS0>+H'0188 | 32 |
| | Next1 Source Address Register 6 | N1SA_6 | RW | H'00000000 | <Base_NS0>+H'018C | 32 |
| | Next1 Destination Address Register 6 | N1DA_6 | RW | H'00000000 | <Base_NS0>+H'0190 | 32 |
| | Next1 Transaction Byte Register 6 | N1TB_6 | RW | H'00000000 | <Base_NS0>+H'0194 | 32 |
| | Current Source Address Register 6 | CRSA_6 | R | H'00000000 | <Base_NS0>+H'0198 | 32 |
| | Current Destination Address Register 6 | CRDA_6 | R | H'00000000 | <Base_NS0>+H'019C | 32 |
| | Current Transaction Byte Register 6 | CRTB_6 | R | H'00000000 | <Base_NS0>+H'01A0 | 32 |
| | Channel Status Register 6 | CHSTAT_6 | R | H'00000000 | <Base_NS0>+H'01A4 | 32 |
| | Channel Control Register 6 | CHCTRL_6 | RW | H'00000000 | <Base_NS0>+H'01A8 | 32 |
| | Channel Configuration Register 6 | CHCFG_6 | RW | H'00000000 | <Base_NS0>+H'01AC | 32 |
| | Channel Interval Register 6 | CHITVL_6 | RW | H'00000000 | <Base_NS0>+H'01B0 | 32 |
| | Channel Extension Register 6 | CHEXT_6 | RW | H'00000000 | <Base_NS0>+H'01B4 | 32 |
| | Next Link Address Register 6 | NXLA_6 | RW | H'00000000 | <Base_NS0>+H'01B8 | 32 |
| | Current Link Address Register 6 | CRLA_6 | R | H'00000000 | <Base_NS0>+H'01BC | 32 |
| 7 | Next0 Source Address Register 7 | N0SA_7 | RW | H'00000000 | <Base_NS0>+H'01C0 | 32 |
| | Next0 Destination Address Register 7 | N0DA_7 | RW | H'00000000 | <Base_NS0>+H'01C4 | 32 |
| | Next0 Transaction Byte Register 7 | N0TB_7 | RW | H'00000000 | <Base_NS0>+H'01C8 | 32 |
| | Next1 Source Address Register 7 | N1SA_7 | RW | H'00000000 | <Base_NS0>+H'01CC | 32 |
| | Next1 Destination Address Register 7 | N1DA_7 | RW | H'00000000 | <Base_NS0>+H'01D0 | 32 |
| | Next1 Transaction Byte Register 7 | N1TB_7 | RW | H'00000000 | <Base_NS0>+H'01D4 | 32 |
| | Current Source Address Register 7 | CRSA_7 | R | H'00000000 | <Base_NS0>+H'01D8 | 32 |
| | Current Destination Address Register 7 | CRDA_7 | R | H'00000000 | <Base_NS0>+H'01DC | 32 |
| | Current Transaction Byte Register 7 | CRTB_7 | R | H'00000000 | <Base_NS0>+H'01E0 | 32 |

Table 14.1 Register Configuration (4/7)

| Channel | Register Name | Abbreviation | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|---|
| 7 | Channel Status Register 7 | CHSTAT_7 | R | H'00000000 | <Base_NS0>+H'01E4 | 32 |
| | Channel Control Register 7 | CHCTRL_7 | RW | H'00000000 | <Base_NS0>+H'01E8 | 32 |
| | Channel Configuration Register 7 | CHCFG_7 | RW | H'00000000 | <Base_NS0>+H'01EC | 32 |
| | Channel Interval Register 7 | CHITVL_7 | RW | H'00000000 | <Base_NS0>+H'01F0 | 32 |
| | Channel Extension Register 7 | CHEXT_7 | RW | H'00000000 | <Base_NS0>+H'01F4 | 32 |
| | Next Link Address Register 7 | NXLA_7 | RW | H'00000000 | <Base_NS0>+H'01F8 | 32 |
| | Current Link Address Register 7 | CRLA_7 | R | H'00000000 | <Base_NS0>+H'01FC | 32 |
| Common for 0 to 7 | DMA Control Registers 0-7 | DCTRL_0_7 | RW | H'00000000 | <Base_NS0>+H'0300 | 32 |
| | DMA Status EN Registers 0-7 | DSTAT_EN_0_7 | R | H'00000000 | <Base_NS0>+H'0310 | 32 |
| | DMA Status ER Registers 0-7 | DSTAT_ER_0_7 | R | H'00000000 | <Base_NS0>+H'0314 | 32 |
| | DMA Status END Registers 0-7 | DSTAT_END_0_7 | R | H'00000000 | <Base_NS0>+H'0318 | 32 |
| | DMA Status TC Registers 0-7 | DSTAT_TC_0_7 | R | H'00000000 | <Base_NS0>+H'031C | 32 |
| | DMA Status SUS Registers 0-7 | DSTAT_SUS_0_7 | R | H'00000000 | <Base_NS0>+H'0320 | 32 |
| 8 | Next0 Source Address Register 8 | N0SA_8 | RW | H'00000000 | <Base_NS0>+H'0400 | 32 |
| | Next0 Destination Address Register 8 | N0DA_8 | RW | H'00000000 | <Base_NS0>+H'0404 | 32 |
| | Next0 Transaction Byte Register 8 | N0TB_8 | RW | H'00000000 | <Base_NS0>+H'0408 | 32 |
| | Next1 Source Address Register 8 | N1SA_8 | RW | H'00000000 | <Base_NS0>+H'040C | 32 |
| | Next1 Destination Address Register 8 | N1DA_8 | RW | H'00000000 | <Base_NS0>+H'0410 | 32 |
| | Next1 Transaction Byte Register 8 | N1TB_8 | RW | H'00000000 | <Base_NS0>+H'0414 | 32 |
| | Current Source Address Register 8 | CRSA_8 | R | H'00000000 | <Base_NS0>+H'0418 | 32 |
| | Current Destination Address Register 8 | CRDA_8 | R | H'00000000 | <Base_NS0>+H'041C | 32 |
| | Current Transaction Byte Register 8 | CRTB_8 | R | H'00000000 | <Base_NS0>+H'0420 | 32 |
| | Channel Status Register 8 | CHSTAT_8 | R | H'00000000 | <Base_NS0>+H'0424 | 32 |
| | Channel Control Register 8 | CHCTRL_8 | RW | H'00000000 | <Base_NS0>+H'0428 | 32 |
| | Channel Configuration Register 8 | CHCFG_8 | RW | H'00000000 | <Base_NS0>+H'042C | 32 |
| | Channel Interval Register 8 | CHITVL_8 | RW | H'00000000 | <Base_NS0>+H'0430 | 32 |
| | Channel Extension Register 8 | CHEXT_8 | RW | H'00000000 | <Base_NS0>+H'0434 | 32 |
| | Next Link Address Register 8 | NXLA_8 | RW | H'00000000 | <Base_NS0>+H'0438 | 32 |
| | Current Link Address Register 8 | CRLA_8 | R | H'00000000 | <Base_NS0>+H'043C | 32 |
| 9 | Next0 Source Address Register 9 | N0SA_9 | RW | H'00000000 | <Base_NS0>+H'0440 | 32 |
| | Next0 Destination Address Register 9 | N0DA_9 | RW | H'00000000 | <Base_NS0>+H'0444 | 32 |
| | Next0 Transaction Byte Register 9 | N0TB_9 | RW | H'00000000 | <Base_NS0>+H'0448 | 32 |
| | Next1 Source Address Register 9 | N1SA_9 | RW | H'00000000 | <Base_NS0>+H'044C | 32 |
| | Next1 Destination Address Register 9 | N1DA_9 | RW | H'00000000 | <Base_NS0>+H'0450 | 32 |
| | Next1 Transaction Byte Register 9 | N1TB_9 | RW | H'00000000 | <Base_NS0>+H'0454 | 32 |
| | Current Source Address Register 9 | CRSA_9 | R | H'00000000 | <Base_NS0>+H'0458 | 32 |
| | Current Destination Address Register 9 | CRDA_9 | R | H'00000000 | <Base_NS0>+H'045C | 32 |
| | Current Transaction Byte Register 9 | CRTB_9 | R | H'00000000 | <Base_NS0>+H'0460 | 32 |
| | Channel Status Register 9 | CHSTAT_9 | R | H'00000000 | <Base_NS0>+H'0464 | 32 |
| | Channel Control Register 9 | CHCTRL_9 | RW | H'00000000 | <Base_NS0>+H'0468 | 32 |
| | Channel Configuration Register 9 | CHCFG_9 | RW | H'00000000 | <Base_NS0>+H'046C | 32 |
| | Channel Interval Register 9 | CHITVL_9 | RW | H'00000000 | <Base_NS0>+H'0470 | 32 |
| | Channel Extension Register 9 | CHEXT_9 | RW | H'00000000 | <Base_NS0>+H'0474 | 32 |
| | Next Link Address Register 9 | NXLA_9 | RW | H'00000000 | <Base_NS0>+H'0478 | 32 |
| | Current Link Address Register 9 | CRLA_9 | R | H'00000000 | <Base_NS0>+H'047C | 32 |

Table 14.1 Register Configuration (5/7)

| Channel | Register Name | Abbreviation | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|---|
| 10 | Next0 Source Address Register 10 | N0SA_10 | RW | H'00000000 | <Base_NS0>+H'0480 | 32 |
| | Next0 Destination Address Register 10 | N0DA_10 | RW | H'00000000 | <Base_NS0>+H'0484 | 32 |
| | Next0 Transaction Byte Register 10 | N0TB_10 | RW | H'00000000 | <Base_NS0>+H'0488 | 32 |
| | Next1 Source Address Register 10 | N1SA_10 | RW | H'00000000 | <Base_NS0>+H'048C | 32 |
| | Next1 Destination Address Register 10 | N1DA_10 | RW | H'00000000 | <Base_NS0>+H'0490 | 32 |
| | Next1 Transaction Byte Register 10 | N1TB_10 | RW | H'00000000 | <Base_NS0>+H'0494 | 32 |
| | Current Source Address Register 10 | CRSA_10 | R | H'00000000 | <Base_NS0>+H'0498 | 32 |
| | Current Destination Address Register 10 | CRDA_10 | R | H'00000000 | <Base_NS0>+H'049C | 32 |
| | Current Transaction Byte Register 10 | CRTB_10 | R | H'00000000 | <Base_NS0>+H'04A0 | 32 |
| | Channel Status Register 10 | CHSTAT_10 | R | H'00000000 | <Base_NS0>+H'04A4 | 32 |
| | Channel Control Register 10 | CHCTRL_10 | RW | H'00000000 | <Base_NS0>+H'04A8 | 32 |
| | Channel Configuration Register 10 | CHCFG_10 | RW | H'00000000 | <Base_NS0>+H'04AC | 32 |
| | Channel Interval Register 10 | CHITVL_10 | RW | H'00000000 | <Base_NS0>+H'04B0 | 32 |
| | Channel Extension Register 10 | CHEXT_10 | RW | H'00000000 | <Base_NS0>+H'04B4 | 32 |
| | Next Link Address Register 10 | NXLA_10 | RW | H'00000000 | <Base_NS0>+H'04B8 | 32 |
| | Current Link Address Register 10 | CRLA_10 | R | H'00000000 | <Base_NS0>+H'04BC | 32 |
| 11 | Next0 Source Address Register 11 | N0SA_11 | RW | H'00000000 | <Base_NS0>+H'04C0 | 32 |
| | Next0 Destination Address Register 11 | N0DA_11 | RW | H'00000000 | <Base_NS0>+H'04C4 | 32 |
| | Next0 Transaction Byte Register 11 | N0TB_11 | RW | H'00000000 | <Base_NS0>+H'04C8 | 32 |
| | Next1 Source Address Register 11 | N1SA_11 | RW | H'00000000 | <Base_NS0>+H'04CC | 32 |
| | Next1 Destination Address Register 11 | N1DA_11 | RW | H'00000000 | <Base_NS0>+H'04D0 | 32 |
| | Next1 Transaction Byte Register 11 | N1TB_11 | RW | H'00000000 | <Base_NS0>+H'04D4 | 32 |
| | Current Source Address Register 11 | CRSA_11 | R | H'00000000 | <Base_NS0>+H'04D8 | 32 |
| | Current Destination Address Register 11 | CRDA_11 | R | H'00000000 | <Base_NS0>+H'04DC | 32 |
| | Current Transaction Byte Register 11 | CRTB_11 | R | H'00000000 | <Base_NS0>+H'04E0 | 32 |
| | Channel Status Register 11 | CHSTAT_11 | R | H'00000000 | <Base_NS0>+H'04E4 | 32 |
| | Channel Control Register 11 | CHCTRL_11 | RW | H'00000000 | <Base_NS0>+H'04E8 | 32 |
| | Channel Configuration Register 11 | CHCFG_11 | RW | H'00000000 | <Base_NS0>+H'04EC | 32 |
| | Channel Interval Register 11 | CHITVL_11 | RW | H'00000000 | <Base_NS0>+H'04F0 | 32 |
| | Channel Extension Register 11 | CHEXT_11 | RW | H'00000000 | <Base_NS0>+H'04F4 | 32 |
| | Next Link Address Register 11 | NXLA_11 | RW | H'00000000 | <Base_NS0>+H'04F8 | 32 |
| | Current Link Address Register 11 | CRLA_11 | R | H'00000000 | <Base_NS0>+H'04FC | 32 |
| 12 | Next0 Source Address Register 12 | N0SA_12 | RW | H'00000000 | <Base_NS0>+H'0500 | 32 |
| | Next0 Destination Address Register 12 | N0DA_12 | RW | H'00000000 | <Base_NS0>+H'0504 | 32 |
| | Next0 Transaction Byte Register 12 | N0TB_12 | RW | H'00000000 | <Base_NS0>+H'0508 | 32 |
| | Next1 Source Address Register 12 | N1SA_12 | RW | H'00000000 | <Base_NS0>+H'050C | 32 |
| | Next1 Destination Address Register 12 | N1DA_12 | RW | H'00000000 | <Base_NS0>+H'0510 | 32 |
| | Next1 Transaction Byte Register 12 | N1TB_12 | RW | H'00000000 | <Base_NS0>+H'0514 | 32 |
| | Current Source Address Register 12 | CRSA_12 | R | H'00000000 | <Base_NS0>+H'0518 | 32 |
| | Current Destination Address Register 12 | CRDA_12 | R | H'00000000 | <Base_NS0>+H'051C | 32 |
| | Current Transaction Byte Register 12 | CRTB_12 | R | H'00000000 | <Base_NS0>+H'0520 | 32 |
| | Channel Status Register 12 | CHSTAT_12 | R | H'00000000 | <Base_NS0>+H'0524 | 32 |
| | Channel Control Register 12 | CHCTRL_12 | RW | H'00000000 | <Base_NS0>+H'0528 | 32 |
| | Channel Configuration Register 12 | CHCFG_12 | RW | H'00000000 | <Base_NS0>+H'052C | 32 |
| | Channel Interval Register 12 | CHITVL_12 | RW | H'00000000 | <Base_NS0>+H'0530 | 32 |

Table 14.1 Register Configuration (6/7)

| Channel | Register Name | Abbreviation | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|---|
| 12 | Channel Extension Register 12 | CHEXT_12 | RW | H'00000000 | <Base_NS0>+H'0534 | 32 |
| | Next Link Address Register 12 | NXLA_12 | RW | H'00000000 | <Base_NS0>+H'0538 | 32 |
| | Current Link Address Register 12 | CRLA_12 | R | H'00000000 | <Base_NS0>+H'053C | 32 |
| 13 | Next0 Source Address Register 13 | N0SA_13 | RW | H'00000000 | <Base_NS0>+H'0540 | 32 |
| | Next0 Destination Address Register 13 | N0DA_13 | RW | H'00000000 | <Base_NS0>+H'0544 | 32 |
| | Next0 Transaction Byte Register 13 | N0TB_13 | RW | H'00000000 | <Base_NS0>+H'0548 | 32 |
| | Next1 Source Address Register 13 | N1SA_13 | RW | H'00000000 | <Base_NS0>+H'054C | 32 |
| | Next1 Destination Address Register 13 | N1DA_13 | RW | H'00000000 | <Base_NS0>+H'0550 | 32 |
| | Next1 Transaction Byte Register 13 | N1TB_13 | RW | H'00000000 | <Base_NS0>+H'0554 | 32 |
| | Current Source Address Register 13 | CRSA_13 | R | H'00000000 | <Base_NS0>+H'0558 | 32 |
| | Current Destination Address Register 13 | CRDA_13 | R | H'00000000 | <Base_NS0>+H'055C | 32 |
| | Current Transaction Byte Register 13 | CRTB_13 | R | H'00000000 | <Base_NS0>+H'0560 | 32 |
| | Channel Status Register 13 | CHSTAT_13 | R | H'00000000 | <Base_NS0>+H'0564 | 32 |
| | Channel Control Register 13 | CHCTRL_13 | RW | H'00000000 | <Base_NS0>+H'0568 | 32 |
| | Channel Configuration Register 13 | CHCFG_13 | RW | H'00000000 | <Base_NS0>+H'056C | 32 |
| | Channel Interval Register 13 | CHITVL_13 | RW | H'00000000 | <Base_NS0>+H'0570 | 32 |
| | Channel Extension Register 13 | CHEXT_13 | RW | H'00000000 | <Base_NS0>+H'0574 | 32 |
| | Next Link Address Register 13 | NXLA_13 | RW | H'00000000 | <Base_NS0>+H'0578 | 32 |
| | Current Link Address Register 13 | CRLA_13 | R | H'00000000 | <Base_NS0>+H'057C | 32 |
| 14 | Next0 Source Address Register 14 | N0SA_14 | RW | H'00000000 | <Base_NS0>+H'0580 | 32 |
| | Next0 Destination Address Register 14 | N0DA_14 | RW | H'00000000 | <Base_NS0>+H'0584 | 32 |
| | Next0 Transaction Byte Register 14 | N0TB_14 | RW | H'00000000 | <Base_NS0>+H'0588 | 32 |
| | Next1 Source Address Register 14 | N1SA_14 | RW | H'00000000 | <Base_NS0>+H'058C | 32 |
| | Next1 Destination Address Register 14 | N1DA_14 | RW | H'00000000 | <Base_NS0>+H'0590 | 32 |
| | Next1 Transaction Byte Register 14 | N1TB_14 | RW | H'00000000 | <Base_NS0>+H'0594 | 32 |
| | Current Source Address Register 14 | CRSA_14 | R | H'00000000 | <Base_NS0>+H'0598 | 32 |
| | Current Destination Address Register 14 | CRDA_14 | R | H'00000000 | <Base_NS0>+H'059C | 32 |
| | Current Transaction Byte Register 14 | CRTB_14 | R | H'00000000 | <Base_NS0>+H'05A0 | 32 |
| | Channel Status Register 14 | CHSTAT_14 | R | H'00000000 | <Base_NS0>+H'05A4 | 32 |
| | Channel Control Register 14 | CHCTRL_14 | RW | H'00000000 | <Base_NS0>+H'05A8 | 32 |
| | Channel Configuration Register 14 | CHCFG_14 | RW | H'00000000 | <Base_NS0>+H'05AC | 32 |
| | Channel Interval Register 14 | CHITVL_14 | RW | H'00000000 | <Base_NS0>+H'05B0 | 32 |
| | Channel Extension Register 14 | CHEXT_14 | RW | H'00000000 | <Base_NS0>+H'05B4 | 32 |
| | Next Link Address Register 14 | NXLA_14 | RW | H'00000000 | <Base_NS0>+H'05B8 | 32 |
| | Current Link Address Register 14 | CRLA_14 | R | H'00000000 | <Base_NS0>+H'05BC | 32 |
| 15 | Next0 Source Address Register 15 | N0SA_15 | RW | H'00000000 | <Base_NS0>+H'05C0 | 32 |
| | Next0 Destination Address Register 15 | N0DA_15 | RW | H'00000000 | <Base_NS0>+H'05C4 | 32 |
| | Next0 Transaction Byte Register 15 | N0TB_15 | RW | H'00000000 | <Base_NS0>+H'05C8 | 32 |
| | Next1 Source Address Register 15 | N1SA_15 | RW | H'00000000 | <Base_NS0>+H'05CC | 32 |
| | Next1 Destination Address Register 15 | N1DA_15 | RW | H'00000000 | <Base_NS0>+H'05D0 | 32 |
| | Next1 Transaction Byte Register 15 | N1TB_15 | RW | H'00000000 | <Base_NS0>+H'05D4 | 32 |
| | Current Source Address Register 15 | CRSA_15 | R | H'00000000 | <Base_NS0>+H'05D8 | 32 |
| | Current Destination Address Register 15 | CRDA_15 | R | H'00000000 | <Base_NS0>+H'05DC | 32 |
| | Current Transaction Byte Register 15 | CRTB_15 | R | H'00000000 | <Base_NS0>+H'05E0 | 32 |
| | Channel Status Register 15 | CHSTAT_15 | R | H'00000000 | <Base_NS0>+H'05E4 | 32 |

Table 14.1 Register Configuration (7/7)

| Channel | Register Name | Abbreviation | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|---|
| 15 | Channel Control Register 15 | CHCTRL_15 | RW | H'00000000 | <Base_NS0>+H'05E8 | 32 |
| | Channel Configuration Register 15 | CHCFG_15 | RW | H'00000000 | <Base_NS0>+H'05EC | 32 |
| | Channel Interval Register 15 | CHITVL_15 | RW | H'00000000 | <Base_NS0>+H'05F0 | 32 |
| | Channel Extension Register 15 | CHEXT_15 | RW | H'00000000 | <Base_NS0>+H'05F4 | 32 |
| | Next Link Address Register 15 | NXLA_15 | RW | H'00000000 | <Base_NS0>+H'05F8 | 32 |
| | Current Link Address Register 15 | CRLA_15 | R | H'00000000 | <Base_NS0>+H'05FC | 32 |
| Common for 8 to 15 | DMA Control Registers 8-15 | DCTRL_8_15 | RW | H'00000000 | <Base_NS0>+H'0700 | 32 |
| | DMA Status EN Registers 8-15 | DSTAT_EN_8_15 | R | H'00000000 | <Base_NS0>+H'0710 | 32 |
| | DMA Status ER Registers 8-15 | DSTAT_ER_8_15 | R | H'00000000 | <Base_NS0>+H'0714 | 32 |
| | DMA Status END Registers 8-15 | DSTAT_END_8_15 | R | H'00000000 | <Base_NS0>+H'0718 | 32 |
| | DMA Status TC Registers 8-15 | DSTAT_TC_8_15 | R | H'00000000 | <Base_NS0>+H'071C | 32 |
| | DMA Status SUS Registers 8-15 | DSTAT_SUS_8_15 | R | H'00000000 | <Base_NS0>+H'0720 | 32 |

### (2) APB I/F

| Base Address Name | Base Address |
|---|---|
| <Base_NS1> | H'0_1183_0000 (Cortex-A55 Address Space) |

Table 14.2 Register Configuration

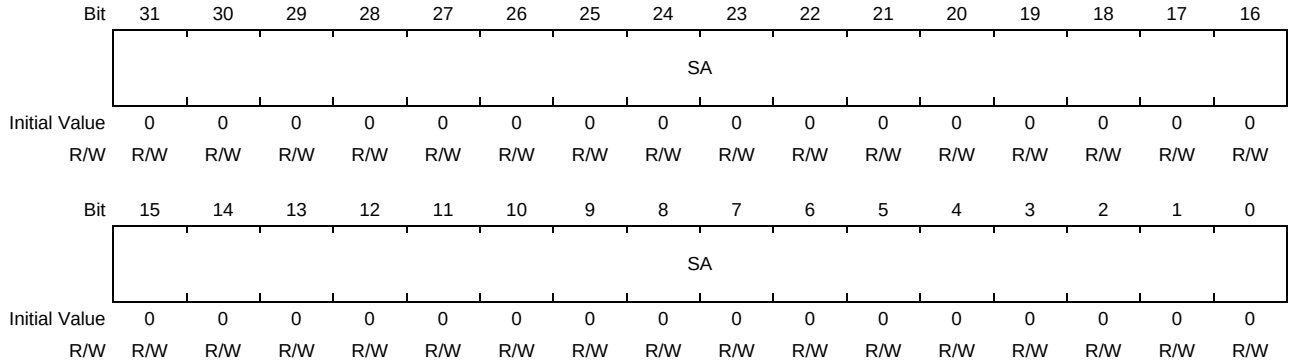
| Channel | Register Name | Abbreviation | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|---|
| 0/1 | DMA Extension Resource Selectors 0 | DMARS0 | RW | H'00000000 | <Base_NS1>+H'0000 | 32 |
| 2/3 | DMA Extension Resource Selectors 1 | DMARS1 | RW | H'00000000 | <Base_NS1>+H'0004 | 32 |
| 4/5 | DMA Extension Resource Selectors 2 | DMARS 2 | RW | H'00000000 | <Base_NS1>+H'0008 | 32 |
| 6/7 | DMA Extension Resource Selectors 3 | DMARS 3 | RW | H'00000000 | <Base_NS1>+H'000C | 32 |
| 8/9 | DMA Extension Resource Selectors 4 | DMARS 4 | RW | H'00000000 | <Base_NS1>+H'0010 | 32 |
| 10/11 | DMA Extension Resource Selectors 5 | DMARS 5 | RW | H'00000000 | <Base_NS1>+H'0014 | 32 |
| 12/13 | DMA Extension Resource Selectors 6 | DMARS 6 | RW | H'00000000 | <Base_NS1>+H'0018 | 32 |
| 14/15 | DMA Extension Resource Selectors 7 | DMARS 7 | RW | H'00000000 | <Base_NS1>+H'001C | 32 |

### 14.4.1 Next Source Address Register n (N0SA_n, N1SA_n)

This register sets the DMA transfer source address (32 bits) of DMA channel n (n = 0 to 15) which is to be executed next. N0SA_n is for the Next0 Register Set, and N1SA_n is for the Next1 Register Set.

In register mode, set this register set by using software. In link mode, the descriptor read data is automatically set in the Next0 register set.

These register set values are loaded to the Current Register Set and used for DMA transfer.




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 0 | SA | All 0 | R/W | Source Address<br>Sets the start address of the DMA transfer source. |

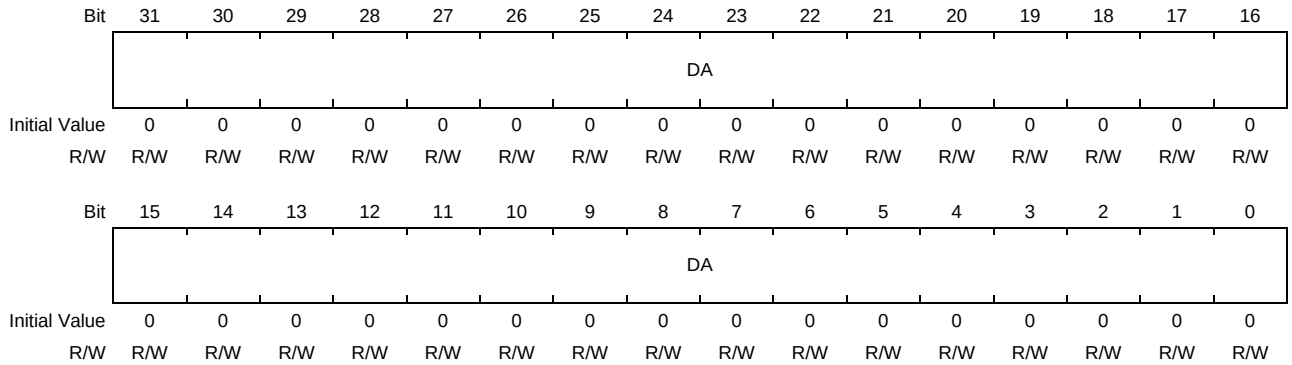
### 14.4.2 Next Destination Address Register n (N0DA_n, N1DA_n)

This register sets the DMA transfer destination address (32 bits) of DMA channel n (n = 0 to 15) which is to be executed next.

N0DA_n is for the Next0 Register Set, and N1DA_n is for the Next1 Register Set.

In register mode, set this register set by using software. In link mode, the descriptor read data is automatically set in the Next0 register set.

These register set values are loaded to the Current Register Set and used for DMA transfer.




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 0 | DA | All 0 | R/W | Destination Address<br>Sets the start address of the DMA transfer destination. |

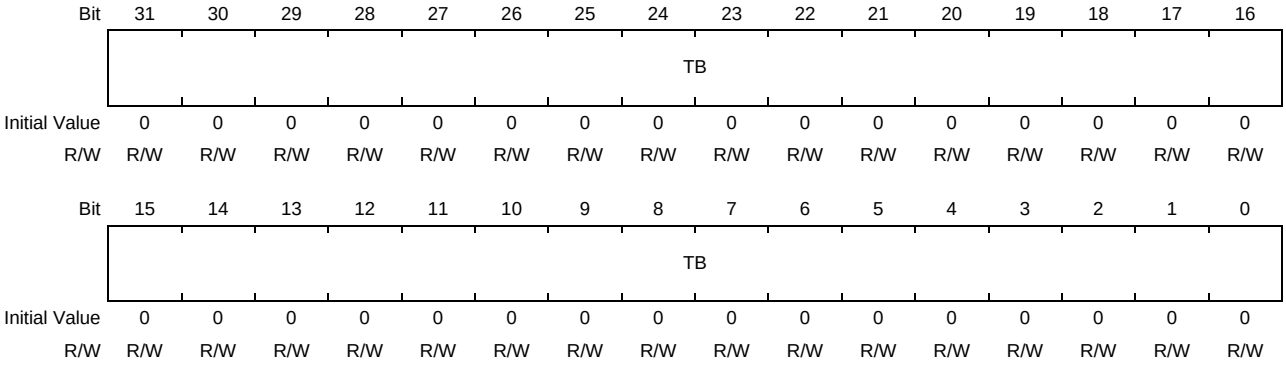
### 14.4.3 Next Transaction Byte Register n (N0TB_n, N1TB_n)

This register sets the total transfer byte count (DMA transaction) of DMA channel (n = 0 to 15) which is to be executed next.

N0TB_n is for the Next0 Register Set, and N1TB_n is for the Next1 Register Set.

In register mode, set this register set by using software. In link mode, the descriptor read data is automatically set in the Next0 register set.

These register set values are loaded to the Current Register Set and used for DMA transfer.




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 0 | TB | All 0 | R/W | Transaction Byte<br>Sets the total transfer byte count.<br>*Note*: Do not start a DMA transaction with 0 set in this register. |

### 14.4.4 Current Source Address Register n (CRSA_n)

This register indicates the DMA transfer source address of DMA channel n (n = 0 to 15).

The values are loaded from the Next0/1 register set in register mode or from the descriptor read data in link mode. This register cannot be written by software.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | CRSA | | | | | | | | | | | | | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | CRSA | | | | | | | | | | | | | | | |
| 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 0 | CRSA | All 0 | R | Current Source Address Register<br>Indicates the read address of the next DMA transaction.<br>The value automatically increments during the DMA transaction.<br>(The value is fixed when 1 is set in SAD of the CHCFG_n register.)<br>The value increments when a read transfer starts.<br>Read this register after DMA stops (0 is set in EN of the CHSTAT_n register). (Any value obtained during the DMA operation should be handled as a reference value.) |

### 14.4.5 Current Destination Address Register n (CRDA_n)

This register indicates the DMA transfer destination address of DMA channel n (n = 0 to 15).

The values are loaded from the Next0/1 register set in register mode or from the descriptor read data in link mode. This register cannot be written by software.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | CRDA | | | | | | | | | | | | | | | |
| 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 |

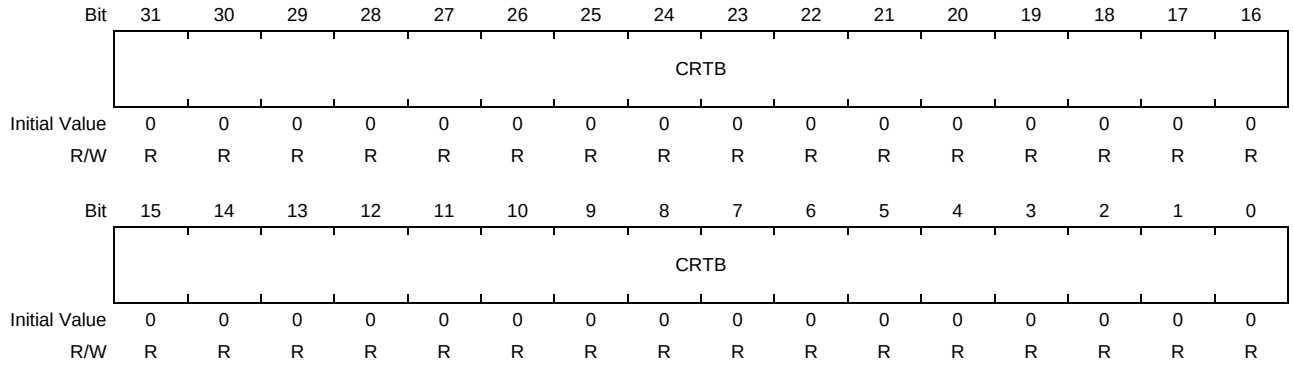
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | CRDA | | | | | | | | | | | | | | | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 0 | CRDA | All 0 | R | Current Destination Address Register<br>Indicates the write address of the next DMA transaction.<br>The value automatically increments during the DMA transaction. (The value is fixed when 1 is set in DAD of the CHCFG_n register.)<br>The value increments when a write transfer starts.<br>Read this register after DMA stops (0 is set in EN of the CHSTAT_n register). (Any value obtained during the DMA operation should be handled as a reference value.) |

### 14.4.6 Current Transaction Byte Register n (CRTB_n)

This register indicates the total transfer byte count of DMA channel n (n = 0 to 15). The value of this register becomes 0 when the transaction ends.

The values are loaded from the Next0/1 register set in register mode or from the descriptor read data in link mode. This register cannot be written by software.




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 0 | CRTB | All 0 | R | Current Transaction Byte Register<br>Indicates the remaining transfer byte count of the currently executed DMA transaction.<br>The value automatically decrements during the DMA transaction.<br>The value decrements when a write transfer is completed.<br>Read this register after DMA stops (0 is set in EN of the CHSTAT_n register). (Any value obtained during the DMA operation should be handled as a reference value.) |

### 14.4.7 Channel Status Register n (CHSTAT_n)

This register indicates the status of DMA channel n (n = 0 to 15).

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | INTMSK |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | MODE | DER | DW | DL | SR | TC | END | ER | SUS | TACT | RQST | 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 | R | R | R | R | R | R | R | R |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 17 | — | All 0 | R | Reserved. Set 0. A read operation results in 0 being read. |
| 16 | INTMSK | 0 | R | Indicates the temporary mask status of the DMA transfer end interrupt.<br>1: Masked temporarily<br>0: Unmasked temporarily<br>Set condition(s):<br>• When SETINTMSK (CHCTRL_n) is set to 1<br>Reset condition(s):<br>• When CLRINTMSK (CHCTRL_n) is set to 1<br>• When SWRST (CHCTRL_n) is set to 1 |
| 15 to 12 | — | All 0 | R | Reserved. Set 0. A read operation results in 0 being read. |
| 11 | MODE | 0 | R | DMA Mode<br>Indicates the DMA mode. It corresponds to the value set in the DMS bit of the CHCFG_n register.<br>0: Register mode<br>1: Link mode |
| 10 | DER | 0 | R | Descriptor Error<br>Indicates whether the link valid value of the read descriptor is invalid (LV = 0) (this is not dependent on the DIM level of the descriptor). If a descriptor error has occurred, the transfer is stopped but no DMA error interrupt occurs.<br>0: Descriptor Error not detected<br>1: Descriptor Error detected<br>Set condition(s):<br>• When the LV value loaded with the descriptor in link mode is 0<br>Reset condition(s):<br>• When SWRST (CHCTRL_n) is set to 1 |
| 9 | DW | 0 | R | Descriptor WriteBack<br>Indicates the descriptor writeback status. The bit maintains 1 if a bus error is received during descriptor writeback.<br>0: Operation other than writeback is being performed for the header in link mode.<br>1: (ER = 0)<br>Writeback is being performed for the header in link mode.<br>(ER = 1)<br>A bus error occurs during writeback for the header in link mode.<br>Set condition(s):<br>• When header writeback in link mode starts<br>Reset condition(s):<br>• When header writeback in link mode ends with an OK response<br>• When SWRST (CHCTRL_n) is set to 1 |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 8 | DL | 0 | R | Descriptor Load<br>Indicates whether the descriptor is being loaded. The bit maintains 1 if a bus error is received during descriptor load.<br>0: Operation other than descriptor load<br>1: (ER = 0)<br>Descriptor load is in progress in link mode.<br>(ER = 1)<br>A bus error occurs during descriptor load in link mode.<br>Set condition(s):<br>• When descriptor load in link mode starts<br>Reset condition(s):<br>• When descriptor load in link mode ends with an OK response<br>• When SWRST (CHCTRL_n) is set to 1 |
| 7 | SR | 0 | R | Selected Register Set<br>Indicates the register set currently selected in register mode.<br>0: Next0 Register Set<br>1: Next1 Register Set<br>Set condition(s):<br>• When RSEL (CHCFG_n) is set to 1<br>Reset condition(s):<br>• When RSEL (CHCFG_n) is set to 0 |
| 6 | TC | 0 | R | Terminal Count<br>Indicates whether the DMA transaction is completed.<br>0: DMA transfer not completed<br>1: DMA transfer completed<br>Set condition(s):<br>• When data equivalent to the total transfer byte count set in the CRTB register has been transferred in register mode<br>• When data equivalent to the total transfer byte count set in the CRTB register has been transferred in link mode, with 1 set in WBD of the descriptor header<br>• When descriptor writeback is completed in link mode, with 0 set in WBD of the descriptor header<br>Clear condition(s):<br>• When the CLRTC (CHCTRL_n) bit is set to 1<br>• When the SWRST (CHCTRL_n) bit is set to 1 |
| 5 | END | 0 | R | DMAEND Interrupted<br>Indicates whether the DMA transaction is completed and whether the DMA transfer end interrupt has occurred.<br>0: DMA transfer not completed<br>1: DMA transfer completed<br>Set condition(s):<br>• When one of the set conditions for the TC bit is met and 0 is set in DEM of the CHCFG_n register<br>• When the descriptor is read in link mode and both LV of the header and DIM are set to 0<br>Clear condition(s):<br>• When CLREND (CHCTRL_n) is set to 1<br>• When SWRST (CHCTRL_n) is set to 1 |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 4 | ER | 0 | R | Error bit<br>Indicates that a DMA error interrupt has occurred because an error response has been received from the transfer source or destination and a bus error has occurred during the DMA transfer.<br>0: No bus error has occurred<br>1: A DMA error interrupt has occurred due to a bus error<br>Set condition(s):<br>• When a bus error has occurred during a bus cycle<br>Clear condition(s):<br>• When SWRST (CHCTRL_n) is set to 1 |
| 3 | SUS | 0 | R | Suspend<br>Indicates whether the channel is suspended.<br>0: Channel_n not suspended<br>1: Channel_n suspended<br>Set condition(s):<br>• When SETSUS (CHCTRL_n) is set to 1 during a DMA transfer on Channel_n, creating a SUSPEND status internally<br>Clear condition(s):<br>• When CLRSUS (CHCTRL_n) is set to 1<br>• When CLREN (CHCTRL_n) is set to 1 |
| 2 | TACT | 0 | R | Transaction Active<br>Indicates whether the DMAC is active. This bit is intended to check that the channel is completely inactive.<br>0: DMA on Channel_n inactive<br>1: DMA on Channel_n active<br>Set condition(s):<br>• When a DMA transaction starts on Channel_n<br>Clear condition(s):<br>• When a DMA transaction is completed |
| 1 | RQST | 0 | R | Request<br>Indicates whether a transfer request is being received.<br>0: DMA transfer request not being received<br>1: DMA transfer request being received<br>Set condition(s):<br>• When the STG bit (CHCTRL_n) is set to 1 (auto request)<br>• When a transfer request is received from the DMA request source set in the CHCFG_n register<br>Clear condition(s):<br>• When SWRST (CHCTRL_n) is set to 1<br>• When CLRRQ (CHCTRL_n) is set to 1<br>• When a transfer is executed on the side specified by REQD (CHCFG_n) in single transfer mode (TM = 0).<br>• When all DMA transactions are completed in register mode (the transaction ends with REN set to 0)<br>• When the DMA transfer of the last descriptor (LE = 1) is completed in link mode<br>• When descriptor read stops (LV = 0) in link mode<br>• When a bus error is received due to an error response |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 0 | EN | 0 | R | Enable<br>Indicates whether the operation of DMA channel n is enabled or disabled.<br>0: Operation disabled<br>1: Operation enabled<br>Set condition(s):<br>• When SETEN (CHCTRL_n) is set to 1<br>Clear condition(s):<br>• When SWRST (CHCTRL_n) is set to 1<br>• When CLREN (CHCTRL_n) is set to 1<br>• When a bus error is received due to an error response during the transfer<br>• When all DMA transactions are completed in register mode (the transaction ends with REN set to 0)<br>• When the DMA transfer of the last descriptor (LE = 1) is completed in link mode (writeback when WBD is set to 0)<br>• When descriptor read stops (LV = 0) in link mode |

**CAUTION**

If the ER bit is set to 1 for any transfer, the whole transfer should be handled as invalid.

To suspend a DMA transaction, mask or clear the transfer request or clear the Enable bit (for the procedure, see **Section 14.7.8(2), Transfer Stop**).

If a transfer request from an on-chip peripheral module input is made concurrently with an auto request (by setting 1 in the STG bit) for the same one channel, the trigger source that takes effect cannot be identified. Make sure that only one of these transfer requests is used in the system.

When transfer is requested by an auto request, wait for the last requested DMA transfer to complete (use the Current Register or other data to check the status) before setting the STG bit for the next transfer request.

### 14.4.8 Channel Control Register n (CHCTRL_n)

This register controls the DMA transfer operation on DMA channel n (n = 0 to 15).

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CLRINTMSK | SETINTMSK |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | CLRSUS | SETSUS | — | CLRTC | CLRENDD | CLRRQ | SWRST | STG | CLREN | SETEN |
| 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/W | R/W | R/W | R/W | R/W | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 18 | — | All 0 | R | Reserved. Set 0. A read operation results in 0 being read. |
| 17 | CLRINTMSK | 0 | R/W | When this bit is set to 1, the mask of the DMA transfer end interrupt is cleared. Also, the INTMSK bit of the CHSTAT_n register is set to 0.<br>If the mask is cleared when 1 is set in both LVINT of the DCTRL register and END of the CHSTAT_n register, the DMA transfer end interrupt becomes active. (It does not become active when 0 is set in LVINT.)<br>A read operation results in 0 being read.<br>1: Clears the mask set by SETINTMSK.<br>0: Does not affect the operation. |
| 16 | SETINTMSK | 0 | R/W | When this bit is set to 1, the DMA transfer end interrupt is temporarily masked. Also, the INTMSK bit of the CHSTAT_n register is set to 1.<br>A read operation results in 0 being read.<br>1: Masks the DMA transfer end interrupt.<br>0: Does not affect the operation. |
| 15 to 10 | — | All 0 | R | Reserved. Set 0. A read operation results in 0 being read. |
| 9 | CLRSUS | 0 | R/W | Clear Suspend<br>Clears the suspend status. Setting this bit to 1 when 1 is set in SUS of the CHSTAT_n register can clear the suspend status.<br>An attempt to read this bit results in 0 being read.<br>1: Clears the suspend status of the current DMA transfer.<br>0: Does not affect the operation. |
| 8 | SETSUS | 0 | R/W | Set Suspend<br>Suspends the current DMA transfer. Setting this bit to 1 when 1 is set in EN of the CHSTAT_n register can suspend the current DMA transfer.<br>An attempt to read this bit results in 0 being read.<br>1: Suspends the current DMA transfer.<br>0: Does not affect the operation. |
| 7 | — | 0 | R | Reserved. Set 0. A read operation results in 0 being read. |
| 6 | CLRTC | 0 | R/W | Clear TC bit<br>Setting this bit to 1 can clear the TC bit of the CHSTAT_n register. An attempt to read this bit results in 0 being read.<br>1: Clears the TC bit.<br>0: Does not affect the operation. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 5 | CLREND | 0 | R/W | Clear End bit<br>Setting this bit to 1 can clear the END bit of the CHSTAT_n register.<br>Also, the DMA transfer end interrupt is cleared.<br>An attempt to read this bit results in 0 being read.<br>1: Clears the END bit.<br>0: Does not affect the operation. |
| 4 | CLRRQ | 0 | R/W | Clear Request bit<br>Setting this bit to 1 can clear the RQST bit of the CHSTAT_n register.<br>An attempt to read this bit results in 0 being read.<br>1: Clears the RQST bit.<br>0: Does not affect the operation. |
| 3 | SWRST | 0 | R/W | Software Reset<br>Setting this bit to 1 can clear the channel status register (CHSTAT_n). When setting this bit to 1, make sure that both the EN bit and TACT bit are set to 0.<br>An attempt to read this bit results in 0 being read.<br>1: Resets the channel status register.<br>0: Does not affect the operation. |
| 2 | STG | 0 | R/W | Software Trigger<br>Setting this bit to 1 sets an auto request. If this bit is set at the same time the SWRST bit is set, the clear operation by the SWRST bit takes precedence.<br>An attempt to read this bit results in 0 being read.<br>1: Sets a transfer request triggered by an auto request (sets 1 in the RQST bit).<br>0: Does not affect the operation. |
| 1 | CLREN | 0 | R/W | Clear Enable<br>Setting this bit to 1 can clear the EN bit (for details, see **Section 14.7.8(2), Transfer Stop**).<br>An attempt to read this bit results in 0 being read.<br>1: Stops the DMA transfer (clears the EN bit).<br>0: Does not affect the operation. |
| 0 | SETEN | 0 | R/W | Set Enable<br>Enables a DMA transfer on DMA channel n. If this bit is set at the same time the SWRST bit is set, the clear operation by the SWRST bit takes precedence and the transfer does not start.<br>An attempt to read this bit results in 0 being read.<br>1: Enables a DMA transfer (sets 1 in the EN bit).<br>0: Does not affect the operation. |

### 14.4.9 Channel Configuration Register n (CHCFG_n)

This register controls the DMA transfer operation on DMA channel n (n = 0 to 15).

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | DMS | REN | RSW | RSEL | SBE | — | — | DEM | — | TM | DAD | SAD | DDS[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 | R | R/W | R | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | SDS[3:0] | | | | — | AM[2:0] | | | — | LVL | HIEN | LOEN | REQD | SEL[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 | R/W | R/W | 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 |
|---|---|---|---|---|
| 31 | DMS | 0 | R/W | DMA Mode Select<br>Sets the DMA mode.<br>0: Register mode (initial value)<br>1: Link mode |
| 30 | REN | 0 | R/W | Register Set Enable<br>After a DMA transaction is completed, DMA transfers are continued using the Next register set selected by RSEL. This bit is valid only in register mode.<br>0: Does not continue DMA transfers.<br>1: Continues DMA transfers.<br>Set condition(s):<br>• When 1 is written to this bit<br>Clear condition(s):<br>• When 0 is written to this bit<br>• When a DMA transaction is completed, with REN set to 1 |
| 29 | RSW | 0 | R/W | Register Select Switch<br>Inverts RSEL automatically after a DMA transaction is completed. This bit is valid only in register mode.<br>0: Does not invert RSEL automatically after a DMA transaction (initial value).<br>1: Inverts RSEL automatically after a DMA transaction. |
| 28 | RSEL | 0 | R/W | Register Set Select<br>Selects the Next register set to be executed next. This bit is valid only in register mode. When RSW is set to 1, this bit is inverted automatically when a DMA transaction is completed.<br>0: Executes the Next0 Register Set (initial value).<br>1: Executes the Next1 Register Set.<br>Transition condition(s):<br>• When a DMA transaction is completed, with RSW set to 1 |
| 27 | SBE | 0 | R/W | Sweep Buffer Enable<br>Selects whether to sweep (write) the data already read into the buffer and stop the DMA transfer if the Enable bit is cleared to 0 during a DMA transaction.<br>The sweep mode is available only when REQD is set to 0.<br>0: Stops the DMA transfer without sweeping the buffer (initial value).<br>1: Stops the DMA transfer after sweeping the buffer. |
| 26, 25 | — | 0 | R | Reserved. Set 0. A read operation results in 0 being read. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 24 | DEM | 0 | R/W | DMA Transfer End Interrupt Mask<br>Masks the DMA transfer end interrupt for register mode transfer.<br>If 1 is set in this bit when a DMA transfer end interrupt is output, the DMA transfer end interrupt signal is not asserted. In this case, DEM is cleared to 0 automatically.<br>0: Does not mask the DMA transfer end interrupt (initial value).<br>1: Masks the DMA transfer end interrupt.<br>Clear condition(s):<br>• When a DMA transaction is completed with DEM set to 1 |
| 23 | — | 0 | R | Reserved. Set 0. A read operation results in 0 being read. |
| 22 | TM | 0 | R/W | Transfer Mode<br>Sets the DMA transfer mode.<br>0: Single transfer mode (initial value)<br>1: Block transfer mode |
| 21 | DAD | 0 | R/W | Sets the destination address counting direction of DMA channel n.<br>0: Increment (initial value)<br>1: Fixed |
| 20 | SAD | 0 | R/W | Sets the source address counting direction of DMA channel n.<br>0: Increment (initial value)<br>1: Fixed |
| 19 to 16 | DDS[3:0] | 0000 | R/W | Destination Data Size<br>Sets the DMA transfer size of the transfer destination. |

| Value | Size | Remark |
|---|---|---|
| 0000 | 8 bits | Initial value |
| 0001 | 16 bits | |
| 0010 | 32 bits | |
| 0011 | 64 bits | |
| 0100 | 128 bits | |
| 0101 | 256 bits | |
| 0110 | 512 bits | |
| 0111 | 1024 bits | |
| Other than the above | — | Setting prohibited |

| | | | | |
|---|---|---|---|---|
| 15 to 12 | SDS[3:0] | 0000 | R/W | Source Data Size<br>Sets the DMA transfer size of the transfer source. |

| Value | Size | Remark |
|---|---|---|
| 0000 | 8 bits | Initial value |
| 0001 | 16 bits | |
| 0010 | 32 bits | |
| 0011 | 64 bits | |
| 0100 | 128 bits | |
| 0101 | 256 bits | |
| 0110 | 512 bits | |
| 0111 | 1024 bits | |
| Other than the above | — | Setting prohibited |

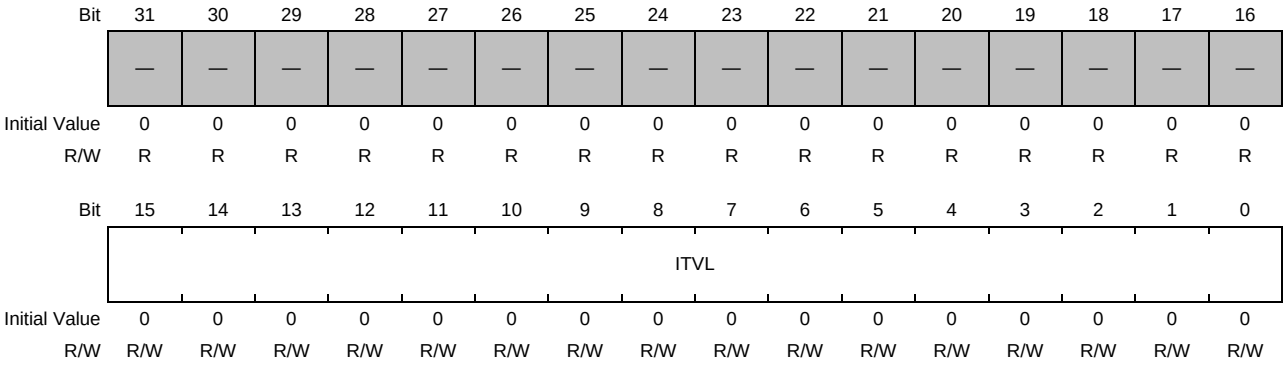
| | | | | |
|---|---|---|---|---|
| 11 | — | 0 | R | Reserved. Set 0. A read operation results in 0 being read. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 10 to 8 | AM[2:0] | 000 | R/W | ACK Mode<br>Sets the DMAACK output mode.<br>In case of Auto-Request Mode, DMAACK is not output. Any value can be set in AM bit.<br>In case of On-Chip Peripheral Module Request Mode, see the **Table 14.3** On-Chip Module Requests and set the appropriate value.<br>000: Not used.<br>001: Level mode (active until the transfer request from an on-chip peripheral module input becomes inactive)<br>01x: Bus cycle mode (active while the DMA transfer is in a bus cycle)<br>1xx: DMAACK not to be output |
| 7 | — | 0 | R | Reserved. Set 0. A read operation results in 0 being read. |
| 6 | LVL | 0 | R/W | Level<br>Selects whether to detect a DMA request based on the level or edge of the signal.<br>0: Detects based on the edge (initial value).<br>1: Detects based on the level. |
| 5 | HIEN | 0 | R/W | High Enable<br>Selects whether to detect a DMA request using the High level or rising edge of the signal.<br>When LVL = 0:<br>HIEN = 1: Detects a request in response to the rising edge of the signal.<br>HIEN = 0: Does not detect a request in response to the rising edge of the signal (initial value).<br>When LVL = 1:<br>HIEN = 1: Detects a request when the signal is at the High level.<br>HIEN = 0: Does not detect a request even when the signal is at the High level (initial value). |
| 4 | LOEN | 0 | R/W | Low Enable<br>Selects whether to detect a DMA request using the Low level or falling edge of the signal.<br>When LVL = 0:<br>LOEN = 1: Detects a request in response to the falling edge of the signal.<br>LOEN = 0: Does not detect a request in response to the falling edge of the signal (initial value).<br>When LVL = 1:<br>LOEN = 1: Detects a request when the signal is at the Low level.<br>LOEN = 0: Does not detect a request even when the signal is at the Low level (initial value). |
| 3 | REQD | 0 | R/W | Request Direction<br>Selects whether DMAREQ selected by the SEL bit is the source or destination. This bit is also used to define when DMAACK is to become active.<br>0: Source; DMAACK is to become active when read (initial value).<br>1: Destination; DMAACK is to become active when written. |
| 2 to 0 | SEL[2:0] | 000 | R/W | These bits are used to set a DMAC channel. Set one of the following values so that the channel set by the SEL bits matches the CHCFG_n channel.<br>000: CH0/CH8<br>001: CH1/CH9<br>010: CH2/CH10<br>011: CH3/CH11<br>100: CH4/CH12<br>101: CH5/CH13<br>110: CH6/CH14<br>111: CH7/CH15 |

### 14.4.10 Channel Interval Register n (CHITVL_n)

This register sets the transfer interval for DMA channel n (n = 0 to 15).

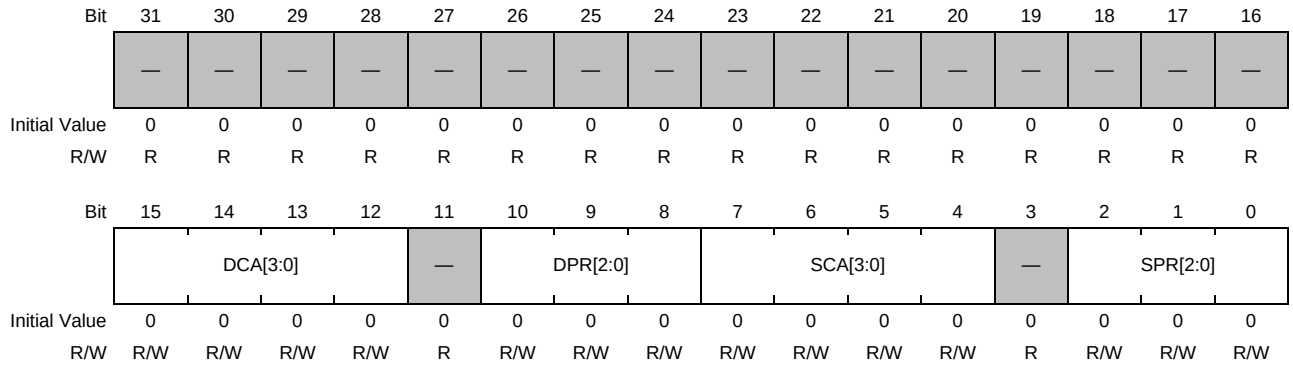
For details, see **Section 14.7.6, Interval Count Function**.




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 16 | — | All 0 | R | Reserved. Set 0. A read operation results in 0 being read. |
| 15 to 0 | ITVL | All 0 | R/W | Sets the channel transfer interval. |

### 14.4.11 Channel Extension Register n (CHEXT_n)

This is an extension register for DMA channel n (n = 0 to 15).




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 16 | — | All 0 | R | Set 0. A read operation results in 0 being read. |
| 15 to 12 | DCA[3:0] | 0000 | R/W | Destination CACHE<br>Sets the value to be output to AWCACHE[3:0] for DMA write transfer.<br>See Note 1 below. |
| 11 | — | 0 | R | Set 0. A read operation results in 0 being read. |
| 10 to 8 | DPR[2:0] | 000 | R/W | Destination PROT<br>Sets the value to be output to AWPROT[2:0] for DMA write transfer.<br>In the case of non-secure access, the value output as AWPROT[1] is fixed to 1 regardless of the setting of the DPR[1] bit.<br>See Note 2 below. |
| 7 to 4 | SCA[3:0] | 0000 | R/W | Source CACHE<br>Sets the value to be output to ARCACHE[3:0] for DMA read transfer.<br>See Note 1 below. |
| 3 | — | 0 | R | Set 0. A read operation results in 0 being read. |
| 2 to 0 | SPR[2:0] | 000 | R/W | Source PROT<br>Sets the value to be output to ARPROT[2:0] for DMA read transfer.<br>In the case of non-secure access, the value output as ARPROT[1] is fixed to 1 regardless of the setting of the SPR[1] bit.<br>See Note 2 below. |

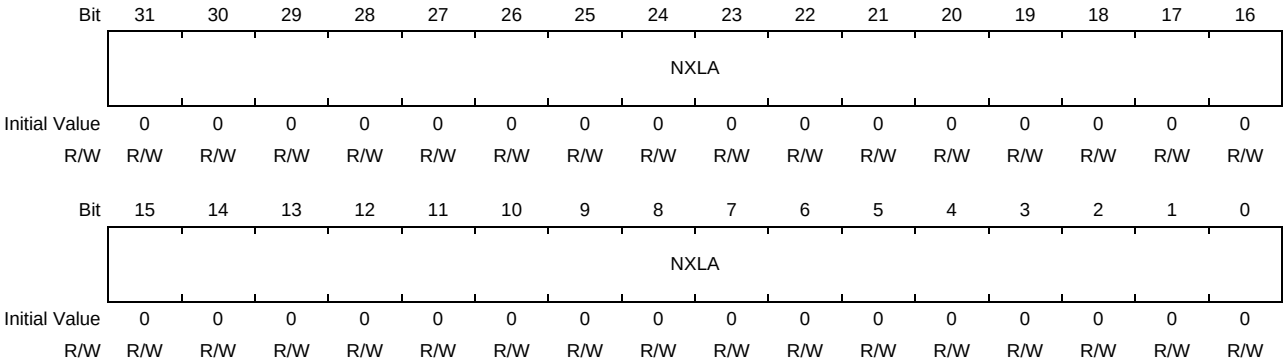
Note 1. Cache support: Bits SCA and DCA are used to change the settings.

Note 2. Protection unit support: Bits SPR and DPR are used to change the settings.
For the setting value, see AMBA AXI Protocol Specification from Arm Limited.

## 14.4.12 Next Link Address Register n (NXLA_n)

This is a 32-bit register that sets the link address of DMA channel n (n = 0 to 15).

For information about the link mode, see **Section 14.6.3, Link Mode**.

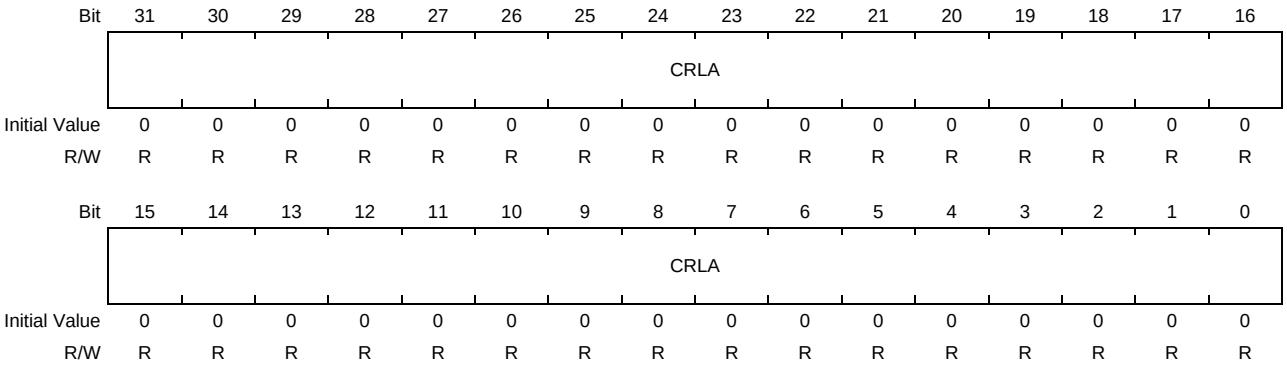



| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 0 | NXLA | All 0 | R/W | Sets a link address. The low-order 2 bits are masked with 0s. Only an address aligned with a 4-byte boundary can be set. |

## 14.4.13 Current Link Address Register n (CRLA_n)

This is a 32-bit register that indicates the link address of DMA channel n (n = 0 to 15).

For information about the link mode, see **Section 14.6.3, Link Mode**.




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 0 | CRLA | All 0 | R | Indicates the address of the currently executed descriptor. |

### 14.4.14 DMA Control Register (DCTRL_0_7, DCTRL_8_15)

This register sets the transfer type for descriptor access and the arbitration between channels.
(DCTRL_0_7 is common for channels 0 to 7 and DCTRL_8_15 is common for channels 8 to 15.)

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | LWCA | | | | — | LWPR | | | LDCA | | | | — | LDPR | | |
| 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/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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | LVINT | PR |
| 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 |
|---|---|---|---|---|
| 31 to 28 | LWCA | 0000 | R/W | Link WriteBack CACHE<br>Sets the value to be output to AWCACHE[3:0] during descriptor writeback in link mode.<br>For the setting value, see Note in **Section 14.4.11, Channel Extension Register n (CHEXT_n)**. |
| 27 | — | 0 | R | Reserved. Set 0. The initial value is 0. |
| 26 to 24 | LWPR | 000 | R/W | Link WriteBack PROT<br>Sets the value to be output to AWPROT[2:0] during descriptor writeback in link mode.<br>In the case of non-secure access, the value output as AWPROT[1] is fixed to 1 regardless of the setting of the LWPR[1] bit.<br>For the setting value, see AMBA AXI Protocol Specification from Arm Limited. |
| 23 to 20 | LDCA | 0000 | R/W | Link Descriptor CACHE<br>Sets the value to be output to ARCACHE[3:0] during descriptor load in link mode.<br>For the setting value, see Note in **Section 14.4.11, Channel Extension Register n (CHEXT_n)**. |
| 19 | — | 0 | R | Reserved. Set 0. The initial value is 0. |
| 18 to 16 | LDPR | 000 | R/W | Link Descriptor PROT<br>Sets the value to be output to ARPROT[2:0] during descriptor load in link mode.<br>In the case of non-secure access, the value output as ARPROT[1] is fixed to 1 regardless of the setting of the LDPR[1] bit.<br>For the setting value, see AMBA AXI Protocol Specification from Arm Limited. |
| 15 to 2 | — | All 0 | R | Reserved. Set 0. A read operation results in 0 being read. |
| 1 | LVINT | 0 | R/W | Sets whether to use pulse output or level output for the DMA transfer end interrupt and DMA error interrupt. Set pulse output for this product.<br>0: Pulse output (initial value)<br>1: Level output |
| 0 | PR | 0 | R/W | Sets the transfer priority control mode between channels (see **Section 14.7.2, Priority Control for DMA Channels**).<br>0: Fixed priority mode (initial value)<br>1: Round robin mode |

### 14.4.15 DMA Status EN Register (DSTAT_EN_0_7)

This register indicates the EN bit status of the CHSTAT_n register (n = 0 to 7).

Writing to this register does not affect the values of the 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | EN7 | EN6 | EN5 | EN4 | EN3 | EN2 | EN1 | EN0 |
| 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 8 | — | All 0 | R | Reserved. Set 0. A read operation results in 0 being read. |
| 7 | EN7 | 0 | R | Indicates the EN bit status of DMA channel 7. |
| 6 | EN6 | 0 | R | Indicates the EN bit status of DMA channel 6. |
| 5 | EN5 | 0 | R | Indicates the EN bit status of DMA channel 5. |
| 4 | EN4 | 0 | R | Indicates the EN bit status of DMA channel 4. |
| 3 | EN3 | 0 | R | Indicates the EN bit status of DMA channel 3. |
| 2 | EN2 | 0 | R | Indicates the EN bit status of DMA channel 2. |
| 1 | EN1 | 0 | R | Indicates the EN bit status of DMA channel 1. |
| 0 | EN0 | 0 | R | Indicates the EN bit status of DMA channel 0. |

### 14.4.16 DMA Status EN Register (DSTAT_EN_8_15)

This register indicates the EN bit status of the CHSTAT_n register (n = 8 to 15).

Writing to this register does not affect the values of the 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | EN15 | EN14 | EN13 | EN12 | EN11 | EN10 | EN9 | EN8 |
| 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 8 | — | All 0 | R | Reserved. Set 0. A read operation results in 0 being read. |
| 7 | EN15 | 0 | R | Indicates the EN bit status of DMA channel 15. |
| 6 | EN14 | 0 | R | Indicates the EN bit status of DMA channel 14. |
| 5 | EN13 | 0 | R | Indicates the EN bit status of DMA channel 13. |
| 4 | EN12 | 0 | R | Indicates the EN bit status of DMA channel 12. |
| 3 | EN11 | 0 | R | Indicates the EN bit status of DMA channel 11. |
| 2 | EN10 | 0 | R | Indicates the EN bit status of DMA channel 10. |
| 1 | EN9 | 0 | R | Indicates the EN bit status of DMA channel 9. |
| 0 | EN8 | 0 | R | Indicates the EN bit status of DMA channel 8. |

### 14.4.17 DMA Status ER Register (DSTAT_ER_0_7)

This register indicates the ER bit status of the CHSTAT_n register (n = 0 to 7).

Writing to this register does not affect the values of the 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | ER7 | ER6 | ER5 | ER4 | ER3 | ER2 | ER1 | ER0 |
| 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 8 | — | All 0 | R | Reserved. Set 0. A read operation results in 0 being read. |
| 7 | ER7 | 0 | R | Indicates the ER bit status of DMA channel 7. |
| 6 | ER6 | 0 | R | Indicates the ER bit status of DMA channel 6. |
| 5 | ER5 | 0 | R | Indicates the ER bit status of DMA channel 5. |
| 4 | ER4 | 0 | R | Indicates the ER bit status of DMA channel 4. |
| 3 | ER3 | 0 | R | Indicates the ER bit status of DMA channel 3. |
| 2 | ER2 | 0 | R | Indicates the ER bit status of DMA channel 2. |
| 1 | ER1 | 0 | R | Indicates the ER bit status of DMA channel 1. |
| 0 | ER0 | 0 | R | Indicates the ER bit status of DMA channel 0. |

### 14.4.18 DMA Status ER Register (DSTAT_ER_8_15)

This register indicates the ER bit status of the CHSTAT_n register (n = 8 to 15).

Writing to this register does not affect the values of the 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | ER15 | ER14 | ER13 | ER12 | ER11 | ER10 | ER9 | ER8 |
| 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 8 | — | All 0 | R | Reserved. Set 0. A read operation results in 0 being read. |
| 7 | ER15 | 0 | R | Indicates the ER bit status of DMA channel 15. |
| 6 | ER14 | 0 | R | Indicates the ER bit status of DMA channel 14. |
| 5 | ER13 | 0 | R | Indicates the ER bit status of DMA channel 13. |
| 4 | ER12 | 0 | R | Indicates the ER bit status of DMA channel 12. |
| 3 | ER11 | 0 | R | Indicates the ER bit status of DMA channel 11. |
| 2 | ER10 | 0 | R | Indicates the ER bit status of DMA channel 10. |
| 1 | ER9 | 0 | R | Indicates the ER bit status of DMA channel 9. |
| 0 | ER8 | 0 | R | Indicates the ER bit status of DMA channel 8. |

### 14.4.19 DMA Status END Register (DSTAT_END_0_7)

This register indicates the END bit status of the CHSTAT_n register (n = 0 to 7).

Writing to this register does not affect the values of the 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | END7 | END6 | END5 | END4 | END3 | END2 | END1 | END0 |
| 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 8 | — | All 0 | R | Reserved. Set 0. A read operation results in 0 being read. |
| 7 | END7 | 0 | R | Indicates the END bit status of DMA channel 7. |
| 6 | END6 | 0 | R | Indicates the END bit status of DMA channel 6. |
| 5 | END5 | 0 | R | Indicates the END bit status of DMA channel 5. |
| 4 | END4 | 0 | R | Indicates the END bit status of DMA channel 4. |
| 3 | END3 | 0 | R | Indicates the END bit status of DMA channel 3. |
| 2 | END2 | 0 | R | Indicates the END bit status of DMA channel 2. |
| 1 | END1 | 0 | R | Indicates the END bit status of DMA channel 1. |
| 0 | END0 | 0 | R | Indicates the END bit status of DMA channel 0. |

### 14.4.20 DMA Status END Register (DSTAT_END_8_15)

This register indicates the END bit status of the CHSTAT_n register (n = 8 to 15).

Writing to this register does not affect the values of the 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | END15 | END14 | END13 | END12 | END11 | END10 | END9 | END8 |
| 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 8 | — | All 0 | R | Reserved. Set 0. A read operation results in 0 being read. |
| 7 | END15 | 0 | R | Indicates the END bit status of DMA channel 15. |
| 6 | END14 | 0 | R | Indicates the END bit status of DMA channel 14. |
| 5 | END13 | 0 | R | Indicates the END bit status of DMA channel 13. |
| 4 | END12 | 0 | R | Indicates the END bit status of DMA channel 12. |
| 3 | END11 | 0 | R | Indicates the END bit status of DMA channel 11. |
| 2 | END10 | 0 | R | Indicates the END bit status of DMA channel 10. |
| 1 | END9 | 0 | R | Indicates the END bit status of DMA channel 9. |
| 0 | END8 | 0 | R | Indicates the END bit status of DMA channel 8. |

### 14.4.21 DMA Status TC Register (DSTAT_TC_0_7)

This register indicates the TC bit status of the CHSTAT_n register (n = 0 to 7).

Writing to this register does not affect the values of the 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | TC7 | TC6 | TC5 | TC4 | TC3 | TC2 | TC1 | TC0 |
| 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 8 | — | All 0 | R | Reserved. Set 0. A read operation results in 0 being read. |
| 7 | TC7 | 0 | R | Indicates the TC bit status of DMA channel 7. |
| 6 | TC6 | 0 | R | Indicates the TC bit status of DMA channel 6. |
| 5 | TC5 | 0 | R | Indicates the TC bit status of DMA channel 5. |
| 4 | TC4 | 0 | R | Indicates the TC bit status of DMA channel 4. |
| 3 | TC3 | 0 | R | Indicates the TC bit status of DMA channel 3. |
| 2 | TC2 | 0 | R | Indicates the TC bit status of DMA channel 2. |
| 1 | TC1 | 0 | R | Indicates the TC bit status of DMA channel 1. |
| 0 | TC0 | 0 | R | Indicates the TC bit status of DMA channel 0. |

### 14.4.22 DMA Status TC Register (DSTAT_TC_8_15)

This register indicates the TC bit status of the CHSTAT_n register (n = 8 to 15).

Writing to this register does not affect the values of the 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | TC15 | TC14 | TC13 | TC12 | TC11 | TC10 | TC9 | TC8 |
| 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 8 | — | All 0 | R | Reserved. Set 0. A read operation results in 0 being read. |
| 7 | TC15 | 0 | R | Indicates the TC bit status of DMA channel 15. |
| 6 | TC14 | 0 | R | Indicates the TC bit status of DMA channel 14. |
| 5 | TC13 | 0 | R | Indicates the TC bit status of DMA channel 13. |
| 4 | TC12 | 0 | R | Indicates the TC bit status of DMA channel 12. |
| 3 | TC11 | 0 | R | Indicates the TC bit status of DMA channel 11. |
| 2 | TC10 | 0 | R | Indicates the TC bit status of DMA channel 10. |
| 1 | TC9 | 0 | R | Indicates the TC bit status of DMA channel 9. |
| 0 | TC8 | 0 | R | Indicates the TC bit status of DMA channel 8. |

### 14.4.23 DMA Status SUS Register (DSTAT_SUS_0_7)

This register indicates the SUS bit status of the CHSTAT_n register (n = 0 to 7).

Writing to this register does not affect the values of the 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | SUS7 | SUS6 | SUS5 | SUS4 | SUS3 | SUS2 | SUS1 | SUS0 |
| 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 8 | — | All 0 | R | Reserved. Set 0. A read operation results in 0 being read. |
| 7 | SUS7 | 0 | R | Indicates the SUS bit status of DMA channel 7. |
| 6 | SUS6 | 0 | R | Indicates the SUS bit status of DMA channel 6. |
| 5 | SUS5 | 0 | R | Indicates the SUS bit status of DMA channel 5. |
| 4 | SUS4 | 0 | R | Indicates the SUS bit status of DMA channel 4. |
| 3 | SUS3 | 0 | R | Indicates the SUS bit status of DMA channel 3. |
| 2 | SUS2 | 0 | R | Indicates the SUS bit status of DMA channel 2. |
| 1 | SUS1 | 0 | R | Indicates the SUS bit status of DMA channel 1. |
| 0 | SUS0 | 0 | R | Indicates the SUS bit status of DMA channel 0. |

### 14.4.24 DMA Status SUS Register (DSTAT_SUS_8_15)

This register indicates the SUS bit status of the CHSTAT_n register (n = 8 to 15).

Writing to this register does not affect the values of the 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | SUS15 | SUS14 | SUS13 | SUS12 | SUS11 | SUS10 | SUS9 | SUS8 |
| 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 8 | — | All 0 | R | Reserved. Set 0. A read operation results in 0 being read. |
| 7 | SUS15 | 0 | R | Indicates the SUS bit status of DMA channel 15. |
| 6 | SUS14 | 0 | R | Indicates the SUS bit status of DMA channel 14. |
| 5 | SUS13 | 0 | R | Indicates the SUS bit status of DMA channel 13. |
| 4 | SUS12 | 0 | R | Indicates the SUS bit status of DMA channel 12. |
| 3 | SUS11 | 0 | R | Indicates the SUS bit status of DMA channel 11. |
| 2 | SUS10 | 0 | R | Indicates the SUS bit status of DMA channel 10. |
| 1 | SUS9 | 0 | R | Indicates the SUS bit status of DMA channel 9. |
| 0 | SUS8 | 0 | R | Indicates the SUS bit status of DMA channel 8. |

### 14.4.25 DMA Extension Resource Selectors 0 to 7 (DMARS0 to DMARS7)

DMARSn (n = 0 to 7) are 32-bit readable/writable registers that specify the DMA transfer sources from peripheral modules in each channel. DMARS0 is for channels 0 and 1, DMARS1 is for channels 2 and 3, and so on.

**Table 14.3** shows the specifiable combinations.

Some on-chip peripheral modules in this product use the same signal both for an interrupt request and for a DMA transfer request. If such a module is selected by a DMARSn register, the signal works as a DMA transfer request signal and interrupt requests to the interrupt controller are masked. To enable the interrupt, clear the setting of DMARSn (set all MID[7:0] and RID[1:0] to 0). See **Table 14.3** for possible combinations of MID[7:0] and RID[1:0].

- DMARS0

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | CH1 MID[7:0] | | | | | | | | CH1 RID[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/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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | CH0 MID[7:0] | | | | | | | | CH0 RID[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/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

- DMARS1

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | CH3 MID[7:0] | | | | | | | | CH3 RID[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/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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | CH2 MID[7:0] | | | | | | | | CH2 RID[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/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

- DMARS2

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | CH5 MID[7:0] | | | | | | | | CH5 RID[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/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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | CH4 MID[7:0] | | | | | | | | CH4 RID[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/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

- DMARS3

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | CH7 MID[7:0] | | | | | | | | CH7 RID[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/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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | CH6 MID[7:0] | | | | | | | | CH6 RID[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/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

- DMARS4

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | CH9 MID[7:0] | | | | | | | | CH9 RID[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/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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | CH8 MID[7:0] | | | | | | | | CH8 RID[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/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

- DMARS5

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | CH11 MID[7:0] | | | | | | | | CH11 RID[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/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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | CH10 MID[7:0] | | | | | | | | CH10 RID[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/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

- DMARS6

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | CH13 MID[7:0] | | | | | | | | CH13 RID[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/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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | CH12 MID[7:0] | | | | | | | | CH12 RID[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/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

- DMARS7

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | CH15 MID[7:0] | | | | | | | | CH15 RID[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/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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | CH14 MID[7:0] | | | | | | | | CH14 RID[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/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

## 14.5 Operation

When there is a DMA transfer request, the DMAC starts the transfer according to the predetermined channel priority order; when the transfer end conditions are satisfied, it ends the transfer. Transfers can be requested in two modes: auto request, and on-chip peripheral module request.

### 14.5.1 Transfer Flow

After the next source address register (N0SA_n, N1SA_n), next destination address register (N0DA_n, N1DA_n), next transaction byte register (N0TB_n, N1TB_n), channel control register (CHCTRL_n), channel configuration register (CHCFG_n), channel extension register (CHEXT_n), DMA control register (DCTRL_0_7, DCTRL_8_15), and DMA extension resource selector (DMARSn) are set for the target transfer conditions, the DMAC transfers data according to the following procedure:

1. Checks to see if transfer is enabled (EN = 0 and TACT = 0 in channel status register).
2. Clears the channel status register (set 1 in the SWRST bit of the channel control register).
3. Enables DMA transfer (set 1 in the SETEN bit of the channel control register).
4. When a transfer request comes and transfer is enabled, the DMAC transfers one transfer unit of data (depending on the DDS[3:0] and SDS[3:0] bit settings). For an auto request, the transfer begins automatically when 1 is set in the STG bit of the channel control register. The CRTB_n value will be decremented by 1 for each transfer.
5. If 0 is set in the REN bit of the channel configuration register when transfer has been completed for the specified count (when CRTB_n reaches 0), transfer ends normally. If the DEM bit of the channel configuration register is set to 0 at this time, a DMA transfer end interrupt is sent to the CPU. If the REN bit is 1 when CRTB_n reaches 0, transfer operations are continued with the values of N0SA_n, N1SA_n, N0DA_n, N1DA_n, N0TB_n, and N1TB_n set by the RSEL bit of the channel configuration register until there are no more transfer requests.
6. When an address error in the DMAC is generated, the transfer is stopped. Transfers are also stopped when 1 is set in the CLREN bit of CHCTRL_n.

### 14.5.2 DMA Transfer Requests

DMA transfer requests are basically generated in either the data transfer source or destination, but they can also be generated in on-chip peripheral modules that are neither the transfer source nor destination. Transfers can be requested in two modes: auto request, and on-chip peripheral module request. On-chip peripheral module request is selected by the DMARS0 to DMARS7 registers.

#### (1) Auto-Request Mode

When there is no transfer request signal from an external source, as in a memory-to-memory transfer or a transfer between memory and an on-chip peripheral module unable to request a transfer, auto-request mode allows the DMAC to automatically generate a transfer request signal internally. When the STG bit in channel control register n is set to 1, the transfer begins so long as the TACT bit in channel status register is 0.

#### (2) On-Chip Peripheral Module Request Mode

In this mode, the transfer is performed in response to the DMA transfer request signal from an on-chip peripheral module.

When a transfer request signal is sent in on-chip peripheral module request mode while DMA transfer is enabled, the DMA transfer is performed.

The DMA transfer request signals to be sent from on-chip peripheral modules or external pin input are listed in **Table 14.3**.

The transfer source or destination is fixed for some on-chip peripheral module requests. For details, see **Table 14.3**.

Table 14.3 On-Chip Module Requests (1/4)

DMA Transfer

| Request Source | DMA Transfer Request Signal | Transfer Source | Transfer Destination | DMARSn | | CHCFG_n*1 | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | MID [7:0] | RID [1:0] | TM | AM [2:0] | LVL | HIEN | LOEN | REQD | SEL[2:0] |
| GTM ch0 | OSTM0TINT (compare match) | Arbitrary | Arbitrary | 0000_1000 | 11 | 0/1 | 010 | 0 | 1 | 0 | 0/1 | CH0:000<br>CH1:001<br>CH2:010<br>CH3:011<br>CH4:100<br>CH5:101<br>CH6:110<br>CH7:111<br>CH8:000<br>CH9:001<br>CH10:010<br>CH11:011<br>CH12:100<br>CH13:101<br>CH14:110<br>CH15:111 |
| GTM ch1 | OSTM1TINT (compare match) | Arbitrary | Arbitrary | 0000_1001 | 11 | 0/1 | 010 | 0 | 1 | 0 | 0/1 | |
| GTM ch2 | OSTM2TINT (compare match) | Arbitrary | Arbitrary | 0000_1010 | 11 | 0/1 | 010 | 0 | 1 | 0 | 0/1 | |
| MTU3 ch0 | TGIA0 (input capture/compare match) | Arbitrary | Arbitrary | 0000_1100 | 11 | 0/1 | 001 | 0 | 1 | 0 | 0/1 | |
| | TGIB0 (input capture/compare match) | Arbitrary | Arbitrary | 0000_1101 | 11 | 0/1 | 001 | 0 | 1 | 0 | 0/1 | |
| | TGIC0 (input capture/compare match) | Arbitrary | Arbitrary | 0000_1110 | 11 | 0/1 | 001 | 0 | 1 | 0 | 0/1 | |
| | TGID0 (input capture/compare match) | Arbitrary | Arbitrary | 0000_1111 | 11 | 0/1 | 001 | 0 | 1 | 0 | 0/1 | |
| MTU3 ch1 | TGIA1 (input capture/compare match) | Arbitrary | Arbitrary | 0001_0000 | 11 | 0/1 | 001 | 0 | 1 | 0 | 0/1 | |
| | TGIB1 (input capture/compare match) | Arbitrary | Arbitrary | 0001_0001 | 11 | 0/1 | 001 | 0 | 1 | 0 | 0/1 | |
| MTU3 ch2 | TGIA2 (input capture/compare match) | Arbitrary | Arbitrary | 0001_0010 | 11 | 0/1 | 001 | 0 | 1 | 0 | 0/1 | |
| | TGIB2 (input capture/compare match) | Arbitrary | Arbitrary | 0001_0011 | 11 | 0/1 | 001 | 0 | 1 | 0 | 0/1 | |
| MTU3 ch3 | TGIA3 (input capture/compare match) | Arbitrary | Arbitrary | 0001_0100 | 11 | 0/1 | 001 | 0 | 1 | 0 | 0/1 | |
| | TGIB3 (input capture/compare match) | Arbitrary | Arbitrary | 0001_0101 | 11 | 0/1 | 001 | 0 | 1 | 0 | 0/1 | |
| | TGIC3 (input capture/compare match) | Arbitrary | Arbitrary | 0001_0110 | 11 | 0/1 | 001 | 0 | 1 | 0 | 0/1 | |
| | TGID3 (input capture/compare match) | Arbitrary | Arbitrary | 0001_0111 | 11 | 0/1 | 001 | 0 | 1 | 0 | 0/1 | |
| MTU3 ch4 | TGIA4 (input capture/compare match) | Arbitrary | Arbitrary | 0001_1000 | 11 | 0/1 | 001 | 0 | 1 | 0 | 0/1 | |
| | TGIB4 (input capture/compare match) | Arbitrary | Arbitrary | 0001_1001 | 11 | 0/1 | 001 | 0 | 1 | 0 | 0/1 | |
| | TGIC4 (input capture/compare match) | Arbitrary | Arbitrary | 0001_1010 | 11 | 0/1 | 001 | 0 | 1 | 0 | 0/1 | |
| | TGID4 (input capture/compare match) | Arbitrary | Arbitrary | 0001_1011 | 11 | 0/1 | 001 | 0 | 1 | 0 | 0/1 | |

Table 14.3 On-Chip Module Requests (2/4)

DMA Transfer

| Request Source | DMA Transfer Request Signal | Transfer Source | Transfer Destination | DMARSn | | CHCFG_n*1 | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | MID [7:0] | RID [1:0] | TM | AM [2:0] | LVL | HIEN | LOEN | REQD | SEL[2:0] |
| MTU3 ch4 | TCIV4 (input capture/compare match) | Arbitrary | Arbitrary | 0001_1100 | 11 | 0/1 | 001 | 0 | 1 | 0 | 0/1 | CH0:000<br>CH1:001<br>CH2:010<br>CH3:011<br>CH4:100<br>CH5:101<br>CH6:110<br>CH7:111<br>CH8:000<br>CH9:001<br>CH10:010<br>CH11:011<br>CH12:100<br>CH13:101<br>CH14:110<br>CH15:111 |
| MTU3 ch5 | TGIU5 (input capture/compare match) | Arbitrary | Arbitrary | 0001_1101 | 11 | 0/1 | 001 | 0 | 1 | 0 | 0/1 | |
| | TGIV5 (input capture/compare match) | Arbitrary | Arbitrary | 0001_1110 | 11 | 0/1 | 001 | 0 | 1 | 0 | 0/1 | |
| | TGIW5 (input capture/compare match) | Arbitrary | Arbitrary | 0001_1111 | 11 | 0/1 | 001 | 0 | 1 | 0 | 0/1 | |
| MTU3 ch6 | TGIA6 (input capture/compare match) | Arbitrary | Arbitrary | 0010_0000 | 11 | 0/1 | 001 | 0 | 1 | 0 | 0/1 | |
| | TGIB6 (input capture/compare match) | Arbitrary | Arbitrary | 0010_0001 | 11 | 0/1 | 001 | 0 | 1 | 0 | 0/1 | |
| | TGIC6 (input capture/compare match) | Arbitrary | Arbitrary | 0010_0010 | 11 | 0/1 | 001 | 0 | 1 | 0 | 0/1 | |
| | TGID6 (input capture/compare match) | Arbitrary | Arbitrary | 0010_0011 | 11 | 0/1 | 001 | 0 | 1 | 0 | 0/1 | |
| MTU3 ch7 | TGIA7 (input capture/compare match) | Arbitrary | Arbitrary | 0010_0100 | 11 | 0/1 | 001 | 0 | 1 | 0 | 0/1 | |
| | TGIB7 (input capture/compare match) | Arbitrary | Arbitrary | 0010_0101 | 11 | 0/1 | 001 | 0 | 1 | 0 | 0/1 | |
| | TGIC7 (input capture/compare match) | Arbitrary | Arbitrary | 0010_0110 | 11 | 0/1 | 001 | 0 | 1 | 0 | 0/1 | |
| | TGID7 (input capture/compare match) | Arbitrary | Arbitrary | 0010_0111 | 11 | 0/1 | 001 | 0 | 1 | 0 | 0/1 | |
| | TCIV7 (input capture/compare match) | Arbitrary | Arbitrary | 0010_1000 | 11 | 0/1 | 001 | 0 | 1 | 0 | 0/1 | |
| MTU3 ch8 | TGIA8 (input capture/compare match) | Arbitrary | Arbitrary | 0010_1001 | 11 | 0/1 | 001 | 0 | 1 | 0 | 0/1 | |
| | TGIB8 (input capture/compare match) | Arbitrary | Arbitrary | 0010_1010 | 11 | 0/1 | 001 | 0 | 1 | 0 | 0/1 | |
| | TGIC8 (input capture/compare match) | Arbitrary | Arbitrary | 0010_1011 | 11 | 0/1 | 001 | 0 | 1 | 0 | 0/1 | |
| | TGID8 (input capture/compare match) | Arbitrary | Arbitrary | 0010_1100 | 11 | 0/1 | 001 | 0 | 1 | 0 | 0/1 | |

Table 14.4 On-Chip Module Requests (3/4)

DMA Transfer

| Request Source | DMA Transfer Request Signal | Transfer Source | Transfer Destination | DMARSn | | CHCFG_n*1 | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | MID [7:0] | RID [1:0] | TM | AM [2:0] | LVL | HIEN | LOEN | REQD | SEL[2:0] |
| SSIF ch0 | INT_ssif_dma_rx_0 (receive data full) | SSIF RDR_0 | Arbitrary | 1001_0101 | 10 | 0 | 010 | 0 | 1 | 0 | 0 | CH0:000<br>CH1:001<br>CH2:010<br>CH3:011<br>CH4:100<br>CH5:101<br>CH6:110<br>CH7:111<br>CH8:000<br>CH9:001<br>CH10:010<br>CH11:011<br>CH12:100<br>CH13:101<br>CH14:110<br>CH15:111 |
| | INT_ssif_dma_tx_0 (transmit data empty) | Arbitrary | SSIF TDR_0 | 1001_0101 | 01 | 0 | 010 | 0 | 1 | 0 | 1 | |
| I2C ch0 | INTRIIC_RI0 (receive data full) | RIIC0DRR | Arbitrary | 1001_1010 | 10 | 0 | 010 | 0 | 1 | 0 | 0 | |
| | INTRIIC_TI0 (transmit data empty) | Arbitrary | RIIC0DRT | 1001_1010 | 01 | 0 | 010 | 0 | 1 | 0 | 1 | |
| I2C ch1 | INTRIIC_RI1 (receive data full) | RIIC1DRR | Arbitrary | 1001_1011 | 10 | 0 | 010 | 0 | 1 | 0 | 0 | |
| | INTRIIC_TI1 (transmit data empty) | Arbitrary | RIIC1DRT | 1001_1011 | 01 | 0 | 010 | 0 | 1 | 0 | 1 | |
| SCIF ch0 | RXI0 (receive FIFO data full) | FRDR0 | Arbitrary | 1001_1110 | 10 | 0 | 100 | 1 | 1 | 0 | 0 | |
| | TXI0 (transmit FIFO data empty) | Arbitrary | FTDR0 | 1001_1110 | 01 | 0 | 100 | 1 | 1 | 0 | 1 | |
| SCIF ch1 | RXI1 (receive FIFO data full) | FRDR1 | Arbitrary | 1001_1111 | 10 | 0 | 100 | 1 | 1 | 0 | 0 | |
| | TXI1 (transmit FIFO data empty) | Arbitrary | FTDR1 | 1001_1111 | 01 | 0 | 100 | 1 | 1 | 0 | 1 | |
| SCIF ch2 | RXI2 (receive FIFO data full) | FRDR2 | Arbitrary | 1010_0000 | 10 | 0 | 100 | 1 | 1 | 0 | 0 | |
| | TXI2 (transmit FIFO data empty) | Arbitrary | FTDR2 | 1010_0000 | 01 | 0 | 100 | 1 | 1 | 0 | 1 | |
| SCIF ch3 | RXI3 (receive FIFO data full) | FRDR3 | Arbitrary | 1010_0001 | 10 | 0 | 100 | 1 | 1 | 0 | 0 | |
| | TXI3 (transmit FIFO data empty) | Arbitrary | FTDR3 | 1010_0001 | 01 | 0 | 100 | 1 | 1 | 0 | 1 | |
| SCIF ch4 | RXI4 (receive FIFO data full) | FRDR4 | Arbitrary | 1010_0010 | 10 | 0 | 100 | 1 | 1 | 0 | 0 | |
| | TXI4 (transmit FIFO data empty) | Arbitrary | FTDR4 | 1010_0010 | 01 | 0 | 100 | 1 | 1 | 0 | 1 | |
| SCIg ch0 | RXI0 (receive FIFO data full) | RDR0 | Arbitrary | 1010_0011 | 10 | 0 | 010 | 1 | 1 | 0 | 0 | |
| | TXI0 (transmit FIFO data empty) | Arbitrary | TDR0 | 1010_0011 | 01 | 0 | 010 | 1 | 1 | 0 | 1 | |
| SCIg ch1 | RXI1 (receive FIFO data full) | RDR1 | Arbitrary | 1010_0100 | 10 | 0 | 010 | 1 | 1 | 0 | 0 | |
| | TXI1 (transmit FIFO data empty) | Arbitrary | TDR1 | 1010_0100 | 01 | 0 | 010 | 1 | 1 | 0 | 1 | |

Table 14.4 On-Chip Module Requests (4/4)

DMA Transfer

| Request Source | DMA Transfer Request Signal | Transfer Source | Transfer Destination | DMARSn | | CHCFG_n*1 | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | MID [7:0] | RID [1:0] | TM | AM [2:0] | LVL | HIEN | LOEN | REQD | SEL[2:0] |
| RSPI ch0 | SPRI0 (receive buffer full) | SPDR0 | Arbitrary | 1010_0101 | 10 | 0 | 010 | 1 | 1 | 0 | 0 | CH0:000<br>CH1:001 |
| | SPTI0 (transmit buffer empty) | Arbitrary | SPDR0 | 1010_0101 | 01 | 0 | 010 | 1 | 1 | 0 | 1 | CH2:010<br>CH3:011 |
| RSPI ch1 | SPRI1 (receive buffer full) | SPDR1 | Arbitrary | 1010_0110 | 10 | 0 | 010 | 1 | 1 | 0 | 0 | CH4:100<br>CH5:101 |
| | SPTI1 (transmit buffer empty) | Arbitrary | SPDR1 | 1010_0110 | 01 | 0 | 010 | 1 | 1 | 0 | 1 | CH6:110<br>CH7:111 |
| | | | | | | | | | | | | CH8:000 |
| | | | | | | | | | | | | CH9:001 |
| | | | | | | | | | | | | CH10:010 |
| | | | | | | | | | | | | CH11:011 |
| | | | | | | | | | | | | CH12:100 |
| | | | | | | | | | | | | CH13:101 |
| | | | | | | | | | | | | CH14:110<br>CH15:111 |

Note 1. CHCFG_n setting value

TM 0: Single transfer
1: Block transfer

AM 001: ACK level output
010: ACK bus cycle output
100: No ACK

LVL 0: REQ edge detection
1: REQ level detection

REQD 0: ACK output at read
1: ACK output at write

## 14.6 DMA Mode

### 14.6.1 Mode Setting

The DMS field of the CHCFG_n register can be used to toggle between register mode and link mode.

Table 14.5 DMA Mode Setting

| DMS (CHCFG_n) | Mode | Description |
|---|---|---|
| 0 | Register mode | A DMA transfer is executed using the values set in the Next Register Set. |
| 1 | Link mode | A DMA transfer is executed using the descriptor set in the Current register. The DMAC repeatedly loads the descriptor and executes the DMA transfer unless otherwise set by the descriptor or stopped by the control register. |

### 14.6.2 Register Mode

In register mode, a DMA transfer is executed using the values set in the internal registers.

Two sets of the source address, destination address, and transfer byte count (Next0 Register Set and Next1 Register Set) can be set. It is possible to select the Next register set to be used for the DMA transfer, as well as to execute two Next register sets continuously for the DMA transfer.

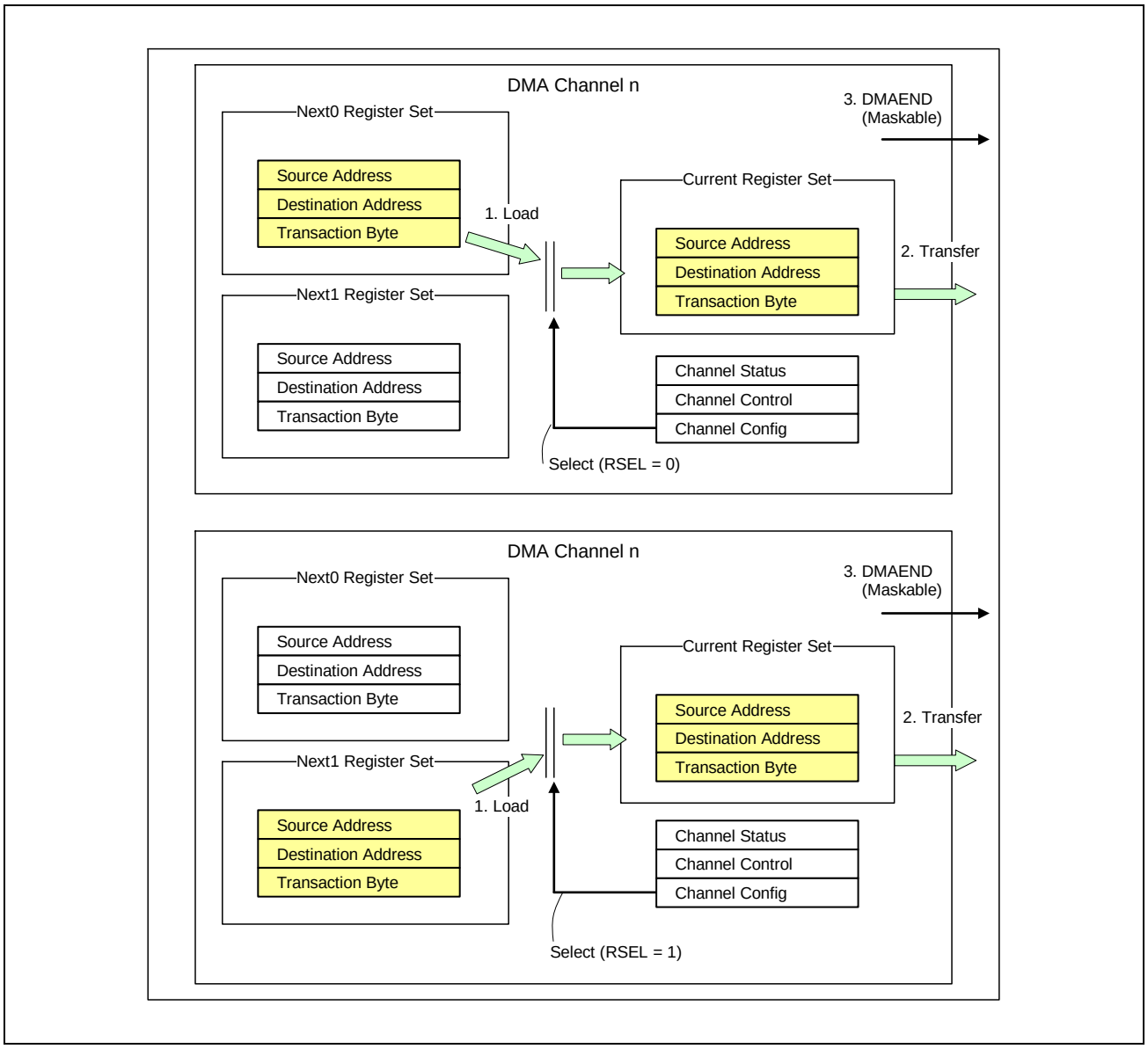



Figure 14.2 Outline of Normal Register Mode

The above figure shows how the transfer is executed when the Next0 Register Set is used (upper part of the figure) and when the Next1 Register Set is used (lower part of the figure).

### (1) Operation Flow

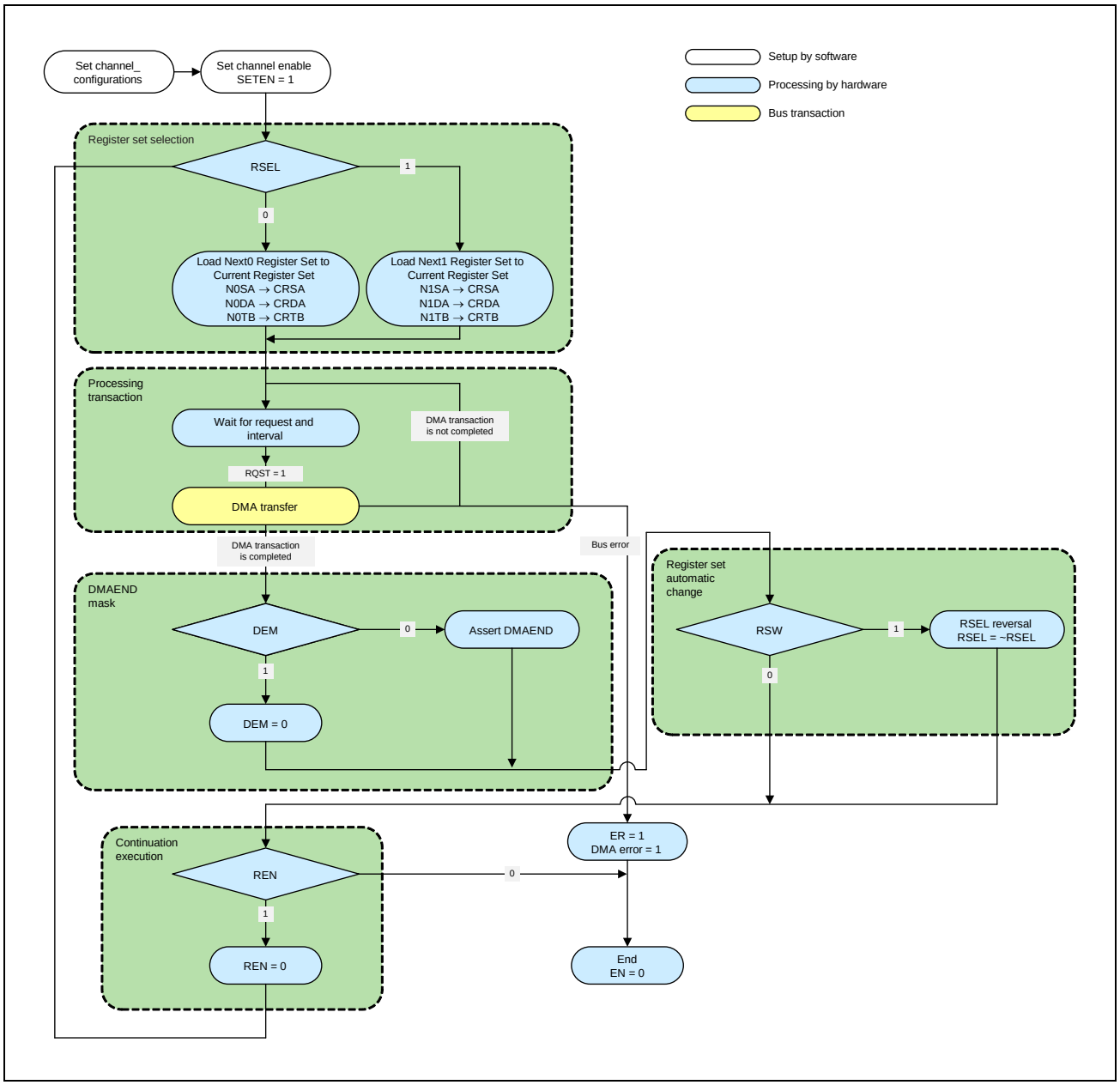



Figure 14.3 Register Mode Flow

<Explanation of the register mode flow>

1. Channel setting (set channel configuration)

   The Next0 or Next1 register set (destination address, source address, and total transfer byte count) is set.

   In the Channel register set, the DMA register set (REQ, DMAACK, transfer size, etc.) is set. (See **Section 14.7, DMA Transfer**.)

2. Register set selection (register set selection)

   When 1 is set in EN, the values set in the Next register set selected by RSEL are loaded to the Current register set.

3. DMA transaction (processing transaction)

   A DMA transfer is executed according to the set values. For details of the transfer, see **Section 14.7, DMA Transfer**.

4. DMA transfer end interrupt mask (DMAINT mask)

   The DMA transfer end interrupt is masked according to the value set in the DEM bit of CHCFG_n. When 1 is set in DEM, the DMA transfer end interrupt is not output. Also, immediately after that, DEM is automatically cleared to 0.

5. Automatic register set change (register set automatic change)

   Whether to use the other Next register set is determined by the value set in the RSW bit of CHCFG_n.

6. Continuation of execution (continuation execution)

   Whether to continue the execution of the DMA transfer is determined by the value set in the REN bit of CHCFG_n. When 1 is set in REN, the execution of the DMA transfer is continued. Also, immediately after that, REN is automatically cleared to 0.

### (2) Register Setting

#### (a) Register mode setting

Select the register set to be executed.

Table 14.6 Register Mode Setting

| DMS (CHCFG_n) | RSEL (CHCFG_n) | Description |
|---|---|---|
| 0 | 0 | Executes the Next0 Register Set. |
| | 1 | Executes the Next1 Register Set. |

#### (b) DMA transfer end interrupt mask setting

The DMA transfer end interrupt can be masked individually for each register set.

Table 14.7 DMAINT Mask Setting

| DEM (CHCFG_n) | Operation |
|---|---|
| 0 | When the DMA transaction is completed, a DMA transfer end interrupt is issued. |
| 1 | Even when the DMA transaction is completed, a DMA transfer end interrupt is not issued. After the DMA transaction is completed, DEM is cleared to 0 by hardware. |

### (c) Automatic register set execution setting

After DMA transfers, the DMA transaction of the selected register set is automatically executed.

Table 14.8 Automatic Register Set Execution Setting

| REN (CHCFG_n) | Operation | Remark |
|---|---|---|
| 0 | When the DMA transaction of the register set selected by RSEL is completed, the EN bit is cleared and the DMA operation ends. | Set this value to execute a DMA transaction once. |
| 1 | When a DMA transaction is completed, the DMAC continues to execute a DMA transfer by using the data set in the selected register set. When continuous transfers are successful, REN is cleared to 0. | Set this value to continuously execute DMA transfers by using the data set in separate register sets. |

### (d) Automatic register set change setting

When 1 is set in REN, the DMAC can automatically change to the register set to be executed next, after a DMA transaction is completed.

Table 14.9 Automatic Register Set Change Setting

| RSW (CHCFG_n) | Operation | Remark |
|---|---|---|
| 0 | If 1 is set in REN when a DMA transaction is completed, the register set is not changed. | Set this value to use only one register set. |
| 1 | If 1 is set in REN when a DMA transaction is completed, the value of RSEL is automatically inverted and the other register set is selected. | Set this value to change the register set. |

### (3) Setting Examples

#### (a) When only the Next0 register set is used

Table 14.10 Register Mode Setting Example 1

| DMS (CHCFG_n) | RSEL (CHCFG_n) | DEM (CHCFG_n) | RSW (CHCFG_n) | REN (CHCFG_n) |
|---|---|---|---|---|
| 0 (Register mode) | 0 (Next0) | 0 (not masked) | 0 (not switched) | 0 (not continuously executed) |

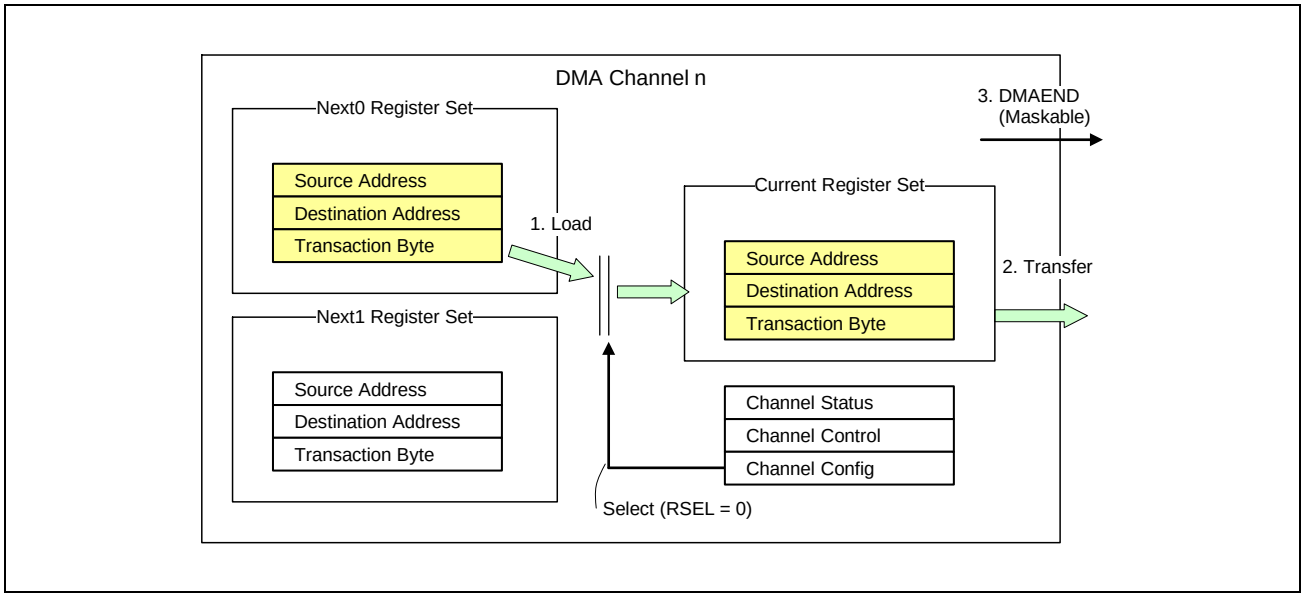



Figure 14.4 Register Mode Setting Example 1

1. 1 is set in EN (SETEN = 1), and the Next0 register set is loaded to the Current register set.
2. A DMA transaction is executed according to the values set in the Current register set and Channel register set.
3. Because 0 is set in DEM, the DMA transfer end interrupt is issued after the DMA transaction is completed.
4. Because 0 is set in REN, EN is cleared to 0 and the DMA transaction ends.

### (b) When two register sets are used continuously

Table 14.11 Automatic Register Set Execution Setting

| DMS (CHCFG_n) | RSEL (CHCFG_n) | DEM (CHCFG_n) | RSW (CHCFG_n) | REN (CHCFG_n) |
|---|---|---|---|---|
| 0 (Register mode) | 0 (Next0) | 1 (masked) | 1 (switched) | 1 (continuously executed) |

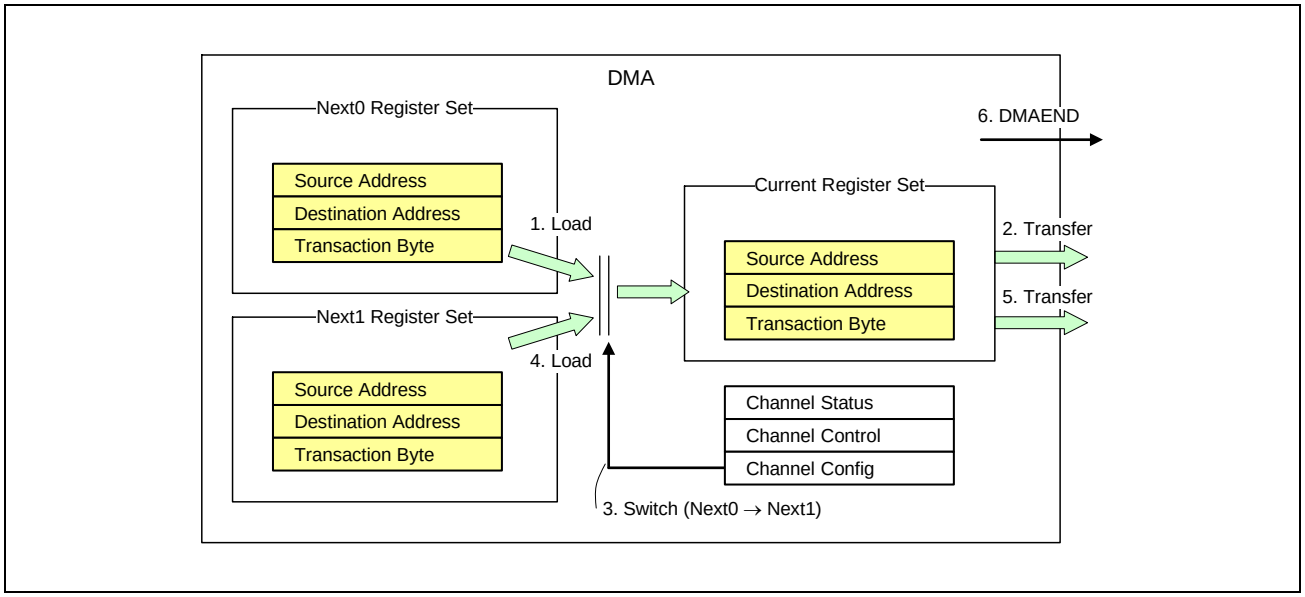



Figure 14.5 Register Mode Setting Example 2

1. 1 is set in EN (SETEN = 1), and the Next0 register set is loaded to the Current register set.
2. A DMA transaction is executed according to the values set in the Current register set and Channel register set.
3. Because 1 is set in DEM, DMA transfer end interrupt is not output after the DMA transaction is completed. Also, DEM is automatically cleared to 0.
4. Because 1 is set in REN, the execution is continued. Also, REN is automatically cleared to 0.
5. Because 1 is set in RSW, the register set to be executed next is switched (RSEL = 0 → 1).
6. The Next1 register set is loaded to the Current register set.
7. A DMA transaction is executed according to the values set in the Current register set and Channel register set.
8. Because 0 is set in DEM, the DMA transfer end interrupt is issued after the DMA transaction is completed.
9. Because 0 is set in REN, EN is cleared to 0 and the DMA transaction ends.

### 14.6.3 Link Mode

In link mode, a descriptor stored in external memory is loaded as set values and a DMA transaction is executed using the loaded values. The DMAC contains a Next Link address and a Current Link address for each channel, and these addresses are used to set the descriptor address to be executed next and to indicate the descriptor address of the currently executed DMA transaction, respectively.

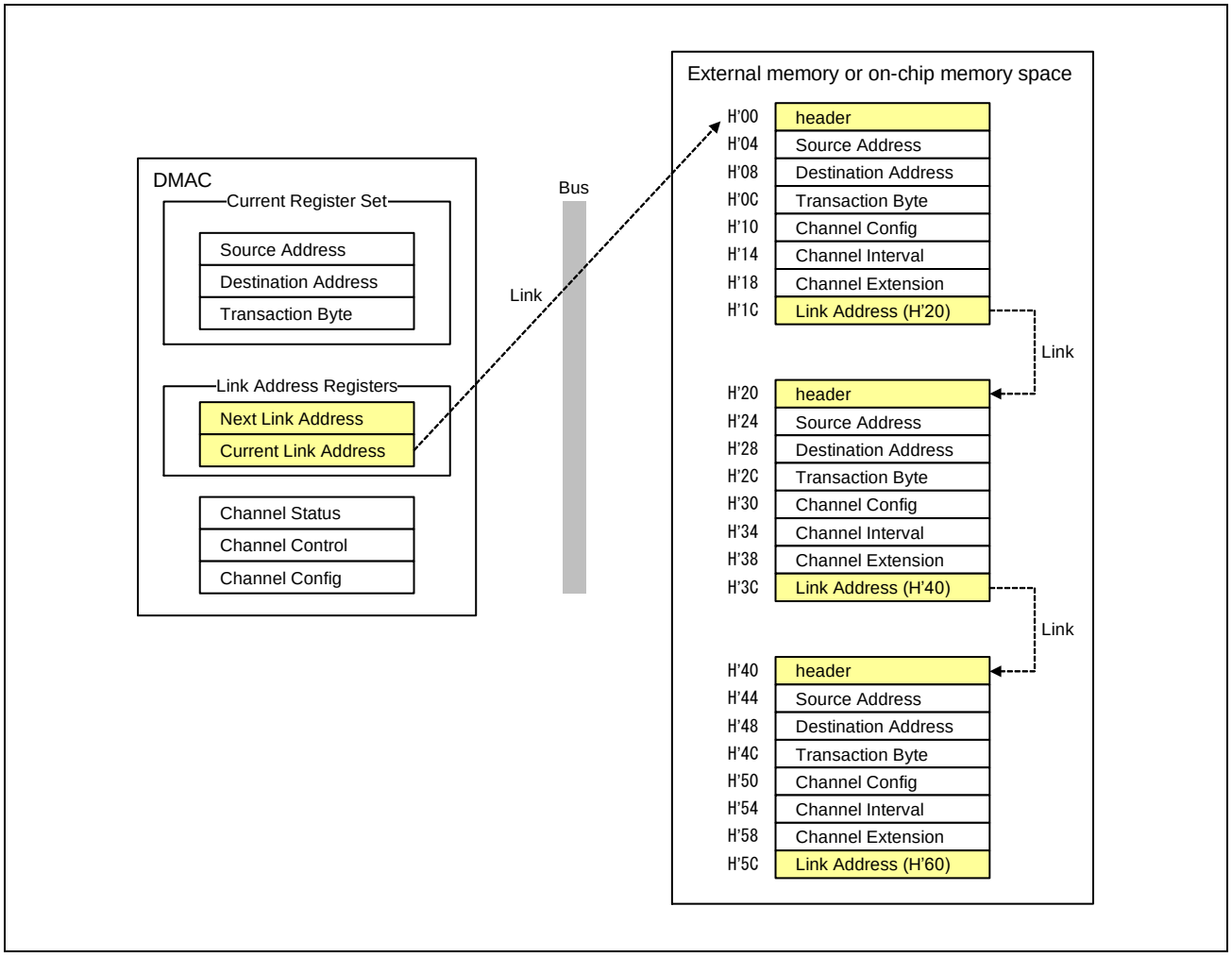



Figure 14.6 Link Mode Outline

### (1) Operation Flow

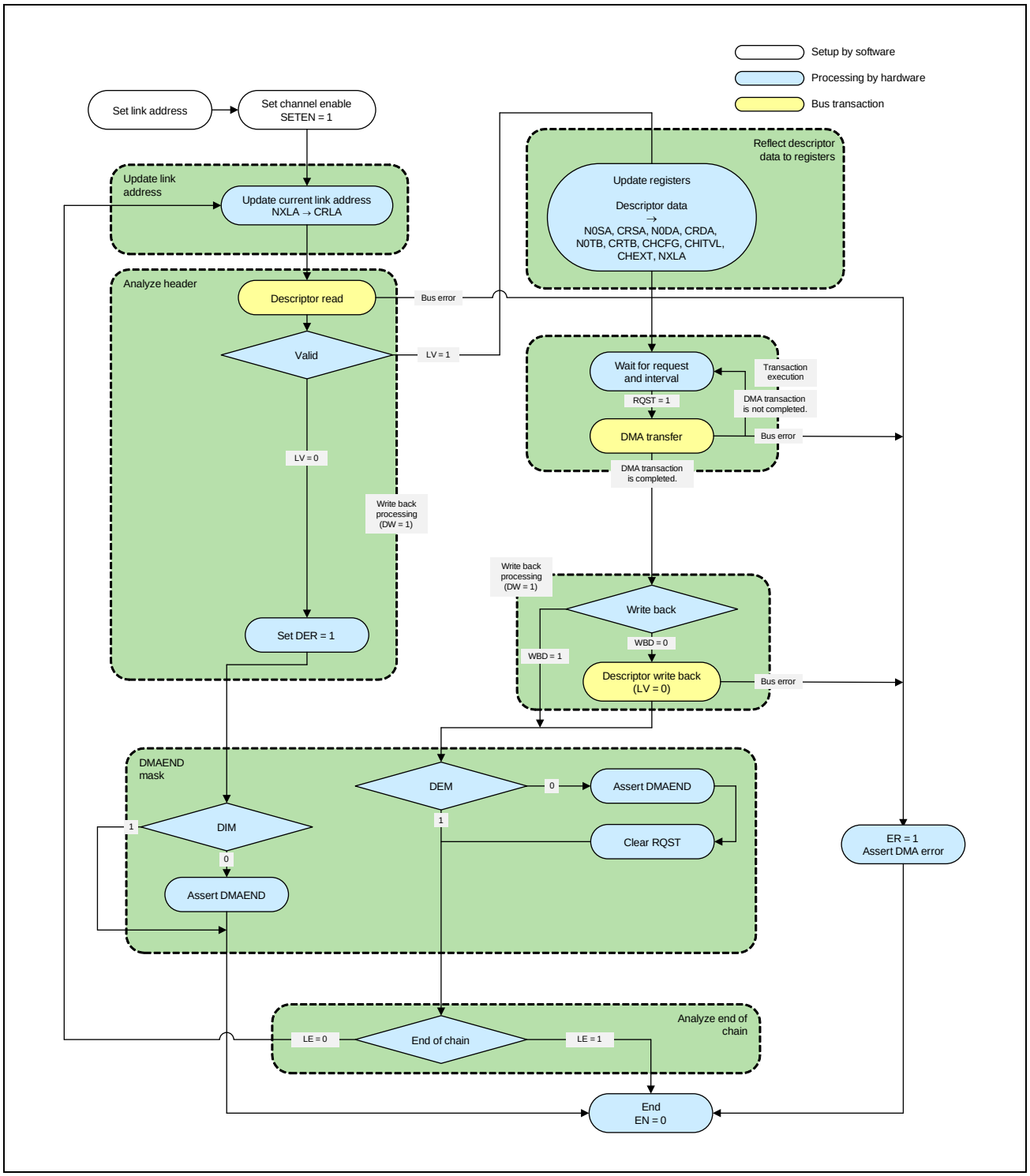



Figure 14.7 Link Mode Flow

<Explanation of the link mode flow>

1. Channel setting

   The start address of the link destination is set in NXLA_n.

2. Link address update

   When 1 is set in EN (1 is set in SETEN), the Link address set in NXLA_n is loaded to CRLA_n.

3. Descriptor load and header analysis

   The DMAC begins to load the descriptor and then analyzes the content of the header. When LV is 0, the DMAC discards the loaded descriptor and sets 1 in DER to end the operation (EN = 0). In this case, if 0 is set in DIM of the header, DMAEND is issued.

4. Descriptor setting

   The loaded descriptor is set in the Current register set and Channel register set. Also, the next link address is set in NXLA_n.

5. DMA transaction

   A DMA transaction is executed according to the set values.

6. Header writeback

   When 0 is set in WBD of the header, the DMAC writes back the header with 0 set in its LV bit.

7. DMAINT mask

   When 0 is set in the DEM bit of CHCFG_n, the DMA transfer end interrupt is issued.

8. Link end analysis

   When 1 is set in LE of the header, the operation is ended by clearing EN to 0, after transfer using the settings of the descriptor is completed. If the setting of LE is 0, the Current registers are then updated and loading of the next descriptor begins. The TEND signal is issued after the transfer of each descriptor.

### (2) Register Setting

#### (a) Link mode setting

To use the link mode, set 1 in the DMS bit of the CHCFG_n register.

Table 14.12 Link Mode Setting

| DMS (CHCFG_n) | Description |
|---|---|
| 1 | Operates in link mode.<br>This bit cannot be changed using a descriptor. |

#### (b) Link address setting

There are two registers that indicate a link address:
Next Link address register and Current Link address register.
To start the link mode, set a link address in the Next Link address register.

The Next Link address indicates the next link address after a descriptor is loaded. The Current Link address indicates the currently executed link address.

Table 14.13 Link Address Register Set

| Register | Description |
|---|---|
| Next Link Address Register (NXLA_n) | Sets and indicates the next link address. Before starting the link mode, set a link address in this register. |
| Current Link Address Register (CRLA_n) | Indicates the currently executed link address. This register is read-only. |

**CAUTION**

In link mode, the settings can be changed by reading a descriptor. It is not possible, however, to synchronize the change of the settings with a peripheral module request or external request. Therefore, when using a peripheral module request or external request, set AM, LVL, HIEN, LOEN, and SEL of the CHCFG_n register before setting Enable and do not change any of these bits in the descriptor.

### (3) Descriptor Setting

In a link address, prepare a descriptor with data arranged in the order shown below.

The DMAC reads the descriptor in burst mode.

#### (a) Descriptor data arrangement

Table 14.14 Descriptor Data Arrangement

| Address | Data | Remarks |
|---|---|---|
| Link address + H'00 | header | |
| Link address + H'04 | Source Address | |
| Link address + H'08 | Destination Address | |
| Link address + H'0C | Transaction Byte | |
| Link address + H'10 | Config | The register mode cannot be set. |
| Link address + H'14 | Interval | |
| Link address + H'18 | Extension | |
| Link address + H'1C | Next Link Address | |

**Note:** As a link address, set an address aligned along the 32-bit boundary.

#### (b) header

The header indicates the status of the descriptor, as shown below.

The DMAC reads this area when a DMA transfer is started in link mode. Also, after a DMA transaction is completed, the DMAC writes back the transfer status to the area.

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| — | — | — | — | — | — | — | — | — | — | — | — | DIM | WBD | LE | LV |

If 0 is set in WBD when the header is read, the DMAC writes back to this 4-byte area after a DMA transaction using a descriptor is completed.

0 is written back to LV, and the values obtained when reading the header are written back to the other fields.

Figure 14.8 Header Area

Table 14.15 Header Area

| Bit Position | Bit Name | Meaning |
|---|---|---|
| 31 to 4 | — | — |
| 3 | DIM | Descriptor Interrupt Mask<br>Sets whether to mask the DMA transfer end interrupt if 0 is set in LV when the header is loaded.<br>0: Issues a DMA transfer end interrupt.<br>1: Does not issue a DMA transfer end interrupt. |
| 2 | WBD | Write Back Disable<br>Sets whether to mask LV bit writeback. When 1 is set in this bit, the DMAC does not perform writeback.<br>0: Writes the LV bit back to 0.<br>1: Does not write back the LV bit. |
| 1 | LE | Link End<br>Indicates whether the link ends with the DMA transaction of this descriptor.<br>Set 1 in this bit to indicate the end of the link.<br>0: The link continues.<br>1: The link ends. |
| 0 | LV | Link Valid<br>Indicates whether this descriptor is valid.<br>If 0 is set in WBD, the DMAC writes 0 in this bit after the DMA transaction written in the descriptor is executed. When setting the header, set 1 in this bit.<br>0: Descriptor invalid<br>1: Descriptor valid |

### (c) Descriptor data other than the header

The data items of the descriptor other than the header are the same as defined in the internal register specifications (note that the DMS bit of the CHCFG_n register cannot be changed using the descriptor). For information about the internal register specifications, see **Section 14.4, Register Descriptions**.

For descriptor setting examples, see **Section 14.8, DMA Setting Examples**.

### (d) CACHE settings for descriptor access

The CACHE settings for descriptor access can be set in LWCA and LDCA of the DMA control register (DCTRL_0_7, DCTRL_8_15). Make these settings as appropriate for the access destination in which the descriptor is prepared.

### (e) Descriptor area and DMA transfer area

The following figure outlines the descriptor area and DMA transfer area that are accessed by the DMAC.

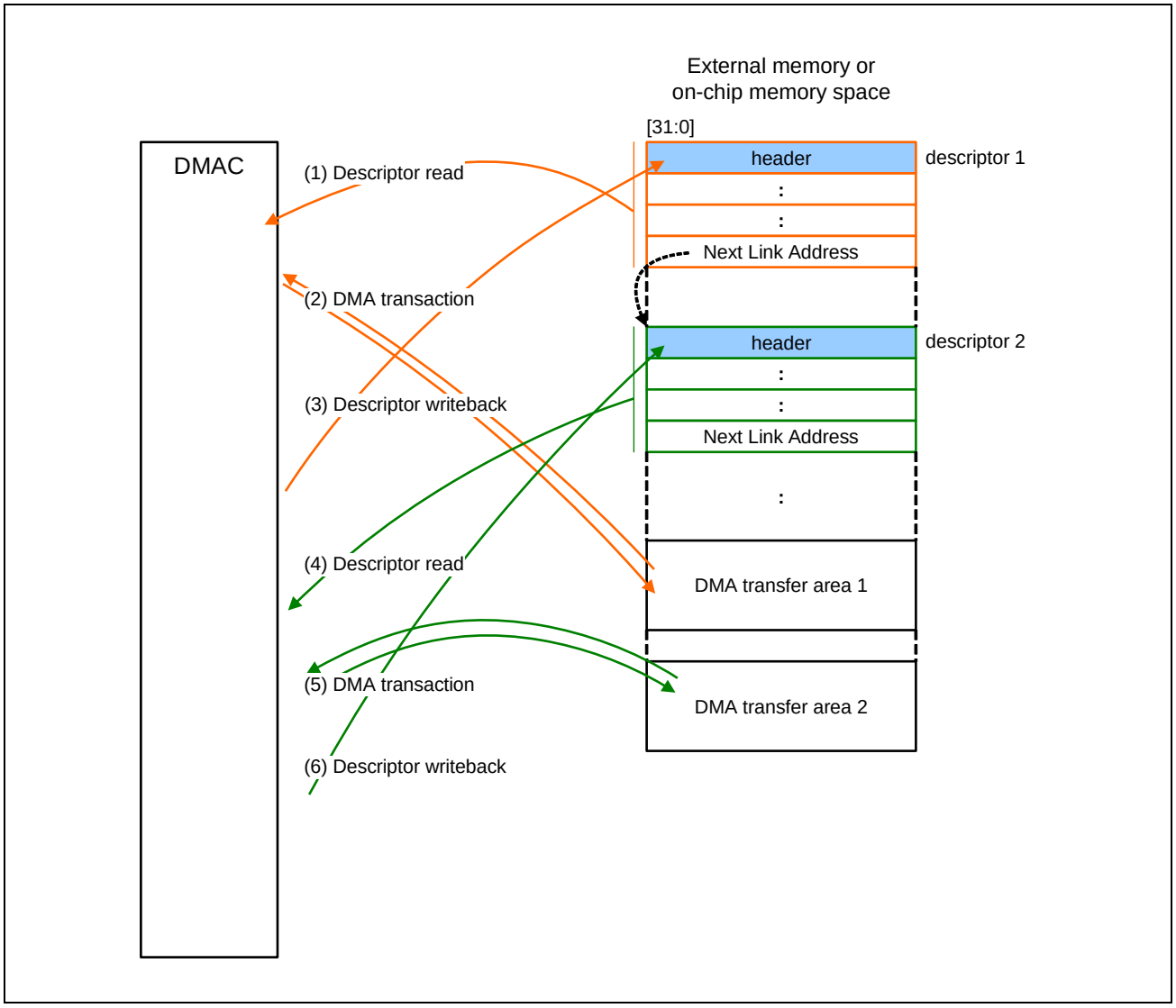



Figure 14.9 Outline of the Descriptor Area and DMA Transfer Area

1. Descriptor read

   The values set in the internal Next Link Address register are loaded to the Current Link Address register, and a descriptor is read from the external memory space (descriptor1) pointed to by the Current Link Address register.

2. DMA transfer

   When 1 is set in the LV bit of the header in the descriptor, a DMA transfer is executed according to the descriptor data.

3. Descriptor writeback

   When 0 is set in the WBD bit of the header after the DMA transfer of the set number of bytes is completed, the DMAC writes back data in word size to the header of descriptor1, with 0 set in LV and the other bits containing the values read in <1>.

4. Descriptor read

   When 0 is set in the LE bit of the header in the last read descriptor (<1>), the next descriptor is read from the address (descriptor2) indicated by Next Link Address in the descriptor.

5. DMA transfer

   When 1 is set in the LV bit of the header in the descriptor, a DMA transfer is executed according to the descriptor data.

6. Descriptor writeback

   When 0 is set in the WBD bit of the header after the DMA transfer of the set number of bytes is completed, the DMAC writes back data in word size to the header of descriptor2, with 0 set in LV and the other bits containing the values read in <4>.

4 through 6 are repeated.

When the header contains 1 in LE and 0 in WBD, the DMAC executes a DMA transfer using the settings of that descriptor, writes back data with 0 set in the LV bit of the header and ends the operation.

When the header contains 1 in both LE and WBD, the DMAC executes a DMA transfer using the settings of that descriptor and ends the operation (without writing back).

When the header contains 0 in LV, the DMAC ends the operation (without executing a DMA transfer).

### (4) Descriptor Configuration Examples

In link mode, a descriptor can be configured as shown below.

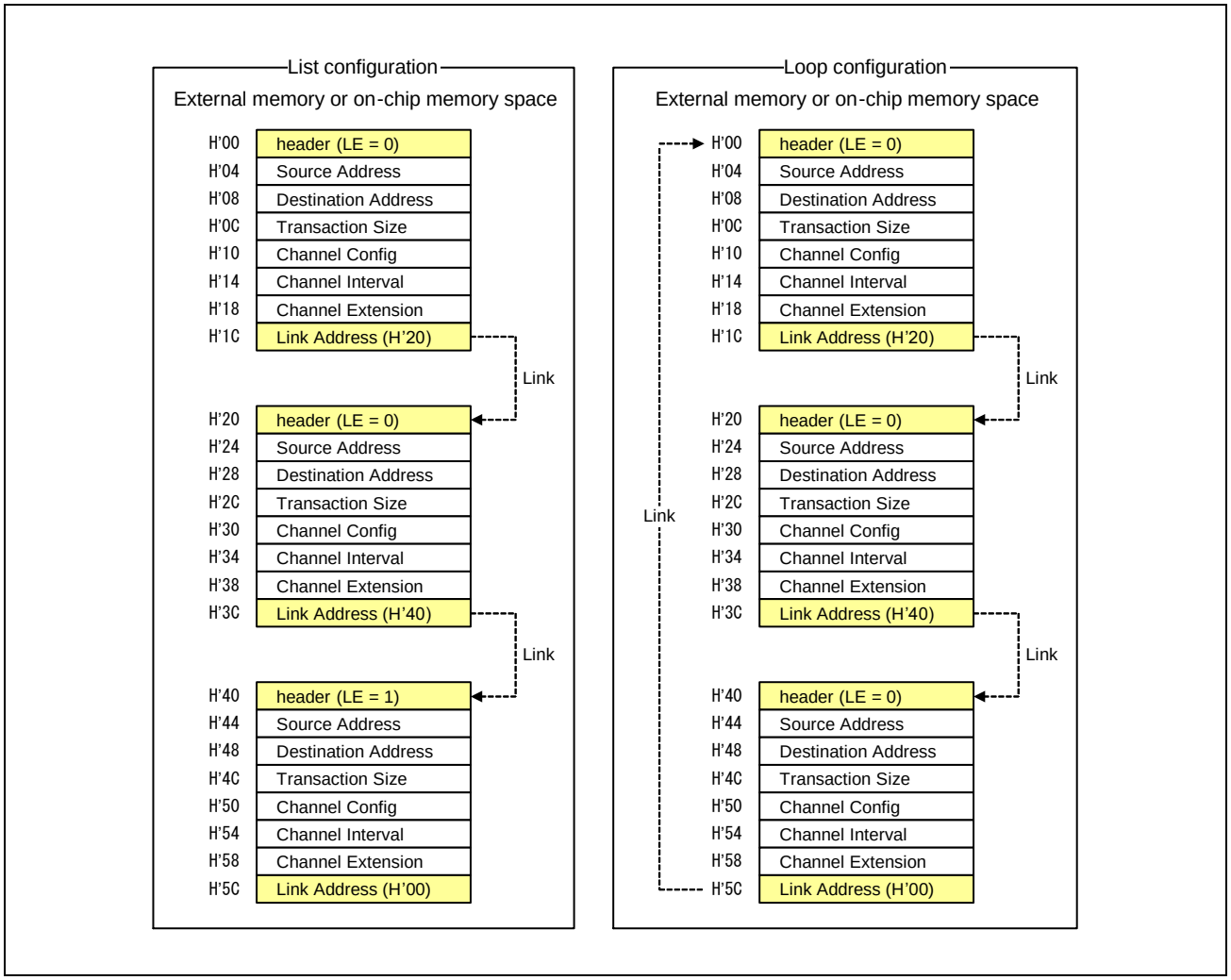



Figure 14.10 Descriptor Configuration Examples

- List configuration
  The link is ended by setting 1 in the LE bit of the header in the last descriptor.
- Loop configuration
  A descriptor can be created with a loop configuration, by setting the address of the top descriptor in the link address of the last descriptor. To end the loop, change the value of the LE bit of the header to 1 before the DMAC reads the descriptor, or follow the transfer suspension procedure.

## 14.7 DMA Transfer

The basic operation of DMA transfer is described here.

### 14.7.1 Transfer Mode

Two transfer modes are supported: single transfer mode and block transfer mode.

To select a transfer mode, set the TM bit of CHCFG_n for each channel.

Table 14.16 Basic Transfer Setting

| Transfer Mode | TM (CHCFG_n) | Function |
|---|---|---|
| Single transfer | 0 | A single DMA transfer is executed in response to a DMAREQ. |
| Block transfer | 1 | In response to a DMAREQ, the DMAC continues to execute the transfer until the DMA transaction is completed. |

#### (1) Single Transfer Mode

When a DMA transfer request is received, a DMA transfer is executed once in the direction indicated by REQD (source or destination). A DMA transfer is executed once each time a transfer request is received, and this operation continues until the number of bytes loaded to CRTB_n is reached (arbitration between channels is accomplished for each DMA transfer).

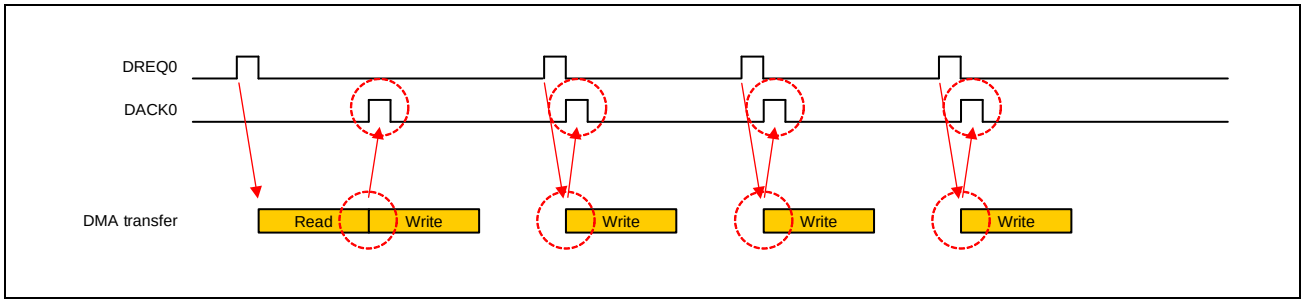



Figure 14.11 Single Transfer Mode (REQD = 1, SDS > DDS)

#### (2) Block Transfer Mode

Once a DMA transfer request is received, the DMAC continues to execute the transfer until data equivalent to the number of bytes loaded to the DMA transfer byte register (CRTB_n register) is transferred (the DMA transaction is completed) (arbitration between channels is accomplished for each DMA transfer).

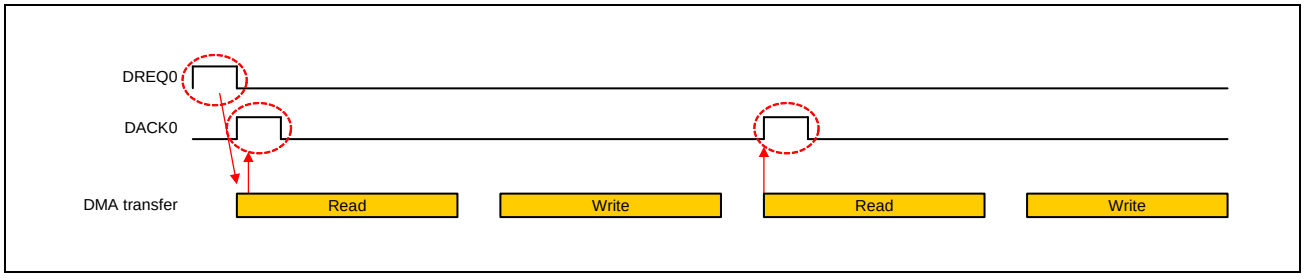



Figure 14.12 Block Transfer Mode (REQD = 0, SDS < DDS)

### 14.7.2 Priority Control for DMA Channels

Within channels 0 to 7 and 8 to 15, two priority control modes are supported: fixed priority mode and round robin mode. Only round robin mode is supported for priority control between the group of channels 0 to 7 and the group of channels 8 to 15. To select a priority control mode, use the PR bit of the DMA control register (DCTRL register). The fixed priority mode is selected when 0 is set in the PR bit, and the round robin mode is selected when 1 is set.

Read priority and write priority are controlled independently.

The DMAC issues transfer requests to different channels concurrently without waiting for the completion of any particular transfer and processes responses in the order it receives them. Therefore, the order in which the channels start transactions is not necessarily consistent with the order in which the transactions end.

Table 14.17 Priority Control Setting

| Mode | PR (DCTRL) | Function | Purpose |
|---|---|---|---|
| Fixed priority | 0 | Requests are controlled based on the fixed order of priority for channels 0 to 7 and 8 to 15 (High: CH0 (CH8) > CH1 (CH9) > CH2 (CH10) > CH3 (CH11) > CH4 (CH12) > CH5 (CH13) > CH6 (CH14) > CH7 (CH15): Low). | Use this mode when the channels have a specific order of priority. |
| Round robin | 1 | Requests are controlled in a round robin fashion. | Use this mode to execute all requests evenly. |

#### (1) Fixed Priority Mode

In fixed priority mode, the channels have a fixed order of priority in channels 0 to 7 and 8 to 15. Round robin mode is used to determine the priority between the group of channels 0 to 7 and the group of channels 8 to 15.

Immediately after a reset, the order of priority is as follows.

High CH0 > CH8 > CH1 > CH9 > CH2 > CH10 > CH3 > CH11 > CH4 > CH12 > CH5 > CH13 > CH6 > CH14 > CH7 > CH15 Low

If there is a transfer request from DMA channel 0 in this state, a transfer is executed on DMA channel 0. After the transfer is completed, the order of priority is as follows.

High CH8 > CH0 > CH9 > CH1 > CH10 > CH2 > CH11 > CH3 > CH12 > CH4 > CH13 > CH5 > CH14 > CH6 > CH15 > CH7 Low

If a DMA transfer request occurs on multiple channels simultaneously, the DMA transfer request of the channel having the smallest channel number is given priority. The following figure shows an example where a DMA transfer request occurs on a channel having a higher priority while a DMA transfer is being executed in fixed priority mode.

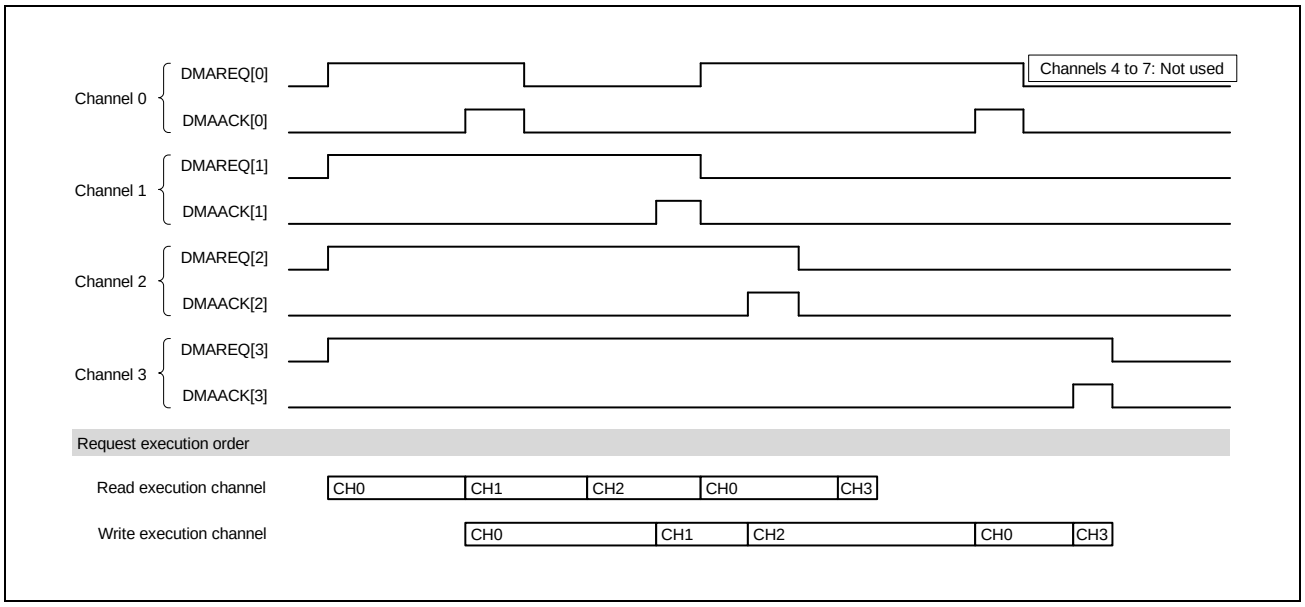



Figure 14.13 Fixed Priority Mode (Number of Channels = 4, REQD = 1)

### (2) Round Robin Mode

In round robin mode, each time a transfer request is received from a channel in the group of channels 0 to 7 and the group of channels 8 to 15, the order of priority is changed in such a way that the channel that executed a transfer last has the lowest priority.
Round robin mode is used to determine the priority between the group of channels 0 to 7 and the group of channels 8 to 15.

Immediately after a reset, the order of priority is the same as that of the fixed priority mode, which is as follows.
High CH0 > CH8 > CH1 > CH9 > CH2 > CH10 > CH3 > CH11 > CH4 > CH12 > CH5 > CH13 > CH6 > CH14 > CH7 > CH15 Low

If a transfer request is received from DMA channel 2 in this state, a transfer is executed on DMA channel 2. After the transfer is completed, the order of priority is as follows.
High CH8 > CH3 > CH9 > CH4 > CH10 > CH5 > CH11 > CH6 > CH12 > CH7 > CH13 > CH0 > CH14 > CH1 > CH15 > CH2 Low

The following figure shows an example where DMA transfers are executed in round robin mode.

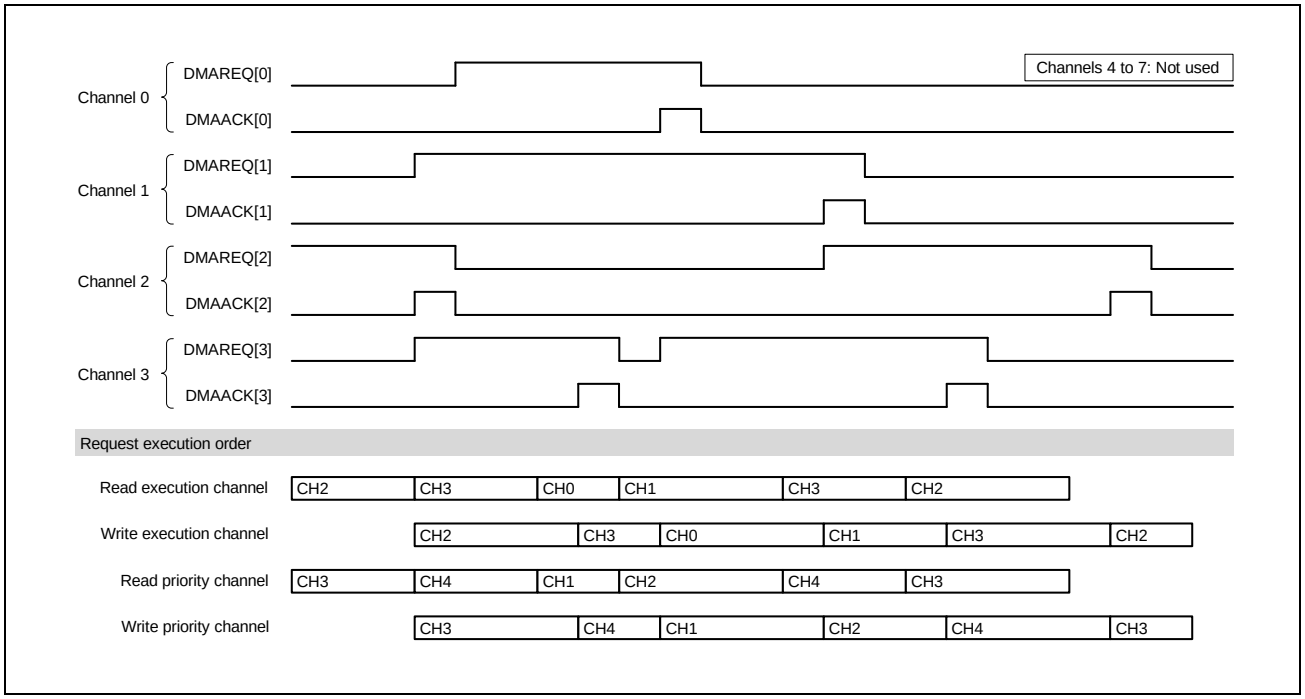



Figure 14.14 Round Robin Mode (Number of Channels = 4, REQD = 0)

The channel whose channel number is the number of the currently transferring channel + 1 gets to execute a DMA transfer next. If there is no transfer request from this channel, the channel whose channel number is the number of this channel + 1 gets to execute a DMA transfer.

### 14.7.3 DMA Transfer Request

Edge detection or level detection can be selected using the LVL bit of the CHCFG_n register.

The HIEN and LOEN bits of the CHCFG_n register are used to select either the rising edge or falling edge in the case of edge detection or either the high level or low level in the case of level detection.

When the transfer request is by an on-chip peripheral module, set the CHCFG_n register according to **Table 14.3**.

#### (1) Edge Detection

Setting 0 in the LVL bit of the CHCFG_n register enables edge detection.

#### (2) Level Detection

Setting 1 in the LVL bit of the CHCFG_n register enables level detection.

### 14.7.4 DMA Transfer End Interrupt

The DMA transfer end interrupt is an interrupt request signal that indicates that a DMA transaction is completed. There is an independent DMA transfer end interrupt for each channel.

When the transfer of data equivalent to the total transfer byte count loaded to the CRTB (Current Transaction Byte) is completed, 1 is set in END of the CHSTAT_n register. In this case, when 0 is set in DEM of the CHCFG_n register, the DMA transfer end interrupt (DMAINTn_NS) is output (n = 0 to 15). (When writeback is performed in link mode, the signal is output after the writeback operation.)

When 0 is set in LV of the header in the read descriptor in link mode, 1 is set in DER of the CHSTAT_n register. In this case, when 0 is set in DIM of the header, the DMA transfer end interrupt is output.

Table 14.18 Assertion Conditions of DMA Transfer End Interrupt

| Source | Condition | DMA Transfer End Interrupt Mask Signal |
|---|---|---|
| DMA transaction end | When the transfer of data equivalent to the total transfer byte count loaded to the CRTB (Current Transaction Byte) is completed with an OKAY response (or after the writeback operation when writeback is performed in link mode) | DEM bit of the CHCFG_n register |
| Descriptor invalid | When 0 is set in LV of the header in the read descriptor in link mode while 0 is set in DIM of the header | DIM bit of the header |

### 14.7.5 DMA Error Interrupt

If an error response is received for a DMA transfer or descriptor access, the DMAC regards it as an error and stops the transfer. Upon receiving an error response, the EN bit of the CHSTAT_n register of transferring channel n is cleared to 0 and 1 is set in the ER bit (n = 0 to 15). Also, the DMA error interrupt (DMAERR_NS) is output.

The DMA error interrupt cannot be masked.

Once an error occurs, the data of the whole transfer cannot be guaranteed. Be sure to start the transaction again from the beginning by following the procedure below.

1. Set 1 in the SWRST bit of the CHCTRL_n register.
2. Set each register again.

### 14.7.6 Interval Count Function

The interval at which a DMA transfer is executed can be adjusted by setting the ITVL bit of the channel interval register (CHITVL_n). This function is intended to prevent the DMA controller from occupying the bus all the time.

When a read or write operation is completed, a countdown starts from the value set in CHITVL_n. The next internal request is not executed until the count value reaches 0.

The following figure shows an example of how this works.

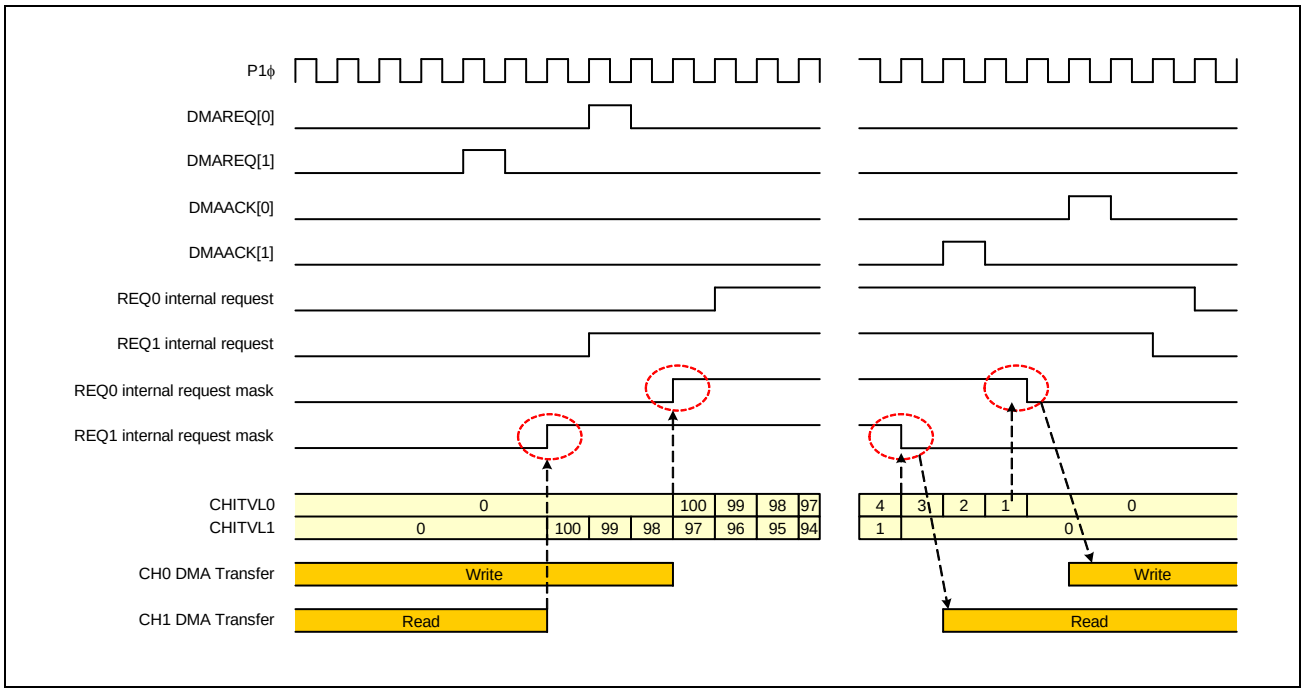



Figure 14.15 Interval Count

### 14.7.7 Difference in Operation Due to the Transfer Size

#### (1) When the Source Transfer Size Is Smaller

When the read of data equivalent to the destination data size is completed, the data is written to the destination.

The following figure shows a timing chart where the source transfer size is 8 bits and the destination transfer size is 32 bits (in the case of rising edge detection).

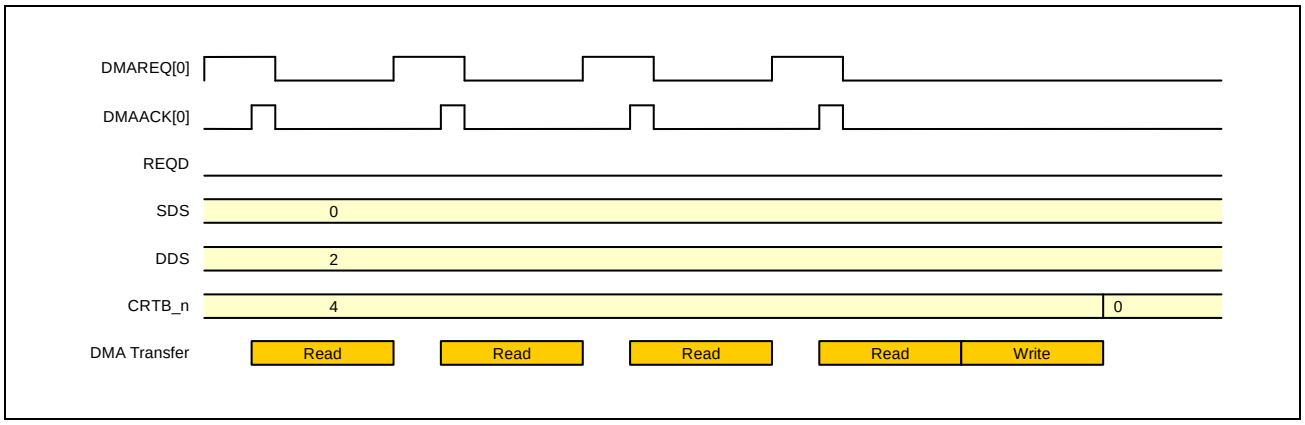



Figure 14.16 When the Source Transfer Size Is Smaller (LVL = 0, HIEN = 1, REQD = 0, SDS < DDS in CHCFG_n)

#### (2) When the Destination Transfer Size Is Smaller

Since the source transfer size is larger, multiple destination writes occur after a single source read. The following figure shows a timing chart where the source transfer size is 64 bits and the destination transfer size is 16 bits (in the case of rising edge detection) (1 is set in REQD of the CHCFG_n register).

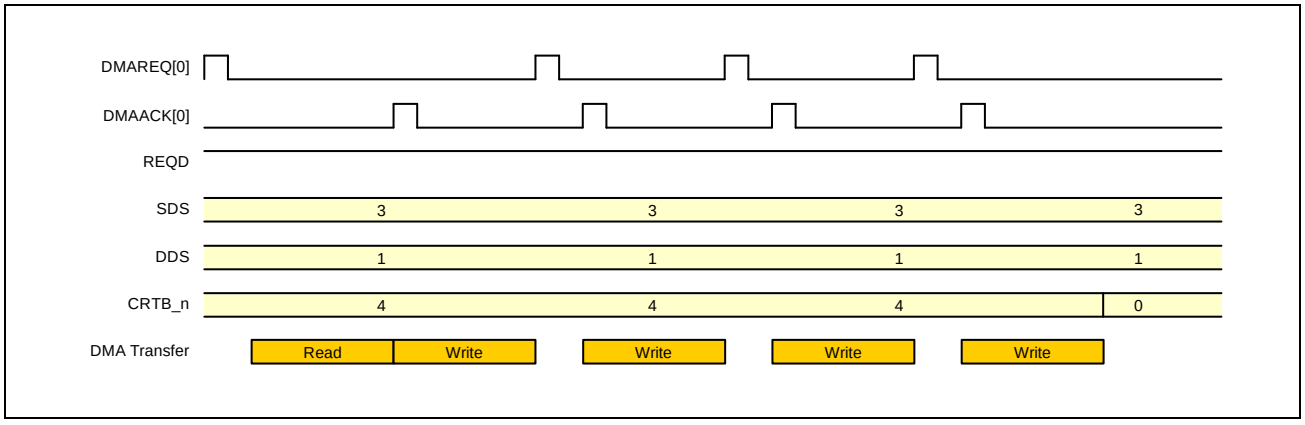



Figure 14.17 When the Destination Transfer Size Is Smaller (LVL = 0, HIEN = 1, REQD = 1, SDS > DDS in CHCFG_n)

### (3) When the Source Transfer Size Is the Same as the Destination Transfer Size

Every time a DMA transfer request is detected, a source read and a destination write occur.

The following figure shows a timing chart where the source transfer size and the destination transfer size are both 8 bits (in the case of rising edge detection, with 1 set in REQD of the CHCFG_n register).

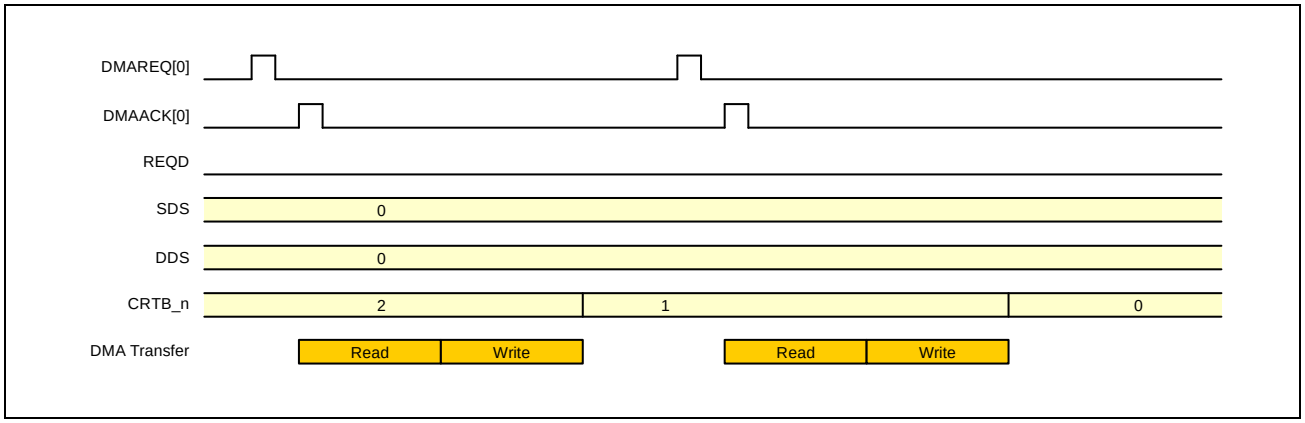



Figure 14.18 When the Source Transfer Size Is the Same as the Destination Transfer Size (LVL = 0, HIEN = 1, REQD = 0, SDS = DDS in CHCFG_n)

## 14.7.8 Transfer Status

The channel status register indicates the status of DMA transfer execution on a channel.

### (1) Suspend

A DMA transfer can be suspended by using the SETSUS bit of CHCTRL_n. In this case, if an ongoing bus cycle exists, the DMAC waits for that cycle to end before suspending the transfer. Writing 1 in the CLRSUS bit restores the DMA transfer from the suspend status.

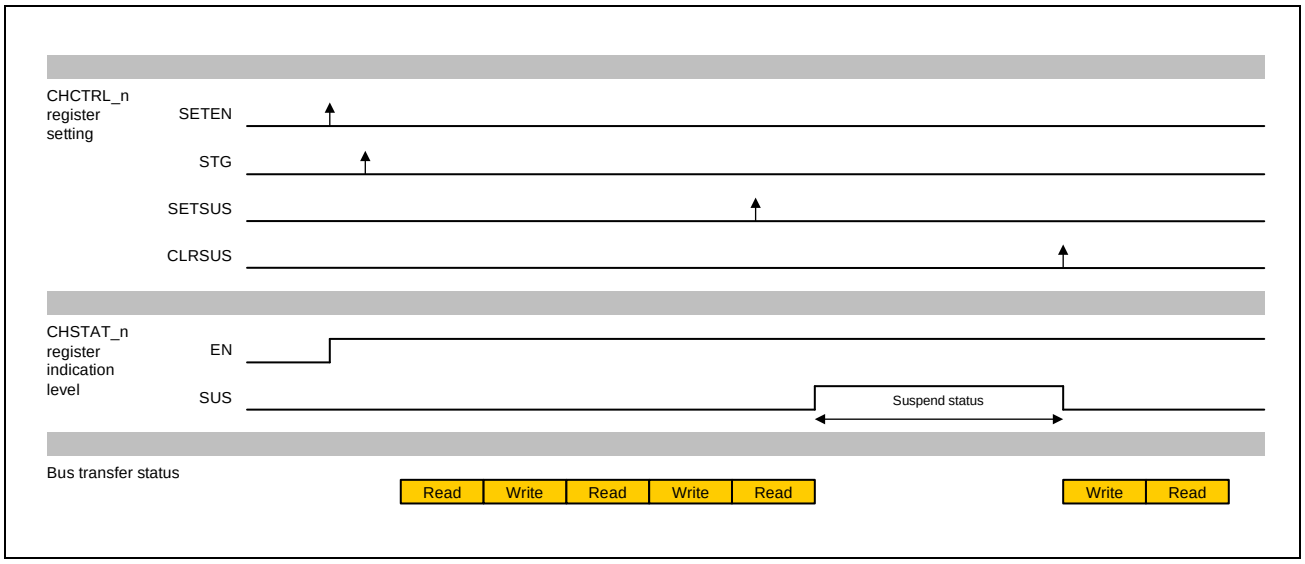



Figure 14.19 DMAC Suspend Status (Auto Request/Block Transfer)

In the above case, the DMA transfer is suspended after the read transfer is completed.

If there is any ongoing DMA transfer, the suspend status starts when that transfer is completed. To make sure that the transfer is suspended, read the CHSTAT or DSTAT_SUS register, after setting SETSUS, and check that 1 is set in the SUS bit for the relevant channel.

### (2) Transfer Stop

If 1 is written to CLREN while a DMA transaction is in progress, the DMA transaction for the corresponding channel can be stopped. For the post-stop processing, two modes are supported: one sweeps out the data remaining in the buffer when the transaction is stopped (SBE = 1) and the other does not (SBE = 0). One of these modes can be selected using the SBE bit of the CHCFG_n register. By default, SBE is set to 0.

When this sweep mode is enabled and CLREN is set to 1, and if a DMA transaction is stopped with data remaining in the DMAC buffer, the transaction is completed after the DMAC sweeps the data.

### (a) Transfer Stop (Buffer Sweep Disabled - SBE = 0)

If 1 is set in CLREN during a DMA transfer, the DMA transfer is stopped. The stop timing depends on the value set in REQD. After stopping a DMA transfer, be sure to set 1 in SWRST to clear the DMA internal status before setting the next transfer.

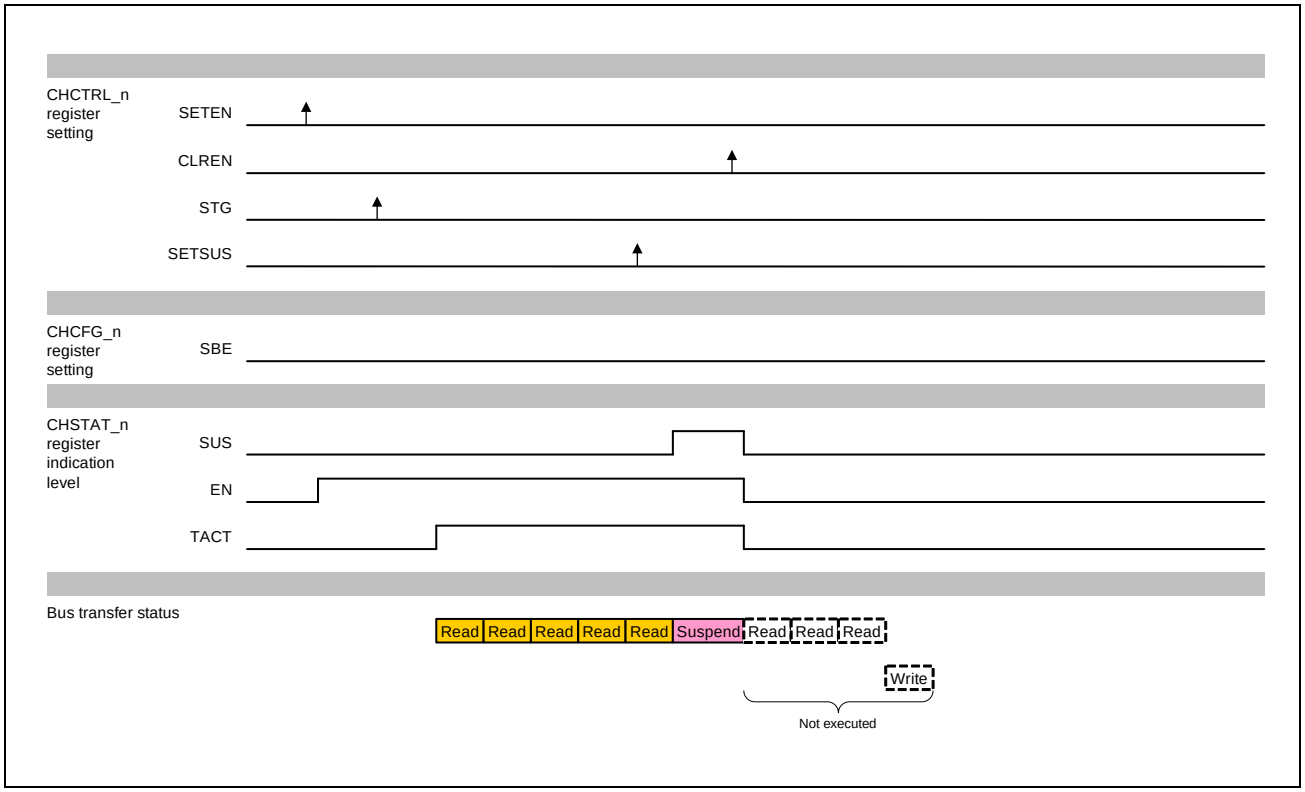



Figure 14.20 DMAC Transfer Stop

- The setting of the TACT bit being 0 indicates that the channel has been brought to a complete stop.
- If an ongoing DMA transfer is stopped before it is completed, the DMA transfer end interrupt is not asserted.
- If 0 is set in REQD, the DMA transfer is stopped when the next read is completed. (If the buffer contains any data that can be written, the DMA transfer is stopped after the data is written.)
- If 1 is set in REQD, the DMA transfer is stopped when the next write is completed.

### (b) Transfer Stop (Buffer Sweep Enabled - SBE = 1)

If 1 is set in CLREN during a DMA transfer, the DMA transfer is stopped. When 0 is set in REQD, the DMA transfer is stopped after the DMAC sweeps (writes) the already read data. If 1 is set in REQD to use hardware requests, do not use the sweep mode. After stopping a DMA transfer, be sure to set 1 in SWRST to clear the DMAC internal status before setting the next transfer.

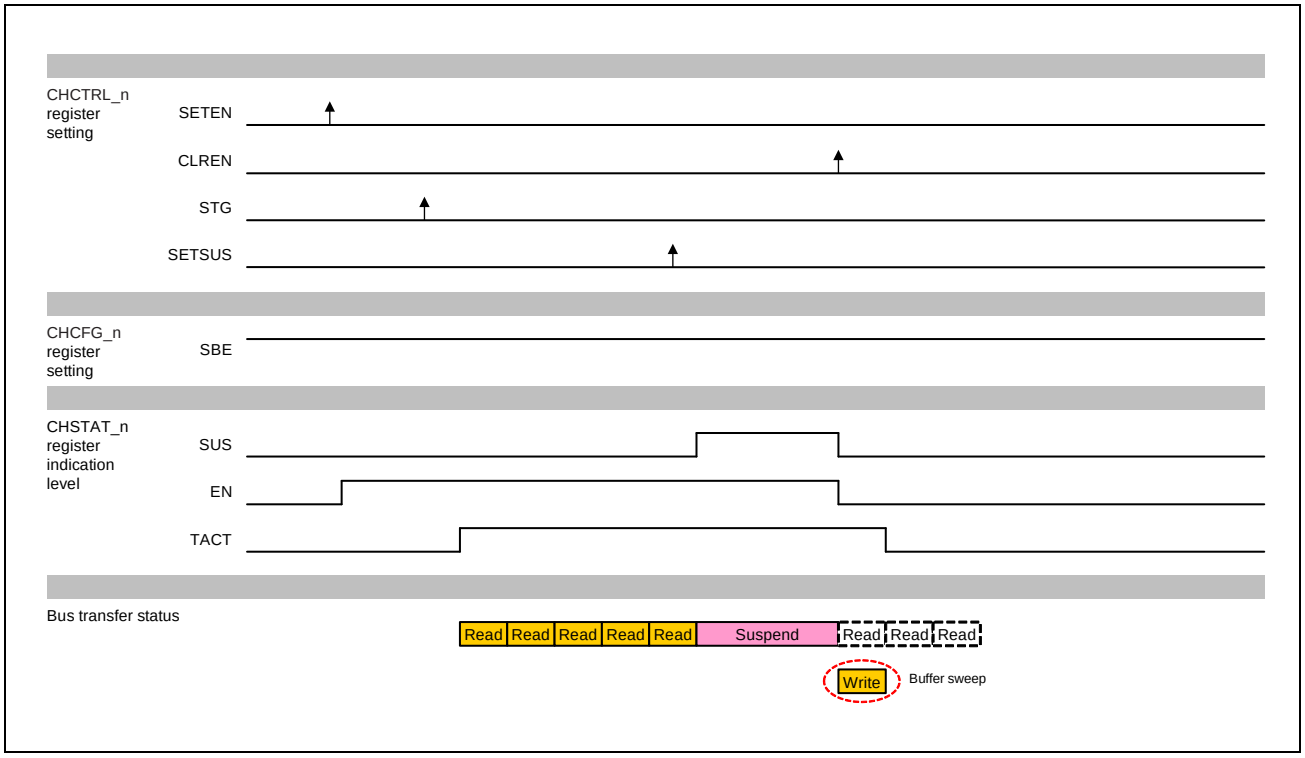



Figure 14.21 DMA Transfer Stop (Buffer Sweep Mode)

- The setting of the TACT bit being 0 indicates that the channel has been brought to a complete stop.
- If a transfer is stopped in sweep mode (SBE = 1) during the fifth read transfer by setting SETSUS and then CLREN, the read data is written before the DMA transfer is stopped.

### (c) Channel Stop Check Method

Even when the EN bit is cleared to 0, the DMA transfer cannot be stopped immediately, if the bus is already executing the transfer. Therefore, in order to make sure that the DMAC has been brought to a complete stop, check that the EN bit and TACT bit are both set to 0.

### (d) Transfer Stop Procedure

The transfer stop procedure is described below.

1. Set 1 in SETSUS of CHCTRL_n.
2. Repeat polling until the SUS bit of CHSTAT_n is set to 1. (If EN is already set to 0, the DMAC has already been stopped. Go to step 6.)
3. Set 1 in CLREN of CHCTRL_n.

4. When 0 is set in SBE, the transfer is stopped according to the value of REQD. When 1 is set in SBE, the sweep mode is enabled. When 1 is set in SBE, set 0 in REQD.
5. Read CHSTAT_n to check that 0 is set in the TACT bit. When TACT is set to 0, it means that the DMAC has been brought to a complete stop. When TACT is set to 1, repeat polling until this bit is set to 0.
6. To execute the next DMA transfer after stopping a transfer, be sure to set 1 in the SWRST (software reset) bit of CHCTRL_n before the next DMA transfer starts.

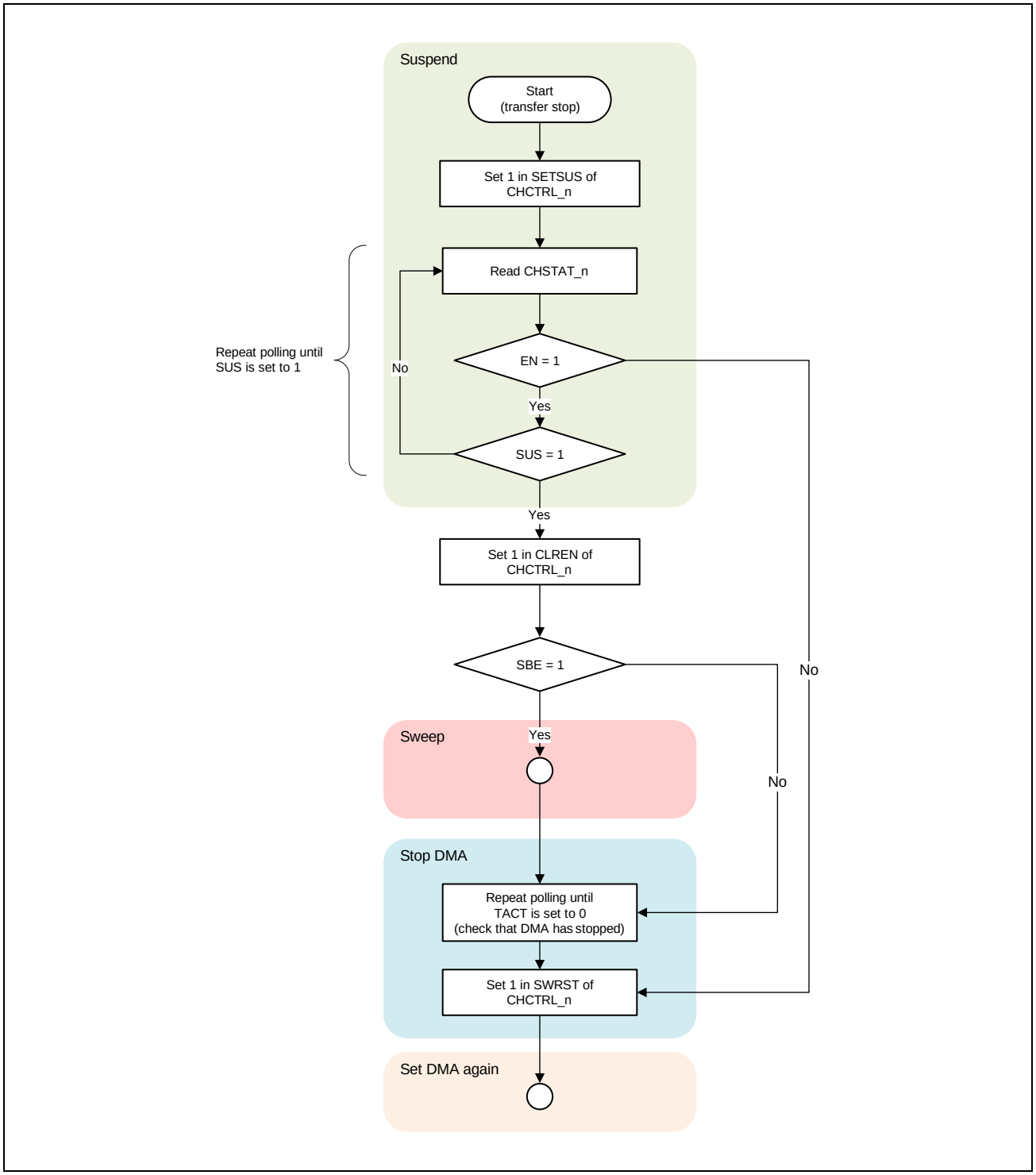



Figure 14.22 Transfer Stop Flow

## 14.8 DMA Setting Examples

Setting examples applicable when DMA transfer is executed using the direct memory access controller are shown in the following.

The transfer conditions for these setting examples are as follows.

Table 14.19 Transfer Condition List for DMA Transfer Setting Examples

| | DMA Mode | Transfer Mode | Transfer Request |
|---|---|---|---|
| Setting example 1 | Register | Single | Hardware |
| Setting example 2 | Register | Block | Software |
| Setting example 3 | Register (continuous execution) | Block | Software |
| Setting example 4 | Link | Block | Software |

For details of the settings, see the individual setting examples.

### 14.8.1 Setting Example 1 (Register Mode/Hardware Request)

The following table shows a setting example applicable when DMA transfer is executed using the settings shown below.

Table 14.20 DMA Transfer Setting Example 1

| Item | | Description | |
|---|---|---|---|
| Channel used | | 3 | |
| DMA mode | | Register | |
| Transfer mode | | Single transfer | |
| Register set used | | Next0 | |
| Source/destination | | Source | Destination |
| | Start address | H’11110000 | H’22220000 |
| | Address direction | Increment | Increment |
| | Data size | 32 bits | 32 bits |
| DMA transfer byte count | | 64 bytes | |
| DMA transfer request | | Rising edge detection by hardware | |
| DMAACK signal | | Level output during read | |
| DMA transfer end interrupt mask | | Not masked | |
| CACHE setting | | Default value | |

#### Setting example 1

N0SA = H’11110000 (source address)
N0DA = H’22220000 (destination address)
N0TB = H’00000040 (transfer byte count)

CHCFG = H’00022123 (configuration)
CHITVL = H’00000000 (interval)
CHEXT = H’00000000 (CACHE setting)

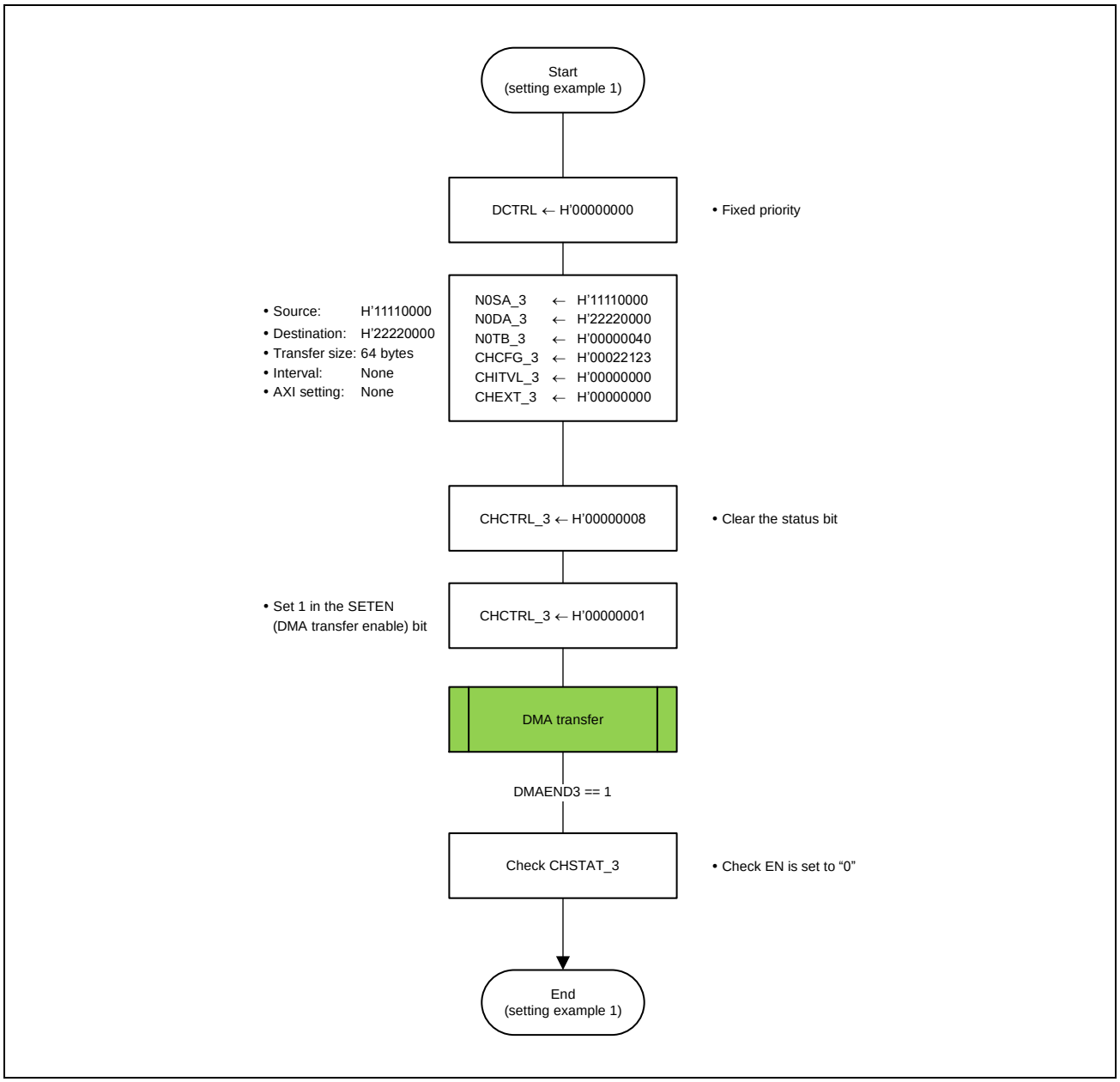



Figure 14.23 Setting Example 1

### 14.8.2 Setting Example 2 (Register Mode/Software Request)

The following table shows a setting example applicable when DMA transfer is executed using the settings shown below.

Table 14.21 DMA Transfer Setting Example 2

| Item | | Description | |
|---|---|---|---|
| Channel used | | 2 | |
| Priority control | | Round robin | |
| DMA mode | | Register | |
| Transfer mode | | Block transfer | |
| Register set used | | Next1 | |
| Source/destination | | Source | Destination |
| | Start address | H'0FFFE000 | H'33330000 |
| | Address direction | Increment | Increment |
| | Data size | 8 bits | 256 bits |
| DMA transfer byte count | | 128 bytes | |
| DMA transfer request | | Auto request | |
| DMAACK signal | | Masked | |
| DMA transfer end interrupt mask | | Not masked | |
| CACHE setting | | Default value | |

**Setting example 2**

DCTRL = H'00000001 (DMA setting)

N1SA = H'0FFFE000 (source address)
N1DA = H'33330000 (destination address)
N1TB = H'00000080 (transfer byte count)

CHCFG = H'10450402 (configuration)
CHITVL = H'00000000 (interval)
CHEXT = H'00000000 (CACHE setting)

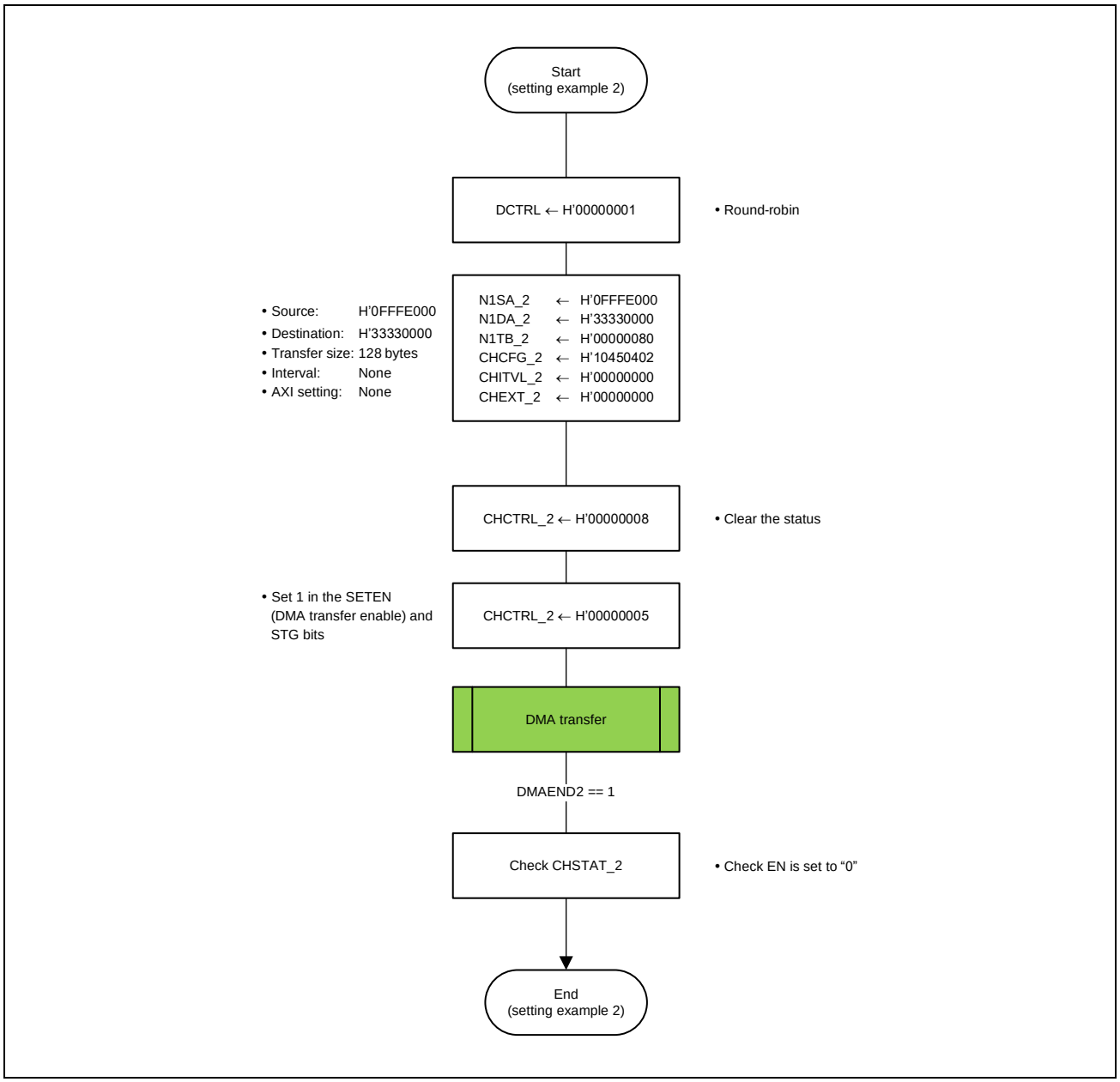



Figure 14.24 Setting Example 2

### 14.8.3 Setting Example 3 (Register Mode/Continuous Execution)

The following table shows a setting example applicable when DMA transfer is executed using the settings shown below.

Table 14.22 DMA Transfer Setting Example 3

| Item | | Description | |
|---|---|---|---|
| Channel used | | 1 | |
| Priority control | | Round robin | |
| DMA mode | | Register | |
| Transfer mode | | Block transfer | |
| Register set used | | Use Next0 and then Next1 continuously | |
| Next0 | | Source | Destination |
| | Start address | H’11110000 | H’33330000 |
| | Address direction | Fixed | Fixed |
| | Data size | 32 bits | 512 bits |
| | DMA transfer byte count | 512 bytes | |
| Next1 | | Source | Destination |
| | Start address | H’22220000 | H’44440000 |
| | Address direction | Fixed | Fixed |
| | Data size | 32 bits | 512 bits |
| | DMA transfer byte count | 2048 bytes | |
| DMA transfer request | | Auto request | |
| DMAACK signal | | Not output | |
| DMA transfer end interrupt mask | | Mask the DMA transfer end interrupt upon completion of Next0 | |
| CACHE setting | | Default value | |

**Setting example 3**

DCTRL = H’00000001 (DMA setting)

N0SA = H’11110000 (source address)
N0DA = H’33330000 (destination address)
N0TB = H’00000200 (transfer byte count)

N1SA = H’22220000 (source address)
N1DA = H’44440000 (destination address)
N1TB = H’00000800 (transfer byte count)

CHCFG = H’61762001 (configuration)
CHITVL = H’00000000 (interval)
CHEXT = H’00000000 (CACHE setting)

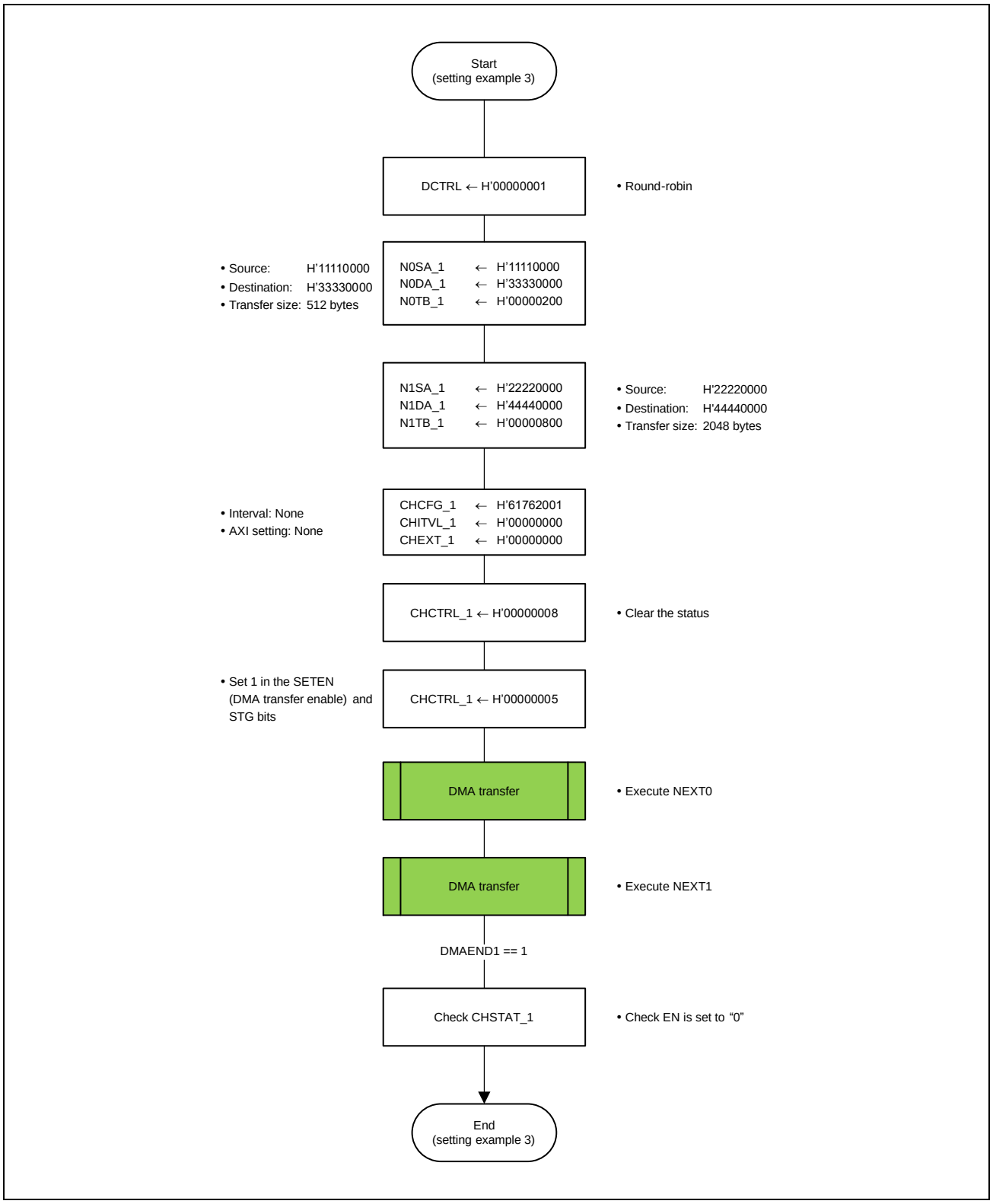



Figure 14.25 Setting Example 3

### 14.8.4 Setting Example 4 (Link Mode)

The following table shows a setting example applicable when DMA transfer is executed using the settings shown below.

Table 14.23 DMA Transfer Setting Example 4

| Item | Description |
|---|---|
| Channel used | 0 |
| Priority control | Round robin |
| DMA mode | Link |
| Transfer mode | Block transfer |
| Register set used | — |
| Descriptor start address | H'0000 1000 |

Table 14.24 DMA Transfer Setting Example 4 (Descriptor 1)

| Item | | Description | |
|---|---|---|---|
| Descriptor start address | | H’0000 1000 | |
| Next descriptor start address | | H’0000 2000 | |
| Transfer mode | | Block transfer | |
| Next0 | | Source | Destination |
| | Start address | H’1111 0000 | H’3333 0000 |
| | Address direction | Increment | Increment |
| | Data size | 32 bits | 32 bits |
| | DMA transfer byte count | 2048 bytes | |
| DMA transfer request | | Auto request trigger (STG) | |
| DMAACK signal | | Not output | |
| DMA transfer end interrupt mask | | Masked | |
| CACHE setting | | Default value | |
| header | | | |
| | DMA interrupt when LV = 1 | Issued (DIM = 0) | |
| | LV writeback | Done (WBD = 0) | |
| | Next link address | Available (LE = 0) | |
| | Descriptor valid | Valid (LV = 1) | |

Table 14.25 DMA Transfer Setting Example 4 (Descriptor 2)

| Item | | Description | |
|---|---|---|---|
| Descriptor start address | | H’0000 2000 | |
| Next descriptor start address | | H’0000 5000 | |
| Transfer mode | | Block transfer | |
| Next0 | | Source | Destination |
| | Start address | H’4444 0000 | H’5555 0000 |
| | Address direction | Increment | Increment |
| | Data size | 64 bits | 256 bits |
| | DMA transfer byte count | 1024 bytes | |
| DMA transfer request | | Auto request trigger (STG) | |
| DMAACK signal | | Not output | |
| DMA transfer end interrupt mask | | Masked | |
| CACHE setting | | Default value | |
| header | | | |
| | DMA interrupt when LV = 1 | Issued (DIM = 0) | |
| | LV writeback | Done (WBD = 0) | |
| | Next link address | Available (LE = 0) | |
| | Descriptor valid | Valid (LV = 1) | |

Table 14.26 DMA Transfer Setting Example 4 (Descriptor 3)

| Item | | Description | |
|---|---|---|---|
| Descriptor start address | | H’0000 5000 | |
| Next descriptor start address | | — | |
| Transfer mode | | Block transfer | |
| Next0 | | Source | Destination |
| | Start address | H’7777 0000 | H’AAAA 0000 |
| | Address direction | Increment | Increment |
| | Data size | 512 bits | 512 bits |
| | DMA transfer byte count | 4096 bytes | |
| DMA transfer request | | Auto request trigger (STG) | |
| DMAACK signal | | Not output | |
| DMA transfer end interrupt mask | | Not masked | |
| CACHE setting | | Default value | |
| header | | | |
| | DMA interrupt when LV = 1 | Issued (DIM = 0) | |
| | LV writeback | Done (WBD = 0) | |
| | Next link address | Not available (LE = 1) | |
| | Descriptor valid | Valid (LV = 1) | |

**Setting example 4**

DCTRL= H'0000 0001 (DMA setting)
NXLA = H'0000 1000 (descriptor start address)
CHCFG = H'8000 0000 (configuration)

Table 14.27 Descriptor Setting

| | Descriptor 1 | Descriptor 2 | Descriptor 3 |
|---|---|---|---|
| header | H'0000 0001 | H'0000 0001 | H'0000 0003 |
| SA (Source Address) | H'1111 0000 | H'4444 0000 | H'7777 0000 |
| DA (Destination Address) | H'3333 0000 | H'5555 0000 | H'AAAA 0000 |
| TB (Transaction Byte) | H'0000 0800 | H'0000 0400 | H'0000 1000 |
| CFG (Configuration) | H'8142 2008 | H'8145 3008 | H'8046 6008 |
| ITVL (Interval) | H'0000 0000 | H'0000 0000 | H'0000 0000 |
| EXT (Extension) | H'0000 0000 | H'0000 0000 | H'0000 0000 |
| NXLA (Next Link Address) | H'0000 2000 | H'0000 5000 | H'0000 0000 |

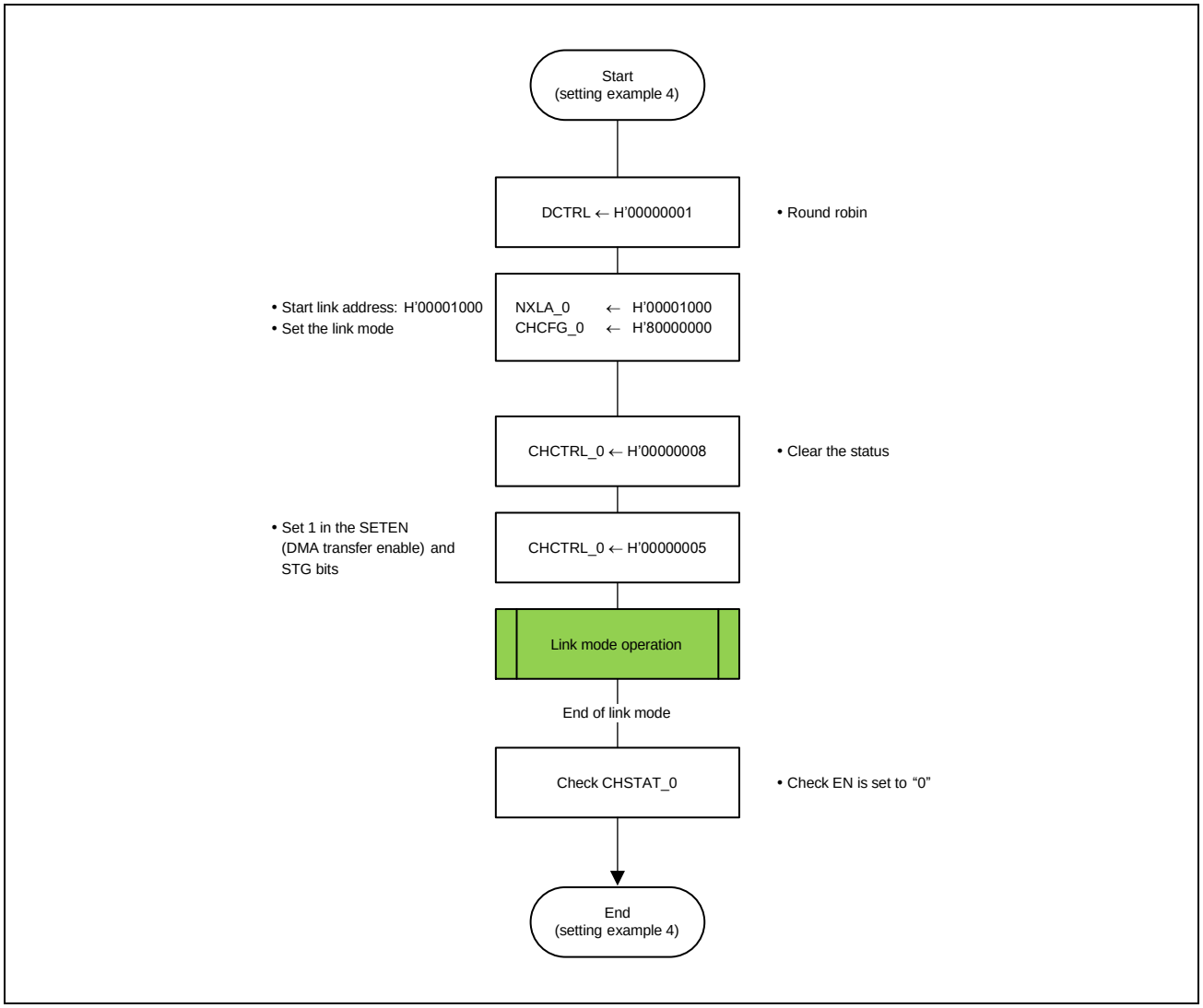



Figure 14.26 Setting Example 4

### 14.8.5 Next Register Set Continuous Execution Setting

The following figure shows the flowchart for executing DMA transfers continuously by using two Next register sets in register mode. While a DMA transaction is being executed using one Next register set, the other Next register set is set in order to continue to execute DMA transfers.

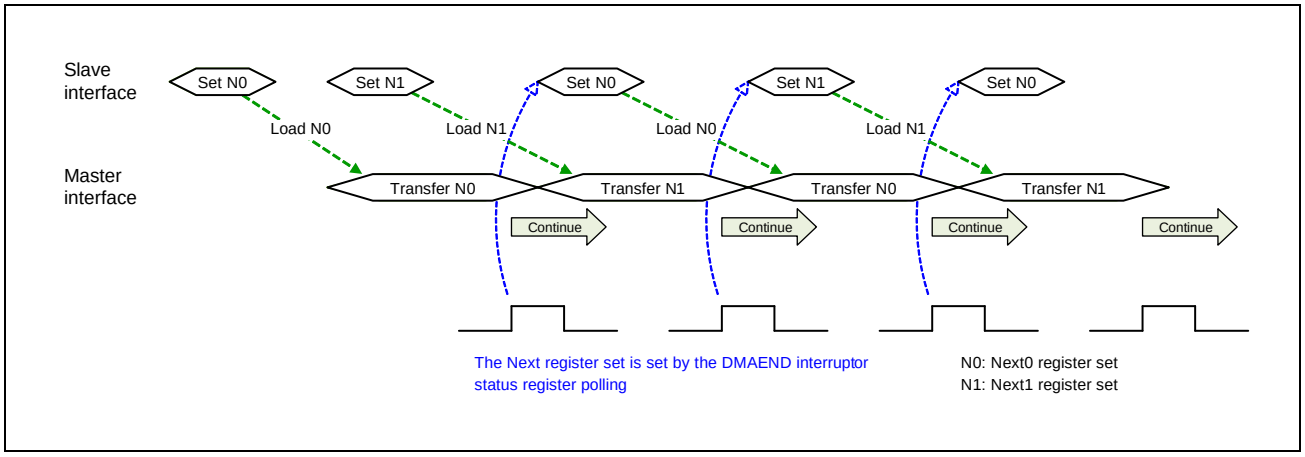



Figure 14.27 Image of Next Register Set Continuous Execution

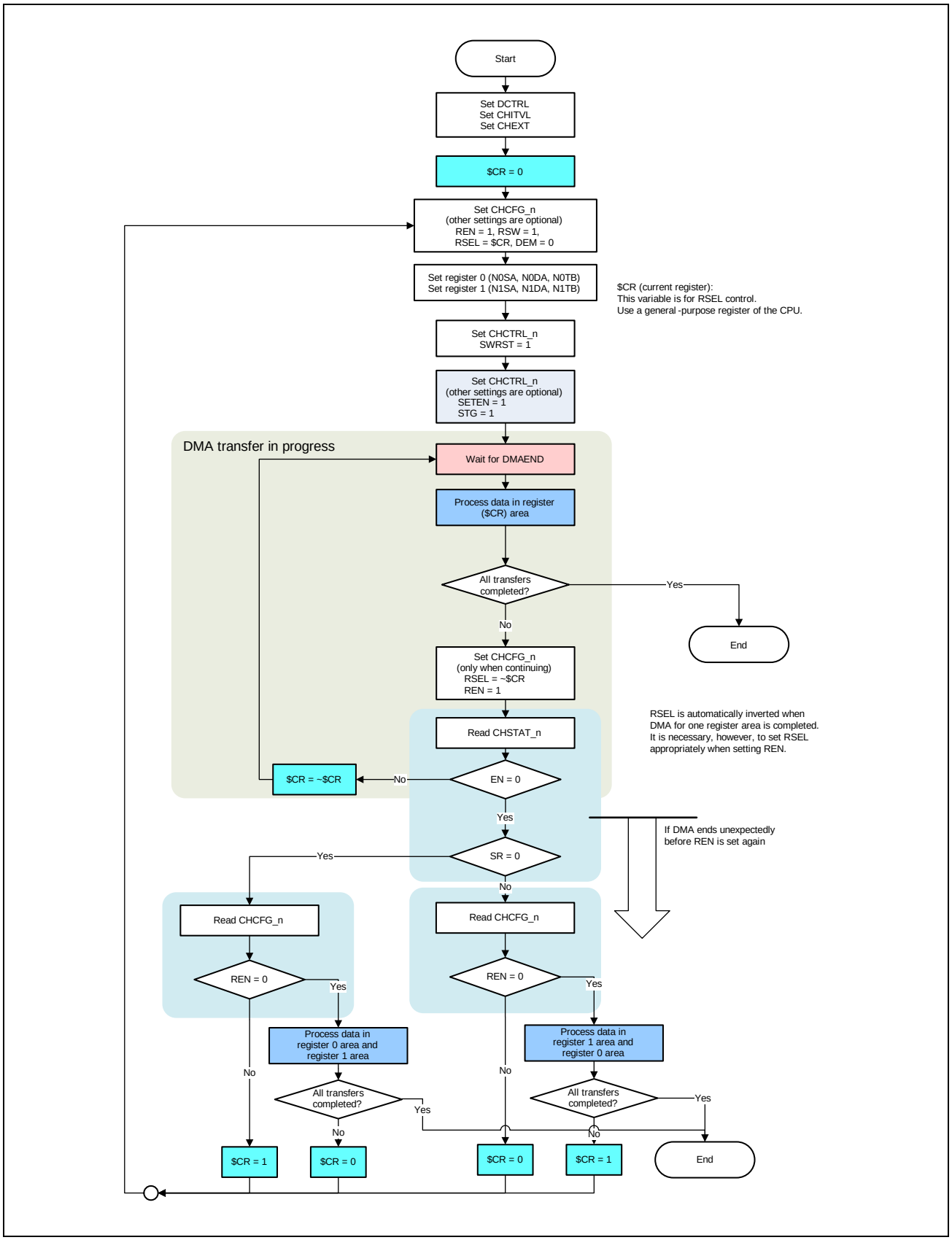



Figure 14.28 Example of Continuous DMA Execution by Using a Next Register Set

- Supplementary information

  First, save the data of the register sets to be used for DMA transfers (0 (N0SA, N0DA, and N0TB) and 1 (N1SA, N1DA, and N1TB)) to a general-purpose register of the CPU (the values of this register is referred to as $CR for the sake of convenience).

  Each time the DMA transfers for one register set are completed (the DMA transfer end interrupt is output), REN is automatically cleared to 0. In order to continue to execute DMA transfers, it is necessary to set REN of the CHCFG_n register every time the DMA transfer end interrupt is asserted. This register also contains the RSEL bit, and the value of this bit needs to be set appropriately as well. Therefore, use $CR.

  In this mode, two Next register sets are executed continuously. However, if CLREN is not set before the DMA transaction is completed (the next DMA transfer end interrupt is output), continuous execution stops. In this case, how much of data has been transferred can be checked by reading the SR and EN bits of the CHSTAT_n register and the REN bit of the CHCFG_n register. To restart the DMA transaction, follow the flowchart shown above.

## 14.9 Points for Caution

### 14.9.1 Problem of the Interrupt Signal Overtaking DMA Transfer

#### 14.9.1.1 Overview

This unit asserts the DMAINT interrupt signal on completion of transfer and this serves as a trigger for the CPU to start processing the data.

However, in cases such as where the device for writing is on another bus, the final data to be written may not yet have reached the device for writing at the time the DMAINT interrupt signal is asserted.

If the CPU accesses a device that has detected the DMAINT interrupt signal before the final data of the DMA transaction have been written to the device, it may be using some old data from before the writing.

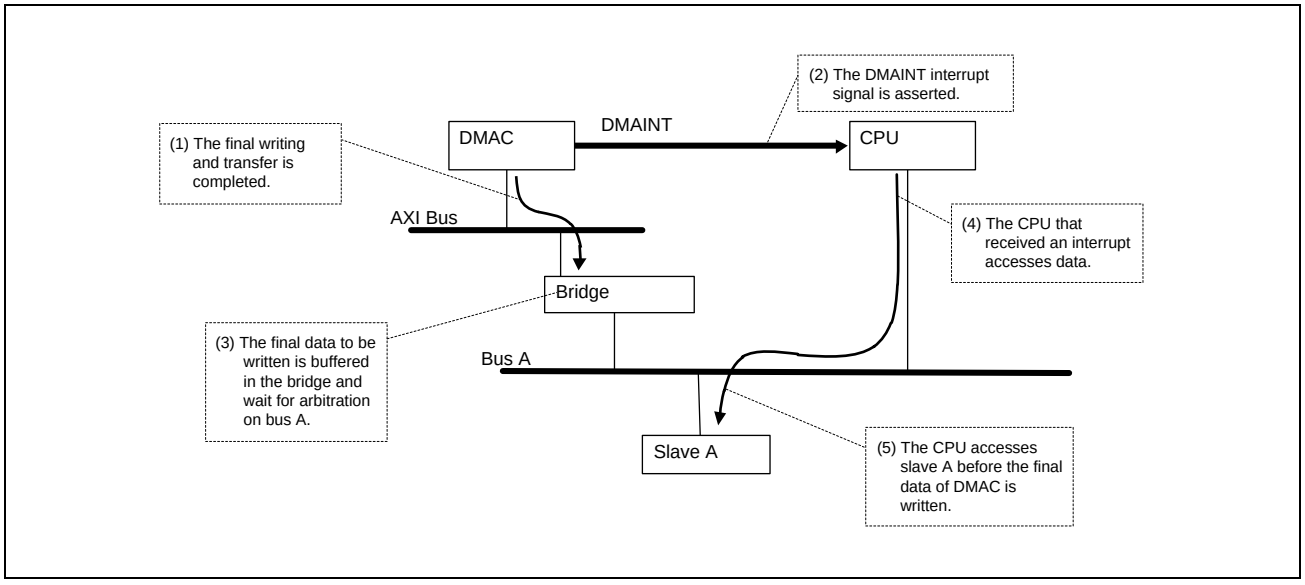



Figure 14-29 Example when Access by the CPU Overtakes Writing of the Final Data by the DMAC

#### 14.9.1.2 Workaround

The following describes workaround for the above problems.

**Setting AWCACHE as non-bufferable**

Set the AWCACHE signal as non-bufferable for DMA transactions. By doing this, the above problems can be avoided because the DMAINT interrupt signal will only be asserted after the completion of writing to the device for access.

Setting all transfers to non-bufferable may reduce the efficiency of transfer. In such cases, divide up the register sets or descriptors by setting most of the transactions to bufferable for transfer, and then set only the last transfer to non-bufferable.

*Exception*)

The effect is not obtained if the device in the access path does not refer to AWCACHE.

# 15. System Counter (SYC)

SYC generates the count value used by the generic timer built into Cortex-A55.

It uses the Timestamp generator, which is one of the components of Arm CoreSight SoC-400, to generate a 64-bit count value. For more information on the Timestamp generator, please refer to *the ARM CoreSight SoC-400 Technical Reference Manual*.

## 15.1 Overview

### 15.1.1 Features

- Count by 24 MHz clock (SYC_CNT_CLK)
- 64 bit gray code counter value generation
  - Convert the value generated by Timestamp generator to Gray code and output
- Access control
  - Access control for the two interfaces of the Timestamp generator
- Halt on Debug function
  - Counting can be stopped/restarted during debugging according to request from CoreSight

### 15.1.2 Block Diagram of SYC

The SYC connection diagram is shown in **Figure 15.1** and the SYC block diagram is shown in **Figure 15.2**.

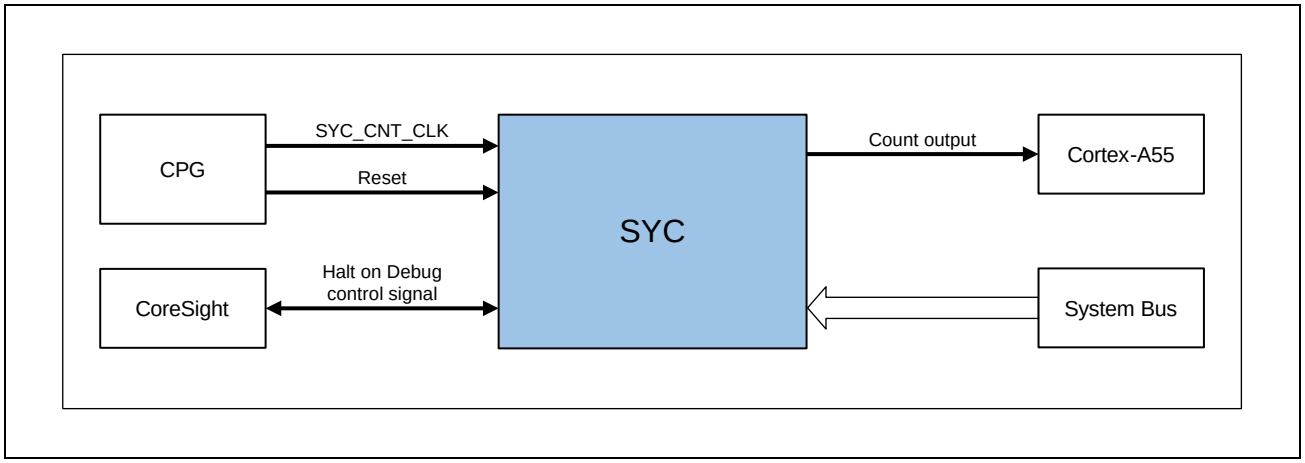



Figure 15.1 SYC Connection Diagram

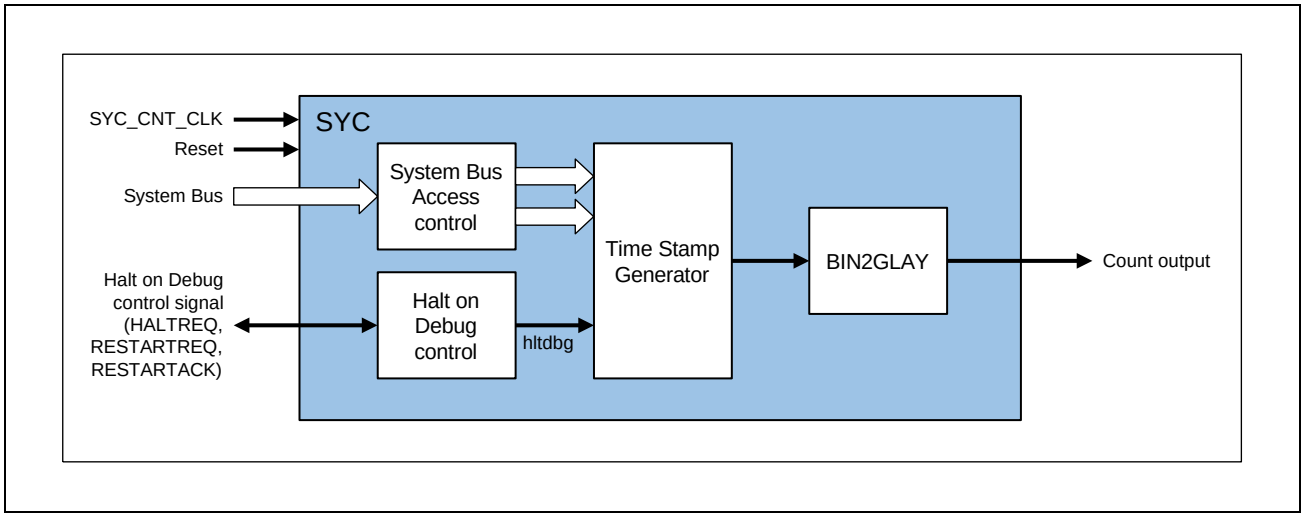



Figure 15.2 SYC Block Diagram

## 15.2 Address Space

**Table 15.1** shows the SYC address space. SYC has an address space of 8 KB. The SYC address space is offset from the base address. The base address of SYC is as follows.

SYC base address: H'0_1100_0000 (Overall Address Space)

Table 15.1 SYC Address Space

| Offset Address | Region Name | Access Target |
|---|---|---|
| H'0000 to H'0FFF | PSELCTRL region | PSELCTRL region & PSELCTRL region Management registers |
| H'1000 to H'1FFF | PSELREAD region | PSELREAD region & PSELREAD region Management registers |

## 15.3 Register Descriptions

For more information on SYC registers, please refer ARM CoreSight SoC-400 Technical Reference Manual 3.10 Timestamp generator.

## 15.4 Operation

### 15.4.1 System Bus Access Control

Access control from one system bus interface of SYC to two APB interfaces of Timestamp Generator is performed.

Controls so that the PSELCTRL area can be accessed when accessing the first half 4 KB of the 8 KB address area of SYC, and the PSELREAD area can be accessed when accessing the latter 4 KB.

### 15.4.2 Halt on Debug Function

SYC supports the Halt on Debug function included in Timestamp Generator. When HALTREQ is asserted from CoreSight, the counting operation of Timestamp Generator is stopped.

When RESTARTREQ is asserted from CoreSight, the counting operation of Timestamp Generator is restarted.

HALTREQ and RESTARTREQ are controlled by CoreSight’s CTI (Cross Trigger Interface).

# 16. Multi-Function Timer Pulse Unit 3 (MTU3a)

NOTE

This LSI does not have A/D converter.

Related descriptions of A/D converter in this section should be ignored.

## 16.1 Overview

This LSI has an on-chip multi-function timer pulse unit 3 (MTU3a), consisting of eight 16-bit timer channels and one 32- bit timer channel.

**Table 16.1** shows the specifications of the MTU and **Table 16.2** lists the functions of the MTU. **Figure 16.1** and **Figure 16.2** are block diagrams of the MTU.

Table 16.1 MTU Specifications (1/2)

| Item | Description |
|---|---|
| Pulse input/output | 28 lines max. |
| Pulse input | 3 lines |
| Count clock | 11 clocks for each channel (14 clocks for MTU0, 12 clocks for MTU2, and 10 clocks for MTU5, four clocks for MTU1-MTU2 combination (when LWA = 1)) |
| Operating frequency | Up to 100 MHz |
| Available operations | [MTU0 to MTU4, MTU6, MTU7, and MTU8]<br>• Waveform output on compare match<br>• Input capture function (noise filter setting available)<br>• Counter-clearing operation<br>• Simultaneous writing to multiple timer counters (TCNT) (excluding MTU8)<br>• Simultaneous clearing on compare match or input capture (excluding MTU8)<br>• Simultaneous input and output to registers in synchronization with counter operations (excluding MTU8)<br>• Up to 12-phase PWM output in combination with synchronous operation (excluding MTU8) |
| | [MTU0 MTU3, MTU4, MTU6, MTU7, and MTU8]<br>• Buffer operation specifiable |
| | [MTU1, MTU2]<br>• Phase counting mode can be specified independently<br>• 32-bit phase counting mode can be specified for interlocked operation of MTU1 and MTU2 (when TMDR3.LWA = 1)<br>• Cascade connection operation available |
| | [MTU3, MTU4, MTU6, and MTU7]<br>• Through interlocked operation of MTU3/4 and MTU6/7, the positive and negative signals in six phases (12 phases in total) can be output in complementary PWM and reset-synchronized PWM operation.<br>• In complementary PWM mode, values can be transferred from buffer registers to temporary registers at crests and troughs of the timer-counter values or when the buffer registers (TGRD registers in MTU4 and MTU7) are written to.<br>• Double-buffering selectable in complementary PWM mode |
| | [MTU3 and MTU4]<br>• Through interlocking with MTU0, a mode for driving AC synchronous motors (brushless DC motors) by using complementary PWM output and reset-synchronized PWM output is settable and allows the selection of two types of waveform output (chopping or level) |

Table 16.1 MTU Specifications (2/2)

| Item | Description |
|---|---|
| Available operations | [MTU5]<br>• Capable of operation as a dead-time compensation counter |
| Interrupt-skipping function | [MTU0/MTU5, MTU1, MTU2, and MTU8]<br>32-bit phase counting mode specifiable by combining MTU1 and MTU2 and through interlocked operation with MTU0/MTU5 and MTU8 |
| Interrupt sources | • In complementary PWM mode, interrupts on crests and troughs of counter values and triggers to start conversion by the A/D converter can be skipped |
| Buffer operation | 43 sources |
| Trigger generation | A/D converter start triggers can be generated |
| | A/D converter start request delaying function enables A/D converter to be started with any desired timing and to be synchronized with PWM output |
| Low power consumption function | The MTU3a can be placed in the module-stop state. |

Table 16.2 MTU Functions (1/2)

| Item | | MTU0 | MTU1 | MTU2 | MTU1 & MTU2 (LWA = 1) | MTU3 | MTU4 | MTU5 | MTU6 | MTU7 | MTU8 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Count clock | | Pϕ/1<br>Pϕ/2<br>Pϕ/4<br>Pϕ/8<br>Pϕ/16<br>Pϕ/32<br>Pϕ/64<br>Pϕ/256<br>Pϕ/1024<br>MTCLKA<br>MTCLKB<br>MTCLKC<br>MTCLKD<br>MTIOC1A | Pϕ/1<br>Pϕ/2<br>Pϕ/4<br>Pϕ/8<br>Pϕ/16<br>Pϕ/32<br>Pϕ/64<br>Pϕ/256<br>Pϕ/1024<br>MTCLKA<br>MTCLKB | Pϕ/1<br>Pϕ/2<br>Pϕ/4<br>Pϕ/8<br>Pϕ/16<br>Pϕ/32<br>Pϕ/64<br>Pϕ/256<br>Pϕ/1024<br>MTCLKA<br>MTCLKB<br>MTCLKC | MTCLKA<br>MTCLKB<br>MTCLKC<br>MTCLKD | Pϕ/1<br>Pϕ/2<br>Pϕ/4<br>Pϕ/8<br>Pϕ/16<br>Pϕ/32<br>Pϕ/64<br>Pϕ/256<br>Pϕ/1024<br>MTCLKA<br>MTCLKB | Pϕ/1<br>Pϕ/2<br>Pϕ/4<br>Pϕ/8<br>Pϕ/16<br>Pϕ/32<br>Pϕ/64<br>Pϕ/256<br>Pϕ/1024<br>MTCLKA<br>MTCLKB | Pϕ/1<br>Pϕ/2<br>Pϕ/4<br>Pϕ/8<br>Pϕ/16<br>Pϕ/32<br>Pϕ/64<br>Pϕ/256<br>Pϕ/1024<br>MTIOC1A | Pϕ/1<br>Pϕ/2<br>Pϕ/4<br>Pϕ/8<br>Pϕ/16<br>Pϕ/32<br>Pϕ/64<br>Pϕ/256<br>Pϕ/1024<br>MTCLKA<br>MTCLKB | Pϕ/1<br>Pϕ/2<br>Pϕ/4<br>Pϕ/8<br>Pϕ/16<br>Pϕ/32<br>Pϕ/64<br>Pϕ/256<br>Pϕ/1024<br>MTCLKA<br>MTCLKB | Pϕ/1<br>Pϕ/2<br>Pϕ/4<br>Pϕ/8<br>Pϕ/16<br>Pϕ/32<br>Pϕ/64<br>Pϕ/256<br>Pϕ/1024<br>MTCLKA<br>MTCLKB |
| External clocks in phase-counting mode | | — | MTCLKA<br>MTCLKB | MTCLKA<br>MTCLKB<br>MTCLKC<br>MTCLKD | MTCLKA<br>MTCLKB<br>MTCLKC<br>MTCLKD | — | — | — | — | — | — |
| General registers (TGR) | | TGRA<br>TGRB<br>TGRE | TGRA<br>TGRB | TGRA<br>TGRB | TGRALW<br>TGRBLW | TGRA<br>TGRB | TGRA<br>TGRB | TGRU<br>TGRV<br>TGRW | TGRA<br>TGRB | TGRA<br>TGRB | TGRA<br>TGRB |
| General registers/ buffer registers | | TGRC<br>TGRD<br>TGRF | — | — | — | TGRC<br>TGRD<br>TGRE | TGRC<br>TGRD<br>TGRE<br>TGRF | — | TGRC<br>TGRD<br>TGRE | TGRC<br>TGRD<br>TGRE<br>TGRF | TGRC<br>TGRD |
| I/O pins | | MTIOC0A<br>MTIOC0B<br>MTIOC0C<br>MTIOC0D | MTIOC1A<br>MTIOC1B | MTIOC2A<br>MTIOC2B | MTIOC1A<br>MTIOC1B | MTIOC3A<br>MTIOC3B<br>MTIOC3C<br>MTIOC3D | MTIOC4A<br>MTIOC4B<br>MTIOC4C<br>MTIOC4D | MTIC5U<br>MTIC5V<br>MTIC5W | MTIOC6A<br>MTIOC6B<br>MTIOC6C<br>MTIOC6D | MTIOC7A<br>MTIOC7B<br>MTIOC7C<br>MTIOC7D | MTIOC8A<br>MTIOC8B<br>MTIOC8C<br>MTIOC8D |
| Counter clear function | | TGR compare match or input capture | TGR compare match or input capture | TGR compare match or input capture | TGRALW/ TGRBLW input capture | TGR compare match or input capture | TGR compare match or input capture | TGR compare match or input capture | TGR compare match or input capture | TGR compare match or input capture | TGR compare match or input capture |
| Compare match output | 0 output | ✓ | ✓ | ✓ | — | ✓ | ✓ | — | ✓ | ✓ | ✓ |
| | 1 output | ✓ | ✓ | ✓ | — | ✓ | ✓ | — | ✓ | ✓ | ✓ |
| | Toggle output | ✓ | ✓ | ✓ | — | ✓ | ✓ | — | ✓ | ✓ | ✓ |
| Input capture function | | ✓ | ✓ | ✓ | ✓*1 | ✓ | ✓ | ✓ | ✓ | ✓ | ✓*2 |
| Synchronous operation | | ✓ | ✓ | ✓ | — | ✓ | ✓ | — | ✓ | ✓ | — |
| PWM mode 1 | | ✓ | ✓ | ✓ | — | ✓ | ✓ | — | ✓ | ✓ | — |
| PWM mode 2 | | ✓ | ✓ | ✓ | — | — | — | — | — | — | — |
| Complementary PWM mode | | — | — | — | — | ✓ | ✓ | — | ✓ | ✓ | — |
| Reset-synchronized PWM mode | | — | — | — | — | ✓ | ✓ | — | ✓ | ✓ | — |

Table 16.2 MTU Functions (2/2)

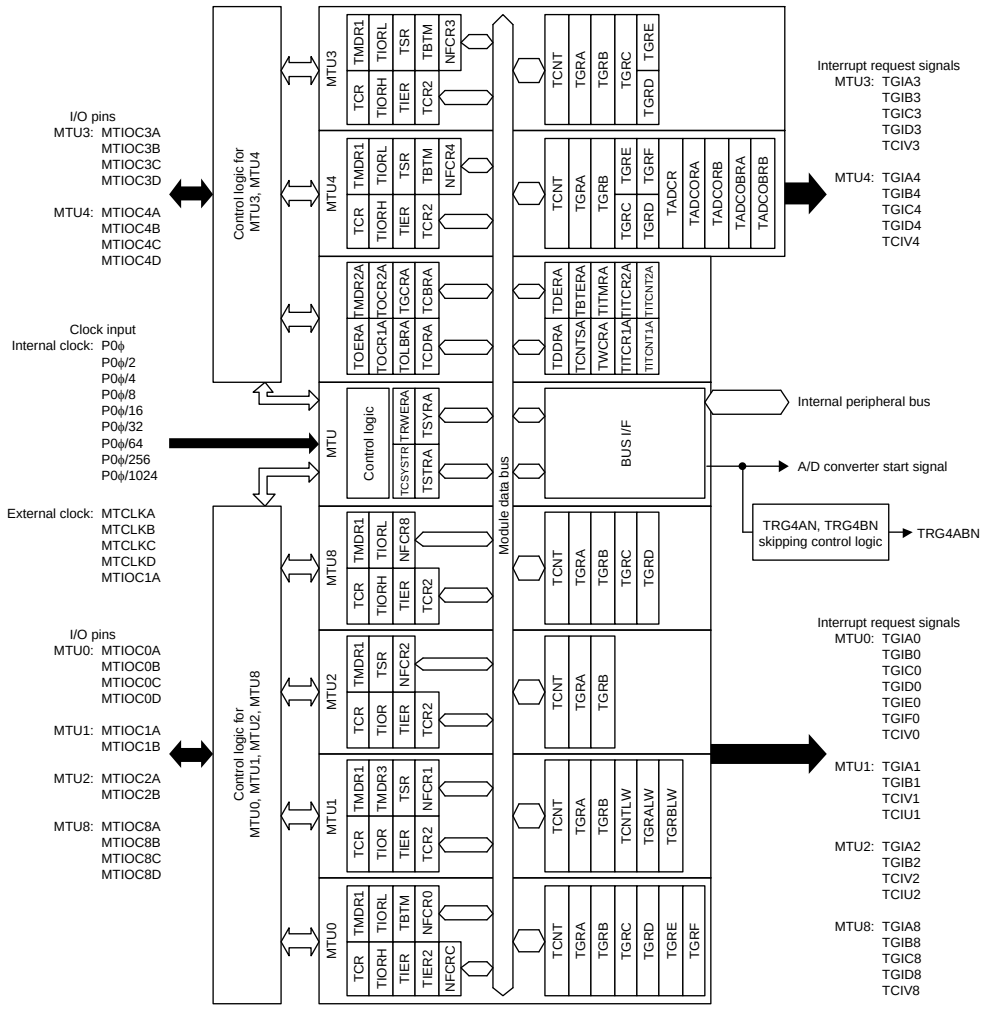
| Item | MTU0 | MTU1 | MTU2 | MTU1 & MTU2 (LWA = 1) | MTU3 | MTU4 | MTU5 | MTU6 | MTU7 | MTU8 |
|---|---|---|---|---|---|---|---|---|---|---|
| AC synchronous motor drive mode | ✓ | — | — | — | ✓ | ✓ | — | — | — | — |
| Phase counting mode | — | ✓ | ✓ | ✓ | — | — | — | — | — | — |
| Buffer operation | ✓ | — | — | — | ✓ | ✓ | — | ✓ | ✓ | ✓ |
| Dead time compensation counter function | — | — | — | — | — | — | — | — | — | — |
| DMAC activation | TGR compare match or input capture | TGR compare match or input capture | TGR compare match or input capture | TGRALW/ TGRBLW input capture | TGR compare match or input capture | TGR compare match or input capture and TCNT overflow/ underflow*3 | TGR compare match or input capture | TGR compare match or input capture | TGR compare match or input capture and TCNT overflow/ underflow*3 | TGR compare match or input capture |
| A/D converter start trigger | TGRA compare match or input capture TGRE compare match | TGRA compare match or input capture | TGRA compare match or input capture | TGRALW input capture | TGRA compare match or input capture | TGRA compare match or input capture, or TCNT underflow (trough) in complementary PWM mode | — | TGRA compare match or input capture | TGRA compare match or input capture, or TCNT underflow (trough) in complementary PWM mode | — |
| Interrupt sources | Seven sources<br>• Compare match or input capture 0A<br>• Compare match or input capture 0B<br>• Compare match or input capture 0C<br>• Compare match or input capture 0D<br>• Compare match0E<br>• Compare match0F<br>• Overflow | Four sources<br>• Compare match or input capture 1A<br>• Compare match or input capture 1B<br>• Overflow<br>• Underflow | Four sources<br>• Compare match or input capture 2A<br>• Compare match or input capture 2B<br>• Overflow<br>• Underflow | Four sources<br>• Input capture 1A<br>• Input capture 1B<br>• Overflow<br>• Underflow | Five sources<br>• Compare match or input capture 3A<br>• Compare match or input capture 3B<br>• Compare match or input capture 3C<br>• Compare match or input capture 3D<br>• Overflow | Five sources<br>• Compare match or input capture 4A<br>• Compare match or input capture 4B<br>• Compare match or input capture 4C<br>• Compare match or input capture 4D<br>• Overflow or underflow*3 | Three sources<br>• Compare match or input capture 5U<br>• Compare match or input capture 5V<br>• Compare match or input capture 5W | Five sources<br>• Compare match or input capture 6A<br>• Compare match or input capture 6B<br>• Compare match or input capture 6C<br>• Compare match or input capture 6D<br>• Overflow | Five sources<br>• Compare match or input capture 7A<br>• Compare match or input capture 7B<br>• Compare match or input capture 7C<br>• Compare match or input capture 7D<br>• Overflow or underflow*3 | Five sources<br>• Compare match or input capture 8A<br>• Compare match or input capture 8B<br>• Compare match or input capture 8C<br>• Compare match or input capture 8D<br>• Overflow |
| A/D converter start request delaying function | — | — | — | — | — | • A/D converter start request at a match between TADCORA and TCNT or A/D converter start request at a match between TADCORB and TCNT | — | — | • A/D converter start request at a match between TADCORA and TCNT or A/D converter start request at a match between TADCORB and TCNT | — |
| Interrupt skipping 1 | — | — | — | — | • Skips TGRA compare match interrupts | • Skips TCIV interrupts | — | • Skips TGRA compare match interrupts | • Skips TCIV interrupts | — |
| Interrupt skipping 2 | — | — | — | — | — | • Skipping in compare count between TADCORA and TCNT, and TADCORB and TCNT | — | — | • Skipping in compare count between TADCORA and TCNT, and TADCORB and TCNT | — |

**Remarks:** ✓: Possible —: Not possible

Note 1. When LWA is 1, the TGRALW capture source can be selected from either of the following: an input from MTIOC1A or MTU0.TGRA compare match/input capture event.
The TGRBLW capture source can be selected from any of the following: an input from MTIOC1B, MTU0.TGRC compare match/ input capture event, or MTU8.TGRC compare match event.

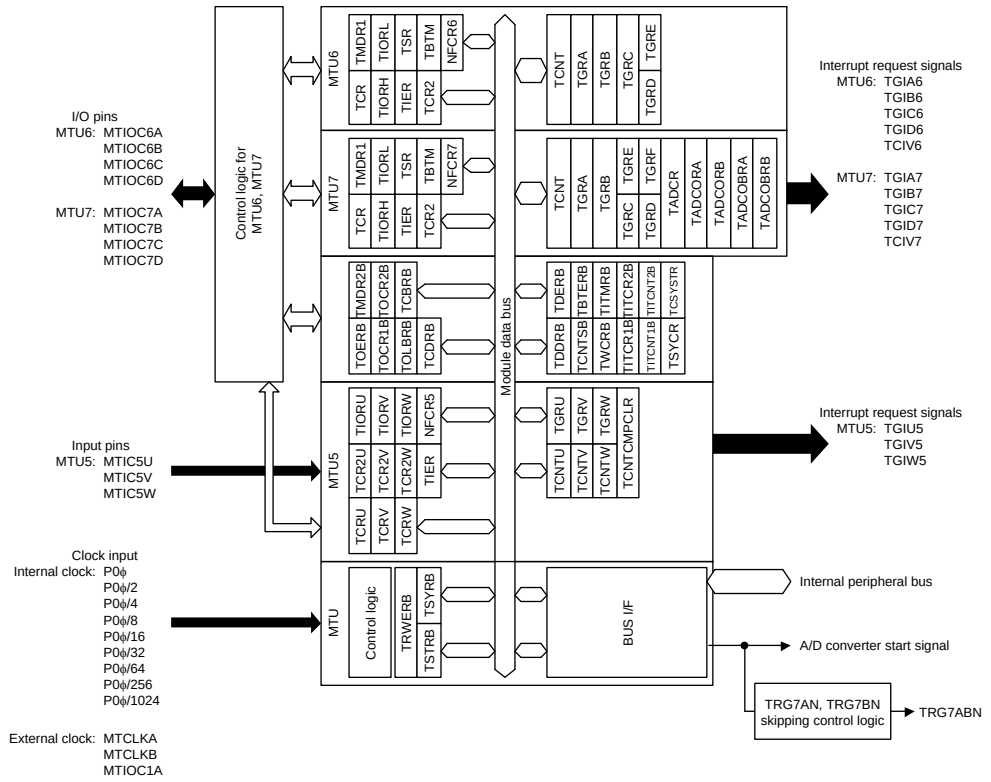
Note 2. Capture in MTU8 is supported only in normal mode.

Note 3. The underflow interrupt source is valid only in complementary PWM mode.




| | |
|---|---|
| TCR: | Timer control register |
| TCR2: | Timer control register 2 |
| TMDR: | Timer mode register 1 |
| TMDR2A: | Timer mode register 2 |
| TMDR3: | Timer mode register 3 |
| 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 |
| TIER2: | Timer interrupt enable register 2 |
| TSR: | Timer status register |
| TBTM: | Timer buffer operation transfer mode register |
| TCNT: | Timer counter |
| TCNTLW: | Timer longword counter |
| TGRA: | Timer general register A |
| TGRALW: | Timer longword general register A |
| TGRB: | Timer general register B |
| TGRBLW: | Timer longword general register B |
| TGRC: | Timer general register C |
| TGRD: | Timer general register D |
| TGRE: | Timer general register E |
| TGRF: | Timer general register F |
| TSTRA: | Timer start register |
| TSYRA: | Timer synchronous register |
| TCSYSTR: | Timer counter synchronous start register |
| TRWERA: | Timer read/write enable register |
| TOERA: | Timer output master enable register |
| TOCR1A: | Timer output control register 1 |
| TOCR2A: | Timer output control register 2 |
| TOLBRA: | Timer output level buffer register |
| TGCRA: | Timer gate control register A |
| TCNTSA: | Timer subcounter |
| TCDRA: | Timer cycle data register |
| TCBRA: | Timer cycle buffer register |
| TDDRA: | Timer dead time data register |
| TDERA: | Timer dead time enable register |
| TBTERA: | Timer buffer transfer set register |
| TWCRA: | Timer waveform control register |
| TICCR: | Timer input capture control register |
| TADCR: | Timer A/D converter start request control register |
| TADCORA: | Timer A/D converter start request cycle set register A |
| TADCORB: | Timer A/D converter start request cycle set register B |
| TADCOBRA: | Timer A/D converter start request cycle set buffer register A |
| TADCOBRB: | Timer A/D converter start request cycle set buffer register B |
| TITMRA: | Timer interrupt skipping mode register |
| TITCR1A: | Timer interrupt skipping set register 1 |
| TITCR2A: | Timer interrupt skipping set register 2 |
| TITCNT1A: | Timer interrupt skipping counter 1 |
| TITCNT2A: | Timer interrupt skipping counter 2 |
| NFCRn: | Noise filter control register n |

Figure 16.1 Block Diagram of MTU (MTU0 to MTU4, MTU8)




| | |
|---|---|
| TCR: | Timer control register |
| TCRU: | Timer control register U |
| TCR2U: | Timer control register 2U |
| TCRV: | Timer control register V |
| TCR2V: | Timer control register 2V |
| TCRW: | Timer control register W |
| TCR2W: | Timer control register 2W |
| TMDR1: | Timer mode register 1 |
| TMDR2B: | Timer mode register 2 |
| TIORH: | Timer I/O control register H |
| TIORL: | Timer I/O control register L |
| TIORU: | Timer I/O control register U |
| TIORV: | Timer I/O control register V |
| TIORW: | Timer I/O control register W |
| TIER: | Timer interrupt enable register |
| TSR: | Timer status register |
| TCNT: | Timer counter |
| TCNTU: | Timer counter U |
| TCNTV: | Timer counter V |
| TCNTW: | Timer counter W |
| 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 |
| TGRU: | Timer general register U |
| TGRV: | Timer general register V |
| TGRW: | Timer general register W |
| TCNTCMPCLR: | Timer compare match clear register |

| | |
|---|---|
| TSTRB: | Timer start register |
| TSYRB: | Timer synchronous register |
| TSYCR: | Timer synchronous clear register |
| TCSYSTR: | Timer counter synchronous start register |
| TRWERB: | Timer read/write enable register |
| TOERB: | Timer output master enable register |
| TOCR1B: | Timer output control register 1 |
| TOCR2B: | Timer output control register 2 |
| TOLBRB: | Timer output level buffer register |
| TCDRB: | Timer cycle data register |
| TCBRB: | Timer cycle buffer register |
| TDDRB: | Timer dead time data register |
| TDERB: | Timer dead time enable register |
| TCNTSB: | Timer subcounter |
| TBTERB: | Timer buffer transfer set register |
| TWCRB: | Timer waveform control register |
| TADCR: | Timer A/D converter start request control register |
| TADCORA: | Timer A/D converter start request cycle set register A |
| TADCORB: | Timer A/D converter start request cycle set register B |
| TADCOBRA: | Timer A/D converter start request cycle set buffer register A |
| TADCOBRB: | Timer A/D converter start request cycle set buffer register B |
| TITMRB: | Timer interrupt skipping mode register |
| TITCR1B: | Timer interrupt skipping set register 1 |
| TITCR2B: | Timer interrupt skipping set register 2 |
| TITCNT1B: | Timer interrupt skipping counter 1 |
| TITCNT2B: | Timer interrupt skipping counter 2 |
| NFCRn | Noise filter control register n |

Figure 16.2 Block Diagram of MTU (MTU5 to MTU7)

**Table 16.3** shows the configuration of pins for the MTU.

Table 16.3 Pin Configuration of the MTU

| Channel | Pin Name | I/O | Function |
|---|---|---|---|
| MTU | MTCLKA | Input | External clock A input pin (MTU1/MTU2 phase counting mode A phase input) |
| | MTCLKB | Input | External clock B input pin (MTU1/MTU2 phase counting mode B phase input) |
| | MTCLKC | Input | External clock C input pin (MTU2 phase counting mode A phase input) |
| | MTCLKD | Input | External clock D input pin (MTU2 phase counting mode B phase input) |
| MTU0 | MTIOC0A | I/O | MTU0 TGRA input capture input/output compare output/PWM output pin |
| | MTIOC0B | I/O | MTU0 TGRB input capture input/output compare output/PWM output pin |
| | MTIOC0C | I/O | MTU0 TGRC input capture input/output compare output/PWM output pin |
| | MTIOC0D | I/O | MTU0 TGRD input capture input/output compare output/PWM output pin |
| MTU1 | MTIOC1A | I/O | MTU1 TGRA input capture input/output compare output/PWM output pin |
| | MTIOC1B | I/O | MTU1 TGRB input capture input/output compare output/PWM output pin |
| MTU2 | MTIOC2A | I/O | MTU2 TGRA input capture input/output compare output/PWM output pin |
| | MTIOC2B | I/O | MTU2 TGRB input capture input/output compare output/PWM output pin |
| MTU3 | MTIOC3A | I/O | MTU3 TGRA input capture input/output compare output/PWM output pin |
| | MTIOC3B | I/O | MTU3 TGRB input capture input/output compare output/PWM output pin |
| | MTIOC3C | I/O | MTU3 TGRC input capture input/output compare output/PWM output pin |
| | MTIOC3D | I/O | MTU3 TGRD input capture input/output compare output/PWM output pin |
| MTU4 | MTIOC4A | I/O | MTU4 TGRA input capture input/output compare output/PWM output pin |
| | MTIOC4B | I/O | MTU4 TGRB input capture input/output compare output/PWM output pin |
| | MTIOC4C | I/O | MTU4 TGRC input capture input/output compare output/PWM output pin |
| | MTIOC4D | I/O | MTU4 TGRD input capture input/output compare output/PWM output pin |
| MTU5 | MTIC5U | Input | MTU5 TGRU input capture input/external pulse input pin |
| | MTIC5V | Input | MTU5 TGRV input capture input/external pulse input pin |
| | MTIC5W | Input | MTU5 TGRW input capture input/external pulse input pin |
| MTU6 | MTIOC6A | I/O | MTU6 TGRA input capture input/output compare output/PWM output pin |
| | MTIOC6B | I/O | MTU6 TGRB input capture input/output compare output/PWM output pin |
| | MTIOC6C | I/O | MTU6 TGRC input capture input/output compare output/PWM output pin |
| | MTIOC6D | I/O | MTU6 TGRD input capture input/output compare output/PWM output pin |
| MTU7 | MTIOC7A | I/O | MTU7 TGRA input capture input/output compare output/PWM output pin |
| | MTIOC7B | I/O | MTU7 TGRB input capture input/output compare output/PWM output pin |
| | MTIOC7C | I/O | MTU7 TGRC input capture input/output compare output/PWM output pin |
| | MTIOC7D | I/O | MTU7 TGRD input capture input/output compare output/PWM output pin |
| MTU8 | MTIOC8A | I/O | MTU8.TGRA input capture input/output compare output pin |
| | MTIOC8B | I/O | MTU8.TGRB input capture input/output compare output pin |
| | MTIOC8C | I/O | MTU8.TGRC input capture input/output compare output pin |
| | MTIOC8D | I/O | MTU8.TGRD input capture input/output compare output pin |

## 16.2 Register Descriptions

**Table 16.4** shows the register configuration. The MTU address space is offset from the base address. The base address of MTU is as follows.

MTU base address: H'0_1000_0000 (Overall Address Space)

Table 16.4 Register configuration (1/5)

| Channel | Register Name | Abbreviation | Address | Access size |
|---|---|---|---|---|
| MTU0 | Timer control register | TCR | H'1300 | 8 |
| | Timer mode register 1 | TMDR1 | H'1301 | 8 |
| | Timer I/O control register H | TIORH | H'1302 | 8 |
| | Timer I/O control register L | TIORL | H'1303 | 8 |
| | Timer interrupt enable register | TIER | H'1304 | 8 |
| | Timer counter | TCNT | H'1306 | 16 |
| | Timer general register A | TGRA | H'1308 | 16 |
| | Timer general register B | TGRB | H'130A | 16 |
| | Timer general register C | TGRC | H'130C | 16 |
| | Timer general register D | TGRD | H'130E | 16 |
| | Timer general register E | TGRE | H'1320 | 16 |
| | Timer general register F | TGRF | H'1322 | 16 |
| | Timer interrupt enable register 2 | TIER2 | H'1324 | 8 |
| | Timer buffer operation transfer mode register | TBTM | H'1326 | 8 |
| | Timer control register 2 | TCR2 | H'1328 | 8 |
| | Noise filter control register 0 | NFCR0 | H'1290 | 8 |
| | Noise filter control register C | NFCRC | H'1299 | 8 |
| MTU1 | Timer control register | TCR | H'1380 | 8 |
| | Timer mode register 1 | TMDR1 | H'1381 | 8 |
| | Timer I/O control register | TIOR | H'1382 | 8 |
| | Timer interrupt enable register | TIER | H'1384 | 8 |
| | Timer status register | TSR | H'1385 | 8 |
| | Timer counter | TCNT | H'1386 | 16 |
| | Timer general register A | TGRA | H'1388 | 16 |
| | Timer general register B | TGRB | H'138A | 16 |
| | Timer input capture control register | TICCR | H'1390 | 8 |
| | Timer mode register 3 | TMDR3 | H'1391 | 8 |
| | Timer control register 2 | TCR2 | H'1394 | 8 |
| | Timer longword counter | TCNTLW | H'13A0 | 32 |
| | Timer longword general register A | TGRALW | H'13A4 | 32 |
| | Timer longword general register B | TGRBLW | H'13A8 | 32 |
| | Noise filter control register 1 | NFCR1 | H'1291 | 8 |
| MTU2 | Timer control register | TCR | H'1400 | 8 |
| | Timer mode register 1 | TMDR1 | H'1401 | 8 |
| | Timer I/O control register | TIOR | H'1402 | 8 |
| | Timer interrupt enable register | TIER | H'1404 | 8 |
| | Timer status register | TSR | H'1405 | 8 |
| | Timer counter | TCNT | H'1406 | 16 |
| | Timer general register A | TGRA | H'1408 | 16 |

Table 16.4 Register configuration (2/5)

| Channel | Register Name | Abbreviation | Address | Access size |
|---|---|---|---|---|
| MTU2 | Timer general register B | TGRB | H'140A | 16 |
| | Timer control register 2 | TCR2 | H'140C | 8 |
| | Noise filter control register 2 | NFCR2 | H'1292 | 8 |
| MTU3 | Timer control register | TCR | H'1200 | 8 |
| | Timer mode register 1 | TMDR1 | H'1202 | 8 |
| | Timer I/O control register H | TIORH | H'1204 | 8 |
| | Timer I/O control register L | TIORL | H'1205 | 8 |
| | Timer interrupt enable register | TIER | H'1208 | 8 |
| | Timer counter | TCNT | H'1210 | 16 |
| | Timer general register A | TGRA | H'1218 | 16 |
| | Timer general register B | TGRB | H'121A | 16 |
| | Timer general register C | TGRC | H'1224 | 16 |
| | Timer general register D | TGRD | H'1226 | 16 |
| | Timer control register 2 | TCR2 | H'124C | 8 |
| | Timer general register E | TGRE | H'1272 | 16 |
| | Timer status register | TSR | H'122C | 8 |
| | Timer buffer operation transfer mode register | TBTM | H'1238 | 8 |
| | Noise filter control register 3 | NFCR3 | H'1293 | 8 |
| MTU4 | Timer control register | TCR | H'1201 | 8 |
| | Timer mode register 1 | TMDR1 | H'1203 | 8 |
| | Timer I/O control register H | TIORH | H'1206 | 8 |
| | Timer I/O control register L | TIORL | H'1207 | 8 |
| | Timer interrupt enable register | TIER | H'1209 | 8 |
| | Timer counter | TCNT | H'1212 | 16 |
| | Timer general register A | TGRA | H'121C | 16 |
| | Timer general register B | TGRB | H'121E | 16 |
| | Timer general register C | TGRC | H'1228 | 16 |
| | Timer general register D | TGRD | H'122A | 16 |
| | Timer control register 2 | TCR2 | H'124D | 8 |
| | Timer general register E | TGRE | H'1274 | 16 |
| | Timer general register F | TGRF | H'1276 | 16 |
| | Timer status register | TSR | H'122D | 8 |
| | Timer buffer operation transfer mode register | TBTM | H'1239 | 8 |
| | Timer A/D converter start request control register | TADCR | H'1240 | 16 |
| | Timer A/D converter start request cycle set register A | TADCORA | H'1244 | 16 |
| | Timer A/D converter start request cycle set register B | TADCORB | H'1246 | 16 |
| | Timer A/D converter start request cycle set buffer register A | TADCOBRA | H'1248 | 16 |
| | Timer A/D converter start request cycle set buffer register B | TADCOBRB | H'124A | 16 |
| | Noise filter control register 4 | NFCR4 | H'1294 | 8 |
| MTU5 | Timer counter U | TCNTU | H'1C80 | 16 |
| | Timer general register U | TGRU | H'1C82 | 16 |
| | Timer control register U | TCRU | H'1C84 | 8 |
| | Timer control register 2U | TCR2U | H'1C85 | 8 |
| | Timer I/O control register U | TIORU | H'1C86 | 8 |
| | Timer counter V | TCNTV | H'1C90 | 16 |

Table 16.4 Register configuration (3/5)

| Channel | Register Name | Abbreviation | Address | Access size |
|---|---|---|---|---|
| MTU5 | Timer general register V | TGRV | H'1C92 | 16 |
| | Timer control register V | TCRV | H'1C94 | 8 |
| | Timer control register 2V | TCR2V | H'1C95 | 8 |
| | Timer I/O control register V | TIORV | H'1C96 | 8 |
| | Timer counter W | TCNTW | H'1CA0 | 16 |
| | Timer general register W | TGRW | H'1CA2 | 16 |
| | Timer control register W | TCRW | H'1CA4 | 8 |
| | Timer control register 2W | TCR2W | H'1CA5 | 8 |
| | Timer I/O control register W | TIORW | H'1CA6 | 8 |
| | Timer interrupt enable register | TIER | H'1CB2 | 8 |
| | Timer start register | TSTR | H'1CB4 | 8 |
| | Timer compare match clear register | TCNTCMPCLR | H'1CB6 | 8 |
| | Noise filter control register 5 | NFCR5 | H'1A95 | 8 |
| MTU6 | Timer control register | TCR | H'1A00 | 8 |
| | Timer mode register 1 | TMDR1 | H'1A02 | 8 |
| | Timer I/O control register H | TIORH | H'1A04 | 8 |
| | Timer I/O control register L | TIORL | H'1A05 | 8 |
| | Timer interrupt enable register | TIER | H'1A08 | 8 |
| | Timer counter | TCNT | H'1A10 | 16 |
| | Timer general register A | TGRA | H'1A18 | 16 |
| | Timer general register B | TGRB | H'1A1A | 16 |
| | Timer general register C | TGRC | H'1A24 | 16 |
| | Timer general register D | TGRD | H'1A26 | 16 |
| | Timer control register 2 | TCR2 | H'1A4C | 8 |
| | Timer general register E | TGRE | H'1A72 | 16 |
| | Timer synchronous clear register | TSYCR | H'1A50 | 8 |
| | Timer status register | TSR | H'1A2C | 8 |
| | Timer buffer operation transfer mode register | TBTM | H'1A38 | 8 |
| | Noise filter control register 6 | NFCR6 | H'1A93 | 8 |
| MTU7 | Timer control register | TCR | H'1A01 | 8 |
| | Timer mode register 1 | TMDR1 | H'1A03 | 8 |
| | Timer I/O control register H | TIORH | H'1A06 | 8 |
| | Timer I/O control register L | TIORL | H'1A07 | 8 |
| | Timer interrupt enable register | TIER | H'1A09 | 8 |
| | Timer counter | TCNT | H'1A12 | 16 |
| | Timer general register A | TGRA | H'1A1C | 16 |
| | Timer general register B | TGRB | H'1A1E | 16 |
| | Timer general register C | TGRC | H'1A28 | 16 |
| | Timer general register D | TGRD | H'1A2A | 16 |
| | Timer control register 2 | TCR2 | H'1A4D | 8 |
| | Timer general register E | TGRE | H'1A74 | 16 |
| | Timer general register F | TGRF | H'1A76 | 16 |
| | Timer status register | TSR | H'1A2D | 8 |
| | Timer buffer operation transfer mode register | TBTM | H'1A39 | 8 |
| | Timer A/D converter start request control register | TADCR | H'1A40 | 16 |

Table 16.4 Register configuration (4/5)

| Channel | Register Name | Abbreviation | Address | Access size |
|---|---|---|---|---|
| MTU7 | Timer A/D converter start request cycle set register A | TADCORA | H'1A44 | 16 |
| | Timer A/D converter start request cycle set register B | TADCORB | H'1A46 | 16 |
| | Timer A/D converter start request cycle set buffer register A | TADCOBRA | H'1A48 | 16 |
| | Timer A/D converter start request cycle set buffer register B | TADCOBRB | H'1A4A | 16 |
| | Noise filter control register 7 | NFCR7 | H'1A94 | 8 |
| MTU8 | Timer control register | TCR | H'1600 | 8 |
| | Timer mode register 1 | TMDR1 | H'1601 | 8 |
| | Timer I/O control register H | TIORH | H'1602 | 8 |
| | Timer I/O control register L | TIORL | H'1603 | 8 |
| | Timer interrupt enable register | TIER | H'1604 | 8 |
| | Timer control register 2 | TCR2 | H'1606 | 8 |
| | Timer counter | TCNT | H'1608 | 32 |
| | Timer general register A | TGRA | H'160C | 32 |
| | Timer general register B | TGRB | H'1610 | 32 |
| | Timer general register C | TGRC | H'1614 | 32 |
| | Timer general register D | TGRD | H'1618 | 32 |
| | Noise filter control register 8 | NFCR8 | H'1298 | 8 |
| MTU | Timer output master enable register A | TOERA | H'120A | 8 |
| | Timer gate control register A | TGCRA | H'120D | 8 |
| | Timer output control register 1A | TOCR1A | H'120E | 8 |
| | Timer output control register 2A | TOCR2A | H'120F | 8 |
| | Timer cycle data register A | TCDRA | H'1214 | 16 |
| | Timer dead time data register A | TDDRA | H'1216 | 16 |
| | Timer subcounter A | TCNTSA | H'1220 | 16 |
| | Timer cycle buffer register A | TCBRA | H'1222 | 16 |
| | Timer interrupt skipping set register 1A | TITCR1A | H'1230 | 8 |
| | Timer interrupt skipping counter 1A | TITCNT1A | H'1231 | 8 |
| | Timer buffer transfer set register A | TBTERA | H'1232 | 8 |
| | Timer dead time enable register A | TDERA | H'1234 | 8 |
| | Timer output level buffer register A | TOLBRA | H'1236 | 8 |
| | Timer interrupt skipping mode register A | TITMRA | H'123A | 8 |
| | Timer interrupt skipping set register 2A | TITCR2A | H'123B | 8 |
| | Timer interrupt skipping counter 2A | TITCNT2A | H'123C | 8 |
| | Timer waveform control register A | TWCRA | H'1260 | 8 |
| | Timer mode register 2A | TMDR2A | H'1270 | 8 |
| | Timer read/write enable register A | TRWERA | H'1284 | 8 |
| | Timer start register A | TSTRA | H'1280 | 8 |
| | Timer synchronous register A | TSYRA | H'1281 | 8 |
| | Timer output master enable register B | TOERB | H'1A0A | 8 |
| | Timer output control register 1B | TOCR1B | H'1A0E | 8 |
| | Timer output control register 2B | TOCR2B | H'1A0F | 8 |
| | Timer cycle data register B | TCDRB | H'1A14 | 16 |
| | Timer dead time data register B | TDDRB | H'1A16 | 16 |
| | Timer subcounter B | TCNTSB | H'1A20 | 16 |
| | Timer cycle buffer register B | TCBRB | H'1A22 | 16 |

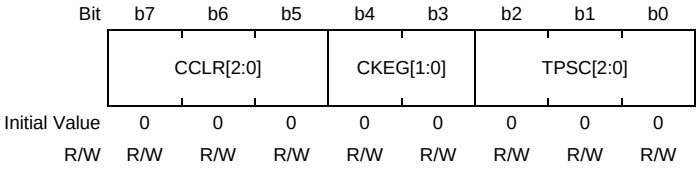
Table 16.4 Register configuration (5/5)

| Channel | Register Name | Abbreviation | Address | Access size |
|---|---|---|---|---|
| MTU | Timer interrupt skipping set register 1B | TITCR1B | H'1A30 | 8 |
| | Timer interrupt skipping counter 1B | TITCNT1B | H'1A31 | 8 |
| | Timer buffer transfer set register B | TBTERB | H'1A32 | 8 |
| | Timer dead time enable register B | TDERB | H'1A34 | 8 |
| | Timer output level buffer register B | TOLBRB | H'1A36 | 8 |
| | Timer interrupt skipping mode register B | TITMRB | H'1A3A | 8 |
| | Timer interrupt skipping set register 2B | TITCR2B | H'1A3B | 8 |
| | Timer interrupt skipping counter 2B | TITCNT2B | H'1A3C | 8 |
| | Timer waveform control register B | TWCRB | H'1A60 | 8 |
| | Timer mode register 2B | TMDR2B | H'1A70 | 8 |
| | Timer start register B | TSTRB | H'1A80 | 8 |
| | Timer synchronous register B | TSYRB | H'1A81 | 8 |
| | Timer read/write enable register B | TRWERB | H'1A84 | 8 |
| | Timer counter synchronous start register | TCSYSTR | H'1282 | 8 |

## 16.2.1 Timer Control Register (TCR)

### (1) MTU0.TCR, MTU1.TCR, MTU2.TCR, MTU3.TCR, MTU4.TCR, MTU6.TCR, MTU7.TCR, MTU8.TCR

**Address(es):** MTU0.TCR H'0_1000_1300, MTU1.TCR H'0_1000_1380, MTU2.TCR H'0_1000_1400, MTU3.TCR H'0_1000_1200, MTU4.TCR H'0_1000_1201, MTU6.TCR H'0_1000_1A00, MTU7.TCR H'0_1000_1A01, MTU8.TCR H'0_1000_1600




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| b2 to b0 | TPSC[2:0] | All 0 | R/W | Time Prescaler Select<br>See **Table 16.7** to **Table 16.10**. |
| b4, b3 | CKEG[1:0] | All 0 | R/W | Clock Edge Select<br>b4 b3<br>0 0: Count at rising edge<br>0 1: Count at falling edge<br>1 x: Count at both edges |
| b7 to b5 | CCLR[2:0] | All 0 | R/W | Counter Clear Source Select<br>See **Table 16.5** and **Table 16.6**. |

**Remarks:** x: Don't care

The TCR register controls the TCNT operation for each channel in combination with the TCR2 register. The MTU has a total of 11 TCR registers, one each for MTU0 to MTU4, MTU 6, MTU7, and MTU8 and three (TCRU, TCRV, and TCRW) for MTU5. TCR values should be specified only while TCNT operation is stopped.

#### TPSC[2:0] Bits (Time Prescaler Select)

These bits select the TCNT count clock source. The clock source can be selected independently for each channel. See **Table 16.7** to **Table 16.10** for details.

#### CKEG[1:0] Bits (Clock Edge Select)

These bits select the count clock source edge, including the MTIOC1A pin. When the internal clock is counted at both edges, the count clock period is halved (e.g. P0ϕ/4 at both edges = P0ϕ/2 at rising edge). If phase counting mode is used on MTU1 and MTU2, the setting of these bits is ignored and the phase counting mode setting has priority. Internal clock edge selection is valid when the count clock source is P0ϕ/2 or slower. When P0ϕ/1 or the overflow/underflow in another channel is selected for the count clock source, a value can be written to these bits but counter operation compiles with the initial value.

#### CCLR[2:0] Bits (Counter Clear Source Select)

These bits select the TCNT counter clearing source. See **Table 16.5** and **Table 16.6** for details.

Table 16.5 CCLR[2:0] (MTU0, MTU3, MTU4, MTU6, MTU7, and MTU8)

| Channel | Bit 7 CCLR2 | Bit 6 CCLR1 | Bit 5 CCLR0 | Description |
|---|---|---|---|---|
| MTU0 MTU3 MTU4 MTU6 MTU7 MTU8 | 0 | 0 | 0 | TCNT clearing disabled |
| | 0 | 0 | 1 | TCNT cleared by TGRA compare match/input capture |
| | 0 | 1 | 0 | TCNT cleared by TGRB compare match/input capture |
| | 0 | 1 | 1 | TCNT cleared by counter clearing in another channel where synchronous clearing or synchronous operation is selected*1 |
| | 1 | 0 | 0 | TCNT clearing disabled |
| | 1 | 0 | 1 | TCNT cleared by TGRC compare match/input capture*2 |
| | 1 | 1 | 0 | TCNT cleared by TGRD compare match/input capture*2 |
| | 1 | 1 | 1 | TCNT cleared by counter clearing in another channel where synchronous clearing or synchronous operation is selected*1 |

Note 1. Synchronous operation is selected by setting a SYNC bit in TSYRA or TSYRB to 1 except for MTU8.

Note 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 16.6 CCLR[2:0] (MTU1 and MTU2)

| Channel | Bit 7 Reserved*2 | Bit 6 CCLR1 | Bit 5 CCLR0 | Description |
|---|---|---|---|---|
| MTU1 MTU2 | 0 | 0 | 0 | TCNT clearing disabled |
| | 0 | 0 | 1 | TCNT cleared by TGRA compare match/input capture |
| | 0 | 1 | 0 | TCNT cleared by TGRB compare match/input capture |
| | 0 | 1 | 1 | TCNT cleared by counter clearing in another channel where synchronous clearing or synchronous operation is selected*1 |

Note 1. Synchronous operation is selected by setting a SYNC bit in TSYRA or TSYRB to 1.

Note 2. Bit 7 is reserved in MTU1 and MTU2. It is read as 0. The write value is ignored.

### (2) MTU5.TCRU, MTU5.TCRV, MTU5.TCRW

**Address(es):** MTU5.TCRU H'0_1000_1C84, MTU5.TCRV H'0_1000_1C94, MTU5.TCRW H'0_1000_1CA4

| Bit | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | TPSC[1:0] | |
| 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 |
|---|---|---|---|---|
| b1, b0 | TPSC[1:0] | All 0 | R/W | Time Prescaler Select<br>See **Table 16.11**. |
| b7 to b2 | — | All 0 | R/W | Reserved<br>These bits are read as 0. The write value should be 0. |

#### TPSC[1:0] Bits (Time Prescaler Select)

These bits select the TCNT count clock source. See **Table 16.11** for details.

## 16.2.2 Timer Control Register 2 (TCR2)

### (1) MTU0.TCR2, MTU3.TCR2, MTU4.TCR2, MTU6.TCR2, MTU7.TCR2, MTU8.TCR2

**Address(es):** MTU0.TCR2 H'0_1000_1328, MTU3.TCR2 H'0_1000_124C, MTU4.TCR2 H'0_1000_124D, MTU6.TCR2 H'0_1000_1A4C, MTU7.TCR2 H'0_1000_1A4D, MTU8.TCR2 H'0_1000_1606

| Bit | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | TPSC2[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 |

### (2) MTU1.TCR2, MTU2.TCR2

**Address(es):** MTU1.TCR2 H'0_1000_1394, MTU2.TCR2 H'0_1000_140C

| Bit | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | PCB[1:0] | | TPSC2[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 |
|---|---|---|---|---|
| b2 to b0 | TPSC2[2:0] | All 0 | R/W | Time Prescaler Select<br>See **Table 16.7** to **Table 16.10**. |
| b4, b3 | PCB[1:0] | All 0 | R/W | Phase Counting Mode Function Expansion Control<br>Functional Expansion Control for Phase Counting Modes 2, 3, and 5 |
| b7 to b5 | — | All 0 | R/W | Reserved<br>These bits are read as 0. The write value should be 0. |

The TCR2 register controls the TCNT operation for each channel in combination with the TCR register. The MTU has a total of 11 TCR2 registers, one each for MTU0 to MTU4, MTU 6, MTU7, and MTU8 and three (TCR2U, TCR2V, and TCR2W) for MTU5. TCR2 values should be specified only while TCNT operation is stopped.

#### TPSC2[2:0] Bits (Time Prescaler Select)

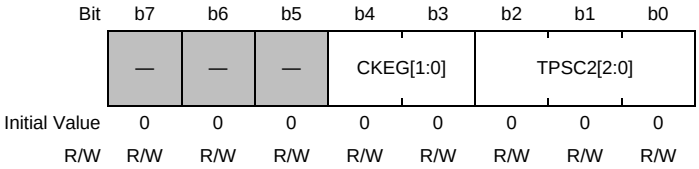
These bits select the TCNT count clock source. The clock source can be selected independently for each channel. See **Table 16.7** to **Table 16.10** for details.

#### PCB[1:0] Bits (Phase Counting Mode Function Expansion Control)

These bits control extended functions for phase counting mode 2, 3, and 5 in MTU1 and MTU2. See **Section 16.3.6, Phase Counting Mode**.

### (3) MTU5.TCR2U, MTU5.TCR2V, MTU5.TCR2W

**Address(es):** MTU5.TCR2U H'0_1000_1C85, MTU5.TCR2V H'0_1000_1C95, MTU5.TCR2W H'0_1000_1CA5




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| b2 to b0 | TPSC2[2:0] | All 0 | R/W | Time Prescaler Select<br>See **Table 16.11**. |
| b4, b3 | CKEG[1:0] | All 0 | R/W | Clock Edge Select<br>b4 b3<br>0 0: Counts at the rising edge.<br>0 1: Counts at the falling edge.<br>1 x: Counts at both edges. |
| b7 to b5 | — | All 0 | R/W | Reserved<br>These bits are read as 0. The write value should be 0. |

**Remarks:** x: Don't care

#### TPSC2[2:0] Bits (Time Prescaler Select)

These bits select the TCNT count clock source. See **Table 16.11** for details.

#### CKEG[1:0] Bits (Clock Edge Select)

These bits select the edge of the count clock source signal output from the MTIOC1A pin.

Table 16.7 TPSC[2:0], TPSC2[2:0] (MTU0)

| | TCR2[2:0] | | | TCR[2:0] | | | |
|---|---|---|---|---|---|---|---|
| | Bit 2 | Bit 1 | Bit 0 | Bit 2 | Bit 1 | Bit 0 | |
| Channel | TPSC22 | TPSC21 | TPSC20 | TPSC2 | TPSC1 | TPSC0 | Description |
| MTU0 | 0 | 0 | 0 | 0 | 0 | 0 | Internal clock: counts on P0ϕ/1 |
| | 0 | 0 | 0 | 0 | 0 | 1 | Internal clock: counts on P0ϕ/4 |
| | 0 | 0 | 0 | 0 | 1 | 0 | Internal clock: counts on P0ϕ/16 |
| | 0 | 0 | 0 | 0 | 1 | 1 | Internal clock: counts on P0ϕ/64 |
| | 0 | 0 | 0 | 1 | 0 | 0 | External clock: counts on MTCLKA pin input |
| | 0 | 0 | 0 | 1 | 0 | 1 | External clock: counts on MTCLKB pin input |
| | 0 | 0 | 0 | 1 | 1 | 0 | External clock: counts on MTCLKC pin input |
| | 0 | 0 | 0 | 1 | 1 | 1 | External clock: counts on MTCLKD pin input |
| | 0 | 0 | 1 | x | x | x | Internal clock: counts on P0ϕ/2 |
| | 0 | 1 | 0 | x | x | x | Internal clock: counts on P0ϕ/8 |
| | 0 | 1 | 1 | x | x | x | Internal clock: counts on P0ϕ/32 |
| | 1 | 0 | 0 | x | x | x | Internal clock: counts on P0ϕ/256 |
| | 1 | 0 | 1 | x | x | x | Internal clock: counts on P0ϕ/1024 |
| | 1 | 1 | 0 | x | x | x | Setting prohibited |
| | 1 | 1 | 1 | x | x | x | External clock: counts on MTIOC1A pin input |

**Remarks:** x: Don’t care

Table 16.8 TPSC[2:0], TPSC2[2:0] (MTU1)

| | TCR2[2:0] | | | TCR[2:0] | | | |
|---|---|---|---|---|---|---|---|
| | Bit 2 | Bit 1 | Bit 0 | Bit 2 | Bit 1 | Bit 0 | |
| Channel | TPSC22 | TPSC21 | TPSC20 | TPSC2 | TPSC1 | TPSC0 | Description |
| MTU1 | 0 | 0 | 0 | 0 | 0 | 0 | Internal clock: counts on P0ϕ/1 |
| | 0 | 0 | 0 | 0 | 0 | 1 | Internal clock: counts on P0ϕ/4 |
| | 0 | 0 | 0 | 0 | 1 | 0 | Internal clock: counts on P0ϕ/16 |
| | 0 | 0 | 0 | 0 | 1 | 1 | Internal clock: counts on P0ϕ/64 |
| | 0 | 0 | 0 | 1 | 0 | 0 | External clock: counts on MTCLKA pin input |
| | 0 | 0 | 0 | 1 | 0 | 1 | External clock: counts on MTCLKB pin input |
| | 0 | 0 | 0 | 1 | 1 | 0 | Internal clock: counts on P0ϕ/256 |
| | 0 | 0 | 0 | 1 | 1 | 1 | Overflow/underflow of MTU2.TCNT |
| | 0 | 0 | 1 | x | x | x | Internal clock: counts on P0ϕ/2 |
| | 0 | 1 | 0 | x | x | x | Internal clock: counts on P0ϕ/8 |
| | 0 | 1 | 1 | x | x | x | Internal clock: counts on P0ϕ/32 |
| | 1 | 0 | 0 | x | x | x | Internal clock: counts on P0ϕ/1024 |
| | 1 | 0 | 1 | x | x | x | Setting prohibited |
| | 1 | 1 | 0 | x | x | x | Setting prohibited |
| | 1 | 1 | 1 | x | x | x | Setting prohibited |

**Note:** This setting has no effect when MTU1 is in phase counting mode.

**Remarks:** x: Don’t care

Table 16.9 TPSC[2:0], TPSC2[2:0] (MTU2)

| | TCR2[2:0] | | | TCR[2:0] | | | |
|---|---|---|---|---|---|---|---|
| | Bit 2 | Bit 1 | Bit 0 | Bit 2 | Bit 1 | Bit 0 | |
| Channel | TPSC22 | TPSC21 | TPSC20 | TPSC2 | TPSC1 | TPSC0 | Description |
| MTU2 | 0 | 0 | 0 | 0 | 0 | 0 | Internal clock: counts on P0ϕ/1 |
| | 0 | 0 | 0 | 0 | 0 | 1 | Internal clock: counts on P0ϕ/4 |
| | 0 | 0 | 0 | 0 | 1 | 0 | Internal clock: counts on P0ϕ/16 |
| | 0 | 0 | 0 | 0 | 1 | 1 | Internal clock: counts on P0ϕ/64 |
| | 0 | 0 | 0 | 1 | 0 | 0 | External clock: counts on MTCLKA pin input |
| | 0 | 0 | 0 | 1 | 0 | 1 | External clock: counts on MTCLKB pin input |
| | 0 | 0 | 0 | 1 | 1 | 0 | External clock: counts on MTCLKC pin input |
| | 0 | 0 | 0 | 1 | 1 | 1 | Internal clock: counts on P0ϕ/1024 |
| | 0 | 0 | 1 | x | x | x | Internal clock: counts on P0ϕ/2 |
| | 0 | 1 | 0 | x | x | x | Internal clock: counts on P0ϕ/8 |
| | 0 | 1 | 1 | x | x | x | Internal clock: counts on P0ϕ/32 |
| | 1 | 0 | 0 | x | x | x | Internal clock: counts on P0ϕ/256 |
| | 1 | 0 | 1 | x | x | x | Setting prohibited |
| | 1 | 1 | 0 | x | x | x | Setting prohibited |
| | 1 | 1 | 1 | x | x | x | Setting prohibited |

**Note:** This setting has no effect when the MTU2 is in phase counting mode.

**Remarks:** x: Don't care

Table 16.10 TPSC[2:0], TPSC2[2:0] (MTU3, MTU4, MTU6, MTU7, and MTU8)

| | TCR2[2:0] | | | TCR[2:0] | | | |
|---|---|---|---|---|---|---|---|
| | Bit 2 | Bit 1 | Bit 0 | Bit 2 | Bit 1 | Bit 0 | |
| Channel | TPSC22 | TPSC21 | TPSC20 | TPSC2 | TPSC1 | TPSC0 | Description |
| MTU3<br>MTU4<br>MTU6<br>MTU7<br>MTU8 | 0 | 0 | 0 | 0 | 0 | 0 | Internal clock: counts on P0ϕ/1 |
| | 0 | 0 | 0 | 0 | 0 | 1 | Internal clock: counts on P0ϕ/4 |
| | 0 | 0 | 0 | 0 | 1 | 0 | Internal clock: counts on P0ϕ/16 |
| | 0 | 0 | 0 | 0 | 1 | 1 | Internal clock: counts on P0ϕ/64 |
| | 0 | 0 | 0 | 1 | 0 | 0 | Internal clock: counts on P0ϕ/256 |
| | 0 | 0 | 0 | 1 | 0 | 1 | Internal clock: counts on P0ϕ/1024 |
| | 0 | 0 | 0 | 1 | 1 | 0 | External clock: counts on MTCLKA pin input |
| | 0 | 0 | 0 | 1 | 1 | 1 | External clock: counts on MTCLKB pin input |
| | 0 | 0 | 1 | x | x | x | Internal clock: counts on P0ϕ/2 |
| | 0 | 1 | 0 | x | x | x | Internal clock: counts on P0ϕ/8 |
| | 0 | 1 | 1 | x | x | x | Internal clock: counts on P0ϕ/32 |
| | 1 | 0 | 0 | x | x | x | Setting prohibited |
| | 1 | 0 | 1 | x | x | x | Setting prohibited |
| | 1 | 1 | 0 | x | x | x | Setting prohibited |
| | 1 | 1 | 1 | x | x | x | Setting prohibited |

**Remarks:** x: Don't care

Table 16.11 TPSC[1:0], TPSC2[2:0] (MTU5)

| | TCR2[2:0] | | | TCR[1:0] | | |
|---|---|---|---|---|---|---|
| | Bit 2 | Bit 1 | Bit 0 | Bit 1 | Bit 0 | |
| Channel | TPSC22 | TPSC21 | TPSC20 | TPSC1 | TPSC0 | Description |
| MTU5 | 0 | 0 | 0 | 0 | 0 | Internal clock: counts on P0ϕ/1 |
| | 0 | 0 | 0 | 0 | 1 | Internal clock: counts on P0ϕ/4 |
| | 0 | 0 | 0 | 1 | 0 | Internal clock: counts on P0ϕ/16 |
| | 0 | 0 | 0 | 1 | 1 | Internal clock: counts on P0ϕ/64 |
| | 0 | 0 | 1 | x | x | Internal clock: counts on P0ϕ/2 |
| | 0 | 1 | 0 | x | x | Internal clock: counts on P0ϕ/8 |
| | 0 | 1 | 1 | x | x | Internal clock: counts on P0ϕ/32 |
| | 1 | 0 | 0 | x | x | Internal clock: counts on P0ϕ/256 |
| | 1 | 0 | 1 | x | x | Internal clock: counts on P0ϕ/1024 |
| | 1 | 1 | 0 | x | x | Setting prohibited |
| | 1 | 1 | 1 | x | x | Internal clock: counts on MTIOC1A pin input |

**Remarks:** x: Don’t care

## 16.2.3 Timer Mode Register 1 (TMDR1)

### (1) MTU0.TMDR1

**Address(es):** MTU0.TMDR1 H'0_1000_1301

| Bit | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | BFE | BFB | BFA | MD[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 |

### (2) MTU1.TMDR1, MTU2.TMDR1

**Address(es):** MTU1.TMDR1 H'0_1000_1381, MTU2.TMDR1 H'0_1000_1401

| Bit | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | MD[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 |

### (3) MTU3.TMDR1, MTU4.TMDR1, MTU6.TMDR1, MTU7.TMDR1, MTU8.TMDR1

**Address(es):** MTU3.TMDR1 H'0_1000_1202, MTU4.TMDR1 H'0_1000_1203, MTU6.TMDR1 H'0_1000_1A02, MTU7.TMDR1 H'0_1000_1A03, MTU8.TMDR1 H'0_1000_1601

| Bit | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | BFB | BFA | MD[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 |
|---|---|---|---|---|
| b3 to b0 | MD[3:0] | All 0 | R/W | Mode Select<br>These bits specify the timer operating mode. See **Table 16.12** for details. |
| b4 | BFA | 0 | R/W | Buffer Operation A<br>0: TGRA and TGRC operate normally<br>1: TGRA and TGRC used together for buffer operation |
| b5 | BFB | 0 | R/W | Buffer Operation B<br>0: TGRB and TGRD operate normally<br>1: TGRB and TGRD used together for buffer operation |
| b6 | BFE | 0 | R/W | Buffer Operation E<br>0: MTU0.TGRE and MTU0.TGRF operate normally<br>1: MTU0.TGRE and MTU0.TGRF used together for buffer operation |
| b7 | — | 0 | R/W | Reserved<br>This bit is read as 0. The write value should be 0. |

The TMDR1 register specifies the operating mode of each channel. The MTU has a total of eight TMDR1 registers, one each for MTU0 to MTU4, MTU 6, MTU7, and MTU8. TMDR1 register values should be specified only while TCNT operation is stopped.

Table 16.12 Operating Mode Setting by MD[3:0] Bits (MTU0 to MTU4, MTU6 to MTU8)

| Bit 3 | Bit 2 | Bit 1 | Bit 0 | | MTU0 | MTU1 | MTU2 | MTU1 & MTU2 (LWA = 1) | MTU3 | MTU4 | MTU6 | MTU7 | MTU8 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MD3 | MD2 | MD1 | MD0 | Description | | | | | | | | | |
| 0 | 0 | 0 | 0 | Normal mode | ✓ | ✓ | ✓ | | ✓ | ✓ | ✓ | ✓ | ✓ |
| 0 | 0 | 0 | 1 | Setting prohibited | | | | | | | | | |
| 0 | 0 | 1 | 0 | PWM mode 1 | ✓ | ✓ | ✓ | | ✓ | ✓ | ✓ | ✓ | |
| 0 | 0 | 1 | 1 | PWM mode 2 | ✓ | ✓ | ✓ | | | | | | |
| 0 | 1 | 0 | 0 | Phase counting mode 1 | | ✓ | ✓ | ✓ | | | | | |
| 0 | 1 | 0 | 1 | Phase counting mode 2 | | ✓ | ✓ | ✓ | | | | | |
| 0 | 1 | 1 | 0 | Phase counting mode 3 | | ✓ | ✓ | ✓ | | | | | |
| 0 | 1 | 1 | 1 | Phase counting mode 4 | | ✓ | ✓ | ✓ | | | | | |
| 1 | 0 | 0 | 0 | Reset-synchronized PWM mode*1 | | | | | ✓ | | ✓ | | |
| 1 | 0 | 0 | 1 | Phase counting mode 5 | | ✓ | ✓ | ✓ | | | | | |
| 1 | 0 | 1 | x | Setting prohibited | | | | | | | | | |
| 1 | 1 | 0 | 0 | Setting prohibited | | | | | | | | | |
| 1 | 1 | 0 | 1 | Complementary PWM mode 1 (transfer at crest)*1 | | | | | ✓ | | ✓ | | |
| 1 | 1 | 1 | 0 | Complementary PWM mode 2 (transfer at trough)*1 | | | | | ✓ | | ✓ | | |
| 1 | 1 | 1 | 1 | Complementary PWM mode 3 (transfer at crest and trough)*1 | | | | | ✓ | | ✓ | | |

**Note:** Only set the corresponding operating mode listed above for each channel.

**Remarks:** x: Don't care

Note 1. Reset-synchronized PWM mode and complementary PWM mode can only be set for MTU3 and MTU6. When MTU3 or MTU6 is set to reset-synchronized PWM mode or complementary PWM mode, the MTU4 or MTU7 settings become ineffective and automatically conform to the MTU3 or MTU6 setting, respectively. MTU4 and MTU7 should be set to the initial values (normal mode).

### BFA Bit (Buffer Operation A)

This bit specifies whether to operate TGRA in the normal way or to use TGRA and TGRC together for buffer operation. When TGRC is used as a buffer register, TGRC input capture/output compare does not take place in modes other than complementary PWM mode, but compare match with TGRC occurs in complementary PWM mode.

If a compare match occurs on MTU4 in the Tb interval in complementary PWM mode, the TGIEC bit in timer interrupt enable register (MTU4.TIER) should be set to 0.

In reset-synchronized PWM mode or complementary PWM mode, buffer operation in MTU3 and MTU4 (or MTU6 and MTU7) depends on the settings in the BFA bit of MTU3.TMDR1 (MTU6.TMDR1). The BFA bit of MTU4.TMDR1 (MTU7.TMDR1) should be set to 0.

In MTU1 and MTU2, which have no TGRC, this bit is reserved. It is read as 0. The write value should be 0. See **Figure 16.50** for an illustration of the Tb interval in complementary PWM mode.

#### BFB Bit (Buffer Operation B)

This bit specifies whether to operate TGRB in the normal way or to use TGRB and TGRD together for buffer operation. When TGRD is used as a buffer register, TGRD input capture/output compare does not take place in modes other than complementary PWM mode, but compare match with TGRD occurs in complementary PWM mode.

If a compare match occurs on MTU4 in the Tb interval in complementary PWM mode, the TGIED bit in timer interrupt enable register (MTU4.TIER) should be set to 0.

In reset-synchronized PWM mode or complementary PWM mode, buffer operation in MTU3 and MTU4 (or MTU6 and MTU7) depends on the settings in the BFB bit of MTU3.TMDR1 (MTU6.TMDR1). The BFB bit of MTU4.TMDR1 (MTU7.TMDR1) should be set to 0.

In MTU1 and MTU2, which have no TGRD, this bit is reserved. It is read as 0. The write value should be 0. See **Figure 16.50** for an illustration of the Tb interval in complementary PWM mode.

#### BFE Bit (Buffer Operation E)

This bit specifies whether to operate MTU0.TGRE and MTU0.TGRF in the normal way or to use them together for buffer operation. Compare match with TGRF occurs even when TGRF is used as a buffer register.

In MTU0 to MTU4, MTU6, MTU7, and MTU8, this bit is reserved. It is read as 0. The write value should be 0.

## 16.2.4 Timer Mode Registers 2 (TMDR2A and TMDR2B)

**Address(es):** MTU.TMDR2A H'0_1000_1270, MTU.TMDR2B H'0_1000_1A70

| Bit | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | DRS |
| 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 |
|---|---|---|---|---|
| b0 | DRS | 0 | R/W | Double Buffer Select<br>0: Double buffer function is disabled<br>1: Double buffer function is enabled |
| b7 to b1 | — | All 0 | R/W | Reserved<br>These bits are read as 0. The write value should be 0. |

TMDR2A and TMDR2B specify the double buffer function in complementary PWM mode 3 (transfer at the crest and trough of the counter value). The MTU has two TMDR2 registers, one each for MTU3 (TMDR2A) and MTU6 (TMDR2B). TMDR2A and TMDR2B values should be specified only while TCNT operation is stopped.

### DRS Bit (Double Buffer Select)

This bit enables or disables the double buffer function in complementary PWM mode.

## 16.2.5 Timer Mode Register 3 (TMDR3)

**Address(es):** MTU1.TMDR3 H'0_1000_1391

| Bit | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | PHCKSEL | LWA |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 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 |
|---|---|---|---|---|
| b0 | LWA | 0 | R/W | MTU1/MTU2 Combination Longword Access Control<br>0: 16-bit access is enabled.<br>1: 32-bit access is enabled. |
| b1 | PHCKSEL | 1 | R/W | External Input Phase Clock Select<br>Selects the external clock pin for phase counting mode.<br>0: MTCLKA and MTCLKB are selected for the external phase clock.<br>1: MTCLKC and MTCLKD are selected for the external phase clock. |
| b7 to b2 | — | All 0 | R/W | Reserved<br>These bits are read as 0. The write value should be 0. |

The TMDR3 register controls longword access to a 32-bit register or counter in a combination of MTU1 and MTU2. There is only one TMDR3 register in MTU1. The counter (TCNTLW), general register A (TGRALW), and general register B (TGRBLW) of MTU1 and MTU2 are accessed in the combinations listed in **Table 16.13**.

### LWA Bit (MTU1/MTU2 Combination Longword Access Control)

This bit selects a 32-bit access in a combination of MTU1 and MTU2.

When the LWA bit is set to 0, MTU1 and MTU2 operate independently as 16-bit timers and the TCNTLW, TGRALW, and TGRBLW registers cannot be accessed.

When the LWA bit is set to 1, MTU1 and MTU2 are cascaded and operate as a 32-bit timer, which is controlled through the TCR, TCR2, TIOR, and TMDR1 registers in MTU1. The TCR, TCR2, TIOR, and TMDR1 register settings in MTU2 are ignored. The 16-bit registers (TCNT, TGRA, and TGRB) in MTU1 and MTU2 cannot be accessed. MTU2 input capture and compare match are disabled.

Note that MTU1 and MTU2 can be cascaded by setting the LWA bit to 1 only in phase-counting mode. Cascade connection cannot be used in normal, PWM1, or PWM2 mode. When setting the LWA bit to 1, be sure to select phase-counting mode.

In addition, initialize the TCNT, TGRA, and TGRB registers in MTU1 and MTU2 before setting the LWA bit to 1.

### PHCKSEL Bit (External Input Phase Clock Select)

When the MTU1 and MTU2 registers are combined for 32-bit phase counting mode or MTU2 phase counting mode, this bit selects either the A- or B-phase signal from the external clock. See **Table 16.66, Clock Input Pins in Phase Counting Mode** for details.

Table 16.13 Setting and Combination of the TMDR3 Register

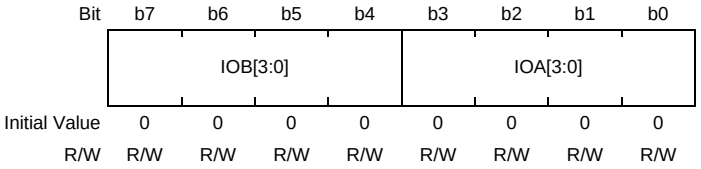
| | TMDR3.LWA = 0 | | TMDR3.LWA = 1 | |
|---|---|---|---|---|
| Register | Symbol | Access mode | Symbol | Access mode |
| Counter in MTU1*1 | MTU1.TCNT | Word | MTU1.TCNT_1_LW | Longword |
| Counter in MTU2 | MTU2.TCNT | Word | | |
| General register A in MTU1 | MTU1.TGRA | Word | MTU1.TGRA_1_LW | Longword |
| General register A in MTU2 | MTU2.TGRA | Word | | |
| General register B in MTU1 | MTU1.TGRB | Word | MTU1.TGRB_1_LW | Longword |
| General register B in MTU2 | MTU2.TGRB | Word | | |

Note 1. When the LWA bit is set to 1, setting the count clock for MTU1 as MTU2.TCNT overflow/underflow is not required.

## 16.2.6 Timer I/O Control Register (TIOR)

### (1) MTU0.TIORH, MTU1.TIOR, MTU2.TIOR, MTU3.TIORH, MTU4.TIORH, MTU6.TIORH, MTU7.TIORH, MTU8.TIORH

**Address(es):** MTU0.TIORH H'0_1000_1302, MTU1.TIOR H'0_1000_1382, MTU2.TIOR H'0_1000_1402, MTU3.TIORH H'0_1000_1204, MTU4.TIORH H'0_1000_1206, MTU6.TIORH H'0_1000_1A04, MTU7.TIORH H'0_1000_1A06, MTU8.TIORH H'0_1000_1602

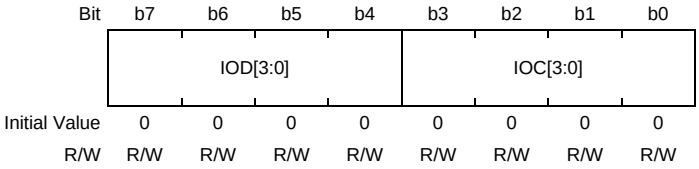



| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| b3 to b0 | IOA[3:0] | All 0 | R/W | I/O Control A*1<br>See the following tables.<br>MTU0.TIORH: **Table 16.28**<br>MTU1.TIOR: **Table 16.30**<br>MTU2.TIOR: **Table 16.31**<br>MTU3.TIORH: **Table 16.32**<br>MTU4.TIORH: **Table 16.34**<br>MTU6.TIORH: **Table 16.36**<br>MTU7.TIORH: **Table 16.38**<br>MTU8.TIORH: **Table 16.40** |
| b7 to b4 | IOB[3:0] | All 0 | R/W | I/O Control B*1<br>See the following tables.<br>MTU0.TIORH: **Table 16.14**<br>MTU1.TIOR: **Table 16.16**<br>MTU2.TIOR: **Table 16.17**<br>MTU3.TIORH: **Table 16.18**<br>MTU4.TIORH: **Table 16.20**<br>MTU6.TIORH: **Table 16.22**<br>MTU7.TIORH: **Table 16.24**<br>MTU8.TIORH: **Table 16.26** |

Note 1. When the value of IOn[3:0] (n = A, B) is changed to the output-prohibited state (0000b or 0100b) during low, high, or toggle output at compare match, this register is in Hi-Z.

### (2) MTU0.TIORL, MTU3.TIORL, MTU4.TIORL, MTU6.TIORL, MTU7.TIORL, MTU8.TIORL

**Address(es):** MTU0.TIORL H'0_1000_1303, MTU3.TIORL H'0_1000_1205, MTU4.TIORL H'0_1000_1207, MTU6.TIORL H'0_1000_1A05,MTU7.TIORL H'0_1000_1A07, MTU8.TIORL H'0_1000_1603

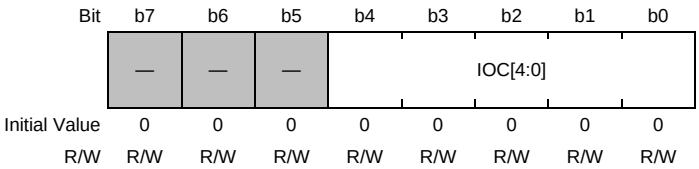



| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| b3 to b0 | IOC[3:0] | All 0 | R/W | I/O Control C*1<br>See the following tables.<br>MTU0.TIORL: **Table 16.29**<br>MTU3.TIORL: **Table 16.33**<br>MTU4.TIORL: **Table 16.35**<br>MTU6.TIORL: **Table 16.37**<br>MTU7.TIORL: **Table 16.39**<br>MTU8.TIORL: **Table 16.41** |
| b7 to b4 | IOD[3:0] | All 0 | R/W | I/O Control D*1<br>See the following tables.<br>MTU0.TIORL: **Table 16.15**<br>MTU3.TIORL: **Table 16.19**<br>MTU4.TIORL: **Table 16.21**<br>MTU6.TIORL: **Table 16.23**<br>MTU7.TIORL: **Table 16.25**<br>MTU8.TIORL: **Table 16.27** |

Note 1. When the value of IOn[3:0] (n = C, D) is changed to the output-prohibited state (0000b or 0100b) during low, high, or toggle output at compare match, this register is in Hi-Z.

### (3) MTU5.TIORU, MTU5.TIORV, MTU5.TIORW

**Address(es):** MTU5.TIORU H'0_1000_1C86, MTU5.TIORV H'0_1000_1C96, MTU5.TIORW H'0_1000_1CA6




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| b4 to b0 | IOC[4:0] | All 0 | R/W | I/O Control C<br>See the following table.<br>MTU5.TIORU, MTU5.TIORV, MTU5.TIORW: **Table 16.42** |
| b7 to b5 | — | All 0 | R/W | Reserved<br>These bits are read as 0. The write value should be 0. |

The TIOR register controls the TGR register. The MTU has a total of 17 TIOR registers, two each for MTU0, MTU3, MTU4, MTU6, MTU7, and MTU8, one each for MTU1 and MTU2, and three (MTU5.TIORU/TIORV/TIORW) for MTU5. The TIOR register should be set when the TMDR register setting is normal mode, PWM mode, or phase counting mode.

Note that TIOR is affected by the TMDR1 setting.

The initial output specified by TIOR is valid when the counter is stopped (the CSTn bits in TSTRA and TSTRB are 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.

Table 16.14 TIORH (MTU0)

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Description | |
|---|---|---|---|---|---|
| IOB3 | IOB2 | IOB1 | IOB0 | MTU0.TGRB Function | MTIOC0B Pin Function |
| 0 | 0 | 0 | 0 | Output compare register | Output prohibited |
| 0 | 0 | 0 | 1 | | Initial output is low.<br>Low output at compare match. |
| 0 | 0 | 1 | 0 | | Initial output is low.<br>High output at compare match. |
| 0 | 0 | 1 | 1 | | Initial output is low.<br>Toggle output at compare match. |
| 0 | 1 | 0 | 0 | | Output prohibited |
| 0 | 1 | 0 | 1 | | Initial output is high.<br>Low output at compare match. |
| 0 | 1 | 1 | 0 | | Initial output is high.<br>High output at compare match. |
| 0 | 1 | 1 | 1 | | Initial output is high.<br>Toggle output at compare match. |
| 1 | 0 | 0 | 0 | Input capture register | Input capture at rising edge. |
| 1 | 0 | 0 | 1 | | Input capture at falling edge. |
| 1 | 0 | 1 | x | | Input capture at both edges. |
| 1 | 1 | x | x | | Capture input source is the clock source for counting in MTU1.<br>Input capture on counting up or down by TCNT (LWA = 0) or TCNTLW (LWA = 1) in MTU1*1 |

**Remarks:** Don’t care

Note 1. Input capture will not be generated in MTU0 if P0ϕ/1 is selected as the count clock for MTU1. Select a clock other than P0ϕ/1.

Table 16.15 TIORL (MTU0)

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Description | |
|---|---|---|---|---|---|
| IOD3 | IOD2 | IOD1 | IOD0 | MTU0.TGRD Function | MTIOC0D Pin Function |
| 0 | 0 | 0 | 0 | Output compare register*1 | Output prohibited |
| 0 | 0 | 0 | 1 | | Initial output is low.<br>Low output at compare match. |
| 0 | 0 | 1 | 0 | | Initial output is low.<br>High output at compare match. |
| 0 | 0 | 1 | 1 | | Initial output is low.<br>Toggle output at compare match. |
| 0 | 1 | 0 | 0 | | Output prohibited |
| 0 | 1 | 0 | 1 | | Initial output is high.<br>Low output at compare match. |
| 0 | 1 | 1 | 0 | | Initial output is high.<br>High output at compare match. |
| 0 | 1 | 1 | 1 | | Initial output is high.<br>Toggle output at compare match. |
| 1 | 0 | 0 | 0 | Input capture register*1 | Input capture at rising edge. |
| 1 | 0 | 0 | 1 | | Input capture at falling edge. |
| 1 | 0 | 1 | x | | Input capture at both edges. |
| 1 | 1 | x | x | | Capture input source is the clock source for counting in MTU1.<br>Input capture on counting up or down by TCNT (LWA = 0) or TCNTLW (LWA = 1) in MTU1*2 |

**Remarks:** Don’t care

Note 1. When the MTU0.TMDR1.BFB is set to 1 and MTU0.TGRD is used as a buffer register, this setting is invalid and input capture/ output compare is not generated.

Note 2. Input capture will not be generated in MTU0 if P0ϕ/1 is selected as the count clock for MTU1. Select a clock other than P0ϕ/1.

Table 16.16 TIOR (MTU1)

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Description | |
|---|---|---|---|---|---|
| IOB3 | IOB2 | IOB1 | IOB0 | MTU1.TGRB/TGRBLW Function | MTIOC1B Pin Function |
| 0 | 0 | 0 | 0 | Output compare register | Output prohibited |
| 0 | 0 | 0 | 1 | | Initial output is low.<br>Low output at compare match. |
| 0 | 0 | 1 | 0 | | Initial output is low.<br>High output at compare match. |
| 0 | 0 | 1 | 1 | | Initial output is low.<br>Toggle output at compare match. |
| 0 | 1 | 0 | 0 | | Output prohibited |
| 0 | 1 | 0 | 1 | | Initial output is high.<br>Low output at compare match. |
| 0 | 1 | 1 | 0 | | Initial output is high.<br>High output at compare match. |
| 0 | 1 | 1 | 1 | | Initial output is high.<br>Toggle output at compare match. |
| 1 | 0 | 0 | 0 | Input capture register | Input capture at rising edge. |
| 1 | 0 | 0 | 1 | | Input capture at falling edge. |
| 1 | 0 | 1 | x | | Input capture at both edges. |
| 1 | 1 | 0 | 0 | | Input capture at occurrence of compare match or input capture in the MTU0.TGRC register |
| 1 | 1 | 1 | x | | Input capture at occurrence of compare match in the MTU8.TGRC register |

**Remarks:** x: Don’t care

Table 16.17 TIOR (MTU2)

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Description | |
|---|---|---|---|---|---|
| IOB3 | IOB2 | IOB1 | IOB0 | MTU2.TGRB Function | MTIOC2B Pin Function |
| 0 | 0 | 0 | 0 | Output compare register | Output prohibited |
| 0 | 0 | 0 | 1 | | Initial output is low.<br>Low output at compare match. |
| 0 | 0 | 1 | 0 | | Initial output is low.<br>High output at compare match. |
| 0 | 0 | 1 | 1 | | Initial output is low.<br>Toggle output at compare match. |
| 0 | 1 | 0 | 0 | | Output prohibited |
| 0 | 1 | 0 | 1 | | Initial output is high.<br>Low output at compare match. |
| 0 | 1 | 1 | 0 | | Initial output is high.<br>High output at compare match. |
| 0 | 1 | 1 | 1 | | Initial output is high.<br>Toggle output at compare match. |
| 1 | x | 0 | 0 | Input capture register | Input capture at rising edge. |
| 1 | x | 0 | 1 | | Input capture at falling edge. |
| 1 | x | 1 | x | | Input capture at both edges. |

**Remarks:** x: Don’t care

Table 16.18 TIORH (MTU3)

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Description | |
|---|---|---|---|---|---|
| IOB3 | IOB2 | IOB1 | IOB0 | MTU3.TGRB Function | MTIOC3B Pin Function |
| 0 | 0 | 0 | 0 | Output compare register | Output prohibited |
| 0 | 0 | 0 | 1 | | Initial output is low.<br>Low output at compare match. |
| 0 | 0 | 1 | 0 | | Initial output is low.<br>High output at compare match. |
| 0 | 0 | 1 | 1 | | Initial output is low.<br>Toggle output at compare match. |
| 0 | 1 | 0 | 0 | | Output prohibited |
| 0 | 1 | 0 | 1 | | Initial output is high.<br>Low output at compare match. |
| 0 | 1 | 1 | 0 | | Initial output is high.<br>High output at compare match. |
| 0 | 1 | 1 | 1 | | Initial output is high.<br>Toggle output at compare match. |
| 1 | x | 0 | 0 | Input capture register | Input capture at rising edge. |
| 1 | x | 0 | 1 | | Input capture at falling edge. |
| 1 | x | 1 | x | | Input capture at both edges. |

**Remarks:** x: Don't care

Table 16.19 TIORL (MTU3)

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Description | |
|---|---|---|---|---|---|
| IOD3 | IOD2 | IOD1 | IOD0 | MTU3.TGRD Function | MTIOC3D Pin Function |
| 0 | 0 | 0 | 0 | Output compare register*1 | Output prohibited |
| 0 | 0 | 0 | 1 | | Initial output is low.<br>Low output at compare match. |
| 0 | 0 | 1 | 0 | | Initial output is low.<br>High output at compare match. |
| 0 | 0 | 1 | 1 | | Initial output is low.<br>Toggle output at compare match. |
| 0 | 1 | 0 | 0 | | Output prohibited |
| 0 | 1 | 0 | 1 | | Initial output is high.<br>Low output at compare match. |
| 0 | 1 | 1 | 0 | | Initial output is high.<br>High output at compare match. |
| 0 | 1 | 1 | 1 | | Initial output is high.<br>Toggle output at compare match. |
| 1 | x | 0 | 0 | Input capture register*1 | Input capture at rising edge. |
| 1 | x | 0 | 1 | | Input capture at falling edge. |
| 1 | x | 1 | x | | Input capture at both edges. |

**Remarks:** x: Don't care

Note 1. When the BFB bit in MTU3.TMDR1 is set to 1 and MTU3.TGRD is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 16.20 TIORH (MTU4)

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Description | |
|---|---|---|---|---|---|
| IOB3 | IOB2 | IOB1 | IOB0 | MTU4.TGRB Function | MTIOC4B Pin Function |
| 0 | 0 | 0 | 0 | Output compare register | Output prohibited |
| 0 | 0 | 0 | 1 | | Initial output is low.<br>Low output at compare match. |
| 0 | 0 | 1 | 0 | | Initial output is low.<br>High output at compare match. |
| 0 | 0 | 1 | 1 | | Initial output is low.<br>Toggle output at compare match. |
| 0 | 1 | 0 | 0 | | Output prohibited |
| 0 | 1 | 0 | 1 | | Initial output is high.<br>Low output at compare match. |
| 0 | 1 | 1 | 0 | | Initial output is high.<br>High output at compare match. |
| 0 | 1 | 1 | 1 | | Initial output is high.<br>Toggle output at compare match. |
| 1 | x | 0 | 0 | Input capture register | Input capture at rising edge. |
| 1 | x | 0 | 1 | | Input capture at falling edge. |
| 1 | x | 1 | x | | Input capture at both edges. |

**Remarks:** x: Don't care

Table 16.21 TIORL (MTU4)

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Description | |
|---|---|---|---|---|---|
| IOD3 | IOD2 | IOD1 | IOD0 | MTU4.TGRD Function | MTIOC4D Pin Function |
| 0 | 0 | 0 | 0 | Output compare register*1 | Output prohibited |
| 0 | 0 | 0 | 1 | | Initial output is low.<br>Low output at compare match. |
| 0 | 0 | 1 | 0 | | Initial output is low.<br>High output at compare match. |
| 0 | 0 | 1 | 1 | | Initial output is low.<br>Toggle output at compare match. |
| 0 | 1 | 0 | 0 | | Output prohibited |
| 0 | 1 | 0 | 1 | | Initial output is high.<br>Low output at compare match. |
| 0 | 1 | 1 | 0 | | Initial output is high.<br>High output at compare match. |
| 0 | 1 | 1 | 1 | | Initial output is high.<br>Toggle output at compare match. |
| 1 | x | 0 | 0 | Input capture register*1 | Input capture at rising edge. |
| 1 | x | 0 | 1 | | Input capture at falling edge. |
| 1 | x | 1 | x | | Input capture at both edges. |

**Remarks:** x: Don't care

Note 1. When the BFB bit in MTU4.TMDR1 is set to 1 and MTU4.TGRD is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 16.22 TIORH (MTU6)

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Description | |
|---|---|---|---|---|---|
| IOB3 | IOB2 | IOB1 | IOB0 | MTU6.TGRB Function | MTIOC6B Pin Function |
| 0 | 0 | 0 | 0 | Output compare register | Output prohibited |
| 0 | 0 | 0 | 1 | | Initial output is low.<br>Low output at compare match. |
| 0 | 0 | 1 | 0 | | Initial output is low.<br>High output at compare match. |
| 0 | 0 | 1 | 1 | | Initial output is low.<br>Toggle output at compare match. |
| 0 | 1 | 0 | 0 | | Output prohibited |
| 0 | 1 | 0 | 1 | | Initial output is high.<br>Low output at compare match. |
| 0 | 1 | 1 | 0 | | Initial output is high.<br>High output at compare match. |
| 0 | 1 | 1 | 1 | | Initial output is high.<br>Toggle output at compare match. |
| 1 | x | 0 | 0 | Input capture register | Input capture at rising edge. |
| 1 | x | 0 | 1 | | Input capture at falling edge. |
| 1 | x | 1 | x | | Input capture at both edges. |

**Remarks:** x: Don't care

Table 16.23 TIORL (MTU6)

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Description | |
|---|---|---|---|---|---|
| IOD3 | IOD2 | IOD1 | IOD0 | MTU6.TGRD Function | MTIOC6D Pin Function |
| 0 | 0 | 0 | 0 | Output compare register*1 | Output prohibited |
| 0 | 0 | 0 | 1 | | Initial output is low.<br>Low output at compare match. |
| 0 | 0 | 1 | 0 | | Initial output is low.<br>High output at compare match. |
| 0 | 0 | 1 | 1 | | Initial output is low.<br>Toggle output at compare match. |
| 0 | 1 | 0 | 0 | | Output prohibited |
| 0 | 1 | 0 | 1 | | Initial output is high.<br>Low output at compare match. |
| 0 | 1 | 1 | 0 | | Initial output is high.<br>High output at compare match. |
| 0 | 1 | 1 | 1 | | Initial output is high.<br>Toggle output at compare match. |
| 1 | x | 0 | 0 | Input capture register*1 | Input capture at rising edge. |
| 1 | x | 0 | 1 | | Input capture at falling edge. |
| 1 | x | 1 | x | | Input capture at both edges. |

**Remarks:** x: Don't care

Note 1. When the BFB bit in MTU6.TMDR1 is set to 1 and MTU6.TGRD register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 16.24 TIORH (MTU7)

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Description | |
|---|---|---|---|---|---|
| IOB3 | IOB2 | IOB1 | IOB0 | MTU7.TGRB Function | MTIOC7B Pin Function |
| 0 | 0 | 0 | 0 | Output compare register | Output prohibited |
| 0 | 0 | 0 | 1 | | Initial output is low.<br>Low output at compare match. |
| 0 | 0 | 1 | 0 | | Initial output is low.<br>High output at compare match. |
| 0 | 0 | 1 | 1 | | Initial output is low.<br>Toggle output at compare match. |
| 0 | 1 | 0 | 0 | | Output prohibited |
| 0 | 1 | 0 | 1 | | Initial output is high.<br>Low output at compare match. |
| 0 | 1 | 1 | 0 | | Initial output is high.<br>High output at compare match. |
| 0 | 1 | 1 | 1 | | Initial output is high.<br>Toggle output at compare match. |
| 1 | x | 0 | 0 | Input capture register | Input capture at rising edge. |
| 1 | x | 0 | 1 | | Input capture at falling edge. |
| 1 | x | 1 | x | | Input capture at both edges. |

**Remarks:** x: Don't care

Table 16.25 TIORL (MTU7)

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Description | |
|---|---|---|---|---|---|
| IOD3 | IOD2 | IOD1 | IOD0 | MTU7.TGRD Function | MTIOC7D Pin Function |
| 0 | 0 | 0 | 0 | Output compare register*1 | Output prohibited |
| 0 | 0 | 0 | 1 | | Initial output is low.<br>Low output at compare match. |
| 0 | 0 | 1 | 0 | | Initial output is low.<br>High output at compare match. |
| 0 | 0 | 1 | 1 | | Initial output is low.<br>Toggle output at compare match. |
| 0 | 1 | 0 | 0 | | Output prohibited |
| 0 | 1 | 0 | 1 | | Initial output is high.<br>Low output at compare match. |
| 0 | 1 | 1 | 0 | | Initial output is high.<br>High output at compare match. |
| 0 | 1 | 1 | 1 | | Initial output is high.<br>Toggle output at compare match. |
| 1 | x | 0 | 0 | Input capture register*1 | Input capture at rising edge. |
| 1 | x | 0 | 1 | | Input capture at falling edge. |
| 1 | x | 1 | x | | Input capture at both edges. |

**Remarks:** x: Don't care

Note 1. When the BFB bit in MTU7.TMDR1 is set to 1 and MTU7.TGRD register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 16.26 TIORH (MTU8)

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Description | |
|---|---|---|---|---|---|
| IOB3 | IOB2 | IOB1 | IOB0 | MTU8.TGRB Function | MTIOC8B Pin Function |
| 0 | 0 | 0 | 0 | Output compare register | Output prohibited |
| 0 | 0 | 0 | 1 | | Initial output is low.<br>Low output at compare match. |
| 0 | 0 | 1 | 0 | | Initial output is low.<br>High output at compare match. |
| 0 | 0 | 1 | 1 | | Initial output is low.<br>Toggle output at compare match. |
| 0 | 1 | 0 | 0 | | Output prohibited |
| 0 | 1 | 0 | 1 | | Initial output is high.<br>Low output at compare match. |
| 0 | 1 | 1 | 0 | | Initial output is high.<br>High output at compare match. |
| 0 | 1 | 1 | 1 | | Initial output is high.<br>Toggle output at compare match. |
| 1 | 0 | 0 | 0 | Input capture register | Input capture at rising edge. |
| 1 | 0 | 0 | 1 | | Input capture at falling edge. |
| 1 | 0 | 1 | x | | Input capture at both edges. |
| 1 | 1 | x | x | | Capture input source is the clock source for counting in MTU1.<br>Input capture on counting up or down by MTU1.TCNT (LWA = 0) or MTU1.TCNTLW (LWA = 1)*1 |

**Remarks:** x: Don’t care

Note 1. Input capture will not be generated in MTU8 if P0ϕ/1 is selected as the count clock for MTU1. Select a clock other than P0ϕ/1.

Table 16.27 TIORL (MTU8)

| Bit 7 | Bit 6 | Bit 5 | Bit 4 | Description | |
|---|---|---|---|---|---|
| IOD3 | IOD2 | IOD1 | IOD0 | MTU8.TGRD Function | MTIOC8D Pin Function |
| 0 | 0 | 0 | 0 | Output compare register*1 | Output prohibited |
| 0 | 0 | 0 | 1 | | Initial output is low.<br>Low output at compare match. |
| 0 | 0 | 1 | 0 | | Initial output is low.<br>High output at compare match. |
| 0 | 0 | 1 | 1 | | Initial output is low.<br>Toggle output at compare match. |
| 0 | 1 | 0 | 0 | | Output prohibited |
| 0 | 1 | 0 | 1 | | Initial output is high.<br>Low output at compare match. |
| 0 | 1 | 1 | 0 | | Initial output is high.<br>High output at compare match. |
| 0 | 1 | 1 | 1 | | Initial output is high.<br>Toggle output at compare match. |
| 1 | x | 0 | 0 | Input capture register | Input capture at rising edge. |
| 1 | x | 0 | 1 | | Input capture at falling edge. |
| 1 | x | 1 | x | | Input capture at both edges. |

**Remarks:** x: Don’t care

Note 1. When the BFB bit of the TMDR1 register is set to 1 and the TGRD register is used as a buffer register in MTU8, this setting is invalid and input capture/output compare is not generated.

Table 16.28 TIORH (MTU0)

| Bit 3 | Bit 2 | Bit 1 | Bit 0 | Description | |
|---|---|---|---|---|---|
| IOA3 | IOA2 | IOA1 | IOA0 | MTU0.TGRA Function | MTIOC0A Pin Function |
| 0 | 0 | 0 | 0 | Output compare register | Output prohibited |
| 0 | 0 | 0 | 1 | | Initial output is low.<br>Low output at compare match. |
| 0 | 0 | 1 | 0 | | Initial output is low.<br>High output at compare match. |
| 0 | 0 | 1 | 1 | | Initial output is low.<br>Toggle output at compare match. |
| 0 | 1 | 0 | 0 | | Output prohibited |
| 0 | 1 | 0 | 1 | | Initial output is high.<br>Low output at compare match. |
| 0 | 1 | 1 | 0 | | Initial output is high.<br>High output at compare match. |
| 0 | 1 | 1 | 1 | | Initial output is high.<br>Toggle output at compare match. |
| 1 | 0 | 0 | 0 | Input capture register | Input capture at rising edge. |
| 1 | 0 | 0 | 1 | | Input capture at falling edge. |
| 1 | 0 | 1 | x | | Input capture at both edges. |
| 1 | 1 | 0 | 0 | | Capture input source is the clock source for counting in MTU1.<br>Input capture on counting up or down by TCNT (LWA = 0) or TCNTLW (LWA = 1) in MTU1*1 |
| 1 | 1 | 1 | x | | Input capture on generation of compare match with MTU8.TGRC |

**Remarks:** x: Don't care

Note 1. Input capture will not be generated in MTU0 if P0ϕ/1 is selected as the count clock for MTU1. Select a clock other than P0ϕ/1.

Table 16.29 TIORL (MTU0)

| Bit 3 | Bit 2 | Bit 1 | Bit 0 | Description | |
|---|---|---|---|---|---|
| IOC3 | IOC2 | IOC1 | IOC0 | MTU0.TGRC Function | MTIOC0C Pin Function |
| 0 | 0 | 0 | 0 | Output compare register*1 | Output prohibited |
| 0 | 0 | 0 | 1 | | Initial output is low.<br>Low output at compare match. |
| 0 | 0 | 1 | 0 | | Initial output is low.<br>High output at compare match. |
| 0 | 0 | 1 | 1 | | Initial output is low.<br>Toggle output at compare match. |
| 0 | 1 | 0 | 0 | | Output prohibited |
| 0 | 1 | 0 | 1 | | Initial output is high.<br>Low output at compare match. |
| 0 | 1 | 1 | 0 | | Initial output is high.<br>High output at compare match. |
| 0 | 1 | 1 | 1 | | Initial output is high.<br>Toggle output at compare match. |
| 1 | 0 | 0 | 0 | Input capture register*1 | Input capture at rising edge. |
| 1 | 0 | 0 | 1 | | Input capture at falling edge. |
| 1 | 0 | 1 | x | | Input capture at both edges. |
| 1 | 1 | x | x | | Capture input source is the clock source for counting in MTU1.<br>Input capture on counting up or down by TCNT (LWA = 0) or TCNTLW (LWA = 1) in MTU1*2 |

**Remarks:** x: Don’t care

Note 1. When the BFA bit in MTU0.TMDR1 is set to 1 and MTU0.TGRC is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Note 2. Input capture will not be generated in MTU0 if P0ϕ/1 is selected as the count clock for MTU1. Select a clock other than P0ϕ/1.

Table 16.30 TIOR (MTU1)

| Bit 3 | Bit 2 | Bit 1 | Bit 0 | Description | |
|---|---|---|---|---|---|
| IOA3 | IOA2 | IOA1 | IOA0 | MTU1.TGRA/TGRALW Function | MTIOC1A Pin Function |
| 0 | 0 | 0 | 0 | Output compare register | Output prohibited |
| 0 | 0 | 0 | 1 | | Initial output is low.<br>Low output at compare match. |
| 0 | 0 | 1 | 0 | | Initial output is low.<br>High output at compare match. |
| 0 | 0 | 1 | 1 | | Initial output is low.<br>Toggle output at compare match. |
| 0 | 1 | 0 | 0 | | Output prohibited |
| 0 | 1 | 0 | 1 | | Initial output is high.<br>Low output at compare match. |
| 0 | 1 | 1 | 0 | | Initial output is high.<br>High output at compare match. |
| 0 | 1 | 1 | 1 | | Initial output is high.<br>Toggle output at compare match. |
| 1 | 0 | 0 | 0 | Input capture register | Input capture at rising edge. |
| 1 | 0 | 0 | 1 | | Input capture at falling edge. |
| 1 | 0 | 1 | x | | Input capture at both edges. |
| 1 | 1 | x | x | | Input capture at generation of MTU0.TGRA compare match/input capture. |

**Remarks:** x: Don’t care

Table 16.31 TIOR (MTU2)

| Bit 3 | Bit 2 | Bit 1 | Bit 0 | Description | |
|---|---|---|---|---|---|
| IOA3 | IOA2 | IOA1 | IOA0 | MTU2.TGRA Function | MTIOC2A Pin Function |
| 0 | 0 | 0 | 0 | Output compare register | Output prohibited |
| 0 | 0 | 0 | 1 | | Initial output is low.<br>Low output at compare match. |
| 0 | 0 | 1 | 0 | | Initial output is low.<br>High output at compare match. |
| 0 | 0 | 1 | 1 | | Initial output is low.<br>Toggle output at compare match. |
| 0 | 1 | 0 | 0 | | Output prohibited |
| 0 | 1 | 0 | 1 | | Initial output is high.<br>Low output at compare match. |
| 0 | 1 | 1 | 0 | | Initial output is high.<br>High output at compare match. |
| 0 | 1 | 1 | 1 | | Initial output is high.<br>Toggle output at compare match. |
| 1 | x | 0 | 0 | Input capture register | Input capture at rising edge. |
| 1 | x | 0 | 1 | | Input capture at falling edge. |
| 1 | x | 1 | x | | Input capture at both edges. |

**Remarks:** x: Don’t care

Table 16.32 TIORH (MTU3)

| Bit 3 | Bit 2 | Bit 1 | Bit 0 | Description | |
|---|---|---|---|---|---|
| IOA3 | IOA2 | IOA1 | IOA0 | MTU3.TGRA Function | MTIOC3A Pin Function |
| 0 | 0 | 0 | 0 | Output compare register | Output prohibited |
| 0 | 0 | 0 | 1 | | Initial output is low.<br>Low output at compare match. |
| 0 | 0 | 1 | 0 | | Initial output is low.<br>High output at compare match. |
| 0 | 0 | 1 | 1 | | Initial output is low.<br>Toggle output at compare match. |
| 0 | 1 | 0 | 0 | | Output prohibited |
| 0 | 1 | 0 | 1 | | Initial output is high.<br>Low output at compare match. |
| 0 | 1 | 1 | 0 | | Initial output is high.<br>High output at compare match. |
| 0 | 1 | 1 | 1 | | Initial output is high.<br>Toggle output at compare match. |
| 1 | x | 0 | 0 | Input capture register | Input capture at rising edge. |
| 1 | x | 0 | 1 | | Input capture at falling edge. |
| 1 | x | 1 | x | | Input capture at both edges. |

**Remarks:** x: Don't care

Table 16.33 TIORL (MTU3)

| Bit 3 | Bit 2 | Bit 1 | Bit 0 | Description | |
|---|---|---|---|---|---|
| IOC3 | IOC2 | IOC1 | IOC0 | MTU3.TGRC Function | MTIOC3C Pin Function |
| 0 | 0 | 0 | 0 | Output compare register*1 | Output prohibited |
| 0 | 0 | 0 | 1 | | Initial output is low.<br>Low output at compare match. |
| 0 | 0 | 1 | 0 | | Initial output is low.<br>High output at compare match. |
| 0 | 0 | 1 | 1 | | Initial output is low.<br>Toggle output at compare match. |
| 0 | 1 | 0 | 0 | | Output prohibited |
| 0 | 1 | 0 | 1 | | Initial output is high.<br>Low output at compare match. |
| 0 | 1 | 1 | 0 | | Initial output is high.<br>High output at compare match. |
| 0 | 1 | 1 | 1 | | Initial output is high.<br>Toggle output at compare match. |
| 1 | x | 0 | 0 | Input capture register*1 | Input capture at rising edge. |
| 1 | x | 0 | 1 | | Input capture at falling edge. |
| 1 | x | 1 | x | | Input capture at both edges. |

**Remarks:** x: Don't care

Note 1. When the BFA bit in MTU3.TMDR1 is set to 1 and MTU3.TGRC is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 16.34 TIORH (MTU4)

| Bit 3 | Bit 2 | Bit 1 | Bit 0 | Description | |
|---|---|---|---|---|---|
| IOA3 | IOA2 | IOA1 | IOA0 | MTU4.TGRA Function | MTIOC4A Pin Function |
| 0 | 0 | 0 | 0 | Output compare register | Output prohibited |
| 0 | 0 | 0 | 1 | | Initial output is low.<br>Low output at compare match. |
| 0 | 0 | 1 | 0 | | Initial output is low.<br>High output at compare match. |
| 0 | 0 | 1 | 1 | | Initial output is low.<br>Toggle output at compare match. |
| 0 | 1 | 0 | 0 | | Output prohibited |
| 0 | 1 | 0 | 1 | | Initial output is high.<br>Low output at compare match. |
| 0 | 1 | 1 | 0 | | Initial output is high.<br>High output at compare match. |
| 0 | 1 | 1 | 1 | | Initial output is high.<br>Toggle output at compare match. |
| 1 | x | 0 | 0 | Input capture register | Input capture at rising edge. |
| 1 | x | 0 | 1 | | Input capture at falling edge. |
| 1 | x | 1 | x | | Input capture at both edges. |

**Remarks:** x: Don't care

Table 16.35 TIORL (MTU4)

| Bit 3 | Bit 2 | Bit 1 | Bit 0 | Description | |
|---|---|---|---|---|---|
| IOC3 | IOC2 | IOC1 | IOC0 | MTU4.TGRC Function | MTIOC4C Pin Function |
| 0 | 0 | 0 | 0 | Output compare register*1 | Output prohibited |
| 0 | 0 | 0 | 1 | | Initial output is low.<br>Low output at compare match. |
| 0 | 0 | 1 | 0 | | Initial output is low.<br>High output at compare match. |
| 0 | 0 | 1 | 1 | | Initial output is low.<br>Toggle output at compare match. |
| 0 | 1 | 0 | 0 | | Output prohibited |
| 0 | 1 | 0 | 1 | | Initial output is high.<br>Low output at compare match. |
| 0 | 1 | 1 | 0 | | Initial output is high.<br>High output at compare match. |
| 0 | 1 | 1 | 1 | | Initial output is high.<br>Toggle output at compare match. |
| 1 | x | 0 | 0 | Input capture register*1 | Input capture at rising edge. |
| 1 | x | 0 | 1 | | Input capture at falling edge. |
| 1 | x | 1 | x | | Input capture at both edges. |

**Remarks:** x: Don't care

Note 1. When the BFA bit in MTU4.TMDR1 is set to 1 and MTU4.TGRC register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 16.36 TIORH (MTU6)

| Bit 3 | Bit 2 | Bit 1 | Bit 0 | Description | |
|---|---|---|---|---|---|
| IOA3 | IOA2 | IOA1 | IOA0 | MTU6.TGRA Function | MTIOC6A Pin Function |
| 0 | 0 | 0 | 0 | Output compare register | Output prohibited |
| 0 | 0 | 0 | 1 | | Initial output is low.<br>Low output at compare match. |
| 0 | 0 | 1 | 0 | | Initial output is low.<br>High output at compare match. |
| 0 | 0 | 1 | 1 | | Initial output is low.<br>Toggle output at compare match. |
| 0 | 1 | 0 | 0 | | Output prohibited |
| 0 | 1 | 0 | 1 | | Initial output is high.<br>Low output at compare match. |
| 0 | 1 | 1 | 0 | | Initial output is high.<br>High output at compare match. |
| 0 | 1 | 1 | 1 | | Initial output is high.<br>Toggle output at compare match. |
| 1 | x | 0 | 0 | Input capture register | Input capture at rising edge. |
| 1 | x | 0 | 1 | | Input capture at falling edge. |
| 1 | x | 1 | x | | Input capture at both edges. |

**Remarks:** x: Don't care

Table 16.37 TIORL (MTU6)

| Bit 3 | Bit 2 | Bit 1 | Bit 0 | Description | |
|---|---|---|---|---|---|
| IOC3 | IOC2 | IOC1 | IOC0 | MTU6.TGRC Function | MTIOC6C Pin Function |
| 0 | 0 | 0 | 0 | Output compare register*1 | Output prohibited |
| 0 | 0 | 0 | 1 | | Initial output is low.<br>Low output at compare match. |
| 0 | 0 | 1 | 0 | | Initial output is low.<br>High output at compare match. |
| 0 | 0 | 1 | 1 | | Initial output is low.<br>Toggle output at compare match. |
| 0 | 1 | 0 | 0 | | Output prohibited |
| 0 | 1 | 0 | 1 | | Initial output is high.<br>Low output at compare match. |
| 0 | 1 | 1 | 0 | | Initial output is high.<br>High output at compare match. |
| 0 | 1 | 1 | 1 | | Initial output is high.<br>Toggle output at compare match. |
| 1 | x | 0 | 0 | Input capture register*1 | Input capture at rising edge. |
| 1 | x | 0 | 1 | | Input capture at falling edge. |
| 1 | x | 1 | x | | Input capture at both edges. |

**Remarks:** x: Don't care

Note 1. When the BFA bit in MTU6.TMDR1 is set to 1 and MTU6.TGRC register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 16.38 TIORH (MTU7)

| Bit 3 | Bit 2 | Bit 1 | Bit 0 | Description | |
|---|---|---|---|---|---|
| IOA3 | IOA2 | IOA1 | IOA0 | MTU7.TGRA Function | MTIOC7A Pin Function |
| 0 | 0 | 0 | 0 | Output compare register | Output prohibited |
| 0 | 0 | 0 | 1 | | Initial output is low.<br>Low output at compare match. |
| 0 | 0 | 1 | 0 | | Initial output is low.<br>High output at compare match. |
| 0 | 0 | 1 | 1 | | Initial output is low.<br>Toggle output at compare match. |
| 0 | 1 | 0 | 0 | | Output prohibited |
| 0 | 1 | 0 | 1 | | Initial output is high.<br>Low output at compare match. |
| 0 | 1 | 1 | 0 | | Initial output is high.<br>High output at compare match. |
| 0 | 1 | 1 | 1 | | Initial output is high.<br>Toggle output at compare match. |
| 1 | x | 0 | 0 | Input capture register | Input capture at rising edge. |
| 1 | x | 0 | 1 | | Input capture at falling edge. |
| 1 | x | 1 | x | | Input capture at both edges. |

**Remarks:** x: Don't care

Table 16.39 TIORL (MTU7)

| Bit 3 | Bit 2 | Bit 1 | Bit 0 | Description | |
|---|---|---|---|---|---|
| IOC3 | IOC2 | IOC1 | IOC0 | MTU7.TGRC Function | MTIOC7C Pin Function |
| 0 | 0 | 0 | 0 | Output compare register*1 | Output prohibited |
| 0 | 0 | 0 | 1 | | Initial output is low.<br>Low output at compare match. |
| 0 | 0 | 1 | 0 | | Initial output is low.<br>High output at compare match. |
| 0 | 0 | 1 | 1 | | Initial output is low.<br>Toggle output at compare match. |
| 0 | 1 | 0 | 0 | | Output prohibited |
| 0 | 1 | 0 | 1 | | Initial output is high.<br>Low output at compare match. |
| 0 | 1 | 1 | 0 | | Initial output is high.<br>High output at compare match. |
| 0 | 1 | 1 | 1 | | Initial output is high.<br>Toggle output at compare match. |
| 1 | x | 0 | 0 | Input capture register*1 | Input capture at rising edge. |
| 1 | x | 0 | 1 | | Input capture at falling edge. |
| 1 | x | 1 | x | | Input capture at both edges. |

**Remarks:** x: Don't care

Note 1. When the BFA bit in MTU7.TMDR1 is set to 1 and MTU7.TGRC register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 16.40 TIORH (MTU8)

| Bit 3 | Bit 2 | Bit 1 | Bit 0 | Description | |
|---|---|---|---|---|---|
| IOA3 | IOA2 | IOA1 | IOA0 | MTU8.TGRA Function | MTIOC8A Pin Function |
| 0 | 0 | 0 | 0 | Output compare register | Output prohibited |
| 0 | 0 | 0 | 1 | | Initial output is low.<br>Low output at compare match. |
| 0 | 0 | 1 | 0 | | Initial output is low.<br>High output at compare match. |
| 0 | 0 | 1 | 1 | | Initial output is low.<br>Toggle output at compare match. |
| 0 | 1 | 0 | 0 | | Output prohibited |
| 0 | 1 | 0 | 1 | | Initial output is high.<br>Low output at compare match. |
| 0 | 1 | 1 | 0 | | Initial output is high.<br>High output at compare match. |
| 0 | 1 | 1 | 1 | | Initial output is high.<br>Toggle output at compare match. |
| 1 | 0 | 0 | 0 | Input capture register | Input capture at rising edge. |
| 1 | 0 | 0 | 1 | | Input capture at falling edge. |
| 1 | x | 1 | x | | Input capture at both edges. |

**Remarks:** x: Don’t care

Table 16.41 TIORL (MTU8)

| Bit 3 | Bit 2 | Bit 1 | Bit 0 | Description | |
|---|---|---|---|---|---|
| IOC3 | IOC2 | IOC1 | IOC0 | MTU8.TGRC Function | MTIOC8C Pin Function |
| 0 | 0 | 0 | 0 | Output compare register*1 | Output prohibited |
| 0 | 0 | 0 | 1 | | Initial output is low.<br>Low output at compare match. |
| 0 | 0 | 1 | 0 | | Initial output is low.<br>High output at compare match. |
| 0 | 0 | 1 | 1 | | Initial output is low.<br>Toggle output at compare match. |
| 0 | 1 | 0 | 0 | | Output prohibited |
| 0 | 1 | 0 | 1 | | Initial output is high.<br>Low output at compare match. |
| 0 | 1 | 1 | 0 | | Initial output is high.<br>High output at compare match. |
| 0 | 1 | 1 | 1 | | Initial output is high.<br>Toggle output at compare match. |
| 1 | x | 0 | 0 | Input capture register | Input capture at rising edge. |
| 1 | x | 0 | 1 | | Input capture at falling edge. |
| 1 | x | 1 | x | | Input capture at both edges. |

**Remarks:** x: Don’t care

Note 1. When the MTU8.TMDR1.BFA bit is set to 1 and the MTU8.TGRC register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 16.42 TIORU, TIORV, and TIORW (MTU5)

| Bit 4 | Bit 3 | Bit 2 | Bit 1 | Bit 0 | Description | |
|---|---|---|---|---|---|---|
| IOC4 | IOC3 | IOC2 | IOC1 | IOC0 | MTU5.TGRU, MTU5.TGRV, MTU5.TGRW Function | MTIC5U, MTIC5V, MTIC5W Pin Function |
| 0 | 0 | 0 | 0 | 0 | Output compare register | No function |
| 0 | 0 | 0 | 0 | 1 | | Setting prohibited |
| 0 | 0 | 0 | 1 | x | | Setting prohibited |
| 0 | 0 | 1 | x | x | | Setting prohibited |
| 0 | 1 | x | x | x | | Setting prohibited |
| 1 | 0 | 0 | 0 | 0 | Input capture register*1 | Setting prohibited |
| 1 | 0 | 0 | 0 | 1 | | Input capture at rising edge. |
| 1 | 0 | 0 | 1 | 0 | | Input capture at falling edge. |
| 1 | 0 | 0 | 1 | 1 | | Input capture at both edges. |
| 1 | 0 | 1 | x | x | | Input capture on generation of compare match with MTU8.TGRC |
| 1 | 1 | 0 | 0 | 0 | | Setting prohibited |
| 1 | 1 | 0 | 0 | 1 | | Measurement of low pulse width of external input signal. Capture at trough in complementary PWM mode. |
| 1 | 1 | 0 | 1 | 0 | | Measurement of low pulse width of external input signal. Capture at crest of complementary PWM mode. |
| 1 | 1 | 0 | 1 | 1 | | Measurement of low pulse width of external input signal. Capture at crest and trough of complementary PWM mode. |
| 1 | 1 | 1 | 0 | 0 | | Setting prohibited |
| 1 | 1 | 1 | 0 | 1 | | Measurement of high pulse width of external input signal. Capture at trough in complementary PWM mode. |
| 1 | 1 | 1 | 1 | 0 | | Measurement of high pulse width of external input signal. Capture at crest of complementary PWM mode. |
| 1 | 1 | 1 | 1 | 1 | | Measurement of high pulse width of external input signal. Capture at crest and trough of complementary PWM mode. |

**Remarks:** x: Don't care

Note 1. Setting the IOC[4:0] bits to "H'19", "H'1A", "H'1B", "H'1D", "H'1E", or "H'1F" is only allowed when the external pulse width measurement function is used or the dead time compensation function, in coordination with MTU6 and MTU7, is used. For details, refer to **Section 16.3.11, External Pulse Width Measurement**, and **Section 16.3.12, Dead Time Compensation**.

## 16.2.7 Timer Compare Match Clear Register (TCNTCMPCLR)

**Address(es):** MTU5.TCNTCMPCLR H'0_1000_1CB6

| Bit | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | CMPCLR5U | CMPCLR5V | CMPCLR5W |
| 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 |
|---|---|---|---|---|
| b0 | CMPCLR5W | 0 | R/W | TCNT Compare Clear 5W<br>0: Disables MTU5.TCNTW to be cleared to H'0000 at MTU5.TCNTW and MTU5.TGRW compare match or input capture<br>1: Enables MTU5.TCNTW to be cleared to H'0000 at MTU5.TCNTW and MTU5.TGRW compare match or input capture |
| b1 | CMPCLR5V | 0 | R/W | TCNT Compare Clear 5V<br>0: Disables MTU5.TCNTV to be cleared to H'0000 at MTU5.TCNTV and MTU5.TGRV compare match or input capture<br>1: Enables MTU5.TCNTV to be cleared to H'0000 at MTU5.TCNTV and MTU5.TGRV compare match or input capture |
| b2 | CMPCLR5U | 0 | R/W | TCNT Compare Clear 5U<br>0: Disables MTU5.TCNTU to be cleared to H'0000 at MTU5.TCNTU and MTU5.TGRU compare match or input capture<br>1: Enables MTU5.TCNTU to be cleared to H'0000 at MTU5.TCNTU and MTU5.TGRU compare match or input capture |
| b7 to b3 | — | All 0 | R/W | Reserved<br>These bits are read as 0. The write value should be 0. |

TCNTCMPCLR specifies requests to clear MTU5.TCNTU, MTU5.TCNTV, and MTU5.TCNTW. The MTU has one TCNTCMPCLR (on MTU5).

## 16.2.8 Timer Interrupt Enable Register (TIER)

### (1) MTU1.TIER, MTU2.TIER

**Address(es):** MTU1.TIER H'0_1000_1384, MTU2.TIER H'0_1000_1404

| Bit | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | TTGE | — | TCIEU | TCIEV | — | — | 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 |

### (2) MTU0.TIER, MTU3.TIER, MTU6.TIER

**Address(es):** MTU0.TIER H'0_1000_1304, MTU3.TIER H'0_1000_1208, MTU6.TIER H'0_1000_1A08

| Bit | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | TTGE | — | — | 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 |

### (3) MTU4.TIER, MTU7.TIER

**Address(es):** MTU4.TIER H'0_1000_1209, MTU7.TIER H'0_1000_1A09

| Bit | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | TTGE | TTGE2 | — | 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 |

### (4) MTU8.TIER

**Address(es):** MTU8.TIER H'0_1000_1604

| Bit | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | 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 |
|---|---|---|---|---|
| b0 | TGIEA | 0 | R/W | TGR Interrupt Enable A<br>0: Interrupt requests (TGIA) disabled<br>1: Interrupt requests (TGIA) enabled |
| b1 | TGIEB | 0 | R/W | TGR Interrupt Enable B<br>0: Interrupt requests (TGIB) disabled<br>1: Interrupt requests (TGIB) enabled |
| b2 | TGIEC | 0 | R/W | TGR Interrupt Enable C<br>0: Interrupt requests (TGIC) disabled<br>1: Interrupt requests (TGIC) enabled |
| b3 | TGIED | 0 | R/W | TGR Interrupt Enable D<br>0: Interrupt requests (TGID) disabled<br>1: Interrupt requests (TGID) enabled |
| b4 | TCIEV | 0 | R/W | Overflow Interrupt Enable<br>0: Interrupt requests (TCIV) disabled<br>1: Interrupt requests (TCIV) enabled |
| b5 | TCIEU | 0 | R/W | Underflow Interrupt Enable<br>0: Interrupt requests (TCIU) disabled<br>1: Interrupt requests (TCIU) enabled |
| b6 | TTGE2 | 0 | R/W | A/D Converter Start Request Enable 2<br>0: A/D converter start request generation by MTUn.TCNT underflow (trough) disabled<br>1: A/D converter start request generation by MTUn.TCNT underflow (trough) enabled |
| b7 | TTGE | 0 | R/W | A/D Converter Start Request Enable<br>0: A/D converter start request generation disabled<br>1: A/D converter start request generation enabled |

**Note:** n = 4, 7

The TIER register enables or disables interrupt requests from each channel. The MTU has a total of ten TIER registers, two for MTU0 and one each for MTU1 to MTU8.

### TGIEA and TGIEB Bits (TGR Interrupt Enable A and B)

Each bit enables or disables interrupt requests (TGIn) (n = A, B).

### TGIEC and TGIED Bits (TGR Interrupt Enable C and D)

Each bit enables or disables an interrupt request (TGIn) (n = C, D).

In MTU1 and MTU2, these bits are reserved. They are read as 0. The write value should be 0.

### TCIEV Bit (Overflow Interrupt Enable)

This bit enables or disables interrupt requests (TCIV).

### TCIEU Bit (Underflow Interrupt Enable)

This bit enables or disables interrupt requests (TCIU).

In MTU0, MTU3, MTU4, MTU6, MTU7, and MTU8, this bit is reserved. It is read as 0. The write value should be 0.

### TTGE2 Bit (A/D Converter Start Request Enable 2)

This bit enables or disables generation of A/D converter start requests by MTUn.TCNT underflow (trough) in complementary PWM mode (n = 4, 7).

In MTU0 to MTU3, MTU6, and MTU8, this bit is reserved. It is read as 0. The write value should be 0.

### TTGE Bit (A/D Converter Start Request Enable)

This bit enables or disables generation of A/D converter start requests by TGRA input capture/compare match. MTU8 is a reserved bit. It is read as 0. The write value should be 0.

## (5) MTU0.TIER2

**Address(es):** MTU0.TIER2 H'0_1000_1324

| Bit | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | TTGE2 | — | — | — | — | — | TGIEF | TGIEE |
| 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 |
|---|---|---|---|---|
| b0 | TGIEE | 0 | R/W | TGR Interrupt Enable E<br>0: Interrupt requests (TGIE) disabled<br>1: Interrupt requests (TGIE) enabled |
| b1 | TGIEF | 0 | R/W | TGR Interrupt Enable F<br>0: Interrupt requests (TGIF) disabled<br>1: Interrupt requests (TGIF) enabled |
| b6 to b2 | — | All 0 | R/W | Reserved<br>These bits are read as 0. The write value should be 0. |
| b7 | TTGE2 | 0 | R/W | A/D Converter Start Request Enable 2<br>0: A/D converter start request generation by compare match between MTU0.TCNT and MTU0.TGRE disabled<br>1: A/D converter start request generation by compare match between MTU0.TCNT and MTU0.TGRE enabled |

### TGIEE and TGIEF Bits (TGR Interrupt Enable E and F)

Each bit enables or disables interrupt requests by compare match between MTU0.TCNT and MTU0.TGRn (n = E, F).

### TTGE2 Bit (A/D Converter Start Request Enable 2)

This bit enables or disables AD converter start requests by compare match between TCNT and TGRE in MTU0.

### (6) MTU5.TIER

**Address(es):** MTU5.TIER H'0_1000_1CB2

| Bit | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | TGIE5U | TGIE5V | TGIE5W |
| 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 |
|---|---|---|---|---|
| b0 | TGIE5W | 0 | R/W | TGR Interrupt Enable 5W<br>0: Interrupt requests TGIW5 disabled<br>1: Interrupt requests TGIW5 enabled |
| b1 | TGIE5V | 0 | R/W | TGR Interrupt Enable 5V<br>0: Interrupt requests TGIV5 disabled<br>1: Interrupt requests TGIV5 enabled |
| b2 | TGIE5U | 0 | R/W | TGR Interrupt Enable 5U<br>0: Interrupt requests TGIU5 disabled<br>1: Interrupt requests TGIU5 enabled |
| b7 to b3 | — | All 0 | R/W | Reserved<br>These bits are read as 0. The write value should be 0. |

#### TGIE5n Bits (TGR Interrupt Enable 5n)

Each bit enables or disables interrupt requests (TGIn5) (n = U, V, W).

## 16.2.9 Timer Status Register (TSR)

### (1) MTU1.TSR, MTU2.TSR

**Address(es):** MTU1.TSR H'0_1000_1385, MTU2.TSR H'0_1000_1405

| Bit | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | TCFD | — | — | — | — | — | — | — |
| Initial Value | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

### (2) MTU3.TSR, MTU4.TSR, MTU6.TSR, MTU7.TSR

**Address(es):** MTU3.TSR H'0_1000_122C, MTU4.TSR H'0_1000_122D, MTU6.TSR H'0_1000_1A2C, MTU7.TSR H'0_1000_1A2D

| Bit | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | TCFD | — | — | — | — | — | — | — |
| Initial Value | 1 | 1 | 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 |
|---|---|---|---|---|
| b5 to b0 | — | All 0 | R/W | Reserved<br>The read value is undefined. The write value should be 1. |
| b6 | — | 1 | R/W | Reserved<br>This bit is read as 1. The write value should be 1. |
| b7 | TCFD | 1 | R | Count Direction Flag<br>0: TCNT counts down<br>1: TCNT counts up |

TSR indicates the states of each of the channels. The MTU has a total of six TSR registers, one each for MTU1 to MTU4, MTU6, and MTU7.

#### TCFD Flag (Count Direction Flag)

Status flag that indicates the direction in which TCNT is counting in MTU1 to MTU4, MTU6, and MTU7.

## 16.2.10 Timer Buffer Operation Transfer Mode Register (TBTM)

### (1) MTU0.TBTM

**Address(es):** MTU0.TBTM H'0_1000_1326

| Bit | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | TTSE | TTSB | TTSA |
| 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 |

### (2) MTU3.TBTM, MTU4.TBTM, MTU6.TBTM, MTU7.TBTM

**Address(es):** MTU3.TBTM H'0_1000_1238, MTU4.TBTM H'0_1000_1239, MTU6.TBTM H'0_1000_1A38, MTU7.TBTM H'0_1000_1A39

| Bit | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | TTSB | TTSA |
| 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 |
|---|---|---|---|---|
| b0 | TTSA | 0 | R/W | Timing Select A<br>0: When compare match A occurs in each channel, data is transferred from TGRC to TGRA<br>1: When TCNT is cleared in each channel, data is transferred from TGRC to TGRA |
| b1 | TTSB | 0 | R/W | Timing Select B<br>0: When compare match B occurs in each channel, data is transferred from TGRD to TGRB<br>1: When TCNT is cleared in each channel, data is transferred from TGRD to TGRB |
| b2 | TTSE | 0 | R/W | Timing Select E<br>0: When compare match E occurs in MTU0, data is transferred from MTU0.TGRF to MTU0.TGRE<br>1: When MTU0.TCNT is cleared in MTU0, data is transferred from MTU0.TGRF to MTU0.TGRE |
| b7 to b3 | — | All 0 | R/W | Reserved<br>These bits are read as 0. The write value should be 0. |

TBTM specifies the timing for transferring data from the buffer register to the timer general register in PWM mode. The MTU has a total of five TBTM registers, one each for MTU0, MTU3, MTU4, MTU6, and MTU7.

#### TTSA Bit (Timing Select A)

This bit specifies the timing for transferring data from TGRC to TGRA in each channel when they are used together for buffer operation. When a channel is not set to PWM mode, do not set the TTSA bit in the channel to 1.

#### TTSB Bit (Timing Select B)

This bit specifies the timing for transferring data from TGRD to TGRB in each channel when they are used together for buffer operation. When a channel is not set to PWM mode, do not set the TTSB bit in the channel to 1.

#### TTSE Bit (Timing Select E)

This bit specifies the timing for transferring data from MTU0.TGRF to MTU0.TGRE when they are used together for buffer operation.

In MTU3, MTU4, MTU6, and MTU7, this bit is reserved. It is read as 0 and the write value should be 0. When a channel is not set to PWM mode, do not set the TTSE bit in the channel to 1.

## 16.2.11 Timer Input Capture Control Register (TICCR)

**Address(es):** MTU1.TICCR H'0_1000_1390

| Bit | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | I2BE | I2AE | I1BE | I1AE |
| 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 |
|---|---|---|---|---|
| b0 | I1AE | 0 | R/W | Input Capture Enable<br>0: Does not include the MTIOC1A pin in the MTU2.TGRA input capture conditions<br>1: Includes the MTIOC1A pin in the MTU2.TGRA input capture conditions |
| b1 | I1BE | 0 | R/W | Input Capture Enable<br>0: Does not include the TMTIOC1B pin in the MTU2.TGRB input capture conditions<br>1: Includes the MTIOC1B pin in the MTU2.TGRB input capture conditions |
| b2 | I2AE | 0 | R/W | Input Capture Enable<br>0: Does not include the MTIOC2A pin in the MTU1.TGRA input capture conditions<br>1: Includes the MTIOC2A pin in the MTU1.TGRA input capture conditions |
| b3 | I2BE | 0 | R/W | Input Capture Enable<br>0: Does not include the MTIOC2B pin in the MTU1.TGRB input capture conditions<br>1: Includes the MTIOC2B pin in the MTU1.TGRB input capture conditions |
| b7 to b4 | — | All 0 | R/W | Reserved<br>These bits are read as 0. The write value should be 0. |

TICCR specifies input capture conditions when MTU1.TCNT and MTU2.TCNT are cascaded. The MTU has one TICCR for MTU1.

## 16.2.12 Timer Synchronous Clear Register (TSYCR)

**Address(es):** MTU6.TSYCR H'0_1000_1A50

| Bit | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | CE0A | CE0B | CE0C | CE0D | CE1A | CE1B | CE2A | CE2B |
| 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 |
|---|---|---|---|---|
| b0 | CE2B | 0 | R/W | Clear Enable 2B<br>0: Disables counter clearing by the MTU2.TGIB2 interrupt generation timing.<br>1: Enables counter clearing by the MTU2.TGIB2 interrupt generation timing. |
| b1 | CE2A | 0 | R/W | Clear Enable 2A<br>0: Disables counter clearing by the MTU2.TGIA2 interrupt generation timing*1.<br>1: Enables counter clearing by the MTU2.TGIA2 interrupt generation timing*1. |
| b2 | CE1B | 0 | R/W | Clear Enable 1B<br>0: Disables counter clearing by the MTU1.TGIB1 interrupt generation timing*1.<br>1: Enables counter clearing by the MTU1.TGIB1 interrupt generation timing*1. |
| b3 | CE1A | 0 | R/W | Clear Enable 1A<br>0: Disables counter clearing by the MTU1.TGIA1 interrupt generation timing*1.<br>1: Enables counter clearing by the MTU1.TGIA1 interrupt generation timing*1. |
| b4 | CE0D | 0 | R/W | Clear Enable 0D<br>0: Disables counter clearing by the MTU0.TGID0 interrupt generation timing*1.<br>1: Enables counter clearing by the MTU0.TGID0 interrupt generation timing*1. |
| b5 | CE0C | 0 | R/W | Clear Enable 0C<br>0: Disables counter clearing by the MTU0.TGIC0 interrupt generation timing*1.<br>1: Enables counter clearing by the MTU0.TGIC0 interrupt generation timing*1. |
| b6 | CE0B | 0 | R/W | Clear Enable 0B<br>0: Disables counter clearing by the MTU0.TGIB0 interrupt generation timing*1.<br>1: Enables counter clearing by the MTU0.TGIB0 interrupt generation timing*1. |
| b7 | CE0A | 0 | R/W | Clear Enable 0A<br>0: Disables counter clearing by the MTU0.TGIA0 interrupt generation timing*1.<br>1: Enables counter clearing by the MTU0.TGIA0 interrupt generation timing*1. |

Note 1. This does not depend on the TIERn.TGIEm bit setting. (n = 0, 1, 2; m = A, B, C, D)

TSYCR specifies synchronous clear conditions for MTU6.TCNT and MTU7.TCNT. The MTU has one TSYCR for MTU1.

### CEnm Bits (Clear Enable nm; n = 0, 1, 2; m = A, B, C, D)

These bits enable or disable counter clearing by the MTUn.TGImn interrupt generation timing.

## 16.2.13 Timer Counter (TCNT)

### (1) MTU0.TCNT to MTU7.TCNT

**Address(es):** MTU0.TCNT H'0_1000_1306, MTU1.TCNT H'0_1000_1386, MTU2.TCNT H'0_1000_1406, MTU3.TCNT H'0_1000_1210, MTU4.TCNT H'0_1000_1212, MTU5.TCNTU H'0_1000_1C80, MTU5.TCNTV H'0_1000_1C90, MTU5.TCNTW H'0_1000_1CA0, MTU6.TCNT H'0_1000_1A10, MTU7.TCNT H'0_1000_1A12

| Bit | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | |
| 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:** TCNT must not be accessed in eight bits; it should be accessed in 16 bits.

### (2) MTU8.TCNT

**Address(es):** MTU8.TCNT H'0_1000_1608

| Bit | b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | |
| 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 | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | |
| 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:** TCNT must not be accessed in 8 or 16 bits; it should be accessed in 32 bits.

MTU0.TCNT to MTU7.TCNT are 16-bit readable/writable counters and MTU8.TCNT is a 32-bit readable/writable counter. The MTU has a total of 11 TCNT counters, one each for MTU0 to MTU4, MTU6, MTU7, and MTU8 and three (TCNTU, TCNTV, and TCNTW) for MTU5. The TCNT counters in MTU0 to MTU4, MTU6, and MTU7 are initialized to H'0000 by a reset, and the TCNT counter in MTU8 is initialized to H'00000000 by a reset. The TCNTU, TCNTV, and TCNTW counters in MTU5 are initialized to H'0000 by a reset. In MTU0 to MTU4, MTU6, and MTU7, the TCNT counters must not be accessed in 8-bit units; they should be accessed in 16-bit units. The MTU8.TCNT counter must not be accessed in 8- or 16-bit units; it should be accessed in 32-bit units.

The MTU1.TCNT and MTU2.TCNT counters are read as H'0000 when TMDR3.LWA is 1. See **Section 16.2.5, Timer Mode Register 3 (TMDR3)** for details.

## 16.2.14 Timer Longword Counter (TCNTLW)

**Address(es):** MTU1.TCNTLW H'0_1000_13A0

| Bit | b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 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 | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 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:** TCNTLW must not be accessed in 8 or 16 bits; it should be accessed in 32 bits.

The TCNTLW counter is a 32-bit readable/writable counter. Only one counter of this type is provided, and is formed by combining MTU1.TCNT and MTU2.TCNT. Such operation is only effective when TMDR3.LWA is 1. The TCNTLW counter is initialized to H'0000 0000 by a reset. This counter is read as H'0000 0000 when TMDR3.LWA is 0. See **Section 16.2.5, Timer Mode Register 3 (TMDR3)** for details. This register can only be used in 32-bit phase counting mode.

## 16.2.15 Timer General Register (TGR)

### (1) MTU0.TGR to MTU7.TGR

**Address(es):** MTU0.TGRA H'0_1000_1308, MTU0.TGRB H'0_1000_130A, MTU0.TGRC H'0_1000_130C, MTU0.TGRD H'0_1000_130E, MTU0.TGRE H'0_1000_1320, MTU0.TGRF H'0_1000_1322, MTU1.TGRA H'0_1000_1388, MTU1.TGRB H'0_1000_138A, MTU2.TGRA H'0_1000_1408, MTU2.TGRB H'0_1000_140A, MTU3.TGRA H'0_1000_1218, MTU3.TGRB H'0_1000_121A, MTU3.TGRC H'0_1000_1224, MTU3.TGRD H'0_1000_1226, MTU3.TGRE H'0_1000_1272, MTU4.TGRA H'0_1000_121C, MTU4.TGRB H'0_1000_121E, MTU4.TGRC H'0_1000_1228, MTU4.TGRD H'0_1000_122A, MTU4.TGRE H'0_1000_1274, MTU4.TGRF H'0_1000_1276, MTU5.TGRU H'0_1000_1C82, MTU5.TGRV H'0_1000_1C92, MTU5.TGRW H'0_1000_1CA2, MTU6.TGRA H'0_1000_1A18, MTU6.TGRB H'0_1000_1A1A, MTU6.TGRC H'0_1000_1A24, MTU6.TGRD H'0_1000_1A26, MTU6.TGRE H'0_1000_1A72, MTU7.TGRA H'0_1000_1A1C, MTU7.TGRB H'0_1000_1A1E, MTU7.TGRC H'0_1000_1A28, MTU7.TGRD H'0_1000_1A2A, MTU7.TGRE H'0_1000_1A74, MTU7.TGRF H'0_1000_1A76

| Bit | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | |
| 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:** TGR must not be accessed in eight bits; it should be accessed in 16 bits. The initial value of TGR is H'FFFF.

### (2) MTU8.TGR

**Address(es):** MTU8.TGRA H'0_1000_160C, MTU8.TGRB H'0_1000_1610, MTU8.TGRC H'0_1000_1614, MTU8.TGRD H'0_1000_1618

| Bit | b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | |
| 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 | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | |
| 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:** TGR must not be accessed in 8 or 16 bits; it should be accessed in 32 bits.

The MTU0.TGR to MTU7.TGR registers are 16-bit readable/writable registers; the MTU8.TGR register is a 32-bit readable/writable register. The MTU has a total of 39 TGR registers, six for MTU0, two each for MTU1 and MTU2, five each for MTU3 and MTU6, six each for MTU4 and MTU7, three for MTU5 and four for MTU8.

The TGRA, TGRB, TGRC, and TGRD registers function as either output compare or input capture registers. The TGRC and TGRD registers for MTU0, MTU3, MTU4, MTU6, MTU7, and MTU8 can also be designated for operation as buffer registers. TGR buffer register combinations are TGRA and TGRC, and TGRB and TGRD.

MTU0.TGRE and MTU0.TGRF function as compare registers. When the MTU0.TCNT count matches the MTU0.TGRE value, an A/D converter start request can be issued. The TGRF register can also be designated for operation as a buffer register. TGR buffer register combination is TGRE and TGRF. MTU5.TGRU, MTU5.TGRV, and MTU5.TGRW function as compare match, input capture, or external pulse width measurement registers.

The MTU1.TGRA, MTU2.TGRA, MTU1.TGRB, and MTU2.TGRB registers are read as H'0000 when TMDR3.LWA is 1. See **Section 16.2.5, Timer Mode Register 3 (TMDR3)** for details.

## 16.2.16 Timer Longword General Registers (TGRALW and TGRBLW)

**Address(es):** MTU1.TGRALW H'0_1000_13A4, MTU1.TGRBLW H'0_1000_13A8

| Bit | b31 | b30 | b29 | b28 | b27 | b26 | b25 | b24 | b23 | b22 | b21 | b20 | b19 | b18 | b17 | b16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 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 | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 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:** TGRALW and TGRBLW must not be accessed in 8 or 16 bits; they should be accessed in 32 bits.

The TGRALW (TGRBLW) register is a 32-bit readable/writable register. Two general registers of this type are provided, and are formed by combining the TGRA (TGRB) registers in MTU1 and MTU2. Such operation is only effective when TMDR3.LWA is 1.

The TGRALW (TGRBLW) register is initialized to H'FFFF FFFF by a reset, but it is read as H'0000 0000 when TMDR3.LWA is 0. See **Section 16.2.5, Timer Mode Register 3 (TMDR3)** for details.

The TGRALW (TGRBLW) register functions as an output compare or input capture register when TMDR3.LWA is 1. This register can only be used in 32-bit phase counting mode.

## 16.2.17 Timer Start Registers (TSTRA, TSTRB, and TSTR)

### (1) MTU.TSTRA (MTU0, MTU1, MTU2, MTU3, MTU4, MTU8)

**Address(es):** MTU.TSTRA H'0_1000_1280

| Bit | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | CST4 | CST3 | — | — | CST8 | CST2 | CST1 | CST0 |
| 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 |
|---|---|---|---|---|
| b0 | CST0 | 0 | R/W | Counter Start 0<br>0: MTU0.TCNT counting is stopped<br>1: MTU0.TCNT performs count operation |
| b1 | CST1 | 0 | R/W | Counter Start 1<br>0: MTU1.TCNT counting is stopped<br>1: MTU1.TCNT performs count operation |
| b2 | CST2 | 0 | R/W | Counter Start 2<br>0: MTU2.TCNT counting is stopped<br>1: MTU2.TCNT performs count operation |
| b3 | CST8 | 0 | R/W | Counter start 8<br>0: MTU8.TCNT counting is stopped<br>1: MTU8.TCNT performs count operation. |
| b5, b4 | — | All 0 | R/W | Reserved<br>These bits are read as 0. The write value should be 0. |
| b6 | CST3 | 0 | R/W | Counter Start 3<br>0: MTU3.TCNT counting is stopped<br>1: MTU3.TCNT performs count operation |
| b7 | CST4 | 0 | R/W | Counter Start 4<br>0: MTU4.TCNT counting is stopped<br>1: MTU4.TCNT performs count operation |

**Note:** When 1 is written to a bit in TCSYSTR, the corresponding bit in TSTRA is also set to 1 automatically.

The TSTRA register starts or stops TCNT operation in MTU0 to MTU4 and MTU8. TSTRA starts or stops TCNT operation in MTU0 to MTU4.

TSTRB starts or stops TCNT operation in MTU6 and MTU7. TSTR starts or stops TCNT operation in MTU5.

Before setting the operating mode in TMDR1 or setting the TCNT count clock in TCR, be sure to stop the TCNT counting.

#### CSTn Bits (Counter Start n) (n = 0, 1, 2, 3, 4, 8)

Each bit starts or stops TCNT in the corresponding channel.

If 0 is written to the CSTn bit during operation with the MTIOC pin designated for output, the counter stops. In this case, the initial output value specified by the TOCR1A or TOCR2A register is output from the MTIOC pin in complementary PWM mode or reset-synchronized PWM mode.

In the other modes, the level of the output compare signal from the MTIOC pin is retained. If TIOR is written to while the CSTn bit is 0, the level of output from the pin will be updated to the specified initial output value.

### (2) MTU.TSTRB(MTU6, MTU7)

**Address(es):** MTU.TSTRB H'0_1000_1A80

| Bit | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | CST7 | CST6 | — | — | — | — | — | — |
| 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 |
|---|---|---|---|---|
| b5 to b0 | — | All 0 | R/W | Reserved<br>These bits are read as 0. The write value should be 0. |
| b6 | CST6 | 0 | R/W | Counter Start 6<br>0: MTU6.TCNT counting is stopped<br>1: MTU6.TCNT performs count operation |
| b7 | CST7 | 0 | R/W | Counter Start 7<br>0: MTU7.TCNT counting is stopped<br>1: MTU7.TCNT performs count operation |

**Note:** When 1 is written to a bit in TCSYSTR, the corresponding bit in TSTRB is also set to 1 automatically.

#### CSTn Bits (Counter Start n) (n = 6, 7)

Each bit starts or stops TCNT in the corresponding channel.

If 0 is written to the CSTn bit during operation with the MTIOC pin designated for output, the counter stops. In this case, the initial output value specified by the TOCR1B or TOCR2B register is output from the MTIOC pin in complementary PWM mode or reset-synchronized PWM mode.

In the other modes, the level of the output compare signal from the MTIOC pin is retained. If TIOR is written to while the CSTn bit is 0, the level of output from the pin will be updated to the specified initial output value.

### (3) MTU5.TSTR(MTU5)

**Address(es):** MTU5.TSTR H'0_1000_1CB4

| Bit | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | CSTU5 | CSTV5 | CSTW5 |
| 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 |
|---|---|---|---|---|
| b0 | CSTW5 | 0 | R/W | Counter Start W5<br>0: MTU5.TCNTW counting is stopped<br>1: MTU5.TCNTW performs count operation |
| b1 | CSTV5 | 0 | R/W | Counter Start V5<br>0: MTU5.TCNTV counting is stopped<br>1: MTU5.TCNTV performs count operation |
| b2 | CSTU5 | 0 | R/W | Counter Start U5<br>0: MTU5.TCNTU counting is stopped<br>1: MTU5.TCNTU performs count operation |
| b7 to b3 | — | All 0 | R/W | Reserved<br>These bits are read as 0. The write value should be 0. |

## 16.2.18 Timer Synchronous Registers (TSYRA and TSYRB)

### (1) MTU.TSYRA(MTU0, MTU1, MTU2, MTU3, MTU4)

**Address(es):** MTU.TSYRA H'0_1000_1281

| Bit | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | SYNC4 | SYNC3 | — | — | — | SYNC2 | SYNC1 | SYNC0 |
| 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 |
|---|---|---|---|---|
| b0 | SYNC0 | 0 | R/W | Timer Synchronous Operation 0<br>0: MTU0.TCNT operates independently (TCNT setting/clearing is not related to other channels).<br>1: MTU0.TCNT performs synchronous operation. (TCNT synchronous setting/synchronous clearing is enabled.) |
| b1 | SYNC1 | 0 | R/W | Timer Synchronous Operation 1<br>0: MTU1.TCNT operates independently (TCNT setting/clearing is not related to other channels).<br>1: MTU1.TCNT performs synchronous operation. (TCNT synchronous setting/synchronous clearing is enabled.) |
| b2 | SYNC2 | 0 | R/W | Timer Synchronous Operation 2<br>0: MTU2.TCNT operates independently (TCNT setting/clearing is not related to other channels).<br>1: MTU2.TCNT performs synchronous operation. (TCNT synchronous setting/synchronous clearing is enabled.) |
| b5 to b3 | — | All 0 | R/W | Reserved<br>These bits are read as 0. The write value should be 0. |
| b6 | SYNC3 | 0 | R/W | Timer Synchronous Operation 3<br>0: MTU3.TCNT operates independently (TCNT setting/clearing is not related to other channels).<br>1: MTU3.TCNT performs synchronous operation. (TCNT synchronous setting/synchronous clearing is enabled.) |
| b7 | SYNC4 | 0 | R/W | Timer Synchronous Operation 4<br>0: MTU4.TCNT operates independently (TCNT setting/clearing is not related to other channels).<br>1: MTU4.TCNT performs synchronous operation. (TCNT synchronous setting/synchronous clearing is enabled.) |

TSYRA selects independent operation or synchronous operation of TCNT in MTU0 to MTU4. TSYRB selects independent operation or synchronous operation of TCNT in MTU6 and MTU7. A channel performs synchronous operation when the corresponding bit in TSYR is set to 1.

#### SYNCn Bits (Timer Synchronous Operation n) (n = 0, 1, 2, 3, 4)

Each bit selects whether operation is independent of or synchronized with other channels.

When synchronous operation is selected, the TCNT synchronous setting 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 the TCR.CCLR[2:0] bits.

### (2) MTU.TSYRB(MTU6, MTU7)

**Address(es):** MTU.TSYRB H'0_1000_1A81

| Bit | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | SYNC7 | SYNC6 | — | — | — | — | — | — |
| 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 |
|---|---|---|---|---|
| b5 to b0 | — | All 0 | R/W | Reserved<br>These bits are read as 0. The write value should be 0. |
| b6 | SYNC6 | 0 | R/W | Timer Synchronous Operation 6<br>0: MTU6.TCNT operates independently (TCNT setting/clearing is not related to other channels).<br>1: MTU6.TCNT performs synchronous operation. (TCNT synchronous setting/synchronous clearing is enabled.) |
| b7 | SYNC7 | 0 | R/W | Timer Synchronous Operation 7<br>0: MTU7.TCNT operates independently (TCNT setting/clearing is not related to other channels).<br>1: MTU7.TCNT performs synchronous operation. (TCNT synchronous setting/synchronous clearing is enabled.) |

#### SYNCn Bits (Timer Synchronous Operation n) (n = 6, 7)

Each bit selects whether operation is independent of or synchronized with other channels.

When synchronous operation is selected, the TCNT synchronous setting 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 TCR.CCLR[2:0].

## 16.2.19 Timer Counter Synchronous Start Register (TCSYSTR)

**Address(es):** MTU.TCSYSTR H'0_1000_1282

| Bit | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | SCH0 | SCH1 | SCH2 | SCH3 | SCH4 | — | SCH6 | SCH7 |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/(W)*1 | R/(W)*1 | R/(W)*1 | R/(W)*1 | R/(W)*1 | R | R/(W)*1 | R/(W)*1 |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| b0 | SCH7 | 0 | R/(W) *1 | Synchronous Start 7<br>0: Does not specify synchronous start for MTU7.TCNT<br>1: Specifies synchronous start for MTU7.TCNT |
| b1 | SCH6 | 0 | R/(W) *1 | Synchronous Start 6<br>0: Does not specify synchronous start for MTU6.TCNT<br>1: Specifies synchronous start for MTU6.TCNT |
| b2 | — | 0 | R | Reserved<br>This bit is read as 0. The write value should be 0. |
| b3 | SCH4 | 0 | R/(W) *1 | Synchronous Start 4<br>0: Does not specify synchronous start for MTU4.TCNT<br>1: Specifies synchronous start for MTU4.TCNT |
| b4 | SCH3 | 0 | R/(W) *1 | Synchronous Start 3<br>0: Does not specify synchronous start for MTU3.TCNT<br>1: Specifies synchronous start for MTU3.TCNT |
| b5 | SCH2 | 0 | R/(W) *1 | Synchronous Start 2<br>0: Does not specify synchronous start for MTU2.TCNT<br>1: Specifies synchronous start for MTU2.TCNT |
| b6 | SCH1 | 0 | R/(W) *1 | Synchronous Start 1<br>0: Does not specify synchronous start for MTU1.TCNT<br>1: Specifies synchronous start for MTU1.TCNT |
| b7 | SCH0 | 0 | R/(W) *1 | Synchronous Start 0<br>0: Does not specify synchronous start for MTU0.TCNT<br>1: Specifies synchronous start for MTU0.TCNT |

Note 1. Only 1 can be written to this bit. It is automatically cleared to 0 when counting begins.

TCSYSTR specifies synchronous start of the counters.

### SCH7 Bit (Synchronous Start 7)

This bit controls synchronous start of TCNT in MTU7. [Clearing condition]

- When the CST7 bit in TSTRB is set to 1 while SCH7 = 1

### SCH6 Bit (Synchronous Start 6)

This bit controls synchronous start of TCNT in MTU6. [Clearing condition]

- When the CST6 bit in TSTRB is set to 1 while SCH6 = 1

**SCH4 Bit (Synchronous Start 4)**

This bit controls synchronous start of TCNT in MTU4. [Clearing condition]

- When the CST4 bit in TSTRA is set to 1 while SCH4 = 1

**SCH3 Bit (Synchronous Start 3)**

This bit controls synchronous start of TCNT in MTU3. [Clearing condition]

- When the CST3 bit in TSTRA is set to 1 while SCH3 = 1

**SCH2 Bit (Synchronous Start 2)**

This bit controls synchronous start of TCNT in MTU2. [Clearing condition]

- When the CST2 bit in TSTRA is set to 1 while SCH2 = 1

**SCH1 Bit (Synchronous Start 1)**

This bit controls synchronous start of TCNT in MTU1. [Clearing condition]

- When the CST1 bit in TSTRA is set to 1 while SCH1 = 1

**SCH0 Bit (Synchronous Start 0)**

This bit controls synchronous start of TCNT in MTU0. [Clearing condition]

- When the CST0 bit in TSTRA is set to 1 while SCH0 = 1

## 16.2.20 Timer Read/Write Enable Registers (TRWERA and TRWERB)

**Address(es):** MTU.TRWERA H'0_1000_1284, MTU.TRWERB H'0_1000_1A84

| Bit | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | RWE |
| Initial Value | 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 |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| b0 | RWE | 1 | R/W | Read/Write Enable<br>0: Read/write access to the registers is disabled<br>1: Read/write access to the registers is enabled |
| b7 to b1 | — | All 0 | R/W | Reserved<br>These bits are read as 0. The write value should be 0. |

TRWERA enables or disables access to the registers and counters that have write-protection capability against accidental modification in MTU3 and MTU4.

TRWERB enables or disables access to the registers and counters that have write-protection capability against accidental modification in MTU6 and MTU7.

### RWE Bit (Read/Write Enable)

This bit enables or disables access to the registers that have write-protection capability against accidental modification.

[Clearing condition]

When 0 is written to the RWE bit after reading RWE = 1

- Registers and Counters having Write-Protection Capability against Accidental Modification (TRWERA)

  24 registers: MTUn.TCR, MTUn.TCR2, MTUn.TMDR1, MTUn.TIORH, MTUn.TIORL, MTUn.TIER, MTUn.TGRA, MTUn.TGRB, TOERA, TOCR1A, TOCR2A, TGCRA, TCDRA, TDDRA, and MTUn.TCNT (n = 3, 4)

- Registers and Counters having Write-Protection Capability against Accidental Modification (TRWERB)

  23 registers: MTUn.TCR, MTUn.TCR2, MTUn.TMDR1, MTUn.TIORH, MTUn.TIORL, MTUn.TIER, MTUn.TGRA, MTUn.TGRB, TOERB, TOCR1B, TOCR2B, TCDRB, TDDRB, and MTUn.TCNT (n = 6, 7)

## 16.2.21 Timer Output Master Enable Registers (TOERA and TOERB)

### (1) MTU.TOERA

**Address(es):** MTU.TOERA H'0_1000_120A

| Bit | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | OE4D | OE4C | OE3D | OE4B | OE4A | OE3B |
| Initial Value | 1 | 1 | 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 |
|---|---|---|---|---|
| b0 | OE3B | 0 | R/W | Master Enable MTIOC3B<br>0: MTU output is disabled*1<br>1: MTU output is enabled |
| b1 | OE4A | 0 | R/W | Master Enable MTIOC4A<br>0: MTU output is disabled*1<br>1: MTU output is enabled |
| b2 | OE4B | 0 | R/W | Master Enable MTIOC4B<br>0: MTU output is disabled*1<br>1: MTU output is enabled |
| b3 | OE3D | 0 | R/W | Master Enable MTIOC3D<br>0: MTU output is disabled*1<br>1: MTU output is enabled |
| b4 | OE4C | 0 | R/W | Master Enable MTIOC4C<br>0: MTU output is disabled*1<br>1: MTU output is enabled |
| b5 | OE4D | 0 | R/W | Master Enable MTIOC4D<br>0: MTU output is disabled*1<br>1: MTU output is enabled |
| b7, b6 | — | All 1 | R/W | Reserved<br>These bits are read as 1. The write value should be 1. |

Note 1. To output the inactive level from each pin when the MTU output is set to disabled, first set the port mode register (PMn) and port register (Pn) of I/O ports to output the inactive level from general I/O ports, and then set the port mode control register (PMCn) to use general I/O ports. For details, refer to **Section 41**, **General Purpose Input Output Port**.

TOERA enables or disables output settings for output pins MTIOC4D, MTIOC4C, MTIOC3D, MTIOC4B, MTIOC4A, and MTIOC3B.

These pins do not output values correctly if the bits of the TOERA register are not appropriately set. Write a desired value to the TOERA register before setting up the TIOR registers in MTU3 and MTU4.

Set the TOERA register after clearing the CST3 and CST4 bits in the TSTRA register to 0 (see **Figure 16.44** and **Figure 16.48**).

### (2) MTU.TOERB

**Address(es):** MTU.TOERB H'0_1000_1A0A

| Bit | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | OE7D | OE7C | OE6D | OE7B | OE7A | OE6B |
| Initial Value | 1 | 1 | 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 |
|---|---|---|---|---|
| b0 | OE6B | 0 | R/W | Master Enable MTIOC6B<br>0: MTU output is disabled*1<br>1: MTU output is enabled |
| b1 | OE7A | 0 | R/W | Master Enable MTIOC7A<br>0: MTU output is disabled*1<br>1: MTU output is enabled |
| b2 | OE7B | 0 | R/W | Master Enable MTIOC7B<br>0: MTU output is disabled*1<br>1: MTU output is enabled |
| b3 | OE6D | 0 | R/W | Master Enable MTIOC6D<br>0: MTU output is disabled*1<br>1: MTU output is enabled |
| b4 | OE7C | 0 | R/W | Master Enable MTIOC7C<br>0: MTU output is disabled*1<br>1: MTU output is enabled |
| b5 | OE7D | 0 | R/W | Master Enable MTIOC7D<br>0: MTU output is disabled*1<br>1: MTU output is enabled |
| b7, b6 | — | All 1 | R/W | Reserved<br>These bits are read as 1. The write value should be 1. |

Note 1. To output the inactive level from each pin when the MTU output is set to disabled, first set the port mode register (PMn) and port register (Pn) of I/O ports to output the inactive level from general I/O ports, and then set the port mode control register (PMCn) to use general I/O ports. For details, refer to **Section 41**, **General Purpose Input Output Port**.

TOERB enables or disables output settings for output pins MTIOC7D, MTIOC7C, MTIOC6D, MTIOC7B, MTIOC7A, and MTIOC6B.

These pins do not output values correctly if the bits of the TOERB register are not appropriately set. Write a desired value to the TOERB register before setting up the TIOR registers in MTU6 and MTU7.

Set the TOERB register after clearing the CST6 and CST7 bits in the TSTRB register to 0 (see **Figure 16.44** and **Figure 16.48**).

## 16.2.22 Timer Output Control Registers 1 (TOCR1A and TOCR1B)

**Address(es):** MTU.TOCR1A H'0_1000_120E, MTU.TOCR1B H'0_1000_1A0E

| Bit | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | PSYE | — | — | TOCL | TOCS | OLSN | OLSP |
| 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 |
|---|---|---|---|---|
| b0 | OLSP | 0 | R/W | Output Level Select P*1 *3<br>See **Table 16.43**. |
| b1 | OLSN | 0 | R/W | Output Level Select N*1 *3<br>See **Table 16.44**. |
| b2 | TOCS | 0 | R/W | TOC Select<br>0: TOCR1j setting is selected (j = A, B)<br>1: TOCR2j setting is selected |
| b3 | TOCL | 0 | R/W | TOC Register Write Protection*2 *4<br>0: Write access to the TOCS, OLSN, and OLSP bits is enabled<br>1: Write access to the TOCS, OLSN, and OLSP bits is disabled |
| b5, b4 | — | All 0 | R/W | Reserved<br>These bits are read as 0. The write value should be 0. |
| b6 | PSYE | 0 | R/W | PWM Synchronous Output Enable<br>0: Toggle output is disabled<br>1: Toggle output is enabled |
| b7 | — | 0 | R/W | Reserved<br>This bit is read as 0. The write value should be 0. |

Note 1. Clearing the TOCR1j.TOCS bit to 0 makes this bit setting valid.

Note 2. Setting the TOCR1j.TOCL bit to 1 prevents accidental modification when the CPU goes out of control.

Note 3. If dead-time is not generated, the negative-phase output is the exact inverse of the positive-phase output. In this case, only the OLSP bit is valid.

Note 4. This bit can be set to 1 only once after a reset. After 1 is written, 0 cannot be written to the bit.

TOCR1A and TOCR1B enable or disable PWM-synchronized toggle output in complementary PWM mode and reset-synchronized PWM mode, and control inversion of PWM output level.

### OLSP Bit (Output Level Select P)

This bit selects the positive-phase output level in reset-synchronized PWM mode and complementary PWM mode. The initial output is selected while the counter is stopped.

### OLSN Bit (Output Level Select N)

This bit selects the negative-phase output level in reset-synchronized PWM mode and complementary PWM mode. The initial output is selected while the counter is stopped.

### TOCS Bit (TOC Select)

This bit selects either the TOCR1j or TOCR2j (j = A, B) setting to be used for the output level in complementary PWM mode and reset-synchronized PWM mode.

### TOCL Bit (TOC Register Write Protection)

This bit enables or disables write access to the TOCS, OLSN, and OLSP bits in TOCR1j (j = A, B).

### PSYE Bit (PWM Synchronous Output Enable)

This bit enables or disables toggle output synchronized with the PWM cycle.

Table 16.43 Output Level Select Function

| Bit 0 | Function | | | |
|---|---|---|---|---|
| | | | Compare Match Output | |
| OLSP | Initial Output | Active Level | Up-Counting | Down-Counting |
| 0 | High level | Low level | Low level | High level |
| 1 | Low level | High level | High level | Low level |

Table 16.44 Output Level Select Function

| Bit 1 | Function | | | |
|---|---|---|---|---|
| | | | Compare Match Output | |
| OLSN | Initial Output | Active Level | Up-Counting | Down-Counting |
| 0 | High level | Low level | High level | Low level |
| 1 | Low level | High level | Low level | High level |

**Note:** The initial output value of negative-phase waveform changes to an active level after the dead time has passed since counting starts.

**Figure 16.3** shows an example of output in complementary PWM mode (one phase) when OLSN = 1 and OLSP = 1.

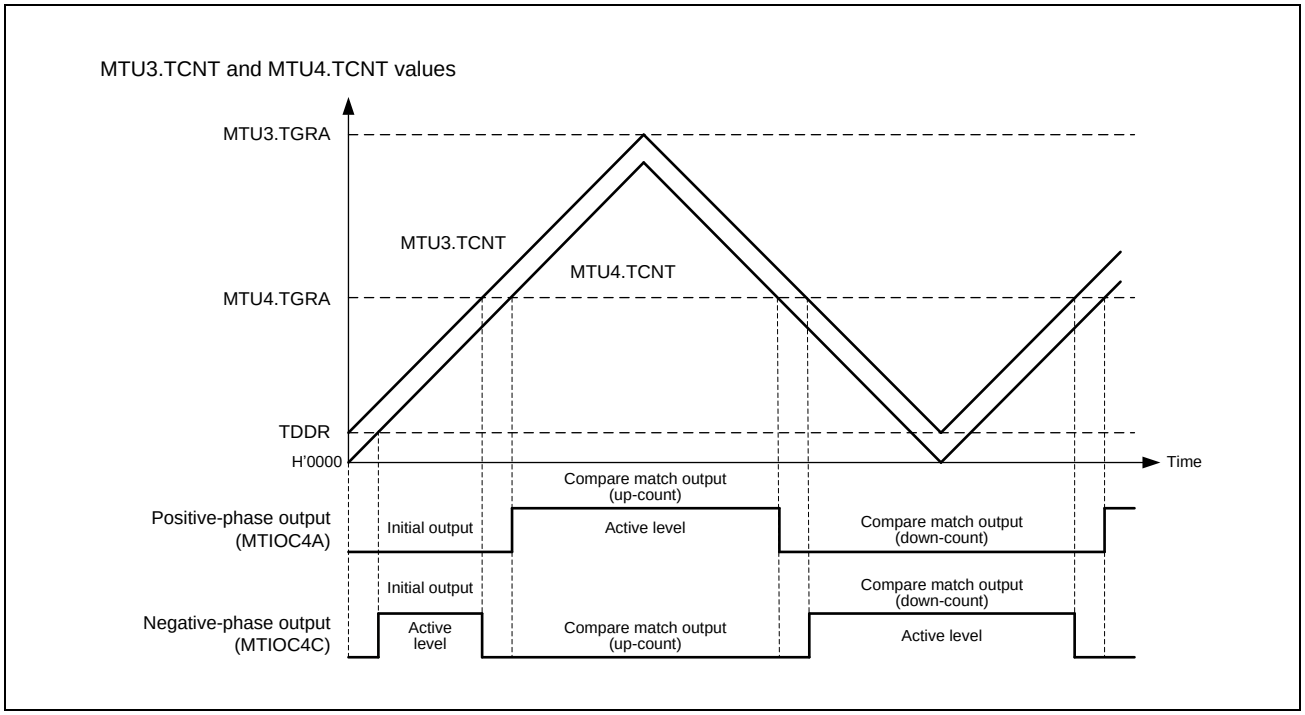



Figure 16.3 Example of Output in Complementary PWM Mode

## 16.2.23 Timer Output Control Registers 2 (TOCR2A and TOCR2B)

**Address(es):** MTU.TOCR2A H'0_1000_120F, MTU.TOCR2B H'0_1000_1A0F

| Bit | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | 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 |
|---|---|---|---|---|
| b0 | OLS1P | 0 | R/W | Output Level Select 1P*1 *2<br>This bit selects the output level on MTIOC3B or MTIOC6B in reset-synchronized PWM mode and complementary PWM mode.<br>See **Table 16.45**. |
| b1 | OLS1N | 0 | R/W | Output Level Select 1N*1 *2<br>This bit selects the output level on MTIOC3D or MTIOC6D in reset-synchronized PWM mode and complementary PWM mode.<br>See **Table 16.46**. |
| b2 | OLS2P | 0 | R/W | Output Level Select 2P*1 *2<br>This bit selects the output level on MTIOC4A or MTIOC7Ain reset-synchronized PWM mode and complementary PWM mode.<br>See **Table 16.47**. |
| b3 | OLS2N | 0 | R/W | Output Level Select 2N*1 *2<br>This bit selects the output level on MTIOC4C or MTIOC7C in reset-synchronized PWM mode and complementary PWM mode.<br>See **Table 16.48**. |
| b4 | OLS3P | 0 | R/W | Output Level Select 3P*1 *2<br>This bit selects the output level on MTIOC4B or MTIOC7B in reset-synchronized PWM mode and complementary PWM mode.<br>See **Table 16.49**. |
| b5 | OLS3N | 0 | R/W | Output Level Select 3N*1 *2<br>This bit selects the output level on MTIOC4D or MTIOC7D in reset-synchronized PWM mode and complementary PWM mode.<br>See **Table 16.50**. |
| b7, b6 | BF[1:0] | All 0 | R/W | TOLBR Buffer Transfer Timing Select<br>These bits select the timing for transferring data from TOLBRj to TOCR2j.<br>See **Table 16.51** for details. |

**Note:** j = A, B

Note 1. Setting the TOCR1j.TOCS bit to 1 makes this bit setting valid.

Note 2. If dead-time is not generated, the negative-phase output is the exact inverse of the positive-phase output. In this case, only the OLSiP bits are valid (i = 1 to 3).

TOCR2A and TOCR2B control inversion of PWM output level in complementary PWM mode and reset-synchronized PWM mode.

The initial output is selected while the counter is stopped.

Table 16.45 MTIOCmB Output Level Select Function

| Bit 0 | Function | | | |
|---|---|---|---|---|
| | | | Compare Match Output | |
| OLS1P | Initial Output | Active Level | Up-Counting | Down-Counting |
| 0 | High level | Low level | Low level | High level |
| 1 | Low level | High level | High level | Low level |

**Note:** m = 3, 6

Table 16.46 MTIOCmD Output Level Select Function

| Bit 1 | Function | | | |
|---|---|---|---|---|
| | | | Compare Match Output | |
| OLS1N | Initial Output | Active Level | Up-Counting | Down-Counting |
| 0 | High level | Low level | High level | Low level |
| 1 | Low level | High level | Low level | High level |

**Note:** m = 3, 6

**Note:** The initial output value of negative-phase waveform changes to an active level after the dead time has passed since counting starts.

Table 16.47 MTIOCmA Output Level Select Function

| Bit 2 | Function | | | |
|---|---|---|---|---|
| | | | Compare Match Output | |
| OLS2P | Initial Output | Active Level | Up-Counting | Down-Counting |
| 0 | High level | Low level | Low level | High level |
| 1 | Low level | High level | High level | Low level |

**Note:** m = 4, 7

Table 16.48 MTIOCmC Output Level Select Function

| Bit 3 | Function | | | |
|---|---|---|---|---|
| | | | Compare Match Output | |
| OLS2N | Initial Output | Active Level | Up-Counting | Down-Counting |
| 0 | High level | Low level | High level | Low level |
| 1 | Low level | High level | Low level | High level |

**Note:** m = 4, 7

**Note:** The initial output value of negative-phase waveform changes to an active level after the dead time has passed since counting starts.

Table 16.49 MTIOCmB Output Level Select Function

| Bit 4 | Function | | | |
|---|---|---|---|---|
| | | | Compare Match Output | |
| OLS3P | Initial Output | Active Level | Up-Counting | Down-Counting |
| 0 | High level | Low level | Low level | High level |
| 1 | Low level | High level | High level | Low level |

**Note:** m = 4, 7

Table 16.50 MTIOCmD Output Level Select Function

| Bit 5 | Function | | | |
|---|---|---|---|---|
| | | | Compare Match Output | |
| OLS3N | Initial Output | Active Level | Up-Counting | Down-Counting |
| 0 | High level | Low level | High level | Low level |
| 1 | Low level | High level | Low level | High level |

**Note:** m = 4, 7

**Note:** The initial output value of negative-phase waveform changes to an active level after the dead time has passed since counting starts.

Table 16.51 Setting of TOCR2j.BF[1:0] Bits

| Bit 7 | Bit 6 | Description | |
|---|---|---|---|
| BF1 | BF0 | Complementary PWM Mode | Reset-Synchronized PWM Mode |
| 0 | 0 | Does not transfer data from the buffer register (TOLBRj) to TOCR2j. | Does not transfer data from the buffer register (TOLBRj) to TOCR2j. |
| | 1 | Transfers data from the buffer register (TOLBRj) to TOCR2j at the crest of the MTUn.TCNT count. | Transfers data from the buffer register (TOLBRj) to TOCR2j when MTUm.TCNT or MTUn.TCNT is cleared. |
| 1 | 0 | Transfers data from the buffer register (TOLBRj) to TOCR2j at the trough of the MTUn.TCNT count. | Setting prohibited |
| | 1 | Transfers data from the buffer register (TOLBRj) to TOCR2j at the crest and trough of the MTUn.TCNT count. | Setting prohibited |

**Note:** n = 4, 7;
m = 3, 6;
j = A, B

## 16.2.24 Timer Output Level Buffer Registers (TOLBRA and TOLBRB)

**Address(es):** MTU.TOLBRA H'0_1000_1236, MTU.TOLBRB H'0_1000_1A36

| Bit | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | 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 |
|---|---|---|---|---|
| b0 | OLS1P | 0 | R/W | Output Level Select 1P<br>Specify the buffer value to be transferred to the OLS1P bit in TOCR2j. |
| b1 | OLS1N | 0 | R/W | Output Level Select 1N<br>Specify the buffer value to be transferred to the OLS1N bit in TOCR2j. |
| b2 | OLS2P | 0 | R/W | Output Level Select 2P<br>Specify the buffer value to be transferred to the OLS2P bit in TOCR2j. |
| b3 | OLS2N | 0 | R/W | Output Level Select 2N<br>Specify the buffer value to be transferred to the OLS2N bit in TOCR2j. |
| b4 | OLS3P | 0 | R/W | Output Level Select 3P<br>Specify the buffer value to be transferred to the OLS3P bit in TOCR2j. |
| b5 | OLS3N | 0 | R/W | Output Level Select 3N<br>Specify the buffer value to be transferred to the OLS3N bit in TOCR2j. |
| b7, b6 | — | All 0 | R/W | Reserved<br>These bits are read as 0. The write value should be 0. |

**Note:** j = A, B

TOLBRA and TOLBRB are buffer registers for TOCR2A and TOCR2B and specify the PWM output level in complementary PWM mode and reset-synchronized PWM mode.

**Figure 16.4** shows an example of the PWM output level setting procedure in buffer operation.

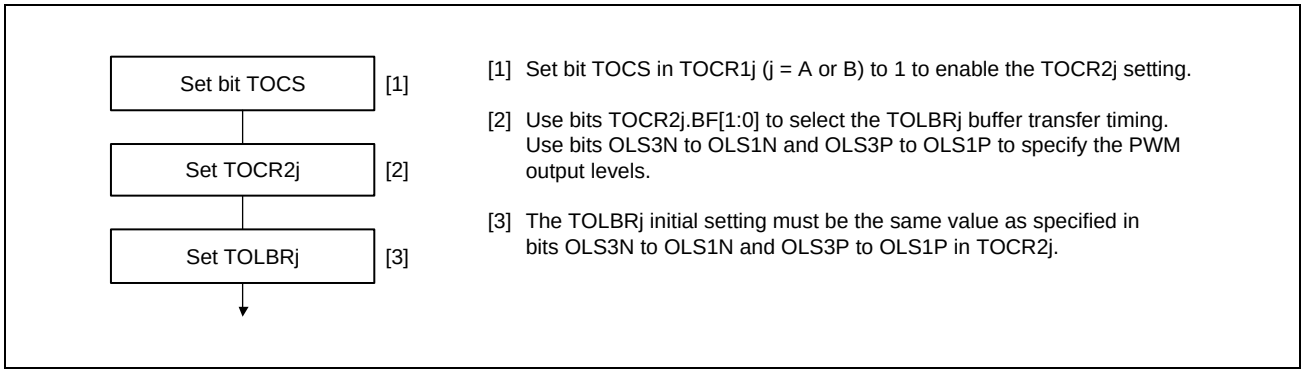



Figure 16.4 Example of PWM Output Level Setting Procedure in Buffer Operation

## 16.2.25 Timer Gate Control Register A (TGCRA)

**Address(es):** MTU.TGCRA H'0_1000_120D

| Bit | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | BDC | N | P | FB | WF | VF | UF |
| Initial Value | 1 | 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 |
|---|---|---|---|---|
| b0 | UF | 0 | R/W | Output Phase Switch<br>These bits turn on or off the positive-phase/negative-phase output. The setting of these bits is valid only when the FB bit is set to 1. In this case, the setting of b0 to b2 is used instead of the external input. See **Table 16.52**. |
| b1 | VF | 0 | R/W | |
| b2 | WF | 0 | R/W | |
| b3 | FB | 0 | R/W | External Feedback Signal Enable<br>0: Output is switched by external input (input sources are TGRA, TGRB, and TGRC input capture signals in MTU0)<br>1: Output is switched by software (TGCRA's UF, VF, and WF settings) |
| b4 | P | 0 | R/W | Positive-Phase Output (P) Control<br>0: Level output<br>1: Reset-synchronized PWM or complementary PWM output |
| b5 | N | 0 | R/W | Negative-Phase Output (N) Control<br>0: Level output<br>1: Reset-synchronized PWM or complementary PWM output |
| b6 | BDC | 0 | R/W | Brushless DC Motor<br>0: Ordinary output<br>1: Functions of this register are effective. |
| b7 | — | 1 | R/W | Reserved<br>This bit is read as 1. The write value should be 1. |

TGCRA controls the output waveform necessary for brushless DC motor control in reset-synchronized PWM mode and complementary PWM mode. TGCRA register settings are ineffective for anything other than complementary PWM mode and reset-synchronized PWM mode.

### UF, VF, and WF Bits (Output Phase Switch)

The setting of these bits is valid only when the FB bit is set to 1. In this case, the setting of b0 to b2 is used instead of the external input. See **Table 16.52** for details.

### FB Bit (External Feedback Signal Enable)

This bit selects whether the positive-/negative-phase output is switched automatically with the TGRA, TGRB, and TGRC input capture signals in MTU0 or by writing 0 or 1 to bits 2 to 0 in TGCRA.

When the FB bit in the TGCRA register is 0, the output signals from MTU3 and MTU4 are switched automatically with the TGRA, TGRB, and TGRC input capture signals in MTU0.

### P Bit (Positive-Phase Output (P) Control)

This bit selects either the level output or the reset-synchronized PWM/complementary PWM output for the positive-phase output pins (MTIOC3B, MTIOC4A, and MTIOC4B pins).

### N Bit (Negative-Phase Output (N) Control)

This bit selects the level output or the reset-synchronized PWM/complementary PWM output for the negative-phase output pins (MTIOC3D, MTIOC4C, and MTIOC4D pins).

### BDC Bit (Brushless DC Motor)

This bit selects whether to make the functions of TGCRA effective or ineffective.

Table 16.52 Output Level Select Function

| Bit 2 | Bit 1 | Bit 0 | Function | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | | MTIOC3B | MTIOC4A | MTIOC4B | MTIOC3D | MTIOC4C | MTIOC4D |
| 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 |

## 16.2.26 Timer Subcounters (TCNTSA and TCNTSB)

**Address(es):** MTU.TCNTSA H'0_1000_1220, MTU.TCNTSB H'0_1000_1A20

| Bit | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | |
| 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:** TCNTSA and TCNTSB must not be accessed in eight bits; it should be accessed in 16 bits.

TCNTSA and TCNTSB are 16-bit read-only counters that are used only in complementary PWM mode. The initial value of TCNTSA and TCNTSB after a reset is H'0000.

## 16.2.27 Timer Cycle Data Registers (TCDRA and TCDRB)

**Address(es):** MTU.TCDRA H'0_1000_1214, MTU.TCDRB H'0_1000_1A14

| Bit | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | |
| 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:** TCDRA and TCDRB must not be accessed in eight bits; it should be accessed in 16 bits.

TCDRA and TCDRB are 16-bit readable/writable registers used only in complementary PWM mode. Set half the PWM carrier cycle as the TCDRA and TCDRB values. The TCDRA and TCDRB registers are constantly compared with the TCNTSA and TCNTSB counters in complementary PWM mode, respectively. When a match occurs, the TCNTSA and TCNTSB counters switch the count direction (down-count to up-count). The initial value of TCDRA and TCDRB after a reset is H'FFFF.

## 16.2.28 Timer Cycle Buffer Registers (TCBRA and TCBRB)

**Address(es):** MTU.TCBRA H'0_1000_1222, MTU.TCBRB H'0_1000_1A22

| Bit | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | |
| 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:** TCBRA and TCBRB must not be accessed in eight bits; it should be accessed in 16 bits.

TCBRA and TCBRB are 16-bit readable/writable registers, used only in complementary PWM mode, that function as buffer registers for TCDRA and TCDRB. The TCBRA and TCBRB values are transferred to TCDRA and TCDRB with the transfer timing set in TMDR1. The initial value of TCBRA and TCBRB after a reset is H'FFFF.

## 16.2.29 Timer Dead Time Data Registers (TDDRA and TDDRB)

**Address(es):** MTU.TDDRA H'0_1000_1216, MTU.TDDRB H'0_1000_1A16

| Bit | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | |
| 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:** TDDRA and TDDRB must not be accessed in eight bits; it should be accessed in 16 bits.

TDDRA and TDDRB are 16-bit readable/writable registers, used only in complementary PWM mode, that specify the MTU3.TCNT (MTU6.TCNT) and MTU4.TCNT (MTU7.TCNT) counter offset value. In complementary PWM mode, when the MTU3.TCNT (MTU6.TCNT) and MTU4.TCNT (MTU7.TCNT) counters are cleared and then restarted, the TDDRA (TDDRB) value is loaded into the MTU3.TCNT (MTU6.TCNT) counter and the count operation starts. The initial value of TDDRA and TDDRB after a reset is H'FFFF.

## 16.2.30 Timer Dead Time Enable Registers (TDERA and TDERB)

**Address(es):** MTU.TDERA H'0_1000_1234, MTU.TDERB H'0_1000_1A34

| Bit | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | TDER |
| Initial Value | 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) |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| b0 | TDER | 1 | R/(W) | Dead Time Enable<br>0: No dead time is generated<br>1: Dead time is generated*1 |
| b7 to b1 | — | All 0 | R/W | Reserved<br>These bits are read as 0. The write value should be 0. |

Note 1. TDDRA and TDDRB must be set to 1 or a larger value.

TDERA and TDERB control dead time generation in complementary PWM mode. The MTU has one TDER each for MTU3 and MTU6. TDERA and TDERB should be modified only while TCNT stops.

### TDER Bit (Dead Time Enable)

This bit specifies whether to generate dead time.

[Clearing condition]

- When 0 is written to TDER after reading TDER = 1

## 16.2.31 Timer Buffer Transfer Set Registers (TBTERA and TBTERB)

**Address(es):** MTU.TBTERA H'0_1000_1232, MTU.TBTERB H'0_1000_1A32

| Bit | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | BTE[1:0] | |
| 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 |
|---|---|---|---|---|
| b1, b0 | BTE[1:0] | All 0 | R/W | Buffer Transfer Disable and Interrupt Skipping Link Setting<br>These bits enable or disable transfer from the buffer registers*1 used in complementary PWM mode to the temporary registers, and specify whether to link the transfer with interrupt skipping function 1.<br>For details, see **Table 16.53**. |
| b7 to b2 | — | All 0 | R/W | Reserved<br>These bits are read as 0. The write value should be 0. |

Note 1. Applicable buffer registers (TBTERA):
MTU3.TGRC, MTU3.TGRD, MTU4.TGRC, MTU4.TGRD, and TCBRA
Applicable buffer registers (TBTERB):
MTU6.TGRC, MTU6.TGRD, MTU7.TGRC, MTU7.TGRD, and TCBRB

TBTERA and TBTERB 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 1 operation.

Table 16.53 Setting of BTE[1:0] Bits in TBTERA and TBTERB

| 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 function 1. |
| | 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 function 1.*2 |
| | 1 | Setting prohibited |

Note 1. Data is transferred according to the MD3 to MD0 bit setting in TMDR1. For details, refer to **Section 16.3.8, Complementary PWM Mode**.

Note 2. When interrupt skipping is disabled (the T3AEN and T4VEN (T6AEN and T7VEN) bits are cleared to 0 in the timer interrupt skipping set register (TITCR1A (TITCR1B)) or the skipping count set bits (T3ACOR and T4VCOR (T6ACOR and T7VCOR)) in TITCR1A (TITCR1B) 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 (TBTERA (TBTERB)) to 0). If link with interrupt skipping is enabled while interrupt skipping is disabled, buffer transfer will not be performed.

## 16.2.32 Timer Waveform Control Registers (TWCRA and TWCRB)

**Address(es):** MTU.TWCRA H'0_1000_1260, MTU.TWCRB H'0_1000_1A60

| Bit | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | CCE | — | — | — | — | — | SCC | WRE |
| Initial Value | 0*2 | 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)*3 |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| b0 | WRE | 0 | R/(W) *3 | Waveform Retain Enable<br>0: Initial values specified in TOCR1A and TOCR2A (TOCR1B and TOCR2B) are output<br>1: Initial output is inhibited |
| b1 | SCC*1 *3 | 0 | R/(W) | Synchronous Clearing Control<br>(Only valid in register TWCRB)<br>0: Clearing of MTU6.TCNT and MTU7.TCNT in response to synchronous clearing for MTU0, MTU1, MTU2–MTU6, MTU7 is enabled.<br>1: Clearing of MTU6.TCNT and MTU7.TCNT in response to synchronous clearing for MTU0, MTU1, MTU2–MTU6, MTU7 is disabled. |
| b6 to b2 | — | All 0 | R/W | Reserved<br>These bits are read as 0. The write value should be 0. |
| b7 | CCE*2 | 0 | R/(W) | Compare Match Clear Enable<br>0: Counters are not cleared at MTU3.TGRA (MTU6.TGRA) compare match<br>1: Counters are cleared at MTU3.TGRA (MTU6.TGRA) compare match |

Note 1. This bit is only valid in register TWCRB and is a reserved bit in register TWCRA.

Note 2. Do not set to 1 when complementary PWM mode 1 is not selected.

Note 3. Do not set to 1 when complementary PWM mode is not selected.

TWCRA and TWCRB control the output waveform when synchronous counter clearing occurs in MTU3.TNCT and MTU4.TNCT (MTU6.TNCT and MTU7.TNCT) in complementary PWM mode and specifies whether to clear the counters at MTU3.TGRA (MTU6.TGRA) compare match.

The CCE bit and WRE bit in TWCRA and TWCRB should be modified only while TCNT stops.

### WRE Bit (Waveform Retain Enable)

This bit selects the waveform output when synchronous counter clearing occurs in complementary PWM mode. The initial output is inhibited with this function 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 values

specified in TOCR1A and TOCR2A (TOCR1B and TOCR2B) are output regardless of the WRE bit setting. The initial values specified in TOCR1A and TOCR2A (TOCR1B and TOCR2B) are also output when synchronous clearing occurs in the Tb interval at the trough immediately after MTU3.TCNT and MTU4.TCNT (MTU6.TCNT and MTU7.TCNT) start operation.

For the Tb interval at the trough in complementary PWM mode, see **Figure 16.50**.

[Setting condition]

- When 1 is written to the WRE bit after reading WRE = 0

### SCC Bit (Synchronous Clearing Control)

The setting of this bit selects whether MTU6.TCNT and MTU7.TCNT are or are not cleared when counter-synchronous clearing is generated for MTU0, MTU1, MTU2–MTU6, MTU7 in complementary PWM mode.

Make the complementary PWM mode settings for MTU6 and MTU7 when this function is in use. When writing a new value to the SCC bit while the counter is operating, do not change the values of the CCE and WRE bits.

Synchronous clearing from the MTU module only becomes disabled due to the setting of the SCC bit when synchronous clearing is generated outside the Tb interval in the trough. If synchronous clearing is generated within the Tb interval in the trough including immediately after the value at which MTU6.TCNT and MTU7.TCNT start, MTU6.TCNT and MTU7.TCNT are cleared.

Regarding the Tb interval in the trough in complementary PWM mode, see **Figure 16.50**.

[Setting condition]

- Writing of 1 to the SCC bit after reading it as 0

The corresponding bit in register TWCRA is reserved and is read as 0. When writing to TWCRA, write 0 to this bit.

### CCE Bit (Compare Match Clear Enable)

This bit specifies whether to clear counters at MTU3.TGRA (MTU6.TGRA) compare match in complementary PWM mode.

[Setting condition]

When 1 is written to CCE after reading CCE = 0

## 16.2.33 Noise Filter Control Register n (NFCRn) (n = 0 to 4, 6, 7, 8, C)

### (1) MTU0.NFCR0, MTU1.NFCR1, MTU2.NFCR2, MTU3.NFCR3, MTU4.NFCR4, MTU6.NFCR6, MTU7.NFCR7, MTU8.NFCR8

**Address(es):** MTU0.NFCR0 H'0_1000_1290, MTU1.NFCR1 H'0_1000_1291, MTU2.NFCR2 H'0_1000_1292, MTU3.NFCR3 H'0_1000_1293, MTU4.NFCR4 H'0_1000_1294, MTU6.NFCR6 H'0_1000_1A93, MTU7.NFCR7 H'0_1000_1A94, MTU8.NFCR8 H'0_1000_1298

| Bit | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | NFCS[1:0] | | NFDEN | NFCEN | NFBEN | NFAEN |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W*1 | R/W*1 | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| b0 | NFAEN | 0 | R/W | Noise Filter A Enable<br>0: The noise filter for the MTIOCnA pin is disabled.<br>1: The noise filter for the MTIOCnA pin is enabled. |
| b1 | NFBEN | 0 | R/W | Noise Filter B Enable<br>0: The noise filter for the MTIOCnB pin is disabled.<br>1: The noise filter for the MTIOCnB pin is enabled. |
| b2 | NFCEN | 0 | R/W*1 | Noise Filter C Enable<br>0: The noise filter for the MTIOCnC pin is disabled.<br>1: The noise filter for the MTIOCnC pin is enabled. |
| b3 | NFDEN | 0 | R/W*1 | Noise Filter D Enable<br>0: The noise filter for the MTIOCnD pin is disabled.<br>1: The noise filter for the MTIOCnD pin is enabled. |
| b5, b4 | NFCS[1:0] | All 0 | R/W | Noise Filter Clock Select<br>b5 b4<br>0 0: P0ϕ/1<br>0 1: P0ϕ/8<br>1 0: P0ϕ/32<br>1 1: Clock source for counting |
| b7, b6 | — | All 0 | R/W | Reserved<br>These bits are read as 0. The write value should be 0. |

Note 1. These bits are reserved in the NFCR1 and NFCR2 registers. These bits are read as 0 and writing to them has no effect.

The NFCRn register (n = 0 to 4, 6, 7, or 8) specifies the noise filter function of the input capture input pin of the corresponding channel.

#### NFAEN Bit (Noise Filter A Enable)

This bit disables or enables the noise filter for input from the MTIOCnA pin. Since changing the value of the bit may lead to the internal generation of an unexpected edge, select the output compare function for the relevant pin in the timer I/O control register or set the TMDR.MD[3:0] bits to a value other than that for normal mode (0000b) before doing so.

### NFBEN Bit (Noise Filter B Enable)

This bit disables or enables the noise filter for input from the MTIOCnB pin. Since changing the value of the bit may lead to the internal generation of an unexpected edge, select the output compare function for the relevant pin in the timer I/O control register or set the TMDR.MD[3:0] bits to a value other than that for normal mode (0000b) before doing so.

### NFCEN Bit (Noise Filter C Enable)

This bit disables or enables the noise filter for input from the MTIOCnC pin. Since changing the value of the bit may lead to the internal generation of an unexpected edge, select the output compare function for the relevant pin in the timer I/O control register or set the TMDR.MD[3:0] bits to a value other than that for normal mode (0000b) before doing so.

### NFDEN Bit (Noise Filter D Enable)

This bit disables or enables the noise filter for input from the MTIOCnD pin. Since changing the value of the bit may lead to the internal generation of an unexpected edge, select the output compare function for the relevant pin in the timer I/O control register or set the TMDR.MD[3:0] bits to a value other than that for normal mode (0000b) before doing so.

### NFCS[1:0] Bits (Noise Filter Clock Select)

These bits set the sampling interval for the noise filters. When setting the NFCS[1:0] bits, wait for two cycles of the selected sampling interval before setting the input-capture function. When the NFCS[1:0] bits are set to 11b, i.e. selecting the external clock as the source to drive counting, wait for two cycles of the external clock before setting the input capture function.

### (2) MTU0.NFCRC

**Address(es):** MTU0.NFCRC H'0_1000_1299

| Bit | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | NFCS[1:0] | | NFDEN | NFCEN | NFBEN | NFAEN |
| 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 |
|---|---|---|---|---|
| b0 | NFAEN | 0 | R/W | Noise Filter A Enable<br>0: The noise filter for the MTCLKA pin is disabled.<br>1: The noise filter for the MTCLKA pin is enabled. |
| b1 | NFBEN | 0 | R/W | Noise Filter B Enable<br>0: The noise filter for the MTCLKB pin is disabled.<br>1: The noise filter for the MTCLKB pin is enabled. |
| b2 | NFCEN | 0 | R/W | Noise Filter C Enable<br>0: The noise filter for the MTCLKC pin is disabled.<br>1: The noise filter for the MTCLKC pin is enabled. |
| b3 | NFDEN | 0 | R/W | Noise Filter D Enable<br>0: The noise filter for the MTCLKD pin is disabled.<br>1: The noise filter for the MTCLKD pin is enabled. |
| b5, b4 | NFCS[1:0] | All 0 | R/W | Noise Filter Clock Select<br>b5 b4<br>0 0: P0ϕ/1<br>0 1: P0ϕ/2<br>1 0: P0ϕ/8<br>1 1: P0ϕ/32 |
| b7, b6 | — | All 0 | R/W | Reserved<br>These bits are read as 0. The write value should be 0. |

The NFCRC register is used in common with all channels and specifies the noise filter function of the external clock pins.

#### NFAEN Bit (Noise Filter A Enable)

This bit disables or enables the noise filter for input from the MTCLKA pin. Since changing the value of the bit may lead to the internal generation of an unexpected edge, do so after stopping the internal counter.

#### NFBEN Bit (Noise Filter B Enable)

This bit disables or enables the noise filter for input from the MTCLKB pin. Since changing the value of the bit may lead to the internal generation of an unexpected edge, do so after stopping the internal counter.

#### NFCEN Bit (Noise Filter C Enable)

This bit disables or enables the noise filter for input from the MTCLKC pin. Since changing the value of the bit may lead to the internal generation of an unexpected edge, do so after stopping the internal counter.

#### NFDEN Bit (Noise Filter D Enable)

This bit disables or enables the noise filter for input from the MTCLKD pin. Since changing the value of the bit may lead to the internal generation of an unexpected edge, do so after stopping the internal counter.

#### NFCS[1:0] Bits (Noise Filter Clock Select)

These bits set the sampling interval for the noise filters. After setting the NFCS[1:0] bits, wait for two cycles of the selected sampling interval to set the input capture function.

## 16.2.34 Noise Filter Control Register 5 (NFCR5)

**Address(es):** MTU5.NFCR5 H'0_1000_1A95

| Bit | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | NFCS[1:0] | | — | NFWEN | NFVEN | NFUEN |
| 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 |
|---|---|---|---|---|
| b0 | NFUEN | 0 | R/W | Noise Filter U Enable<br>0: The noise filter for the MTIC5U pin is disabled.<br>1: The noise filter for the MTIC5U pin is enabled. |
| b1 | NFVEN | 0 | R/W | Noise Filter V Enable<br>0: The noise filter for the MTIC5V pin is disabled.<br>1: The noise filter for the MTIC5V pin is enabled. |
| b2 | NFWEN | 0 | R/W | Noise Filter W Enable<br>0: The noise filter for the MTIC5W pin is disabled.<br>1: The noise filter for the MTIC5W pin is enabled. |
| b3 | — | 0 | R/W | Reserved<br>These bits are read as 0. The write value should be 0. |
| b5, b4 | NFCS[1:0] | All 0 | R/W | Noise Filter Clock Select<br>b5 b4<br>0 0: P0ϕ/1<br>0 1: P0ϕ/8<br>1 0: P0ϕ/32<br>1 1: Clock source for counting |
| b7, b6 | — | All 0 | R/W | Reserved<br>These bits are read as 0. The write value should be 0. |

### NFUEN Bit (Noise Filter U Enable)

This bit disables or enables the noise filter for input from the MTIC5U pin. Since unexpected edges may be internally generated when the value of this bit is changed, select the output compare function for the relevant pin in the timer I/O control register.

### NFVEN Bit (Noise Filter V Enable)

This bit disables or enables the noise filter for input from the MTIC5V pin. Since unexpected edges may be internally generated when the value of this bit is changed, select the output compare function for the relevant pin in the timer I/O control register.

### NFWEN Bit (Noise Filter W Enable)

This bit disables or enables the noise filter for input from the MTIC5W pin. Since unexpected edges may be internally generated when the value of this bit is changed, select the output compare function for the relevant pin in the timer I/O control register.

### NFCS[1:0] Bits (Noise Filter Clock Select)

These bits set the sampling interval for the noise filters. When setting the NFCS[1:0] bits, wait for two cycles of the selected sampling interval before setting the input-capture function.

## 16.2.35 Timer A/D Converter Start Request Control Register (TADCR)

### (1) MTU4.TADCR

**Address(es):** MTU4.TADCR H'0_1000_1240

| Bit | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 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/W | R/W | 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 |
|---|---|---|---|---|
| b0 | ITB4VE | 0 | R/W | TCIV4 Interrupt Skipping Link Enable*1 *2 *3<br>0: A/D converter start request signal TRG4BN and TCIV4 interrupt skipping 1 are not linked<br>1: A/D converter start request signal TRG4BN and TCIV4 interrupt skipping 1 are linked |
| b1 | ITB3AE | 0 | R/W | TGIA3 Interrupt Skipping Link Enable*1 *2 *3<br>0: A/D converter start request signal TRG4BN and TGIA3 interrupt skipping 1 are not linked<br>1: A/D converter start request signal TRG4BN and TGIA3 interrupt skipping 1 are linked |
| b2 | ITA4VE | 0 | R/W | TCIV4 Interrupt Skipping Link Enable*1 *2 *3<br>0: A/D converter start request signal TRG4AN and TCIV4 interrupt skipping 1 are not linked<br>1: A/D converter start request signal TRG4AN and TCIV4 interrupt skipping 1 are linked |
| b3 | ITA3AE | 0 | R/W | TGIA3 Interrupt Skipping Link Enable*1 *2 *3<br>0: A/D converter start request signal TRG4AN and TGIA3 interrupt skipping 1 are not linked<br>1: A/D converter start request signal TRG4AN and TGIA3 interrupt skipping 1 are linked |
| b4 | DT4BE | 0 | R/W | Down-Count TRG4BN Enable*3<br>0: A/D converter start requests (TRG4BN) disabled during MTU4.TCNT down-count operation<br>1: A/D converter start requests (TRG4BN) enabled during MTU4.TCNT down-count operation |
| b5 | UT4BE | 0 | R/W | Up-Count TRG4BN Enable<br>0: A/D converter start requests (TRG4BN) disabled during MTU4.TCNT up-count operation<br>1: A/D converter start requests (TRG4BN) enabled during MTU4.TCNT up-count operation |
| b6 | DT4AE | 0 | R/W | Down-Count TRG4AN Enable*3<br>0: A/D converter start requests (TRG4AN) disabled during MTU4.TCNT down-count operation<br>1: A/D converter start requests (TRG4AN) enabled during MTU4.TCNT down-count operation |
| b7 | UT4AE | 0 | R/W | Up-Count TRG4AN Enable<br>0: A/D converter start requests (TRG4AN) disabled during MTU4.TCNT up-count operation<br>1: A/D converter up requests (TRG4AN) enabled during MTU4.TCNT down-count operation |
| b13 to b8 | — | All 0 | R/W | Reserved<br>These bits are read as 0. The write value should be 0. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| b15, b14 | BF[1:0] | All 0 | R/W | MTU4.TADCOBRA/TADCOBRB Transfer Timing Select<br>See **Table 16.54** for details.<br>These bits specify the transfer timing from MTU4.TADCOBRA and MTU4.TADCOBRB to MTU4.TADCORA and MTU4.TADCORB. |

**Note:** The TADCR register in MTU4 must not be accessed in eight bits; it should be accessed in 16 bits.

Note 1. Clear the bit to 0 when interrupt skipping is disabled (the T3AEN or T4VEN bit in TITCR1A is cleared to 0 or the T3ACOR or T4VCOR bits in TITCR1A are cleared to 0).

Note 2. If link with interrupt skipping is enabled while interrupt skipping is disabled, A/D converter start requests will not be issued.

Note 3. Clear the bit to 0 when complementary PWM mode is not selected.

TADCR enables or disables A/D converter start requests and specifies whether to link A/D converter start requests with interrupt skipping function. The MTU has one TADCR each for MTU4 and MTU7.

Table 16.54 Setting of Transfer Timing by TADCR.BF[1:0] Bits (MTU4)

| Bit 15 | Bit 14 | Description | | | |
|---|---|---|---|---|---|
| BF[1] | BF[0] | Complementary PWM Mode | Reset-Synchronized PWM Mode | PWM Mode 1 | Normal Mode |
| 0 | 0 | Does not transfer data from the cycle set buffer register (TADCOBRA or TADCOBRB in MTU4) to the cycle set register (TADCORA or TADCORB in MTU4). | Does not transfer data from the cycle set buffer register (TADCOBRA or TADCOBRB in MTU4) to the cycle set register (TADCORA or TADCORB in MTU4). | Does not transfer data from the cycle set buffer register (TADCOBRA or TADCOBRB in MTU4) to the cycle set register (TADCORA or TADCORB in MTU4). | Does not transfer data from the cycle set buffer register (TADCOBRA or TADCOBRB in MTU4) to the cycle set register (TADCORA or TADCORB in MTU4). |
| 0 | 1 | Transfers data from the cycle set buffer register (TADCOBRA or TADCOBRB in MTU4) to the cycle set register (TADCORA or TADCORB in MTU4) at the crest of the TCNT count in MTU4. | Transfers data from the cycle set buffer register (TADCOBRA or TADCOBRB in MTU4) to the cycle set register (TADCORA or TADCORB in MTU4) when a compare match occurs between TCNT and TGRA in MTU3. | Transfers data from the cycle set buffer register (TADCOBRA or TADCOBRB in MTU4) to the cycle set register (TADCORA or TADCORB in MTU4) when a compare match occurs between TCNT and TGRA in MTU4. | Transfers data from the cycle set buffer register (TADCOBRA or TADCOBRB in MTU4) to the cycle set register (TADCORA or TADCORB in MTU4) when a compare match occurs between TCNT and TGRA in MTU4. |
| 1 | 0 | Transfers data from the cycle set buffer register (TADCOBRA or TADCOBRB in MTU4) to the cycle set register (TADCORA or TADCORB in MTU4) at the trough of the TCNT count in MTU4. | Setting prohibited | Setting prohibited | Setting prohibited |
| 1 | 1 | Transfers data from the cycle set buffer register (TADCOBRA or TADCOBRB in MTU4) to the cycle set register (TADCORA or TADCORB in MTU4) at the crest and trough of the TCNT count in MTU4. | Setting prohibited | Setting prohibited | Setting prohibited |

### (2) MTU7.TADCR

**Address(es):** MTU7, TADCR H'0_1000_1A40

| Bit | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | BF[1:0] | | — | — | — | — | — | — | UT7AE | DT7AE | UT7BE | DT7BE | ITA6AE | ITA7VE | ITB6AE | ITB7VE |
| 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 |
|---|---|---|---|---|
| b0 | ITB7VE | 0 | R/W | TCIV7 Interrupt Skipping Link Enable*1 *2 *3<br>0: A/D converter start request signal TRG7BN and TCIV7 interrupt skipping 1 are not linked<br>1: A/D converter start request signal TRG7BN and TCIV7 interrupt skipping 1 are linked |
| b1 | ITB6AE | 0 | R/W | TGIA6 Interrupt Skipping Link Enable*1 *2 *3<br>0: A/D converter start request signal TRG7BN and TGIA6 interrupt skipping 1 are not linked<br>1: A/D converter start request signal TRG7BN and TGIA6 interrupt skipping 1 are linked |
| b2 | ITA7VE | 0 | R/W | TCIV7 Interrupt Skipping Link Enable*1 *2 *3<br>0: A/D converter start request signal TRG7AN and TCIV7 interrupt skipping 1 are not linked<br>1: A/D converter start request signal TRG7AN and TCIV7 interrupt skipping 1 are linked |
| b3 | ITA6AE | 0 | R/W | TGIA6 Interrupt Skipping Link Enable*1 *2 *3<br>0: A/D converter start request signal TRG7AN and TGIA6 interrupt skipping 1 are not linked<br>1: A/D converter start request signal TRG7AN and TGIA6 interrupt skipping 1 are linked |
| b4 | DT7BE | 0 | R/W | Down-Count TRG7BN Enable*3<br>0: A/D converter start requests (TRG7BN) disabled during MTU7.TCNT down-count operation<br>1: A/D converter start requests (TRG7BN) enabled during MTU7.TCNT down-count operation |
| b5 | UT7BE | 0 | R/W | Up-Count TRG7BN Enable<br>0: A/D converter start requests (TRG7BN) disabled during MTU7.TCNT up-count operation<br>1: A/D converter start requests (TRG7BN) enabled during MTU7.TCNT up-count operation |
| b6 | DT7AE | 0 | R/W | Down-Count TRG7AN Enable*3<br>0: A/D converter start requests (TRG7AN) disabled during MTU7.TCNT down-count operation<br>1: A/D converter start requests (TRG7AN) enabled during MTU7.TCNT down-count operation |
| b7 | UT7AE | 0 | R/W | Up-Count TRG7AN Enable<br>0: A/D converter start requests (TRG7AN) disabled during MTU7.TCNT up-count operation<br>1: A/D converter up requests (TRG7AN) enabled during MTU7.TCNT down-count operation |
| b13 to b8 | — | All 0 | R/W | Reserved<br>These bits are read as 0. The write value should be 0. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| b15, b14 | BF[1:0] | All 0 | R/W | MTU7.TADCOBRA/TADCOBRB Transfer Timing Select<br>See **Table 16.55** for details.<br>These bits specify the transfer timing from MTU7.TADCOBRA and MTU7.TADCOBRB to MTU7.TADCORA and MTU7.TADCORB. |

**Note:** The TADCR register in MTU7 must not be accessed in eight bits; it should be accessed in 16 bits.

Note 1. Clear the bit to 0 when interrupt skipping is disabled (the T6AEN or T7VEN bit in TITCR1B is cleared to 0 or the T6ACOR or T7VCOR bits in TITCR1B are cleared to 0).

Note 2. If link with interrupt skipping is enabled while interrupt skipping is disabled, A/D converter start requests will not be issued.

Note 3. Clear the bit to 0 when complementary PWM mode is not selected.

Table 16.55 Setting of Transfer Timing by TADCR.BF[1:0] Bits (MTU7)

| Bit 15 | Bit 14 | Description | | | |
|---|---|---|---|---|---|
| BF[1] | BF[0] | Complementary PWM Mode | Reset-Synchronized PWM Mode | PWM Mode 1 | Normal Mode |
| 0 | 0 | Does not transfer data from the cycle set buffer register (TADCOBRA or TADCOBRB in MTU7) to the cycle set register (TADCORA or TADCORB in MTU7). | Does not transfer data from the cycle set buffer register (TADCOBRA or TADCOBRB in MTU7) to the cycle set register (TADCORA or TADCORB in MTU7). | Does not transfer data from the cycle set buffer register (TADCOBRA or TADCOBRB in MTU7) to the cycle set register (TADCORA or TADCORB in MTU7). | Does not transfer data from the cycle set buffer register (TADCOBRA or TADCOBRB in MTU7) to the cycle set register (TADCORA or TADCORB in MTU7). |
| 0 | 1 | Transfers data from the cycle set buffer register (TADCOBRA or TADCOBRB in MTU7) to the cycle set register (TADCORA or TADCORB in MTU7) at the crest of the TCNT count in MTU7. | Transfers data from the cycle set buffer register (TADCOBRA or TADCOBRB in MTU7) to the cycle set register (TADCORA or TADCORB in MTU7) when a compare match occurs between TCNT and TGRA in MTU6. | Transfers data from the cycle set buffer register (TADCOBRA or TADCOBRB in MTU7) to the cycle set register (TADCORA or TADCORB in MTU7) when a compare match occurs between TCNT and TGRA in MTU7. | Transfers data from the cycle set buffer register (TADCOBRA or TADCOBRB in MTU7) to the cycle set register (TADCORA or TADCORB in MTU7) when a compare match occurs between TCNT and TGRA in MTU7. |
| 1 | 0 | Transfers data from the cycle set buffer register (TADCOBRA or TADCOBRB in MTU7) to the cycle set register (TADCORA or TADCORB in MTU7) at the trough of the TCNT count in MTU7. | Setting prohibited | Setting prohibited | Setting prohibited |
| 1 | 1 | Transfers data from the cycle set buffer register (TADCOBRA or TADCOBRB in MTU7) to the cycle set register (TADCORA or TADCORB in MTU7) at the crest and trough of the TCNT count in MTU7. | Setting prohibited | Setting prohibited | Setting prohibited |

## 16.2.36 Timer A/D Converter Start Request Cycle Set Registers (TADCORA and TADCORB)

**Address(es):** MTU4.TADCORA H'0_1000_1244, MTU4.TADCORB H'0_1000_1246, MTU7.TADCORA H'0_1000_1A44, MTU7.TADCORB H'0_1000_1A46

| Bit | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | |
| 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 and TADCORB must not be accessed in eight bits; it should be accessed in 16 bits.

Note 1. When the A/D converter start request delaying function linked with skipping function 1 (for details, see **Section 16.3.9(5), A/D Converter Start Request Delaying Function Linked with Interrupt Skipping Function 1**) is used, the value of this register should be H'0002 to TCDRA setting - 2 in MTU4 and H'0002 to TCDRB setting - 2 in MTU7.

Note 2. When interrupt skipping function 2 is used and the difference between the TADCORA value and the TADCORB value is small, the skipping count may not be counted correctly and the A/D converter start request may not be generated with the expected timing in some cases. The TADCORA and TADCORB values should satisfy the following conditions.

(1) When skipping function 2 is specified with the skipping count set to 0
- The difference between the TADCORA and TADCORB values should be equal to or greater than 4.
- The TADCORA compare interval should be equal to or greater than 4 P0ϕ cycles (the TADCORA update value should be the previous value + 4 or greater, or previous value - 4 or smaller).
- The TADCORB compare interval should be equal to or greater than 4 P0ϕ cycles (the TADCORB update value should be the previous value + 4 or greater, or previous value - 4 or smaller).

(2) When skipping function 2 is specified with the skipping count set to 1 or greater
- The difference between the TADCORA and TADCORB values should be equal to or greater than 2.
- The TADCORB compare interval should be equal to or greater than 2 P0ϕ cycles (the TADCORB update value should be the previous value + 2 or greater, or previous value - 2 or smaller).

TADCORA and TADCORB are 16-bit readable/writable registers that issue a corresponding A/D converter start request when the MTUn.TCNT (n = 4, 7) count reaches the value in TADCORA or TADCORB.

MTUn.TADCORA and TADCORB are initialized to H'FFFF by a reset.

## 16.2.37 Timer A/D Converter Start Request Cycle Set Buffer Registers (TADCOBRA and TADCOBRB)

**Address(es):** MTU4.TADCOBRA H'0_1000_1248, MTU4.TADCOBRB H'0_1000_124A, MTU7.TADCOBRA H'0_1000_1A48, MTU7.TADCOBRB H'0_1000_1A4A

| Bit | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | | | | | | | | |
| 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 and TADCOBRB must not be accessed in eight bits; it should be accessed in 16 bits.

TADCOBRA and TADCOBRB are 16-bit readable/writable registers whose values are transferred to TADCORA and TADCORB, respectively, when the crest or trough of the MTUn.TCNT count is reached.

TADCOBRA and TADCOBRB are initialized to H'FFFF by a reset.

## 16.2.38 Timer Interrupt Skipping Mode Registers (TITMRA and TITMRB)

**Address(es):** MTU.TITMRA H'0_1000_123A, MTU.TITMRB H'0_1000_1A3A

| Bit | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | TITM |
| 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 |
|---|---|---|---|---|
| b0 | TITM | 0 | R/W | Interrupt Skipping Function Select<br>Selects one of the two types of interrupt skipping functions.<br>0: Selects interrupt skipping function 1*1<br>1: Selects interrupt skipping function 2*2 |
| b7 to b1 | — | All 0 | R/W | Reserved<br>These bits are read as 0. The write value should be 0. |

Note 1. TITCR1A or TITCR1B is used to enable interrupt skipping function 1.

Note 2. TITCR2A or TITCR2B is used to enable interrupt skipping function 2.

TITMRA and TITMRB are used to select either of two skipping functions for the TITMRA and TITMRB registers.

## 16.2.39 Timer Interrupt Skipping Set Registers 1 (TITCR1A and TITCR1B)

### (1) MTU.TITCR1A

**Address(es):** MTU.TITMRA H'0_1000_1230

| Bit | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | T3AEN | T3ACOR[2:0] | | | T4VEN | T4VCOR[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 |
|---|---|---|---|---|
| b2 to b0 | T4VCOR[2:0] | All 0 | R/W | TCIV4 Interrupt Skipping Count Setting<br>These bits specify the TCIV4 interrupt skipping count within the range from 0 to 7.*1<br>For details, see **Table 16.56**. |
| b3 | T4VEN | 0 | R/W | T4VEN<br>0: TCIV4 interrupt skipping disabled<br>1: TCIV4 interrupt skipping enabled |
| b6 to b4 | T3ACOR[2:0] | All 0 | R/W | TGIA3 Interrupt Skipping Count Setting<br>These bits specify the TGIA3 interrupt skipping count within the range from 0 to 7.<br>For details, see **Table 16.57**. |
| b7 | T3AEN | 0 | R/W | T3AEN<br>0: TGIA3 interrupt skipping disabled<br>1: TGIA3 interrupt skipping enabled |

Note 1. 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 TITCR1A.T3AEN and TITCR1A.T4VEN bits to 0 to clear the skipping counter (TITCNT1A).

### (2) MTU.TITCR1B

**Address(es):** MTU.TITCR1B H'0_1000_1A30

| Bit | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | T6AEN | T6ACOR[2:0] | | | T7VEN | T7VCOR[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 |
|---|---|---|---|---|
| b2 to b0 | T7VCOR[2:0] | All 0 | R/W | TCIV7 Interrupt Skipping Count Setting<br>These bits specify the TCIV7 interrupt skipping count within the range from 0 to 7.*1<br>For details, see **Table 16.58**. |
| b3 | T7VEN | 0 | R/W | T7VEN<br>0: TCIV7 interrupt skipping disabled<br>1: TCIV7 interrupt skipping enabled |
| b6 to b4 | T6ACOR[2:0] | All 0 | R/W | TGIA6 Interrupt Skipping Count Setting<br>These bits specify the TGIA6 interrupt skipping count within the range from 0 to 7.*1<br>For details, see **Table 16.59**. |
| b7 | T6AEN | 0 | R/W | T6AEN<br>0: TGIA6 interrupt skipping disabled<br>1: TGIA6 interrupt skipping enabled |

Note 1. 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 TITCR1B.T6AEN and TITCR1B.T7VEN bits to 0 to clear the skipping counter (TITCNT1B).

TITCR1A and TITCR1B enable or disable interrupt skipping and specify the interrupt skipping count.

The setting in the TITCR1A (TITCR1B) register is valid only while the TITM bit in TITMRA (TITMRB) is set to 0; when the TITM bit in TITMRA (TITMRB) is set to 1, the setting in TITCR1A (TITCR1B) is cleared.

Table 16.56 Setting of Interrupt Skipping Count by T4VCOR[2:0] Bits

| Bit 2 | Bit 1 | Bit 0 | |
|---|---|---|---|
| T4VCOR2 | T4VCOR1 | T4VCOR0 | Description |
| 0 | 0 | 0 | Does not skip TCIV4 interrupts. |
| | | 1 | Sets the TCIV4 interrupt skipping count to 1. |
| | 1 | 0 | Sets the TCIV4 interrupt skipping count to 2. |
| | | 1 | Sets the TCIV4 interrupt skipping count to 3. |
| 1 | 0 | 0 | Sets the TCIV4 interrupt skipping count to 4. |
| | | 1 | Sets the TCIV4 interrupt skipping count to 5. |
| | 1 | 0 | Sets the TCIV4 interrupt skipping count to 6. |
| | | 1 | Sets the TCIV4 interrupt skipping count to 7. |

Table 16.57 Setting of Interrupt Skipping Count by T3ACOR[2:0] Bits

| Bit 6 | Bit 5 | Bit 4 | |
|---|---|---|---|
| T3ACOR2 | T3ACOR1 | T3ACOR0 | Description |
| 0 | 0 | 0 | Does not skip TGIA3 interrupts. |
| | | 1 | Sets the TGIA3 interrupt skipping count to 1. |
| | 1 | 0 | Sets the TGIA3 interrupt skipping count to 2. |
| | | 1 | Sets the TGIA3 interrupt skipping count to 3. |
| 1 | 0 | 0 | Sets the TGIA3 interrupt skipping count to 4. |
| | | 1 | Sets the TGIA3 interrupt skipping count to 5. |
| | 1 | 0 | Sets the TGIA3 interrupt skipping count to 6. |
| | | 1 | Sets the TGIA3 interrupt skipping count to 7. |

Table 16.58 Setting of Interrupt Skipping Count by T7VCOR[2:0] Bits

| Bit 2 | Bit 1 | Bit 0 | |
|---|---|---|---|
| T7VCOR2 | T7VCOR1 | T7VCOR0 | Description |
| 0 | 0 | 0 | Does not skip TCIV7 interrupts. |
| | | 1 | Sets the TCIV7 interrupt skipping count to 1. |
| | 1 | 0 | Sets the TCIV7 interrupt skipping count to 2. |
| | | 1 | Sets the TCIV7 interrupt skipping count to 3. |
| 1 | 0 | 0 | Sets the TCIV7 interrupt skipping count to 4. |
| | | 1 | Sets the TCIV7 interrupt skipping count to 5. |
| | 1 | 0 | Sets the TCIV7 interrupt skipping count to 6. |
| | | 1 | Sets the TCIV7 interrupt skipping count to 7. |

Table 16.59 Setting of Interrupt Skipping Count by T6ACOR[2:0] Bits

| Bit 6 | Bit 5 | Bit 4 | |
|---|---|---|---|
| T6ACOR2 | T6ACOR1 | T6ACOR0 | Description |
| 0 | 0 | 0 | Does not skip TGIA6 interrupts. |
| | | 1 | Sets the TGIA6 interrupt skipping count to 1. |
| | 1 | 0 | Sets the TGIA6 interrupt skipping count to 2. |
| | | 1 | Sets the TGIA6 interrupt skipping count to 3. |
| 1 | 0 | 0 | Sets the TGIA6 interrupt skipping count to 4. |
| | | 1 | Sets the TGIA6 interrupt skipping count to 5. |
| | 1 | 0 | Sets the TGIA6 interrupt skipping count to 6. |
| | | 1 | Sets the TGIA6 interrupt skipping count to 7. |

## 16.2.40 Timer Interrupt Skipping Counters 1 (TITCNT1A and TITCNT1B)

### (1) MTU.TITCNT1A

**Address(es):** MTU.TITCNT1A H'0_1000_1231

| Bit | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | T3ACNT[2:0] | | | — | T4VCNT[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 |
|---|---|---|---|---|
| b2 to b0 | T4VCNT[2:0] | All 0 | R | TCIV4 Interrupt Counter<br>While the T4VEN bit in TITCR1A is set to 1, the count in these bits is incremented every time a TCIV4 interrupt occurs. |
| b3 | — | 0 | R | Reserved<br>This bit is read as 0. |
| b6 to b4 | T3ACNT[2:0] | All 0 | R | TGIA3 Interrupt Counter<br>While the T3AEN bit in TITCR1A is set to 1, the count in these bits is incremented every time a TGIA3 interrupt occurs. |
| b7 | — | 0 | R | Reserved<br>This bit is read as 0. |

**Note:** To clear the TITCNT1A, clear the TITCR1A.T3AEN and TITCR1A.T4VEN bits to 0.

TITCNT1A and TITCNT1B are 8-bit readable/writable counters. TITCNTA and TITCNTB retain their values even after stopping the count operation of MTU3.TCNT and MTU4.TCNT (MTU6.TCNT and MTU7.TCNT).

#### T4VCNT[2:0] Bits (TCIV4 Interrupt Counter)

[Clearing conditions]

- When the TITM bit in TITMRA is 1
- When the T4VEN bit in TITCR1A is cleared to 0
- When the T4VCOR[2:0] bits in TITCR1A are cleared to 000b
- When the T4VCNT[2:0] bits in TITCNT1A match the T4VCOR[2:0] bits in TITCR1A

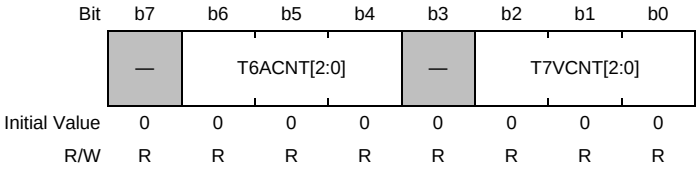
#### T3ACNT[2:0] Bits (TGIA3 Interrupt Counter)

[Clearing conditions]

- When the TITM bit in TITMRA is 1
- When the T3AEN bit in TITCR1A is cleared to 0
- When the T3ACOR[2:0] bits in TITCR1A are cleared to 000b
- When the T3ACNT[2:0] bits in TITCNT1A match the T3ACOR[2:0] bits in TITCR1A

### (2) MTU.TITCNT1B

**Address(es):** MTU.TITCNT1B H'0_1000_1A31




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| b2 to b0 | T7VCNT[2:0] | All 0 | R | TCIV7 Interrupt Counter<br>While the T7VEN bit in TITCR1B is set to 1, the count in these bits is incremented every time a TCIV7 interrupt occurs. |
| b3 | — | 0 | R | Reserved<br>This bit is read as 0. |
| b6 to b4 | T6ACNT[2:0] | All 0 | R | TGIA6 Interrupt Counter<br>While the T6AEN bit in TITCR1B is set to 1, the count in these bits is incremented every time a TGIA6 interrupt occurs. |
| b7 | — | 0 | R | Reserved<br>This bit is read as 0. |

**Note:** To clear the TITCNT1B, clear the TITCR1B.T6AEN and TITCR1B.T7VEN bits to 0.

#### T7VCNT[2:0] Bits (TCIV7 Interrupt Counter)

[Clearing conditions]

- When the TITM bit in TITMRB is 1
- When the T7VEN bit in TITCR1B is cleared to 0
- When the T7VCOR[2:0] bits in TITCR1B are cleared to 000b
- When the T7VCNT[2:0] bits in TITCNT1B match the T7VCOR[2:0] bits in TITCR1B

#### T6ACNT[2:0] Bits (TGIA6 Interrupt Counter)

[Clearing conditions]

- When the TITM bit in TITMRB is 1
- When the T6AEN bit in TITCR1B is cleared to 0
- When the T6ACOR[2:0] bits in TITCR1B are cleared to 000b
- When the T6ACNT[2:0] bits in TITCNT1B match the T6ACOR[2:0] bits in TITCR1B

## 16.2.41 Timer Interrupt Skipping Set Registers 2 (TITCR2A and TITCR2B)

### (1) MTU.TITCR2A

**Address(es):** MTU.TITCR2A H'0_1000_123B

| Bit | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | TRG4COR[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 |
|---|---|---|---|---|
| b2 to b0 | TRG4COR [2:0] | All 0 | R/W | TRG4AN/TRG4BN Interrupt Skipping Count Setting<br>These bits specify the TRG4AN/TRG4BN interrupt skipping count within the range from 0 to 7.<br>For details, see **Table 16.60**. |
| b7 to b3 | — | All 0 | R/W | Reserved<br>These bits are read as 0. The write value should be 0. |

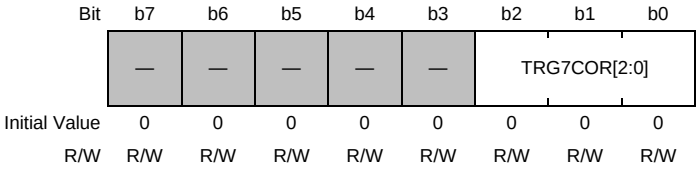
TITCR2A and TITCR2B specify the interrupt skipping count for TRG4AN and TRG4BN (TRG7AN and TRG7BN). This setting is valid only while TITMRA or TITMRB is set to 1.

Table 16.60 Setting of Interrupt Skipping Count by TRG4COR[2:0] Bits

| Bit 2 | Bit 1 | Bit 0 | |
|---|---|---|---|
| TRG4COR2 | TRG4COR1 | TRG4COR0 | Description |
| 0 | 0 | 0 | Does not skip TRG4AN and TRG4BN interrupts. |
| | | 1 | Sets the TRG4AN and TRG4BN interrupt skipping count to 1. |
| | 1 | 0 | Sets the TRG4AN and TRG4BN interrupt skipping count to 2. |
| | | 1 | Sets the TRG4AN and TRG4BN interrupt skipping count to 3. |
| 1 | 0 | 0 | Sets the TRG4AN and TRG4BN interrupt skipping count to 4. |
| | | 1 | Sets the TRG4AN and TRG4BN interrupt skipping count to 5. |
| | 1 | 0 | Sets the TRG4AN and TRG4BN interrupt skipping count to 6. |
| | | 1 | Sets the TRG4AN and TRG4BN interrupt skipping count to 7. |

### (2) MTU.TITCR2B

**Address(es):** MTU.TITCR2B H'0_1000_1A3B




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| b2 to b0 | TRG7COR [2:0] | All 0 | R/W | TRG7AN/TRG7BN Interrupt Skipping Count Setting<br>These bits specify the TRG7AN/TRG7BN interrupt skipping count within the range from 0 to 7.<br>For details, see **Table 16.61**. |
| b7 to b3 | — | All 0 | R/W | Reserved<br>These bits are read as 0. The write value should be 0. |

Table 16.61 Setting of Interrupt Skipping Count by TRG7COR[2:0] Bits

| Bit 2 | Bit 1 | Bit 0 | |
|---|---|---|---|
| TRG7COR2 | TRG7COR1 | TRG7COR0 | Description |
| 0 | 0 | 0 | Does not skip TRG7AN and TRG7BN interrupts. |
| | | 1 | Sets the TRG7AN and TRG7BN interrupt skipping count to 1. |
| | 1 | 0 | Sets the TRG7AN and TRG7BN interrupt skipping count to 2. |
| | | 1 | Sets the TRG7AN and TRG7BN interrupt skipping count to 3. |
| 1 | 0 | 0 | Sets the TRG7AN and TRG7BN interrupt skipping count to 4. |
| | | 1 | Sets the TRG7AN and TRG7BN interrupt skipping count to 5. |
| | 1 | 0 | Sets the TRG7AN and TRG7BN interrupt skipping count to 6. |
| | | 1 | Sets the TRG7AN and TRG7BN interrupt skipping count to 7. |

## 16.2.42 Timer Interrupt Skipping Counters 2 (TITCNT2A and TITCNT2B)

### (1) MTU.TITCNT2A

**Address(es):** MTU.TITCNT2A H'0_1000_123C

| Bit | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | TRG4CNT[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 |
|---|---|---|---|---|
| b2 to b0 | TRG4CNT [2:0] | All 0 | R | TRG4AN/TRG4BN Interrupt Counter<br>These bits start counting from the value set in TRG4COR[2:0] and the count decrements every time TRG4AN or TRG4BN is generated. When the count reaches 0 and is reloaded, the TRG4AN and TRG4BN interrupts become valid. |
| b7 to b3 | — | All 0 | R | Reserved<br>These bits are read as 0. |

TITCNT2A and TITCNT2B start counting from the values set in the TRG4COR[2:0] and TRG7COR[2:0] bits and the count decrements every time TRG4AN or TRG4BN (TITCNT2A) is generated or TRG7AN or TRG7BN (TITCNT2B) is generated. When the count reaches 0 and is reloaded, the TRG4AN and TRG4BN interrupts or the TRG7AN and TRG7BN interrupts become valid.

#### TRG4CNT[2:0] Bits (TRG4AN/TRG4BN Interrupt Counter)

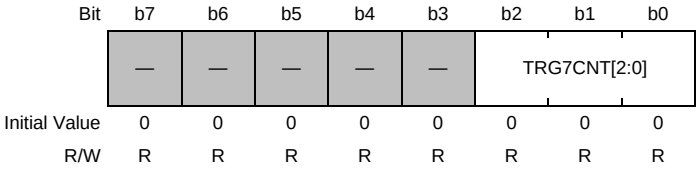
These bits start counting from the value set in the TRG4COR[2:0] bits and the count decrements every time a TRG4AN or TRG4BN interrupt is generated. When the count reaches 0 and is reloaded, the TRG4AN and TRG4BN interrupts become valid.

[Clearing conditions]

- When the TITM bit in TITMRA is 0
- When the TRG4COR[2:0] bits in TITCR2A are cleared to 000b
- When the count of TRG4AN and TRG4BN occurrence matches the TRG4COR[2:0] value in TITCR2A

### (2) MTU.TITCNT2B

**Address(es):** MTU.TITCNT2B H'0_1000_1A3C




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| b2 to b0 | TRG7CNT [2:0] | All 0 | R | TRG7AN/TRG7BN Interrupt Counter<br>These bits start counting from the value set in TRG7COR[2:0] and the count decrements every time TRG7AN or TRG7BN is generated. When the count reaches 0 and is reloaded, the TRG7AN and TRG7BN interrupts become valid. |
| b7 to b3 | — | All 0 | R | Reserved<br>These bits are read as 0. |

#### TRG7CNT[2:0] Bits (TRG7AN/TRG7BN Interrupt Counter)

These bits start counting from the value set in the TRG7COR[2:0] bits and the count decrements every time a TRG7AN or TRG7BN interrupt is generated. When the count reaches 0 and is reloaded, the TRG7AN and TRG7BN interrupts become valid.

[Clearing conditions]

- When the TITM bit in TITMRB is 0
- When the TRG7COR[2:0] bits in TITCR2B are cleared to 000b
- When the count of TRG7AN and TRG7BN occurrence matches the TRG7COR[2:0] value in TITCR2B

### 16.2.43 Bus Master Interface

The timer counter (MTU8.TCNT), general registers (MTU8.TGRn) for MTU8, and MTU1.TCNTLW, MTU1.TGRALW, and MTU1.TGRBLW registers when TMDR3.LWA = 1 are 32-bit registers. A 32-bit data bus to the bus master enables 32-bit read/write access. 8- and 16-bit read/write are not allowed. Access these registers in 32-bit units.

Excluding MTU8, the timer counters (MTU0.TCNT to MTU7.TCNT), general registers (MTU0.TGRn to MTU7.TGRn), timer subcounters (TCNTSA and TCNTSB), timer cycle buffer registers (TCBRA and TCBRB), timer dead time data registers (TDDRA and TDDRB), timer cycle data registers (TCDRA and TCDRB), timer A/D converter start request control registers (MTU4.TADCR and MTU7.TADCR), timer A/D converter start request cycle set registers (MTU4.TADCORA, MTU4.TADCORB, MTU7.TADCORA, and MTU7.TADCORB), and timer A/D converter start request cycle set buffer registers (MTU4.TADCOBRA, MTU4.TADCOBRB, MTU7.TADCOBRA, and MTU7.TADCOBRB) are 16-bit registers. A 16-bit data bus to the bus master enables 16-bit read/write access. 8-bit read/ write is not allowed. Access the registers in 16-bit units.

All registers other than the above registers are 8-bit registers. Read from/write to these registers in 8-bit units.

# 16.3 Operation

## 16.3.1 Basic Functions

Each channel has TCNT and TGR. TCNT in each channel performs up-counting and is also capable of free-running count, periodic count, and external event count operations.

Each TGR can be used as an input capture register or an output compare register.

### (1) Counter Operation

When one of bits CST0 to CST4 and CST8 in TSTRA, bits CST6 and CST7 in TSTRB, and bits CSTU5, CSTV5, and CSTW5 in MTU5.TSTR is set to 1, TCNT for the corresponding channel begins counting. TCNT can perform various count operations such as free-running count or periodic count operations.

#### (a) Example of Count Operation Setting Procedure

**Figure 16.5** shows an example of the count operation setting procedure.

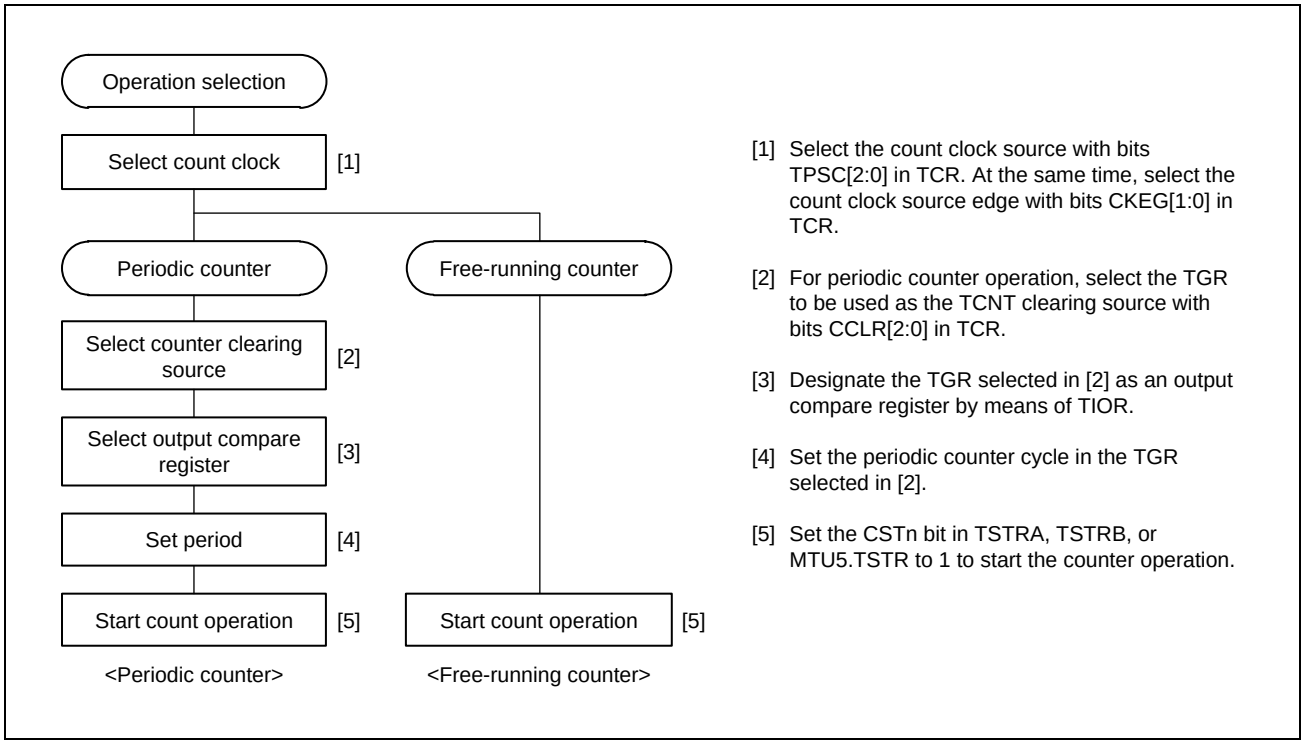



Figure 16.5 Example of Count Operation Setting Procedure

### (b) Free-Running Count Operation and Periodic Count Operation

Immediately after a reset, all TCNT counters are designated as free-running counters. When the CSTn bit in TSTRA, TSTRB, or TSTR in MTU5 is set to 1, the corresponding TCNT counter starts up-counting as a free-running counter. When TCNT overflows (from H'FFFF to H'0000), if the corresponding TIER.TCIEV bit is 1, an interrupt request is issued to the CPU. After an overflow, TCNT starts counting up again from H'0000.

**Figure 16.6** illustrates free-running counter operation.

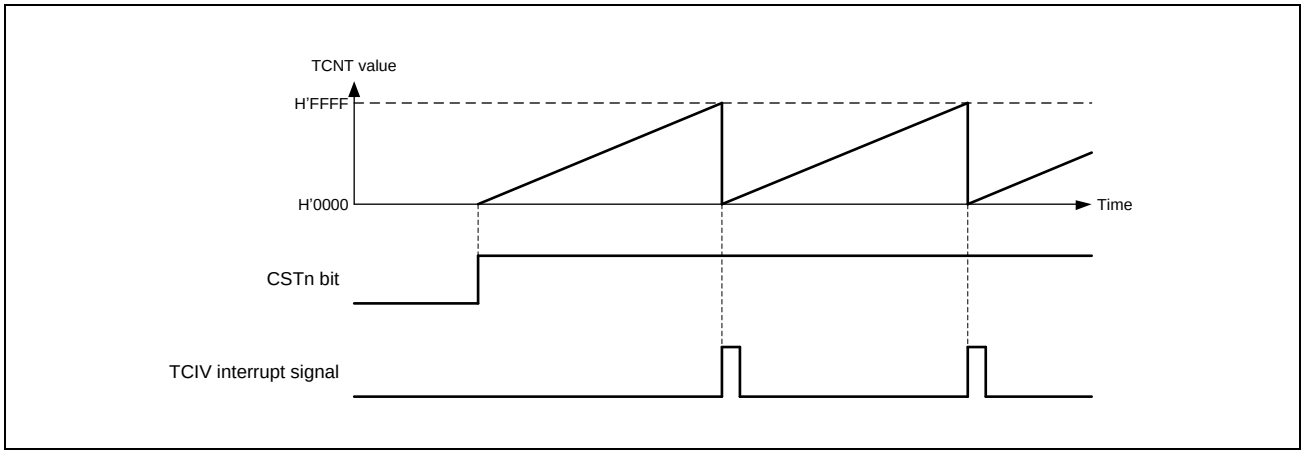



Figure 16.6 Free-Running Counter Operation

When compare match is selected as the TCNT clearing source, TCNT for the relevant channel performs periodic count operation. TGR for setting the cycle is designated as an output compare register, and counter clearing by compare match is selected by means of bits CCLR[2:0] in TCR. After the settings have been made, TCNT starts up-counting as a periodic counter when the corresponding CSTn bit in TSTRA, TSTRB, or TSTR in MTU5 is set to 1. When the count matches the value in TGR, TCNT is cleared to H'0000.

If the value of the corresponding TIER.TGIE bit is 1 at this point, an interrupt request is issued to the CPU. After a compare match, TCNT starts counting up again from H'0000.

**Figure 16.7** illustrates periodic counter operation.

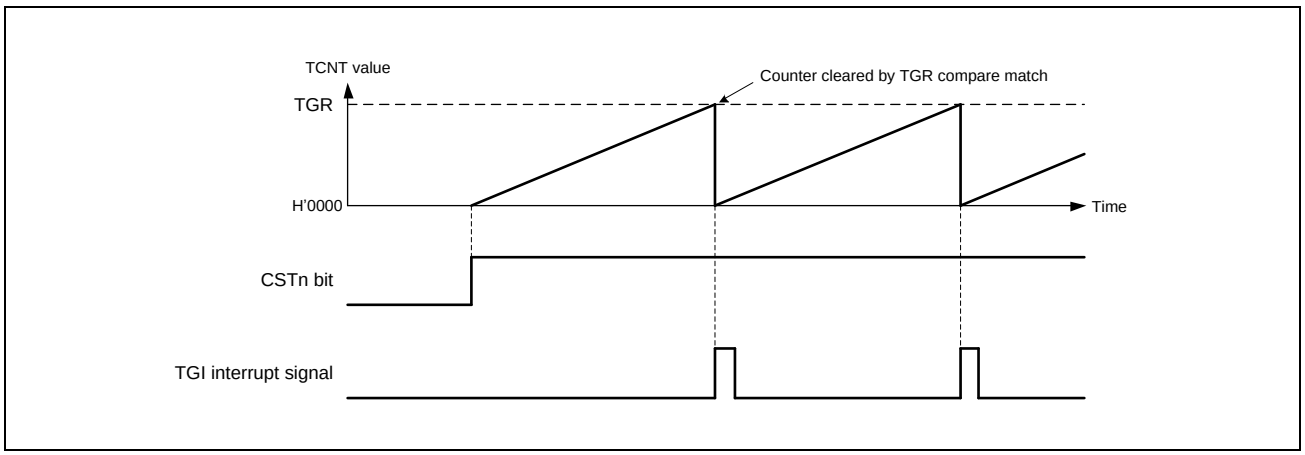



Figure 16.7 Periodic Counter Operation

### (2) Waveform Output by Compare Match

Upon compare match, low, high, or toggle output from the corresponding pin can be performed. The compare match output operation is not available in MTU5.

#### (a) Example of Procedure for Setting Waveform Output by Compare Match

**Figure 16.8** shows an example of the procedure for setting waveform output by compare match

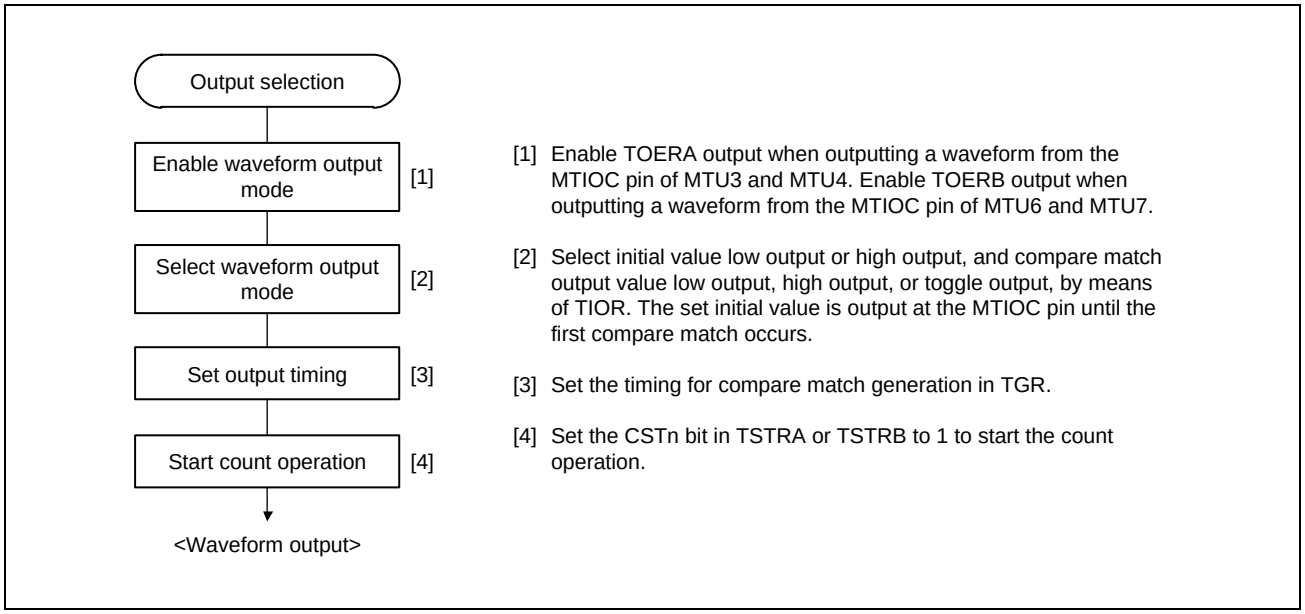



Figure 16.8 Example of Procedure for Setting Waveform Output by Compare Match

### (b) Examples of Waveform Output Operation

**Figure 16.9** shows an example of low output and high output.

In this example, TCNT has been designated as a free-running counter, and settings have been made so that high is output by compare match A and low is output by compare match B. When the pin level is the same as the specified level, the pin level does not change.

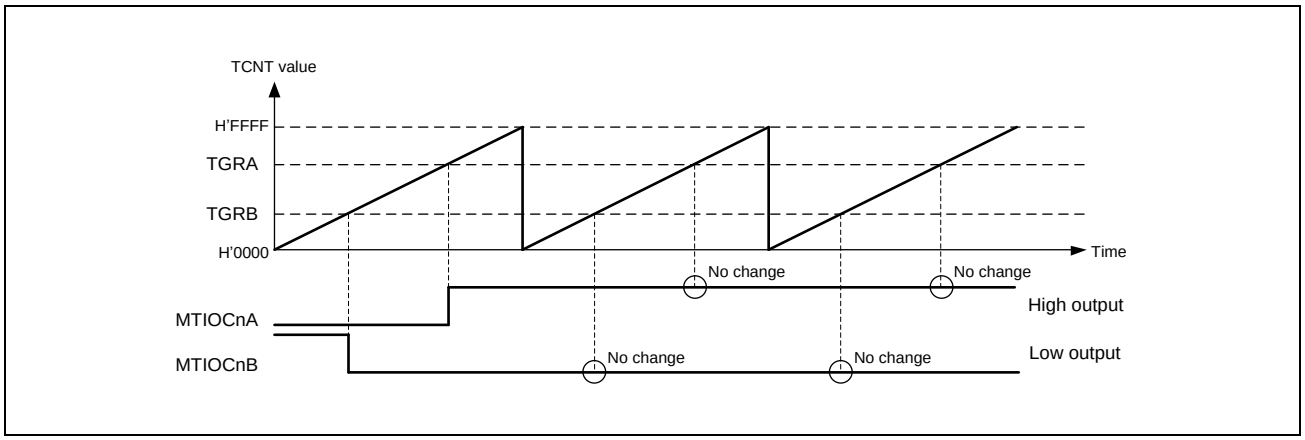



Figure 16.9 Example of low output and high output Operation (n = 0 to 4, 6, 7, 8)

**Figure 16.10** 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 so that the output is toggled by both compare match A and compare match B.

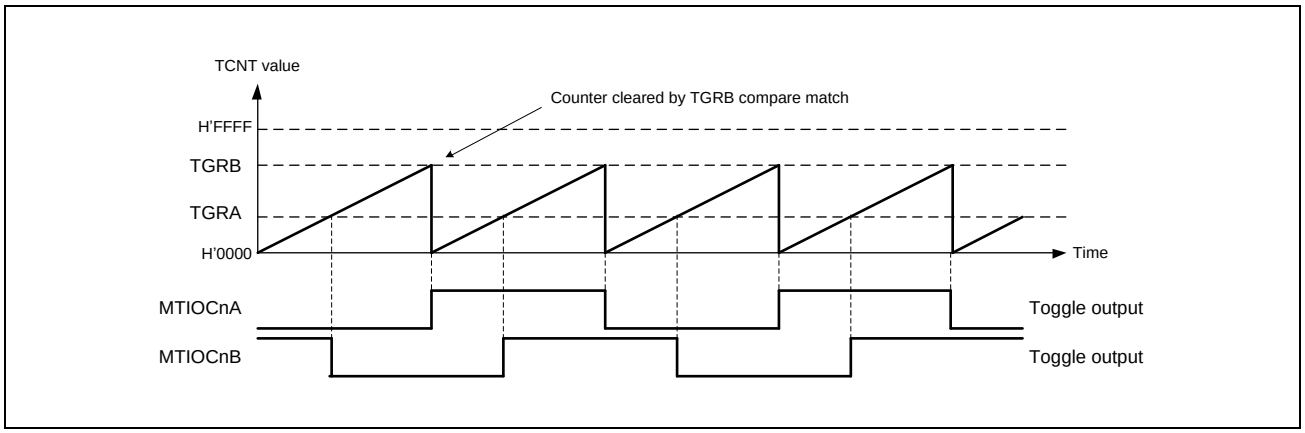



Figure 16.10 Example of Toggle Output Operation (n = 0 to 4, 6, 7, 8)

### (3) Input Capture Function

The TCNT value can be transferred to TGR on detection of the MTIOCnm, MTIC5U, MTIC5V, or MTIC5W pin (n = 0 to 4, 6, 7, 8; m = A to D), input edge.

The rising edge, falling edge, or both edges can be selected as the detection edge. For MTU0 and MTU1, another channel's count clock or compare match signal can also be specified as the input capture source.

NOTE

When another channel's count clock is used as the input capture input for MTU0 and MTU1, P0ϕ/1 should not be selected as the count clock used for input capture input. Input capture will not be generated if P0ϕ/1 is selected.

#### (a) Example of Input Capture Operation Setting Procedure

**Figure 16.11** shows an example of the input capture operation setting procedure.

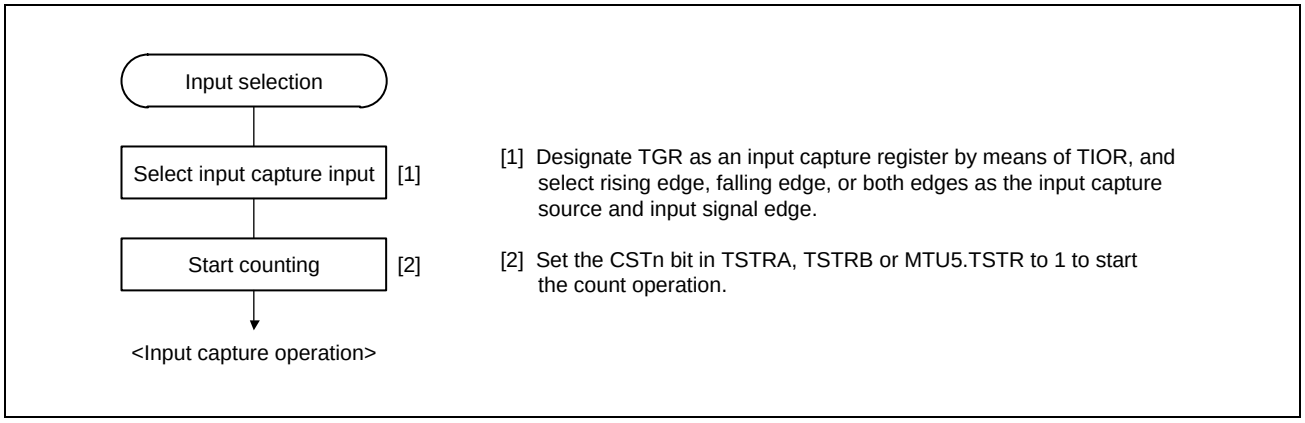



Figure 16.11 Example of Input Capture Operation Setting Procedure

### (b) Example of Input Capture Operation

**Figure 16.12** shows an example of input capture operation.

In this example, both rising and falling edges have been selected as the MTIOCnA pin input capture input edge, the falling edge has been selected as the MTIOCnB pin input capture input edge, and counter clearing by TGRB input capture has been designated for TCNT. (n = 0 to 4, 6, 7, 8)

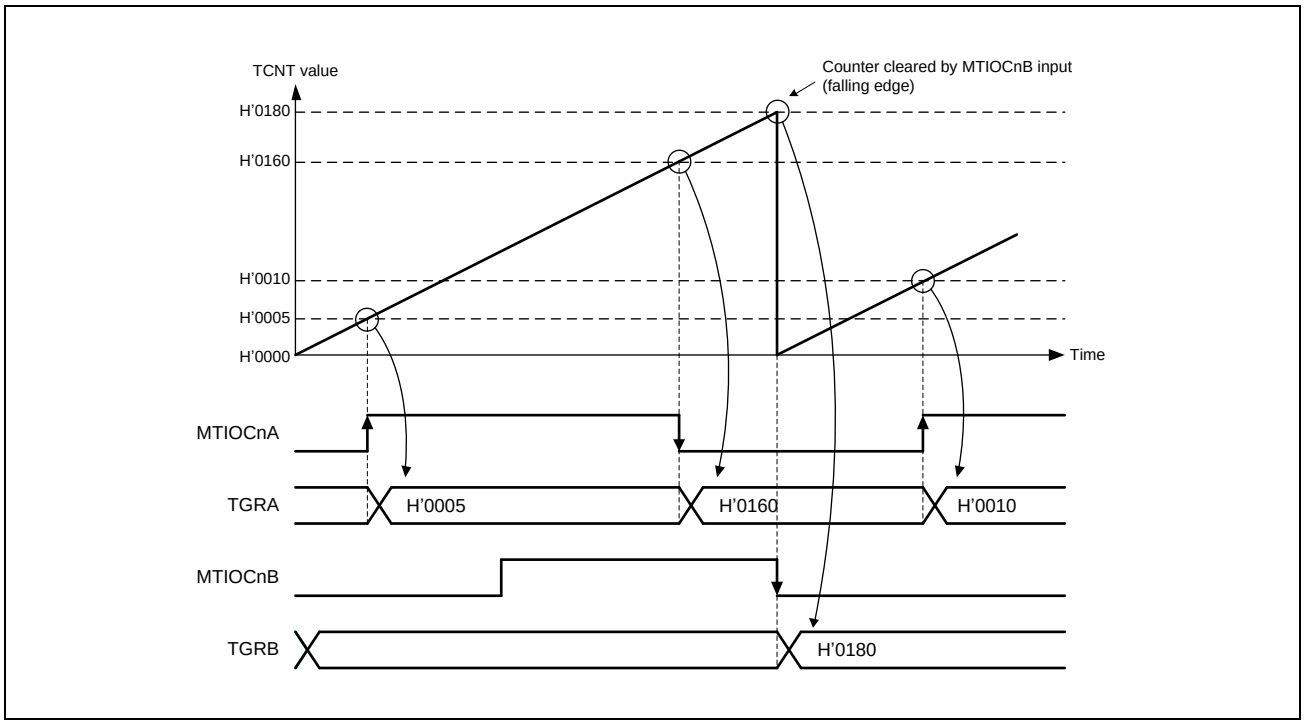



Figure 16.12 Example of Input Capture Operation (n = 0 to 4, 6, 7, 8)

## 16.3.2 Synchronous Operation

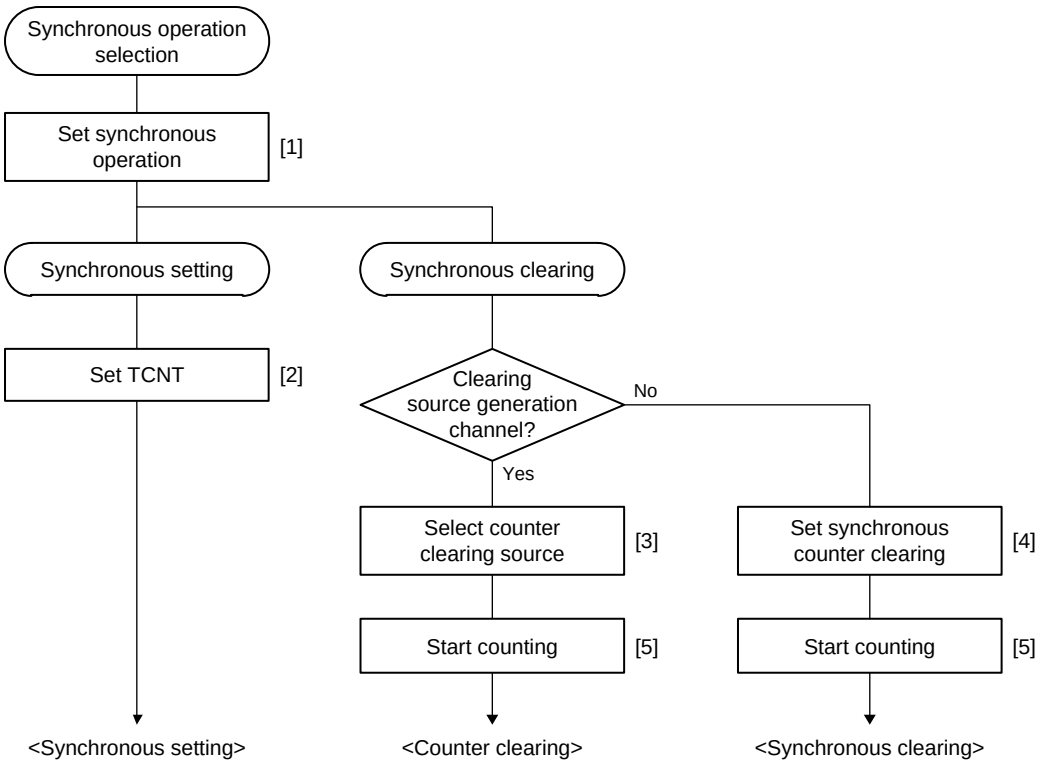
In synchronous operation, the values in multiple TCNT counters can be modified simultaneously (synchronous setting). In addition, multiple TCNT counters can be cleared simultaneously (synchronous clearing) by making the appropriate setting in TCR.

Synchronous operation can increase the number of TGR registers assigned to the same time base.

MTU0 to MTU4, MTU6, and MTU7 can all be designated for synchronous operation. MTU5 and MTU8 cannot be used for synchronous operation.

### (1) Example of Synchronous Operation Setting Procedure

**Figure 16.13** shows an example of the synchronous operation setting procedure.




[1] Set to 1 in the SYNC bits in TSYRA and TSYRB 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 the CCLR[2:0] bits in TCR to specify TCNT clearing by input capture/output compare, etc.
[4] Use the CCLR[2:0] bits in TCR to designate synchronous clearing for the counter clearing source.
[5] Set to 1 in the CSTn bits in TSTRA and TSTRB for the relevant channels, to start the count operation.

Figure 16.13 Example of Synchronous Operation Setting Procedure

### (2) Example of Synchronous Operation

**Figure 16.14** shows an example of synchronous operation.

In this example, synchronous operation and PWM mode 1 have been designated for MTU0 to MTU 2, MTU0.TGRB compare match has been set as the counter clearing source in MTU0, and synchronous clearing has been set for the counter clearing source in MTU1 and MTU2.

Three-phase PWM waveforms are output from pins MTIOC0A, MTIOC1A, and MTIOC2A. At this time, synchronous setting and synchronous clearing by MTU0.TGRB compare match are performed for the TCNT counters in MTU0 to MTU2, and the data set in MTU0.TGRB is used as the PWM cycle.

For details of PWM modes, see **Section 16.3.5, PWM Modes**.

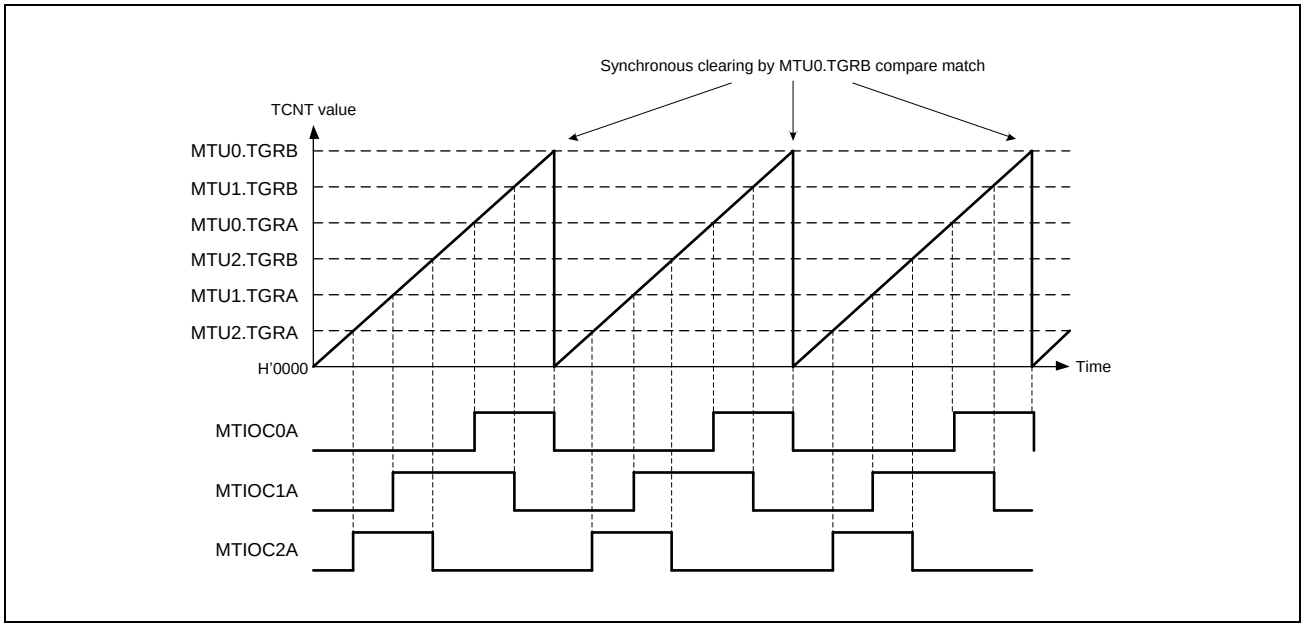



Figure 16.14 Example of Synchronous Operation

### 16.3.3 Buffer Operation

Buffer operation, provided for MTU0, MTU3, MTU4, MTU6, MTU7, and MTU8, enables TGRC and TGRD to be used as buffer registers. In MTU0, 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

MTU0.TGRE cannot be designated as an input capture register and can only operate as a compare match register.

**Table 16.62** shows the register combinations used in buffer operation.

Table 16.62 Register Combinations in Buffer Operation

| Channel | Timer General Register | Buffer Register |
|---|---|---|
| MTU0 | TGRA | TGRC |
| | TGRB | TGRD |
| | TGRE | TGRF |
| MTU3 | TGRA | TGRC |
| | TGRB | TGRD |
| MTU4 | TGRA | TGRC |
| | TGRB | TGRD |
| MTU6 | TGRA | TGRC |
| | TGRB | TGRD |
| MTU7 | TGRA | TGRC |
| | TGRB | TGRD |
| MTU8 | TGRA | TGRC |
| | TGRB | TGRD |

#### 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 16.15**.

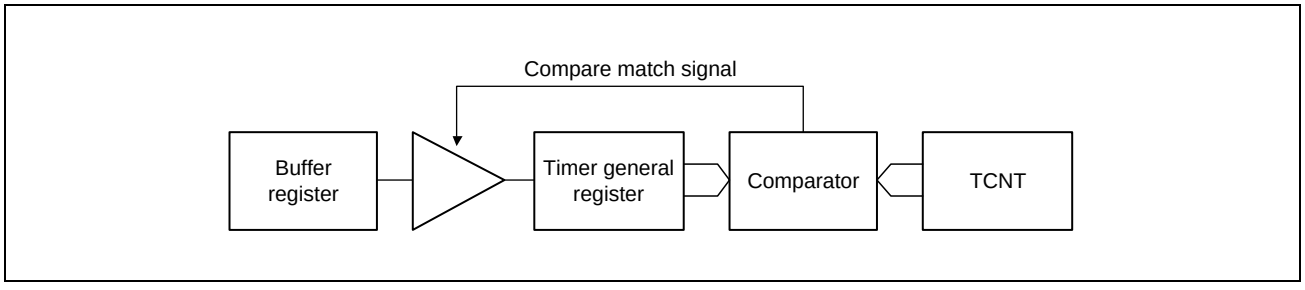



Figure 16.15 Compare Match Buffer Operation

**When TGR is an input capture register**

When an input capture occurs, the value in TCNT is transferred to TGR and the value previously held in TGR is transferred to the buffer register.

This operation is illustrated in **Figure 16.16**.

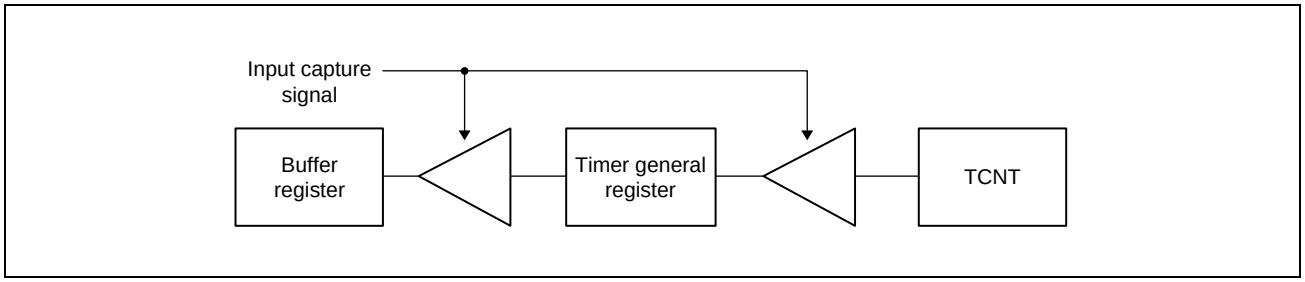



Figure 16.16 Input Capture Buffer Operation

### (1) Example of Buffer Operation Setting Procedure

**Figure 16.17** shows an example of the buffer operation setting procedure.

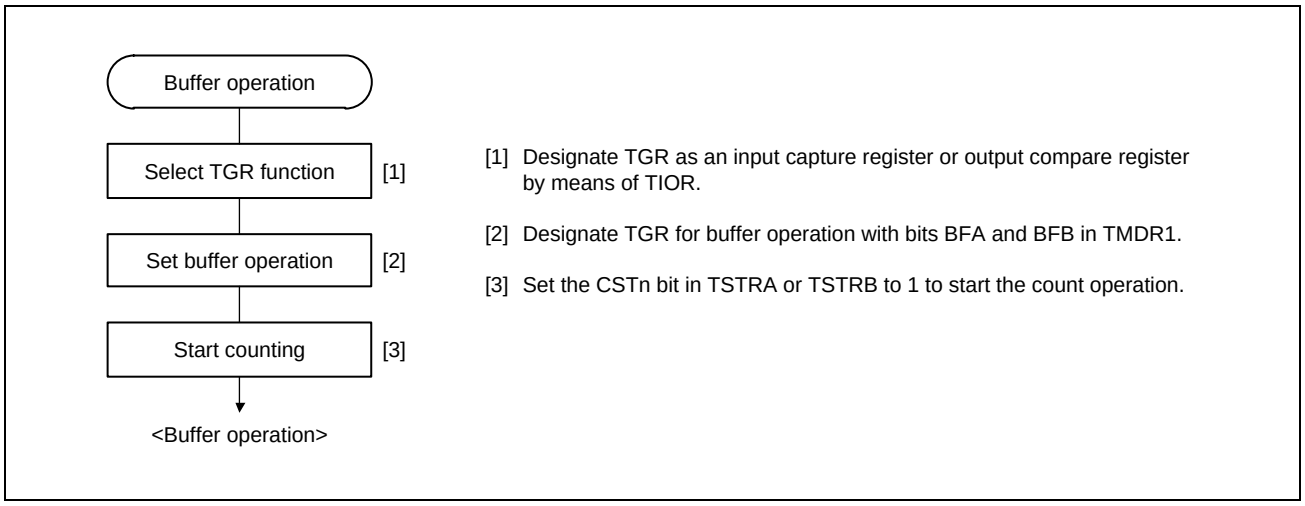



Figure 16.17 Example of Buffer Operation Setting Procedure

### (2) Examples of Buffer Operation

#### (a) When TGR is an Output Compare Register

**Figure 16.18** shows an operation example in which PWM mode 1 has been designated for MTU0, and buffer operation has been designated for TGRA and TGRC. The settings used in this example are TCNT clearing by compare match B, high output at compare match A, and low 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 compare match A occurs.

For details of PWM modes, see **Section 16.3.5, PWM Modes**.

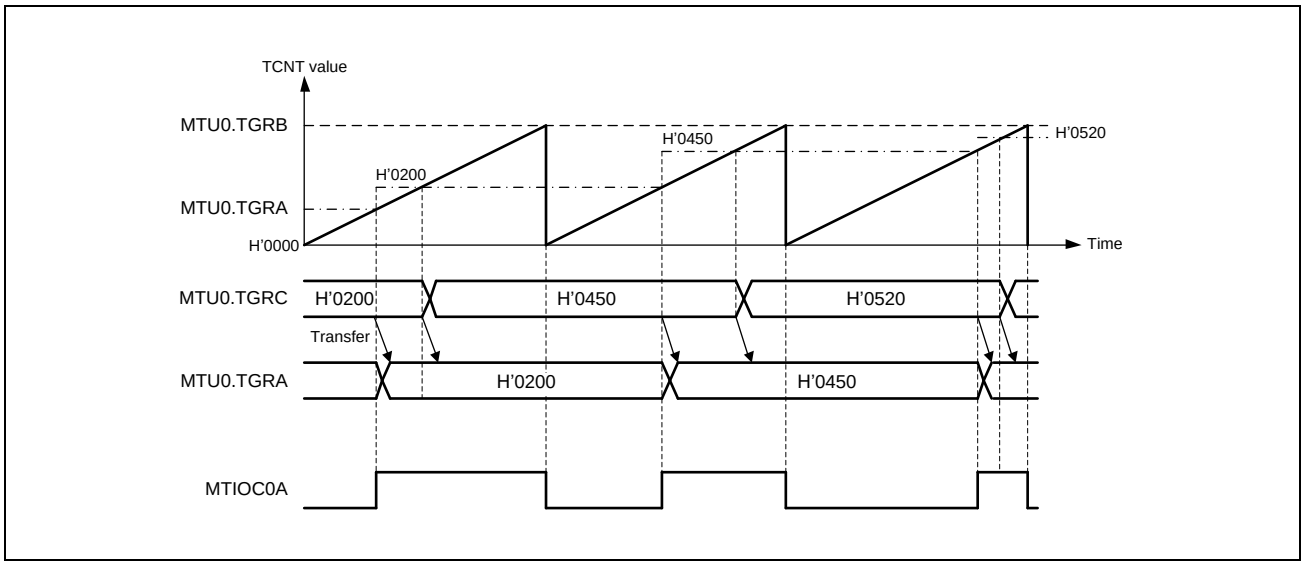



Figure 16.18 Example of Buffer Operation (1)

### (b) When TGR is an Input Capture Register

**Figure 16.19** 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 MTIOCnA pin input capture input edge. (n = 0 to 4, 6, 7, 8)

As buffer operation has been set, when the TCNT value is transferred to TGRA upon occurrence of input capture A, the value previously stored in TGRA is simultaneously transferred to TGRC.

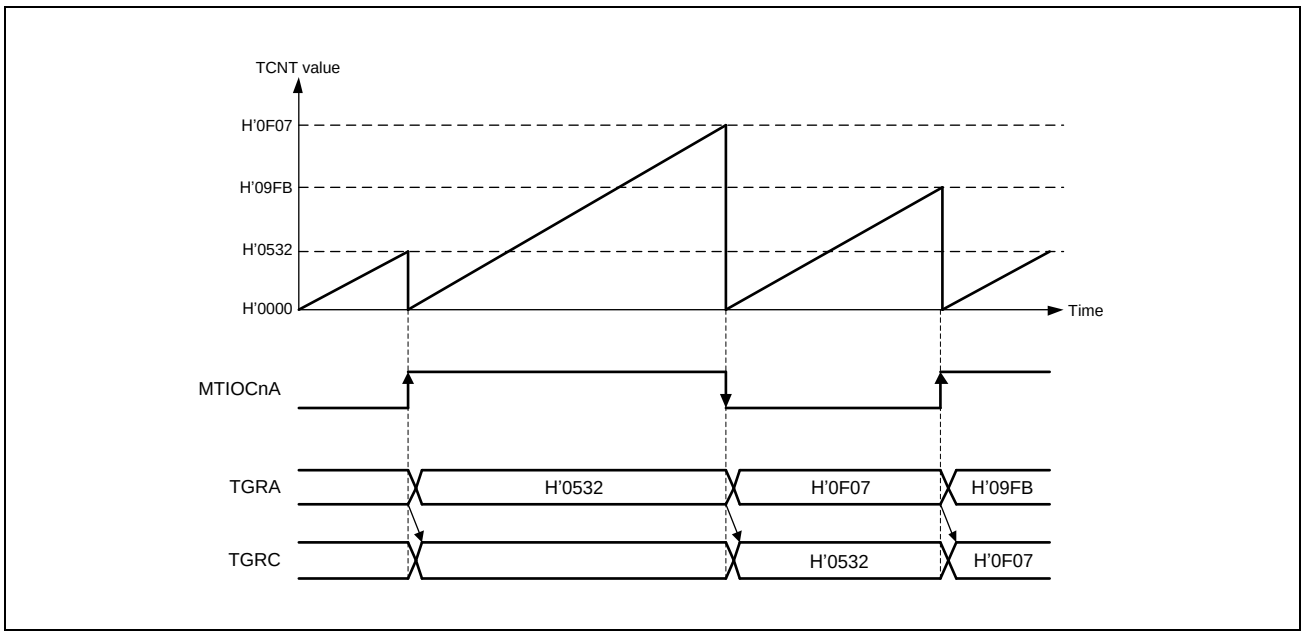



Figure 16.19 Example of Buffer Operation (2) (n = 0 to 4, 6, 7, 8)

### (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 MTU0 or in PWM mode 1 for MTU3, MTU4, MTU6, and MTU7 by setting the buffer operation transfer mode registers (MTUn.TBTM (n = 0, 3, 4, 6, 7)). Either compare match (value after reset) 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 CCLR[2:0] bits in TCR

NOTE

TBTM must be modified only while TCNT stops.

**Figure 16.20** shows an operation example in which PWM mode 1 is designated for MTU0 and buffer operation is designated for MTU0.TGRA and MTU0.TGRC. The settings used in this example are MTU0.TCNT clearing by compare match B, high output at compare match A, and low output at compare match B. The TTSA bit in MTU0.TBTM is set to 1.

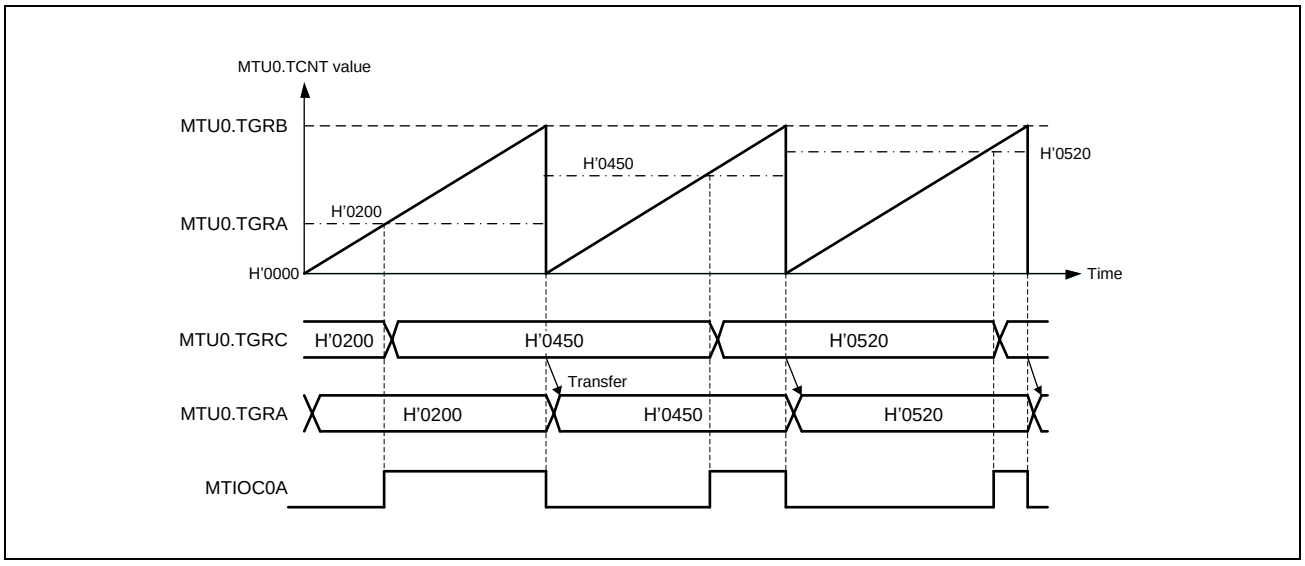



Figure 16.20 Example of Buffer Operation When MTU0.TCNT Clearing is Selected for MTU0.TGRC to MTU0.TGRA Transfer Timing

### 16.3.4 Cascaded Operation

In cascaded operation, two 16-bit counters in different channels are used together as a 32-bit counter.

There are two functions for connecting MTU1 and MTU2 to use as a 32-bit counter: cascade connection to be set when the MTU1.TMDR3.LWA bit is 0, and cascade connection 32-bit phase counting mode to be set when the MTU1.TMDR3.LWA bit is 1. For details on cascade connection 32-bit phase counting mode, refer to **Section 16.3.6.2, Cascade Connection 32-Bit Phase Counting Mode**. This section describes the cascade connection function to be set when the MTU1.TMDR3.LWA bit is 0.

This function works when the LWA bit of TMDR3 in MTU1 is set to 0 and the TPSC[2:0] bits of TCR in MTU1 are set so that TCNT in MTU1 counts at an overflow/underflow of TCNT in MTU2.

Underflow occurs only when the MTU2 to which the lower 16 bits allocated is in phase counting mode.

**Table 16.63** shows the register combinations used in cascaded operation.

NOTE

When phase counting mode is set for MTU1, the count clock setting is invalid and the counters operate independently in phase counting mode.

Table 16.63 Cascaded Combinations

| Combination | Upper 16 Bits | Lower 16 Bits |
|---|---|---|
| MTU1 and MTU2 | MTU1.TCNT | MTU2.TCNT |

For simultaneous input capture of MTU1.TCNT and MTU2.TCNT during cascaded operation, additional input capture input pins can be specified by the input capture control register (TICCR). The input-capture condition is of edges in the signal produced by taking the logical OR of the input level on the main input pin and the input level on the added input pin. Accordingly, if either is at the high level, a change in the level of the other will not produce an edge for detection. For details, see **Section 16.3.4(4), Cascaded Operation Example (c)**. For input capture in cascade connection, refer to **Section 16.6.20, Simultaneous Input Capture in MTU1.TCNT and MTU2.TCNT in Cascade Connection**.

**Table 16.64** shows the TICCR setting and input capture input pins.

Table 16.64 TICCR Setting and Input Capture Input Pins

| Target Input Capture | TICCR Setting | Input Capture Input Pin |
|---|---|---|
| Input capture from MTU1.TCNT to MTU1.TGRA | I2AE bit = 0 (Initial value) | MTIOC1A |
| | I2AE bit = 1 | MTIOC1A, MTIOC2A |
| Input capture from MTU1.TCNT to MTU1.TGRB | I2BE bit = 0 (Initial value) | MTIOC1B |
| | I2BE bit = 1 | MTIOC1B, MTIOC2B |
| Input capture from MTU2.TCNT to MTU2.TGRA | I1AE bit = 0 (Initial value) | MTIOC2A |
| | I1AE bit = 1 | MTIOC2A, MTIOC1A |
| Input capture from MTU2.TCNT to MTU2.TGRB | I1BE bit = 0 (Initial value) | MTIOC2B |
| | I1BE bit = 1 | MTIOC2B, MTIOC1B |

### (1) Example of Cascaded Operation Setting Procedure

**Figure 16.21** shows an example of the cascaded operation setting procedure.

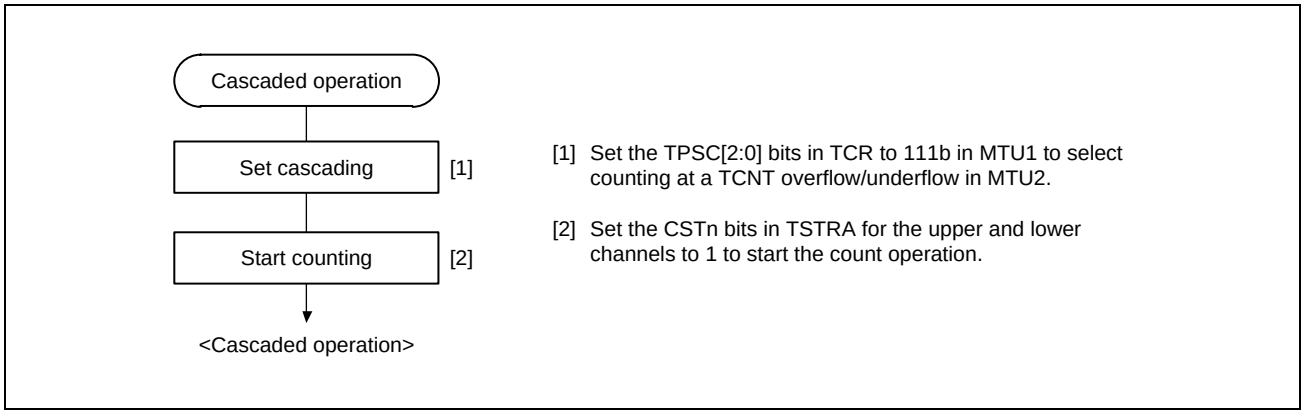



Figure 16.21 Cascaded Operation Setting Procedure

### (2) Cascaded Operation Example (a)

**Figure 16.22** shows the operation when MTU1.TCNT is set for counting at MTU2.TCNT overflow/underflow and MTU2 is set for phase counting mode 1 while MTU1.TCNT and MTU2.TCNT are cascaded.

MTU1.TCNT is incremented by MTU2.TCNT overflow and decremented by MTU2.TCNT underflow.

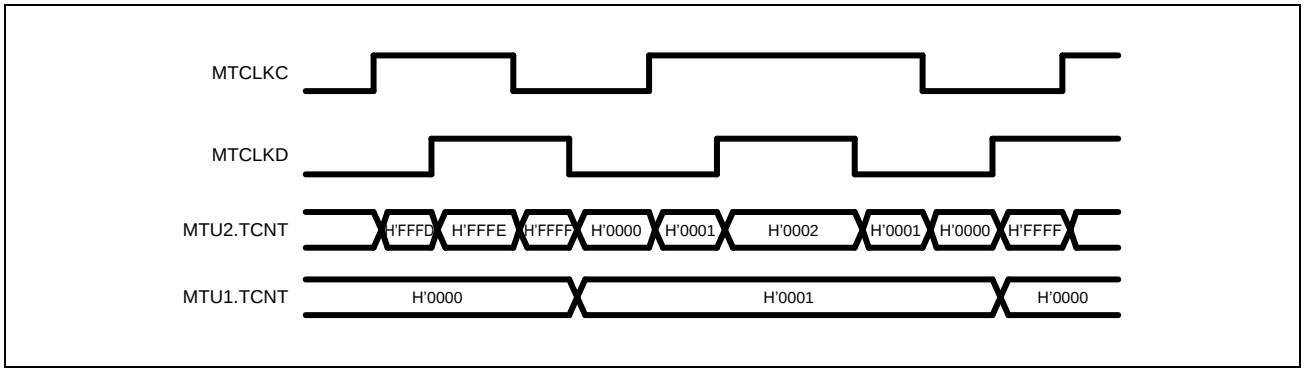



Figure 16.22 Cascaded Operation Example (a)

### (3) Cascaded Operation Example (b)

**Figure 16.23** illustrates the operation when MTU1.TCNT and MTU2.TCNT have been cascaded and the I2AE bit in TICCR has been set to 1 to include the MTIOC2A pin in the MTU1.TGRA input capture conditions. In this example, the MTU1.TIOR.IOA[3:0] bits have selected the MTIOC1A rising edge for the input capture timing while the MTU2.TIOR.IOA[3:0] bits have selected the MTIOC2A rising edge for the input capture timing.

Under these conditions, the rising edge of both MTIOC1A and MTIOC2A is used for the MTU1.TGRA input capture condition. For the MTU2.TGRA input capture condition, the MTIOC2A rising edge is used.

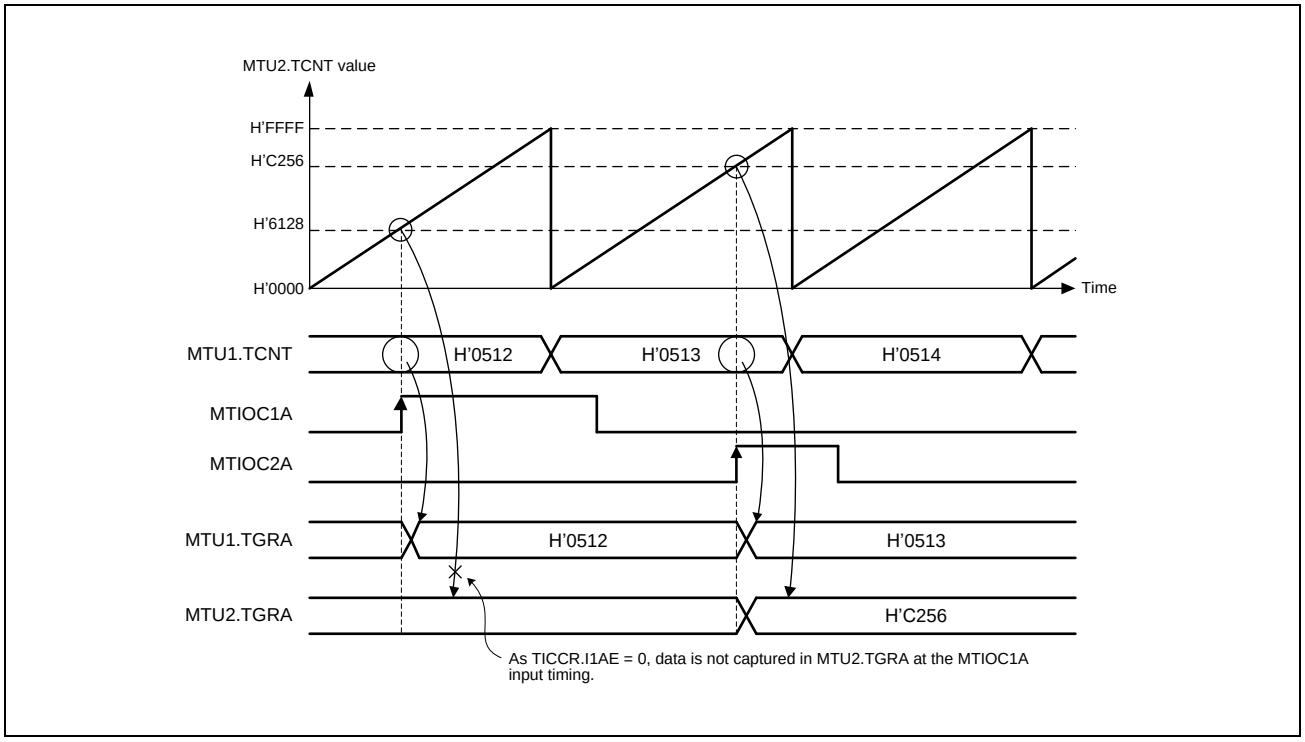
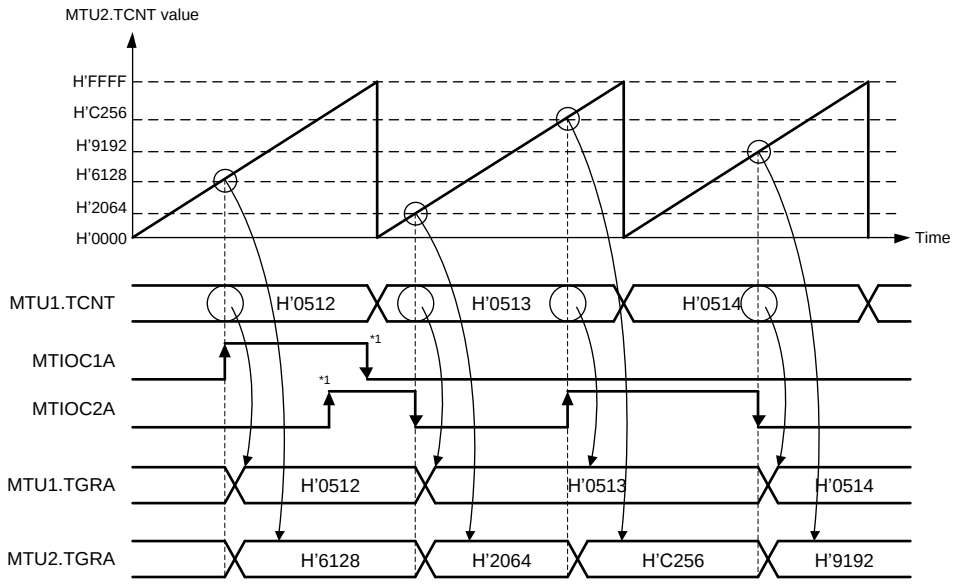



Figure 16.23 Cascaded Operation Example (b)

### (4) Cascaded Operation Example (c)

**Figure 16.24** illustrates the operation when MTU1.TCNT and MTU2.TCNT have been cascaded and the TICCR.I2AE and I1AE bits have been set to 1 to include the MTIOC2A and MTIOC1A pins in the MTU1.TGRA and MTU2.TGRA input capture conditions, respectively. In this example, the IOA[3:0] bits in both MTU1.TIOR and MTU2.TIOR have selected both the rising and falling edges for the input capture timing. Under these conditions, the ORed result of MTIOC1A and MTIOC2A input is used for the MTU1.TGRA and MTU2.TGRA input capture conditions.




Note 1. When either of the input signals is driven high, the selected edge of the other cannot act as the input capture condition.

Figure 16.24 Cascaded Operation Example (c)

### (5) Cascaded Operation Example (d)

**Figure 16.25** illustrates the operation when MTU1.TCNT and MTU2.TCNT have been cascaded and the TICCR.I2AE bit has been set to 1 to include the MTIOC2A pin in the MTU1.TGRA input capture conditions. In this example, the IOA[3:0] bits in MTU1.TIOR have selected occurrence of MTU0.TGRA compare match or input capture for the input capture timing while the IOA[3:0] bits in MTU2.TIOR have selected the MTIOC2A rising edge for the input capture timing.

Under these conditions, as MTU1.TIOR has selected occurrence of MTU0.TGRA compare match or input capture for the input capture timing, the MTIOC2A edge is not used for MTU1.TGRA input capture condition although the I2AE bit in TICCR has been set to 1.

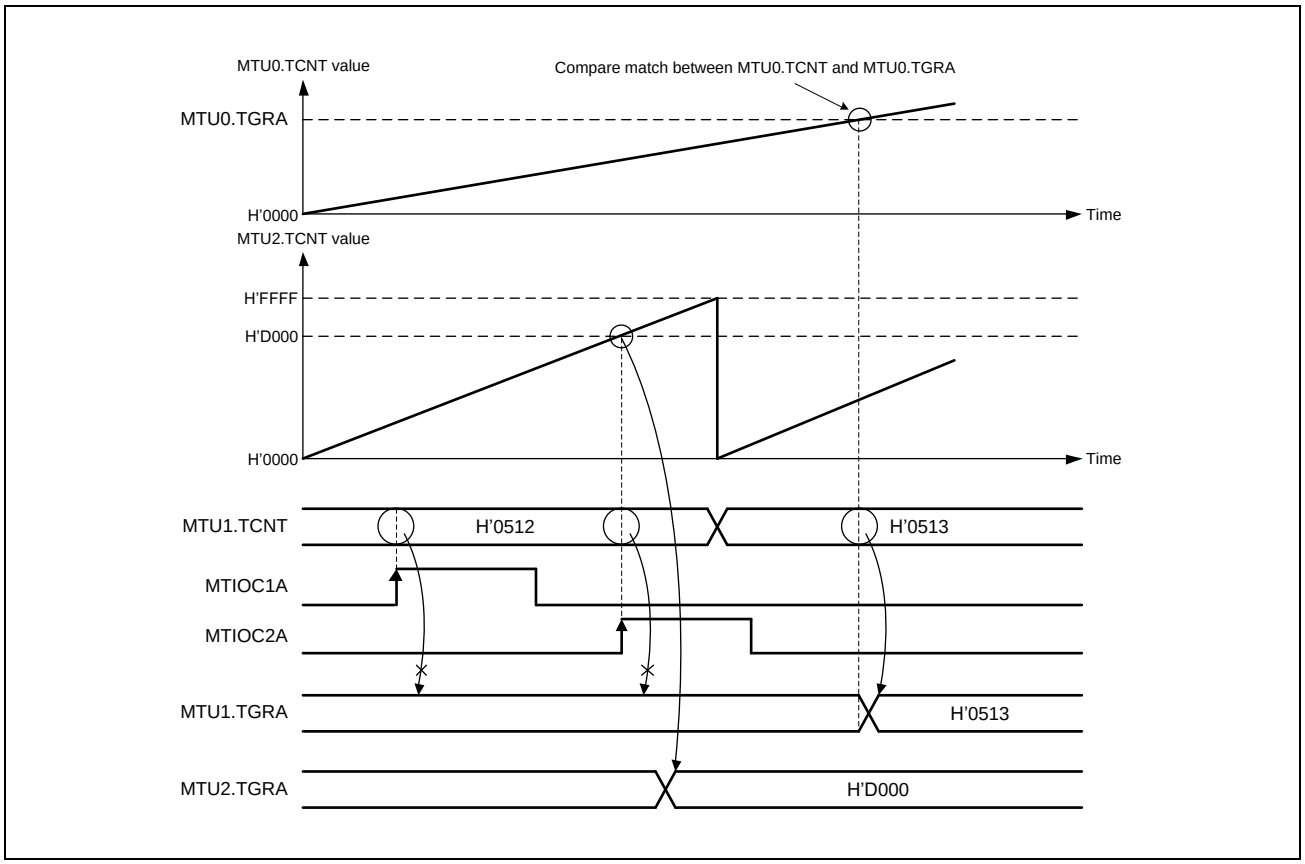



Figure 16.25 Cascaded Operation Example (d)

### 16.3.5 PWM Modes

PWM modes are provided to output PWM waveforms from the external pins. The output level can be selected as low, high, or toggle output in response to a compare match of each TGR.

PWM waveforms in the range of 0% to 100% duty cycle can be output according to the TGR settings.

By designating TGR compare match as the counter clearing source, the PWM cycle can be specified in that register. Every channel except MTU5 and MTU8 can be set to PWM mode independently. Synchronous operation is also possible between the channels placed in PWM mode or between a channel in PWM mode and another channel in a different mode.

There are two PWM modes as described below.

#### PWM Mode 1

PWM waveforms are output from the MTIOCnA and MTIOCnC pins by pairing TGRA with TGRB and TGRC with TGRD. The levels specified by the TIOR.IOA[3:0] and IOC[3:0] bits are output from the MTIOCnA and MTIOCnC pins at compare matches A and C, and the level specified by the TIOR.IOB[3:0] and IOD[3:0] bits are output at compare matches B and D (n = 0 to 4, 6, 7). The initial output value is set in TGRA or TGRC. If the values set in paired TGRs are identical, the output value does not change even when a compare match occurs.

In PWM mode 1, PWM waveforms in up to 12 phases can be output.

#### PWM Mode 2

PWM waveform output is generated using one TGR as the cycle register and the others as duty registers. The level specified in TIOR is output at compare matches. Upon counter clearing by a cycle register compare match, the initial value set in TIOR is output from each pin. If the values set in the cycle and duty registers are identical, the output value does not change even when a compare match occurs.

In PWM mode 2, up to eight phases of PWM waveforms can be output when synchronous clearing is used as synchronous operation in the channels that cannot be placed in PWM mode 2.

The correspondence between PWM output pins and registers is shown in **Table 16.65**.

Table 16.65 PWM Output Registers and Output Pins

| Channel | Register | Output Pins | |
|---|---|---|---|
| | | PWM Mode 1 | PWM Mode 2 |
| MTU0 | TGRA | MTIOC0A | MTIOC0A |
| | TGRB | | MTIOC0B |
| | TGRC | MTIOC0C | MTIOC0C |
| | TGRD | | MTIOC0D |
| MTU1 | TGRA | MTIOC1A | MTIOC1A |
| | TGRB | | MTIOC1B |
| MTU2 | TGRA | MTIOC2A | MTIOC2A |
| | TGRB | | MTIOC2B |
| MTU3 | TGRA | MTIOC3A | Setting prohibited |
| | TGRB | | |
| | TGRC | MTIOC3C | |
| | TGRD | | |
| MTU4 | TGRA | MTIOC4A | |
| | TGRB | | |
| | TGRC | MTIOC4C | |
| | TGRD | | |
| MTU6 | TGRA | MTIOC6A | |
| | TGRB | | |
| | TGRC | MTIOC6C | |
| | TGRD | | |
| MTU7 | TGRA | MTIOC7A | |
| | TGRB | | |
| | TGRC | MTIOC7C | |
| | TGRD | | |

**Note:** In PWM mode 2, PWM waveform output is not possible for the TGR register in which the PWM cycle is set.

### (1) Example of PWM Mode Setting Procedure

**Figure 16.26** shows an example of the PWM mode setting procedure.

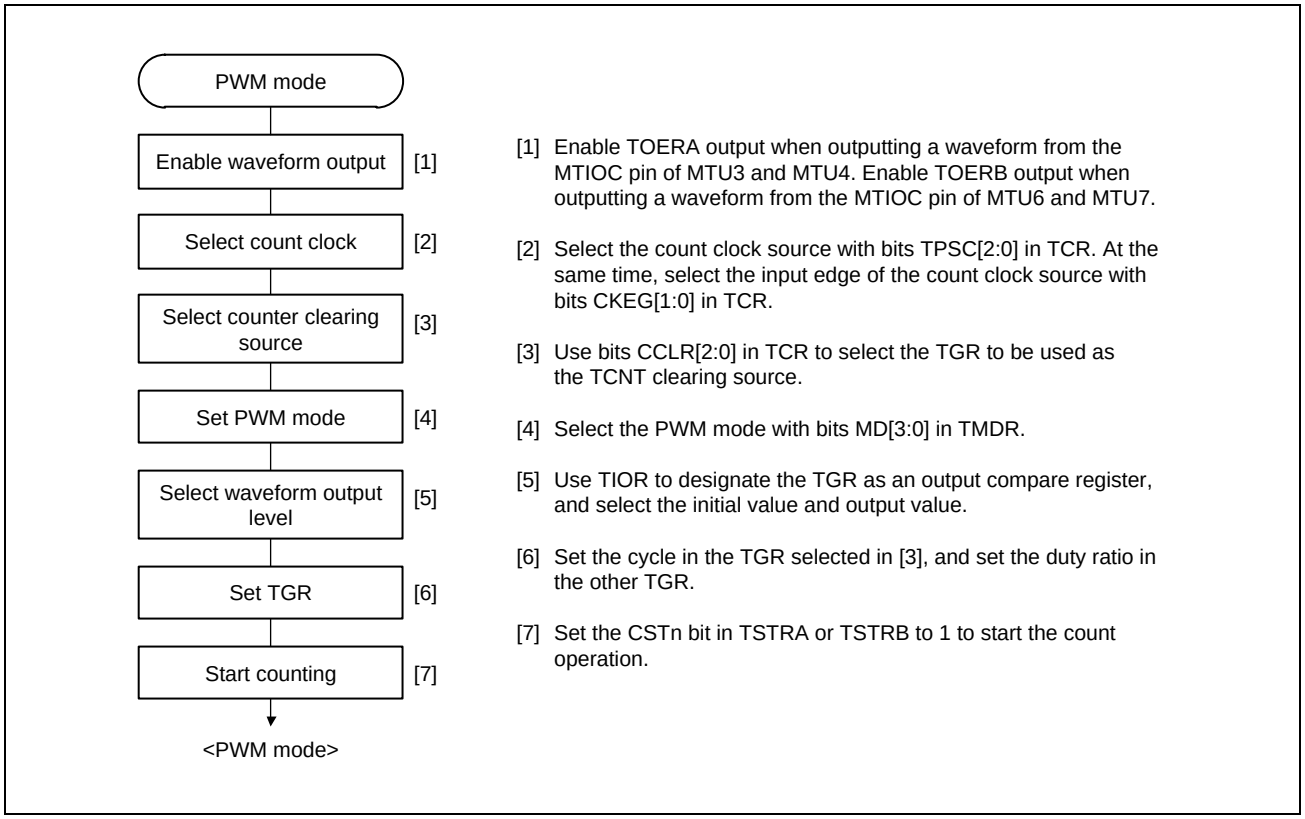



Figure 16.26 Example of PWM Mode Setting Procedure

### (2) Examples of PWM Mode Operation

**Figure 16.27** shows an example of operation in PWM mode 1.

In this example, TGRA compare match is set as the TCNT clearing source, 0 is set as the initial output value and output value for TGRA, and 1 is set as the output value for TGRB.

In this case, the value set in TGRA is used as the cycle, and the value set in TGRB is used as the duty ratio.

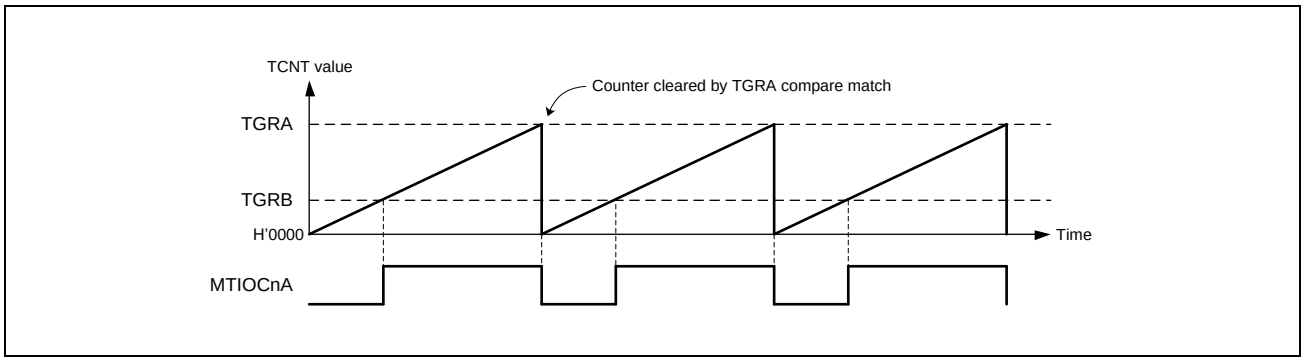



Figure 16.27 Example of PWM Mode 1 Operation (n = 0 to 4, 6, 7)

**Figure 16.28** shows an example of operation in PWM mode 2.

In this example, synchronous operation is designated for MTU0 and MTU1, MTU1.TGRB compare match is set as the TCNT clearing source, and low is set as the initial output value and high as the output value for the other TGR registers (MTU0.TGRA to MTU0.TGRD and MTU1.TGRA), outputting 5-phase PWM waveforms.

In this case, the value set in MTU1.TGRB is used as the cycle, and the values set in the other TGRs are used as the duty ratio.

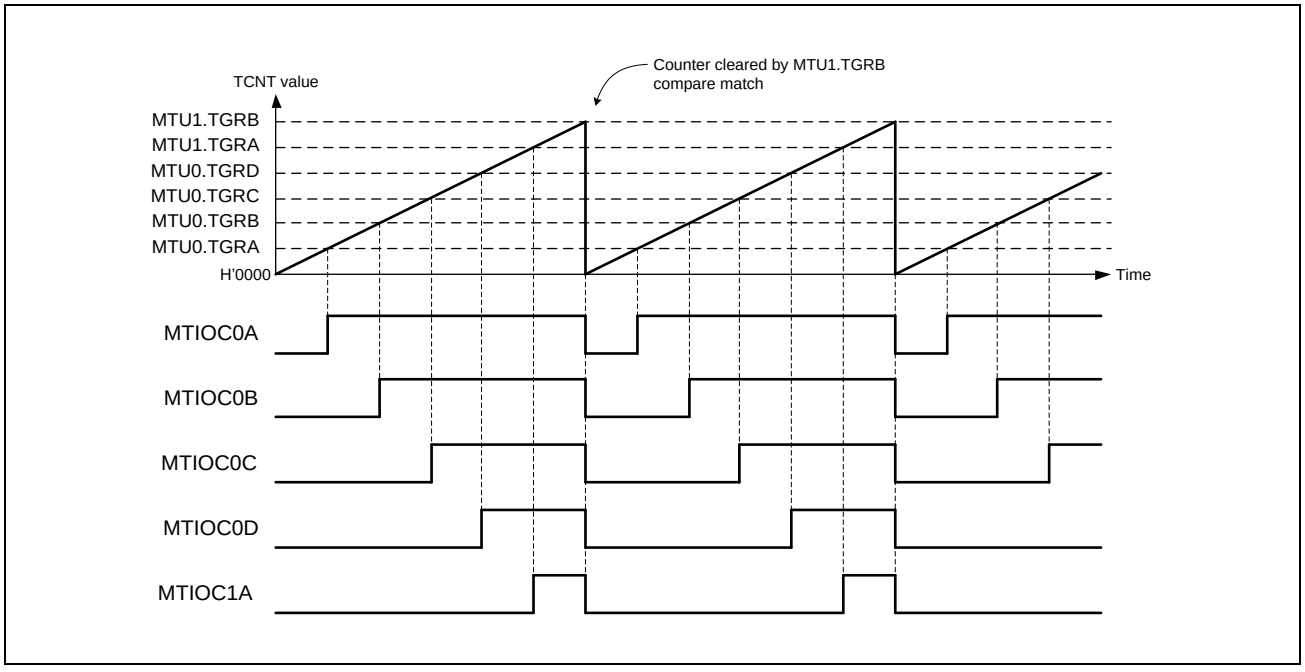



Figure 16.28 Example of PWM Mode 2 Operation

**Figure 16.29** shows examples of PWM waveform output with 0% duty and 100% duty in PWM mode 1. In these examples, TGRA compare match is selected as the TCNT clearing source, the initial output value and the output value for TGRA are set to the low level, and the output value for TGRB is set to the high level.

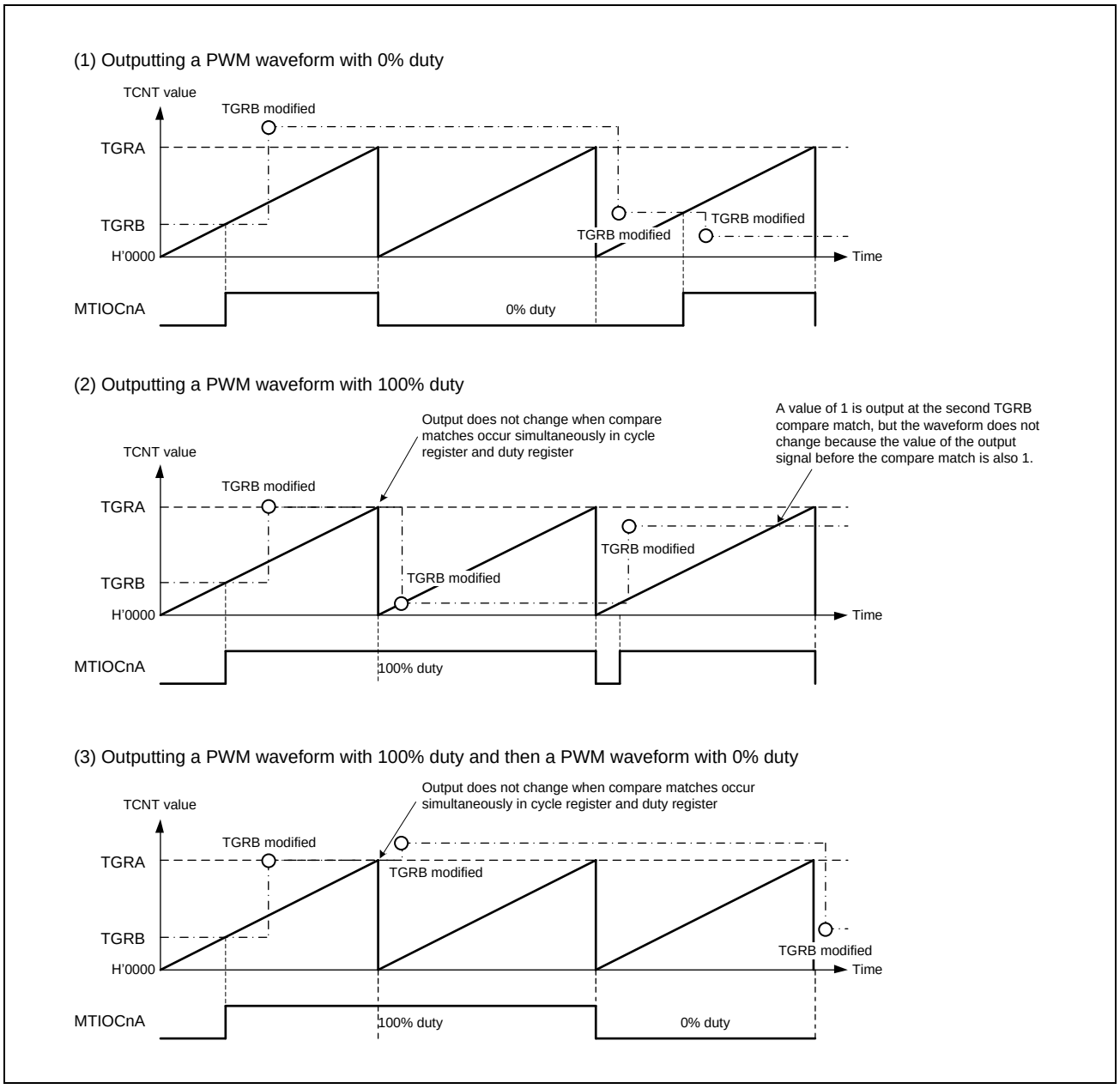



Figure 16.29 Examples of PWM Mode Operation (PWM Waveform Output with 0% Duty and 100% Duty) (n = 0 to 4, 6, 7)

### 16.3.6 Phase Counting Mode

There are two phase counting modes: 16-bit phase counting mode in which MTU1 and MTU2 operate independently, and cascade connection 32-bit phase counting mode in which MTU1 and MTU2 are cascaded.

In phase counting mode, the phase difference between two external input clocks is detected and the corresponding TCNT is incremented or decremented.

Two external clock input pins for each phase counting mode are not affected by the settings of TCR.TPSC[2:0], TCR2.TPSC2[2:0], and CKEG[1:0]. The two external clock input pins used in 16-bit phase counting mode and cascade connection 32-bit phase counting mode of MTU2 can be selected by MTU1.TMDR3.PHCKSEL. In a phase counting mode other than 16-bit phase counting mode and cascade connection 32-bit phase counting mode of MTU2, MTCLKA and MTCLKB are selected for A-phase and B-phase, respectively. In phase counting mode, the external clock pins MTCLKA, MTCLKB, MTCLKC, and MTCLKD are used for two-phase encoder pulse input.

**Table 16.66** lists the external clock input pins to be connected in each phase counting mode.

Table 16.66 Clock Input Pins in Phase Counting Mode

| | | External Clock Input Pins | |
|---|---|---|---|
| Phase Counting Mode | TMDR3.PHCKSEL bit | A-Phase | B-Phase |
| MTU1 16-bit phase counting mode | x (Don't care) | MTCLKA | MTCLKB |
| MTU2 16-bit phase counting mode | 0 | MTCLKA | MTCLKB |
| | 1 (initial value) | MTCLKC | MTCLKD |
| Cascade connection 32-bit phase counting mode | 0 | MTCLKA | MTCLKB |
| | 1 (initial value) | MTCLKC | MTCLKD |

### 16.3.6.1 16-Bit Phase Counting Mode

When the MTU1.TMDR3.LWA is 0, 16-bit phase counting mode can be set individually for MTU1 and MTU2.

In 16-bit phase counting mode, the phase difference between two external input clocks is detected and the 16-bit counter TCNT of the corresponding channel is incremented or decremented.

When 16-bit phase counting mode is specified, an external clock is selected as the count clock and TCNT operates as an up-counter/down-counter regardless of the setting of TCR.TPSC[2:0], TCR2.TPSC2[2:0], and CKEG[1:0]. However, the functions of TCR.CCLR[1:0],TIOR, TIER, and TGR are enabled, and input capture/compare match and interrupt functions can be used.

These external input pins can be used for two-phase encoder pulse input.

If an overflow occurs during TCNT up-counting, a TCIV interrupt is generated when the corresponding TCIEV bit in the TIER register is set to 1.

If an underflow occurs during TCNT down-counting, a TCIU interrupt is generated when the corresponding TCIEU bit in the TIER register is set to 1.

The TCFD bit in TSR is the count direction flag. Read the TCFD flag to check whether TCNT is counting up or down.

#### (1) Example of 16-Bit Phase Counting Mode Setting Procedure

**Figure 16.30** shows an example of the phase counting mode setting procedure.

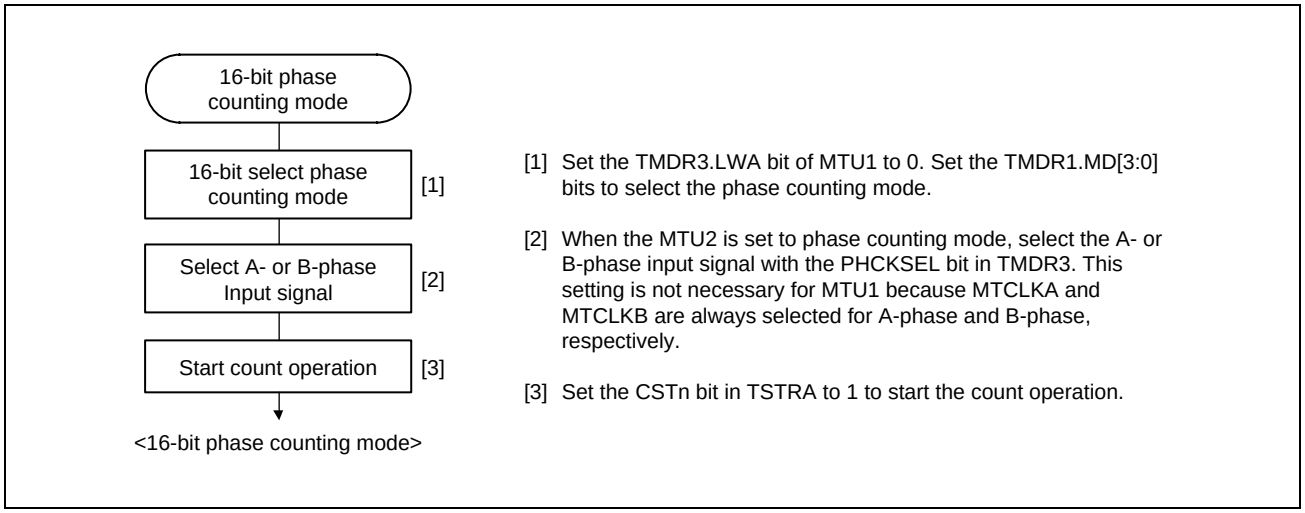



Figure 16.30 Example of 16-Bit Phase Counting Mode Setting Procedure

### (2) Examples of 16-Bit Phase Counting Mode Operation

In phase counting mode, TCNT is incremented or decremented according to the phase difference between two external clocks. There are five modes according to the count conditions. Each mode operates under the condition PHCKSEL = 1, which means the phase clock for MTU1 is input from MTCLKA or MTCLKB and that for MTU2 is input from MTCLKC or MTCLKD.

#### (a) Phase Counting Mode 1

**Figure 16.31** shows an example of operation in phase counting mode 1, and **Table 16.67** summarizes the TCNT up- counting and down-counting conditions.

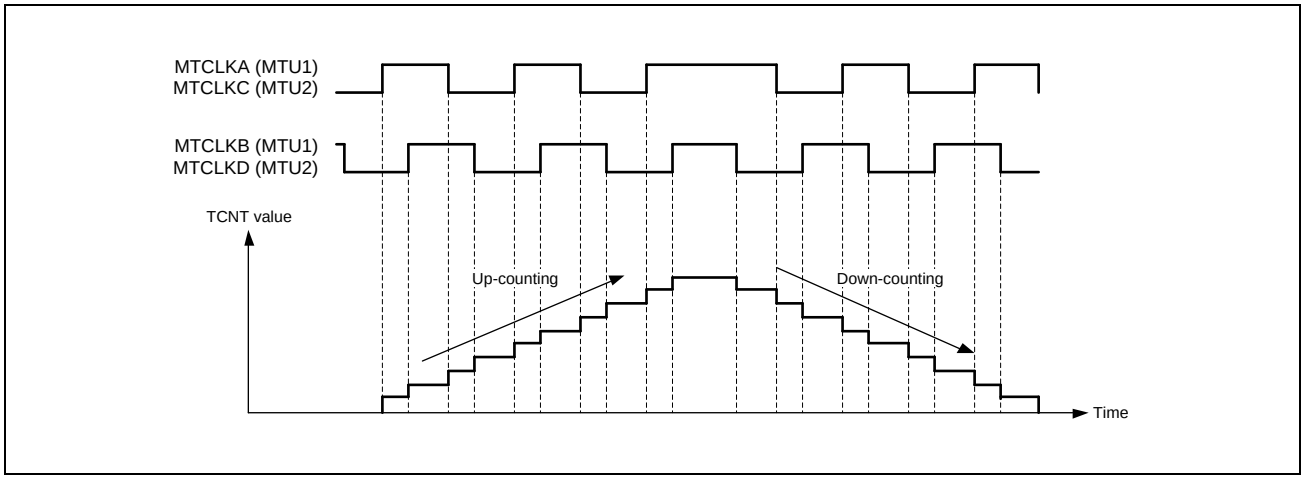



Figure 16.31 Example of Operation in Phase Counting Mode 1

Table 16.67 Up-Counting and Down-Counting Conditions in Phase Counting Mode 1

| MTCLKA (MTU1)<br>MTCLKC (MTU2) | MTCLKB (MTU1)<br>MTCLKD (MTU2) | Operation |
|---|---|---|
| High | ↑ | Up-counting |
| Low | ↓ | |
| ↑ | Low | |
| ↓ | High | |
| High | ↓ | Down-counting |
| Low | ↑ | |
| ↑ | High | |
| ↓ | Low | |

**Remarks:** ↑ : Rising edge
↓ : Falling edge

### (b) Phase Counting Mode 2

**Figure 16.32** to **Figure 16.34** show the examples of operation in phase counting mode 2 and **Table 16.68** summarizes the TCNT up-counting and down-counting conditions.

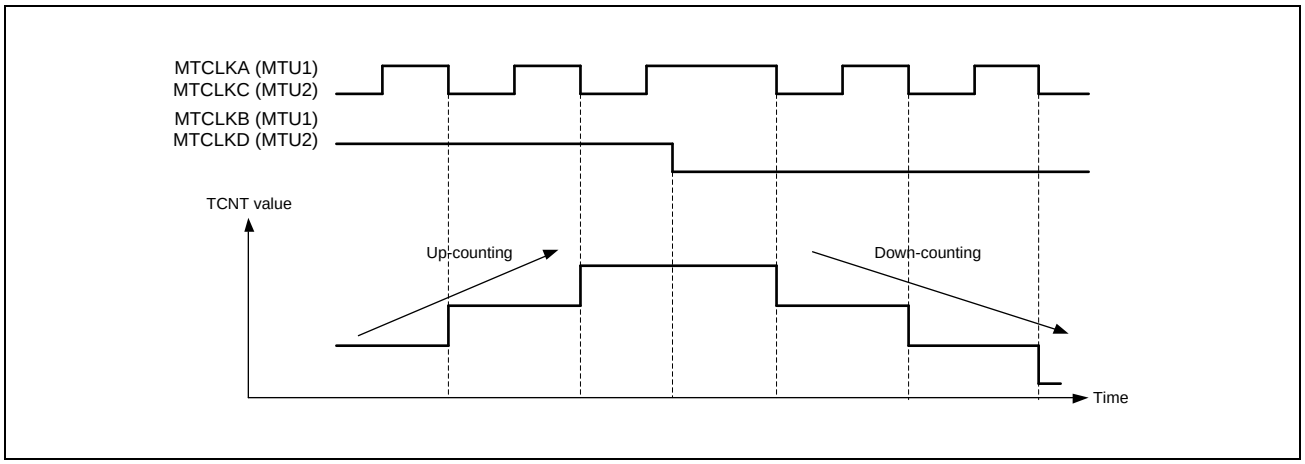



Figure 16.32 Example of Operation in Phase Counting Mode 2 (When MTUn.TCR2.PCB[1:0] is 00b (n = 1, 2))

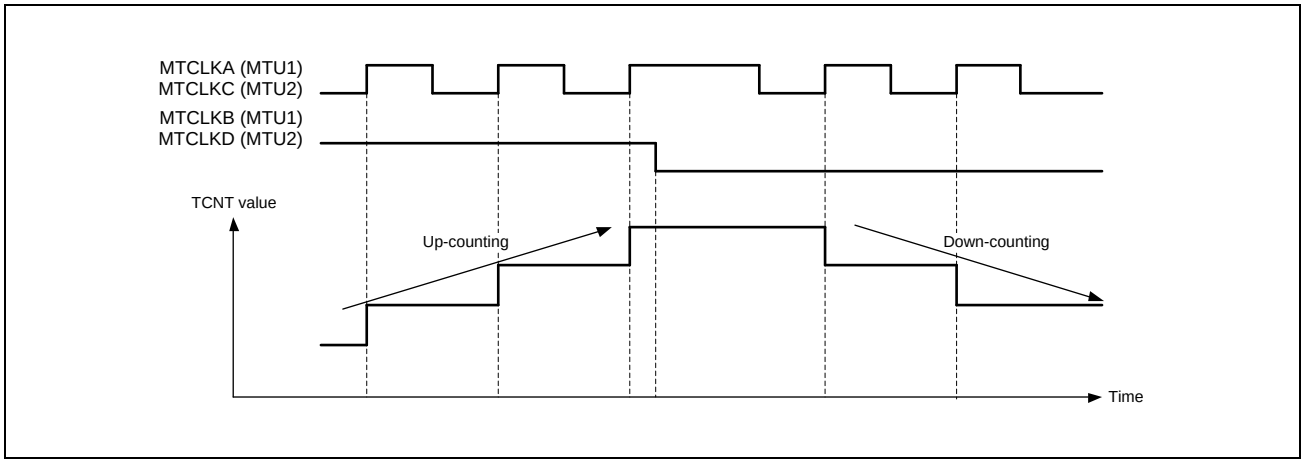



Figure 16.33 Example of Operation in Phase Counting Mode 2 (When MTUn.TCR2.PCB[1:0] is 01b (n = 1, 2))

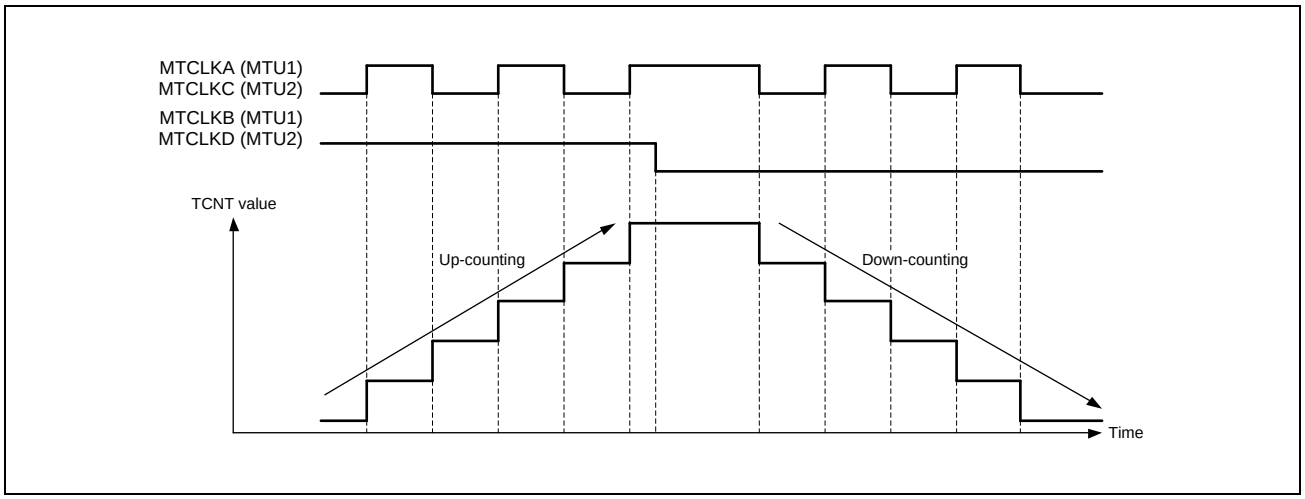



Figure 16.34 Example of Operation in Phase Counting Mode 2 (When MTUn.TCR2.PCB[1:0] is 1xb (n = 1, 2))

Table 16.68 Up-Counting and Down-Counting Conditions in Phase Counting Mode 2

| PCB[1:0] | MTCLKA (MTU1) MTCLKC (MTU2) | MTCLKB (MTU1) MTCLKD (MTU2) | Operation |
|---|---|---|---|
| 00b | High | ↑ | Not counted (Don't care) |
| | Low | ↓ | |
| | ↑ | Low | |
| | ↓ | High | Up-counting |
| | High | ↓ | |
| | Low | ↑ | |
| | ↑ | High | |
| | ↓ | Low | Down-counting |
| 01b | High | ↑ | Not counted (Don't care) |
| | Low | ↓ | |
| | ↑ | Low | Down-counting |
| | ↓ | High | Not counted (Don't care) |
| | High | ↓ | |
| | Low | ↑ | |
| | ↑ | High | Up-counting |
| | ↓ | Low | Not counted (Don't care) |
| 1xb | High | ↑ | Not counted (Don't care) |
| | Low | ↓ | |
| | ↑ | Low | Down-counting |
| | ↓ | High | Up-counting |
| | High | ↓ | Not counted (Don't care) |
| | Low | ↑ | |
| | ↑ | High | Up-counting |
| | ↓ | Low | Down-counting |

**Remarks:** ↑ : Rising edge
↓ : Falling edge

### (c) Phase Counting Mode 3

**Figure 16.35** to **Figure 16.37** show the examples of operation in phase counting mode 3 and **Table 16.69** summarizes the TCNT up-counting and down-counting conditions.

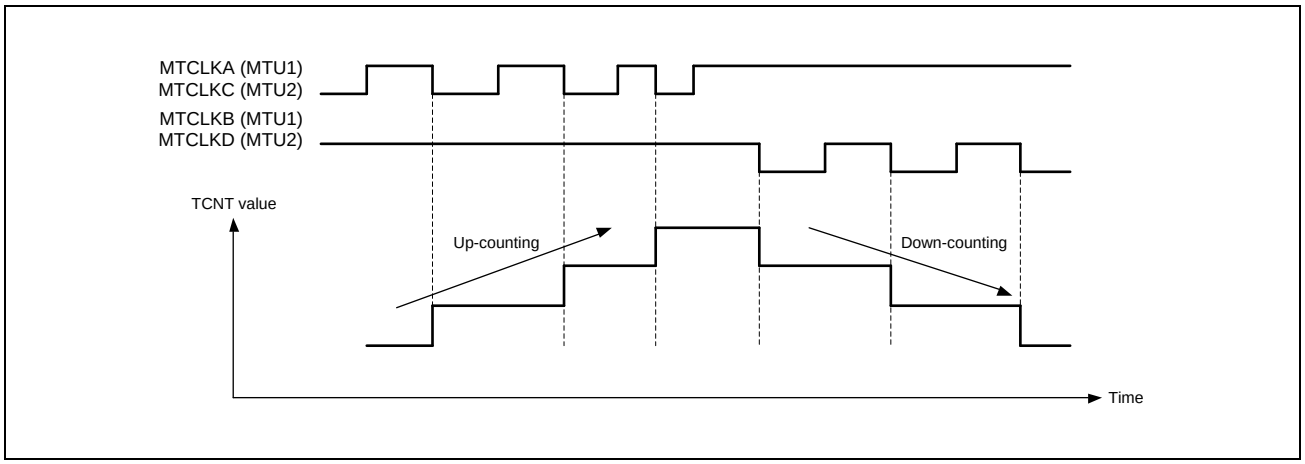



Figure 16.35 Example of Operation in Phase Counting Mode 3 (When MTUn.TCR2.PCB[1:0] is 00b (n = 1, 2))

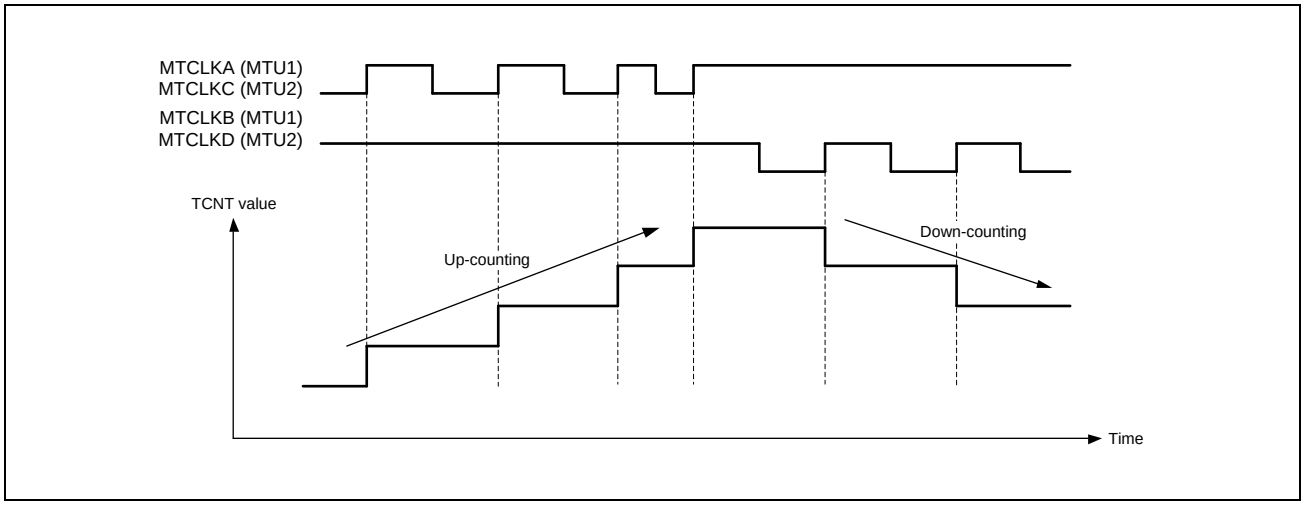



Figure 16.36 Example of Operation in Phase Counting Mode 3 (When MTUn.TCR2.PCB[1:0] is 01b (n = 1, 2))

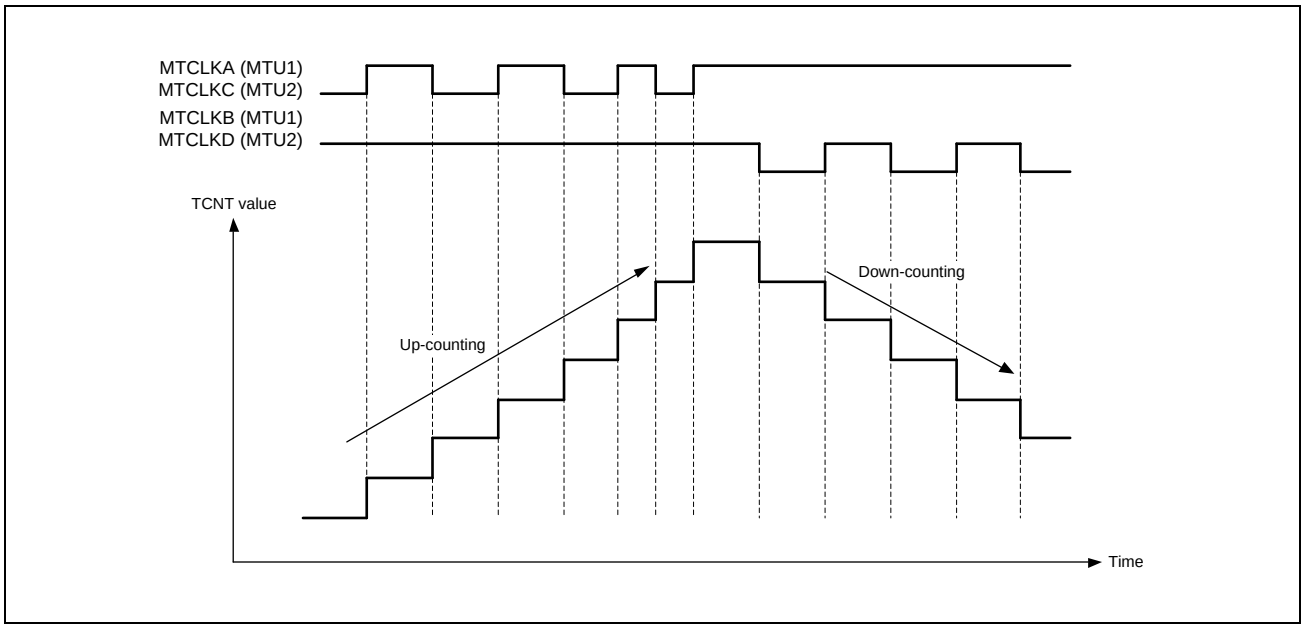



Figure 16.37 Example of Operation in Phase Counting Mode 3 (When MTUn.TCR2.PCB[1:0] is 1xb (n = 1, 2))

Table 16.69 Up-Counting and Down-Counting Conditions in Phase Counting Mode 3

| PCB[1:0] | MTCLKA (MTU1)<br>MTCLKC (MTU2) | MTCLKB (MTU1)<br>MTCLKD (MTU2) | Operation |
|---|---|---|---|
| 00b | High | ↑ | Not counted (Don't care) |
| | Low | ↓ | |
| | ↑ | Low | |
| | ↓ | High | Up-counting |
| | High | ↓ | Down-counting |
| | Low | ↑ | Not counted (Don't care) |
| | ↑ | High | |
| | ↓ | Low | |
| 01b | High | ↑ | Down-counting |
| | Low | ↓ | Not counted (Don't care) |
| | ↑ | Low | |
| | ↓ | High | |
| | High | ↓ | |
| | Low | ↑ | |
| | ↑ | High | Up-counting |
| | ↓ | Low | Not counted (Don't care) |
| 1xb | High | ↑ | Down-counting |
| | Low | ↓ | Not counted (Don't care) |
| | ↑ | Low | |
| | ↓ | High | Up-counting |
| | High | ↓ | Down-counting |
| | Low | ↑ | Not counted (Don't care) |
| | ↑ | High | Up-counting |
| | ↓ | Low | Not counted (Don't care) |

**Remarks:** ↑ : Rising edge
↓ : Falling edge

### (d) Phase Counting Mode 4

**Figure 16.38** shows an example of operation in phase counting mode 4, and **Table 16.70** summarizes the TCNT up- counting and down-counting conditions.

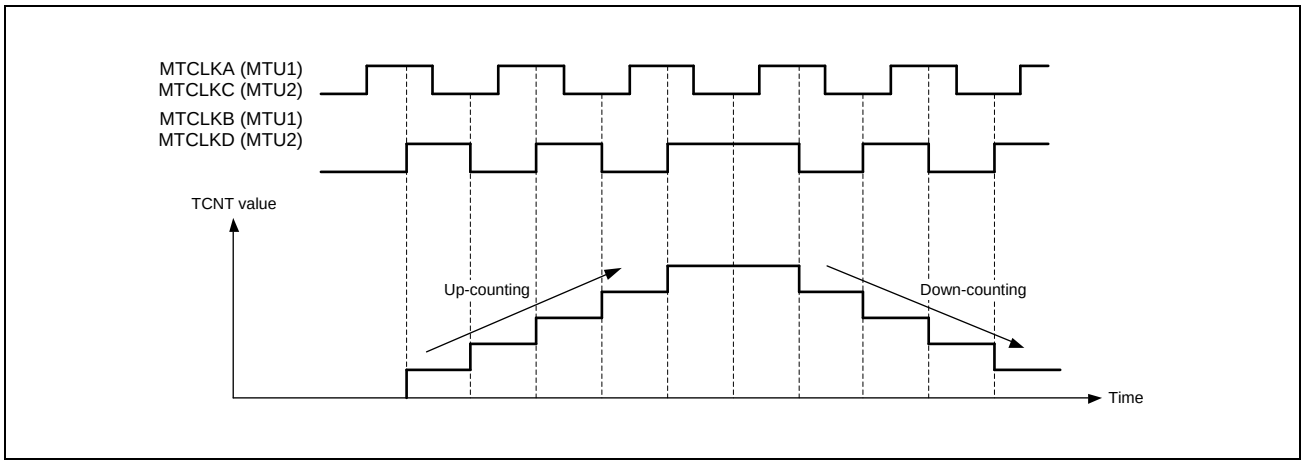



Figure 16.38 Example of Operation in Phase Counting Mode 4

Table 16.70 Up-Counting and Down-Counting Conditions in Phase Counting Mode 4

| MTCLKA (MTU1)<br>MTCLKC (MTU2) | MTCLKB (MTU1)<br>MTCLKD (MTU2) | Operation |
|---|---|---|
| High | ↑ | Up-counting |
| Low | ↓ | |
| ↑ | Low | Not counted (Don't care) |
| ↓ | High | |
| High | ↓ | Down-counting |
| Low | ↑ | |
| ↑ | High | Not counted (Don't care) |
| ↓ | Low | |

**Remarks:** ↑ : Rising edge
↓ : Falling edge

### (e) Phase Counting Mode 5

**Figure 16.39** and **Figure 16.40** show the examples of operation in phase counting mode 5 and **Table 16.71** summarizes the TCNT up-counting and down-counting conditions.

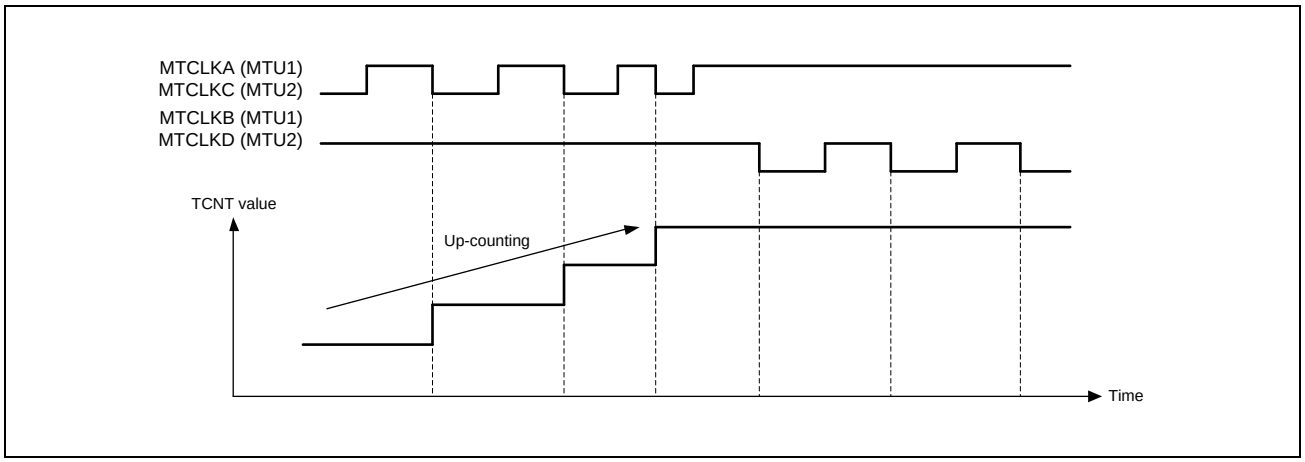



Figure 16.39 Example of Operation in Phase Counting Mode 5 (When MTUn.TCR2.PCB[1:0] = 0xb (n = 1, 2))

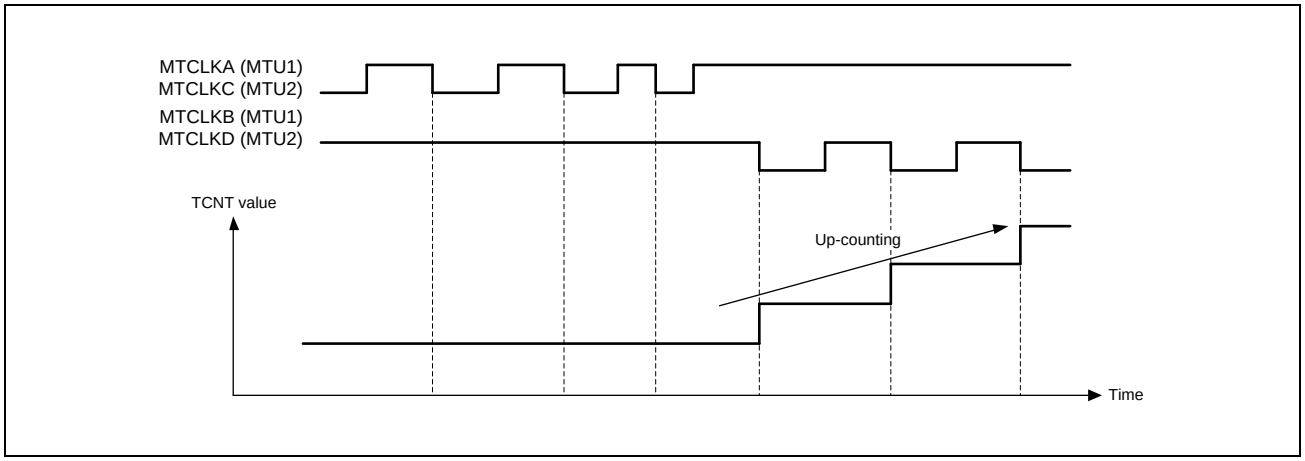



Figure 16.40 Example of Operation in Phase Counting Mode 5 (When MTUn.TCR2.PCB[1:0] = 1xb (n = 1, 2))

Table 16.71 Up-Counting and Down-Counting Conditions in Phase Counting Mode 5

| PCB[1:0] | MTCLKA (MTU1)<br>MTCLKC (MTU2) | MTCLKB (MTU1)<br>MTCLKD (MTU2) | Operation |
|---|---|---|---|
| 0xb | High |  | Not counted (Don't care) |
| | Low |  | |
| |  | Low | |
| |  | High | Up-counting |
| | High |  | Not counted (Don't care) |
| | Low |  | |
| |  | High | |
| |  | Low | Up-counting |
| 1xb | High |  | Not counted (Don't care) |
| | Low |  | Up-counting |
| |  | Low | Not counted (Don't care) |
| |  | High | |
| | High |  | Up-counting |
| | Low |  | Not counted (Don't care) |
| |  | High | |
| |  | Low | |

**Remarks:**  : Rising edge
 : Falling edge

### (3) 16-Bit Phase Counting Mode Application Example

**Figure 16.41** shows an example in which MTU1 is in phase counting mode, and MTU1 is coupled with MTU0 to input 2-phase encoder pulses of a servo motor in order to detect position or speed.

MTU1 is set to phase counting mode 1, and the encoder pulse A-phase and B-phase are input to MTCLKA and MTCLKB.

In MTU0, MTU0.TGRC compare match is specified as the TCNT clearing source and MTU0.TGRA and MTU0.TGRC are used for the compare match function and are set with the speed control cycle and position control cycle.

MTU0.TGRB is used for input capture, with MTU0.TGRB and MTU0.TGRD operating in buffer mode. The MTU1 count clock is designated as the MTU0.TGRB input capture source, and the widths of 2-phase encoder 4-multiplication pulses are detected.

MTU1.TGRA and MTU1.TGRB for MTU1 are designated for the input capture function and MTU0.TGRA and MTU0.TGRC compare matches in MTU0 are selected as the input capture sources to store the up/down-counter values for the control cycles.

This procedure enables the accurate detection of position and speed.

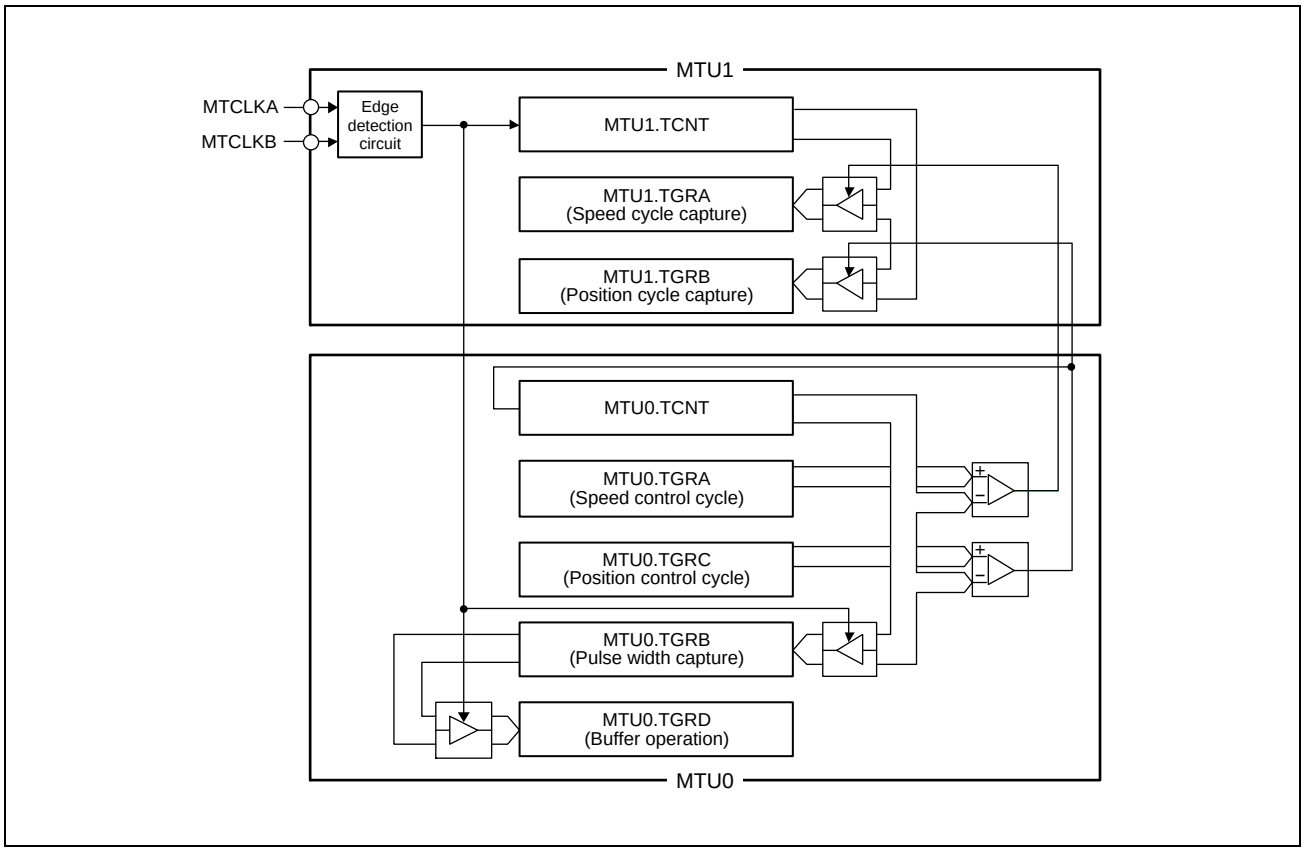



Figure 16.41 16-Bit Phase Counting Mode Application Example

#### 16.3.6.2 Cascade Connection 32-Bit Phase Counting Mode

When MTU1 is set to phase counting mode by setting MTU1.TMDR3.LWA = 1, MTU1 and MTU2 are connected to operate in cascade connection 32-bit phase counting mode as shown in **Figure 16.42**. When this mode is used, the TCR, TCR2, TIOR, TIER, TGR, and TSR registers are controlled by MTU1 and the settings of MTU2 are disabled. See **Figure 16.43** for the procedure for setting cascade connection 32-bit phase counting mode.

In this mode, three-phase (A, B, and Z) signals can be input. As an encoder pulse signal, the external input phase clocks MTCLKA and MTCLKB or MTCLKC and MTCLKD can be selected for A-phase and B-phase, and MTIOC1A can be selected for Z-phase, respectively. See **Table 16.70** for selecting external clock input of A-phase and B-phase. A counter event is generated using an A-phase or B-phase pulse and counted by the 32-bit counter MTU1.TCNTLW.

An input capture can also be generated using a Z-phase signal; thus, an angular velocity can be measured using the value captured in the general register.

Furthermore, MTU8 can be used as a channel for measuring a 1-ms interval, and a compare match signal can be output at a 1-ms interval to the MTU1 and MTU2, which operate in cascade connection 32-bit phase counting mode. That is, a compare match signal of MTU8 is used as a capture signal of MTU1 and MTU2, and the number of A-phase and B-phase pulses for a 1-ms period can be measured.

When MTU0 or MTU5 is specified as the channel for measuring Z-phase signal pulses, this compare match signal of TGRC in MTU8 can be output as a capture signal or a clear signal to MTU0 or MTU5; thus, the Z-phase count can be measured at 1-ms intervals.

In addition, a counter event signal of combined MTU1 and MTU2 can be used as a capture signal of TGRD in MTU8, and measurement can be performed including the intervals of A, B, or both phase pulses. In this case, the TGRD register in MTU8 should be set to buffer operation.

Refer to **Section 16.3.4, Cascaded Operation**, for details on the cascade connection function for connecting MTU1 and MTU2 in a mode other than cascade connection 32-bit phase counting mode.

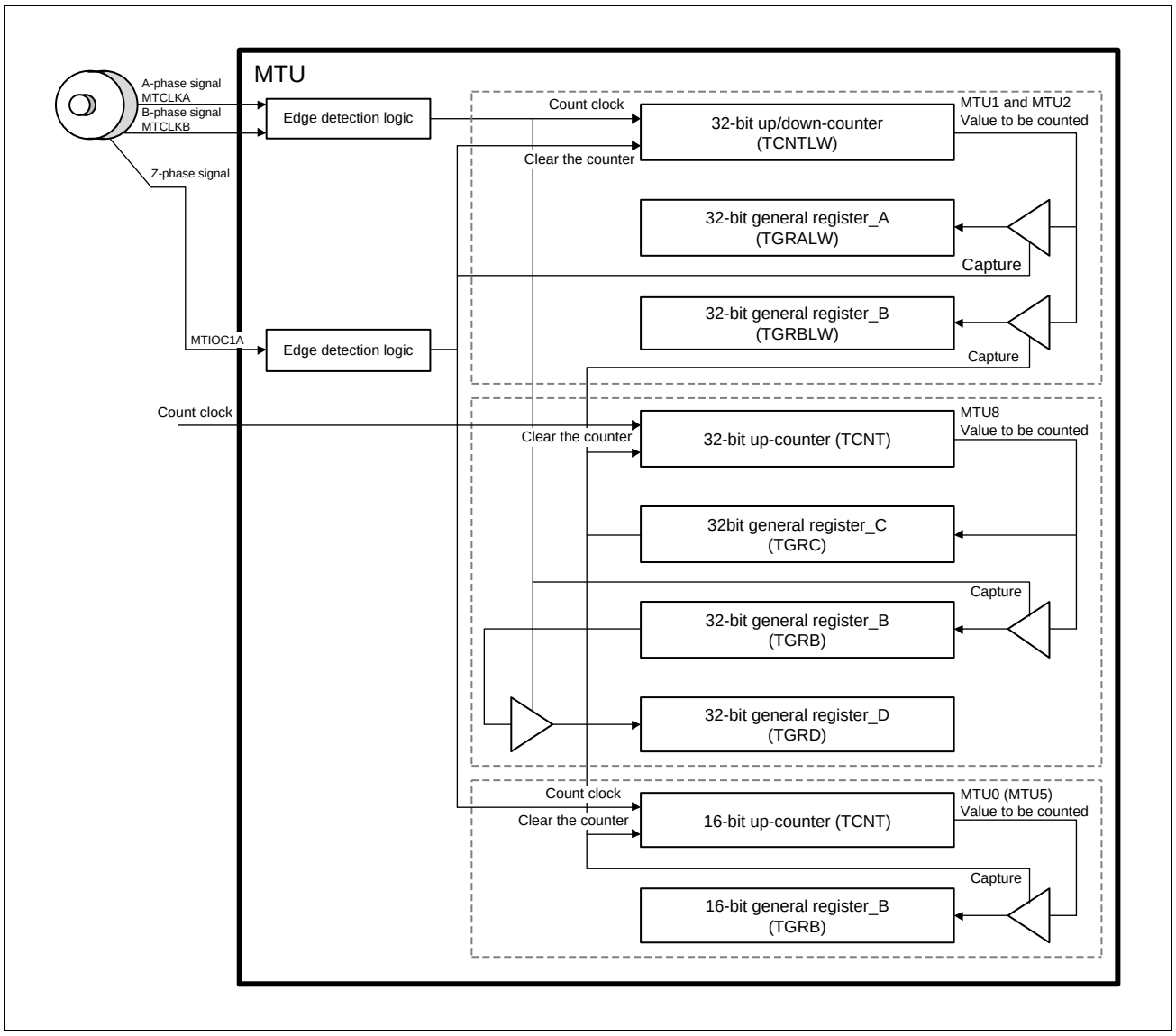
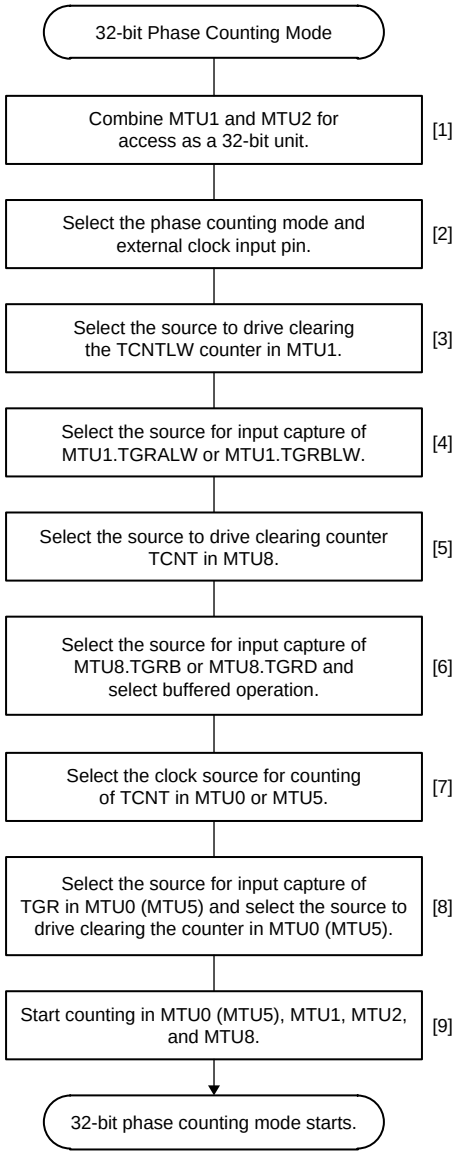



Figure 16.42 Block Diagram for Operation in Cascade Connection 32-Bit Phase Counting Mode

### (1) Example of Setting Cascade Connection 32-Bit Phase Counting Mode

**Figure 16.43** shows an example of the procedure for setting cascade connection 32-bit phase counting mode.




[1] Set the LWA bit in MTU1.TMDR3 to 1 and combine MTU1 and MTU3 for access as a 32-bit unit.*1

[2] Select phase counting mode with the MD[3:0] bits in MTU1.TMDR1 and select the external clock input pin with the PHCKSEL bits in MTU1.TMDR3.

[3] Set the CCLR[2:0] bits of TCR in MTU1 to 001b to select input capture of TGRALW in MTU1 as the source of the signal to drive clearing the TCNTLW counter in MTU1.

[4] Set the IOA[3:0] bits in MTU1.TIOR to select the effective edge (rising, falling or both) of the signal produced by input capture of TGRALW in MTU1. After that, set the IOB[3:0] bits in MTU1.TIOR to 1110b to select the compare match signal of MTU8.TGRC as the source to drive input capture of MTU1.TGRBLW.

[5] Set the CCLR[2:0] bits in MTU8.TCR to 101b to select the compare match signal of MTU8.TGRC as the source to drive clearing counter TCNT in MTU8.

[6] Select the source for input capture of MTU8.TGRB as the clock for counting of TCNTLW in MTU1 by setting the IOB[3:0] bits in MTU8.TIORH to 1110b. At the same time, select buffered operation for TGRB and TGRD by setting the BFB bit in MTU8.TMDR1 to 1.

[7] Set the TPSC2[2:0] bits in MTU0.TCR2 to 111b to select the MTIOC1A pin as the clock source for counting of TCNT in MTU0.
(When using TCNT in MTU5, set the TPSC2[2:0] bits in MTU5.TCR2 to 111b).

[8] Select the compare match signal of MTU8.TGRC as the source to drive input capture of MTU0.TGRA by setting the IOA[3:0] bits in MTU0.TIORH to 1110b. (When using TCNT in MTU5, set the TIORU, TIORV, or TIORW bit in MTU5 is set to 10100b).
After that, select input capture of MTU0.TGRA as the source to drive clearing counter TCNT in MTU0 by setting the CCLR[2:0] bits in MTU0.TCR to 001b.

[9] Start counting in MTU0, MTU1, MTU2, and MTU8 by setting the CST0, CST1, CST2, and CST8 bits in TSTRA to 1 at the same time.
(When using TCNT in MTU5, set the counter corresponding to TSTR in MTU5 to 1).

Note 1. When the LWA bit in MTU1.TMDR3 is set to 1, the TCR, TCR2, TMDR, and TIOR registers in MTU1 are effective in controlling cascaded operation. In this case, the settings of these registers in MTU2 are ignored.

Figure 16.43 Procedure for Setting Cascade Connection 32-Bit Phase Counting Mode

### 16.3.7 Reset-Synchronized PWM Mode

In the reset-synchronized PWM mode, three phases of positive and negative PWM waveforms (six phases in total) that share a common wave transition point can be output by combining MTU3 and MTU4 and MTU6 and MTU7.

When set for reset-synchronized PWM mode, the MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4C, MTIOC4B, MTIOC4D, MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7C, MTIOC7B, and MTIOC7D pins function as PWM output pins and timer counters 6 and 12 (MTU3.TCNT and MTU6.TCNT) functions as an up-counter.

**Table 16.72** shows the PWM output pins used. **Table 16.73** shows the settings of the registers.

Table 16.72 Output Pins for Reset-Synchronized PWM Mode

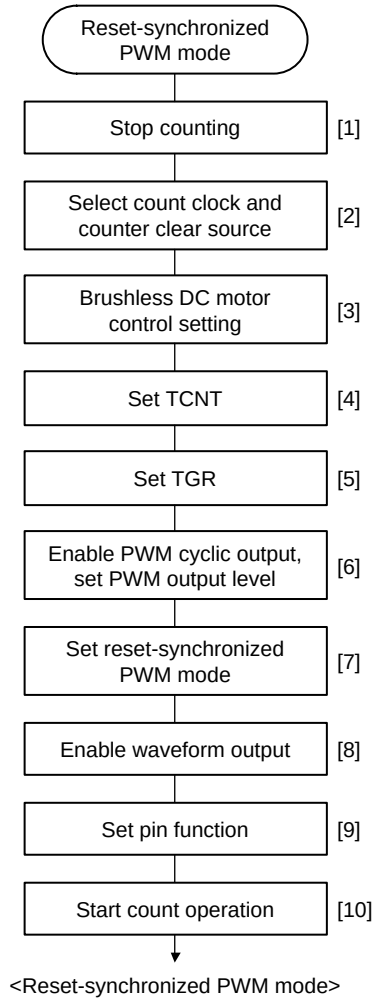
| Channel | Output Pin | Description |
|---|---|---|
| MTU3 | MTIOC3B | PWM output pin 1 |
| | MTIOC3D | PWM output pin 1' (negative-phase waveform of PWM output 1) |
| MTU4 | MTIOC4A | PWM output pin 2 |
| | MTIOC4C | PWM output pin 2' (negative-phase waveform of PWM output 2) |
| | MTIOC4B | PWM output pin 3 |
| | MTIOC4D | PWM output pin 3' (negative-phase waveform of PWM output 3) |
| MTU6 | MTIOC6B | PWM output pin 4 |
| | MTIOC6D | PWM output pin 4' (negative-phase waveform of PWM output 4) |
| MTU7 | MTIOC7A | PWM output pin 5 |
| | MTIOC7C | PWM output pin 5' (negative-phase waveform of PWM output 5) |
| | MTIOC7B | PWM output pin 6 |
| | MTIOC7D | PWM output pin 6' (negative-phase waveform of PWM output 6) |

Table 16.73 Register Settings for Reset-Synchronized PWM Mode

| Register | Setting |
|---|---|
| MTU3.TCNT | Initial setting (H’0000) |
| MTU4.TCNT | Initial setting (H’0000) |
| MTU3.TGRA | Set the count cycle for MTU3.TCNT |
| MTU3.TGRB | Set the transition point of the PWM waveform to be output from the MTIOC3B and MTIOC3D pins |
| MTU4.TGRA | Set the transition point of the PWM waveform to be output from the MTIOC4A and MTIOC4C pins |
| MTU4.TGRB | Set the transition point of the PWM waveform to be output from the MTIOC4B and MTIOC4D pins |
| MTU6.TCNT | Initial setting (H’0000) |
| MTU7.TCNT | Initial setting (H’0000) |
| MTU6.TGRA | Set the count cycle for MTU6.TCNT |
| MTU6.TGRB | Set the transition point of the PWM waveform to be output from the MTIOC6B and MTIOC6D pins |
| MTU7.TGRA | Set the transition point of the PWM waveform to be output from the MTIOC7A and MTIOC7C pins |
| MTU7.TGRB | Set the transition point of the PWM waveform to be output from the MTIOC7B and MTIOC7D pins |

### (1) Example of Procedure for Setting Reset-Synchronized PWM Mode

**Figure 16.44** shows an example of the procedure for specifying the reset-synchronized PWM mode.




<Reset-synchronized PWM mode>

[1] Clear the TSTRA.CST3 (TSTRB.CST6) and TSTRA.CST4 (TSTRB.CST7) bits to 0 to stop the TCNT count operation. Specify the reset-synchronized PWM mode while MTU3.TCNT (MTU6.TCNT) and MTU4.TCNT (MTU7.TCNT) are stopped.

[2] Set the TCR.TPSC[2:0] and TCR.CKEG[1:0] bits of MTU3 (MTU6) to select the count clock source and clock edge for MTU3 (MTU6). Set the TCR.CCLR[2:0] bits to select TGRA compare match as the counter clear source.

[3] To control a brushless DC motor, set the TGCRA.BDC bit and set the feedback signal input source and output chopping or gate signal direct output (only when MTU3 and MTU4 are used).

[4] Set MTU3.TCNT (MTU6.TCNT) and MTU4.TCNT (MTU7.TCNT) to H'0000.

[5] MTU3.TGRA (MTU6.TGRA) is the cycle register. Set the PWM cycle in MTU3.TGRA (MTU6.TGRA).
Set the PWM output transition timing in MTU3.TGRB (MTU6.TGRB), MTU4.TGRA (MTU7.TGRA), and MTU4.TGRB (MTU7.TGRB).
Note that the setting values should be within the range of compare match with MTU3.TCNT (MTU6.TCNT).

$X \leq$ MTU3.TGRA (X: Set value)

[6] Set the PSYE bit in the TOCR1A (TOCR1B) register to enable or disable toggle output synchronized with the PMW cycle, and set the OLSP and OLSN bits to select the PWM output level.
When specifying the PWM output level using the TOCR2A (TOCR2B) and TOLBRA (TOLBRB) in buffer operation, see Timer Output Level Buffer Registers (TOLBRA and TOLBRB).

[7] Set the MD[3:0] bits in MTU3.TMDR1 (MTU6.TMDR1) register to 1000b to select the reset-synchronized PWM mode. Specify the buffer operation (BFA = 1) of the MTU3.TGRA (MTU6.TGRA) register and the buffer operation (BFB = 1) of the MTU3.TGRB register when necessary. Set the MTU4.TMDR1 (MTU7.TMDR1) register to the initial value (normal operation; BFA = 0 and BFB = 0).

[8] Set the TOERA (TOERB) register to enable or disable output from the PWM output pin. Then, set the MTU3.TIORH (MTU6.TIORH), MTU3.TIORL (MTU6.TIORL), MTU4.TIORH (MTU7.TIORH), and MTU4.TIORL (MTU7.TIORL) registers to H'00.

[9] Set the port mode control registers (PMCn) and Pmn port fuction control reigister (PFCm) of the GPIO ports.

[10] Set the TSTRA.CST3 (TSTRB.CST6) bit to 1 to start count operation.

**Note:** The output waveform starts to toggle at the point of MTU3.TCNT (MTU6.TCNT) = MTU3.TGRA (MTU6.TGRA) = X by setting X = MTU3.TGRA (MTU6.TGRA), i.e., cycle = duty ratio.

Figure 16.44 Procedure for Selecting Reset-Synchronized PWM Mode

### (2) Example of Reset-Synchronized PWM Mode Operation

**Figure 16.45** shows an example of operation in the reset-synchronized PWM mode.

MTU3.TCNT and MTU4.TCNT (MTU6.TCNT and MTU7.TCNT) operate as up-counters. The counters are cleared when a compare match occurs between MTU3.TCNT (MTU6.TCNT) and MTU3.TGRA (MTU6.TGRA), and then begin incrementing from H’0000. The output from the PWM pins toggles every time a compare match occurs in MTU3.TGRB (MTU6.TGRB), MTU4.TGRA (MTU7.TGRA), and MTU4.TGRB (MTU7.TGRB) and the counters are cleared.

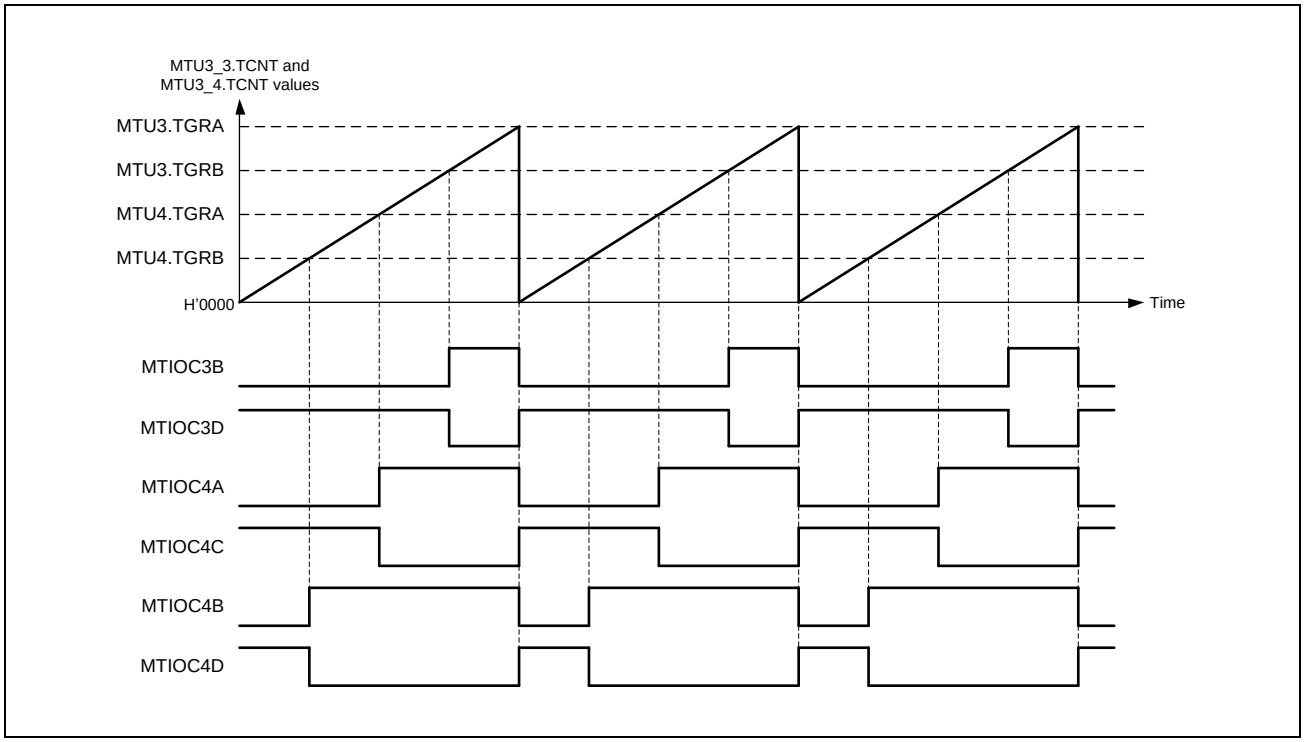



Figure 16.45 Example of Reset-Synchronized PWM Mode Operation
(When OLSN = 1 and OLSP = 1 in MTU3.TOCR1 and MTU4.TOCR1)

### 16.3.8 Complementary PWM Mode

In complementary PWM mode, dead time can be set for PWM waveforms to be output. The dead time is the period during which the upper and lower arm transistors are set to the inactive level in order to prevent short-circuiting of the arms.

Six positive-phase and six negative-phase PWM waveforms (12 phases in total) with dead time can be output by combining MTU3/ MTU4 and MTU6/MTU7. PWM waveforms without dead time can also be output.

In complementary PWM mode, MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4B, MTIOC4C, MTIOC4D, MTIOC6B,

MTIOC6D, MTIOC7A, MTIOC7B, MTIOC7C, and MTIOC7D pins function as PWM output pins, and the MTIOC3A and MTIOC6A pins can be set for toggle output synchronized with the PWM cycle.

MTU3.TCNT, MTU4.TCNT, MTU6.TCNT, and MTU7.TCNT function as up/down-counters.

**Table 16.74** shows the PWM output pins used. **Table 16.75** shows the settings of the registers used.

A function to directly cut off the PWM output by using an external signal is supported as a port function.

Table 16.74 Output Pins for Complementary PWM Mode

| Channel | Output Pin | Description |
|---|---|---|
| MTU3 | MTIOC3A | Toggle output synchronized with PWM cycle (or I/O port) |
| | MTIOC3B | PWM output pin 1 |
| | MTIOC3C | I/O port*1 |
| | MTIOC3D | PWM output pin 1' (negative-phase waveform output of PWM output 1) |
| MTU4 | MTIOC4A | PWM output pin 2 |
| | MTIOC4C | PWM output pin 2' (negative-phase waveform output of PWM output 1) |
| | MTIOC4B | PWM output pin 3 |
| | MTIOC4D | PWM output pin 3' (negative-phase waveform output of PWM output 1) |
| MTU6 | MTIOC6A | Toggle output synchronized with PWM cycle (or I/O port) |
| | MTIOC6B | PWM output pin 4 |
| | MTIOC6C | I/O port*1 |
| | MTIOC6D | PWM output pin 4' (negative-phase waveform output of PWM output 4) |
| MTU7 | MTIOC7A | PWM output pin 5 |
| | MTIOC7C | PWM output pin 5' (negative-phase waveform output of PWM output 5) |
| | MTIOC7B | PWM output pin 6 |
| | MTIOC7D | PWM output pin 6' (negative-phase waveform output of PWM output 6) |

Note 1. Avoid setting the MTIOC3C and MTIOC6C pins as timer I/O pins in complementary PWM mode.

Table 16.75 Register Settings for Complementary PWM Mode (1/2)

| Channel | Counter / Register | Description | Read/Write from CPU |
|---|---|---|---|
| MTU3 | TCNT | Starts up-counting from the value set in the dead time register | Maskable by TRWERA setting*1 |
| | TGRA | Set MTU3.TCNT upper limit value (1/2 carrier cycle + dead time) | Maskable by TRWERA setting*1 |
| | TGRB | PWM output 1 compare register | Maskable by TRWERA setting*1 |
| | TGRC | MTU3.TGRA buffer register | Readable/writable |
| | TGRD | PWM output 1/MTU3.TGRB buffer register | Readable/writable |
| | TGRE | MTU3.TGRB buffer register B (when double buffer function is used) | Readable/writable |
| MTU4 | TCNT | Starts up-counting after being initialized to H'0000 | Maskable by TRWERA setting*1 |
| | TGRA | PWM output 2 compare register | Maskable by TRWERA setting*1 |
| | TGRB | PWM output 3 compare register | Maskable by TRWERA setting*1 |
| | TGRC | PWM output 2/MTU4.TGRA buffer register | Readable/writable |
| | TGRD | PWM output 3/MTU4.TGRB buffer register | Readable/writable |
| | TGRE | MTU4.TGRA buffer register B (when double buffer function is used) | Readable/writable |
| | TGRF | MTU4.TGRB buffer register B (when double buffer function is used) | Readable/writable |
| MTU6 | TCNT | Starts up-counting from the value set in the dead time register | Maskable by TRWERB setting*2 |
| | TGRA | Set MTU6.TCNT upper limit value (1/2 carrier cycle + dead time) | Maskable by TRWERB setting*2 |
| | TGRB | PWM output 4 compare register | Maskable by TRWERB setting*2 |
| | TGRC | MTU6.TGRA buffer register | Readable/writable |
| | TGRD | PWM output 4/MTU6.TGRB buffer register | Readable/writable |
| | TGRE | MTU6.TGRB buffer register B (when double buffer function is used) | Readable/writable |
| MTU7 | TCNT | Starts up-counting after being initialized to H'0000 | Maskable by TRWERB setting*2 |
| | TGRA | PWM output 5 compare register | Maskable by TRWERB setting*2 |
| | TGRB | PWM output 6 compare register | Maskable by TRWERB setting*2 |
| | TGRC | PWM output 5/MTU7.TGRA buffer register | Readable/writable |
| | TGRD | PWM output 6/MTU7.TGRB buffer register | Readable/writable |
| | TGRE | MTU7.TGRA buffer register B (when double buffer function is used) | Readable/writable |
| | TGRF | MTU7.TGRB buffer register B (when double buffer function is used) | Readable/writable |

Note 1. Access can be enabled or disabled according to the setting in TRWERA (timer read/write enable register A).

Note 2. Access can be enabled or disabled according to the setting in TRWERB (timer read/write enable register B).

Table 16.76 Register Settings for Complementary PWM Mode (2/2)

| Counter / Register | Description | Read/Write from CPU |
|---|---|---|
| Timer dead time data register A (TDDRA) | Set MTU4.TCNT and MTU3.TCNT offset value (dead time value) | Maskable by TRWERA setting*1 |
| Timer dead time data register B (TDDRB) | Set MTU7.TCNT and MTU6.TCNT offset value (dead time value) | Maskable by TRWERB setting*2 |
| Timer cycle data register A (TCDRA) | Set MTU4.TCNT upper limit value (1/2 carrier cycle) | Maskable by TRWERA setting*1 |
| Timer cycle data register B (TCDRB) | Set MTU7.TCNT upper limit value (1/2 carrier cycle) | Maskable by TRWERB setting*2 |
| Timer cycle buffer register A (TCBRA) | TCDRA buffer register | Readable/writable |
| Timer cycle buffer register B (TCBRB) | TCDRB buffer register | Readable/writable |
| Subcounter A (TCNTSA) | Subcounter A for dead time generation | Read-only |
| Subcounter B (TCNTSB) | Subcounter B for dead time generation | Read-only |
| Temporary register 1A (TEMP1A) | PWM output 1/MTU3.TGRB temporary register A | Not readable or writable |
| Temporary register 1B (TEMP1B) | PWM output 1/MTU3.TGRB temporary register B (when double buffer function is used) | Not readable or writable |
| Temporary register 2A (TEMP2A) | PWM output 2/MTU4.TGRA temporary register A | Not readable or writable |
| Temporary register 2B (TEMP2B) | PWM output 2/MTU4.TGRA temporary register B (when double buffer function is used) | Not readable or writable |
| Temporary register 3A (TEMP3A) | PWM output 3/MTU4.TGRB temporary register A | Not readable or writable |
| Temporary register 3B (TEMP3B) | PWM output 3/MTU4.TGRB temporary register B (when double buffer function is used) | Not readable or writable |
| Temporary register 4A (TEMP4A) | PWM output 4/MTU6.TGRB temporary register A | Not readable or writable |
| Temporary register 4B (TEMP4B) | PWM output 4/MTU6.TGRB temporary register B (when double buffer function is used) | Not readable or writable |
| Temporary register 5A (TEMP5A) | PWM output 5/MTU7.TGRA temporary register A | Not readable or writable |
| Temporary register 5B (TEMP5B) | PWM output 5/MTU7.TGRA temporary register B (when double buffer function is used) | Not readable or writable |
| Temporary register 6A (TEMP6A) | PWM output 6/MTU7.TGRB temporary register A | Not readable or writable |
| Temporary register 6B (TEMP6B) | PWM output 6/MTU7.TGRB temporary register B (when double buffer function is used) | Not readable or writable |

Note 1. Access can be enabled or disabled according to the setting in TRWERA (timer read/write enable register A).

Note 2. Access can be enabled or disabled according to the setting in TRWERB (timer read/write enable register B).

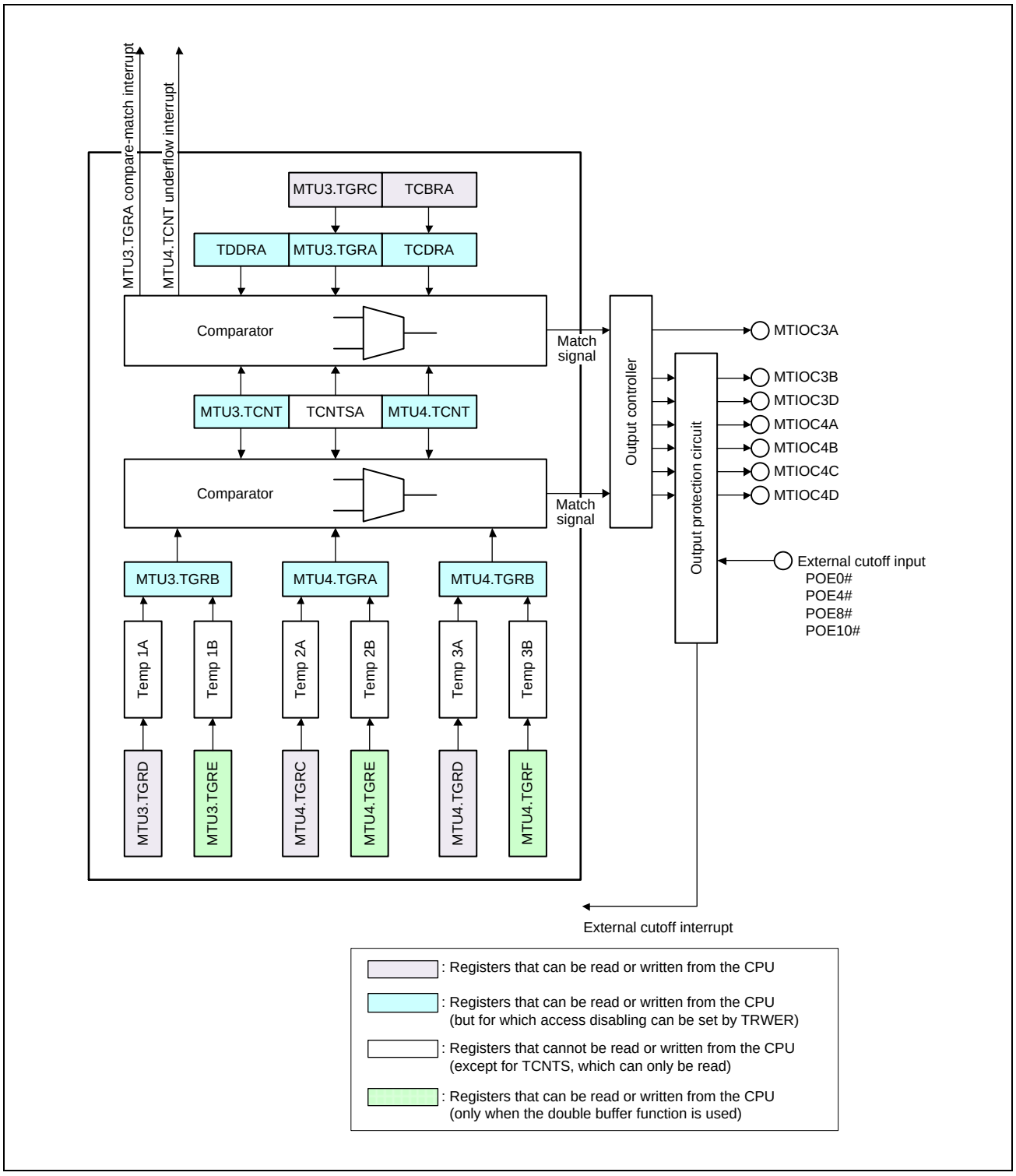



Figure 16.46 Block Diagram of MTU3 and MTU4 in Complementary PWM Mode

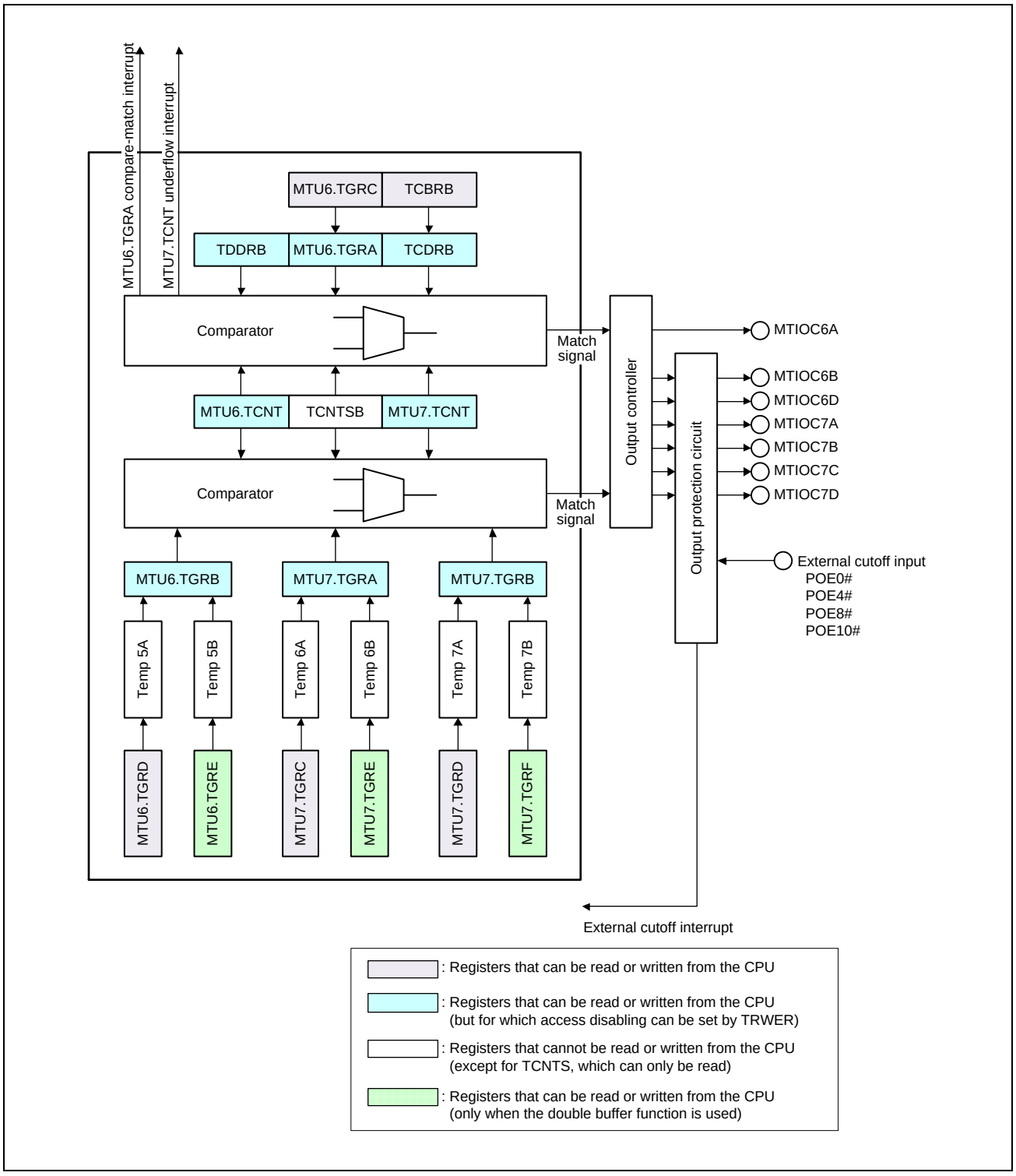
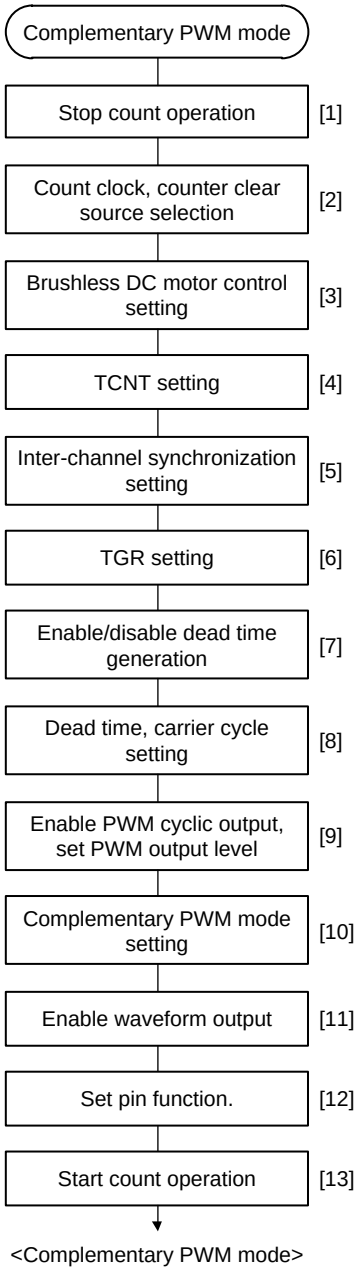



Figure 16.47 Block Diagram of MTU6 and MTU7 in Complementary PWM Mode

### (1) Example of Complementary PWM Mode Setting Procedure

**Figure 16.48** shows an example of the complementary PWM mode setting procedure.




[1] Clear the TSTRA.CST3 (TSTRB.CST6) and TSTRA.CST4 (TSTRB.CST7) bits to 0 to stop the TCNT count operation.
Specify complementary PWM mode while MTU3.TCNT (MTU6.TCNT) and MTU4.TCNT (MTU7.TCNT) are stopped.

[2] Set the TCR.TPSC[2:0] and TCR.CKEG[1:0] bits of MTU3 and MTU4 (MTU6 and MTU7) to select the same count clock source and clock edge for MTU3 and MTU4 (MTU6 and MTU7).
Set the TCR.CCLR[2:0] bits only when restarting by synchronous clearing with another channel during complementary PWM operation.

[3] To control a brushless DC motor, set the TGCRA.BDC bit and set the feedback signal input source and output chopping or gate signal direct output (only when MTU3 and MTU4 are used).

[4] Set the dead time in MTU3.TCNT (MTU6.TCNT). Set MTU4.TCNT (MTU7.TCNT) to H'0000.

[5] Make this setting only when restarting by synchronous clearing with another channel during complementary PWM mode operation. In this case, set the TSYRA register (TSYRB) to synchronize the channel generating the synchronous clear with MTU3 and MTU4 (MTU6 and MTU7).

[6] When MTU3 and MTU6 are used, set the PWM output duty ratio in the duty registers (MTU3.TGRB, MTU4.TGRA, and MTU4.TGRB) and buffer registers (MTU3.TGRD, MTU4.TGRC, and MTU4.TGRD).
Be sure to set the same value in each pair of the duty registers (MTU3.TGRD, MTU4.TGRC, and MTU4.TGRD) and buffer registers (MTU3.TGRB, MTU4.TGRA, and MTU4.TGRB).
Only when the double buffer function is used, set the PWM output duty value − 1 in the buffer registers (MTU3.TGRE, MTU4.TGRE, and MTU4.TGRF).
When MTU6 and MTU7 are used, set the value to the following registers:
Duty registers (MTU6.TGRB, MTU7.TGRA, and MTU7.TGRB)
Buffer registers (MTU6.TGRD, MTU7.TGRC, and MTU7.TGRD)
Buffer registers when the double buffer function is used (MTU6.TGRE, MTU7.TGRE, and MTU7.TGRF)

[7] This setting is necessary only when no dead time should be generated. Make appropriate settings in the TDERA (TDERB) register so that no dead time generated.

[8] Set the dead time in the TDDRA (TDDRB) register, 1/2 the PWM cycle in the TCDRA (TCDRB) register and TCBRA (TCBRB) register, and 1/2 the PWM cycle plus the dead time in MTU3.TGRA (MTU6.TGRA) and MTU3.TGRC (MTU6.TGRC).
When no dead time generation is selected, set 1 in the TDDRA (TDDRB) register and 1/2 the PWM cycle + 1 in MTU3.TGRA (MTU6.TGRA) and MTU3.TGRC (MTU6.TGRC).

[9] Set the PSYE bit in the TOCR1A register to enable or disable toggle output synchronized with the PWM cycle, and set the OLSP and OLSN bits to select the PWM output level. When specifying the PWM output level using the TOCR2A (TOCR2B) and TOLBRA (TOLBRA) in buffer operation, see Timer Output Level Buffer Registers (TOLBRA and TOLBRB).

[10] Select complementary PWM mode in MTU3.TMDR1 (MTU6.TMDR1). Specify the buffer operation (BFA = 1) of MTU3.TGRA (MTU6.TGRA) and the buffer operation (BFB = 1) of MTU3.TGRB (MTU6.TGRB).
Only when the double buffer function is used, set the TMDR2A (TMDR2B) register to enable the double buffer function.

[11] Set the TOERA (TOERB) register to enable or disable output from the PWM output pin. Then, set the MTU3.TIORH (MTU6.TIORH), MTU3.TIORL (MTU6.TIORL), MTU4.TIORH (MTU7.TIORH), and MTU4.TIORL (MTU7.TIORL) to H'00.

[12] Set the port mode registers (PMn) and Pmn port function control register (PFCm) of the GPIO ports.

[13] Set the TSTRA.CST3 (TSTRB.CST6) and TSTRA.CST4 (TSTRB.CST7) bits to 1 simultaneously to start count operation.

Figure 16.48 Example of Complementary PWM Mode Setting Procedure

### (2) Outline of Complementary PWM Mode Operation

In complementary PWM mode, six phases (three positive and three negative) PWM waveforms can be output. **Figure 16.49** illustrates counter operation in complementary PWM mode (MTU3 and MTU4), and **Figure 16.50** shows an example of operation in complementary PWM mode.

#### (a) Counter Operation

In complementary PWM mode, three counters — MTU3.TCNT, MTU4.TCNT, and TCNTSA (MTU6.TCNT, MTU7.TCNT, and TCNTSB) — in each unit perform up-/down-count operations.

MTU3.TCNT (MTU6.TCNT) is automatically initialized to the value set in TDDRA (TDDRB) when complementary PWM mode is selected and the CST3 bit in TSTRA (TSTRB) is 0. When the CST3 bit is set to 1, MTU3.TCNT (MTU6.TCNT) counts up to the value set in MTU3.TGRA (MTU6.TGRA), then switches to down-counting when it matches MTU3.TGRA (MTU6.TGRA). When the MTU4.TCNT (MTU7.TCNT) value matches H’0000, MTU3.TCNT (MTU6.TCNT) switches to up-counting, and the operation is repeated in this way.

TCNT in MTU4 (MTU7) should be initialized to H’0000 after a reset. When the CST4 bit is set to 1, MTU4.TCNT (MTU7.TCNT) counts up in synchronization with MTU3.TCNT (MTU6.TCNT), and switches to down-counting when MTU3.TCNT (MTU6.TCNT) matches MTU3.TGRA (MTU6.TGRA). On reaching H’0000, MTU4.TCNT (MTU7.TCNT) switches to up-counting, and the operation is repeated in this way. TCNTSA (TCNTSB) is a read-only counter. It does not need to be initialized after a reset.

In counting up by MTU3.TCNT and MTU4.TCNT (or MTU6.TCNT and MTU7.TCNT), MTU3.TCNT (or MTU6.TCNT) starts counting up when it matches TCDRA (or TCDRB) and switches to counting down when it matches MTU3.TGRA (or MTU6.TGRA). Furthermore, when MTU4.TCNT (or MTU7.TCNT) matches TDDRA (or TDDRB), TCNTSA (or TCNTSB) is set to the value in MTU3.TGRA (or MTU6.TGRA) and counting is stopped.

When MTU4.TCNT (MTU7.TCNT) matches TDDRA (TDDRB) during down-counting of MTU3.TCNT and MTU4.TCNT (MTU6.TCNT and MTU7.TCNT), TCNTSA (TCNTSB) starts up-counting, and when MTU4.TCNT (MTU7.TCNT) matches H’0000, the operation switches to down-counting. When MTU3.TCNT (MTU6.TCNT) matches TCDRA (TCDRB), TCNTSA (TCNTSB) is cleared to H’0000 and stops counting.

TCNTSA (TCNTSB) is compared with the compare register and temporary register, in which the PWM duty is specified, only during the count operation.

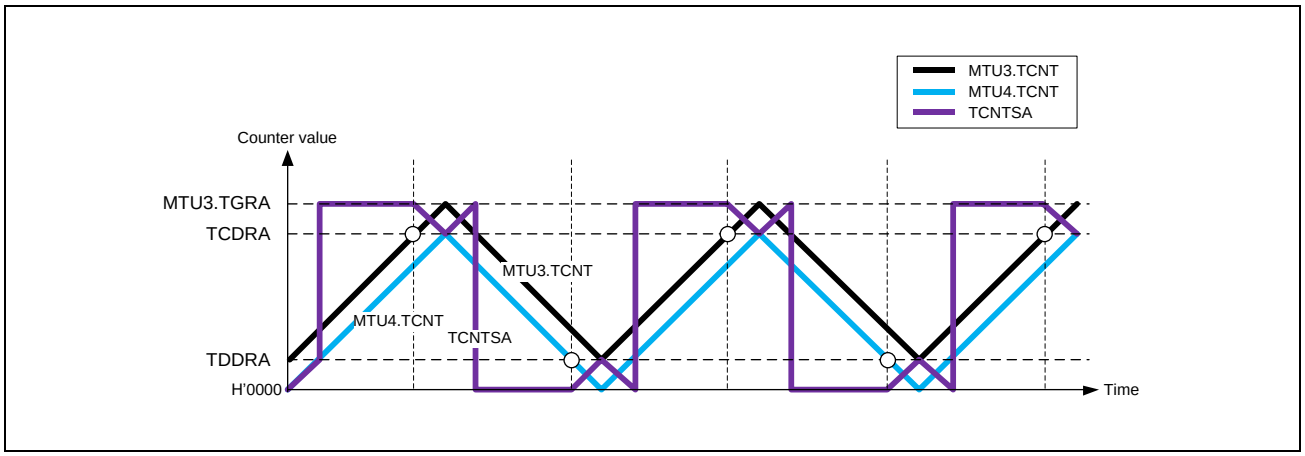



Figure 16.49 Count Operation in Complementary PWM Mode (MTU3 and MTU4)

### (b) Register Operation

In complementary PWM mode, nine registers (compare registers, buffer registers, and temporary registers) are used to control the duty ratio for the PWM output. **Figure 16.50** shows an example of operation in complementary PWM mode (MTU3 and MTU4).

MTU3.TGRB, MTU4.TGRA, and MTU4.TGRB (MTU6.TGRB, MTU7.TGRA, and MTU7.TGRB) are constantly compared with the counters to generate PWM waveforms. When these registers match the counter, the value set in the OLSN and OLSP bits in the timer output control registers (TOCR1A and TOCR1B) is output from the PWM output pin. MTU3.TGRD, MTU4.TGRC, and MTU4.TGRD (MTU6.TGRD, MTU7.TGRC, and MTU7.TGRD) are buffer registers for these compare registers.

When the double buffer function is used, MTU3.TGRE, MTU4.TGRE, and MTU4.TGRF (MTU6.TGRE, MTU7.TGRE, and MTU7.TGRF) are also used as buffer registers B. For details of double buffer operation, refer to **Section 16.3.8(2)(s), Double Buffer Function in Complementary PWM Mode**.

Data in a compare register can be changed by writing new data to the corresponding buffer register. The buffer registers can be read or written at any time.

When modifying data in a buffer register, write to MTU4.TGRD (MTU7.TGRD) last and enable data transfer from the buffer register to a temporary register. At this time, transfer from the TCBRA (TCBRB) register and MTU3.TGRC (MTU6.TGRC) register, which operate as buffer registers for the timer cycle registers, to temporary registers is also enabled. Data is transferred to all five temporary registers at the same time.

When transfer is enabled in the Ta interval, data written to a buffer register is transferred to the temporary register. The data is not transferred to the temporary register in the Tb1 and Tb2 intervals. Data enabled for transfer in this interval is transferred to the temporary register at the end of this interval.

The value transferred to a temporary register is transferred to the compare register at the end of the Tb1 interval (when matches MTU3.TGRA (MTU6.TGRA) while TCNTSA (TCNTSB) is counting up), or at the end of the Tb2 interval (when matches H’0000 while TCNTSA (TCNTSB) is counting down). The timing for transfer from the temporary register to the compare register can be selected with bits MD[3:0] in the timer mode register 1 (TMDR1). **Figure 16.50** shows an example in which the trough is selected for the transfer timing.

In the Tb interval in which data is not transferred to the temporary register (Tb1 in **Figure 16.50**), 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 output phase; the compare register contains the pre-change data and the temporary register contains new data. In this interval, three counters MTU3.TCNT, MTU4.TCNT, and TCNTSA (MTU6.TCNT, MTU7.TCNT, and TCNTSB) and two registers (compare register and temporary register) are compared, and PWM output is controlled accordingly.

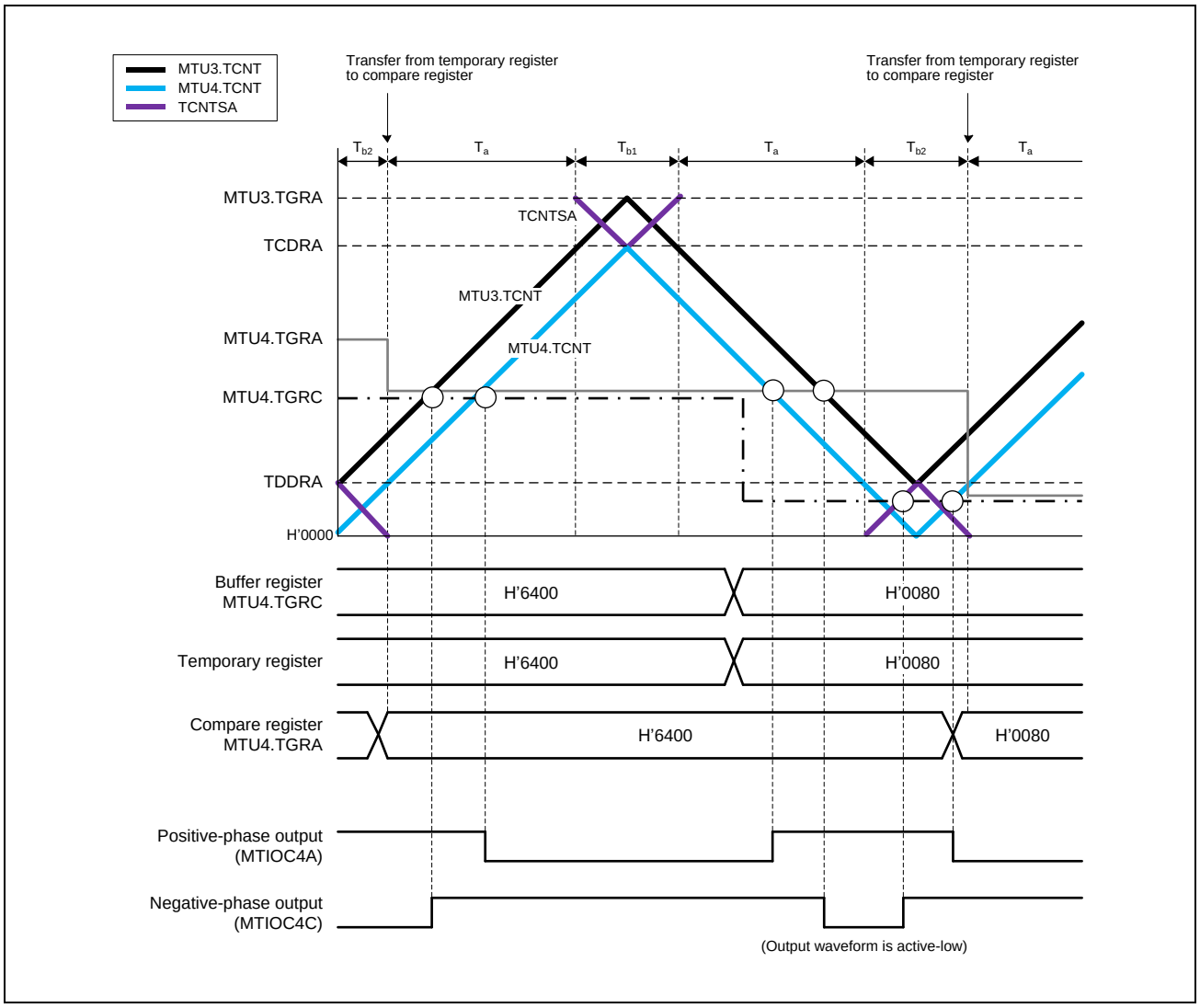



Figure 16.50 Example of Operation in Complementary PWM Mode (MTU3 and MTU4)

### (c) Initial Setting

In complementary PWM mode, there are nine registers that require initial setting. 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 MTU3.TMDR1.MD[3:0] (MTU6.TMDR1.MD[3:0]) bits, initial values should be set in the following registers.

The TOCR1A, TOCR2A, TOCR1B, and TOCR2B registers are used to set the PWM output level. MTU3.TGRC (MTU6.TGRC) operates as the buffer register for MTU3.TGRA (MTU6.TGRA), and should be set with 1/2 the PWM cycle + dead time Td. The timer cycle buffer register (TCBRA or TCBRB) operates as the buffer register for the timer cycle data register (TCDRA or TCDRB), and should be set with 1/2 the PWM cycle. Set dead time Td in the timer dead time data register (TDDRA or TDDRB).

When dead time is not needed, the TDER bit in the timer dead time enable register (TDERA or TDERB) should be cleared to 0, MTU3.TGRC and MTU3.TGRA (MTU6.TGRC and MTU6.TGRA) should be set to 1/2 the PWM carrier cycle + 1, and TDDRA (TDDRB) should be set to 1.

Set the respective initial PWM duty values in three buffer registers A (MTU3.TGRD, MTU4.TGRC, and MTU4.TGRD (MTU6.TGRD, MTU7.TGRC, and MTU7.TGRD)).

Set the respective (initial PWM duty – 1) values in three buffer registers B (MTU3.TGRE, MTU4.TGRE, and MTU4.TGRF (MTU6.TGRE, MTU7.TGRE, and MTU7.TGRF)) only when the double buffer function is used.

The values set in the five buffer registers excluding TDDRA (TDDRB) are transferred to the corresponding compare registers as soon as complementary PWM mode is set.

Set MTU4.TCNT (MTU7.TCNT) to H’0000 before setting complementary PWM mode.

Table 16.77 Registers and Counters Requiring Initial Setting

| Register and Counter | Setting |
|---|---|
| TOCR1A, TOCR2A, TOCR1B, TOCR2B | PWM output level |
| MTU3.TGRC MTU6.TGRC | 1/2 PWM cycle + dead time Td<br>(1/2 PWM cycle + 1 when dead time generation is disabled by the TDERA or TDERB setting) |
| TDDRA, TDDRB | Dead time Td<br>(1 when dead time generation is disabled by the TDERA or TDERB setting) |
| TCBRA, TCBRB | 1/2 PWM cycle |
| MTU3.TGRD, MTU4.TGRC, MTU4.TGRD<br>MTU6.TGRD, MTU7.TGRC, MTU7.TGRD | Initial PWM duty ratio value for each phase |
| MTU3.TGRE, MTU4.TGRE, MTU4.TGRF<br>MTU6.TGRE, MTU7.TGRE, MTU7.TGRF | Initial PWM duty ratio – 1 value for each phase<br>(only when double buffer function is used) |
| MTU4.TCNT MTU7.TCNT | H’0000 |

**Note:** The value set in MTU3.TGRC (MTU6.TGRC) should be the sum of 1/2 the PWM cycle set in TCBRA (TCBRB) and dead time Td set in TDDRA (TDDRB). When dead time generation is disabled by TDERA (TDERB), TGRC should be set to 1/2 the PWM cycle + 1.

### (d) PWM Output Level Setting

In complementary PWM mode, the PWM output level is set with bits OLSN and OLSP in timer output control register 1 (TOCR1A or TOCR1B) or bits OLS1P to OLS3P and OLS1N to OLS3N in timer output control register 2 (TOCR2A or TOCR2B).

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, dead time can be set for PWM output.

The dead time is set in the timer dead time data register (TDDRA or TDDRB). The value set in TDDRA (TDDRB) is used as the MTU3.TCNT (MTU6.TCNT) counter start value and creates a non-overlapping interval between MTU3.TCNT (MTU6.TCNT) and MTU4.TCNT (MTU7.TCNT). Complementary PWM mode should be cleared before changing the contents of TDDRA (TDDRB).

### (f) Dead Time Suppressing

Dead time generation is suppressed by clearing the TDER bit in the timer dead time enable register (TDERA or TDERB) to 0. TDERA (TDERB) can be cleared to 0 only when 0 is written to it after reading TDER = 1.

MTU3.TGRA and MTU4.TGRC (MTU6.TGRA and MTU7.TGRC) should be set to 1/2 PWM cycle + 1 and the timer dead time data register (TDDRA or TDDRB) should be set to 1.

By the above settings, PWM waveforms without dead time can be obtained. **Figure 16.51** shows an example of operation without dead time (MTU3 and MTU4).

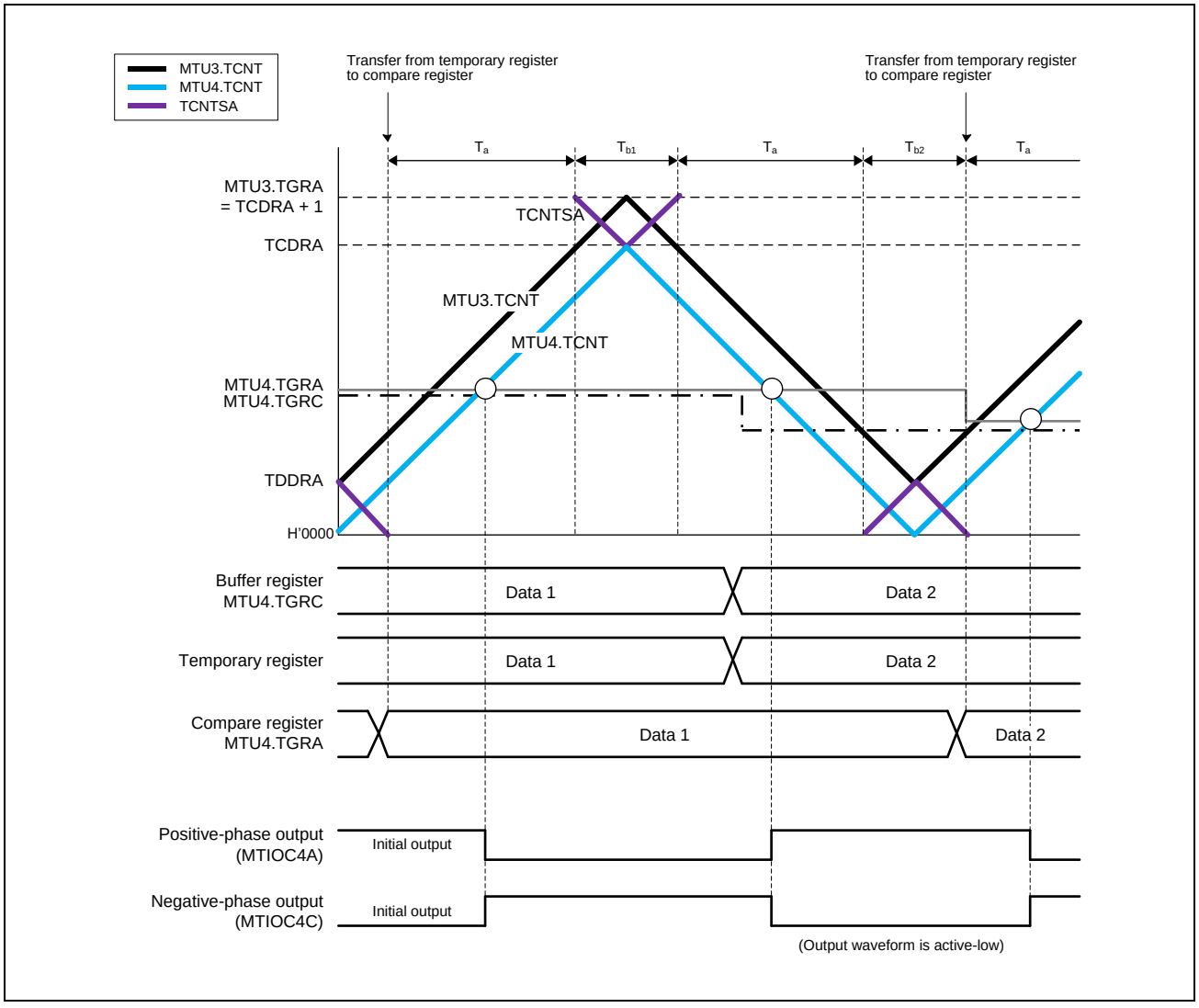



Figure 16.51 Example of Operation without Dead Time (MTU3 and MTU4)

### (g) PWM Cycle Setting

In complementary PWM mode, the PWM cycle is set in two registers—MTU3.TGRA (MTU6.TGRA), in which the MTU3.TCNT (MTU6.TCNT) upper limit value is set, and TCDRA (TCDRB), in which the MTU4.TCNT (MTU7.TCNT) upper limit value is set. The settings should be made so as to achieve the following relationship between these two registers:

With dead time: MTU3.TGRA (MTU6.TGRA) setting = TCDRA (TCDRB) setting + TDDRA (TDDRB) setting
Without dead time: MTU3.TGRA (MTU6.TGRA) setting = TCDRA (TCDRB) setting + 1

In addition, the settings should be made so as to achieve the following relationship between the TCDRA (TCDRB) register and the TDDRA (TDDRB) register:

TCDRA (TCDRB) setting > TDDRA (TDDRB) setting × 2 + 2

The MTU3.TGRA and TCDRA (MTU6.TGRA and TCDRB) settings are made by setting values in buffer registers MTU3.TGRC and TCBRA (MTU6.TGRC and TCBRB). When data is written to MTU4.TGRD (MTU7.TGRD) to enable transfers, the values set in MTU3.TGRC and TCBRA (MTU6.TGRC and TCBRB) are transferred simultaneously to the MTU3.TGRA and TCDRA (MTU6.TGRA and TCDRB) with the transfer timing selected with the MTU3.TMDR1.MD[3:0] (MTU6.TMDR1.MD[3:0]) bits.

The new PWM cycle is reflected from the next cycle when data is updated at the crest, or from the current cycle when updated in the trough. **Figure 16.52** illustrates the operation when the PWM cycle is updated at the crest.

See the following **Section 16.3.8(2)(h), Register Data Updating**, for the method of updating the data in each buffer register.

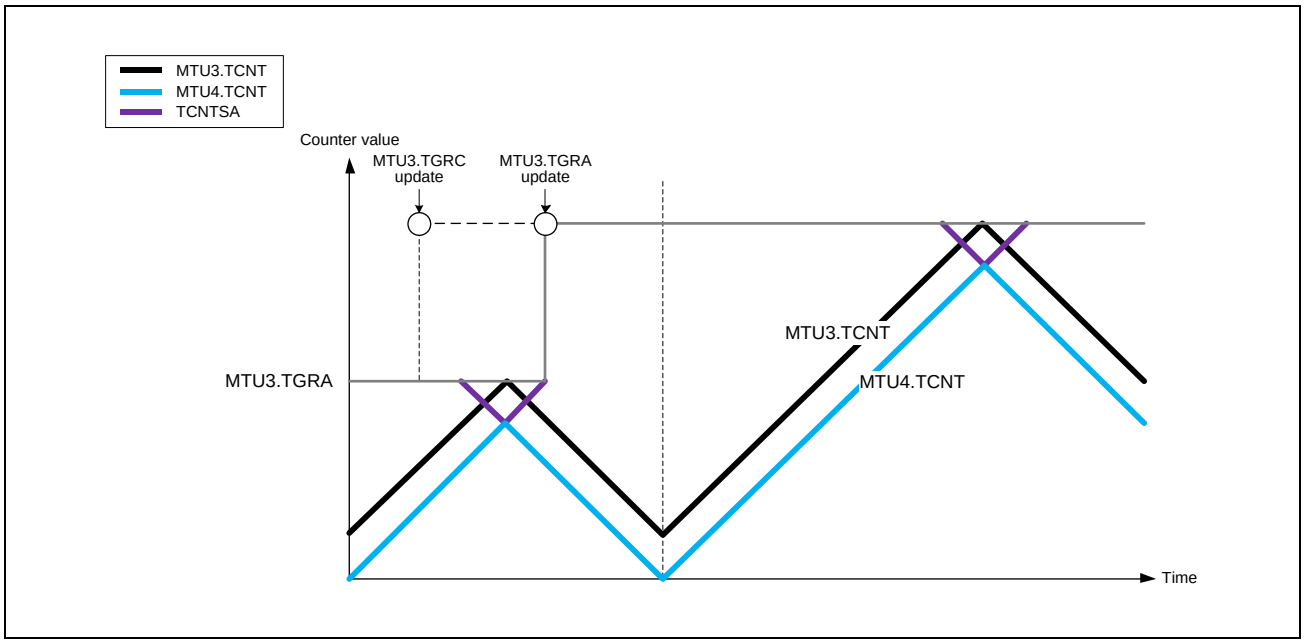



Figure 16.52 Example of PWM Cycle Updating (MTU3 and MTU4)

### (h) Register Data Updating

In complementary PWM mode, buffer registers are used to update the data in five compare registers for PWM duty and PWM cycle. The update data can be written to the buffer registers at any time.

There is a temporary register between each of these registers and its buffer register. While subcounter TCNTSA (TCNTSB) is not counting, if buffer register data is updated, the temporary register value also changes. Data is not transferred from buffer registers to temporary registers while TCNTSA (TCNTSB) is counting; in this case, the value written to a buffer register is transferred after TCNTSA (TCNTSB) halts.

The temporary register value is transferred to the compare register at the data update timing set with MTU3.TMDR1.MD[3:0] (MTU6.TMDR1.MD[3:0]) bits. **Figure 16.53** shows an example of data updating in complementary PWM mode (MTU3 and MTU4). This example shows the mode in which data is updated at both the counter crest and trough.

When updating buffer register data, be sure to write to MTU4.TGRD (MTU7.TGRD) at the end of the update. Data is transferred from buffer registers to the temporary registers simultaneously for all five registers after the write to MTU4.TGRD (MTU7.TGRD).

Even when not updating all five registers or when not updating the MTU4.TGRD (MTU7.TGRD) data, be sure to write to MTU4.TGRD (MTU7.TGRD) after writing data to the registers to be updated. In this case, the data written to MTU4.TGRD (MTU7.TGRD) should be the same as the data prior to the write operation.

See **Section 16.3.8(2)(s), Double Buffer Function in Complementary PWM Mode**, for data updating when the double buffer function is used.

● Data update timing: counter crest and trough

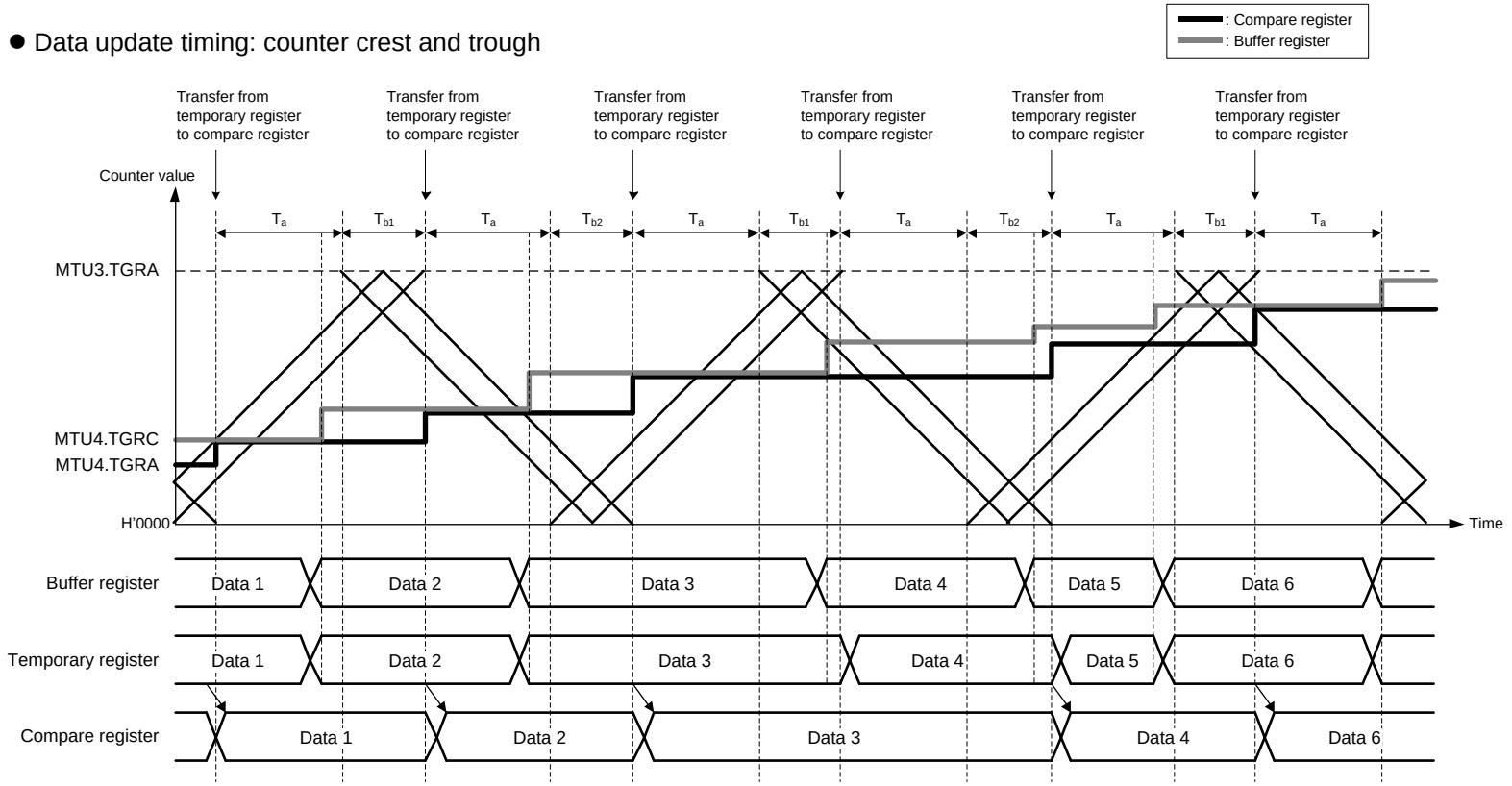



Figure 16.53 Example of Data Updating in Complementary PWM Mode (MTU3 and MTU4)

### (i) Initial Output in Complementary PWM Mode

In complementary PWM mode, the initial output is determined by the setting of the OLSN and OLSP bits in the TOCR1A (TOCR1B) register or the OLS1N to OLS3N and OLS1P to OLS3P bits in the TOCR2A register (TOCR2B). This initial output is the non-active level of the PWM output and continues from when complementary PWM mode is set with the MTU3.TMDR1 (MTU6.TMDR1) until MTU4.TCNT (MTU7.TCNT) exceeds the value set in the TDDRA (TDDRB) register. **Figure 16.54** shows an example of the initial output in complementary PWM mode.

An example of the waveform when the initial PWM duty ratio value is smaller than the TDDRA (TDDRB) value is shown in **Figure 16.55**.

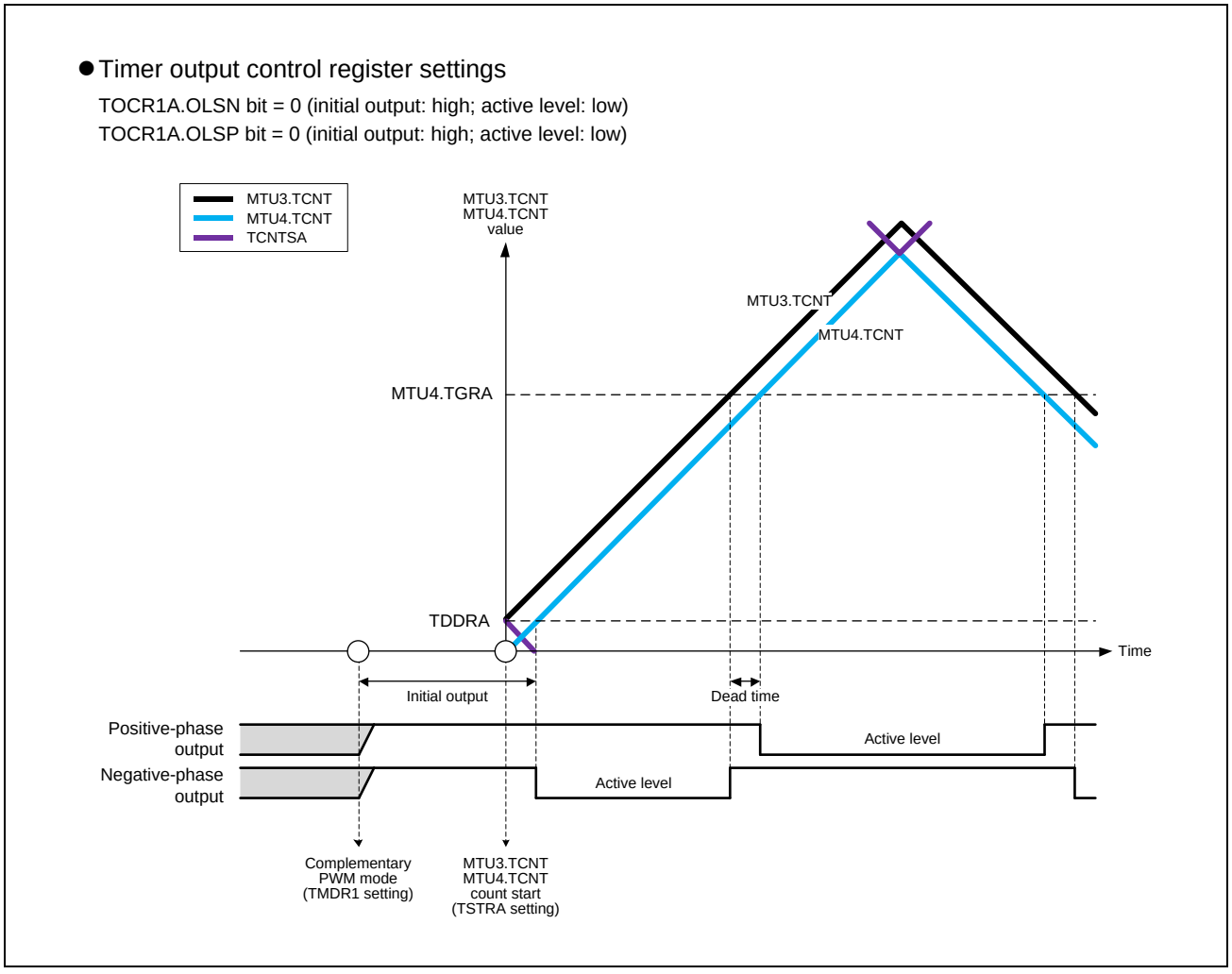



Figure 16.54 Example of Initial Output in Complementary PWM Mode (MTU3 and MTU4) (1)

● Timer output control register settings

TOCR1A.OLSN bit = 0 (initial output: high; active level: low)
TOCR1A.OLSP bit = 0 (initial output: high; active level: low)

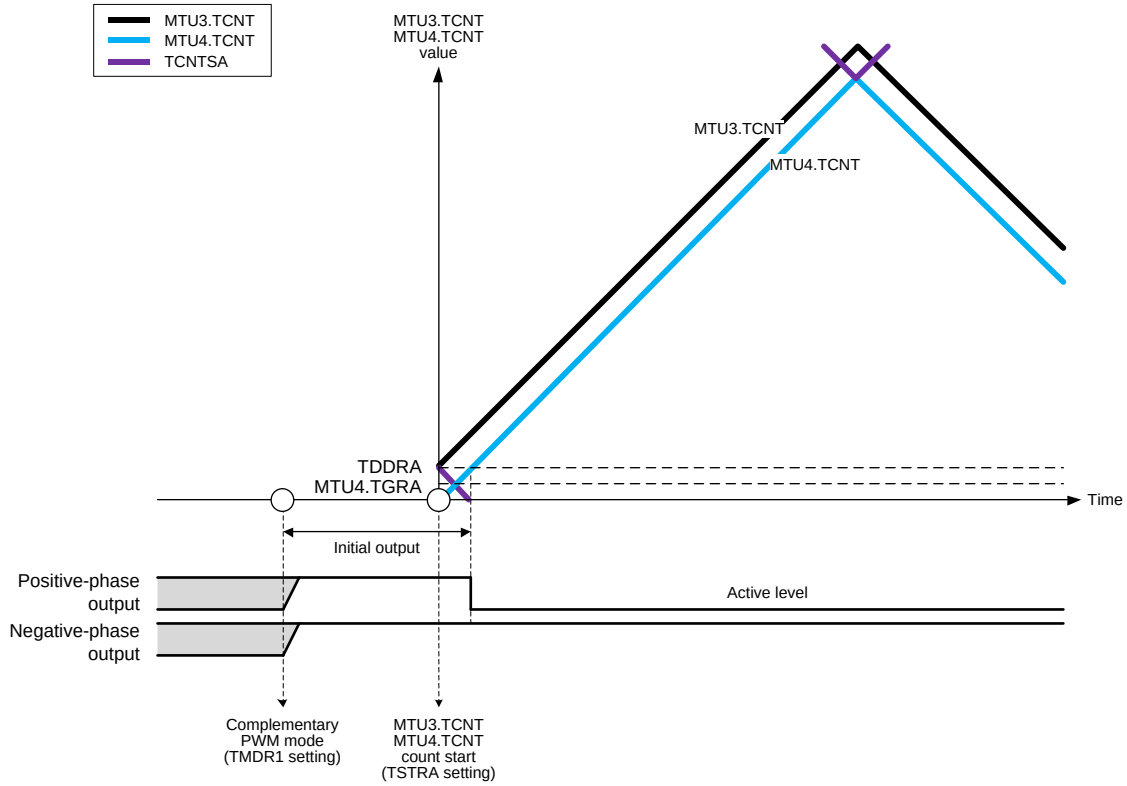



Figure 16.55 Example of Initial Output in Complementary PWM Mode (MTU3 and MTU4) (2)

### (j) Method for Generating PWM Output in Complementary PWM Mode

In complementary PWM mode, six phases (three positive and three negative) PWM waveforms can be output. Dead time can be set for PWM waveforms to be output.

A PWM waveform is generated by output of the level selected in the timer output control register in the event of a compare match between a counter and a compare register. While TCNTSA (TCNTSB) is counting, the compare register and temporary register values are simultaneously compared to create consecutive PWM output from 0 to 100% duty ratio. The relative timing of turn-on and turn-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 turn-on times do not overlap. **Figure 16.56** to **Figure 16.58** show examples of waveform generation in complementary PWM mode.

The positive-phase and negative-phase turn-off timing is generated by a compare match with the counter indicated by a solid line, and the turn-on timing is generated by a compare match with the counter indicated by a dotted line, which operates with a delay equal to the dead time behind the counter indicated by a solid line. In the T1 period, compare match a that turns off the negative phase has the highest priority, and compare matches before a are ignored. In the T2 period, compare match c that turns off the positive phase has the highest priority, and compare matches before c are ignored.

In most cases, compare matches occur in the order a → b → c → d (or c → d → a' → b') as shown in **Figure 16.56**.

If compare matches deviate from the a → b → c → d order, since the time for which the negative phase is off is shorter than twice the dead time, the positive phase is not turned on. If compare matches deviate from the c → d → a' → b' order, since the time for which the positive phase is off is shorter than twice the dead time, the negative phase is not turned on. As shown in **Figure 16.57**, if compare match c follows compare match a before compare match b, compare match b is ignored and the negative phase is turned on by compare match d. This is because turning off 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 because the positive phase goes from off to off).

Similarly, in the example in **Figure 16.58**, turning off the negative phase has priority due to the occurrence of compare match a' (negative-phase off timing) before compare match d (negative-phase on timing). 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.

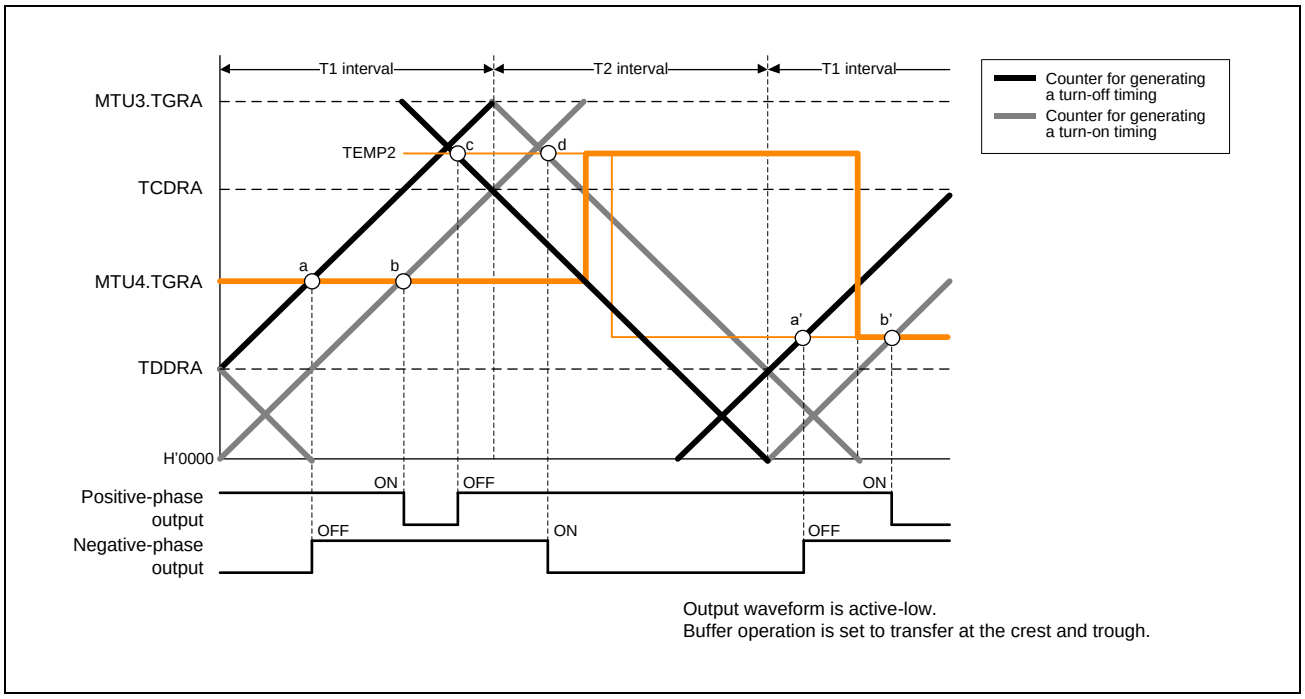



Figure 16.56 Example of Waveform Output in Complementary PWM Mode (MTU3 and MTU4) (1)

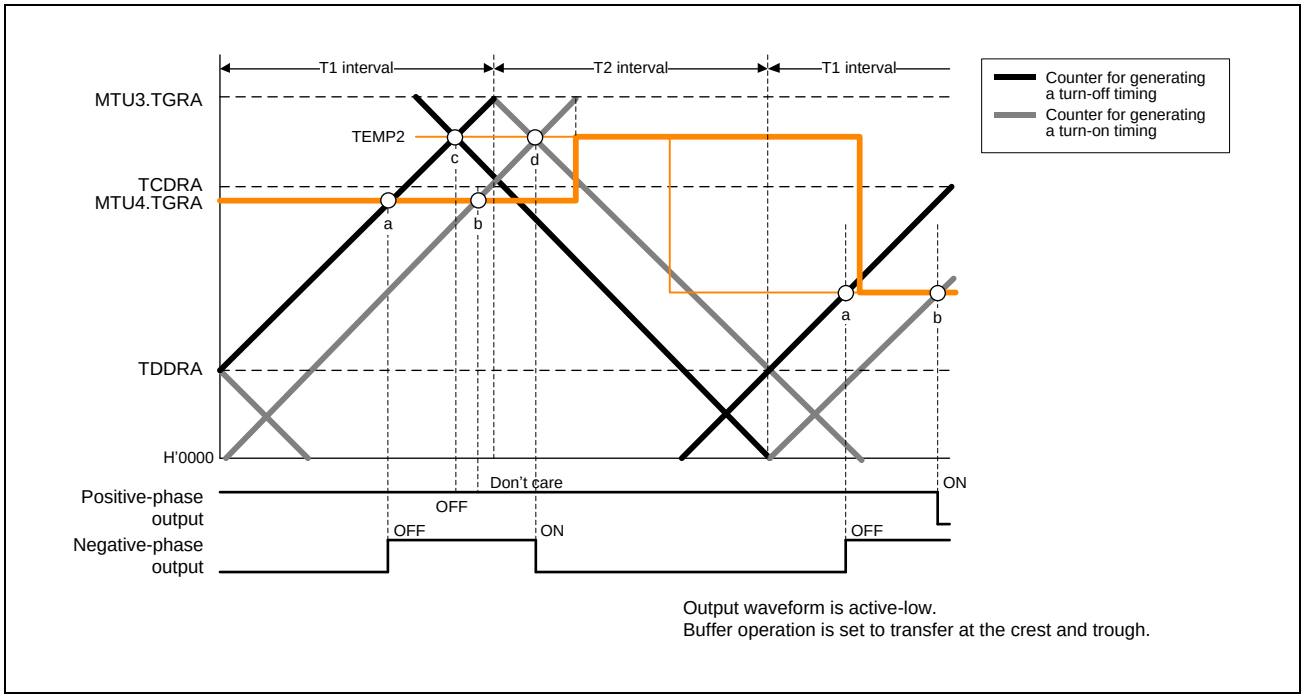



Figure 16.57 Example of Waveform Output in Complementary PWM Mode (MTU3 and MTU4) (2)

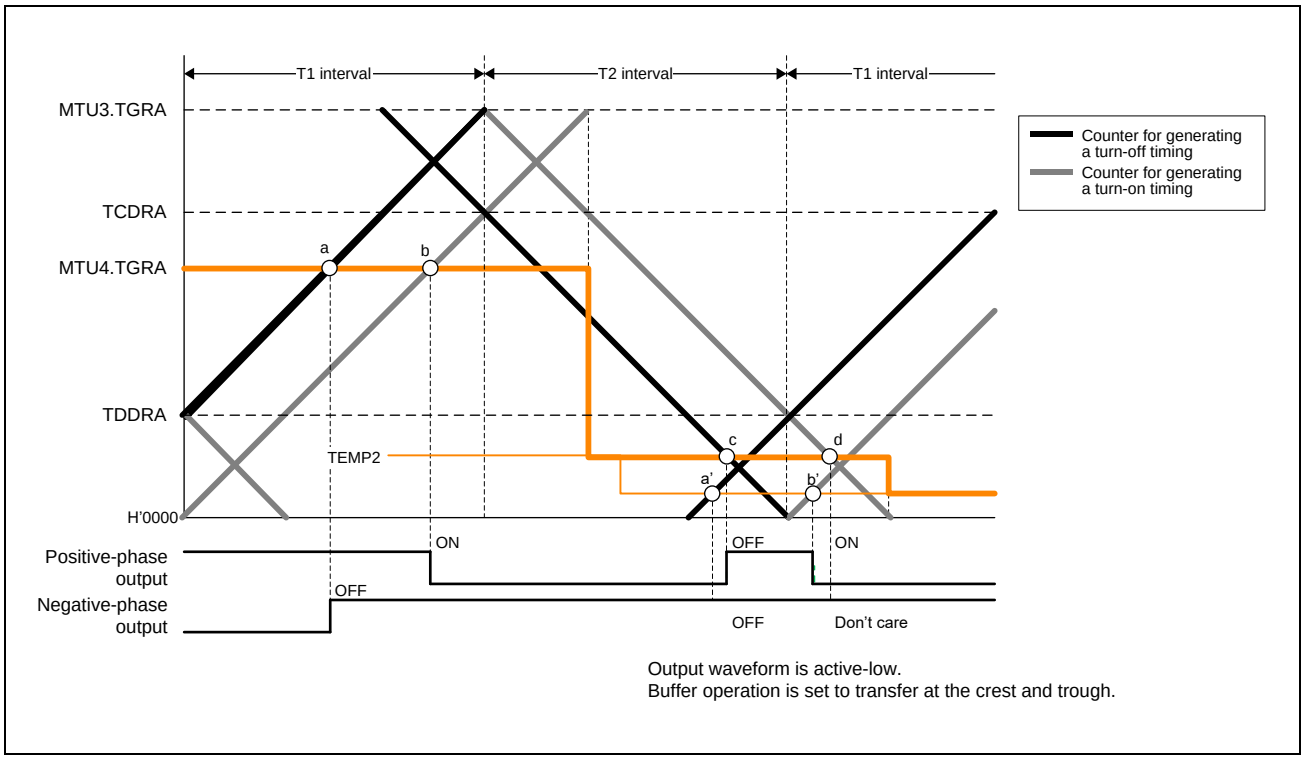



Figure 16.58 Example of Waveform Output in Complementary PWM Mode (MTU3 and MTU4) (3)

### (k) 0% and 100% Duty Ratio Output in Complementary PWM Mode

In complementary PWM mode, 0% and 100% duty PWM output can be output as required. **Figure 16.59** to **Figure 16.63** show output examples.

A 100% duty waveform is output when the compare register value is set to H'0000. The waveform in this case has a positive phase with a 100% on-state. A 0% duty waveform is output when the compare register value is set to the same value as MTU3.TGRA (MTU6.TGRA). The waveform in this case has a positive phase with a 100% off-state.

Turn-on and turn-off compare matches occur simultaneously, but if a turn-on compare match and turn-off compare match for the same phase occur simultaneously, both compare matches are ignored and the waveform does not change.

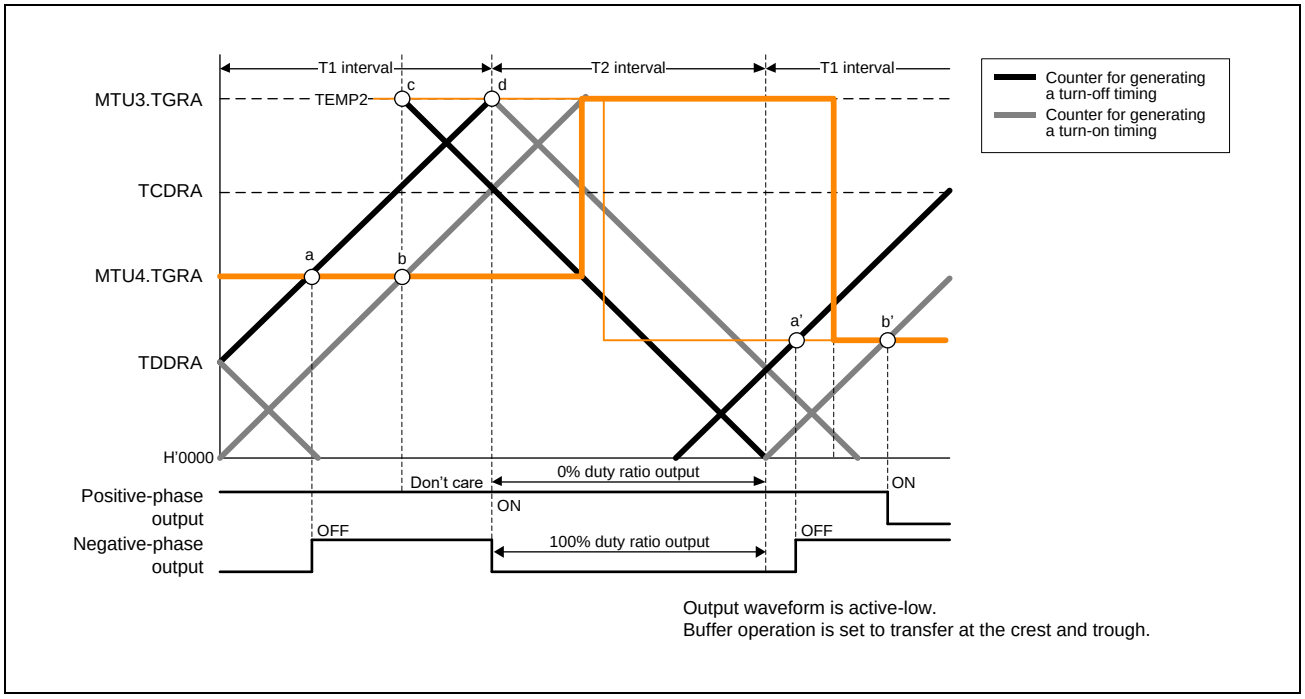



Figure 16.59 Example of 0% and 100% Waveform Output in Complementary PWM Mode (MTU3 and MTU4) (1)

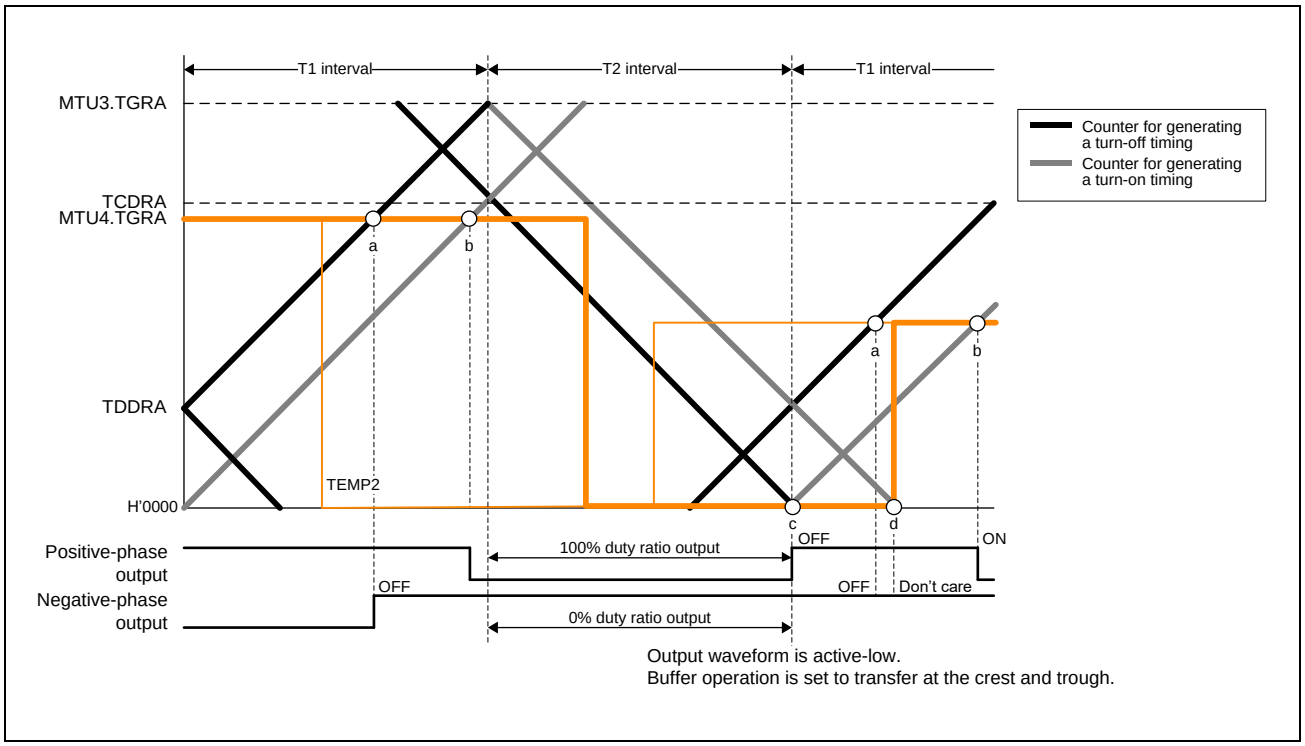



Figure 16.60 Example of 0% and 100% Waveform Output in Complementary PWM Mode (MTU3 and MTU4) (2)

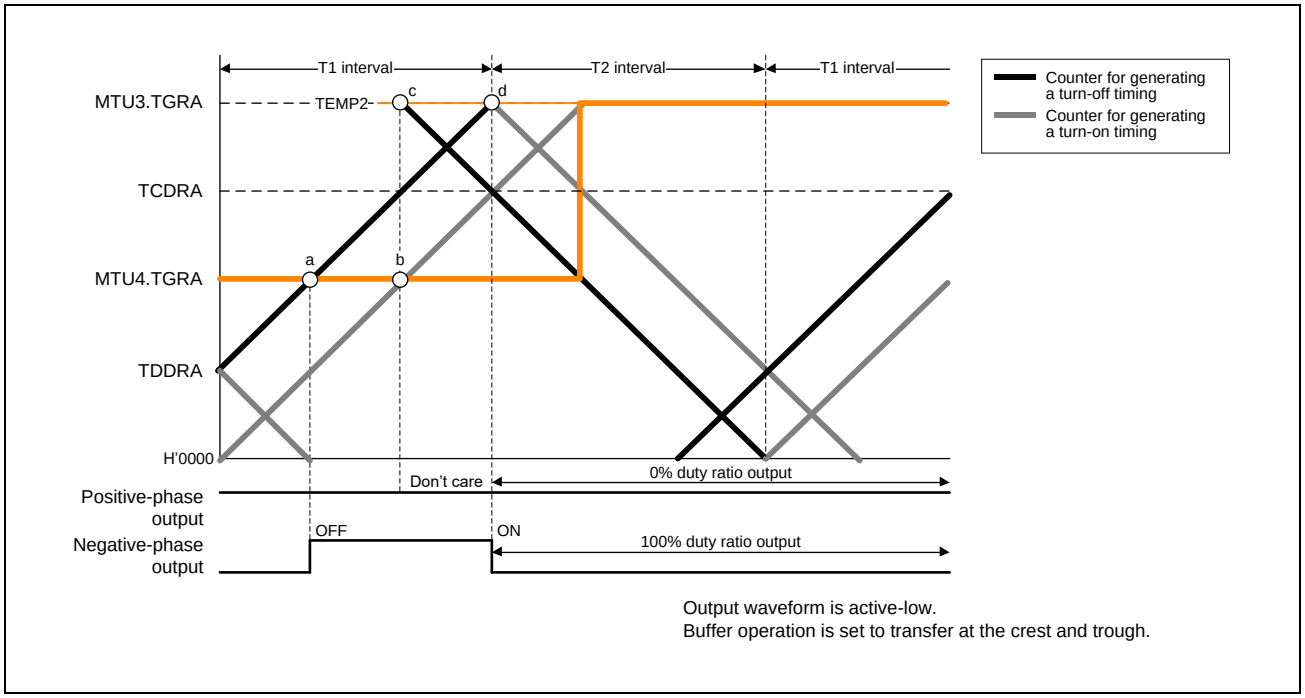



Figure 16.61 Example of 0% and 100% Waveform Output in Complementary PWM Mode (MTU3 and MTU4) (3)

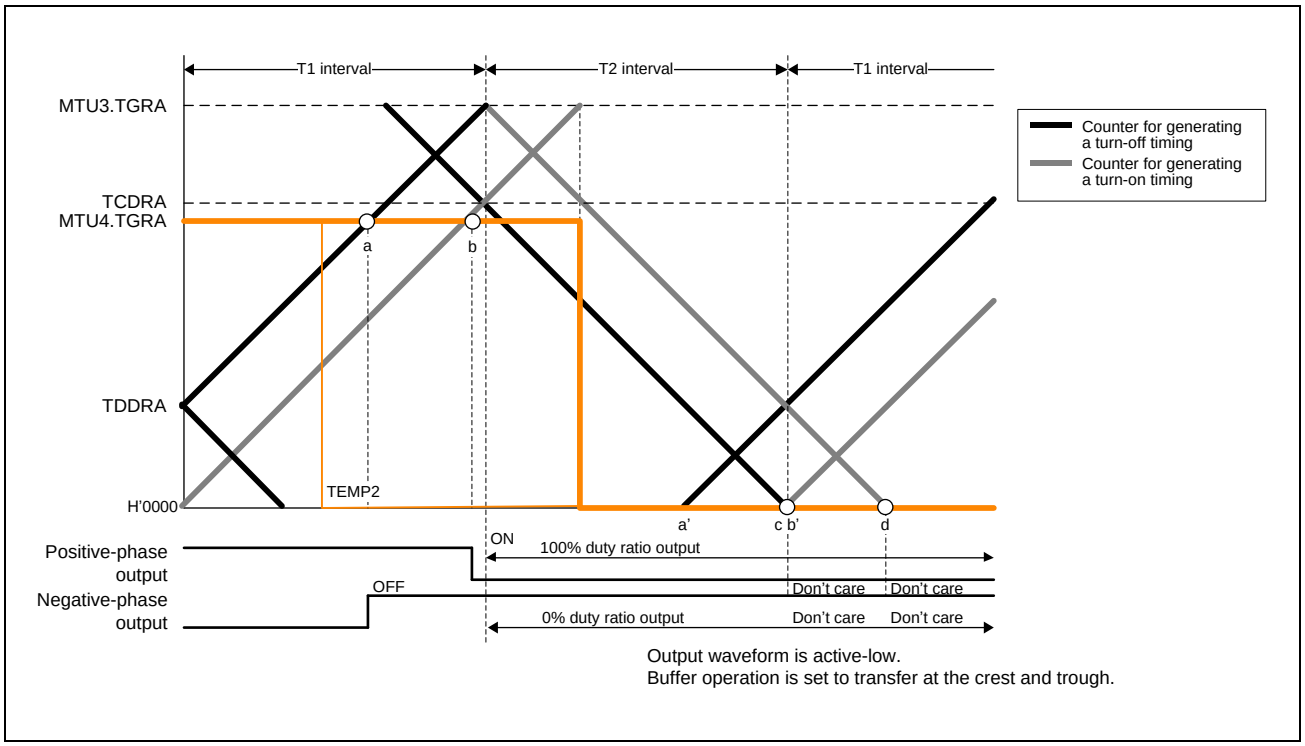



Figure 16.62 Example of 0% and 100% Waveform Output in Complementary PWM Mode (MTU3 and MTU4) (4)

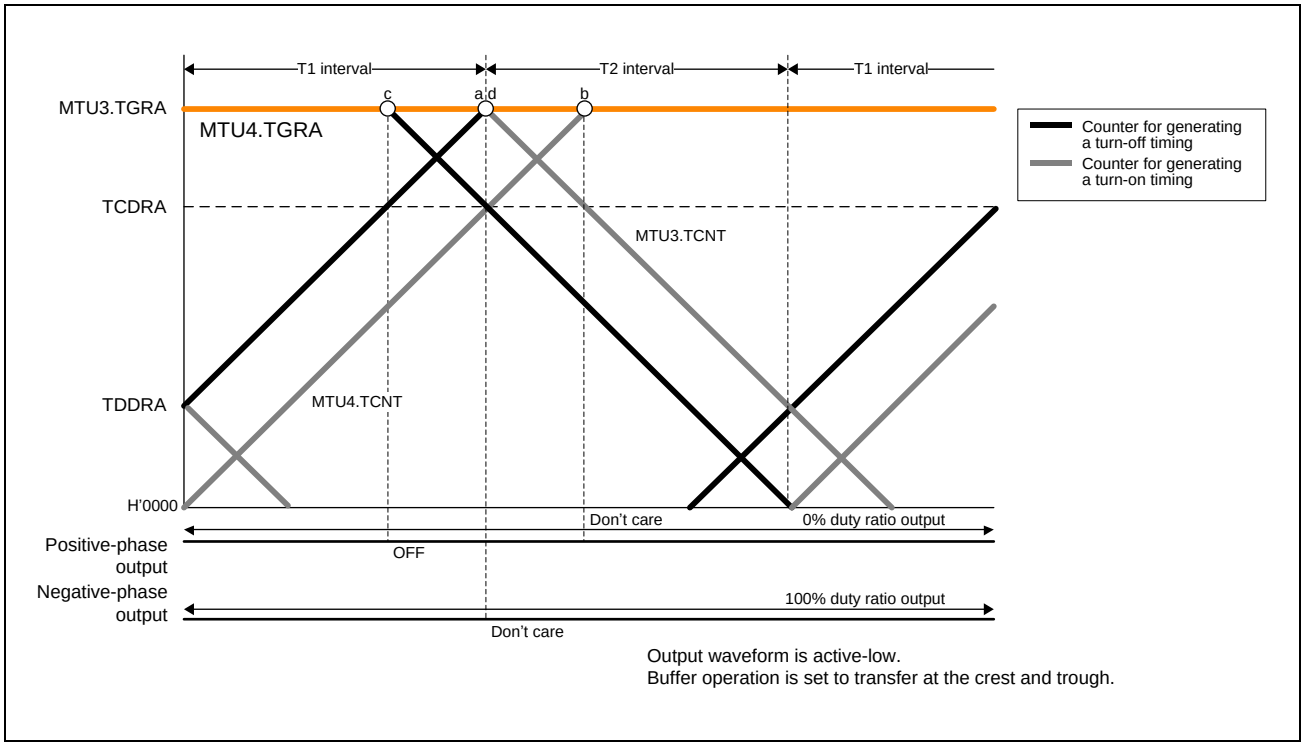



Figure 16.63 Example of 0% and 100% Waveform Output in Complementary PWM Mode (MTU3 and MTU4) (5)

### (l) Toggle Output Synchronized with PWM Cycle

In complementary PWM mode, toggle output from the PWM output pin in synchronization with the PWM cycle can be enabled by setting the PSYE bit in the TOCR1A (TOCR1B) register to 1. An example of a toggle output waveform is shown in **Figure 16.64**.

This output is toggled by a compare match between MTU3.TCNT and MTU3.TGRA (MTU6.TCNT and MTU6.TGRA) and a compare match between MTU4.TCNT (MTU7.TCNT) and H’0000.

The MTIOC3A (MTIOC6A) pin is assigned for this toggle output. The initial output is high-level output.

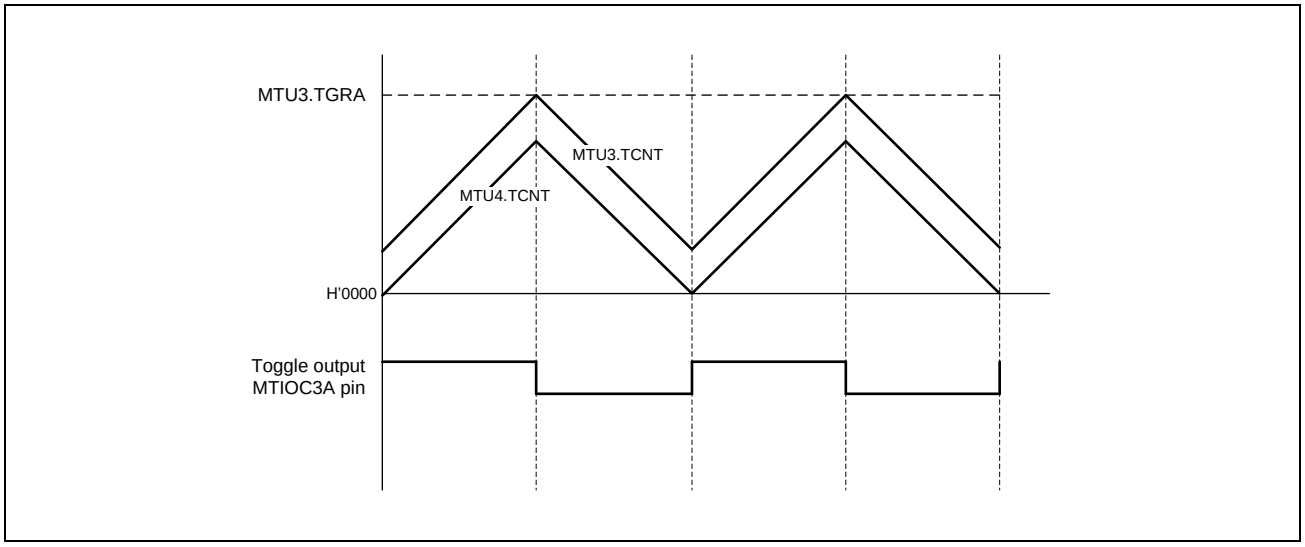



Figure 16.64 Example of Toggle Output Waveform Synchronized with PWM Output (MTU3 and MTU4)

### (m) Counter Clearing by Another Channel

In complementary PWM mode, MTU3.TCNT, MTU4.TCNT, and TCNTSA (MTU6.TCNT, MTU7.TCNT, and TCNTSB) can be cleared by another channel source when a mode for synchronization with another channel is specified through the TSYRA (TSYRB) register and synchronous clearing is selected with MTU3.TCR.CCLR[2:0] (MTU6.TCR.CCLR[2:0]) bits.

**Figure 16.65** illustrates an example of this operation.

Use of this function enables a counter to be cleared and restarted through an external signal.

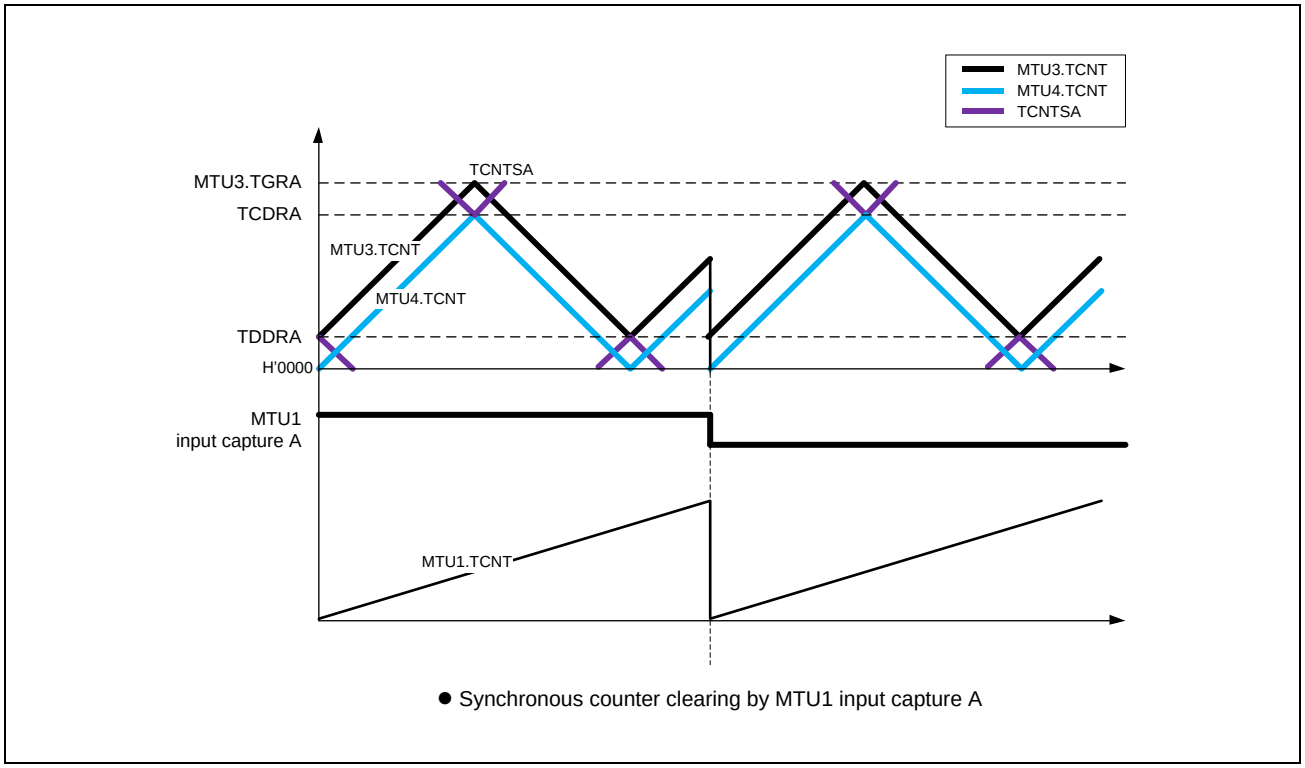



Figure 16.65 Counter Clearing Synchronized with Another Channel (MTU3 and MTU4)

### (n) Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

Setting the WRE bit in TWCRA (TWCRB) to 1 suppresses initial output when synchronous counter clearing occurs in the Tb interval (Tb2 interval) at the trough in complementary PWM mode and controls abrupt change in duty cycle at synchronous counter clearing.

Initial output suppression through the WRE bit = 1 is applicable only when synchronous clearing occurs in the Tb2 interval as indicated by (10) or (11) in **Figure 16.66**. When synchronous clearing occurs outside that interval, the initial value specified by the OLSN and OLSP bits in TOCR1A (TOCR1B) is output. Even in the Tb2 interval, if synchronous clearing occurs in the initial value output period (indicated by (1) in **Figure 16.66**) immediately after the counters start operation, initial value output is not suppressed.

This function can be used in both channel combinations of MTU 3 and MTU4 and MTU6 and MTU7. In MTU3 and MTU4, synchronous clearing in MTU0, MTU1, and MTU2 can cause counter clearing; in MTU6 and MTU7, compare match or input capture generated in MTU0, MTU1, and MTU2 can cause counter clearing.

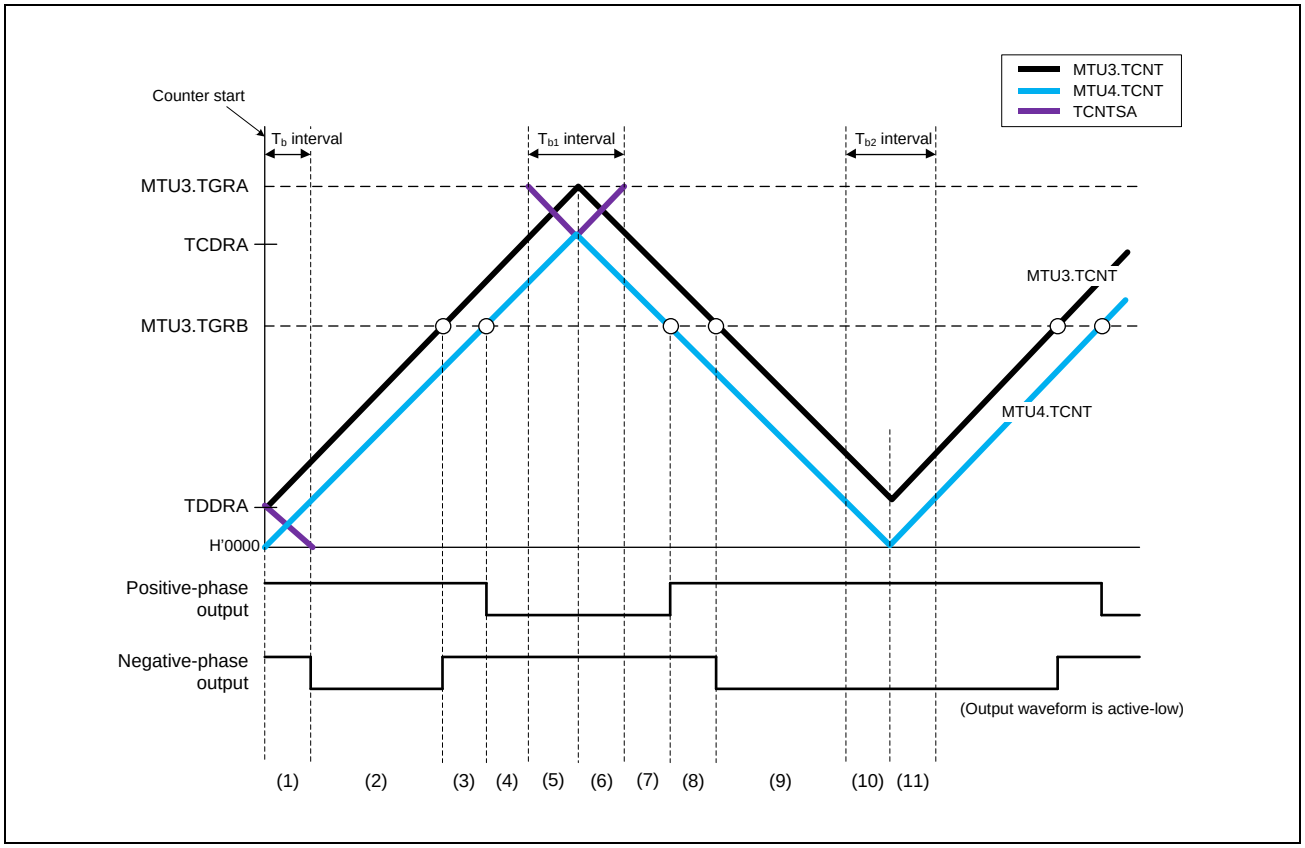



Figure 16.66 Timing for Synchronous Counter Clearing (MTU3 and MTU4)

- 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 16.67**.

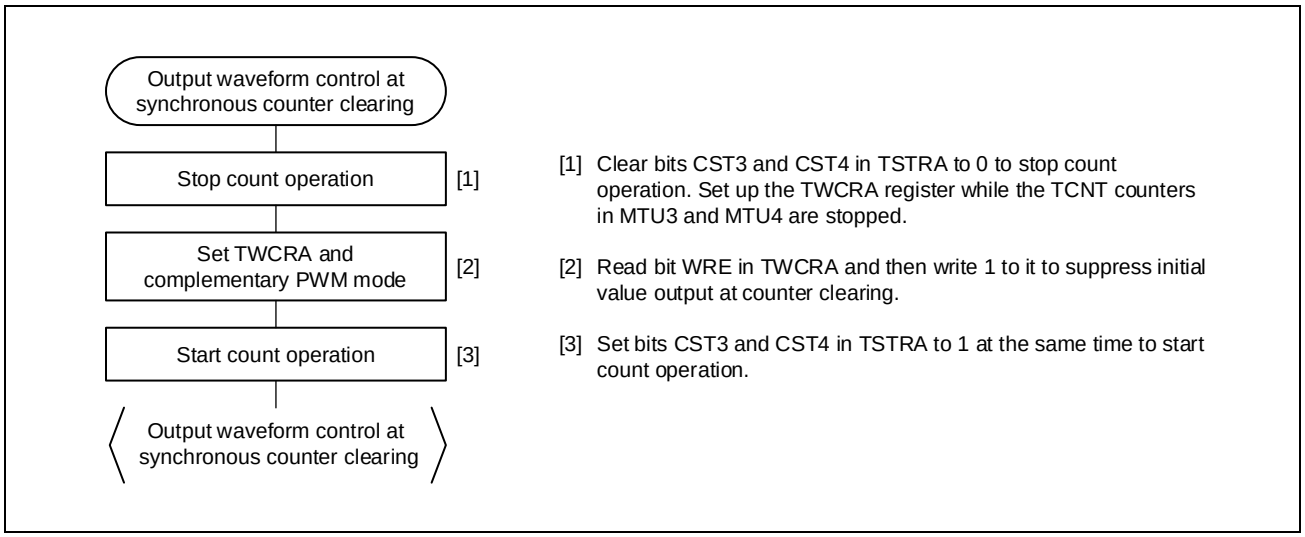



Figure 16.67 Example of Procedure for Setting Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode (MTU3 and MTU4)

- Examples of Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

**Figure 16.68** to **Figure 16.71** show examples of output waveform control in which MTU3 and MTU4 operate in complementary PWM mode and synchronous counter clearing is generated while the WRE bit in TWCRA is set to 1. In the examples shown in **Figure 16.68** to **Figure 16.71**, synchronous counter clearing occurs at timing (3), (6), (8), and (11) shown in **Figure 16.66**, respectively.
In MTU6 and MTU7, these examples are equivalent to the cases when MTU6 and MTU7 operate in complementary PWM mode and synchronous counter clearing is generated while the SCC bit is cleared to 0 and the WRE bit is set to 1 in TWCRB.

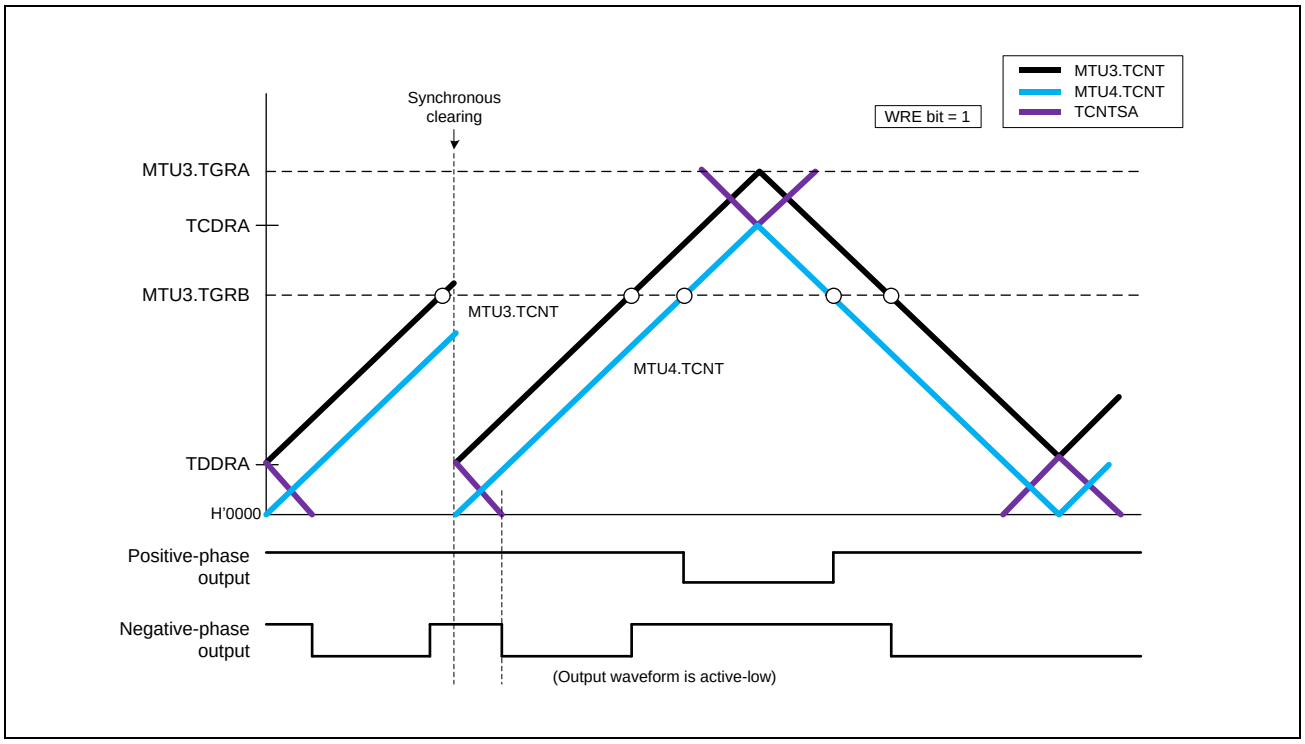



Figure 16.68 Example of Synchronous Clearing in Dead Time during Up-Counting
(Timing (3) in **Figure 16.66**; TWCRA.WRE Bit is 1)

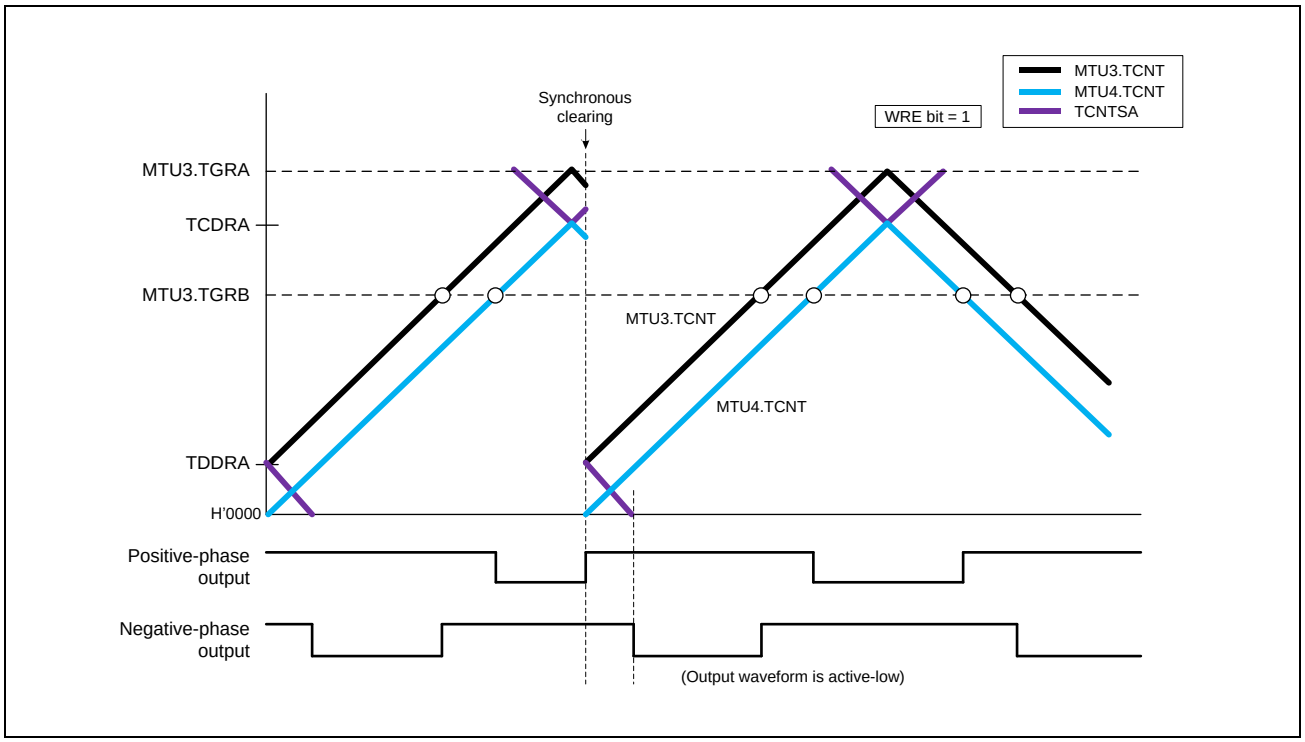



Figure 16.69 Example of Synchronous Clearing in Tb1 interval
(Timing (6) in **Figure 16.66**; TWCRA.WRE Bit is 1)

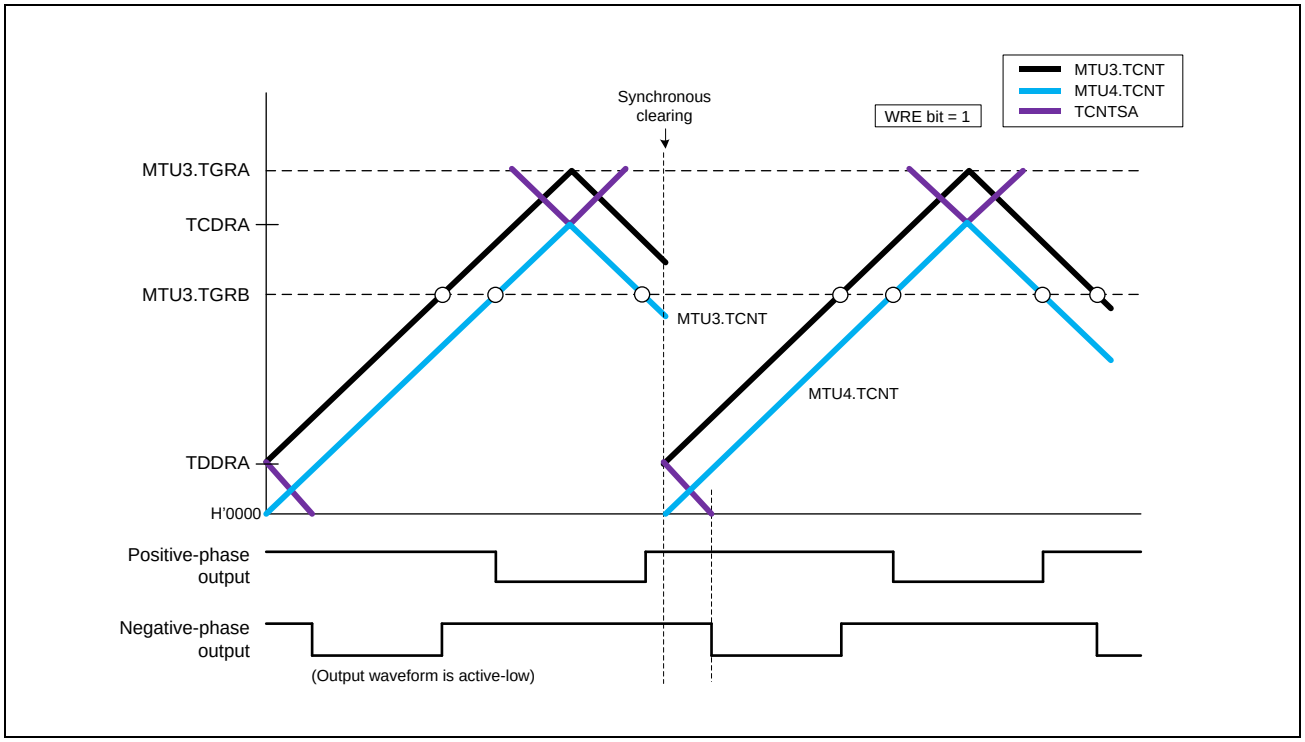



Figure 16.70 Example of Synchronous Clearing in Dead Time during Down-Counting
(Timing (8) in **Figure 16.66**; TWCRA.WRE Bit is 1)

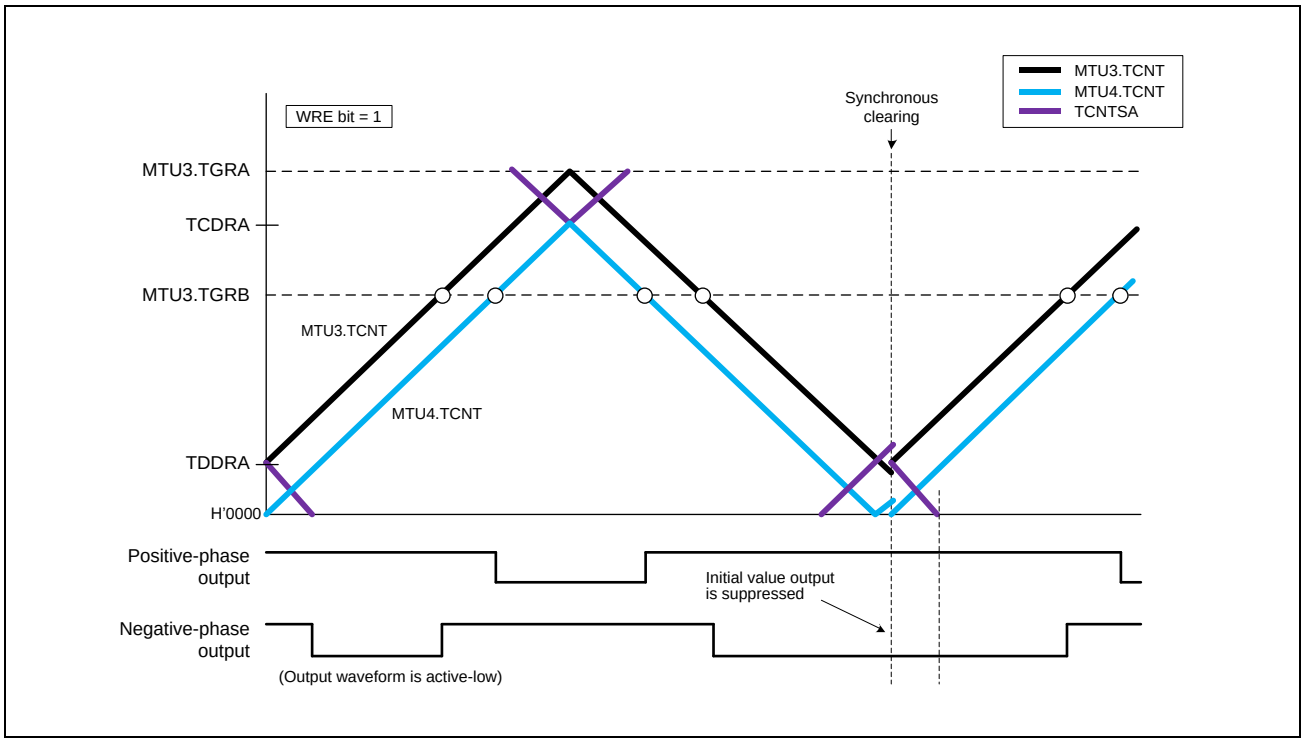



Figure 16.71 Example of Synchronous Clearing in Tb2 interval
(Timing (11) in **Figure 16.66**; TWCRA.WRE Bit is 1)

### (o) Suppressing Synchronous Counter Clearing for MTU0 to MTU2 and MTU6 and MTU7

In MTU6 and MTU7, setting the SCC bit in TWCRB to 1 suppresses synchronous counter clearing caused by MTU0 to MTU2.

Synchronous counter clearing is suppressed only within the interval shown in **Figure 16.72**. When using this function, MTU6 and MTU7 should be set to complementary PWM mode.

For details of synchronous clearing caused by MTU0 to MTU2, refer to **Section 16.3.10(2), Synchronous Counter Clearing for MTU6 and MTU7**.

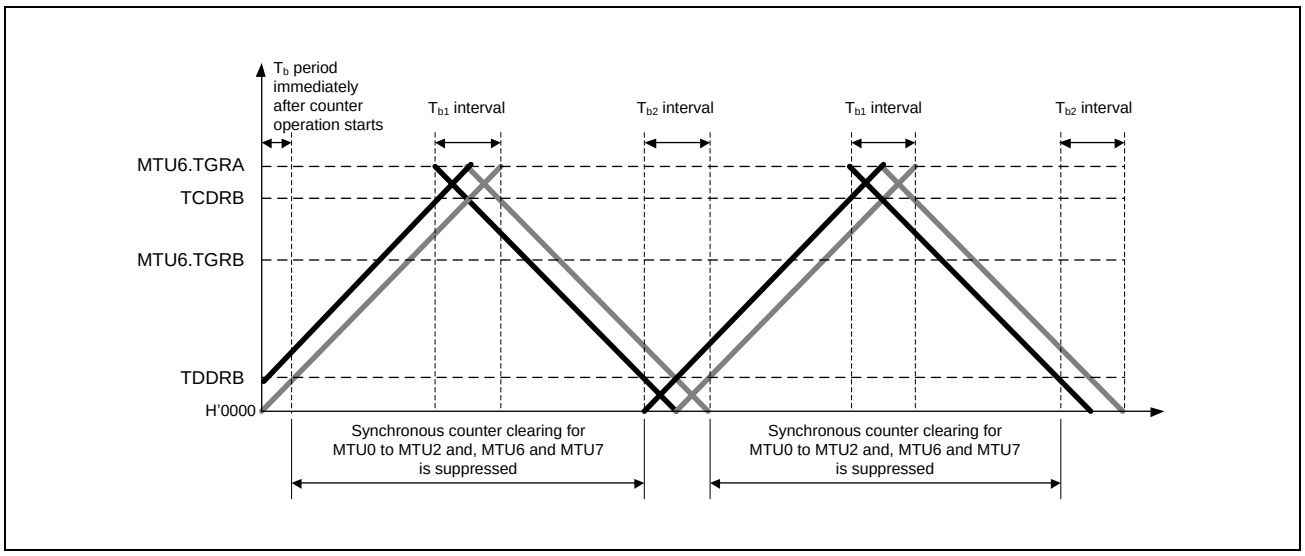
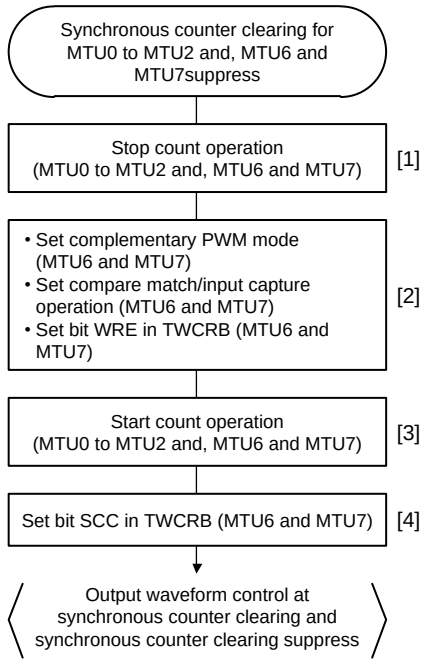



Figure 16.72 Synchronous Clearing-Suppressed Interval Specified by TWCRB.SCC Bit for MTU0, MTU1 and MTU2, and MTU6 and MTU7

- Example of Procedure for Suppressing Synchronous Counter Clearing for MTU0 to MTU2, and MTU6 and MTU7

An example of the procedure for suppressing synchronous counter clearing for MTU0 to MTU2, and MTU6 and MTU7 is shown in **Figure 16.73**.




[1] Clear bits CSTn of the timer start register (TSTRB) in MTU6 and MTU7 to 0 to stop count operation. Clear bits CSTn of TSTRA in MTU0 to MTU2 to 0 to stop count operation.

[2] Set the complementary PWM mode in MTU6 and MTU7 and compare match/input capture operation in MTU0 to MTU2. When bit WRE in TWCRA or TWCRB should be set, make appropriate setting here.

[3] Set bits CST6 and CST7 in MTU6 and MTU7 to 1 to start count operation. For synchronous counter clearing for MTU0 to MTU2 and MTU6 and MTU7, set bits CSTn of TSTRA in MTU0 to MTU2 to 1 to start count operation in any one of MTU0.TCNT to MTU2.TCNT.

[4] Read TWCRB and then set bit SCC in TWCRB to 1 to suppress synchronous counter clearing for MTU0 to MTU2 and MTU6 and MTU7*1. Here, do not modify the CCE and WRE bit values in TWCRB of MTU6 and MTU7. Synchronous counter clearing for MTU0 to MTU2 and MTU6 and MTU7 is suppressed in the synchronous counter clearing-suppressed periods.

Note 1. The TWCRB.SCC bit value can be modified during count operation. However, if a synchronous clearing occurs when bit TWCRB.SCC is modified from 0 to 1, the synchronous clearing may not be suppressed. If a synchronous clearing occurs when bit TWCRB.SCC is modified from 1 to 0, the synchronous clearing may be suppressed.

Figure 16.73 Example of Procedure for Suppressing Synchronous Counter Clearing for MTU0 to MTU2, and MTU6 and MTU7

- Examples of Suppression of Synchronous Counter Clearing for MTU 0 to MTU2, and MTU6 and MTU7

**Figure 16.74** to **Figure 16.77** show examples of operation in which MTU6 and MTU7 operate in complementary PWM mode and synchronous counter clearing for MTU 0 to MTU2, and MTU6 and MTU7 is suppressed by setting the SCC bit in TWCRB in MTU6 and MTU7 to 1. In the examples shown in **Figure 16.74** to **Figure 16.77**, synchronous counter clearing occurs at timing (3), (6), (8), and (11) shown in **Figure 16.66**, respectively. In these examples, the WRE bit in TWCRB in MTU6 and MTU7 is set to 1.

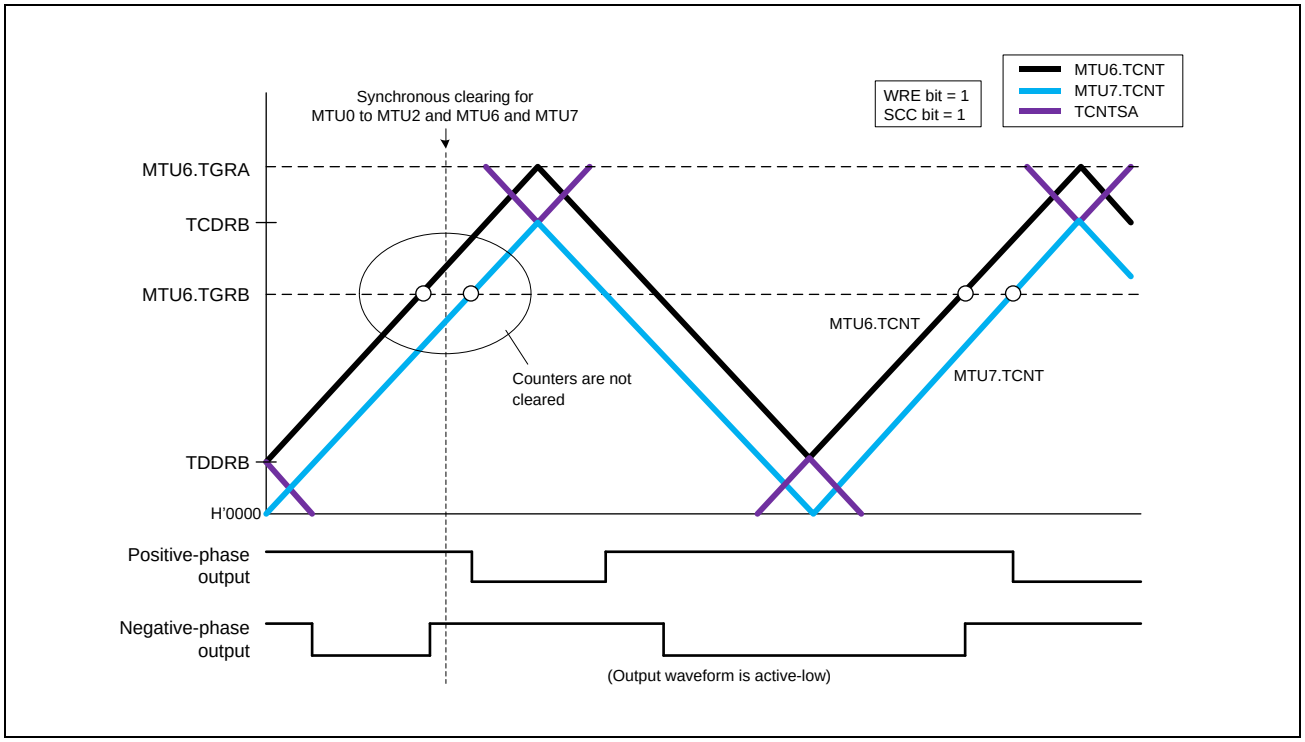



Figure 16.74 Example of Synchronous Clearing in Dead Time during Up-Counting
(Timing (3) in **Figure 16.66**; TWCRB.WRE Bit is 1 and SCC Bit is 1 in MTU6 and MTU7)

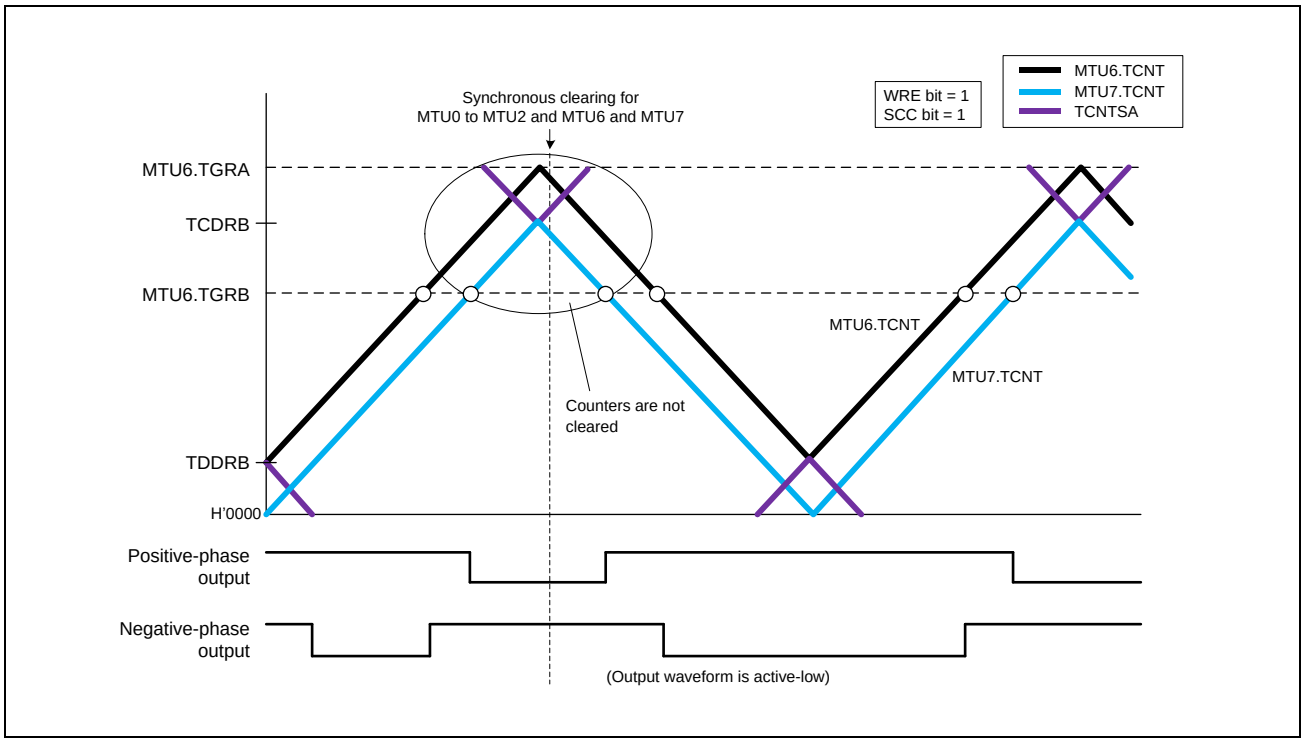



Figure 16.75 Example of Synchronous Clearing in Tb1 interval
(Timing (6) in **Figure 16.66**; TWCRB.WRE Bit is 1 and SCC Bit is 1 in MTU6 and MTU7)

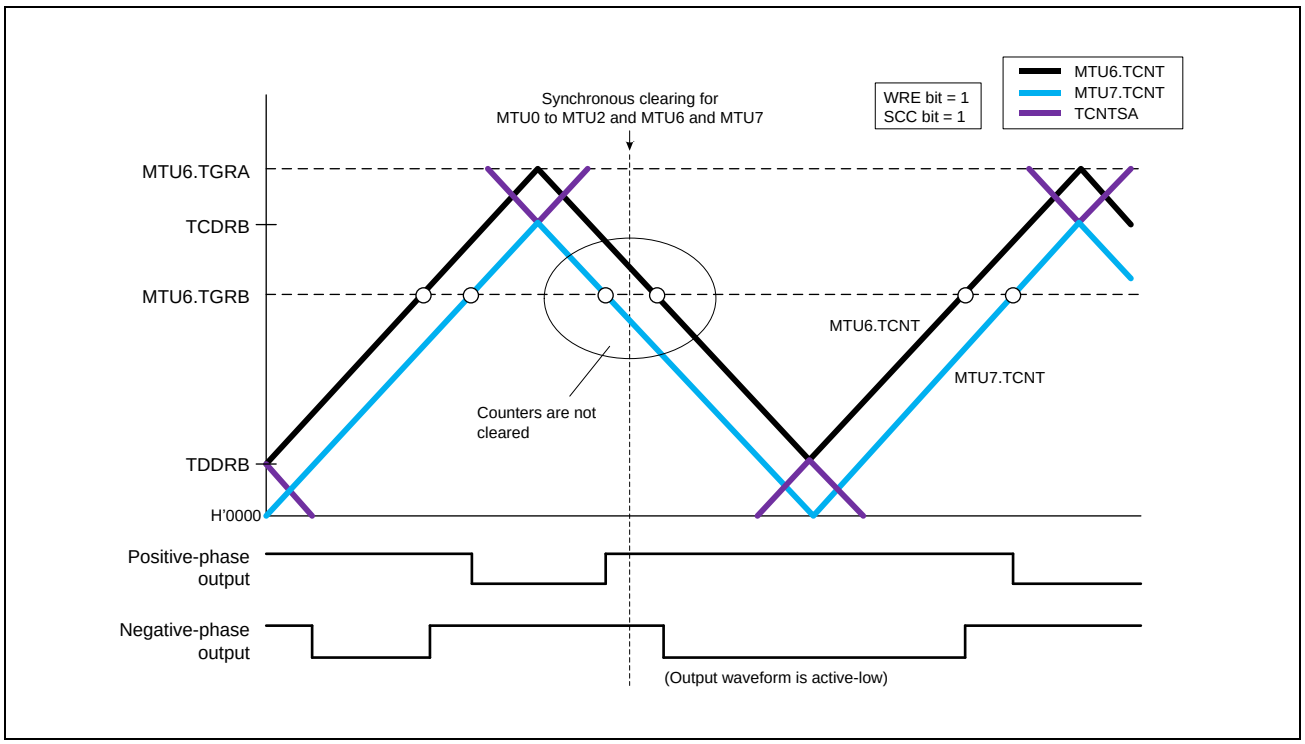



Figure 16.76 Example of Synchronous Clearing in Dead Time during Down-Counting
(Timing (8) in **Figure 16.66**; TWCRB.WRE Bit is 1 and SCC Bit is 1 in MTU6 and MTU7)

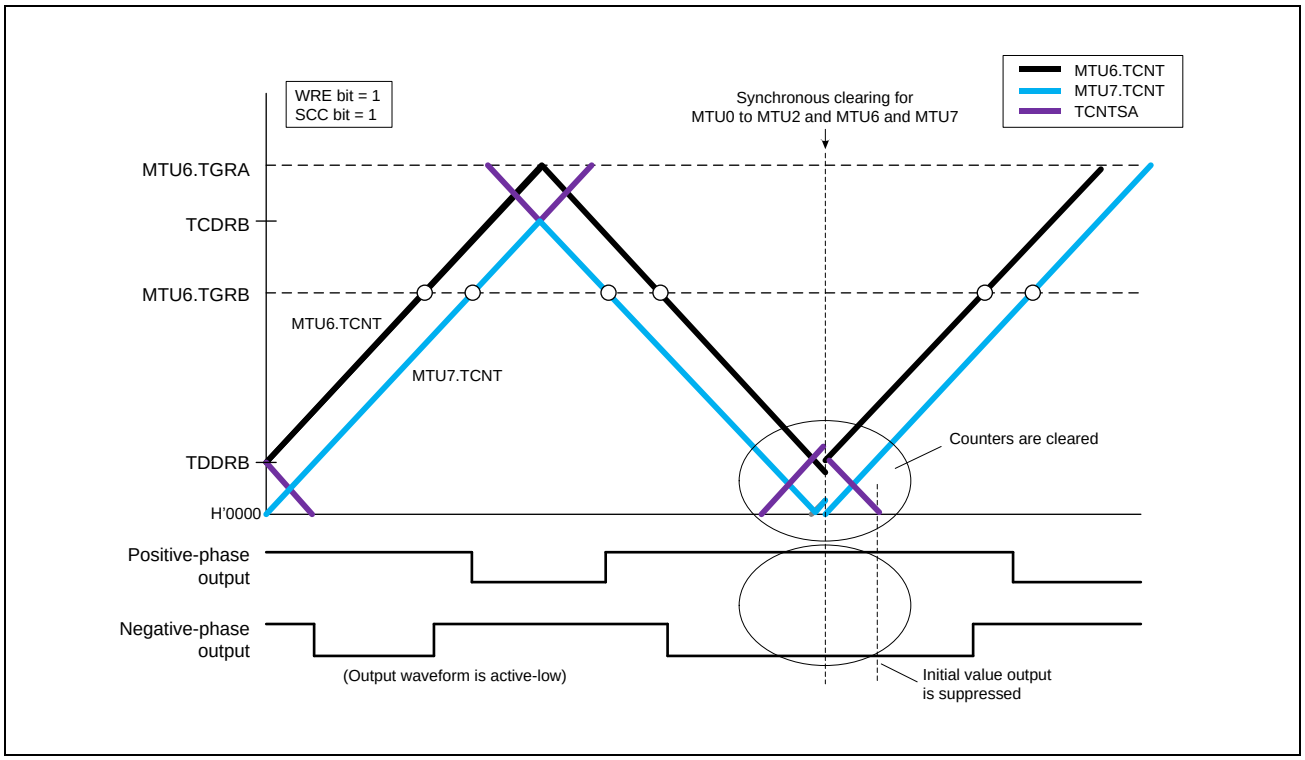



Figure 16.77 Example of Synchronous Clearing in Tb2 interval
(Timing (11) in **Figure 16.66**; TWCRB.WRE Bit is 1 and SCC Bit is 1 in MTU6 and MTU7)

### (p) Counter Clearing by MTU3.TGRA (MTU6.TGRA) Compare Match

In complementary PWM mode, MTU3.TCNT, MTU4.TCNT, and TCNTSA (MTU6.TCNT, MTU7.TCNT, and TCNTSB) can be cleared by MTU3.TGRA (MTU6.TGRA) compare match when the TWCRA.CCE (TWCRB.CCE) bit.

**Figure 16.78** 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 1 in the SYNC0 to SYNC4 or SYNC6 to SYNC7 bits in the timer synchronous register (TSYRA or TSYRB) and the CEOA to CEOD or CEIA to CEID bits in TSYCR).
3. Do not set the PWM duty value to H'0000.
4. Do not set the PSYE bit in timer output control register 1 (TOCR1A or TOCR1B) to 1.

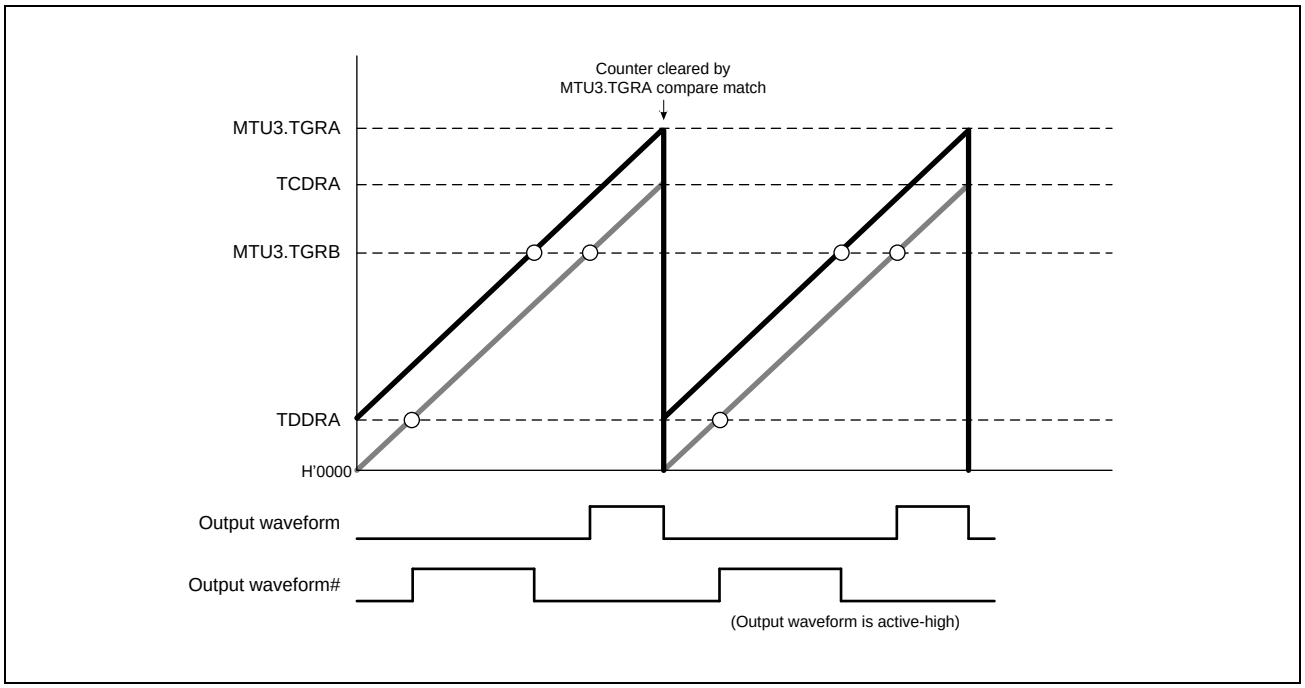



Figure 16.78 Example of Counter Clearing Operation by MTU3.TGRA Compare Match

### (q) Example of Waveform Output for Driving AC Synchronous Motor (Brushless DC Motor)

In complementary PWM mode when MTU3 and MTU4 are used, a brushless DC motor can easily be controlled using TGCRA. **Figure 16.79** to **Figure 16.82** show examples of brushless DC motor driving waveforms created using TGCRA.

To switch the output phases for a 3-phase brushless DC motor by means of external signals detected with a Hall element, etc., clear the TGCRA.FB bit to 0. In this case, the external signals indicating the magnetic pole position should be input to pins MTIOC0A, MTIOC0B, and MTIOC0C in MTU0 (make appropriate settings with the port mode registers (PMn) and Pmn port function control register (PFCm) of the GPIO ports). When an edge is detected at pin MTIOC0A, MTIOC0B, or MTIOC0C, the output on/off state is switched automatically.

When the TGCRA.FB bit is 1, the output on/off state is switched when the UF, VF, or WF bit in TGCRA is cleared to 0 or set to 1.

The driving waveforms are output from the 6-phase PWM output pins for complementary PWM mode.

With this 6-phase output, while the output is turned on, chopping output is available through complementary PWM mode output function by setting the N bit or P bit in TGCRA to 1. When the N bit or P bit is 0, the level output is selected.

The active level of the 6-phase output (on output level) can be set with the TOCR1A.OLSN and TOCR1A.OLSP bits regardless of the setting of the N and P bits.

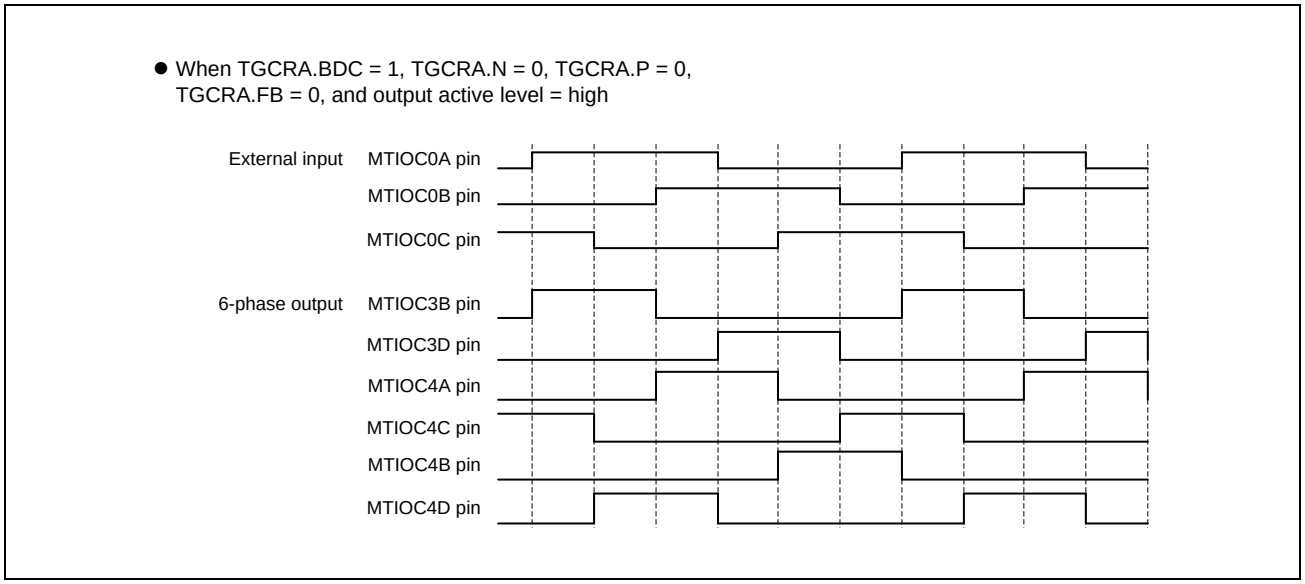



Figure 16.79 Example of Output Phase Switching by External Input (1)

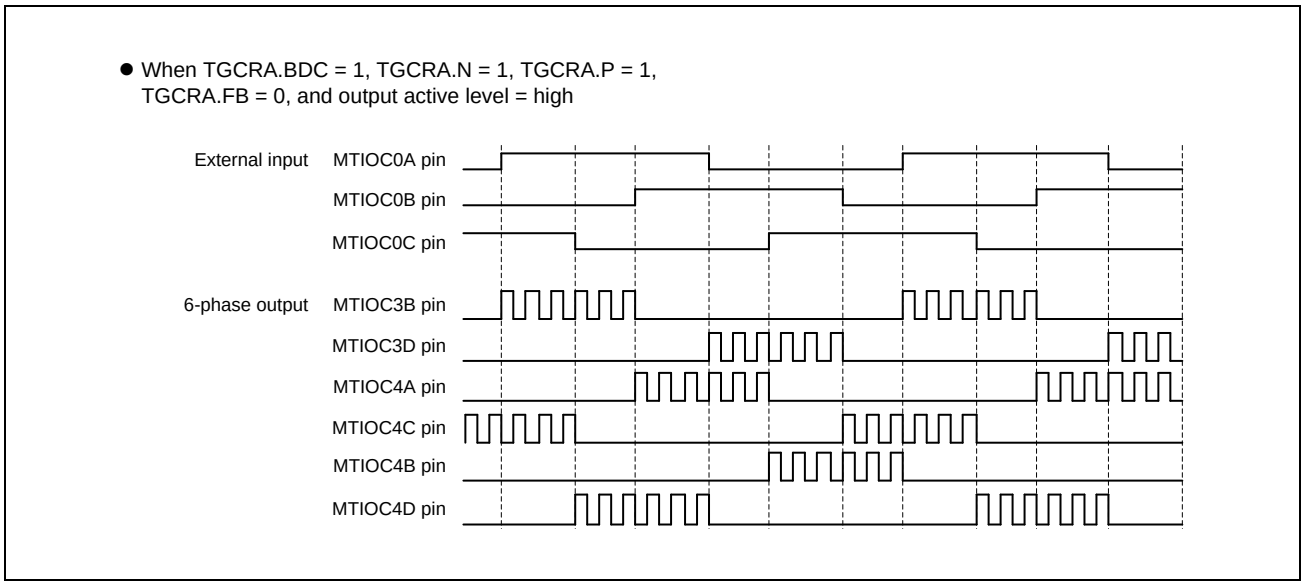



Figure 16.80 Example of Output Phase Switching by External Input (2)

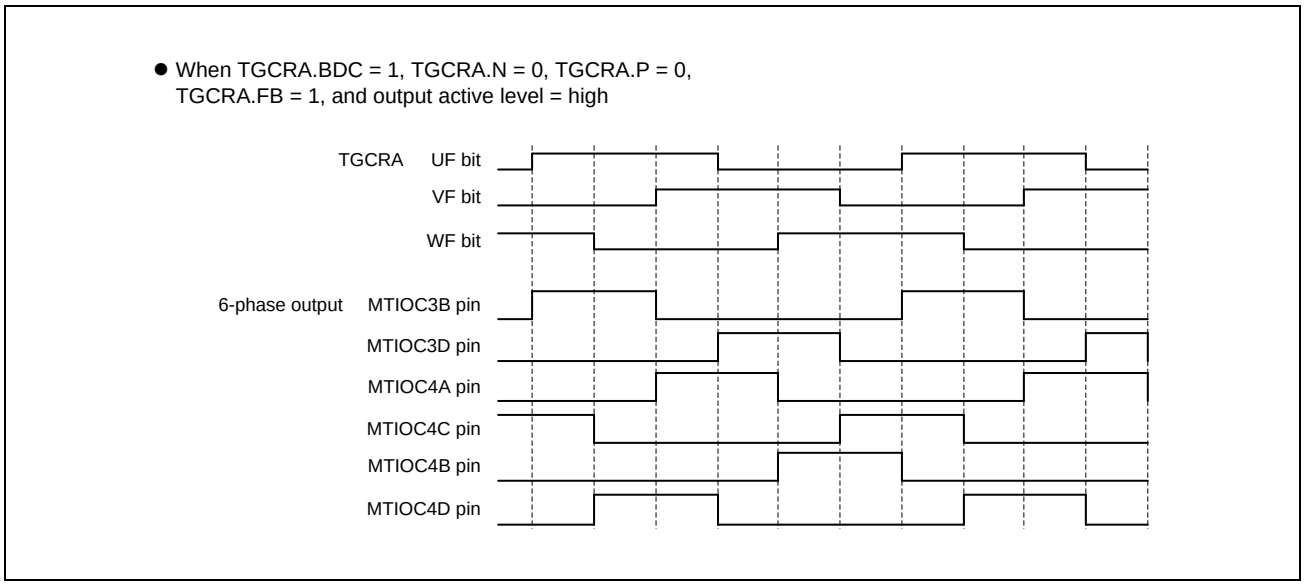



Figure 16.81 Example of Output Phase Switching through UF, VF, and WF Bit Settings (1)

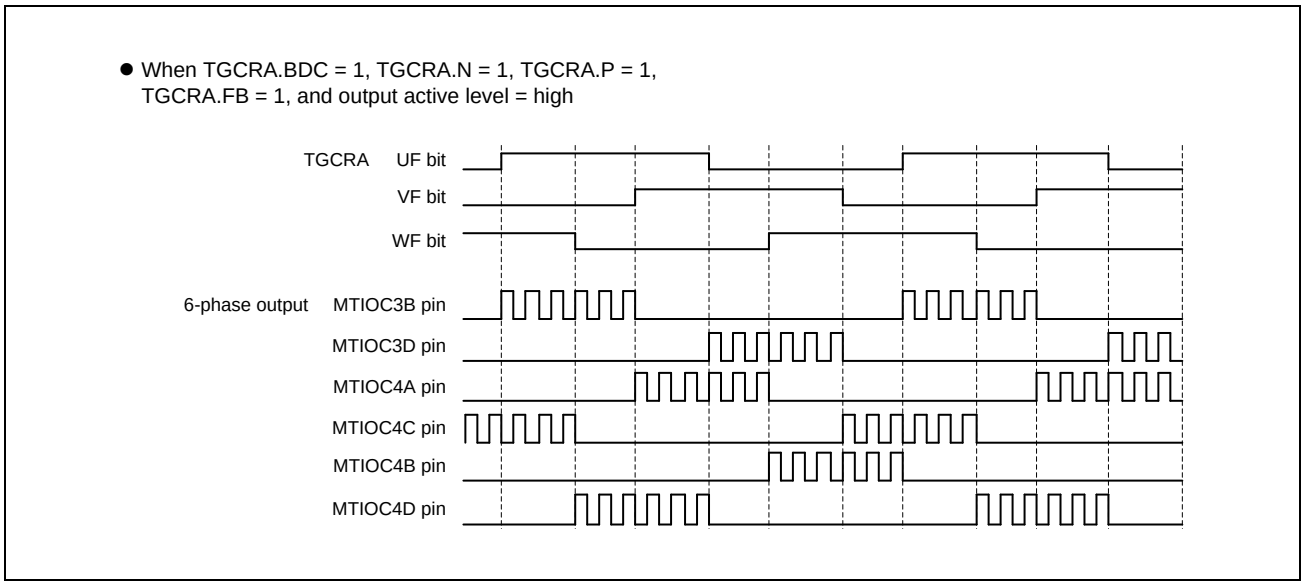



Figure 16.82 Example of Output Phase Switching through UF, VF, and WF Bit Settings (2)

### (r) A/D Converter Start Request Setting

In complementary PWM mode, an A/D converter start request can be issued using MTU3.TGRA (MTU6.TGRA) compare match, MTU4.TCNT (MTU7.TCNT) underflow (trough), or compare match on a channel other than MTU3 and MTU4 (MTU6 and MTU7).

When start requests using MTU3.TGRA (MTU6.TGRA) compare match are specified, A/D conversion can be started at the crest of the MTU3.TCNT (MTU6.TCNT) count.

A/D converter start requests can be specified by setting the TIER.TTGE bit. To issue an A/D converter start request at an MTU4.TCNT (MTU7.TCNT) underflow (trough), set the MTU4.TIER.TTGE2 (MTU7.TIER.TTGE2) bit to 1.

### (s) Double Buffer Function in Complementary PWM Mode

In complementary PWM mode 3 (transfer at the crest and trough), the PWM output setting resolution can be improved from ±2 to ±1 by setting the TMDR2A.DRS (TMDR2B.DRS) bit to 1.

When setting buffer registers A (TGRD in MTU3, TGRC and TGRD in MTU4, TGRD in MTU6, and TGRC and TGRD in MTU7), set also buffer registers B (TGRE in MTU3, TGRE and TGRF in MTU4, TGRE in MTU6, and TGRE and TGRF in MTU7) at the same time. Each buffer register B should be set to the buffer register A value or (buffer register A value – 1). For details of the setting procedure, refer to **Section 16.3.8(1), Example of Complementary PWM Mode Setting Procedure**.

NOTE

When a buffer register B is set to the buffer register A value, symmetric PWM waveforms are output. When a buffer register B is set to (buffer register A value – 1), asymmetric PWM waveforms are output.

**Figure 16.83** shows an example of double buffer operation. Each register data is transferred as follows.

- After TGRD (buffer A) in MTU4 or MTU7 is written to, data is transferred from TGRD (buffer A) in MTU4 or MTU7 to Temp3A or Temp6A (temporary A) and from TGRF (buffer B) in MTU4 or MTU7 to Temp3B or Temp6B (temporary B).
- With timing (1) in the figure, data is transferred from Temp3A or Temp6A (temporary A) to TGRB (compare) in MTU4 or MTU7.
- With timing (2) in the figure, data is transferred from Temp3B or Temp6B (temporary B) to TGRB (compare) in MTU4 or MTU7.

In the crest interval (Tb1 interval), the compare register and temporary register A are valid; in the trough interval (Tb2 interval), the compare register and temporary register B are valid.

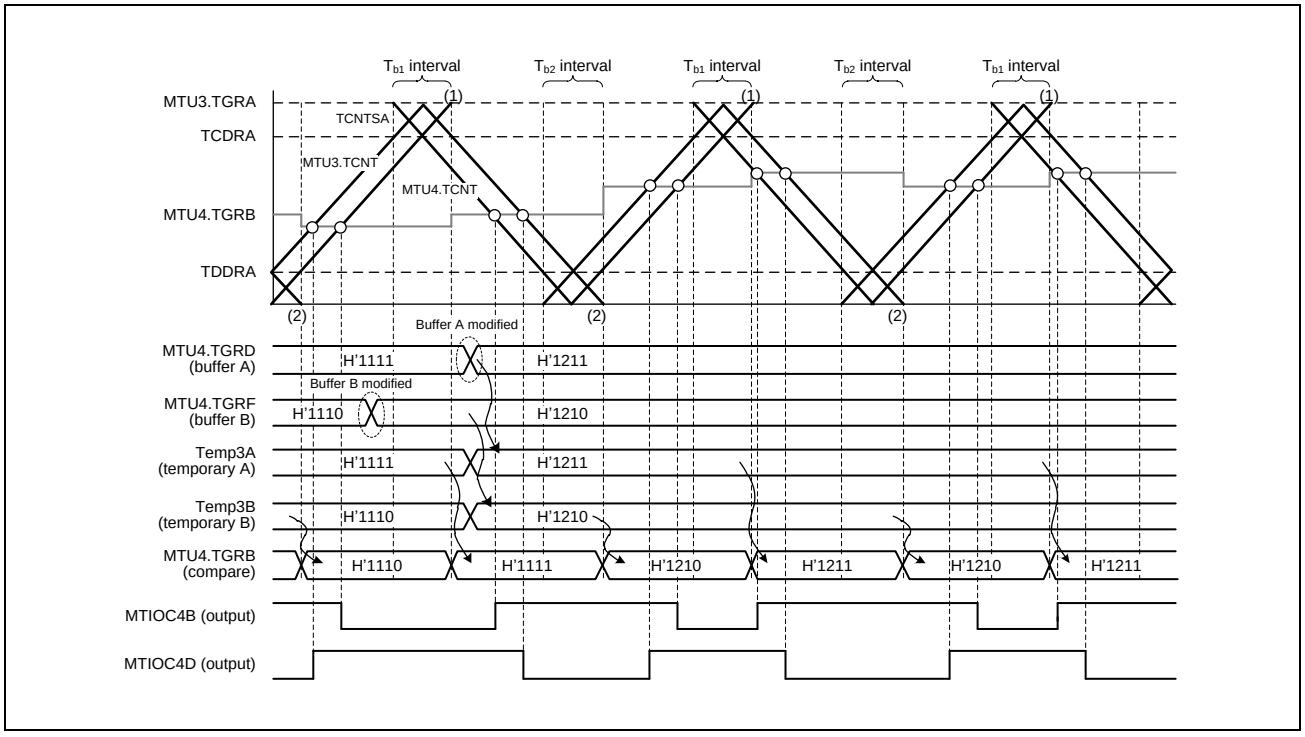



Figure 16.83 Example of Double Buffer Operation

**Figure 16.84** shows an example when the buffer write value is smaller than the TDDRA (TDDRB) value, and **Figure 16.85** shows an example when the write value is greater than TCDRA (TCDRB).

In the crest interval, the output is controlled according to the compare match with the compare register or temporary register A; in the trough interval, the output is controlled according to the compare match with the compare register or temporary register B.

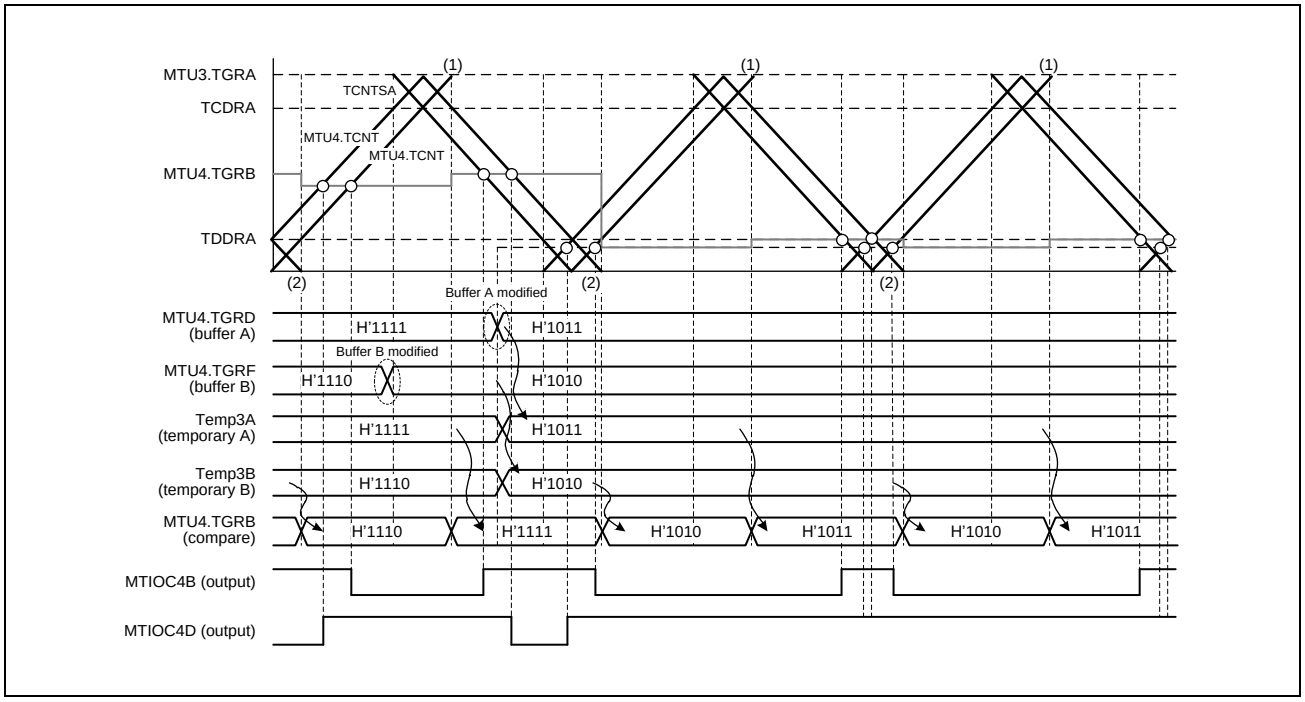



Figure 16.84 Example of Double Buffer Operation (Buffer Write Value is Smaller than TDDRA)

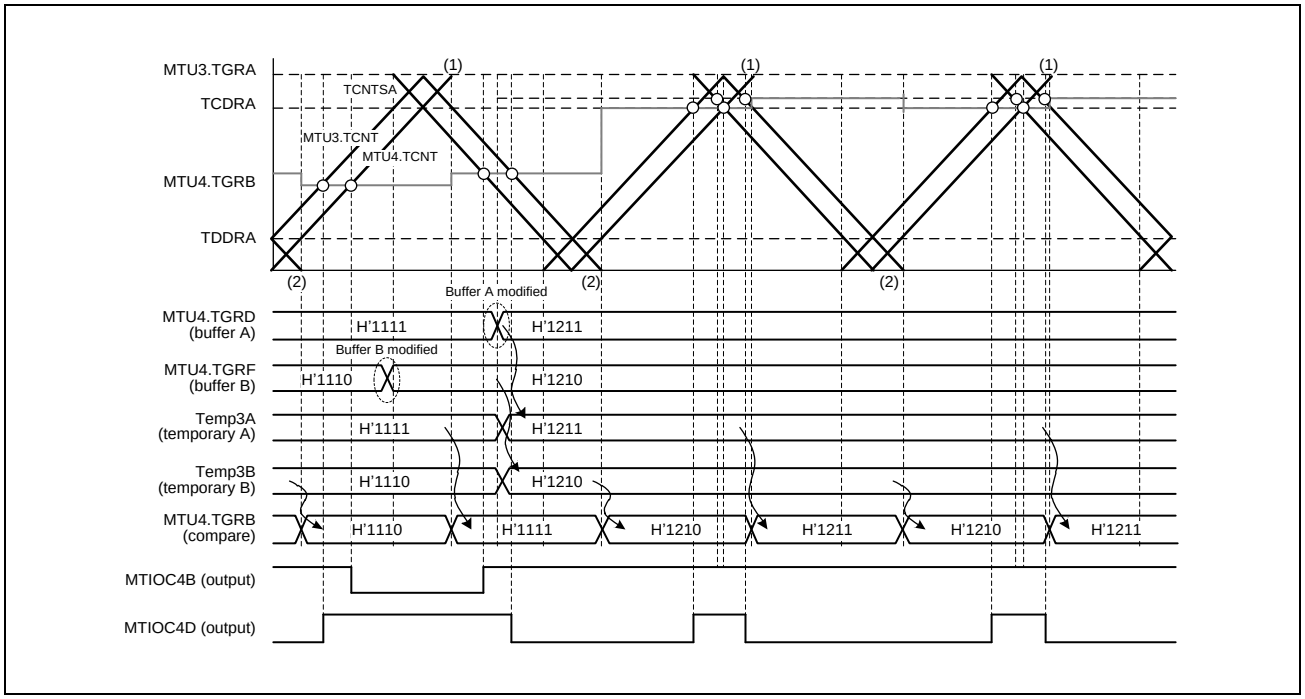



Figure 16.85 Example of Double Buffer Operation (Buffer Write Value is Greater than TCDRA)

### (3) Interrupt Skipping Function 1 in Complementary PWM Mode

Interrupts TGIA3 (TGIA6) (at the crest) and TCIV4 (TCIV7) (at the trough) in MTU3 and MTU4 (MTU6 and MTU7) can be skipped up to seven times by making settings in the TITCR1A (TITCR1B) register.

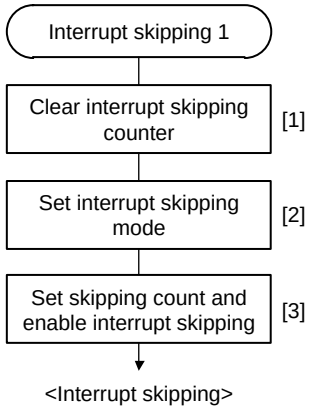
Transfers from a buffer register to a temporary register or a compare register can be skipped in coordination with interrupt skipping by making settings in the TBTERA (TBTERB) register. For the linkage with buffer registers, refer to description (c), Buffer Transfer Control Linked with Interrupt Skipping, 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 MTU4.TADCR (MTU7.TADCR) register. For the linkage with the A/D converter start request delaying function, refer to **Section 16.3.9, A/D Converter Start Request Delaying Function**.

The TITCR1A (TITCR1B) register should be set while interrupt skipping function 1 is selected by setting the TITM bit in the timer interrupt skipping mode register (TITMRA or TITMRB) to 0, TGIA3 (TGIA6) interrupt requests are disabled by setting the MTU3.TIER (MTU6.TIER) register, TCIV4 (TCIV7) interrupt requests are disabled by setting the MTU4.TIER (MTU7.TIER) register, and a compare match is not generated. Before changing the skipping count, be sure to clear the T3AEN (T6AEN) and T4VEN (T7VEN) bits to 0 to clear the skipping counter.

#### (a) Example of Interrupt Skipping Function 1 Setting Procedure

**Figure 16.86** shows an example of the interrupt skipping function 1 setting procedure. **Figure 16.87** shows the periods during which interrupt skipping count can be changed.




[1] Set bits T3AEN and T4VEN in the timer interrupt skipping set register 1 (TITCR1A) to 0 to clear the skipping counter.

[2] Set bit TITM in the timer interrupt skipping mode register (TITMRA) to 0 to select interrupt skipping function 1.

[3] Specify the interrupt skipping count within the range from 0 to 7 times in bits T3ACOR2 to T3ACOR0 and T4VCOR2 to T4VCOR0 in TITCR1A, and enable interrupt skipping through bits T3AEN and T4VEN.

**Note:** The setting of TITCR1A must be done while the TGIA3 and TCIV4 interrupt requests are disabled by the settings of MTU3.TIER and MTU4.TIER under the conditions in which a compare match is not generated. Before changing the skipping count, be sure to clear the T3AEN and T4VEN bits to 0 to clear the skipping counter.

Figure 16.86 Example of Interrupt Skipping Function 1 Setting Procedure

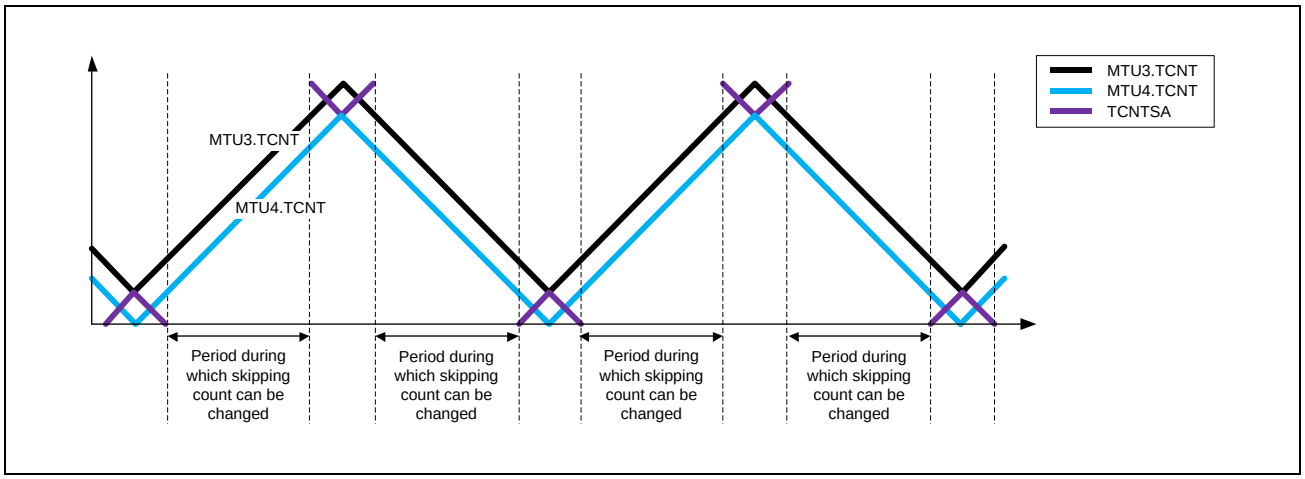



Figure 16.87 Periods during which Interrupt Skipping Count can be Changed

### (b) Example of Interrupt Skipping Function 1

**Figure 16.88** shows an example of TGIA3 (TGIA6) interrupt skipping in which the interrupt skipping count is set to three by the T3ACOR (T6ACOR) bits and the T3AEN (T6AEN) bit is set to 1 in the TITCR1A (TITCR1B) register.

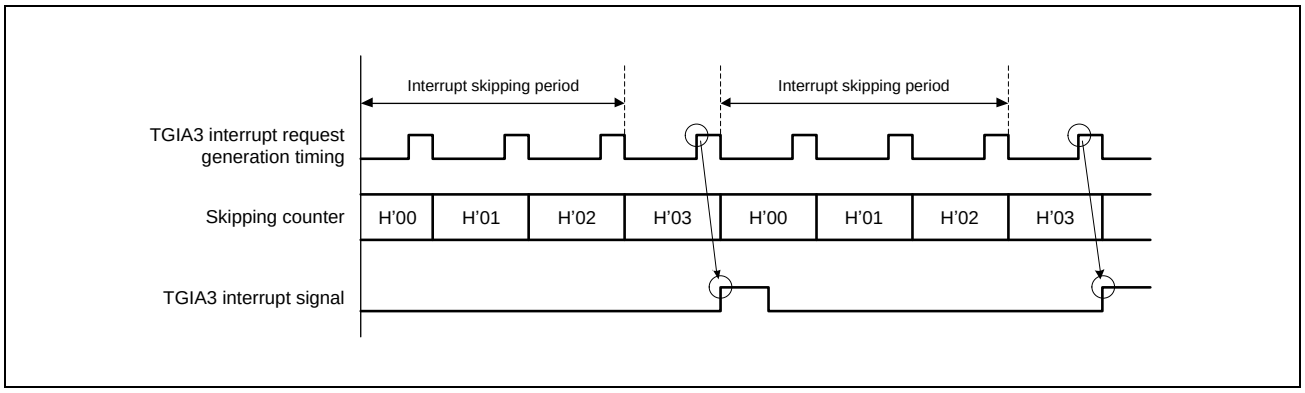



Figure 16.88 Example of Interrupt Skipping Function 1

### (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 BTE[1:0] bits in the TBTERA (TBTERB) register.

**Figure 16.89** shows an example of operation when buffer transfer is disabled (BTE[1:0] = 01b). While this setting is valid, data is not transferred from the buffer register to the temporary register.

**Figure 16.90** shows an example of operation when buffer transfer is linked with interrupt skipping (BTE[1:0] = 10b). While this setting is valid, data is not transferred from the buffer register outside the buffer transfer-enabled period.

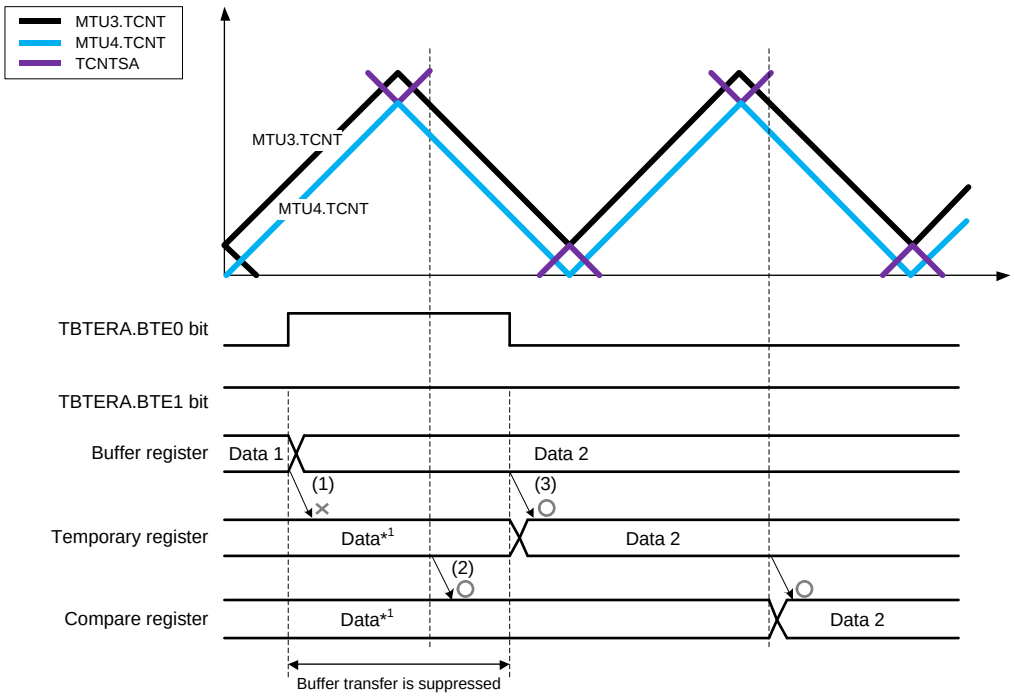
Note that the buffer transfer-enabled period differs depending on whether only the T3AEN (T6AEN) bit in the TITCR1A (TITCR1B) register is set to 1, only the T4VEN (T7VEN) bit in the TITCR1A (TITCR1B) register is set to 1, or both the T3AEN and T4VEN (T6AEN and T7VEN) bits are set to 1. **Figure 16.91** shows the relationship between the T3AEN (T6AEN) and T4VEN (T7VEN) bit settings in TITCR1A (TITCR1B) and buffer transfer-enabled period.

NOTE

This function must be used in combination with interrupt skipping function 1.

When interrupt skipping is disabled (the T3AEN and T4VEN (T6AEN and T7VEN) bits in the timer interrupt skipping set register 1 (TITCR1A or TITCR1B) are cleared to 0 or the skipping count set bits (T3ACOR and T4VCOR (T6ACOR and T7VCOR)) in TITCR1A (TITCR1B) are cleared to 0), make sure that buffer transfer is not linked with interrupt skipping (clear the BTE1 bit in TBTERA or TBTERB to 0).

If buffer transfer is linked with interrupt skipping while interrupt skipping is disabled, buffer transfer is never performed.



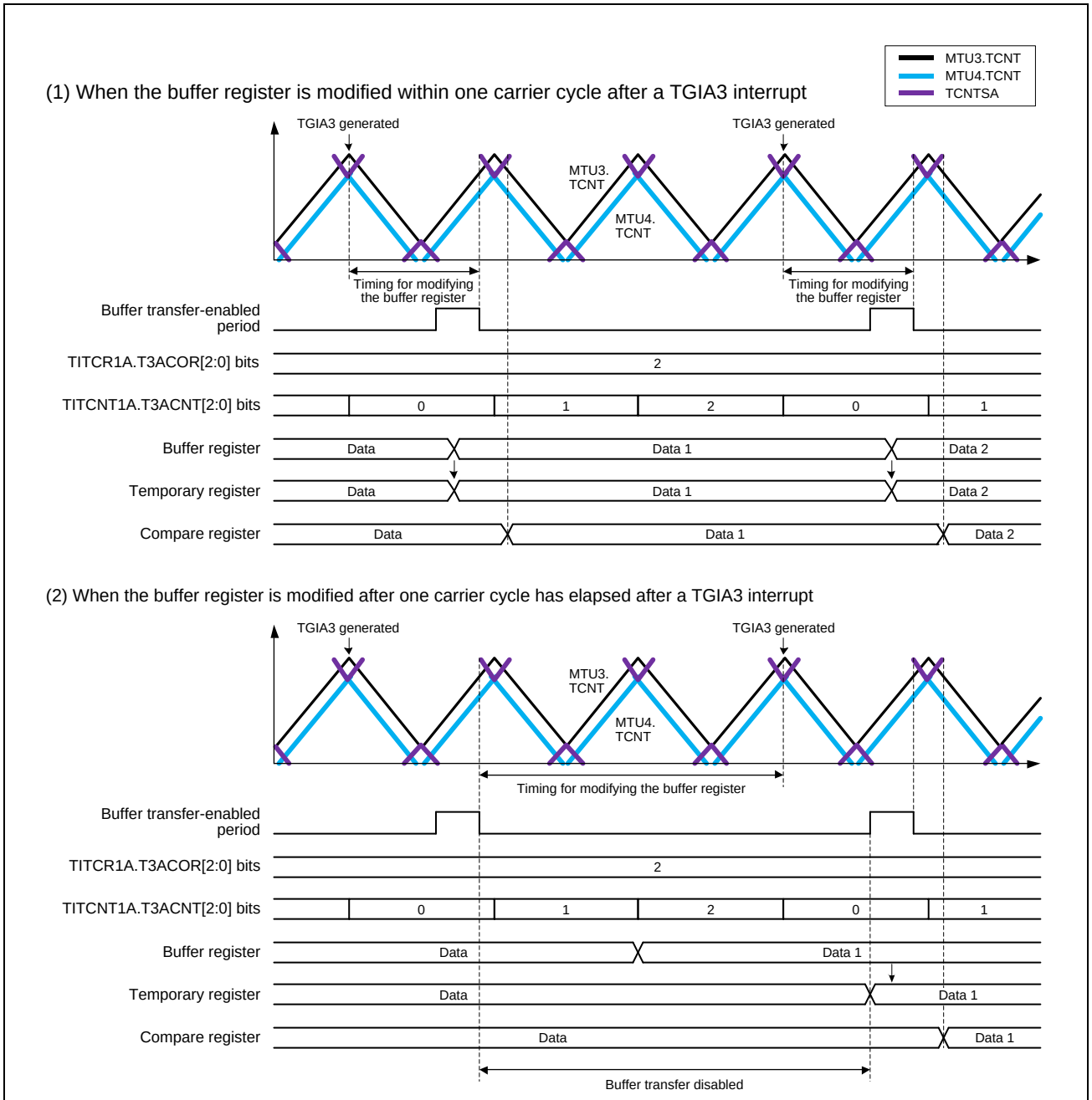

(1) No data is transferred from the buffer register to the temporary register in the buffer transfer-disabled period (bits BTE[1:0] in TBTERA are set to 01b).

(2) Data is transferred from the temporary register to the compare 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.

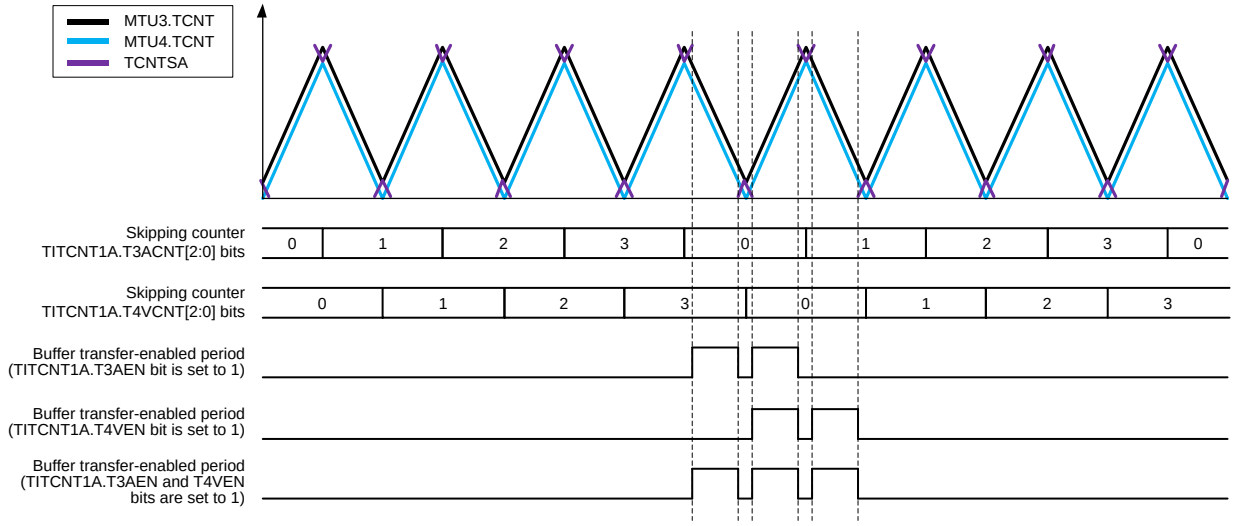
Note 1. When buffer transfer at the crest is selected.

Figure 16.89 Example of Operation When Buffer Transfer is Disabled (BTE[1:0] = 01b)




**Note:** Buffer transfer at the crest is selected.
The skipping count is set to two. The TITCR1A.T3AEN bit is set to 1.

Figure 16.90 Example of Operation When Buffer Transfer is Linked with Interrupt Skipping (BTE[1:0] = 10b)




**Note:** The skipping count is set to three.
Buffer operation is set to transfer at the crest and trough.

Figure 16.91 Relationship between Bits T3AEN and T4VEN in TITCR1A and Buffer Transfer-Enabled Period

### (4) Complementary PWM Mode Output Protection Functions

The following output protection functions are provided for complementary PWM mode.

#### (a) Register and Counter Miswrite Prevention Function

Access from the CPU to the mode registers, control registers, compare registers, and counters can be enabled or disabled by setting the RWE bit in the TRWERA (TRWERB) register. The applicable registers are some of the registers in MTU3, MTU4, MTU6, and MTU7 shown below:

MTU3.TCR, MTU4.TCR, MTU3.TCR2, MTU4.TCR2, MTU3.TMDR1, MTU4.TMDR1, MTU3.TIORH, MTU4.TIORH, MTU3.TIORL, MTU4.TIORL, MTU3.TIER, MTU4.TIER, MTU3.TCNT, MTU4.TCNT, MTU3.TGRA, MTU4.TGRA, MTU3.TGRB, MTU4.TGRB, MTU.TOERA, MTU.TOCR1A, MTU.TOCR2A, MTU.TGCRA, MTU.TCDRA, MTU.TDDRA MTU6.TCR, MTU7.TCR, MTU6.TCR2, MTU7.TCR2, MTU6.TMDR1, MTU7.TMDR1, MTU6.TIORH, MTU7.TIORH, MTU6.TIORL, MTU7.TIORL, MTU6.TIER, MTU7.TIER, MTU6.TCNT, MTU7.TCNT, MTU6.TGRA, MTU7.TGRA, MTU6.TGRB, MTU7.TGRB, MTU.TOERB, MTU.TOCR1B, MTU.TOCR2B, MTU.TCDRB, and MTU.TDDRB

47 registers in total

This function can disable CPU access to the mode registers, control registers, and counters to prevent miswriting due to CPU runaway. In the access-disabled state, the applicable registers are read as undefined and writing to these registers is ignored.

#### (b) Halting of PWM Output by External Signal

The PWM output pins of MTU0, MTU3, MTU4, MTU6, and MTU7 can be set to the high-impedance state automatically.

See **Section 17, Port Output Enable 3 (POE3)**, for details.

## 16.3.9 A/D Converter Start Request Delaying Function

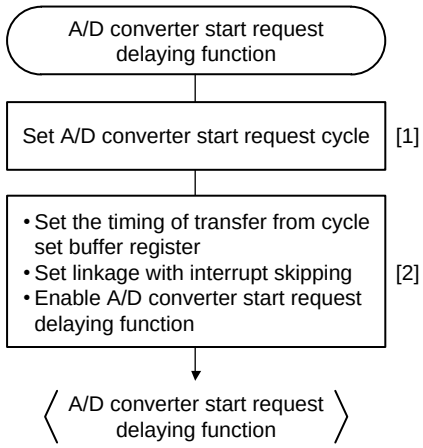
A/D converter start requests can be issued in MTU4 or MTU7 by making settings in the timer A/D converter start request control register (MTU4.TADCR or MTU7.TADCR), timer A/D converter start request cycle set registers (MTU4.TADCORA and MTU4.TADCORB, or MTU7.TADCORA and MTU7.TADCORB), and timer A/D converter start request cycle set buffer registers (MTU4.TADCOBRA and MTU4.TADCOBRB, or MTU7.TADCOBRA and MTU7.TADCOBRB).

The A/D converter start request delaying function compares MTU4.TCNT with MTU4.TADCORA or MTU4.TADCORB (MTU7.TCNT with MTU7.TADCORA or MTU7.TADCORB), and when their values match, the function issues a respective A/D converter start request (TRG4AN or TRG4BN (TRG7AN or TRG7BN)).

A/D converter start requests (TRG4AN and TRG4BN (TRG7AN and TRG7BN)) can be skipped in coordination with interrupt skipping by making settings in the ITA3AE, ITA4VE, ITB3AE, and ITB4VE bits in the TADCR register in MTU4 (the ITA6AE, ITA7VE, ITB6AE, and ITB7VE bits in the TADCR register in MTU7).

### (1) Example of Procedure for Specifying A/D Converter Start Request Delaying Function

**Figure 16.92** 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 (MTU4.TADCOBRA or MTU4.TADCOBRB) and timer A/D converter start request cycle register (MTU4.TADCORA or MTU4.TADCORB).
(When the timing of transfer from the cycle buffer register to cycle register is set to the trough, or both trough and crest in step [2] below, the same initial value must be specified in the cycle buffer register and cycle register because transfer is performed immediately after counting starts.)

[2] Use bits BF[1:0] in the timer A/D converter start request control register (TADCR in MTU4) to specify the timing of transfer from the timer A/D converter start request cycle buffer register to the A/D converter start request cycle register.
- Specify whether to link with interrupt skipping through bits ITA3AE, ITA4VE, ITB3AE, and ITB4VE.
- Use bits UT4AE, DT4AE, UT4BE, and DT4BE to enable A/D converter start requests (TRG4AN or TRG4BN).

**Note 1.** Set up TADCR in MTU4 while TCNT in MTU4 is stopped.

**Note 2.** Set bit BF1 of TADCR in MTU4 to 0 when complementary PWM mode is not selected.

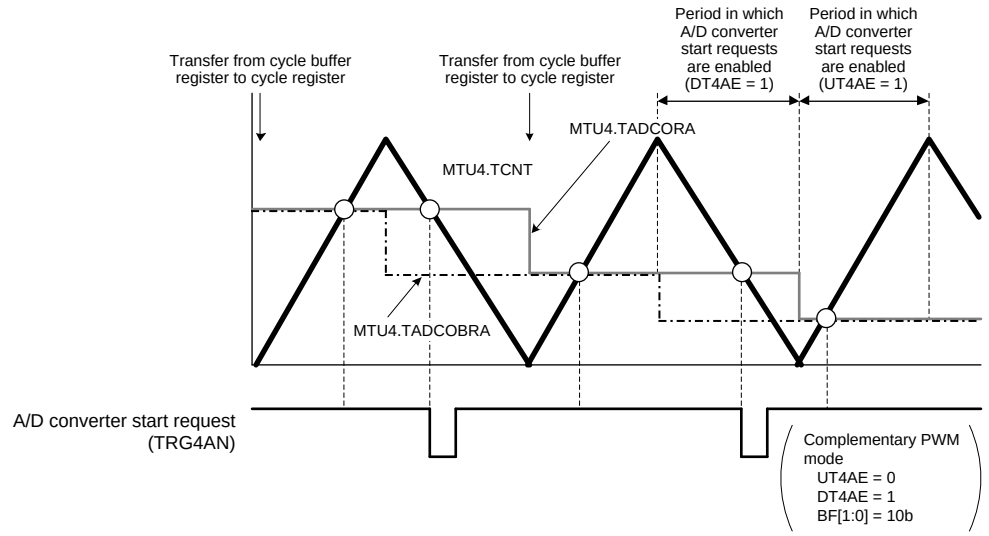
**Note 3.** Set bits ITA3AE, ITA4VE, ITB3AE, ITB4VE, DT4AE, and DT4BE to 0 when complementary PWM mode is not selected.

**Note 4.** Set bits ITA3AE, ITA4VE, ITB3AE, and ITB4VE to 0 when interrupt skipping is disabled.

Figure 16.92 Example of Procedure for Specifying A/D Converter Start Request Delaying Function (MTU3 and MTU4)

### (2) Basic Example of A/D Converter Start Request Delaying Function Operation

**Figure 16.93** shows a basic example of A/D converter start request signal (TRG4AN (TRG7AN)) operation when the trough of MTU4.TCNT (MTU7.TCNT) is specified for the buffer transfer timing and an A/D converter start request signal is output during MTU4.TCNT (MTU7.TCNT) down-counting.




Period in which A/D converter start requests are enabled (UT4AE = 1): 0 ≤ MTU4.TCNT ≤ TCDR – 1
Period in which A/D converter start requests are enabled (DT4AE = 1): TCDR ≥ MTU4.TCNT ≥ 1
UT4AE, DT4AE, BF[1:0]: Bits in TADCR

Figure 16.93 Basic Example of A/D Converter Start Request Signal (TRG4AN) Operation

### (3) Period in which A/D Converter Start Requests are Enabled

An A/D converter start request (TRG4AN or TRG4BN) can be issued when the TCNT counter value in MTU4 (MTU7) matches the value of the corresponding cycle set register (TADCORA or TADCORB) in MTU4 (MTU7) within the period specified by the UT4AE or UT4BE (UT7AE or UT7BE) bit of TADCR in MTU4 (MTU7).

When the UT4AE or UT4BE (UT7AE or UT7BE) bit is set to 1 in complementary PWM mode, issuing of AD converter start requests is enabled for the period of TCNT up-counting (0 ≤ TCNT ≤ TCDR - 1) in MTU4 (MTU7). When the DT4AE or DT4BE (DT7AE or DT7BE) bit is set to 1, issuing of AD converter start requests is enabled for the period of TCNT down-counting (TCDR ≥ TCNT ≥ 1) in MTU4 (MTU7) (**Figure 16.93**).

### (4) Buffer Transfer

The data in the timer A/D converter start request cycle set registers (MTU4.TADCORA and MTU4.TADCORB, or MTU7.TADCORA and MTU7.TADCORB) is updated by writing data to the timer A/D converter start request cycle set buffer registers (MTU4.TADCOBRA and MTU4.TADCOBRB, or MTU7.TADCOBRA and MTU7.TADCOBRB). Data is transferred from the buffer registers to the respective cycle set registers at the timing selected with the BF[1:0] bits in the MTU4.TADCR (MTU7.TADCR) register.

In complementary PWM mode, data is also transferred from the timer A/D converter start request cycle set buffer registers to the timer A/D converter start request cycle set registers when MTU4.TGRD (MTU7.TGRD) register is updated.

When using buffer transfer in complementary PWM mode, pay attention to the buffer transfer timing. For details, refer to **Section 16.6.26, Notes on A/D Converter Start Request Delaying Function in Complementary PWM Mode**. When complementary PWM mode is not selected, be sure to clear the BF1 bit of the TADCR register in MTU4 or MTU7.

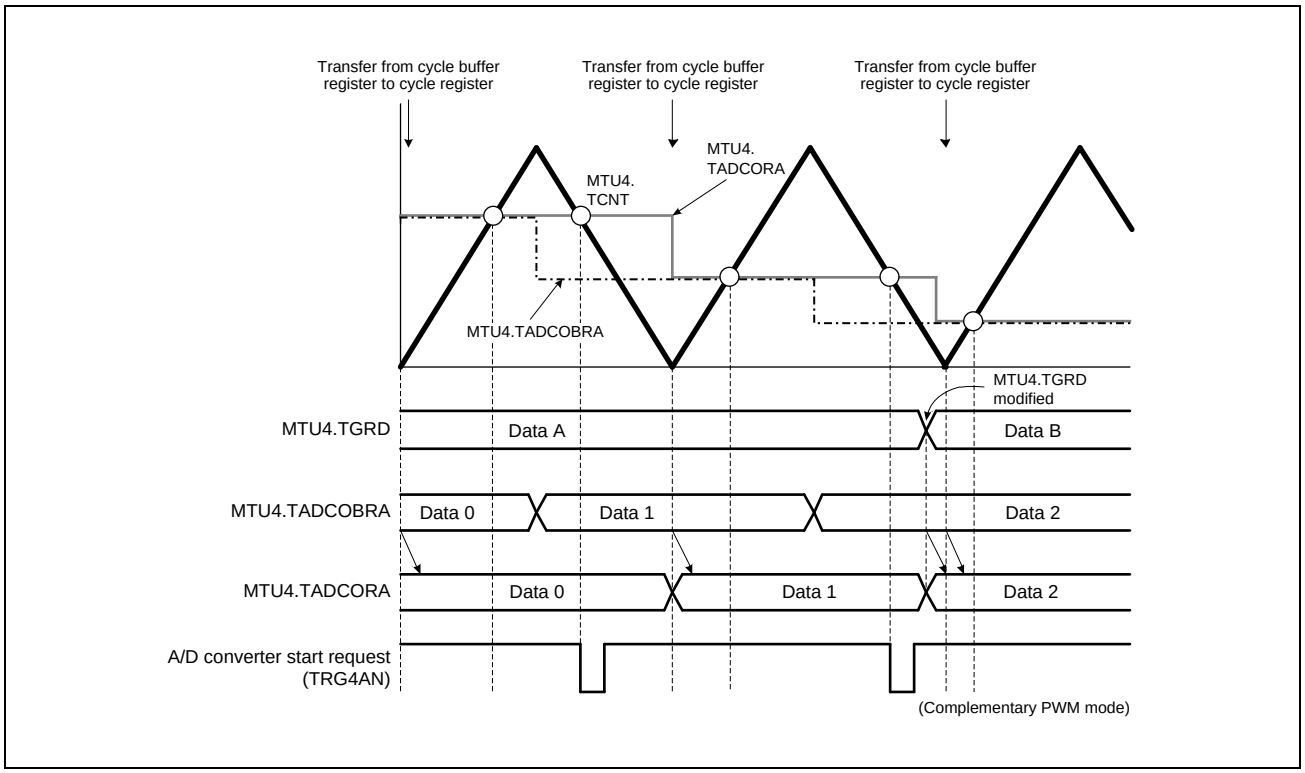



Figure 16.94 Example of A/D Converter Start Request Signal (TRG4AN) and Buffer Transfer Operation

### (5) A/D Converter Start Request Delaying Function Linked with Interrupt Skipping Function 1

In complementary PWM mode, A/D converter start requests (TRG4AN and TRG4BN (TRG7AN and TRG7BN)) can be issued in coordination with interrupt skipping 1 by setting the ITA3AE, ITA4VE, ITB3AE, and ITB4VE (ITA6AE, ITA7VE, ITB6AE, and ITB7VE) bits of the TADCR register in MTU4 (MTU7).

**Figure 16.95** shows an example of A/D converter start request signal (TRG4AN (TRG7AN)) operation when TRG4AN (TRG7AN) output is enabled during MTU4.TCNT (MTU7.TCNT) up-counting and down-counting and A/D converter start requests are linked with interrupt skipping 1.

**Figure 16.96** shows another example of A/D converter start request signal (TRG4AN (TRG7AN)) operation when TRG4AN (TRG7AN) output is enabled during MTU4.TCNT (MTU7.TCNT) up-counting and A/D converter start requests are linked with interrupt skipping 1.

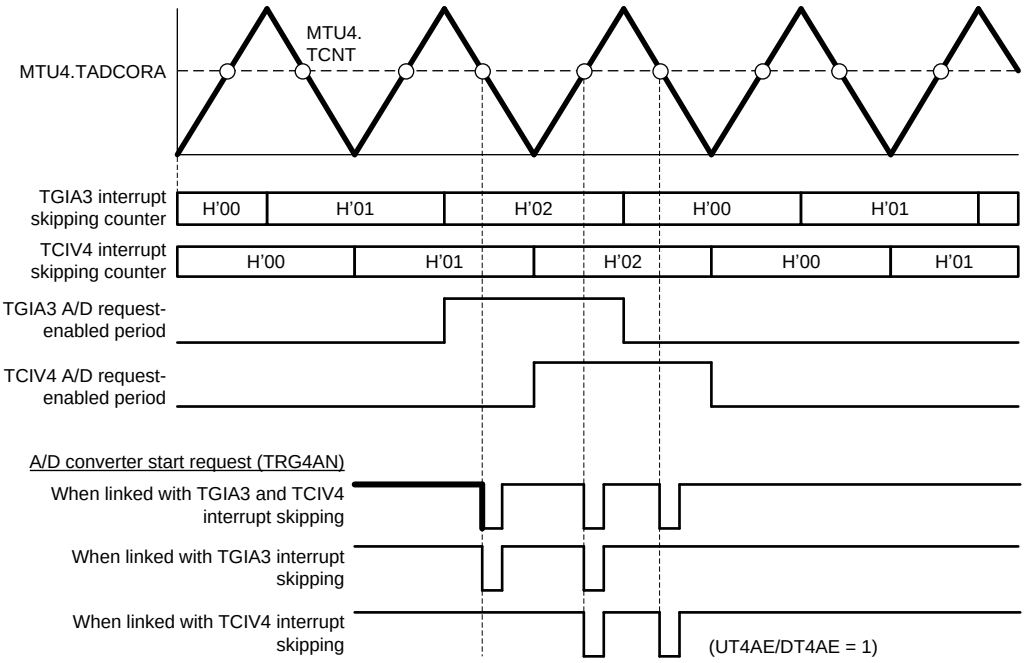
When complementary PWM mode is not selected, A/D converter start request delaying function cannot be used in coordination with interrupt skipping 1; clear the ITA3AE, ITA4VE, ITB3AE, and ITB4VE (ITA6AE, ITA7VE, ITB6AE, and ITB7VE) bits of the TADCR register to 0 in MTU4 (MTU7).

NOTE

This function should be used in combination with interrupt skipping 1.

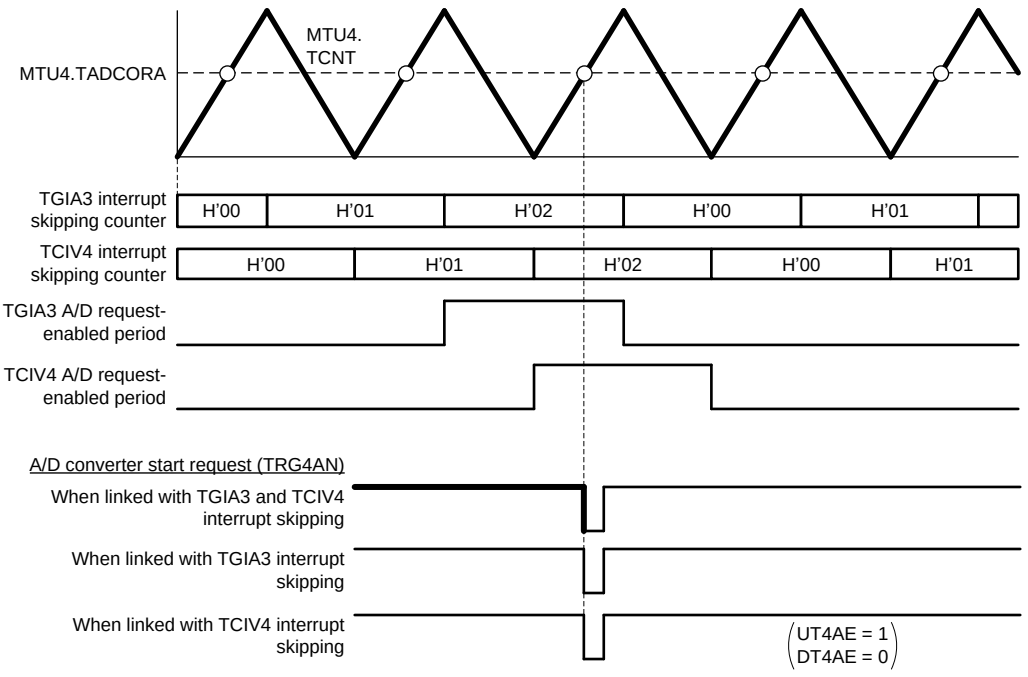
When interrupt skipping is disabled (the T3AEN and T4VEN (T6AEN and T7VEN) bits in the timer interrupt skipping set register (TITCR1A (TITCR1B)) are cleared to 0 or the skipping count set bits (T3ACOR and T4VCOR (T6ACOR and T7VCOR)) in TITCR1A (TITCR1B) are cleared to 0), make sure that A/D converter start requests are not linked with interrupt skipping 1 (clear the ITA3AE, ITA4VE, ITB3AE, and ITB4VE (ITA6AE, ITA7VE, ITB6AE, and ITB7VE) bits in the timer A/D converter start request control register (MTU4.TADCR (MTU7.TADCR)) to 0).

When this function is used, MTU4.TADCORA and MTU4.TADCORB (MTU7.TADCORA and MTU7.TADCORB) should be set with the value ranging H'0002 to the value set in TCDRA minus 2 (value set in TCDRB minus 2).




**Note:** When the interrupt skipping count is set to two.

Figure 16.95 Example of A/D Converter Start Request Signal (TRG4AN) Operation Linked with Interrupt Skipping Function 1 (UT4AE and DT4AE = 1)




**Note:** When the interrupt skipping count is set to two.

Figure 16.96 Example of A/D Converter Start Request Signal (TRG4AN) Operation Linked with Interrupt Skipping Function 1 (UT4AE = 1, DT4AE = 0)

### (6) A/D Converter Start Request Delaying Function Linked with Interrupt Skipping Function 2

By setting the TITM bit to 1 in the TITMRA (TITMRB) register, the counter starts down-counting from the value (0 to 7) set in the TRG4COR[2:0] (TRG7COR[2:0]) bits in TITCR2A (TITCR2B) register every time an A/D converter start trigger (TRG4AN or TRG4BN (TRG7AN or TRG7BN)) is generated. When the counter value reaches 0 and is reloaded, the TRG4AN and TRG4BN (TRG7AN and TRG7BN) interrupts become valid and an A/D converter start request signal (TRG4ABN (TRG7ABN)) is output.

This function is valid only when the A/D converter request delaying function is enabled.

#### (a) Example of Procedure for Setting Interrupt Skipping Function 2

**Figure 16.97** shows an example of procedure for setting interrupt skipping function 2.

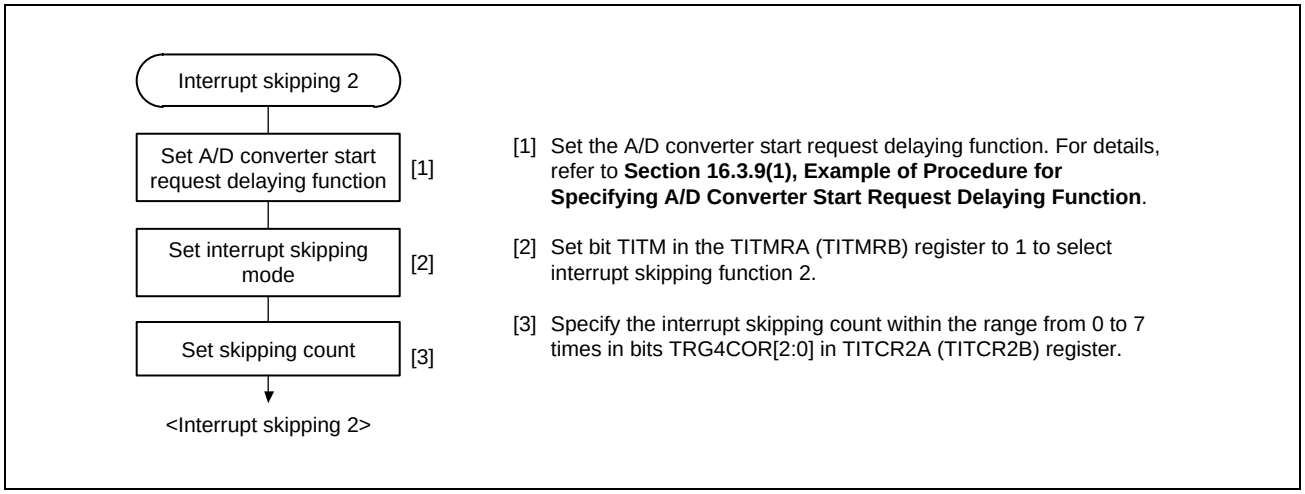



Figure 16.97 Example of Procedure for Setting Interrupt Skipping Function 2

### (b) Example of Interrupt Skipping Function 2 Operation

**Figure 16.98** shows an example of interrupt skipping 2 operation.

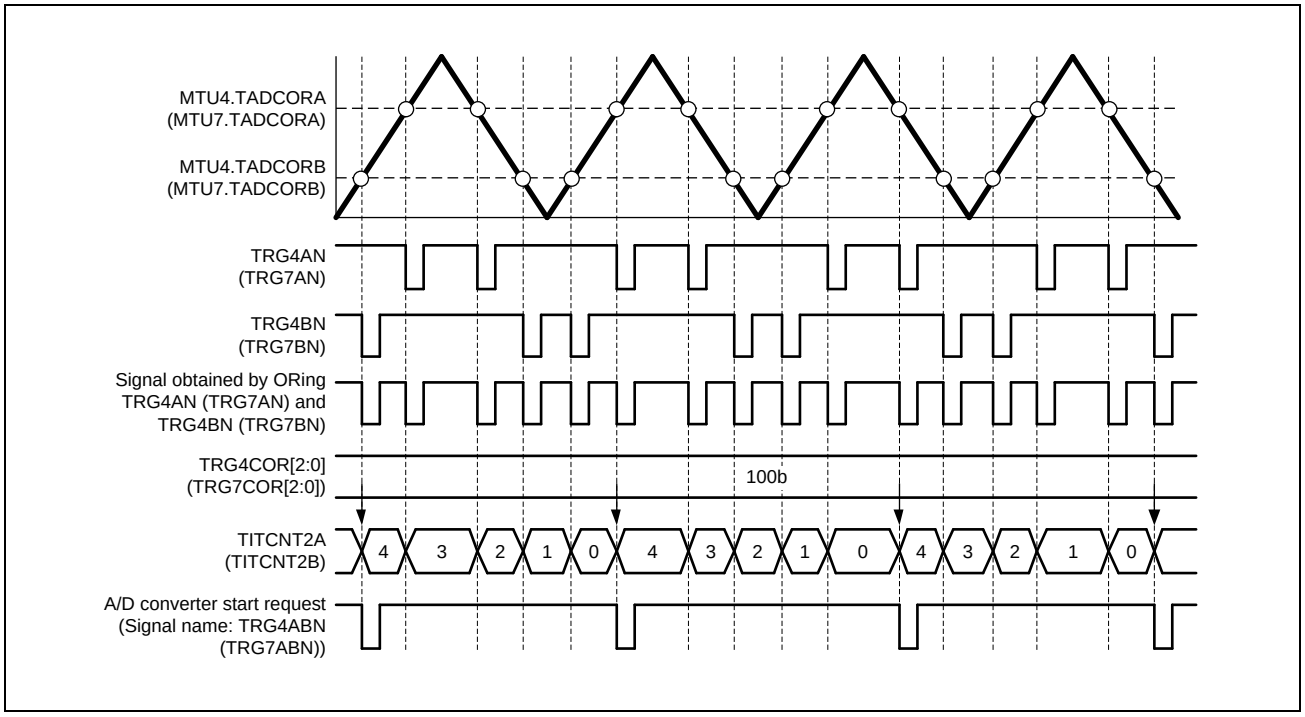



Figure 16.98 Example of Interrupt Skipping 2 Operation (Skipping Count is Set to Four)

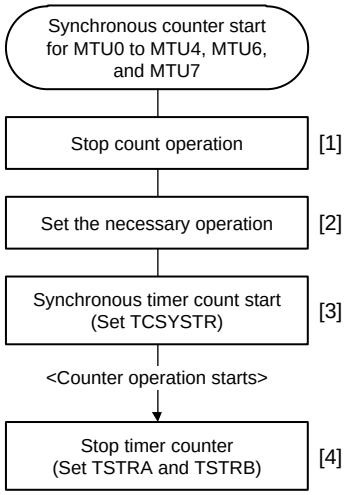
## 16.3.10 Synchronous Operation of MTU0 to MTU4, MTU6, and MTU7

### (1) Synchronous Counter Start for MTU0 to MTU4, MTU6, and MTU7

The counters in MTU0 to MTU4, MTU6, and MTU7 can be started synchronously by making the TCSYSTR settings.

#### (a) Example of Procedure for Setting Synchronous Counter Start for MTU0 to MTU4, MTU6, and MTU7

**Figure 16.99** shows an example of the procedure for specifying synchronous counter start in MTU0 to MTU4, MTU6, and MTU7.




[1] Set TSTRA and TSTRB to stop the counters used for synchronous start operation.

[2] Specify the necessary operation with appropriate registers such as TCR, TMDR1, and TMDR2.

[3] In TCSYSTR, set the bits corresponding to the counters to be started synchronously to 1 at the same time. When TCSYSTR is set, TSTRA and TSTRB for the target timer counters are automatically set appropriately, the counters start synchronously, and TCSYSTR is automatically cleared.

[4] To stop a timer counter, clear the bit in TSTRA or TSTRB that corresponds to the counter to 0.

**Note 1.** Clearing a bit in TCSYSTR corresponding to an operating counter to 0 does not stop the counter.

**Note 2.** To start counters synchronously in reset-synchronized PWM mode or complementary PWM mode using MTU3 and MTU4 or MTU6 and MTU7, set the bits in TCSYSTR that correspond to MTU 3 to MTU4, MTU6, and MTU7 to 1 at the same time.

Figure 16.99 Example of Procedure for Specifying Synchronous Counter Start in MTU0 to MTU4, MTU6, and MTU7

### (b) Examples of Synchronous Counter Start Operation

**Figure 16.100** shows an example of the synchronous counter start operation in MTU0 to MTU4, MTU6, and MTU7.

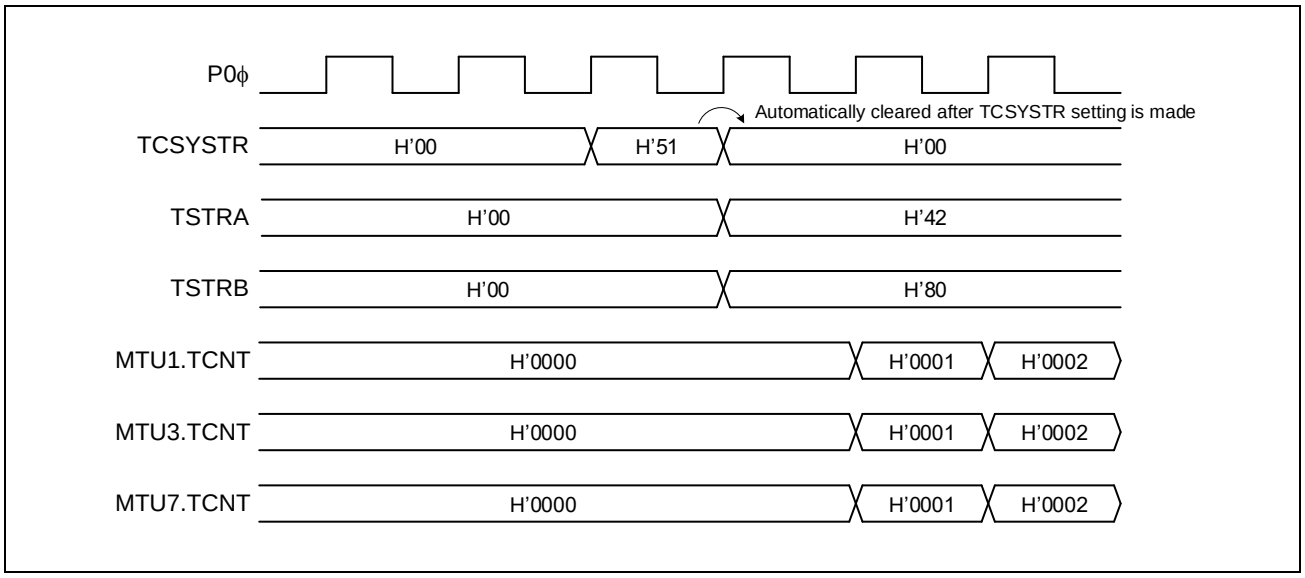


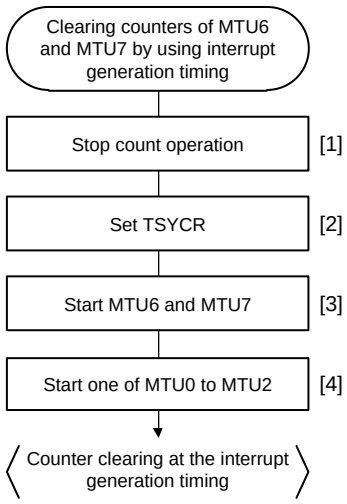

Figure 16.100 Example of Synchronous Counter Start Operation in MTU0 to MTU4, MTU6, and MTU7

### (2) Synchronous Counter Clearing for MTU6 and MTU7

The counters in MTU6 and MTU7 can be cleared by the TGImn interrupt generation timing (m = A to D; n = 0 to 2) through the TSYCR setting.

#### (a) Example of Procedure for Specifying Synchronous Counter Clearing for MTU6 and MTU7

**Figure 16.101** shows an example of the procedure for specifying synchronous counter clearing in MTU6 and MTU7 by using the interrupt generation timing.




[1] Set TSTRA and TSTRB to stop the counters used for this function.

[2] Use TSYCR to specify the source to be used for the MTU6.TCNT and MTU7.TCN counter clearing.

[3] Start MTU6.TCNT or MTU7.TCNT.

[4] Start MTU0.TCNT, MTU1.TCNT, or MTU2.TCNT.

**Note:** The TSYCR setting is ignored while the counter is stopped. The setting becomes valid after MTU6.TCNT or MTU7.TCNT is started.

Figure 16.101 Example of Procedure for Specifying Synchronous Counter Clearing in MTU6 and MTU7

### (b) Examples of Synchronous Counter Clearing in MTU6 and MTU7

**Figure 16.102** and **Figure 16.103** show examples of synchronous counter clearing in MTU6 and MTU7 by using the interrupt generation timing.

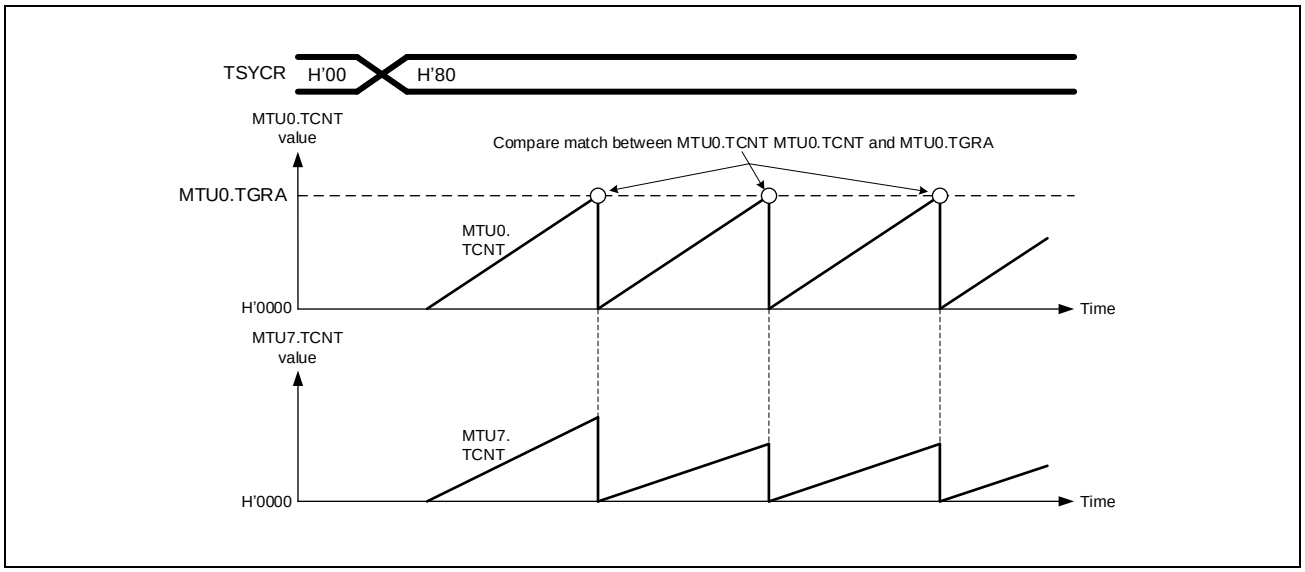



Figure 16.102 Example of Synchronous Counter Clearing in MTU6 and MTU7 (1)

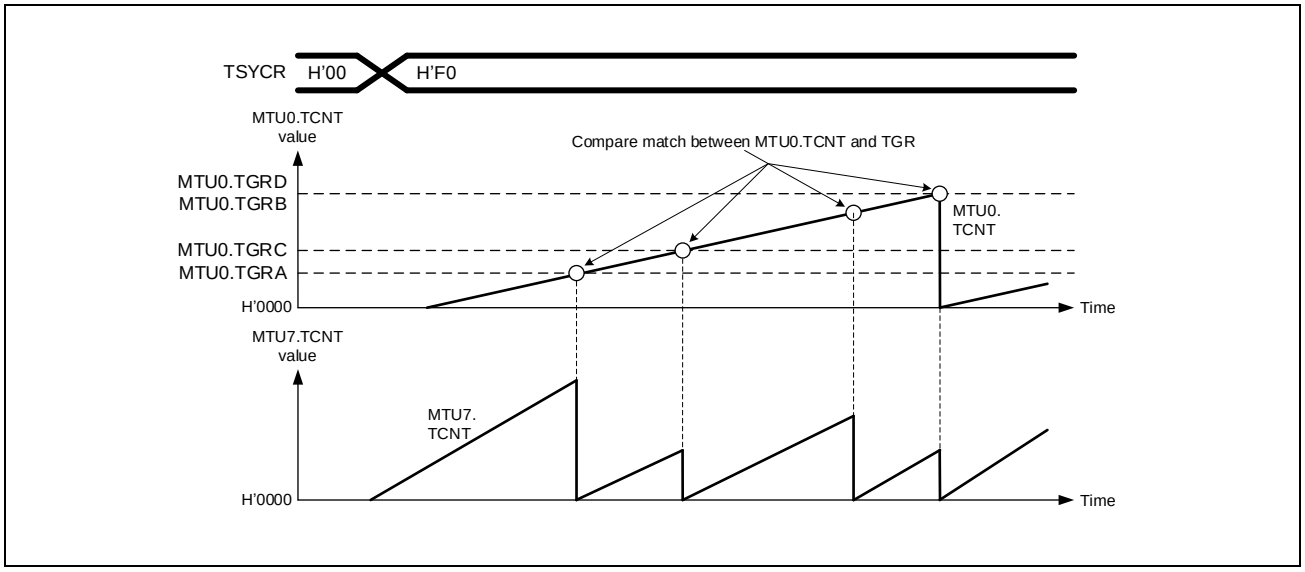


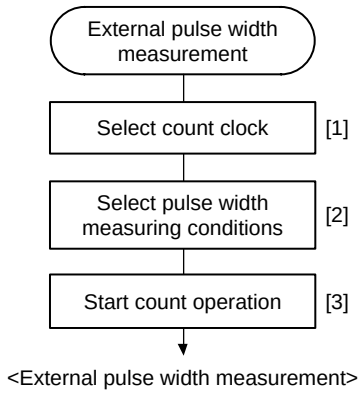

Figure 16.103 Example of Synchronous Counter Clearing in MTU6 and MTU7 (2)

### 16.3.11 External Pulse Width Measurement

The pulse widths of up to three external input lines can be measured in MTU5.

When the IOC[4:0] bits in MTU5.TIORU, MTU5.TIORV, MTU5.TIORW are set for pulse width measurement, the pulse width of the signal input to the MTIC5U, MTIC5V, and MTIC5W pins are measured. TCNTU, TCNTV, and TCNTW count up while the level specified by the IOC[4:0] bits is input.

**Figure 16.104** shows an example of setting external pulse width measurement, and **Figure 16.105** an example of external pulse width measurement.




[1] Use the TPSC1 and TPSC0 bits in MTU5.TCRU, MTU5.TCRV, or MTU5.TCRW to select the count clock.

[2] In MTU5.TIORU, MTU5.TIORV, or MTU5.TIORW, select the high level or low level for the pulse width measuring condition.

[3] Set bit CSTU5, CSTV5, or CSTW5 in MTU5.TSTR to 1 to start count operation.

**Note:** When MTU6 and MTU7 are not operating in complementary PWM mode, note the following 1 to 3.
1. Set the CMPCLR5U, CMPCLR5V, and CMPCLR5W bits in the TCNTCMPCLR register to 0.
2. Set the TGIE5U, TGIE5V, and TGIE5W bits in MTU5.TIER to 0.
3. The TCNT value is not captured in the TGR register.

Figure 16.104 Example of External Pulse Width Measurement Setting Procedure

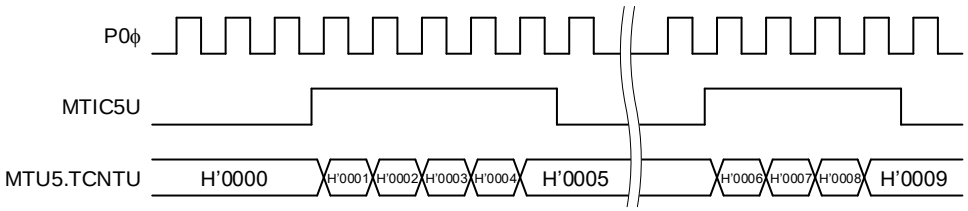



Figure 16.105 Example of External Pulse Width Measurement (Measuring High Pulse Width)

## 16.3.12 Dead Time Compensation

MTU5 to MTU7 can be used in combination to compensate for the delay in the dead time (the delay between complementary PWM output and inverter output).

**Figure 16.106** shows an example of a motor control circuit in which the combination of MTU5 to MTU7 is used to compensate for the dead time delay.

The external pulse measurement function of MTU5 allows the specification of a correction to the duty cycle specified in the PWM output compare registers by measuring the delay between the complementary PWM output and inverter output. This can be used for dead time compensation for the PWM output waveform during complementary PWM operation of MTU6 and MTU7 (**Figure 16.107**).

**Figure 16.108** shows the procedure for setting dead time compensation using MTU5 to MTU7. For details on MTU5 operation at this time, refer to **Section 16.3.13, TCNTU, TCNTV, and TCNTW Capture at Crest and/or Trough in Complementary PWM Mode**.

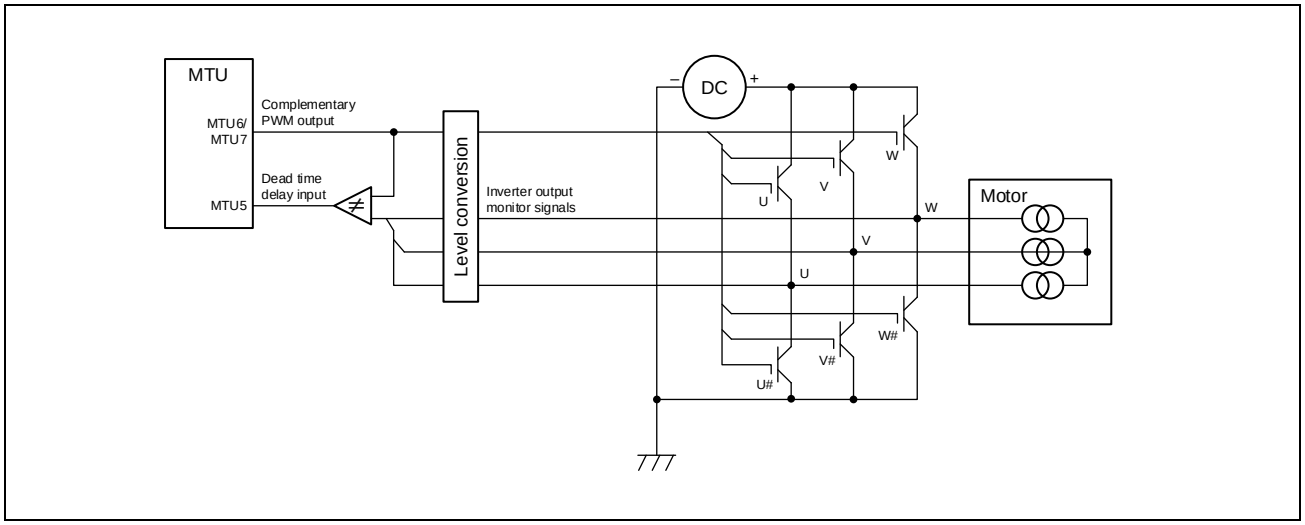



Figure 16.106 Motor Control Circuit Example

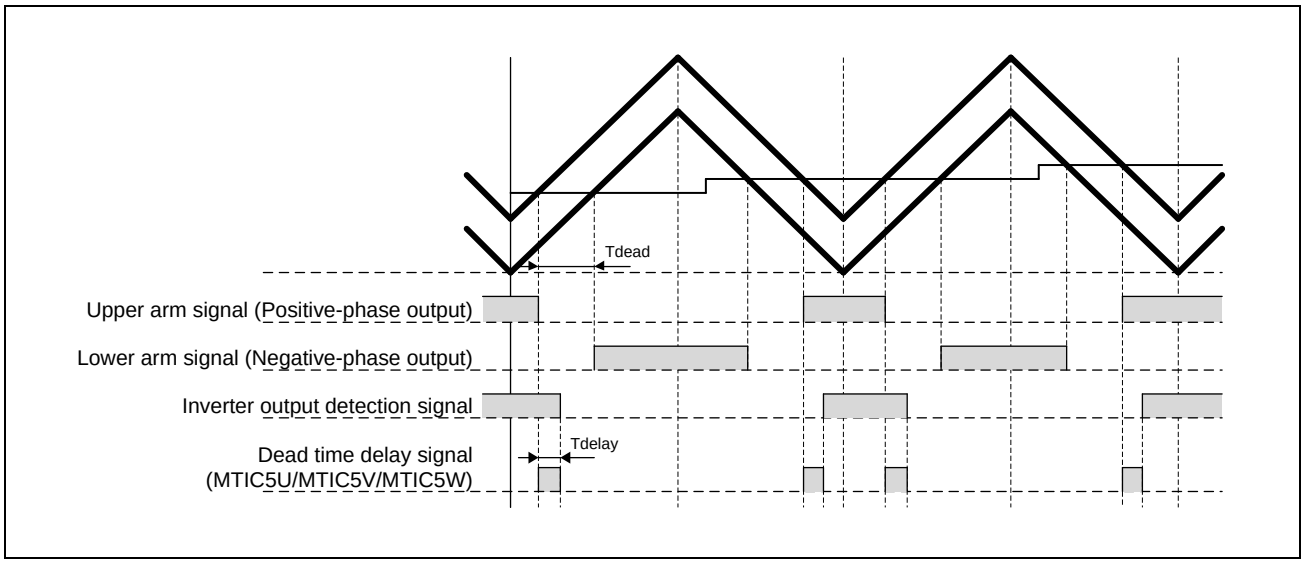
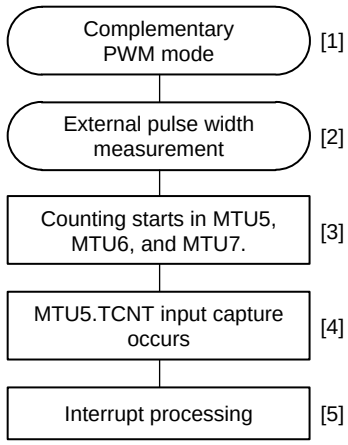



Figure 16.107 Delay in Dead Time in Complementary PWM Operation

### (1) Example of Dead Time Compensation Setting Procedure

**Figure 16.108** shows an example of dead time compensation setting procedure by using three counters in MTU5.




[1] Set MTU6 and MTU7 to complementary PWM mode. For details, refer to **Section 16.3.8, Complementary PWM Mode**.

[2] Specify the external pulse width measurement function for the target TIOR in MTU5. For details, refer to **Section 16.3.11, External Pulse Width Measurement**.

[3] Set bits CST6 and CST7 in TSTRB and bits CSTU5, CSTV5, and CSTW5 in MTU5.TSTR to 1 to start count operation.

[4] When the capture condition specified in TIOR is satisfied, the MTU5.TCNT value is captured in MTU5.TGR.
To prevent the generation of the MTU5.TGIm5 interrupt (m = U, V, W) request under the capture condition specified in TIOR, do not enable interrupt requests in MTU5.TIER.

[5] For U-phase dead time compensation, when an interrupt is generated at the crest (TGIA6) or trough (TCIV7) in complementary PWM mode, read the MTU5.TGRU value, calculate the difference in time between MTU6.TGRB and MTU5.TGRU, and write the corrected value to MTU6.TGRD in the interrupt processing.
For the V phase and W phase, read the MTU5.TGRV and MTU5.TGRW values and write the corrected values to MTU7.TGRC and TGRD, respectively, in the same way as for U-phase compensation.
The MTU5.TCNT value should be cleared through the TCNTCMPCLR setting or by software.

**Note:** The port mode registers (PMn) and Pmn port function control register (PFCm) of the GPIO ports should be set up in advance.
As the interrupt flag is set under the capture condition specified in TIOR, do not enable interrupt requests in MTU5.TIER.

Figure 16.108 Example of Dead Time Compensation Setting Procedure

### 16.3.13 TCNTU, TCNTV, and TCNTW Capture at Crest and/or Trough in Complementary PWM Mode

The external pulse width measurement function of MTU5 can be used to transfer the values of TCNTU, TCNTV, and TCNTW to TGRU, TGRV, and TGRW at the crests, troughs, or both crests and troughs of the complementary PWM output when MTU6 and MTU7 are being used in complementary PWM mode. The type of transfer timing is specified in TIORU, TIORV, and TIORW. When the CMPCLR5U, CMPCLR5V, and CMPCLR5W bits in the TCNTCNPCLR register are set to 1, TCNTU, TCNTV, and TCNTW are cleared to H'0000 at the timing for transfer to TGRU, TGRV, and TGRW.

Note that the TCNTU, TCNTV, and TCNTW capture operation at the crest and/or trough of complementary PWM output in MTU5 is not available when MTU3 and MTU4 are used in complementary PWM mode.

**Figure 16.109** shows an operation example in which TCNTU is used as a free-running counter without being cleared and the value is captured in TGRU at the crest and trough in complementary PWM mode.

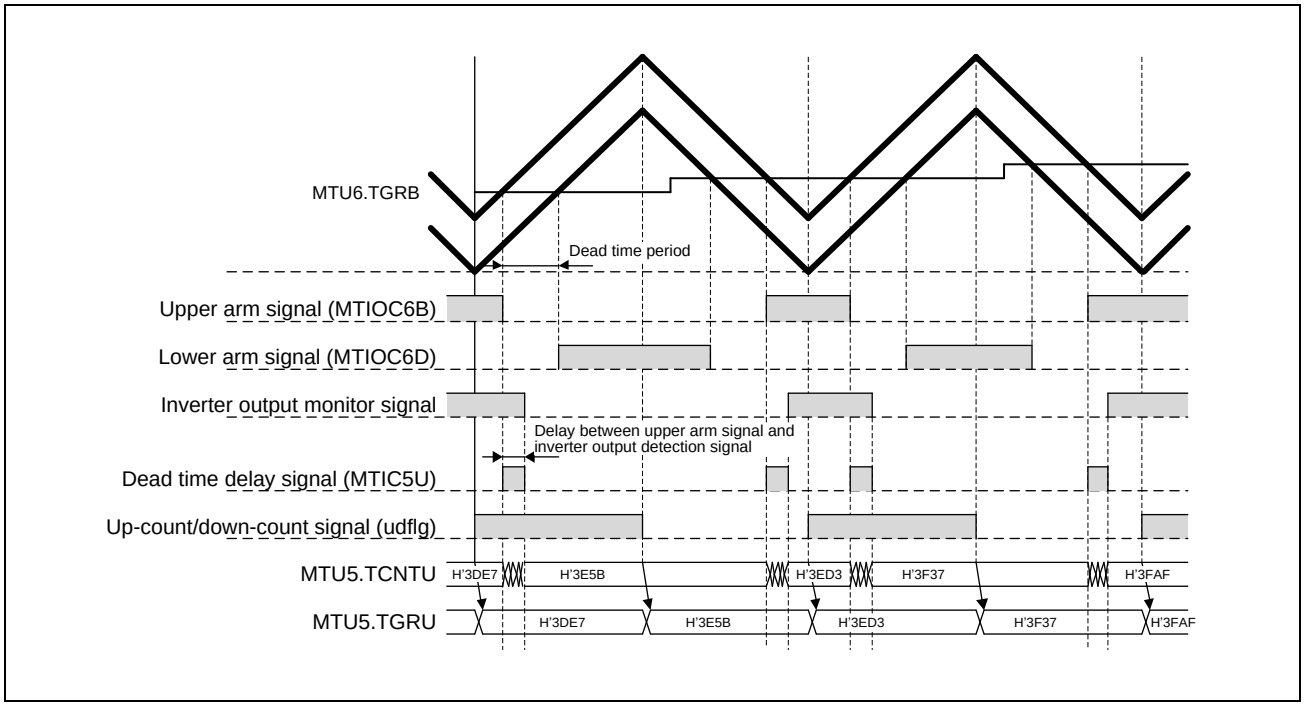



Figure 16.109 TCNTU Capture at Crest and Trough in Complementary PWM Operation

### 16.3.14 Noise Filter Function

The input capture input pins and external pulse input pins have a noise filter function.

Set the NFCRn register (n = 0 to 7, C) to enable or disable the noise filter function and set the sampling clock. The noise filter for each pin can be enabled or disabled individually, and the sampling clock can be set for each channel.

**Figure 16.110** shows the timing of noise filtering.

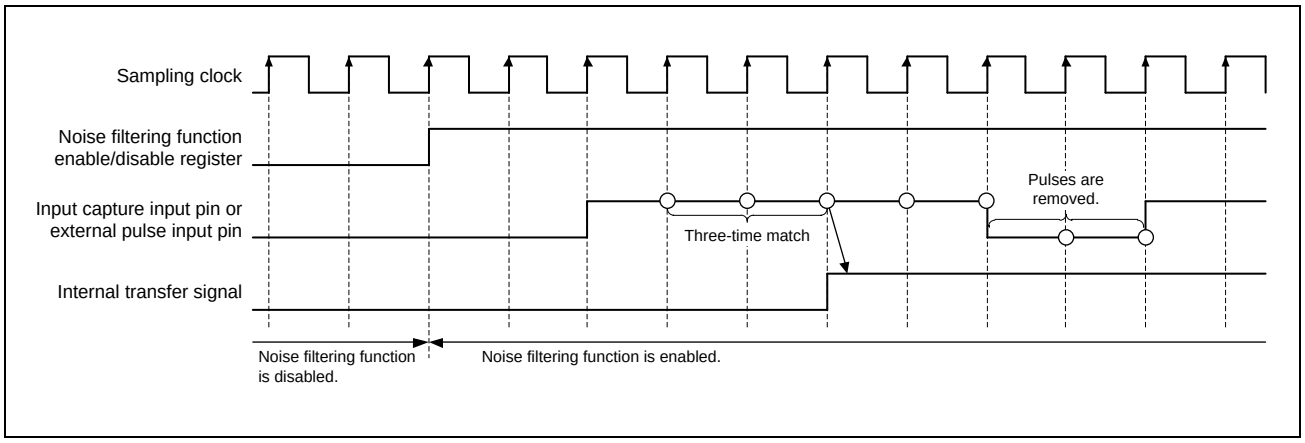



Figure 16.110 Timing of Noise Filtering

## 16.4 Interrupt Sources

### 16.4.1 Interrupt Sources and Priorities

There are three kinds of interrupt source; TGR input capture/compare match, TCNT overflow, and TCNT underflow. Each interrupt source has its own enable/disable bit, allowing the generation of interrupt request signals to be enabled or disabled individually.

When an interrupt source is generated, if the corresponding enable/disable bit in TIER is set to 1, an interrupt is requested.

Relative channel priorities can be changed by the interrupt controller; however the priority within a channel is fixed. For details, refer to **Section 8, Interrupt Controller**. **Table 16.78** lists the MTU interrupt sources.

Table 16.78 MTU Interrupt Sources (1/2)

| Channel | Name | Interrupt Source | DMAC Activation |
|---|---|---|---|
| MTU0 | TGIA0 | MTU0.TGRA input capture/compare match | Possible |
| | TGIB0 | MTU0.TGRB input capture/compare match | Possible |
| | TGIC0 | MTU0.TGRC input capture/compare match | Possible |
| | TGID0 | MTU0.TGRD input capture/compare match | Possible |
| | TCIV0 | MTU0.TCNT overflow | Not possible |
| | TGIE0 | MTU0.TGRE compare match | Not possible |
| | TGIF0 | MTU0.TGRF compare match | Not possible |
| MTU1 | TGIA1 | MTU1.TGRA input capture/compare match | Possible |
| | TGIB1 | MTU1.TGRB input capture/compare match | Possible |
| | TCIV1 | MTU1.TCNT overflow | Not possible |
| | TCIU1 | MTU1.TCNT underflow | Not possible |
| MTU2 | TGIA2 | MTU2.TGRA input capture/compare match | Possible |
| | TGIB2 | MTU2.TGRB input capture/compare match | Possible |
| | TCIV2 | MTU2.TCNT overflow | Not possible |
| | TCIU2 | MTU2.TCNT underflow | Not possible |
| MTU3 | TGIA3 | MTU3.TGRA input capture/compare match | Possible |
| | TGIB3 | MTU3.TGRB input capture/compare match | Possible |
| | TGIC3 | MTU3.TGRC input capture/compare match | Possible |
| | TGID3 | MTU3.TGRD input capture/compare match | Possible |
| | TCIV3 | MTU3.TCNT overflow | Not possible |
| MTU4 | TGIA4 | MTU4.TGRA input capture/compare match | Possible |
| | TGIB4 | MTU4.TGRB input capture/compare match | Possible |
| | TGIC4 | MTU4.TGRC input capture/compare match | Possible |
| | TGID4 | MTU4.TGRD input capture/compare match | Possible |
| | TCIV4 | MTU4.TCNT overflow/underflow*1 | Possible |
| MTU5 | TGIU5 | MTU5.TGRU input capture/compare match | Possible |
| | TGIV5 | MTU5.TGRV input capture/compare match | Possible |
| | TGIW5 | MTU5.TGRW input capture/compare match | Possible |
| MTU6 | TGIA6 | MTU6.TGRA input capture/compare match | Possible |
| | TGIB6 | MTU6.TGRB input capture/compare match | Possible |
| | TGIC6 | MTU6.TGRC input capture/compare match | Possible |
| | TGID6 | MTU6.TGRD input capture/compare match | Possible |
| | TCIV6 | MTU6.TCNT overflow | Not possible |

Table 16.78 MTU Interrupt Sources (2/2)

| Channel | Name | Interrupt Source | DMAC Activation |
|---|---|---|---|
| MTU7 | TGIA7 | MTU7.TGRA input capture/compare match | Possible |
| | TGIB7 | MTU7.TGRB input capture/compare match | Possible |
| | TGIC7 | MTU7.TGRC input capture/compare match | Possible |
| | TGID7 | MTU7.TGRD input capture/compare match | Possible |
| | TCIV7 | MTU7.TCNT overflow/underflow*1 | Possible |
| MTU8 | TGIA8 | MTU8.TGRA input capture/compare match | Possible |
| | TGIB8 | MTU8.TGRB input capture/compare match | Possible |
| | TGIC8 | MTU8.TGRC input capture/compare match | Possible |
| | TGID8 | MTU8.TGRD input capture/compare match | Possible |
| | TCIV8 | MTU8.TCNT overflow | Possible |

**Note:** This table shows the initial state immediately after a reset. The relative channel priorities can be changed by the interrupt controller.

Note 1. The underflow interrupt source is valid only in complementary PWM mode.

#### (1) Input Capture/Compare Match Interrupt

If the TIER.TGIE bit is set to 1 when a TGR input capture/compare match occurs on a channel, an interrupt is requested. The MTU has 33 input capture/compare match interrupts (six for MTU0, four each for MTU3, MTU4, MTU6, MTU7, and MTU8, two each for MTU1 and MTU2, and three for MTU5).

#### (2) Overflow Interrupt

If the TIER.TCIEV bit is set to 1 when a TCNT overflow occurs on a channel, clearing an interrupt is requested. The MTU provides a total of eight overflow interrupts (one for each channel except MTU5).

In complementary PWM mode, an overflow interrupt is generated even when an underflow occurs in TCNT in MTU4 or MTU7.

#### (3) Underflow Interrupt

If the TIER.TCIEU bit is set to 1 when a TCNT underflow occurs on a channel, an interrupt is requested. The MTU has two underflow interrupts (one each for MTU1, and MTU2).

### 16.4.2 DMAC Activation

#### (1) DMAC Activation

The DMAC can be activated by the TGR input capture/compare match interrupt and the overflow interrupt in MTU4 and MTU7. For details, see **Section 14, Direct Memory Access Controller**.

The MTU provides a total of 33 interrupts (input capture/compare match and overflow interrupts) that can be used as DMAC activation sources: four each for MTU0, MTU3, MTU6, and MTU8, two each for MTU1 and MTU2, five each for MTU4 and MTU7, and three for MTU5.

### 16.4.3 A/D Converter Activation

The A/D converter can be activated by one of the following three methods in the MTU. **Table 16.79** 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 MTU4.TCNT (MTU7.TCNT) 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 MTU4.TIER (MTU7.TIER) is set to 1, the A/D converter can be activated at the trough of MTU4.TCNT (MTU7.TCNT) count (MTU4.TCNT (MTU7.TCNT) = H'0000).

A/D converter start request signal TRGAnN is issued to the A/D converter under either of the following conditions (n = 0 to 4, 6, or 7).

- When a TGRA input capture/compare match occurs on a channel while the TIER.TTG bit is set to 1
- When the MTU4.TCNT (MTU7.TCNT) count reaches the trough (MTU4.TCNT (MTU7.TCNT) = H'0000) during complementary PWM operation while the TTGE2 bit in MTU4.TIER (MTU7.TIER) is set to 1

When either condition is satisfied, if A/D converter start signal TRGAnN from the MTU is selected as the trigger in the A/D converter, A/D conversion will start.

#### (2) A/D Converter Activation by Compare Match between MTU0.TCNT and MTU0.TGRE

A/D converter start request signal TRG0N is issued to the A/D converter when a compare match occurs between MTU0.TCNT and MTU0.TGRE.

When a compare match occurs between MTU0.TCNT and MTU0.TGRE in MTU0 while the TTGE2 bit.TIER2 is set to 1, A/D converter start request TGR0N is issued to the A/D converter. If A/D converter start signal TRG0N from the MTU 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 (TRG7AN or TRG7BN) when the MTU4.TCNT (MTU7.TCNT) count matches the MTU4.TADCORA or MTU4.TADCORB (MTU7.TADCORA or MTU7.TADCORB) value if the UT4AE, DT4AE, UT4BE, or DT4BE (UT7AE, DT7AE, UT7BE, or DT7BE) bit in the A/D converter start request control register (MTU4.TADCR (MTU7.TADCR)) is set to 1. For details, refer to **Section 16.3.9, A/D Converter Start Request Delaying Function**.

A/D conversion will start when TRG4AN (TRG7AN) is generated if A/D converter start signal TRG4AN (TRG7AN) from the MTU is selected as the trigger in the A/D converter, when TRG4BN (TRG7BN) is generated if TRG4BN (TRG7BN) from the MTU is selected as the trigger in the A/D converter, or when TRG4ABN (TRG7ABN) is generated if TRG4ABN (TRG7ABN) from the MTU is selected as the trigger in the A/D converter.

Table 16.79 Interrupt Sources and A/D Converter Start Request Signals

| Target Registers | Interrupt Source | A/D Converter Start Request Signal |
|---|---|---|
| MTU0.TGRA and MTU0.TCNT | Input capture/compare match | TRGA0N |
| MTU1.TGRA and MTU1.TCNT | | TRGA1N |
| MTU2.TGRA and MTU2.TCNT | | TRGA2N |
| MTU3.TGRA and MTU3.TCNT | | TRGA3N |
| MTU4.TGRA and MTU4.TCNT*1 | | TRGA4N |
| MTU4.TCNT | MTU4.TCNT trough in complementary PWM mode | |
| MTU6.TGRA and MTU6.TCNT | Input capture/compare match | TRGA6N |
| MTU7.TGRA and MTU7.TCNT*1 | | TRGA7N |
| MTU7.TCNT | MTU7.TCNT trough in complementary PWM mode | |
| MTU0.TGRE and MTU0.TCNT | Compare match | TRG0N |
| MTU4.TADCORA and MTU4.TCNT | | TRG4AN |
| MTU4.TADCORB and MTU4.TCNT | | TRG4BN |
| MTU7.TADCORA and MTU7.TCNT | | TRG7AN |
| MTU7.TADCORB and MTU7.TCNT | | TRG7BN |
| MTU4.TADCORA and MTU4.TCNT, MTU4.TADCORB and MTU4.TCNT | Compare match (interrupt skipping function 2) | TRG4ABN |
| MTU7.TADCORA and MTU7.TCNT, MTU7.TADCORB and MTU7.TCNT | | TRG7ABN |

Note 1. Since PWM waveforms are generated in complementary PWM mode, MTU4.TGRA (MTU7.TGRA) compare match not only with MTU4.TCNT (MTU7.TCNT) but also with MTU3.TCNT (MTU6.TCNT) and TCNTSA (TCNTSB) is detected. Accordingly, when compare match with MTU3.TCNT (MTU6.TCNT) and TCNTSA (TCNTSB) occurs, TRGA4N (TRGA7N) is also generated.
When MTU3 and MTU 4 (MTU 6 and MTU7) are made to operate in complementary PWM mode for generating an A/D converter start request, use the A/D converter start request by compare match between MTU4.TCNT (MTU7.TCNT) and MTU4.TADCORA or TADCORB (MTU7.TADCORA or TADCORB).

# 16.5 Operation Timing

## 16.5.1 Input/Output Timing

### (1) TCNT Count Timing

**Figure 16.111** and **Figure 16.112** show the TCNT count timing in internal clock operation, **Figure 16.113** shows the TCNT count timing in external clock operation (normal mode), and **Figure 16.114** shows the TCNT count timing in external clock operation (phase counting mode).

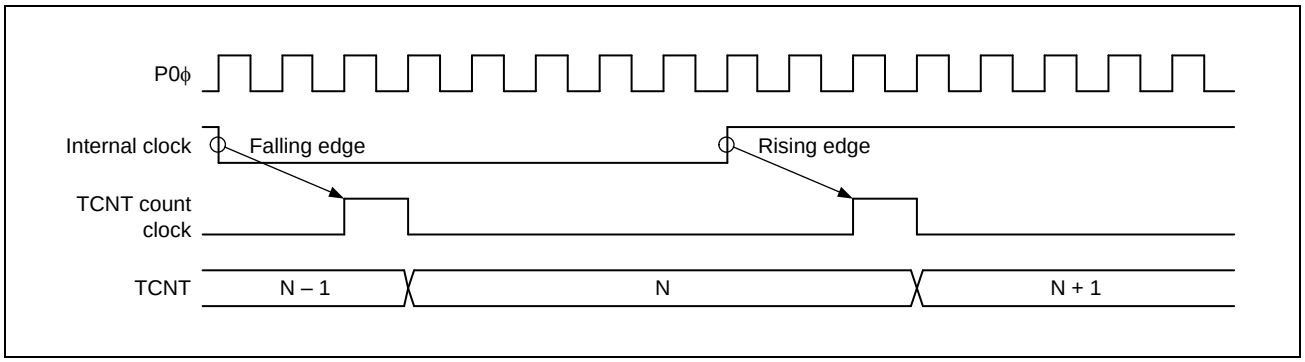



Figure 16.111 Count Timing in Internal Clock Operation (MTU0 to MTU4 and MTU6 to MTU8)

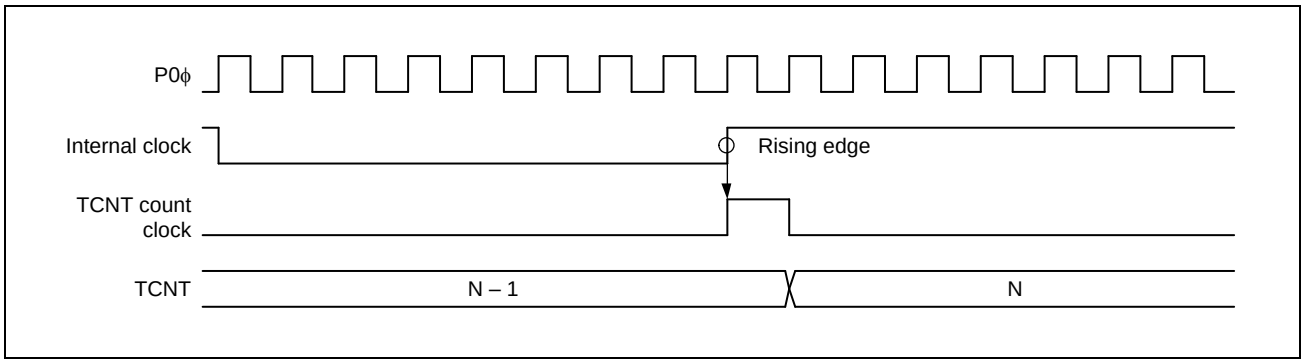



Figure 16.112 Count Timing in Internal Clock Operation (MTU5)

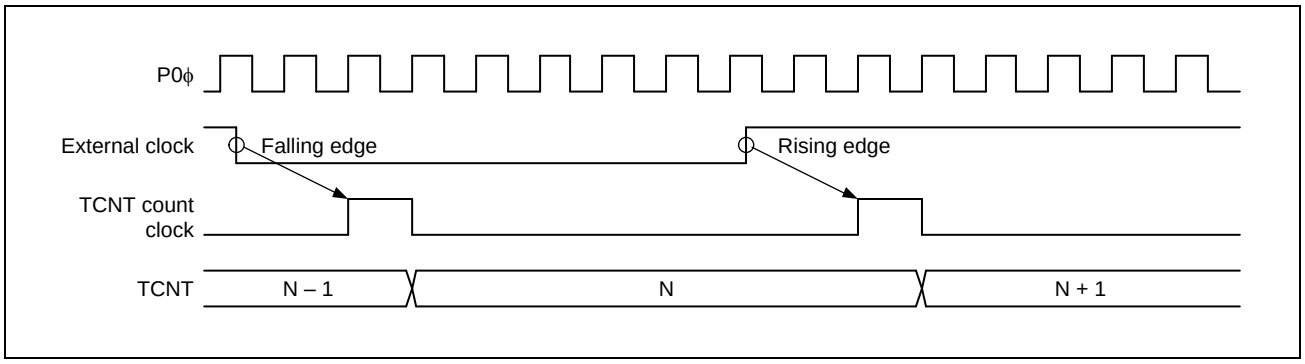



Figure 16.113 Count Timing in External Clock Operation (MTU0 to MTU4, MTU6 to MTU8)

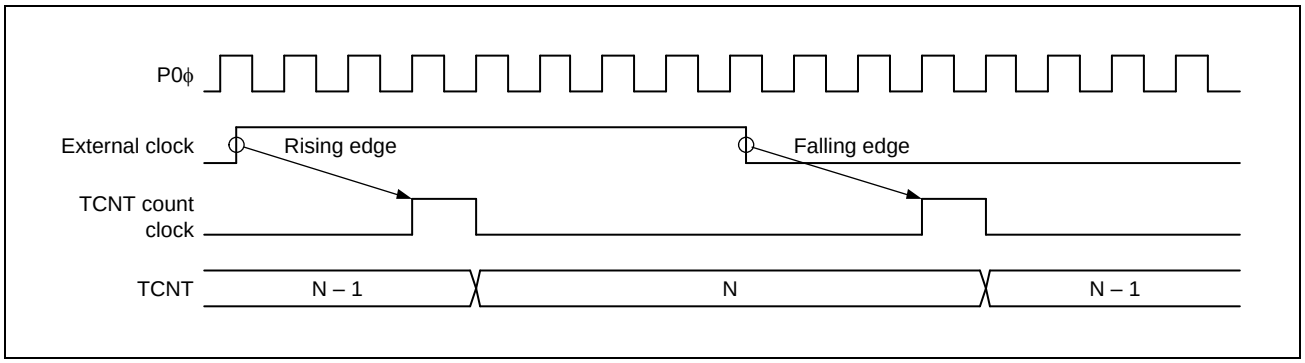



Figure 16.114 Count Timing in External Clock Operation (Phase Counting Mode)

### (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 value set in TIOR is output from MTIOCnm pin (n = 0 to 4, 6, 7, 8; m = A to D). After a match between TCNT and TGR, the compare match signal is not generated until the TCNT count clock is generated.

**Figure 16.115** shows the output compare output timing (normal mode or PWM mode) and **Figure 16.116** shows the output compare output timing (complementary PWM mode or reset-synchronized PWM mode).

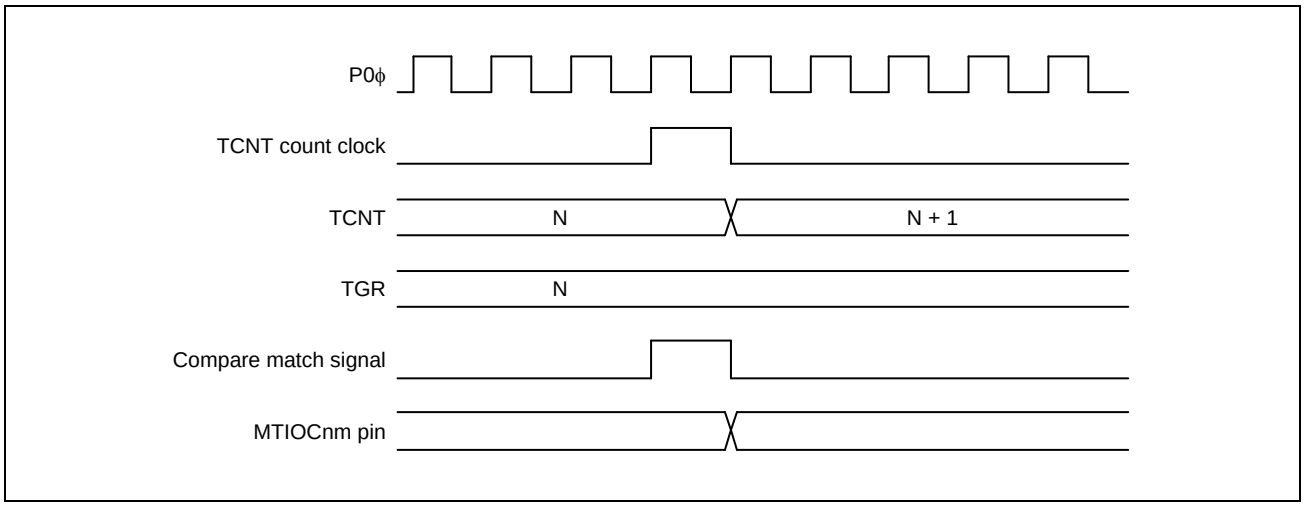



Figure 16.115 Output Compare Output Timing (Normal Mode or PWM Mode) (n = 0 to 4, 6, 7, 8; m = A to D)

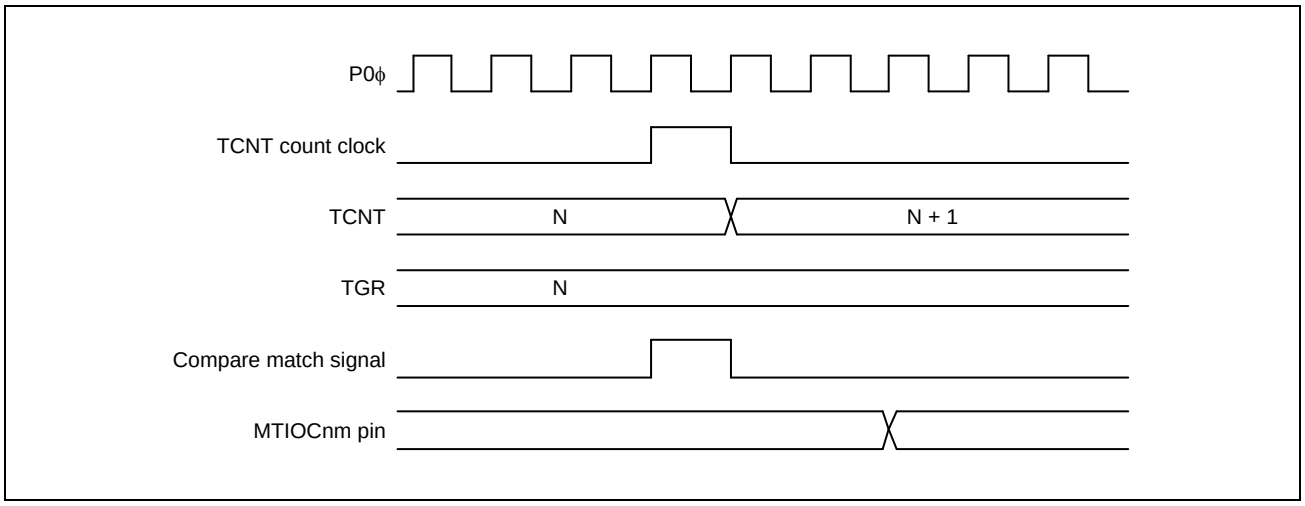



Figure 16.116 Output Compare Output Timing (Complementary PWM Mode or Reset-Synchronized PWM Mode) (n = 0 to 4, 6, 7, 8; m = A to D)

### (3) Input Capture Signal Timing

**Figure 16.117** shows the input capture signal timing.

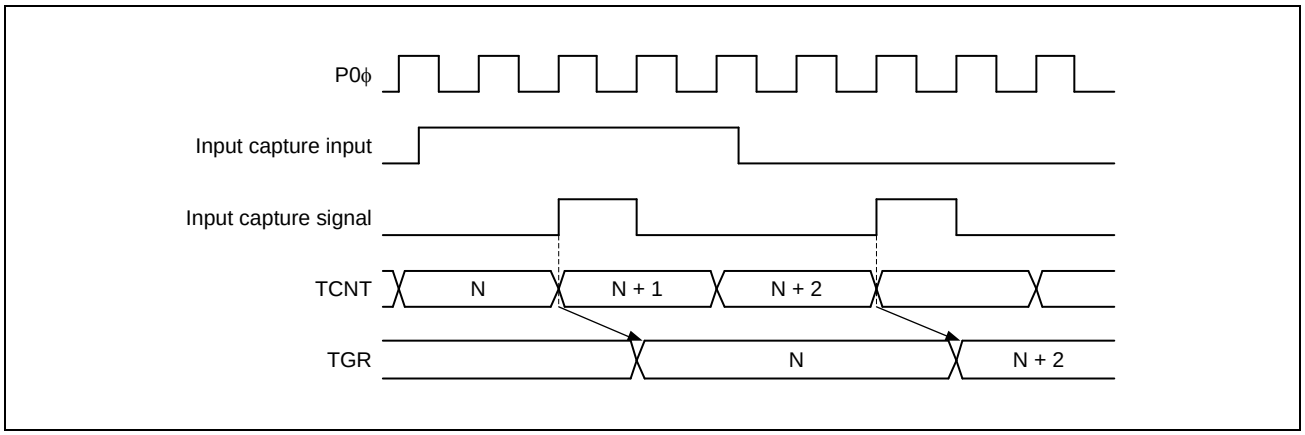



Figure 16.117 Input Capture Input Signal Timing

### (4) Timing for Counter Clearing by Compare Match/Input Capture

**Figure 16.118** and **Figure 16.119** show the timing when counter clearing on compare match is specified, and **Figure 16.120** shows the timing when counter clearing on input capture is specified.

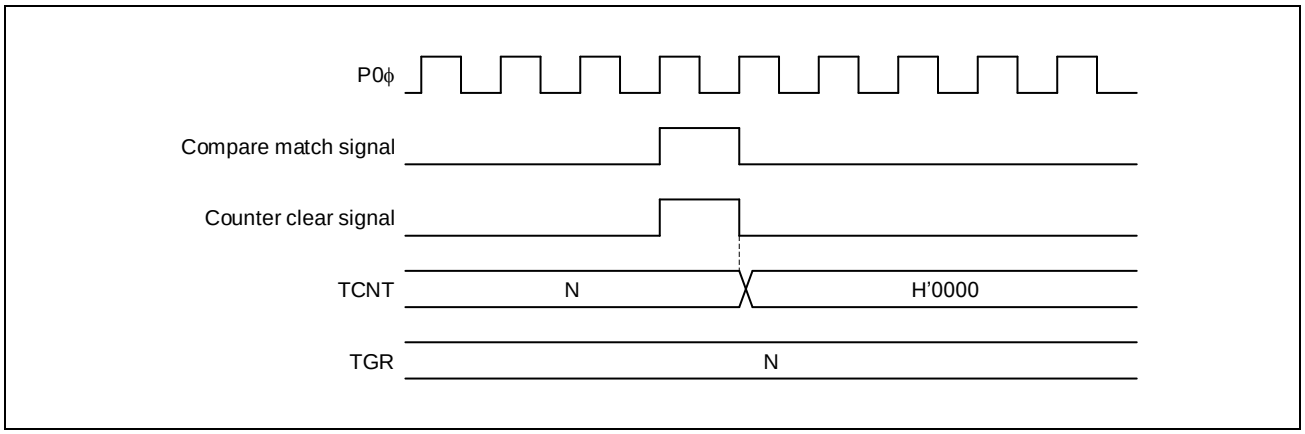



Figure 16.118 Counter Clear Timing (Compare Match) (MTU0 to MTU4 and MTU6 to MTU8)

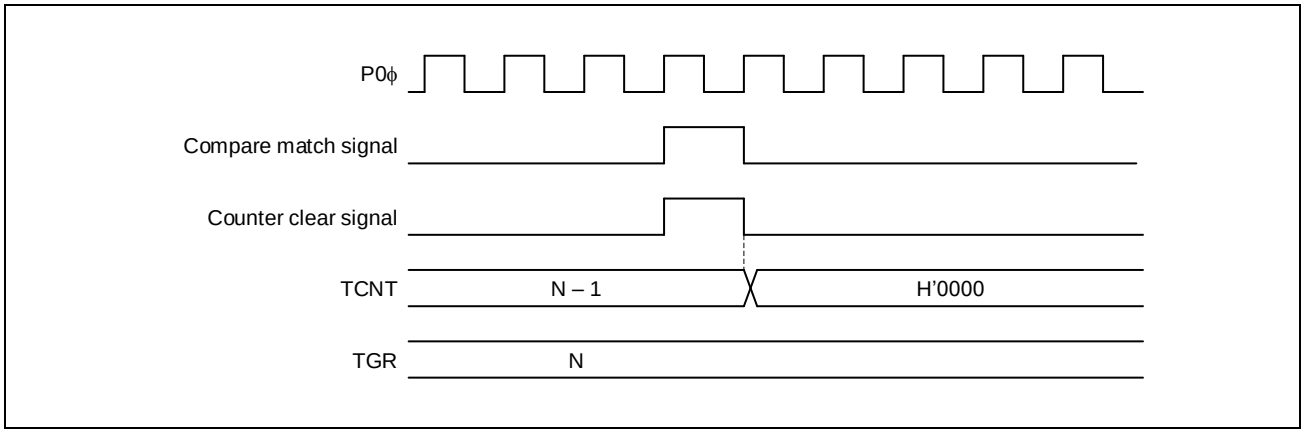



Figure 16.119 Counter Clear Timing (Compare Match) (MTU5)

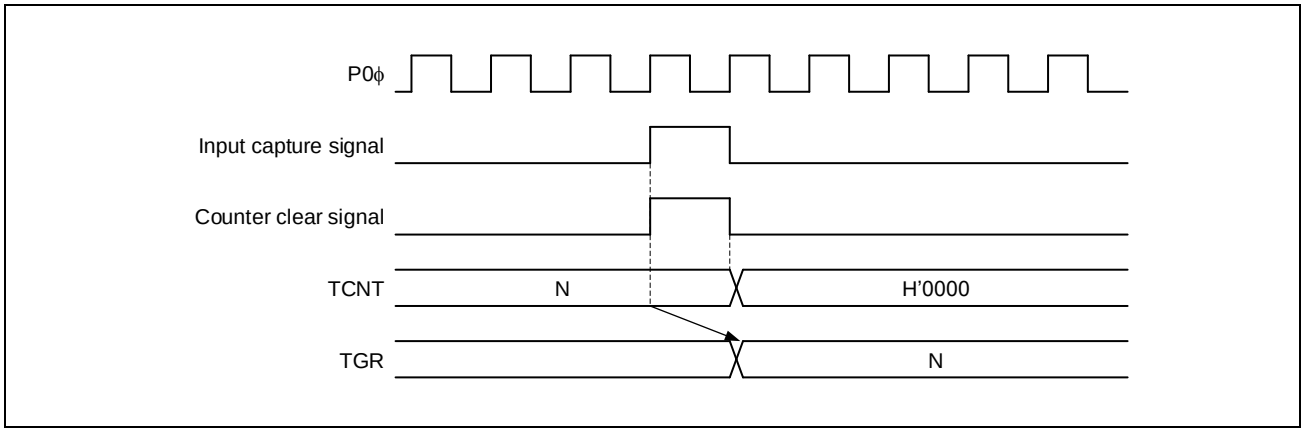



Figure 16.120 Counter Clear Timing (Input Capture) (MTU0 to MTU8)

### (5) Buffer Operation Timing

**Figure 16.121** to **Figure 16.123** show the timing in buffer operation.

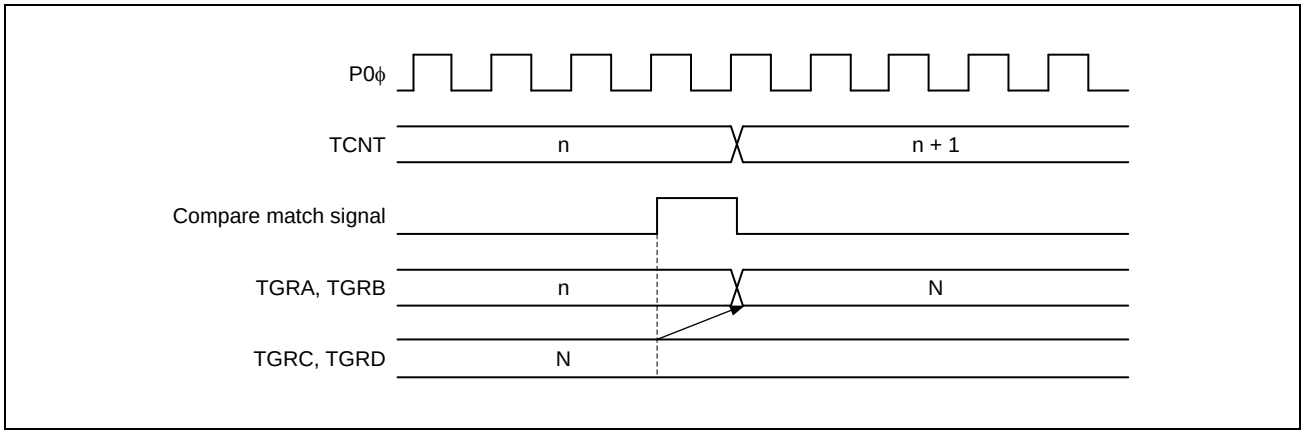



Figure 16.121 Buffer Operation Timing (Compare Match)

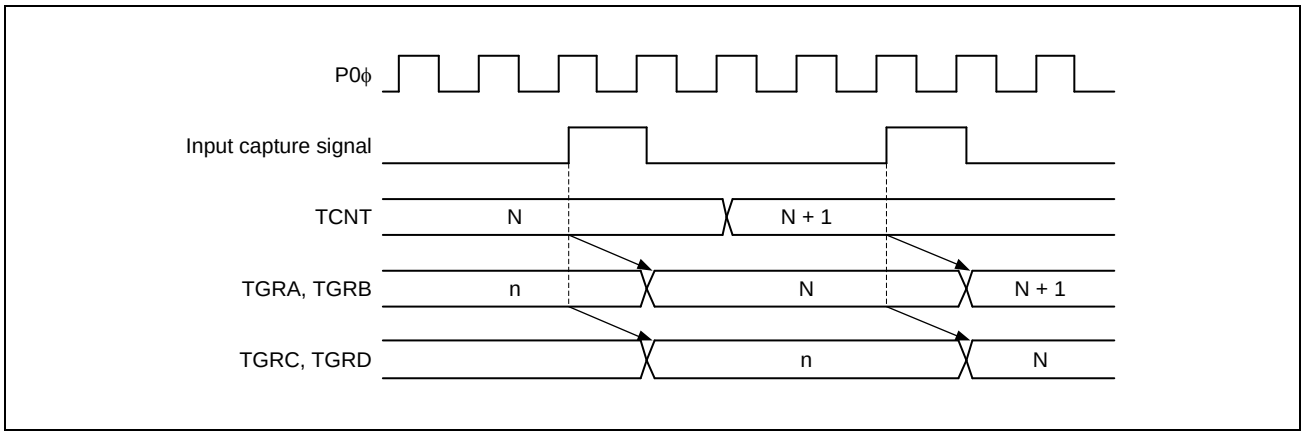



Figure 16.122 Buffer Operation Timing (Input Capture)

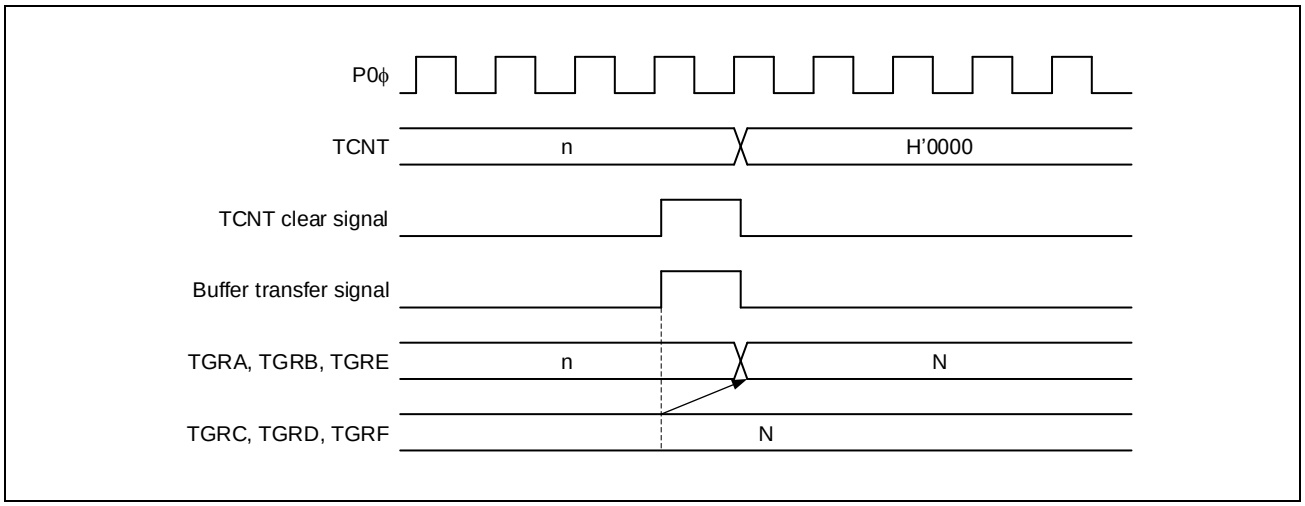



Figure 16.123 Buffer Operation Timing (When TCNT Cleared)

### (6) Buffer Transfer Timing (Complementary PWM Mode)

**Figure 16.124** to **Figure 16.126** show the buffer transfer timing in complementary PWM mode.

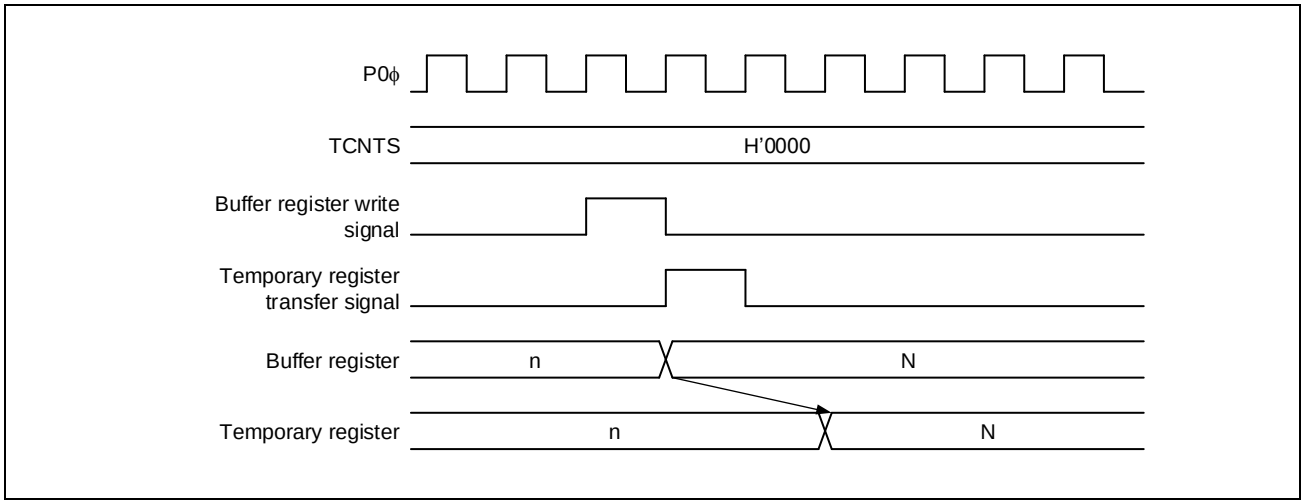



Figure 16.124 Transfer Timing from Buffer Register to Temporary Register (TCNTSA Stopped)

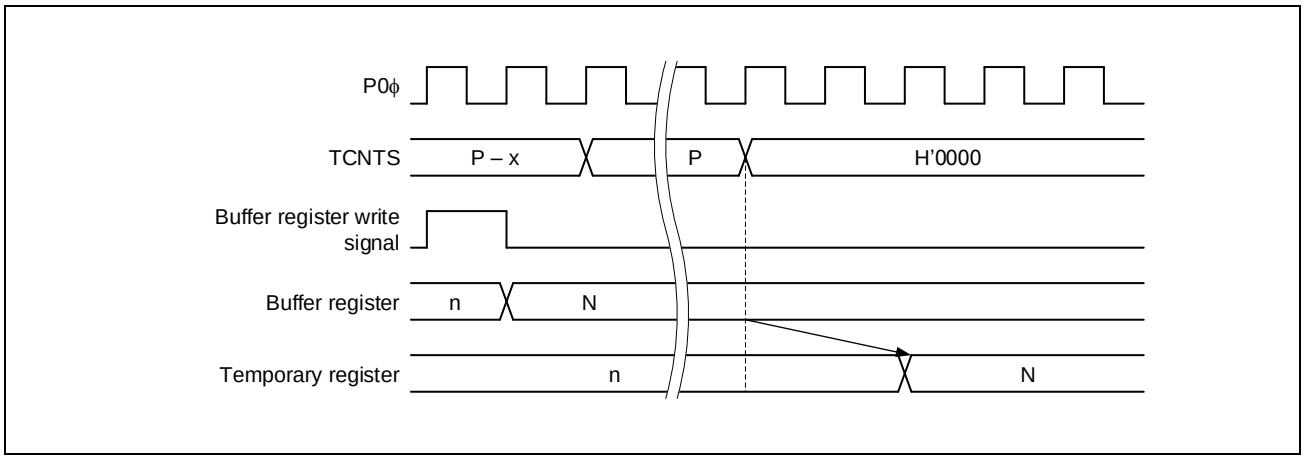



Figure 16.125 Transfer Timing from Buffer Register to Temporary Register (TCNTSA Operating)

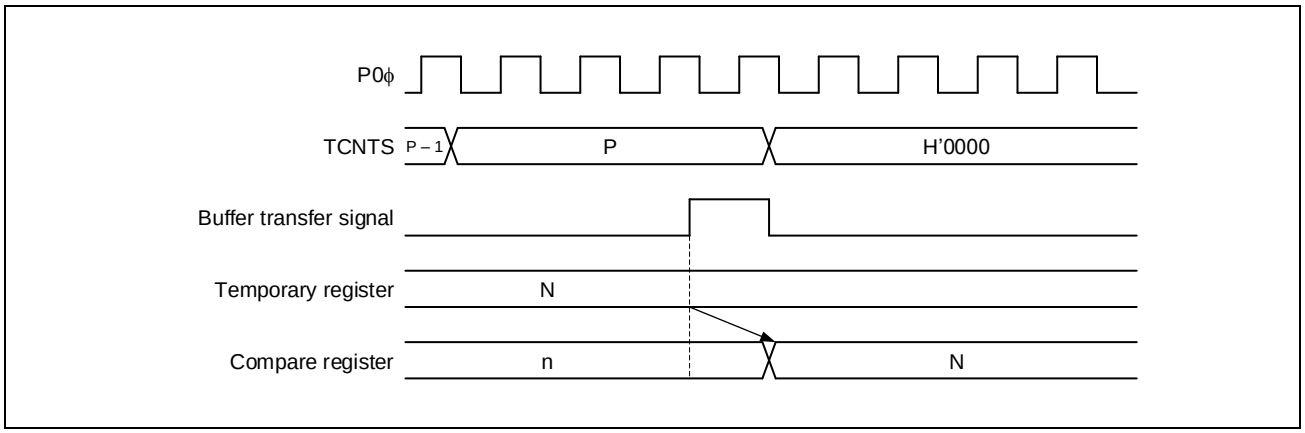



Figure 16.126 Transfer Timing from Temporary Register to Compare Register

## 16.5.2 Interrupt Signal Timing

### (1) TGI Interrupt Timing by Compare Match

**Figure 16.127** and **Figure 16.128** show the TGI interrupt request signal timing when a compare match occurs.

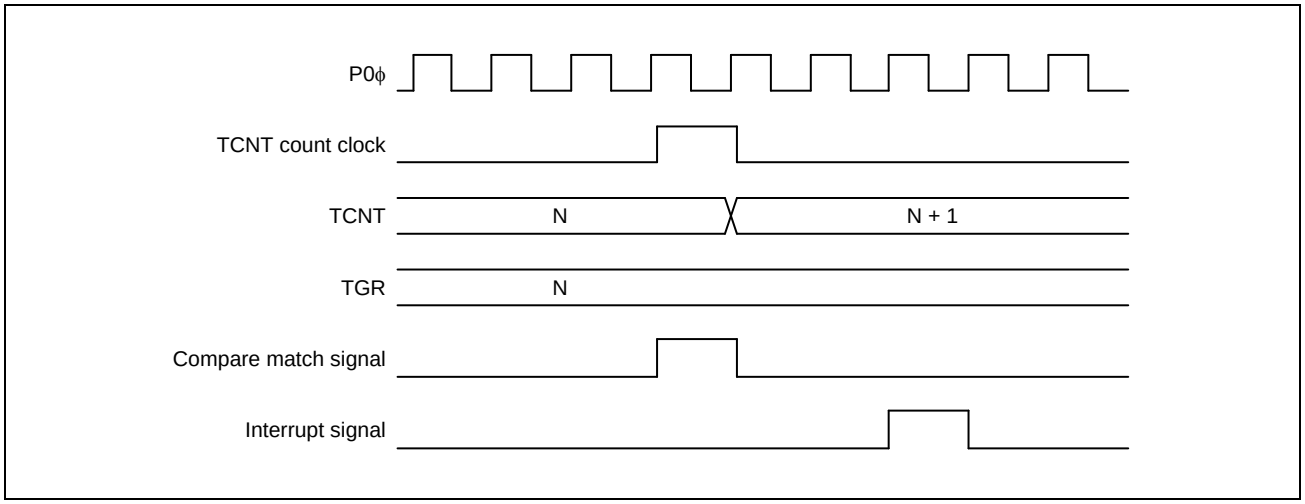



Figure 16.127 TGI Interrupt Timing (Compare Match) (MTU0 to MTU4 and MTU6 to MTU8)

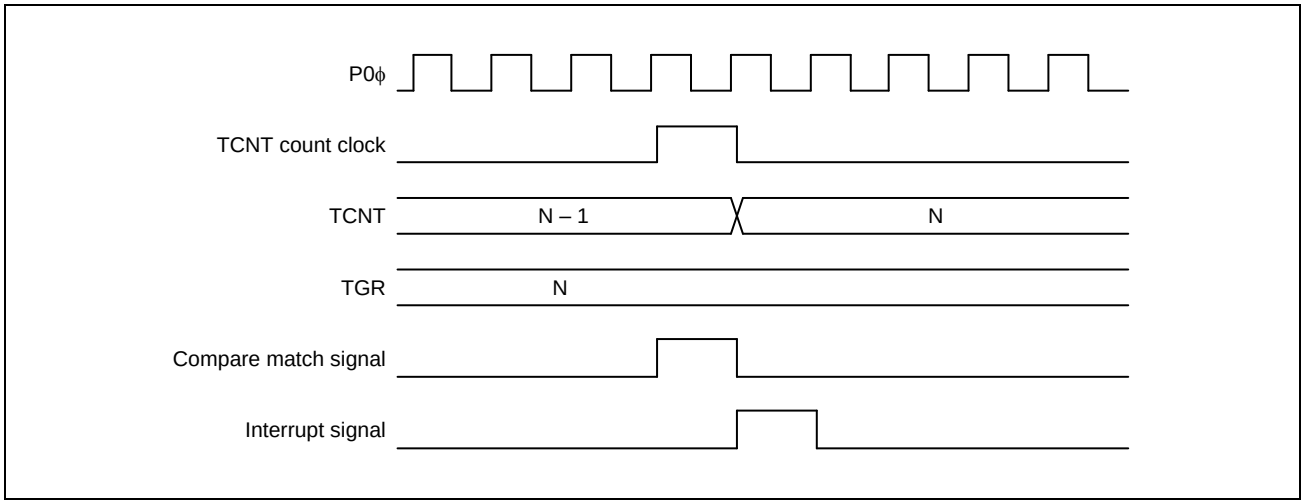



Figure 16.128 TGI Interrupt Timing (Compare Match) (MTU5)

### (2) TGI Interrupt Timing by Input Capture

**Figure 16.129** and **Figure 16.130** show the TGI interrupt request signal timing when an input capture occurs.

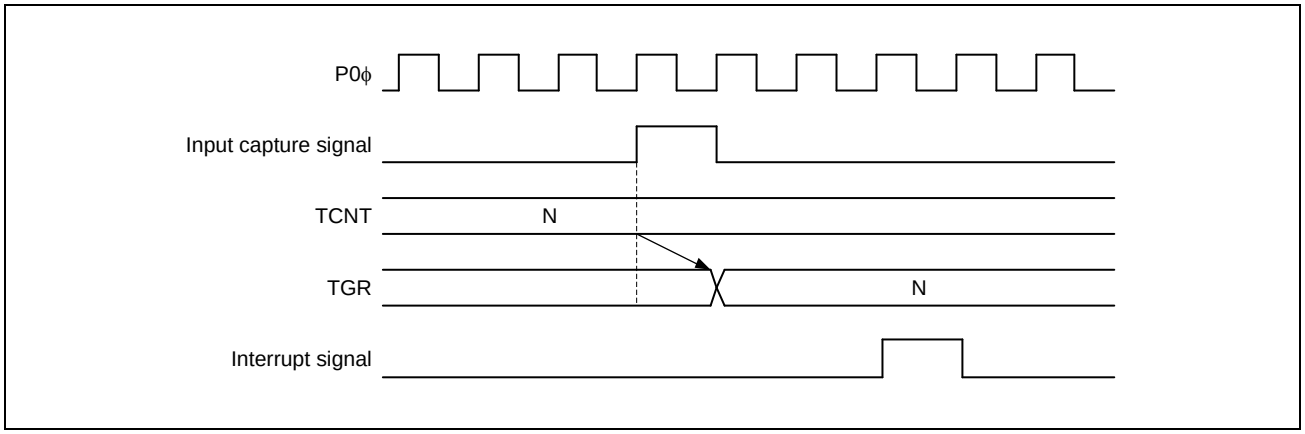



Figure 16.129 TGI Interrupt Timing (Input Capture) (MTU0 to MTU4 and MTU6 to MTU8)

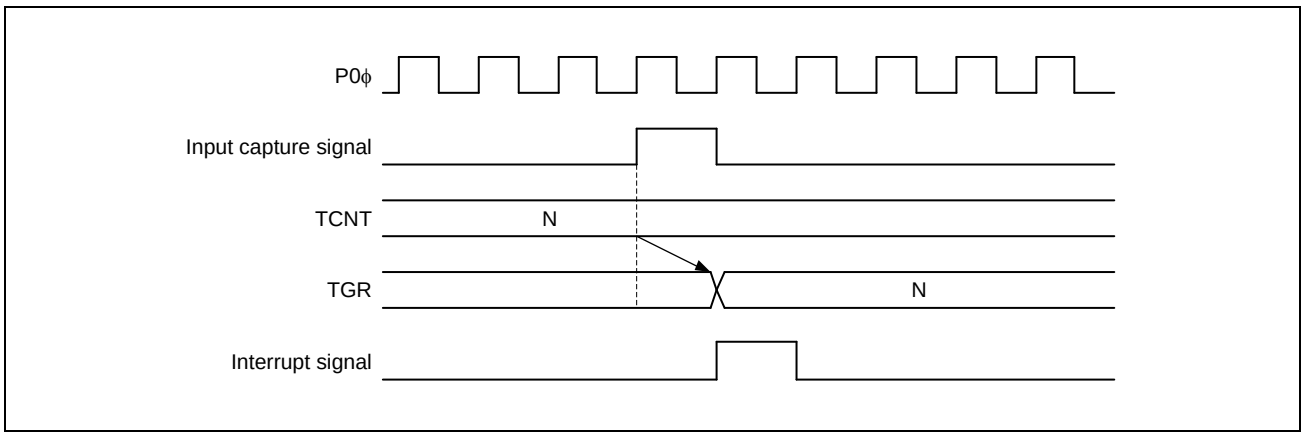



Figure 16.130 TGI Interrupt Timing (Input Capture) (MTU5)

### (3) TCFV and TCFU Interrupt Timing

**Figure 16.131** shows the TCIV interrupt request signal timing when an overflow is generated.

**Figure 16.132** shows the TCIU interrupt request signal timing when an underflow is generated.

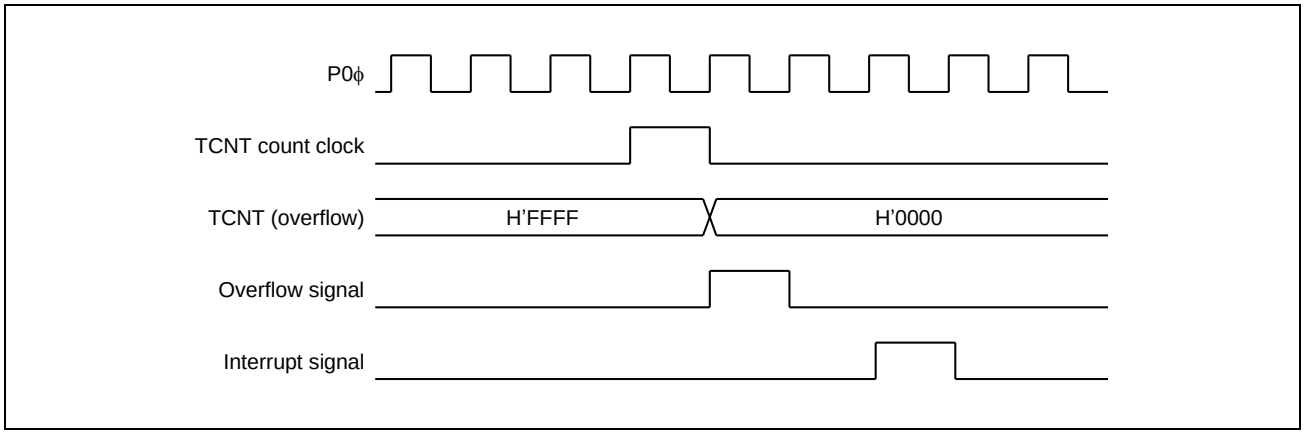



Figure 16.131 TCIV Interrupt Timing

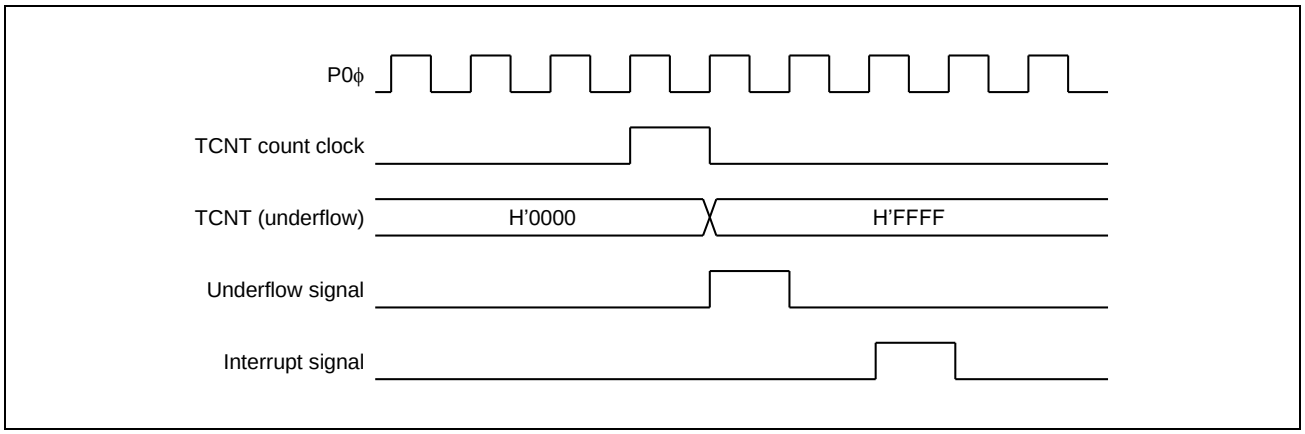


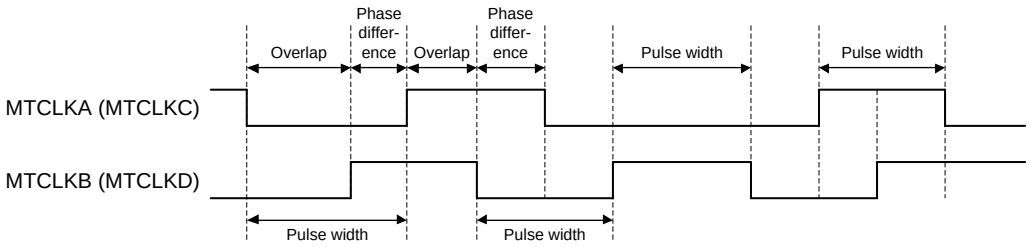

Figure 16.132 TCIU Interrupt Timing

# 16.6 Usage Note

## 16.6.1 Count Clock Restrictions

The count clock source pulse width must be at least three P0ϕ clock cycles for single-edge detection, and at least five P0ϕ clock cycles for both-edge detection. The MTU 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 three P0ϕ clock cycles, and the pulse width must be at least 5P0ϕ clock cycles. **Figure 16.133** shows the input clock conditions in phase counting mode.




**Note:** Phase difference and overlap: 3 P0ϕ clock cycles or more

**Note:** Pulse width: 5 P0ϕ clock cycles or more

Figure 16.133 Phase Difference, Overlap, and Pulse Width in Phase Counting Mode

## 16.6.2 Note on Cycle 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 TCNU updates the matched count value). Consequently, the actual counter frequency is given by the following formula:

- MTU0 to MTU4 and MTU6 to MTU8

$$f = \frac{CNTCLK}{N + 1}$$

- MTU5

$$f = \frac{CNTCLK}{N}$$

f: Counter frequency

CNTCLK: The count clock frequency set by TCR.TPSC[2:0] and TCR2.TPSC2[2:0]

N: TGR setting

### 16.6.3 Contention between TCNT Write and Clear Operations

If the counter clear signal is generated in the TCNT write cycle, TCNT clearing takes precedence and TCNT write operation is not performed.

**Figure 16.134** shows the timing in this case.

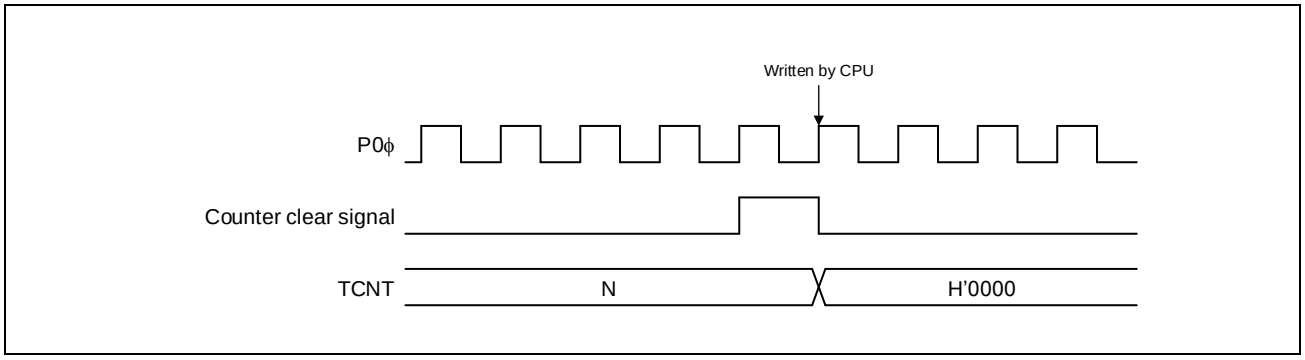



Figure 16.134 Contention between TCNT Write and Clear Operations

### 16.6.4 Contention between TCNT Write and Increment Operations

If incrementing occurs in a TCNT write cycle, TCNT write operation takes precedence and TCNT is not incremented.

**Figure 16.135** shows the timing in this case.

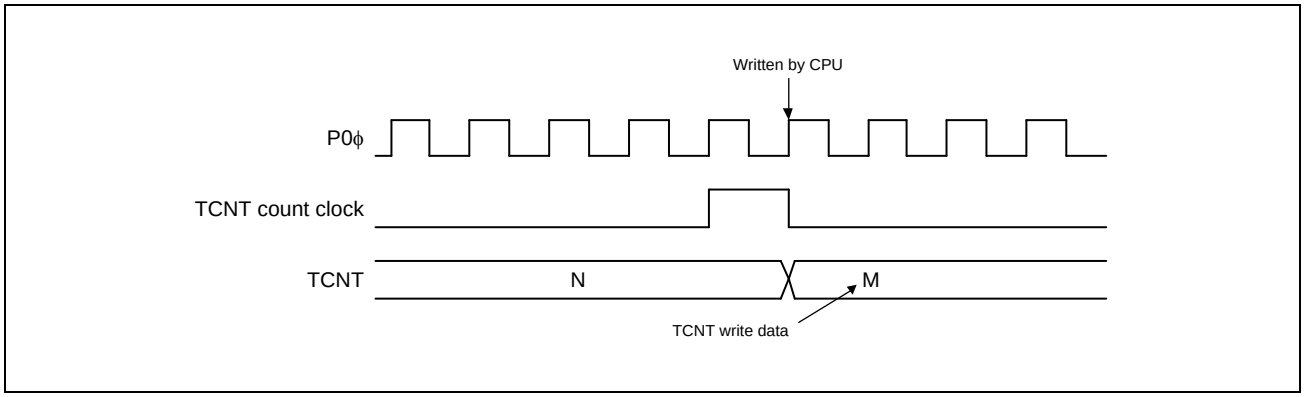



Figure 16.135 Contention between TCNT Write and Increment Operations

### 16.6.5 Contention between TGR Write Operation and Compare Match

If a compare match occurs in a TGR write cycle, TGR write operation is executed and the compare match signal is also generated.

**Figure 16.136** shows the timing in this case.

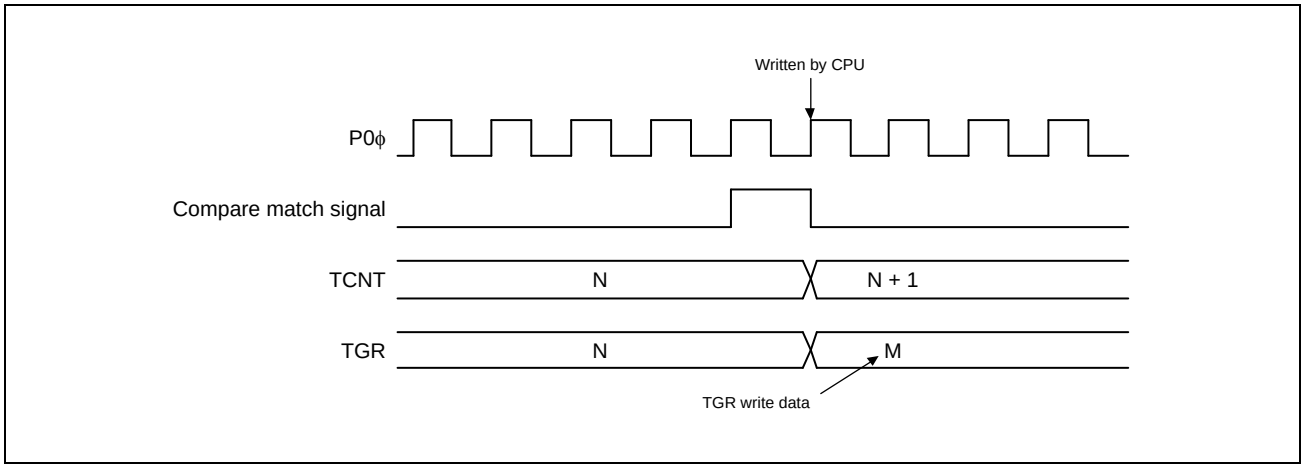



Figure 16.136 Contention between TGR Write Operation and Compare Match

### 16.6.6 Contention between Buffer Register Write Operation and Compare Match

If a compare match occurs in the T2 state in a TGR write cycle, the data before write operation is transferred to TGR by the buffer operation.

**Figure 16.137** shows the timing in this case.

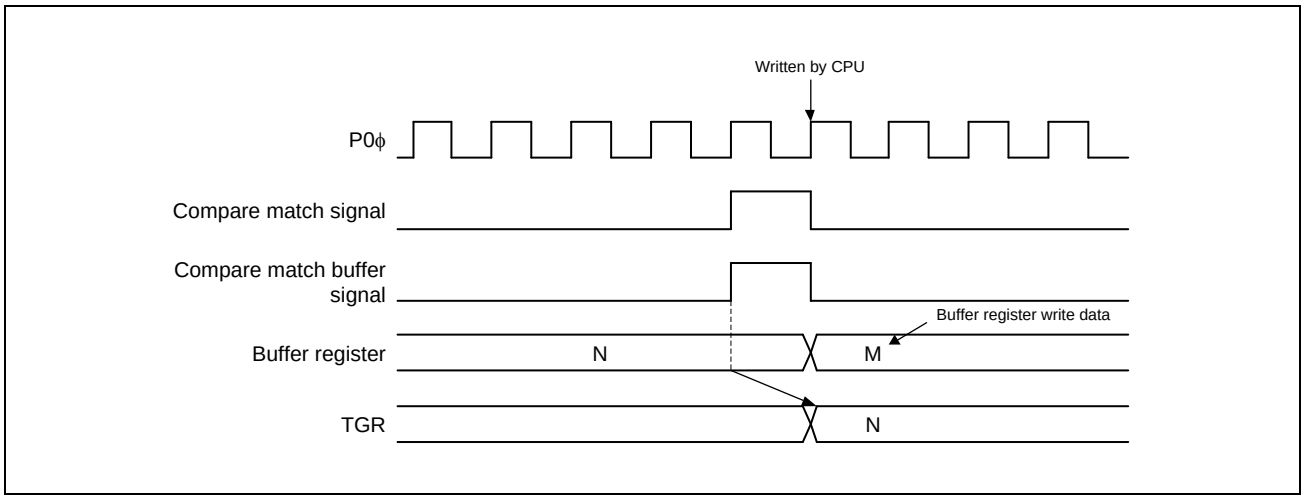



Figure 16.137 Contention between Buffer Register Write Operation and Compare Match

### 16.6.7 Contention between Buffer Register Write and TCNT Clear Operations

When the buffer transfer timing is set to the TCNT clear timing by the timer buffer transfer mode register (TBTM), if TCNT is cleared during the TGR write cycle, the data before write operation is transferred to TGR by the buffer operation.

**Figure 16.138** shows the timing in this case.

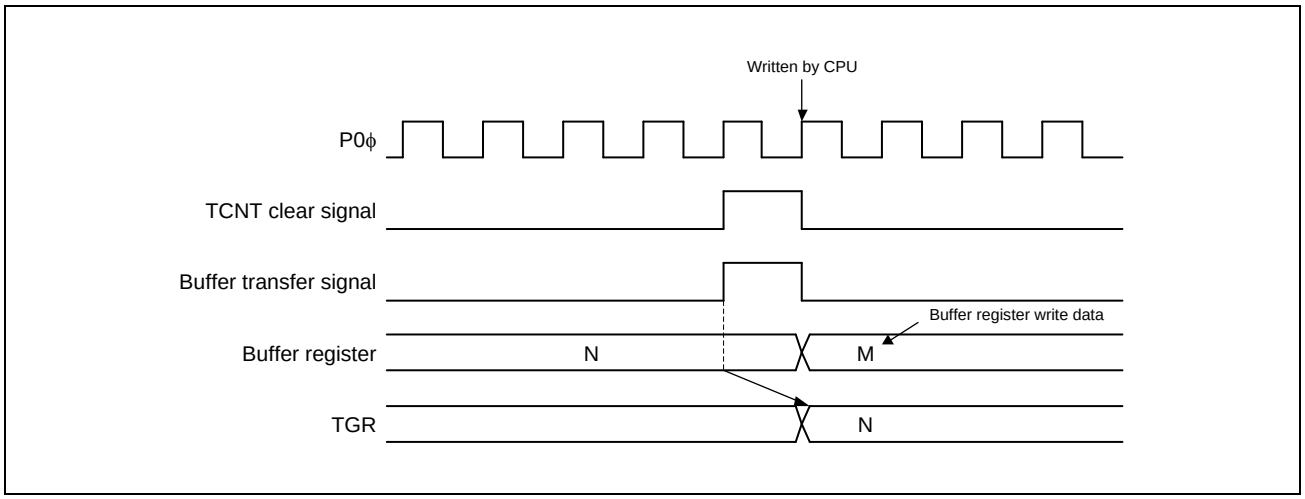



Figure 16.138 Contention between Buffer Register Write and TCNT Clear Operations

### 16.6.8 Contention between TGR Read Operation and Input Capture

If an input capture signal is generated in a TGR read cycle, the data before input capture transfer is read.

**Figure 16.139** shows the timing in this case.

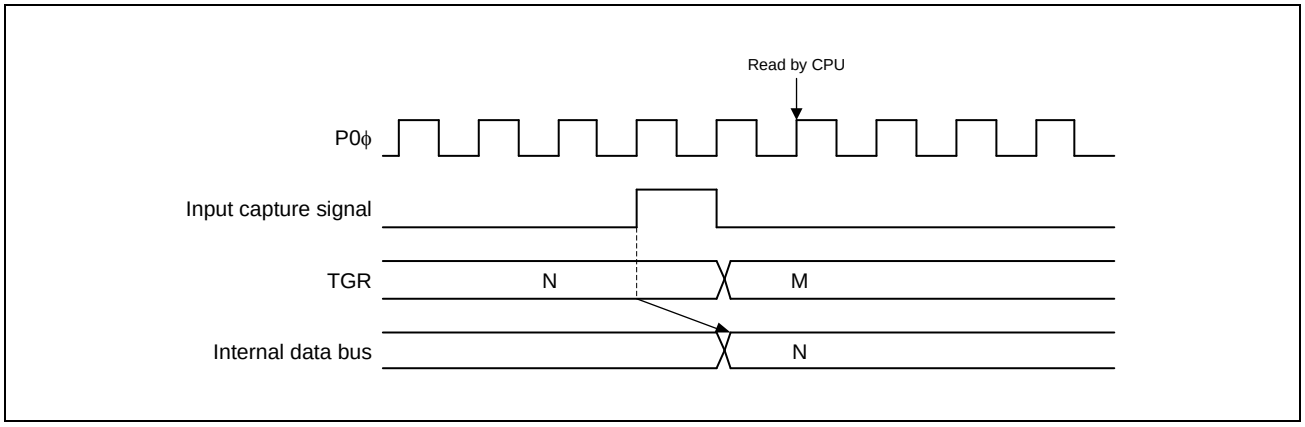



Figure 16.139 Contention between TGR Read Operation and Input Capture (MTU0 to MTU8)

### 16.6.9 Contention between TGR Write Operation and Input Capture

If an input capture signal is generated in the TGR write cycle, the input capture operation takes precedence and the TGR write operation is not performed in MTU0 to MTU4 and MTU6 to MTU8. In MTU5, the TGR write operation is performed and the input capture signal is generated.

**Figure 16.140** and **Figure 16.141** show the timing in this case.

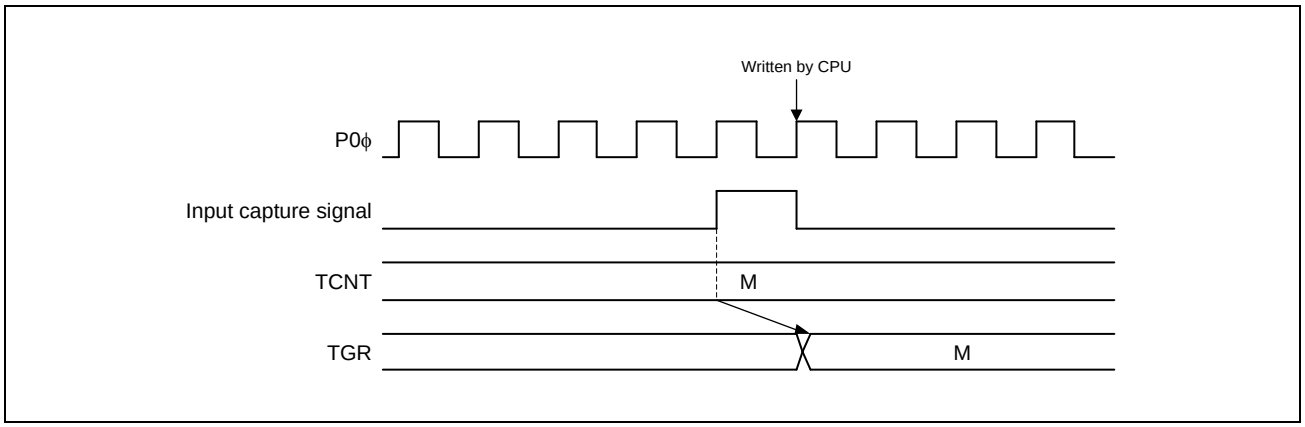



Figure 16.140 Contention between TGR Write Operation and Input Capture (MTU0 to MTU4 and MTU6 to MTU8)

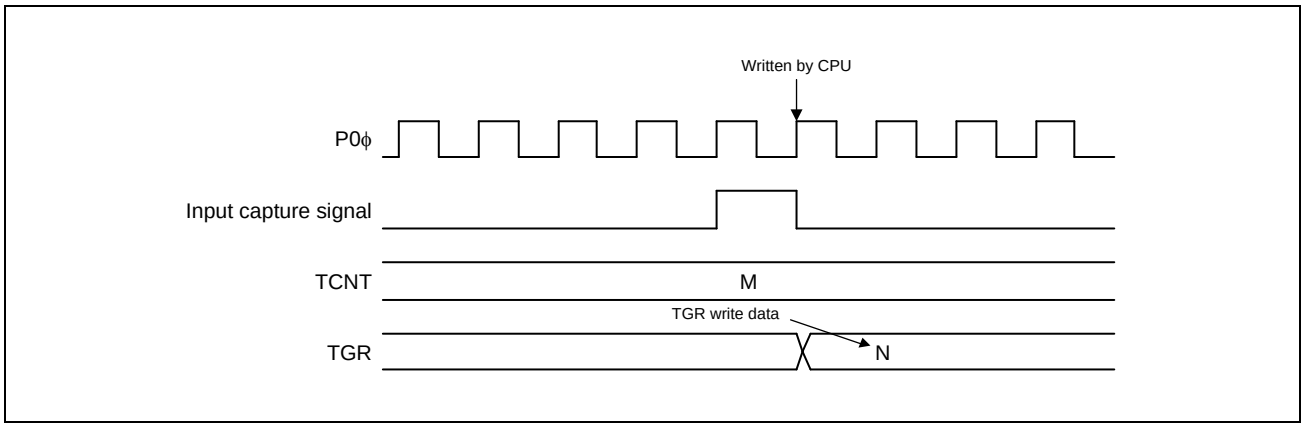



Figure 16.141 Contention between TGR Write Operation and Input Capture (MTU5)

### 16.6.10 Contention between Buffer Register Write Operation and Input Capture

If an input capture signal is generated in the buffer register write cycle, the buffer operation takes precedence and the buffer register write operation is not performed.

**Figure 16.142** shows the timing in this case.

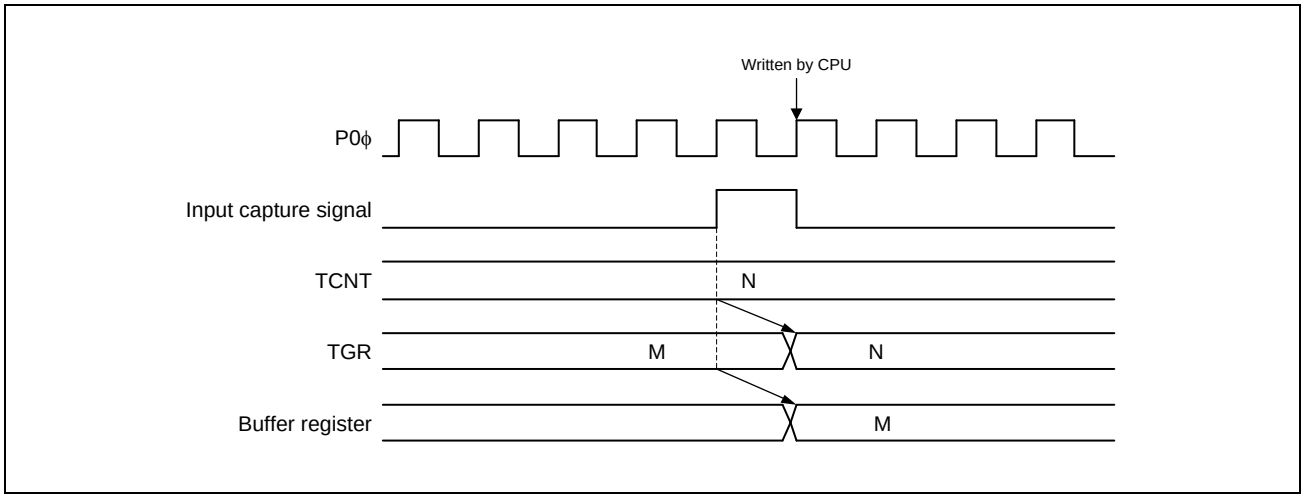



Figure 16.142 Contention between Buffer Register Write Operation and Input Capture

### 16.6.11 Contention between MTU2.TCNT Write Operation and Overflow/Underflow in Cascaded Operation

With timer counters TCNT in MTU1 and MTU2 cascaded, when a contention occurs between the counting by TCNT in MTU1 (a TCNT overflow/underflow in MTU2) and the writing to TCNT in MTU2, the TCNT write operation is performed in MTU2 and the TCNT count signal in MTU1 is disabled. In this case, if TGRA in MTU1 works as a compare match register and there is a match between the values of TGRA and TCNT in MTU1, a compare match signal is issued.

Furthermore, when the TCNT count clock in MTU1 is selected as the input capture source of MTU0, TGRA to TGRD in MTU0 work in input capture mode. In addition, when the TGRC compare match/input capture in MTU0 is selected as the input capture source of TGRB in MTU1, TGRB in MTU1 works in input capture mode.

**Figure 16.143** shows the timing in this case.

When setting the TCNT clearing function in cascaded operation, be sure to synchronize MTU1 and MTU2.

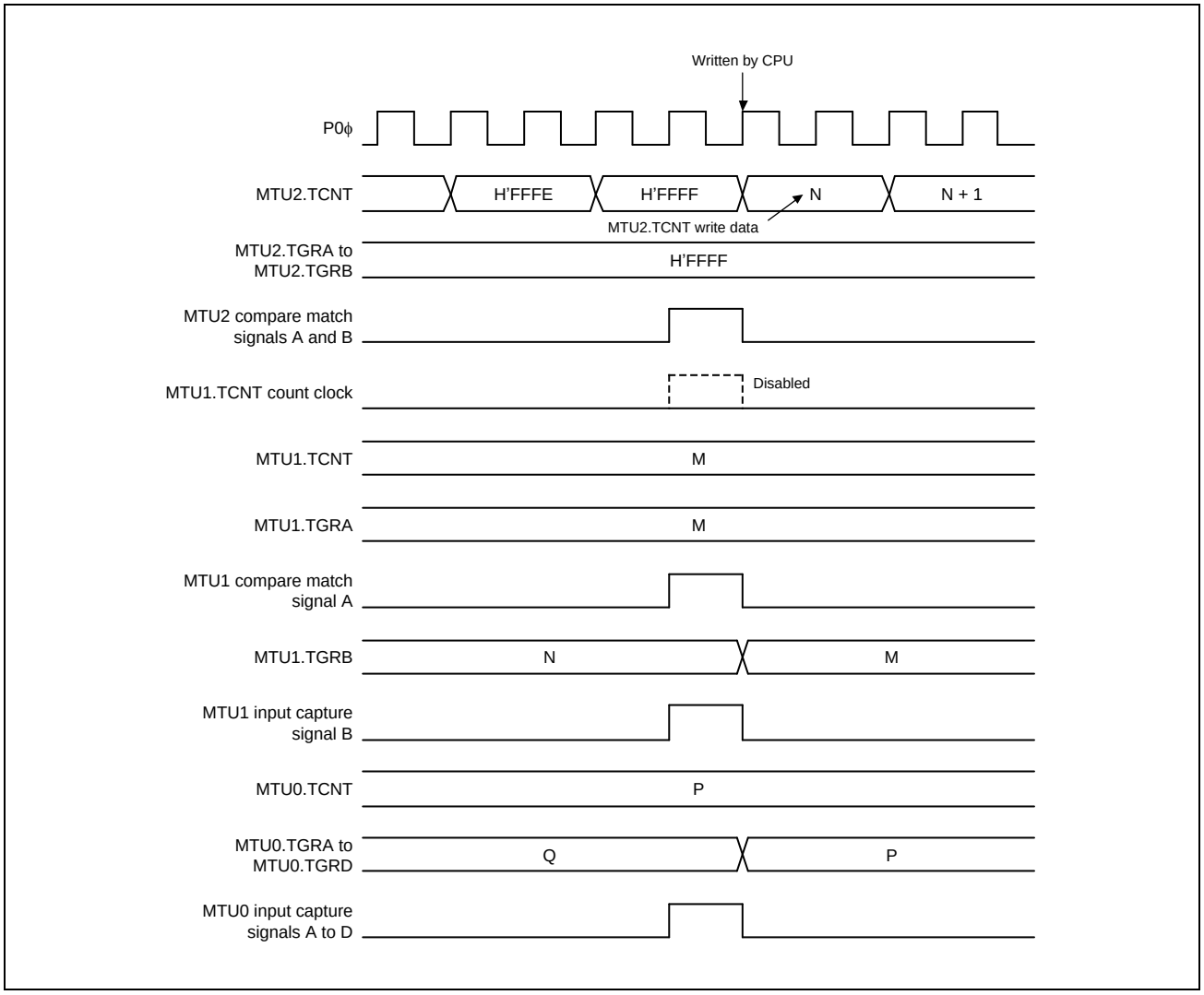



Figure 16.143 Contention between MTU2.TCNT Write Operation and Overflow/Underflow in Cascaded Operation

### 16.6.12 Counter Value When Count Operation is Stopped in Complementary PWM Mode

When counting operation in MTU3.TCNT and MTU4.TCNT (MTU6.TCNT and MTU7.TCNT) is stopped in complementary PWM mode, the MTU3.TCNT (MTU6.TCNT) value is set to the timer dead time register (TDDRA (TDDRB)) value and MTU4.TCNT (MTU7.TCNT) is set to H'0000.

When operation is restarted in complementary PWM mode, counting begins automatically from the initial setting state.

**Figure 16.144** shows this operation.

When counting begins in another operating mode, be sure to make initial settings in MTU3.TCNT and MTU4.TCNT (MTU6.TCNT and MTU7.TCNT).

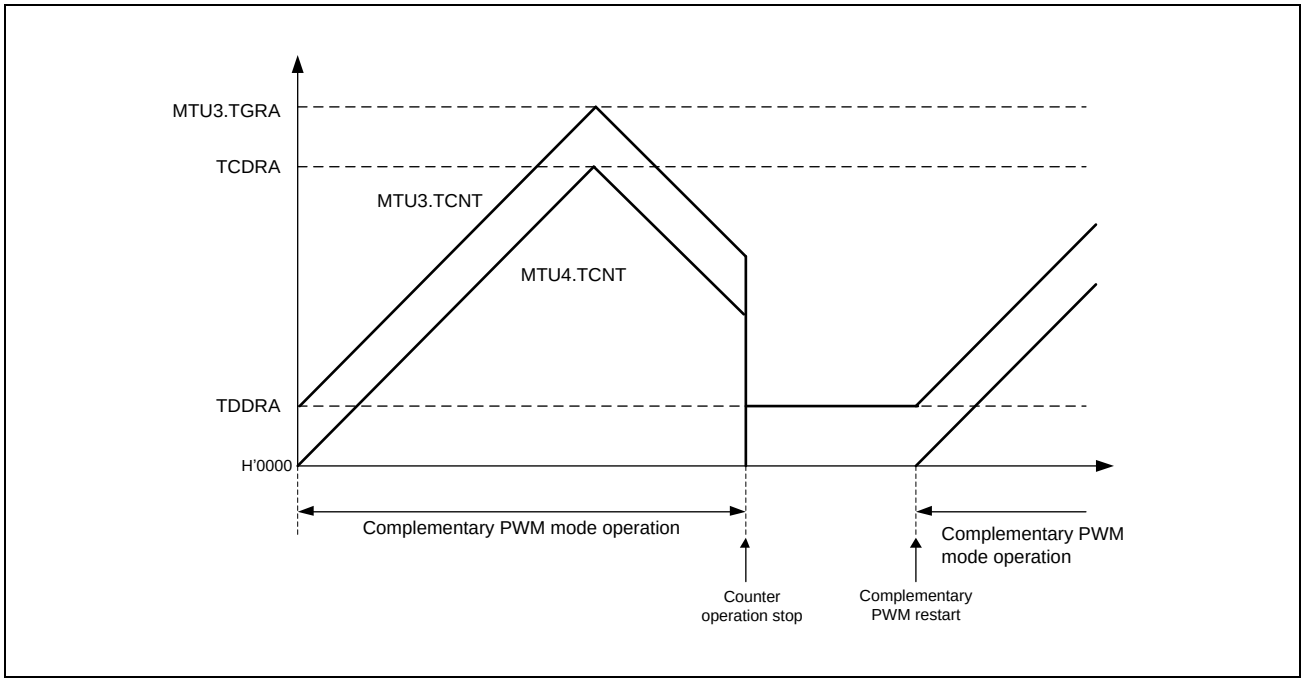



Figure 16.144 Counter Value When Stopped in Complementary PWM Mode

### 16.6.13 Buffer Operation Setting in Complementary PWM Mode

When modifying the PWM cycle set register (MTU3.TGRA or MTU6.TGRA), timer cycle data register (TCDRA or TCDRB), and duty set registers (MTU3.TGRB, MTU4.TGRA, and MTU4.TGRB (MTU6.TGRB, MTU7.TGRA, and MTU7.TGRB)) in complementary PWM mode, be sure to use buffer operation. In addition, set the BFA and BFB bits in MTU4.TMDR1 (MTU7.TMDR1) to 0. The MTIOC4C (MTIOC7C) pin cannot output waveforms if the BFA bit in MTU4.TMDR1 (MTU7.TMDR1) is set to 1. Likewise, the MTIOC4D (MTIOC7D) pin cannot output waveforms if the BFB bit in MTU4.TMDR1 (MTU7.TMDR1) is set to 1.

In complementary PWM mode, buffer operation in MTU3 and MTU4 (or MTU6 and MTU7) depends on the settings in bits BFA and BFB of MTU3.TMDR1 (MTU6.TMDR1). When the BFA bit in MTU3.TMDR1 (MTU6.TMDR1) is set to 1, MTU3.TGRC (MTU6.TGRC) functions as a buffer register for MTU3.TGRA (MTU6.TGRA). At the same time, MTU4.TGRC (MTU7.TGRC) functions as a buffer register for MTU4.TGRA (MTU7.TGRA), and TCBRA (TCBRB) functions as a buffer register for TCDRA (TCDRB).

### 16.6.14 Buffer Operation and Compare Match in Reset-Synchronized PWM Mode

When setting buffer operation in reset-synchronized PWM mode, set the BFA and BFB bits in MTU4.TMDR1 (MTU7.TMDR1) to 0. The MTIOC4C (MTIOC7C) pin cannot output waveforms if the BFA bit in MTU4.TMDR1 (MTU7.TMDR1) is set to 1. Likewise, the MTIOC4D (MTIOC7D) pin cannot output waveforms if the BFB bit in MTU4.TMDR1 (MTU7.TMDR1) is set to 1.

In reset-synchronized PWM mode, buffer operation in MTU3 and MTU4 (or MTU 6 and MTU7) depends on the settings in the BFA and BFB bits of MTU3.TMDR1 (MTU6.TMDR1). For example, if the BFA bit in MTU3.TMDR1 (MTU6.TMDR1) is set to 1, MTU3.TGRC (MTU6.TGRC) functions as a buffer register for MTU3.TGRA (MTU6.TGRA). At the same time, MTU4.TGRC (MTU7.TGRC) functions as a buffer register for MTU4.TGRA (MTU7.TGRA).

While the MTU3.TGRC (MTU6.TGRC) and MTU3.TGRD (MTU6.TGRD) are operating as buffer registers, a TGImn interrupt (m = C or D; n = 3, 4, 6, or 7) is not generated.

**Figure 16.145** shows an example of MTU3.TGR (MTU6.TGR), MTU4.TGR (MTU7.TGR), MTIOC3 (MTIOC6), and MTIOC4 (MTIOC7) operation with the BFA and BFB bits in MTU3.TMDR1 (MTU6.TMDR1) set to 1 and the BFA and BFB bits in MTU4.TMDR1 (MTU7.TMDR1) set to 0.

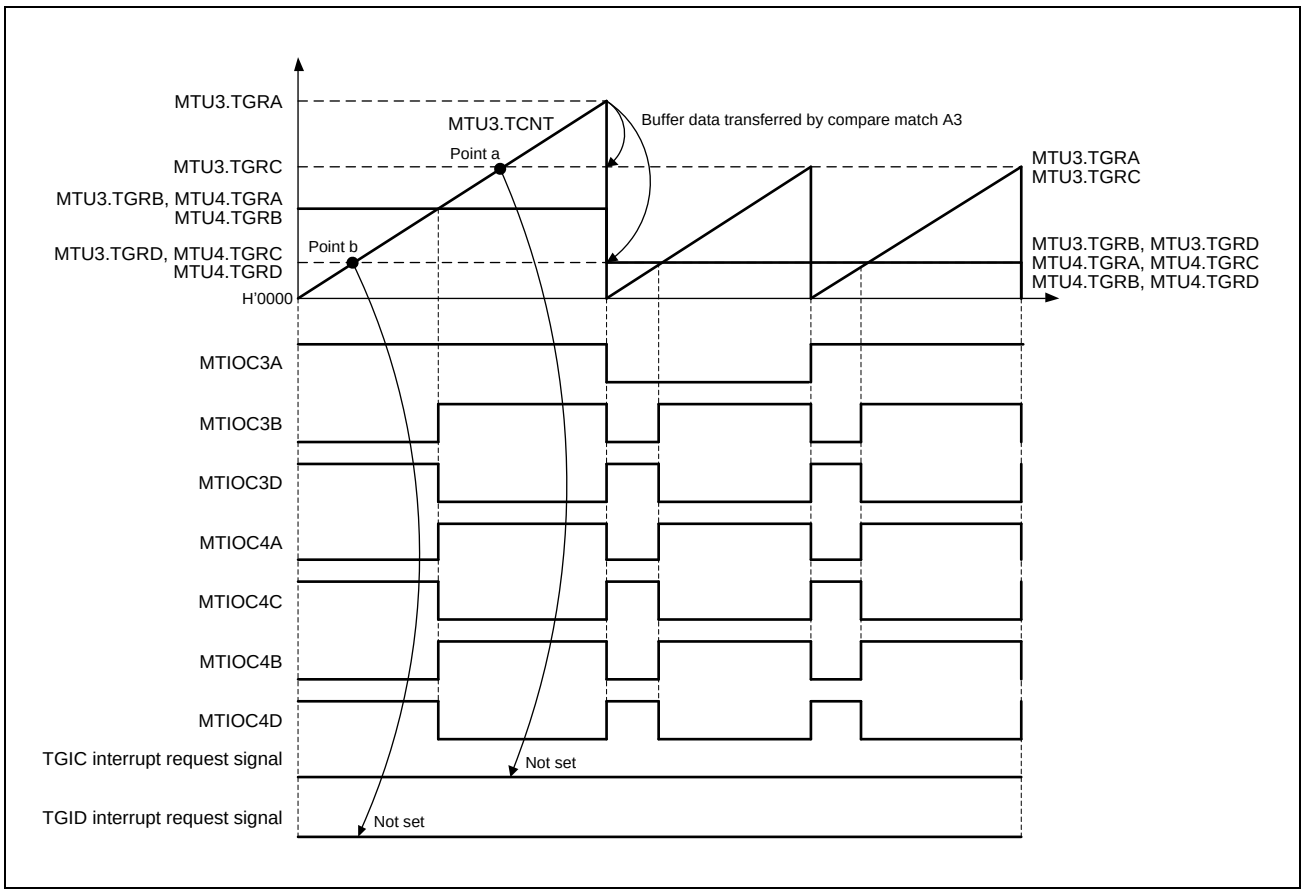



Figure 16.145 Buffer Operation and Compare Match in Reset-Synchronized PWM Mode

### 16.6.15 Overflow in Reset-Synchronized PWM Mode

After reset-synchronized PWM mode is selected, the TCNT counters in MTU3 and MTU4 (MTU6 and MTU7) start counting when the CST3 (CST6) bit of TSTRA (TSTRB) is set to 1. In this state, the TCNT count clock source and count edge in MTU4 (MTU7) are determined by the setting of TCR in MTU3 (MTU6).

In reset-synchronized PWM mode, with cycle register TGRA in MTU3 (MTU6) set to H'FFFF and the TGRA compare match in MTU3 (MTU6) selected as the counter clearing source, the TCNT counters in MTU3 and MTU4 (MTU6 and MTU7) count up to H'FFFF, then a compare match occurs with TGRA in MTU3 (MTU6), and the TCNT counters in MTU3 and MTU4 (MTU6 and MTU7) are both cleared. In this case, a TCIVn interrupt (n = 3, 4, 6, or 7) is not generated.

**Figure 16.146** shows an example of the operation in reset-synchronized PWM mode with cycle register TGRA in MTU3 (MTU6) set to H'FFFF and the TGRA compare match in MTU3 (MTU6) specified for the counter clearing source.

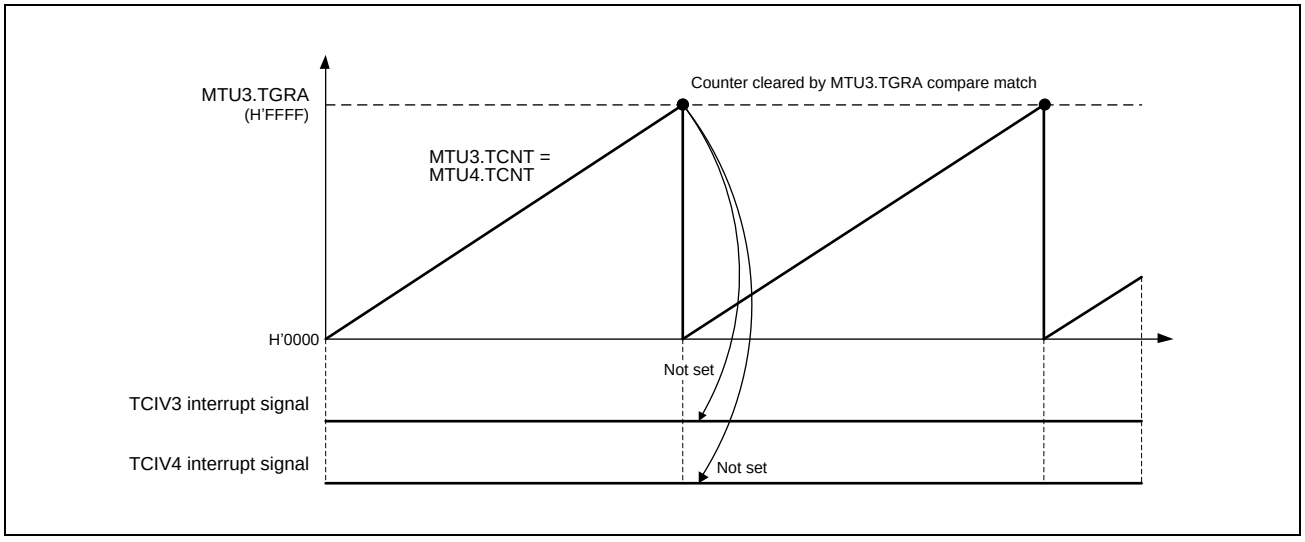



Figure 16.146 Overflow in Reset-Synchronized PWM Mode

### 16.6.16 Contention between Overflow/Underflow and Counter Clearing

If an overflow/underflow and counter clearing occur simultaneously, a TCIVn interrupt (n = 0 to 4 or 6 to 8) nor a TCIUn interrupt (n = 1 or 2) is not generated and TCNT clearing takes precedence.

**Figure 16.147** shows the operation timing when a TGR compare match is specified as the clearing source and TGR is set to H'FFFF.

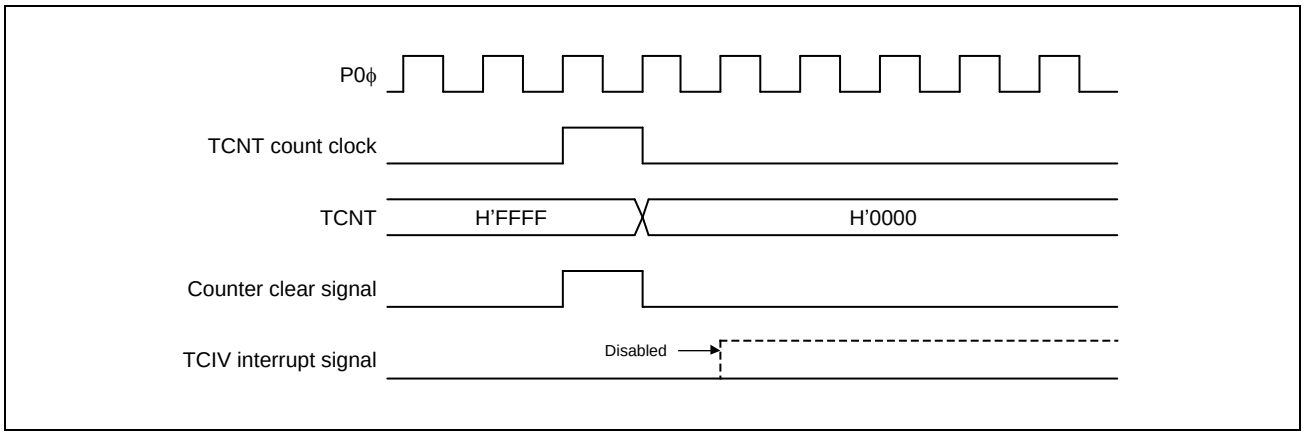



Figure 16.147 Contention between Overflow and Counter Clearing

### 16.6.17 Contention between TCNT Write Operation and Overflow/Underflow

If TCNT counts up or down in a TCNT write cycle and an overflow or an underflow occurs, the TCNT write operation takes precedence. A TCIVn interrupt (n = 0 to 4 or 6 to 8) nor a TCIUn interrupt (n = 1 or 2) is not generated.

**Figure 16.148** shows the operation timing when there is contention between TCNT write operation and overflow.

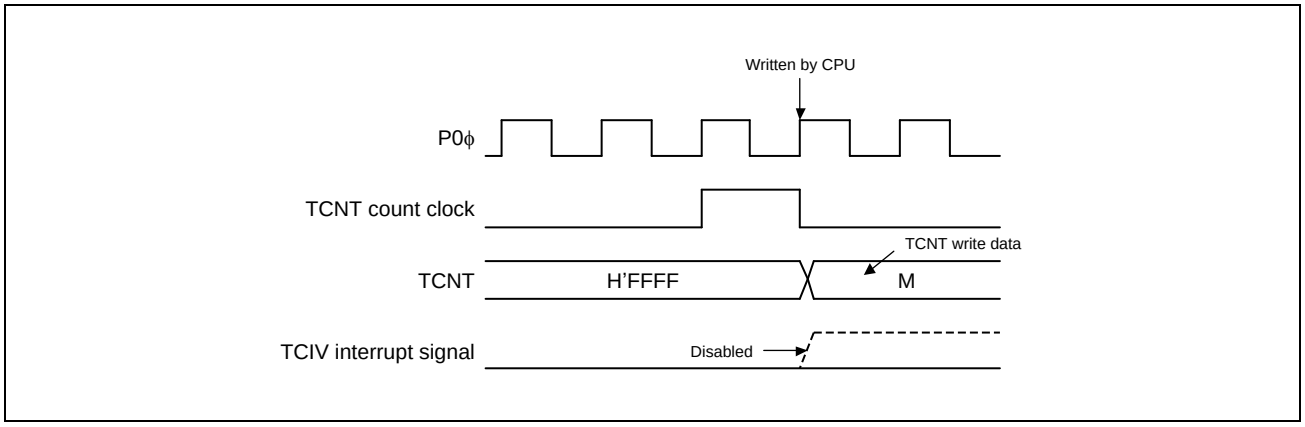



Figure 16.148 Contention between TCNT Write Operation and Overflow

### 16.6.18 Note on Transition from Normal Mode or PWM Mode 1 to Reset-Synchronized PWM Mode

When making a transition from normal mode or PWM mode 1 to reset-synchronized PWM mode in MTU3 and MTU4 (or MTU6 and MTU7), if the counter is stopped while the output pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4C, MTIOC4B, MTIOC4D, MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7C, MTIOC7B, and MTIOC7D) are held at a high level and then operation is started after a transition to reset-synchronized PWM mode, the initial pin output will not be correct.

When making a transition from normal mode to reset-synchronized PWM mode, write H'11 to MTU3.TIORH, MTU3.TIORL, MTU4.TIORH, and MTU4.TIORL (MTU6.TIORH, MTU6.TIORL, MTU7.TIORH, and MTU7.TIORL) to initialize the output pin state to a low level, then set the registers to the initial value (H'00) before making the mode transition.

When making a transition from PWM mode 1 to reset-synchronized PWM mode, switch to normal mode, initialize the output pin state to a low level, and then set the registers to the initial value (H'00) before making the transition to reset-synchronized PWM mode.

### 16.6.19 Output Level in Complementary PWM Mode and Reset-Synchronized PWM Mode

When MTU3 and MTU4 (or MTU6 and MTU7) are in complementary PWM mode or reset-synchronized PWM mode, the PWM waveform output level is determined by the OLSP and OLSN bits in the timer output control register (TOCR1A or TOCR1B). In complementary PWM mode or reset-synchronized PWM mode, TIOR should be set to H'00. The output level in negative phase when the TDER bit in TDERA (TDERB) is set to 0 in complementary PWM mode (the dead time is not generated) does not depend on the setting of the OLSN bit in TOCR1A (TOCR1B). It is equivalent to the inverted level of positive phase output based on the setting of the OLSP bit in TOCR1A (TOCR1B).

### 16.6.20 Simultaneous Input Capture in MTU1.TCNT and MTU2.TCNT in Cascade Connection

When timer counters 1 and 2 (MTU1.TCNT and MTU2.TCNT) operate as a 32-bit counter in cascade connection, the cascaded counter value cannot be captured successfully in some cases even if input-capture input is simultaneously done to MTIOC1A and MTIOC2A or to MTIOC1B and MTIOC2B. This is because the input timing of MTIOC1A and MTIOC2A or to MTIOC1B and MTIOC2B may not be the same when external input-capture signals input into MTU1.TCNT and MTU2.TCNT are taken in synchronization with the internal clock.

For example, MTU1.TCNT (the counter for upper 16 bits) does not capture the count-up value by an overflow from MTU2.TCNT (the counter for lower 16 bits) but captures the count value before the up-counting. In this case, the values of MTU1.TCNT = H'FFF1 and MTU2.TCNT = H'0000 should be transferred to MTU1.TGRA and MTU2.TGRA or to MTU1.TGRB and MTU2.TGRB, but the values of MTU1.TCNT = H'FFF0 and MTU2.TCNT = H'0000 are erroneously transferred.

The MTU has a function that allows simultaneous capture of MTU1.TCNT and MTU2.TCNT with a single input capture input. This function can be used to read the 32-bit counter such that MTU1.TCNT and MTU2.TCNT are captured at the same time. For details, see **Section 16.2.11, Timer Input Capture Control Register (TICCR)**.

### 16.6.21 Interrupt-Skipping Function 2

When interrupt-skipping function 2 is in use and the difference between the values in MTU4.TADCORA and MTU4.TADCORB is small, correct counting of the number skipped may not be possible, in which case requests for A/D conversion will not be generated with the expected timing. The conditions listed below thus apply to these settings.

For MTU6 and MTU7, the same conditions apply to the settings of MTU7.TADCORA and MTU7.TADCORB.

(1) When the number skipped is zero for skipping function 2

- The difference between the values in MTU4.TADCORA and MTU4.TADCORB must be at least four.
- The interval of comparison for MTU4.TADCORA must be at least four cycles of P0ϕ clock (the updated value of MTU4.TADCORA is set to the previous value plus or minus at least four).
- The interval of comparison for MTU4.TADCORB must be at least four cycles of P0ϕ clock (the updated value of MTU4.TADCORB is set to the previous value plus or minus at least four).

(2) When the number skipped is one or more for skipping function 2

- The difference between the values in MTU4.TADCORA and MTU4.TADCORB must be at least two.
- The interval of comparison for MTU4.TADCORB must be at least two cycles of P0ϕ (the updated value of MTU4.TADCORB is set to the previous value plus or minus at least two).

### 16.6.22 Notes When Complementary PWM Mode Output Protection Function is Not Used

The complementary PWM mode output protection function is initially enabled. For details, refer to **Section 17, Port Output Enable 3 (POE3)**.

### 16.6.23 Notes Regarding Timer Counter (MTU5.TCNT) and Timer General Register (MTU5.TGR)

Do not set TGRj (j = U, V, or W) in MTU5 to the value of the corresponding TCNTj in MTU5 plus one while counting by TCNTj in MTU5 is stopped. If such a setting is made, a compare match will be generated even though counting is stopped.

In this case, if the value of the compare match enable bit (TGIE5j bit of TIER in MTU5) is 1 (enabled), a compare match interrupt will also be generated. If the value of the timer compare match clear register is 1 (enabled), the TCNTj counter in MTU5 is automatically cleared to H’0000 when the compare match is generated, regardless of whether compare match interrupts are enabled or disabled.

### 16.6.24 Notes to Prevent Malfunctions in Synchronous Clearing for Complementary PWM Mode

If control of the output waveform is enabled (the WRE bit of TWCRA or TWCRB is 1) at the time of synchronous counter clearing in complementary PWM mode, satisfaction of either condition 1 or 2 below has the following effects.

- Dead time on the PWM output pins is shortened (or disappears).
- The active level is output on the negative phase PWM output pins beyond the period for active-level output.

Condition 1:

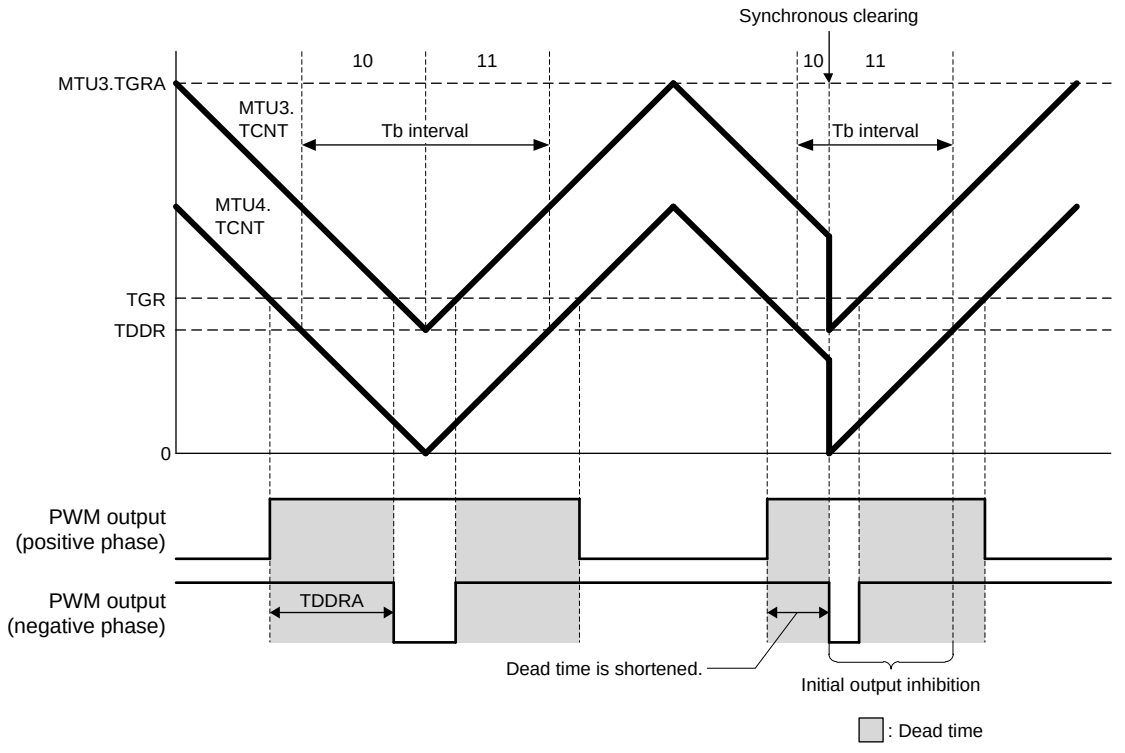
In portion (10) of the initial output inhibition period in **Figure 16.149**, synchronous clearing occurs within the dead-time period for PWM output.

Condition 2:

In portions (10) and (11) of the initial output inhibition period in **Figure 16.150**, synchronous clearing occurs when any condition from among TGRB in MTU3 (MTU6) ≤ TDDR (TDDRB), TGRA in MTU4 (MTU7) ≤ TDDR (TDDRB), and TGRB in MTU4 (MTU7) ≤ TDDR (TDDRB) is satisfied.

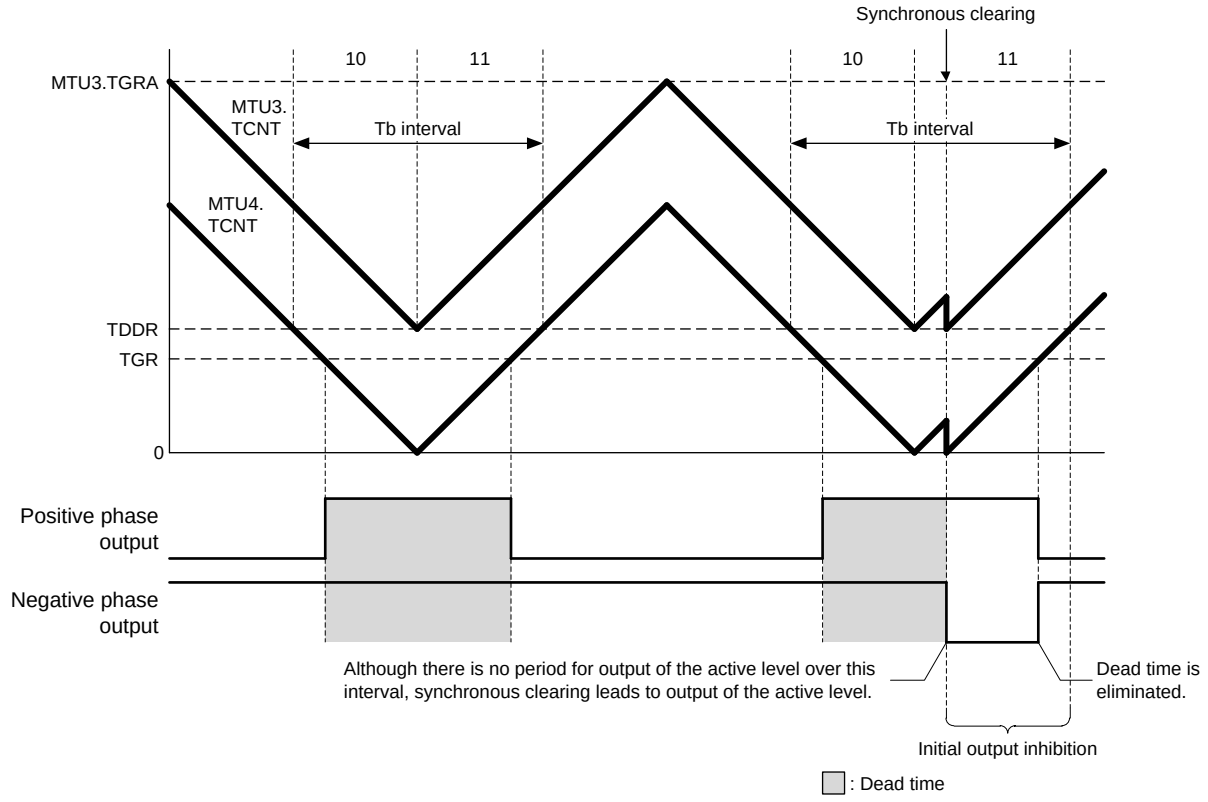
The following method avoids the above phenomena.

Ensure that synchronous clearing proceeds with the value of each comparison register (TGRB in MTU3 (MTU6), TGRA in MTU4 (MTU7), and TGRB in MTU4 (MTU7)) set to at least double the value of the TDDRA register (TDDRB register).




**Note:** PWM output is active low.

Figure 16.149 Example of Synchronous Clearing (When Condition 1 Applies)




**Note:** PWM output is active low.

Figure 16.150 Example of Synchronous Clearing (When Condition 2 Applies)

### 16.6.25 Continuous Output of Interrupt Signal in Response to a Compare Match

When the TGR register is set to H’0000, the P0ϕ/1 clock is set as the count clock, and compare match is set as the trigger for clearing of the count clock, the value of the TCNT counter remains H’0000, and the interrupt signal will be output continuously (i.e. its level will be flat) rather than output over a single cycle. Consequently, interrupts will not be detected in response to second and subsequent compare matches.

**Figure 16.151** shows the timing for continuous output of the interrupt signal in response to a compare match.

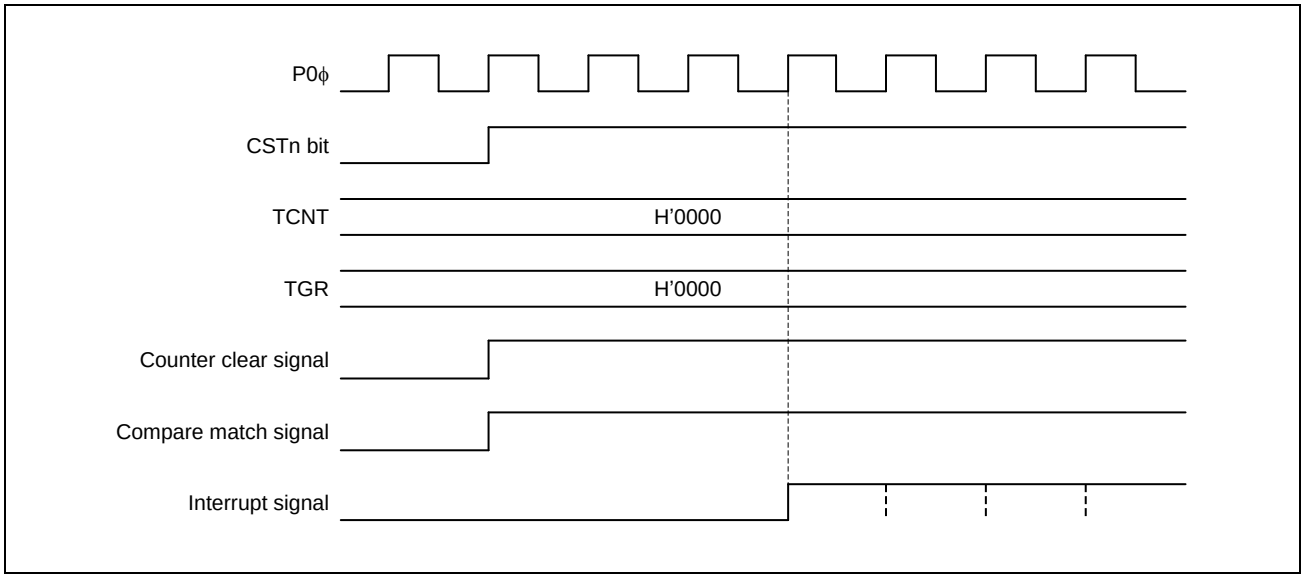
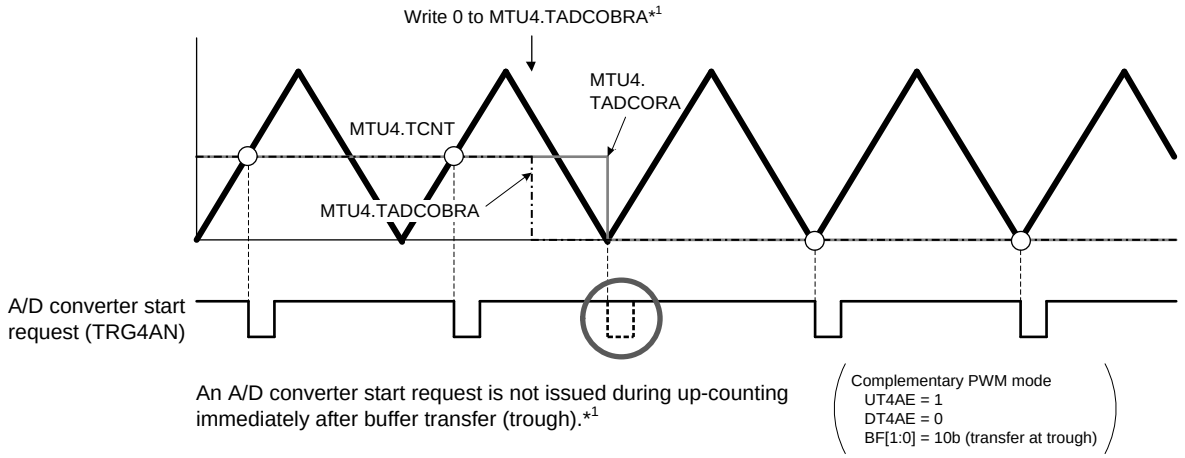



Figure 16.151 Continuous Output of Interrupt Signal in Response to a Compare Match

### 16.6.26 Notes on A/D Converter Start Request Delaying Function in Complementary PWM Mode

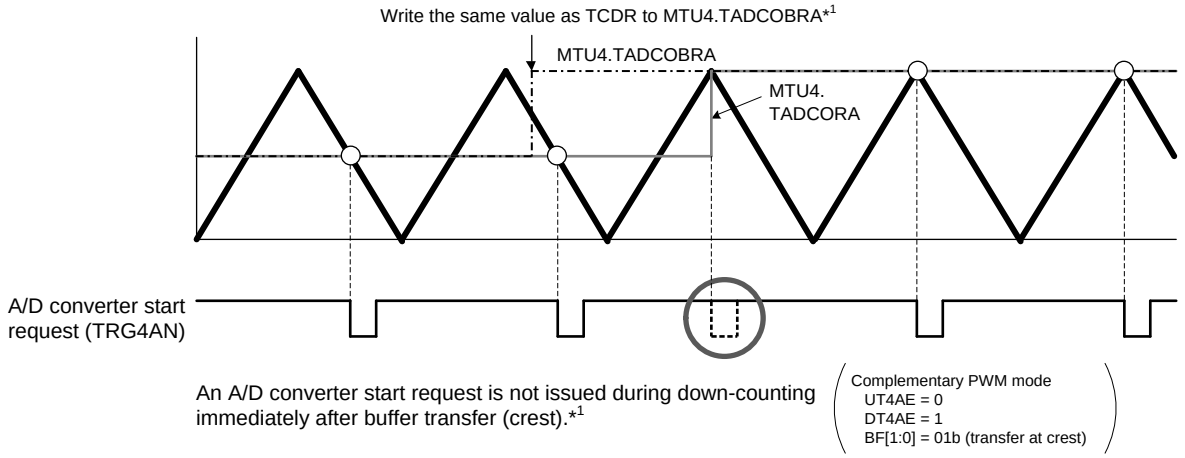
- When the TADCOBRA or TADCOBRB register in MTU4 (MTU7) is set to 0, the UT4AE or UT4BE (UT7AE or UT7BE) bit of TADCR in MTU4 (MTU7) is set to 1, and buffer transfer is performed at the trough of the value of the TCNT counter in MTU4 (MTU7), an A/D converter start request is not issued during up-counting immediately after the transfer (**Figure 16.152**).
- When the TADCOBRA or TADCOBRB register in MTU4 (MTU7) is set to the same value as that in the TCDR register, the DT4AE or DT4BE (DT7AE or DT7BE) bit of TADCR in MTU4 (MTU7) is set to 1, and buffer transfer is performed at the crest of the value of the TCNT counter in MTU4 (MTU7), an A/D converter start request is not issued during down-counting immediately after the transfer (**Figure 16.153**).
- When using A/D converter start requests in coordination with the interrupt skipping function, set up the TADCORA or TADCORB register in MTU4 (MTU7) so that the setting satisfies the condition "2 ≤ TADCORA or TADCORB in MTUn ≤ TCDR - 2" (n = 4 or 7).




**Note:** UT4AE, DT4AE, BF[1:0]: Bits in TADCR

Note 1. An A/D converter start request is issued when TCDR − 1 ≥ MTU4.TADCOBRA/TADCOBRB ≥ 1 is written.

Figure 16.152 A/D Converter Start Request when TADCOBRA is Set to 0 in MTU4




**Note:** UT4AE, DT4AE, BF[1:0]: Bits in TADCR

Note 1. An A/D converter start request is issued when TCDR −1 ≥ MTU4.TADCOBRA/TADCOBRB ≥ 1 is written.

Figure 16.153 A/D Converter Start Request when TADCOBRA is Set to the Same Value as TCDR in MTU4

# 16.7 MTU Output Pin Initialization

## 16.7.1 Operating Modes

The MTU has the following six operating modes. Waveforms can be output in any of these modes.

- Normal mode (MTU0 to MTU4 and MTU6 to MTU8)
- PWM mode 1 (MTU0 to MTU4, MTU6, and MTU)
- PWM mode 2 (MTU0 to MTU2)
- Phase counting modes 1 to 5 (MTU1 and MTU2)
- Complementary PWM mode (MTU3, MTU4, MTU6, and MTU7)
- Reset-synchronized PWM mode (MTU3, MTU4, MTU6, and MTU7)

This section describes how to initialize the MTU output pins in each of these modes.

## 16.7.2 Operation in Case of Re-Setting Due to Error during Operation

If an error occurs during MTU operation, MTU output should be cut off by the system. The output can be cut off by allowing non-active level output from the pins by setting the pins as general output ports using the port mode registers (PMn), port registers (Pn), and port mode control registers (PMCn) of the I/O ports. MTU output can be disabled through TIOR settings. Complementary PWM output (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4B, MTIOC4C, MTIOC4D, MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7B, MTIOC7C, and MTIOC7D) should be specified through TOERA and TOERB settings. For PWM output pins, output can also be cut by hardware, using port output enable 3 (POE3). The pin initialization procedures for re-setting due to an error during operation and the procedures for restarting in a different mode after re-setting are described below.

The MTU has six operating modes, as stated above. There are thus 36 mode transition combinations, but some transitions are not available with certain channel and mode combinations. Available mode transition combinations are shown in **Table 16.80**.

Table 16.80 Mode Transition Combinations

| | 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) | Not available | Not available |
| PCM | (17) | (18) | (19) | (20) | Not available | Not available |
| CPWM | (21) | (22) | Not available | Not available | (23) (24) | (25) |
| RPWM | (26) | (27) | Not available | Not available | (28) | (29) |

**Note:** Normal: Normal mode
PWM1: PWM mode 1
PWM2: PWM mode 2
PCM: Phase counting modes 1 to 5
CPWM: Complementary PWM mode
RPWM: Reset-synchronized PWM mode

### 16.7.3 Overview of Initialization Procedures and Mode Transitions in Case of Error during Operation

- When making a transition to a mode (Normal, PWM1, PWM2, or PCM) in which the pin output level is selected by the timer I/O control register (TIOR) setting, initialize the pins by means of TIOR setting.
- In PWM mode 1, waveforms are not output to the MTIOCnB and MTIOCnD pins (n = 3, 4, 6, or 7). If the MTIOCnB or MTIOCnD output is selected as the function of the pins, they enter high-impedance state. To output a specified level, make necessary settings for general output ports in the port mode registers (PMn), port registers (Pn), and port mode control registers (PMCn) of the I/O ports.
- In PWM mode 2, waveforms are not output to the pins corresponding to the TGR registers used as cycle registers. If the MTIOCnm output (n = 0 to 2 and m = A to D) is selected as the function of the pins, they enter high-impedance state. To output a specified level, make necessary settings for general output ports in the port mode registers (PMn), port registers (Pn), and port mode control registers (PMCn) of the I/O ports.
- In normal mode or PWM mode 2, if TGRC and TGRD operate as buffer registers, waveforms are not output to the corresponding MTIOCnC and MTIOCnD pins (n = 0, 3, 4, 6, or 7). If the MTIOCnC or MTIOCnD output is selected as the function of the pins, they enter high-impedance state. To output a specified level, make necessary settings for general output ports in the port mode registers (PMn), port registers (Pn), and port mode control registers (PMCn) of the I/O ports.
- In PWM mode 1, if either TGRC or TGRD operates as a buffer register, waveforms are not output to the corresponding MTIOCnC and MTIOCnD pins (n = 0, 3, 4, 6, or 7). If the MTIOCnC or MTIOCnD output is selected as the function of the pins, they enter high-impedance state. To output a specified level, make necessary settings for general output ports in the port mode registers (PMn), port registers (Pn), and port mode control registers (PMCn) of the I/O ports.
- When making a transition to a mode (CPWM or RPWM) in which the pin output level is selected by the timer output control register (TOCR1A, TOCR2A, TOCR1B, or TOCR2B) setting, temporarily disable output in MTU3 and MTU4 (or MTU6 and MTU7) with the timer output master enable register (TOERA or TOERB). If the MTIOCnm output (n =3, 4, 6, or 7 and m = A to D) is selected as the function of pins, they enter high-impedance state. To output a specified level, make necessary settings for general output ports in the port mode registers (PMn), port registers (Pn), and port mode control registers (PMCn) of the I/O ports. Switch to normal mode, perform initialization with TIOR, restore TIOR to its initial value, then operate the MTU in accordance with the mode setting procedure (TOCR1A setting, TOCR2A setting, TMDR1 setting, and TOERA setting (TOCR1B setting, TOCR2B setting, TMDR1 setting, and TOERB setting)).

NOTE

Channel number is substituted for “n” indicated in this section.

Pin initialization procedures are described below for the numbered combinations in **Table 16.80**. The active level is assumed to be low.

### (1) Operation When Error Occurs in Normal Mode and Operation is Restarted in Normal Mode

**Figure 16.154** shows a case in which an error occurs in normal mode and operation is restarted in normal mode after re- setting.

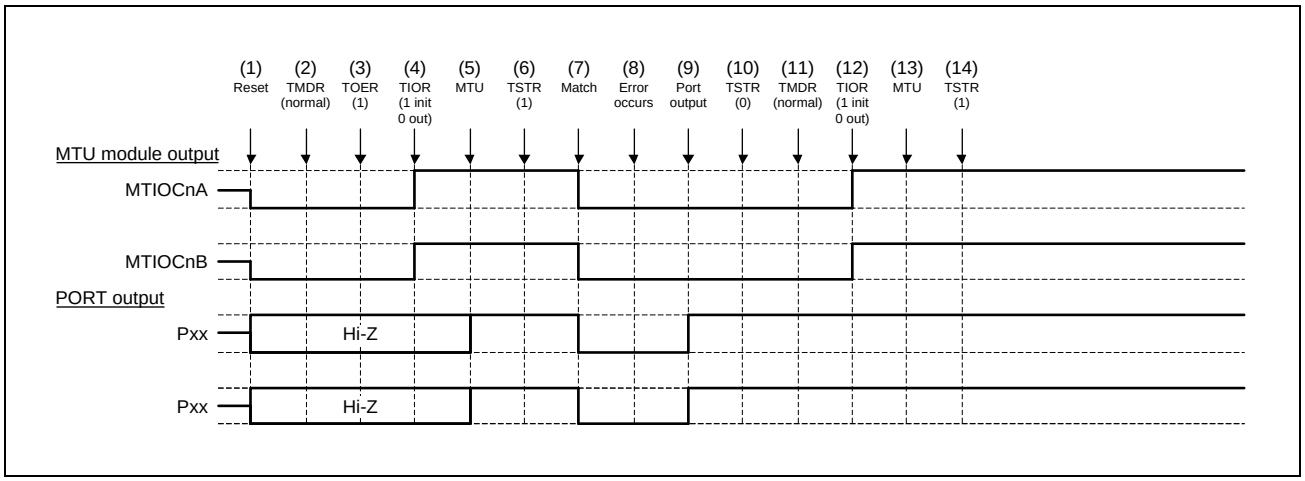



Figure 16.154 Error Occurrence in Normal Mode, Recovery in Normal Mode

(1) After a reset, the MTU output goes low and the ports enter high-impedance state.

(2) After a reset, the TMDR1 setting is for normal mode.

(3) For MTU3 and MTU4 (MTU6 and MTU7), enable output with TOERA (TOERB) before initializing the pins with TIOR.

(4) Initialize the pins with TIOR. (In the example, the initial output is a high level, and a low level is output on compare match occurrence.)

(5) Set MTU output using the port mode control registers (PMCn) and port function control registers (PFCm) corresponding to the GPIO ports.

(6) Start count operation by setting TSTRA (TSTRB).

(7) Output goes low on compare match occurrence.

(8) An error occurs.

(9) Allow non-active level output by setting the pins as general output ports using the port mode registers (PMn), port registers (Pn), and port mode control registers (PMCn) of the I/O ports.

(10) Stop count operation by setting TSTRA (TSTRB).

(11) This step is not necessary when restarting in normal mode.

(12) Initialize the pins with TIOR.

(13) Set MTU output using the port mode control registers (PMCn) and Port function control registers (PFCm) corresponding to the GPIO ports.

(14) Restart operation by setting TSTRA (TSTRB).

### (2) Operation When Error Occurs in Normal Mode and Operation is Restarted in PWM Mode 1

**Figure 16.155** shows a case in which an error occurs in normal mode and operation is restarted in PWM mode 1 after re-setting.

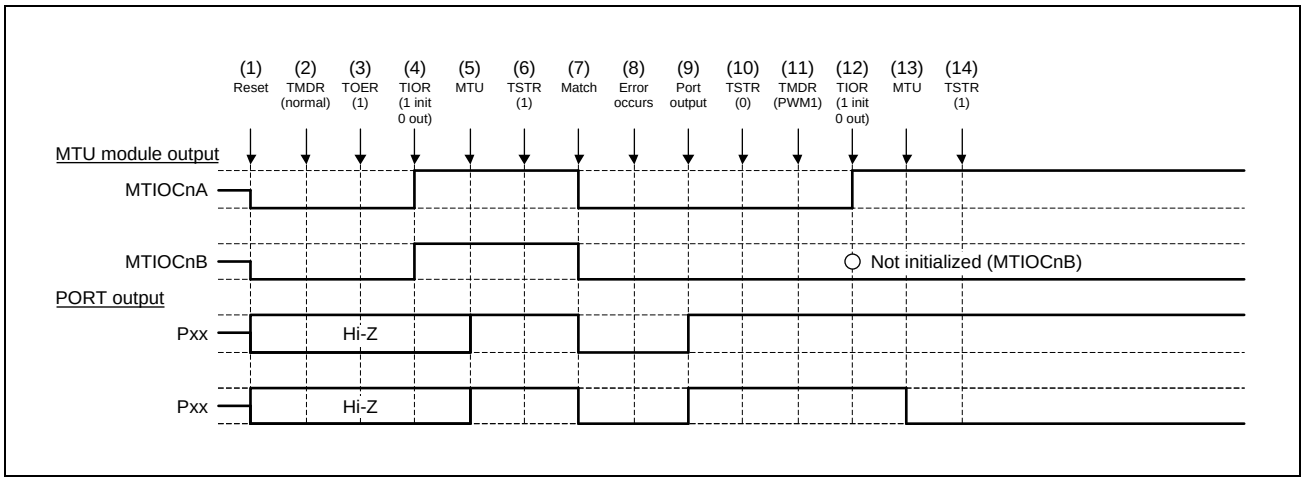



Figure 16.155 Error Occurrence in Normal Mode, Recovery in PWM Mode 1

(1) to (10) are the same as in **Figure 16.154**.

(11) Set PWM mode 1.

(12) Initialize the pins with TIOR. (In PWM mode 1, waveforms are not output to the MTIOCnB (MTIOCnD) pins. To output a specified level, make necessary settings for general output ports in the port mode registers (PMn) and port registers (Pn) of the I/O ports.)

(13) Set MTU output using the port mode control registers (PMCn) and Port function control registers (PFCm) corresponding to the GPIO ports.

(14) Restart operation by setting TSTRA (TSTRB).

### (3) Operation When Error Occurs in Normal Mode and Operation is Restarted in PWM mode 2

**Figure 16.156** shows a case in which an error occurs in normal mode and operation is restarted in PWM mode 2 after re-setting.

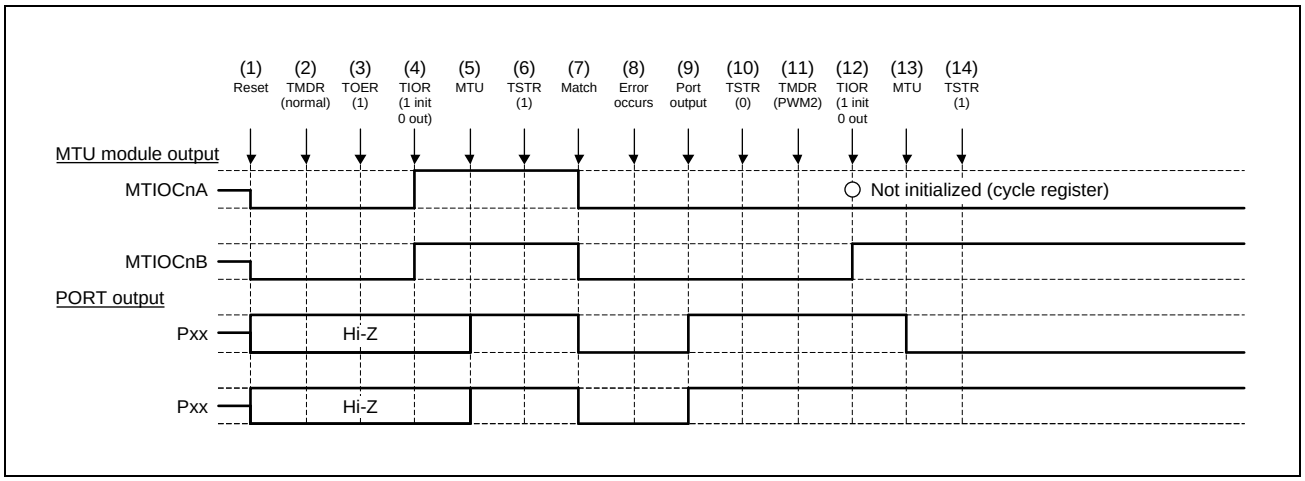



Figure 16.156 Error Occurrence in Normal Mode, Recovery in PWM Mode 2

(1) to (10) are the same as in **Figure 16.154**.

(11) Set PWM mode 2.

(12) Initialize the pins with TIOR. (In PWM mode 2, waveforms are not output to the pins corresponding to the TGR registers used as cycle registers. To output a specified level, set up the general output ports with the port mode registers (PMn) and port registers (Pn) of the I/O ports.)

(13) Set MTU output using the port mode control registers (PMCn) and Port function control registers (PFCm) corresponding to the GPIO ports.

(14) Restart operation by setting TSTRA.

NOTE

PWM mode 2 can only be selected for MTU0 to MTU2, and therefore TOERA setting is not necessary.

### (4) Operation When Error Occurs in Normal Mode and Operation is Restarted in Phase Counting Mode

**Figure 16.157** shows a case in which an error occurs in normal mode and operation is restarted in phase counting mode after re-setting.

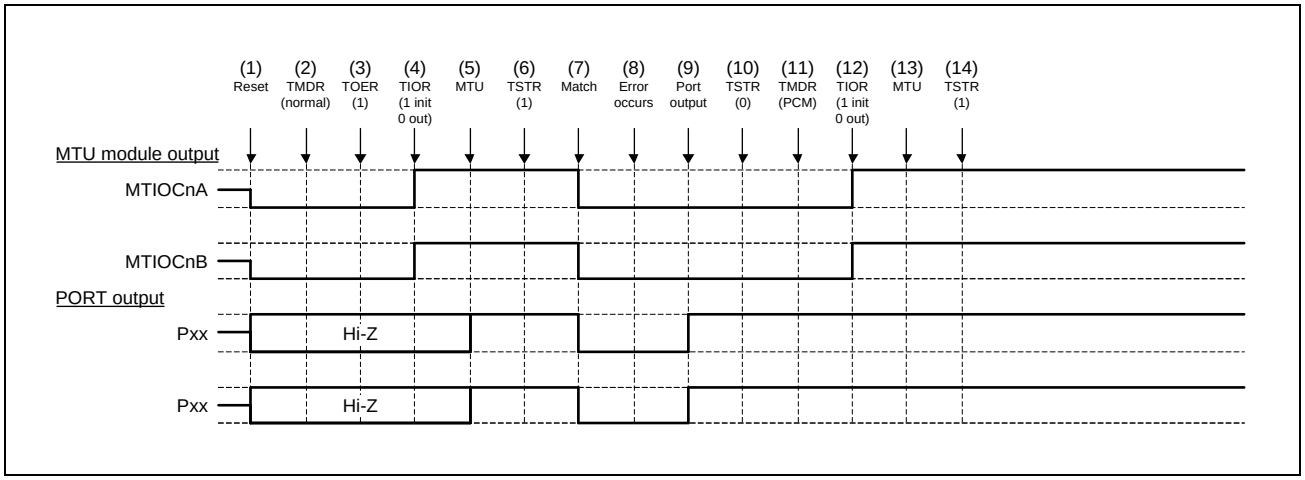



Figure 16.157 Error Occurrence in Normal Mode, Recovery in Phase Counting Mode

(1) to (10) are the same as in **Figure 16.154**.

(11) Set the phase counting mode.

(12) Initialize the pins with TIOR.

(13) Set MTU output using the port mode control registers (PMCn) and Port function control registers (PFCm) corresponding to the GPIO ports.

(14) Restart operation by setting TSTRA.

NOTE

The phase counting mode can only be selected for MTU1 and MTU2, and therefore TOERA setting is not necessary.

### (5) Operation When Error Occurs in Normal Mode and Operation is Restarted in Complementary PWM Mode

**Figure 16.158** shows a case in which an error occurs in normal mode and operation is restarted in complementary PWM mode after re-setting.

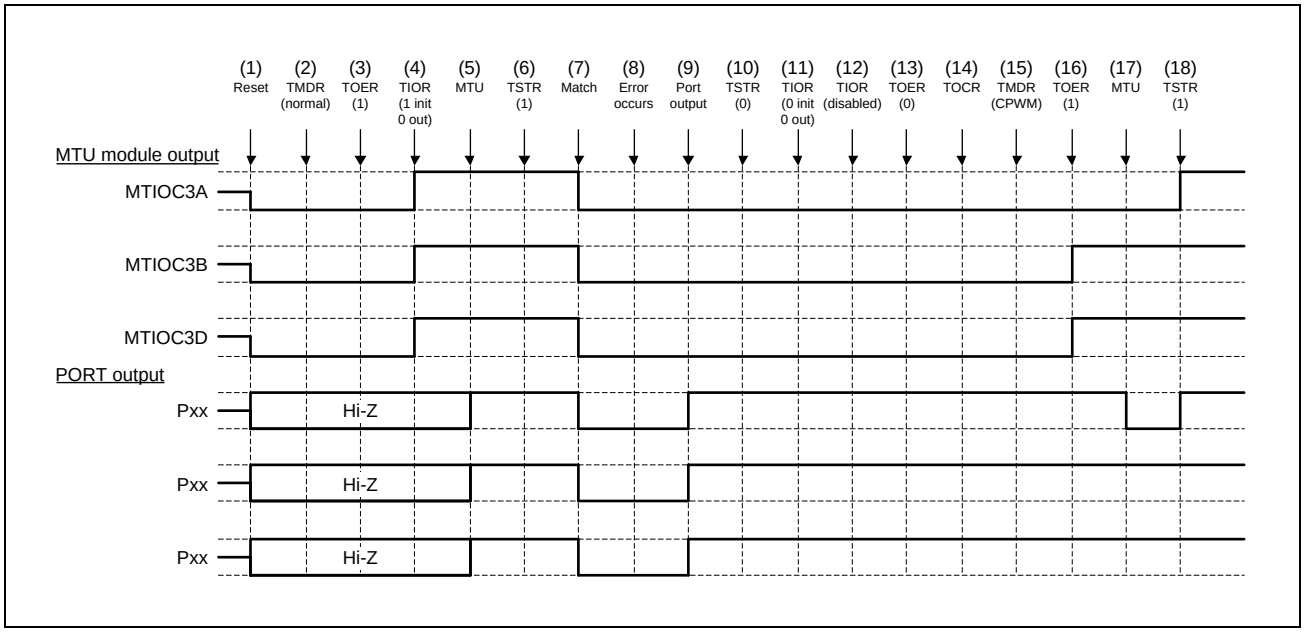



Figure 16.158 Error Occurrence in Normal Mode, Recovery in Complementary PWM Mode

(1) to (10) are the same as in **Figure 16.154**.

(11) Initialize the normal mode waveform generation section with TIOR.

(12) Disable operation of the normal mode waveform generation section with TIOR.

(13) Disable output in MTU3 and MTU4 (MTU6 and MTU7) with TOERA (TOERB).

(14) Select the complementary PWM output level and enable or disable cyclic output with TOCR1A and TOCR2A (TOCR1B and TOCR2B).

(15) Set complementary PWM mode.

(16) Enable output in MTU3 and MTU4 (MTU6 and MTU7) with TOERA (TOERB).

(17) Set MTU output using the port mode control registers (PMCn) and Port function control registers (PFCm) corresponding to the GPIO ports.

(18) Restart operation by setting TSTRA (TSTRB).

### (6) Operation When Error Occurs in Normal Mode and Operation is Restarted in Reset-Synchronized PWM Mode

**Figure 16.159** shows a case in which an error occurs in normal mode and operation is restarted in reset-synchronized PWM mode after re-setting.

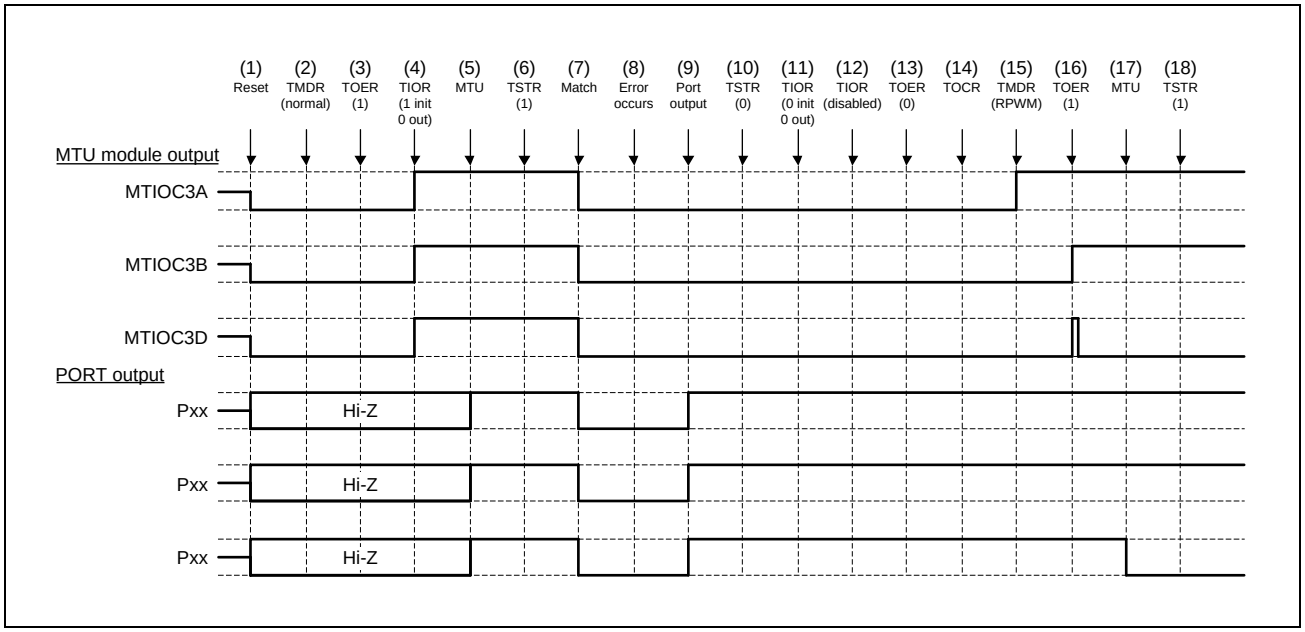



Figure 16.159 Error Occurrence in Normal Mode, Recovery in Reset-Synchronized PWM Mode

(1) to (13) are the same as in **Figure 16.156**.

(14) Select the reset-synchronized PWM output level and enable or disable cyclic output with TOCR1A and TOCR2A (TOCR1B and TOCR2B).

(15) Set reset-synchronized PWM mode.

(16) Enable output from MTU3 and MTU4 (MTU6 and MTU7) with TOERA (TOERB).

(17) Set MTU output using the port mode control registers (PMCn) and Port function control registers (PFCm) corresponding to the GPIO ports.

(18) Restart operation by setting TSTRA (TSTRB).

### (7) Operation When Error Occurs in PWM Mode 1 and Operation is Restarted in Normal Mode

**Figure 16.160** shows a case in which an error occurs in PWM mode 1 and operation is restarted in normal mode after re-setting.

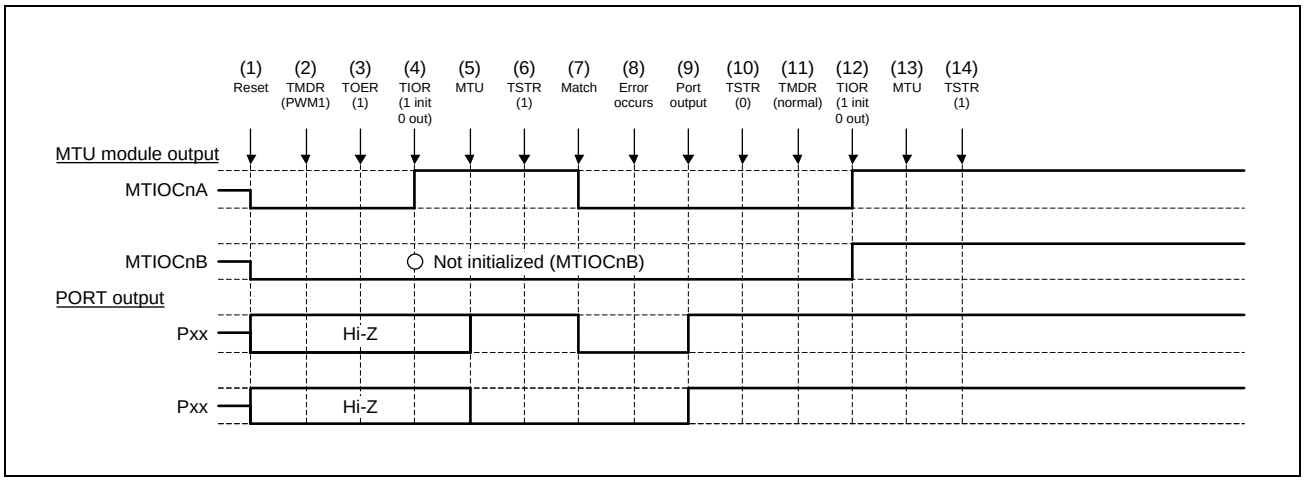



Figure 16.160 Error Occurrence in PWM Mode 1, Recovery in Normal Mode

(1) After a reset, the MTU output goes low and the ports enter high-impedance state.

(2) Set PWM mode 1.

(3) For MTU3 and MTU4 (MTU6 and MTU7), enable output with TOERA (TOERB) before initializing the pins with TIOR.

(4) Initialize the pins with TIOR. (In the example, the initial output is a high level, and a low level is output on compare match occurrence. In PWM mode 1, the MTIOCnB side is not initialized.)

(5) Set MTU output using the port mode control registers (PMCn) and Port function control registers (PFCm) corresponding to the GPIO ports.

(6) Start count operation by setting TSTRA (TSTRB).

(7) Output goes low on compare match occurrence.

(8) An error occurs.

(9) Allow non-active level output by setting the pins as general output ports using the port mode registers (PMn), port registers (Pn), and port mode control registers (PMCn) of the I/O ports.

(10) Stop count operation by setting TSTRA (TSTRB).

(11) Set normal mode.

(12) Initialize the pins with TIOR.

(13) Set MTU output using the port mode control registers (PMCn) and Port function control registers (PFCm) corresponding to the GPIO ports.

(14) Restart operation by setting TSTRA (TSTRB).

### (8) Operation When Error Occurs in PWM Mode 1 and Operation is Restarted in PWM mode 1

**Figure 16.161** shows a case in which an error occurs in PWM mode 1 and operation is restarted in PWM mode 1 after re-setting.

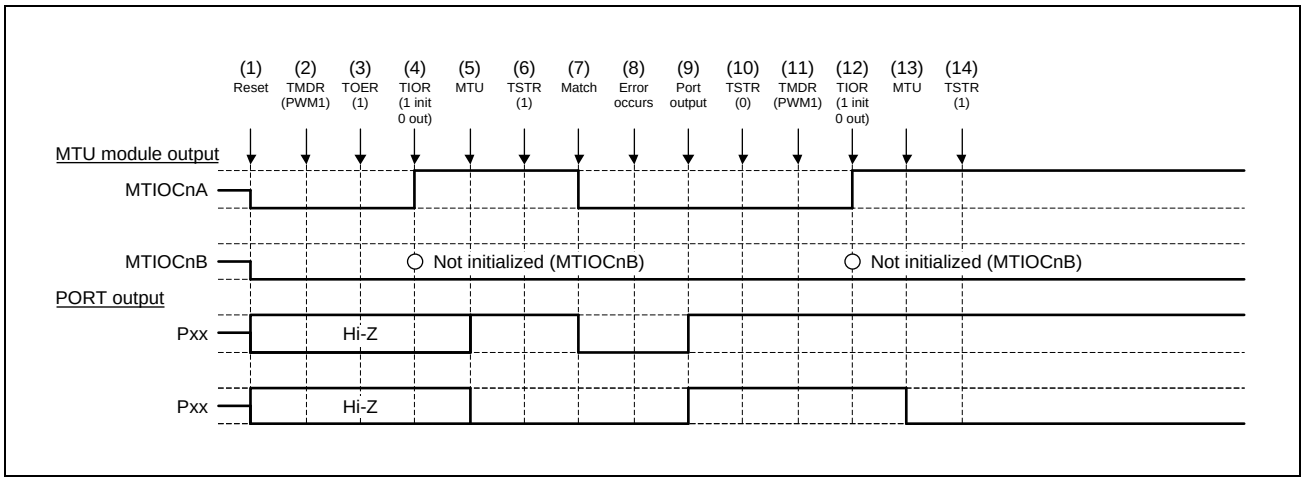



Figure 16.161 Error Occurrence in PWM Mode 1, Recovery in PWM Mode 1

(1) to (10) are the same as in **Figure 16.160**.

(11) This step is not necessary when restarting in PWM mode 1.

(12) Initialize the pins with TIOR. (In PWM mode 1, waveforms are not output to the MTIOCnB (MTIOCnD) pins. To output a specified level, make necessary settings for general output ports in the port mode registers (PMn) and port registers (Pn) of the I/O ports.)

(13) Set MTU output using the port mode control registers (PMCn) and Port function control registers (PFCm) corresponding to the GPIO ports.

(14) Restart operation by setting TSTRA (TSTRB).

### (9) Operation When Error Occurs in PWM Mode 1 and Operation is Restarted in PWM mode 2

**Figure 16.162** shows a case in which an error occurs in PWM mode 1 and operation is restarted in PWM mode 2 after re-setting.

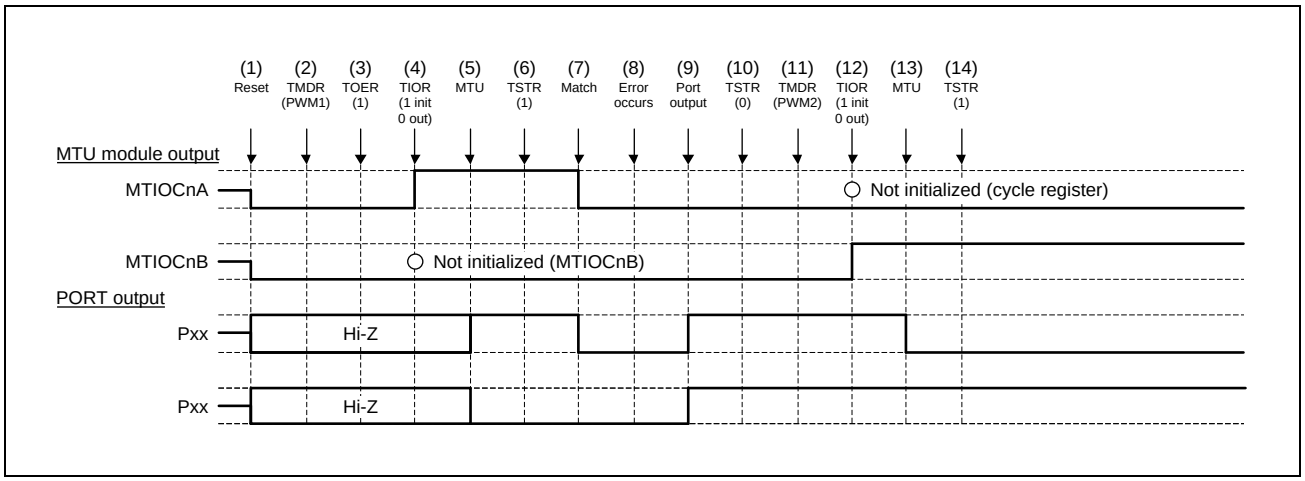



Figure 16.162 Error Occurrence in PWM Mode 1, Recovery in PWM Mode 2

(1) to (10) are the same as in **Figure 16.160**.

(11) Set PWM mode 2.

(12) Initialize the pins with TIOR. (In PWM mode 2, waveforms are not output to the pins corresponding to the TGR registers used as cycle registers.
To output a specified level, make necessary settings for general output ports in the port mode registers (PMn) and port registers (Pn) of the I/O ports.)

(13) Set MTU output using the port mode control registers (PMCn) and Port function control registers (PFCm) corresponding to the GPIO ports.

(14) Restart operation by setting TSTRA.

NOTE

PWM mode 2 can only be selected for MTU0 to MTU2, and therefore TOERA setting is not necessary.

### (10) Operation When Error Occurs in PWM Mode 1 and Operation is Restarted in Phase Counting Mode

**Figure 16.163** shows a case in which an error occurs in PWM mode 1 and operation is restarted in phase counting mode after re-setting.

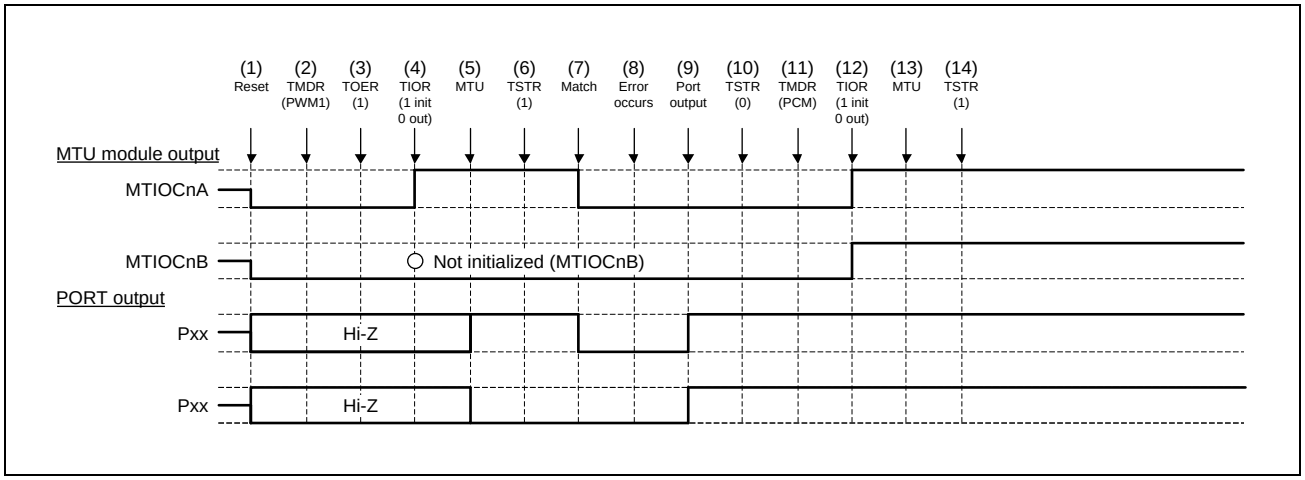



Figure 16.163 Error Occurrence in PWM Mode 1, Recovery in Phase Counting Mode

(1) to (10) are the same as in **Figure 16.160**.

(11) Set the phase counting mode.

(12) Initialize the pins with TIOR.

(13) Set MTU output using the port mode control registers (PMCn) and Port function control registers (PFCm) corresponding to the GPIO ports.

(14) Restart operation by setting TSTRA.

NOTE

The phase counting mode can only be selected for MTU1 and MTU2, and therefore TOERA setting is not necessary.

### (11) Operation When Error Occurs in PWM Mode 1 and Operation is Restarted in Complementary PWM Mode

**Figure 16.164** shows a case in which an error occurs in PWM mode 1 and operation is restarted in complementary PWM mode after re-setting.

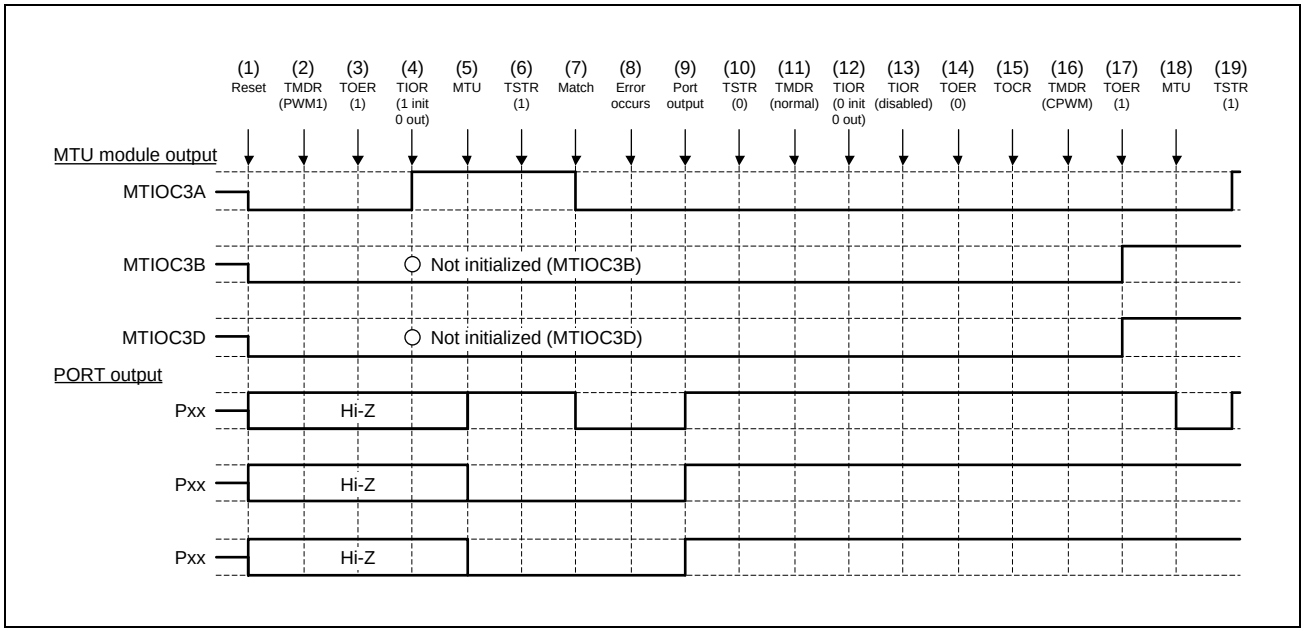



Figure 16.164 Error Occurrence in PWM Mode 1, Recovery in Complementary PWM Mode

(1) to (10) are the same as in **Figure 16.160**.

(11) Set normal mode to initialize 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 output in MTU3 and MTU4 (MTU6 and MTU7) with TOERA (TOERB).

(15) Select the complementary PWM output level and enable or disable cyclic output with TOCR1A and TOCR2A (TOCR1B and TOCR2B).

(16) Set complementary PWM mode.

(17) Enable output in MTU3 and MTU4 (MTU6 and MTU7) with TOERA (TOERB).

(18) Set MTU output using the port mode control registers (PMCn) and Port function control registers (PFCm) corresponding to the GPIO ports.

(19) Restart operation by setting TSTRA (TSTRB).

### (12) Operation When Error Occurs in PWM Mode 1 and Operation is Restarted in Reset- Synchronized PWM Mode

**Figure 16.165** shows a case in which an error occurs in PWM mode 1 and operation is restarted in reset-synchronized PWM mode after re-setting.

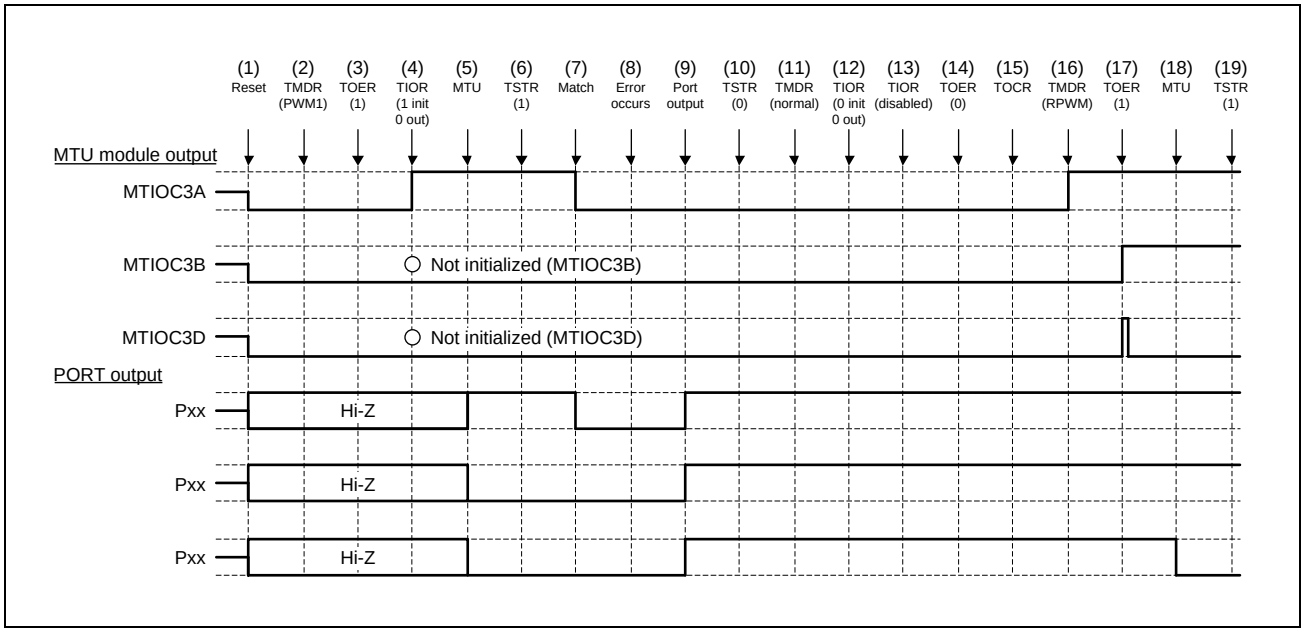



Figure 16.165 Error Occurrence in PWM Mode 1, Recovery in Reset-Synchronized PWM Mode

(1) to (14) are the same as in **Figure 16.164**.

(15) Select the reset-synchronized PWM output level and enable or disable cyclic output with TOCR1A and TOCR2A (TOCR1B and TOCR2B).

(16) Set reset-synchronized PWM mode.

(17) Enable output in MTU3 and MTU4 (MTU6 and MTU7) with TOERA (TOERB).

(18) Set MTU output using the port mode control registers (PMCn) and Port function control registers (PFCm) corresponding to the GPIO ports.

(19) Restart operation by setting TSTRA (TSTRB).

### (13) Operation When Error Occurs in PWM Mode 2 and Operation is Restarted in Normal Mode

**Figure 16.166** shows a case in which an error occurs in PWM mode 2 and operation is restarted in normal mode after re-setting.

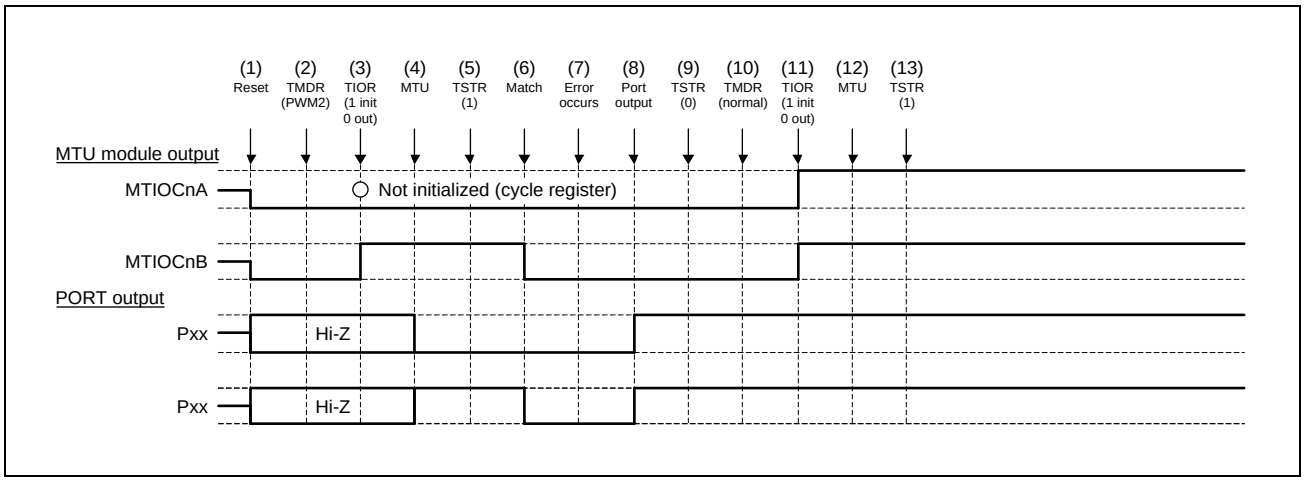



Figure 16.166 Error Occurrence in PWM Mode 2, Recovery in Normal Mode

(1) After a reset, the MTU output goes low and the ports enter high-impedance state.

(2) Set PWM mode 2.

(3) Initialize the pins with TIOR. (In the example, the initial output is a high level, and a low level is output on compare match occurrence. In PWM mode 2, the pins corresponding to the TGR registers used as cycle registers are not initialized. In the example, TGRA in MTUn is used as a cycle register.)

(4) Set MTU output using the port mode control registers (PMCn) and Port function control registers (PFCm) corresponding to the GPIO ports.

(5) Start count operation by setting TSTRA.

(6) Output goes low on compare match occurrence.

(7) An error occurs.

(8) Allow non-active level output by setting the pins as general output ports using the port mode registers (PMn), port registers (Pn), and port mode control registers (PMCn) of the I/O ports.

(9) Stop count operation by setting TSTRA.

(10) Set normal mode.

(11) Initialize the pins with TIOR.

(12) Set MTU output using the port mode control registers (PMCn) and Port function control registers (PFCm) corresponding to the GPIO ports.

(13) Restart operation by setting TSTRA.

### (14) Operation When Error Occurs in PWM Mode 2 and Operation is Restarted in PWM Mode 1

**Figure 16.167** shows a case in which an error occurs in PWM mode 2 and operation is restarted in PWM mode 1 after re-setting.

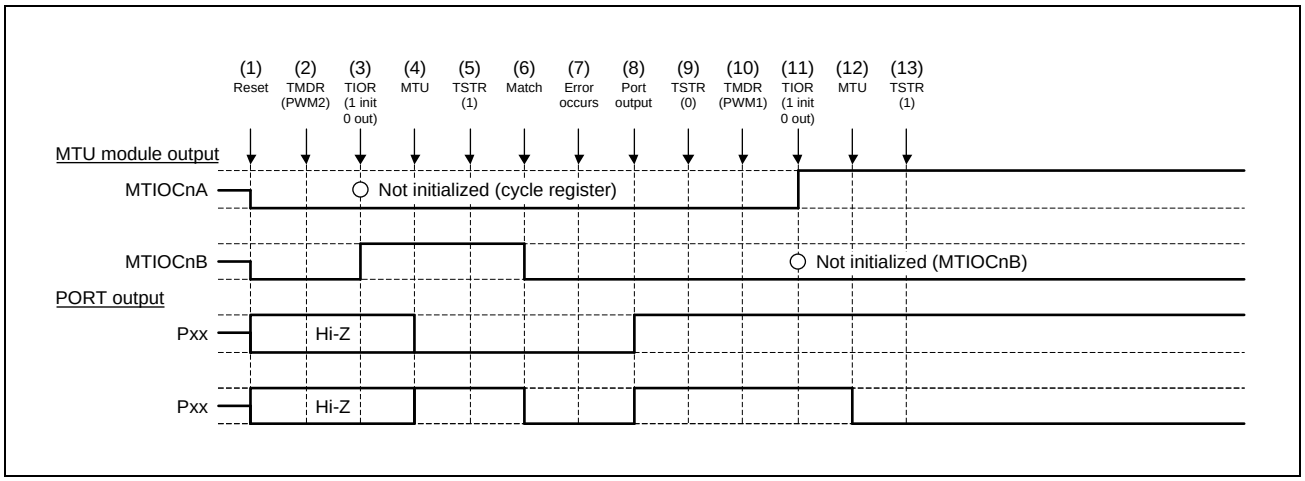



Figure 16.167 Error Occurrence in PWM Mode 2, Recovery in PWM Mode 1

(1) to (9) are the same as in **Figure 16.166**.

(10) Set PWM mode 1.

(11) Initialize the pins with TIOR. (In PWM mode 1, waveforms are not output to the MTIOCnB (MTIOCnD) pins. To output a specified level, make necessary settings for general output ports in the port mode registers (PMn) and port registers (Pn) of the I/O ports.)

(12) Set MTU output using the port mode control registers (PMCn) and Port function control registers (PFCm) corresponding to the GPIO ports.

(13) Restart operation by setting TSTRA.

### (15) Operation When Error Occurs in PWM Mode 2 and Operation is Restarted in PWM Mode 2

**Figure 16.168** shows a case in which an error occurs in PWM mode 2 and operation is restarted in PWM mode 2 after re-setting.

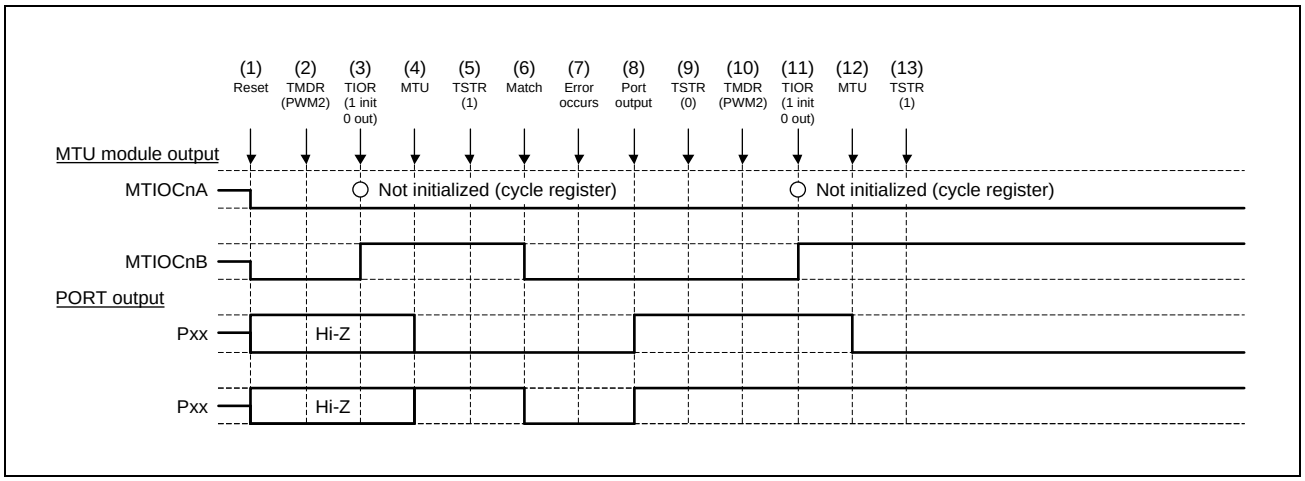



Figure 16.168 Error Occurrence in PWM Mode 2, Recovery in PWM Mode 2

(1) to (9) are the same as in **Figure 16.166**.

(10) This step is not necessary when restarting in PWM mode 2.

(11) Initialize the pins with TIOR. (In PWM mode 2, waveforms are not output to the pins corresponding to the TGR registers used as cycle registers.
To output a specified level, make necessary settings for general output ports in the port mode registers (PMn) and port registers (Pn) of the I/O ports.)

(12) Set MTU output using the port mode control registers (PMCn) and Port function control registers (PFCm) corresponding to the GPIO ports.

(13) Restart operation by setting TSTRA.

### (16) Operation When Error Occurs in PWM Mode 2 and Operation is Restarted in Phase Counting Mode

**Figure 16.169** shows a case in which an error occurs in PWM mode 2 and operation is restarted in phase counting mode after re-setting.

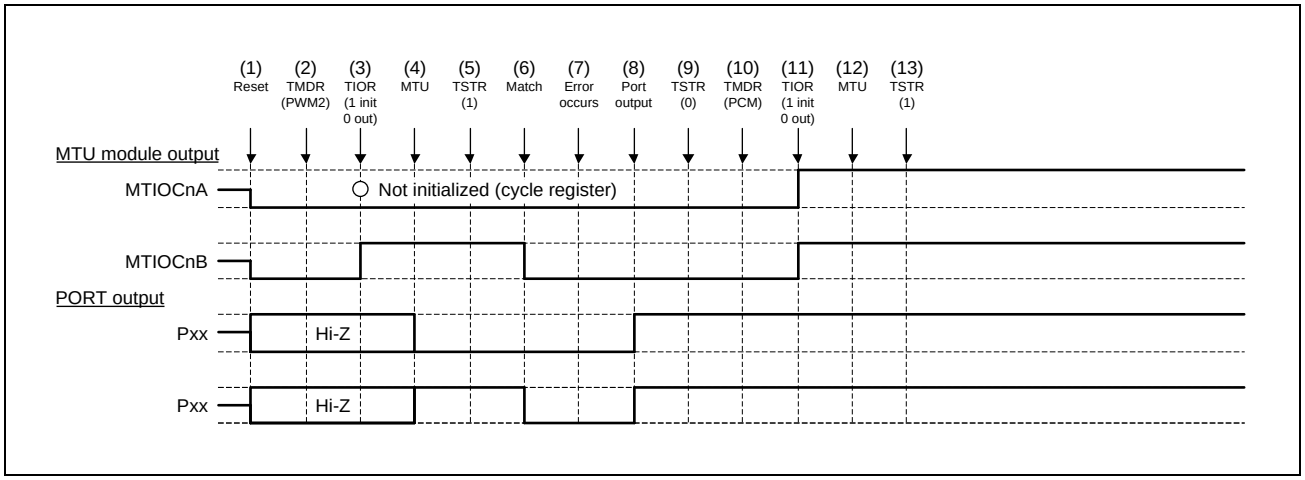



Figure 16.169 Error Occurrence in PWM Mode 2, Recovery in Phase Counting Mode

(1) to (9) are the same as in **Figure 16.166**.

(10) Set the phase counting mode.

(11) Initialize the pins with TIOR.

(12) Set MTU output using the port mode control registers (PMCn) and Port function control registers (PFCm) corresponding to the GPIO ports.

(13) Restart operation by setting TSTRA.

### (17) Operation When Error Occurs in Phase Counting Mode and Operation is Restarted in Normal Mode

**Figure 16.170** shows a case in which an error occurs in phase counting mode and operation is restarted in normal mode after re-setting.

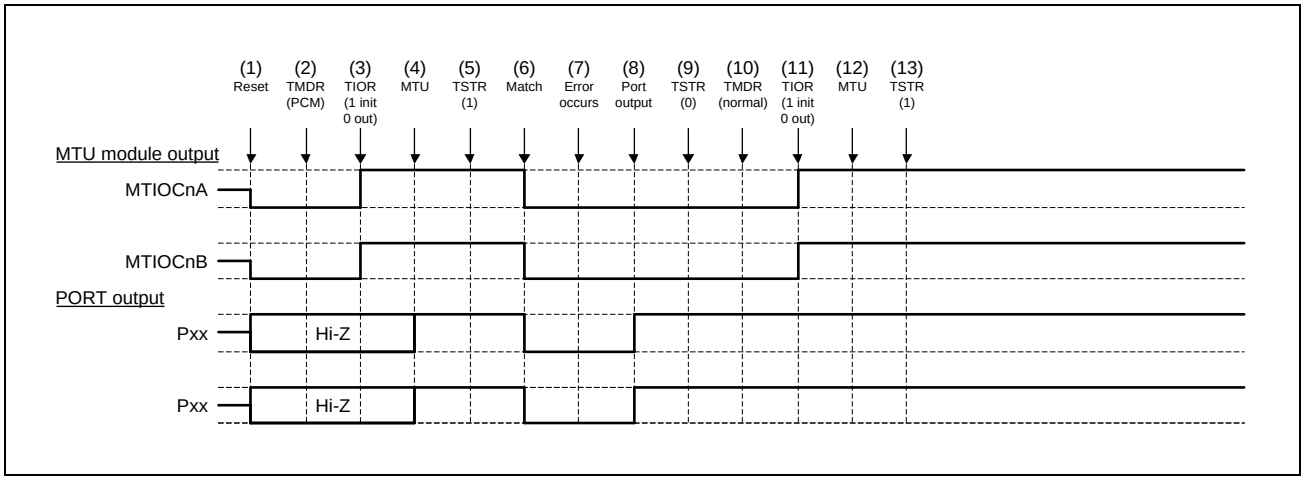



Figure 16.170 Error Occurrence in Phase Counting Mode, Recovery in Normal Mode

(1) After a reset, the MTU output goes low and the ports enter high-impedance state.

(2) Set phase counting mode.

(3) Initialize the pins with TIOR. (In the example, the initial output is a high level, and a low level is output on compare match occurrence.)

(4) Set MTU output using the port mode control registers (PMCn) and Port function control registers (PFCm) corresponding to the GPIO ports.

(5) Start count operation by setting TSTRA.

(6) Output goes low on compare match occurrence.

(7) An error occurs.

(8) Allow non-active level output by setting the pins as general output ports using the port mode registers (PMn), port registers (Pn), and port mode control registers (PMCn) of the I/O ports.

(9) Stop count operation by setting TSTRA.

(10) Set normal mode.

(11) Initialize the pins with TIOR.

(12) Set MTU output using the port mode control registers (PMCn) and Port function control registers (PFCm) corresponding to the GPIO ports.

(13) Restart operation by setting TSTRA.

### (18) Operation When Error Occurs in Phase Counting Mode and Operation is Restarted in PWM Mode 1

**Figure 16.171** shows a case in which an error occurs in phase counting mode and operation is restarted in PWM mode 1 after re-setting.

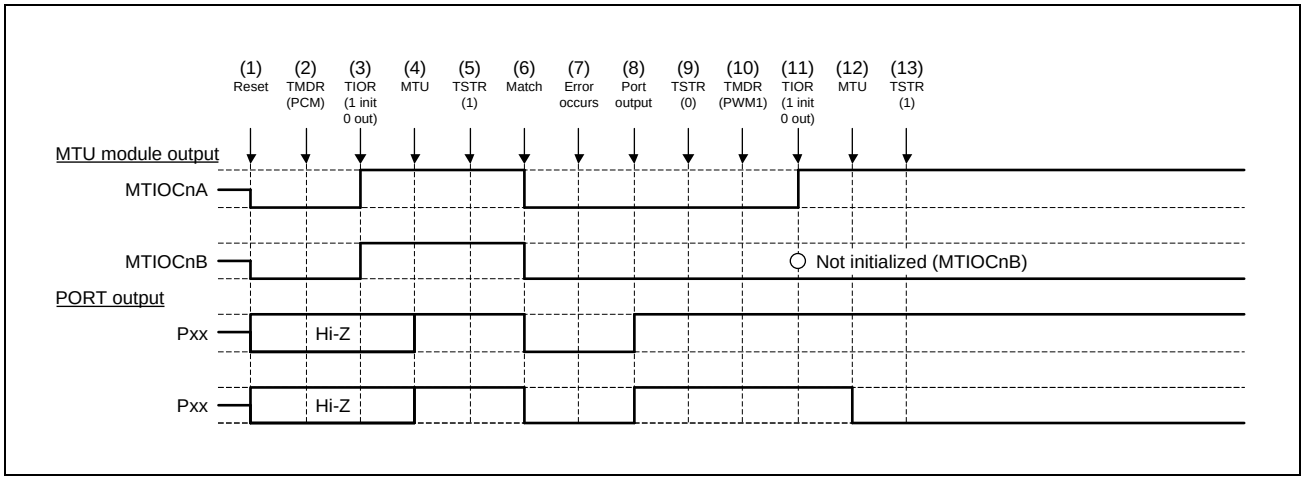



Figure 16.171 Error Occurrence in Phase Counting Mode, Recovery in PWM Mode 1

(1) to (9) are the same as in **Figure 16.170**.

(10) Set PWM mode 1.

(11) Initialize the pins with TIOR. (In PWM mode 1, waveforms are not output to the MTIOCnB (MTIOCnD) pins. To output a specified level, make necessary settings for general output ports in the port mode registers (PMn) and port registers (Pn) of the I/O ports.)

(12) Set MTU output using the port mode control registers (PMCn) and Port function control registers (PFCm) corresponding to the GPIO ports.

(13) Restart operation by setting TSTRA.

### (19) Operation When Error Occurs in Phase Counting Mode and Operation is Restarted in PWM Mode 2

**Figure 16.172** shows a case in which an error occurs in phase counting mode and operation is restarted in PWM mode 2 after re-setting.

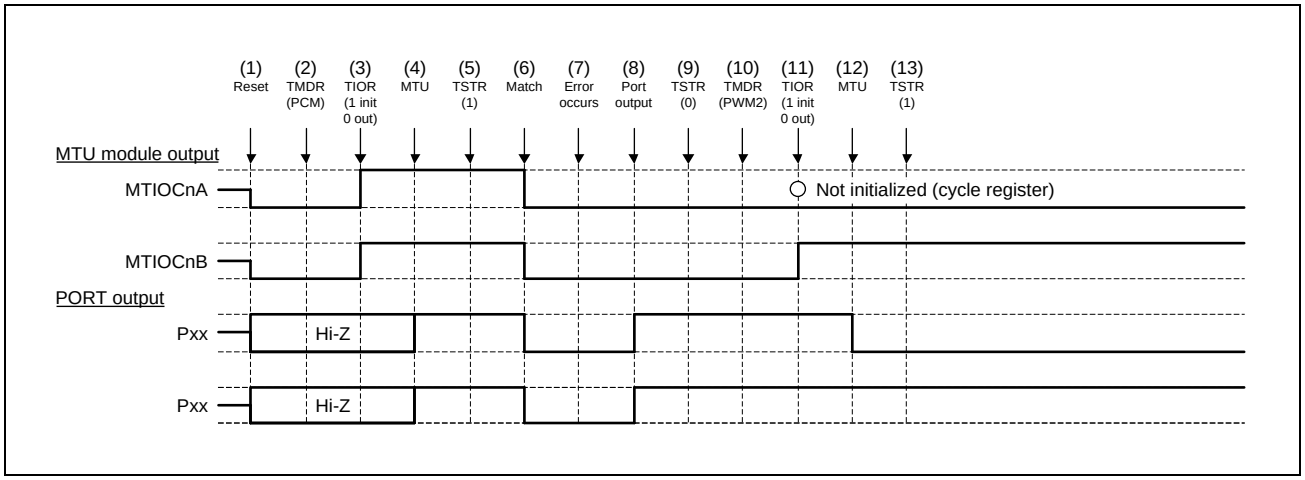



Figure 16.172 Error Occurrence in Phase Counting Mode, Recovery in PWM Mode 2

(1) to (9) are the same as in **Figure 16.170**.

(10) Set PWM mode 2.

(11) Initialize the pins with TIOR. (In PWM mode 2, waveforms are not output to the pins corresponding to the TGR registers used as cycle registers.
To output a specified level, make necessary settings for general output ports in the port mode registers (PMn) and port registers (Pn) of the I/O ports.)

(12) Set MTU output using the port mode control registers (PMCn) and Port function control registers (PFCm) corresponding to the GPIO ports.

(13) Restart operation by setting TSTRA.

### (20) Operation When Error Occurs in Phase Counting Mode and Operation is Restarted in Phase Counting Mode

**Figure 16.173** shows a case in which an error occurs in phase counting mode and operation is restarted in phase counting mode after re-setting.

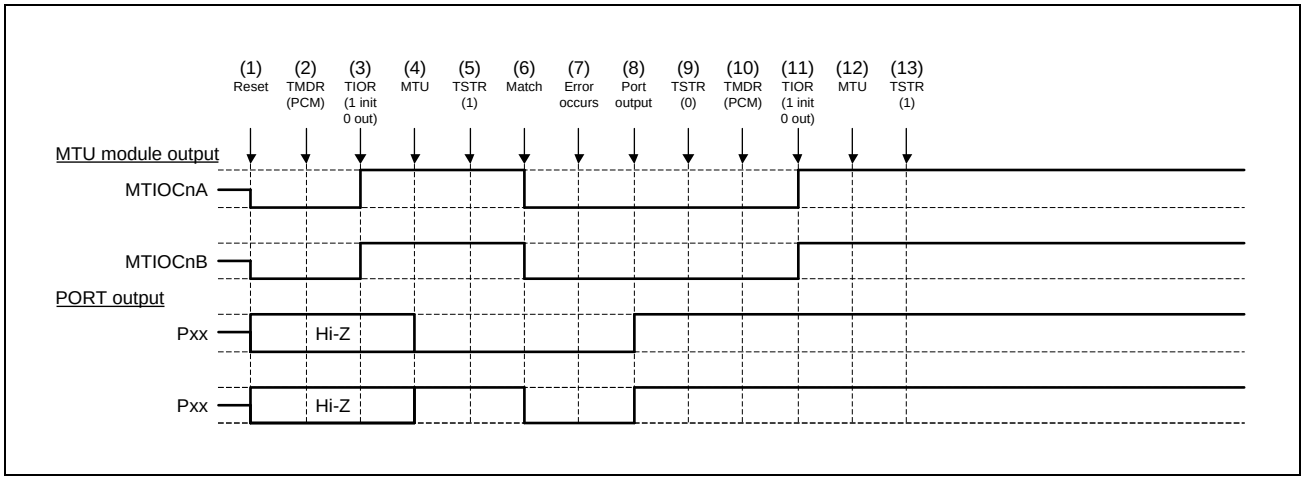



Figure 16.173 Error Occurrence in Phase Counting Mode, Recovery in Phase Counting Mode

(1) to (9) are the same as in **Figure 16.170**.

(10) This step is not necessary when restarting in phase counting mode.

(11) Initialize the pins with TIOR.

(12) Set MTU output using the port mode control registers (PMCn) and Port function control registers (PFCm) corresponding to the GPIO ports.

(13) Restart operation by setting TSTRA.

### (21) Operation When Error Occurs in Complementary PWM Mode and Operation is Restarted in Normal Mode

**Figure 16.174** shows a case in which an error occurs in complementary PWM mode and operation is restarted in normal mode after re-setting.

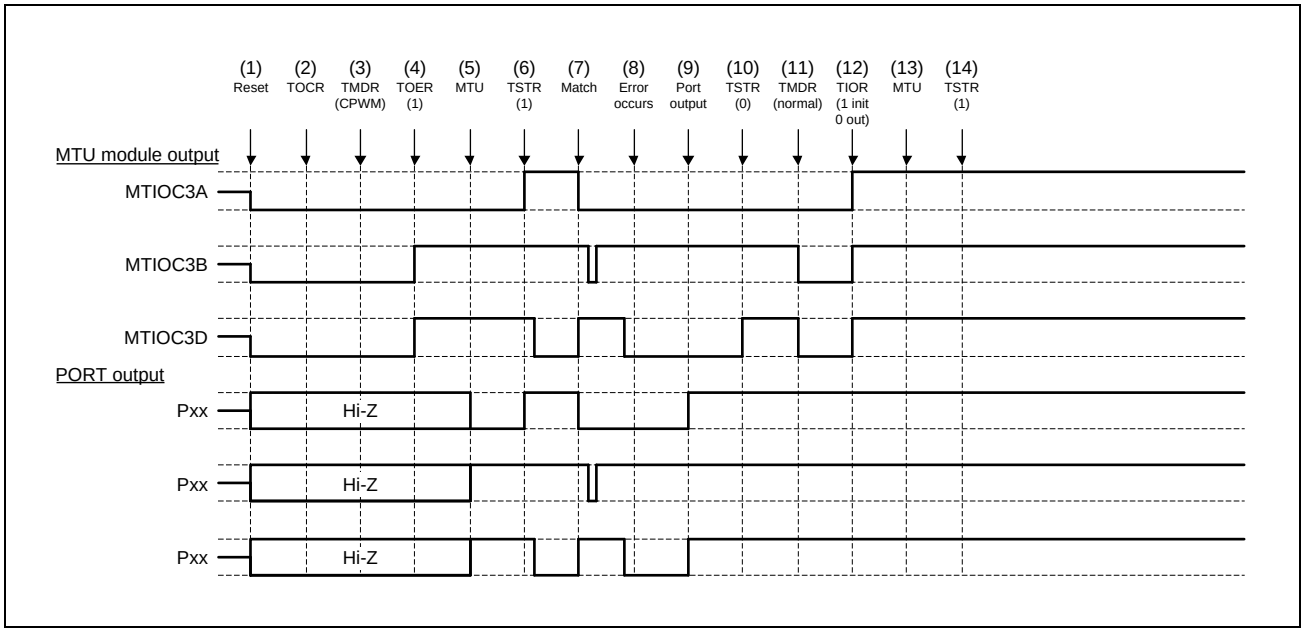



Figure 16.174 Error Occurrence in Complementary PWM Mode, Recovery in Normal Mode

(1) After a reset, the MTU3 output goes low and the ports enter high-impedance state.

(2) Select the complementary PWM output level and enable or disable cyclic output with TOCR1A and TOCR2A (TOCR1B and TOCR2B).

(3) Set complementary PWM mode.

(4) Enable output in MTU3 and MTU4 (MTU6 and MTU7) with TOERA (TOERB).

(5) Set MTU output using the port mode control registers (PMCn) and Port function control registers (PFCm) corresponding to the GPIO ports.

(6) Start count operation by setting TSTRA (TSTRB).

(7) The complementary PWM waveform is output on compare match occurrence.

(8) An error occurs.

(9) Allow non-active level output by setting the pins as general output ports using the port mode registers (PMn), port registers (Pn), and port mode control registers (PMCn) of the I/O ports.

(10) Stop count operation by setting TSTRA (TSTRB). (MTU output becomes the initial complementary PWM output value).

(11) Set normal mode (MTU output goes low).

(12) Initialize the pins with TIOR.

(13) Set MTU output using the port mode control registers (PMCn) and Port function control registers (PFCm) corresponding to the GPIO ports.

(14) Restart operation by setting TSTRA (TSTRB).

### (22) Operation When Error Occurs in Complementary PWM Mode and Operation is Restarted in PWM Mode 1

**Figure 16.175** shows a case in which an error occurs in complementary PWM mode and operation is restarted in PWM mode 1 after re-setting.

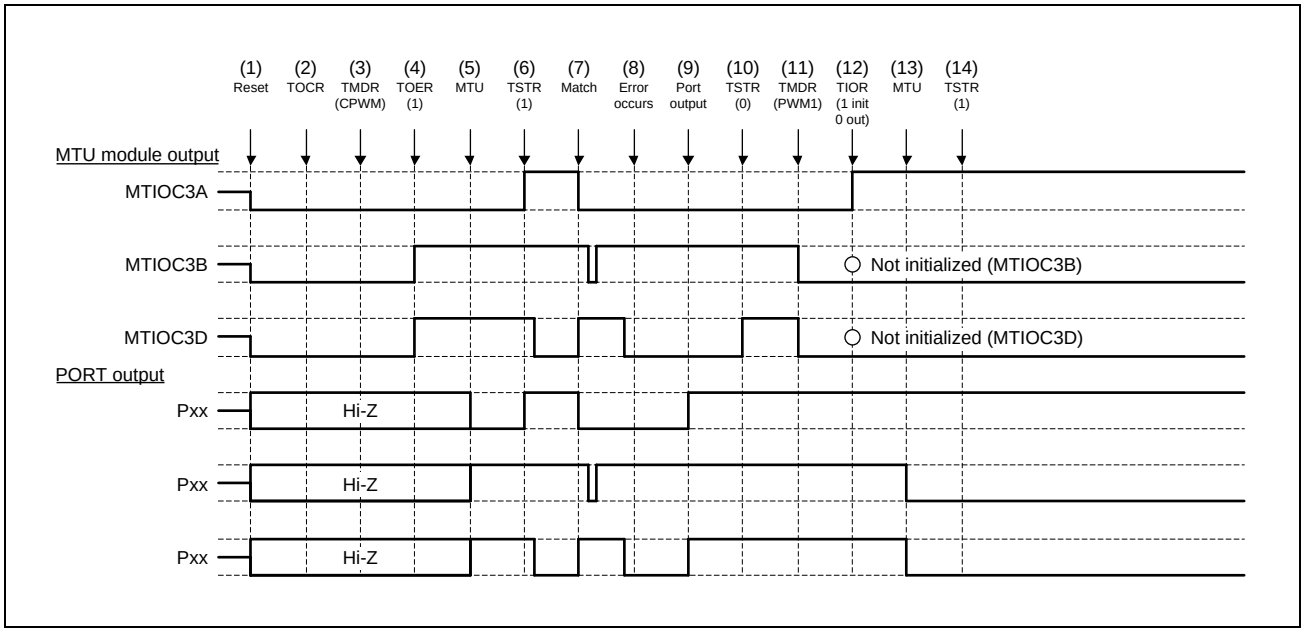



Figure 16.175 Error Occurrence in Complementary PWM Mode, Recovery in PWM Mode 1

(1) to (10) are the same as in **Figure 16.174**.

(11) Set PWM mode 1 (MTU output goes low).

(12) Initialize the pins with TIOR. (In PWM mode 1, waveforms are not output to the MTIOCnB (MTIOCnD) pins. To output a specified level, make necessary settings for general output ports in the port mode registers (PMn) and port registers (Pn) of the I/O ports.)

(13) Set MTU output using the port mode control registers (PMCn) and Port function control registers (PFCm) corresponding to the GPIO ports.

(14) Restart operation by setting TSTRA (TSTRB).

### (23) Operation When Error Occurs in Complementary PWM Mode and Operation is Restarted in Complementary PWM Mode

**Figure 16.176** shows a case in which 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 of stopping the counter).

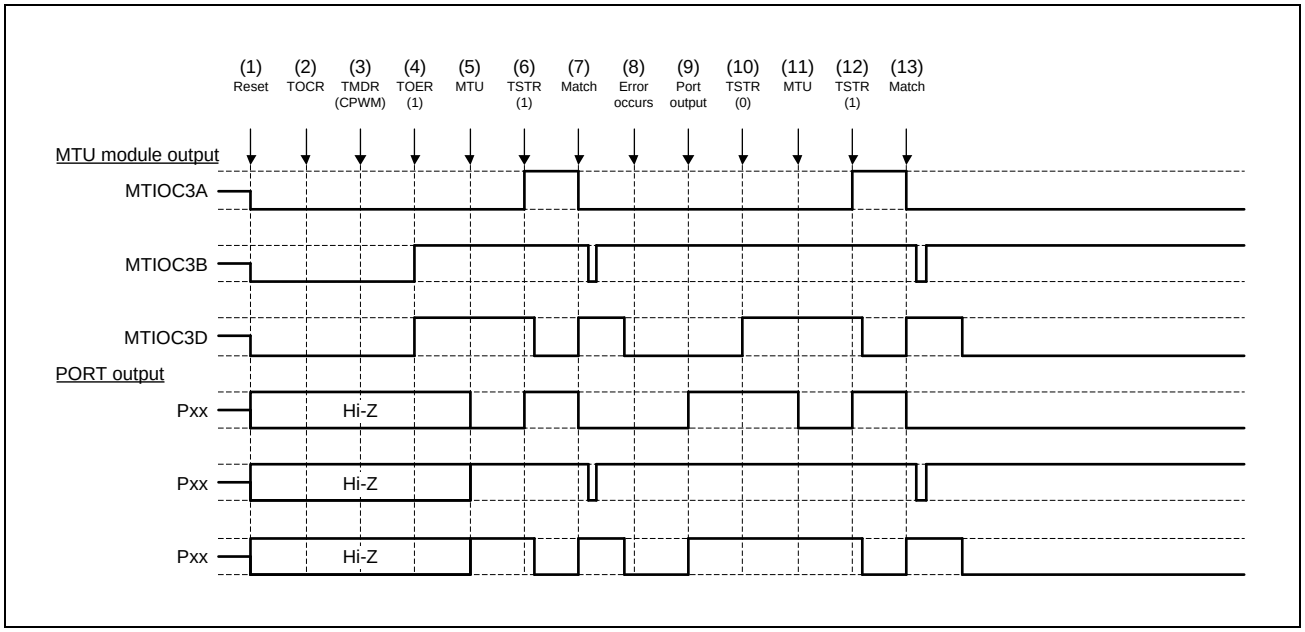



Figure 16.176 Error Occurrence in Complementary PWM Mode, Recovery in Complementary PWM Mode (1)

(1) to (10) are the same as in **Figure 16.174**.

(11) Set MTU output using the port mode control registers (PMCn) and Port function control registers (PFCm) corresponding to the GPIO ports.

(12) Restart operation by setting TSTRA (TSTRB).

(13) The complementary PWM waveform is output on compare match occurrence.

### (24) Operation When Error Occurs in Complementary PWM Mode and Operation is Restarted in Complementary PWM Mode with New Settings

**Figure 16.177** shows a case in which an error occurs in complementary PWM mode and operation is restarted in complementary PWM mode after re-setting (operation is restarted using new cycle and duty ratio settings).

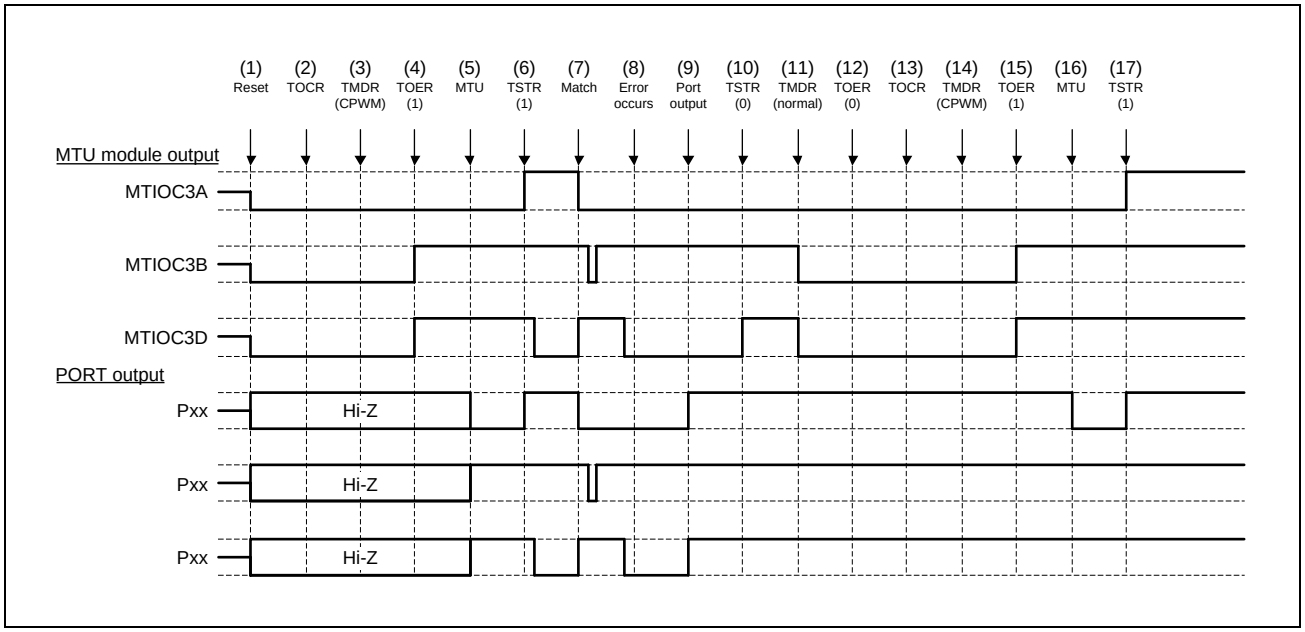



Figure 16.177 Error Occurrence in Complementary PWM Mode, Recovery in Complementary PWM Mode (2)

(1) to (10) are the same as in **Figure 16.174**.

(11) Set normal mode and make new settings (MTU output goes low).

(12) Disable output in MTU3 and MTU4 (MTU6 and MTU7) with TOERA (TOERB).

(13) Select the complementary PWM output level and enable or disable cyclic output with TOCR1A and TOCR2A (TOCR1B and TOCR2B).

(14) Set complementary PWM mode.

(15) Enable output in MTU3 and MTU4 (MTU6 and MTU7) with TOERA (TOERB).

(16) Set MTU output using the port mode control registers (PMCn) and Port function control registers (PFCm) corresponding to the GPIO ports.

(17) Restart operation by setting TSTRA (TSTRB).

### (25) Operation When Error Occurs in Complementary PWM Mode and Operation is Restarted in Reset-Synchronized PWM Mode

**Figure 16.178** shows a case in which an error occurs in complementary PWM mode and operation is restarted in reset- synchronized PWM mode after re-setting.

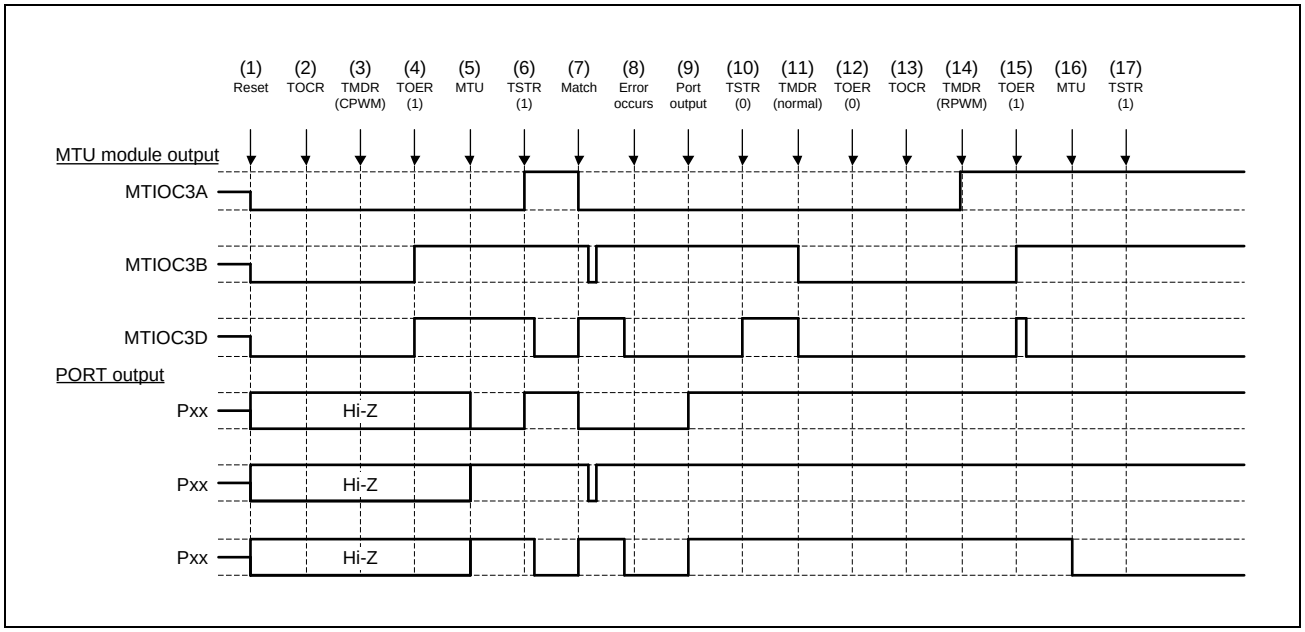



Figure 16.178 Error Occurrence in Complementary PWM Mode, Recovery in Reset-Synchronized PWM Mode

(1) to (10) are the same as in **Figure 16.174**.

(11) Set normal mode (MTU output goes low).

(12) Disable output in MTU3 and MTU4 (MTU6 and MTU7) with TOERA (TOERB).

(13) Select the reset-synchronized PWM output level and enable or disable cyclic output with TOCR1A and TOCR2A (TOCR1B and TOCR2B).

(14) Set reset-synchronized PWM mode.

(15) Enable output in MTU3 and MTU4 (MTU6 and MTU7) with TOERA (TOERB).

(16) Set MTU output using the port mode control registers (PMCn) and Port function control registers (PFCm) corresponding to the GPIO ports.

(17) Restart operation by setting TSTRA (TSTRB).

### (26) Operation When Error Occurs in Reset-Synchronized PWM Mode and Operation is Restarted in Normal Mode

**Figure 16.179** shows a case in which an error occurs in reset-synchronized PWM mode and operation is restarted in normal mode after re-setting.

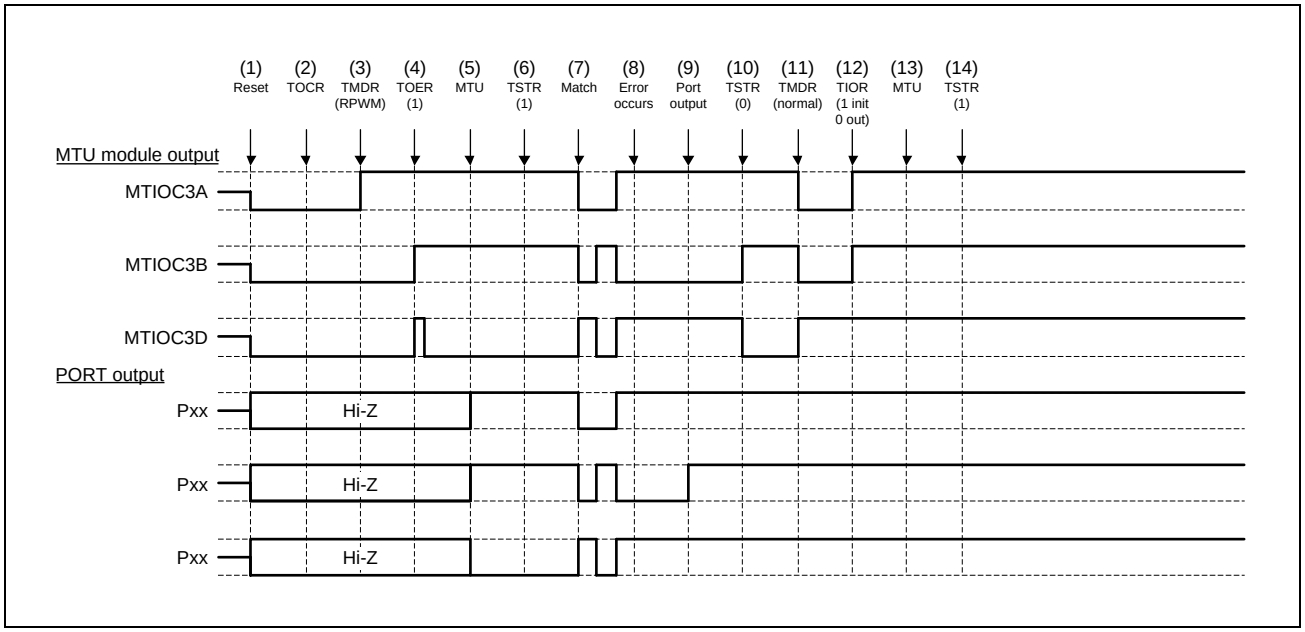



Figure 16.179 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Normal Mode

(1) After a reset, the MTU output goes low and the ports enter high-impedance state.

(2) Select the reset-synchronized PWM output level and enable or disable cyclic output with TOCR1A and TOCR2A (TOCR1B and TOCR2B).

(3) Set reset-synchronized PWM mode.

(4) Enable output in MTU3 and MTU4 (MTU6 and MTU7) with TOERA (TOERB).

(5) Set MTU output using the port mode control registers (PMCn) and Port function control registers (PFCm) corresponding to the GPIO ports.

(6) Start count operation by setting TSTRA (TSTRB).

(7) The reset-synchronized PWM waveform is output on compare match occurrence.

(8) An error occurs.

(9) Allow non-active level output by setting the pins as general output ports using the port mode registers (PMn), port registers (Pn), and port mode control registers (PMCn) of the I/O ports.

(10) Stop count operation by setting TSTRA (TSTRB). (MTU output becomes the initial reset-synchronized PWM output value.)

(11) Set normal mode (positive-phase MTU output goes low, and negative-phase output goes high).

(12) Initialize the pins with TIOR.

(13) Set MTU output using the port mode control registers (PMCn) and Port function control registers (PFCm) corresponding to the GPIO ports.

(14) Restart operation by setting TSTRA (TSTRB).

### (27) Operation When Error Occurs in Reset-Synchronized PWM Mode and Operation is Restarted in PWM Mode 1

**Figure 16.180** shows a case in which an error occurs in reset-synchronized PWM mode and operation is restarted in PWM mode 1 after re-setting.

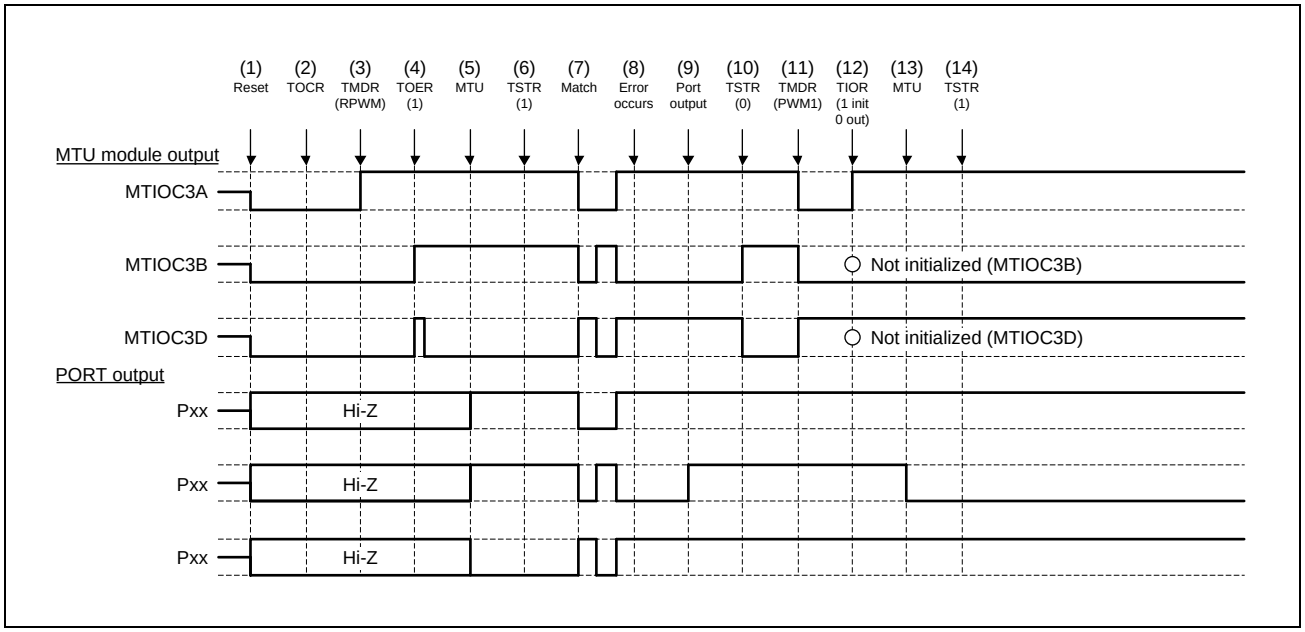



Figure 16.180 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in PWM Mode 1

(1) to (10) are the same as in **Figure 16.179**.

(11) Set PWM mode 1 (positive-phase MTU output goes low, and negative-phase output goes high).

(12) Initialize the pins with TIOR. (In PWM mode 1, waveforms are not output to the MTIOCnB (MTIOCnD) pins. To output a specified level, make necessary settings for general output ports in the port mode registers (PMn) and port registers (Pn) of the I/O ports.)

(13) Set MTU output using the port mode control registers (PMCn) and Port function control registers (PFCm) corresponding to the GPIO ports.

(14) Restart operation by setting TSTRA (TSTRB).

### (28) Operation When Error Occurs in Reset-Synchronized PWM Mode and Operation is Restarted in Complementary PWM Mode

**Figure 16.181** shows a case in which an error occurs in reset-synchronized PWM mode and operation is restarted in complementary PWM mode after re-setting.

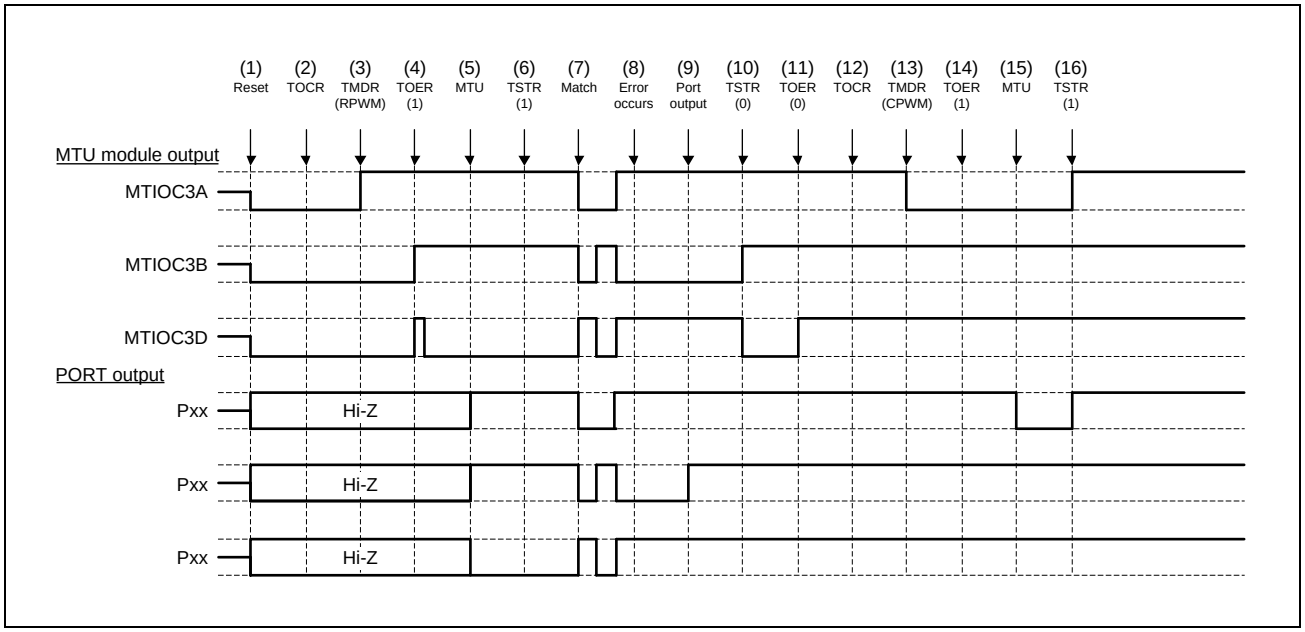



Figure 16.181 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Complementary PWM Mode

(1) to (10) are the same as in **Figure 16.179**.

(11) Disable output in MTU3 and MTU4 (MTU6 and MTU7) with TOERA (TOERB).

(12) Select the complementary PWM output level and enable or disable cyclic output with TOCR1A and TOCR2A (TOCR1B and TOCR2B).

(13) Set complementary PWM mode (MTU cyclic output pin goes low).

(14) Enable output in MTU3 and MTU4 (MTU6 and MTU7) with TOERA (TOERB).

(15) Set MTU output using the port mode control registers (PMCn) and Port function control registers (PFCm) corresponding to the GPIO ports.

(16) Restart operation by setting TSTRA (TSTRB).

### (29) Operation When Error Occurs in Reset-Synchronized PWM Mode and Operation is Restarted in Reset-Synchronized PWM Mode

**Figure 16.182** shows a case in which an error occurs in reset-synchronized PWM mode and operation is restarted in reset-synchronized PWM mode after re-setting.

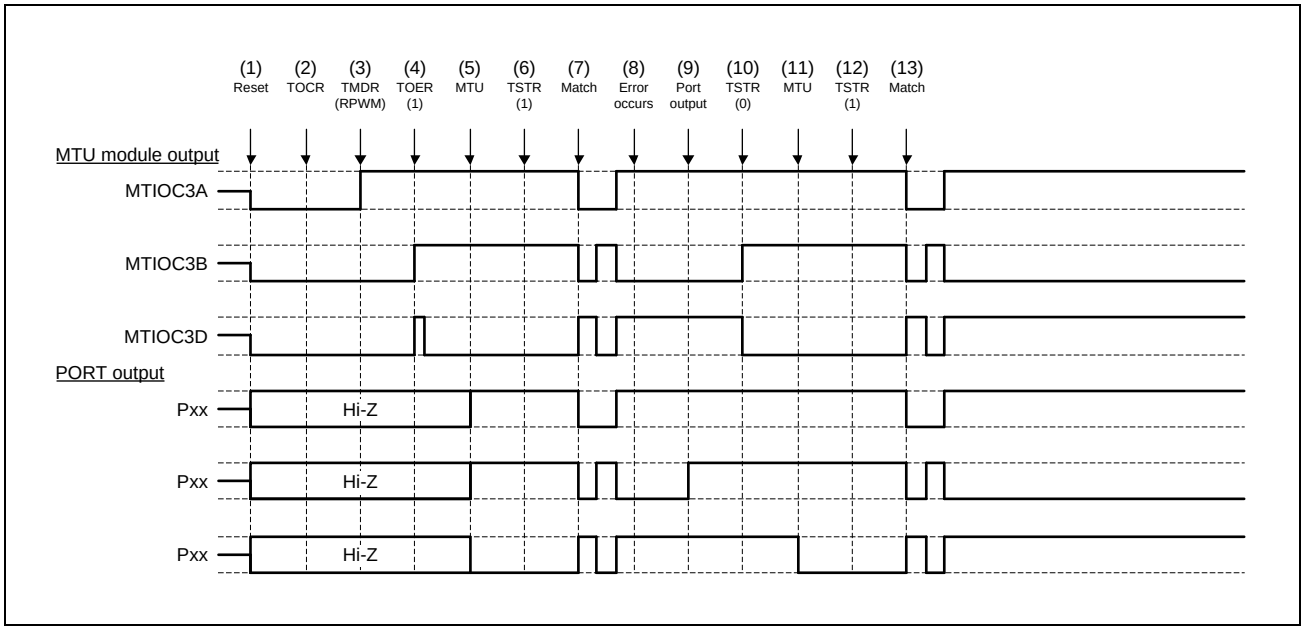



Figure 16.182 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Reset-Synchronized PWM Mode

(1) to (10) are the same as in **Figure 16.179**.

(11) Set MTU output using the port mode control registers (PMCn) and Port function control registers (PFCm) corresponding to the GPIO ports.

(12) Restart operation by setting TSTRA (TSTRB).

(13) The reset-synchronized PWM waveform is output on compare match occurrence.

# 17. Port Output Enable 3 (POE3)

The port output enable 3 (POE3) can be used to place output pins for the MTU3a in the high-impedance state in response to various conditions.

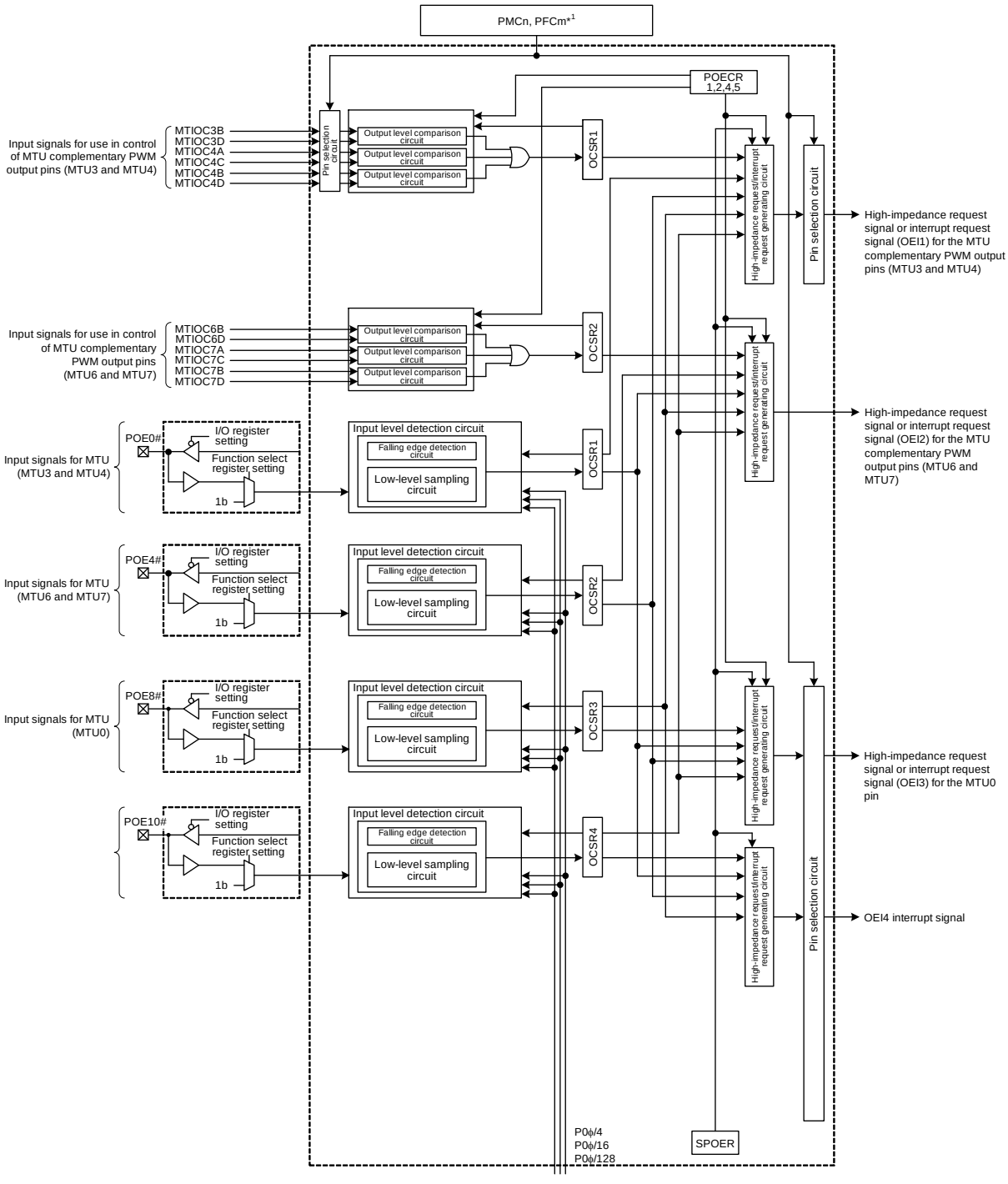
## 17.1 Overview

**Table 17.1** lists the specifications of the POE3, and **Figure 17.1** shows a block diagram of the POE3.

Table 17.1 POE3 Specifications

| Item | Description |
|---|---|
| Target pins to be placed in the high-impedance state | • MTU3a output pins<br>MTU0 pin (MTIOC0A, MTIOC0B, MTIOC0C, MTIOC0D)<br>MTU3 pin (MTIOC3B, MTIOC3D)<br>MTU4 pin (MTIOC4A, MTIOC4B, MTIOC4C, MTIOC4D)<br>MTU6 pin (MTIOC6B, MTIOC6D)<br>MTU7 pin (MTIOC7A, MTIOC7B, MTIOC7C, MTIOC7D) |
| Conditions for the high-impedance state | • Setting pins as inputs: setting the POE0#, POE4#, POE8#, or POE10# pins as inputs (falling edge or low- level sampling).<br>• Short-circuits between output pins: A match (short circuit) between the output signal levels at the active level over one or more cycle on (the following combination of pins)<br>1: MTIOC3B and MTIOC3D<br>2: MTIOC4A and MTIOC4C<br>3: MTIOC4B and MTIOC4D<br>4: MTIOC6B and MTIOC6D<br>5: MTIOC7A and MTIOC7C<br>6: MTIOC7B and MTIOC7D<br>• SPOER register setting being made |
| Function | • Each of the POE0#, POE4#, POE8#, and POE10# input pins can be set for falling edge, Pϕ/4 × 16, Pϕ/ 16 × 16, or Pϕ/128 × 16 low-level sampling.<br>• Pins for the MTU complementary PWM output, and MTU0 pin can be placed in high-impedance state by POE0#, POE4#, POE8#, and POE10# pin falling-edge or low-level sampling.<br>• Pins for the MTU complementary PWM output can be placed in high-impedance state when output levels of the MTU complementary PWM output pins are compared and simultaneous active-level output continues for one cycle or more.<br>• Pins for the MTU complementary PWM output, and MTU0 can be placed in the high-impedance state by modifying the settings of the SPOER register of POE3.<br>• Interrupts can be generated by input-level sampling or output level comparison results. |

The POE3 has input level detection circuits, pin selection circuits, output level comparison circuits, and a high-impedance request/interrupt request generating circuit as shown in **Figure 17.1**.




Note 1. Refer to chapter of GPIO.

Figure 17.1 POE3 Block Diagram

**Table 17.2** shows input/output pins to be used by the POE3.

Table 17.2 POE3 Input/Output Pins

| Pin Name | I/O | Description |
|---|---|---|
| POE0# | Input | Input pin for the request signal to place the MTU3 and MTU4 pins for MTU complementary PWM output in high-impedance state. In accord with register settings, this pin is also capable of placing the MTU0, MTU6, and MTU7 pins in the high-impedance state. |
| POE4# | Input | Input pin for the request signal to place the MTU6 and MTU7 pins for MTU complementary PWM output in high-impedance state. In accord with register settings, this pin is also capable of placing the MTU0, MTU3, and MTU4 pins in the high-impedance state. |
| POE8# | Input | Input pin for the request signal to place the pins for MTU0 in high-impedance state. In accord with register settings, this pin is also capable of placing the MTU3 and MTU4 pins or MTU6 and MTU7 pins for MTU complementary PWM output in high-impedance state. |
| POE10# | Input | In accord with register settings, this pin is also capable of placing the MTU3 and MTU4 pins or MTU6 and MTU7 pins for MTU complementary PWM output, and MTU0 pins in high-impedance state. |

**Table 17.3** shows output level comparisons with pin combinations.

Table 17.3 Pin Combinations

| Pin Combination | I/O | Description |
|---|---|---|
| MTIOC3B and MTIOC3D | Output | The MTU3 and MTU4 pins for MTU complementary PWM output set in the M3SELR, M4SELR1, and M4SELR2 registers are placed in high-impedance state when both pins of a pair simultaneously output the active level*1 for one or more cycles of the POE3_CLKM_POE (P0ϕ). Pin combinations for output comparison and high-impedance control can be selected by registers of POE3.<br>*Note 1.* The low level is output when the OLSP bit in TOCR1A of MTUn is 0 with the TOCS bit in TOCR1A of MTUn cleared to 0, or the high level is output when the OLSP bit is 1. Otherwise, the low level is output when the OLS3N, OLS3P, OLS2N, OLS2P, OLS1N, and OLS1P bits in TOCR2A of MTUn are 0 with the TOCS bit in TOCR1A of MTUn set to 1, or the high level is output when these bits are 1. |
| MTIOC4A and MTIOC4C | Output | |
| MTIOC4B and MTIOC4D | Output | |
| MTIOC6B and MTIOC6D | Output | The MTU6 and MTU7 pins for MTU complementary PWM output are placed in high-impedance state when both pins of a pair simultaneously output the active level*1 for one or more cycles of the POE3_CLKM_POE (P0ϕ). Pin combinations for output comparison and high-impedance control can be selected by registers of POE3.<br>*Note 1.* The low level is output when the OLSP bit in TOCR1B of MTUn is 0 with the TOCS bit in TOCR1B of MTUn cleared to 0, or the high level is output when the OLSP bit is 1. Otherwise, the low level is output when the OLS3N, OLS3P, OLS2N, OLS2P, OLS1N, and OLS1P bits in TOCR2B of MTUn are 0 with the TOCS bit in TOCR1B of MTUn set to 1, or the high level is output when these bits are 1. |
| MTIOC7A and MTIOC7C | Output | |
| MTIOC7B and MTIOC7D | Output | |

## 17.2 Register Descriptions

**Table 17.4** shows the register configuration. The POE3 registers are initialized by a reset.

Base Address: H'0_1004_9800 (Cortex-A55 Address Space)

Table 17.4 Register configuration

| Register Name | Abbreviation | Offset Address | Access size |
|---|---|---|---|
| Input level control/status register 1 | ICSR1 | H'00 | 16 |
| Output level control/status register 1 | OCSR1 | H'02 | 16 |
| Input level control/status register 2 | ICSR2 | H'04 | 16 |
| Output level control/status register 2 | OCSR2 | H'06 | 16 |
| Input level control/status register 3 | ICSR3 | H'08 | 16 |
| Software port output enable register | SPOER | H'0A | 8 |
| Port output enable control register 1 | POECR1 | H'0B | 8 |
| Port output enable control register 2 | POECR2 | H'0C | 16 |
| Port output enable control register 4 | POECR4 | H'10 | 16 |
| Port output enable control register 5 | POECR5 | H'12 | 16 |
| Input level control/status register 4 | ICSR4 | H'16 | 16 |

### 17.2.1 Input Level Control/Status Register 1 (ICSR1)

ICSR1 selects the input modes for the POE0# pins, controls the enable/disable of interrupts, and indicates status.

**Address(es):** H'1004_9800

| Bit | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | POE0F | — | — | — | PIE1 | — | — | — | — | — | — | POE0M[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)*2 | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W*1 | R/W*1 |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| b1, b0 | POE0M[1:0] | All 0 | R/W*1 | POE0 Mode Select<br>b1 b0<br>0 0: Accepts a request on the falling edge of POE0# input.<br>0 1: Accepts a request when POE0# input has been sampled 16 times at Pϕ/4 clock pulses and all are low level.<br>1 0: Accepts a request when POE0# input has been sampled 16 times at Pϕ/16 clock pulses and all are low level.<br>1 1: Accepts a request when POE0# input has been sampled 16 times at Pϕ/128 clock pulses and all are low level. |
| b7 to b2 | — | All 0 | R/W | Reserved<br>These bits are read as 0. The write value should be 0. |
| b8 | PIE1 | 0 | R/W | Port Interrupt Enable 1<br>0: Interrupt requests disabled<br>1: Interrupt requests enabled |
| b11 to b9 | — | All 0 | R/W | Reserved<br>These bits are read as 0. The write value should be 0. |
| b12 | POE0F | 0 | R/(W)*2 | POE0 Flag<br>0: Indicates that a high-impedance request has not been input to the POE0# pin.<br>1: Indicates that a high-impedance request has been input to the POE0# pin. |
| b15 to b13 | — | All 0 | R/W | Reserved<br>These bits are read as 0. The write value should be 0. |

Note 1. Can be modified only once after a reset. After the bits are modified once, writing these bits does not modify their values until the register is reset.

Note 2. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

#### POE0M[1:0] Bits (POE0 Mode Select)

These bits select the input mode of the POE0# pin.

#### PIE1 Bit (Port Interrupt Enable 1)

This bit enables or disables interrupt requests when any one of the POE0F bit of the ICSR1 is set to 1.

#### POE0F Flag (POE0 Flag)

This flag indicates that a high-impedance request has been input to the POE0# pin.

[Setting condition]

- When the input set by POE0M[1:0] occurs at the POE0# pin

[Clearing condition]

- By writing 0 to POE0F after reading POE0F = 1

## 17.2.2 Input Level Control/Status Register 2 (ICSR2)

ICSR2 selects the input mode for the POE4# pin, controls the enable/disable of interrupts, and indicates status.

**Address(es):** H'1004_9804

| Bit | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | POE4F | — | — | — | PIE2 | — | — | — | — | — | — | POE4M[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)*2 | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W*1 | R/W*1 |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| b1, b0 | POE4M[1:0] | All 0 | R/W*1 | POE4 Mode Select<br>b1 b0<br>0 0: Accepts a request on the falling edge of POE4# input<br>0 1: Accepts a request when POE4# input has been sampled 16 times at P0ϕ/4 clock pulses and all are low level.<br>1 0: Accepts a request when POE4# input has been sampled 16 times at P0ϕ/16 clock pulses and all are low level.<br>1 1: Accepts a request when POE4# input has been sampled 16 times at P0ϕ/128 clock pulses and all are low level. |
| b7 to b2 | — | All 0 | R/W | Reserved<br>These bits are read as 0. The write value should be 0. |
| b8 | PIE2 | 0 | R/W | Port Interrupt Enable 2<br>0: Interrupt requests disabled<br>1: Interrupt requests enabled |
| b11 to b9 | — | All 0 | R/W | Reserved<br>These bits are read as 0. The write value should be 0. |
| b12 | POE4F | 0 | R/(W)*2 | POE4 Flag<br>0: Indicates that a high-impedance request has not been input to the POE4# pin.<br>1: Indicates that a high-impedance request has been input to the POE4# pin. |
| b15 to b13 | — | All 0 | R/W | Reserved<br>These bits are read as 0. The write value should be 0. |

Note 1. Can be modified only once after a reset. After the bits are modified once, writing these bits does not modify their values until the register is reset.

Note 2. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

### POE4M[1:0] Bits (POE4 Mode Select)

These bits select the input mode of the POE4# pin.

### PIE2 Bit (Port Interrupt Enable 2)

This bit enables or disables interrupt requests when the POE4F flag in ICSR2 is set to 1.

### POE4F Flag (POE4 Flag)

This flag indicates that a high-impedance request has been input to the POE4# pin.

[Setting condition]

- When the input set by POE4M[1:0] occurs at the POE4# pin

[Clearing condition]

- By writing 0 to POE4F after reading POE4F = 1

### 17.2.3 Input Level Control/Status Register 3 (ICSR3)

ICSR3 selects the input mode for the POE8# pin, controls the enable/disable of interrupts, and indicates status.

**Address(es):** H'1004_9808

| Bit | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | POE8F | — | — | POE8E | PIE3 | — | — | — | — | — | — | POE8M[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)*2 | R/W | R/W | R/W*1 | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W*1 | R/W*1 |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| b1, b0 | POE8M[1:0] | All 0 | R/W*1 | POE8 Mode Select<br>b1 b0<br>0 0: Accepts a request on the falling edge of POE8# input<br>0 1: Accepts a request when POE8# input has been sampled 16 times at P0ϕ/4 clock pulses and all are low level.<br>1 0: Accepts a request when POE8# input has been sampled 16 times at P0ϕ/16 clock pulses and all are low level.<br>1 1: Accepts a request when POE8# input has been sampled 16 times at P0ϕ/128 clock pulses and all are low level. |
| b7 to b2 | — | All 0 | R/W | Reserved<br>These bits are read as 0. The write value should be 0. |
| b8 | PIE3 | 0 | R/W | Port Interrupt Enable 3<br>0: Interrupt requests disabled<br>1: Interrupt requests enabled |
| b9 | POE8E | 0 | R/W*1 | POE8 High- Impedance Enable<br>0: Does not place the pin in high-impedance state.<br>1: Places the pin in high-impedance state. |
| b11, b10 | — | All 0 | R/W | Reserved<br>These bits are read as 0. The write value should be 0. |
| b12 | POE8F | 0 | R/(W)*2 | POE8 Flag<br>0: Indicates that a high-impedance request has not been input to the POE8# pin.<br>1: Indicates that a high-impedance request has been input to the POE8# pin. |
| b15 to b13 | — | All 0 | R/W | Reserved<br>These bits are read as 0. The write value should be 0. |

Note 1. Can be modified only once after a reset. After the bits are modified once, writing these bits does not modify their values until the register is reset.

Note 2. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

#### POE8M[1:0] Bits (POE8 Mode Select)

These bits select the input mode of the POE8# pin.

#### PIE3 Bit (Port Interrupt Enable 3)

This bit enables or disables interrupt requests when the POE8F bit in ICSR3 is set to 1.

#### POE8E Bit (POE8 High-Impedance Enable)

This bit specifies whether to place the corresponding pin in high-impedance state when the POE8F bit is set to 1.

**POE8F Flag (POE8 Flag)**

This flag indicates that a high-impedance request has been input to the POE8# pin.

[Setting condition]

- When the input set by POE8M[1:0] occurs at the POE8# pin

[Clearing condition]

- By writing 0 to POE8F after reading POE8F = 1

## 17.2.4 Input Level Control/Status Register 4 (ICSR4)

ICSR4 selects the POE10# pin input mode, controls the enable/disable of interrupts, and indicates status.

**Address(es):** H'1004_9816

| Bit | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | POE10F | — | — | POE10E | PIE4 | — | — | — | — | — | — | POE10M[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)*2 | R/W | R/W | R/W*1 | R/W*1 | R/W | R/W | R/W | R/W | R/W | R/W | R/W*1 | R/W*1 |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| b1, b0 | POE10M[1:0] | All 0 | R/W*1 | POE10 Mode Select<br>b1 b0<br>0 0: Accepts a request on the falling edge of POE10# input<br>0 1: Accepts a request when POE10# input has been sampled 16 times at P0ϕ/4 clock pulses and all are low level.<br>1 0: Accepts a request when POE10# input has been sampled 16 times at P0ϕ/16 clock pulses and all are low level.<br>1 1: Accepts a request when POE10# input has been sampled 16 times at P0ϕ/128 clock pulses and all are low level. |
| b7 to b2 | — | All 0 | R/W | Reserved<br>These bits are read as 0. The write value should be 0. |
| b8 | PIE4 | 0 | R/W | Port Interrupt Enable 4<br>0: Interrupt requests disabled<br>1: Interrupt requests enabled |
| b9 | POE10E | 0 | R/W*1 | POE10 High- Impedance Enable<br>0: Does not place the pin in high-impedance state.<br>1: Places the pin in high-impedance state. |
| b11, b10 | — | All 0 | R/W | Reserved<br>These bits are read as 0. The write value should be 0. |
| b12 | POE10F | 0 | R/(W)*2 | POE10 Flag<br>0: Indicates that a high-impedance request has not been input to the POE10# pin.<br>1: Indicates that a high-impedance request has been input to the POE10# pin. |
| b15 to b13 | — | All 0 | R/W | Reserved<br>These bits are read as 0. The write value should be 0. |

Note 1. Can be modified only once after a reset. After the bits are modified once, writing these bits does not modify their values until the register is reset.

Note 2. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

### POE10M[1:0] Bits (POE10 Mode Select)

These bits select the input mode of the POE10# pin.

### PIE4 Bit (Port Interrupt Enable 4)

This bit enables or disables interrupt requests when the POE10F bit is set to 1.

### POE10E Bit (POE10 High-Impedance Enable)

This bit specifies whether to place the corresponding pin in high-impedance state when the POE10F bit is set to 1.

**POE10F Bit (POE10 Flag)**

This flag indicates that a request for the high-impedance state has been input to the POE10# pin.

[Setting condition]

- When the input set by POE10M[1:0] occurs at the POE10# pin

[Clearing condition]

- By writing 0 to POE10F after reading POE10F = 1

## 17.2.5 Output Level Control/Status Register 1 (OCSR1)

OCSR1 controls the enable/disable of output level comparison and interrupts, and indicates status.

**Address(es):** H'1004_9802

| Bit | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | OSF1 | — | — | — | — | — | OCE1 | OIE1 | — | — | — | — | — | — | — | — |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/(W)*2 | R/W | R/W | R/W | R/W | R/W | R/W*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 |
|---|---|---|---|---|
| b7 to b0 | — | All 0 | R/W | Reserved<br>These bits are read as 0. The write value should be 0. |
| b8 | OIE1 | 0 | R/W | Output Short Interrupt Enable 1<br>0: Interrupt requests disabled<br>1: Interrupt requests enabled |
| b9 | OCE1 | 0 | R/W*1 | Output Short High-Impedance Enable 1<br>0: Does not place the pins in high-impedance state.<br>1: Places the pins in high-impedance state. |
| b14 to b10 | — | All 0 | R/W | Reserved<br>These bits are read as 0. The write value should be 0. |
| b15 | OSF1 | 0 | R/(W)*2 | Output Short Flag 1<br>0: Indicates that outputs have not simultaneously become an active level.<br>1: Indicates that outputs have simultaneously become an active level. |

Note 1. Can be modified only once after a reset. After the bit is modified once, writing this bit does not modify its value until the register is reset.

Note 2. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

### OIE1 Bit (Output Short Interrupt Enable 1)

This bit enables or disables interrupt requests when the OSF1 bit in OCSR1 is set to 1.

### OCE1 Bit (Output Short High-Impedance Enable 1)

This bit specifies whether to place the pins in high-impedance state when the OSF1 bit in OCSR1 is set to 1.

### OSF1 Flag (Output Short Flag 1)

This flag indicates that any one of the three pairs of two-phase MTU3 and MTU4 pins for MTU complementary PWM output to be compared has simultaneously become an active level.

[Setting condition]

- When any one of the three pairs of two-phase outputs has simultaneously become an active level

[Clearing condition]

- By writing 0 to OSF1 after reading OSF1 = 1

## 17.2.6 Output Level Control/Status Register 2 (OCSR2)

OCSR2 controls the enable/disable of output level comparison and interrupts, and indicates status.

**Address(es):** H'1004_9806

| Bit | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | OSF2 | — | — | — | — | — | OCE2 | OIE2 | — | — | — | — | — | — | — | — |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/(W)*2 | R/W | R/W | R/W | R/W | R/W | R/W*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 |
|---|---|---|---|---|
| b7 to b0 | — | All 0 | R/W | Reserved<br>These bits are read as 0. The write value should be 0. |
| b8 | OIE2 | 0 | R/W | Output Short Interrupt Enable 2<br>0: Interrupt requests disabled<br>1: Interrupt requests enabled |
| b9 | OCE2 | 0 | R/W*1 | Output Short High-Impedance Enable 2<br>0: Does not place the pins in high-impedance state.<br>1: Places the pins in high-impedance state. |
| b14 to b10 | — | All 0 | R/W | Reserved<br>These bits are read as 0. The write value should be 0. |
| b15 | OSF2 | 0 | R/(W)*2 | Output Short Flag 2<br>0: Indicates that outputs have not simultaneously become an active level.<br>1: Indicates that outputs have simultaneously become an active level. |

Note 1. Can be modified only once after a reset. After the bit is modified once, writing this bit does not modify its value until the register is reset.

Note 2. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

### OIE2 Bit (Output Short Interrupt Enable 2)

This bit enables or disables interrupt requests when the OSF2 bit in OCSR2 is set to 1.

### OCE2 Bit (Output Short High-Impedance Enable 2)

This bit specifies whether to place the pins in high-impedance state when the OSF2 bit in OCSR2 is set to 1.

### OSF2 Flag (Output Short Flag 2)

This flag indicates that any one of the three pairs of two-phase MTU6 and MTU7 pins for MTU complementary PWM output to be compared has simultaneously become an active level.

[Setting condition]

- When any one of the three pairs of two-phase outputs has simultaneously become an active level

[Clearing condition]

- By writing 0 to OSF2 after reading OSF2 = 1

### 17.2.7 Software Port Output Enable Register (SPOER)

SPOER controls high-impedance state of the pins.

**Address(es):** H'1004_980A

| Bit | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | MTUCH0HIZ | MTUCH67HIZ | MTUCH34HIZ |
| 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 |
|---|---|---|---|---|
| b0 | MTUCH34HIZ | 0 | R/W | MTU3 or MTU4 Output High-Impedance Enable<br>0: Does not place the pins in high-impedance state.<br>1: Places the pins in high-impedance state. |
| b1 | MTUCH67HIZ | 0 | R/W | MTU6 and MTU7 Output High- Impedance Enable<br>0: Does not place the pins in high-impedance state.<br>1: Places the pins in high-impedance state. |
| b2 | MTUCH0HIZ | 0 | R/W | MTU0 Output High-Impedance Enable<br>0: Does not place the pins in high-impedance state.<br>1: Places the pins in high-impedance state. |
| b7 to b3 | — | All 0 | R/W | Reserved<br>These bits are read as 0. The write value should be 0. |

#### MTUCH34HIZ Bit (MTU3 or MTU4 Output High-Impedance Enable)

This bit specifies whether to place the MTU complementary PWM output pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4B, MTIOC4C, MTIOC4D) in high-impedance state.

[Setting condition]

- By writing 1 to MTUCH34HIZ

[Clearing conditions]

- Reset
- By writing 0 to MTUCH34HIZ after reading MTUCH34HIZ = 1*[1]

**Note 1.** To write 0 to this bit, be sure to read 1 and then write 0.

#### MTUCH67HIZ Bit (MTU6 and MTU7 Output High-Impedance Enable)

This bit specifies whether to place the MTU complementary PWM output pins (MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7B, MTIOC7C, and MTIOC7D) in high-impedance state.

[Setting condition]

- By writing 1 to MTUCH67HIZ

[Clearing conditions]

- Reset
- By writing 0 to MTUCH67HIZ after reading MTUCH67HIZ = 1*[1]

**Note 1.** To write 0 to this bit, be sure to read 1 and then write 0.

**MTUCH0HIZ Bit (MTU0 Output High-Impedance Enable)**

This bit specifies whether to place the MTU0 pins in high-impedance state.

[Setting condition]

- By writing 1 to MTUCH0HIZ

[Clearing conditions]

- Reset
- By writing 0 to MTUCH0HIZ after reading MTUCH0HIZ = 1*1

**Note 1.** To write 0 to this bit, be sure to read 1 and then write 0.

## 17.2.8 Port Output Enable Control Register 1 (POECR1)

POECR1 controls high-impedance state of the MTU0 pins.

**Address(es):** H'1004_980B

| Bit | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | MTU0DZE | MTU0CZE | MTU0BZE | MTU0AZE |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R/W | R/W*1 | R/W*1 | R/W*1 | R/W*1 |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| b0 | MTU0AZE | 0 | R/W*1 | MTIOC0A High-Impedance Enable<br>0: Does not place the pin in high-impedance state.<br>1: Places the pin in high-impedance state. |
| b1 | MTU0BZE | 0 | R/W*1 | MTIOC0B High-Impedance Enable<br>0: Does not place the pin in high-impedance state.<br>1: Places the pin in high-impedance state. |
| b2 | MTU0CZE | 0 | R/W*1 | MTIOC0C High-Impedance Enable<br>0: Does not place the pin in high-impedance state.<br>1: Places the pin in high-impedance state. |
| b3 | MTU0DZE | 0 | R/W*1 | MTIOC0D High-Impedance Enable<br>0: Does not place the pin in high-impedance state.<br>1: Places the pin in high-impedance state. |
| b7 to b4 | — | All 0 | R/W | Reserved<br>These bits are read as 0. The write value should be 0. |

Note 1. Can be modified only once after a reset. After the bit is modified once, writing this bit does not modify its value until the register is reset.

### MTU0AZE Bit (MTIOC0A High-Impedance Enable)

This bit specifies whether to place the MTIOC0A output for the MTU0 pin in high-impedance state when any of the POE8F flag in ICSR3, MTUCH0HIZ bit in SPOER or, as additionally specified in POECR5, the POEmF (n = 1, 2, 4; m = 0, 4, 10) flag in ICSRn, is set to 1.

### MTU0BZE Bit (MTIOC0B High-Impedance Enable)

This bit specifies whether to place the MTIOC0B output for the MTU0 pin in high-impedance state when any of the POE8F flag in ICSR3, MTUCH0HIZ bit in SPOER or, as additionally specified in POECR5, the POEmF (n = 1, 2, 4; m = 0, 4, 10) flag in ICSRn, is set to 1.

### MTU0CZE Bit (MTIOC0C High-Impedance Enable)

This bit specifies whether to place the MTIOC0C output for the MTU0 pin in high-impedance state when any of the POE8F flag in ICSR3, MTUCH0HIZ bit in SPOER or, as additionally specified in POECR5, the POEmF (n = 1, 2, 4; m = 0, 4, 10) flag in ICSRn, is set to 1.

### MTU0DZE Bit (MTIOC0D High-Impedance Enable)

This bit specifies whether to place the MTIOC0D output for the MTU0 pin in high-impedance state when any of the POE8F flag in ICSR3, MTUCH0HIZ bit in SPOER or, as additionally specified in POECR5, the POEmF (n = 1, 2, 4; m = 0, 4, 10) flag in ICSRn, is set to 1.

## 17.2.9 Port Output Enable Control Register 2 (POECR2)

POECR2 controls high-impedance state of the MTU complementary PWM output pins (MTU3, MTU4, MTU6, and MTU7 pins).

**Address(es):** H'1004_980C

| Bit | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | MTU3BDZE | MTU4ACZE | MTU4BDZE | — | — | — | — | — | MTU6BDZE | MTU7ACZE | MTU7BDZE |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W*1 | R/W*1 | R/W*1 | R/W | R/W | R/W | R/W | R/W | R/W*1 | R/W*1 | R/W*1 |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| b0 | MTU7BDZE | 1 | R/W*1 | MTIOC7B/7D High-Impedance Enable*2<br>0: Does not place the pins in high-impedance state.<br>1: Places the pins in high-impedance state. |
| b1 | MTU7ACZE | 1 | R/W*1 | MTIOC7A/7C High-Impedance Enable*2<br>0: Does not place the pins in high-impedance state.<br>1: Places the pins in high-impedance state. |
| b2 | MTU6BDZE | 1 | R/W*1 | MTIOC6B/6D High-Impedance Enable*2<br>0: Does not place the pins in high-impedance state.<br>1: Places the pins in high-impedance state. |
| b7 to b3 | — | All 0 | R/W | Reserved<br>These bits are read as 0. The write value should be 0. |
| b8 | MTU4BDZE | 1 | R/W*1 | MTIOC4B/4D High-Impedance Enable<br>0: Does not place the pins in high-impedance state.<br>1: Places the pins in high-impedance state. |
| b9 | MTU4ACZE | 1 | R/W*1 | MTIOC4A/4C High-Impedance Enable<br>0: Does not place the pins in high-impedance state.<br>1: Places the pins in high-impedance state. |
| b10 | MTU3BDZE | 1 | R/W*1 | MTIOC3B/3D High-Impedance Enable<br>0: Does not place the pins in high-impedance state.<br>1: Places the pins in high-impedance state. |
| b15 to b11 | — | All 0 | R/W | Reserved<br>These bits are read as 0. The write value should be 0. |

Note 1. Can be modified only once after a reset. After the bit is modified once, writing this bit does not modify its value until the register is reset.

Note 2. Set this bit to 0 when MTU6 or MTU7 is not used.

### MTU7BDZE Bit (MTIOC7B/7D High-Impedance Enable)

This bit specifies whether to place the MTIOC7B output and MTIOC7D output for the MTU3 pin in high-impedance state when any one of the OSF2 flag in OCSR2, POE4F flag in ICSR2, MTUCH67HIZ bit in SPOER, OSTSTF flag in ICSR6 (when the OSTSTE bit is 1), or, as additionally specified in POECR4, the POEmF (n = 1, 3, 4; m = 0, 8, 10) flag in ICSRn, is set to 1.

### MTU7ACZE Bit (MTIOC7A/7C High-Impedance Enable)

This bit specifies whether to place the MTIOC7A output and MTIOC7C output for the MTU3 pin in high-impedance state when any one of the OSF2 flag in OCSR2, POE4F flag in ICSR2, MTUCH67HIZ bit in SPOER, OSTSTF flag in ICSR6 (when the OSTSTE bit is 1), or, as additionally specified in POECR4, the POEmF (n = 1, 3, 4; m = 0, 8, 10) flag in ICSRn, is set to 1.

#### MTU6BDZE Bit (MTIOC6B/6D High-Impedance Enable)

This bit specifies whether to place the MTIOC6B output and MTIOC6D output for the MTU3 pin in high-impedance state when any one of the OSF2 flag in OCSR2, POE4F flag in ICSR2, MTUCH67HIZ bit in SPOER, OSTSTF flag in ICSR6 (when the OSTSTE bit is 1), or, as additionally specified in POECR4, the POEmF (n = 1, 3, 4; m = 0, 8, 10) flag in ICSRn, is set to 1.

#### MTU4BDZE Bit (MTIOC4B/4D High-Impedance Enable)

This bit specifies whether to place the MTIOC4B output and MTIOC4D output for the MTU4 pin in high-impedance state when any one of the OSF1 flag in OCSR1, POE0F flag in ICSR1, MTUCH34HIZ bit in SPOER, OSTSTF flag in ICSR6 (when the OSTSTE bit is 1), or, as additionally specified in POECR4, the POEmF (n = 2 to 4; m = 4, 8, 10) flag in ICSRn, is set to 1.

#### MTU4ACZE Bit (MTIOC4A/4C High-Impedance Enable)

This bit specifies whether to place the MTIOC4A output and MTIOC4C output for the MTU4 pin in high-impedance state when any one of the OSF1 flag in OCSR1, POE0F flag in ICSR1, MTUCH34HIZ bit in SPOER, OSTSTF flag in ICSR6 (when the OSTSTE bit is 1), or, as additionally specified in POECR4, the POEmF (n = 2 to 4; m = 4, 8, 10) flag in ICSRn, is set to 1.

#### MTU3BDZE Bit (MTIOC3B/3D High-Impedance Enable)

This bit specifies whether to place the MTIOC3B output and MTIOC3D output for the MTU3 pin in high-impedance state when any one of the OSF1 flag in OCSR1, POE0F flag in ICSR1, MTUCH34HIZ bit in SPOER, OSTSTF flag in ICSR6 (when the OSTSTE bit is 1), or, as additionally specified in POECR4, the POEmF (n = 2 to 4; m = 4, 8, 10) flag in ICSRn, is set to 1.

## 17.2.10 Port Output Enable Control Register 4 (POECR4)

The POECR4 is used to extend the control conditions of the high-impedance state for the MTU3, MTU4, MTU6, and MTU7 pins for the MTU complementary PWM output.

For details about the targets and conditions of high-impedance control, see **Figure 17.2**.

**Address(es):** H'1004_9810

| Bit | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | IC4ADDMT67 | IC3ADDMT67 | — | IC1ADDMT67 | — | — | — | — | IC4ADDMT34 | IC3ADDMT34 | IC2ADDMT34 | — | — |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| R/W | R/W | R/W | R/W | R/W*1 | R/W*1 | R/W | R/W*1 | R/W | R/W | R/W | R/W | R/W*1 | R/W*1 | R/W*1 | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| b0 | — | 0 | R/W | Reserved<br>This bit is read as 0. The write value should be 0. |
| b1 | — | 1 | R/W | Reserved<br>This bit is read as 1. The write value should be 1. |
| b2 | IC2ADDMT34ZE | 0 | R/W*1 | MTU3 and MTU4 High- Impedance POE4F Add<br>0: Does not add the pins to the high-impedance control conditions.<br>1: Adds the pins to the high-impedance control conditions. |
| b3 | IC3ADDMT34ZE | 0 | R/W*1 | MTU3 and MTU4 High- Impedance POE8F Add<br>0: Does not add the pins to the high-impedance control conditions.<br>1: Adds the pins to the high-impedance control conditions. |
| b4 | IC4ADDMT34ZE | 0 | R/W*1 | MTU3 and MTU4 High- Impedance POE10F Add<br>0: Does not add the pins to the high-impedance control conditions.<br>1: Adds the pins to the high-impedance control conditions. |
| b8 to b5 | — | All 0 | R/W | Reserved<br>These bits are read as 0. The write value should be 0 |
| b9 | IC1ADDMT67ZE | 0 | R/W*1 | MTU6 and MTU7 High- Impedance POE0F Add<br>0: Does not add the pins to the high-impedance control conditions.<br>1: Adds the pins to the high-impedance control conditions. |
| b10 | — | 1 | R/W | Reserved<br>This bit is read as 1. The write value should be 1. |
| b11 | IC3ADDMT67ZE | 0 | R/W*1 | MTU6 and MTU7 High- Impedance POE8F Add<br>0: Does not add the pins to the high-impedance control conditions.<br>1: Adds the pins to the high-impedance control conditions. |
| b12 | IC4ADDMT67ZE | 0 | R/W*1 | MTU6 and MTU7 High- Impedance POE10F Add<br>0: Does not add the pins to the high-impedance control conditions.<br>1: Adds the pins to the high-impedance control conditions. |
| b15 to b13 | — | All 0 | R/W | Reserved<br>These bits are read as 0. The write value should be 0. |

Note 1. Can be modified only once after a reset. After the bit is modified once, writing this bit does not modify its value until the register is reset.

#### IC2ADDMT34ZE Bit (MTU3 and MTU4 High-Impedance POE4F Add)

Adds the POE4F flag in ICSR2 (POE4#) to the high-impedance control conditions for the MTU3 and MTU4 pins (MTIOC3B/MTIOC3D/MTIOC4A/MTIOC4C/MTIOC4B/MTIOC4D).

#### IC3ADDMT34ZE Bit (MTU3 and MTU4 High-Impedance POE8F Add)

Adds the POE8F flag in ICSR3 (POE8#) to the high-impedance control conditions for the MTU3 and MTU4 pins (MTIOC3B/MTIOC3D/MTIOC4A/MTIOC4C/MTIOC4B/MTIOC4D).

#### IC4ADDMT34ZE Bit (MTU3 and MTU4 High-Impedance POE10F Add)

Adds the POE10F flag in ICSR4 (POE10#) to the high-impedance control conditions for the MTU3 and MTU4 pins (MTIOC3B/MTIOC3D/MTIOC4A/MTIOC4C/MTIOC4B/MTIOC4D).

#### IC1ADDMT67ZE Bit (MTU6 and MTU7 High-Impedance POE0F Add)

Adds the POE0F flag in ICSR1 (POE0#) to the high-impedance control conditions for the MTU6, and MTU7 pins (MTIOC6B/MTIOC6D/MTIOC7A/MTIOC7C/MTIOC7B/MTIOC7D).

#### IC3ADDMT67ZE Bit (MTU6 and MTU7 High-Impedance POE8F Add)

Adds the POE8F flag in ICSR3 (POE8#) to the high-impedance control conditions for the MTU6, and MTU7 pins (MTIOC6B/MTIOC6D/MTIOC7A/MTIOC7C/MTIOC7B/MTIOC7D).

#### IC4ADDMT67ZE Bit (MTU6 and MTU7 High-Impedance POE10F Add)

Adds the POE10F flag in ICSR4 (POE10#) to the high-impedance control conditions for the MTU6, and MTU7 pins (MTIOC6B/MTIOC6D/MTIOC7A/MTIOC7C/MTIOC7B/MTIOC7D).

## 17.2.11 Port Output Enable Control Register 5 (POECR5)

The POECR5 is used to extend the control conditions of the high-impedance for the MTU0 pin.

For details about the targets and conditions of high-impedance control, see **Figure 17.2**.

**Address(es):** H'1004_9812

| Bit | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | IC4ADDMT0ZE | — | IC2ADDMT0ZE | IC1ADDMT0ZE | — |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 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*1 | R/W | R/W*1 | R/W*1 | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| b0 | — | 0 | R/W | Reserved<br>This bit is read as 0. The write value should be 0. |
| b1 | IC1ADDMT0ZE | 0 | R/W*1 | MTU0 High-Impedance POE0F Add<br>0: Does not add the pins to the high-impedance control conditions.<br>1: Adds the pins to the high-impedance control conditions. |
| b2 | IC2ADDMT0ZE | 0 | R/W*1 | MTU0 High-Impedance POE4F Add<br>0: Does not add the pins to the high-impedance control conditions.<br>1: Adds the pins to the high-impedance control conditions. |
| b3 | — | 1 | R/W | Reserved<br>This bit is read as 1. The write value should be 1. |
| b4 | IC4ADDMT0ZE | 0 | R/W*1 | MTU0 High-Impedance POE10F Add<br>0: Does not add the pins to the high-impedance control conditions.<br>1: Adds the pins to the high-impedance control conditions. |
| b15 to b5 | — | All 0 | R/W | Reserved<br>These bits are read as 0. The write value should be 0. |

Note 1. Can be modified only once after a reset. After the bit is modified once, writing this bit does not modify its value until the register is reset.

### IC1ADDMT0ZE Bit (MTU0 High-Impedance POE0F Add)

Adds the POE0F flag in ICSR1 (POE0#) to the high-impedance control conditions for the MTU0 pins (MTIOC0A, MTIOC0B, MTIOC0C, MTIOC0D).

### IC2ADDMT0ZE Bit (MTU0 High-Impedance POE4F Add)

Adds the POE4F flag in ICSR2 (POE4#) to the high-impedance control conditions for the MTU0 pins (MTIOC0A, MTIOC0B, MTIOC0C, MTIOC0D).

### IC4ADDMT0ZE Bit (MTU0 High-Impedance POE10F Add)

Adds the POE10F flag in ICSR4 (POE10#) to the high-impedance control conditions for the MTU0 pins (MTIOC0A, MTIOC0B, MTIOC0C, MTIOC0D).

## 17.3 Operation

**Table 17.5** shows the target pins for high-impedance control and conditions to place the pins in high-impedance state.

Table 17.5 Target Pins and Conditions for High-Impedance Control (1/2)

| Pins | Conditions | Detailed Conditions |
|---|---|---|
| MTU3 pins (MTIOC3B and MTIOC3D) | • Operation for detection of the POE0# input level<br>• Operation for comparison of the output levels on the MTIOC3B and MTIOC3D pins<br>• SPOER setting<br>• Additional conditions of the POECR4 | MTU3BDZE•<br>((POE0F)<br>+(OSF1•OCE1)<br>+(MTUCH34HIZ)<br>+(IC2ADDMT34ZE•POE4F)<br>+(IC3ADDMT34ZE•POE8E•POE8F)<br>+(IC4ADDMT34ZE•ICSR4.POE10E•ICSR4.POE10F) |
| MTU4 pins (MTIOC4A and MTIOC4C) | • Operation for detection of the POE0# input level<br>• Operation for comparison of the output levels on the MTIOC4A and MTIOC4C pins<br>• SPOER setting<br>• Additional conditions of the POECR4 | MTU4ACZE•<br>((POE0F)<br>+(OSF1•OCE1)<br>+ (MTUCH34HIZ)<br>+(IC2ADDMT34ZE•POE4F)<br>+(IC3ADDMT34ZE•POE8E•POE8F)<br>+(IC4ADDMT34ZE•ICSR4.POE10E•ICSR4.POE10F) |
| MTU4 pins (MTIOC4B and MTIOC4D) | • Operation for detection of the POE0# input level<br>• Operation for comparison of the output levels on the MTIOC4B and MTIOC4D pins<br>• SPOER setting<br>• Additional conditions of the POECR4 | MTU4BDZE•<br>((POE0F)<br>+(OSF1•OCE1)<br>+(MTUCH34HIZ)<br>+(IC2ADDMT34ZE•POE4F)<br>+(IC3ADDMT34ZE•POE8E•POE8F)<br>+(IC4ADDMT34ZE•ICSR4.POE10E•ICSR4.POE10F) |
| MTU6 pins (MTIOC6B and MTIOC6D) | • Operation for detection of the POE4# input level<br>• Operation for comparison of the output levels on the MTIOC6B and MTIOC6D pins<br>• SPOER setting<br>• Additional conditions of the POECR4 | MTU6BDZE•<br>((POE4F)<br>+(OSF2•OCE2)<br>+(MTUCH67HIZ)<br>+(IC1ADDMT67ZE•POE0F)<br>+(IC3ADDMT67ZE•POE8E•POE8F)<br>+(IC4ADDMT67ZE•ICSR4.POE10E•ICSR4.POE10F) |
| MTU7 pins (MTIOC7A and MTIOC7C) | • Operation for detection of the POE4# input level<br>• Operation for comparison of the output levels on the MTIOC7A and MTIOC7C pins<br>• SPOER setting<br>• Additional conditions of the POECR4 | MTU7ACZE•<br>((POE4F)<br>+(OSF2•OCE2)<br>+(MTUCH67HIZ)<br>+(IC1ADDMT67ZE•POE0F)<br>+(IC3ADDMT67ZE•POE8E•POE8F)<br>+(IC4ADDMT67ZE•ICSR4.POE10E•ICSR4.POE10F) |
| MTU7 pins (MTIOC7B and MTIOC7D) | • Operation for detection of the POE4# input level<br>• Operation for comparison of the output levels on the MTIOC7B and MTIOC7D pins<br>• SPOER setting<br>• Additional conditions of the POECR4 | MTU7BDZE•<br>((POE4F)<br>+(OSF2•OCE2)<br>+(MTUCH67HIZ)<br>+(IC1ADDMT67ZE•POE0F)<br>+(IC3ADDMT67ZE•POE8E•POE8F)<br>+(IC4ADDMT67ZE•ICSR4.POE10E•ICSR4.POE10F) |
| MTU0 pin (MTIOC0A) | • Operation for detection of the POE8# input level<br>• SPOER setting<br>• Additional conditions of the POECR5 | MTU0AZE• ((POE8F•POE8E)<br>+(MTUCH0HIZ)<br>+(IC1ADDMT0ZE•POE0F)<br>+(IC2ADDMT0ZE•POE4F)<br>+(IC4ADDMT0ZE•ICSR4.POE10E•ICSR4.POE10F) |

Table 17.5 Target Pins and Conditions for High-Impedance Control (2/2)

| Pins | Conditions | Detailed Conditions |
|---|---|---|
| MTU0 pin (MTIOC0B) | • Operation for detection of the POE8# input level<br>• SPOER setting<br>• Additional conditions of the POECR5 | MTU0BZE• ((POE8F•POE8E)<br>+(MTUCH0HIZ)<br>+(IC1ADDMT0ZE•POE0F)<br>+(IC2ADDMT0ZE•POE4F)<br>+(IC4ADDMT0ZE•ICSR4.POE10E•ICSR4.POE10F) |
| MTU0 pin (MTIOC0C) | • Operation for detection of the POE8# input level<br>• SPOER setting<br>• Additional conditions of the POECR5 | MTU0CZE• ((POE8F•POE8E)<br>+(MTUCH0HIZ)<br>+(IC1ADDMT0ZE•POE0F)<br>+(IC2ADDMT0ZE•POE4F)<br>+(IC4ADDMT0ZE•ICSR4.POE10E•ICSR4.POE10F) |
| MTU0 pin (MTIOC0D) | • Operation for detection of the POE8# input level<br>• SPOER setting<br>• Additional conditions of the POECR5 | MTU0DZE• ((POE8F•POE8E)<br>+(MTUCH0HIZ)<br>+(IC1ADDMT0ZE•POE0F)<br>+(IC2ADDMT0ZE•POE4F)<br>+(IC4ADDMT0ZE•ICSR4.POE10E•ICSR4.POE10F) |

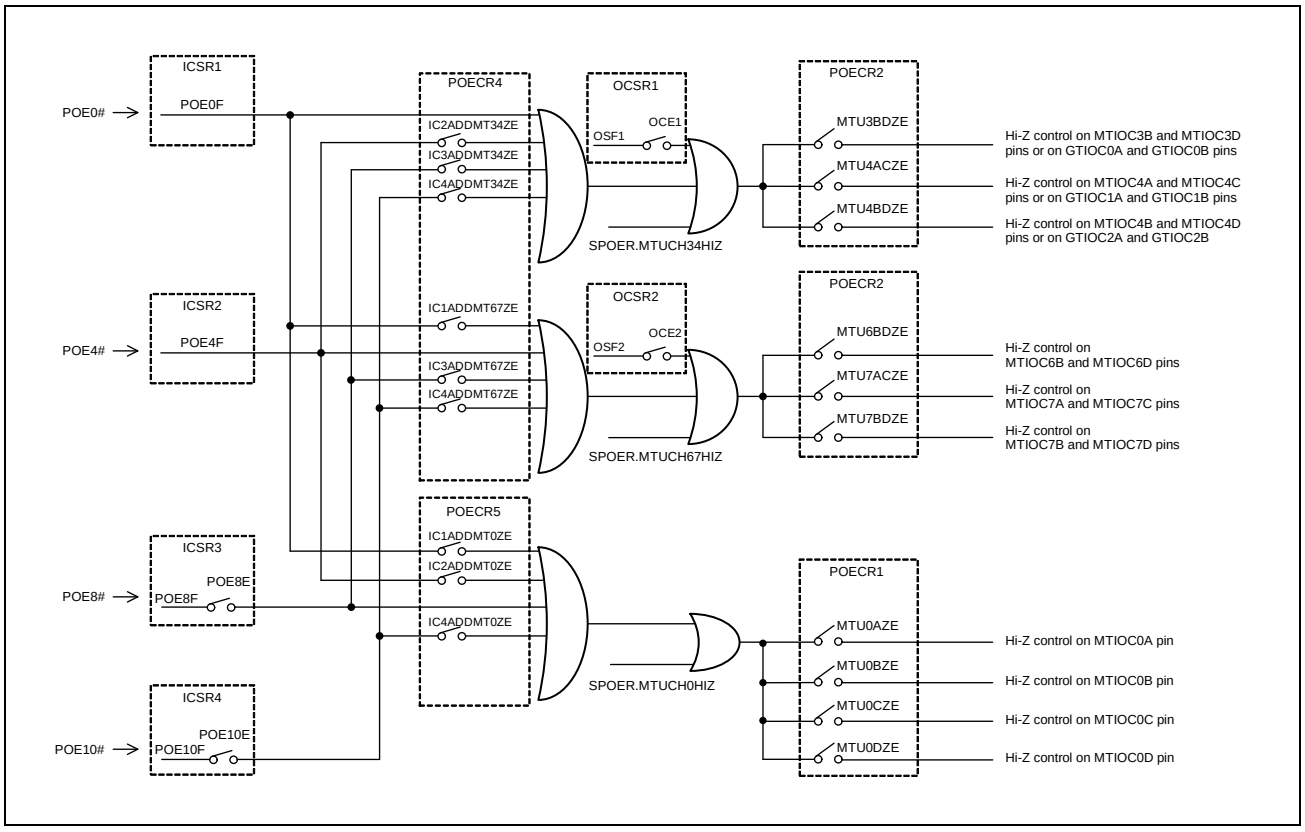



Figure 17.2 Target Pins and Conditions for High-Impedance Control

High-impedance requests to individual pins can be controlled by the settings in the POECR1 to POECR2 registers.

The following input pins can be added to high-impedance control conditions by the settings in the POECR4 to POECR5 registers: input pins other than the POE0# pin for the MTU3 and MTU4 pins; input pins other than the POE4# pin for the MTU6 and MTU7 pins; input pins other than the POE8# pin for the MTU0 pins.

For example, setting the IC2ADDMT34ZE bit in POECR4 to 1 outputs high-impedance requests to the MTU3 and MTU4 pins even when the POE4# input is detected.

High-impedance requests due to the detection of POE8# input or POE10# input can be controlled by the settings in the ICSR3 to ICSR4 registers. (The ICSR1 and ICSR2 registers do not control enabling or disabling of output of high-impedance requests.)

High-impedance requests to the MTU3, MTU4, MTU6, and MTU7 pins as the results of output level comparison can be controlled by the settings in the OCSR1 and OCSR2 registers.

### 17.3.1 MTU Pin Selection

In this LSI, each terminal function for MTU is assigned to multiple ports. **Table 17.6** shows the correspondence table of MTU terminal functions and compatible ports. The pin used as the MTU must be set separately in the general-purpose I / O port register. (Set by PMCn and PFCm registers of IO_TOP)

Table 17.6 Correspondence between MTU Pins (1/2)

| MTU Pin Functions | Corresponding Ports |
|---|---|
| MTCLKA | P1_1 |
| | P10_0 |
| MTCLKB | P1_2 |
| | P10_1 |
| MTCLKC | P1_3 |
| | P10_2 |
| MTCLKD | P1_4 |
| | P10_3 |
| MTIOC0A | P0_0 |
| | P10_0 |
| MTIOC0B | P0_1 |
| | P10_1 |
| MTIOC0C | P0_2 |
| | P10_2 |
| MTIOC0D | P0_3 |
| | P10_3 |
| MTIOC1A | P0_0 |
| | P7_1 |
| | P11_2 |
| | P20_1 |
| MTIOC1B | P0_1 |
| | P8_0 |
| | P11_3 |
| | P20_2 |
| MTIOC2A | P5_3 |
| | P8_1 |
| | P12_0 |
| | P20_3 |
| MTIOC2B | P5_4 |
| | P8_2 |
| | P12_1 |
| | P20_4 |
| MTIOC3A | P1_4 |
| MTIOC3B | P2_0 |
| MTIOC3C | P2_1 |
| MTIOC3D | P2_2 |
| MTIOC4A | P0_0 |
| MTIOC4B | P0_1 |
| MTIOC4C | P0_2 |
| MTIOC4D | P0_3 |

Table 17.6 Correspondence between MTU Pins (2/2)

| MTU Pin Functions | Corresponding Ports |
|---|---|
| MTIC5U | P2_0 |
| MTIC5V | P2_1 |
| MTIC5W | P2_2 |
| MTIOC6A | P3_1 |
| MTIOC6B | P3_2 |
| MTIOC6C | P3_3 |
| MTIOC6D | P4_0 |
| MTIOC7A | P4_1 |
| MTIOC7B | P4_2 |
| MTIOC7C | P4_5 |
| MTIOC7D | P5_0 |
| MTIOC8A | P5_1 |
| MTIOC8B | P5_2 |
| MTIOC8C | P5_3 |
| MTIOC8D | P5_4 |

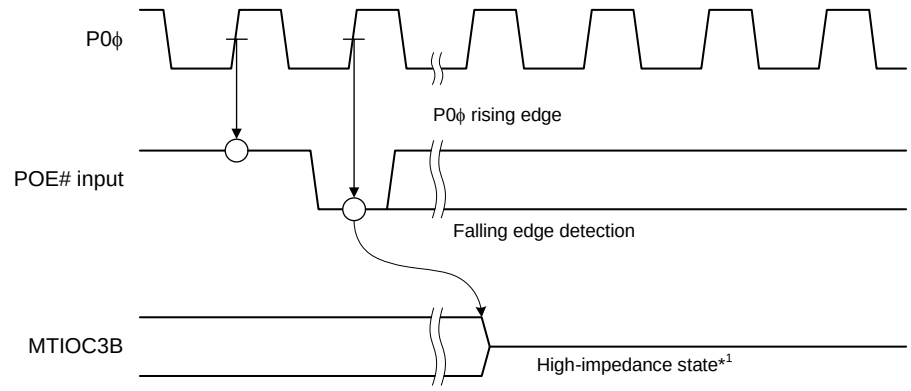
## 17.3.2 Input Level Detection Operation

If the input conditions set by ICSR1 to ICSR4 occur on the POE0#, POE4#, POE8#, and POE10# pins, the MTU3 and MTU4 or MTU6 and MTU7 pins for the MTU complementary PWM output, and MTU0 pin are placed in high-impedance state. Note however, that these pins are still placed in the high-impedance state even when the MTU functions are not selected for the pins.

### (1) Falling Edge Detection

When a change from a high to low level is input to the POE0#, POE4#, POE8#, and POE10# pins, the pins for the MTU complementary PWM output, and pin functions multiplexed with the MTU0 pins are placed in high-impedance state.

**Figure 17.3** shows a sample timing after the level changes in input to the POE0#, POE4#, POE8#, and POE10# pins until the respective pins enter high-impedance state.

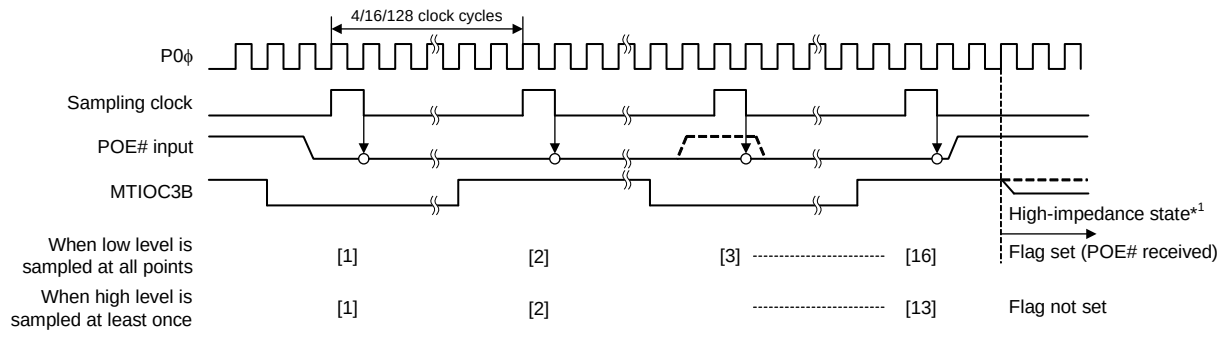



Note 1. Other pins also enter the high-impedance state at the same timing.

Figure 17.3 Falling Edge Detection

### (2) Low-Level Detection

**Figure 17.4** shows the low-level detection operation. Sixteen continuous low levels are sampled with the sampling clock selected by ICSR1 to ICSR4. If even one high level is detected during this interval, the low level is not accepted. The timing when pins for the MTU complementary PWM output, and MTU0 pins enter the high-impedance state after the sampling clock is input is the same in both falling-edge detection and in low-level detection.




Note 1. Other pins also enter the high-impedance state at the same timing.

Figure 17.4 Low-Level Detection Operation

### 17.3.3 Output Level Compare Operation

**Figure 17.5** shows an example of the output level compare operation for the combination of MTIOC3B and MTIOC3D. The operation is the same for the other pin combinations.

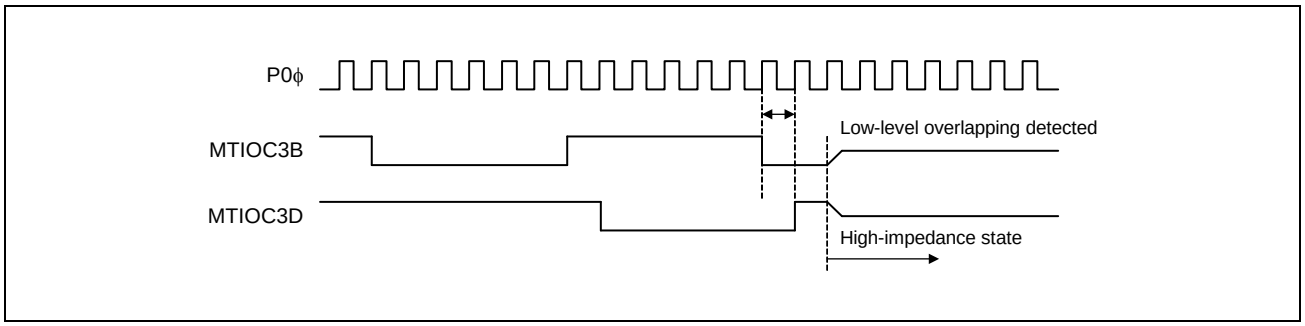



Figure 17.5 Output Level Compare Operation

### 17.3.4 High-Impedance Control Using Registers

The high-impedance state of the MTU pins (MTU0, MTU3, MTU4, MTU6, and MTU7) can be directly controlled by using the software port output enable register (SPOER).

For instance, setting the MTUCH34HIZ bit in SPOER to 1 places the MTU3 and MTU4 pins specified by the port output enable control register 2 (POECR2) in the high-impedance state.

The high-impedance state of other pins can also be controlled by setting the appropriate bits in SPOER.

### 17.3.5 Additional Functions for Controlling High-Impedance States

Settings in port enable registers 4 to 5 (POECR4 and POECR5) can add further high-impedance control conditions for the MTU complementary PWM output and MTU0 pins.

For instance, the settings listed below can be added as high-impedance control conditions for the MTU3 and MTU4 pins.

- Setting the IC2ADDMT34ZE bit in POECR4 to 1 adds the input level detection by the POE4#
- Setting the IC3ADDMT34ZE bit in POECR4 to 1 and adds the input level detection by the POE8#
- Setting the IC4ADDMT34ZE bit in POECR4 to 1 and adds the input level detection by the POE10# (ICSR4.POE10F)

The high-impedance state of other pins can also be controlled by setting the appropriate bits in the POECR4 and POECR5.

### 17.3.6 Release from High-Impedance State

MTU pins which have entered high-impedance state due to input-level detection can be released from the state either by returning them to their initial state with a reset, or by clearing all of the ICSR1.POE0F, ICSR2.POE4F, ICSR3.POE8F, and ICSR4.POE10F flags. However, note that when low-level sampling is selected with the ICSR1.POE0M[1:0], ICSR2.POE4M[1:0], ICSR3.POE8M[1:0], ICSR4.POE10M[1:0], and ICSR5.POE10M[1:0] bits, just writing 0 to a flag is ignored (the flag is not cleared); flags can be cleared by writing 0 to it only after a high level is input to the POE0#, POE4#, POE8#, and POE10# pins and is sampled.

MTU pins which have entered high-impedance state due to output level detection can be released from the state either by returning them to their initial state with a reset, or by clearing the OCSR1.OSF1 flag or the OCSR2.OSF2 flag.

## 17.4 POE3 Setting Procedure

**Figure 17.6** shows the POE3 configuration procedure. As an example, high impedance control by comparing the output levels of the MTU 3 terminals (MTIOC3B / MTIOC3D) is shown. In **Figure 17.6**, select P32_1 for the MTIOC3B terminal and P36_1 for the MTIOC3D terminal.

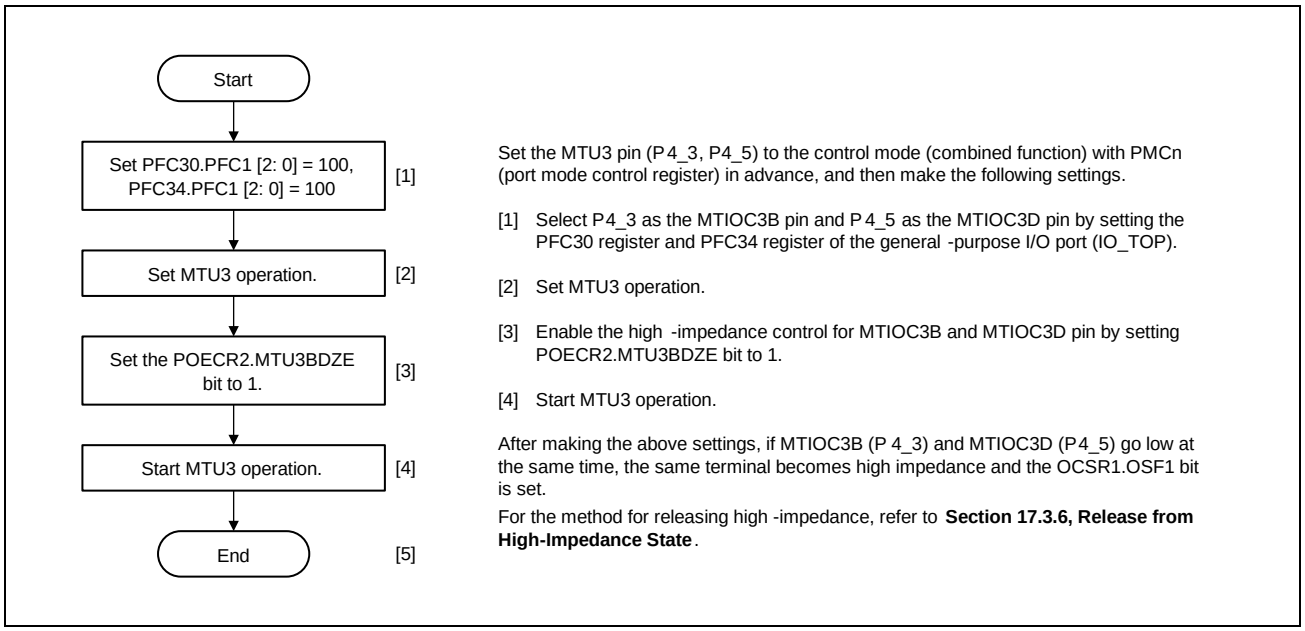



Figure 17.6 Procedure for Setting POE3

## 17.5 Interrupts

The POE3 issues a request to generate an interrupt when the specified condition is satisfied during input level detection or output level comparison. **Table 17.7** shows the interrupt sources and their conditions.

Table 17.7 Interrupt Sources and Conditions

| Name | Interrupt Source | Interrupt Flag | Condition |
|---|---|---|---|
| OEI1 | Output enable interrupt 1 | POE0F and OSF1 | PIE1•POE0F+OIE1•OSF1 |
| OEI2 | Output enable interrupt 2 | POE4F and OSF2 | PIE2•POE4F+OIE2•OSF2 |
| OEI3 | Output enable interrupt 3 | POE8F | PIE3•POE8F |
| OEI4 | Output enable interrupt 4 | ICSR4.POE10F | PIE4•ICSR4.POE10F |

## 17.6 Usage Notes

### 17.6.1 High-Impedance Control when MTU6 and MTU7 are not Used

When MTU6 and MTU7 are not to be used, set the POECR2.MTU6BDZE, MTU7ACZE, and MTU7BDZE bits to 0 to disable high-impedance control.

# 20. General Timer (GTM)

## 20.1 Functional Overview

The General timer has the following features.

- Two operating modes
  - Interval timer mode
  - Free-running comparison mode
- Choice between startup of DMA by compare match and generation of interrupt

### 20.1.1 Features of GTM

#### Channels

This product has the following number of channels of the General timer.

Table 20.1 Channels of General Timer

| General Timer | |
|---|---|
| Number of Channels | 3 |
| Name | OSTMn |

**Note:** n = 0, 1, 2

#### Meaning of n

Throughout this section, the individual channels of the General timer are identified by the index “n” (n = 0, 1, 2), for example OSTMnTO for the General timer n output register.

#### Register address

The register addresses of the General timer are given as offsets from the individual base addresses <OSTMn_base>. The register base addresses of each OSTMn are listed in the following table.

Table 20.2 Register Base Addresses

| Base Address Name | Base Address |
|---|---|
| <OSTM0_base> | H’0_1280_1000 |
| <OSTM1_base> | H’0_1280_1400 |
| <OSTM2_base> | H’0_1280_1800 |

### Interrupts

The General timers can generate the following interrupt requests.

Table 20.3 OSTMn Interrupt Requests

| OSTMn Signal | Function | Startup of Direct Memory Access Controller |
|---|---|---|
| OSTM0TINT | OSTM0 interrupt | ✓ |
| OSTM1TINT | OSTM1 interrupt | ✓ |
| OSTM2TINT | OSTM2 interrupt | ✓ |

## 20.2 Registers

The General timer is controlled by the following registers.

### 20.2.1 Registers Overview

The list of OSTMn (n = 0, 1, 2) registers and the memory addresses are as follows.

For the base addresses, see **Table 20.2**.

For the actual addresses, the offset values indicated in the following table are added to the base addresses.

Table 20.4 Register Configuration

| Register Name | Function | R/W | Reset Value | Access Unit (bit) | | | Address |
|---|---|---|---|---|---|---|---|
| | | | | 8 | 16 | 32 | |
| OSTMnCMP | OSTM compare register | R/W | H'0000 0000 | — | — | ✓ | <OSTMn_base> + H'00 |
| OSTMnCNT | OSTM counter register | R | H'FFFF FFFF | — | — | ✓ | <OSTMn_base> + H'04 |
| OSTMnTE | OSTM count enable status register | R | H'00 | ✓ | — | — | <OSTMn_base> + H'10 |
| OSTMnTS | OSTM count start trigger register | W | H'00 | ✓ | — | — | <OSTMn_base> + H'14 |
| OSTMnTT | OSTM count stop trigger register | W | H'00 | ✓ | — | — | <OSTMn_base> + H'18 |
| OSTMnCTL | OSTM control register | R/W | H'00 | ✓ | — | — | <OSTMn_base> + H'20 |

## 20.2.2 Details of OSTM Registers

### 20.2.2.1 OSTMnCMP — OSTM Compare Register

Depending on the mode of operation, this register holds the start value for the down-counter or the value for comparison with that of the counter.

**Access Size:** This register is readable/writable in 32-bit units.

**Address(es):** <OSTMn_base>

**Initial Value:** H'0000 0000

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | OSTMnCMP[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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | OSTMnCMP[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 | OSTMnCMP[31:0] | All 0 | R/W | • In interval timer mode: start value of the down-counter<br>• In free-running comparison mode: value for comparison |

### 20.2.2.2 OSTMnCNT — OSTM Counter Register

This register indicates the counter value of the timer.

**Access Size:** This register is readable in 32-bit units.

**Address(es):** <OSTMn_base> + H'4

**Initial Value:** The initial value depends on the operating mode of the General timer. Refer to **Table 20.5**

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | OSTMnCNT[31:16] | | | | | | | | | | | | | | | |
| Initial Value | *1 | *1 | *1 | *1 | *1 | *1 | *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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | OSTMnCNT[15:0] | | | | | | | | | | | | | | | |
| Initial Value | *1 | *1 | *1 | *1 | *1 | *1 | *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 0 | OSTMnCNT [31:0] | *1 | R | 32-bit counter value |

Note 1. The initial value depends on the operating mode of the General timer. Refer to **Table 20.5**

The following table shows the correspondence between operating mode, counting direction and initial value. The initial value is the value read from the counter after a change to the operating mode.

Table 20.5 Correspondence between Operating Mode, Counting Direction and Initial Value

| Timer Operating Mode | OSTMnCTL.OSTMnMD1 | Counting Direction | Initial value |
|---|---|---|---|
| Interval timer mode | 0*1 | Down | H'FFFF FFFF |
| Free-running comparison mode | 1 | Up | H'0000 0000 |

Note 1. Value after reset

### 20.2.2.3 OSTMnTE — OSTM Count Enable Status Register

This register indicates whether the counter is enabled or disabled.

**Access Size:** This register is readable in 8-bit units.

**Address(es):** <OSTMn_base>+ H'10

**Initial Value:** H'00

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | OSTMnTE |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 to 1 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 0 | OSTMnTE | 0 | R | This bit indicates whether the counter is enabled or disabled.<br>0: Counter disabled<br>1: Counter enabled<br>This bit is set to 1 in response to OSTMnTS.OSTMnTS being set to 1.<br>This bit is reset to 0 in response to OSTMnTT.OSTMnTT being set to 1. |

Note 1. When OSTMnTE = 0, the counter retains its value.
If the counter is restarted, it
• restarts counting down from the value in the OSTMnCMP register if it is in interval timer mode or
• restarts counting up from the counter value H'0000 0000 if it is in free running comparison mode.

Note 1. For debugging, the GTM counter can be paused by the counter stop request.

### 20.2.2.4 OSTMnTS — OSTM Count Start Trigger Register

This register starts the counter.

**Access Size:** This register is writable in 8-bit units. It is always read as H'00.

**Address(es):** <OSTMn_base>+ H'14

**Initial Value:** H'00

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | OSTMnTS |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 to 1 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 0 | OSTMnTS | 0 | W | This bit starts the counter.<br>0: This setting has no effect.<br>1: Starts the counter and sets OSTMnTE.OSTMnTE = 1.<br>• In interval timer mode, a forced restart is executed if this bit is set while OSTMnTE.OSTMnTE = 1.<br>• In free-running comparison mode, setting this bit is ignored as long as OSTMnTE.OSTMnTE = 1. |

### 20.2.2.5 OSTMnTT — OSTM Count Stop Trigger Register

This register stops the counter.

**Access Size:** This register is writable in 8-bit units. It is always read as H'00.

**Address(es):** <OSTMn_base>+ H'18

**Initial Value:** H'00

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | OSTMnTT |
| Initial Value | | | | | | | | |
| R/W | R | R | R | R | R | R | R | W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 to 1 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 0 | OSTMnTT | 0 | W | Stops the counter.<br>0: This setting has no effect.<br>1: Stops the counter and clears the OSTMnTE.OSTMnTE bit. |

### 20.2.2.6 OSTMnCTL — OSTM Control Register

This register specifies the operating mode for the counter and controls enabling/disabling of OSTMnTINT interrupt requests when counting starts.

**Access Size:** This register is readable/writable in 8-bit units. Writing to this register is only possible if the counter is disabled (OSTMnTOE.OSTMnTOE = 0).

**Address(es):** <OSTMn_base>+ H'20

**Initial Value:** H'00

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | OSTMnMD1 | OSTMnMD0 |
| 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<br>These bits are always read as 0. The write value should always be 0. |
| 1 | OSTMnMD1 | 0 | R/W | Specifies the operating mode for the counter.<br>0: Interval timer mode<br>1: Free-running comparison mode |
| 0 | OSTMnMD0 | 0 | R/W | Controls enabling/disabling of OSTMnTINT interrupt requests when counting starts.<br>0: Disables the interrupts when counting starts.<br>1: Enables the interrupts when counting starts. |

## 20.3 Functional Description

Each General timer is a 32-bit timer/counter.

The settings for operating mode specify the direction of counting (up or down) and the generation of interrupt requests.

### 20.3.1 Block Diagram

The following block diagram shows the main components of GTM.

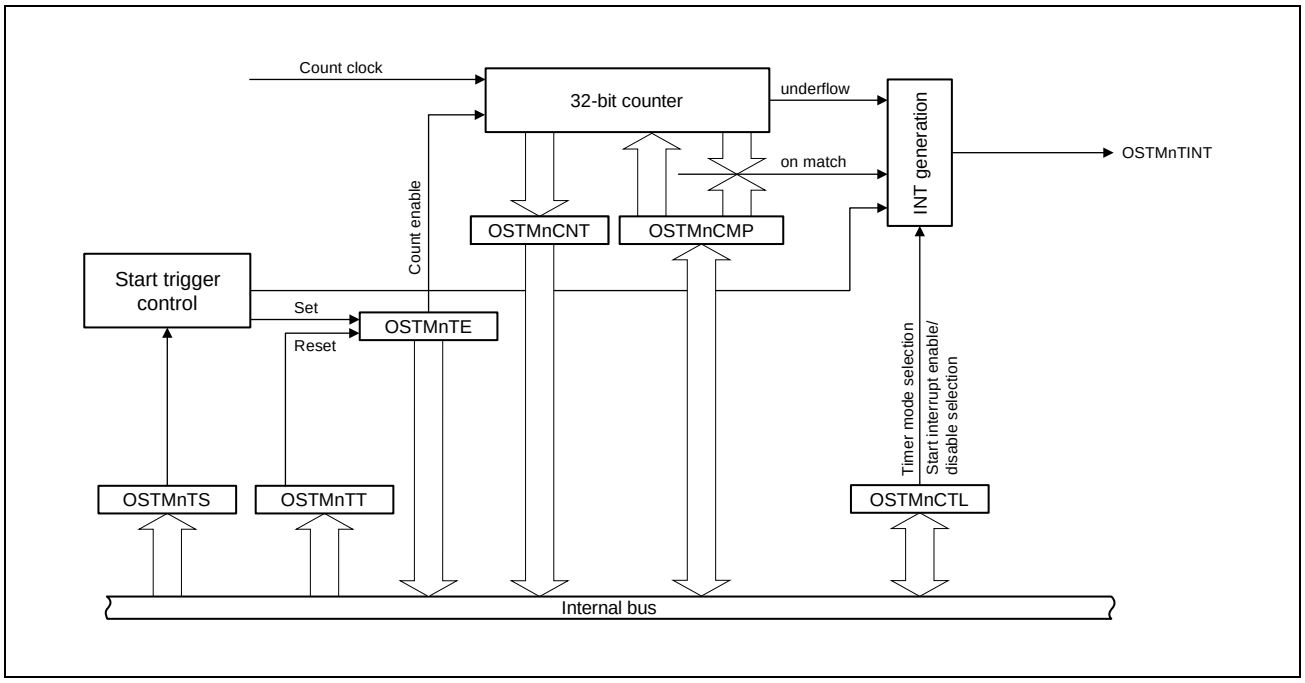



Figure 20.1 Block Diagram of GTM

### 20.3.2 Count Clock

The count clock of OSTMn is P0ϕ (OSTMn_PCLK).

### 20.3.3 Generation of Interrupt Request

An OSTMnTINT interrupt request is generated whenever the counter reaches H’0000 0000 (in interval timer mode) or matches the comparison value (in free-running comparison mode).

An interrupt request can also be generated on starting and restarting of the counter. This is controlled by the OSTMnCTL.OSTMnMD0 bit.

This operation is shown in the following figure.

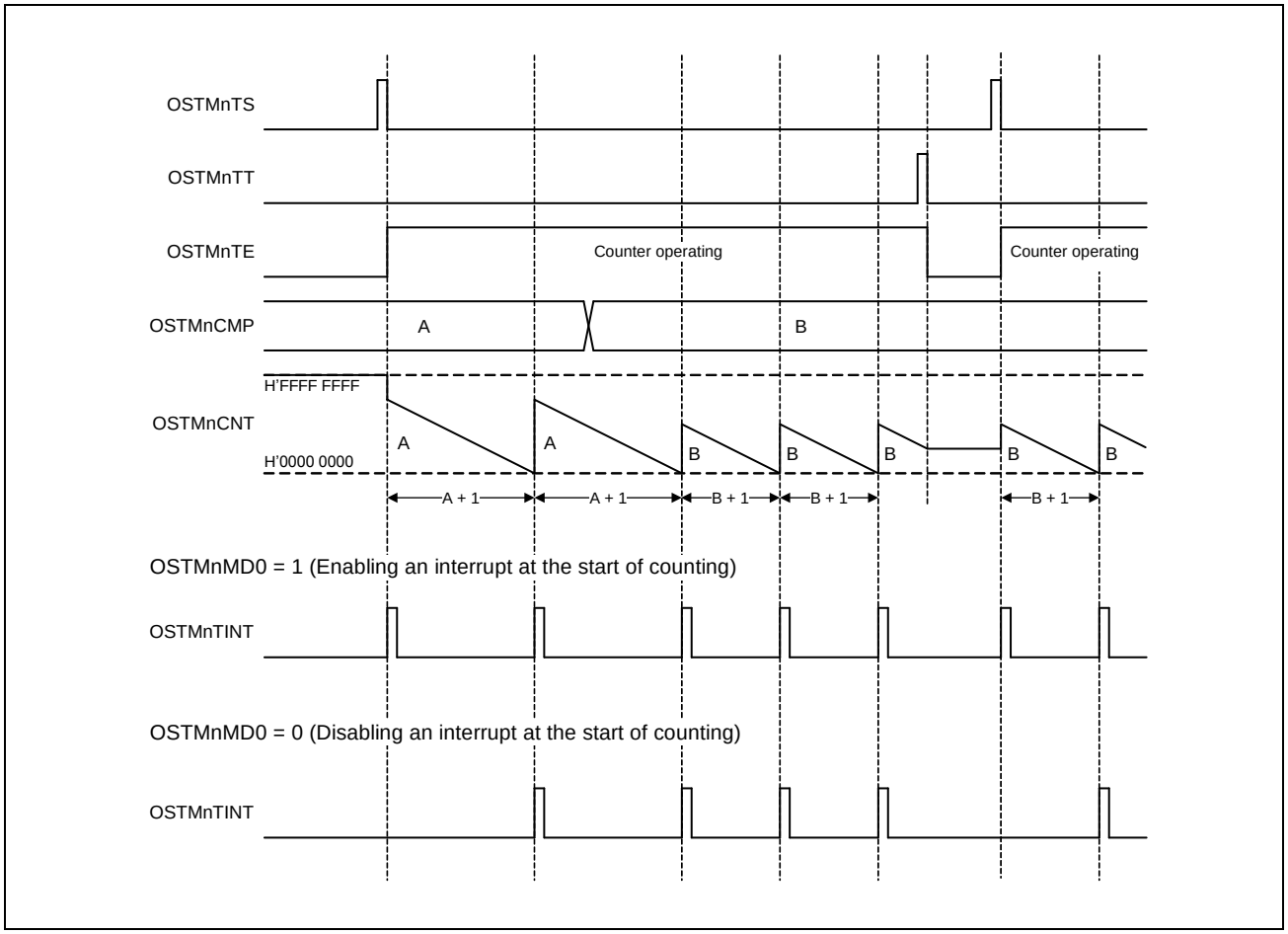



Figure 20.2 Generating an Interrupt when Counting Starts (in Interval Timer Mode)

### 20.3.4 Starting and Stopping the Timer

The General timer is started and stopped as follows.

#### Starting the timer

The timer is started in either of the following way:

- setting the OSTMnTS.OSTMnTSF bit to 1

Status bit OSTMnTE.OSTMnTE is set to 1.

The counter starts to count up or down in accord with the settings for operating mode.

#### Stopping the timer

Setting the OSTMnTT.OSTMnTT bit to 1 stops the timer.

This also clears the OSTMnTE.OSTMnTE status flag.

## 20.3.5 Interval Timer Mode

Select the interval timer mode when the General timer is to be used as a reference timer for generating interrupt requests at a fixed interval.

### 20.3.5.1 Basic Operation in Interval Timer Mode

In interval timer mode, the timer counts down from the value specified in the OSTMnCMP register. An OSTMnTINT interrupt request is generated when the counter reaches H'0000 0000.

Select interval timer mode by setting OSTMnCTL.OSTMnMD1 = 0.

New values can be written to the OSTMnCMP register at any time. If it is rewritten during count operation, the counter loads the new OSTMnCMP value when the next H'0000 0000 is reached.

#### Cycles of OSTMnTINT output

The cycle of OSTMnTINT output is as follows.

- OSTMnTINT generation cycle = counter-clock cycle × (OSTMnCMP + 1)

The following figure shows the basic operation of OSTM when counter-start interrupts is enabled in interval timer mode.

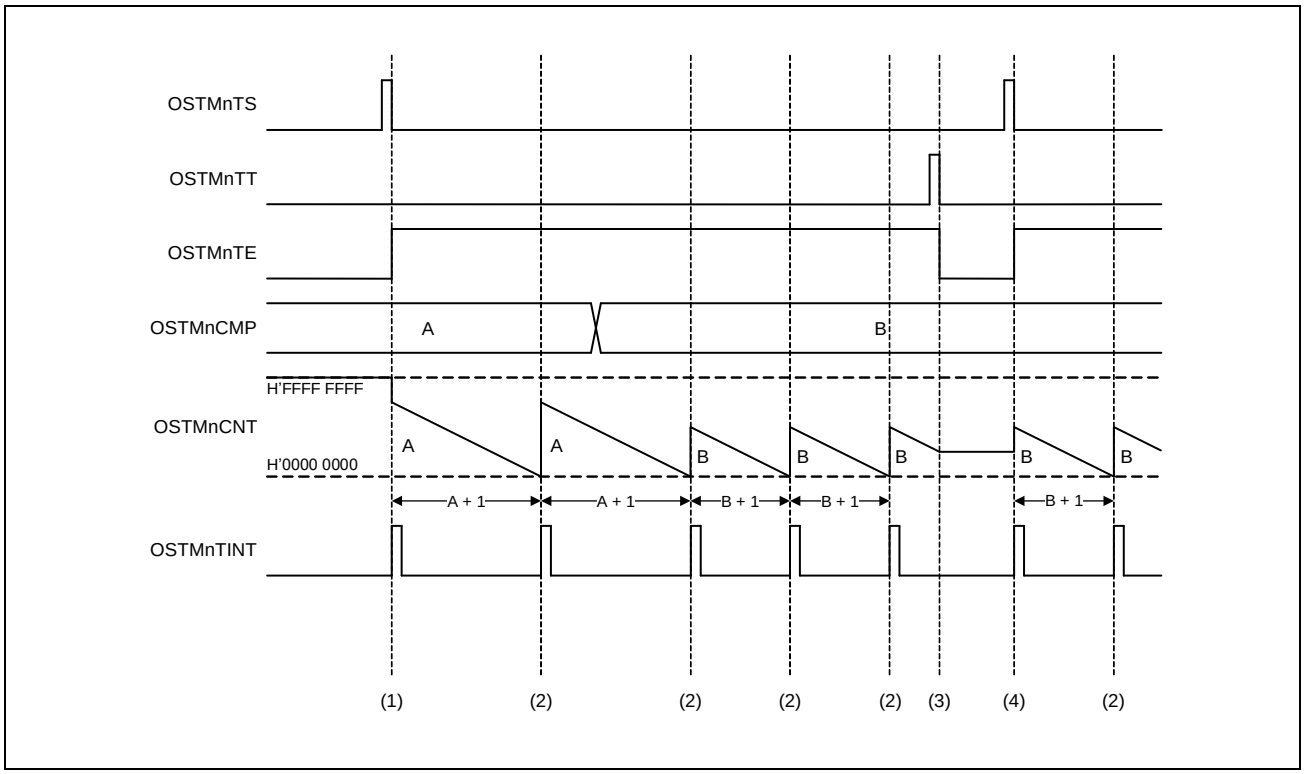



Figure 20.3 Timing Diagram of OSTM in Interval Timer Mode

The timing diagram above shows the following:

(1) The counter starts counting when OSTMnTS.OSTMnTS = 1. The OSTMnTE.OSTMnTE bit is set to indicate enabling of the counter.
The counter starts counting down from the value of OSTMnCMP.
If OSTMnCTL.OSTMnMD0 is 1, OSTMTINT interrupt requests are generated at the start of counting. The OSTMnCNT register contains the current value as the counter.

(2) When the counter reaches H’0000 0000, an OSTMTINT interrupt request is generated. The counter loads the new start value from OSTMnCMP and continues counting down.

(3) When the counter is stopped (OSTMnTT.OSTMnTT = 1), the OSTMnTE.OSTMnTE bit is cleared to indicate disabling of the counter. The counter retains its current value until it is restarted.

(4) When counting is restarted (OSTMnTS.OSTMnTS = 1), the counter loads the new start value from OSTMnCMP and starts counting down.

### Forced restart

The counter is forcibly restarted by setting OSTMnTS.OSTMnTS = 1 during counting.

The counter loads the start value from the OSTMnCMP register and continues to count down.

The following figure shows the forced restart of the General Timer in interval timer mode, with counter-start interrupts enabled (OSTMnCTL.OSTMnMD0 = 1).

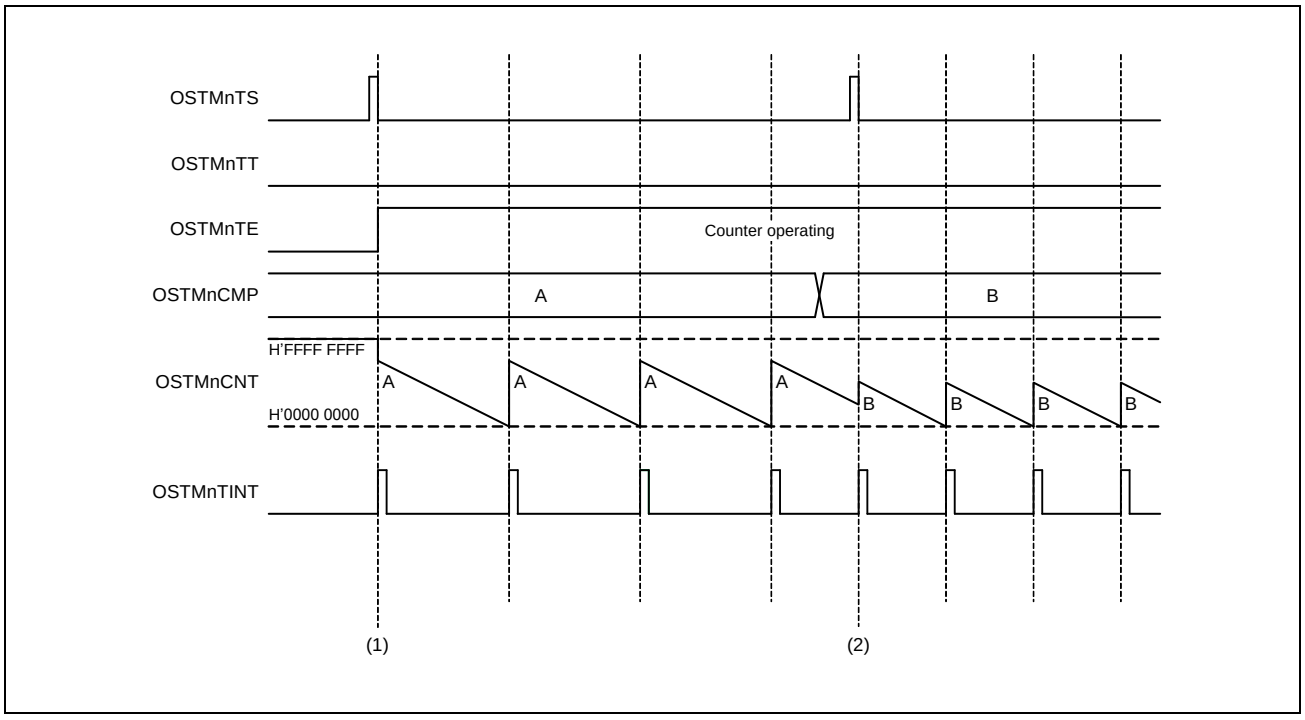



Figure 20.4 Timing Diagram of Forced Restart in Interval Timer Mode

Operations shown in the above timing diagram are as follows.

(1) The counter is started and stopped as described under **Figure 20.3, Timing Diagram of OSTM in Interval Timer Mode**.

(2) Setting OSTMnTS.OSTMnTS = 1 restarts the counter while counting is in progress (i.e. while OSTMnTE.OSTMnTE = 1).
The counter immediately restarts counting down, starting with the current value of OSTMnCMP.
When OSTMnCTL.OSTMnMD0 = 1, an OSTMTINT interrupt request is generated when counting starts.

### 20.3.5.2 Operation when OSTMnCMP = H’0000 0000

When OSTMnCMP = H’0000 0000, OSTM behaves as follows.

- When the counter is enabled, the OSTMTINT interrupt request is always set to 1.

The following figure shows operations of OSTM when OSTMnCMP = H’0000 0000, and counter-start interrupts are enabled.

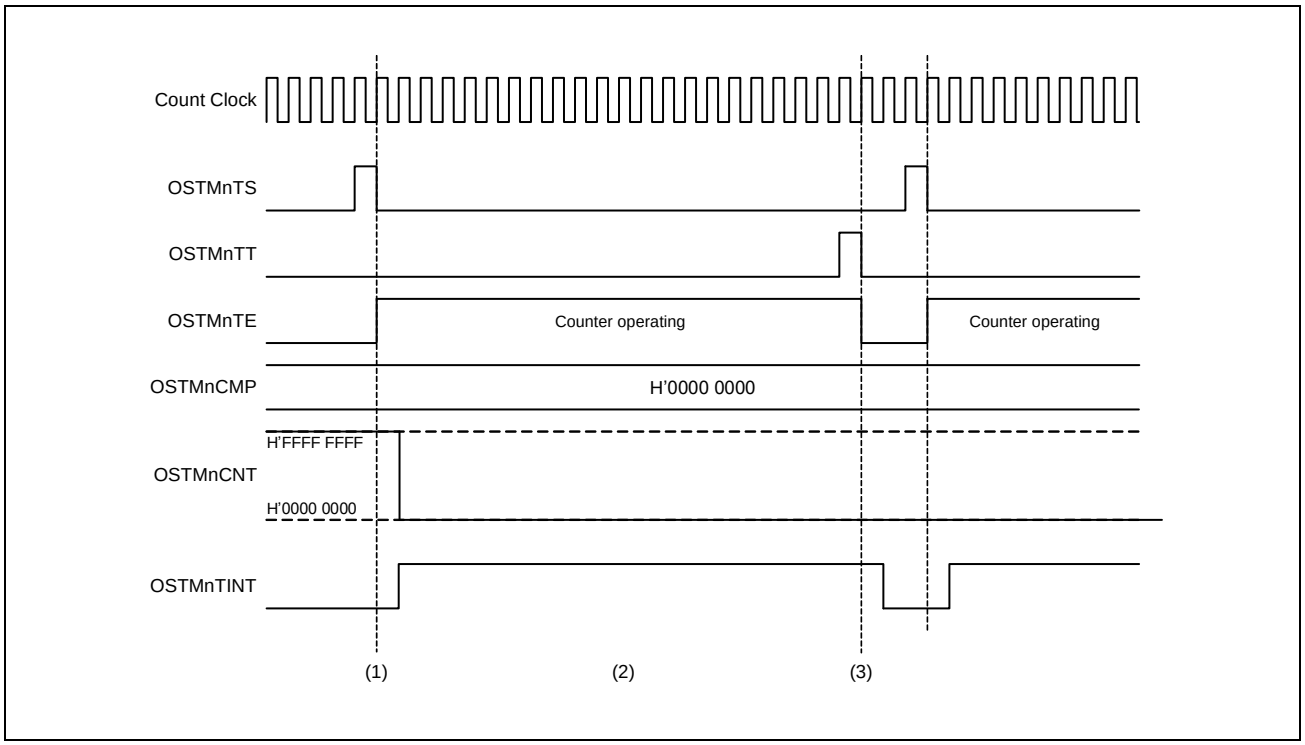



Figure 20.5 Timing Diagram when OSTMnCMP = H’0000 0000 in Interval Timer Mode

The timing diagram above shows the following operations:

(1) The counter is reloaded with the value in OSTMnCMP as soon as it starts counting, so the value H’0000 0000 is retained in OSTMnCMP.

(2) The OSTMTINT interrupt request is continuously asserted.

(3) After the counter stops, the OSTMTINT interrupt request signal is deasserted.

(4) When interrupts on starting of the counter are disabled, no interrupt is generated when counting starts.

## 20.3.6 Free-Running Comparison Mode

### 20.3.6.1 Basic Operation in Free-Running Comparison Mode

In free-running comparison mode, the counter counts up from H’0000 0000 to H’FFFF FFFF. An OSTMnTINT interrupt request is output when the current value of the counter matches the value of the OSTMnCMP register. The free-running comparison mode is selected by setting the OSTMnCTL.OSTMnMD1 bit to 1.

New values can be written to the OSTMnCMP register at any time.

The following figure shows the basic operation of OSTM in free-run compare mode with the start of counting enabled (OSTMnCTL.OSTMnMD0 = 1).

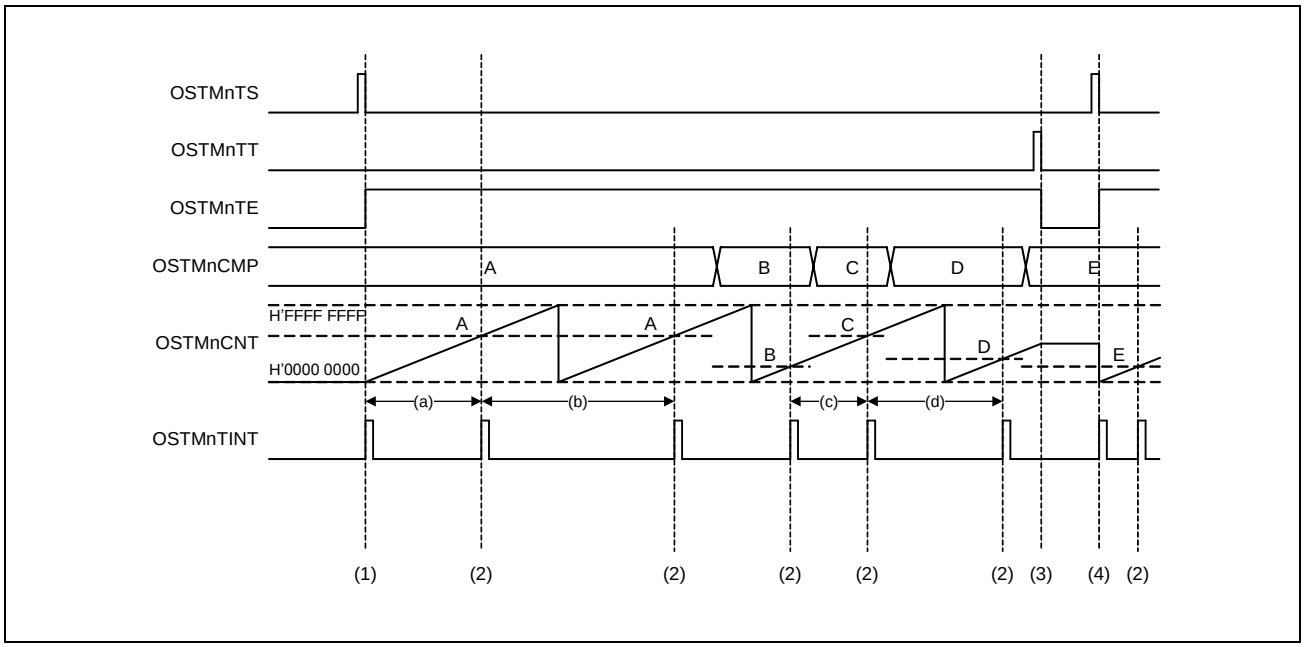



Figure 20.6 Timing Diagram of OSTM in Free-Run Compare Mode

The timing diagram above shows the following:

(1) The counter starts counting when OSTMnTS.OSTMnTS = 1.
The OSTMnTE.OSTMnTE bit is set to indicate enabling of the counter. The counter counts up from H’0000 0000 to H’FFFF FFFF. The OSTMnCNT register is the counter, so it contains the current value.
When OSTMnCTL.OSTMnMD0 = 1, an OSTMTINT interrupt request is generated at the start of counting.

(2) When the current counter value matches the value in the OSTMnCMP register, an OSTMTINT interrupt request is generated.

(3) When the counter is stopped (OSTMnTT.OSTMnTT = 1), the OSTMnTE.OSTMnTE bit is cleared to indicate disabling of the counter.
The counter retains its current value until it is restarted.

(4) Counting by the counter restarts from H’0000 0000 when OSTMnTS.OSTMnTS = 1.

### OSTMTINT period

The OSTMTINT generation period is different at the start of counting and depends on the old and new compare values if OSTMnCMP is rewritten during operation.

Table 20.6 OSTMTINT Generation Timing

| Old Value for Comparison | New Value for Comparison | Counter Value at Time of Rewriting | Period of OSTMTINT Generation | Label in Timing Diagram |
|---|---|---|---|---|
| Counter starts | | | (A + 1) × counter clock period | (a) |
| A | A | No rewriting | (H'FFFF FFFF + 1) × counter clock period | (b) |
| B | C > B | B < counter value < C | (C – B) × counter clock period | (c) |
| C | D < C | Counter value > D,C | (H'FFFF FFFF – C + D + 1) × counter clock period | (d) |

### Forced restart

Forced restarting does not proceed during counting even if the OSTMnTS.OSTMnTS bit is set. The counter ignores the attempted setting and continues counting.

### 20.3.6.2 Operation when OSTMnCMP = H’0000 0000

The following figure shows the operation of OSTM when OSTMnCMP = H’0000 0000, and counter-start interrupts are enabled (OSTMnCTL.OSTMnMD0 = 1).

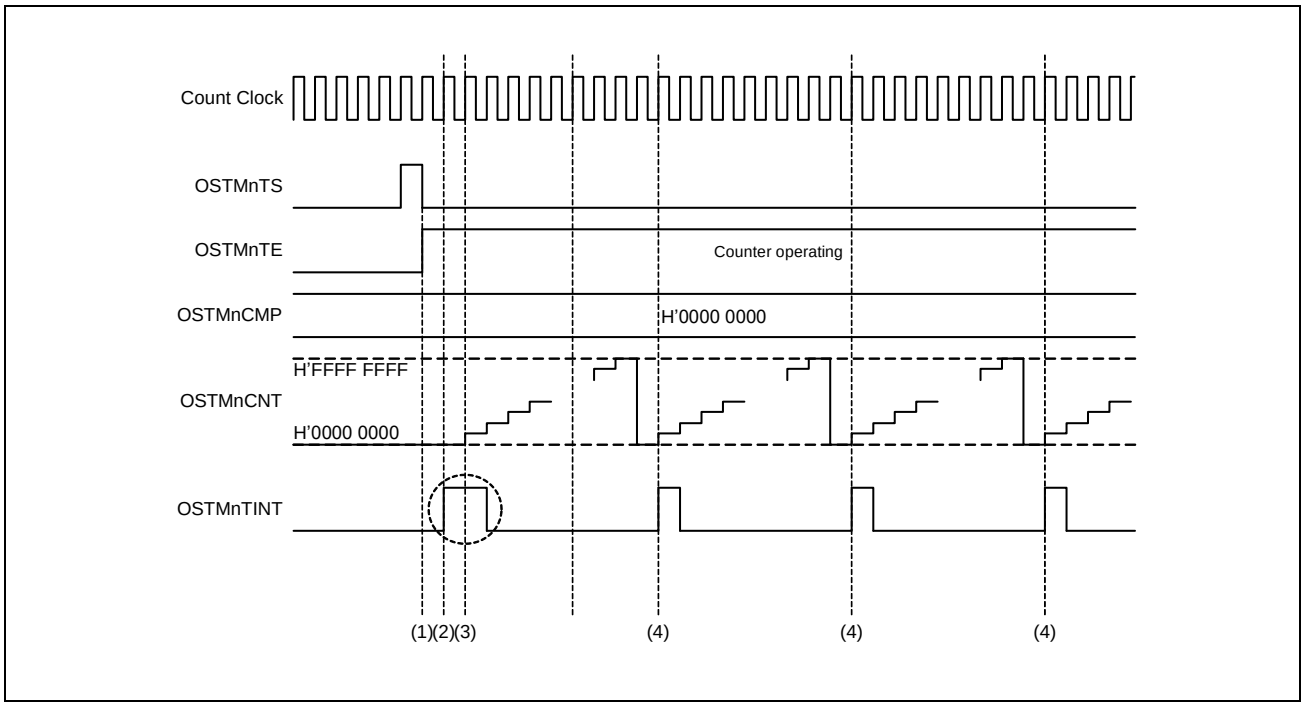



Figure 20.7 Timing Diagram when OSTMnCMP = H’0000 0000 in Free-Run Compare Mode

The timing diagram above shows the following operations.

(1) Once the counter starts, it counts up from H’0000 0000 to H’FFFF FFFF.

(2) An OSTMTINT interrupt request is generated when counting starts.

(3) If the current counter value matches OSTMnCMP, an OSTMTINT interrupt request is generated. If OSTMnCMP = H’0000 0000 in the above case, OSTMTINT is generated over two clock cycles.

(4) Every (H’FFFF FFFF + 1) clock cycles the OSTMTINT interrupt request is asserted.

When interrupts on starting of the counter are disabled, no interrupt is generated when counting starts.

# 21. Watchdog Timer (WDT)

This LSI has 1 channel of watchdog timer as **Table 21.1** and generates a reset request signal when the counter value is not rewritten and overflows due to system runaway.

Table 21.1 WDT Channels

| Channel | Functions |
|---|---|
| WDT CH0 | WDT to check the operation of Cortex-A55-CPU Core0 |

Also, a reset request signal or interrupt signal can be generated by inputting a parity error signal due to a parity error in the CPU of this LSI.

## 21.1 Features

All WDT module has the following functions implemented in common as basic WDT functions.

### 21.1.1 Normal Watchdog Timer Function

- Bus interface: Compatible with AMBA® 2.0 APB.
- Operating frequency: Bus interface 100 [MHz], counter control 24 [MHz]
- The counter operation and the bus interface are asynchronous and can operate without depending on the size relationship of the clock cycles of each other.
- A 32-bit counter that operates with the clock (WDTn_CLK (n = 0)) input to this module.
- An interrupt request signal is generated every cycle set in the counter of this module.
- When WDTn_CLK (n = 0) = 24 MHz, the counter cycle can be set in 43.69 msec unit from 43.69 msec to 178956.97 msec.
- After the interrupt request signal is generated, if the software does not clear the watchdog timer until the next counter overflow, the reset request signal is generated.
- The set value of each register and the eElapsed time of the watchdog timer can be read.

### 21.1.2 Reset Request Function due to CPU Parity Error

- A reset request signal (WDTRSTB) or interrupt signal (PERROUT) is generated by inputting a parity error signal.
- The reset request signal is generated as the OR of the reset request of this module and the reset request of the parity error circuit.
- The polarity of the parity error signal can be set with this module.
- Function can be turned ON/OFF for each 32-bit parity error signal.
- The output of the reset request signal and the interrupt signal can be switched by inputting the parity error signal.
- It is possible to force a parity error to occur by register setting.

### 21.1.3 Internal Block Diagram

**Figure 21.1** shows the internal block of WDT. WDT consists of a WDT-CORE part that implements the function of a normal watchdog timer and a part that detects a parity error.

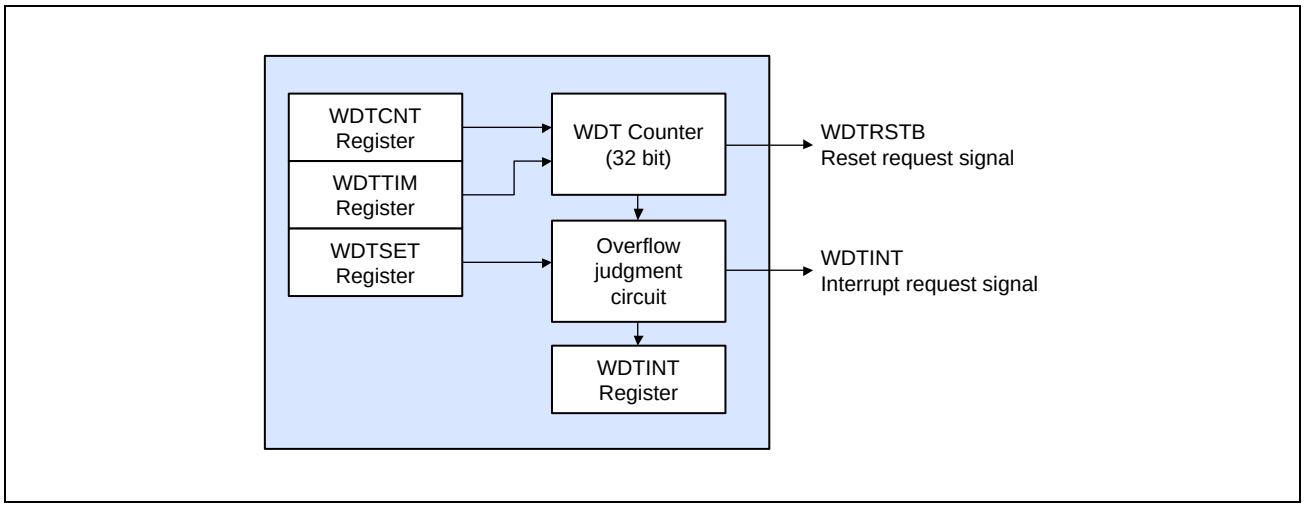



Figure 21.1 WDT_CORE Internal Block

In addition to the above WDT-Core function, WDT receives a parity error interrupt signal generated by the CPU when a Parity Error occurs in the CPU monitored by WDT and generates a reset request signal or interrupt request signal.

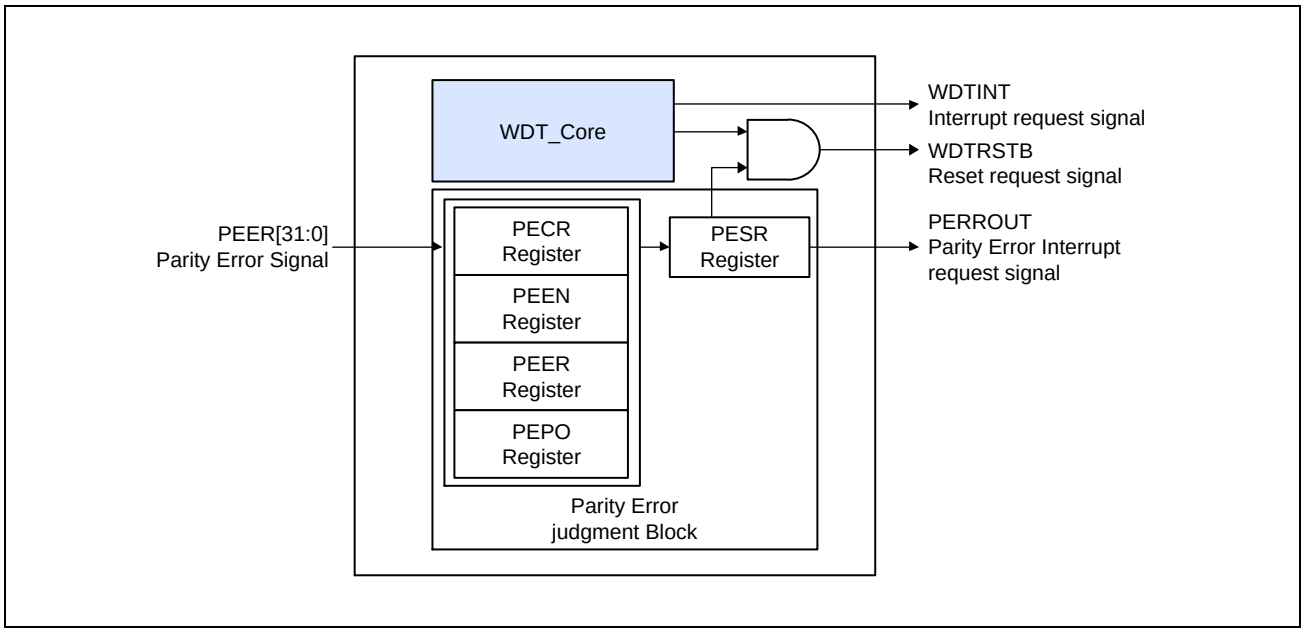



Figure 21.2 WDT_CORE and Parity Error handling block

### 21.1.4 Input/Output Pin

**Table 21.2** shows the pin configuration.

Table 21.2 Pin Configuration

| | | | |
|---|---|---|---|
| Watchdog timer overflow/ CPU parity error | WDTOVF_PERROUT# | Output | Watchdog timer counter overflow signal output/ Interrupt signal output when CPU parity error occurs |

## 21.2 Register Configuration

**Table 21.3** shows the base address for WDT channel.

Table 21.3 Base Address for each WDT channel

| | Channel | WDT | Base Address |
|---|---|---|---|
| Cortex-A55 Address Space | WDT CH0 | WDT to check the operation of Cortex-A55-CPU Core0 | H'0_1280_0800 |

**Table 21.4** shows WDT Register Configuration on all WDT Channel.

Table 21.4 WDT Register Configuration

| Channel | Register Name | Abbreviation | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|---|
| WDT CH0 | WDT Control Register_0 | WDTCNT_0 | R/W | H'0000_0000 | H'0000 | 32 |
| | WDT Period Setting Register_0 | WDTSET_0 | R/W | H'FFF0_0000 | H'0004 | 32 |
| | WDT Elapsed Time Register_0 | WDTTIM_0 | R/W | H'0000_0000 | H'0008 | 32 |
| | WDT Interrupt control Register_0 | WDTINT_0 | R/W | H'0000_0000 | H'000C | 32 |
| | Parity Error Control Register_0 | PECR_0 | R/W | H'0000_0000 | H'0010 | 32 |
| | Parity Error forced Enable Register_0 | PEEN_0 | R/W | H'0000_0000 | H'0014 | 32 |
| | Party Status Register_0 | PESR_0 | R/W | H'0000_0000 | H'0018 | 32 |
| | Parity Error Enable Register_0 | PEER_0 | R/W | H'0000_0000 | H'001C | 32 |
| | Parity Error Polarity setting Register_0 | PEPO_0 | R/W | H'0000_0000 | H'0020 | 32 |

# 21.3 Register Descriptions

## 21.3.1 WDT Control Register_n (WDTCNT_n) (n = 0)

This register sets the operation enable of the watchdog timer function. Once the software activates the watchdog timer, the watchdog timer does not stop*1 until it is reset. Also, the setting register*2 of the watchdog timer cannot be changed. The setting register change must be completed*2 before enabling the watchdog timer operation.

**Note 1.** For debugging, the internal counter can be paused with the CNTSTOP signal.

**Note 2.** WDT cycle setting register (WDTSET) and WDT eElapsed time register (WDTTIM)

**Note 3.** When changing the setting register, wait until the write data value of the setting register is reflected before enabling the watchdog timer operation. The reflection time of the setting register is 6 × P0ϕ + 9 × OSCCLK or more. If this reflection time is not observed, the setting register value will not be reflected and will be ignored.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | WDTEN |
| 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<br>When read, the initial value is read. The written value will be ignored. |
| 0 | WDTEN | 0 | R/W | Watchdog timer enable register<br>1: Watchdog timer operation enabled<br>0: Disabled |

### 21.3.2 WDT Period Setting Register_n (WDTSET_n) (n = 0)

This register is used to set the cycle of the 32-bit counter that composes the watchdog timer. Writing is not possible while the watchdog timer is running. If you write continuously, the data written later will be valid. Calculate the watchdog timer cycle using the following formula.

$$\text{Watchdog timer cycle} = \text{WDTn_CLK (n = 0) cycle} \times 1024 \times 1024 \times (\text{WDTTIME setting value} + 1)$$

$$\text{WDTTIME setting value} = \frac{\text{WDT cycle}}{(\text{WDTn_CLK (n = 0) cycle} \times 1024 \times 1024) - 1}$$

For example, WDTn_CLK (n = 0) = 24 MHz, the counter cycle can be set in 43.69 msec unit from 43.69 msec to 178956.97 msec.

It can be set in units. The setting value of WDTTIME is the value set in WDTSET[31:20], and the value from H’000 to H’FFF can be set.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | WDTTIME11 | WDTTIME10 | WDTTIME9 | WDTTIME8 | WDTTIME7 | WDTTIME6 | WDTTIME5 | WDTTIME4 | WDTTIME3 | WDTTIME2 | WDTTIME1 | WDTTIME0 | — | — | — | — |
| Initial Value | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 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 | 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 20 | WDTTIME [11:0] | All 1 | R/W | Watchdog timer period cycle setting<br>It cannot be written while the watchdog timer is running. |
| 19 to 0 | — | All 0 | R | Reserved<br>When read, the initial value is read. The written value will be ignored. |

**CAUTION**

The value that the watchdog timer compares when an interrupt occurs is “WDTSET[31:0] + H’000F_FFFF”. Therefore, when the setting value of this register is H’0010_0000, the interrupt occurrence count is “H’001F_FFFF” (not H’0010_0000).

### 21.3.3 WDT Elapsed Time Register_n (WDTTIM_n (n = 0))

This register indicates the 32-bit count value that composes the watchdog timer. If the watchdog timer is read during operation, the eElapsed time from the time it is cleared to the time of reading can be read.

By writing to this register when the watchdog timer is stopped, the value at the start of counting can be set in the 32-bit counter that configures the watchdog timer. When a value larger than the counter cycle (value when clearing) set in the WDT cycle setting register is written and the watchdog timer operation is started, the 32-bit counter once overflows and returns to 0. It operates at the set correct cycle.

You cannot write while the watchdog timer is running. If you write continuously, the data written later will be valid.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | CRTTIME31 | CRTTIME30 | CRTTIME29 | CRTTIME28 | CRTTIME27 | CRTTIME26 | CRTTIME25 | CRTTIME24 | CRTTIME23 | CRTTIME22 | CRTTIME21 | CRTTIME20 | CRTTIME19 | CRTTIME18 | CRTTIME17 | CRTTIME16 |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | CRTTIME15 | CRTTIME14 | CRTTIME13 | CRTTIME12 | CRTTIME11 | CRTTIME10 | CRTTIME9 | CRTTIME8 | CRTTIME7 | CRTTIME6 | CRTTIME5 | CRTTIME4 | CRTTIME3 | CRTTIME2 | CRTTIME1 | CRTTIME0 |
| 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 | CRTTIME [31:0] | All 0 | R/W | Watchdog timer count value.<br>It cannot be written while the watchdog timer is running. |

**CAUTION**

When setting a value in the WDTTIM register, do not set a value that is WDTSET[31:0] register + H'000F_FFFF.

### 21.3.4 WDT Interrupt Control Register_n (WDTINT_n (n = 0))

This register reads the interrupt status of the watchdog timer, clears interrupts, and clears the counters that compose the watchdog timer. Set the INTDISP bit of this register to “1” within the time set by WDTSET. If not set, the WDTINT pin is asserted. After that, within the time set by WDTSET

When the INTDISP bit is not set to “1”, the WDTRSTB pin is asserted.

When performing continuous write access to the WDTINT register, leave a write interval of 5 × P0ϕ + 5 × OSCCLK or more. If the write interval is not observed, the register will not be written.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | INTDISP |
| 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<br>When read, the initial value is read. The written value will be ignored. |
| 0 | INTDISP | 0 | R/W | Register for interrupt status read and clear<br>Read: Interrupt status<br>1: With interrupt<br>0: Without interrupt<br>Write: Interrupt clear<br>1: Interrupt clear<br>0: Invalid (no change) |

### 21.3.5 Parity Error Control Register_n (PECR_n (n = 0))

This register controls whether a reset (WDTRSTB) or an interrupt (PERROUT) is generated due to a CPU parity error.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PECR31 | PECR30 | PECR29 | PECR28 | PECR27 | PECR26 | PECR25 | PECR24 | PECR23 | PECR22 | PECR21 | PECR20 | PECR19 | PECR18 | PECR17 | PECR16 |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PECR15 | PECR14 | PECR13 | PECR12 | PECR11 | PECR10 | PECR9 | PECR8 | PECR7 | PECR6 | PECR5 | PECR4 | PECR3 | PECR2 | PECR1 | PECR0 |
| 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 | PECR [31:0] | All 0 | R/W | Parity error control<br>Controls the following operations when the value is 1 for each bit of PESR[31:0]<br>0: Reset (WDTRSTB) is generated<br>1: Generate interrupt (PEEROUT) |

### 21.3.6 Parity Error Forced Enable Register_n (PEEN_n (n = 0))

This register forcibly asserts reset (WDTRSTB) or PERROUT even when no CPU parity error has occurred.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | PEEN |
| 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<br>When read, the initial value is read. The written value will be ignored. |
| 0 | PEEN | 0 | R/W | Parity error forced enable<br>0: Reset (WDTRSTB) or PERROUT generation follows the operation of the parity error register<br>1: Force reset (WDTRSTB) or assert PERROUT |

### 21.3.7 Parity Error Status Register_n (PESR_n (n = 0))

This register holds the cause of CPU parity error.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PESR31 | PESR30 | PESR29 | PESR28 | PESR27 | PESR26 | PESR25 | PESR24 | PESR23 | PESR22 | PESR21 | PESR20 | PESR19 | PESR18 | PESR17 | PESR16 |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PESR15 | PESR14 | PESR13 | PESR12 | PESR11 | PESR10 | PESR9 | PESR8 | PESR7 | PESR6 | PESR5 | PESR4 | PESR3 | PESR2 | PESR1 | PESR0 |
| 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 | PESR[31:0] | All 0 | R/W | Parity error flag 0 to 31 Indicates that a parity error has occurred.<br>Read: Interrupt status<br>1: With interrupt<br>0: Without interrupt<br>Write: Interrupt clear<br>1: Interrupt clear<br>0: Invalid (nothing changed) |

### 21.3.8 Parity Error Enable Register_n (PEER_n (n = 0))

This register controls the assertion of the WDTRSTB signal and PERROUT signal due to a CPU parity error. Each bit controls the behavior of parity error.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PEER31 | PEER30 | PEER29 | PEER28 | PEER27 | PEER26 | PEER25 | PEER24 | PEER23 | PEER22 | PEER21 | PEER20 | PEER19 | PEER18 | PEER17 | PEER16 |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PEER15 | PEER14 | PEER13 | PEER12 | PEER11 | PEER10 | PEER9 | PEER8 | PEER7 | PEER6 | PEER5 | PEER4 | PEER3 | PEER2 | PEER1 | PEER0 |
| 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 | PEER[31:0] | All 0 | R/W | Parity error enable 31-0<br>Controls the operation of PESR.<br>0: Disables reset (WDTRSTB) and PERROUT generation operations<br>1: Permits reset (WDTRSTB) and PERROUT generation operation |

### 21.3.9 Parity Error Polarity Setting Register_n (PEPO_n (n = 0))

This register sets the polarity of CPU parity error.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PEPO31 | PEPO30 | PEPO29 | PEPO28 | PEPO27 | PEPO26 | PEPO25 | PEPO24 | PEPO23 | PEPO22 | PEPO21 | PEPO20 | PEPO19 | PEPO18 | PEPO17 | PEPO16 |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PEPO15 | PEPO14 | PEPO13 | PEPO12 | PEPO11 | PEPO10 | PEPO9 | PEPO8 | PEPO7 | PEPO6 | PEPO5 | PEPO4 | PEPO3 | PEPO2 | PEPO1 | PEPO0 |
| 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 | PEPO [31:0] | All 0 | R/W | Active polarity of parity 31-0<br>Set the Active polarity (H/L) of each parity error signal connected to PEERR.<br>0: Active High<br>1: Active Low |

### 21.3.10 Register Setting Order for Each WDT Channel

The following shows the register setting sequence for controlling the watchdog timer and parity error circuit of each channel. The control of the original watchdog timer and the parity error circuit do not affect each other, so there is no restriction on the setting order in the following diagrams (1) and (2).

The parity error circuit cannot accept the parity error that occurred before setting PEER = 1. When rewriting PEEN from 0 to 1, it is possible to rewrite at any timing. However, when rewriting PEEN from 1 to 0, once PEER = 0 and (2) flow execution is required.

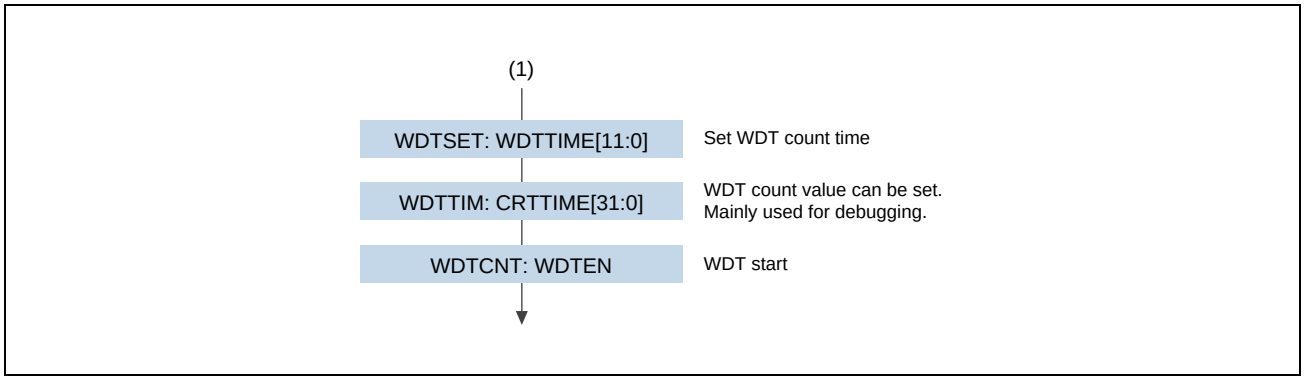



Figure 21.3 WDT Register Setting Order

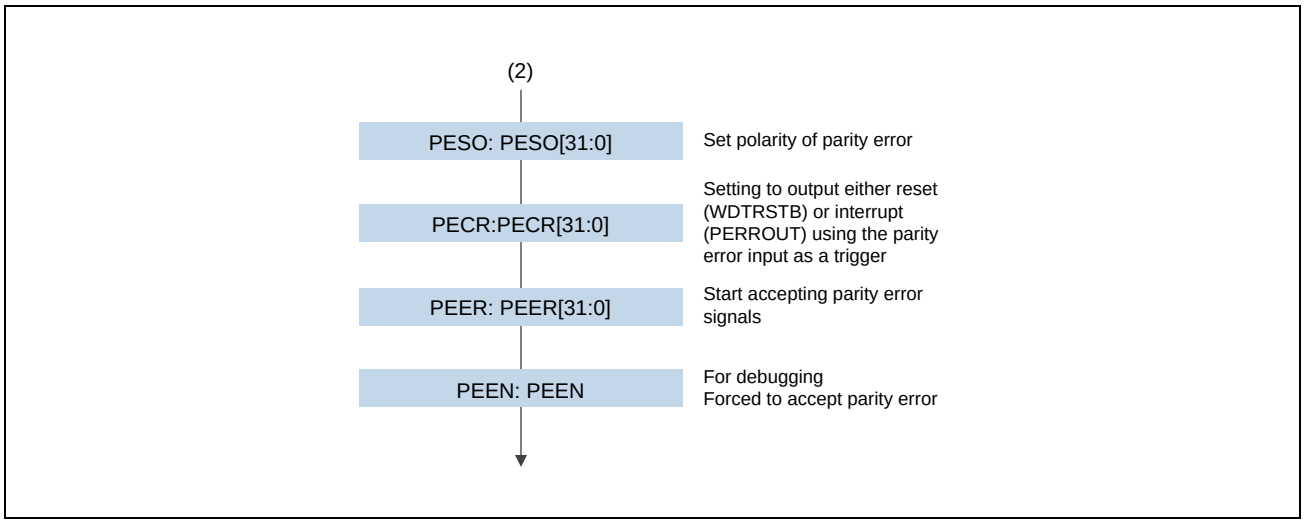



Figure 21.4 Parity Error Circuit Register Setting Order

## 21.4 Operation

### 21.4.1 WDT_CORE Operaton Timing

**Figure 21.5** shows the operation diagram of the WDT-Core block. When 1 is set in Bit [0] of the WDTCNT register, the 32-bit internal counter WDT-Counter starts counting. If the counter value matches the value set in the WDTSET register, the counter becomes overflow, clears the counter to 0, and then generates an interrupt request signal.

An interrupt signal is output at the first counter overflow, and a reset request signal is output if the counter is not cleared and the second counter overflow occurs.

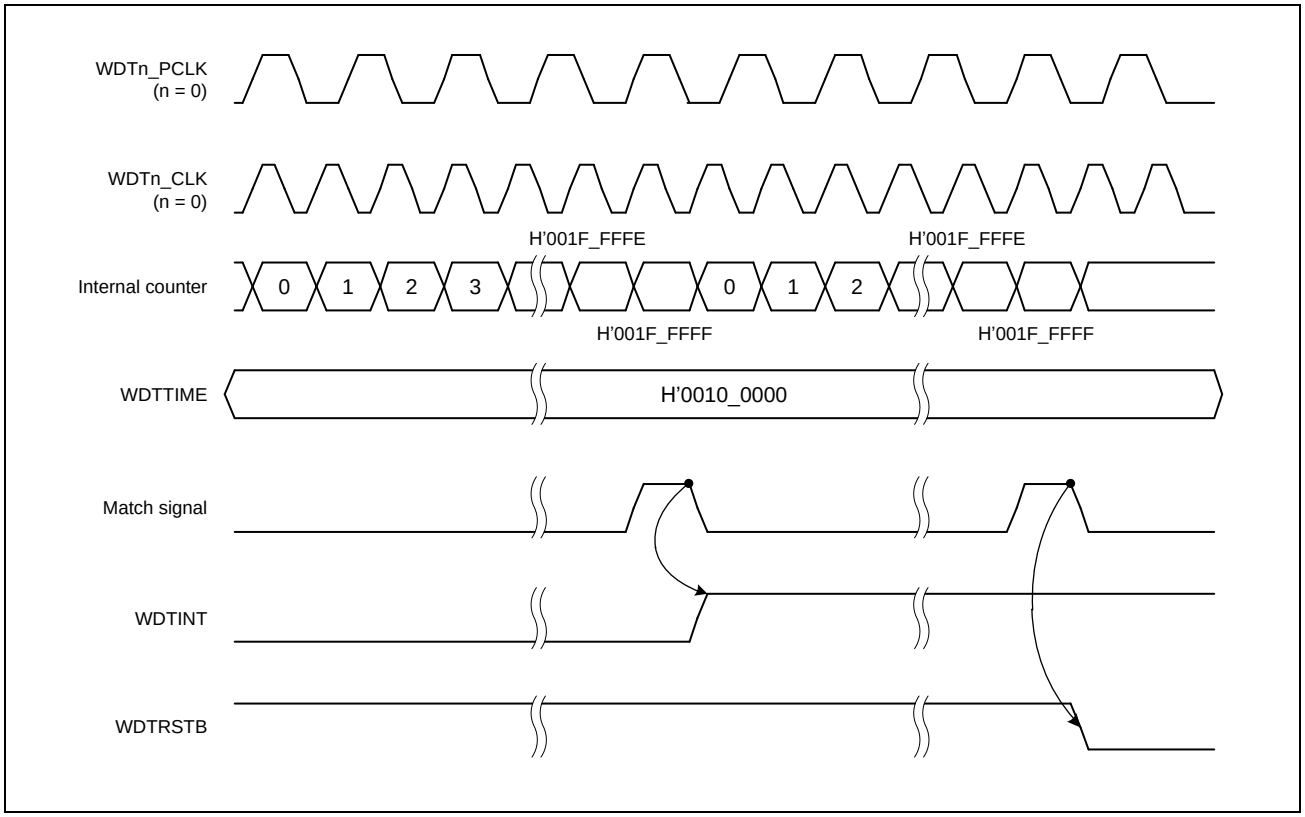



Figure 21.5 WDT Output Signal Operation Timing

### 21.4.2 WDT Interrupt Control Register (WDTINT) Specification for Write Access Interval

After the write access to the WDT interrupt control register is completed, the written data is synchronized with WDTn_CLK (n = 0), so a synchronization period is required. When performing write access to the WDTINT register, it is necessary to leave a write interval of 5 × P0ϕ + 5 × OSCCLK or more between the next write access.*1

**Note 1.** If the write interval is not observed, the interrupt request may not be cleared.
In this case, after making a write access and confirming that the interrupt factor has been cleared, perform the following write access.

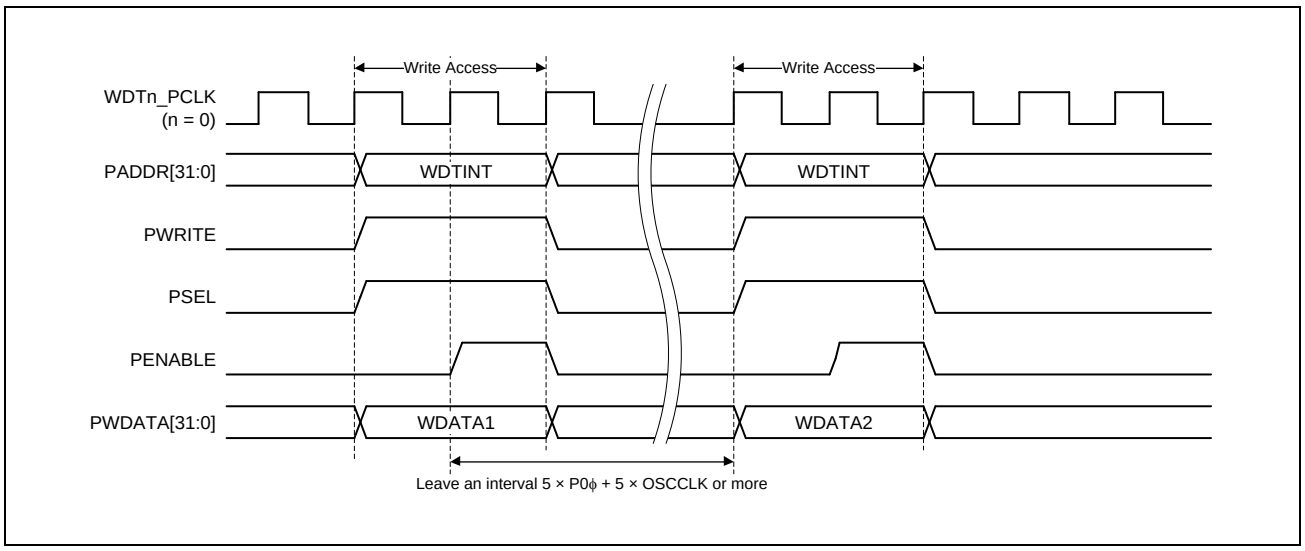



Figure 21.6 WDTINT Write Access Interval Regulation

### 21.4.3 Watchdog Timer Operation Reflection Timing of Register Setting Value

Each register other than the WDT interrupt control register (WDTINT) is always synchronized with WDTn_CLK (n = 0) at regular intervals. The timing at which the written data is reflected in the watchdog timer operation is in the range of small 2 × P0ϕ + 4 × OSCCLK) to large value (6 × P0ϕ +9 × OSCCLK).

**Figure 21.7** (Watchdog timer operation reflection timing 1 of register setting value) shows how write data is small and synchronized.

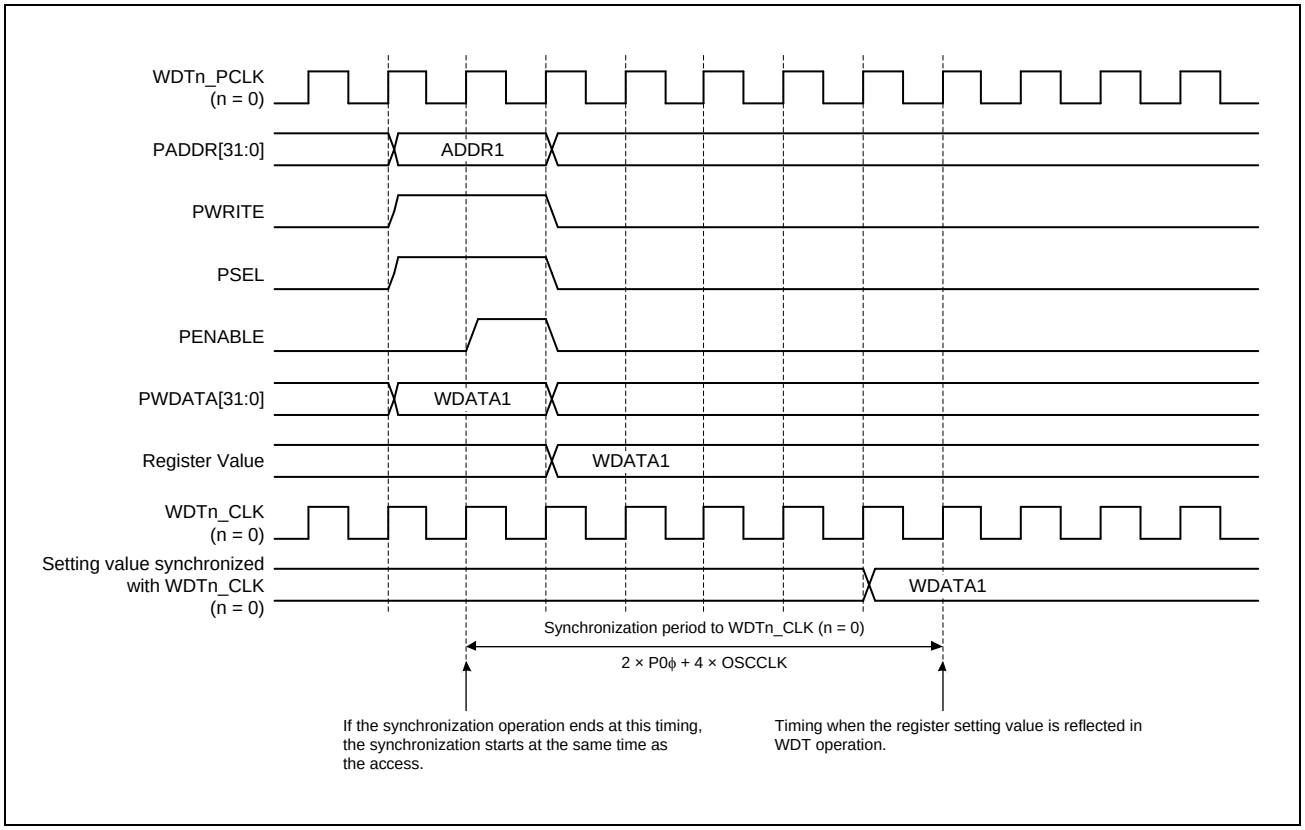



Figure 21.7 Watchdog Timer Operation Reflection Timing of Register Setting Value - 1

In addition, **Figure 21.8** (Watchdog timer operation reflection timing 2 of register setting value) shows how write data is large and synchronized.

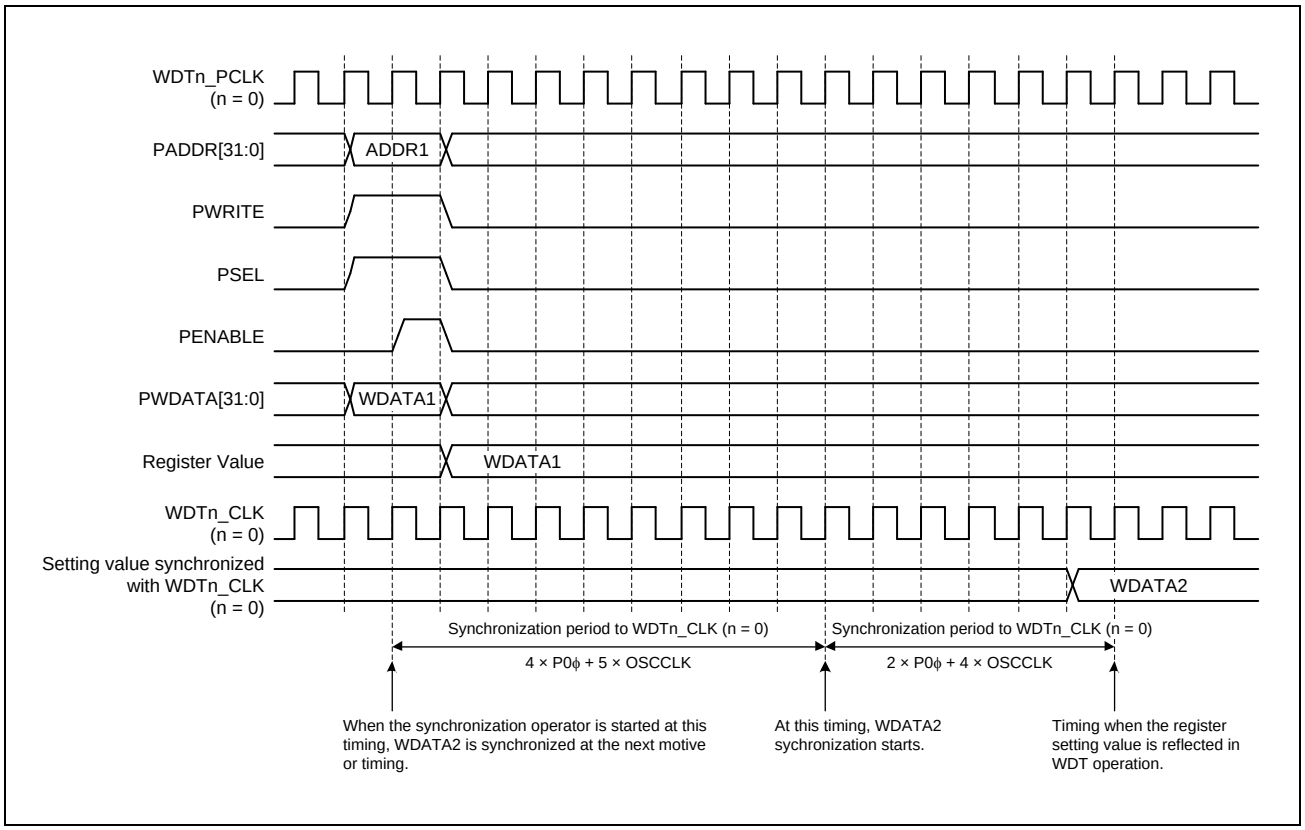



Figure 21.8 Watchdog Timer Operation Reflection Timing of Register Setting Value - 2

### 21.4.4 Timing of Reflecting WDT Counter Value to WDTTIM Register

The value stored in the WDTTIM register is the value obtained by synchronizing the counter value with P0φ. Therefore, the interval at which the value is updated in the WDTTIM register is 5 × OSCCLK+5 × P0ϕ at most, and the value before the actual counter value is read.

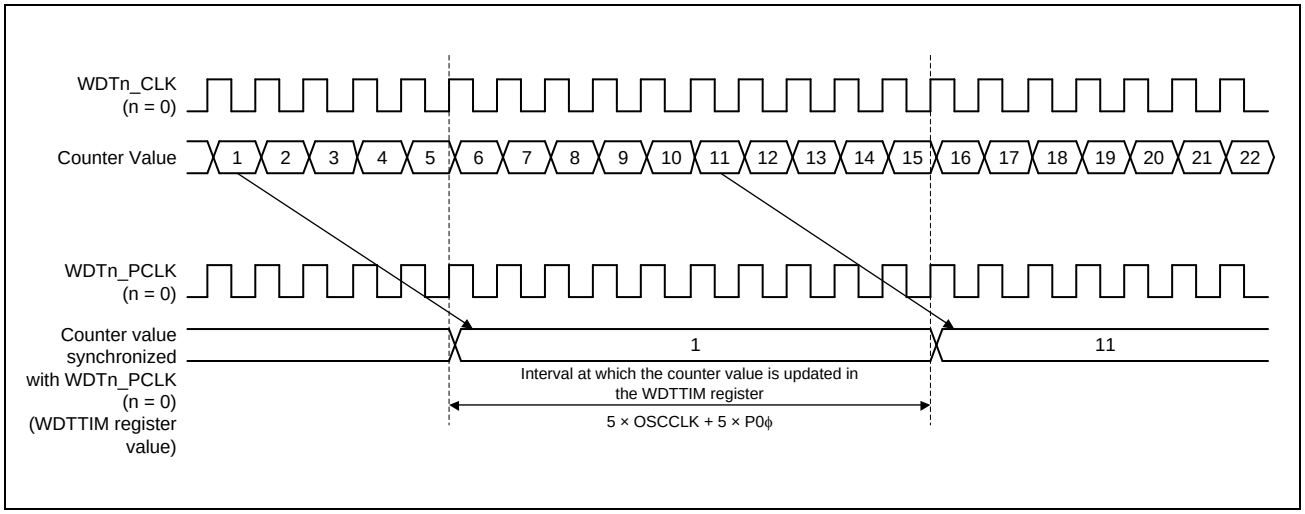



Figure 21.9 Timing of Reflecting the Counter Value to the WDTTIM Register

### 21.4.5 Conditions Under which WDTRSTB and PERROUT Operate

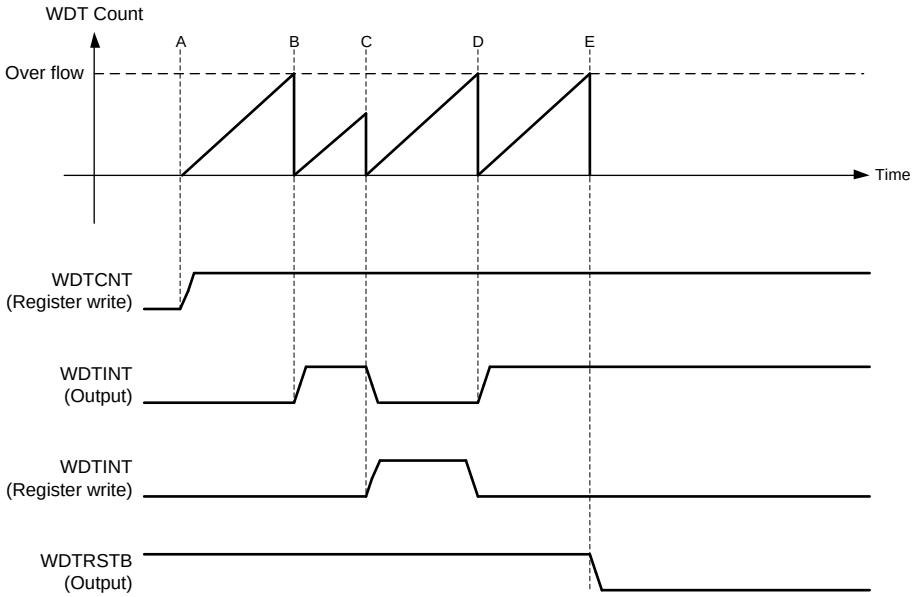
**Table 21.5** shows the conditions under which WDTRSTB and PERROUT operate.

When PEER = 0, WDTRSTB follows the watchdog timer operation described in **Section 21.4.1** to **Section 21.4.5** regardless of the occurrence of parity error. PERROUT holds 0. (No. 1, No. 2) No. 3 to No. 6 show the operating conditions when the parity error input is enabled by PEER = 1. When a parity error occurs (No. 3, No. 4), PESR holds the error status, and WDTRSTB or PERROUT is asserted depending on the value of PECR. If no parity error occurs, PESR = 0, WDTRSTB propagates the WDT-Core logic, and PERROUT is in the unasserted state. (No. 5, No. 6) When PEEN = 1 forces a parity error to occur, WDTRSTB or PERROUT is asserted depending on the PECR value regardless of the parity error.

Table 21.5 WDTRSTB/PERROUT Operating Conditions

| No. | PEERR | PEPO | PEER | PEEN | PECR | PESR | WDTRSTB | PERROUT |
|---|---|---|---|---|---|---|---|---|
| 1 | X | 1/0 | 0 | 0 | 0 | 0 (Fixed) | Original WDT | 0: de-assert |
| 2 | | 1/0 | 0 | 0 | 1 | 0 (Fixed) | Original WDT | 0: de-assert |
| 3 | With parity | 1/0 | 1 | 0 | 0 | 1 | 0: assert | 0: de-assert |
| 4 | | 1/0 | 1 | 0 | 1 | 1 | Original WDT | 1: assert |
| 5 | No parity | 1/0 | 1 | 0 | 0 | 0 | Original WDT | 0: de-assert |
| 6 | | 1/0 | 1 | 0 | 1 | 0 | Original WDT | 0: de-assert |
| 7 | X | X | X | 1 | 0 | X | 0: assert | 0: de-assert |
| 8 | | X | X | 1 | 1 | X | Original WDT | 1: assert |

The timing waveforms for No. 1 to No. 4 in **Table 21.5** are shown below.




**Note:** A) Set WDTCNT = 1 to start WDT counting.

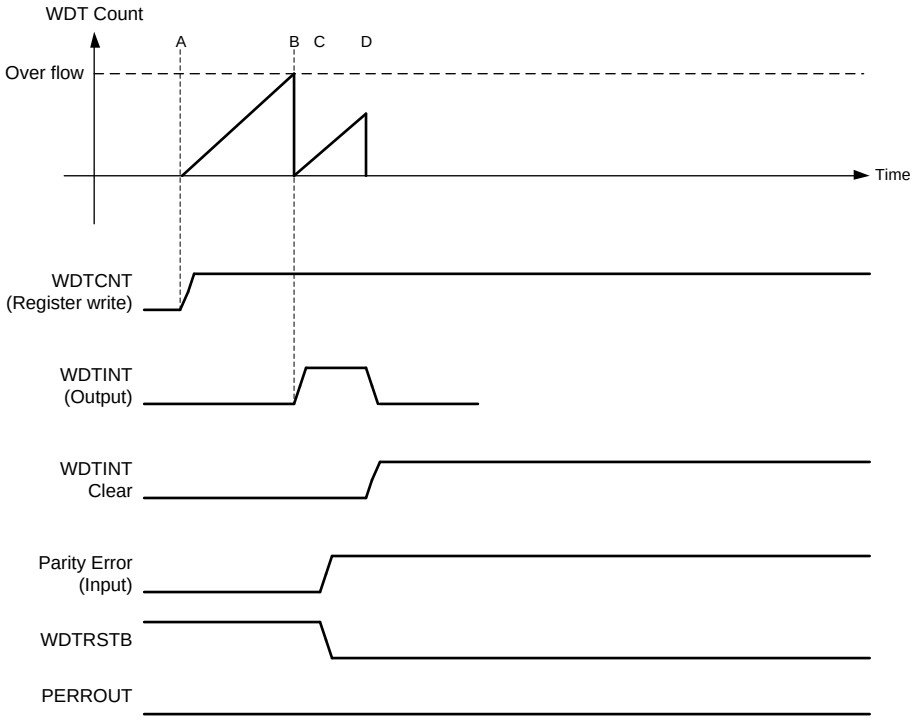
B) The interrupt request signal (WDTINT) is asserted at the first overflow of the WDT count.

C) If WDTINT (INTDISP) = 1 is written before the WDT count overflows, the interrupt request is cleared and the WDT count becomes zero.

D) Again, the interrupt request signal is asserted at the first overflow.

E) When the WDT count overflows for the second time, WDTRSTB is asserted.

Figure 21.10 Normal Watchdog Timer Operation No. 1/No. 2



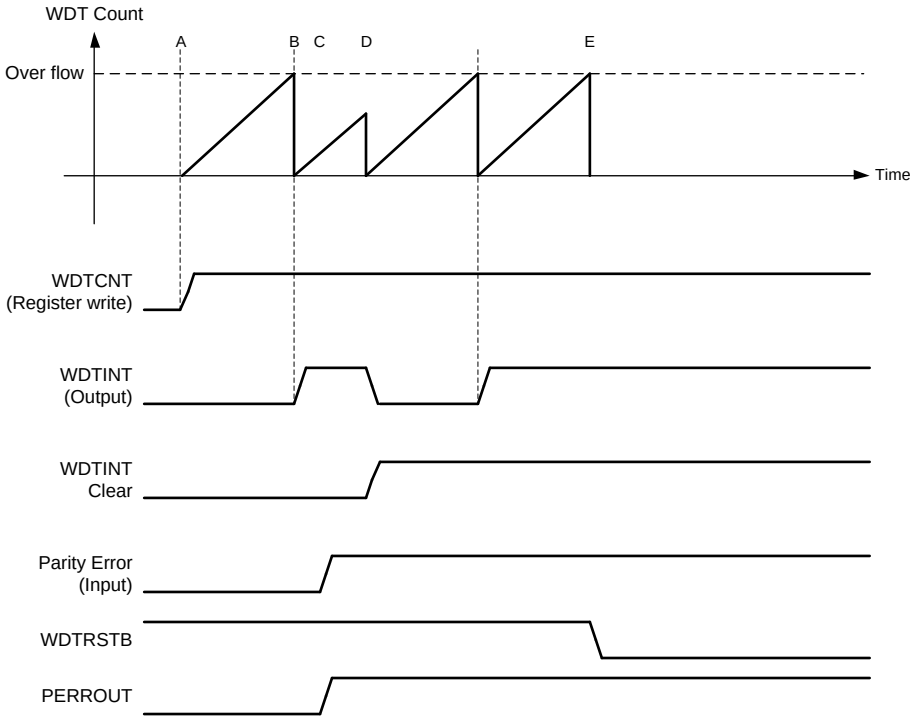

**Note:** A) Set WDTCNT = 1 to start WDT counting.

B) The interrupt request signal (WDTINT) is asserted at the first overflow of the WDT count.

C) When a parity error occurs, WDTRSTB = 0. At the same time, hold PERROUT = 0.

D) Even if WDTCount is reset by WDTINT = 1, WDTRSTB = 0 and PERROUT = 0 are retained.

Figure 21.11 Watchdog Timer Operation when a Parity Error Signal Occurs No. 3




**Note:** A) Set WDTCNT = 1 to start WDT counting.

B) The interrupt request signal (WDTINT) is asserted at the first overflow of the WDT count.

C) When a parity error occurs, WDTRSTB = 1. At the same time, hold PERROUT = 1.

D) Even if WDTCount is reset by WDTINT = 1, WDTRSTB = 1 and PERROUT = 1 are retained.

E) WDTRSTB = 0 is asserted due to WDT Count overflow.

Figure 21.12 Watchdog Timer Operation when a Parity Error Signal Occurs No. 4

## 21.5 Reset Control

### 21.5.1 WDTRST Control of Each WDT Channels

WDT CH0 generates WDTINT every cycle set in WDTSET. The CPU (Cortex-A55-Core0) receives the WDT CH0 interrupt signal WDTINT, clears the WDT CH0 counter at regular intervals, and suppresses the overflow of the WDT counter.

If the counter is not cleared before the cycle set for each WDT elapses, the uncleared WDT sends a reset request signal to the CPG. Based on the setting of the built-in register CPG_WDTRST_SEL, CPG can select whether to reset only the CPU corresponding to the WDT that caused the reset from the WDT reset request signal from each channel, or to reset the entire system. is. You can also choose whether to assert the WDTOVF_PERROUT# pin to notify the outside world that the entire system has been reset.

(For details about CPG_WDTRST_SEL register in CPG, refer to the CPG chapter.)

Table 21.6 WDT Reset Target by CPG_WDTRST_SEL Register

| Bit | Initial Value | R/W | Reset Factor | Function |
|---|---|---|---|---|
| CPG_WDTRST_SEL[0] | 0 | R/W | WDT CH0 | When the target WDT executes a reset request due to overflow etc.<br>0: Mask system reset<br>1: System reset implementation |
| CPG_WDTRST_SEL[4] | 0 | R/W | WDT CH0 | When the target WDT executes a reset request due to overflow etc.<br>0: Mask WDTOVF_PERROUT assert<br>1: WDTOVF_PERROUT assert |
| CPG_WDTRST_SEL[8] | 0 | R/W | WDT CH0 | When either WDT CH0 makes a reset request<br>0: Mask the Cold-Reset of Cortex-A55<br>1: Implemented Cortex-A55 Cold-Reset |

When the WDT system reset is executed, it is also possible to set the CPG_WDTRST_SEL[4] bit to notify the system outside the LSI from the WDTOVF_PERROUT# pin of the GPIO pin that the system reset has occurred due to the WDT overflow. When using WDTOVF_PERROUT# as a signal to notify the system reset to the outside, set the CPG_WDTRST_SEL[4] bit to 1 when CPG_WDTRST_SEL[0] is 1 as shown below.

CPG_WDTRST_SEL[4] Set 1 when CPG_WDTRST_SEL[0] = 1

For example, if CPG_WDTRST_SEL[0] = 0 and WDTRST_SEL[4] = 1 are set, the system reset will not be executed even if WDT CH0 asserts a reset request because CPG_WDTRST_SEL[0] = 0, but WDTRST_SEL[4] = 1 so WDTOVF_PERROUT# pin will be asserted. In other words, the system reset execution status and WDTOVF_PERROUT do not match.

## 21.6 Usage Note

[Cautions]

- When setting a value in the WDTTIM register, do not set a value that is WDTSET register + H'F_FFFF.
- The reset request signal WDTRSTB and WDT interrupt signal WDTINT cannot be masked.

# 22. Serial Communications Interface with FIFO (SCIFA)

This LSI has five channels of serial communication interface (SCIFA) with FIFO that support both asynchronous and clock synchronous serial communication. The SCIFA has 16-stage FIFO buffers for transmission and reception, respectively, for each channel that enable this LSI to perform efficient high-speed continuous communication.

## 22.1 Overview

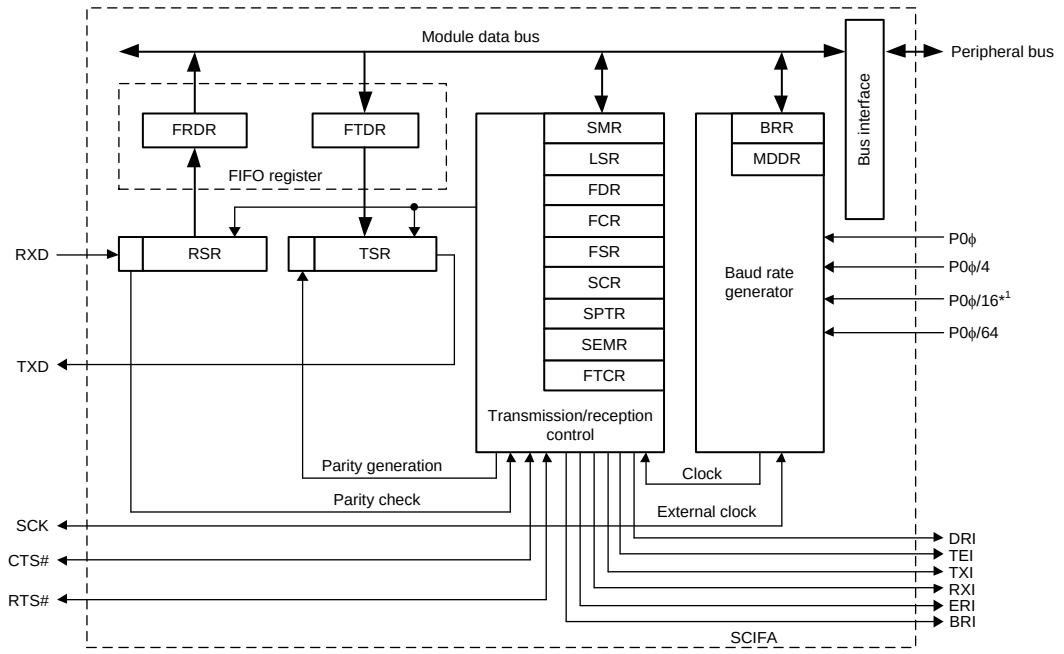
**Table 22.1** lists the specifications of the SCIFA.

Table 22.1 Specifications of SCIFA

| Item | | Description |
|---|---|---|
| Channel | | 5 channels (ch 0,1,2,3,4) |
| Serial communication method | | Asynchronous communication mode and clock synchronous communication mode |
| Transfer speed | | Selectable bit rate with an on-chip baud rate generator |
| Full duplex communication | | Transmitting section: realizes continuous data transmission using 16-stage FIFO buffer<br>Receiving section: realizes continuous data reception using 16-stage FIFO buffer |
| Data transmission | | Selectable either LSB-first or MSB-first transfer |
| Interrupt source | | The following six sources:<br>• Transmit-end (TEIF)<br>• Transmit-FIFO-data-empty (TXIF)<br>• Receive-FIFO-data-full (RXIF)<br>• Receive-data-ready (DRIF)*1<br>• Receive-error (ERIF)<br>• Break detection or overrun (BRIF) |
| Asynchronous communication mode | Character length | 7 or 8 bits |
| | Transmission stop bit length | 1 or 2 bits |
| | Parity | Even, odd, or none |
| | Receive error detection | Parity, overrun, and framing errors |
| | Hardware flow control | Controls data transmission and reception using the CTS# and RTS# pins.<br>Not applicable to channel 3 (SCIF3) and channel 4 (SCIF4) |
| | Break detection | Break signal detection function by hardware. |
| | Clock source | Selectable from internal or external clock |
| | Noise cancellation | Incorporates a digital noise filter in the RXD pin input path. |
| Clock synchronous communication mode | Character length | 8 bits |
| | Receive error detection | Detects an overrun error as a receive error. |
| | Clock source | Selectable either internal or external clock |
| Bit rate modulation | | Enables errors to be decreased by correcting the output of the on-chip baud rate generator. |

Note 1. Effective only for asynchronous communication mode

**Figure 22.1** shows a block diagram of the SCIFA.




RSR: Receive shift register
FRDR: Receive FIFO data register
TSR: Transmit shift register
FTDR: Transmit FIFO data register
SMR: Serial mode register
SCR: Serial control register
FSR: Serial status register
MDDR: Modulation duty register
BRR: Bit rate register
SPTR: Serial port register
FCR: FIFO control register
FDR: FIFO data count register
LSR: Line status register
SEMR: Serial extended mode register
FTCR: FIFO trigger control register

Note 1. The Pϕ/16 clock is not input in the case of the channel 0 interface.

Figure 22.1 Block Diagram of SCIFA

**Table 22.2** lists the input/output pins of the SCIFA.

Table 22.2 Pin Configuration of the SCIFA

| Channel | Item | Pin Name | I/O | Function |
|---|---|---|---|---|
| 0 to 4 (n = 0-4) | Serial clock pin | SCIFn_SCK | I/O | Transmission/reception clock input/output, general output |
| | Receive data pin | SCIFn_RXD | Input | Receive data input |
| | Transmit data pin | SCIFn_TXD | Output | Transmit data output |
| 0 to 2 (n = 0-2) | Transmission/reception start control pin | SCIFn_CTS# | I/O | Input for hardware flow control (transmission enable signal) / general output |
| | | SCIFn_RTS# | Output | Output for hardware flow control (transmission request signal) / general output |

**Note:** Channels of each pin is omitted.

## 22.2 Register Descriptions

Table 22.3 Base Address Table

| Address Space | Channel 0 | Channel 1 | Channel 2 | Channel 3 | Channel 4 |
|---|---|---|---|---|---|
| Cortex-A55 Address Space | H'0_1004_B800 | H'0_1004_BC00 | H'0_1004_C000 | H'0_1004_C400 | H'0_1004_C800 |

Table 22.4 List of Registers

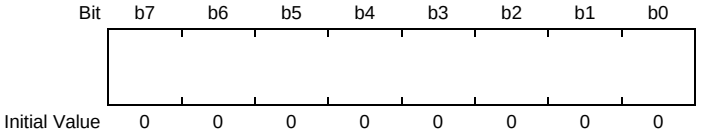
| Register Name | Abbreviation | R/W | Initial Value | Offset Address | Access Size |
|---|---|---|---|---|---|
| Serial mode register | SMR | R/W | H'0000 | H'00 | 16 |
| Bit rate register | BRR | R/W | H'FF | H'02 | 8 |
| Modulation duty register | MDDR | R/W | H'FF | H'02 | 8 |
| Serial control register | SCR | R/W | H'0000 | H'04 | 16 |
| Transmit FIFO data register | FTDR | W | Undefined | H'06 | 8 |
| Serial status register | FSR | R/W | H'0020 | H'08 | 16 |
| Receive FIFO data register | FRDR | R | Undefined | H'0A | 8 |
| FIFO control register | FCR | R/W | H'0000 | H'0C | 16 |
| FIFO data count register | FDR | R | H'0000 | H'0E | 16 |
| Serial port register | SPTR | R/W | H'00xx | H'10 | 16 |
| Line status register | LSR | R/W | H'0000 | H'12 | 16 |
| Serial extended mode register | SEMR | R/W | H'00 | H'14 | 8 |
| FIFO trigger control register | FTCR | R/W | H'1F1F | H'16 | 16 |

**Note:** BRR and MDDR are located in the same address. Setting the MDDRS bit of the SEMR register switches these registers.

### 22.2.1 Receive Shift Register (RSR)

The RSR register receives serial data and temporally stores the data. The SCIFA stores the serial data input via the RXD pin into the RSR register and converts the data to the parallel form. When one byte of data has been received, it is automatically transferred to the receive FIFO data register (FRDR).

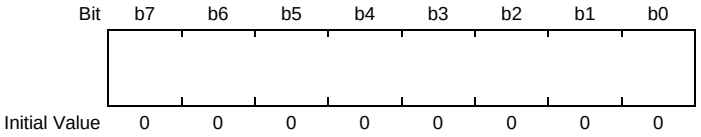
The CPU cannot read from or write to the RSR register directly.




### 22.2.2 Receive FIFO Data Register (FRDR)

The FRDR register is a 8-bit, 16-stage FIFO register that stores the received serial data. When the SCIFA receives one byte of serial data, it transfers the received data from the receive shift register (RSR) to the FRDR register and completes the receive operation. Continuous reception is possible until the received 16 bytes of data are stored. If the FRDR register is read when there is no received data in the FRDR register, an undefined value is read.

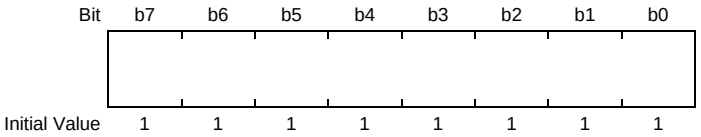
When the FRDR register is full of received data, subsequently received serial data is lost.




### 22.2.3 Transmit Shift Register (TSR)

The SCIFA transfers the transmit data from the transmit FIFO data register (FTDR) to the TSR register, and then transmits the data serially to the TXD pin. After transmitting one byte of data, the SCIFA automatically transfers the next transmit data from the FTDR register into the TSR register and starts transmission.

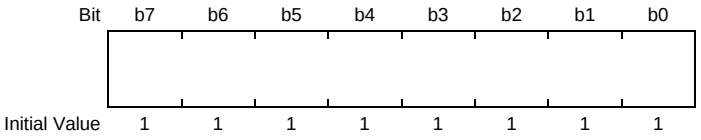
The CPU cannot read from or write to the TSR register directly.



### 22.2.4 Transmit FIFO Data Register (FTDR)

The FTDR register is a 8-bit, 16-stage FIFO register that stores serial transmission data. When the SCIFA detects that the transmit shift register (TSR) is empty, it transmits data written in the FTDR register to the TSR register and starts serial transmission. Continuous serial transmission is executed until there is no transmit data left in the FTDR register. Writing the transmit data to the FTDR register should be done when a transmit data empty interrupt (TXI) request is generated. When the FTDR register becomes full of transmit data (16 bytes), no more data can be written. Even if new data is written, the data is ignored.

CPU can read from the FTDR register but cannot write to it.



### 22.2.5 Serial Mode Register (SMR)

The SMR register specifies the SCIFA serial communication format and selects the clock source for the baud rate generator.

The CPU can always read and write to the SMR register.

| Bit | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | CM | CHR | PE | PM | 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/W | R/W | R/W | R/W | 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 |
|---|---|---|---|---|
| b1, b0 | CKS[1:0] | All 0 | R/W | Clock Select<br>b1 b0<br>0 0: 1 × Pϕ*1<br>0 1: 1/4 × Pϕ*1<br>1 0: 1/16 × Pϕ*1,*2<br>1 1: 1/64 × Pϕ*1 |
| b2 | — | 0 | R/W | Reserved<br>This bit is read as 0. The write value should be 0. |
| b3 | STOP | 0 | R/W | Stop Bit Length<br>0: One stop bit<br>1: Two stop bits |
| b4 | PM | 0 | R/W | Parity Mode<br>0: Even parity<br>1: Odd parity |
| b5 | PE | 0 | R/W | Parity Enable<br>0: Parity bit addition or check is disabled.<br>1: Parity bit addition or check is enabled. |
| b6 | CHR | 0 | R/W | Character Length<br>0: 8-bit data<br>1: 7-bit data*3 |
| b7 | CM | 0 | R/W | Communication Mode<br>0: Asynchronous mode<br>1: Clock synchronous mode |
| b15 to b8 | — | All 0 | R/W | Reserved<br>These bits are read as 0. The write value should be 0. |

Note 1. Pϕ: SCIFn_CLK_PCK (n = 0 to 4)

Note 2. This setting is not applicable to channel 0. It is only applicable to channels 1 to 4.

Note 3. When 7-bit data is selected, the MSB (bit 7) of the transmit FIFO data register is not transmitted.

#### CKS[1:0] Bits (Clock Select)

Select an internal clock source for the on-chip baud rate generator. For further information on the clock source, bit rate register settings, and baud rates, see **Section 22.2.8, Bit Rate Register (BRR)**.

#### STOP Bit (Stop Bit Length)

Selects one bit 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.

*Note*: When transmitting with one stop bit, a single 1 bit (stop bit) is added at the end of each transmission character.

*Note*: When transmitting with two stop bits, two 1 bits (stop bits) are added at the end of each transmission character.

#### PM Bit (Parity Mode)

Selects either the even or odd parity check. The setting of this bit is effective only when the parity enable (PE) bit of this register is set to 1 in asynchronous mode. The setting of this bit is ignored in clock synchronous mode, or when parity addition/check is disabled in asynchronous mode.

*Note*: If even parity is selected, the parity bit is added to data to be transmitted to make the total number of 1s even in the transmission character and parity bit combined. When receiving, the SCIFA verifies that the total number of 1s in the received character and parity bit combined is even.

*Note*: If odd parity is selected, the parity bit is added to data to be transmitted to make the total number of 1s odd in the transmission character and parity bit combined. When receiving, the SCIFA verifies that the total number of 1s in the received character and parity bit combined is odd.

#### PE Bit (Parity Enable)

Selects whether to add a parity bit on data transmission and whether to enable/disable the parity check on data reception in asynchronous mode. In clock synchronous mode, a parity bit is neither added nor checked, regardless of the setting of this bit.

*Note*: When this bit is set to 1, an even or odd parity bit specified in the PM bit is added to data to be transmitted. The SCIFA verifies whether the parity bit of the received data is even or odd as specified in the PM bit when receiving.

#### CHR Bit (Character Length)

Selects 7- or 8-bit data length in asynchronous mode. In clock synchronous mode, the data length is always 8 bits, regardless of the CHR setting.

#### CM Bit (Communication Mode)

Selects whether the SCIFA operates in asynchronous or clock synchronous mode.

### 22.2.6 Serial Control Register (SCR)

The SCR register enables or disables the SCIFA transmission/reception and interrupt requests, and selects the transmit/receive clock source. The CPU can always read from and write to the SCR register.

| Bit | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | TIE | RIE | TE | RE | REIE | TEIE | CKE[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 |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| b1, b0 | CKE[1:0] | All 0 | R/W | Clock Enable<br>In asynchronous mode:<br>b1 b0<br>0 0: Internal clock or SCK pin is used for input pin (input signal is ignored). The SCK pin state depends on the SCKIO and SCKDT bits in SPTR.<br>0 1: Internal clock or SCK pin is used for clock output (The output clock frequency is 16 or 8 times of the bit rate).<br>1 0: External clock or SCK pin is used for clock input (The input clock frequency is 16 or 8 times of the bit rate).<br>1 1: Setting prohibited<br>In clock synchronous mode:<br>b1 b0<br>0 0: Internal clock or SCK pin is used for synchronous clock output.<br>0 1: Internal clock or SCK pin is used for synchronous clock output.<br>1 0: External clock or SCK pin is used for synchronous clock input.<br>1 1: Setting prohibited |
| b2 | TEIE*1 | 0 | R/W | Transmit End Interrupt Enable<br>0: Transmit end interrupt (TEI) request is disabled.<br>1: Transmit end interrupt (TEI) request is enabled. |
| b3 | REIE | 0 | R/W | Receive Error Interrupt Enable<br>0: Receive-error interrupt (ERI) and break interrupt (BRI) requests are disabled.<br>1: Receive-error interrupt (ERI) and break interrupt (BRI) requests are enabled. |
| b4 | RE | 0 | R/W | Receive Enable<br>0: Data reception is disabled.<br>1: Data reception is enabled. |
| b5 | TE | 0 | R/W | Transmit Enable<br>0: Data transmission is disabled.<br>1: Data transmission is enabled. |
| b6 | RIE | 0 | R/W | Receive Interrupt Enable<br>0: Receive-FIFO-data-full interrupt (RXI), receive-data ready interrupt (DRI), receive-error interrupt (ERI), and break interrupt (BRI) requests are disabled.<br>1: Receive-FIFO-data-full interrupt (RXI), receive-data ready interrupt (DRI), receive-error interrupt (ERI), and break interrupt (BRI) requests are enabled. |
| b7 | TIE | 0 | R/W | Transmit Interrupt Enable<br>0: Transmit-FIFO-data-empty interrupt request (TXI) is disabled.<br>1: Transmit-FIFO-data-empty interrupt request (TXI) is enabled. |
| b15 to b8 | — | All 0 | R/W | Reserved<br>These bits are read as 0. The write value should be 0. |

Note 1. TEI interrupt requests can be cleared by reading 1 from the TEND flag, and then clearing the setting to 0, or by setting the TEIE bit to 0.

### CKE[1:0] Bits (Clock Enable)

Select the SCIFA clock source and enable or disable clock output from the SCK pin. Depending on the settings of these bits, the SCK pin can be used for serial clock output or serial clock input. If the SCK pin is set for the synchronous clock output in the clock synchronous mode, set the CM bit in the SMR register to 1, and then set the CKE[1:0] bits. The settings of the CKE[1:0] bits are listed in **Table 22.17**.

### REIE Bit (Receive Error Interrupt Enable)

Specifies whether to enable or disable a receive-error interrupt (ERI) request and a break interrupt (BRI) request. The setting of this bit is only valid when the RIE bit is set to 0.

*Note*: ERI interrupt requests can be cleared by reading 1 from the ER bit in the FSR register, and then clearing the setting to 0, or by clearing both the RIE and REIE bits in this register to 0. BRI interrupt requests can be cleared by reading 1 from the BRK bit in the FSR register, or from the ORER flag in the LSR register, and then clearing the setting to 0, or by clearing both the RIE and REIE bits in this register to 0.

### RE Bit (Receive Enable)

Specifies whether to enable or disable the serial data reception.

*Note*: Setting this bit to 0 does not affect the receive flags (DR, ER, BRK, RDF, FER, and PER in the FSR register, and ORER in the LSR register). These flags retain their previous values.

*Note*: Serial reception starts when a start bit is detected in asynchronous mode, or a synchronous clock input is detected in clock synchronous mode. Before setting this bit to 1, be sure to set the serial mode register (SMR) and the FIFO control register (FCR) to select the receive format and reset the receive FIFO.

### TE Bit (Transmit Enable)

Specifies whether to enable or disable the serial data transmission.

*Note*: Serial transmission starts after writing of data to be transmitted into the FTDR register under this condition. Before setting this bit to 1, be sure to set the serial mode register (SMR) and the FIFO control register (FCR) to select the transmit format and reset the transmit FIFO.

### RIE Bit (Receive Interrupt Enable)

Specifies whether to enable or disable a receive-FIFO-data-full (RXI) interrupt request when the RDF flag in the serial status register (FSR) is set to 1, a receive-data ready (DRI) interrupt request when the DR flag in the FSR register is set to 1, a receive-error (ERI) interrupt request when the ER flag in the FSR register is set to 1, and a break (BRI) interrupt request when the BRK flag in the FSR register or the ORER flag in the line status register (LSR) is set to 1.

*Note*: RXI interrupt requests can be cleared by reading 1 from the DR or RDF flag in the FSR register, then clearing the flag to 0, or by clearing the RIE bit to 0. DRI interrupt requests can be cleared by reading 1 from the DR flag in the FSR register, and then clearing the setting to 0, or by clearing the RIE bit in this register to 0. Receive error interrupt (ERI) requests and break interrupt (BRI) requests can be cleared by clearing both the RIE and REIE bits in this register to 0.

### TIE Bit (Transmit Interrupt Enable)

Specifies whether to enable or disable a transmit-FIFO-data-empty interrupt (TXI) request when the serial transmit data is transferred from the transmit FIFO data register (FTDR) into the transmit shift register (TSR), the quantity of data in the transmit FIFO data register falls below the specified trigger number for transmission, and the TDFE flag in the serial status register (FSR) is set to 1.

*Note*: TXI interrupt requests can be cleared either by writing a greater quantity of transmit data than the specified transmission trigger number into the FTDR register, reading 1 from the TDFE flag, and then clearing the TDFE flag to 0, or by clearing this bit to 0.

### 22.2.7 Serial Status Register (FSR)

The FSR register is a 16-bit register. The 8 lower-order bits indicate the status flag representing the SCIFA operating state.

The CPU can always read and write to the FSR register, but cannot write 1 to the status flags (ER, TEND, TDFE, BRK, RDF, and DR bits) in this register. These flags can be only cleared to 0 when they have first been read (after being set to 1). b3 (FER) and b2 (PER) are read-only bits that cannot be written.

| Bit | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | ER | TEND | TDFE | BRK | FER | PER | RDF | DR |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 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)*1 | R(/W)*1 | R(/W)*1 | R(/W)*1 | R | R | R(/W)*1 | R(/W)*1 |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| b0 | DR | 0 | R/(W)*1 | Receive Data Ready Flag<br>0: Reception is in progress, or no received data has remained in the FRDR register after normally completed receiving.<br>1: Next receive data has not been received. |
| b1 | RDF | 0 | R/(W)*1 | Receive FIFO Data Full Flag<br>0: The quantity of receive data in the FRDR register falls below the specified reception trigger number.<br>1: The quantity of receive data written in the FRDR register is equal to or greater than the specified reception trigger number. |
| b2 | PER | 0 | R | Parity Error Flag*4<br>0: No receive parity error occurred in the next receive data read from the FRDR register.<br>1: A receive parity error occurred in the next receive data read from the FRDR register. |
| b3 | FER | 0 | R | Framing Error Flag*4<br>0: No receive framing error occurred in the next data read from the FRDR register.<br>1: A receive framing error occurred in the next data read from the FRDR register. |
| b4 | BRK | 0 | R/(W)*1 | Break Detect Flag<br>0: No break signal is received.<br>1: A break signal is received.*2 |
| b5 | TDFE | 1 | R/(W)*1 | Transmit FIFO Data Empty Flag<br>0: The quantity of transmit data written in the FTDR register exceeds the specified transmission trigger number.<br>1: The quantity of transmit data written in the FTDR register is equal to or less than the specified transmission trigger number.*3 |
| b6 | TEND | 0 | R/(W)*1 | Transmit End Flag<br>0: Transmission is in the waiting state or in progress.<br>1: Transmission is completed. |
| b7 | ER | 0 | R/(W)*1 | Receive Error Flag<br>0: Reception is in progress or has normally completed.<br>1: A framing error or parity error has occurred during reception. |
| b15 to b8 | — | All 0 | R/W | Reserved<br>These bits are read as 0. The write value should be 0. |

Note 1. 0 can be only written to clear the flag after 1 is read.

Note 2. When a break signal is detected, transfer of the receive data (H'00) to the FRDR register stops after the detection. When the break ends and the receive signal becomes mark state (high level), the transfer of receive data resumes.

Note 3. Since the FTDR register is a 16-byte FIFO register, the maximum quantity of data that can be written when TDFE is 1 is "16 minus the number of non-transmitted data units". If additional data is written, the data is ignored. The quantity of data in the FTDR register is indicated by the 8 higher-order bits of the FDR register.

Note 4. When the DMAC is used to read data, the generation of errors cannot be checked by reading this flag.

### DR Bit (Receive Data Ready Flag)

Indicates that the quantity of data stored in the receive FIFO data register (FRDR) falls below the specified reception trigger number, and that no next data has been received yet after the elapse of 15 ETUs from the last stop bit in asynchronous mode. In clock synchronous mode, this bit is not set to 1.

[Setting condition]

- DR is set to 1 when the FRDR register contains less data than the specified reception trigger number, and no next data has been received yet after the elapse of 15 ETUs*[1] from the last stop bit.

[Clearing conditions]

When either of the following is satisfied:

- DR is cleared to 0 when DR = 1 is read and then 0 is written to the DR flag.
- DR is cleared to 0 when all received data in the FRDR register are read.

**Note 1.** This is equivalent to one and a half (1.5) frames in the 8-bit format with one stop bit (ETU: elementary time unit).

*Note*: When the RE bit in SCR is cleared to 0, the DR bit is not affected and retains its previous value.

### RDF Bit (Receive FIFO Data Full Flag)

Indicates that receive data has been transferred to the receive FIFO data register (FRDR), and the quantity of data in FRDR becomes equal to or greater than the specified reception trigger number.

[Setting condition]

- RDF is set to 1 when the quantity of receive data which is equal to or greater than the specified reception trigger number are stored in the FRDR register*[1].

[Clearing conditions]

- RDF is cleared to 0 when RDF = 1 is read and then 0 is written to this bit.
- RDF is cleared to 0 when the FRDR register is read.

**Note 1.** Since the FRDR register is a 16-byte FIFO register, the maximum quantity of data that can be read when this bit is 1 is equivalent to the specified reception trigger number. If an attempt is made to read after all the data in the FRDR register has been read, the read data is undefined. The quantity of receive data in the FRDR register is indicated by the 8 lower-order bits of the FDR register.

#### PER Bit (Parity Error Flag)

Indicates whether there is a parity error in the data read from the receive FIFO data register (FRDR) in asynchronous mode.

[Setting condition]

- PER is set to 1 when a parity error is present in the next data read from the FRDR register.

[Clearing condition]

- PER is cleared to 0 when no parity error is present in the next data read from the FRDR register.

#### FER Bit (Framing Error Flag)

Indicates whether there is a framing error in the data read from the receive FIFO data register (FRDR) in asynchronous mode.

[Setting condition]

- FER is set to 1 when a framing error is present in the next data read from the FRDR register.

[Clearing condition]

- FER is cleared to 0 when no framing error is present in the next data read from the FRDR register.

#### BRK Bit (Break Detect Flag)

Indicates that a break signal has been detected in receive data.

[Setting condition]

- BRK is set to 1 when data including a framing error is received, and the framing error is followed by at least one frame of data received at the space 0 level (low level).

[Clearing condition]

- BRK is cleared to 0 when software reads BRK after it has been set to 1 and then writes 0 to BRK.

#### TDFE Bit (Transmit FIFO Data Empty Flag)

Indicates that data has been transferred from the transmit FIFO data register (FTDR) into the transmit shift register (TSR), the quantity of data in the FRDR register becomes equal to or less than the specified transmission trigger number, and writing of transmit data to the FRDR register is enabled.

[Setting conditions]

When either of the following is satisfied:

- TDFE is set to 1 when the TE bit in SCR is 0.
- TDFE is set to 1 when the quantity of transmit data written in the FRDR register is equal to or less than the specified transmission trigger number.

[Clearing conditions]

- TDFE is cleared to 0 when 0 is written in the TDFE bit after reading TDFE = 1.
- When transmit data is written to the FTDR register

### TEND Bit (Transmit End Flag)

Indicates that the FRDR register contains no more valid data and transmission is completed when transmitting the last bit of the transmit data.

[Setting condition]

When the following is satisfied:

- TEND is set to 1 when the FTDR register does not contain transmit data when the last bit of the serial transmission data is transmitted.

[Clearing conditions]

- When transmit data is written to the FTDR register
- When 0 is written to TEND after it has been read as 1

### ER Bit (Receive Error Flag)

Indicates the occurrence of a framing error, or of a parity error when receiving the parity-added data*[1].

[Setting conditions]

When either of the following is satisfied:

- ER is set to 1 when the stop bit is found to be 0 after checking whether the stop bit of the received data is 1 at the end of one data receive operation*[1].
- ER is set to 1 when the total number of 1s in the received data and parity bit combined does not match the even or odd parity setting specified by the PM bit in the SMR register.

[Clearing condition]

- When 0 is written to ER after it has been read as 1

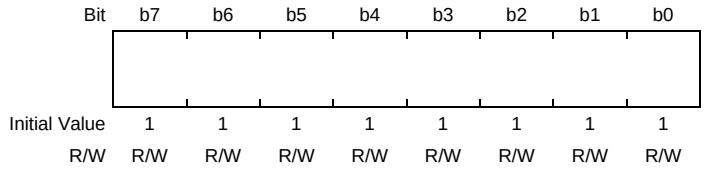
**Note 1.** Clearing the RE bit to 0 in the SCR register does not affect this bit, which retains its previous value. Even if a receive error occurs, the receive data is transferred to the FRDR register and the receive operation is continued. Whether the data read from the FRDR register includes a receive error can be detected by the FER and PER bits in the FSR register.

*Note*: In two stop bits mode, only the first stop bit is checked; the second stop bit is not checked.

### 22.2.8 Bit Rate Register (BRR)

The BRR register is an 8-bit register that, together with the baud rate generator clock source selected by the CKS[1:0] bits in the serial mode register (SMR), determines the serial transmit/receive bit rate.

This register is located in the same address as that of the MDDR register and selected when the MDDRS bit in SEMR is 0. The CPU can read and write to BRR. Writing to BRR should be executed when TE = RE = 0 in the SCR register.




The BRR setting is calculated using the following formulae.

[Asynchronous mode]

- When the baud rate generator is in normal mode (SEMR.BGDM = 0):

$$N = \frac{P0\phi}{64 \times 2^{2n-1} \times B} \times 10^6 - 1$$

(When operating on the base clock with a frequency 16 times the bit rate (SMER.ABCS0 = 0))

$$N = \frac{P0\phi}{32 \times 2^{2n-1} \times B} \times 10^6 - 1$$

(When operating on the base clock with a frequency 8 times the bit rate (SMER.ABCS0 = 1))

- When the baud rate generator is in double-speed mode (SEMR.BGDM = 1):

$$N = \frac{P0\phi}{32 \times 2^{2n-1} \times B} \times 10^6 - 1$$

(When operating on the base clock with a frequency 16 times the bit rate (SMER.ABCS0 = 0))

$$N = \frac{P0\phi}{16 \times 2^{2n-1} \times B} \times 10^6 - 1$$

(When operating on the base clock with a frequency 8 times the bit rate (SMER.ABCS0 = 1))

[Clock synchronous mode]

$$N = \frac{P0\phi}{8 \times 2^{2n-1} \times B} \times 10^6 - 1$$

B: Bit rate (bit/s)
N: Setting of the BRR register ($0 \leq N \leq 255$) (The setting must satisfy the electrical characteristics).
P0ϕ: 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 22.5**).

NOTE

The MDDR register is used to adjust the bit rate. For details, see **Section 22.2.9, Modulation Duty Register (MDDR)**.

Table 22.5 SMR Register Setting

| n | Clock Source | SMR Register Settings | |
|---|---|---|---|
| | | CKS1 | CKS0 |
| 0 | P0ϕ | 0 | 0 |
| 1 | P0ϕ/4 | 0 | 1 |
| 2 | P0ϕ/16 | 1 | 0 |
| 3 | P0ϕ/64 | 1 | 1 |

The bit rate error in asynchronous mode is calculated using the following formulae.

- When the baud rate generator is in normal mode (SEMR.BGDM = 0):

$$\text{Error}\ (\%) = \left\{ \frac{\text{P0}\phi \times 10^6}{(\text{N}+1) \times \text{B} \times 64 \times 2^{2\text{n}-1}} - 1 \right\} \times 100$$

(When operating on the base clock with a frequency 16 times the bit rate (SMER.ABCS0 = 0))

$$\text{Error}\ (\%) = \left\{ \frac{\text{P0}\phi \times 10^6}{(\text{N}+1) \times \text{B} \times 32 \times 2^{2\text{n}-1}} - 1 \right\} \times 100$$

(When operating on the base clock with a frequency 8 times the bit rate (SMER.ABCS0 = 1))

- When the baud rate generator is in double-speed mode (SEMR.BGDM = 1):

$$\text{Error}\ (\%) = \left\{ \frac{\text{P0}\phi \times 10^6}{(\text{N}+1) \times \text{B} \times 32 \times 2^{2\text{n}-1}} - 1 \right\} \times 100$$

(When operating on the base clock with a frequency 16 times the bit rate (SMER.ABCS0 = 0))

$$\text{Error}\ (\%) = \left\{ \frac{\text{P0}\phi \times 10^6}{(\text{N}+1) \times \text{B} \times 16 \times 2^{2\text{n}-1}} - 1 \right\} \times 100$$

(When operating on the base clock with a frequency 8 times the bit rate (SMER.ABCS0 = 1))

**Table 22.6** list the examples of the BRR register setting in asynchronous mode, and **Table 22.7** list the examples of the BRR register setting in clock synchronous mode.

Table 22.6 Bit Rates and BRR Register Settings in Asynchronous Mode

| | P0ϕ (MHz) | | |
|---|---|---|---|
| | 100 | | |
| Bit Rate (bps) | n | N | Error (%) |
| 150 | — | — | — |
| 300 | 3 | 162 | −0.15 |
| 600 | 2 | 80 | 0.47 |
| 1200 | 2 | 162 | −0.15 |
| 2400 | 2 | 80 | 0.47 |
| 4800 | 1 | 162 | −0.15 |
| 9600 | 1 | 80 | 0.47 |
| 14400 | 0 | 216 | 0.01 |
| 19200 | 0 | 162 | −0.15 |
| 28800 | 0 | 108 | −0.45 |
| 31250 | 0 | 99 | 0.00 |
| 38400 | 0 | 80 | 0.47 |
| 115200 | 0 | 26 | 0.47 |
| 500000 | 0 | 5 | *1 |

**Note:** These values assume bits SEMR.ABCS0 and SEMR.BGDM are both 0.
When either the SEMR.ABCS0 bit or SEMR.BGDM bit is set to 1, the bit rate is doubled.
When bits SEMR.ABCS0 and SEMR.BGDM are both 1, the bit rate is quadrupled.
Configure settings so the range of error is no greater than 1%.

Note 1. Values for the blank cells in the table can be set using the MDDR register.
For details, see **Section 22.2.9, Modulation Duty Register (MDDR)** and the **Table 22.12**.

Table 22.7 Bit Rates and BRR Register Settings in Clock Synchronous Mode

| | Pϕ (MHz) | |
|---|---|---|
| | 100 | |
| Bit Rate (bps) | n | N |
| 250 | — | — |
| 500 | — | — |
| 1000 | — | — |
| 2500 | 3 | 155 |
| 5000 | 3 | 77 |
| 10000 | 3 | 38 |
| 25000 | 1 | 249 |
| 50000 | 1 | 124 |
| 100000 | 0 | 249 |
| 250000 | 0 | 99 |
| 500000 | 0 | 49 |
| 1000000 | 0 | 24 |
| 2500000 | 0 | 9 |
| 5000000 | 0 | 4 |

**Note:** Continuous transmission or reception is not possible.
Set the BRR register so that the range of error can fall within 1% or less.

**Remarks:** —: Setting is prohibited.

**Table 22.8** lists the maximum bit rates for various frequencies in asynchronous mode when the baud rate generator is used. **Table 22.9** lists the maximum bit rates for various frequencies in clock synchronous mode when the baud rate generator is used. **Table 22.10** and **Table 22.11** list the maximum rates for external clock inputs in asynchronous mode and clock synchronous mode, respectively.

Table 22.8 Maximum Bit Rates for Various Frequencies with Baud Rate Generator (in Asynchronous Mode)

| | | Settings | |
|---|---|---|---|
| Pϕ (MHz) | Maximum Bit Rate (bit/s) | n | N |
| 100 | 12500000 | 0 | 0 |

**Note:** These values assume bits SEMR.ABCS0 and SEMR.BGDM are both 1. When either the SEMR.ABCS0 bit or SEMR.BGDM bit is set to 1, the bit rate is 1/2. When bits SEMR.ABCS0 and SEMR.BGDM are both 1, the bit rate is 1/4.

Table 22.9 Maximum Bit Rates for Various Frequencies with Baud Rate Generator (in Clock Synchronous Mode)

| | Discontinuous Transmission/Reception | | | Continuous Transmission/Reception | | |
|---|---|---|---|---|---|---|
| | | Settings | | | Settings | |
| Pϕ (MHz) | Maximum Bit Rate (bit/s) | n | N | Maximum Bit Rate (bit/s) | n | N |
| 100 | 25000000 | 0 | 0 | 12500000 | 0 | 1 |

Table 22.10 Maximum Bit Rates with External Clock Input (in Asynchronous Mode)

| Pϕ (MHz) | External Input Clock (MHz) | Maximum Bit Rate (bit/s) |
|---|---|---|
| 100 | 25.00 | 3125000 |

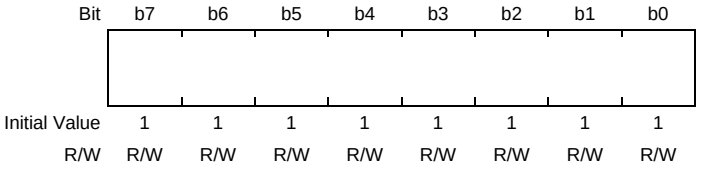
**Note:** This is an example when the SEMR.ABCS0 bit is 1. When the ABCS0 bit is set to 0, the bit rate is 1/2.

Table 22.11 Maximum Bit Rates with External Clock Input (in Clock Synchronous Mode)

| Pϕ (MHz) | External Input Clock (MHz) | Maximum Bit Rate (bit/s) |
|---|---|---|
| 100 | 8.33 | 8330000 |

### 22.2.9 Modulation Duty Register (MDDR)

The MDDR register corrects the bit rate adjusted by the BRR register. The value after reset of this register is H'FF. When the BRME bit in SEMR is set to 1, the bit rate generated by the on-chip baud rate generator is evenly corrected according to the settings of MDDR (MDDR/256). The relationship between the MDDR register setting and the bit rate (B) is given by the following formula. The MDDR register is located in the same address as that of the BRR register and is selected when the MDDRS bit in SEMR is 1. This register is only writable when TE = RE = 0 in the SCR register. b7 in this register is fixed to 1.



| Bit | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| 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 formulae below show the relationships between the MDDR setting and the bit rate (B) when the bit rate modulation function is used.

[Asynchronous mode]

- When the baud rate generator is in normal mode (SEMR.BGDM = 0):

$$B = \frac{P0\phi \times 10^6}{64 \times 2^{2n-1} \times (256/MDDR) \times (N+1)}$$

(When operating on the base clock with a frequency 16 times the bit rate (SEMR.ABCS0 = 0))

$$B = \frac{P0\phi \times 10^6}{32 \times 2^{2n-1} \times (256/MDDR) \times (N+1)}$$

(When operating on the base clock with a frequency 8 times the bit rate (SEMR.ABCS0 = 1))

- When the baud rate generator is in double-speed mode (SEMR.BGDM = 1):

$$B = \frac{P0\phi \times 10^6}{32 \times 2^{2n-1} \times (256/MDDR) \times (N+1)}$$

(When operating on the base clock with a frequency 16 times the bit rate (SEMR.ABCS0 = 0))

$$B = \frac{P0\phi \times 10^6}{64 \times 2^{2n-1} \times (256/MDDR) \times (N+1)}$$

(When operating on the base clock with a frequency 8 times the bit rate (SEMR.ABCS0 = 1))

[Clock synchronous mode]

$$B = \frac{P0\phi \times 10^6}{8 \times 2^{2n-1} \times (256/MDDR) \times (N+1)}$$

When the bit rate modulation is used, the bit rate average error is given by the following formulae.

- When the baud rate generator is in normal mode (SEMR.BGDM = 0):

$$\text{Error}\ (\%) = \{ \frac{P0\phi \times 10^6}{B \times 64 \times 2^{2n-1} \times (256/MDDR) \times (N+1)} - 1 \} \times 100$$

(When operating on the base clock with a frequency 16 times the bit rate (SEMR.ABCS0 = 0))

$$\text{Error}\ (\%) = \{ \frac{P0\phi \times 10^6}{B \times 32 \times 2^{2n-1} \times (256/MDDR) \times (N+1)} - 1 \} \times 100$$

(When operating on the base clock with a frequency 8 times the bit rate (SEMR.ABCS0 = 1))

- When the baud rate generator is in double-speed mode (SEMR.BGDM = 1):

$$\text{Error}\ (\%) = \{ \frac{P0\phi \times 10^6}{B \times 32 \times 2^{2n-1} \times (256/MDDR) \times (N+1)} - 1 \} \times 100$$

(When operating on the base clock with a frequency 16 times the bit rate (SEMR.ABCS0 = 0))

$$\text{Error}\ (\%) = \{ \frac{P0\phi \times 10^6}{B \times 16 \times 2^{2n-1} \times (256/MDDR) \times (N+1)} - 1 \} \times 100$$

(When operating on the base clock with a frequency 8 times the bit rate (SEMR.ABCS0 = 1))

B: Bit rate (bits/s)
N: BRR register setting ($0 \leq N \leq 255$)
The setting must satisfy the electrical characteristics).
P0ϕ: Operating frequency for peripheral modules (MHz)
MDDR: MDDR setting ($128 \leq MDDR \leq 256$)
n: Baud rate generator clock source (n = 0, 1, 2, 3) (For the clock sources and values of n, see **Table 22.5**).

Table 22.12 Bit Rates and BRR and MDDR Registers Settings in Asynchronous Mode

| | Pϕ (MHz) | | | |
|---|---|---|---|---|
| | 100 | | | |
| Bit Rate (bps) | n | N | MDDR | Error (%) |
| 150 | 3 | 252 | 199 | 0.02 |
| 300 | 3 | 161 | 255 | 0.08 |
| 600 | 3 | 80 | 255 | 0.08 |
| 1200 | 2 | 161 | 255 | 0.08 |
| 2400 | 2 | 80 | 255 | 0.08 |
| 4800 | 1 | 161 | 255 | 0.08 |
| 9600 | 1 | 80 | 255 | 0.08 |
| 14400 | — | — | — | — |
| 19200 | 0 | 161 | 255 | 0.08 |
| 28800 | 0 | 107 | 254 | 0.08 |
| 31250 | — | — | — | — |
| 38400 | 0 | 80 | 253 | 0.08 |
| 115200 | 0 | 26 | 255 | 0.08 |
| 500000 | 0 | 5 | 246 | 0.10 |

**Note:** These values assume bits SEMR.ABCS0 and SEMR.BGDM are both 0. When either the SEMR.ABCS0 bit or SEMR.BGDM bit is set to 1, the bit rate is doubled. When bits SEMR.ABCS0 and SEMR.BGDM are both 1, the bit rate is quadrupled. Configure settings so the range of error is no greater than 1%.

## 22.2.10 FIFO Control Register (FCR)

The FCR register resets the quantity of data in the transmit FIFO data register (FTDR) and the receive FIFO data register (FRDR) and specifies the number of triggers. This register also specifies whether to enable the loop-back test.

The CPU can always read and write to the FCR register.

| Bit | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | 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/W | R/W | R/W | R/W | 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 |
|---|---|---|---|---|
| b0 | LOOP | 0 | R/W | Loop-Back Test<br>0: Loop back test is disabled.<br>1: Loop back test is enabled |
| b1 | RFRST | 0 | R/W | Receive FIFO Data Register Reset<br>0: Normal operation<br>1: Resets the FRDR register. |
| b2 | TFRST | 0 | R/W | Transmit FIFO Data Register Reset<br>0: Normal operation<br>1: Resets the FTDR register. |
| b3 | MCE | 0 | R/W | Modem Control Enable<br>0: Model signal is disabled.*1<br>1: Model signal is enabled. |
| b5, b4 | TTRG[1:0] | All 0 | R/W | Transmit FIFO Data Trigger Number Select<br>b5 b4<br>0 0: 8 (8)*2<br>0 1: 4 (12)*2<br>1 0: 2 (14)*2<br>1 1: 0 (16)*2 |
| b7, b6 | RTRG[1:0] | All 0 | R/W | Receive FIFO Data Trigger Number Select<br>In asynchronous mode:<br>b7 b6<br>0 0: 1<br>0 1: 4<br>1 0: 8<br>1 1: 14<br>In clock synchronous mode:<br>b7 b6<br>0 0: 1<br>0 1: 2<br>1 0: 8<br>1 1: 14 |
| b10 to b8 | RSTRG[2:0] | All 0 | R/W | RTS# Output Active Trigger Number Select<br>b10 b8<br>0 0 0: 15<br>0 0 1: 1<br>0 1 0: 4<br>0 1 1: 6<br>1 0 0: 8<br>1 0 1: 10<br>1 1 0: 12<br>1 1 1: 14 |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| b15 to b11 | — | All 0 | R/W | Reserved<br>These bits are read as 0. The write value should be 0. |

Note 1. The CTS# input level does not affect the transmit operation. Similarly, the RTS# input level does not affect the receive operation.

Note 2. Values in parentheses mean the number of empty bytes in the FTDR register when the TDFE flag is set to 1 and a transmit FIFO data empty interrupt (TXI) request is generated.

#### LOOP Bit (Loop-Back Test)

Internally connects between the transmit output pin (TXD) and the receive input pin (RXD) and between the RTS# pin and the CTS# pin, to perform loop-back testing.

#### RFRST Bit (Receive FIFO Data Register Reset)

Disables the receive data in the receive FIFO data register (FRDR) and makes the data to the empty state. If you set this bit to 1, be sure to clear it to 0 afterward.

#### TFRST Bit (Transmit FIFO Data Register Reset)

Disables the transmit data in the transmit FIFO data register (FTDR) and makes the data to the empty state. If you set this bit to 1, be sure to clear it to 0 afterward.

#### MCE Bit (Modem Control Enable)

Specifies whether to enable or disable the modem control signals, CTS# and RTS#. In clock synchronous mode, this bit should always be set to 0.

#### TTRG[1:0] Bits (Transmit FIFO Data Trigger Number Select)

Specify the reference quantity of data for transmission (i.e., the threshold number of entries to trigger the writing of further data for transmission) for setting of the TDFE flag in the serial status register (FSR). When the number of entries for transmission in the transmission FIFO, i.e. the number of entries written to the transmit FIFO data register (FTDR) that are yet to be transmitted, falls to or below the specified trigger number for transmission, the TDFE flag is set to 1 and a transmit FIFO data empty interrupt (TXI) request is generated.

The setting in these bits is valid when the TTRGS bit in the FTCR register is 0. When the TTRGS bit in the FTCR register is 1, the setting of the TFTC[4:0] bits in the FTCR register is valid.

#### RTRG[1:0] Bits (Receive FIFO Data Trigger Number Select)

Specify the reference quantity of receive data (i.e., the threshold number of entries to trigger the reading of received data) for setting of the RDF flag in the serial status register (FSR). When the number of entries in the reception FIFO, i.e. the number of entries yet to be read from the receive FIFO data register (FRDR), rises to or above the specified trigger number for reception, the RDF flag is set to 1 and a receive FIFO data full interrupt (RXI) request is generated.

The setting in these bits is valid when the RTRGS bit in the FTCR register is 0. When the RTRGS bit in the FTCR register is 1, the setting of the RFTC[4:0] bits in the FTCR register is valid.

#### RSTRG[2:0] Bits (RTS# Output Active Trigger Number Select)

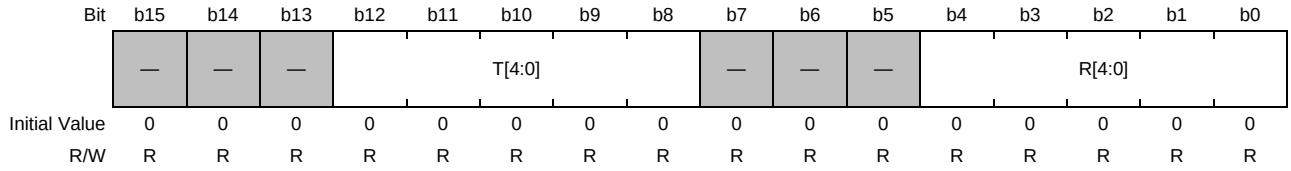
When the number of entries in the reception FIFO, i.e. the number of entries yet to be read from the receive FIFO data register (FRDR), rises to or above the specified trigger number, the RTS# signal is in the high state.

The setting in these bits is only valid when a modem signal is enabled by the MCE bit in this register in asynchronous mode.

## 22.2.11 FIFO Data Count Register (FDR)

The FDR register indicates the quantity of data stored in the transmit FIFO data register (FTDR) and the receive FIFO data register (FRDR).

This register indicates the quantity of transmit data in the FTDR register with the 8 higher-order bits, and the quantity of receive data in the FRDR register with the 8 lower-order bits. The CPU can always read the FDR register.




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| b4 to b0 | R[4:0] | All 0 | R | Receive Data Quantity in FRDR<br>Indicate the quantity of receive data stored in the FRDR register. |
| b7 to b5 | — | All 0 | R | Reserved<br>These bits are read as 0. |
| b12 to b8 | T[4:0] | All 0 | R | Non-Transmitted Data Quantity in FTDR<br>Indicate the quantity of non-transmitted data stored in the FTDR register. |
| b15 to b13 | — | All 0 | R | Reserved<br>These bits are read as 0. |

### R[4:0] Bits

Indicate the quantity of receive data stored in the FRDR register.

H’00 means no received data, and H’10 means that all of the received data is stored in the FRDR register.

### T[4:0] Bits

Indicate the quantity of non-transmitted data stored in the FTDR register.

H’00 means no transmit data, and H’10 means that all of the data for transmission is stored in the FTDR register.

## 22.2.12 Serial Port Register (SPTR)

The SPTR register controls input/output and data of the pins multiplexed to SCIFA function. The CPU can always read and write to the SPTR register.

NOTE

b6, b4, b2, and b0 of this register respectively indicate the input status of their corresponding pins. See the descriptions for each bit for details. Writings to these bits in 1-bit unit are handled as read-modify-write, which may lead to undesired values to be written. To avoid this, when modifying the SPB2DT or SPB2IO bit, for example, write the other bit (the bit used in combination) at the same time.

| Bit | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | RTS2IO | RTS2DT | CTS2IO | CTS2DT | SCKIO | SCKDT | SPB2IO | SPB2DT |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | x | 0 | x | 0 | x | 0 | x |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

**Remarks:** x: Undefined

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| b0 | SPB2DT | x | R/W | Serial Port Break Data Select<br>Controls the TXD pin in combination with the TE bit in the SCR register and the SPB2IO bit. See **Table 22.15**. |
| b1 | SPB2IO | 0 | R/W | Serial Port Break Input/Output<br>Controls the TXD pin in combination with the TE bit in the SCR register and the SPB2DT bit. |
| b2 | SCKDT | x | R/W | SCK Port Data Select<br>Controls the SCK pin in combination with the CM bit in the SMR register, the SCKIO bit, and the CKE1 and CKE0 bits in the SCR register. See **Table 22.17**. |
| b3 | SCKIO | 0 | R/W | SCK Port Input/Output<br>Controls the SCK pin in combination with the CM bit in the SMR register, the SCKDT bit, and the CKE1 and CKE0 bits in the SCR register. See **Table 22.17**. |
| b4 | CTS2DT | x | R/W | CTS# Port Data Select<br>Controls the CTS# pin in combination with MCE bit in FCR and CTS2IO bit. See **Table 22.14**.<br>The SCIF 3 and SCIF 4 channels are not supported. |
| b5 | CTS2IO | 0 | R/W | CTS# Port Output Specify<br>Controls the CTS# pin in combination with MCE bit in FCR and CTS2IO bit. See **Table 22.14**.<br>The SCIF 3 and SCIF 4 channels are not supported. |
| b6 | RTS2DT | x | R/W | RTS# Port Data Select<br>Controls the RTS# pin in combination with MCE bit in FCR and RTS2IO bit. See **Table 22.13**.<br>The SCIF 3 and SCIF 4 channels are not supported. |
| b7 | RTS2IO | 0 | R/W | RTS# Port Output Specify<br>Controls the RTS# pin in combination with MCE bit in FCR and RTS2IO bit. See **Table 22.13**.<br>The SCIF 3 and SCIF 4 channels are not supported. |
| b15 to b8 | — | All 0 | R/W | Reserved<br>These bits are read as 0. The write value should be 0. |

**Remarks:** x: Undefined

### SPB2DT Bit (Serial Port Break Data Select)

This bit specifies the output level of the TXD pin when the setting of the SCR.TE bit is 0. The RXD pin input status can be read from this bit regardless of the SPB2IO bit setting. However, the RXD pin function must have been selected with the general I/O port.

### SPB2IO Bit (Serial Port Break Input/Output)

Controls the TXD pin in combination with the TE bit in the SCR register and the SPB2DT bit.

### SCKDT Bit (SCK Port Data Select)

The SCK pin status can be read from this bit regardless of the SCKIO bit setting. (When the SCK pin is used for input, the input signal is invalid (has no means) but the pin status can be read.) However, the SCK pin function must have been selected with the general I/O port.

### SCKIO Bit (SCK Port Input/Output)

Specifies input or output status of the SCK pin. This bit controls the SCK pin in combination with the SCKDT bit, the CM bit in the SMR register, and the CKE1 and CKE0 bits in the SCR register.

### CTS2DT Bit (CTS# Port Data Select)

The status of the CTS# pin can be read from this bit regardless of the CTS2IO bit setting. However, the CTS# pin function must have been selected with the general I/O port.

### RTS2DT Bit (RTS# Port Data Select)

The status of the RTS# pin can be read from this bit regardless of the RTS2IO bit setting. However, the RTS# pin function must have been selected with the general I/O port

Table 22.13 RTS# Pin Status

| FCR.MCE Bit Setting | RTS2IO Bit Setting | RTS2DT Bit Setting | RTS# Pin Status |
|---|---|---|---|
| 0 | 0 | × | Setting prohibited*1 |
| 0 | 1 | 0 | Low output |
| 0 | 1 | 1 | High output |
| 1 | × | × | Modem control output |

**Remarks:** ×: Don't care

Note 1. There is no problem with the initial setting if the RTS# pin is not used.

Table 22.14 CTS# Pin Status

| FCR.MCE Bit Setting | CTS2IO Bit Setting | CTS2DT Bit Setting | CTS# Pin Status |
|---|---|---|---|
| 0 | 0 | × | Setting prohibited*1 |
| 0 | 1 | 0 | Low output |
| 0 | 1 | 1 | High output |
| 1 | × | × | Modem control input |

**Remarks:** ×: Don't care

Note 1. There is no problem with the initial setting if the CTS# pin is not used.

Table 22.15 TXD Pin Status

| SCR.TE Bit Setting | SPB2IO Bit Setting | SPB2DT Bit Setting | TXD Pin Status |
|---|---|---|---|
| 0 | 0 | × | Setting prohibited |
| 0 | 1 | 0 | Low output |
| 0 | 1 | 1 | High output |
| 1 | × | × | Transmit data output |

**Remarks:** ×: Don't care

## 22.2.13 Line Status Register (LSR)

The LSR register is a 16-bit register. The PER and FER bits indicate the number of receive errors in the receive FIFO data register. 1 cannot be written to the OREER status flag. The flag should be read as 1 prior to clearing it to 0.

| Bit | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | PER[3:0] | | | | — | — | FER[3:0] | | | | — | 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)*1 |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| b0 | ORER | 0 | R/(W) *1 | Overrun Error Flag<br>0: Reception is in progress or has normally completed.<br>1: An overrun error has occurred during reception. |
| b1 | — | 0 | R | Reserved<br>This bit is read as 0. |
| b5 to b2 | FER[3:0] | All 0 | R | Framing Error Count<br>Indicates the quantity of data with a framing error among the receive data stored in the receive FIFO data register (FRDR). |
| b7, b6 | — | All 0 | R | Reserved<br>These bits are read as 0. |
| b11 to b8 | PER[3:0] | All 0 | R | Parity Error Count<br>Indicates the quantity of data with a parity error among the receive data stored in the receive FIFO data register (FRDR). |
| b15 to b12 | — | All 0 | R | Reserved<br>These bits are read as 0. |

Note 1. To clear the flag, 0 can be only written after 1 is read.

### ORER Bit (Overrun Error Flag)

Indicates that receive operation abnormally stops due to occurrence of an overrun error. This flag is not affected and retains its previous state if the RE bit in the serial control register (SCR) is cleared to 0. The receive FIFO data register (FRDR) retains the data before an overrun error occurred, and newly received data is lost. When the ORER bit is set to 1, the SCIFA cannot continue subsequent serial reception.

[Setting condition]

- When the next serial reception is completed with the receive FIFO in full state (16-byte data is received)

[Clearing condition]

- When 0 is written to ORER after being read as 1.

*Note*: When the internal clock is selected while the SCIFA is in clock synchronous mode, the amount of receive data can be controlled, so no overrun occurs.

### FER[3:0] Bits (Framing Error Count)

The values of bits 5 to 2 indicate the quantity of data with a framing error after the ER bit in the FSR register is set. Reading 0000 from the FER[3:0] bits means all 16-byte receive data in the FRDR register have a framing error.

### PER[3:0] Bits (Parity Error Count)

The values of bits 11 to 8 indicate the quantity of data with a parity error after the ER bit in the FSR register is set. Reading 0000 from the PER[3:0] bits means all 16-byte receive data in the FRDR register have a parity error.

## 22.2.14 Serial Extended Mode Register (SEMR)

The SEMR register specifies either LSB or MSB first, enables the noise cancellation, operation in normal or double-speed mode of the baud rate generator, and bit rate modulation, and selects the modulation register and the sampling count (either 8 or 16 times).

| Bit | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | BGDM | — | BRME | MDDRS | DIR | NFEN | — | ABCS0 |
| 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 |
|---|---|---|---|---|
| b0 | ABCS0 | 0 | R/W | Asynchronous Base Clock Select<br>0: Operates on a frequency 16 times the transfer rate as the base clock.<br>1: Operates on a frequency 8 times the transfer rate as the base clock. |
| b1 | — | 0 | R/W | Reserved<br>This bit is read as 0. The write value should be 0. |
| b2 | NFEN | 0 | R/W | Noise Cancellation Enable<br>0: Noise cancellation for the RxD pin is disabled.<br>1: Noise cancellation for the RxD pin is enabled. |
| b3 | DIR | 0 | R/W | Data Transfer Direction Select<br>0: Transmits the data in the FTDR register by the LSB-first method.<br>The received data is stored in the FRDR register by the LSB- first method.<br>1: Transmits the data in the FTDR register by the MSB-first method.<br>The received data is stored in the FRDR register by the MSB- first method. |
| b4 | MDDRS | 0 | R/W | Modulation Duty Register Select<br>0: BRR register is accessible.<br>1: MDDR register is accessible. |
| b5 | BRME | 0 | R/W | Bit Rate Modulation Enable<br>0: Bit rate modulation is disabled.<br>1: Bit rate modulation is enabled. |
| b6 | — | 0 | R/W | Reserved<br>This bit is read as 0. The write value should be 0. |
| b7 | BGDM | 0 | R/W | Baud Rate Generator Double- Speed Mode Select<br>0: Baud rate generator normal mode:<br>Baud rate generator operates on the clock signal produced by dividing the clock source by two.<br>1: Baud rate generator double-speed mode:<br>Baud rate generator operates on the clock signal produced by the clock source (no frequency division). |

### ABCS0 Bit (Asynchronous Base Clock Select)

Selects the base clock for 1-bit period in asynchronous mode.

This bit setting is valid only in asynchronous mode (i.e., when the CM bit in the SMR register is 0).

### NFEN Bit (Noise Cancellation Enable)

Reduces noise of the input to the RxD pin. This function is only valid in asynchronous mode. For details, see **Section 22.7, Noise Cancellation**.

In clock synchronous mode, this bit should always be set to 0.

### DIR Bit (Data Transfer Direction Select)

Selects the serial communication format. This bit is valid only when the transmit/receive data is in 8-bit formats.*1

**Note 1.** Asynchronous mode or clock synchronous mode with the 8-bit data length

### MDDRS Bit (Modulation Duty Register Select)

Selects the register to be enabled access to it.

### BRME Bit (Bit Rate Modulation Enable)

Specifies whether to enable or disable the bit rate modulation.

### BGDM Bit (Baud Rate Generator Double-Speed Mode Select)

Selects operating mode of the baud rate generator. When setting 1 in this bit, the baud rate generator included in the SCIFA operates in double-speed mode. The setting of this bit is only effective in asynchronous mode (SMR.CM bit = 0) when the internal clock is selected as the clock source (SCR.CKE[1:0] = 00b). Use normal mode under any other settings.

## 22.2.15 FIFO Trigger Control Register (FTCR)

The FTCR register is a 16-bit register that specifies FIFO trigger conditions. The CPU can always read from and write to the FTCR register.

| Bit | b15 | b14 | b13 | b12 | b11 | b10 | b9 | b8 | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RTRGS | — | — | RFTC[4:0] | | | | | TTRGS | — | — | TFTC[4:0] | | | | |
| Initial Value | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 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 | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| b4 to b0 | TFTC[4:0] | All 1 | R/W | Transmit FIFO Data Trigger Number<br>H'00: Transmit data trigger number is 0.<br>H'0F: Transmit data trigger number is 15.<br>Do not set H'10 to H'1F in these bits. |
| b6, b5 | — | All 0 | R/W | Reserved<br>These bits are read as 0. The write value should be 0. |
| b7 | TTRGS | 0 | R/W | Transmit Trigger Select<br>0: TTRG[1:0] bits in FCR are valid.<br>1: TFTC[4:0] bits in FTCR are valid. |
| b12 to b8 | RFTC[4:0] | All 1 | R/W | Receive FIFO Data Trigger Number<br>H'01: Receive data trigger number is 1.<br>H'10: Receive data trigger number is 16.<br>Do not set H'00 and H'11 to H'1F in these bits. |
| b14, b13 | — | All 0 | R/W | Reserved<br>These bits are read as 0. The write value should be 0. |
| b15 | RTRGS | 0 | R/W | Receive Trigger Select<br>0: RTRG[1:0] bits in FCR are valid.<br>1: RFTC[4:0] bits in FTCR are valid. |

### TFTC[4:0] Bits (Transmit FIFO Data Trigger Number)

Specify the reference quantity of data for transmission (i.e., the threshold number of entries to trigger the writing of further data for transmission) for setting of the TDFE flag in the serial status register (FSR).

When the number of entries for transmission in the transmission FIFO, i.e. the number of entries written to the transmit FIFO data register (FTDR) that are yet to be transmitted, falls to or below the specified trigger number for transmission, the TDFE flag is set to 1 and a transmit FIFO data empty interrupt (TXI) request is generated.

### RFTC[4:0] Bits (Receive FIFO Data Trigger Number)

Specify the reference quantity of receive data (i.e., the threshold number of entries to trigger the reading of received data) for setting of the RDF flag in the serial status register (FSR).

When the number of entries in the reception FIFO, i.e. the number of entries yet to be read from the receive FIFO data register (FRDR), rises to or above the specified trigger number for reception, the RDF flag is set to 1 and a receive FIFO data full interrupt (RXI) request is generated.

## 22.3 Operation

### 22.3.1 Overview

For serial communication, the SCIFA can select either asynchronous mode in which characters are synchronized individually or a clock synchronous mode in which communication is synchronized with clock pulses.

The SCIFA has a 16-stage FIFO buffer for both transmission and receptions, reducing the overhead of the CPU and enabling continuous high-speed communication. The RTS# and CTS# signals are provided as modem control signals. Selection of a transmission/reception format is enabled with the serial mode register (SMR). **Table 22.16** shows the transmission format which can be selected in the serial mode register (SMR). As shown in **Table 22.17**, the SCIFA clock source can be selected by the combination of the CKE1 and CKE0 bits in the serial control register (SCR).

#### (1) Asynchronous Mode

- Data length is selectable either 7 or 8 bits
- Parity addition and 1- or 2-bit stop bit addition are selectable.
  (The combination of the preceding selections determines the transmission/reception format and character length).
- In reception, it is possible to detect framing errors, parity errors, receive FIFO data full, overrun errors, receive data ready, and breaks.
- The stored data quantities are indicated in the FIFO data count register (FDR), respectively for transmit and receive FIFO data.
- An internal or external clock can be selected as the SCIFA clock source.
  When an internal clock is selected, the SCIFA operates using the clock of on-chip baud rate generator and can output the clock with a frequency 16 (or 8) times the bit rate.
  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

- The transmission/reception format is fixed to the 8-bit data length.
- In reception, it is possible to detect overrun errors (ORER).
- An internal or external clock can be selected as the SCIFA clock source.
  When an internal clock is selected, the SCIFA 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 SCIFA operates on the input synchronous clock not using the on-chip baud rate generator.

Table 22.16 SMR Register Settings and SCIFA Communication Formats

| SMR Register | | | | | SCIFA Transmission/Reception Format | | |
|---|---|---|---|---|---|---|---|
| b7 | b6 | b5 | b3 | | | | |
| CM | CHR | PE | STOP | Mode | Data Length | Parity Bit | Stop Bit Length |
| 0 | 0 | 0 | 0 | Asynchronous mode | 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 | × | × | × | Clock synchronous mode | 8 bits | Not set | None |

**Remarks:** ×: Don't care

Table 22.17 SMR, SCR, and SPTR Register Settings and SCIFA Clock Source Selection

| SMR Register | SCR Register | | SPTR Register | | | | |
|---|---|---|---|---|---|---|---|
| b7 | b1 | b0 | b3 | b2 | | | |
| CM | CKE1 | CKE0 | SCKIO | SCKDT | Mode | Clock Source | SCK Pin Function |
| 0 | 0 | 0 | 0 | × | Asynchronous mode | Internal | Input pin (input signal invalid) (Initial state) |
| | | | 1 | 0 | | | SCK pin state: Low |
| | | | 1 | 1 | | | SCK pin state: High |
| | | 1 | × | × | | | Outputs a clock with frequency 16/8 times the bit rate*1 |
| | 1 | 0 | × | × | | External | Inputs a clock with frequency 16/8 times the bit rate*2 |
| | | 1 | × | × | | Setting prohibited | |
| 1 | 0 | × | × | × | Clock synchronous mode | Internal | Outputs the synchronous clock |
| | 1 | 0 | × | × | | External | Inputs the synchronous clock |
| | | 1 | × | × | | Setting prohibited | |

**Remarks:** ×: Don't care

Note 1. SEMR.ABCS0 = 0: Output a clock that has a frequency 16 times the bit rate. SEMR.ABCS0 = 1: Output a clock that has a frequency 8 times the bit rate.

Note 2. SEMR.ABCS0 = 0: Input a clock that has a frequency 16 times the bit rate. SEMR.ABCS0 = 1: Input a clock that has a frequency 8 times the bit rate.

### 22.3.2 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 SCIFA are independent, so full duplex communication is possible. The transmitter and receiver are 16-stage FIFO buffered, so data can be written and read while transmitting and receiving are in progress, enabling continuous transmission and reception.

**Figure 22.2** shows the general format of asynchronous serial communication.

In asynchronous serial communication, the communication line is normally held in the mark (high) state. The SCIFA monitors the line and starts serial communication when the line goes to the space (low) state, considered as a start bit. One serial character consists of a start bit (low), data (LSB first when LSB-first transfer is selected), parity bit (high or low), and stop bit (high), in that order.

When receiving in asynchronous mode, the SCIFA synchronizes at the falling edge of the start bit. The SCIFA samples each data bit on the eighth pulse of a clock with a frequency 16 or 8 times the bit rate*[1]. Receive data is latched at the center of each bit.

**Note 1.** When the SEMR.ABCS0 bit = 0, data is sampled on the eighth pulse of a clock with a frequency 16 times the bit rate.
When the SEMR.ABCS0 bit = 1, data is sampled on the fourth pulse of a clock with a frequency 8 times the bit rate.

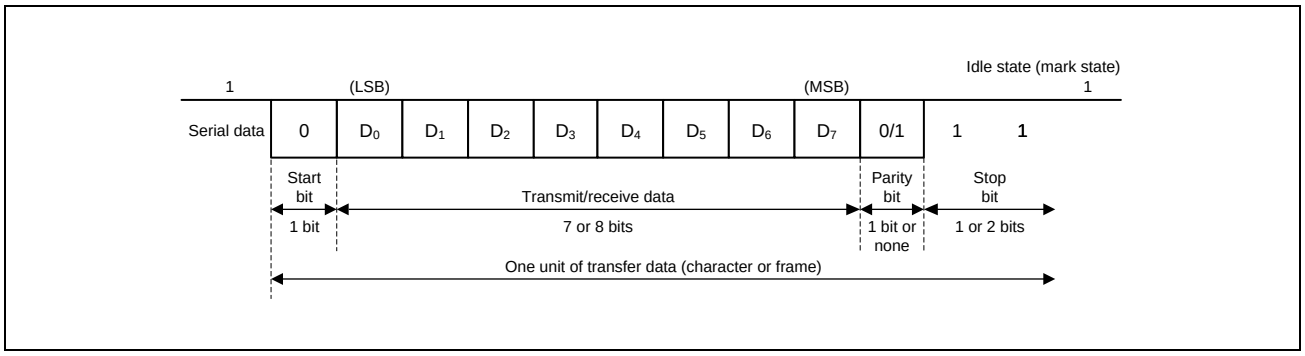



Figure 22.2 Data Format in Asynchronous Communication
(8-Bit Data with Parity and Two Stop Bits when LSB-First Transfer is Selected)

### (1) Transmit/Receive Formats

**Table 22.18** lists the eight communications formats that can be selected in asynchronous mode. The format is selected by setting in the serial mode register (SMR).

Table 22.18 Serial Communications Formats (in Asynchronous Mode)

| SMR Setting | | | 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 | | |
| | | 1 | START | 8-bit data | | | | | | | | STOP | STOP | |
| | 1 | 0 | START | 8-bit data | | | | | | | | P | STOP | |
| | | 1 | START | 8-bit data | | | | | | | | P | STOP | STOP |
| 1 | 0 | 0 | START | 7-bit data | | | | | | | STOP | | | |
| | | 1 | START | 7-bit data | | | | | | | STOP | STOP | | |
| | 1 | 0 | START | 7-bit data | | | | | | | P | STOP | | |
| | | 1 | START | 7-bit data | | | | | | | P | STOP | STOP | |

**Note:** START: Start bit
STOP: Stop bit
P: Parity bit

### (2) Clock

An SCIFA transmit/receive clock can be selected from two types of clock sources: the internal clock generated by the on- chip baud rate generator, the external clock input from the SCK pin. The clock source is selected by the settings of the CM bit in the serial mode register (SMR), the CKE[1:0] bits in the serial control register (SCR), and the ACS0 bit in the serial extended mode register (SEMR). For clock source selection, refer to **Table 22.17**.

When an external clock is input at the SCK pin, it must have a frequency equal to 16/8 times the desired bit rate.

When the SCIFA operates on an internal clock, it can output a clock signal on the SCK pin. The frequency of this output clock is 16/8 times the desired bit rate.

### (3) Transmitting and Receiving Data

#### SCIFA Initialization (in Asynchronous Mode)

Before transmitting or receiving data, clear the TE and RE bits to 0 in the serial control register (SCR), and then initialize the SCIFA as follows.

When changing operating mode or communication format, always clear the TE and RE bits in the SCR register to 0 before following the procedure given below. Clearing TE to 0 initializes the transmit shift register (TSR). Clearing TE and RE to 0, however, does not initialize the serial status register (FSR), transmit FIFO data register (FTDR), or receive FIFO data register (FRDR), which retain their previous contents. Clear TE to 0 after all transmit data has been transmitted and the TEND flag in the FSR register is set. The TE bit can be cleared to 0 during transmission, but the transmit data (the TXD pin output level) after the TE bit is cleared to 0 depends on the settings of the SPB2IO and SPB2DT bits in the SPTR register. Set the TFRST bit in the FCR register to 1 and reset the FTDR register before TE is set to 1 again to start transmission.

When an external clock is used, the clock should not be stopped during initialization or subsequent operation. SCIFA operation becomes unreliable if the clock is stopped. **Figure 22.3** shows a sample flowchart for initializing the SCIFA in asynchronous mode.

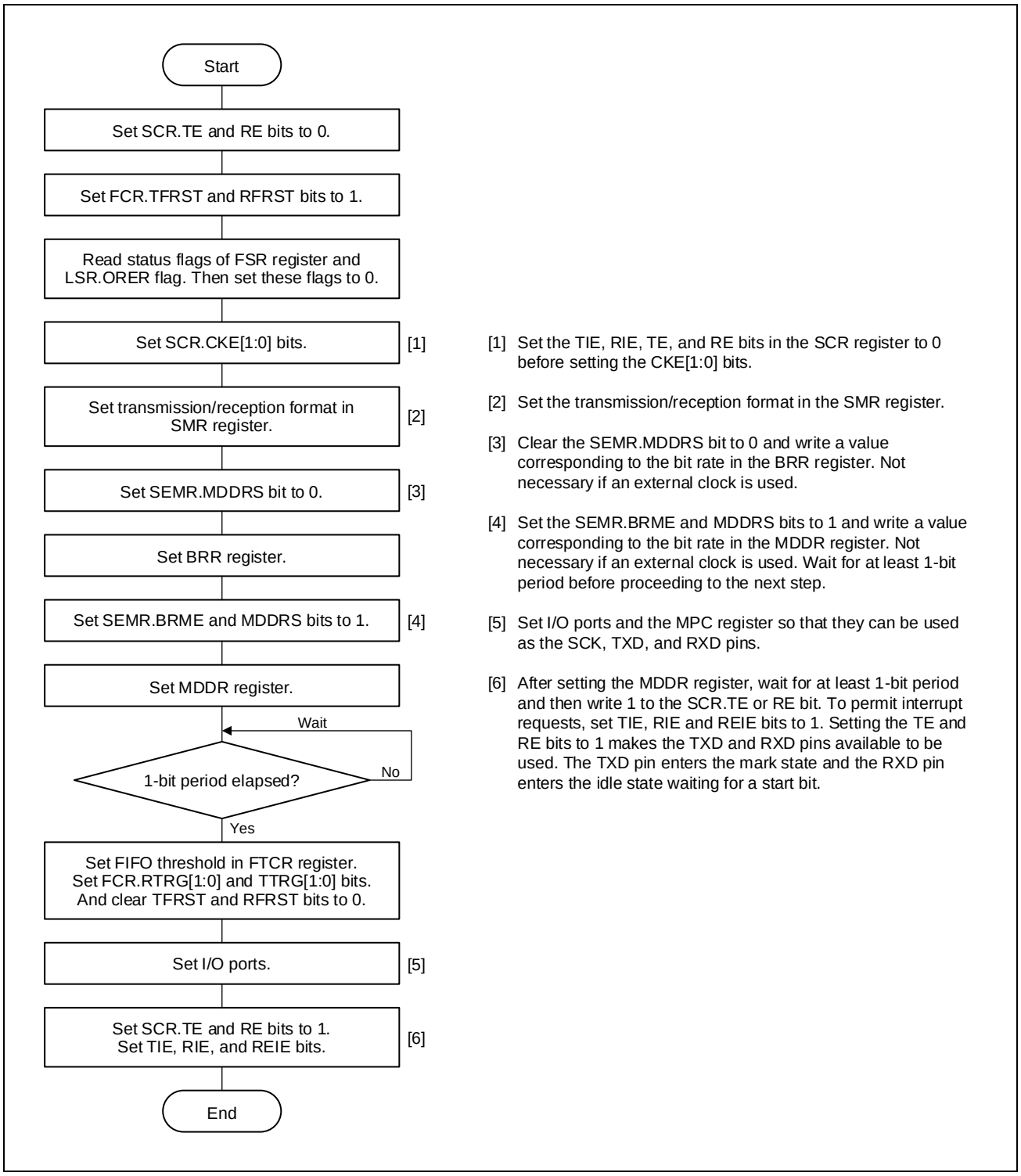
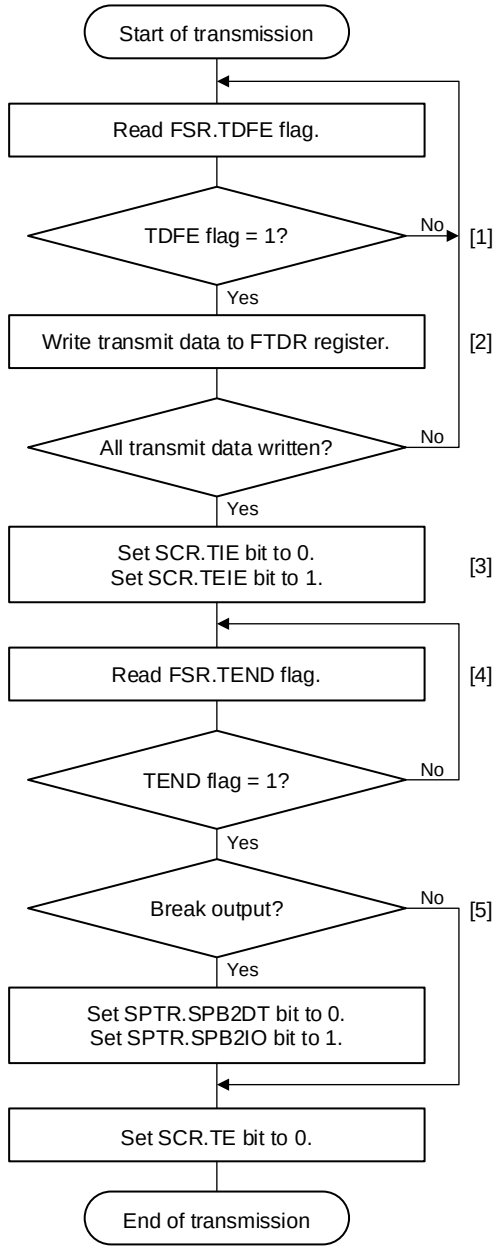



Figure 22.3 Sample Flowchart for SCIFA Initialization in Asynchronous Mode

### Transmitting Serial Data (in Asynchronous Mode)

**Figure 22.4** shows a sample flowchart for serial transmission in asynchronous mode.

Follow the procedure given below for serial data transmission after enabling the SCIFA for transmission.




[1] Check the SCIFA status and write the transmit data:
Read the FSR register, confirm that the TDFE flag is set to 1, and then write the transmit data to the FTDR register. The transmit data quantity that can be written is “16 - FDR.T[4:0]”. A change of the TDFE flag from 0 to 1 can be known by a TXIF interrupt request.

[2] Serial transmission continuation procedure:
When transmit data is written to the FTDR register, the TDFE flag is automatically cleared to 0.

[3] Serial transmission end procedure (when TXIF or TEIF interrupt is used):
After the last transmission data is written to the FTDR register, set the TIE bit to 0 and the TEIE bit to 1 in the SCR register. If the DMAC is activated to write the transmit data to the FTDR register, enable the transmit end interrupt. This will generate a CPU interrupt request when the last transmit data is written in the FTDR register. The TIE and TEIE bit settings will be changed by the CPU interrupt process.
This procedure is unnecessary if the TXIF or TEIF interrupt is not used.

[4] Serial transmission end confirmation:
Read the FSR register and confirm that the TEND flag has changed to 1.
A change of the TEND flag from 0 to 1 can be known by a TEIF interrupt request.

[5] Break output during serial transmission:
To cause a break to be output during serial transmission, set the SPB2DT bit to 0 and the SPB2IO bit to 1 in the SPTR register. Then set the SCR.TE bit to 0.*1

In Step [1], the number of data bytes that can be written can be checked by the FDR.T[4:0] flag (non-transmitted data quantity in the FTDR register).

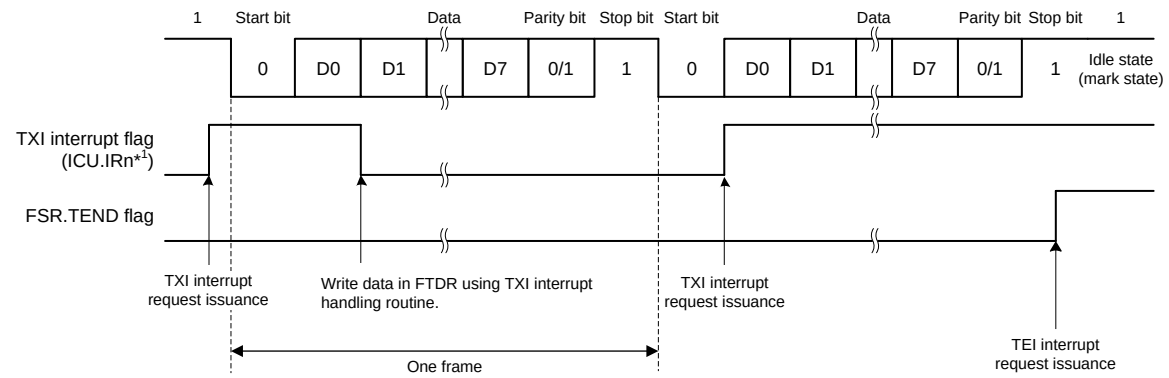
Note 1. The SPB2DT bit of the SPTR register indicates the input status of the RXD pin. Writings to the SPB2DT and SPB2IO bits of this register in 1-bit unit are handled as read-modify-write, which may lead to undesired values to be written to the SPB2DT bit. To avoid this, when modifying the SPB2DT or SPB2IO bit, for example, write the other bit (the bit used in combination) at the same time.

Figure 22.4 Sample Flowchart for Transmitting Serial Data in Asynchronous Mode

In asynchronous mode, the SCIFA performs serial transmission as described below.

1. When data is written into the transmit FIFO data register (FTDR) by the TXI interrupt processing routine, the SCIFA transfers the data from the FTDR register to the transmit shift register (TSR) and starts transmission. Confirm that the TDFE flag in the serial status register (FSR) is set to 1 before writing transmit data to the FTDR register. The number of data bytes that can be written is “16 minus the number of non-transmitted data units”.
2. When data is transferred from the FTDR register to the TSR register and transmission is started, consecutive transmit operations are performed until there is no transmit data left in the FTDR register. When the number of transmit data bytes in the FTDR register becomes equal to or less than the transmission trigger number specified in the FIFO control register (FCR) or FIFO trigger control register (FTCR), the TDFE flag is set. If the TIE bit in the serial control register (SR) is set to 1 at this time, a transmit-FIFO-data-empty interrupt (TXI) request is generated. The serial transmit data is output from the TXD pin in the following order.
   a) Start bit: One-bit 0 is output.
   b) Transmit data: 8- or 7-bit data is output in LSB-first order (when LSB-first transfer is selected).
   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 output.
3. The SCIFA checks the transmit data of the FTDR register at the timing for sending the stop bit. If data is present, the data is transferred from the FTDR register to the TSR register, the stop bit is output, and then serial transmission of the next frame is started. If there is no data to be transmitted, the TEND flag in the FSR register is set to 1, the stop bit is output, and then the SCIFA enters the mark state (high level) in which 1 is output continuously.

**Figure 22.5** shows an example of the operation for transmission in asynchronous mode.




Note 1. For the corresponding interrupt number, see **Section 8, Interrupt Controller**.

Figure 22.5 Example of Transmit Operation in Asynchronous Mode
(8-Bit Data with Parity and One Stop Bit when LSB-First Transfer is Selected)

4. When modem control is enabled, transmission can be stopped/resumed by the input level to the CTS# pin. When a high level is input to the CTS# pin during transmission, the SCIFA enters the mark state (high level) after completion of one-frame data transmission. When a low level is input to the CTS# pin, output of the next data to be transmitted begins with a start bit. **Figure 22.6** shows an example of the operation for transmission when using the modem control function.

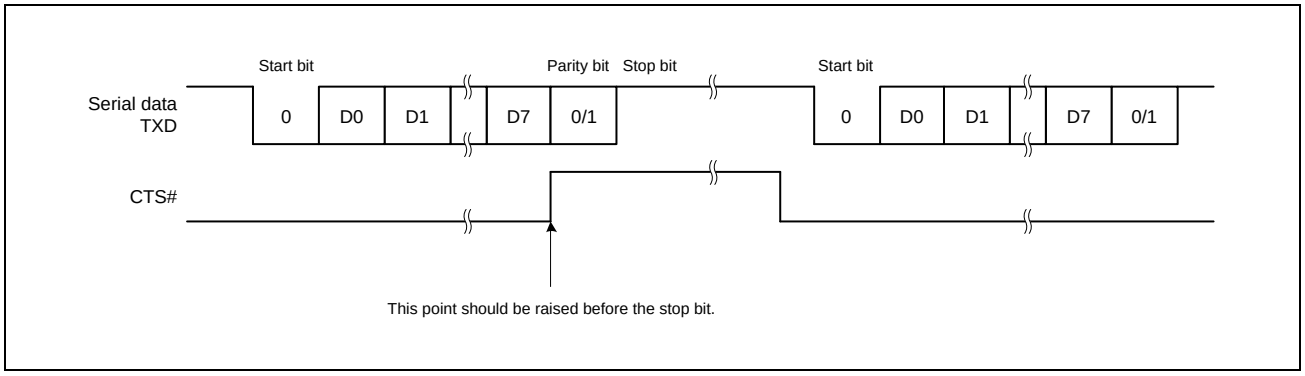
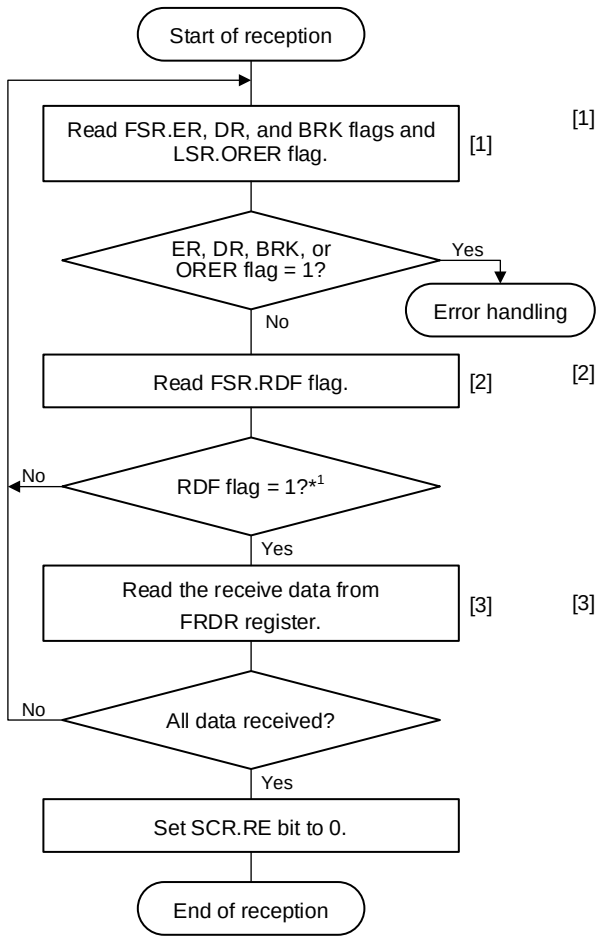



Figure 22.6 Example of Transmit Operation in Asynchronous Mode Using Modem Control Function (CTS#)

Receiving Serial Data (in Asynchronous Mode)

**Figure 22.7** and **Figure 22.8** show sample flowcharts for serial reception in asynchronous mode. Follow the procedure given below for serial data reception after enabling the SCIFA for reception.



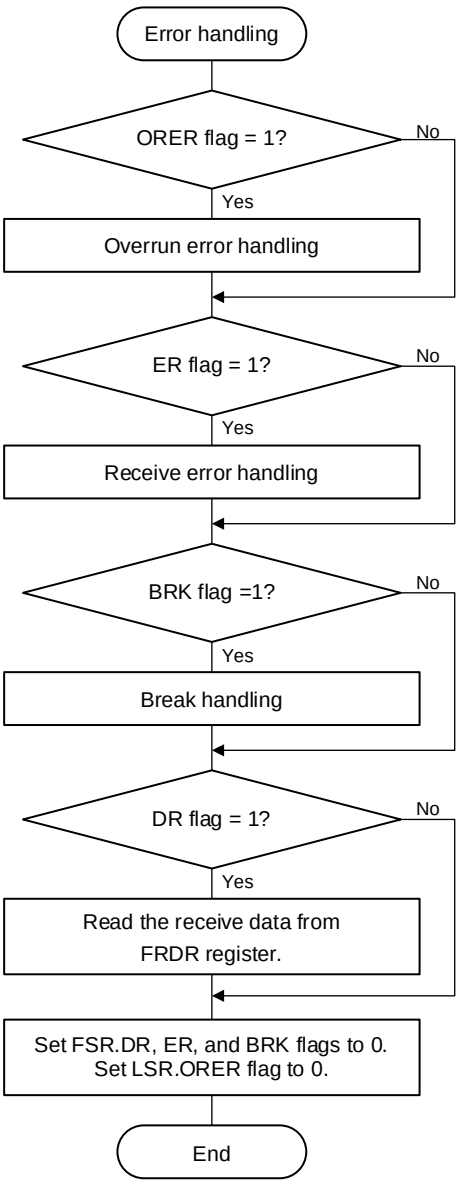

[1] Receive error handling and break detection:
When a receive error occurs, read the ER, DR, and BRK flags in the FSR register and the LSR.ORER flag to identify the error. After handling the error as specified, clear the ER, DR, BRK, and ORER flags to 0.
In the case where a framing error occurs, a break can be detected by reading the value of the RXD pin.
Error occurrence can be known by a BRIF, ERIF, or DRIF interrupt request.

[2] Check the SCIFA status and read the receive data:
Read the FSR register, confirm that the RDF flag is set to 1, and then read the receive data at least as much as the amount specified by the receive FIFO threshold from those stored in the FEDR register. The receive data quantity that can be read can be checked by the FDR.R[4:0] flag.
A change of the RDF flag from 0 to 1 can be known by an RXIF interrupt request.

[3] Serial reception continuation procedure:
When receive data is read from the FRDR register, the RDF flag is automatically cleared to 0.

Note 1. The number of all the received data units should be an integral multiple of the receive FIFO threshold. If not, process returns to Step [1] and enters an infinite loop. In this case, it is necessary to set the timeout processing using a timer.

Figure 22.7 Sample Flowchart for Receiving Serial Data in Asynchronous Mode (1)




- Whether a framing error or parity error has occurred in the received data that is read from the FRDR register can be checked with the FER and PER flags in the FSR register.
- When a break signal is received, the received data is not transferred to the FRDR register while the BRK flag is set to 1. However, note that the last data in the FRDR register is H'00 and the break data in which a framing error occurred is stored.

Figure 22.8 Sample Flowchart for Receiving Serial Data in Asynchronous Mode (2)

In asynchronous mode, the SCIFA performs serial reception as described below.

1. The SCIFA monitors the communication line, and if a 0 start bit is detected, it performs internal synchronization to start reception.
2. The received data is stored into the RSR register in LSB-to-MSB order (when LSB-first transfer is selected).
3. The parity bit and stop bit are received.
   After receiving these bits, the SCIFA carries out the following checks.
   a) Stop bit check: The SCIFA checks whether the stop bit is 1. If there are two stop bits, only the first is checked.
   b) The SCIFA checks whether receive data can be transferred from the receive shift register (RSR) to the receive FIFO data register (FRDR).
   c) Parity bit check: The SCIFA checks whether the parity bit is an expected value.
   d) Overrun error check: The SCIFA checks whether the ORER flag is 0, indicating that the overrun error has not occurred.
   e) Break check: The SCIFA checks whether 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 the FRDR register.

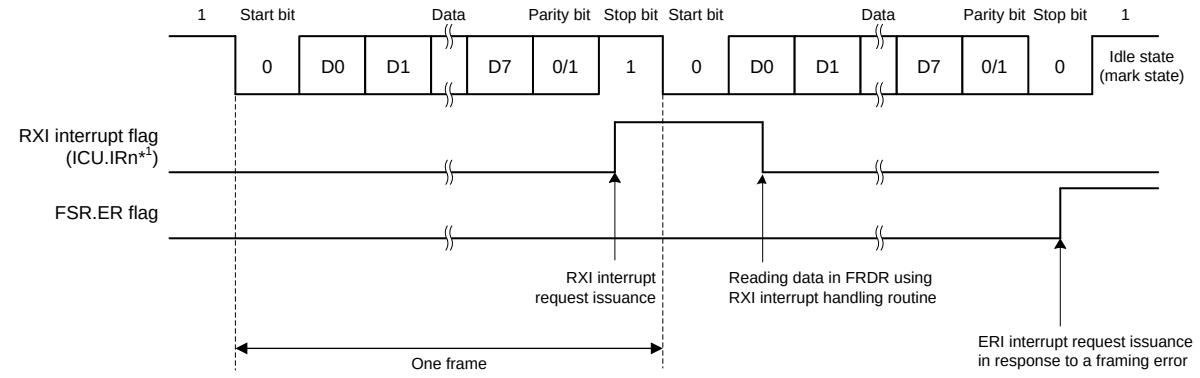
NOTE

When a parity error or a framing error occurs, reception is not suspended.

4. When receive data units equaling or exceeding the specified reception trigger number are stored in the receive FIFO data register (FRDR) and the RDF flag is changed to 1, a receive FIFO data full interrupt (RXI) request is generated while the RIE bit in the SCR register is set to 1. When the quantity of data in the FRDR register falls below the specified reception trigger number and the RIE bit in the SCR register is set to 1, a receive data ready interrupt (DRI) request is generated if no next data is received after the elapse of 15 ETUs*[1] from the last stop bit (the DR flag in the FSR register is 1). When the ER flag in the FSR register is changed to 1, a receive error interrupt (ERI) request is generated while the RIE or REIE bit in the SCR register is set to 1. When the BRK or ORER flag is changed to 1 in the FSR register, a break reception interrupt (BRI) request is generated while the RIE or REIE bit in the SCR register is set to 1.

**Note 1.** It is equivalent to 1 and half frames of 8-bit format with one stop bit (ETU: Element Time Unit).

**Figure 22.9** shows an example of the operation for reception in asynchronous mode.




Note 1. For the corresponding interrupt number, see **Section 8, Interrupt Controller**.

Figure 22.9 Example of SCIFA Receive Operation in Asynchronous Mode
(8-Bit Data with Parity and One Stop Bit when LSB-First Transfer is Selected)

5. When modem control is enabled, the RTS# signal that indicates the FRDR register has space is output. When the RTS# pin is at low level, reception is possible. The RTS# pin being at the high level indicates that the number of entries in the FRDR register is equal to or greater than the threshold for output of the active level of the RTS# signal and that the transmission of further data needs to be suspended until the FRDR register has enough space. **Figure 22.10** shows an example of the operation for reception in asynchronous mode when using the modem control function.

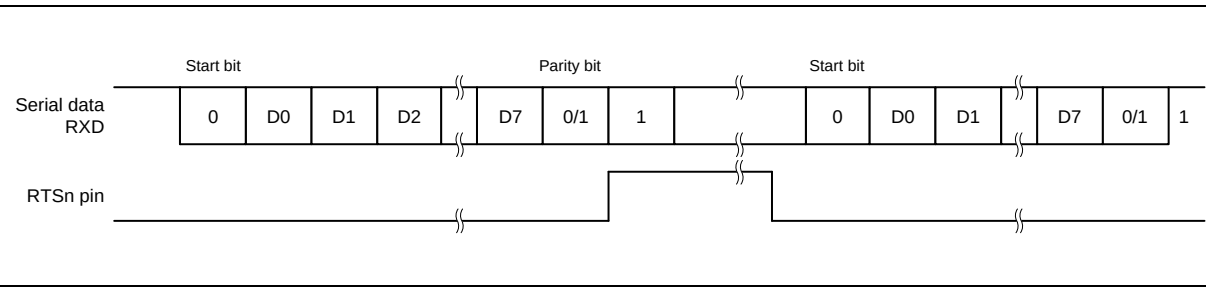


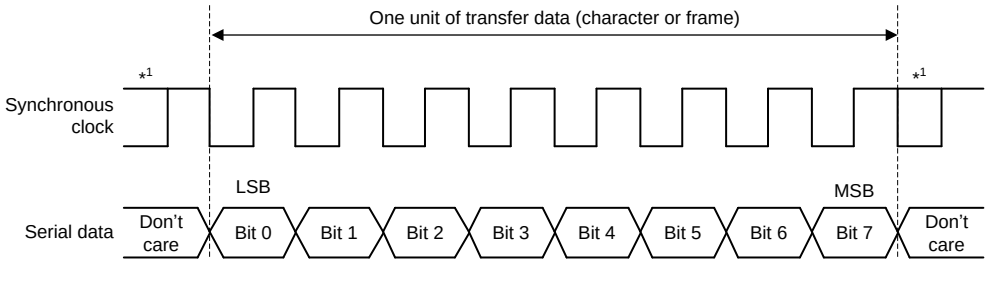

Figure 22.10 Example of SCIFA Receive Operation in Asynchronous Mode Using Model Control Function (RTS#)

### 22.3.3 Operation in Clock Synchronous Mode

In clock synchronous mode, the SCIFA transmits and receives data in synchronization with clock pulses. This mode is suitable for high-speed serial communication.

Full-duplex communication is possible because the SCIFA transmitter and receiver are independent and share the same clock. Since the transmitter and the receiver have 16-stage FIFO buffers, respectively, continuous transmission or reception is possible by reading or writing data while transmission or reception is in progress.

**Figure 22.11** shows the general format in clock synchronous serial communication.




Note 1. High level except in continuous transmission/reception

Figure 22.11 Data Format in Clock Synchronous Communication (when LSB-First Transfer is Selected)

In clock synchronous serial communication, each data bit is output on the communication line from one falling edge of the synchronous clock to the next. Data is guaranteed valid at the rising edge of the synchronous 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 (when LSB-first transfer is selected).

In clock synchronous mode, the SCIFA receives data by synchronizing with the rising edge of the synchronous 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 or an external clock input from the SCK pin can be selected as the SCIFA transmit/receive clock according to the settings of the CM bit in the serial mode register (SMR) and the CKE[1:0] bits in the serial control register (SCR).

When the SCIFA operates on an internal clock, it outputs the synchronous clock signal at the SCK pin. Eight synchronous clock pulses are output per transmitted or received character. Unless the SCIFA is transmitting or receiving, the synchronous clock signal remains in the high state. When the SCIFA only receives data on an internal clock, the internal clock signal outputs while the RE bit in the SCR register is 1 until the number of data units in the receive FIFO reaches the specified reception trigger number.

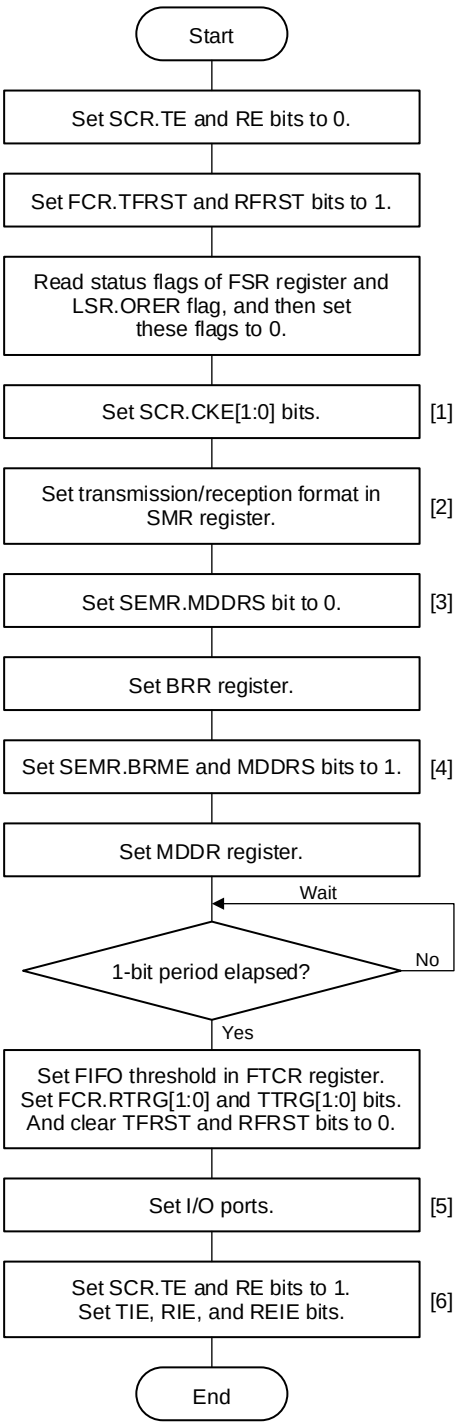
### (3) Transmitting and Receiving Data

#### SCIFA Initialization (in Clock Synchronous Mode)

Before transmitting or receiving, clear the TE and RE bits to 0 in the serial control register (SCR), and then initialize the SCIFA by performing the following procedure.

Similarly, before changing the mode or communication format, clear the TE and RE bits to 0, and then change it by performing the following procedure. Clearing TE to 0 initializes the transmit shift register (TSR). Clearing RE to 0, however, does not initialize the RDF, PER, FER, and ORER flags and the receive FIFO data register (FRDR), which retain their previous contents.

**Figure 22.12** shows a sample flowchart for initializing the SCIFA in clock synchronous mode.




[1] Set the TIE, RIE, TE, and RE bits in the SCR register to 0 before setting the CKE[1:0] bits.

[2] Set the transmission/reception format in the SMR register.

[3] Clear the SEMR.MDDRS bit to 0 and write a value corresponding to the bit rate in the BRR register. Not necessary if an external clock is used.

[4] Set the SEMR.BRME and MDDRS bits to 1 and write a value corresponding to the bit rate in the MDDR register. Not necessary if an external clock is used. Wait for at least 1-bit period before proceeding to the next step.

[5] Set I/O ports and the MPC register so that they can be used as the SCK, TXD, and RXD pin.

[6] After setting the MDDR register, wait for at least 1-bit, and then write 1 to the SCR.TE or RE bit. To permit interrupt requests, set TIE, RIE, and REIE bits to 1. Setting the TE and RE bits to 1 makes the TXD and RXD pins available to be used. In transmission, those pins enter the mark state, and in reception, they enter the idle state waiting for a start bit.

Figure 22.12 Sample Flowchart for SCIFA Initialization in Clock Synchronous Mode

### Transmitting Serial Data (in Clock Synchronous Mode)

**Figure 22.13** shows a sample flowchart for transmitting serial data in clock synchronous mode.

Follow the procedure given below for serial data transmission after enabling the SCIFA for transmission.

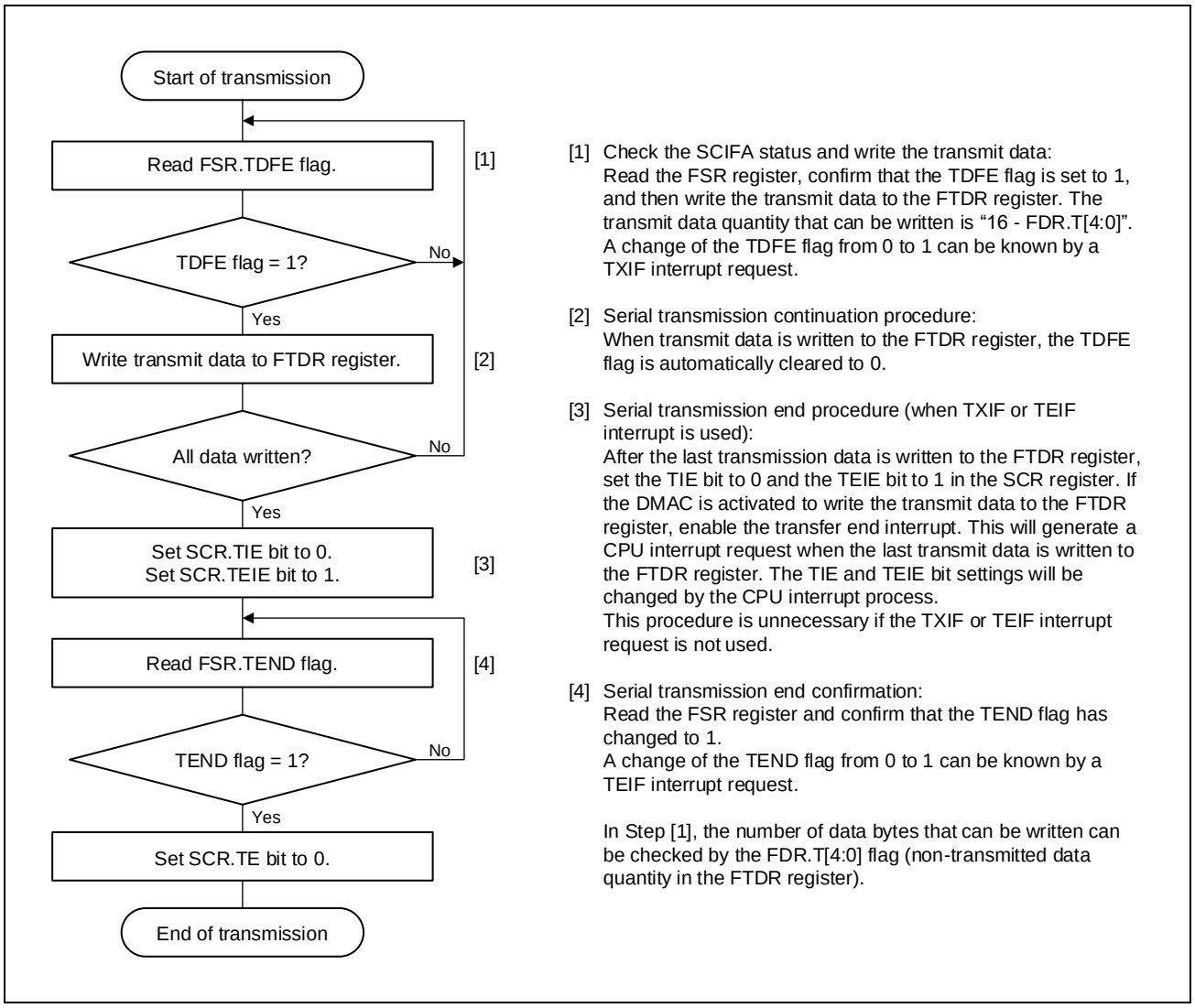
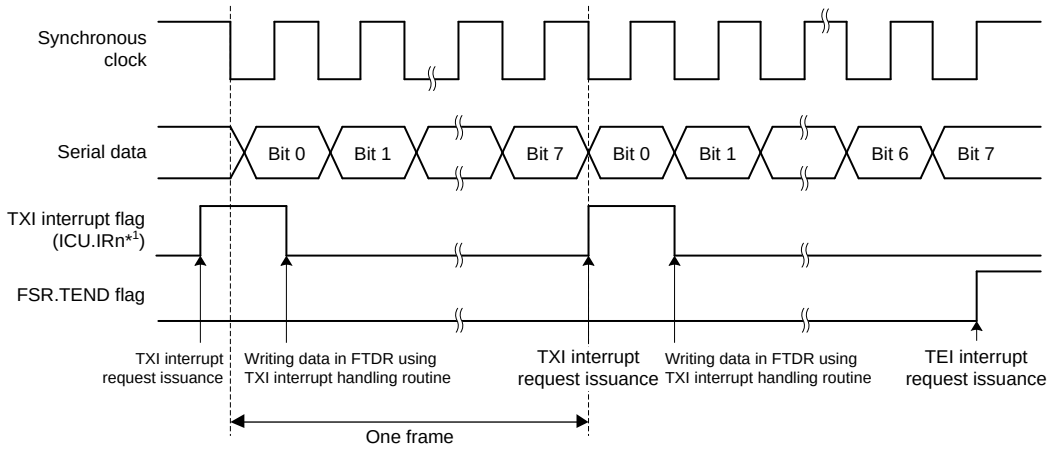



Figure 22.13 Sample Flowchart for Transmitting Serial Data in Clock Synchronous Mode

In clock synchronous mode, the SCIFA performs serial transmission as described below.

1. When data is written into the transmit FIFO data register (FTDR) by the TXI interrupt processing routine, the SCIFA transfers the data from the FTDR register to the transmit shift register (TSR) and starts transmission. Confirm that the TDFE flag in the serial status register (FSR) is set to 1 before writing transmit data to the FTDR register. The number of data bytes that can be written is “16 minus the specified number of non-transmitted data units”.
2. When data is transferred from the FTDR register to the TSR register and transmission is started, consecutive transmit operations are performed until there is no transmit data left in the FTDR register. When the number of transmit data bytes in the FTDR register becomes equal to or less than the transmission trigger number set in the FIFO control register (FCR) or FIFO trigger control register (FTCR), the TDFE flag in the FSR register is set. If the TIE bit in the serial control register (SCR) is set to 1 at this time, a transmit-FIFO-data-empty interrupt (TXI) request is generated.
   If clock output mode is selected, the SCIFA outputs eight synchronous clock pulses. If an external clock source is selected, the SCIFA outputs data in synchronization with the input clock. Data is output from the TXD pin in order from the LSB (b0) to the MSB (b7) (when LSB-first transfer is selected).
3. The SCIFA checks the transmit data of the FTDR register at the timing for sending the MSB (bit 7). If data is present, the data is transferred from the FTDR register to the TSR register, and then serial transmission of the next frame is started. If there is no data, the TXD pin holds the output level of the last data after the TEND flag in the FSR register is set to 1 and the MSB (bit 7) is output.
4. After the end of serial transmission, the SCK pin is held in the high state.

**Figure 22.14** shows an example of SCIFA transmit operation in clock synchronous mode.



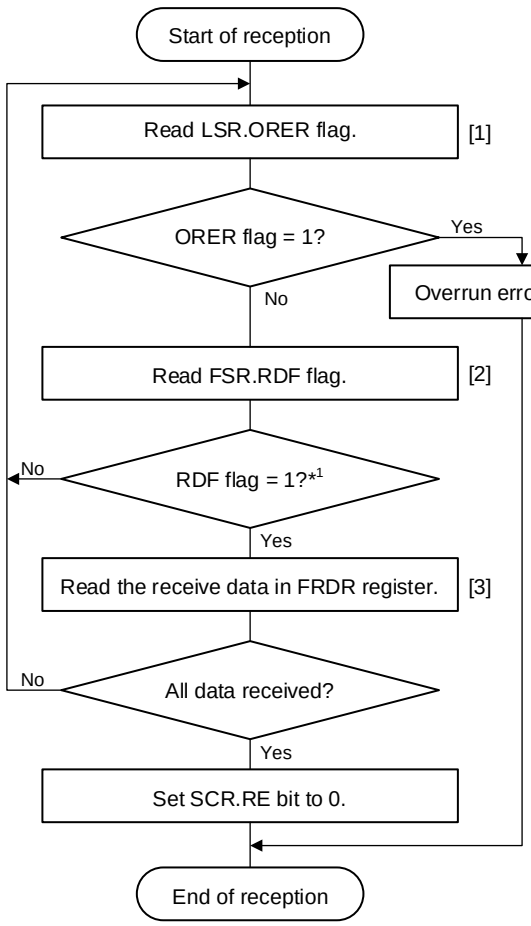

**Note:** For the corresponding interrupt number, see **Section 8, Interrupt Controller**.

Figure 22.14 Example of SCIFA Transmit Operation in Clock Synchronous Mode (when LSB-First Transfer is Selected)

### Receiving Serial Data (in Clock Synchronous Mode)

**Figure 22.15** and **Figure 22.16** show sample flowcharts for receiving serial data in clock synchronous mode.

Follow the procedure given below for serial data reception after enabling the SCIFA for reception. When switching from asynchronous mode to clock synchronous mode without SCIFA initialization, make sure that the ORER, PER, and FER flags in the line status register (LSR) are cleared to 0.




[1] Receive error handling:
When a receive error occurs, read the LSR.ORER flag and handles the error as specified, and then clear the ORER flag to 0. Reception cannot be restarted while the ORER flag is set to 1. A change of the ORER flag from 0 to 1 can be known by a BRIF interrupt request.

[2] Check the SCIFA status and read the receive data:
Read the FSR register, confirm that the RDF flag is set to 1, and then read the receive data at least as much as the amount specified by the receive FIFO threshold from those stored in the FRDR register. The receive data quantity that can be read can be checked by the FDR.R[4:0] flag.
A change of the RDF flag from 0 to 1 can be known by an RXIF interrupt request.

[3] Serial reception continuation procedure:
When receive data is read from the FRDR register, the RDF flag is automatically cleared to 0.

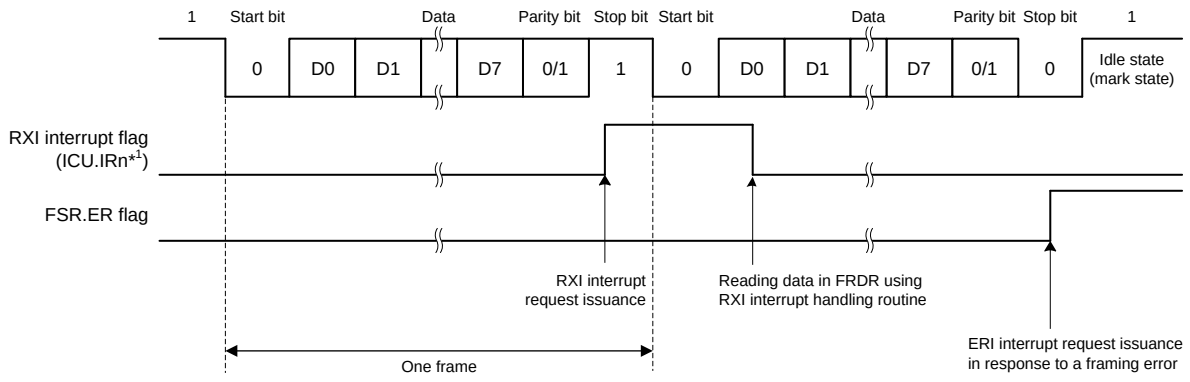
Note 1. The number of all the received data units should be an integral multiple of the receive FIFO threshold. If not, process returns to Step [1] and enters an infinite loop. In this case, it is necessary to set the timeout processing using a timer.

Figure 22.15 Sample Flowchart for Receiving Serial Data in Clock Synchronous Mode

In clock synchronous mode, the SCIFA performs serial reception as described below.

1. The SCIFA synchronizes with the synchronous clock input or output and starts reception.
2. Receive data is stored into the receive shift register (RSR) in order from the LSB to the MSB (when LSB-first transfer is selected). After receiving the data, the SCIFA checks whether the receive data can be transferred from the RSR register to the FRDR register. If data can be transferred, the SCIFA stores the received data in the FRDR register. If an overrun error is detected during the error check, further reception is not performed.
3. After the received data units equaling or exceeding the specified reception trigger number are stored in the FRDR register and the RDF flag is set to 1, a receive-data-full interrupt (RXI) request is generated when the RIE bit in the serial control register (SCR) is set to 1. When the ORER flag in the line status register (LSR) is set to 1 and the RIE or REIE bit in the SCR register is also set to 1, a break interrupt (BRI) request is generated.

**Figure 22.16** shows an example of SCIFA receive operation in clock synchronous mode.




Note 1. For the corresponding interrupt number, see **Section 8, Interrupt Controller**.

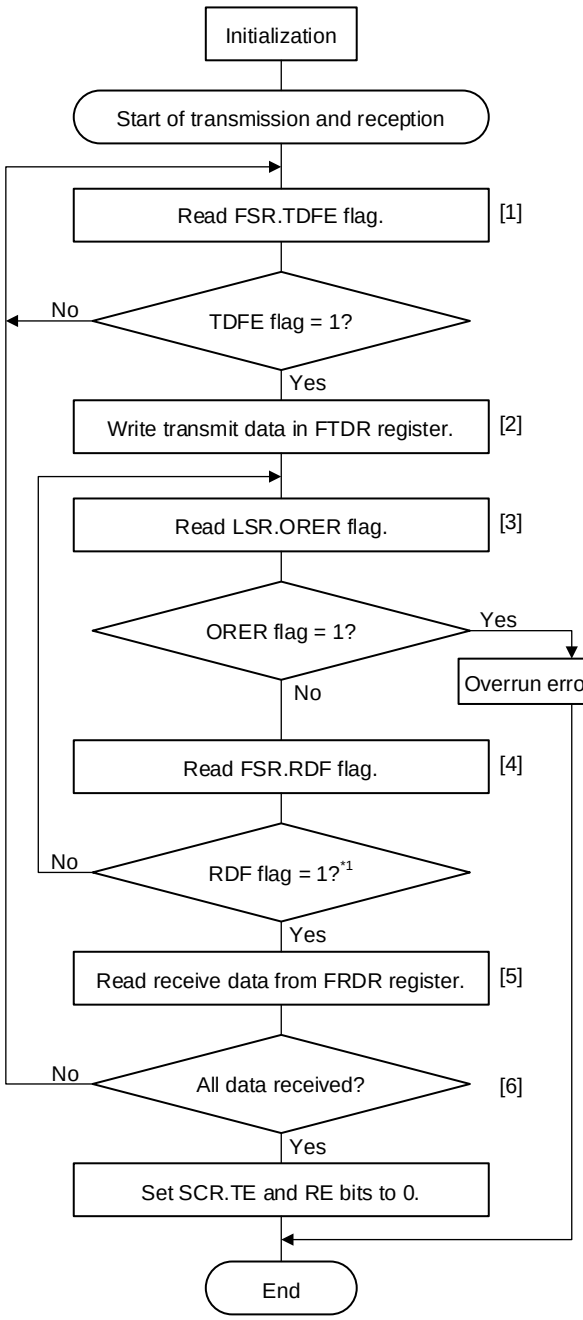
Figure 22.16 Example of SCIFA Receive Operation (when LSB-First Transfer is Selected)

### Transmitting and Receiving Serial Data Simultaneously (in Clock Synchronous Mode)

**Figure 22.17** shows a sample flowchart for transmitting and receiving serial data simultaneously in clock synchronous mode.

In simultaneous transmission/reception of serial data, number of receive data = number of transmit data = number of transmit data to be written to the FTDR register.

Follow the procedure given below for the simultaneous transmission/reception of serial data, after enabling the SCIFA for transmission/reception.




[1] Check the SCIFA status and write the transmit data:
Read the FSR register, confirm that the TDFE flag is set to 1, and then write the transmit data in the FTDR register.
A change of the TDFE flag from 0 to 1 can be known by a TXIF interrupt request.

[2] Serial transmission continuation procedure:
When transmit data is written to the FTDR register, the TDFE flag is automatically cleared to 0.

[3] Receive error handling:
When a receive error occurs, read the LSR.ORER flag and handles the error as specified, and then clear the ORER flag to 0. Reception cannot be restarted while the ORER flag is set to 1.
A change of the ORER flag from 0 to 1 can be known by a BRIF interrupt request.

[4] Check the SCIFA status and read the receive data:
Read the FSR register, confirm that the RDF flag is set to 1, and then read the receive data stored in the FRDR register.
A change of the RDF flag from 0 to 1 can be known by an RXIF interrupt request.

[5] Serial reception continuation procedure:
When receive data is read from the FRDR register, the RDF flag is automatically cleared to 0.

[6] Serial transmission/reception continuation procedure (supplementary):
To continue serial transmission/reception, read the FRDR register and clear the RDF flag to 0 before the FRDR register overflows and the ORER flag changes to 1. In addition, before the last bit of the last data written to the FTDR register is transmitted, write the data in the FTDR register and clear the TDFE flag to 0.
When writing data to the FTDR register or reading data from the FRDR register, the TDFE flag and RDF flag are automatically cleared to 0.

**Note:** When switching the operation from transmission or reception to the simultaneous transmission and reception, clear the TE and RE bits to 0 and then simultaneously set these bits to 1.

Note 1. The number of all the received data units should be an integral multiple of the receive FIFO threshold. If not, process returns to Step [1] and enters an infinite loop. In this case, it is necessary to set the timeout processing using a timer.

Figure 22.17 Sample Flowchart for Simultaneous Transmitting/Receiving Serial Data in Clock Synchronous Mode

## 22.4 Bit Rate Modulation

Using the bit rate modulation, the bit rate can be corrected by skipping the specified number of clock pulses input to the baud rate generator. To correct the bit rate, only the number of clock pulses specified in the MDDR register are enabled among 256 internal clock pulses specified by the CKS1 and CKS0 bits in the SMR register in a way that forms average intervals.

**Figure 22.18** shows an example where P0φ is selected by the CKS[1:0] bits in SMR and BRR and MDDR are set to 0 and 160, respectively, in asynchronous mode. In this example, the cycle of the base clock is evenly corrected (256/160) and the bit rate is also corrected (160/256). Note that skipping an internal clock causes bias and expansion or contraction is generated in the pulse width of the base clock.

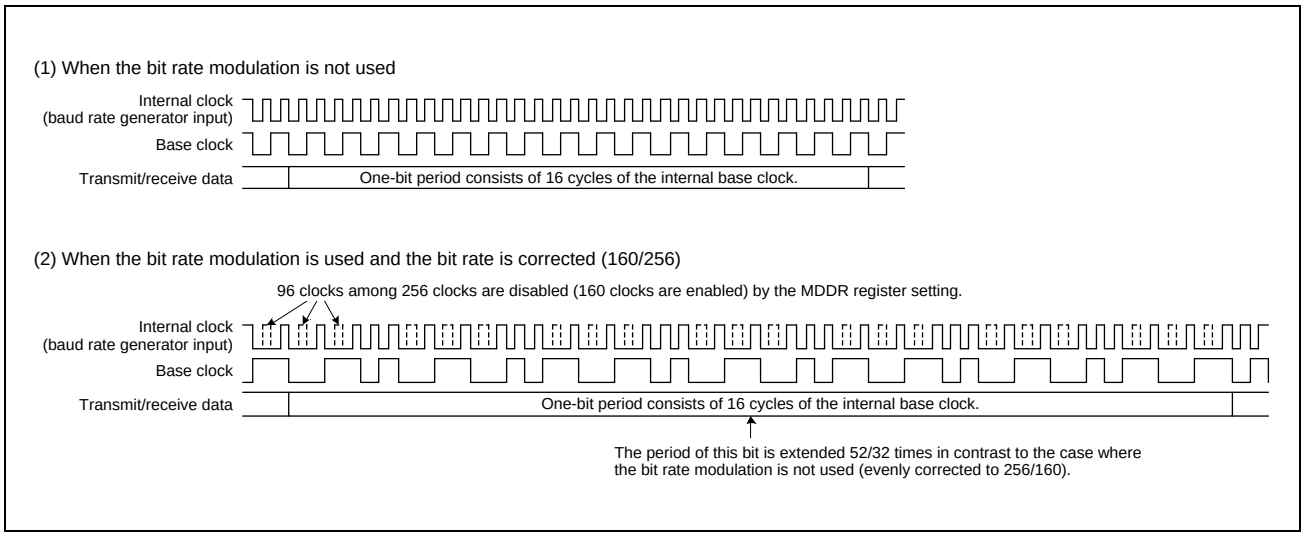



Figure 22.18 Example of Internal Base Clock when Bit Modulation is Used

## 22.5 Interrupt Sources

The SCIFA has six interrupt sources: transmit-FIFO-data-empty (TXI), receive-error (ERI), receive-FIFO-data-full (RXI), break (BRI), transmit-end (TEI), receive-data-ready (DRI). The TEI, DRI, ERI, and BRI interrupts share the same vector number.

**Table 22.19** shows the interrupt sources and priority. The interrupt sources can be enabled or disabled using the TIE, RIE, REIE, TEIE bits in the SCR register and are separately input to the interrupt controller.

When the quantity of transmit data written in the FTDR register as a result of transmission is equal to or less than the specified transmission trigger number, the TDFE flag in the serial status register (FSR) is set to 1 and a TXI interrupt request is generated.

When the data units equaling or exceeding the specified transmission trigger number are stored in the receive FIFO register (FRDR) and the RDF flag in the FSR register is set to 1, a receive data full interrupt (RXI) request is generated. When the quantity of received data in the FRDR register is below the specified reception trigger number and no next data has been received yet even after the period of 15 ETUs elapsed*1 from the last stop bit, the DR flag in the FSR register is set to 1 and a receive data ready interrupt (DRI) request is generated. In clock synchronous mode, a DRI interrupt request is not generated.

When the BRK flag in the FSR register or the ORER flag in the LSR register is set to 1, a BRI interrupt request is issued. When the ER flag in the FSR register is set to 1, an ERI interrupt request is issued.

When the TEND flag in the FSR register is set to 1, a TEI interrupt request is issued.

When the RIE bit is cleared to 0 and the REIE bit in the SCR register is set to 1, an ERI and a BRI interrupt requests are issued but an RXI interrupt request is not.

An TXI interrupt indicates that transmit data can be written and an RXI interrupt indicates that receive data is stored in the FRDR register.

**Note 1.** It is equivalent to 1 and half frames of 8-bit format with one stop bit (ETU: Element Time Unit).

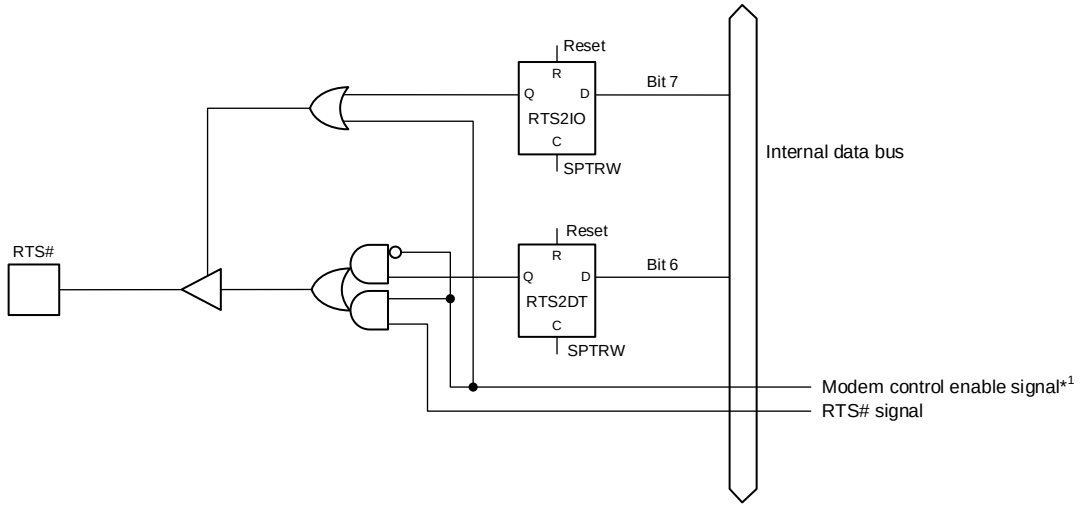
Table 22.19 SCIFA Interrupt Sources

| Name | Level/Edge | Interrupt Source | Interrupt Enable Bit | DMAC Activation | Priority |
|---|---|---|---|---|---|
| BRI | Level | Interrupt caused by break (BRK) or overrun (ORER). | RIE or REIE | Impossible | High |
| ERI | Level | Interrupt caused by framing or parity (ER). | RIE or REIE | Impossible | |
| RXI | Level | Interrupt caused by receive FIFO data full (RDF). | RIE | Possible | |
| TXI | Level | Interrupt caused by transmit FIFO data empty (TDFE). | TIE | Possible | |
| TEI | Level | Interrupt caused by transmit end (TEND). | TEIE | Impossible | |
| DRI | Level | Interrupt caused by receive data ready (DR). | RIE | Impossible | Low |

**Note:** The TEI and DRI interrupts share the same vector number.
The ERI and BRI interrupts share the same vector number.
If CPU processing is used, clear the flag after the block transfer. If the DMAC is activated, access to the flags is prohibited.

## 22.6 Serial Port Register (SPTR) and SCIFA-Related Pins

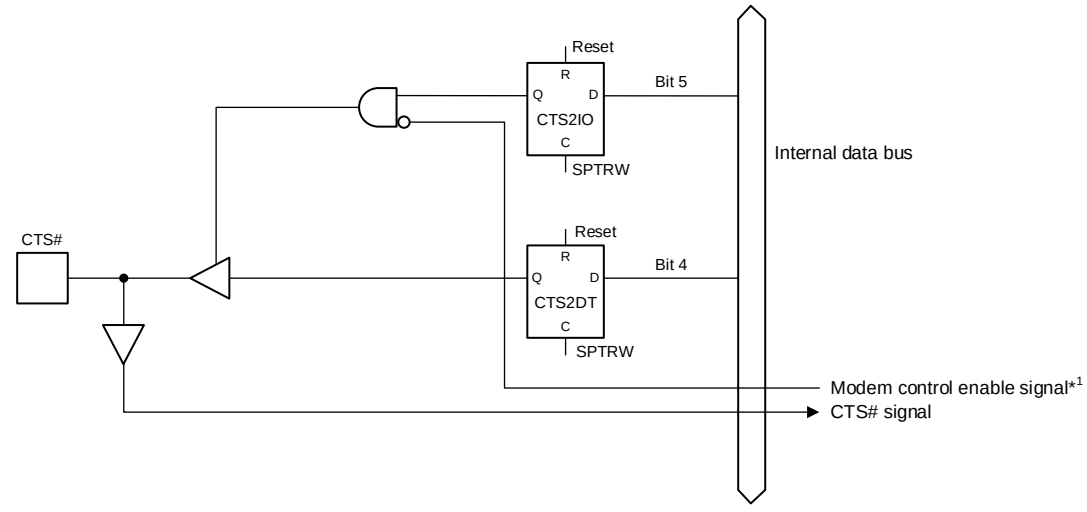
**Figure 22.19** to **Figure 22.22** show the relationships between the SPTR register and the SCIFA-related pins.




**Note:** SPTRW: SPTR write

Note 1. These signals allow the RTS# pin to enter the modem control state with the MCE bits in FCR.

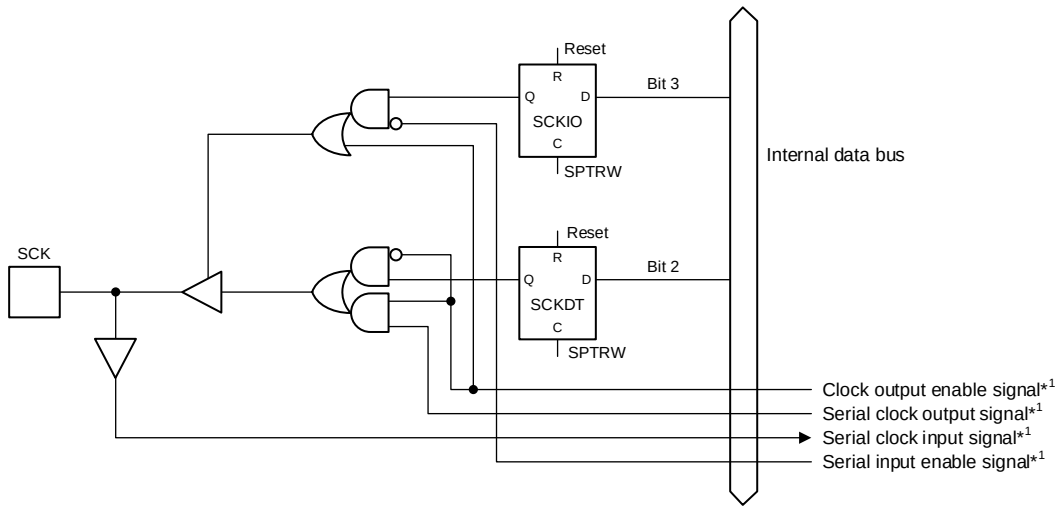
Figure 22.19 RTS2IO Bit and RTS2DT Bit in the SPTR Register, and RTS# Pin




**Note:** SPTRW: SPTR write

Note 1. These signals allow the CTS# pin to enter the modem control state with the MCE bits in FCR.

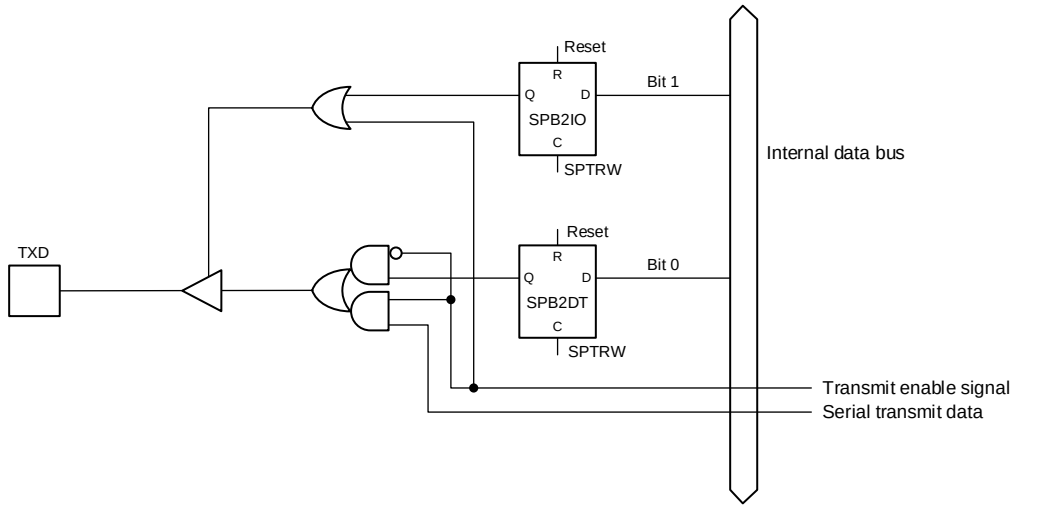
Figure 22.20 CTS2IO Bit and CTS2DT Bit in the SPTR Register, and CTS# Pin




**Note:** SPTRW: SPTR write

Note 1. These signals control the SCK pin according to the settings of the CM bit in SMR and CKE[1:0] bits in SCR.

Figure 22.21 SCKIO Bit and SCKDT Bit in the SPTR Register, and SCK Pin




**Note:** SPTRW: SPTR write

Figure 22.22 SPB2IO Bit and SPB2DT Bit in the SPTR Register, and TXD Pin

## 22.7 Noise Cancellation

**Figure 22.23** shows the configuration of the noise filter used for noise cancellation. The noise filter consists of a two-stage flip-flop circuits and a match detection circuit. When the input signals of the noise filter and the output signals of the two-stage flip-flop circuits completely match, the matched level is conveyed as an internal signal. Unless otherwise matched, the previous value is retained. (When the levels sampled on three consecutive cycles of the sampling clock of the noise filter match, the signal is considered valid. If three consecutive sampled values do not match, the signal is considered to be noise rather than a received signal).

In asynchronous mode, the noise cancellation can be applied to the receive signal input to the RXDn pin. The receive level of the RXDn pin is taken in the flip-flop circuit of the noise filter on the base clock (the clock with a frequency 16 or 8 times the transfer rate*[1]).

If the base clock is stopped once with the noise filter enabled and then the base clock input is restarted again, the noise filter operation resumes from the state where the clock was stopped. When SCR.RE is set to 0 during input of the base clock, the noise filter outputs 0 as the internal RxDn signal. The internal match detector continues operating even while operations for reception are stopped, and the result from the last time previous consecutive samples matched is output at the same time as operations for reception are resumed.

**Note 1.** A frequency 16 times bit rate when the SEMR.ABCS0 bit and the SEMR.BGDM bit are both 0, and a frequency 8 times bit rate when either the SEMR.ABCS0 bit or the SEMR.BGDM bit is 1.

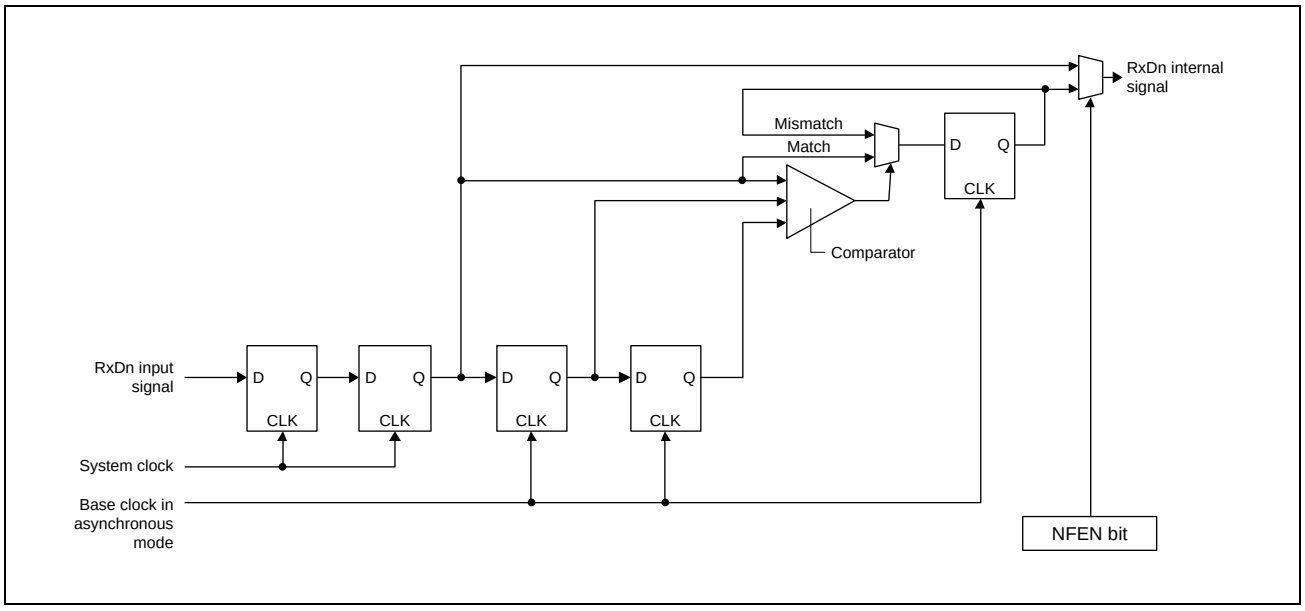



Figure 22.23 Block Diagram of Digital Noise Filter Circuit

## 22.8 Usage Notes

The following is the notes on using the SCIFA.

### 22.8.1 FTDR Register Writing and TDFE Flag

The TDFE flag in the serial status register (FSR) is set when the number of transmit data bytes written in the transmit FIFO data register (FTDR) has fallen below the transmission trigger number set by bits TTRG[1:0] in the FIFO control register (FCR) or bits TFTC[4:0] in the FIFO trigger control register (FTCR). After the TDFE flag is set, transmit data up to the number of empty bytes in the FTDR register can be written, allowing efficient continuous transmission.

However, if the number of data bytes written in the FTDR register is equal to or less than the specified transmission trigger number, the TDFE flag will be set to 1 again even after being read as 1 and cleared to 0.

The number of transmit data bytes in the FTDR register can be checked by the 8 higher-order bits of the FIFO data count register (FDR).

### 22.8.2 FRDR Register Reading and RDF Flag

The RDF flag in the serial status register (FSR) is set when the number of receive data bytes in the receive FIFO data register (FRDR) has become equal to or greater than the reception trigger number set by bits RTRG[1:0] in the FIFO control register (FCR) or bits RFTC[4:0] in the FIFO trigger control register (FTCR). After the RDF flag is set, receive data equivalent to the trigger number can be read from the FRDR register, allowing efficient continuous reception.

However, if the number of data bytes in the FRDR register exceeds the reception trigger number, the RDF flag will be set to 1 again even after being read as 1 and cleared to 0.

The number of receive data bytes in the FRDR register can be checked by the 8 lower-order bits of the FIFO data count register (FDR).

### 22.8.3 Break Detection and Processing

When a framing error (FER) is detected, a break signal can be detected by reading the RXD pin value directly. In a break, the input from the RXD pin becomes all low. Therefore, the FER flag in the serial status register (FSR) is set to 1 and the parity error flag (PER) may also be set to 1.

Upon detection of a break signal, the SCIFA stops the received data transfer to the FRDR register but continues the receive operation.

### 22.8.4 Writing to the SPTR Register

b6, b4, b2, and b0 of the SPTR register respectively indicate the input status of their corresponding pins. (See the description of each bit of **Section 22.2.12, Serial Port Register (SPTR)** for details.)

Writings to these bits in 1-bit unit are handled as read-modify-write, which may lead to undesired values to be written. To avoid this, when modifying the SPB2DT or SPB2IO bit, for example, write the other bit (the bit used in combination) at the same time.

### 22.8.5 Break Signal Transmission

The output signal from the TXD pin is determined by the SPB2IO bit and the SPB2DT bit in the serial port register (SPTR). The break signal can be sent by using these bits.

The TXD pin does not function as a transmit data output pin during the period from when the SCIFA is initialized to when the TE bit in the SCR register is set to 1 (transmission possible). The TXD pin status during this period is replaced by the SPB2DT bit value. Therefore, the SPB2IO and SPB2DT bits in the SPTR register must have been set to 1 (high output) at first (mark (high) status).

To transmit the break signal during serial transmission, set the SPB2IO bit in the SPTR register to 1, clear the SPB2DT bit to 0 (specify a low level), and then clear the TE bit in the SCR register to 0 (transmission stop). Clearing the TE bit to 0 initializes the transmitter regardless of the current transmission status, and outputs a low level from the TXD pin.

### 22.8.6 Receive Data Sampling Timing and Receive Margin in Asynchronous Mode

The SCIFA operates on a base clock with a frequency 16 times the transfer rate*[1]. In reception, the SCIFA internally latches the received data at the rising edge of the eighth base clock pulse*[1]. The timing is shown in **Figure 22.24**.

**Note 1.** This is an example when the SEMR.ABCS0 bit is 0. When the ABCS0 bit is 1, a frequency of 8 times the bit rate is used as a base clock and receive data is sampled at the rising edge of the 4th pulse of the base clock. The receive margin in asynchronous mode can therefore be expressed as shown in equation 1.

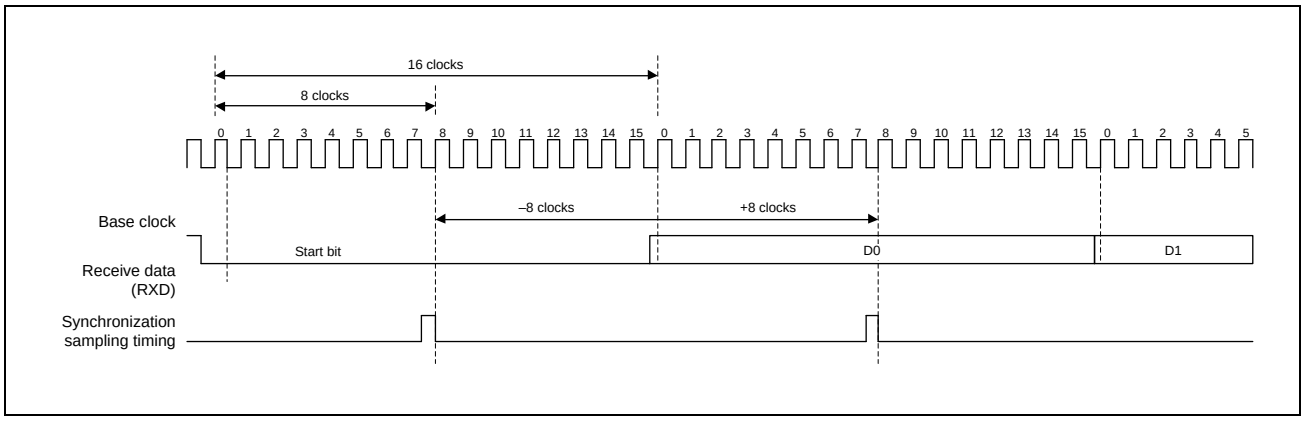



Figure 22.24 Receive Data Sampling Timing in Asynchronous Mode

The receive margin in asynchronous mode can therefore be expressed as shown in equation 1.

[Equation 1]

$$M = \{0.5 - \frac{1}{2N} - (L - 0.5)F - \frac{|D-0.5|}{N}(1 + F)\} \times 100[\%]$$

Where: M: Receive margin (%)

N: Ratio of clock frequency to bit rate (N = 16)

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 and D = 0.5, the receive margin is 46.875%, as given by equation 2.

[Equation 2]

When D = 0.5 and F = 0:

$$M = (0.5 - 1/(2 \times 16)) \times 100\% = 46.875\%$$

This is a theoretical value. A reasonable margin to allow in system designs is 20% to 30%.

### 22.8.7 Note on FER Flag and PER Flag in Serial Status Register (FSR)

The FER flag and PER flag in the serial status register (FSR) are status flags that apply to next entry to be read from the receive FIFO data register (FRDR). After the CPU or DMAC reads the receive FIFO data register, the flags of framing errors and parity errors in the receive data will be cleared. To check the received data for the states of framing errors and parity errors, only read the receive FIFO data register after reading the serial status register.

### 22.8.8 Notes on External Clock Input in Clock Synchronous Mode

Before setting the TE and RE bits in the serial control register (SCR) to 1, wait for four or more cycles of the peripheral operating clock after the external clock (SCK) is changed from 0 (low) to 1 (high). To input the external clock (SCK) (to start communication), wait for one or more cycles of the external clock after the TE and RE bits in the SCR register are set to 1.

### 22.8.9 Module Standby Mode Setting

SCIFA operation can be disabled or enabled using the standby control register. As the initial setting, the SCIFA operation is halted. Register access is enabled by clearing module standby mode. For details, refer to **Section 42, Low Power Mode**.

### 22.8.10 Notes on Operation for Reception when an Internal Clock is Selected in Clock Synchronous Mode

When an internal clock is selected as the clock for reception in clock-synchronous mode, if the number of data stored through the receive FIFO data register (FRDR) becomes equal to or greater than the specified reception trigger number, the RDF flag is set, the RXI interrupt request is generated and, at the same time, output of the synchronizing clock and reception of serial data are stopped. Once the number of data are again less than the specified reception trigger number, output of the synchronizing clock and the reception of serial data are restarted. In addition, if an internal clock is selected for reception in clock synchronous mode, the ORER flag is not set to 1 since no overrun occurs. Accordingly, overruns (indicated by the ORER flag) cannot be used as a BRI interrupt source.

# 23. Serial Communications Interface (SCIg)

This MCU has two independent serial communications interface (SCI) channels. The SCIs consist of the SCI0 and SCI1 modules, referred to in common as “SCIg modules”.

The SCIg modules (SCI0 and SCI1) can handle both asynchronous and clock synchronous serial communications. Asynchronous serial data communications can be carried out with standard asynchronous communications chips such as a Universal Asynchronous Receiver/Transmitter (UART) or Asynchronous Communications Interface Adapter (ACIA). As an extended function in asynchronous communications mode, the SCI also supports smart card (IC card) interfaces conforming to ISO/IEC 7816-3 (standard for Identification Cards).

## 23.1 Overview

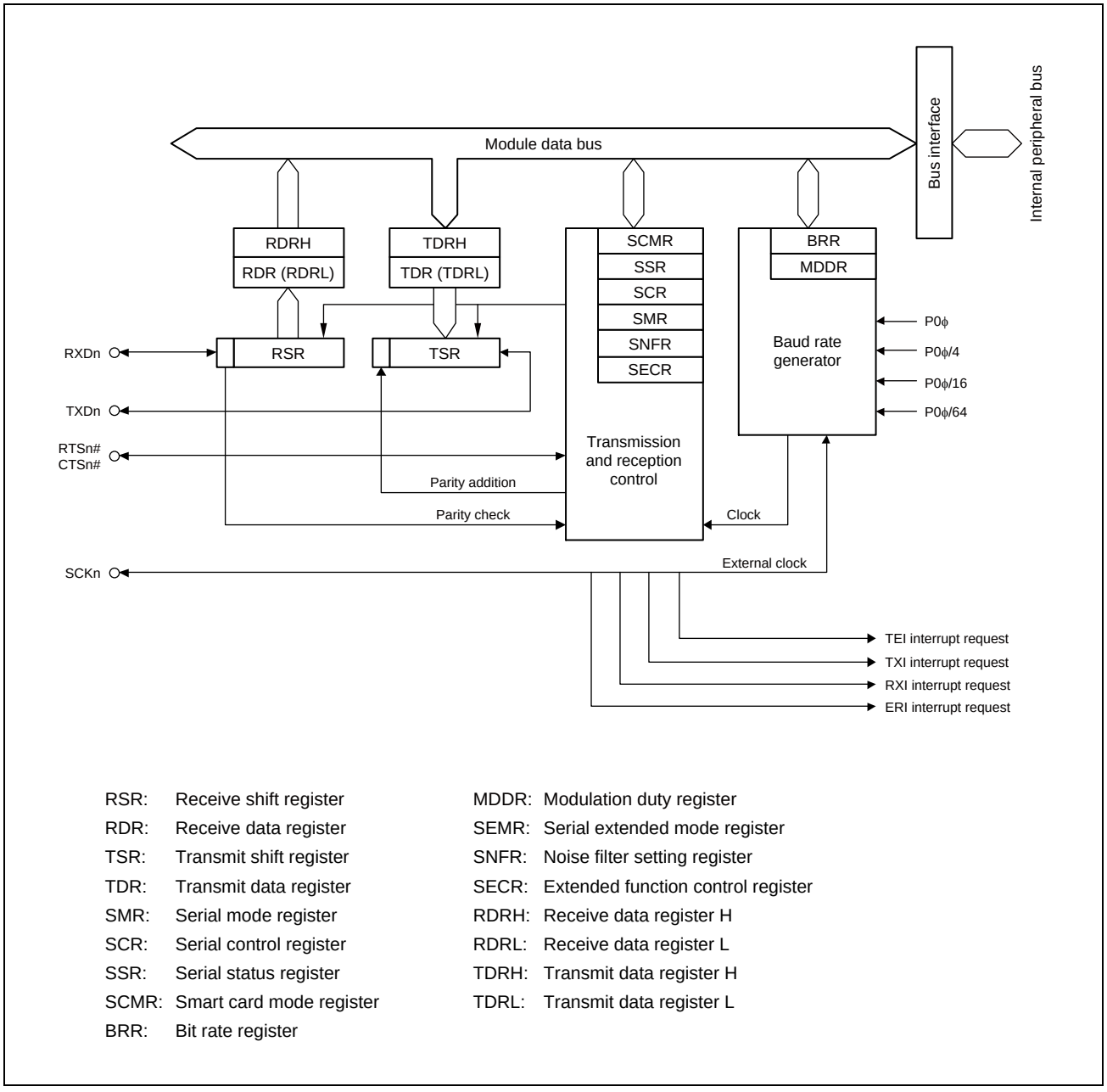
**Table 23.1** lists the specifications of the SCIg modules. **Figure 23.1** shows the block diagram of the SCI0 and SCI1 modules.

Table 23.1 SCIg Specifications (1/2)

| Item | | Description |
|---|---|---|
| Serial communications modes | | • Asynchronous<br>• Clock synchronous<br>• Smart card interface<br>• IrDA (only for channel 0) |
| Transfer speed | | Bit rate specifiable with the on-chip baud rate generator. |
| Full-duplex communications | | Transmitter: Continuous transmission possible using double-buffer structure.<br>Receiver: Continuous reception possible using double-buffer structure. |
| I/O pins | | See **Table 23.2**. |
| Data transfer | | Selectable as LSB first or MSB first transfer |
| Interrupt sources | | Transmit end, transmit data empty, receive data full, and receive error |
| Low power consumption function | | Module stop state can be set for each channel. |
| Asynchronous mode | Data length | 7, 8, or 9 bits |
| | Transmission stop bit | 1 or 2 bits |
| | Parity | Even parity, odd parity, or no parity |
| | Receive error detection | Parity, overrun, and framing errors |
| | Hardware flow control | CTSn# and RTSn# pins can be used in controlling transmission/reception. |
| | Start-bit detection | Low level or falling edge is selectable. |
| | Break detection | When a framing error occurs, a break can be detected by reading the RXDn pin level directly. |
| | Clock source | An internal or external clock can be selected. |
| | Double-speed mode | Baud rate generator double-speed mode is selectable. |
| | Multi-processor communications function | Serial communication among multiple processors |
| | Noise cancellation | The signal paths from input on the RXDn pins incorporate digital noise filters. |
| Clock synchronous mode | Data length | 8 bits |
| | Receive error detection | Overrun error |
| | Hardware flow control | CTSn# and RTSn# pins can be used in controlling transmission/reception. |

Table 23.1 SCIg Specifications (2/2)

| Item | | Description |
|---|---|---|
| Smart card interface mode | Error processing | An error signal can be automatically transmitted when detecting a parity error during reception |
| | | Data can be automatically retransmitted when receiving an error signal during transmission |
| | Data type | Both direct convention and inverse convention are supported. |
| Bit rate modulation function | | Correction of outputs from the on-chip baud rate generator can reduce errors. |




RSR: Receive shift register
RDR: Receive data register
TSR: Transmit shift register
TDR: Transmit data register
SMR: Serial mode register
SCR: Serial control register
SSR: Serial status register
SCMR: Smart card mode register
BRR: Bit rate register
MDDR: Modulation duty register
SEMR: Serial extended mode register
SNFR: Noise filter setting register
SECR: Extended function control register
RDRH: Receive data register H
RDRL: Receive data register L
TDRH: Transmit data register H
TDRL: Transmit data register L

Figure 23.1 Block Diagram of SCIg (SCI0 and SCI1)

**Table 23.2** lists the pin configuration of the SCIs for the individual modes.

Table 23.2 SCI Pin Configuration in Asynchronous Mode and Clock Synchronous Mode

| Channel | Pin Name | I/O | Function |
|---|---|---|---|
| SCI0 | SCI0_SCK | I/O | SCI0 clock input/output |
| | SCI0_RXD | Input | SCI0 receive data input |
| | SCI0_TXD | Output | SCI0 transmit data output |
| | SCI0_CTS#/RTS# | I/O | SCI0 transfer start control input/output |
| SCI1 | SCI1_SCK | I/O | SCI1 clock input/output |
| | SCI1_RXD | Input | SCI1 receive data input |
| | SCI1_TXD | Output | SCI1 transmit data output |
| | SCI1_CTS#/RTS# | I/O | SCI1 transfer start control input/output |

## 23.2 Register Descriptions

Table 23.3 Register configuration

BASE Address (Cortex A55 Address Space): (ch0) H'0_1004_D000 (ch1) H'0_1004_D400

| Channel | Register Name | Abbreviation | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|---|
| 0 | Serial Mode Register | SMR | R/W | H'00 | BASE(ch0) + H'0000 | 8 |
| | Bit Rate Register | BRR | R/W | H'FF | BASE(ch0) + H'0001 | 8 |
| | Serial Control Register | SCR | R/W | H'00 | BASE(ch0) + H'0002 | 8 |
| | Transmit Data Register | TDR | R/W | H'FF | BASE(ch0) + H'0003 | 8 |
| | Serial Status Register | SSR | R/W*1 | H'84 | BASE(ch0) + H'0004 | 8 |
| | Receive Data Register | RDR | R | H'00 | BASE(ch0) + H'0005 | 8 |
| | Smart Card Mode Register | SCMR | R/W*1 | H'F2 | BASE(ch0) + H'0006 | 8 |
| | Serial Extended Mode Register | SEMR | R | H'00 | BASE(ch0) + H'0007 | 8 |
| | Noise Filter Setting Register | SNFR | R | H'00 | BASE(ch0) + H'0008 | 8 |
| | Extended Function Control Register | SECR | R/W | H'00 | BASE(ch0) + H'000D | 8 |
| | Transmit Data Register H | TDRH | R/W | H'FF | BASE(ch0) + H'000E | 8 |
| | Transmit Data Register L | TDRL | R/W | H'FF | BASE(ch0) + H'000F | 8 |
| | Transmit Data Register HL | TDRHL | R/W | H'FFFF | BASE(ch0) + H'000E | 16 |
| | Receive Data Register H | RDRH | R | H'00 | BASE(ch0) + H'0010 | 8 |
| | Receive Data Register L | RDRL | R | H'00 | BASE(ch0) + H'0011 | 8 |
| | Receive Data Register HL | RDRHL | R | H'0000 | BASE(ch0) + H'0010 | 16 |
| | Modulation Duty Register | MDDR | R/W | H'FF | BASE(ch0) + H'0012 | 8 |
| 1 | Serial Mode Register | SMR | R/W | H'00 | BASE(ch1) + H'0000 | 8 |
| | Bit Rate Register | BRR | R/W | H'FF | BASE(ch1) + H'0001 | 8 |
| | Serial Control Register | SCR | R/W | H'00 | BASE(ch1) + H'0002 | 8 |
| | Transmit Data Register | TDR | R/W | H'FF | BASE(ch1) + H'0003 | 8 |
| | Serial Status Register | SSR | R/W*1 | H'84 | BASE(ch1) + H'0004 | 8 |
| | Receive Data Register | RDR | R | H'00 | BASE(ch1) + H'0005 | 8 |
| | Smart Card Mode Register | SCMR | R/W*1 | H'F2 | BASE(ch1) + H'0006 | 8 |
| | Serial Extended Mode Register | SEMR | R | H'00 | BASE(ch1) + H'0007 | 8 |
| | Noise Filter Setting Register | SNFR | R | H'00 | BASE(ch1) + H'0008 | 8 |
| | Extended Function Control Register | SECR | R/W | H'00 | BASE(ch1) + H'000D | 8 |
| | Transmit Data Register H | TDRH | R/W | H'FF | BASE(ch1) + H'000E | 8 |
| | Transmit Data Register L | TDRL | R/W | H'FF | BASE(ch1) + H'000F | 8 |
| | Transmit Data Register HL | TDRHL | R/W | H'FFFF | BASE(ch1) + H'000E | 16 |
| | Receive Data Register H | RDRH | R | H'00 | BASE(ch1) + H'0010 | 8 |
| | Receive Data Register L | RDRL | R | H'00 | BASE(ch1) + H'0011 | 8 |
| | Receive Data Register HL | RDRHL | R | H'0000 | BASE(ch1) + H'0010 | 16 |
| | Modulation Duty Register | MDDR | R/W | H'FF | BASE(ch1) + H'0012 | 8 |

Note 1. Partly, R only

### 23.2.1 Receive Shift Register (RSR)

RSR is a shift register which is used to receive serial data input from the RXDn pin and converts it into parallel data. When one frame of data has been received, it is automatically transferred to the RDR register.

The RSR register cannot be directly accessed by the CPU.

### 23.2.2 Receive Data Register (RDR)

| Bit | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | | | | | | | | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R |

RDR is an 8-bit register that stores receive data.

When one frame of serial data has been received, the received serial data is transferred from RSR to RDR. Then the RSR register can receive the next data.

Since RSR and RDR function as a double buffer in this way, continuous receive operations can be performed.

Read RDR only once after a receive data full interrupt (RXI) has occurred. Note that if next one frame of data is received before reading receive data from RDR, an overrun error occurs.

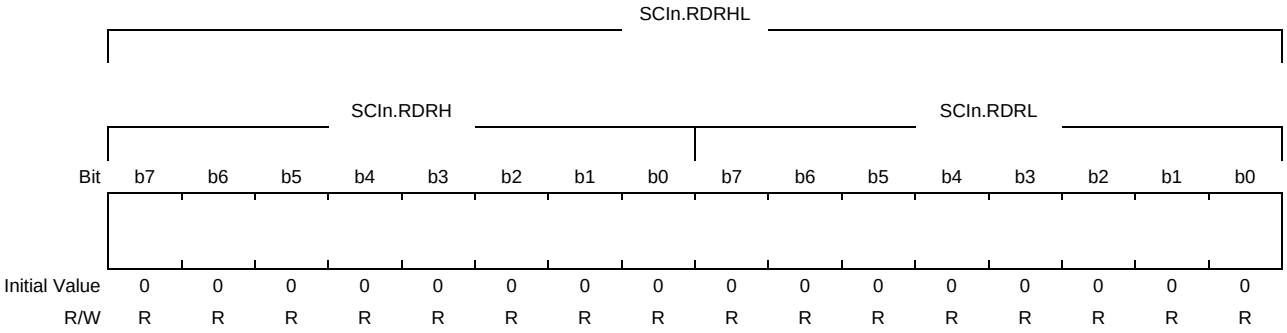
RDR cannot be written to by the CPU.

### 23.2.3 Receive Data Register H, L, HL (RDRH, RDRL, RDRHL)

**Receive Data Register H (RDRH)**

**Receive Data Register L (RDRL)**

**Receive Data Register HL (RDRHL)**




RDRH and RDRL are 8-bit registers that store receive data. Use these registers when asynchronous mode and 9-bit data length are selected.

RDRL is the shadow register of RDR; i.e. access to RDRL is equivalent to access to RDR.

After one frame of data is received, the received data is transferred from the RSR register to these registers, thus allowing the RSR register to receive the next data.

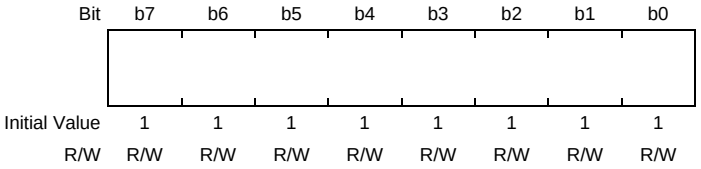
The RSR, RDRH and RDRL registers have a double-buffered construction to enable continuous reception.

Read RDRH and RDRL should be performed only once in the order from RDRH to RDRL when a receive data full interrupt (RXI) request is issued. Note that an overrun error occurs when the next frame of data is received before the received data has been read from RDRL.

The CPU cannot write to the RDRH and RDRL registers. Bits 0 to 7 in RDRH are fixed to 0. These bits are read as 0. The write value should be 0.

The RDRHL register can be accessed in 16-bit units.

### 23.2.4 Transmit Data Register (TDR)




TDR is an 8-bit register that stores transmit data.

When the SCI detects that TSR is empty, it transfers the transmit data written in TDR to TSR and starts transmission. The double-buffered structures of TDR and TSR enable continuous serial transmission. If the next transmit data has already been written to TDR when one frame of data is transmitted, the SCI transfers the written data to TSR to continue transmission.

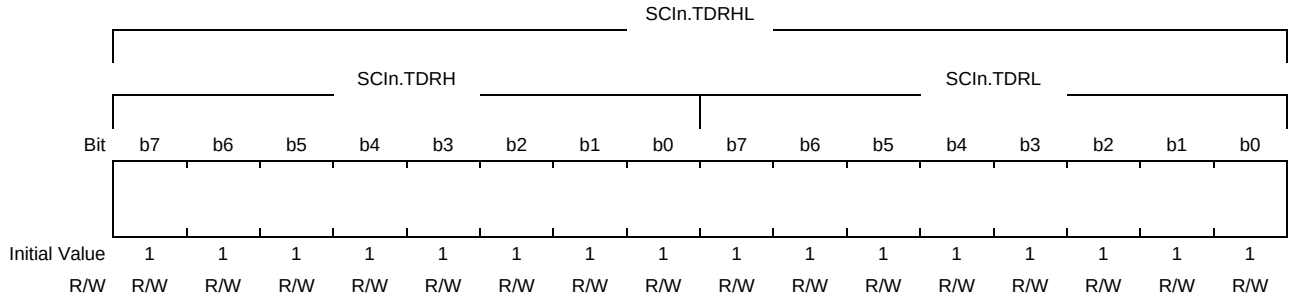
The CPU is able to read from or write to TDR at any time. Only write transmit data to TDR once after each instance of the transmit data empty interrupt (TXI).

## 23.2.5 Transmit Data Register H, L, HL (TDRH, TDRL, TDRHL)

**Transmit Data Register H (TDRH)**

**Transmit Data Register L (TDRL)**

**Transmit Data Register HL (TDRHL)**




TDRH and TDRL are 8-bit registers that store transmit data. Use these registers when asynchronous mode and 9-bit data length are selected.

TDRL is the shadow register of TDR; i.e. access to TDRL is equivalent to access to TDR.

When empty space is detected in the TSR register, the transmit data stored in the TDRH and TDRL registers is transferred to TSR; i.e., transmitting is started.

The TSR, TDRH and TDRL registers have a double-buffered construction to realize continuous reception. When the next data to be transmitted is stored in TDRL after one frame of data has been transmitted, the transmitting operation is continued by transfer to the TSR register.

The CPU can read and write to the TDRH and TDRL registers. Bits 0 to 7 in RDRH are fixed to 1. These bits are read as 1. The write value should be 1.

Writing transmit data to the TDRH and TDRL registers should be performed only once in the order from TDRH to TDRL when a transmit data empty interrupt (TXI) request is issued.

The TDRHL register can be accessed in 16-bit units.

## 23.2.6 Transmit Shift Register (TSR)

TSR is a shift register that transmits serial data.

To perform serial data transmission, the SCI first automatically transfers transmit data from TDR to TSR, and then sends the data to the TXDn pin.

TSR cannot be directly accessed by the CPU.

## 23.2.7 Serial Mode Register (SMR)

NOTE

Some bits in SMR have different functions in smart card interface mode and non-smart card interface mode.

### (1) Non-Smart Card Interface Mode (SCMR.SMIF = 0)

| Bit | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | CM | CHR | PE | PM | STOP | MP | CKS[1:0] | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W*4 | R/W*4 | R/W*4 | R/W*4 | R/W*4 | R/W*4 | R/W*4 | R/W*4 |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| b1, b0 | CKS[1:0] | 00b | R/W*4 | Clock Select<br>b1 b0<br>0 0: P0φ clock (n = 0)*1<br>0 1: P0φ/4 clock (n = 1)*1<br>1 0: P0φ/16 clock (n = 2)*1<br>1 1: P0φ/64 clock (n = 3)*1 |
| b2 | MP | 0b | R/W*4 | Multi-Processor Mode<br>(Valid only in asynchronous mode)<br>0: Multi-processor communications function is disabled<br>1: Multi-processor communications function is enabled |
| b3 | STOP | 0b | R/W*4 | Stop Bit Length<br>(Valid only in asynchronous mode)<br>0: 1 stop bit<br>1: 2 stop bits |
| b4 | PM | 0b | R/W*4 | Parity Mode<br>(Valid only when the PE bit is 1)<br>0: Selects even parity<br>1: Selects odd parity |
| b5 | PE | 0b | R/W*4 | Parity Enable<br>(Valid only in asynchronous mode)<br>● When transmitting<br>0: Parity bit addition is not performed<br>1: The parity bit is added<br>● When receiving<br>0: Parity bit checking is not performed<br>1: The parity bit is checked |
| b6 | CHR | 0b | R/W*4 | Character Length<br>(Valid only in asynchronous mode*2)<br>Selects in combination with the SCMR.CHR1 bit.<br>CHR1 CHR<br>0 0: Transmit/receive in 9-bit data length<br>0 1: Transmit/receive in 9-bit data length<br>1 0: Transmit/receive in 8-bit data length (initial value)<br>1 1: Transmit/receive in 7-bit data length*3 |
| b7 | CM | 0b | R/W*4 | Communications Mode<br>0: Asynchronous mode<br>1: Clock synchronous mode |

Note 1. n is the decimal notation of the value of n in BRR (refer to **Section 23.2.11, Bit Rate Register (BRR)**).

Note 2. In other than asynchronous mode, this bit setting is invalid and a fixed data length of 8 bits is used.

Note 3. LSB first is fixed and the MSB (bit 7) in TDR is not transmitted in transmission.

Note 4. Writable only when TE in SCR = 0 and RE in SCR = 0 (both serial transmission and reception are disabled).

#### CKS[1:0] Bits (Clock Select)

These bits select the clock source for the on-chip baud rate generator.

For the relation between the settings of these bits and the baud rate, refer to **Section 23.2.11, Bit Rate Register (BRR)**.

#### MP Bit (Multi-Processor Mode)

Disables/enables the multi-processor communications function. The settings of the PE bit and PM bit are invalid in multi-processor mode.

#### STOP Bit (Stop Bit Length)

Selects the stop bit length in transmission.

In reception, only the first stop bit is checked regardless of this bit setting. If the second stop bit is 0, it is treated as the start bit of the next transmit frame.

#### PM Bit (Parity Mode)

Selects the parity mode (even or odd) for transmission and reception. The setting of the PM bit is invalid in multi-processor mode.

#### PE Bit (Parity Enable)

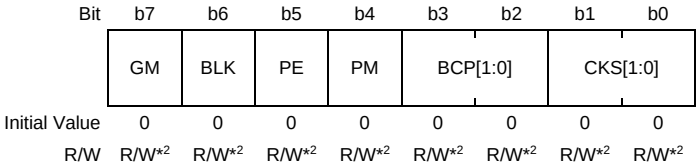
When this bit is set to 1, the parity bit is added to transmit data, and the parity bit is checked in reception. Irrespective of the setting of the PE bit, the parity bit is not added or checked in multi-processor format.

#### CHR Bit (Character Length)

Selects the data length for transmission and reception. Selects in combination with the CHR1 bit in SCMR.

In other than asynchronous mode, a fixed data length of 8 bits is used.

### (2) Smart Card Interface Mode (SCMR.SMIF = 1)




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| b1, b0 | CKS[1:0] | 00b | R/W*2 | Clock Select<br>b1 b0<br>0 0: P0ϕ clock (n = 0)*1<br>0 1: P0ϕ/4 clock (n = 1)*1<br>1 0: P0ϕ/16 clock (n = 2)*1<br>1 1: P0ϕ/64 clock (n = 3)*1 |
| b3, b2 | BCP[1:0] | 00b | R/W*2 | Base Clock Pulse<br>Selects the number of base clock cycles in combination with the SCMR.BCP2 bit.<br>**Table 23.4** lists the combinations of the SCMR.BCP2 bit and SMR.BCP[1:0] bits. |
| b4 | PM | 0b | R/W*2 | Parity Mode<br>(Valid only when the PE bit is 1)<br>0: Selects even parity<br>1: Selects odd parity |
| b5 | PE | 0b | R/W*2 | Parity Enable<br>When this bit is set to 1, a parity bit is added to transmit data, and the parity of received data is checked. Set this bit to 1 in smart card interface mode. |
| b6 | BLK | 0b | R/W*2 | Block Transfer Mode<br>0: Normal mode operation<br>1: Block transfer mode operation |
| b7 | GM | 0b | R/W*2 | GSM Mode<br>0: Normal mode operation<br>1: GSM mode operation |

Note 1. n is the decimal notation of the value of n in BRR (refer to **Section 23.2.11, Bit Rate Register (BRR)**).

Note 2. Writable only when TE in SCR = 0 and RE in SCR = 0 (both serial transmission and reception are disabled).

#### CKS[1:0] Bits (Clock Select)

These bits select the clock source for the on-chip baud rate generator.

For the relationship between the settings of these bits and the baud rate, refer to **Section 23.2.11, Bit Rate Register (BRR)**.

#### BCP[1:0] Bits (Base Clock Pulse)

These bits select the number of base clock cycles in a 1-bit data transfer time in smart card interface mode. Set these bits in combination with the SCMR.BCP2 bit.

For details, refer to **Section 23.6.4, Receive Data Sampling Timing and Reception Margin**.

Table 23.4 Combinations of the SCMR.BCP2 Bit and SMR.BCP[1:0] Bits

| SCMR.BCP2 Bit | SMR.BCP[1:0] Bits | | Number of Base Clock Cycles for 1-Bit Transfer Period |
|---|---|---|---|
| 0 | 0 | 0 | 93 clock cycles (S = 93)*1 |
| 0 | 0 | 1 | 128 clock cycles (S = 128)*1 |
| 0 | 1 | 0 | 186 clock cycles (S = 186)*1 |
| 0 | 1 | 1 | 512 clock cycles (S = 512)*1 |
| 1 | 0 | 0 | 32 clock cycles (S = 32)*1 (Initial Value) |
| 1 | 0 | 1 | 64 clock cycles (S = 64)*1 |
| 1 | 1 | 0 | 372 clock cycles (S = 372)*1 |
| 1 | 1 | 1 | 256 clock cycles (S = 256)*1 |

Note 1. S is the value of S in BRR (refer to **Section 23.2.11, Bit Rate Register (BRR)**).

### PM Bit (Parity Mode)

Selects the parity mode for transmission and reception (even or odd).

For details on the usage of this bit in smart card interface mode, refer to **Section 23.6.2, Data Format (Except in Block Transfer Mode)**.

### PE Bit (Parity Enable)

Set the PE bit to 1.

The parity bit is added to transmit data before transmission, and the parity bit is checked in reception.

### BLK Bit (Block Transfer Mode)

Setting this bit to 1 allows block transfer mode operation.

For details, refer to **Section 23.6.3, Block Transfer Mode**.

### GM Bit (GSM Mode)

Setting this bit to 1 allows GSM mode operation.

In GSM mode, the SSR.TEND flag set timing is put forward to 11.0 etu (elementary time unit = 1-bit transfer time) from the start and the clock output control function is appended. For details, refer to **Section 23.6.6, Serial Data Transmission (Except in Block Transfer Mode)** and **Section 23.6.8, Clock Output Control**.

## 23.2.8 Serial Control Register (SCR)

NOTE

Some bits in SCR have different functions in smart card interface mode and non-smart card interface mode.

### (1) Non-Smart Card Interface Mode (SCMR.SMIF = 0)

| Bit | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | TIE | RIE | TE | RE | MPIE | TEIE | CKE[1:0] | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W*2 | R/W*2 | R/W | R/W | R/W*1 | R/W*1 |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| b1, b0 | CKE[1:0] | 00b | R/W*1 | Clock Enable<br>• For SCI0 and SCI1<br>(Asynchronous mode)<br>b1 b0<br>0 0: On-chip baud rate generator<br>The SCKn pin functions as I/O port.<br>0 1: On-chip baud rate generator<br>The clock with the same frequency as the bit rate is output from the SCKn pin.<br>1 x: External clock<br>The clock with a frequency 16 times the bit rate should be input from the SCKn pin. Input a clock signal with a frequency eight times the bit rate when the SEMR.ABCS bit is 1.<br>(Clock synchronous mode)<br>b1 b0<br>0 x: Internal clock<br>The SCKn pin functions as the clock output pin.<br>1 x: External clock<br>The SCKn pin functions as the clock input pin. |
| b2 | TEIE | 0b | R/W | Transmit End Interrupt Enable<br>0: A TEI interrupt request is disabled<br>1: A TEI interrupt request is enabled |
| b3 | MPIE | 0b | R/W | Multi-Processor Interrupt Enable<br>(Valid in asynchronous mode when SMR.MP = 1)<br>0: Normal reception<br>1: When the data with the multi-processor bit set to 0 is received, the data is not read, and setting the status flags ORER and FER in SSR to 1 is disabled. When the data with the multi-processor bit set to 1 is received, the MPIE bit is automatically cleared to 0, and normal reception is resumed. |
| b4 | RE | 0b | R/W*2 | Receive Enable<br>0: Serial reception is disabled<br>1: Serial reception is enabled |
| b5 | TE | 0b | R/W*2 | Transmit Enable<br>0: Serial transmission is disabled<br>1: Serial transmission is enabled |
| b6 | RIE | 0b | R/W | Receive Interrupt Enable<br>0: RXI and ERI interrupt requests are disabled<br>1: RXI and ERI interrupt requests are enabled |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| b7 | TIE | 0b | R/W | Transmit Interrupt Enable<br>0: A TXI interrupt request is disabled<br>1: A TXI interrupt request is enabled |

**Remarks:** x: Don't care

Note 1. Writable only when TE = 0 and RE = 0.

Note 2. 1 can be written only when TE = 0 and RE = 0, while the SMR.CM bit is 1. After setting TE or RE to 1, only 0 can be written to TE and RE. While the SMR.CM bit is 0 and the SIMR1.IICM bit is 0, writing is enabled under any condition.

### CKE[1:0] Bits (Clock Enable)

These bits select the clock source and SCKn pin function.

The combination of the settings of these bits and of the SEMR.ACS0 bit sets the internal TMR clock.

### TEIE Bit (Transmit End Interrupt Enable)

Enables or disables a TEI interrupt request.

A TEI interrupt request is disabled by setting the TEIE bit to 0.

### MPIE Bit (Multi-Processor Interrupt Enable)

When this bit is set to 1 and the data with the multi-processor bit set to 0 is received, the data is not read and setting the status flags ORER and FER in SSR to 1 is disabled. When the data with the multi-processor bit set to 1 is received, the MPIE is automatically cleared to 0, and normal reception is resumed. For details, refer to **Section 23.4, Multi-Processor Communications Function**.

When the receive data includes the MPB bit is SSR set to 0, the receive data is not transferred from the RSR to the RDR, a receive error is not detected, and setting the flags ORER and FER to 1 is disabled.

When the receive data includes the MPB bit set to 1, the MPB bit is set to 1, the MPIE bit is automatically cleared to 0, the RXI and ERI interrupt requests are enabled (if the RIE bit in SCR is set to 1), and setting the flags ORER and FER to 1 is enabled.

MPIE should be set to 0 if multi-processor communications function is not to be used.

### RE Bit (Receive Enable)

Enables or disables serial reception.

When this bit is set to 1, serial reception is started by detecting the start bit in asynchronous mode or the synchronous clock input in clock synchronous mode. Note that SMR should be set prior to setting the RE bit to 1 in order to designate the reception format.

Even if reception is halted by setting the RE bit to 0, the ORER, FER, and PER flags in SSR are not affected and the previous value is retained.

### TE Bit (Transmit Enable)

Enables or disables serial transmission.

When this bit is set to 1, serial transmission is started by writing transmit data to TDR. Note that SMR should be set prior to setting the TE bit to 1 in order to designate the transmission format.

**RIE Bit (Receive Interrupt Enable)**

Enables or disables RXI and ERI interrupt requests.

An RXI interrupt request is disabled by setting the RIE bit to 0.

An ERI interrupt request can be canceled by reading 1 from the ORER, FER, or PER flag in SSR and then setting the flag to 0, or setting the RIE bit to 0.

### (2) Smart Card Interface Mode (SCMR.SMIF = 1)

| Bit | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | TIE | RIE | TE | RE | MPIE | TEIE | CKE[1:0] | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W*2 | R/W*2 | R/W | R/W | R/W*1 | R/W*1 |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| b1, b0 | CKE[1:0] | 00b | R/W*1 | Clock Enable<br>• When SMR.GM = 0<br>b1 b0<br>0 0: Output disabled (The SCKn pin is available for use as an I/O port according to the I/O port settings.)<br>0 1: Clock output<br>1 x: (Setting prohibited)<br>• When SMR.GM = 1<br>b1 b0<br>0 0: Output fixed low<br>x 1: Clock output<br>1 0: Output fixed high |
| b2 | TEIE | 0b | R/W | Transmit End Interrupt Enable<br>This bit should be 0 in smart card interface mode. |
| b3 | MPIE | 0b | R/W | Multi-Processor Interrupt Enable<br>This bit should be 0 in smart card interface mode. |
| b4 | RE | 0b | R/W*2 | Receive Enable<br>0: Serial reception is disabled<br>1: Serial reception is enabled |
| b5 | TE | 0b | R/W*2 | Transmit Enable<br>0: Serial transmission is disabled<br>1: Serial transmission is enabled |
| b6 | RIE | 0b | R/W | Receive Interrupt Enable<br>0: RXI and ERI interrupt requests are disabled<br>1: RXI and ERI interrupt requests are enabled |
| b7 | TIE | 0b | R/W | Transmit Interrupt Enable<br>0: A TXI interrupt request is disabled<br>1: A TXI interrupt request is enabled |

**Remarks:** x: Don't care

Note 1. Writable only when TE = 0 and RE = 0.

Note 2. 1 can be written only when TE = 0 and RE = 0. After setting TE or RE to 1, only 0 can be written in TE and RE.

For details on interrupt requests, refer to **Section 23.8, Interrupt Sources**.

#### CKE[1:0] Bits (Clock Enable)

These bits control the clock output from the SCKn pin.

In GSM mode, clock output can be dynamically switched. For details, refer to **Section 23.6.8, Clock Output Control**.

#### TEIE Bit (Transmit End Interrupt Enable)

This bit should be 0 in smart card interface mode.

#### MPIE Bit (Multi-Processor Interrupt Enable)

This bit should be 0 in smart card interface mode.

#### RE Bit (Receive Enable)

Enables or disables serial reception.

When this bit is set to 1, serial reception is started by detecting the start bit. Note that SMR should be set prior to setting the RE bit to 1 in order to designate the reception format.

Even if reception is halted by setting the RE bit to 0, the ORER, FER, and PER flags in SSR are not affected and the previous value is retained.

#### TE Bit (Transmit Enable)

Enables or disables serial transmission.

When this bit is set to 1, serial transmission is started by writing transmit data to TDR. Note that SMR should be set prior to setting the TE bit to 1 in order to designate the transmission format.

#### RIE Bit (Receive Interrupt Enable)

Enables or disables RXI and ERI interrupt requests.

An RXI interrupt request is disabled by setting the RIE bit to 0.

An ERI interrupt request can be canceled by reading 1 from the ORER, FER, or PER flag in SSR and then setting the flag to 0, or setting the RIE bit to 0.

## 23.2.9 Serial Status Register (SSR)

Some bits in SSR have different functions in smart card interface mode and non-smart card interface mode.

### (1) Non-Smart Card Interface Mode (SCMR.SMIF = 0)

| Bit | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | TDRE | RDRF | ORER | FER | PER | TEND | MPB | MPBT |
| Initial Value | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| R/W | R/(W)*2 | R/(W)*2 | R/(W)*1 | R/(W)*1 | R/(W)*1 | R | R | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| b0 | MPBT | 0b | R/W | Multi-Processor Bit Transfer<br>Sets the multi-processor bit for adding to the transmission frame<br>0: Data transmission cycles<br>1: ID transmission cycles |
| b1 | MPB | 0b | R | Multi-Processor<br>Value of the multi-processor bit in the reception frame<br>0: Data transmission cycles<br>1: ID transmission cycles |
| b2 | TEND | 1b | R | Transmit End Flag<br>0: A character is being transmitted.<br>1: Character transfer has been completed. |
| b3 | PER | 0b | R/(W) *1 | Parity Error Flag<br>0: No parity error occurred<br>1: A parity error has occurred |
| b4 | FER | 0b | R/(W) *1 | Framing Error Flag<br>0: No framing error occurred<br>1: A framing error has occurred |
| b5 | ORER | 0b | R/(W) *1 | Overrun Error Flag<br>0: No overrun error occurred<br>1: An overrun error has occurred |
| b6 | RDRF | 0b | R/(W) *2 | Receive Data Full Flag<br>0: When data is transferred from RDR<br>1: When data has been received normally, and transferred from RSR to RDR |
| b7 | TDRE | 1b | R/(W) *2 | Transmit Data Empty Flag<br>0: When data is transferred to TDR<br>1: When data is transferred from TDR to TSR |

Note 1. Only 0 can be written to this bit, to clear the flag.

Note 2. Write 1 when writing is necessary.

#### MPB Bit (Multi-Processor)

Holds the value of the multi-processor bit in the reception frame. This bit does not change when the SCR.RE bit is 0.

#### TEND Flag (Transmit End Flag)

Indicates completion of transmission.

[Setting conditions]

- When the SCR.TE bit is set to 0 (serial transmission is disabled)
  When the SCR.TE bit is changed from 0 to 1, the TEND flag is not affected and retains the value 1.
- When the TDR register is not updated at the time of transmission of the tail-end bit of a character being transmitted

[Clearing condition]

- When transmit data are written to the TDR register while the SCR.TE bit is 1

#### PER Flag (Parity Error Flag)

Indicates that a parity error has occurred during reception in asynchronous mode and the reception ends abnormally.

[Setting condition]

- When a parity error is detected during reception
  Although receive data when the parity error occurs is transferred to RDR, no RXI interrupt request occurs. Note that when the PER flag is being set to 1, the subsequent receive data is not transferred to RDR.

[Clearing condition]

- When 0 is written to PER after reading PER = 1
  Even when the SCR.RE bit is set to 0 (serial reception is disabled), the PER flag is not affected and retains its previous value.

#### FER Flag (Framing Error Flag)

Indicates that a framing error has occurred during reception in asynchronous mode and the reception ends abnormally.

[Setting condition]

- When the stop bit is 0
  In 2-stop-bit mode, only the first stop bit is checked whether it is 1 but the second stop bit is not checked. Note that although receive data when the framing error occurs is transferred to RDR, no RXI interrupt request occurs. In addition, when the FER flag is being set to 1, the subsequent receive data is not transferred to RDR.

[Clearing condition]

- When 0 is written to FER after reading FER = 1
  Even when the SCR.RE bit is set to 0, the FER flag is not affected and retains its previous value.

#### ORER Flag (Overrun Error Flag)

Indicates that an overrun error has occurred during reception and the reception ends abnormally.

[Setting condition]

- When the next data is received before receive data is read from RDR
  In RDR, receive data prior to an overrun error occurrence is retained, but data received after the overrun error occurrence is lost. When the ORER flag is set to 1, subsequent serial reception cannot be performed. Note that, in clock synchronous mode, serial transmission also cannot continue.

[Clearing condition]

- When 0 is written to ORER after reading ORER = 1
  Even when the SCR.RE bit is set to 0, the ORER flag is not affected and retains its previous value.

#### RDRF Flag (Receive Data Full Flag)

Indicates whether RDR has received data.

[Setting condition]

- When data has been received normally, and transferred from RSR to RDR

[Clearing condition]

- When data is transferred from RDR

#### TDRE Flag (Transmit Data Empty Flag)

Indicates whether TDR has data to be transmitted.

[Setting condition]

- When data is transferred from TDR to TSR

[Clearing condition]

- When data is transferred to TDR

### (2) Smart Card Interface Mode (SCMR.SMIF = 1)

| Bit | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | TDRE | RDRF | ORER | ERS | PER | TEND | MPB | MPBT |
| Initial Value | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| R/W | R/(W)*2 | R/(W)*2 | R/(W)*1 | R/(W)*1 | R/(W)*1 | R | R | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| b0 | MPBT | 0b | R/W | Multi-Processor Bit Transfer<br>This bit should be set to 0 in smart card interface mode. |
| b1 | MPB | 0b | R | Multi-Processor<br>This bit is not used in smart card interface mode. It should be set to 0. |
| b2 | TEND | 1b | R | Transmit End Flag<br>0: A character is being transmitted.<br>1: Character transfer has been completed. |
| b3 | PER | 0b | R/(W) *1 | Parity Error Flag<br>0: No parity error occurred<br>1: A parity error has occurred |
| b4 | ERS | 0b | R/(W) *1 | Error Signal Status Flag<br>0: Low error signal not responded<br>1: Low error signal responded |
| b5 | ORER | 0b | R/(W) *1 | Overrun Error Flag<br>0: No overrun error occurred<br>1: An overrun error has occurred |
| b6 | RDRF | 0b | R/(W) *2 | Receive Data Full Flag<br>0: When data is transferred from RDR<br>1: When data has been received normally, and transferred from RSR to RDR |
| b7 | TDRE | 1b | R/(W) *2 | Transmit Data Empty Flag<br>0: When data is transferred to TDR<br>1: When data is transferred from TDR to TSR |

Note 1. Only 0 can be written to this bit, to clear the flag.

Note 2. Write 1 when writing is necessary.

### TEND Flag (Transmit End Flag)

With no error signal from the receiving side, this bit is set to 1 when further data for transfer is ready to be transferred to the TDR register.

[Setting conditions]

- When the SCR.TE bit = 0 (serial transmission is disabled)
  When the SCR.TE bit is changed from 0 to 1, the TEND flag is not affected and retains the value 1.
- When a specified period has elapsed after the latest transmission of 1 byte, the ERS flag is 0, and the TDR register is not updated
  The set timing is determined by register settings as listed below.
  When SMR.GM = 0 and SMR.BLK = 0, 12.5 etu after the start of transmission
  When SMR.GM = 0 and SMR.BLK = 1, 11.5 etu after the start of transmission
  When SMR.GM = 1 and SMR.BLK = 0, 11.0 etu after the start of transmission
  When SMR.GM = 1 and SMR.BLK = 1, 11.0 etu after the start of transmission

[Clearing condition]

- When transmit data are written to the TDR register while the SCR.TE bit is 1

### PER Flag (Parity Error Flag)

Indicates that a parity error has occurred during reception in asynchronous mode and the reception ends abnormally.

[Setting condition]

- When a parity error is detected during reception
  Although receive data when the parity error occurs is transferred to RDR, no RXI interrupt request occurs. Note that when the PER flag is being set to 1, the subsequent receive data is not transferred to RDR.

[Clearing condition]

- When 0 is written to PER after reading PER = 1
  Even when the SCR.RE bit is set to 0 (serial reception is disabled), the PER flag is not affected and retains its previous value.

### ERS Flag (Error Signal Status Flag)

[Setting condition]

- When a low error signal is sampled

[Clearing condition]

- When 0 is written to ERS after reading ERS = 1
  Even when the SCR.RE bit is set to 0, the ERS flag is not affected and retains its previous value.

#### ORER Flag (Overrun Error Flag)

Indicates that an overrun error has occurred during reception and the reception ends abnormally.

[Setting condition]

- When the next data is received before receive data is read from RDR
  In RDR, the receive data prior to an overrun error occurrence is retained, but data received following the overrun error occurrence is lost. When the ORER flag is set to 1, subsequent serial reception cannot be performed.

[Clearing condition]

- When 0 is written to ORER after reading ORER = 1
  Even when the SCR.RE bit is set to 0, the ORER flag is not affected and retains its previous value.

#### RDRF Flag (Receive Data Full Flag)

Indicates whether RDR has received data.

[Setting condition]

- When data has been received normally, and transferred from RSR to RDR

[Clearing condition]

- When data is transferred from RDR

#### TDRE Flag (Transmit Data Empty Flag)

Indicates whether TDR has data to be transmitted.

[Setting condition]

- When data is transferred from TDR to TSR

[Clearing condition]

- When data is transferred to TDR

## 23.2.10 Smart Card Mode Register (SCMR)

| Bit | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | BCP2 | — | — | CHR1 | SDIR | SINV | — | SMIF |
| Initial Value | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 0 |
| R/W | R/W*1 | R/W | R/W | R/W*1 | R/W*1 | R/W*1 | R/W | R/W*1 |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| b0 | SMIF | 0b | R/W*1 | Smart Card Interface Mode Select<br>0: Non-smart card interface mode (Asynchronous mode, clock synchronous mode,)<br>1: Smart card interface mode |
| b1 | — | 1b | R/W | Reserved<br>This bit is read as 1. The write value should be 1. |
| b2 | SINV | 0b | R/W*1 | Transmitted/Received Data Invert<br>0: TDR contents are transmitted as they are. Receive data is stored as it is in RDR.<br>1: TDR contents are inverted before being transmitted. Receive data is stored in inverted form in RDR. |
| b3 | SDIR | 0b | R/W*1 | Transmitted/Received Data Transfer Direction<br>This bit can be used in the following modes.<br>• Smart card interface mode<br>• Asynchronous mode (multi-processor mode)<br>• Clock synchronous mode<br>0: Transfer with LSB first<br>1: Transfer with MSB first |
| b4 | CHR1 | 1b | R/W*1 | Character Length 1<br>(Only valid in asynchronous mode)*2<br>Selects in combination with the SMR.CHR bit.<br>CHR1 CHR<br>0 0: Transmit/receive in 9-bit data length<br>0 1: Transmit/receive in 9-bit data length<br>1 0: Transmit/receive in 8-bit data length (initial value)<br>1 1: Transmit/receive in 7-bit data length*3 |
| b6, b5 | — | 11b | R/W | Reserved<br>These bits are read as 1. The write value should be 1. |
| b7 | BCP2 | 1b | R/W*1 | Base Clock Pulse 2<br>Selects the number of base clock cycles in combination with the SMR.BCP[1:0] bits.<br>**Table 23.5** lists the combinations of the SCMR.BCP2 bit and SMR.BCP[1:0] bits. |

Note 1. Writable only when TE in SCR = 0 and RE in SCR = 0 (both serial transmission and reception are disabled).

Note 2. The setting is invalid and a fixed data length of 8 bits is used in modes other than asynchronous mode.

Note 3. LSB first should be selected and the value of MSB (b7) in TDR cannot be transmitted.

### SMIF Bit (Smart Card Interface Mode Select)

When this bit is set to 1, smart card interface mode is selected.

When this bit is set to 0, non-smart card interface mode, i.e., asynchronous mode (including multi-processor mode), clock synchronous mode is selected.

### SINV Bit (Transmitted/Received Data Invert)

Inverts the transmit/receive data logic level. This bit does not affect the logic level of the parity bit. To invert the parity bit, invert the PM bit in SMR.

### CHR1 bit (Character Length 1)

Selects the data length of transmit/receive data. Selects in combination with the CHR bit in SMR.

A fixed data length of 8 bits is used in modes other than asynchronous mode.

### BCP2 Bit (Base Clock Pulse 2)

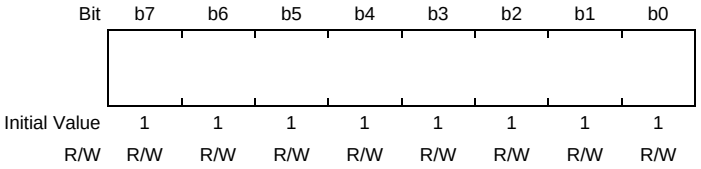
Selects the number of base clock cycles in a 1-bit data transfer time in smart card interface mode. Set this bit in combination with the SMR.BCP[1:0] bits.

Table 23.5 Combinations of the SCMR.BCP2 Bit and SMR.BCP[1:0] Bits

| SCMR.BCP2 Bit | SMR.BCP[1:0] Bits | | Number of Base Clock Cycles for 1-Bit Transfer Period |
|---|---|---|---|
| 0 | 0 | 0 | 93 clock cycles (S = 93)*1 |
| 0 | 0 | 1 | 128 clock cycles (S = 128)*1 |
| 0 | 1 | 0 | 186 clock cycles (S = 186)*1 |
| 0 | 1 | 1 | 512 clock cycles (S = 512)*1 |
| 1 | 0 | 0 | 32 clock cycles (S = 32)*1 (Initial Value) |
| 1 | 0 | 1 | 64 clock cycles (S = 64)*1 |
| 1 | 1 | 0 | 372 clock cycles (S = 372)*1 |
| 1 | 1 | 1 | 256 clock cycles (S = 256)*1 |

Note 1. S is the value of S in BRR (refer to **Section 23.2.11, Bit Rate Register (BRR)**).

### 23.2.11 Bit Rate Register (BRR)




BRR is an 8-bit register that adjusts the bit rate.

As each SCI channel has independent baud rate generator control, different bit rates can be set for each. **Table 23.6** shows the relationship between the setting (N) in the BRR and the bit rate (B) for normal asynchronous mode, multi-processor communication, clock synchronous mode, and smart card interface mode.

The initial value of BRR is H'FF.

BRR can be read from by the CPU, but it can be written to only when the TE and RE bits in SCR are 0.

Table 23.6 Relationship between N Setting in BRR and Bit Rate B

| Mode | SEMR BGDM Bit | SEMR ABCS Bit | BRR Setting | Error |
|---|---|---|---|---|
| Asynchronous or multi-processor communication | 0 | 0 | $N = \frac{P0\phi \times 10^6}{64 \times 2^{2n-1} \times B} - 1$ | $Error\ (\%) = \{\frac{P0\phi \times 10^6}{B \times 64 \times 2^{2n-1} \times (N+1)} - 1\} \times 100$ |
| | 0 | 1 | $N = \frac{P0\phi \times 10^6}{32 \times 2^{2n-1} \times B} - 1$ | $Error\ (\%) = \{\frac{P0\phi \times 10^6}{B \times 32 \times 2^{2n-1} \times (N+1)} - 1\} \times 100$ |
| | 1 | 0 | | |
| | 1 | 1 | $N = \frac{P0\phi \times 10^6}{16 \times 2^{2n-1} \times B} - 1$ | $Error\ (\%) = \{\frac{P0\phi \times 10^6}{B \times 16 \times 2^{2n-1} \times (N+1)} - 1\} \times 100$ |
| Clock synchronous | | | $N = \frac{P0\phi \times 10^6}{8 \times 2^{2n-1} \times B} - 1$ | |
| Smart card interface | | | $N = \frac{P0\phi \times 10^6}{S \times 2^{2n+1} \times B} - 1$ | $Error\ (\%) = \{\frac{P0\phi \times 10^6}{B \times S \times 2^{2n+1} \times (N+1)} - 1\} \times 100$ |

**Remarks:** B: Bit rate (bps)
N: BRR setting for on-chip baud rate generator (0 ≤ N ≤ 255)
P0ϕ: Operating frequency (MHz)
n and S: Determined by the settings of the SMR and SCMR registers as listed in the table below.

Table 23.7 Clock Source Settings

| SMR.CKS[1:0] Bit Setting | Clock Source | n |
|---|---|---|
| 0 0 | P0$\phi$ clock | 0 |
| 0 1 | P0$\phi$/4 clock | 1 |
| 1 0 | P0$\phi$/16 clock | 2 |
| 1 1 | P0$\phi$/64 clock | 3 |

Table 23.8 Base Clock Settings in Smart Card Interface Mode

| SCMR.BCP2 Bit Setting | SMR.BCP[1:0] Bit Setting | Base Clock Cycles for 1-bit Period | S |
|---|---|---|---|
| 0 | 0 0 | 93 clock cycles | 93 |
| 0 | 0 1 | 128 clock cycles | 128 |
| 0 | 1 0 | 186 clock cycles | 186 |
| 0 | 1 1 | 512 clock cycles | 512 |
| 1 | 0 0 | 32 clock cycles | 32 |
| 1 | 0 1 | 64 clock cycles | 64 |
| 1 | 1 0 | 372 clock cycles | 372 |
| 1 | 1 1 | 256 clock cycles | 256 |

**Table 23.9** lists examples of N settings in BRR in normal asynchronous mode. **Table 23.10** lists the maximum bit rate settable for each operating frequency. Examples of BRR (N) settings in clock synchronous mode are listed in **Table 23.12**. Examples of BRR (N) settings in smart card interface mode are listed in **Table 23.14**. In smart card interface mode, the number of base clock cycles S in a 1-bit data transfer time can be selected. For details, refer to **Section 23.6.4, Receive Data Sampling Timing and Reception Margin**. **Table 23.11** and **Table 23.13** list the maximum bit rates with external clock input.

When either the asynchronous mode base clock select bit (ABCS) or the baud rate generator double-speed mode select bit (BGDM) in the serial extended mode register (SEMR) is set to 1 in asynchronous mode, the bit rate becomes twice that listed in **Table 23.9**. When both of those registers are set to 1, the bit rate becomes four times the listed value.

Table 23.9 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode)

| | Operating Frequency P0$\phi$ (MHz) | | |
|---|---|---|---|
| | 100 | | |
| Bit Rate (bps) | n | N | Error (%) |
| 110 | | | |
| 150 | | | |
| 300 | 3 | 162 | -0.15 |
| 600 | 3 | 80 | 0.47 |
| 1200 | 2 | 162 | -0.15 |
| 2400 | 2 | 80 | 0.47 |
| 4800 | 1 | 162 | -0.15 |
| 9600 | 1 | 80 | 0.47 |
| 19200 | 0 | 162 | -0.15 |
| 31250 | 0 | 99 | 0.00 |
| 38400 | 0 | 80 | 0.47 |

**Note:** This is an example when the ABCS and BGDM bits in SEMR are 0.
When either the ABCS bit or BGDM bit is set to 1, the bit rate doubles.
When both ABCS and BGDM bits in SEMR are set to 1, the bit rate increases four times.

Table 23.10 Maximum Bit Rate for Each Operating Frequency (Asynchronous Mode)

| | SEMR Settings | | | | |
|---|---|---|---|---|---|
| P0ϕ (MHz) | BGDM Bit | ABCS Bit | n | N | Maximum Bit Rate (bps) |
| 100 | 0 | 0 | 0 | 0 | 3125000 |
| | | 1 | 0 | 0 | 6250000 |
| | 1 | 0 | 0 | 0 | 6250000 |
| | | 1 | 0 | 0 | 12500000 |

Table 23.11 Maximum Bit Rate with External Clock Input (Asynchronous Mode)

| | Maximum Bit Rate (bps) | | |
|---|---|---|---|
| P0ϕ (MHz) | External Input Clock (MHz) | SEMR.ABCS Bit = 0 | SEMR.ABCS Bit = 1 |
| 100 | 25.000 | 1562500 | 3125000 |

Table 23.12 BRR Settings for Various Bit Rates (Clock Synchronous Mode)

| | Operating Frequency P0ϕ (MHz) | |
|---|---|---|
| | 100 | |
| Bit Rate (bps) | n | N |
| 110 | — | — |
| 250 | — | — |
| 500 | — | — |
| 1 k | — | — |
| 2.5 k | 3 | 155 |
| 5 k | 3 | 77 |
| 10 k | 2 | 155 |
| 25 k | 1 | 249 |
| 50 k | 1 | 124 |
| 100 k | 0 | 249 |
| 250 k | 0 | 99 |
| 500 k | 0 | 49 |
| 1 M | 0 | 24 |
| 2.5 M | 0 | 9 |
| 5 M | 0 | 4 |
| 7.5 M | 0 | 2 |

**Note:** Continuous transmission or reception is impossible. After transmitting/receiving one frame of data, there is an interval of a 1-bit period before starting transmitting/receiving the next frame of data. The output of the synchronization clock is stopped for a 1-bit period. For this reason, it takes 9 bits worth of time to transfer one frame (8 bits) of data, and the average transfer rate is 8/9 times the bit rate.

**Remarks:** Space: Setting prohibited.
—: Can be set, but an error will occur.

Table 23.13 Maximum Bit Rate with External Clock Input (Clock Synchronous Mode)

| P0φ (MHz) | External Input Clock (MHz) | Maximum Bit Rate (Mbps) |
|---|---|---|
| 100 | 16.6667 | 16.6667 |

Table 23.14 BRR Settings for Various Bit Rates (Smart Card Interface Mode, n = 0, S = 372)

| Bit Rate (bps) | P0φ (MHz) | n | N | Error (%) |
|---|---|---|---|---|
| 9600 | 100 | 0 | 13 | 0.01 |

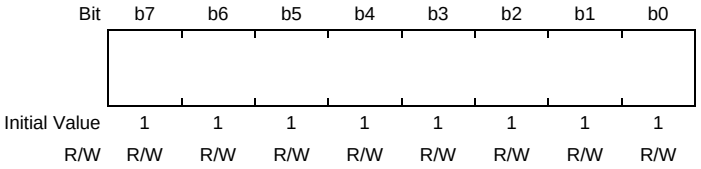
Table 23.15 Maximum Bit Rate for Each Operating Frequency (Smart Card Interface Mode, S = 32)

| P0φ (MHz) | Maximum Bit Rate (bps) | n | N |
|---|---|---|---|
| 100.00 | 1562500 | 0 | 0 |

Table 23.16 Maximum Bit Rate for Each Operating Frequency (Smart Card Interface Mode, S = 372)

| P0φ (MHz) | Maximum Bit Rate (bps) | n | N |
|---|---|---|---|
| 100.00 | 134409 | 0 | 0 |

### 23.2.12 Modulation Duty Register (MDDR)




MDDR corrects the bit rate adjusted by the BRR register.

When the BRME bit in SEMR is set to 1, the bit rate generated by the on-chip baud rate generator is evenly corrected according to the settings of MDDR (M/256). The relationship between the MDDR setting (M) and the bit rate (B) is given in **Table 23.17**.

The initial value of MDDR is H'FF. Bit 7 in this register is fixed to 1.

The CPU can read the MDDR register, but this register is only writable when the TE and RE bits in SCR are 0.

Table 23.17 Relationship between MDDR Setting (M) and Bit Rate (B)
When Bit Rate Modulation Function is Used

| Mode | SEMR Setting BGDM Bit | SEMR Setting ABCS Bit | BRR Setting | Error |
|---|---|---|---|---|
| Asynchronous or multi-processor communication | 0 | 0 | $N = \frac{P0\phi \times 10^6}{64 \times 2^{2n-1} \times (256/M) \times B} - 1$ | $Error\ (\%) = \{\frac{P0\phi \times 10^6}{B \times 64 \times 2^{2n-1} \times (256M) \times (N+1)} - 1\} \times 100$ |
| | 0 | 1 | $N = \frac{P0\phi \times 10^6}{32 \times 2^{2n-1} \times (256/M) \times B} - 1$ | $Error\ (\%) = \{\frac{P0\phi \times 10^6}{B \times 32 \times 2^{2n-1} \times (256M) \times (N+1)} - 1\} \times 100$ |
| | 1 | 0 | | |
| | 1 | 1 | $N = \frac{P0\phi \times 10^6}{16 \times 2^{2n-1} \times (256/M) \times B} - 1$ | $Error\ (\%) = \{\frac{P0\phi \times 10^6}{B \times 16 \times 2^{2n-1} \times (256M) \times (N+1)} - 1\} \times 100$ |
| Clock synchronous | | | $N = \frac{P0\phi \times 10^6}{8 \times 2^{2n-1} \times (256/M) \times B} - 1$ | |
| Smart card interface | | | $N = \frac{P0\phi \times 10^6}{S \times 2^{2n-1} \times (256/M) \times B} - 1$ | $Error\ (\%) = \{\frac{P0\phi \times 10^6}{B \times S \times 2^{2n-1} \times (256M) \times (N+1)} - 1\} \times 100$ |

**Note:** Do not use this function in clock synchronous mode with the highest speed setting (SMR.CKS[1:0] = 00b, SCR.CKE[1] = 0, and BRR = 0).

**Remarks:** B: Bit rate (bps)
M: MDDR setting (128 ≤ MDDR ≤ 256)
N: BRR setting for baud rate generator (0 ≤ N ≤ 255)
P0ϕ: Operating frequency (MHz)
n and S: Determined by the settings of the SMR and SCMR registers as listed in **Table 23.7** and **Table 23.8**, **Section 23.2.11, Bit Rate Register (BRR)**.

## 23.2.13 Serial Extended Mode Register (SEMR)

| Bit | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | RXDESEL | BGDM | NFEN | ABCS | — | BRME | — | ACS0 |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W*1 | R/W | R/W*1 | R/W*1 | R/W | R/W | R/W | R/W*1 |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| b0 | ACS0 | 0b | R/W*1 | Asynchronous Mode Clock Source Select<br>(Valid only in asynchronous mode)<br>0: External clock input<br>1: Reserved bit |
| b1 | — | 0b | R/W | Reserved<br>This bit is read as 0. The write value should be 0. |
| b2 | BRME | 0b | R/W | Bit Rate Modulation Enable<br>0: Bit rate modulation function is disabled.<br>1: Bit rate modulation function is enabled. |
| b3 | — | 0b | R/W | Reserved<br>This bit is read as 0. The write value should be 0. |
| b4 | ABCS | 0b | R/W*1 | Asynchronous Mode Base Clock Select<br>(Valid only in asynchronous mode)<br>0: Selects 16 base clock cycles for 1-bit period.<br>1: Selects 8 base clock cycles for 1-bit period. |
| b5 | NFEN | 0b | R/W*1 | Digital Noise Filter Function Enable<br>(In asynchronous mode)<br>0: Noise cancellation function for the RXDn input signal is disabled.<br>1: Noise cancellation function for the RXDn input signal is enabled. |
| b6 | BGDM | 0b | R/W | Baud Rate Generator Double-Speed Mode Select<br>(Only valid the CKE[1] bit in SCR is 0 in asynchronous mode).<br>0: Baud rate generator outputs the clock with normal frequency.<br>1: Baud rate generator outputs the clock with doubled frequency. |
| b7 | RXDESEL | 0b | R/W*1 | Asynchronous Start Bit Edge Detection Select<br>(Valid only in asynchronous mode)<br>0: The low level on the RXDn pin is detected as the start bit.<br>1: A falling edge on the RXDn pin is detected as the start bit. |

Note 1. Writable only when TE in SCR = 0 and RE in SCR = 0 (both serial transmission and reception are disabled).

SEMR selects the clock source for 1-bit period in asynchronous mode.

### ACS0 Bit (Asynchronous Mode Clock Source Select)

Selects the clock source in the asynchronous mode.

The ACS0 bit is valid in asynchronous mode (CM bit in SMR = 0) and when an external clock input is selected (CKE[1:0] bits in SCR = 10b or 11b).

Set the ACS0 bit to 0 in other than asynchronous mode.

### BRME bit (Bit Rate Modulation Enable)

Enables and disables the bit rate modulation function. The bit rate generated by on-chip baud rate generator is evenly corrected when this function is enabled.

#### NFEN Bit (Digital Noise Filter Function Enable)

This bit enables or disables the digital noise filter function.

When the function is enabled, noise cancellation is applied to the RXDn input signal in asynchronous mode. In any mode other than above, set the NFEN bit to 0 to disable the digital noise filter function.

When the function is disabled, input signals are transferred as is, as internal signals.

#### BGDM bit (Baud Rate Generator Double-Speed Mode Select)

Selects the cycle of output clock for the baud rate generator.

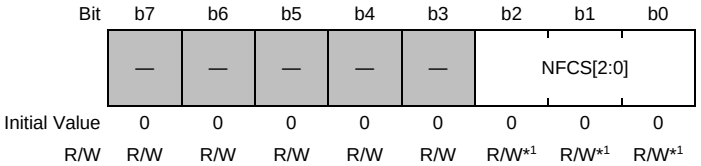
This bit is valid when the on-chip baud rate generator is selected as the clock source (SCR.CKE[1] = 0) in asynchronous mode (SMR.CM = 0). For the clock output from the baud rate generator, either normal or doubled frequency can be selected. The base clock is generated by the clock output from the baud rate generator. When the BGDM bit is set to 1, the base clock cycle is halved and the bit rate is doubled. Set this bit to 0 in modes other than asynchronous mode.

#### RXDESEL Bit (Asynchronous Start Bit Edge Detection Select)

Selects the detection method of the start bit for reception in asynchronous mode. When a break occurs, data receiving operation depends on the settings of this bit. Set this bit to 1 when reception should be stopped while a break occurs or when reception should be started without retaining the RXDn pin input at high level for the period of one data frame or longer after completion of the break.

Set this bit to 0 in modes other than asynchronous mode.

## 23.2.14 Noise Filter Setting Register (SNFR)




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| b2 to b0 | NFCS[2:0] | All 0 | R/W*1 | Noise Filter Clock Select<br>In asynchronous mode, the standard setting for the base clock is as follows.<br>b2 b0<br>0 0 0: The clock signal divided by 1 is used with the noise filter.<br>Settings other than above are prohibited. |
| b7 to b3 | — | All 0 | R/W | Reserved<br>These bits are read as 0. The write value should be 0. |

Note 1. Writing to these bits is only possible when the RE and TE bits in the SCR are 0 (serial reception and transmission disabled).

### NFCS[2:0] Bits (Noise Filter Clock Select)

These bits select the sampling clock for the digital noise filter. To use the noise filter in asynchronous mode, set these bits to 000b.

## 23.2.15 Extended function control register (SECR)

| Bit | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | CTSE | — |
| 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*1 | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| b0 | — | 0b | R/W | Reserved<br>This bit is read as 0. The write value should be 0. |
| b1 | CTSE | 0b | R/W*1 | CTS Enable<br>0: CTS function is disabled (RTS output function is enabled).<br>1: CTS function is enabled. |
| b2 to b7 | — | All 0 | R/W | Reserved<br>These bits are read as 0. The write value should be 0. |

Note 1. Writing to the bit is only possible when the SCR.RE and SCR.TE bits are 0.

SECR is used to select the extension settings in asynchronous and clock synchronous modes.

### CTSE Bit (CTS Enable)

Set this bit to 1 if the SSn# pin is to be used for inputting of the CTS control signal to control of transmission and reception. The RTS signal is output when this bit is set to 0. Set this bit to 0 in smart card interface mode. Do not set both the CTSE and SSE bits to enabled (even if this setting is made, operation is the same as that when these bits are set to 0).

## 23.3 Operation in Asynchronous Mode

**Figure 23.2** shows the general format for asynchronous serial communications.

One frame consists of a start bit (low level), transmit/receive data, a parity bit, and stop bits (high level).

In asynchronous serial communications, the communications line is usually held in the mark state (high level).

The SCI monitors the communications line. When the SCI detects a low, it regards that as a start bit and starts serial communication.

Inside the SCI, the transmitter and receiver are independent units, enabling full-duplex communications. Both the transmitter and the receiver also have a double-buffered structure, so that data can be read or written during transmission or reception, enabling continuous data transmission and reception.

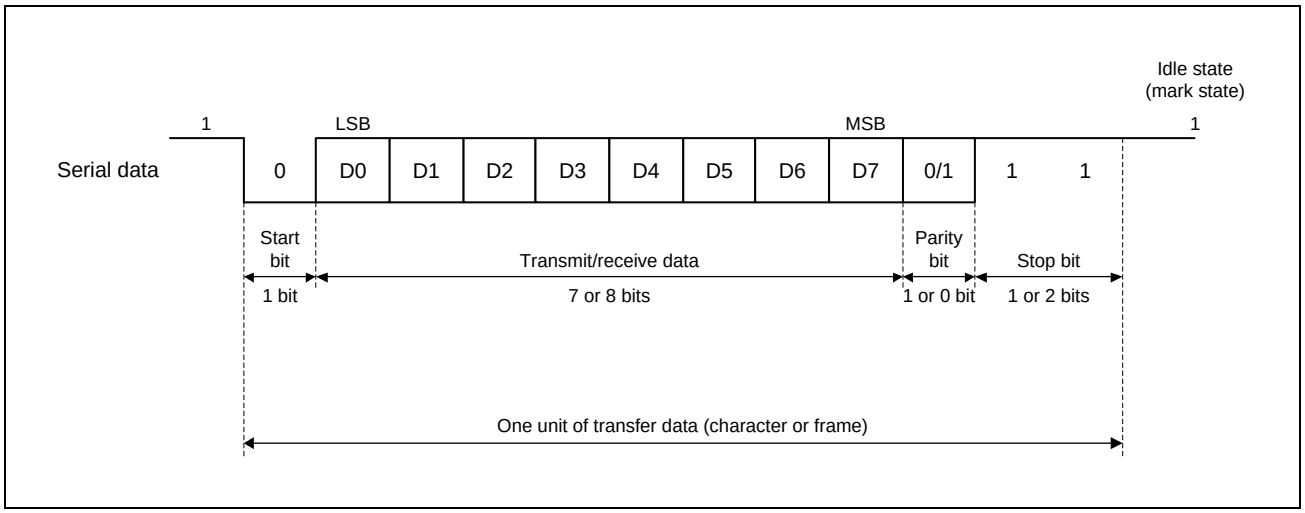



Figure 23.2 Data Format in Asynchronous Serial Communications (Example with 8-Bit Data, Parity, 2 Stop Bits)

### 23.3.1 Serial Data Transfer Format

**Table 23.18** lists the serial data transfer formats that can be used in asynchronous mode.

Any of 18 transfer formats can be selected according to the SMR and SCMR setting. For details of multi-processor function, refer to **Section 23.4, Multi-Processor Communications Function**.

Table 23.18 Serial Transfer Formats (Asynchronous Mode)

| SCMR Setting | SMR Setting | | | | Serial Transfer Format and Frame Length | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| CHR1 | CHR | PE | MP | STOP | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
| 0 | 0 | 0 | 0 | 0 | S | 9-bit data | | | | | | | | | STOP | | |
| 0 | 0 | 0 | 0 | 1 | S | 9-bit data | | | | | | | | | STOP | STOP | |
| 0 | 0 | 1 | 0 | 0 | S | 9-bit data | | | | | | | | | P | STOP | |
| 0 | 0 | 1 | 0 | 1 | S | 9-bit data | | | | | | | | | P | STOP | STOP |
| 1 | 0 | 0 | 0 | 0 | S | 8-bit data | | | | | | | | STOP | | | |
| 1 | 0 | 0 | 0 | 1 | S | 8-bit data | | | | | | | | STOP | STOP | | |
| 1 | 0 | 1 | 0 | 0 | S | 8-bit data | | | | | | | | P | STOP | | |
| 1 | 0 | 1 | 0 | 1 | S | 8-bit data | | | | | | | | P | STOP | STOP | |
| 1 | 1 | 0 | 0 | 0 | S | 7-bit data | | | | | | | STOP | | | | |
| 1 | 1 | 0 | 0 | 1 | S | 7-bit data | | | | | | | STOP | STOP | | | |
| 1 | 1 | 1 | 0 | 0 | S | 7-bit data | | | | | | | P | STOP | | | |
| 1 | 1 | 1 | 0 | 1 | S | 7-bit data | | | | | | | P | STOP | STOP | | |
| 0 | 0 | — | 1 | 0 | S | 9-bit data | | | | | | | | | MPB | STOP | |
| 0 | 0 | — | 1 | 1 | S | 9-bit data | | | | | | | | | MPB | STOP | STOP |
| 1 | 0 | — | 1 | 0 | S | 8-bit data | | | | | | | | MPB | STOP | | |
| 1 | 0 | — | 1 | 1 | S | 8-bit data | | | | | | | | MPB | STOP | STOP | |
| 1 | 1 | — | 1 | 0 | S | 7-bit data | | | | | | | MPB | STOP | | | |
| 1 | 1 | — | 1 | 1 | S | 7-bit data | | | | | | | MPB | STOP | STOP | | |

**Note:** S: Start bit
STOP: Stop bit
P: Parity bit
MPB: Multi-processor bit

### 23.3.2 Receive Data Sampling Timing and Reception Margin in Asynchronous Mode

In asynchronous mode, the SCI operates on a base clock with a frequency of 16 times*1 the bit rate.

In reception, the SCI samples the falling edge of the start bit using the base clock, and performs internal synchronization. Since receive data is sampled at the rising edge of the 8th pulse*1 of the base clock, data is latched at the middle of each bit, as shown in **Figure 23.3**. Thus the reception margin in asynchronous mode is determined by formula (1) below.

$$M = \left|(0.5 - \frac{1}{2N}) - (L - 0.5)F - \frac{|D - 0.5|}{N}(1 + F)\right| \times 100\ [\%] \cdots \text{Formula (1)}$$

M: Reception margin
N: Ratio of bit rate to clock (N = 16 when ABCS in SEMR = 0, N = 8 when ABCS in SEMR = 1)
D: Duty cycle of clock (D = 0.5 to 1.0)
L: Frame length (L = 9 to 13)
F: Absolute value of clock frequency deviation

Assuming values of F = 0 and D = 0.5 in formula (1), the reception margin is determined by the formula below.

$$M = \{0.5 - 1/(2 \times 16)\} \times 100\ (\%) = 46.875\%$$

However, this is only the computed value, and a margin of 20% to 30% should be allowed in system design.

**Note 1.** This is an example when the ABCS bit in SEMR is 0. When the ABCS bit is 1, a frequency of 8 times the bit rate is used as a base clock and receive data is sampled at the rising edge of the 4th pulse of the base clock.

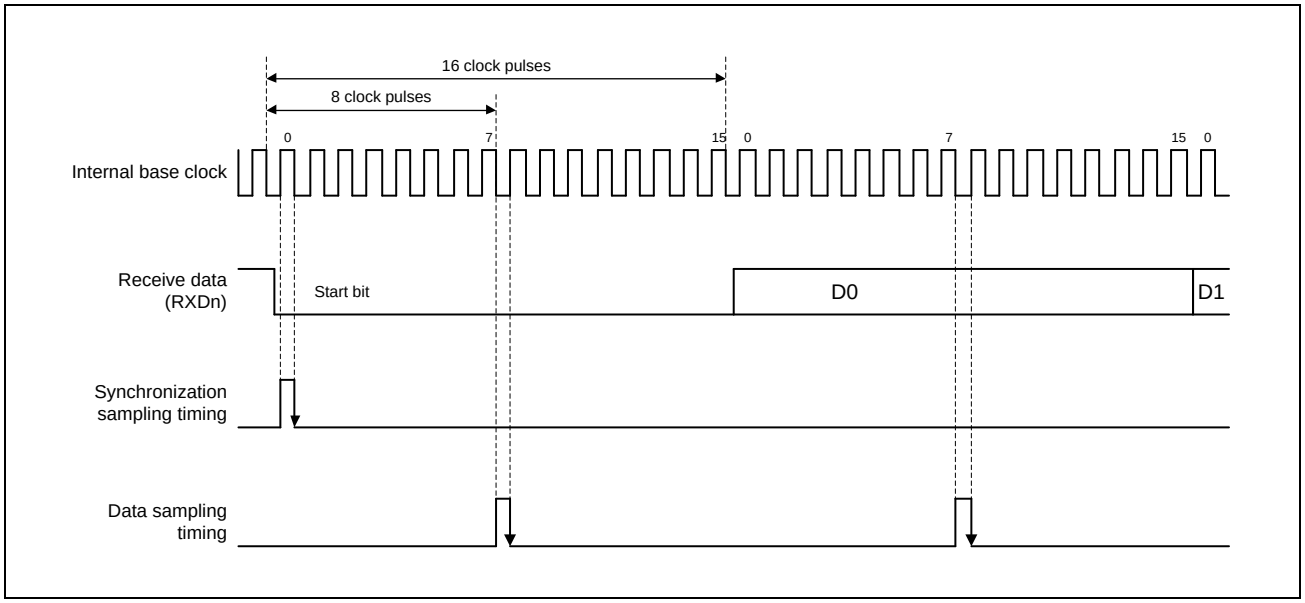



Figure 23.3 Receive Data Sampling Timing in Asynchronous Mode

### 23.3.3 Clock

Either an internal clock generated by the on-chip baud rate generator or an external clock input to the SCKn pin can be selected as the SCI's transfer clock, according to the setting of the CM bit in SMR and the CKE[1:0] bits in SCR. When an external clock is input to the SCKn pin, the clock frequency should be 16 times the bit rate (when ABCS in SEMR = 0) and 8 times the bit rate (when ABCS in SEMR = 1).

When the SCI is operated on an internal clock, the clock can be output from the SCKn pin. The frequency of the clock output in this case is equal to the bit rate, and the phase is such that the rising edge of the clock is in the middle of the transmit data, as shown in **Figure 23.4**.

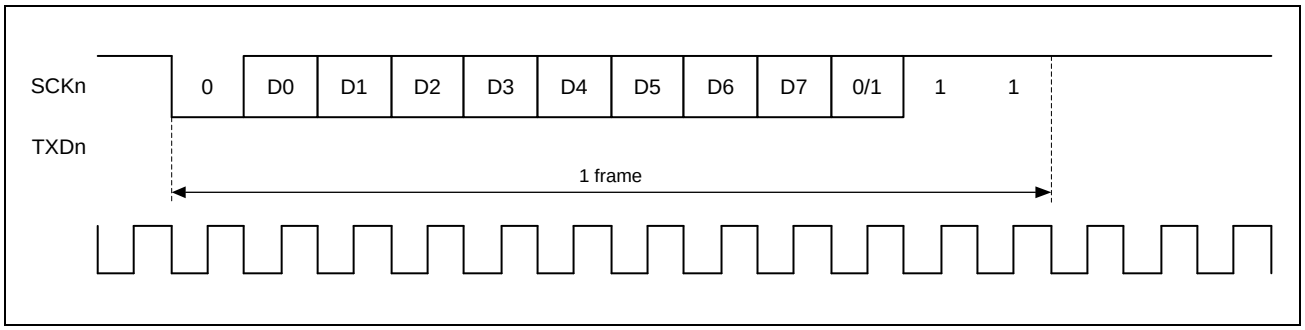



Figure 23.4 Phase Relationship between Output Clock and Transmit Data
(Asynchronous Mode: SMR.CHR = 0, PE = 1, MP = 0, STOP = 1)

### 23.3.4 Double-Speed Mode

The output clock frequency of the on-chip baud rate generator is doubled by setting the SEMR.BGDM bit to 1, enabling high-speed communication at a doubled bit rate. If the SEMR.ABCS bit is set to 1 under the above condition, the number of base clock cycles changes from 16 to 8, so the bit rate becomes four times faster than the initial state.

As shown by Formula (1) in **Section 23.3.2, Receive Data Sampling Timing and Reception Margin in Asynchronous Mode**, setting the SEMR.ABCS bit to 1 changes the number of cycles to 8, and the sampling interval becomes longer. This causes the reception margin to decrease. Therefore, setting the SEMR.BGDM bit to 1 and the SEMR.ABCS bit to 0 is recommended instead of setting the SEMR.BGDM bit to 0 and the SEMR.ABCS bit to 1 for high-speed operation at a doubled bit rate.

### 23.3.5 CTS and RTS Functions

The CTS function is the use of input on the CTSn# pin in transmission control. Setting the SECR.CTSE bit to 1 enables the CTS function. When the CTS function is enabled, placing the low level on the CTSn# pin causes transmission to start.

Applying the high level to the CTS# pin while transmission is in progress does not affect transmission of the current frame, which continues.

In the RTS function, by using the function of output on the RTSn# pin, a low level is output when reception becomes possible. Conditions for output of the low and high level are shown below.

[Conditions for low-level output]

Satisfaction of all conditions listed below

- The value of the RE bit in the SCR is 1
- Reception is not in progress
- There are no received data yet to be read
- The ORER, FER, and PER flags in the SSR are all 0

[Condition for high-level output]

- The conditions for low-level output have not been satisfied.

### 23.3.6 SCI Initialization (Asynchronous Mode)

Before transmitting and receiving data, start by writing the initial value H'00 to SCR and then continue through the procedure for SCI given in **Figure 23.5**. Whenever the operating mode or transfer format is changed, SCR must be initialized before the change is made.

When the external clock is used in asynchronous mode, ensure that the clock signal is supplied even during initialization. Note that setting the SCR.RE bit to 0 initializes neither the ORER, FER, and PER flags in SSR nor RDR, RDRH, and RDRL.

Moreover, note that switching the value of the SCR.TE bit from 1 to 0 or 0 to 1 while the SCR.TIE bit is 1 leads to the generation of a TXI interrupt request.

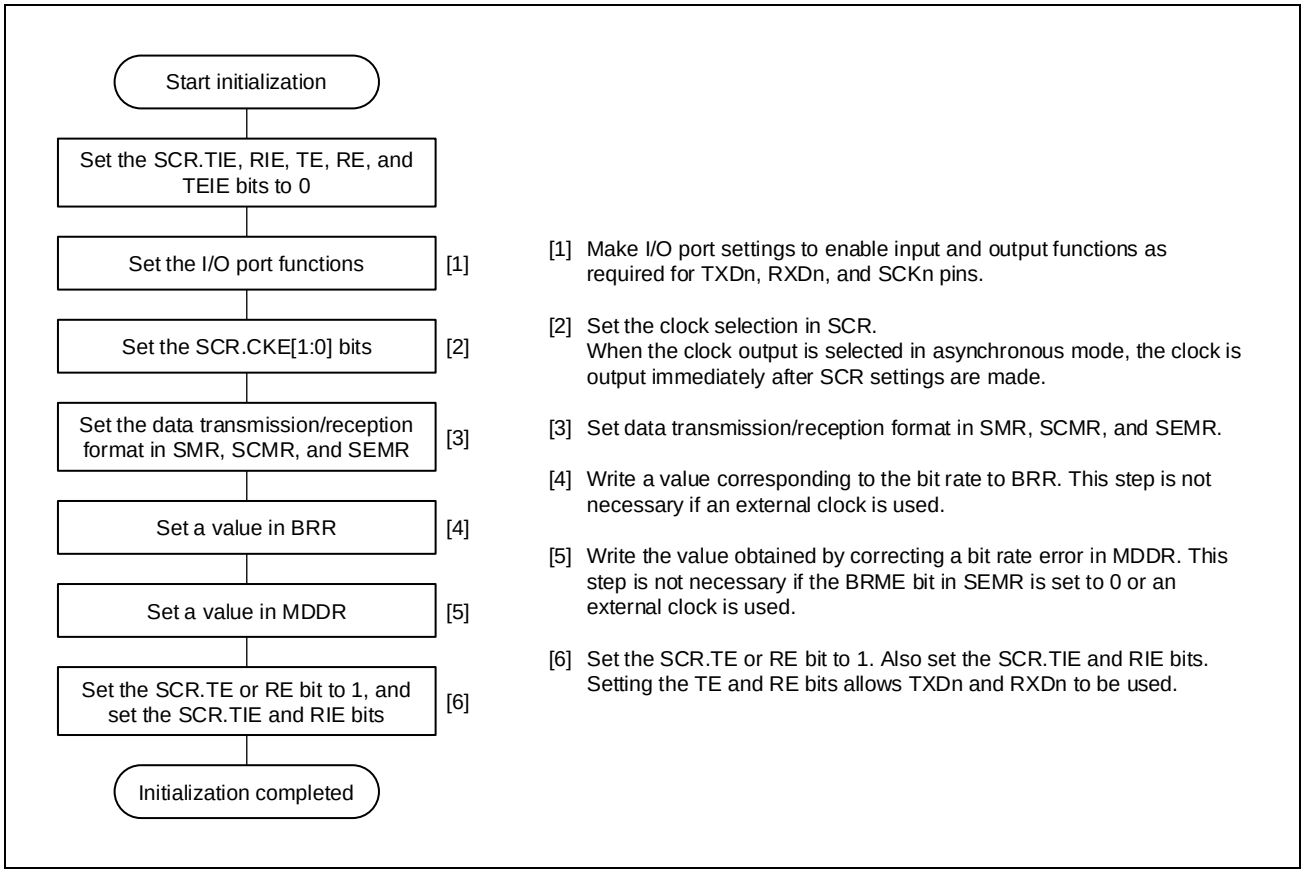



Figure 23.5 Sample SCI Initialization Flowchart (Asynchronous Mode)

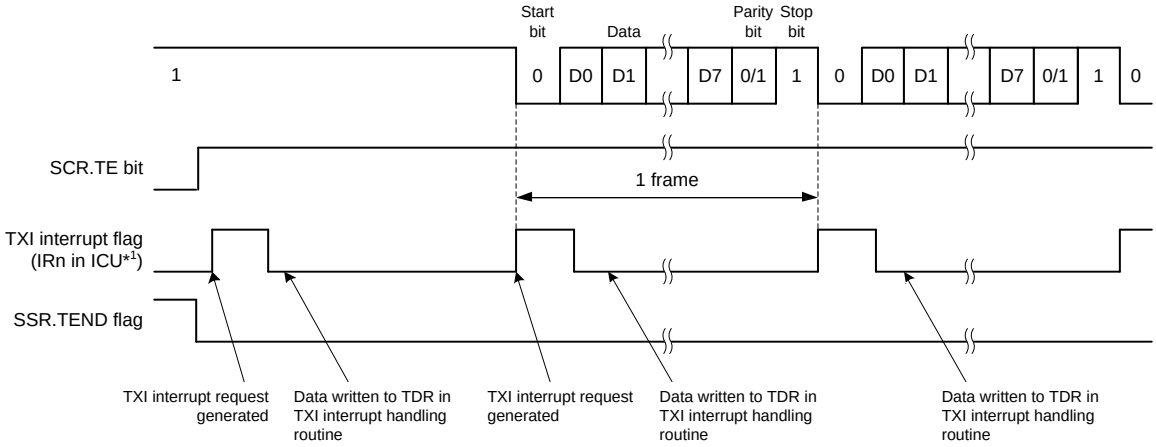
### 23.3.7 Serial Data Transmission (Asynchronous Mode)

**Figure 23.6** to **Figure 23.8** show an example of the operation for serial transmission in asynchronous mode. In serial transmission, the SCI operates as described below.

1. The SCI transfers data from TDR*1 to TSR when data is written to TDR*1 in the TXI interrupt handling routine. The TXI interrupt request at the beginning of transmission is generated when the TE bit in SCR is set to 1 after the TIE bit in SCR is set to 1 or when these 2 bits are set to 1 simultaneously by a single instruction.
2. Transmission starts after the CTSE bit in SECR is set to 0 (CTS function is disabled) and a low level on the CTSn# pin causes data transfer from TDR*1 to TSR. If the TIE bit in SCR is 1 at this time, a TXI interrupt request is generated. Continuous transmission is obtainable by writing the next transmit data to TDR*1 in the TXI interrupt handling routine before transmission of the current transmit data is completed. When TEI interrupt requests are in use, set the SCR.TIE bit to 0 (a TXI interrupt request is disabled) and the SCR.TEIE bit to 1 (a TEI interrupt request is enabled) after the last of the data to be transmitted are written to the TDR*1*2 from the handling routine for TXI requests.
3. Data is sent from the TXDn pin in the following order: start bit, transmit data, parity bit or multi-processor bit (may be omitted depending on the format), and stop bit.
4. The SCI checks for updating of (writing to) TDR*3 at the time of stop bit output.
5. When TDR*3 is updated, setting of the CTSE bit in SECR to 0 (CTS function is disabled) or a low level input on the CTSn# pin cause the next transfer of the next transmit data from TDR*1 to TSR and sending of the stop bit, after which serial transmission of the next frame starts.
6. If TDR*3 is not updated, the TEND flag in SSR is set to 1, the stop bit is sent, and then the mark state is entered in which 1 is output. If the TEIE bit in SCR is 1 at this time, the TEND flag in SSR is set to 1 and a TEI interrupt request is generated.

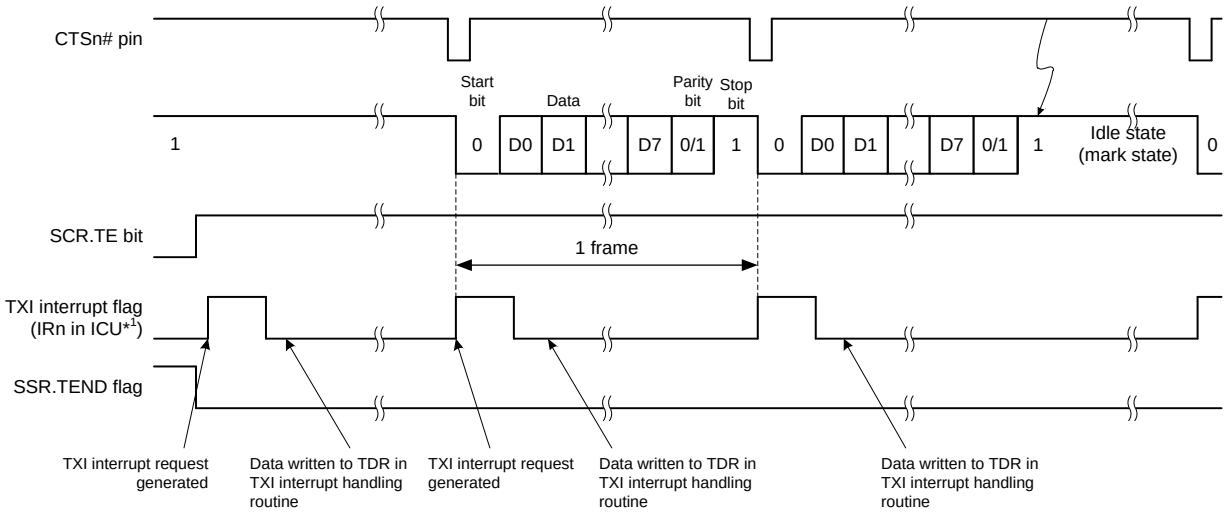
**Note 1.** Write data not to TDR but to the TDRH and TDRL registers when 9-bit data length is selected.
**Note 2.** Write data in the order from TDRH to TDRL when 9-bit data length is selected.
**Note 3.** The SCI checks for updating of the TDRL register only and does not check for updating of the TDRH register when 9-bit data length is selected.

**Figure 23.9** shows a sample flowchart for serial transmission in asynchronous mode.



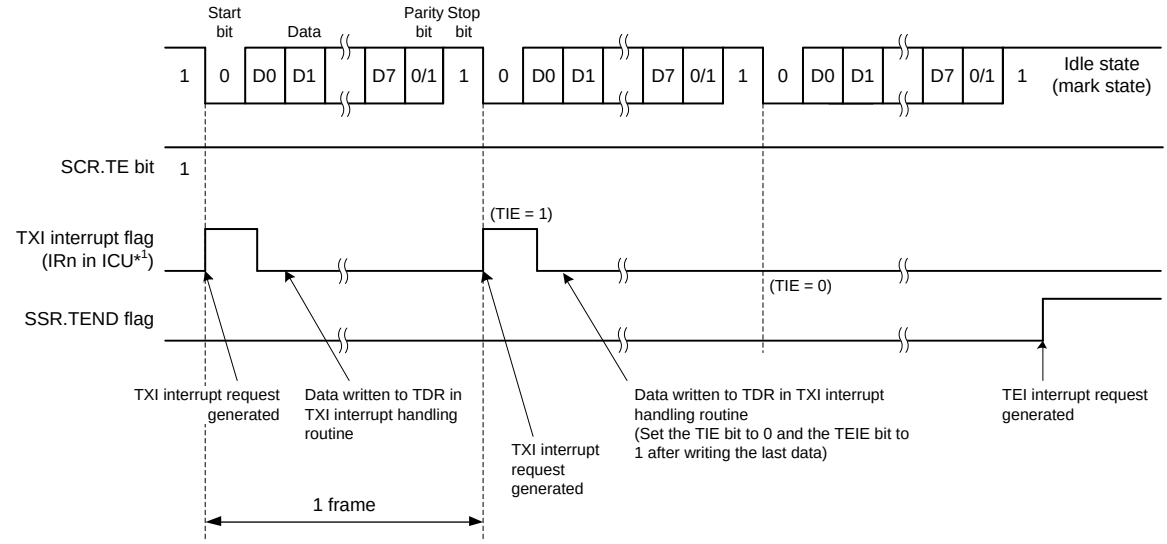

Note 1. Refer to **Section 8, Interrupt Controller** for details on the corresponding interrupt vector number.

Figure 23.6 Example of Operation for Serial Transmission in Asynchronous Mode (1)
(with 8-Bit Data, Parity, 1 Stop Bit, CTS Function Not Used, at the Beginning of Transmission)



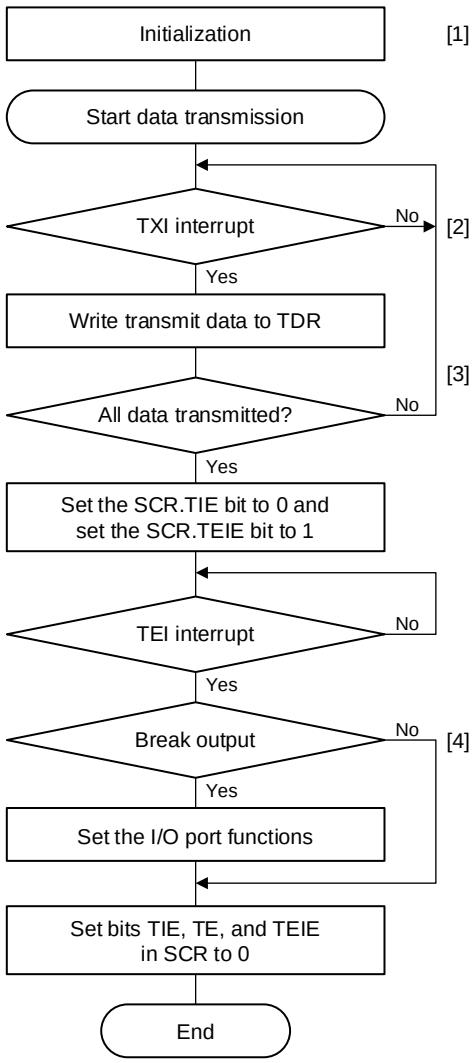

Note 1. Refer to **Section 8, Interrupt Controller** for details on the corresponding interrupt vector number.

Figure 23.7 Example of Operation for Serial Transmission in Asynchronous Mode (2)
(with 8-Bit Data, Parity, 1 Stop Bit, CTS Function Used, at the Beginning of Transmission)




Note 1. Refer to **Section 8, Interrupt Controller** for details on the corresponding interrupt vector number.

Figure 23.8 Example of Operation for Serial Transmission in Asynchronous Mode (3)
(with 8-Bit Data, Parity, 1 Stop Bit, CTS Function Not Used, from the Middle of Transmission until Transmission Completion)




[1] SCI initialization:
Set data transmission.
After the TE bit in SCR is set to 1, 1 is output for a frame, and transmission is enabled.

[2] Transmit data write to TDR by a TXI interrupt request:
When transmit data is transferred from TDR to TSR, a transmit data empty interrupt (TXI) request is generated.
Write transmit data to TDR once in the TXI interrupt handling routine.

[3] Serial transmission continuation procedure:
To continue serial transmission, write transmit data to TDR once using a TXI interrupt request.
Transmit data can also be written to TDR by activating the DMAC.
When TEI interrupt requests are in use, set the SCR.TIE bit to 0 and the SCR.TEIE bit to 1 after the last of the data to be transmitted are written to the TDR.

[4] Break output at the end of serial transmission:
To output a break in serial transmission, set the I/O port function corresponding to the TXDn pin (corresponding to output of the low level), and after switching the TXDn pin to the general I/O port function, set the TE bit in the SCR to 0.

**Note:** The TDR register becomes the TDRH and TDRL registers when 9-bit data length is selected. Write data in the order from TDRH to TDRL.

Figure 23.9 Example of Serial Transmission Flowchart in Asynchronous Mode

### 23.3.8 Serial Data Reception (Asynchronous Mode)

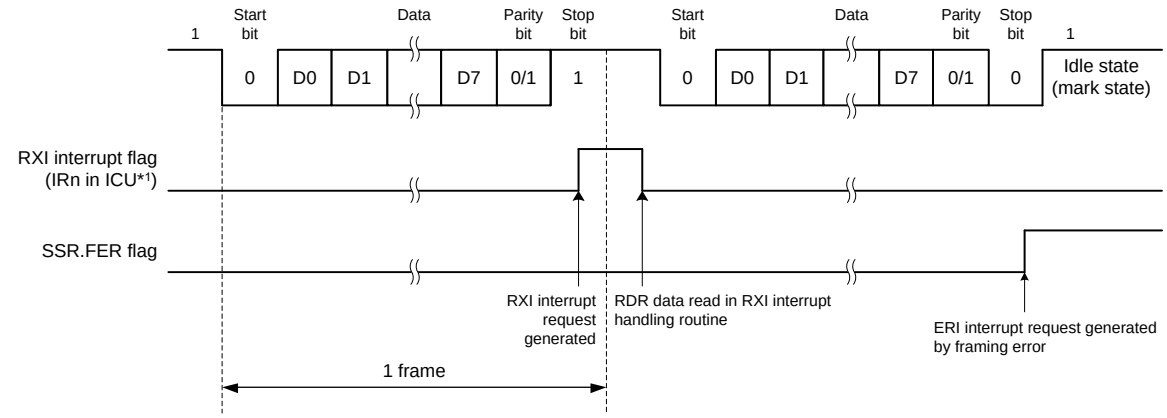
**Figure 23.10** and **Figure 23.11** show an example of the operation for serial data reception in asynchronous mode. In serial data reception, the SCI operates as described below.

1. When the value of the RE bit in SCR becomes 1, the output signal on the RTSn# pin goes to the low level.
2. When the SCI monitors the communications line and detects a start bit, it performs internal synchronization, stores receive data in RSR, and checks the parity bit and stop bit.
3. If an overrun error occurs, the ORER flag in SSR is set to 1. If the RIE bit in SCR is 1 at this time, an ERI interrupt request is generated. Receive data is not transferred to RDR*1.
4. If a parity error is detected, the PER bit in SSR is set to 1 and receive data is transferred to RDR*1. If the RIE bit in SCR is 1 at this time, an ERI interrupt request is generated.
5. If a framing error (when the stop bit is 0) is detected, the FER bit in SSR is set to 1 and receive data is transferred to RDR*1. If the RIE bit in SCR is 1 at this time, an ERI interrupt request is generated.
6. When reception finishes successfully, receive data is transferred to RDR*1. If the RIE bit in SCR is 1 at this time, an RXI interrupt request is generated. Continuous reception is enabled by reading the receive data transferred to RDR*1 in this RXI interrupt handling routine before reception of the next receive data is completed. Reading the received data that have been transferred to RDR*1 causes the RTSn# pin to output the low level.

**Note 1.** Read data not in RDR but in the RDRH and RDRL registers when 9-bit data length is selected.

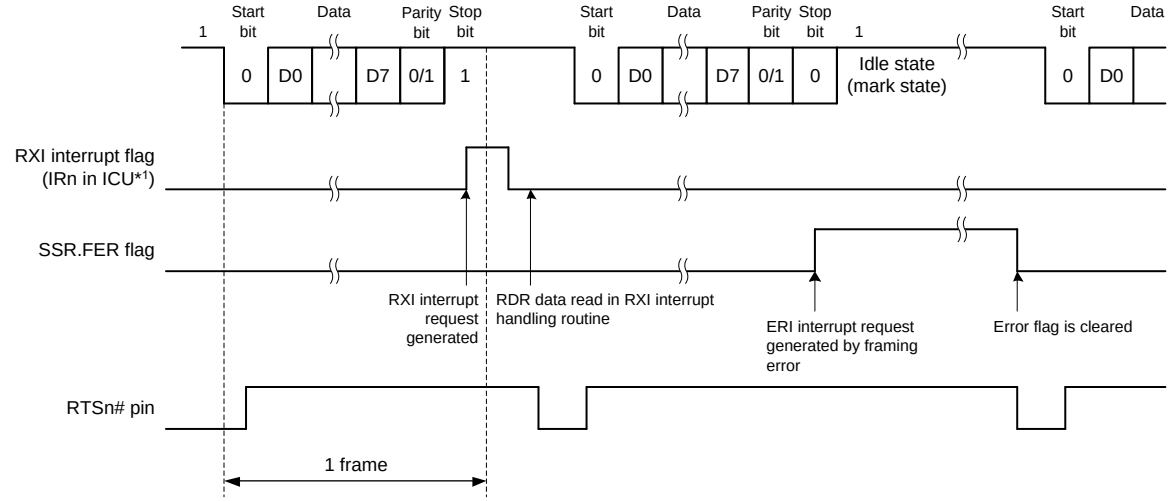
NOTE

The SCI checks for reading of the RDRL register only and does not check for reading of the RDRH register when 9-bit data length is selected.




Note 1. Refer to **Section 8, Interrupt Controller** for details on the corresponding interrupt vector number.

Figure 23.10 Example of SCI Operation for Serial Reception in Asynchronous Mode (1)
(When RTS Function is Not Used) (Example with 8-Bit Data, Parity, 1 Stop Bit)




Note 1. Refer to **Section 8, Interrupt Controller** for details on the corresponding interrupt vector number.

Figure 23.11 Example of SCI Operation for Serial Reception in Asynchronous Mode (2)
(When RTS Function is Used) (Example with 8-Bit Data, Parity, 1 Stop Bit)

**Table 23.19** lists the states of the flags in the SSR status register and receive data handling when a receive error is detected.

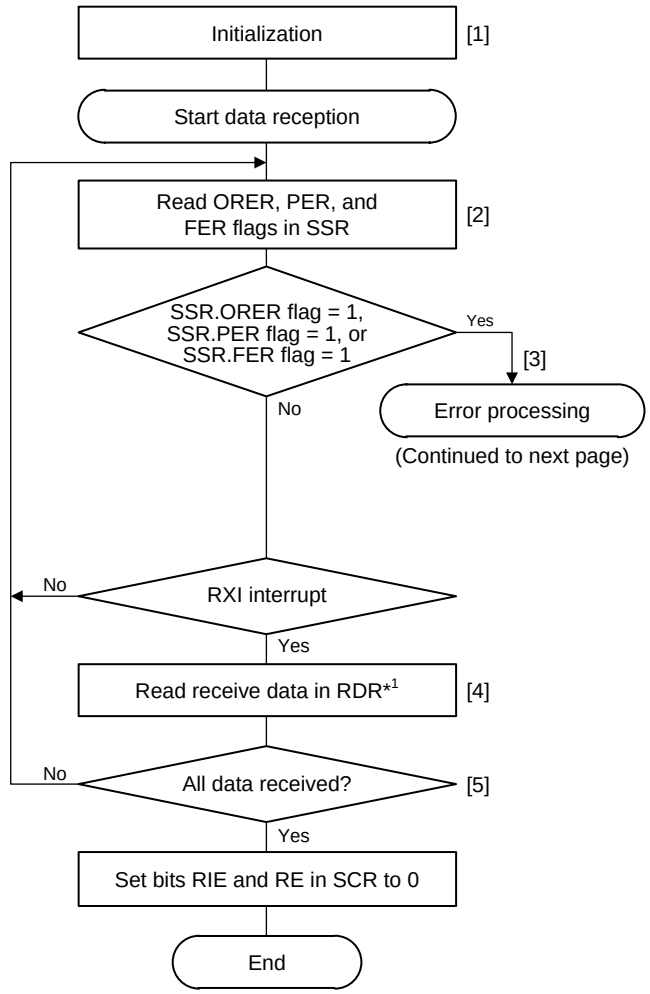
If a receive error is detected, an ERI interrupt request is generated but an RXI interrupt request is not generated. Data reception cannot be resumed while the receive error flag is 1. Accordingly, set the ORER, FER, and PER bits to 0 before resuming reception. Moreover, be sure to read the RDR (or the RDRL) during overrun error processing. When a reception is forcibly terminated by setting the SCR.RE bit to 0 during operation, read the RDR (or the RDRL) register because received data which has not yet been read may be left in RDR (or the RDRL).

**Figure 23.12** and **Figure 23.13** show samples of flowcharts for serial data reception.

Table 23.19 Flags in the SSR Status Register and Receive Data Handling

| Flags in the SSR Status Register | | | | |
|---|---|---|---|---|
| ORER | FER | PER | Receive Data | Receive Error Type |
| 1 | 0 | 0 | Lost | Overrun error |
| 0 | 1 | 0 | Transferred to RDR*1 | Framing error |
| 0 | 0 | 1 | Transferred to RDR*1 | Parity error |
| 1 | 1 | 0 | Lost | Overrun error + framing error |
| 1 | 0 | 1 | Lost | Overrun error + parity error |
| 0 | 1 | 1 | Transferred to RDR*1 | Framing error + parity error |
| 1 | 1 | 1 | Lost | Overrun error + framing error + parity error |

Note 1. Read data not in RDR but in the RDRH and RDRL registers when 9-bit data length is selected.




[1] SCI initialization:
Set data reception.

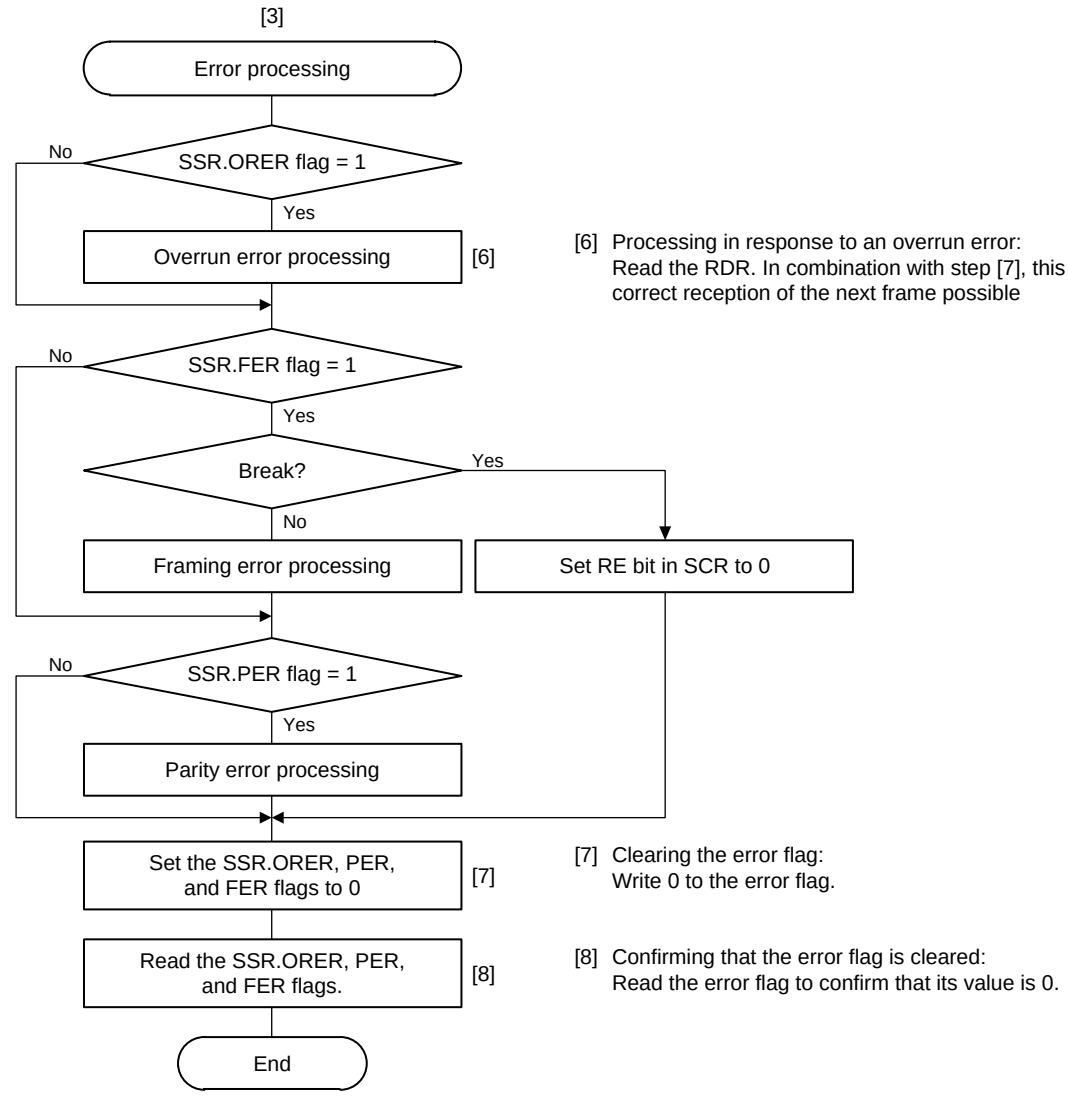
[2][3] Receive error processing and break detection:
If a receive error occurs, an ERI interrupt is generated. An error is identified by reading the ORER, PER, and FER flags in SSR. After performing the appropriate error processing, be sure to set the ORER, PER, and FER flags to 0. Reception cannot be resumed if any of these flags is set to 1. In the case of a framing error, a break can be detected by reading the value of the input port corresponding to the RXDn pin.

[4] Read the receive data in RDR once in the RXI interrupt handling routine.

[5] Serial reception continuation procedure:
To continue serial reception, before the stop bit of the current frame is received, read data from RDR in the RXI interrupt processing routine.
The RDR data can also be read by activating the DMAC.

Note 1. The RDR register becomes the RDRH and RDRL registers when 9-bit data length is selected. Read data in the order from RDRH to RDRL.

Figure 23.12 Example Flowchart of Serial Reception in Asynchronous Mode (1)




Note 1. The RDR register becomes the RDRL register when 9-bit data length is selected. Reading the RDRH register is not necessary.

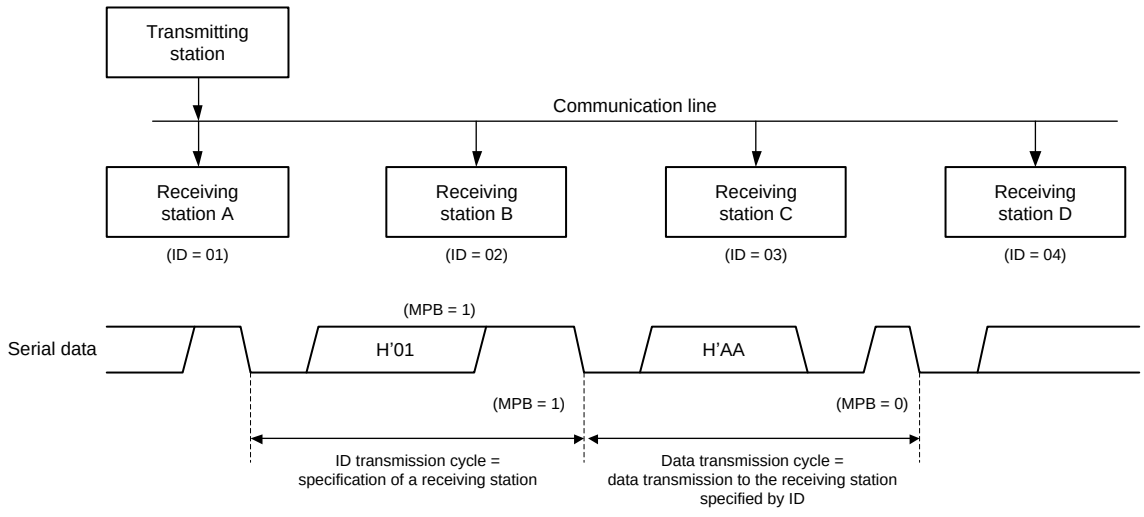
Figure 23.13 Example Flowchart of Serial Reception in Asynchronous Mode (2)

## 23.4 Multi-Processor Communications Function

Using the multi-processor communication functions enables to transmit and receive data by sharing a communication line between multiple processors by using asynchronous serial communication in which the multi-processor bit is added. In multi-processor communication, a unique ID code is allocated to each receiving station. Serial communication cycles consist of an ID transmission cycle to specify the receiving station and a data transmission cycle to transmit data to the specified receiving station. The multi-processor bit is used to distinguish between the ID transmission cycle and the data transmission cycle. When the multi-processor bit is set to 1, it indicates the ID transmission cycle and when the multi- processor bit is set to 0, it indicates the data transmission cycle. **Figure 23.14** shows an example of communication between processors by using a multi-processor format. First, a transmitting station transmits communication data in which the multi-processor bit set to 1 is added to the ID code of the receiving station. Next, the transmitting station transmits the communication data in which the multi-processor bit set to 0 is added to the transmit data. Upon receiving the communication data in which the multi-processor bit is set to 1, the receiving station compares the received ID with the ID of the receiving station itself and if the two match, receives the communication data that is subsequently transmitted. If the received ID does not match with the ID of the receiving station, the receiving station skips the communication data until again receiving the communication data in which the multi-processor bit is set to 1.

For supporting this function, the SCI provides the MPIE bit in SCR. When the MPIE bit is set to 1, transfer of receive data from the RSR to the RDR (the RDRH and RDRL registers when 9-bit data length is selected), detection of a receive error, and setting the respective status flags ORER and FER in SSR are disabled until reception of data in which the multi-processor bit is set to 1. Upon receiving a reception character in which the multi-processor bit is set to 1, the MPBT bit in SSR is set to 1 and the MPIE bit in SCR is automatically cleared, thus returning to a normal reception operation. During this time, an RXI interrupt is generated if the RIE bit in SCR is set.

When the multi-processor format is specified, specification of the parity bit is disabled. Apart from this, there is no difference from the operation in the normal asynchronous mode. A clock which is used for the multi-processor communication is also the same as the clock used in the normal asynchronous mode.

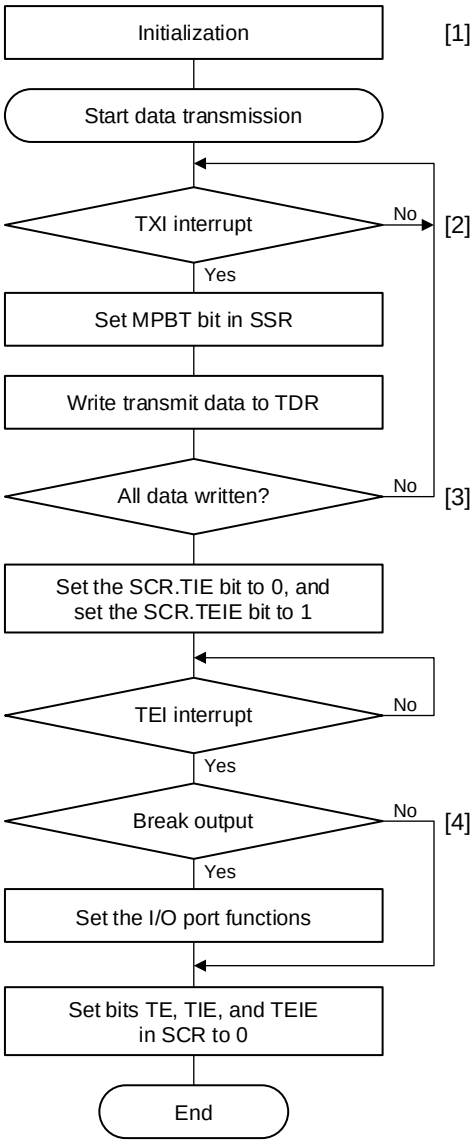



**Note:** MPB: Multi-processor bit

Figure 23.14 An Example of Communication using the Multi-Processor Format
(Example of Transmission of Data H'AA to Receiving Station A)

### 23.4.1 Multi-Processor Serial Data Transmission

**Figure 23.15** is a sample flowchart of multi-processor data transmission. In the ID transmission cycle, the ID should be transmitted with the MPBT bit in SSR set to 1. In the data transmission cycle, the data should be transmitted with the MPBT bit set to 0. The other operations are the same as the operations in asynchronous mode.




[1] SCI initialization:
Set data transmission.
After the TE bit in SCR is set to 1, 1 is output for a frame, and transmission is enabled.

[2] TXI interrupt request:
When transmit data is transferred from TDR to TSR, a transmit data empty interrupt (TXI) request is generated.
Set the MPBT bit in SSR to 0 or 1, and write transmit data to TDR once in the TXI interrupt handling routine.

[3] Serial transmission continuation procedure:
To continue serial transmission, write transmit data to TDR once using a TXI interrupt request.
Transmit data can also be written to TDR by activating the DMAC.
When TEI interrupt requests are in use, set the SCR.TIE bit to 0 and the SCR.TEIE bit to 1 after the last of the data to be transmitted are written to the TDR.

[4] Break output at the end of serial transmission:
To output a break in serial transmission, set the I/O port function corresponding to the TXDn pin (corresponding to output of the low level), and after switching the TXDn pin to the general I/O port function, set the TE bit in the SCR to 0.

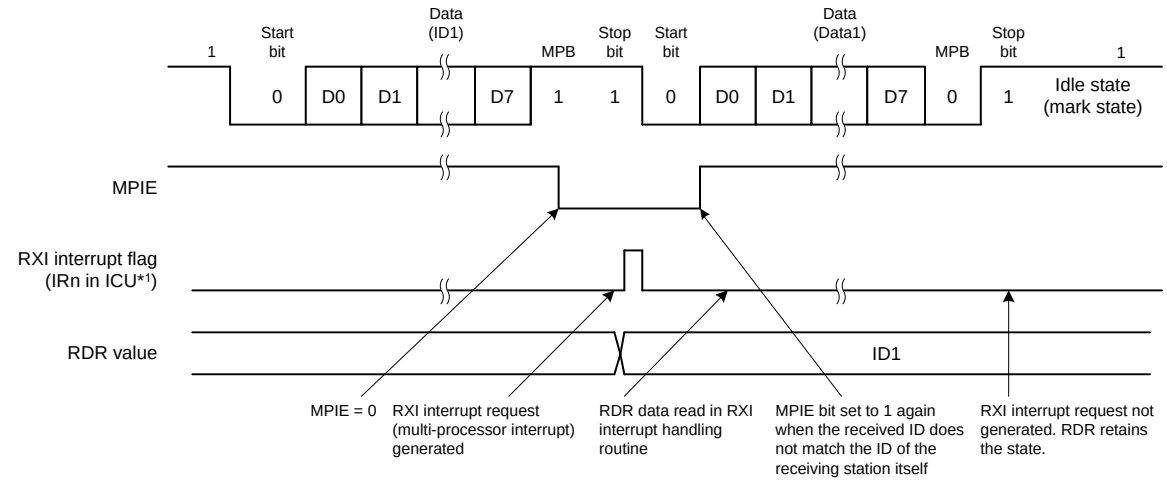
**Note:** The TDR register becomes the TDRH and TDRL registers when 9-bit data length is selected. Write data in the order from TDRH to TDRL.

Figure 23.15 Example of Multi-Processor Serial Transmission Flowchart

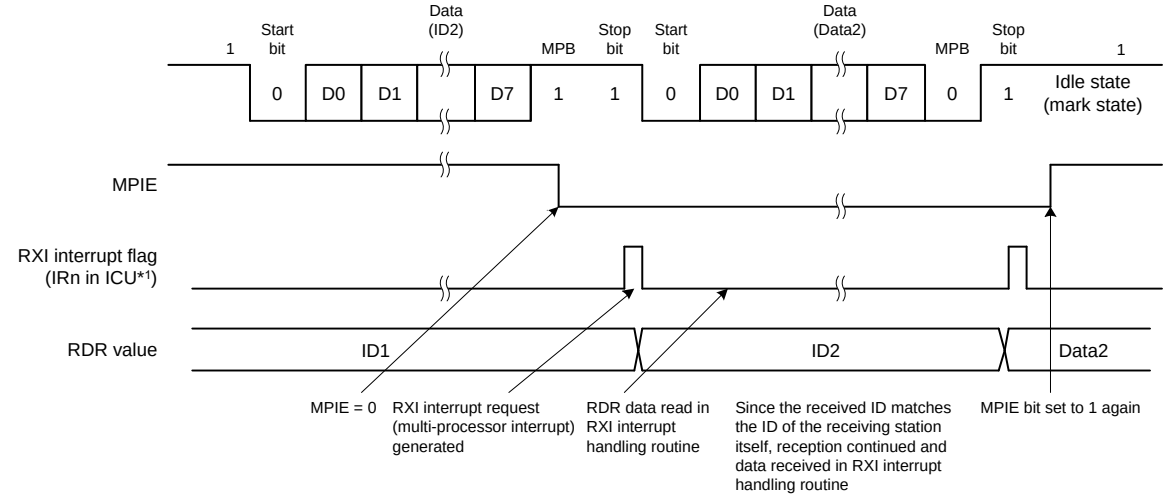
### 23.4.2 Multi-Processor Serial Data Reception

**Figure 23.17** and **Figure 23.18** are sample flowcharts of multi-processor data reception. When the MPIE bit in SCR is set to 1, reading the communication data is skipped until reception of the communication data in which the multi-processor bit is set to 1. When the communication data in which the multi-processor bit is set to 1 is received, the received data is transferred to RDR (the RDRH and RDRL registers when 9-bit data length is selected). During this time, the RXI interrupt request is generated. The other operations are the same as the operations in asynchronous mode.

**Figure 23.16** is the example of operation for reception.



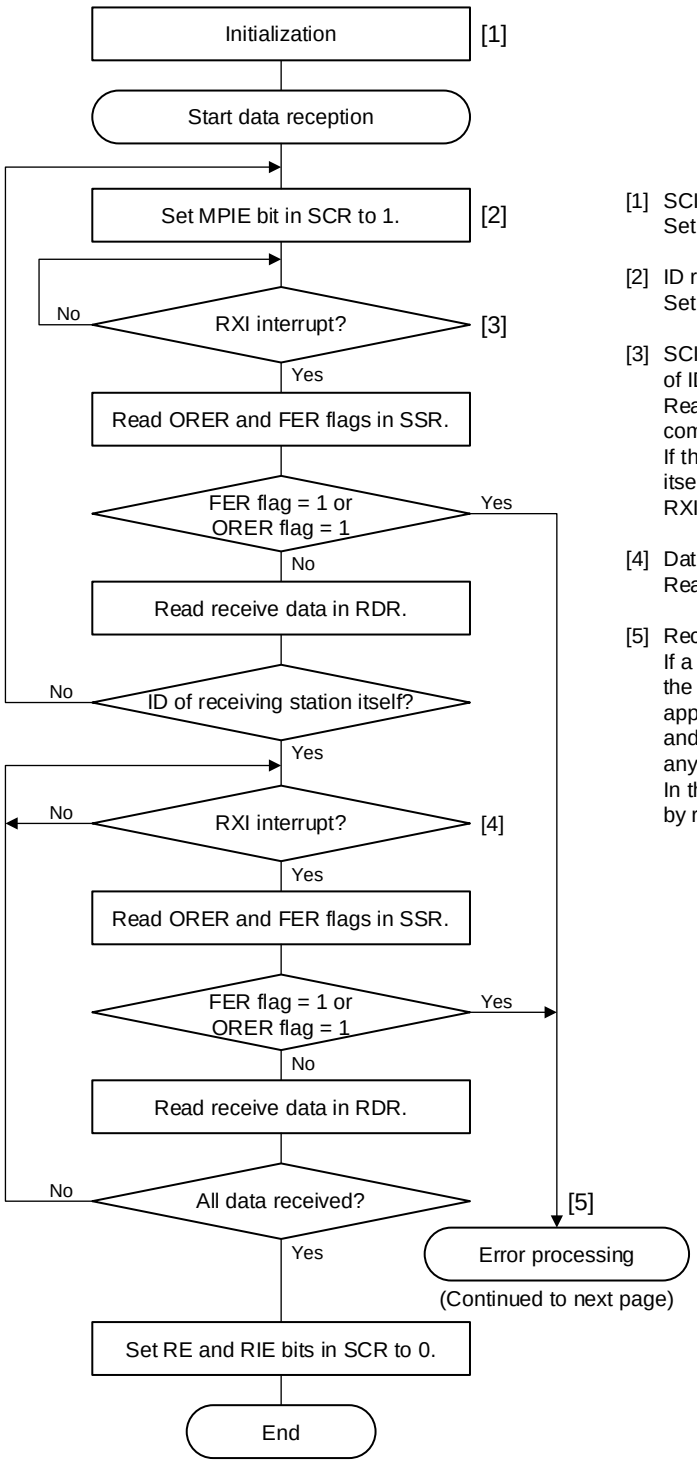

(a) When the received ID does not match the ID of the receiving station itself




(b) When the received ID matches the ID of the receiving station itself

Note 1. Refer to **Section 8, Interrupt Controller** for details on the corresponding interrupt vector number.

Figure 23.16 Example of SCI Reception (8-Bit Data/Multi-Processor Bit/1 Stop Bit)




[1] SCI initialization:
Set data reception.

[2] ID reception cycle:
Set the MPIE bit in SCR to 1 and wait for ID reception.

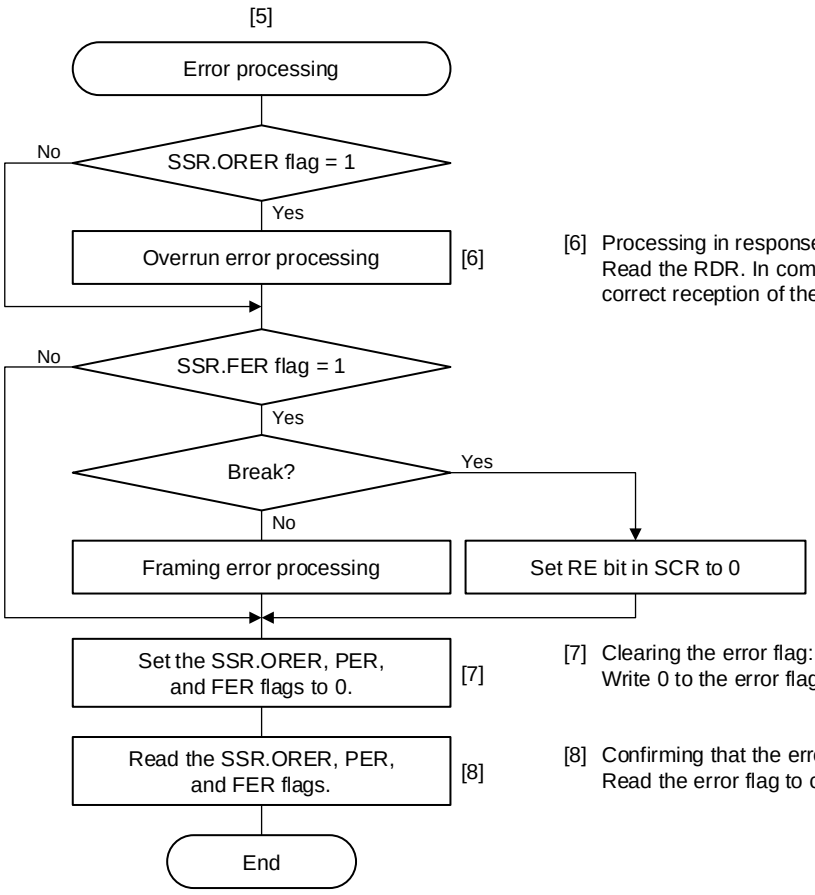
[3] SCI status confirmation and reception and comparison of ID:
Read data in RDR at the first RXI interrupt, and compare it with the ID of the receiving station itself.
If the ID does not match the ID of the receiving station itself, set the MPIE bit to 1 again, and wait for another RXI interrupt RXI interrupt request.

[4] Data reception at an RXI interrupt:
Read data in RDR once in the RXI interrupt routine.

[5] Receive error processing and break detection:
If a receive error occurs, an error is identified by reading the ORER and FER flags in SSR. After performing the appropriate error processing, be sure to set the ORER and FER flags to 0. Reception cannot be resumed if any of these flags is set to 1.
In the case of a framing error, a break can be detected by reading the value of the RXDn pin.

**Note:** The RDR register becomes the RDRH and RDRL registers when 9-bit data length is selected. Read data in the order from RDRH to RDRL.

Figure 23.17 Example of Multi-Processor Serial Reception Flowchart (1)




[6] Processing in response to an overrun error:
Read the RDR. In combination with step [7], this will make correct reception of the next frame possible.

[7] Clearing the error flag:
Write 0 to the error flag.

[8] Confirming that the error flag is cleared:
Read the error flag to confirm that its value is 0.

**Note:** The RDR register becomes the RDRH and RDRL registers when 9-bit data length is selected. Read data in the order from RDRH to RDRL.

Figure 23.18 Example of Multi-Processor Serial Reception Flowchart (2)

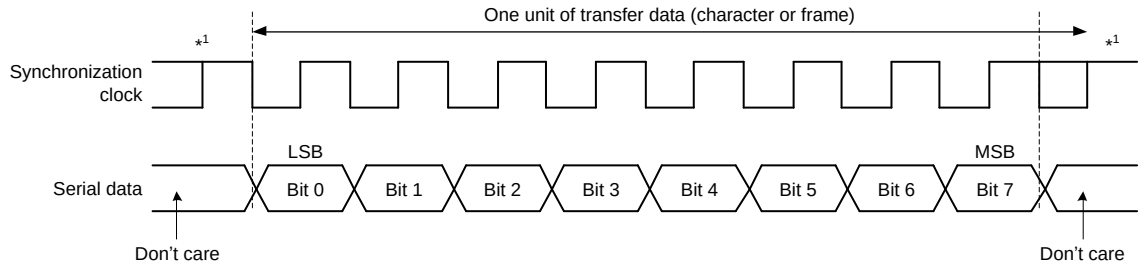
## 23.5 Operation in Clock Synchronous Mode

**Figure 23.19** shows the data format for clock synchronous serial data communications.

In clock synchronous mode, data is transmitted or received in synchronization with clock pulses. One character in transfer data consists of 8-bit data. In clock synchronous mode, no parity bit can be added.

In data transmission, the SCI outputs data from one falling edge of the synchronization clock to the next. In data reception, the SCI receives data in synchronization with the rising edge of the synchronization clock. After 8-bit data is output, the communication line holds the last bit output state.

Inside the SCI, the transmitter and receiver are independent units, enabling full-duplex communications by use of a common clock. Both the transmitter and the receiver also have a double-buffered structure, so that the next transmit data can be written during transmission or the previous receive data can be read during reception, enabling continuous data transfer.




Note 1. Holds a high level except during continuous transfer.

Figure 23.19 Data Format in Clock Synchronous Serial Communications (LSB First)

### 23.5.1 Clock

Either an internal clock generated by the on-chip baud rate generator or an external synchronization clock input at the SCKn pin can be selected, according to the setting of the SCR.CKE[1:0] bits.

When the SCI is operated on an internal clock, the synchronization clock is output from the SCKn pin. Eight synchronization clock pulses are output in the transfer of one character, and when no transfer is performed the clock is held high. However, when only data reception is performed while the CTS function is disabled, the synchronization clock output is started at the same time when the SCR.RE bit set to 1. The synchronization clock is stopped at the high level when an overrun error occurs or the SCR.RE bit is set to 0.

When only data reception is performed and the CTS function is enabled, the clock output is not started even when the SCR.RE bit set to 1 if the CTSn# pin input is high when the SCR.RE bit is 0. The synchronization clock output is started when the SCR.RE bit is set to 1 and the CTSn# pin input is low. After that, if the CTSn# pin input is high on completion of the frame reception, the synchronization clock output is stopped at the high level. If the CTSn# pin input continues to be low, the synchronization clock is stopped at the high level when an overrun error occurs or the SCR.RE bit is set to 0.

### 23.5.2 CTS and RTS Functions

In the CTS function, CTSn# pin input is used to control reception/transmission start when the clock source is the internal clock. Setting the SECR.CTSE bit to 1 enables the CTS function.

When the CTS function is enabled, placing the low level on the CTSn# pin causes reception/transmission to start.

In the RTS function, RTSn# pin output is used to request reception/transmission start when the clock source is an external synchronizing clock. A low level is output when serial communications become possible. Conditions for output of the low and high level are shown below.

[Conditions for low-level output] Satisfaction of all conditions listed below

- The value of the RE or TE bit in the SCR is 1
- Neither transmission nor reception is in progress
- There are no received data yet to be read (when the SCR.RE bit is 1)
- Transmit data has been written (when the SCR.TE bit is 1)
- ORER flag in SSR is 0

[Condition for high-level output]

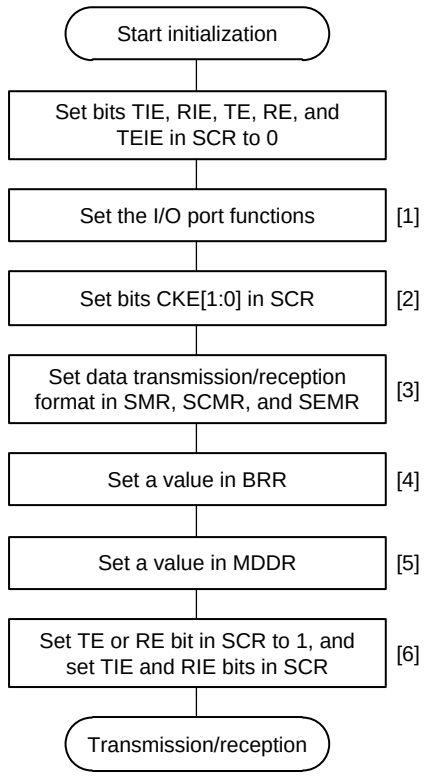
- The conditions for low-level output have not been satisfied.

### 23.5.3 SCI Initialization (Clock Synchronous Mode)

Before transmitting and receiving data, start by writing the initial value H'00 to the SCR and then continue through the procedure for SCI given in **Figure 23.20**. Whenever the operating mode or transfer format is changed, the SCR must be initialized before the change is made.

Note that setting the SCR.RE bit to 0 initializes neither the ORER, FER, and PER flags in SSR nor RDR.

Moreover, note that switching the value of the SCR.TE bit from 1 to 0 or 0 to 1 while the SCR.TIE bit is 1 leads to the generation of a TXI interrupt request.




[1] Make I/O port settings to enable input and output functions as required for TXDn, RXDn, and SCKn pins.

[2] Set the clock selection in SCR.
When an internal clock is selected, the SCK pin functions as the clock output pin.

[3] Set the data transmission/reception format in SMR, SCMR, and SEMR.

[4] Write a value corresponding to the bit rate to BRR. This step is not necessary if an external clock is used.

[5] Write the value obtained by correcting a bit rate error in MDDR. This step is not necessary if the BRME bit in SEMR is cleared to 0 or an external clock is used.

[6] Set the TE bit or RE bit in SCR to 1. Also set the TIE and RIE bits in SCR.
Setting the TE and RE bits allows TXDn and RXDn to be used.

**Note:** In simultaneous transmit and receive operations, the TE and RE bits in SCR should both be set to 0 or 1 simultaneously.

Figure 23.20 Example of SCI Initialization Flowchart (Clock Synchronous Mode)

### 23.5.4 Serial Data Transmission (Clock Synchronous Mode)

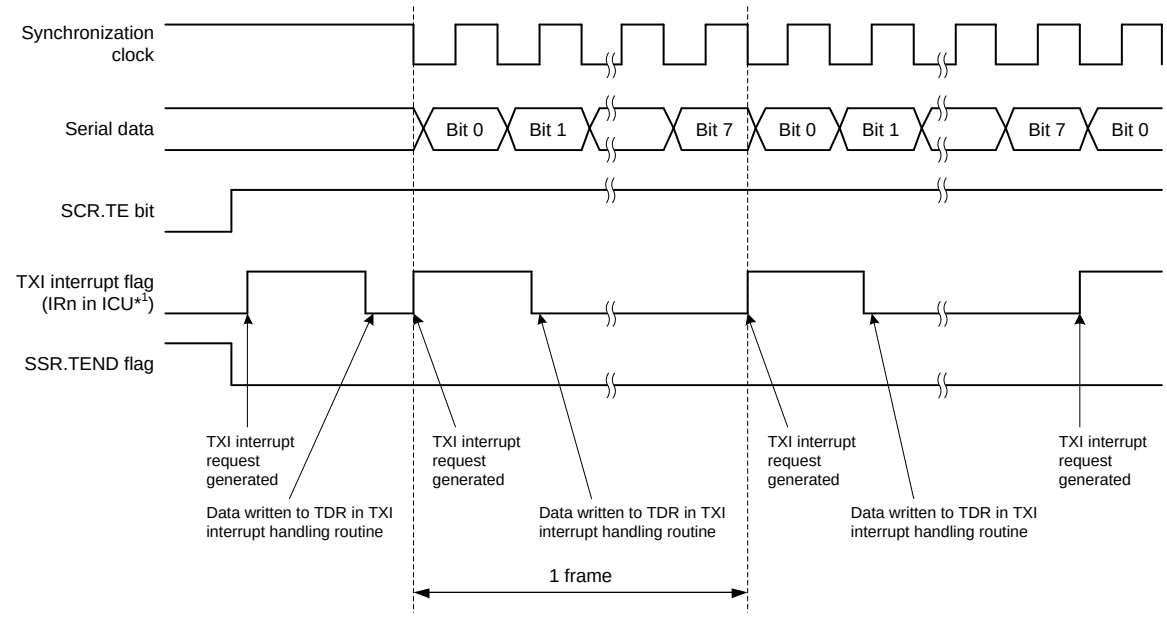
**Figure 23.21**, **Figure 23.22**, and **Figure 23.23** show an example of the operation for serial transmission in clock synchronous mode.

In serial data transmission, the SCI operates as described below.

1. The SCI transfers data from TDR to TSR when data is written to TDR in the TXI interrupt handling routine. The TXI interrupt request at the beginning of transmission is generated when the TE bit in SCR is set to 1 after the TIE bit in SCR is set to 1 or when these 2 bits are set to 1 simultaneously by a single instruction.
2. After transferring data from TDR to TSR, the SCI starts transmission. When the SCR.TIE bit is set to 1 at this time, a TXI interrupt request is generated. Continuous transmission is enabled by writing the next transmit data to TDR in this TXI interrupt handling routine before transmission of the current transmit data has finished. When TEI interrupt requests are in use, set the SCR.TIE bit to 0 (a TXI interrupt request is disabled) and the SCR.TEIE bit to 1 (a TEI interrupt request is enabled) after the last of the data to be transmitted are written to the TDR from the handling routine for TXI requests.
3. 8-bit data is sent from the TXDn pin in synchronization with the output clock when clock output mode has been specified and in synchronization with the input clock when use of an external clock has been specified. Output of the clock signal is suspended until the input CTS signal is at the low level while the CTSE bit in SECR is 1 (CTS function is enabled).
4. The SCI checks for updating of (writing to) the TDR at the time of the last bit output.
5. When TDR is updated, the next transmit data is transferred from TDR to TSR, and serial transmission of the next frame is started.
6. If TDR is not updated, set the SSR.TEND flag to 1 and the TXDn pin retains the output state of the last bit. If the TEIE bit in SCR is 1 at this time, a TEI interrupt request is generated. The SCKn pin is held high.

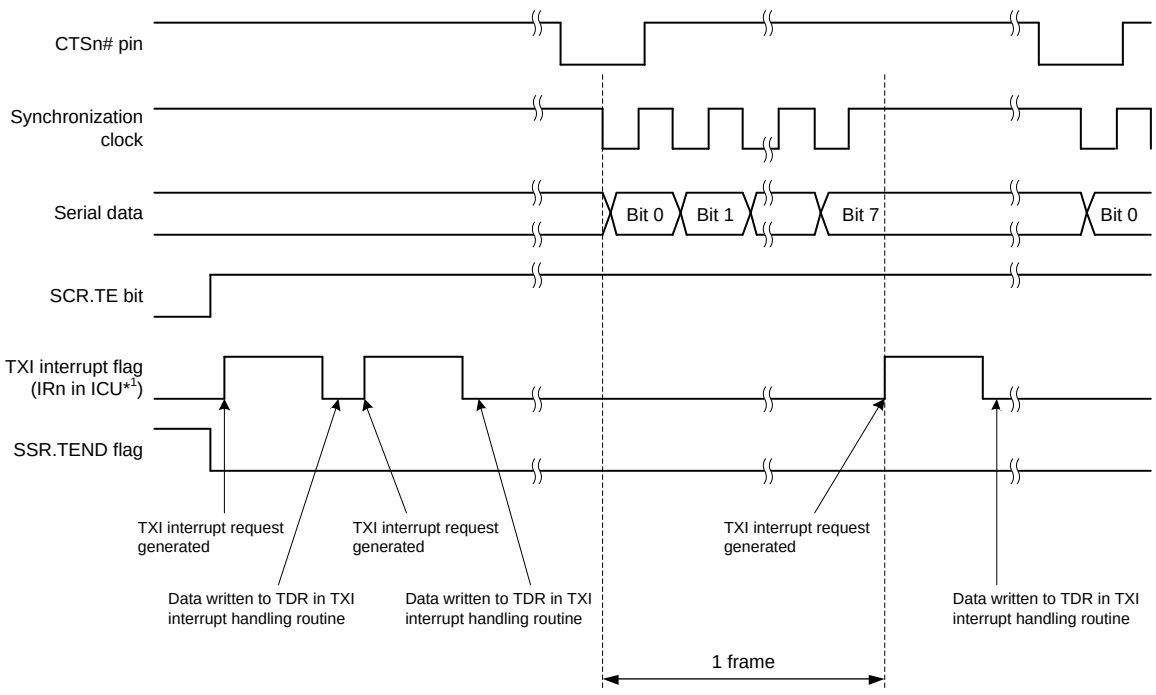
**Figure 23.24** shows a sample flowchart of serial data transmission.

Transmission will not start while a receive error flag (ORER, FER, or PER in SSR) is set to 1. Be sure to set the receive error flags to 0 before starting transmission. Note that setting the RE bit in SCR to 0 does not clear the receive error flags.



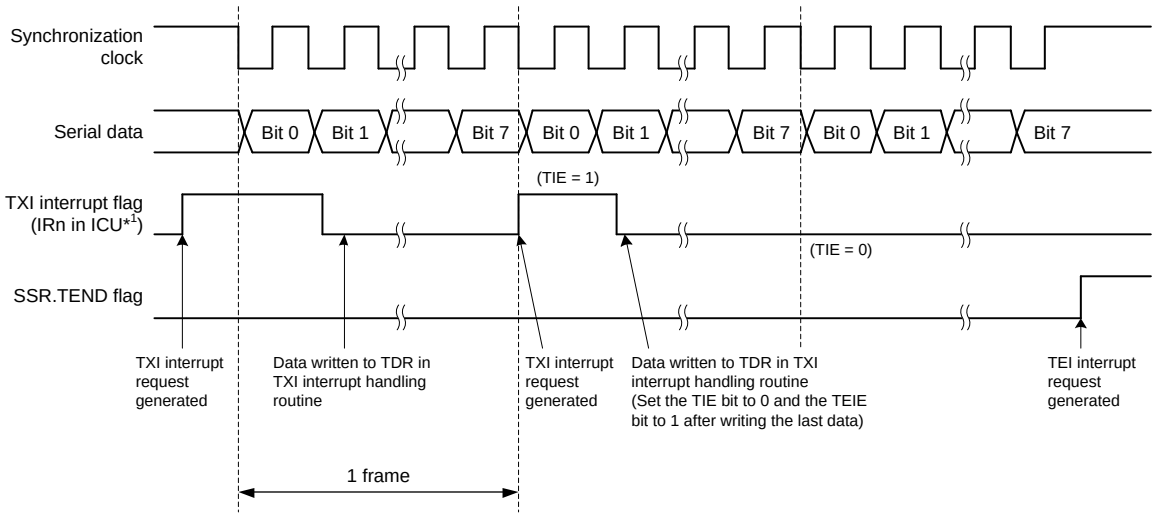

Note 1. Refer to **Section 8, Interrupt Controller** for details on the corresponding interrupt vector number.

Figure 23.21 Example of Serial Data Transmission in Clock Synchronous Mode (1)
(When the CTS Function is Not Used at the Beginning of Transmission)



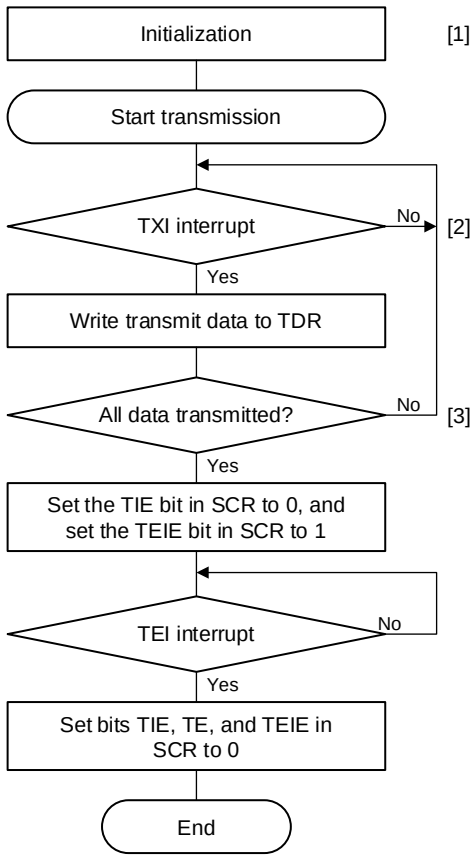

Note 1. Refer to **Section 8, Interrupt Controller** for details on the corresponding interrupt vector number.

Figure 23.22 Example of Serial Data Transmission in Clock Synchronous Mode (2)
(When the CTS Function is Used at the Beginning of Transmission)




Note 1. Refer to **Section 8, Interrupt Controller** for details on the corresponding interrupt vector number.

Figure 23.23 Example of Serial Data Transmission in Clock Synchronous Mode (3)
(From the Middle of Transmission until Transmission Completion)




[1] SCI initialization:
Set data transmission.

[2] Writing transmit data write to TDR by a TXI interrupt request:
When transmit data is transferred from TDR to TSR, a transmit data empty interrupt (TXI) request is generated.
Transmit data is written to TDR once from the handling routine for TXI requests.

[3] Serial transmission continuation procedure:
To continue serial transmission, write transmit data to TDR upon accepting a transmit data empty interrupt (TXI) request. Transmit data can also be written to TDR by activating the DMAC by the TXI interrupt request. When TEI interrupt requests are in use, set the SCR.TIE bit to 0 and the SCR.TEIE bit to 1 after the last of the data to be transmitted are written to the TDR.

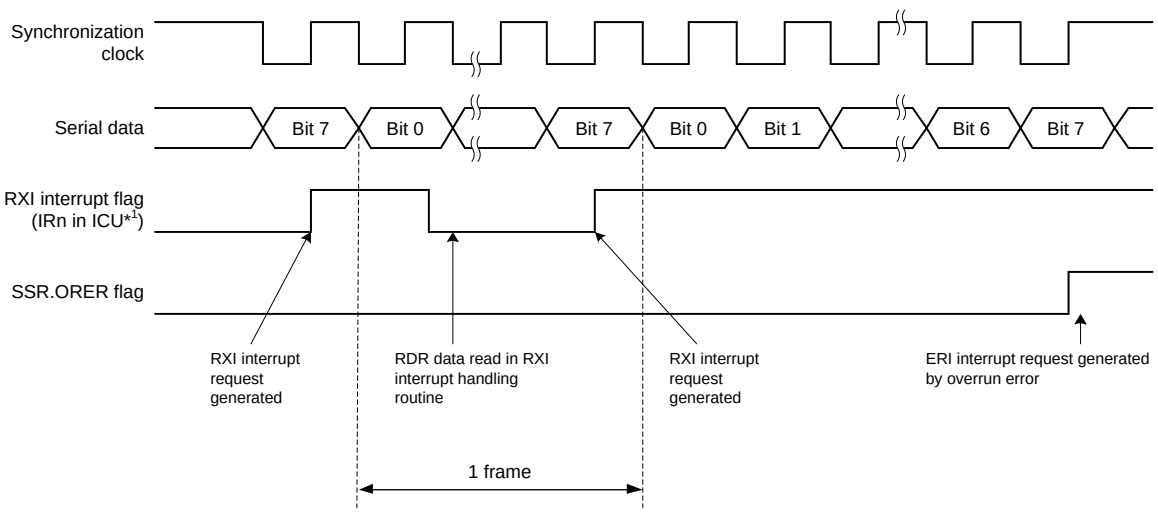
**Note:** When the external clock is in use (the value of the SCR.CKE[1:0] bits is 10b or 11b), the rising edge on the SCK pin for the last bit sets the SSR.TEND flag to 1. Setting the SCR.TE bit to 0 immediately after this may lead to insufficient received-data hold time on the receiver side.

Figure 23.24 Example Flowchart of Serial Transmission in Clock Synchronous Mode

### 23.5.5 Serial Data Reception (Clock Synchronous Mode)

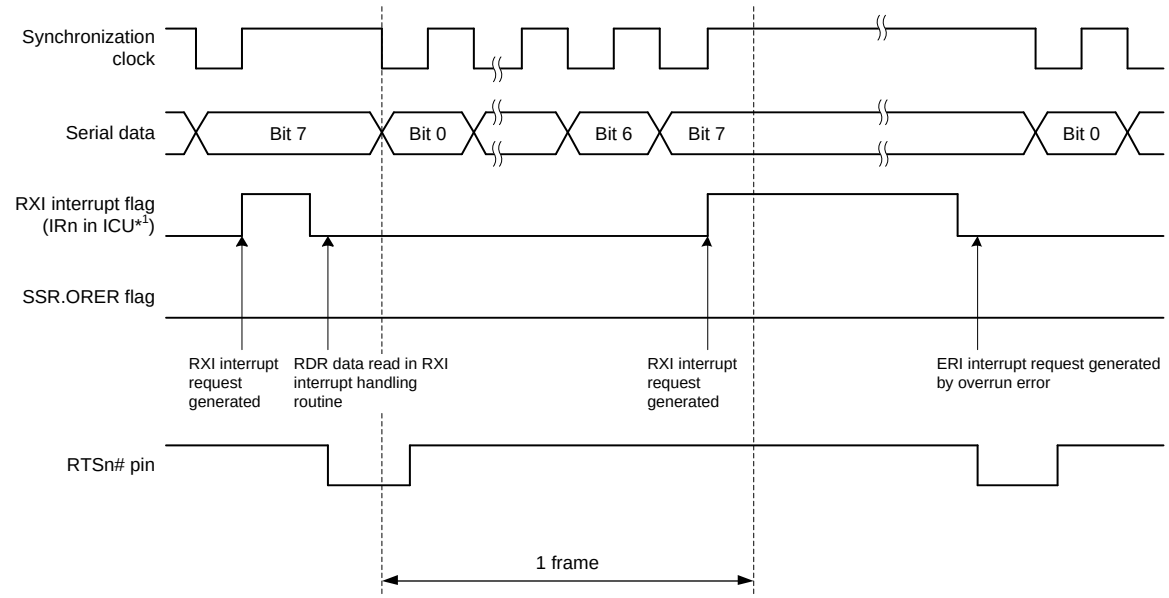
**Figure 23.25** and **Figure 23.26** show an example of SCI operation for serial reception in clock synchronous mode. In serial data reception, the SCI operates as described below.

1. The value of the RE bit in SCR becoming 1 places the signal output on the RTSn# pin at the low level (when the RTS function is in use).
2. The SCI performs internal initialization and starts receiving data in synchronization with a synchronization clock input or output, and stores the receive data in RSR.
3. If an overrun error occurs, the ORER bit in SSR is set to 1. If the RIE bit in SCR is 1 at this time, an ERI interrupt request is generated. Receive data is not transferred to RDR.
4. When reception finishes successfully, receive data is transferred to RDR. If the RIE bit in SCR is 1 at this time, an RXI interrupt request is generated. Continuous reception is enabled by reading the receive data transferred to RDR in this RXI interrupt handling routine before reception of the next receive data is completed. Reading out the received data that have been transferred to RDR causes the RTSn# pin to output the low level (when the RTS function is in use).




Note 1. Refer to **Section 8, Interrupt Controller** for details on the corresponding interrupt vector number.

Figure 23.25 Example of Operation for Serial Reception in Clock Synchronous Mode (1)
(When RTS Function is Not Used)




Note 1. Refer to **Section 8, Interrupt Controller** for details on the corresponding interrupt vector number.

Figure 23.26 Example of Operation for Serial Reception in Clock Synchronous Mode (2) (When RTS Function is Used)

Data transfer cannot be resumed while a receive error flag is 1. Accordingly, clear the ORER, FER, and PER bits in SSR to 0 before resuming reception. Moreover, be sure to read the RDR during overrun error processing. When a reception is forcibly terminated by setting the SCR.RE bit to 0 during operation, read RDR because received data which has not yet been read may be left in RDR.

**Figure 23.27** shows a sample flowchart for serial data reception.

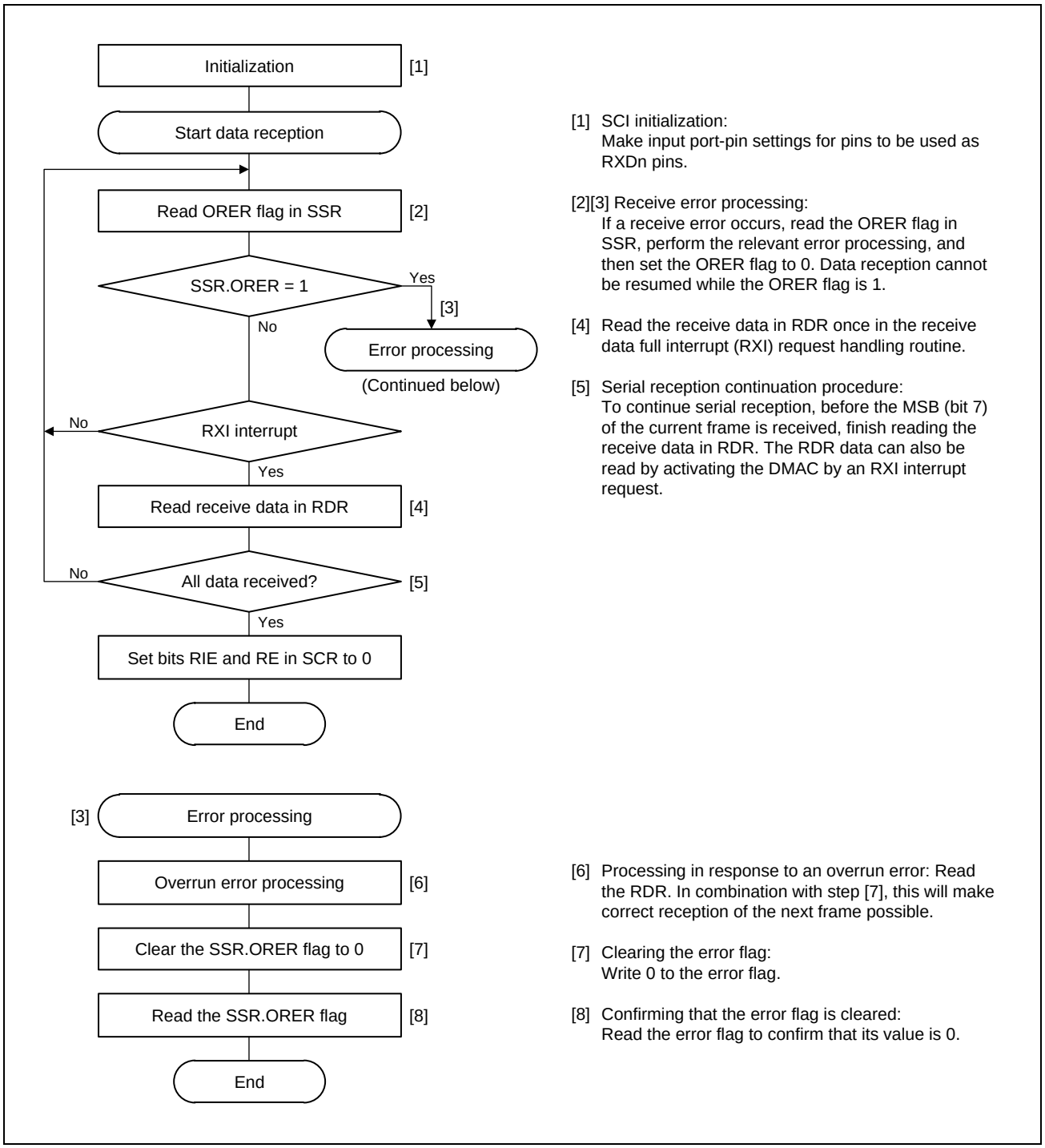



Figure 23.27 Example Flowchart of Serial Reception in Clock Synchronous Mode

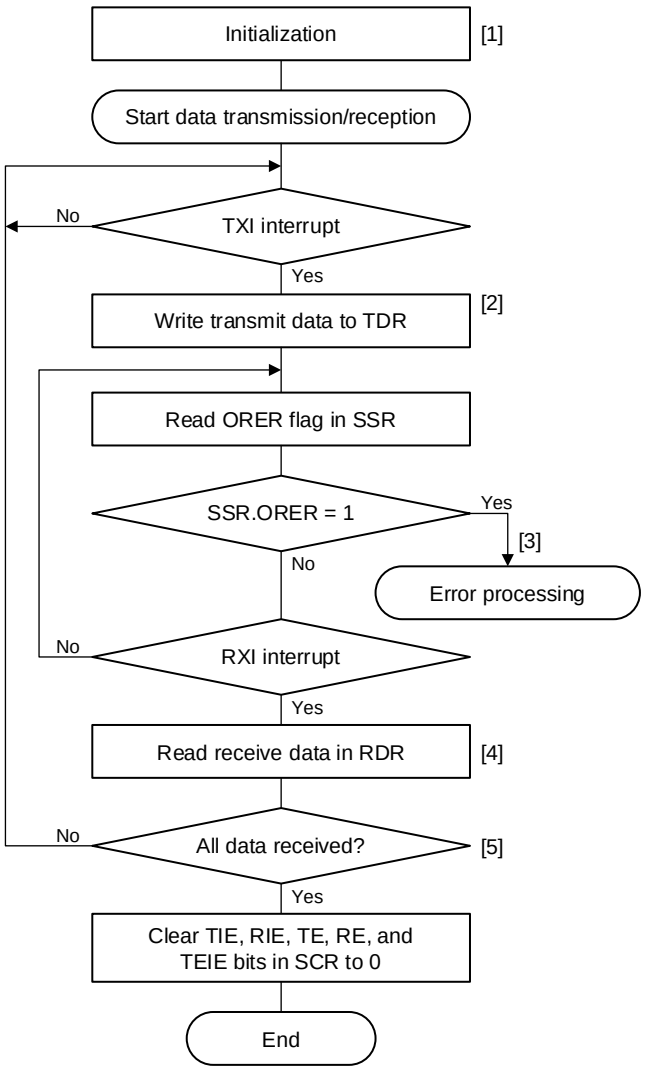
### 23.5.6 Simultaneous Serial Data Transmission and Reception (Clock Synchronous Mode)

**Figure 23.28** shows a sample flowchart for simultaneous serial transmit and receive operations in clock synchronous mode.

After initializing the SCI, the following procedure should be used for simultaneous serial data transmit and receive operations.

To switch from transmit mode to simultaneous transmit and receive mode, check that the SCI has finished transmission by reading that the TEND flag in SSR is 1, and then initialize the SCR register. Then set the TIE, RIE, TE, and RE bits in SCR to 1 simultaneously by a single instruction.

To switch from receive mode to simultaneous transmit and receive mode, check that the SCI has finished reception, and then set the RIE and RE bits to 0. Then check that the receive error flags (ORER, FER, and PER in SSR) are 0, and then set the TIE, RIE, TE, and RE bits in SCR to 1 simultaneously by a single instruction.




[1] SCI initialization:
The TXDn pin can act as the output pin for transmitted data and the RXDn pin can act as the input pin for received data at the same time.

[2] Transmit data write:
Write transmit data to TDR once in the TXI interrupt handling routine.

[3] Receive error processing:
If a receive error occurs, read the ORER flag in SSR, perform the relevant error processing, and then set the ORER flag to 0. Data reception cannot be resumed while the ORER flag is 1.

[4] Reading receive data:
Read the receive data in RDR once in the RXI interrupt request handling routine.

[5] Serial transmission/reception continuation procedure:
To continue serial transmission and reception, before the MSB (bit 7) of the current frame is received, finish reading the receive data in RDR by the RXI interrupt. Also, before the MSB (bit 7) of the current frame is transmitted, write data to TDR by the TXI interrupt.
Transmit data can also be written to TDR by activating the DMAC by a transmit data empty interrupt (TXI) request. Similarly, the RDR data can also be read by activating the DMAC by a receive data full interrupt (RXI) request.

**Note:** When switching from transmit or receive operation to simultaneous transmit and receive operations first set the TIE, RIE, TE, RE, and TEIE bits in SCR to 0, and then set TIE, RIE, TE, and RE bits to 1 simultaneously.

Figure 23.28 Example Flowchart of Simultaneous Serial Transmission and Reception in Clock Synchronous Mode

## 23.6 Operation in Smart Card Interface Mode

The SCI supports smart card (IC card) interfaces conforming to ISO/IEC 7816-3 (standard for Identification Cards), as an extended function of the SCI.

Smart card interface mode can be selected using the appropriate register.

### 23.6.1 Sample Connection

**Figure 23.29** shows a sample connection between a smart card (IC card) and this MCU.

As in the figure, since this MCU communicates with an IC card using a single transmission line, interconnect the TXDn and RXDn pins and pull up the data transmission line to VCC using a resistor.

Setting the TE and RE bits in SCR to 1 with an IC card disconnected enables closed transmission/reception allowing self-diagnosis.

To supply an IC card with the clock pulses generated by the SCI, input the SCKn pin output to the CLK pin of an IC card. The output port of this MCU can be used to output a reset signal.

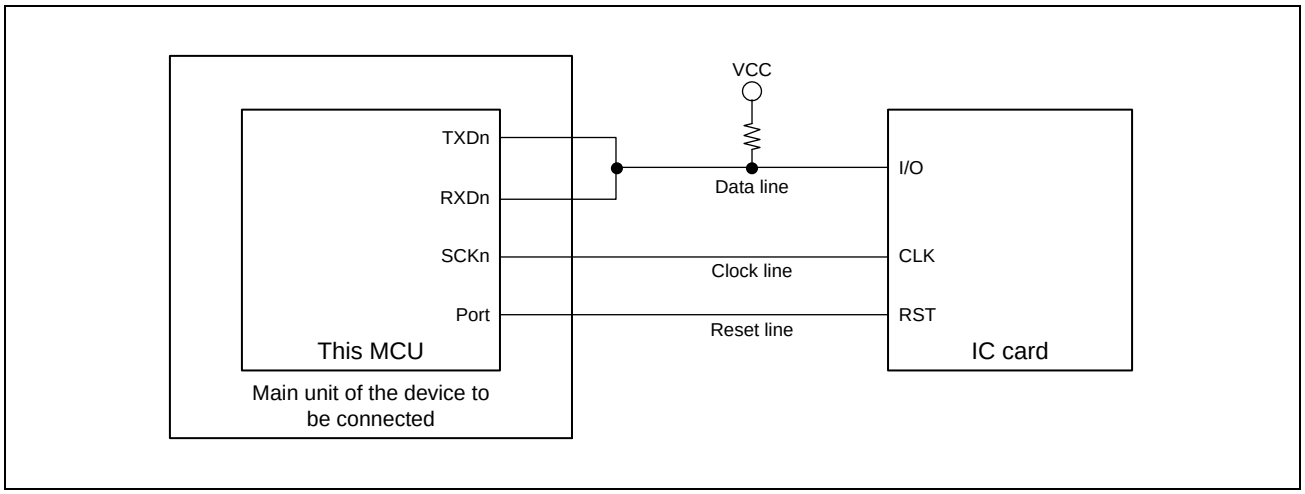
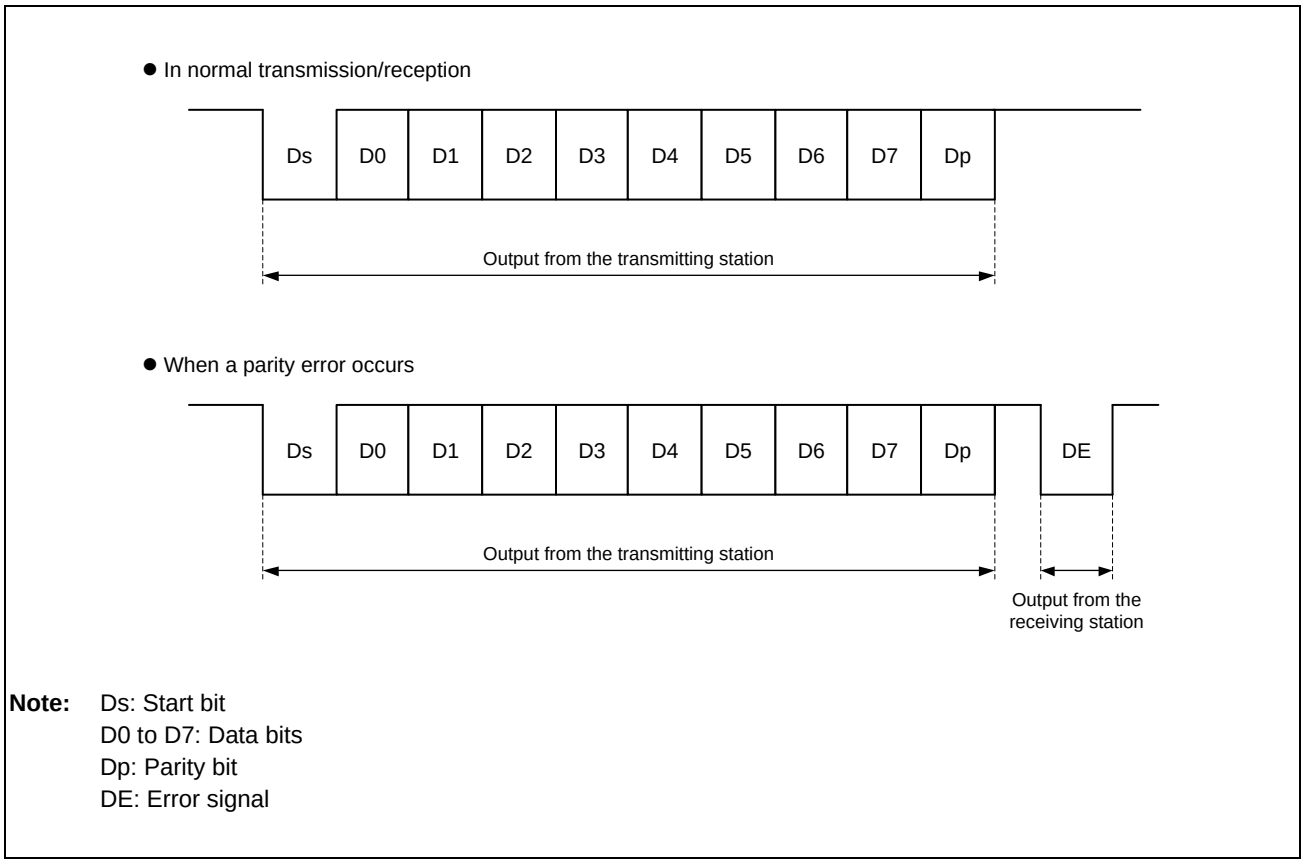



Figure 23.29 Sample Connection with a Smart Card (IC Card)

### 23.6.2 Data Format (Except in Block Transfer Mode)

**Figure 23.30** shows the data transfer formats in smart card interface mode.

- One frame consists of 8-bit data and a parity bit in asynchronous mode.
- During transmission, at least 2 etu (elementary time unit: time required for transferring 1 bit) is secured as a guard time from the end of the parity bit until the start of the next frame.
- If a parity error is detected during reception, a low-level error signal is output for 1 etu after 10.5 etu has passed from the start bit.
- If an error signal is sampled during transmission, the same data is automatically retransmitted after at least 2 etu.




**Note:** Ds: Start bit
D0 to D7: Data bits
Dp: Parity bit
DE: Error signal

Figure 23.30 Data Formats in Smart Card Interface Mode

For communications with IC cards of the direct convention type and inverse convention type, follow the procedure below.

### (1) Direct Convention Type

For the direct convention type, logic levels 1 and 0 correspond to states Z and A, respectively, and data is transferred with LSB first as the start character, as shown in **Figure 23.31**. Therefore, data in the start character in the figure is H'3B.

When using the direct convention type, write 0 to both the SDIR and SINV bits in SCMR. Write 0 to the PM bit in SMR in order to use even parity, which is prescribed by the smart card standard.

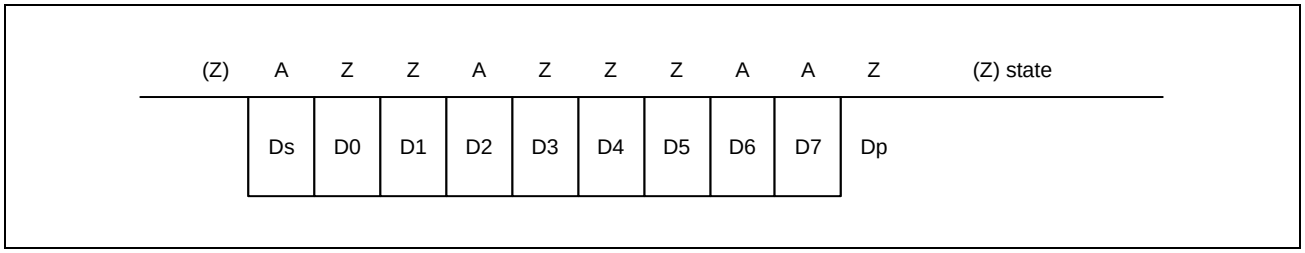



Figure 23.31 Direct Convention (SDIR in SCMR = 0, SINV in SCMR = 0, PM in SMR = 0)

### (2) Inverse Convention Type

For the inverse convention type, logic levels 1 and 0 correspond to states A and Z, respectively and data is transferred with MSB first as the start character, as shown in **Figure 23.32**. Therefore, data in the start character in the figure is H'3F. When using the inverse convention type, write 1 to both the SDIR and SINV bits in SCMR. The parity bit is logic level 0 to produce even parity, which is prescribed by the smart card standard, and corresponds to state Z. Since the SINV bit of this MCU only inverts data bits D7 to D0, write 1 to the PM bit in SMR to invert the parity bit for both transmission and reception.

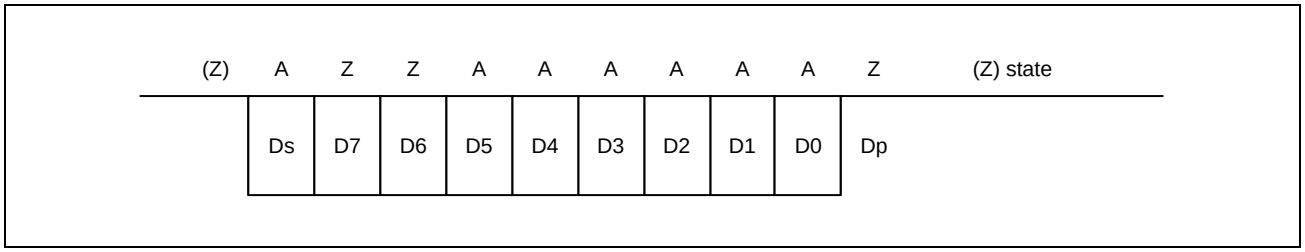



Figure 23.32 Inverse Convention (SDIR in SCMR = 1, SINV in SCMR = 1, PM in SMR = 1)

### 23.6.3 Block Transfer Mode

Block transfer mode is different from normal smart card interface mode in the following respects.

- Even if a parity error is detected during reception, no error signal is output. Since the PER bit in SSR is set by error detection, clear the PER bit before receiving the parity bit of the next frame.
- During transmission, at least 1 etu is secured as a guard time from the end of the parity bit until the start of the next frame.
- Since the same data is not retransmitted during transmission, the TEND flag in SSR is set 11.5 etu after transmission start.
- In block transfer mode, the ERS flag in SSR indicates the error signal status as in normal smart card interface mode, but the flag is read as 0 because no error signal is transferred.

### 23.6.4 Receive Data Sampling Timing and Reception Margin

Only the internal clock generated by the on-chip baud rate generator can be used as a transfer clock in smart card interface mode.

In this mode, the SCI can operate on a base clock with a frequency of 32, 64, 372, 256, 93, 128, 186, or 512 times the bit rate according to the settings of the BCP2 bit in SCMR and the BCP[1:0] bits in SMR (the frequency is always 16 times the bit rate in normal asynchronous mode).

For data reception, the falling edge of the start bit is sampled with the base clock to perform internal synchronization. Receive data is sampled on the 16th, 32nd, 186th, 128th, 46th, 64th, 93rd, and 256th rising edges of the base clock so that it can be latched at the middle of each bit as shown in **Figure 23.33**. The reception margin here is determined by the following formula.

$$M = \left| (0.5 - \frac{1}{2N}) - (L - 0.5)F - \frac{|D - 0.5|}{N}(1 + F) \right| \times 100\ [\%]$$

M: Reception margin (%)
N: Ratio of bit rate to clock (N = 32, 64, 372, 256) D: Duty cycle of clock (D = 0 to 1.0)
L: Frame length (L = 10)
F: Absolute value of clock frequency deviation

Assuming values of F = 0, D = 0.5, and N = 372 in the above formula, the reception margin is determined by the formula below.

$$M = \{0.5 - 1/(2 \times 372)\} \times 100\ [\%] = 49.866\%$$

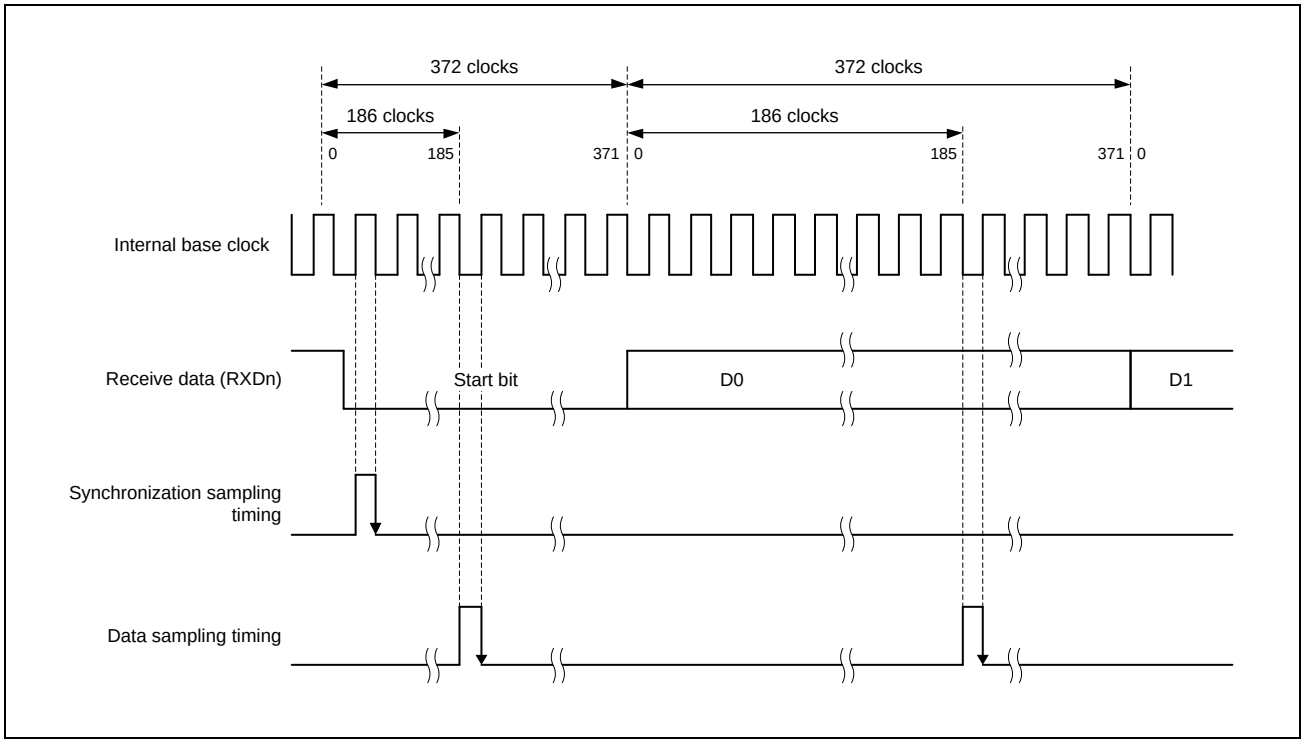



Figure 23.33 Receive Data Sampling Timing in Smart Card Interface Mode (When Clock Frequency is 372 Times the Bit Rate)

### 23.6.5 SCI Initialization (Smart Card Interface Mode)

Initialize the SCI following the example of flowchart shown in **Figure 23.34**.

Be sure to initialize the SCI before switching from transmission mode to reception mode and vice versa. Even if the RE bit is set to 0, the RDR register is not initialized.

To change reception mode to transmission mode, first check that reception has completed, and then initialize the SCI. At the end of initialization, set TE = 1 and RE = 0. Reception completion can be verified by reading the RXI request, ORER, or PER flag in SSR.

To change transmission mode to reception mode, first check that transmission has completed, and then initialize the SCI. At the end of initialization, set TE = 0 and RE = 1. Transmission completion can be verified by reading the TEND flag in SSR.

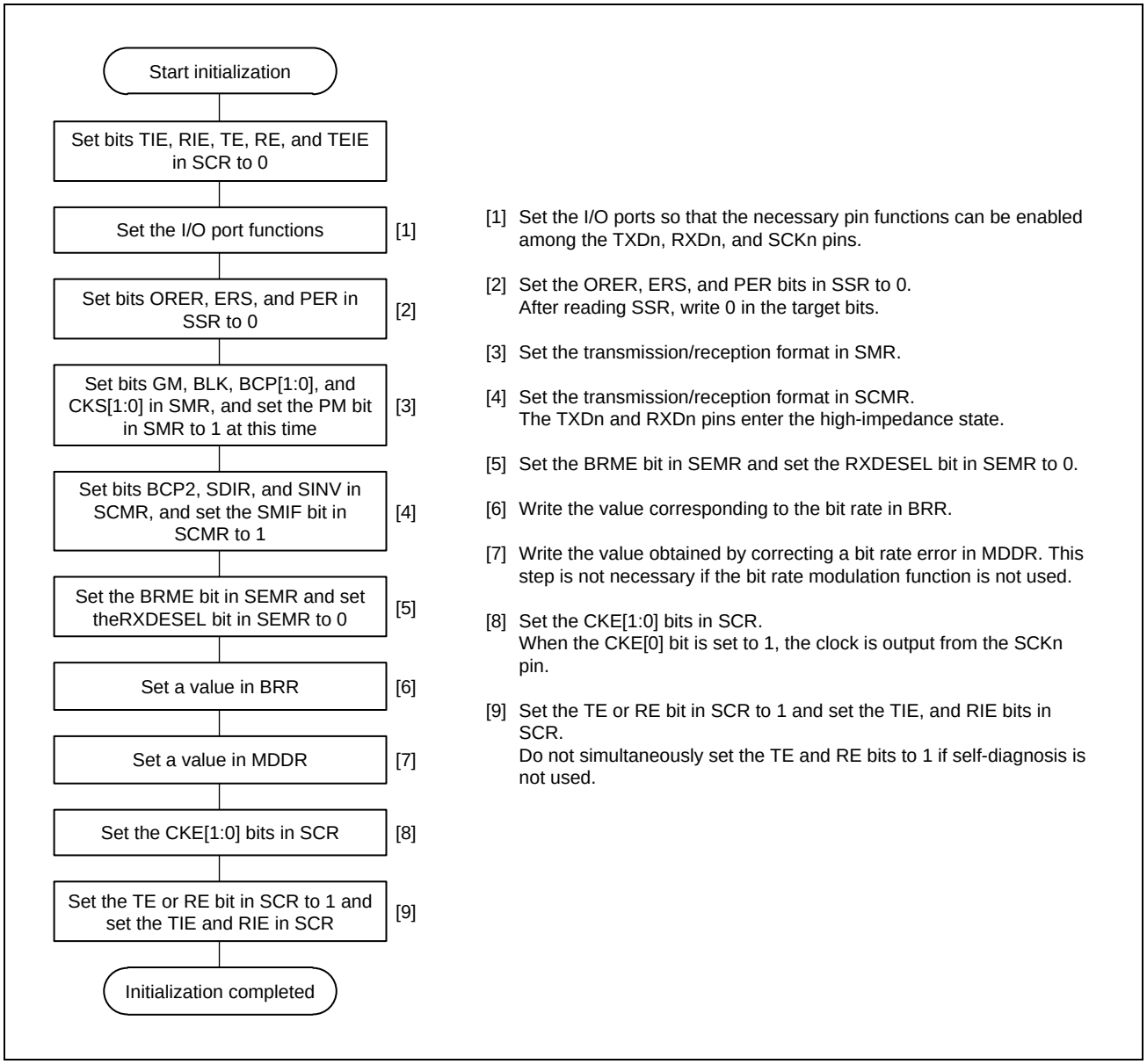



Figure 23.34 Example of SCI Initialization Flowchart (Smart Card Interface Mode)

### 23.6.6 Serial Data Transmission (Except in Block Transfer Mode)

Serial data transmission in smart card interface mode (except in block transfer mode), in that an error signal is sampled and data can be retransmitted, is different from that in non-smart card interface mode. **Figure 23.35** shows the data retransfer operation during transmission.

1. When an error signal from the receiver end is sampled after one-frame data has been transmitted, the ERS flag in SSR is set to 1. If the RIE bit in SCR is 1 at this time, an ERI interrupt request is generated. Clear the ERS flag to 0 before the next parity bit is sampled.
2. For a frame in which an error signal is received, the TEND flag in SSR is not set. Data is retransferred from TDR to TSR allowing automatic data retransmission.
3. If no error signal is returned from the receiver, the ERS flag is not set to 1.
4. In this case, the SCI judges that transmission of one-frame data (including retransfer) has been completed, and the TEND flag is set. If the TIE bit in SCR is 1 at this time, a TXI interrupt request is generated. Writing transmit data to TDR starts transmission of the next data.

**Figure 23.37** shows a sample flowchart of serial transmission. All the processing steps are automatically performed using a TXI interrupt request to activate the DMAC.

When the TEND flag in SSR is set to 1 in transmission, if the TIE bit in SCR is 1, a TXI interrupt request is generated. The DMAC is activated by a TXI interrupt request if the TXI interrupt request is specified as a source of DMAC activation beforehand, allowing transfer of transmit data. The TEND flag is automatically set to 0 when the DMAC transfers the data.

If an error occurs, the SCI automatically retransmits the same data. During this retransmission, the TEND flag is kept to 0 and the DMAC is not activated. Therefore, the SCI and DMAC automatically transmit the specified number of bytes, including retransmission in the case of error occurrence. However, since the ERS flag is not automatically cleared, set the RIE bit to 1 beforehand to enable an ERI interrupt request to be generated at error occurrence, and clear the ERS flag to 0. When transmitting/receiving data using the DMAC, be sure to make settings to enable the DMAC before making SCI settings.

For DMAC settings, refer to **Section 14, Direct Memory Access Controller**.

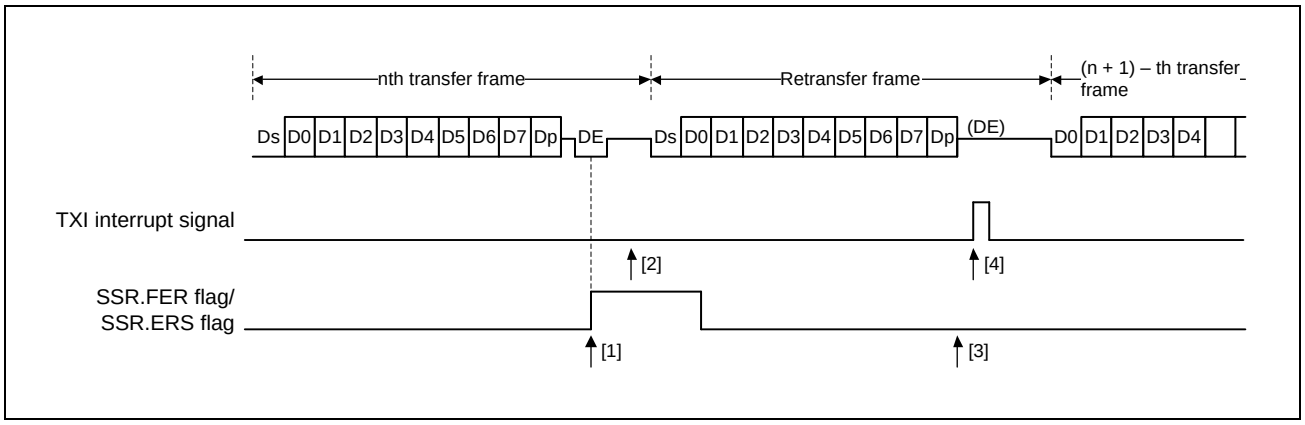



Figure 23.35 Data Retransfer Operation in SCI Transmission Mode

Note that the SSR.TEND flag is set in different timings depending on the GM bit setting in SMR. **Figure 23.36** shows the TEND flag generation timing.

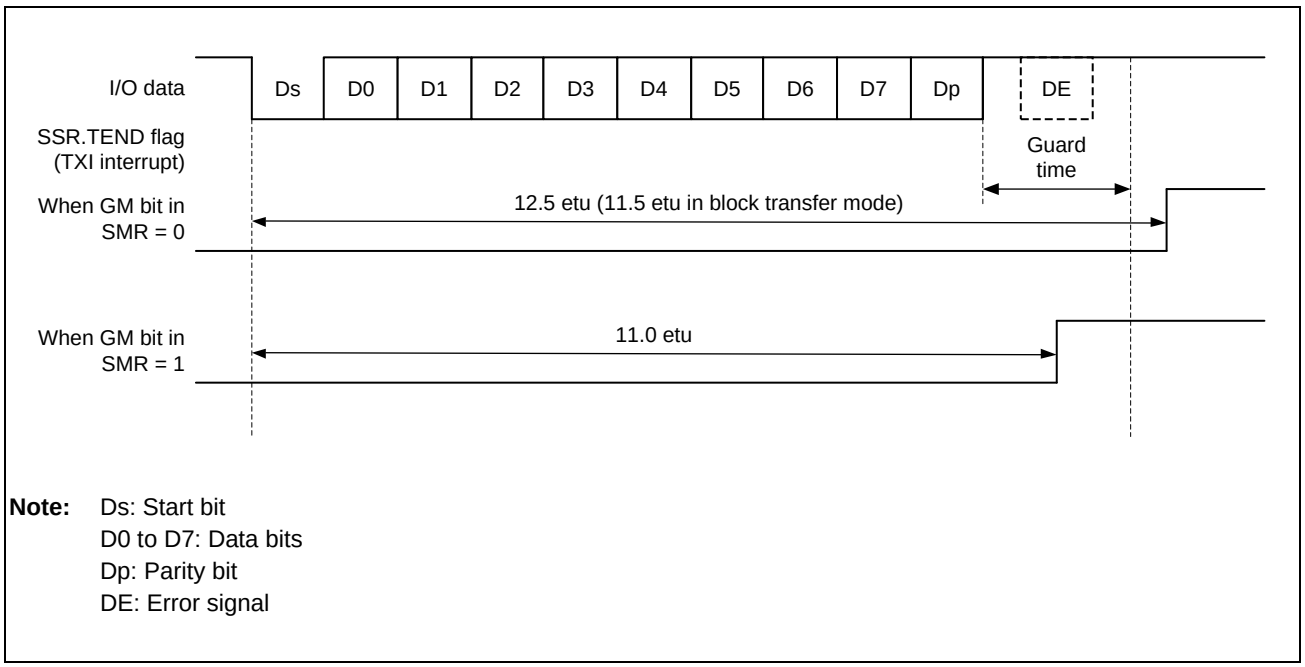



Figure 23.36 SSR.TEND Flag Generation Timing during Transmission

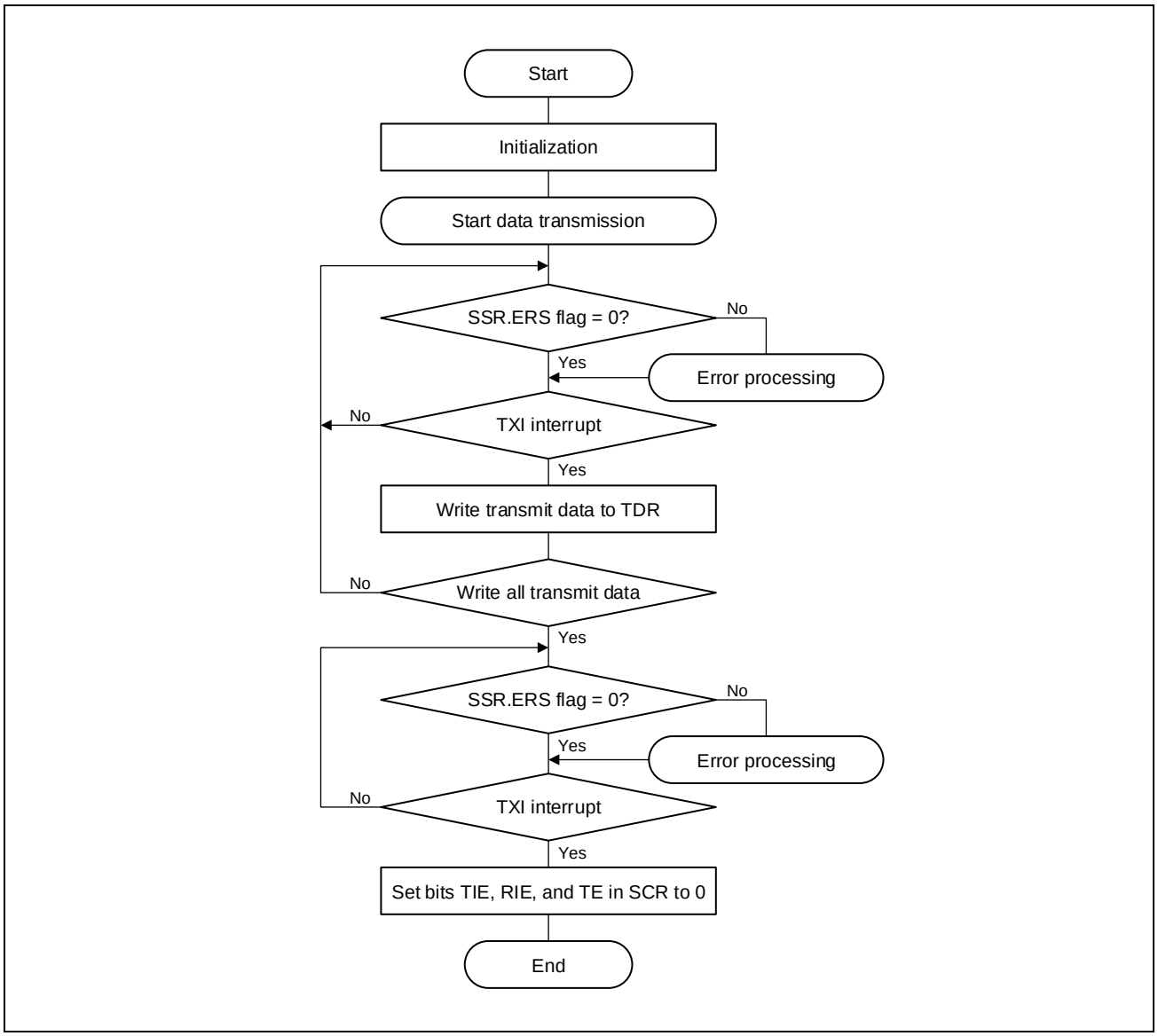



Figure 23.37 Sample Smart Card Interface Transmission Flowchart

### 23.6.7 Serial Data Reception (Except in Block Transfer Mode)

Serial data reception in smart card interface mode is similar to that in non-smart card interface mode. **Figure 23.38** shows the data retransfer operation in reception mode.

1. If a parity error is detected in receive data, the PER flag in SSR is set to 1. When the RIE bit in SCR is 1 at this time, an ERI interrupt request is generated. Clear the PER flag to 0 before the next parity bit is sampled.
2. For a frame in which a parity error is detected, no RXI interrupt is generated.
3. When no parity error is detected, the PER flag in SSR is not set to 1.
4. In this case, data is determined to have been received successfully. When the RIE bit in SCR is 1, an RXI interrupt request is generated.

**Figure 23.39** shows a sample flowchart for serial data reception. All the processing steps are automatically performed using an RXI interrupt request to activate the DMAC.

In reception, setting the RIE bit to 1 allows an RXI interrupt request to be generated. The DMAC is activated by an RXI interrupt request if the RXI interrupt request is specified as a source of DMAC activation beforehand, allowing transfer of receive data.

If an error occurs during reception and either the ORER or PER flag in SSR is set to 1, a receive error interrupt (ERI) request is generated. Clear the error flag after the error occurrence. If an error occurs, the DMAC is not activated and receive data is skipped. Therefore, the number of bytes of receive data specified in the DMAC is transferred.

Even if a parity error occurs and the PER flag is set to 1 during reception, receive data is transferred to RDR, thus allowing the data to be read.

When a reception is forcibly terminated by setting the RE bit in SCR to 0 during operation, read the RDR register because the received data which has not yet been read may be left in RDR.

NOTE

For operations in block transfer mode, refer to **Section 23.3, Operation in Asynchronous Mode**.

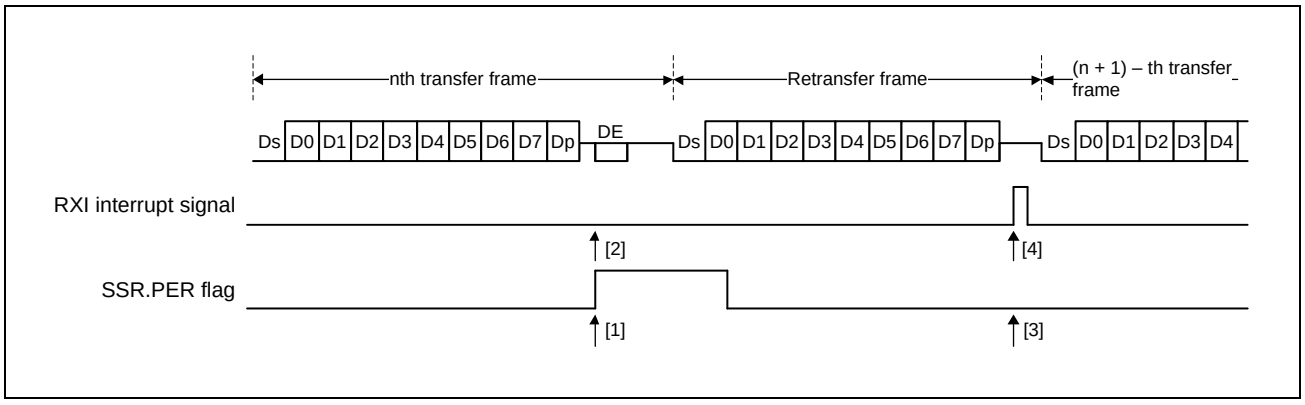



Figure 23.38 Data Retransfer Operation in SCI Reception Mode (Data Retransfer Operation during Reception)

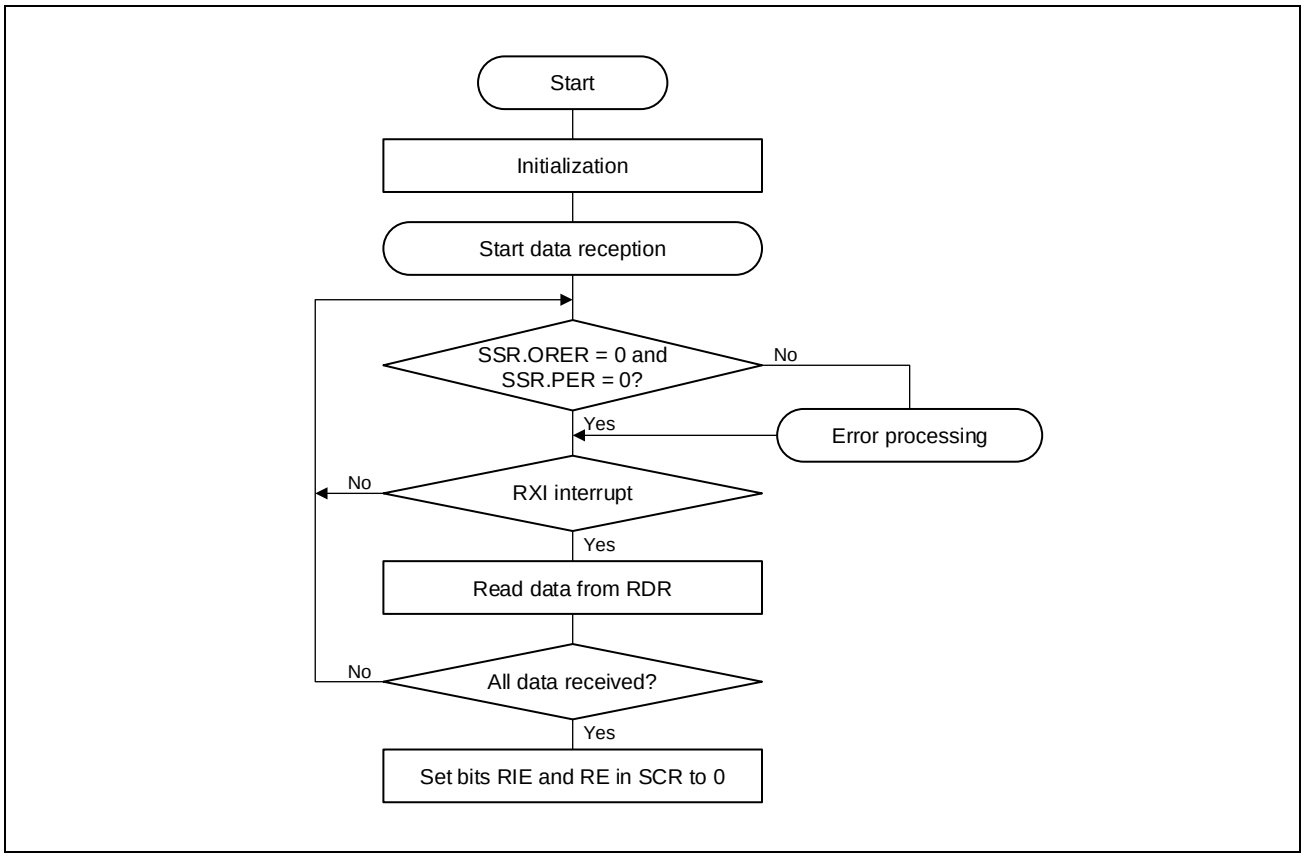



Figure 23.39 Sample Smart Card Interface Reception Flowchart

### 23.6.8 Clock Output Control

Clock output can be fixed using the CKE[1:0] bits in SCR when the GM bit in SMR is 1. Specifically, the minimum width of a clock pulse can be specified.

**Figure 23.40** shows an example of clock output fixing timing when the CKE[0] bit is controlled with GM = 1 and CKE[1] = 0.

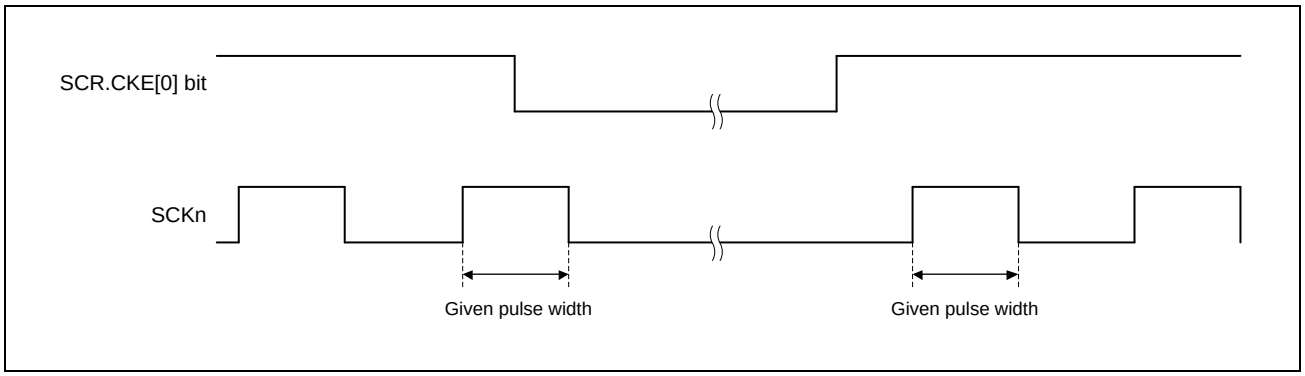



Figure 23.40 Clock Output Fixing Timing

At power-on, use the following procedure to secure the appropriate clock duty cycle.

#### (1) At Power-On

To secure the appropriate clock duty cycle simultaneously with power-on, use the following procedure.

1. Initially, port input is enabled in the high-impedance state. To fix the potential level, use a pull-up or pull-down resistor.
2. Fix the SCKn pin to the specified output by setting the SCR.CKE[1] bit and I/O port functions.
3. Set SMR and SCMR to enable smart card interface mode.
4. Set the SCR.CKE[0] bit to 1 to start clock output.

## 23.7 Noise Cancellation Function

**Figure 23.41** shows the configuration of the noise filter used for noise cancellation. The noise filter consists of two stages of flip-flop circuits and a match-detection circuit. When the level on the pin matches in three consecutive samples taken at the set sampling interval, the matching level continues to be conveyed internally until the level on the pin again matches in three consecutive samples.

In asynchronous mode, the noise cancellation function can be applied on the RXDn input signal. The period of the base clock (1/16th of a bit-period when SEMR.ABCS = 0 and 1/8th of a bit-period when SEMR.ABCS = 1) is the sampling interval.

If the base clock is stopped with the noise filter enabled and then the clock input is started again, the noise filter operation resumes from where the clock was stopped. If SCR.TE and SCR.RE are set to 0 during base clock input, all of the noise filter flip-flop values are initialized to 1. Accordingly, if the input data is 1 when reception operation resumes, it is determined that a level match is detected and is conveyed to the internal signal. When the level being input corresponds to 0, the initial output of the noise filter is retained until the level matches in three consecutive samples.

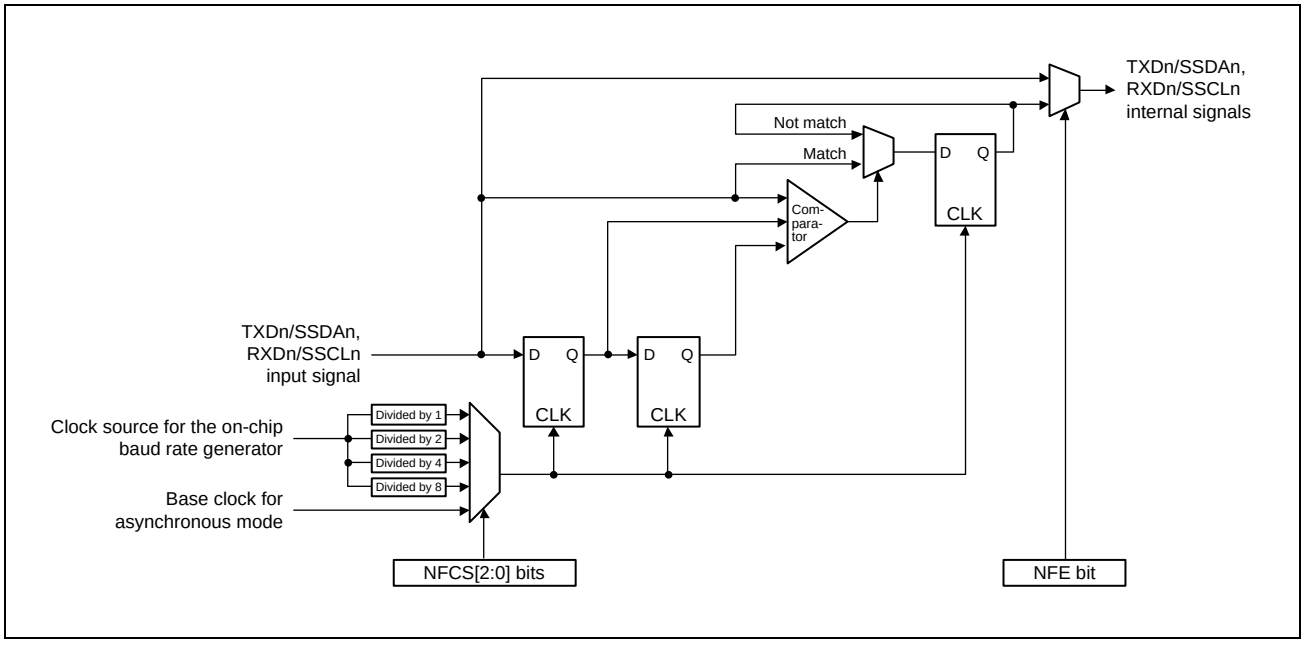



Figure 23.41 Block Diagram of Digital Noise Filter Circuit

## 23.8 Interrupt Sources

### 23.8.1 Buffer Operations for TXI and RXI Interrupts

If the conditions for a TXI and RXI interrupt are satisfied while the interrupt status flag in the interrupt controller is 1, the interrupt controller does not output the interrupt request but retains it internally (with a capacity for retention of one request per source).

When the value of the interrupt status flag in the interrupt controller becomes 0, the interrupt request retained within the interrupt controller is output. The internally retained interrupt request is automatically discarded once the actual interrupt is output. Clearing of the corresponding interrupt enable bit (the TIE or RIE bit in the SCR) can also be used to discard an internally retained interrupt request.

### 23.8.2 Interrupts in Asynchronous Mode and Clock Synchronous Mode

**Table 23.20** lists interrupt sources in asynchronous mode and clock synchronous mode. A different interrupt vector is assigned to each interrupt source, and individual interrupt sources can be enabled or disabled with the enable bits in SCR. If the SCR.TIE bit is 1, a TXI interrupt request is generated when transmit data is transferred from the TDR or TDRL register*[1] to the TSR. A TXI interrupt request can also be generated by setting the SCR.TE bit to 1 after setting the SCR.TIE bit to 1 or by using a single instruction to set the SCR.TE and SCR.TIE bit to 1 at the same time. A TXI interrupt request can activate the DMAC to handle data transfer.

A TXI interrupt request is not generated by setting the SCR.TE bit to 1 while the setting of the SCR.TIE bit is 0 or by setting the SCR.TIE bit to 1 while the setting of the SCR.TE bit is 1.*[1]

When new data is not written by the time of transmission of the last bit of the current transmit data and the setting of the SCR.TEIE bit is 1, the SSR.TEND flag becomes 1 and a TEI interrupt request is generated. Furthermore, when the setting of the SCR.TE bit is 1, the SSR.TEND flag retains the value 1 until further transmit data are written to the TDR or TDRL register*[1], and setting the SCR.TEIE bit to 1 leads to the generation of a TEI interrupt request.

Writing data to the TDR or TDRL register*[1] leads to clearing of the SSR.TEND flag and, after a certain time, discarding of the TEI interrupt request.

If the SCR.RIE bit is 1, an RXI interrupt request is generated when received data is stored in the RDR. An RXI interrupt request can activate the DMAC to handle data transfer.

Setting of any from among the ORER, FER, and PER flags in the SSR to 1 while the SCR.RIE bit is 1 leads to the generation of an ERI interrupt request. An RXI interrupt request is not generated at this time. Clearing all three flags (ORER, FER, and PER) leads to discarding of the ERI interrupt request.

**Note 1.** In the case where asynchronous mode and 9-bit data length are selected

NOTE

To temporarily prohibit TXI interrupts at the time of transmission of the last of the data and so on when you wish a new round of transmission to start after handling of the transmission-completed interrupt, control prohibiting and permitting of the interrupt by using the interrupt request enable bit in the interrupt controller rather than using the SCR.TIE bit. This can prevent the suppression of TXI interrupt requests in the transfer of new data.

Table 23.20 Interrupt Sources

| Name | Interrupt Source | Interrupt Flag | DMAC Activation | Priority |
|---|---|---|---|---|
| ERI | Receive error | ORER, FER, or PER | Not possible | High |
| RXI | Receive data full | — | Possible | |
| TXI | Transmit data empty | — | Possible | |
| TEI | Transmit end | TEND | Not possible | Low |

### 23.8.3 Interrupts in Smart Card Interface Mode

**Table 23.21** lists interrupt sources in smart card interface mode. A transmit end interrupt (TEI) request cannot be used in this mode.

Table 23.21 SCI Interrupt Sources

| Name | Interrupt Source | Interrupt Flag | DMAC Activation | Priority |
|---|---|---|---|---|
| ERI | Receive error or error signal detection | ORER, PER, or ERS | Not possible | High |
| RXI | Receive data full | — | Possible | |
| TXI | Transmit data empty | TEND | Possible | Low |

Data transmission/reception using the DMAC is also possible in smart card interface mode, similar to in the normal SCI mode. In transmission, when the TEND flag in SSR is set to 1, a TXI interrupt request is generated. This TXI interrupt request activates the DMAC allowing transfer of transmit data if the TXI request is specified beforehand as a source of DMAC activation. The TEND flag is automatically set to 0 when the DMAC transfers the data.

If an error occurs, the SCI automatically retransmits the same data. During the retransmission, the TEND flag is kept to 0 and the DMAC is not activated. Therefore, the SCI and DMAC automatically transmit the specified number of bytes, including retransmission in the case of error occurrence. However, the ERS flag in SSR is not automatically cleared to 0 at error occurrence. Therefore, the ERS flag must be cleared by previously setting the RIE bit in SCR to 1 to enable an ERI interrupt request to be generated at error occurrence.

When transmitting/receiving data using the DMAC, be sure to make settings to enable the DMAC before making SCI settings. For DMAC settings, refer to **Section 14, Direct Memory Access Controller**.

In reception, an RXI interrupt request is generated when receive data is set to RDR. This RXI interrupt request activates the DMAC allowing transfer of receive data if the RXI request is specified beforehand as a source of DMAC activation. If an error occurs, the error flag is set. Therefore, the DMAC is not activated and an ERI interrupt request is issued to the CPU instead; the error flag must be cleared.

## 23.9 Usage Notes

### 23.9.1 Break Detection and Processing

When a framing error is detected, a break can be detected by reading the RXDn pin value directly. In a break, the input from the RXDn pin becomes all 0s, and so the FER flag in SSR is set to 1 (framing error has occurred), and the PER flag in SSR may also be set to 1 (parity error has occurred). The SCI continues the receive operation even after a break is received. Therefore, note that even if the FER flag is set to 0 (no framing error occurred), it will be set to 1 again. When the SEMR.RXDESEL bit is 1, the SCI sets the SSR.FER flag to 1 and stops receiving operation until a start bit of the next data frame is detected. If the SSR.FER flag is set to 0 at this time, the SSR.FER flag retains 0 during the break.

When the RXDn pin is set to 1 and the break ends, detecting the beginning of the start bit at the first falling edge of the RXDn pin allows the SCI to start the receiving operation.

### 23.9.2 Mark State and Production of Breaks

When the SCR.TE bit is 0 (serial transmission is disabled), setting the I/O port function makes selection of the level and direction (input or output) of the TXDn pin possible. If this is done, the TXDn pin can be placed in the mark state to send a break at the time of data transmission. Until the SCR.TE bit is set to 1 (serial transmission is enabled), the I/O port function is used to set the TXDn pin to output 1 and set the pin mode to a general I/O port pin, and thus place the communication line in the mark state (state of having the value 1). On the other hand, to output a break at the time of data transmission, set the TXDn pin to output 0 and make the pin mode settings for a general I/O port pin. When the SCR.TE bit is set to 0, the transmitter is initialized regardless of the current state of transmission.

### 23.9.3 Receive Error Flags and Transmit Operations (Clock Synchronous Mode)

Transmission cannot be started when a receive error flag (ORER) in SSR is set to 1, even if data is written to TDR. Be sure to set the receive error flags to 0 before starting transmission. Note also that the receive error flags cannot be set to 0 even if the RE bit in SCR is set to 0 (serial reception is disabled).

### 23.9.4 Writing Data to TDR

Data can be written to TDR, TDRH, and TDRL. However, if new data is written to TDR, TDRH, and TDRL when transmit data is remaining in TDR, TDRH, and TDRL, the previous data in TDR, TDRH, and TDRL is lost because it has not been transferred to TSR yet. Be sure to write transmit data to TDR, TDRH, and TDRL in the TXI interrupt request handling routine.

### 23.9.5 Restrictions on Clock Synchronous Transmission (Clock Synchronous Mode)

When the external clock source is used as a synchronization clock, the following restrictions apply.

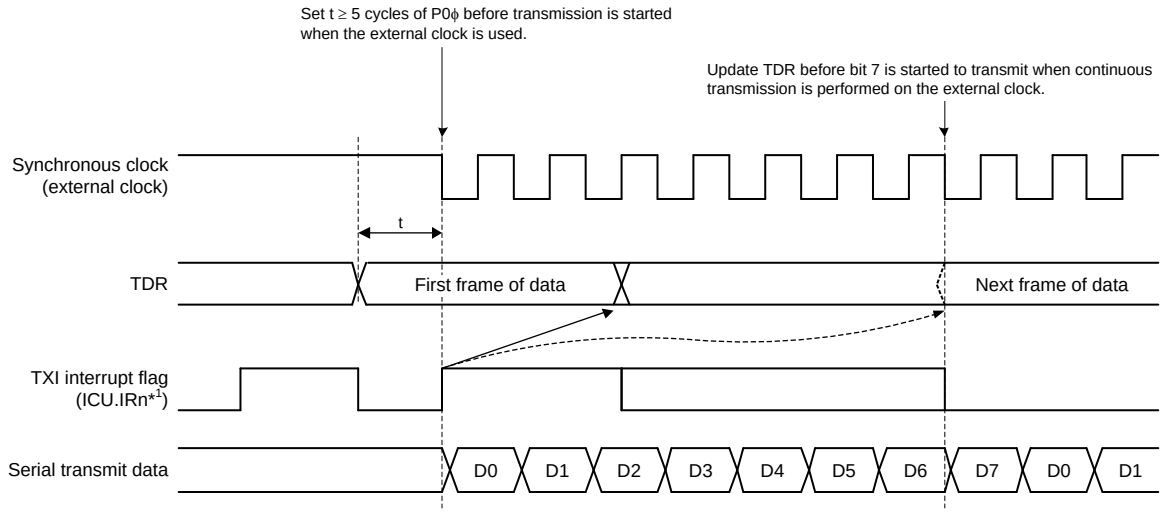
#### (1) Start of transmission

Update TDR by the CPU or DMAC and wait for at least five P0ϕ cycles before allowing the transmit clock to be input (see **Figure 23.42**).

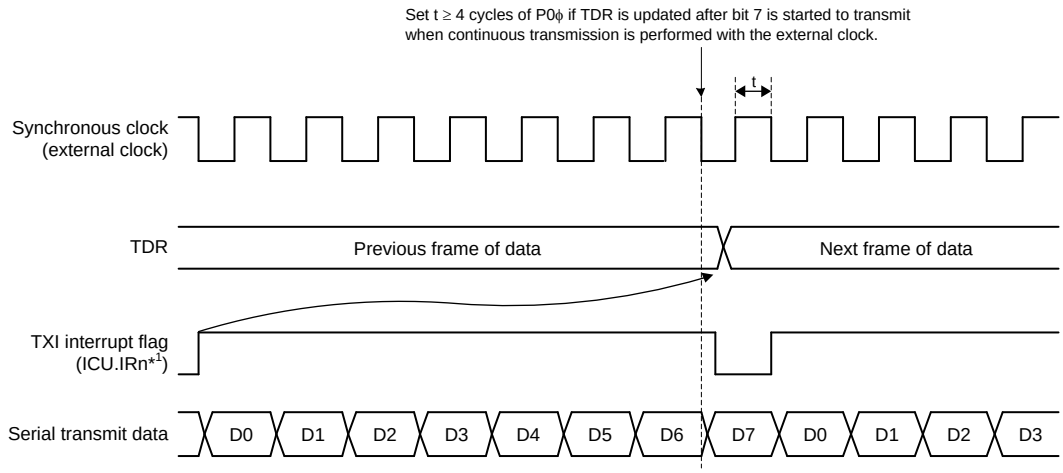
#### (2) Continuous transmission

(a) Write the next transmit data to TDR or TDRL before the falling edge of the transmit clock (bit 7) (see **Figure 23.42**).

(b) When updating TDR after bit 7 has started to transmit, update TDR while the synchronization clock is in the low-level period, and set the high-level width of the transmit clock (bit 7) to four P0ϕ cycles or longer (see **Figure 23.42**).




(a) Start of transmission and Continuous transmission (1)




(b) Continuous transmission (2)

Note 1. Refer to **Section 8, Interrupt Controller** for details on the corresponding interrupt vector number.

Figure 23.42 Restrictions on Use of External Clock in Clock Synchronous Transmission

### 23.9.6 Restrictions on Using DMAC

When using the DMAC or DTC to read RDR, RDRH, and RDRL, be sure to set the receive data full interrupt (RXI) as the activation source of the relevant SCI.

### 23.9.7 Notes on Starting Transfer

At the point where transfer starts when the interrupt status flag (IRn.IR bit) in the interrupt controller is 1, follow the procedure below to clear interrupt requests before permitting operations (by setting the SCR.TE or SCR.RE bit to 1). For details on the interrupt status flag, refer to **Section 8, Interrupt Controller**.

- Confirm that transfer has stopped (the setting of the SCR.TE or SCR.RE bits is 0).
- Set the corresponding interrupt enable bit (SCR.TIE or SCR.RIE) to 0.
- Read the corresponding interrupt enable bit (SCR.TIE or SCR.RIE bit) to check that it has become 0.
- Set the interrupt status flag (IRn.IR bit) in the interrupt controller to 0.

### 23.9.8 External Clock Input in Clock Synchronous Mode

In clock synchronous mode, the external clock SCKn must be input as follows:

High-pulse period, low-pulse period = 2 P0ϕ cycles or more, period = 6 P0ϕ cycles or more

### 23.9.9 Note on Transmit Enable Bit (TE bit)

When setting the SCR.TE bit to 0 (serial transmission is disabled) while the pin function is “TXDn”, output of the pin becomes high impedance.

Prevent the TXDn line from becoming high impedance by any of the following ways:

(1) Connect a pull-up resistor to the TXDn line.

(2) Change the pin function to “general-purpose I/O port, output” before setting the SCR.TE bit to 0. Set the SCR.TE bit to 1 before changing the pin function to “TXDn”.

## 23.10 IrDA Communications

The channel 0 serial communications interface (SCI) is capable of working with the Infrared Data Association (IrDA) module to handle transfer through IrDA communications in conformance with version 1.0 of the IrDA protocol.

Using the IRE bit in IRCR to enable the IrDA function leads to encoding and decoding of the SCI0_TXD and SCI0_RXD signals of the channel 0 serial communications interface, respectively, to convert them to and from waveforms in conformance with version 1.0 of the IrDA protocol. Infrared transfer in conformance with version 1.0 of the IrDA protocol can be handled by connecting a transceiver for use in infrared communications.

In version 1.0 of the IrDA protocol, IrDA transfer starts at a transfer rate of 9600 bps. The transfer rate is changeable during transfer as required. The IrDA module itself does not have a function for automatically changing the transfer rate. Accordingly, change the transfer rate by changing that for the serial communications interface.

A block diagram and pin configuration for IrDA transfer are shown in **Figure 23.43** and **Table 23.22**, respectively.

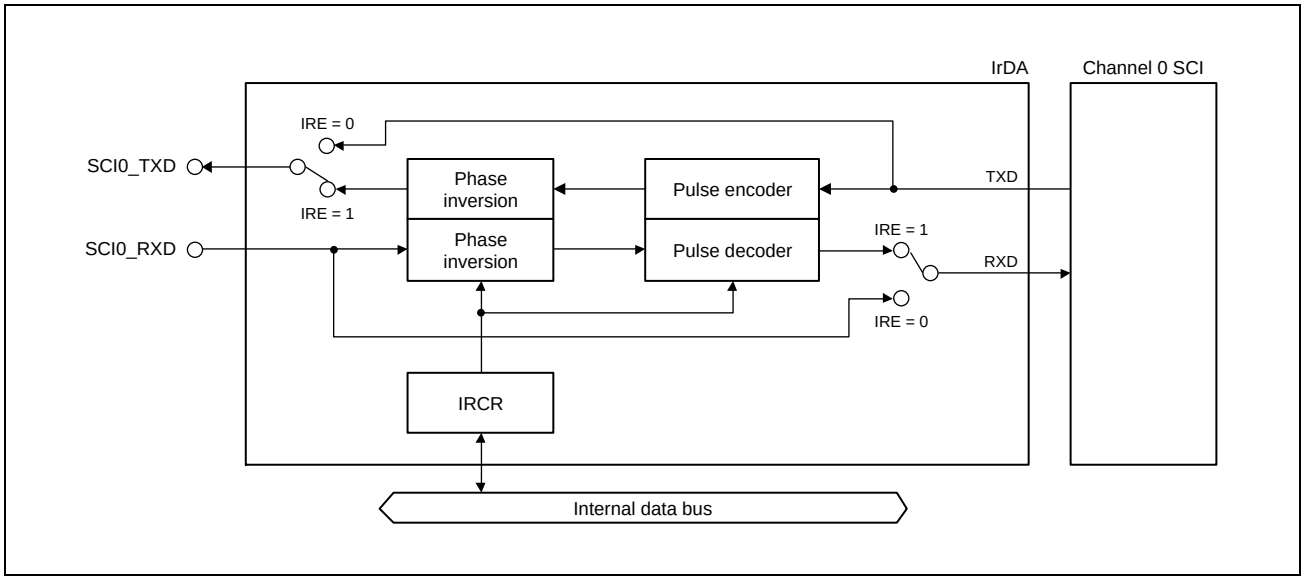



Figure 23.43 Block Diagram

Table 23.22 Pin Configuration

| Pin Name | Symbol | I/O | Function |
|---|---|---|---|
| IrDA data transmission pin | SCI0_TXD | Output | IrDA data transmission output |
| IrDA data reception pin | SCI0_RXD | Input | IrDA data reception input |

## 23.11 Description of the IrDA Register

**Table 23.23** shows the register configuration.

BASE Address (Cortex-A55 Address Space): H’0_1004_CC00

Table 23.23 Register Configuration

| Register Name | Abbreviation | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|
| IrDA Control Register | IRCR | R/W | H’00 | H’0_1004_CC00 | 8 |

### 23.11.1 IrDA Control Register (IRCR)

IRCR sets the operation of the IrDA module.

| Bit | b7 | b6 | b5 | b4 | b3 | b2 | b1 | b0 |
|---|---|---|---|---|---|---|---|---|
| | IRE | IRCKS[2:0] | | | IRTXINV | IRRXINV | — | — |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | | | | | | | | |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| b1, b0 | — | | R | Reserved<br>These bits are read as 0. The write value should be 0. |
| b2 | IRRXINV | | R/W | SCI_RXD0 Data Level Switching<br>Sets inversion of the logic level of inputs on SCI0_RXD.<br>0: The input on SCI0_RXD is used without change as the received data.<br>1: The input on SCI0_RXD is inverted before it is treated as the received data. |
| b3 | IRTXINV | | R/W | SCI_TXD0 Data Level Switching<br>Sets inversion of the logic level of outputs on SCI0_TXD.<br>0: Data for transmission is output without change to SCI0_TXD.<br>The pulse-width setting in the IRCKS bits determines the period over which the pulses are at the high level.<br>1: Data for transmission is inverted before being output to SCI0_TXD.<br>The pulse-width setting in the IRCKS bits determines the period over which the pulses are at the low level. |
| b6 to b4 | IRCKS[2:0] | | R/W | IrDA Clock Select<br>Selects the pulse width for use in encoding output pulses from SCI0_TXD when the setting of the IRE bit is 1.<br>000: Bit rate × 3/16<br>Settings other than above are prohibited. |
| b7 | IRE | | R/W | IrDA Enable<br>Sets the function for use with the SCI0_TXD and SCI0_RXD pins as normal serial or IrDA.<br>0: Disables the IrDA function.<br>Data from the TXD pin of the channel 0 serial communications interface is output without change to SCI0_TXD.<br>Data on SCI0_RXD is output without change to the RXD pin of the channel 0 serial communications interface.<br>1: Enables the IrDA function.<br>Data from the TXD pin of the channel 0 serial communications interface is encoded before being output through SCI0_TXD.<br>Data on SCI0_RXD is decoded before being output to the RXD pin of the channel 0 serial communications interface. |

# 23.12 IrDA Operation

## 23.12.1 Flow of Settings for IrDA Operation

Set up IrDA operation by following the procedure listed below.

1. Make the general-purpose I/O port-pin settings.
2. Set the IRCR register.
3. Set the registers related to the serial communications interface.

## 23.12.2 Transmission

In transmission with the IrDA function enabled, serial data (UART frames of data) from the TXD pin of the serial communications interface are converted to IR frames (see **Figure 23.44**). If the IRTXINV bit is set to 0, when the value of the serial data is 0, a high-level pulse with the width equivalent to three sixteenths of the bit period is output to the SCI0_TXD pin by default. In the IrDA protocol, the specifiable high-level pulse widths are from a minimum of 1.41 µs to a maximum of (3/16 + 2.5%) × bit period or (3/16 × bit period) + 1.08 µs. On the other hand, if the value of the serial data is 1, no high-level pulse is output.

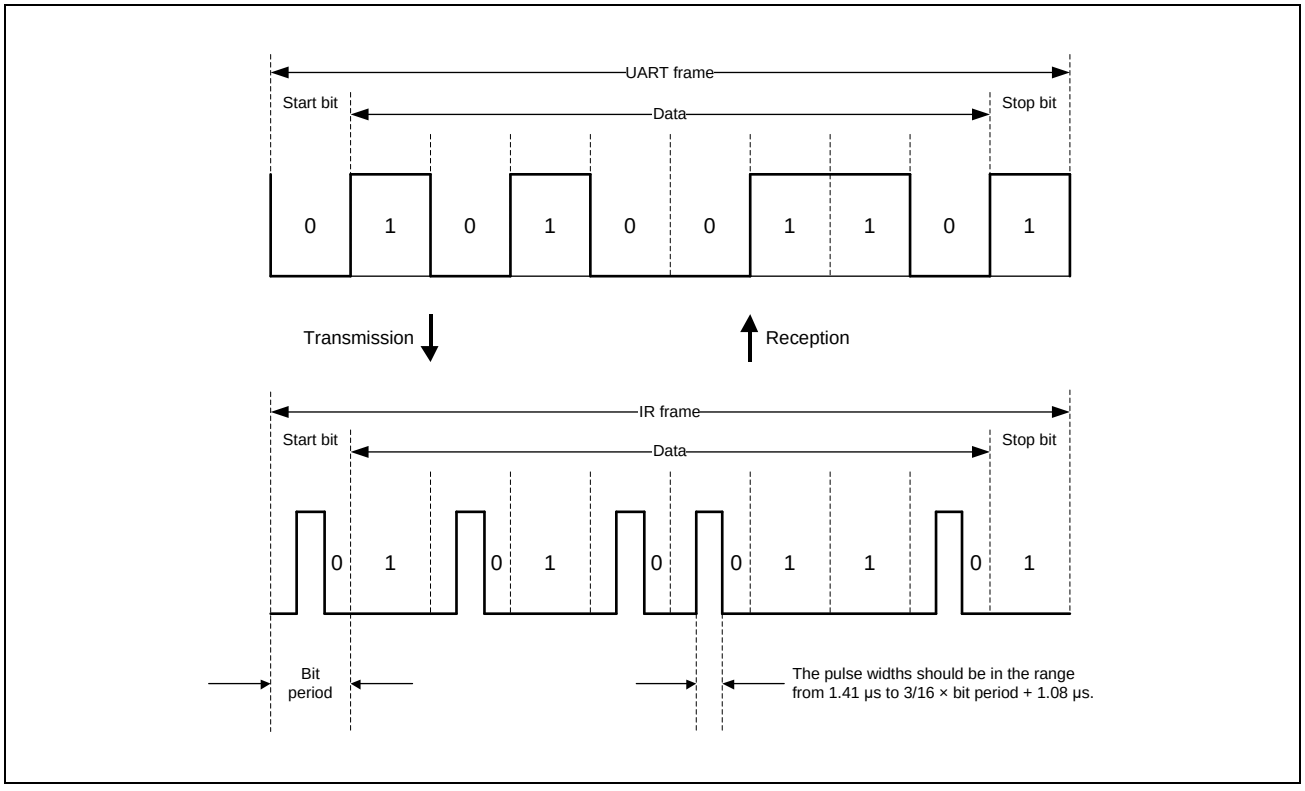



Figure 23.44 Example of the Operation in IrDA Transfer

### 23.12.3 Reception

In reception with the IrDA function enabled, IR frames of data from the SCI0_RXD pin are converted to serial data and output to the RXD pin of the serial communications interface. If the setting of the IRRXINV bit is 0, the value 0 is output when a high-level pulse is detected and the value 1 is output when no high-level pulse is detected during each bit period. Note that pulses with a width shorter than the minimum pulse width of 1.41 μs cannot be recognized.

### 23.12.4 Relationship between Bit Rate and High-Level Pulse Width

The relationship between the Bit Rate during transmission and the High-Level Pulse is shown in **Table 23.24**.

Table 23.24 Upper Row: Bit Rate (bps); Lower Row: Bit Period × 3/16 (μs)

| | | | | | |
|---|---|---|---|---|---|
| 2400 | 9600 | 19200 | 38400 | 57600 | 115200 |
| 78.13 | 19.53 | 9.77 | 4.88 | 3.26 | 1.63 |

## 23.13 Note on IrDA Usage

### 23.13.1 Minimum Pulse Width in Reception

Pulses with a width shorter than the minimum pulse width of 1.41 μs cannot be recognized

### 23.13.2 Base Clock in Asynchronous Mode for the Serial Communications Interface

The IrDA module works with the serial communications interface by receiving the base clock with a frequency 16 times the bit rate for communications from the interface. For the serial communications interface, either 16 or 8 cycles of the base clock are selectable as the bit period. On the other hand, the IrDA module is only capable of operation when the setting for the bit period of the serial communications interface is 16 cycles of the base clock.

# 24. Renesas Serial Peripheral Interface

This LSI circuit includes three independent Renesas serial peripheral interfaces.

This module is capable of full-duplex synchronous serial communication.

## 24.1.1 Features

This module has the following features.

### SPI transfer functions

- Use of MOSI (master out/slave in), MISO (master in/slave out), SSL (slave select), and RSPCK (SPI clock) signals allows for serial communications through SPI operation (four-wire method).
- Capable of serial communications in master/slave mode
- Supports mode fault error detection (only in SPI slave mode)
- Supports overrun error detection (only in SPI slave mode)
- Switching of the polarity of the serial transfer clock
- Switching of the clock phase of serial transfer

### Data format

- MSB-first/LSB-first selectable
- Transfer bit-length is selectable as 8, 16, or 32 bits.

### Bit rate

- RSPCK can be divided by a maximum of 4096 in master mode
- RSPCK can be generated by dividing P0ϕ by the on-chip baud rate generator.
- An externally input clock can be used as a serial clock.

### Buffer configuration

- 8 bytes for transmission and 32 bytes for reception

### SSL control function

- One SSL signal for each channel
  - In master mode, outputs SSL signal.
  - In slave mode, inputs SSL signal.
- Controllable delay from SSL output assertion to RSPCK operation (RSPCK delay)
  - Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units)
- Controllable delay from RSPCK stoppage to SSL output negation (SSL negation delay)
  - Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units)
- Controllable wait for next-access SSL output assertion (next-access delay)
  - Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units)
- Function for changing SSL polarity

### Control in master transfer

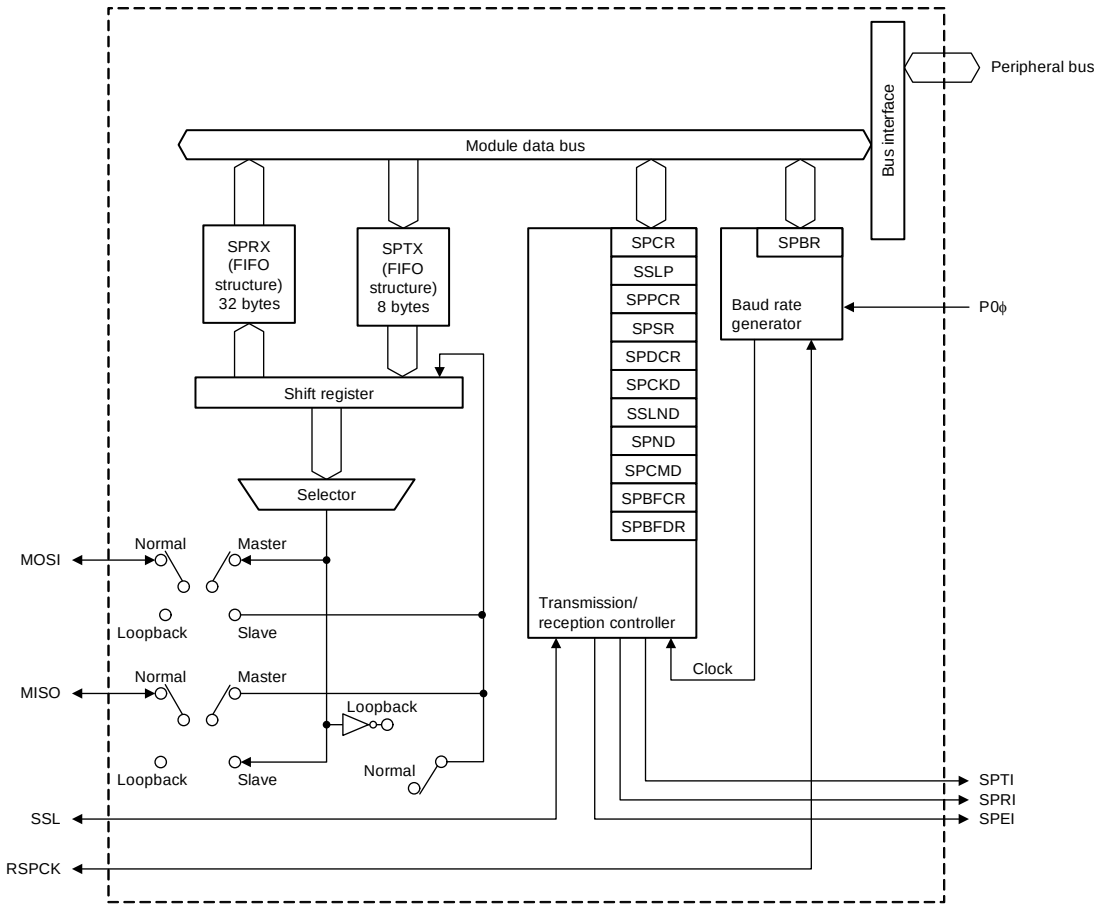
- A transfer of up to four commands can be executed sequentially in looped execution.
- For each command, the following can be set:
  - SSL signal value, bit rate, RSPCK polarity/phase, transfer data length, LSB/MSB first, burst, RSPCK delay, SSL negation delay, and next-access delay.
  - A transfer can be initiated by writing to the transmit buffer.
  - A transfer can be initiated by clearing the SPTEF bit.
  - MOSI signal value specifiable in SSL negation

### Interrupt sources

- Maskable interrupt sources:
  - Receive interrupt (receive buffer full)
  - Transmit interrupt (transmit buffer empty)
  - Error interrupt (mode fault, overrun)

### Others

- Provides loop back mode
- Provides a function for disabling (initializing) this module




[Legend]

| | | | |
|---|---|---|---|
| SPCR: | Control register | SPCMD: | Command register |
| SSLP: | Slave select polarity register | SPBR: | Bit rate register |
| SPPCR: | Pin control register | SPTX: | Transmission buffer (Data register write side) |
| SPSR: | Status register | SPRX: | Receive buffer (Data register read side) |
| SPSCR: | Sequence control register | SPBFCR: | Buffer control register |
| SPSSR: | Sequence status register | SPBFDR: | Buffer data count setting register |
| SPDCR: | Data control register | SPTI: | Transmit interrupt |
| SPCKD: | Cock delay register | SPRI: | Receive interrupt |
| SSLND: | Slave select negate delay register | SPEI: | Error interrupt |
| SPND: | Next-access delay register | | |

Figure 24.1 Block Diagram (for One Channel)

## 24.2 Input/Output Pins

**Table 24.1** shows the pin configuration. This module automatically switches the input/output direction of the SSL pin. SSL is set as an output in master mode and as an input in slave mode. Pins RSPCK, MOSI, and MISO are automatically set as inputs or outputs according to the setting of master or slave and the level input on SSL (see **Section 24.4.2, Pin Control**).

Table 24.1 Pin Configuration

| Channel | Pin Name | Pin Name | I/O | Function |
|---|---|---|---|---|
| 0 | Clock pin | RSPI0_CK | I/O | Clock input/output |
| | Master transmit data pin | RSPI0_MOSI | I/O | Master transmit data |
| | Slave transmit data pin | RSPI0_MISO | I/O | Slave transmit data |
| | Slave select 0 pin | RSPI0_SSL | I/O | Slave selection |
| 1 | Clock pin | RSPI1_CK | I/O | Clock input/output |
| | Master transmit data pin | RSPI1_MOSI | I/O | Master transmit data |
| | Slave transmit data pin | RSPI1_MISO | I/O | Slave transmit data |
| | Slave select 0 pin | RSPI1_SSL | I/O | Slave selection |

**Note:** In the description of the pins, the channel is omitted, and pin names are described as RSPCK, MOSI, MISO, and SSL.

## 24.3 Register Descriptions

**Table 24.2** shows the register configuration. The address of the RSPI register is represented by the offset address from the base address. RSPI base address is as follows:

Base address: H'0_1004_0000 (Overall Address Space)

Table 24.2 Register Configuration

| Channel | Register Name | Abbreviation*1 | R/W | Initial Value | Offset Address | Access Size |
|---|---|---|---|---|---|---|
| 0 | Control register_0 | SPCR_0 | R/W | H'00 | H'AC00 | 8 |
| | Slave select polarity register_0 | SSLP_0 | R/W | H'00 | H'AC01 | 8 |
| | Pin control register_0 | SPPCR_0 | R/W | H'00 | H'AC02 | 8 |
| | Status register_0 | SPSR_0 | R/(W)*2 | H'60 | H'AC03 | 8 |
| | Data register_0 | SPDR_0 | R/W | Undefined | H'AC04 | 8, 16, 32 |
| | Sequence control register_0 | SPSCR_0 | R/W | H'00 | H'AC08 | 8 |
| | Sequence status register_0 | SPSSR_0 | R | H'00 | H'AC09 | 8 |
| | Bit rate register_0 | SPBR_0 | R/W | H'FF | H'AC0A | 8 |
| | Data control register_0 | SPDCR_0 | R/W | H'20 | H'AC0B | 8 |
| | Clock delay register_0 | SPCKD_0 | R/W | H'00 | H'AC0C | 8 |
| | Slave select negation delay register_0 | SSLND_0 | R/W | H'00 | H'AC0D | 8 |
| | Next-access delay register_0 | SPND_0 | R/W | H'00 | H'AC0E | 8 |
| | Command register0_0 | SPCMD0_0 | R/W | H'070D | H'AC10 | 16 |
| | Command register1_0 | SPCMD1_0 | R/W | H'070D | H'AC12 | 16 |
| | Command register2_0 | SPCMD2_0 | R/W | H'070D | H'AC14 | 16 |
| | Command register3_0 | SPCMD3_0 | R/W | H'070D | H'AC16 | 16 |
| | Buffer control register_0 | SPBFCR_0 | R/W | H'00 | H'AC20 | 8 |
| | Buffer data count setting register_0 | SPBFDR_0 | R | H'0000 | H'AC22 | 16 |
| 1 | Control register_1 | SPCR_1 | R/W | H'00 | H'B000 | 8 |
| | Slave select polarity register_1 | SSLP_1 | R/W | H'00 | H'B001 | 8 |
| | Pin control register_1 | SPPCR_1 | R/W | H'00 | H'B002 | 8 |
| | Status register_1 | SPSR_1 | R/(W)*2 | H'60 | H'B003 | 8 |
| | Data register_1 | SPDR_1 | R/W | Undefined | H'B004 | 8, 16, 32 |
| | Sequence control register_1 | SPSCR_1 | R/W | H'00 | H'B008 | 8 |
| | Sequence status register_1 | SPSSR_1 | R | H'00 | H'B009 | 8 |
| | Bit rate register_1 | SPBR_1 | R/W | H'FF | H'B00A | 8 |
| | Data control register_1 | SPDCR_1 | R/W | H'20 | H'B00B | 8 |
| | Clock delay register_1 | SPCKD_1 | R/W | H'00 | H'B00C | 8 |
| | Slave select negation delay register_1 | SSLND_1 | R/W | H'00 | H'B00D | 8 |
| | Next-access delay register_1 | SPND_1 | R/W | H'00 | H'B00E | 8 |
| | Command register0_1 | SPCMD0_1 | R/W | H'070D | H'B010 | 16 |
| | Command register1_1 | SPCMD1_1 | R/W | H'070D | H'B012 | 16 |
| | Command register2_1 | SPCMD2_1 | R/W | H'070D | H'B014 | 16 |
| | Command register3_1 | SPCMD3_1 | R/W | H'070D | H'B016 | 16 |
| | Buffer control register_1 | SPBFCR_1 | R/W | H'00 | H'B020 | 8 |
| | Buffer data count setting register_1 | SPBFDR_1 | R | H'0000 | H'B022 | 16 |

Note 1. In the description of the register names, the channel is omitted.

Note 2. Only 0 can be written to clear the flag.

### 24.3.1 Control Register (SPCR)

SPCR sets the operating mode. If the MSTR and MODFEN bits are changed while the function of this module is enabled by setting the SPE bit to 1, subsequent operations cannot be guaranteed.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SPRIE | SPE | SPTIE | SPEIE | MSTR | MODFEN | — | — |
| 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 |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | SPRIE | 0 | R/W | Receive Interrupt Enable<br>Enables or disables generation of receive interrupt requests (SPRI) when the number of receive data units in the receive buffer (SPRX) is equal to or greater than the specified receive buffer data triggering number and the SPRF flag in SPSR is set to 1.<br>0: Disables the generation of receive interrupt requests.<br>1: Enables the generation of receive interrupt requests. |
| 6 | SPE | 0 | R/W | Function Enable<br>Setting this bit to 1 enables the module function. When the MODF bit in the status register (SPSR) is 1, the SPE bit cannot be set to 1 (see **Section 24.4.6, Error Detection**). Setting the SPE bit to 0 disables the module function, and initializes a part of the module function (see **Section 24.4.7, Initialization**).<br>0: Disables the module function.<br>1: Enables the module function. |
| 5 | SPTIE | 0 | R/W | Transmit Interrupt Enable<br>Enables or disables generation of transmit interrupt requests (SPTI) when the number of transmit data units in the transmit buffer (SPTX) is equal to or less than the specified transmit buffer data triggering number and the SPTEF flag in SPSR is set to 1.<br>0: Disables the generation of transmit interrupt requests.<br>1: Enables the generation of transmit interrupt requests. |
| 4 | SPEIE | 0 | R/W | Error Interrupt Enable<br>Enables or disables the generation of error interrupt requests when this module detects a mode fault error and sets the MODF bit in the status register (SPSR) to 1, or when this module detects an overrun error and sets the OVRF bit in SPSR to 1 (see **Section 24.4.6, Error Detection**).<br>0: Disables the generation of error interrupt requests.<br>1: Enables the generation of error interrupt requests.<br>*Note*: This bit is valid only in SPI slave mode. |
| 3 | MSTR | 0 | R/W | Master/Slave Mode Select<br>Selects master/slave mode. According to MSTR bit settings, this module determines the direction of the RSPCK, MOSI, MISO, and SSL pins.<br>0: Slave mode<br>1: Master mode |
| 2 | MODFEN | 0 | R/W | Mode Fault Error Detection Enable<br>Enables or disables the detection of a mode fault error (see **Section 24.4.6, Error Detection**).<br>0: Disables the detection of a mode fault error.<br>1: Enables the detection of a mode fault error.<br>*Note*: This bit is valid only in SPI slave mode. When master mode is specified with the MSTR bit, this bit should always be cleared to 0. |
| 1, 0 | — | All 0 | R | Reserved<br>The write value should always be 0. Otherwise, operation cannot be guaranteed. |

### 24.3.2 Slave Select Polarity Register (SSLP)

SSLP sets the polarity of the SSL signal. If the contents of SSL0P are changed while the function of this module is enabled by setting the SPE bit in the control register (SPCR) to 1, subsequent operations cannot be guaranteed.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | SSL0P |
| 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<br>The write value should always be 0. Otherwise, operation cannot be guaranteed. |
| 0 | SSL0P | 0 | R/W | SSL Signal Polarity Setting<br>Sets the polarity of the SSL signal. The value of SSL0P indicates the active polarity of the SSL signal.<br>0: SSL signal 0-active<br>1: SSL signal 1-active |

### 24.3.3 Pin Control Register (SPPCR)

SPPCR sets the modes of the pins. If the contents of this register are changed while the function of this module is enabled by setting the SPE bit in the control register (SPCR) to 1, subsequent operations cannot be guaranteed.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | MOIFE | MOIFV | — | — | — | SPLP |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R/W | R/W | R | R | R | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7, 6 | — | All 0 | R | Reserved<br>The write value should always be 0. Otherwise, operation cannot be guaranteed. |
| 5 | MOIFE | 0 | R/W | MOSI Idle Value Fixing Enable<br>Fixes the MOSI output value when this module in master mode is in an SSL negation period (including the SSL retention period during a burst transfer). When MOIFE is 0, this module outputs the last output value from the previous serial transfer during the SSL negation period to the MOSI pin. (The value is undefined when CPHA is 0). When MOIFE is 1, this module outputs the fixed value set in the MOIFV bit to the MOSI pin.<br>0: MOSI output value equals the last output value from previous transfer. (The value is undefined when CPHA is 0).<br>1: MOSI output value equals the value set in the MOIFV bit. |
| 4 | MOIFV | 0 | R/W | MOSI Idle Fixed Value<br>If the MOIFE bit is 1 in master mode, this module, according to MOIFV bit settings, determines the MOSI signal value during the SSL negation period (including the SSL retention period during a burst transfer).<br>0: MOSI Idle fixed value equals 0.<br>1: MOSI Idle fixed value equals 1. |
| 3 to 1 | — | All 0 | R | Reserved<br>The write value should always be 0. Otherwise, operation cannot be guaranteed. |
| 0 | SPLP | 0 | R/W | Loopback<br>When the SPLP bit is set to 1, this module shuts off the path between the MISO pin and the shift register, and between the MOSI pin and the shift register, and connects (reverses) the input path and the output path for the shift register.<br>0: Normal mode<br>1: Loopback mode |

### 24.3.4 Status Register (SPSR)

SPSR indicates the operating status.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SPRF | TEND | SPTEF | — | — | MODF | — | OVRF |
| Initial Value | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R/(W)*1 | R | R/(W)*1 |

Note 1. Only 0 can be written to clear the flag after reading 1.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | SPRF | 0 | R | Receive Buffer Full Flag<br>Indicates that the number of receive data units in the receive buffer (SPRX) is equal to or greater than the receive buffer data triggering number specified in the buffer control register (SPBFCR).<br>0: The number of receive data units in the receive buffer is less than the receive buffer data triggering number.<br>1: The number of receive data units in the receive buffer is equal to or greater than the receive buffer data triggering number.<br>[Clearing conditions]<br>• The receive buffer data is read until the number of data units in the receive buffer becomes less than the specified receive buffer data triggering number.<br>• Receive buffer data reset is enabled.<br>• Power-on reset<br>[Setting condition]<br>• The number of data units in the receive buffer is equal to or greater than the specified receive buffer data triggering number. |
| 6 | TEND | 1 | R | Transmit End<br>This bit is set to 1 when transmission is completed, and this bit is 0 when transmission is not completed.<br>[Clearing condition]<br>• When transmit data are transferred from the transmit register to the shift register.<br>[Setting condition]<br>• When the number of data units in the transmit buffer (SPTX) is zero when a serial transfer is completed.<br>*Note*: This bit is valid only in SPI master mode. |
| 5 | SPTEF | 1 | R | Transmit Buffer Empty Flag<br>Indicates that the number of transmit data units in the transmit buffer (SPTX) is equal to or less than the transmit buffer data triggering number specified in the buffer control register (SPBFCR).<br>0: The number of transmit data units in the transmit buffer is equal to or greater than the specified transmit buffer data triggering number.<br>1: The number of transmit data units in the transmit buffer is less than the specified transmit buffer data triggering number.<br>[Clearing condition]<br>• When data is written to the transmit buffer until the number of transmit data units in the transmit buffer exceeds the specified transmit buffer data triggering number.<br>[Setting conditions]<br>• When the number of transmit data units in the transmit buffer is less than the specified transmit buffer data triggering number.<br>• When transmit buffer data reset is enabled.<br>• Power-on reset |
| 4, 3 | — | All 0 | R | Reserved<br>The write value should always be 0. Otherwise, operation cannot be guaranteed. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 2 | MODF | 0 | R/(W) *1 | Mode Fault Error Flag<br>Indicates the occurrence of a mode fault error. If the MODFEN bit is set to 1 when this module is in slave mode and the SSL pin is negated before the RSPCK cycle necessary for data transfer ends, this module detects a mode fault error. The active level of the SSL signal is determined by the SSL0P bit in the slave select polarity register (SSLP).<br>[Clearing conditions]<br>• SPSR is read when the MODF bit is 1, and then 0 is written to the MODF bit.<br>• Power-on reset<br>0: No mode fault error occurred<br>1: A mode fault error occurred<br>*Note*: This bit is valid only in SPI slave mode. |
| 1 | — | 0 | R | Reserved<br>The write value should always be 0. Otherwise, operation cannot be guaranteed. |
| 0 | OVRF | 0 | R/(W) *1 | Overrun Error Flag<br>Indicates the occurrence of an overrun error. If a serial transfer ends when there is not enough space for receiving the specified length of data in the receive buffer (SPRX), this module detects an overrun error, and sets the OVRF bit to 1.<br>[Clearing conditions]<br>• SPSR is read when the OVRF bit is 1, and then 0 is written to the OVRF bit.<br>• Power-on reset<br>0: No overrun error occurred<br>1: An overrun error occurred<br>*Note*: This bit is valid only in SPI slave mode. |

Note 1. Only 0 can be written to clear the flag after reading 1.

### 24.3.5 Data Register (SPDR)

SPDR is a buffer that holds data for transmission and reception.

The transmit buffer (SPTX) and receive buffer (SPRX) are independent and are mapped to SPDR.

SPDR should be read or written to in byte, word, or longword units according to the access width specification bit (SPLW) in the data control register (SPDCR).

The bit length to be used is determined by the data length specification bits (SPB3 to SPB0) in the command register (SPCMD).

The access width set by SPDCR must agree with the data length set by SPCMD.

When data is written to SPDR, the data will be written to the transmit buffer from SPDR if the transmit buffer has a space equal to or more than the SPDR access width. If there is not enough space, data will not be written to the transmit buffer. Even if an attempt is made to write data to the buffer, the data is ignored.

When data is read from SPDR, receive data in the receive buffer will be read. If SPDR is read when there is no receive data in the receive buffer, the read value is undefined.

When SPDR is written to with the longword-, word-, or byte-access width, the transmit data should be written to address 0 irrespective of the access width. If data is written to the other addresses, the data is not guaranteed.

When SPDR is read with the longword-, word-, or byte-access width, the receive data should be read from address 0. If data is read from the other addresses, the data is not guaranteed.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | SPD31 | SPD30 | SPD29 | SPD28 | SPD27 | SPD26 | SPD25 | SPD24 | SPD23 | SPD22 | SPD21 | SPD20 | SPD19 | SPD18 | SPD17 | SPD16 |
| Initial Value | Undefined | Undefined | Undefined | Undefined | Undefined | Undefined | Undefined | Undefined | Undefined | Undefined | Undefined | Undefined | Undefined | Undefined | Undefined | Undefined |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | SPD15 | SPD14 | SPD13 | SPD12 | SPD11 | SPD10 | SPD9 | SPD8 | SPD7 | SPD6 | SPD5 | SPD4 | SPD3 | SPD2 | SPD1 | SPD0 |
| Initial Value | Undefined | Undefined | Undefined | Undefined | Undefined | Undefined | Undefined | Undefined | Undefined | Undefined | Undefined | Undefined | Undefined | Undefined | Undefined | Undefined |
| R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

### 24.3.6 Sequence Control Register (SPSCR)

SPSCR sets the sequence control method when this module operates in master mode. If the contents of SPSCR are changed while the MSTR and SPE bits in the control register (SPCR) are 1 with the function of this module enabled in master mode, the subsequent operation cannot be guaranteed.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | SPSLN1 | SPSLN0 |
| 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<br>The write value should always be 0. Otherwise, operation cannot be guaranteed. |
| 1 | SPSLN1 | 0 | R/W | Sequence Length Specification<br>These bits specify a sequence length when this module in master mode performs sequential operations. This module in master mode changes command registers 0 to 3 (SPCMD0 to SPCMD3) to be referenced and the order in which they are referenced according to the sequence length that is set in the SPSLN1 and SPSLN0 bits.<br>The relationship among the setting of bits SPSLN1 and SPSLN0, sequence length, and SPCMD0 to SPCMD3 referenced by this module is shown below. In slave mode, SPCMD0 is always referenced.<br><br>Sequence Length / Referenced SPCMD #<br>00: 1 — 0 → 0 → …<br>01: 2 — 0 → 1 → 0 → …<br>10: 3 — 0 → 1 → 2 → 0 → …<br>11: 4 — 0 → 1 → 2 → 3 → 0 → … |
| 0 | SPSLN0 | 0 | R/W | |

### 24.3.7 Sequence Status Register (SPSSR)

SPSSR indicates the sequence control status when this module operates in master mode.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | SPCP1 | SPCP0 |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 to 2 | — | All 0 | R | Reserved<br>The write value should always be 0. Otherwise, operation cannot be guaranteed. |
| 1 | SPCP1 | 0 | R | Command Pointer<br>During sequence control, these bits indicate one of the command registers 0 to 3 (SPCMD0 to SPCMD3) that is currently pointed to by the pointer. The relationship between the setting of SPCP1 and SPCP0 and SPCMD0 to SPCMD3 is shown below.<br>For the sequence control, see **Section 24.4.8(1)(c), Sequence Control**.<br>00: SPCMD0<br>01: SPCMD1<br>10: SPCMD2<br>11: SPCMD3 |
| 0 | SPCP0 | 0 | R | |

### 24.3.8 Bit Rate Register (SPBR)

SPBR sets the bit rate in master mode. If the contents of SPBR are changed while the MSTR and SPE bits in the control register (SPCR) are 1 with the function of this module enabled in master mode, the subsequent operation cannot be guaranteed.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | SPR7 | SPR6 | SPR5 | SPR4 | SPR3 | SPR2 | SPR1 | SPR0 |
| 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 |

When this module is used in slave mode, the bit rate depends on the bit rate of the input clock regardless of the settings of SPBR and BRDV.

The bit rate is determined by combinations of SPBR settings and the bit settings in the BRDV1 and BRDV0 bits in the command registers (SPCMD0 to SPCMD3). The equation for calculating the bit rate is given below. In the equation, n denotes an SPBR setting (1, 2, …, 255), and N denotes a BRDV1 and BRDV0 bit setting (0, 1, 2, 3).

$$\text{Bit rate} = \frac{\text{f (P0}\phi\text{)}}{2 \times (n + 1) \times 2^{N}}$$

**Table 24.3** shows examples of the relationship between the SPBR register and BRDV1 and BRDV0 bit settings.

Table 24.3 Relationship between SPBR and BRDV1 and BRDV0 Settings

| SPBR (n) | BRDV[1:0] (N) | Division Ratio | Bit Rate |
|---|---|---|---|
| | | | P0ϕ = 100 MHz |
| 1 | 0 | 4 | 25.00 Mbps |
| 2 | 0 | 6 | 16.67 Mbps |
| 3 | 0 | 8 | 12.50 Mbps |
| 4 | 0 | 10 | 10.00 Mbps |
| 5 | 0 | 12 | 8.33 Mbps |
| 5 | 1 | 24 | 4.17 Mbps |
| 5 | 2 | 48 | 2.08 Mbps |
| 5 | 3 | 96 | 1.04 Mbps |
| 255 | 3 | 4096 | 24.41 Kbps |

### 24.3.9 Data Control Register (SPDCR)

SPDCR selects the width to access SPDR from longword-, word-, and byte-width, and enables or disables dummy data transmission for the master mode operation.

If the contents of SPDCR are changed while bit TEND in the status register (SPSR) indicates that transmission is not completed, the subsequent operation cannot be guaranteed.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | TXDMY | SPLW1 | SPLW0 | — | — | — | — | — |
| Initial Value | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | R/W | R | R | R | R | R |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | TXDMY | 0 | R/W | Dummy Data Transmission Enable<br>Enables or disables dummy data transmission.<br>When communication is performed with this bit set to 1, dummy data is transmitted from the MOSI pin and a serial communication can be performed even if there is no transmit data in the transmit buffer.<br>Specifically, if there is no transmit data in the transmit buffer and this bit is set to 1, dummy data is transferred to the shift register. Data previously transmitted from the pin is used as dummy data. If this bit is set to 1 after the initialization and a transfer is performed, the transmitted dummy data is undefined.<br>0: Disables dummy data transmission.<br>1: Enables dummy data transmission.<br>*Note*: This bit is valid only in the master mode. |
| 6 | SPLW1 | 0 | R/W | Access Width Specification<br>Specifies the width for accessing the data register (SPDR). If the length of data transferred to SPDR does not agree with these bit settings, operation is not guaranteed.*1<br>00: Setting prohibited<br>01: SPDR is accessed in bytes (8 bits).<br>10: SPDR is accessed in words (16 bits).<br>11: SPDR is accessed in longwords (32 bits). |
| 5 | SPLW0 | 1 | R/W | |
| 4 to 0 | — | All 0 | R | Reserved<br>The write value should always be 0. Otherwise, operation cannot be guaranteed. |

Note 1. The data length is specified by the SPB3 to SPB0 bits in the command register (SPCMD).
See **Section 24.3.5, Data Register (SPDR)**.

### 24.3.10 Clock Delay Register (SPCKD)

SPCKD sets a period from the beginning of SSL signal assertion to RSPCK oscillation (RSPCK delay) when the SCKDEN bit in the command register (SPCMD) is 1. If the contents of SPCKD are changed while the MSTR and SPE bits in the control register (SPCR) are 1 with the function of this module enabled in master mode, the subsequent operation cannot be guaranteed.

When using this module in slave mode, set 000b to SCKDL2 to SCKDL0.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | SCKDL2 | SCKDL1 | SCKDL0 |
| 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<br>The write value should always be 0. Otherwise, operation cannot be guaranteed. |
| 2 | SCKDL2 | 0 | R/W | RSPCK Delay Setting<br>These bits set an RSPCK delay value when the SCKDEN bit in SPCMD is 1.<br>The relationship between the setting of SCKDL2 to SCKDL0 and the RSPCK delay value is shown below.<br>000: 1 RSPCK<br>001: 2 RSPCK<br>010: 3 RSPCK<br>011: 4 RSPCK<br>100: 5 RSPCK<br>101: 6 RSPCK<br>110: 7 RSPCK<br>111: 8 RSPCK |
| 1 | SCKDL1 | 0 | R/W | |
| 0 | SCKDL0 | 0 | R/W | |

### 24.3.11 Slave Select Negation Delay Register (SSLND)

SSLND sets a period (SSL negation delay) from the transmission of a final RSPCK edge to the negation of the SSL signal during a serial transfer by this module in master mode. If the contents of SSLND are changed while the MSTR and SPE bits in the control register (SPCR) are 1 with the function of this module enabled in master mode, the subsequent operation cannot be guaranteed.

When using this module in slave mode, set 000b to SLNDL2 to SLNDL0.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | SLNDL2 | SLNDL1 | SLNDL0 |
| 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<br>The write value should always be 0. Otherwise, operation cannot be guaranteed. |
| 2 | SLNDL2 | 0 | R/W | SSL Negation Delay Setting<br>These bits set an SSL negation delay when the SLNDEN bit in SPCMD is 1.<br>The relationship between the setting of SLNDL2 to SLNDL0 and the SSL negation delay value is shown below.<br>000: 1 RSPCK<br>001: 2 RSPCK<br>010: 3 RSPCK<br>011: 4 RSPCK<br>100: 5 RSPCK<br>101: 6 RSPCK<br>110: 7 RSPCK<br>111: 8 RSPCK |
| 1 | SLNDL1 | 0 | R/W | |
| 0 | SLNDL0 | 0 | R/W | |

### 24.3.12 Next-Access Delay Register (SPND)

SPND sets a non-active period (next-access delay) after termination of a serial transfer when the SPNDEN bit in the command register (SPCMD) is 1. If the contents of SPND are changed while the MSTR and SPE bits in the control register (SPCR) are 1 with the function of this module enabled in master mode, the subsequent operation cannot be guaranteed.

When using this module in slave mode, set 000b to SPNDL2 to SPNDL0.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | SPNDL2 | SPNDL1 | SPNDL0 |
| 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<br>The write value should always be 0. Otherwise, operation cannot be guaranteed. |
| 2 | SPNDL2 | 0 | R/W | Next-Access Delay Setting<br>These bits set a next-access delay when the SPNDEN bit in SPCMD is 1.<br>The relationship between the setting of SPNDL2 to SPNDL0 and the next-access delay value is shown below.<br>000: 1 RSPCK + 2 PΦ<br>001: 2 RSPCK + 2 PΦ<br>010: 3 RSPCK + 2 PΦ<br>011: 4 RSPCK + 2 PΦ<br>100: 5 RSPCK + 2 PΦ<br>101: 6 RSPCK + 2 PΦ<br>110: 7 RSPCK + 2 PΦ<br>111: 8 RSPCK + 2 PΦ |
| 1 | SPNDL1 | 0 | R/W | |
| 0 | SPNDL0 | 0 | R/W | |

### 24.3.13 Command Register (SPCMD)

Each channel has four command registers (SPCMD0 to SPCMD3). SPCMD0 to SPCMD3 are used to set a transfer format for master mode operation. Some of the bits in SPCMD0 are used to set a transfer mode for slave mode operation. In master mode, this module sequentially references SPCMD0 to SPCMD3 according to the settings in bits SPSLN1 and SPSLN0 in the sequence control register (SPSCR), and executes the serial transfer that is set in the referenced SPCMD. While bit TEND in the status register (SPSR) indicates that transmission is not completed, correct operation of this module cannot be guaranteed if SPCMD is changed that is referred by this module. SPCMD referenced by this module in master mode can be checked by means of bits SPCP1 and SPCP0 in the sequence status register (SPSSR). When the function of this module in slave mode is enabled, operation cannot be guaranteed if the value set in SPCMD0 is changed.

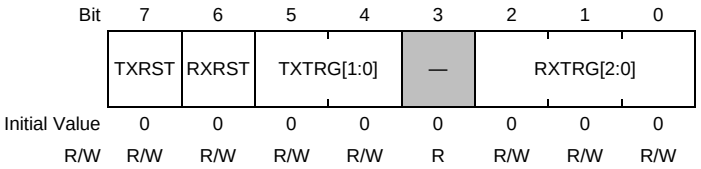
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | SCKDEN | SLNDEN | SPNDEN | LSBF | SPB3 | SPB2 | SPB1 | SPB0 | SSLKP | — | — | — | BRDV1 | BRDV0 | CPOL | CPHA |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 |
| R/W | R/W | R/W | 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 |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 | SCKDEN | 0 | R/W | RSPCK Delay Setting Enable<br>Sets the period from the point this module in master mode activates the SSL signal until the RSPCK starts oscillation (RSPCK delay). If the SCKDEN bit is 0, this module sets the RSPCK delay to 1 RSPCK. If the SCKDEN bit is 1, this module starts the oscillation of RSPCK at an RSPCK delay in compliance with the clock delay register (SPCKD) settings.<br>To use this module in slave mode, the SCKDEN bit should be set to 0.<br>0: An RSPCK delay of 1 RSPCK<br>1: An RSPCK delay equal to SPCKD settings. |
| 14 | SLNDEN | 0 | R/W | SSL Negation Delay Setting Enable<br>Sets the period from the point this module in master mode stops RSPCK oscillation until this module sets the SSL signal inactive (SSL negation delay). If the SLNDEN bit is 0, this module sets the SSL negation delay to 1 RSPCK. If the SLNDEN bit is 1, this module negates the SSL signal at an SSL negation delay in compliance with the slave select negation delay register (SSLND) settings.<br>To use this module in slave mode, the SLNDEN bit should be set to 0.<br>0: An SSL negation delay of 1 RSPCK<br>1: An SSL negation delay equal to SSLND settings. |
| 13 | SPNDEN | 0 | R/W | Next-Access Delay Enable<br>Sets the period from the point this module in master mode terminates a serial transfer and sets the SSL signal inactive until this module enables the SSL signal assertion for the next access (next-access delay). If the SPNDEN bit is 0, this module sets the next-access delay to 1 RSPCK + 2P0ϕ. If the SPNDEN bit is 1, this module inserts a next-access delay in compliance with the next-access delay register (SPND) settings.<br>To use this module in slave mode, the SPNDEN bit should be set to 0.<br>0: A next-access delay of 1 RSPCK + 2 P0ϕ<br>1: A next-access delay equal to SPND settings. |
| 12 | LSBF | 0 | R/W | LSB First<br>Sets the data format in master mode or slave mode to MSB first or LSB first.<br>0: MSB first<br>1: LSB first |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 11 | SPB3 | 0 | R/W | Data Length Setting<br>These bits set a transfer data length in master mode or slave mode.<br>0100 to 0111: 8 bits<br>1111: 16 bits<br>0010, 0011: 32 bits<br>Others: Setting prohibited |
| 10 | SPB2 | 1 | R/W | |
| 9 | SPB1 | 1 | R/W | |
| 8 | SPB0 | 1 | R/W | |
| 7 | SSLKP | 0 | R/W | SSL Signal Level Keeping<br>When this module in master mode performs a serial transfer, this bit specifies whether the SSL signal level for the current command is to be kept or negated between the SSL negation timing associated with the current command and the SSL assertion timing associated with the next command.<br>To use this module in slave mode, the SSLKP bit should be set to 0.<br>0: Negates the SSL signal upon completion of transfer.<br>1: Keeps the SSL signal level from the end of the transfer until the beginning of the next access. |
| 6 to 4 | — | All 0 | R | Reserved<br>The write value should always be 0. Otherwise, operation cannot be guaranteed. |
| 3 | BRDV1 | 1 | R/W | Bit Rate Division Setting<br>These bits are used to determine the bit rate. A bit rate is determined by combinations of bits BRDV1 and BRDV 0 and the settings in the bit rate register (SPBR) (see **Section 24.3.8, Bit Rate Register (SPBR)**). The settings in SPBR determine the base bit rate. The settings in bits BRDV1 and BRDV0 are used to select a bit rate which is obtained by dividing the base bit rate by 1, 2, 4, or 8. In the bits SPCMD0 to SPCMD3, different BRDV1 and BRDV0 settings can be specified. This permits the execution of serial transfers at a different bit rate for each command.<br>00: Select the base bit rate.<br>01: Select the base bit rate divided by 2.<br>10: Select the base bit rate divided by 4.<br>11: Select the base bit rate divided by 8. |
| 2 | BRDV0 | 1 | R/W | |
| 1 | CPOL | 0 | R/W | RSPCK Polarity Setting<br>Sets an RSPCK polarity in master or slave mode. When data communication is performed between the Renesas serial peripheral interface module and the other modules, the same RSPCK polarity should be set for both modules.<br>0: RSPCK = 0 when idle<br>1: RSPCK = 1 when idle |
| 0 | CPHA | 1 | R/W | RSPCK Phase Setting<br>Sets an RSPCK phase in master or slave mode. When data communication is performed between the Renesas serial peripheral interface module and the other modules, the same RSPCK phase should be set for both modules.<br>0: Data sampling on odd edge, data variation on even edge<br>1: Data variation on odd edge, data sampling on even edge |

## 24.3.14 Buffer Control Register (SPBFCR)

SPBFCR resets the number of data units in the transmit buffer (SPTX) or receive buffer (SPRX) and sets the number of triggering data units.




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | TXRST | 0 | R/W | Transmit Buffer Data Reset<br>Resets the transmit buffer to an empty state. Transmit data in the transmit buffer becomes invalid when this bit is set to 1.<br>0: Disables the reset operation.*1<br>1: Enables the reset operation<br>*Note 1.* The reset operation is performed after a power-on reset. |
| 6 | RXRST | 0 | R/W | Receive Buffer Data Reset<br>Resets the receive buffer to an empty state. Receive data in the receive buffer becomes invalid when this bit is set to 1.<br>0: Disables the reset operation.*1<br>1: Enables the reset operation<br>*Note 1.* The reset operation is performed after a power-on reset. |
| 5, 4 | TXTRG[1:0] | 00 | R/W | Transmit Buffer Data Triggering Number*2<br>Specifies the timing at which the transmit buffer empty state is determined, that is when the SPTEF flag in the status register is set. When the number of bytes of data in the transmit buffer (SPTX) is equal to or less than the specified triggering number, the SPTEF flag is set to 1.<br>00: 7 bytes (1)*1<br>01: 6 bytes (2)*1<br>10: 4 bytes (4)*1<br>11: 0 bytes (8)*1<br>*Note 1.* The value in the parenthesis shows the number of available bytes in the transmit buffer (SPTX).<br>*Note 2.* When transferring the data by using DMA, set the following values according to the data transfer length.<br>(The data transfer length is set in SPBn (n = 0-3) of the command register (SPCMD)).<br>1) Data transfer length is 1 byte<br>SPBFCR (TXTRG): 01 (6 bytes) or 10 (4 byte) or 11 (0 byte)<br>2) Data transfer length is 2 bytes<br>SPBFCR (TXTRG): 10 (4 bytes) or 11 (0 byte)<br>3) Data transfer length is 4bytes<br>SPBFCR (TXTRG): 11 (0 byte) |
| 3 | — | 0 | R | Reserved<br>The write value should always be 0. Otherwise, operation cannot be guaranteed. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 2 to 0 | RXTRG[2:0] | 000 | R/W | Receive Buffer Data Triggering Number<br>Specifies the timing at which the receive buffer full state is determined, that is when the SPRF flag in the status register is set. When the number of bytes of data in the receive buffer (SPRX) is equal to or greater than the specified triggering number, the SPRF flag is set to 1.<br>000: 1 byte (31)*1<br>001: 2 bytes (30)*1<br>010: 4 bytes (28)*1<br>011: 8 bytes (24)*1<br>100: 16 bytes (16)*1<br>101: 24 bytes (8)*1<br>110: 32 bytes (0)*1<br>111: 5 bytes (27)*1<br>*Note 1.* The value in the parenthesis shows the number of available bytes in the receive buffer (SPRX). |

### 24.3.15 Buffer Data Count Setting Register (SPBFDR)

SPBFDR indicates the number of data units stored in the transmit buffer (SPTX) and receive buffer (SPRX). The upper eight bits indicate the number of transmit data units in SPTX and the lower eight bits indicate the number of receive data units in SPRX.

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | T[3:0] | | | | — | — | R[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 | R | R | R | R | R |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 to 12 | — | All 0 | R | Reserved<br>The write value should always be 0. Otherwise, operation cannot be guaranteed. |
| 11 to 8 | T[3:0] | 0000 | R | Indicates the number of bytes of data to be transmitted in SPTX.<br>0000 indicates that SPTX is empty.<br>1000 indicates that SPTX is full. |
| 7, 6 | — | All 0 | R | Reserved<br>The write value should always be 0. Otherwise, operation cannot be guaranteed. |
| 5 to 0 | R[5:0] | 000000 | R | Shows the number of bytes of received data in SPRX.<br>000000 indicates that SPRX is empty.<br>100000 indicates that SPRX is full. |

## 24.4 Operation

In this section, the serial transfer period means a period from the beginning of driving valid data to the fetching of the final valid data.

### 24.4.1 Overview of Operations

This module is capable of serial transfers in slave mode and master mode. A particular mode of this module can be selected by using the MSTR bit in the control register (SPCR). **Table 24.4** gives the relationship between the modes and SPCR settings, and a description of each mode.

Table 24.4 Relationship between Modes and SPCR and Description of Each Mode

| Mode | Slave (SPI Operation) | Master (SPI Operation) |
|---|---|---|
| MSTR bit setting | 0 | 1 |
| MODFEN bit setting | 0 or 1 | 0 |
| RSPCK signal | Input | Output |
| MOSI signal | Input | Output |
| MISO signal | Output/Hi-Z | Input |
| SSL signal | Input | Output |
| SSL polarity modification function | Supported | Supported |
| Transfer rate | Up to P0ϕ/8 | Up to P0ϕ/2<br>(Up to P0φ/4 when P0φ is 100MHz) |
| Clock source | RSPCK input | On-chip baud rate generator |
| Clock polarity | Two | Two |
| Clock phase | Two | Two |
| First transfer bit | MSB/LSB | MSB/LSB |
| Transfer data length | 8, 16, or 32 bits | 8, 16, or 32 bits |
| Burst transfer | Possible (CPHA = 1) | Possible (CPHA = 0, 1) |
| RSPCK delay control | Not supported | Supported |
| SSL negation delay control | Not supported | Supported |
| Next-access delay control | Not supported | Supported |
| Transfer activation method | SSL input active or RSPCK oscillation | Transmit buffer is written when SPE = 1 |
| Sequence control | Not supported | Supported |
| Transmit buffer empty detection | Supported | Supported |
| Receive buffer full detection | Supported | Supported |
| Overrun error detection | Supported | Not Supported |
| Mode fault error detection | Supported (MODFEN = 1) | Not supported |

### 24.4.2 Pin Control

According to the MSTR bit in the control register (SPCR), this module can automatically switch pin directions and output modes. **Table 24.5** shows the relationship between pin states and bit settings.

Table 24.5 Relationship between Pin States and Bit Settings

| Mode | Pin | Pin State*1 |
|---|---|---|
| Master mode (SPI operation) (MSTR = 1) | RSPCK | CMOS output |
| | SSL | CMOS output |
| | MOSI | CMOS output |
| | MISO | Input |
| Slave mode (SPI operation) (MSTR = 0) | RSPCK | Input |
| | SSL | Input |
| | MOSI | Input |
| | MISO*1 | CMOS output/Hi-Z |

Note 1. When SSL is at the non-active level or the SPE bit in SPCR is cleared to 0, the pin state is Hi-Z.

This module in master mode (SPI operation) determines MOSI signal values during the SSL negation period (including the SSL retention period during a burst transfer) according to MOIFE and MOIFV bit settings in SPPCR, as shown in **Table 24.6**.

Table 24.6 MOSI Signal Value Determination during SSL Negation Period

| MOIFE | MOIFV | MOSI Signal Value during SSL Negation Period |
|---|---|---|
| 0 | 0, 1 | Last output value from previous transfer (The value is undefined when CPHA is 0) |
| 1 | 0 | Always 0 |
| 1 | 1 | Always 1 |

## 24.4.3 System Configuration Example

### (1) Master/Slave (with This LSI Acting as Master)

**Figure 24.2** shows a master/slave system configuration example when this LSI is used as a master. In master/slave configuration, the SSL output of this LSI (master) is not used. The SSL input of the slave is fixed to the low level, and the slave is always maintained in a selected state. In the transfer format corresponding to the case where the CPHA bit in the control register (SPCR) is 0, there are slave devices for which the SSL signal cannot be fixed to the active level. In situations where the SSL signal cannot be fixed, the SSL output of this LSI should be connected to the SSL input of the slave device.

This LSI (master) always drives the RSPCK and MOSI. The slave always drives the MISO.

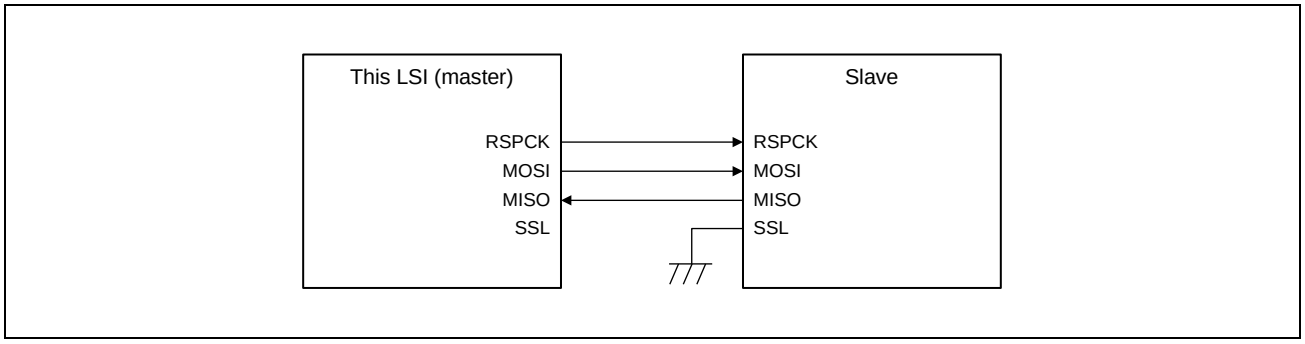



Figure 24.2 Master/Slave Configuration Example (This LSI = Master)

### (2) Master/Slave (with This LSI Acting as Slave)

**Figure 24.3** shows a master/slave system configuration example when this LSI is used as a slave. When this LSI is to operate as a slave, the SSL pin is used as SSL input. The master always drives the RSPCK and MOSI. This LSI (slave) always drives the MISO. When SSL is at the non-active level, the pin state is Hi-Z.

In the slave configuration in which the CPHA bit in the command register (SPCMD) is set to 1, the SSL input of this LSI (slave) is fixed to the 0 level, this LSI (slave) is always maintained in a selected state, and in this manner it is possible to execute serial transfer (**Figure 24.4**).

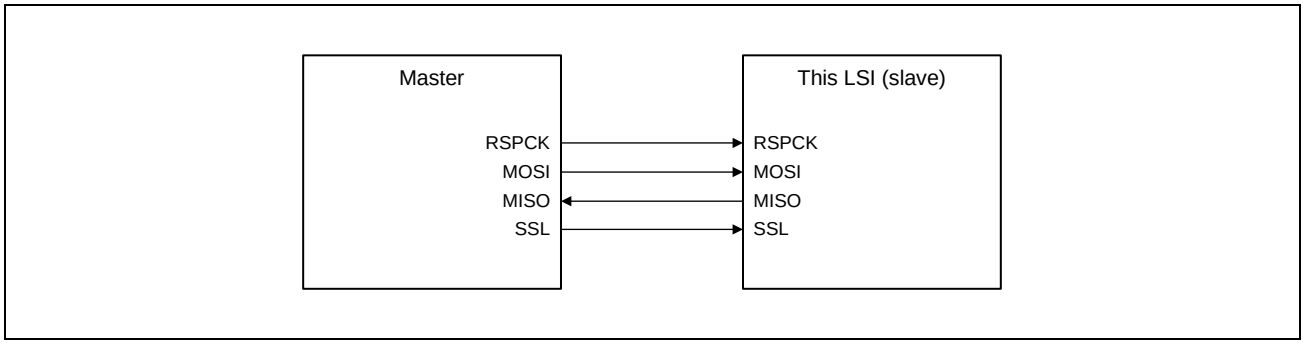



Figure 24.3 Master/Slave Configuration Example (This LSI = Slave)

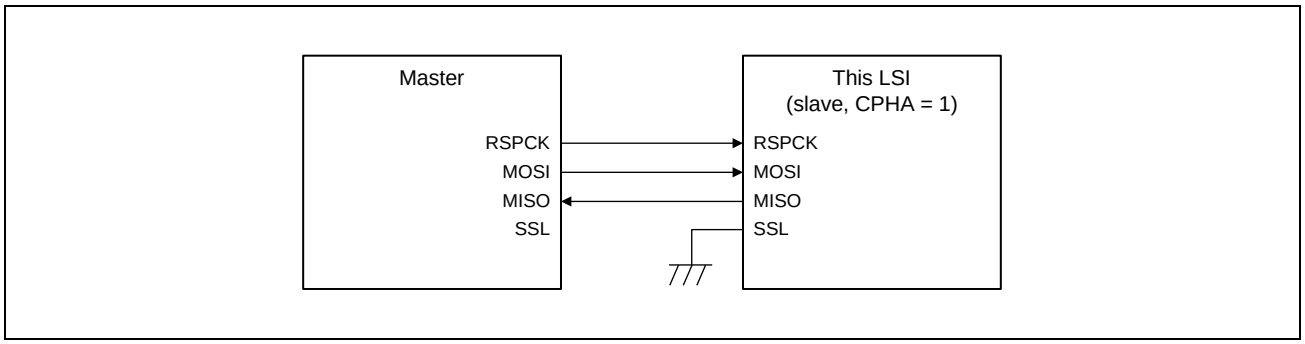



Figure 24.4 Master/Slave Configuration Example (This LSI = Slave, CPHA = 1)

### (3) Master/Multi-Slave (with This LSI Acting as Slave)

**Figure 24.5** shows a master/multi-slave system configuration example when this LSI is used as a slave. In the example of **Figure 24.5**, the system is comprised of a master and two LSIs (slave X and slave Y).

The RSPCK and MOSI outputs of the master are connected to the RSPCK and MOSI inputs of the LSIs (slave X and slave Y). The MISO outputs of the LSIs (slave X and slave Y) are all connected to the MISO input of the master. SSLX and SSLY outputs of the master are connected to the SSL inputs of the LSIs (slave X and slave Y), respectively.

The master always drives RSPCK, MOSI, SSLX, and SSLY. Of the LSIs (slave X and slave Y), the slave that receives low level input into the SSL0 input drives MISO.

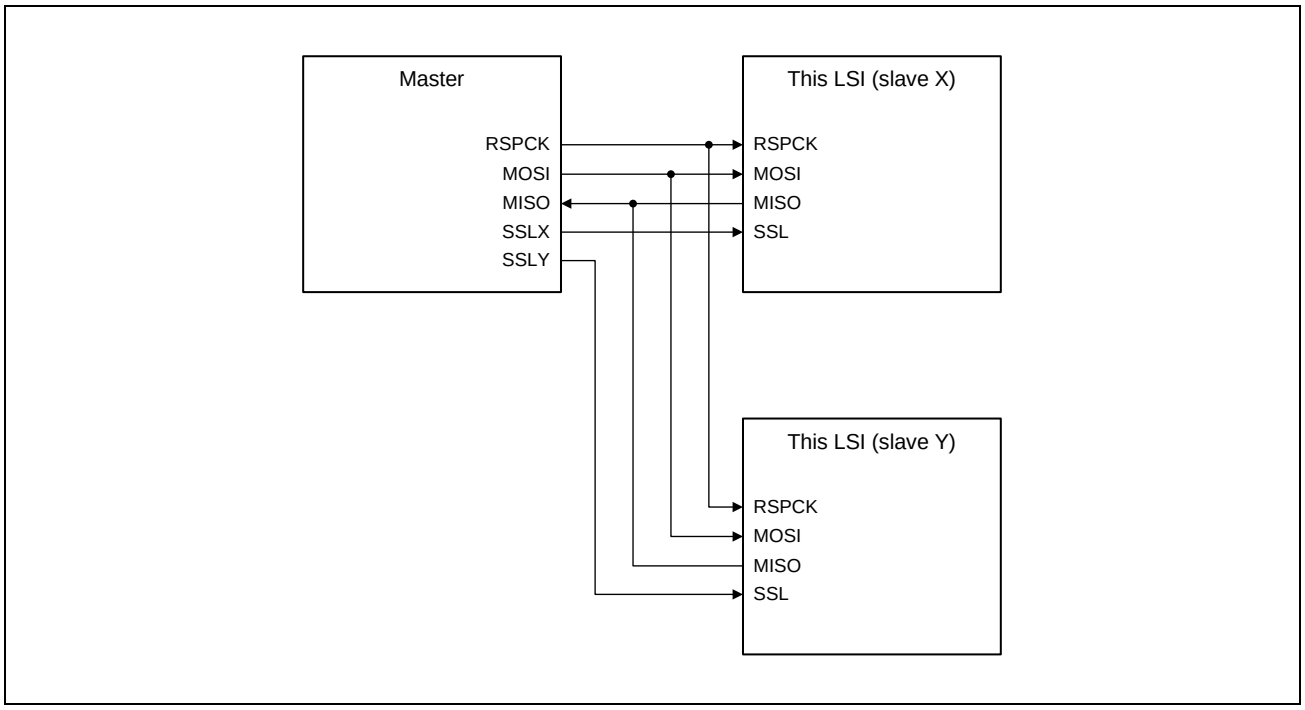



Figure 24.5 Master/Multi-Slave Configuration Example (This LSI = Slave)

### 24.4.4 Transfer Format

#### (1) CPHA = 0

**Figure 24.6** shows a sample transfer format for the serial transfer of 8-bit data when the CPHA bit in the command register (SPCMD) is 0. In **Figure 24.6**, RSPCK (CPOL = 0) indicates the RSPCK signal waveform when the CPOL bit in SPCMD is 0; RSPCK (CPOL = 1) indicates the RSPCK signal waveform when the CPOL bit is 1. The sampling timing represents the timing at which this module fetches serial transfer data into the shift register. The input/output directions of the signals depend on the settings of this module. For details, see **Section 24.4.2, Pin Control**.

When the CPHA bit is 0, the driving of valid data to the MOSI and MISO signals commences at an SSL signal assertion timing. The first RSPCK signal change timing that occurs after the SSL signal assertion becomes the first transfer data fetching timing. After this timing, data is sampled at every 1 RSPCK cycle. The change timing for MOSI and MISO signals is always 1/2 RSPCK cycle after the transfer data fetch timing. The settings in the CPOL bit do not affect the RSPCK signal operation timing; they only affect the signal polarity.

t1 denotes a period from an SSL signal assertion to RSPCK oscillation (RSPCK delay). t2 denotes a period from the cessation of RSPCK oscillation to an SSL signal negation (SSL negation delay). t3 denotes a period in which SSL signal assertion is suppressed for the next transfer after the end of serial transfer (next-access delay). t1, t2, and t3 are controlled by a master device running on the system. For a description of t1, t2, and t3 when this module is in master mode, see **Section 24.4.3(1), Master/Slave (with This LSI Acting as Master)**.

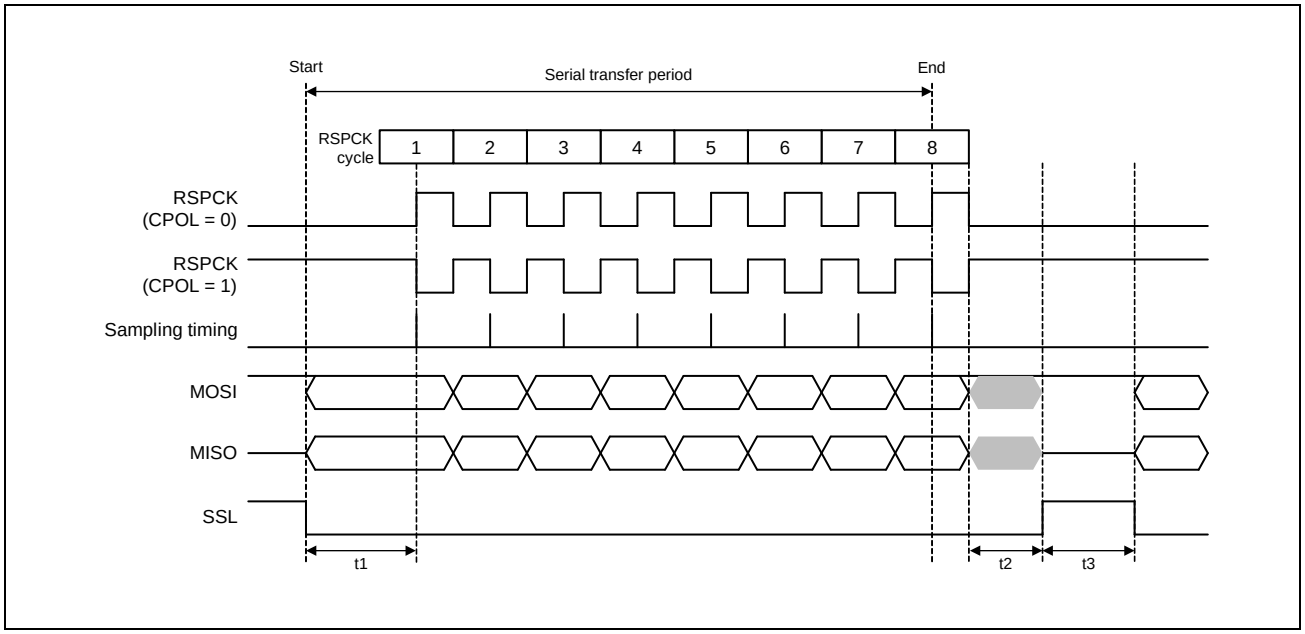



Figure 24.6 Transfer Format (CPHA = 0)

### (2) CPHA = 1

**Figure 24.7** shows a sample transfer format for the serial transfer of 8-bit data when the CPHA bit in the command register (SPCMD) is 1. In **Figure 24.7**, RSPCK (CPOL = 0) indicates the RSPCK signal waveform when the CPOL bit in SPCMD is 0; RSPCK (CPOL = 1) indicates the RSPCK signal waveform when the CPOL bit is 1. The sampling timing represents the timing at which this module fetches serial transfer data into the shift register. The input/output directions of the signals depend on the modes (master or slave). For details, see **Section 24.4.2, Pin Control**.

When the CPHA bit is 1, the driving of invalid data to the MOSI and MISO signals commences at an SSL signal assertion timing. The driving of valid data to the MOSI and MISO signals commences at the first RSPCK signal change timing that occurs after the SSL signal assertion. After this timing, data is updated at every 1 RSPCK cycle. The transfer data fetch timing is always 1/2 RSPCK cycle after the data update timing. The settings in the CPOL bit do not affect the RSPCK signal operation timing; they only affect the signal polarity.

t1, t2, and t3 are the same as those in the case of CPHA = 0. For a description of t1, t2, and t3 when this module is in master mode, see **Section 24.4.3(1), Master/Slave (with This LSI Acting as Master)**.

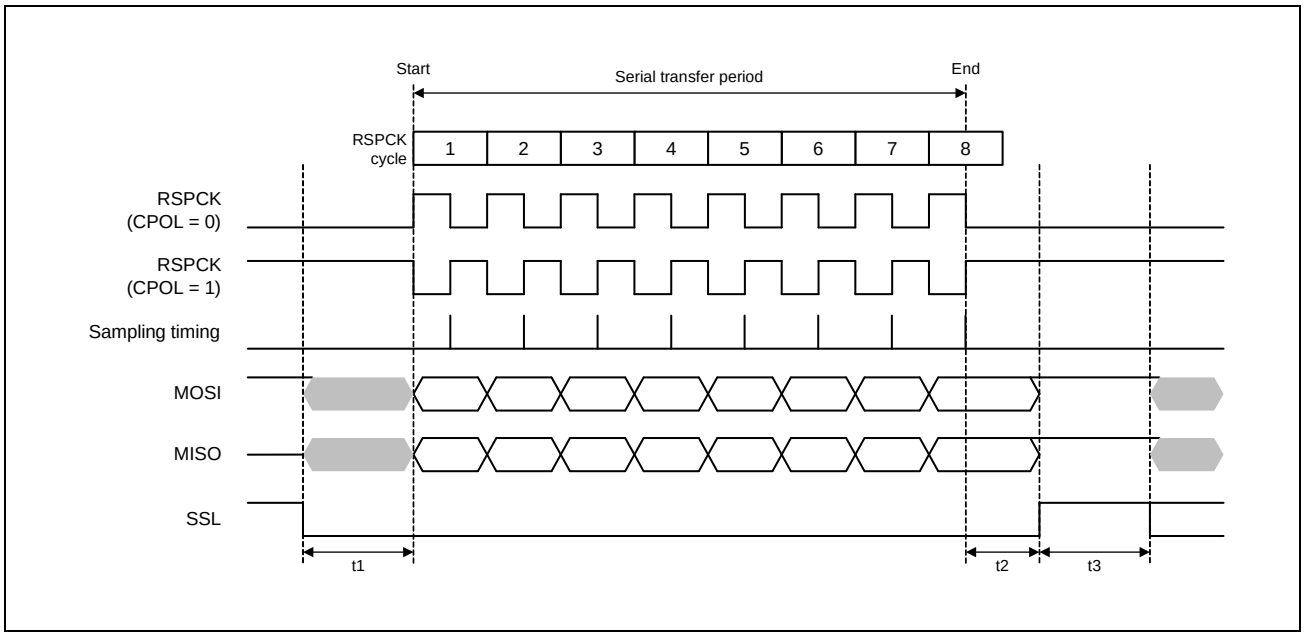



Figure 24.7 Transfer Format (CPHA = 1)

### 24.4.5 Data Format

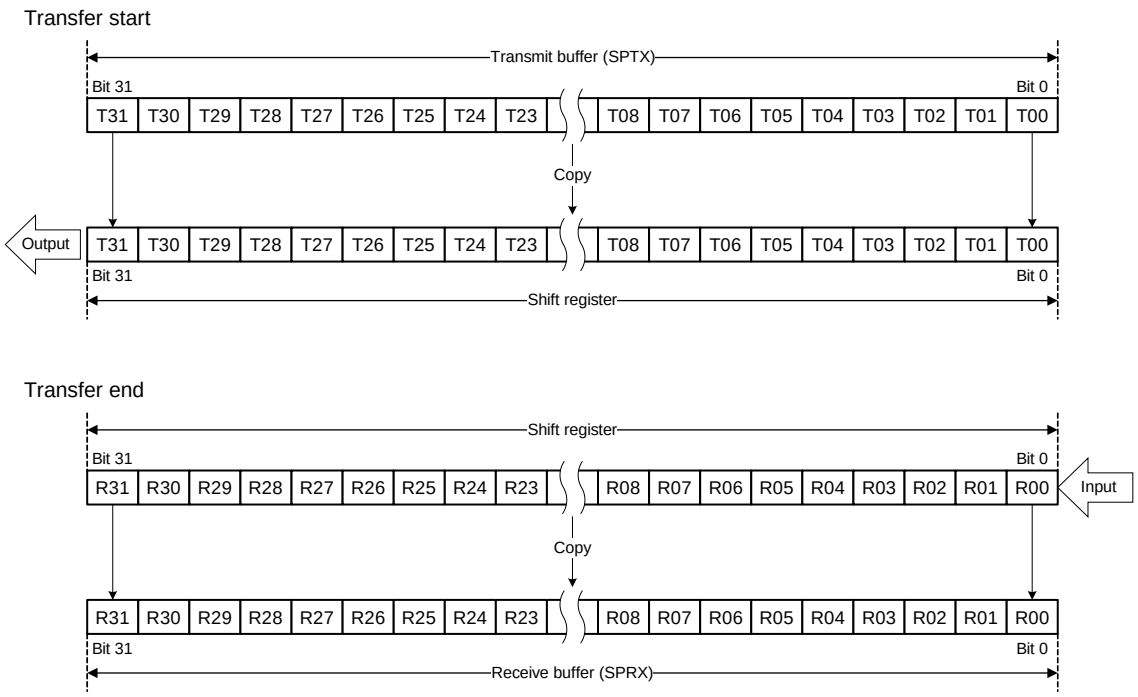
The data format depends on the settings in the command register (SPCMD). Irrespective of MSB/LSB first, this module treats the range from the LSB of the data register (SPDR) to the assigned data length as transfer data.

#### (1) MSB First Transfer (32-Bit Data)

**Figure 24.8** shows the operation of the transmit buffer (SPTX) and the shift register when this module performs a 32-bit data length MSB-first data transfer.

The CPU or direct memory access controller writes T31 to T00 to the transmit buffer of SPDR. If the shift register is empty, this module copies the data in the transmit buffer to the shift register, and fully populates the shift register. When serial transfer starts, this module outputs data from the MSB (bit 31) in the shift register, and shifts in the data from the LSB (bit 0) in the shift register. When the RSPCK cycle required for the serial transfer of 32 bits has passed, data R31 to R00 is stored in the shift register. In this state, this module copies the data from the shift register to the receive buffer, and empties the shift register. If the receive buffer does not have a space for the receive data length after the receive data has been copied from the shift register to the receive buffer, another serial transfer will not be started. In order to start another serial transfer, data for the receive data length should be read from the receive buffer to secure the necessary space in the receive buffer.

If another serial transfer is started before the CPU or direct memory access controller writes to the transmit buffer, received data R31 to R00 is shifted out from the shift register.




**Note:** Output = MOSI (master)/MISO (slave), input = MISO (master)/MOSI (slave)

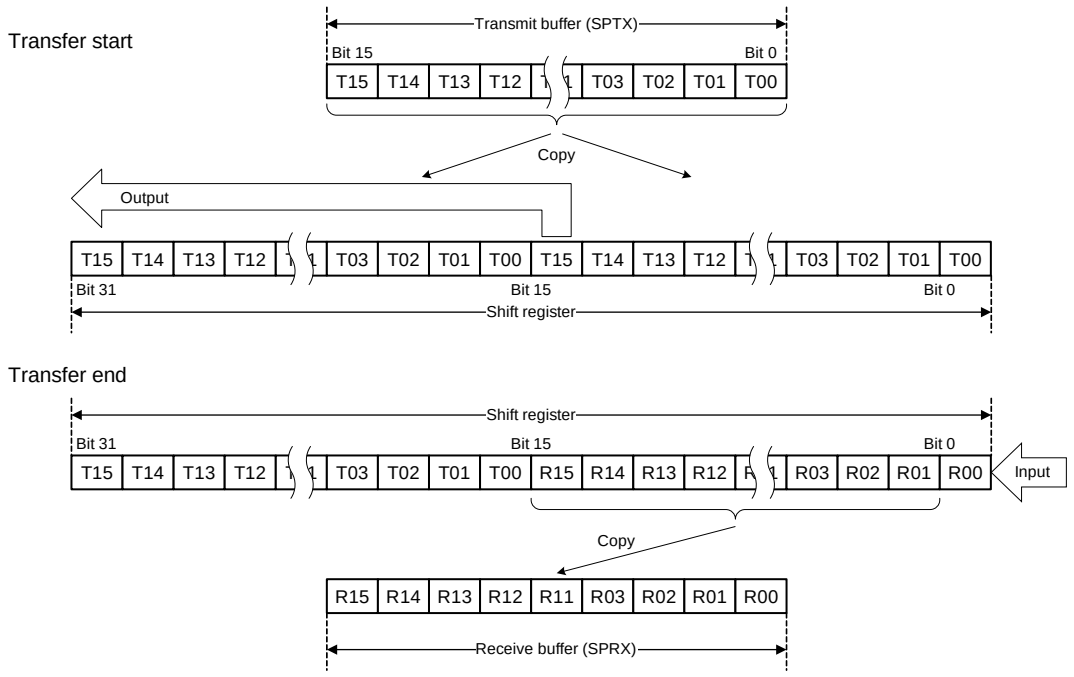
Figure 24.8 MSB First Transfer (32-Bit Data)

### (2) MSB First Transfer (16-Bit Data)

**Figure 24.9** shows the operation of the transmit buffer (SPTX) and the shift register when this module performs a 16-bit data length MSB-first data transfer.

The CPU or direct memory access controller writes T15 to T00 to the transmit buffer. If the shift register is empty, this module copies the data in the transmit buffer to the shift register, and fully populates the shift register. When serial transfer starts, this module outputs data from bit 15 in the shift register, and shifts in the data from the LSB (bit 0) in the shift register. When the RSPCK cycle required for the serial transfer of 16 bits has passed, received data R15 to R00 is stored in bits 15 to 0 in the shift register. After completion of the serial transfer, data that existed before the transfer is retained in bits 31 to 16 in the shift register. In this state, this module copies the data from the shift register to the receive buffer, and empties the shift register. If the receive buffer does not have a space for the receive data length after receive data has been copied from the shift register to the receive buffer, another serial transfer will not be started. In order to start another serial transfer, data for the receive data length should be read from the receive buffer to secure the necessary space in the receive buffer.

If another serial transfer is started before the CPU or direct memory access controller writes to the transmit buffer, received data R15 to R00 is shifted out from the shift register.




**Note:** Output = MOSI (master)/MISO (slave), input = MISO (master)/MOSI (slave)

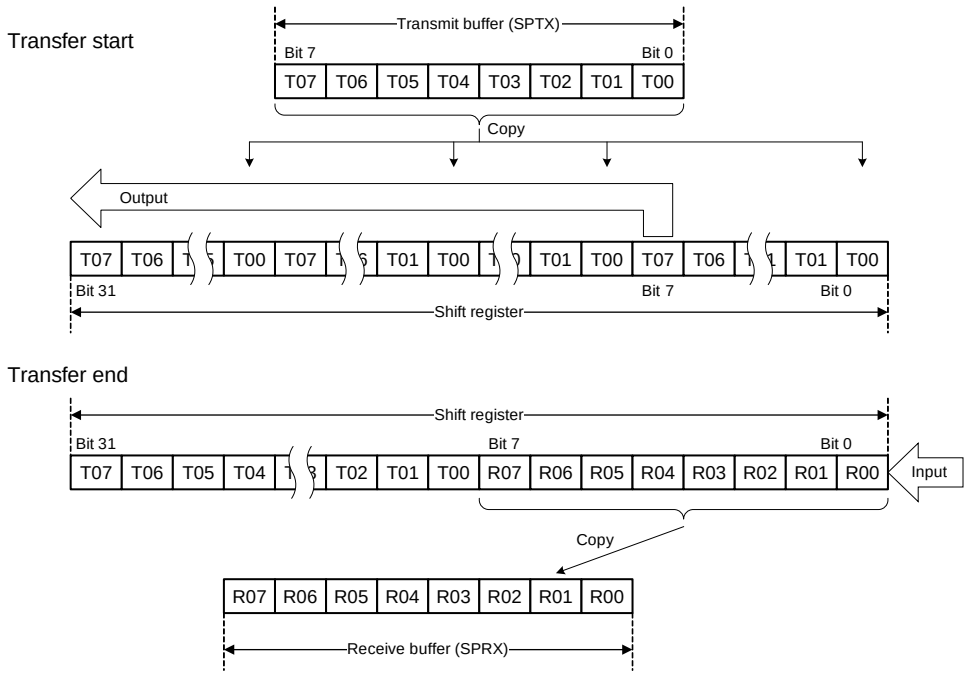
Figure 24.9 MSB First Transfer (16-Bit Data)

### (3) MSB First Transfer (8-Bit Data)

**Figure 24.10** shows the operation of the transmit buffer (SPTX) and the shift register when this module performs an 8-bit data length MSB-first data transfer.

The CPU or direct memory access controller writes T07 to T00 to the transmit buffer. If the shift register is empty, this module copies the data in the transmit buffer to the shift register, and fully populates the shift register. When serial transfer starts, this module outputs data from bit 7 in the shift register, and shifts in the data from the LSB (bit 0) in the shift register. When the RSPCK cycle required for the serial transfer of 8 bits has passed, received data R07 to R00 is stored in bits 7 to 0 in the shift register. After completion of the serial transfer, data that existed before the transfer is retained in bits 31 to 8 in the shift register. In this state, this module copies the data from the shift register to the receive buffer, and empties the shift register. If the receive buffer does not have a space for the receive data length after receive data has been copied from the shift register to the receive buffer, another serial transfer will not be started. In order to start another serial transfer, data for the receive data length should be read from the receive buffer to secure the necessary area in the receive buffer.

If another serial transfer is started before the CPU or direct memory access controller writes to the transmit buffer, received data R07 to R00 is shifted out from the shift register.




**Note:** Output = MOSI (master)/MISO (slave), input = MOSI (master)/MISO (slave)

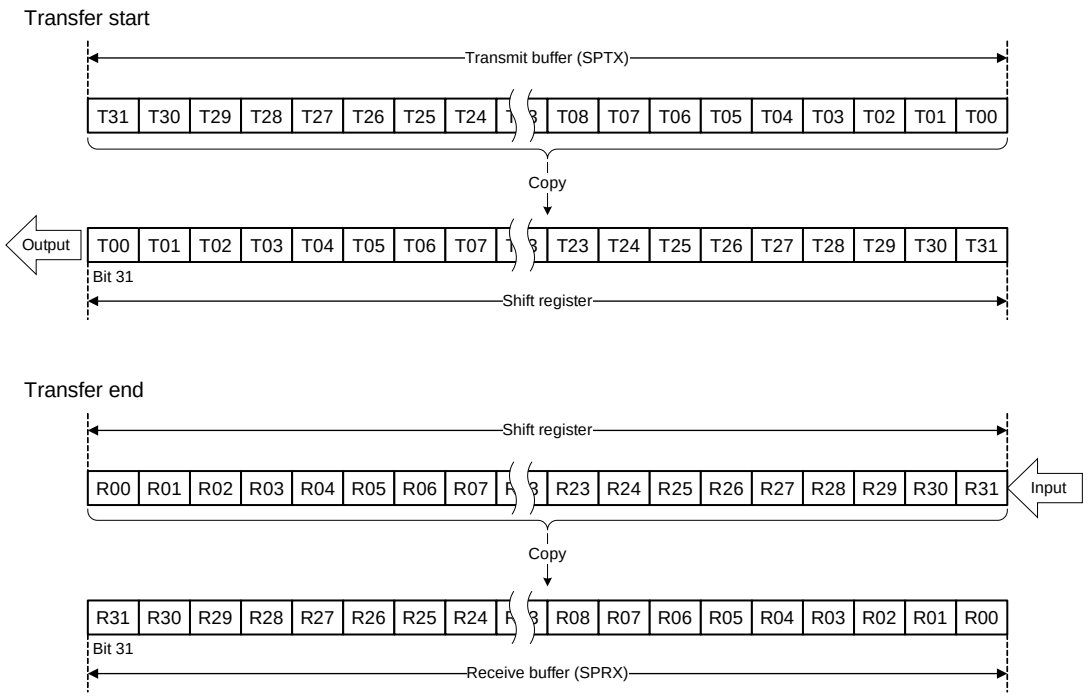
Figure 24.10 MSB First Transfer (8-Bit Data)

### (4) LSB First Transfer (32-Bit Data)

**Figure 24.11** shows the operation of the transmit buffer (SPTX) and the shift register when this module performs a 32-bit data length LSB-first data transfer.

The CPU or direct memory access controller writes T31 to T00 to the transmit buffer. If the shift register is empty, this module reverses the order of the bits of the data in the transmit buffer, copies it to the shift register, and fully populates the shift register. When serial transfer starts, this module outputs data from the MSB (bit 31) in the shift register, and shifts in the data from the LSB (bit 0) in the shift register. When the RSPCK cycle required for the serial transfer of 32 bits has passed, data R00 to R31 is stored in the shift register. In this state, this module copies the data, in which the order of the bits is reversed, from the shift register to the receive buffer, and empties the shift register. If the receive buffer does not have a space for the receive data length after receive data has been copied from the shift register to the receive buffer, another serial transfer will not be started. In order to start another serial transfer, data for the receive data length should be read from the receive buffer to secure the necessary space in the receive buffer.

If another serial transfer is started before the CPU or direct memory access controller writes to the transmit buffer of the SPDR, received data R00 to R31 is shifted out from the shift register.




**Note:** Output = MOSI (master)/MISO (slave), input = MISO (master)/MOSI (slave)

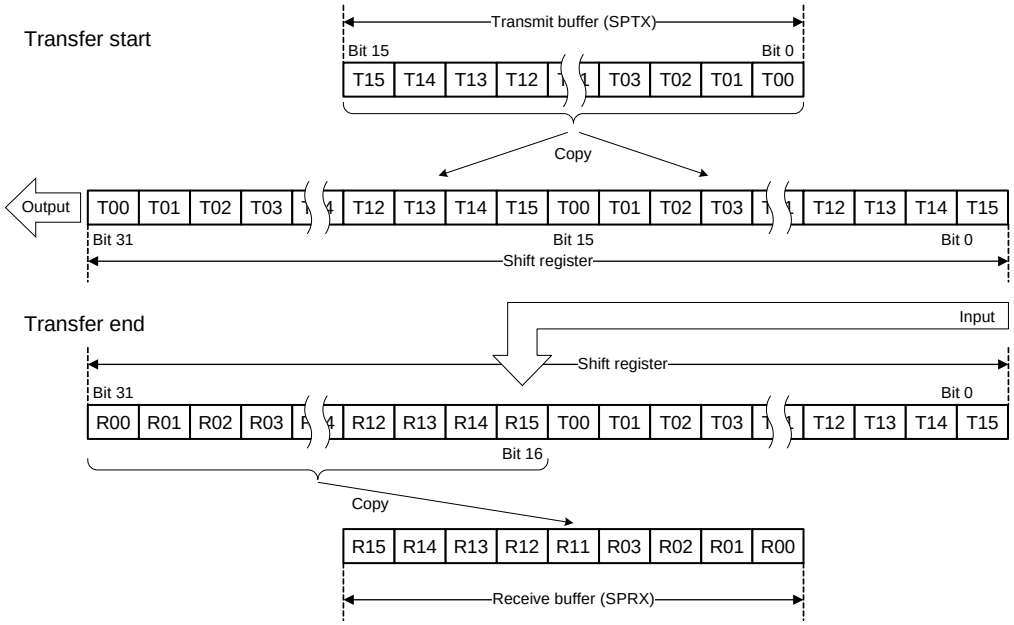
Figure 24.11 LSB First Transfer (32-Bit Data)

### (5) LSB First Transfer (16-Bit Data)

**Figure 24.12** shows the operation of the transmit buffer (SPTX) and the shift register when this module performs a 16-bit data length LSB-first data transfer.

The CPU or direct memory access controller writes T15 to T00 to the transmit buffer. If the shift register is empty, this module reverses the order of the bits of the data in the transmit buffer, copies it to the shift register, and fully populates the shift register. When serial transfer starts, this module outputs data from the MSB (bit 31) in the shift register, and shifts in the data from bit 16 in the shift register. When the RSPCK cycle required for the serial transfer of 16 bits has passed, received data R00 to R15 is stored in bits 31 to 16 in the shift register. After completion of the serial transfer, data that existed before the transfer is retained in bits 15 to 0 in the shift register. In this state, this module copies the data, in which the order of the bits is reversed, from the shift register to the receive buffer of SPDR, and empties the shift register. If the receive buffer does not have a space for the receive data length after receive data has been copied from the shift register to the receive buffer, another serial transfer will not be started. In order to start another serial transfer, data for the receive data length should be read from the receive buffer to secure the necessary space in the receive buffer.

If another serial transfer is started before the CPU or direct memory access controller writes to the transmit buffer of SPDR, received data R00 to R15 is shifted out from the shift register.




**Note:** Output = MOSI (master)/MISO (slave), input = MISO (master)/MOSI (slave)

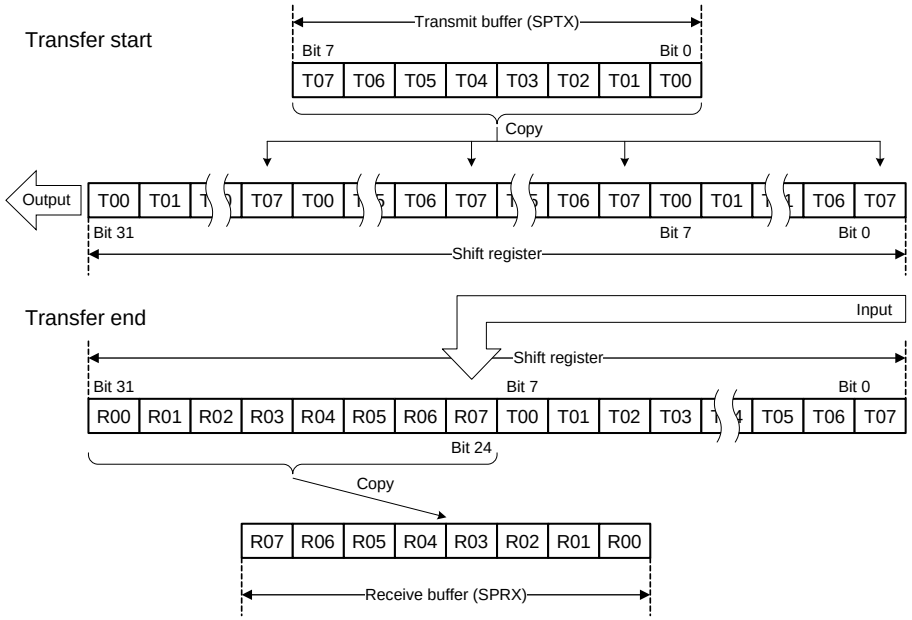
Figure 24.12 LSB First Transfer (16-Bit Data)

### (6) LSB First Transfer (8-Bit Data)

**Figure 24.13** shows the operation of the transmit buffer (SPTX) and the shift register when this module performs an 8-bit data length LSB-first data transfer.

The CPU or direct memory access controller writes T07 to T00 to the transmit buffer. If the shift register is empty, this module reverses the order of the bits of the data in the transmit buffer, copies it to the shift register, and fully populates the shift register. When serial transfer starts, this module outputs data from the MSB (bit 31) in the shift register, and shifts in the data from bit 24 in the shift register. When the RSPCK cycle required for the serial transfer of 8 bits has passed, received data R00 to R07 is stored in bits 31 to 24 of the shift register. After completion of the serial transfer, data that existed before the transfer is retained in bits 23 to 0 in the shift register. In this state, this module copies the data, in which the order of the bits is reversed, from the shift register to the receive buffer of SPDR, and empties the shift register. If the receive buffer does not have a space for the receive data length after the receive data has been copied from the shift register to the receive buffer, another serial transfer will not be started. In order to start another serial transfer, data for the receive data length should be read from the receive buffer to secure the necessary space in the receive buffer.

If another serial transfer is started before the CPU or direct memory access controller writes to the transmit buffer of SPDR, received data R00 to R07 is shifted out from the shift register.




**Note:** Output = MOSI (master)/MISO (slave), input = MISO (master)/MOSI (slave)

Figure 24.13 LSB First Transfer (8-Bit Data)

### 24.4.6 Error Detection

In the normal serial transfer, the data written to the transmit buffer of the data register (SPDR) is serially transmitted, and the serially received data can be read from the receive buffer of SPDR. If access is made to SPDR, depending on the status of the transmit buffer/receive buffer or the status at the beginning or end of serial transfer, in some cases non-normal transfers can be executed.

If a non-normal transfer operation occurs, this module detects the event as an overrun error or a mode fault error. **Table 24.7** shows the relationship between non-normal transfer operations and the error detection function.

Table 24.7 Relationship between Non-Normal Transfer Operations and Error Detection Function

| | Occurrence Condition | Operation | Error Detection |
|---|---|---|---|
| A | SPDR is written when the transmit buffer is full. | Missing write data. | None |
| B | Serial transfer is started in slave mode when transmit data is still not loaded on the shift register. | Data received in previous serial transfer is serially transmitted. | None |
| C | SPDR is read when the receive buffer is empty. | The output data is undefined. | None |
| D | Serial transfer terminates when the receive buffer is full. | Missing serial receive data. | Overrun error (only in slave mode) |
| E | The SSL input signal is negated during serial transfer in slave mode. | Serial transfer suspended.<br>Missing send/receive data.<br>Operation disabled. | Mode fault error |

On operation A shown in **Table 24.7**, this module does not detect an error. Whether SPDR can be written to or not can be checked using the T[3:0] bits in the buffer data count setting register (SPBFDR).

Likewise, this module does not detect an error on operation B. In a serial transfer that was started before the shift register was updated, this module sends the data that was received in the previous serial transfer, and does not treat the operation indicated in B as an error. Note that the received data from the previous serial transfer is retained in the receive buffer of SPDR, thus it can be correctly read.

Similarly, this module does not detect an error on operation C. To prevent extraneous data from being read, the number of receive data units stored in the receive buffer should be read from the R[5:0] bits in the buffer data count setting register (SPBFDR).

An overrun error shown in D is described in **Section 24.4.6(1), Overrun Error**. A mode fault error shown in E is described in **Section 24.4.6(2), Mode Fault Error**.

### (1) Overrun Error

If serial transfer ends when the receive buffer of the data register (SPDR) is full, this module detects an overrun error, and sets the OVRF bit in SPSR to 1. When the OVRF bit is 1, this module does not copy data from the shift register to the receive buffer so that the data prior to the occurrence of the error is retained in the receive buffer. To reset the OVRF bit in SPSR to 0, either perform a power-on reset, or write a 0 to the OVRF bit after SPSR has been read with the OVRF bit set to 1.

**Figure 24.14** shows an example of operation of the SPRF and OVRF bits in SPSR. The SPSR and SPDR accesses shown in **Figure 24.14** indicate the condition of accesses to SPSR and SPDR, respectively, where I denotes an idle cycle, W a write cycle, and R a read cycle. In the example of **Figure 24.14**, this module performs an 8-bit serial transfer in which the CPHA bit in the command register (SPCMD) is 1, and CPOL is 0. The numbers given under the RSPCK waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).

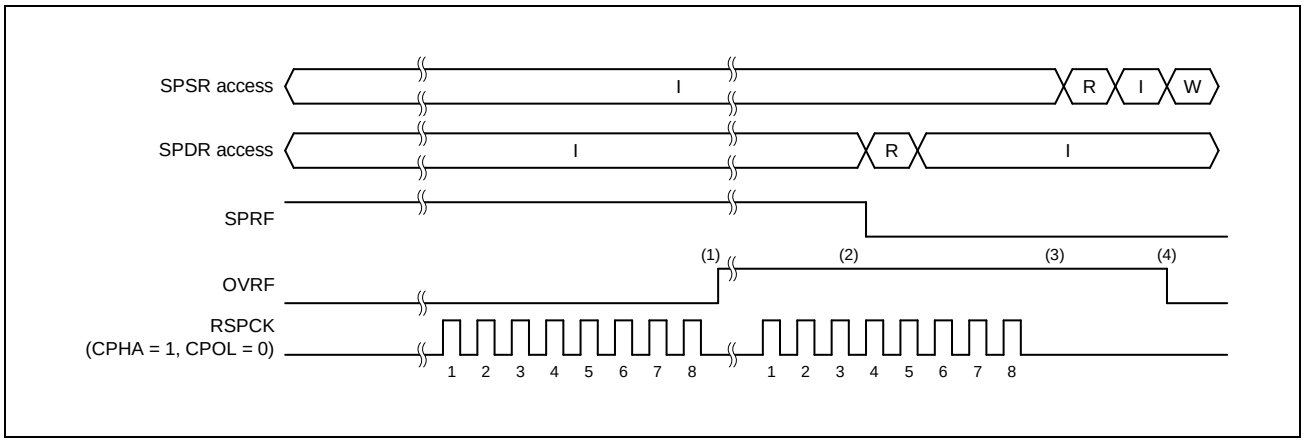



Figure 24.14 SPRF and OVRF Bit Operation Example

The operation of the flags at the timing shown in steps (1) to (4) in the figure is described below.

1. If a serial transfer terminates when the receive buffer does not have a space for the receive data length, this module detects an overrun error, and sets the OVRF bit to 1. This module does not copy the data in the shift register to the receive buffer.
2. The OVFR bit is not cleared even when SPDR is read and thus the number of data bytes in the receive buffer becomes less than the number of the receive buffer data triggering number specified by the RXTRG bits.
3. If the serial transfer terminates in an overrun error state, this module determines that the shift register is empty; in this manner, data transfer is enabled from the transmit buffer to the shift register.
4. If 0 is written to the OVRF bit after SPSR is read with OVRF = 1, this module clears the OVRF bit.

The occurrence of an overrun can be checked either by reading SPSR or by using an error interrupt and reading SPSR. When using an error interrupt, set the SPEIE bit in the control register (SPCR) to 1. When executing a serial transfer without using an error interrupt, measures should be taken to ensure the early detection of overrun errors, such as reading SPSR immediately after SPDR is read.

The OVRF bit is cleared to 0 under the following conditions:

- After SPSR is read in a condition in which the OVRF bit is set to 1, 0 is written to the OVRF bit.
- Power-on reset

NOTE

When the receive buffer has area enough to store receive data with an overrun error, this module receives receive data.

### (2) Mode Fault Error

When the MSTR bit is 0, this module operates in slave mode. This module detects a mode fault error if the SSL input signal is negated during the serial transfer period (from the time the driving of valid data is started to the time the final valid data is fetched) when the MODFEN bit is 1 in slave mode.

Upon detecting a mode fault error, this module stops driving of the output signals and clears the SPE bit in SPCR to 0. When the SPE bit is cleared to 0, the function of this module is disabled, and this module stops driving external signals. For details of disabling the function of this module by clearing the SPE bit to 0, see **Section 24.4.7, Initialization**.

The occurrence of a mode fault error can be checked either by reading SPSR or by using an error interrupt and reading SPSR. When using an error interrupt, set the SPEIE bit in the control register (SPCR) to 1. To detect a mode fault error without using an error interrupt, it is necessary to poll SPSR.

When the MODF bit is 1, writing 1 to the SPE bit is ignored. To enable the function of this module after the detection of a mode fault error, the MODF bit must be set to 0. The MODF bit is cleared to 0 under the following conditions:

- After SPSR is read in a condition where the MODF bit has turned 1, 0 is written to the MODF bit.
- Power-on reset

### 24.4.7 Initialization

If 0 is written to the SPE bit in the control register (SPCR) or this module clears the SPE bit to 0 because of the detection of a mode fault error, this module disables the module function, and initializes a part of the module function. When a power-on reset is generated, this module initializes all of the module function. An explanation follows of initialization by the clearing of the SPE bit.

#### (1) Initialization by Clearing SPE Bit

When the SPE bit in SPCR is cleared, this module performs the following initialization:

- Suspending any serial transfer that is being executed
- Stopping the driving of output signals (Hi-Z) in slave mode
- Initializing the internal state
- Initializing the TEND bit in SPSR

Initialization by the clearing of the SPE bit does not initialize the control bits of this module. For this reason, this module can be started in the same transfer mode as prior to the initialization if the SPE bit is re-set to 1.

## 24.4.8 SPI Operation

### (1) Multi-Master Mode Operation

This section explains the operation in multi-master mode.

#### (a) Starting Serial Transfer

A serial transfer is started when transmit data is copied from the transmit buffer to the shift register, the shift register becomes full, and the receive buffer has a space for the receive data length. If transmit data has already been written to the shift register, data is not copied from the transmit buffer to the shift register.

For details of the transfer format, see **Section 24.4.4, Transfer Format**.

#### (b) Terminating Serial Transfer

Irrespective of the CPHA bit in the command register (SPCMD), this module terminates the serial transfer after transmitting an RSPCK edge corresponding to the final sampling timing. After the serial transfer is completed, receive data is copied from the shift register to the receive buffer. If the receive buffer does not have a space for the receive data length after receive data is copied from the shift register to the receive buffer, another serial transfer will not be performed. In order to perform another serial transfer, data for the receive data length should be read from the receive buffer to secure the space for the receive data.

It should be noted that the final sampling timing varies depending on the bit length of transfer data. In master mode, the data length depends on the settings in bits SPB3 to SPB0 in SPCMD. For details on the transfer format, see **Section 24.4.4, Transfer Format**.

#### (c) Sequence Control

The transfer format that is employed in master mode is determined by the sequence control register (SPSCR), command registers 0 to 3 (SPCMD0 to SPCMD3), the bit rate register (SPBR), the clock delay register (SPCKD), the slave select negation delay register (SSLND), and the next-access delay register (SPND).

SPSCR is a register used to determine the sequence configuration for serial transfers that are executed by this module in master mode. The following items are set in command registers SPCMD0 to SPCMD3: SSL output signal value, MSB/ LSB first, data length, some of the bit rate settings, RSPCK polarity/phase, whether SPCKD is to be referenced, whether SSLND is to be referenced, and whether SPND is to be referenced. SPBR holds some of the bit rate settings; SPCKD, a clock delay value; SSLND, an SSL negation delay; and SPND, a next-access delay value.

According to the sequence length that is assigned to SPSCR, this module makes up a sequence comprised of a part or all of SPCMD0 to SPCMD3. This module contains a pointer to the SPCMD that makes up the sequence. The value of this pointer can be checked by reading bits SPCP1 and SPCP0 in the sequence status register (SPSSR). When the SPE bit in the control register (SPCR) is set to 1 and the function of this module is enabled, this module loads the pointer to the commands in SPCMD0, and incorporates the SPCMD0 settings into the transfer format at the beginning of serial transfer. This module increments the pointer each time the next-access delay period for a data transfer ends. Upon completion of the serial transfer that corresponds to the final command comprising the sequence, this module sets the pointer in SPCMD0, and in this manner the sequence is executed repeatedly.

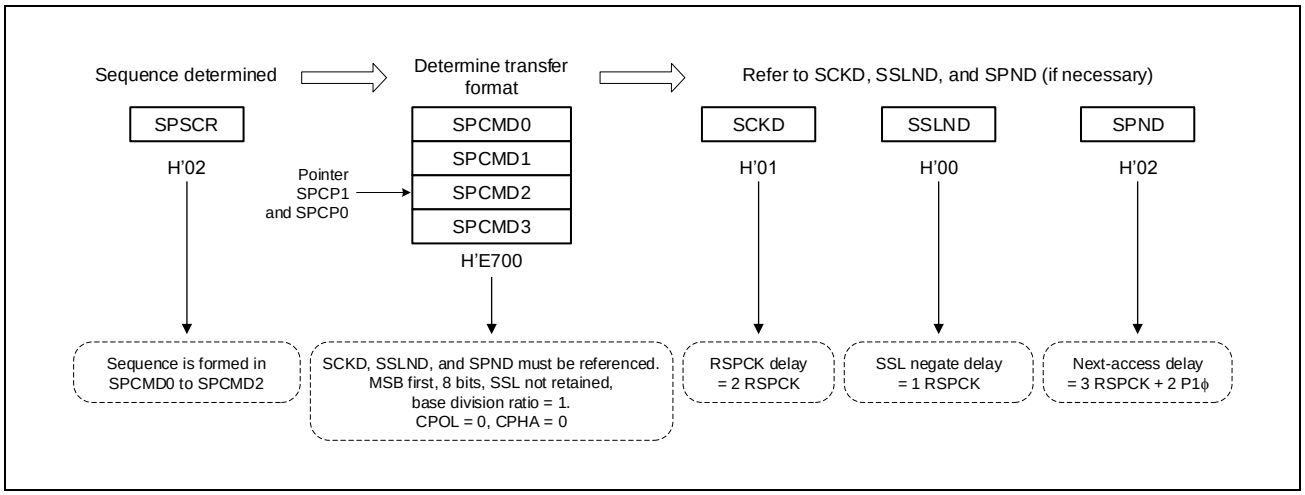



Figure 24.15 Determination Procedure of Serial Transfer Mode in Master Mode

### (d) Burst Transfer

If the SSLKP bit in the command register (SPCMD) that this module references during the current serial transfer is 1, this module keeps the SSL signal level during the serial transfer until the beginning of the SSL signal assertion for the next serial transfer. If the SSL signal level for the next serial transfer is the same as the SSL signal level for the current serial transfer, this module can execute continuous serial transfers while keeping the SSL signal assertion status (burst transfer).

**Figure 24.16** shows an example of an SSL signal operation for the case where a burst transfer is implemented using SPCMD0 and SPCMD1 settings. The text below explains operations (1) to (7) as depicted in **Figure 24.16**. It should be noted that the polarity of the SSL output signal depends on the settings in the slave select polarity register (SSLP).

1. Based on SPCMD0, this module asserts the SSL signal and inserts RSPCK delays.
2. Serial transfers are executed according to SPCMD0.
3. SSL negation delays are inserted.
4. Because the SSLKP bit in SPCMD0 is 1, this module keeps the SSL signal value on SPCMD0. This period is sustained, at the shortest, for a period equal to the next-access delay of SPCMD0. If the shift register is empty after the passage of a minimum period, this period is sustained until such time as the transmit data is stored in the shift register for another transfer.
5. Based on SPCMD1, this module asserts the SSL signal and inserts RSPCK delays.
6. Serial transfers are executed according to SPCMD1.
7. Because the SSLKP bit in SPCMD1 is 0, this module negates the SSL signal. In addition, a next-access delay is inserted according to SPCMD1.

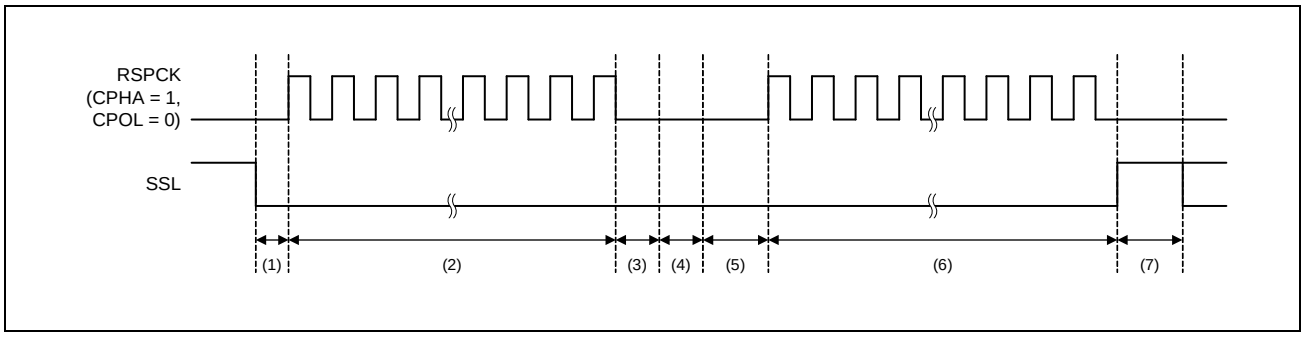



Figure 24.16 Example of Burst Transfer Operation using SSLKP Bit

If the SSL signal settings in the SPCMD in which 1 is assigned to the SSLKP bit are different from the SSL signal output settings in the SPCMD to be used in the next transfer, this module switches the SSL signal status to SSL signal assertion ((5) in **Figure 24.16**) corresponding to the command for the next transfer. Notice that if such an SSL signal switching occurs, the slaves that drive the MISO signal compete, and the possibility arises of the collision of signal levels.

This module in master mode references within the module the SSL signal operation for the case where the SSLKP bit is not used. Even when the CPHA bit in SPCMD is 0, this module can accurately start serial transfers by asserting the SSL signal for the next transfer. For this reason, burst transfers in master mode can be executed irrespective of CPHA bit settings (see **Section 24.4.8(2), Slave Mode Operation**).

### (e) RSPCK Delay (t1)

The RSPCK delay value in master mode depends on SCKDEN bit settings in the command register (SPCMD) and on clock delay register (SPCKD) settings. This module determines the SPCMD to be referenced during serial transfer by pointer control, and determines an RSPCK delay value during serial transfer by using the SCKDEN bit in the selected SPCMD and SPCKD, as shown in **Table 24.8**. For a definition of RSPCK delay, see **Section 24.4.4, Transfer Format**.

Table 24.8 Relationship among SCKDEN and SPCKD Settings and RSPCK Delay Values

| SCKDEN | SPCKD | RSPCK Delay Value |
|---|---|---|
| 0 | 000 to 111 | 1 RSPCK |
| 1 | 000 | 1 RSPCK |
| | 001 | 2 RSPCK |
| | 010 | 3 RSPCK |
| | 011 | 4 RSPCK |
| | 100 | 5 RSPCK |
| | 101 | 6 RSPCK |
| | 110 | 7 RSPCK |
| | 111 | 8 RSPCK |

### (f) SSL Negation Delay (t2)

The SSL negation delay value in master mode depends on SLNDEN bit settings in the command register (SPCMD) and on SSL negation delay register (SSLND) settings. This module determines the SPCMD to be referenced during serial transfer by pointer control, and determines an SSL negation delay value during serial transfer by using the SLNDEN bit in the selected SPCMD and SSLND, as shown in **Table 24.9**. For a definition of SSL negation delay, see **Section 24.4.4, Transfer Format**.

Table 24.9 Relationship among SLNDEN and SSLND Settings and SSL Negation Delay Values

| SLNDEN | SSLND | SSL Negation Delay Value |
|---|---|---|
| 0 | 000 to 111 | 1 RSPCK |
| 1 | 000 | 1 RSPCK |
| | 001 | 2 RSPCK |
| | 010 | 3 RSPCK |
| | 011 | 4 RSPCK |
| | 100 | 5 RSPCK |
| | 101 | 6 RSPCK |
| | 110 | 7 RSPCK |
| | 111 | 8 RSPCK |

### (g) Next-Access Delay (t3)

The next-access delay value in master mode depends on SPNDEN bit settings in the command register (SPCMD) and on next-access delay register (SPND) settings. This module determines the SPCMD to be referenced during serial transfer by pointer control, and determines a next-access delay value during serial transfer by using the SPNDEN bit in the selected SPCMD and SPND, as shown in **Table 24.10**. For a definition of next-access delay, see **Section 24.4.4, Transfer Format**.

Table 24.10 Relationship among SPNDEN and SPND Settings and Next-Access Delay Values

| SPNDEN | SPND | Next-Access Delay Value |
|---|---|---|
| 0 | 000 to 111 | 1 RSPCK + 2 P0ϕ |
| 1 | 000 | 1 RSPCK + 2 P0ϕ |
| | 001 | 2 RSPCK + 2 P0ϕ |
| | 010 | 3 RSPCK + 2 P0ϕ |
| | 011 | 4 RSPCK + 2 P0ϕ |
| | 100 | 5 RSPCK + 2 P0ϕ |
| | 101 | 6 RSPCK + 2 P0ϕ |
| | 110 | 7 RSPCK + 2 P0ϕ |
| | 111 | 8 RSPCK + 2 P0ϕ |

### (h) Initialization Flowchart

**Figure 24.17** is a flowchart illustrating an example of initialization in SPI operation when this module is used in master mode. For a description of how to set up the interrupt controller, direct memory access controller, and input/output ports, see the descriptions given in the individual blocks.

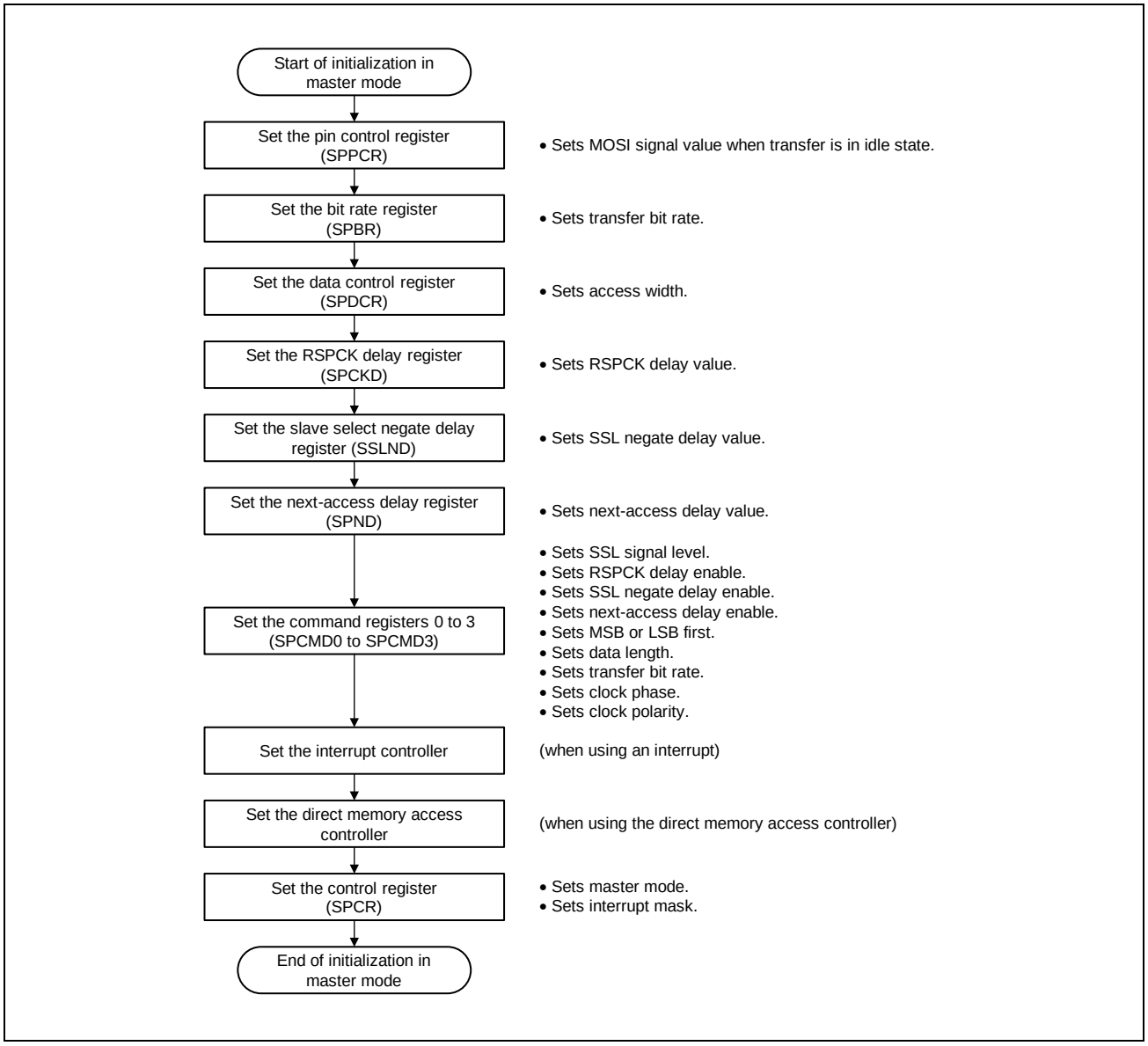



Figure 24.17 Example of Initialization Flowchart in Master Mode

### (i) Transfer Operation Flowchart

**Figure 24.18** is a flowchart illustrating a transfer in SPI operation when this module is used in master mode.

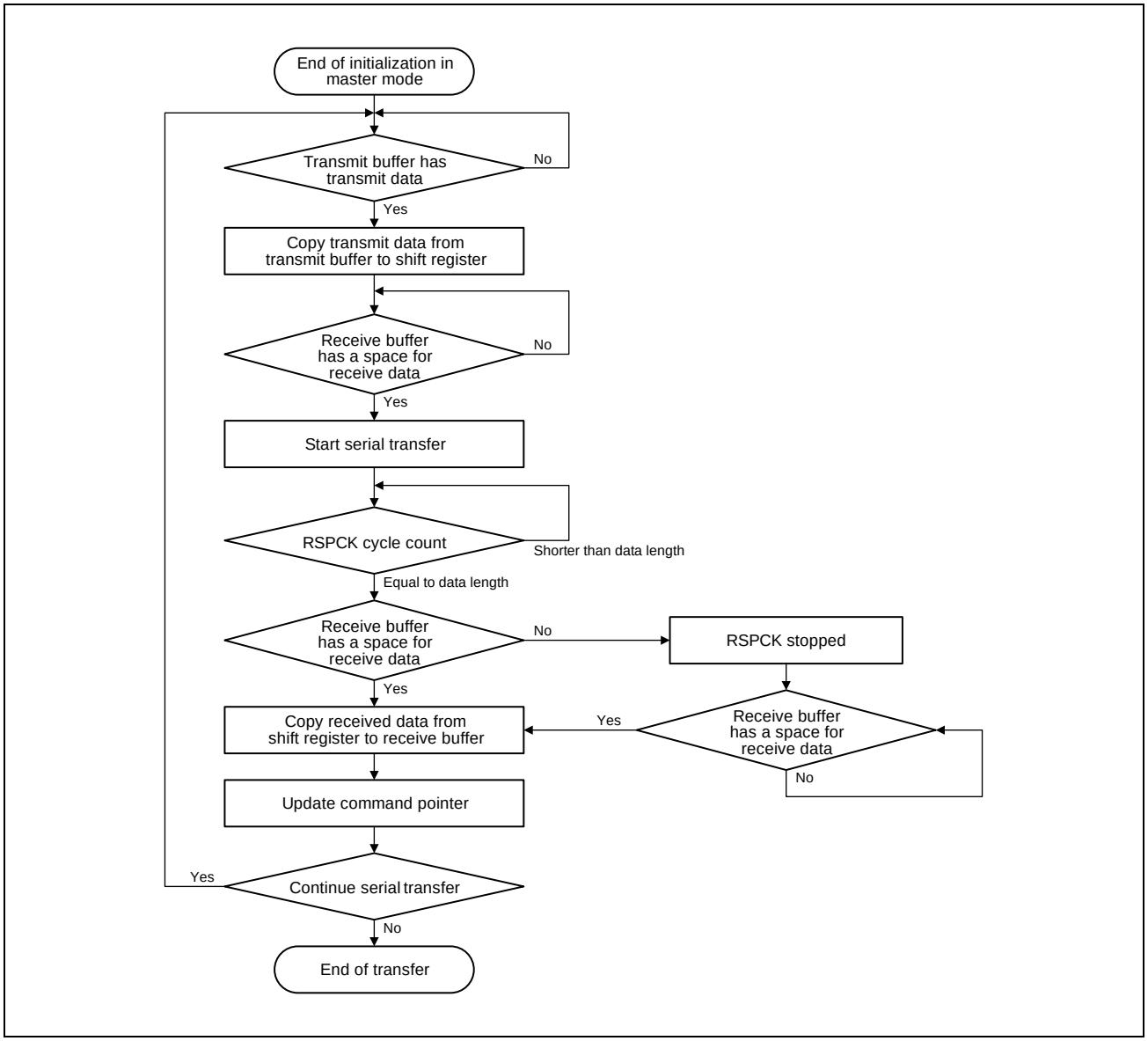



Figure 24.18 Transfer Operation Flowchart in Master Mode

### (2) Slave Mode Operation

#### (a) Starting Serial Transfer

If this module detects an SSL input signal assertion when the CPHA bit in the command register 0 (SPCMD0) is 0, this module is required to start driving valid data to the MISO output signal. For this reason, when the CPHA bit is 0, the asserting of the SSL input signal triggers the start of a serial transfer.

If this module detects the first RSPCK edge in an SSL signal asserted condition when the CPHA bit is 1, this module is required to start driving valid data to the MISO output signal. For this reason, when the CPHA bit is 1, the first RSPCK edge in an SSL signal asserted condition triggers the start of a serial transfer.

When detecting the start of a serial transfer in a condition in which the shift register is empty, this module changes the status of the shift register to “full”, so that data cannot be copied from the transmit buffer to the shift register when serial transfer is in progress. If the shift register was full before the serial transfer started, this module leaves the status of the shift register intact, in the full state.

Irrespective of CPHA bit settings, this module starts driving MISO output signals at the SSL signal assertion timing. Whether the data output from this module is valid or invalid differs depending on CPHA bit settings.

For details on the transfer format, see **Section 24.4.4, Transfer Format**. The polarity of the SSL input signal depends on the setting of the SSL0P bit in the slave select polarity register (SSLP).

#### (b) Terminating Serial Transfer

Irrespective of the CPHA bit in the command register 0 (SPCMD0), this module terminates the serial transfer after detecting an RSPCK edge corresponding to the final sampling timing. When the receive buffer has an enough space for receive data, this module copies received data from the shift register to the receive buffer of the data register (SPDR) upon termination of the serial transfer. Irrespective of the value of the SPRF bit, this module changes the status of the shift register to “empty” upon termination of the serial transfer. If this module detects an SSL input signal negation from the beginning of serial transfer to the end of serial transfer, a mode fault error occurs (see **Section 24.4.6, Error Detection**).

The final sampling timing changes depending on the bit length of the transfer data. In slave mode, the data length depends on the settings in bits SPB3 to SPB0 bits in SPCMD0. The polarity of the SSL input signal depends on the setting in the SSL0P bit in the slave select polarity register (SSLP). For details on the transfer format, see **Section 24.4.4, Transfer Format**.

#### (c) Notes on Slave Operations

If the CPHA bit in the command register 0(SPCMD0) is 0, this module starts serial transfers when it detects the assertion edge for an SSL input signal. In the type of configuration shown in **Figure 24.4** as an example, if this module is used in single-slave mode, the SSL signal is always fixed at active state. Therefore, when the CPHA bit is set to 0, this module cannot correctly start a serial transfer. To correctly execute send/receive operation in a configuration in which the SSL input signal is fixed at active state, the CPHA bit should be set to 1. When it is necessary to set the CPHA bit to 0, the SSL input signal should not be fixed.

### (d) Burst Transfer

If the CPHA bit in the command register 0 (SPCMD0) is 1, continuous serial transfer (burst transfer) can be executed while retaining the assertion state for the SSL input signal. If the CPHA bit is 1, the period from the first RSPCK edge to the sampling timing for the reception of the final bit in an SSL signal active state corresponds to a serial transfer period. Even when the SSL input signal remains at the active level, this module can accommodate burst transfers because it can detect the start of access.

If the CPHA bit is 0, for the reason given in **Section 24.4.8(2)(c), Notes on Slave Operations**, second and subsequent serial transfers during the burst transfer cannot be executed correctly.

### (e) Initialization Flowchart

**Figure 24.19** is a flowchart illustrating an example of initialization in SPI operation when this module is used in slave mode. For a description of how to set up the interrupt controller, direct memory access controller, and input/output ports, see the descriptions given in the individual blocks.

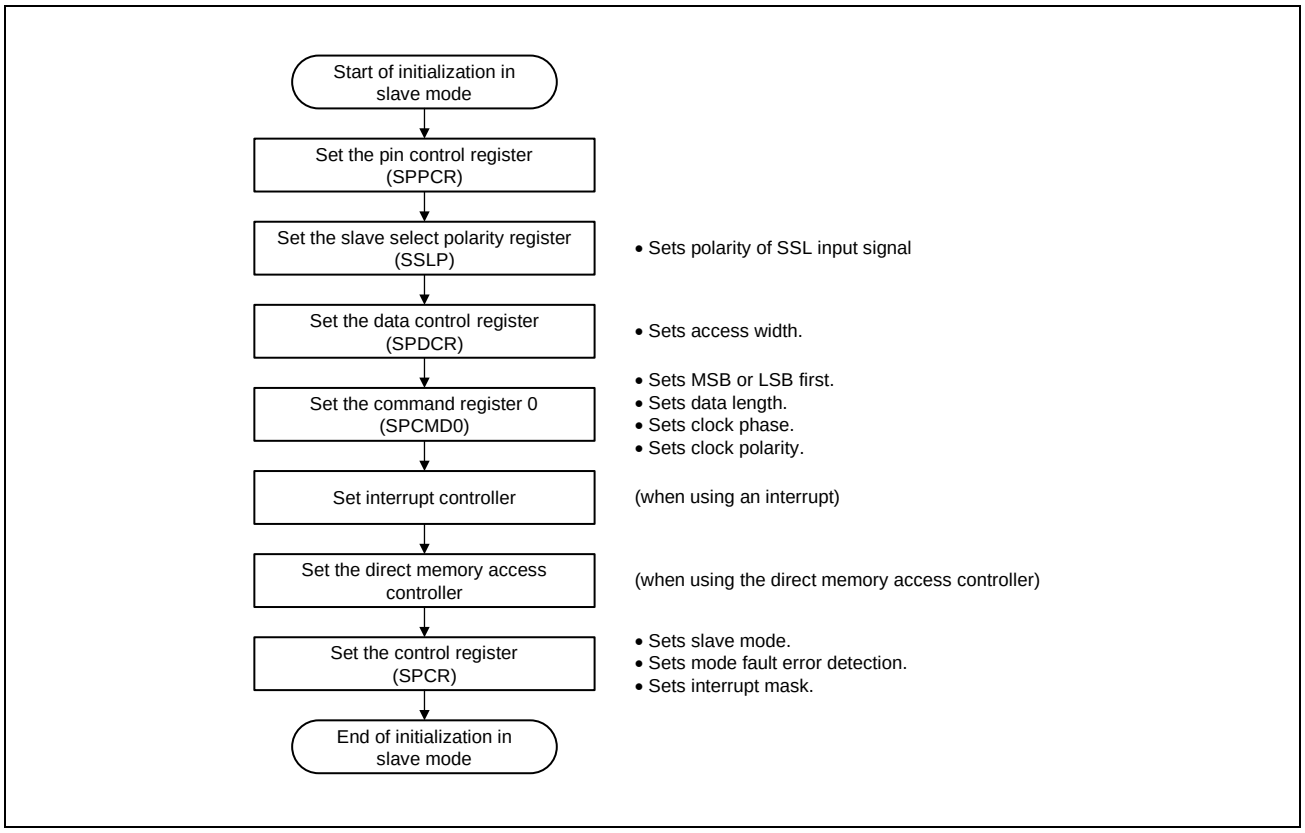



Figure 24.19 Example of Initialization Flowchart in Slave Mode

### (f) Transfer Operation Flowchart (CPHA = 0)

**Figure 24.20** is a flowchart illustrating a transfer in SPI operation when this module is used in slave mode with the CPHA bit in the command register 0 (SPCMD0) set to 0.

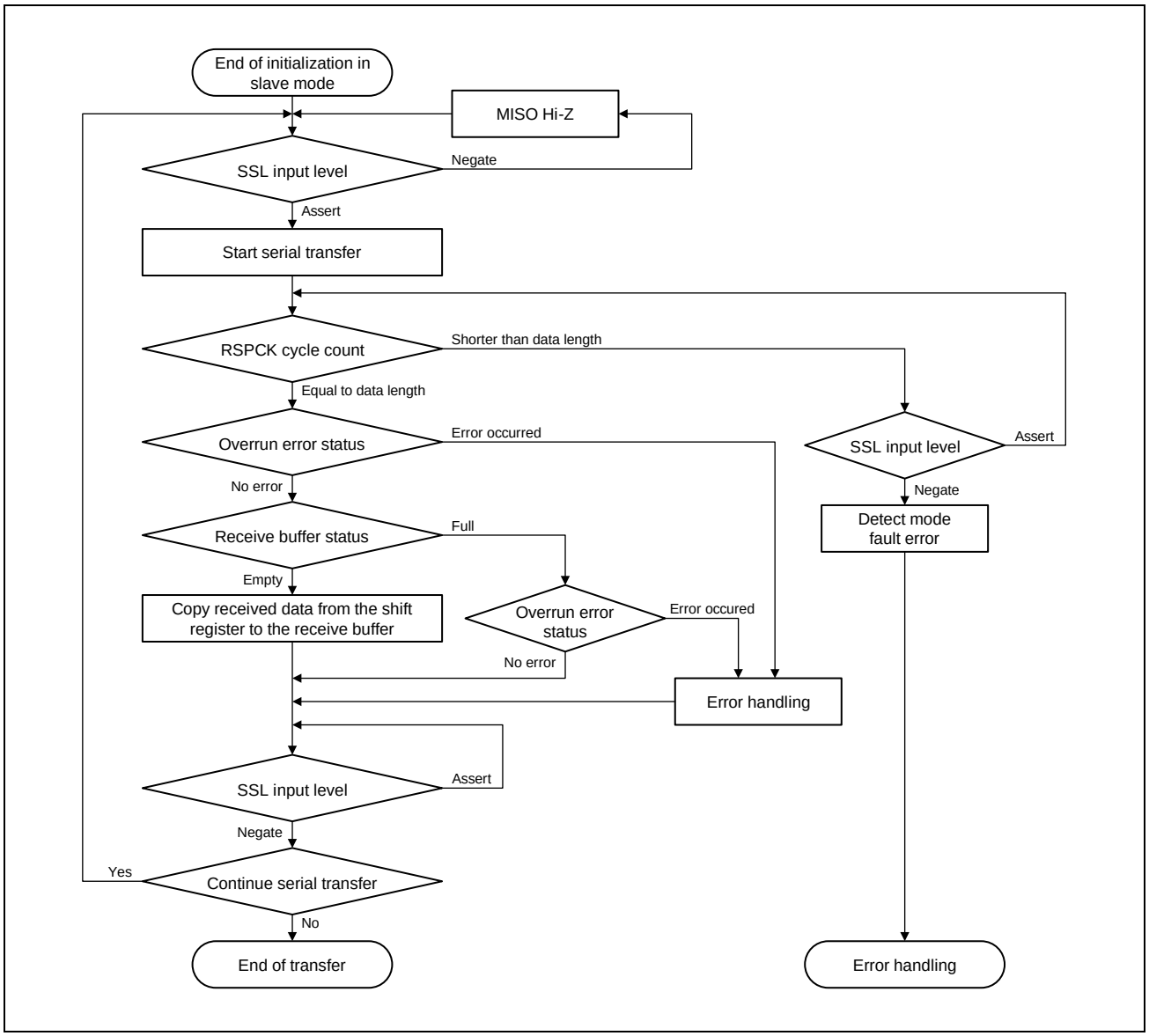



Figure 24.20 Transfer Operation Flowchart in Slave (CPHA = 0)

### (g) Transfer Operation Flowchart (CPHA = 1)

**Figure 24.21** is a flowchart illustrating a transfer in SPI operation when this module is used in slave mode with the CPHA bit in the command register 0 (SPCMD0) and the MODFEN bit in the control register (SPCR) set to 1, respectively. The subsequent operation is not guaranteed when the serial transfer is started with the MODFEN bit set to 0 and the SSL input level is negated with the number of RSPCK cycles shorter than the data length.

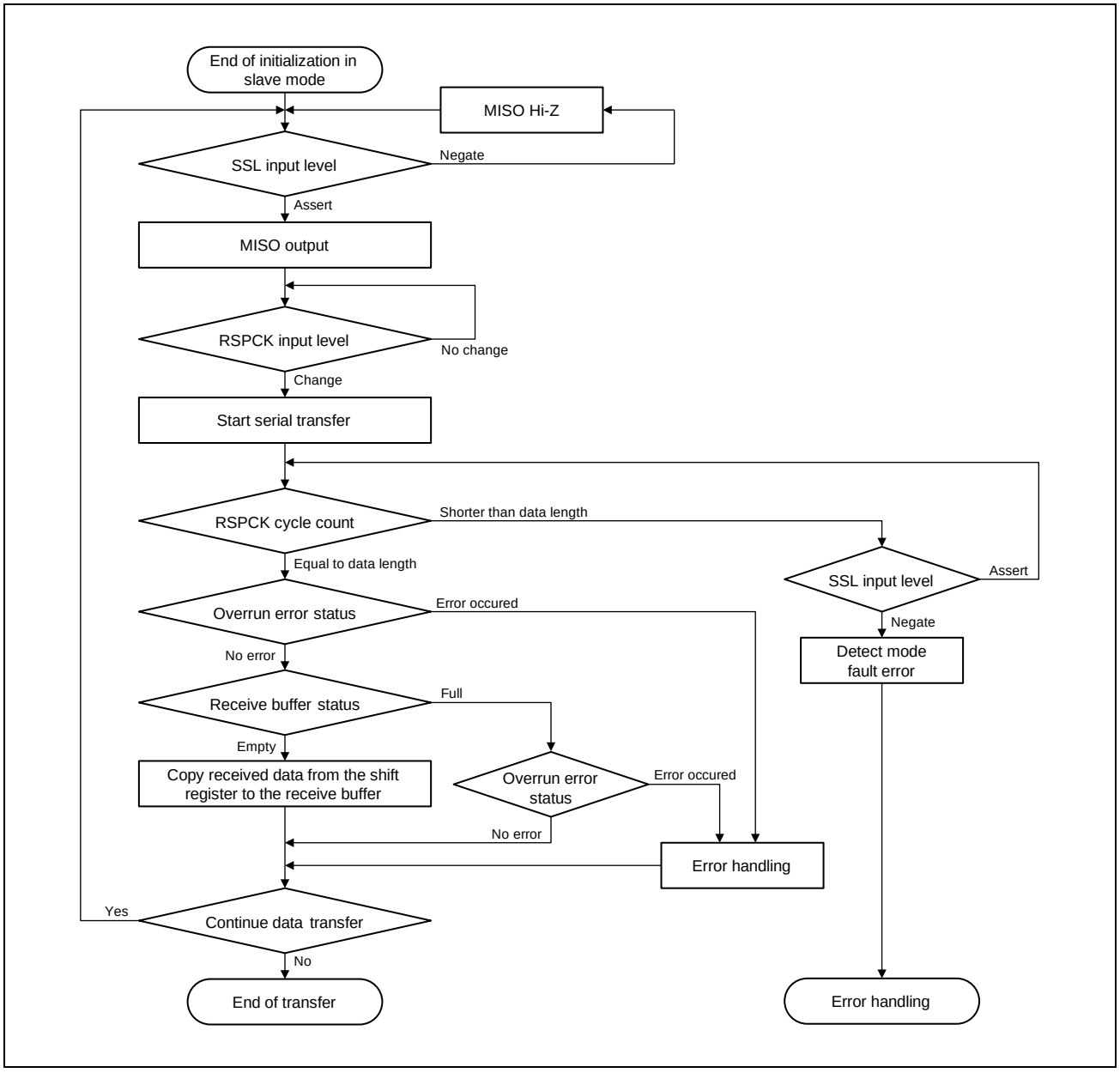



Figure 24.21 Transfer Operation Flowchart in Slave Mode (CPHA = 1)

### 24.4.9 Error Handling

**Figure 24.22** and **Figure 24.23** show the error handling. The following error handling is used to return from the error state after an error in master or slave mode.

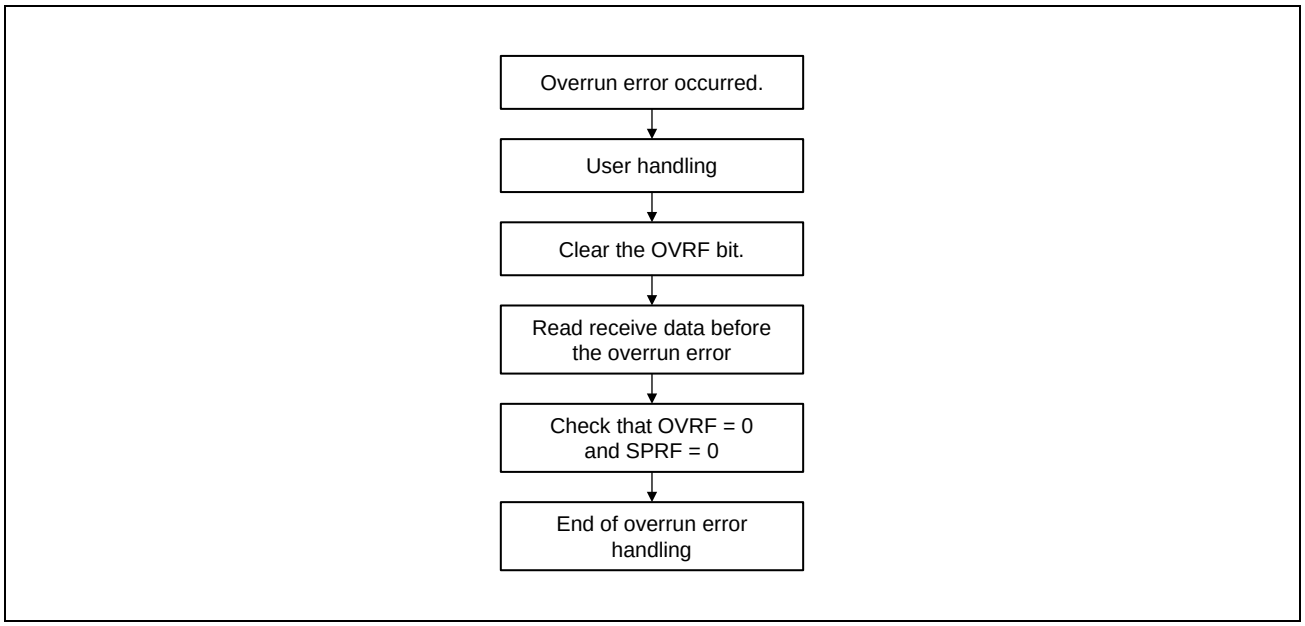



Figure 24.22 Error Handling (Overrun Error)

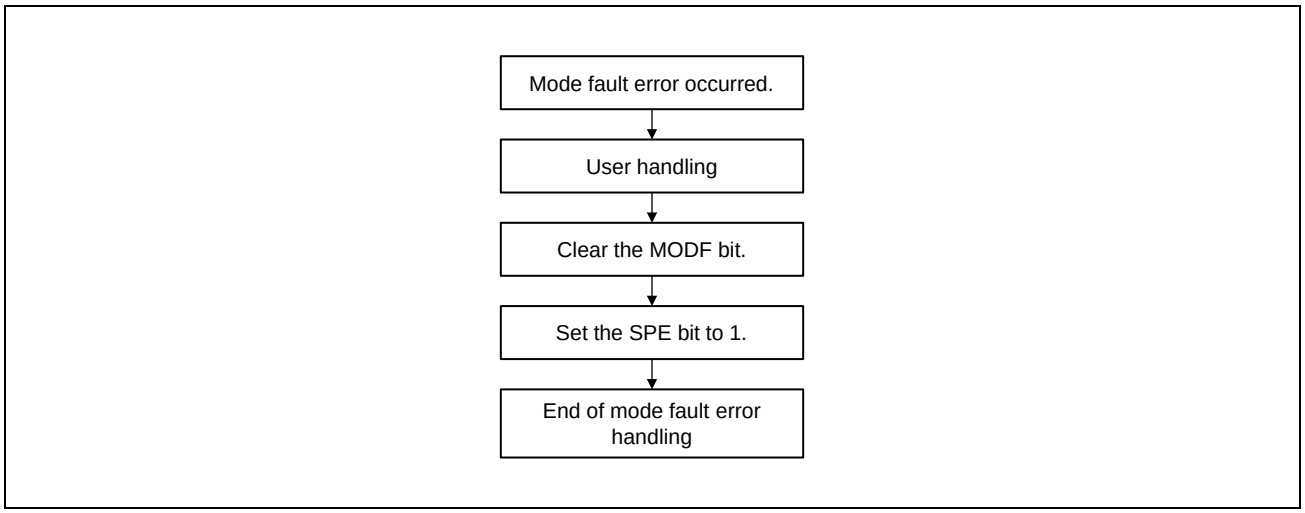



Figure 24.23 Error Handling (Mode Fault Error)

### 24.4.10 Loopback Mode

When 1 is written to the SPLP bit in the pin control register (SPPCR), this module shuts off the path between the MISO pin and the shift register, and between the MOSI pin and the shift register, and connects the input path and the output path (reversed) of the shift register. This is called loopback mode. When a serial transfer is executed in loopback mode, the transmit data becomes the received data. **Figure 24.24** shows the configuration of the shift register input/output paths for the case where this module in master mode is set in loopback mode.

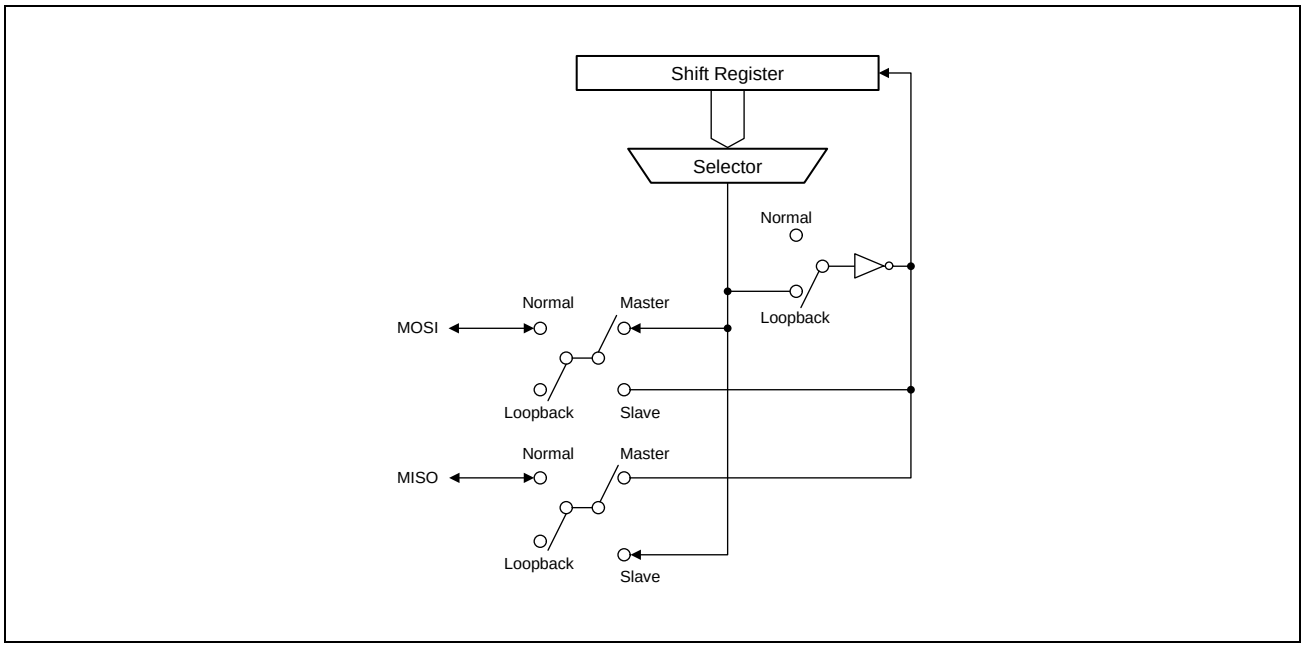



Figure 24.24 Configuration of Shift Register Input/Output Paths in Loopback Mode (Master Mode)

### 24.4.11 Interrupt Sources

This module has interrupt sources of receive buffer full, transmit buffer empty, mode fault, and overrun. In addition, the direct memory access controller can be activated by the receive buffer full or transmit buffer empty interrupt for data transfer.

**Table 24.11** shows the interrupt sources.

When any of the interrupt conditions in **Table 24.11** is met, an interrupt is generated. The interrupt sources should be cleared with data transfer by the CPU or direct memory access controller.

Table 24.11 Interrupt Sources

| Name | Interrupt Source | Abbreviation | Interrupt Condition | Activation of Direct Memory Access Controller |
|---|---|---|---|---|
| SPRI | Receive buffer full | RXI | (SPRIE = 1) • (SPRF = 1) | Possible |
| SPTI | Transmit buffer empty | TXI | (SPTIE = 1) • (SPTEF = 1) | Possible |
| SPEI | Mode fault | MOI | (SPEIE = 1) • (MODF = 1) | — |
| | Overrun | OVI | (SPEIE = 1) • (OVRF = 1) | — |

# 25. SPI Multi I/O Bus Controller

The SPI multi I/O bus controller enables the direct connection of serial flash memory device to this LSI chip.

## 25.1 Features

This module allows the connected serial flash memory device to be accessed by reading the external address space, or using Manual mode to transmit and receive data.

### 25.1.1 Serial Flash Memory Interface

- SPI_PVDD: 1.8V/3.3V
- QSPI0_SPCLK: 66MHz (SDR), 50MHz (DDR)
- A data bus size of 1 bit or 4 bits can be selected for one serial flash memory device.
- Both single data rate (SDR) and double data rate (DDR) transfers are supported.
- Supports master operation. (Slave operation is not supported.)

### 25.1.2 External Address Space Read Mode

- Read access from the bus master to the SPI multi I/O space is automatically converted to a read command, and the read data is returned to the bus master.
- Normal read operation and burst read operation
- Efficient data reception due to built-in read cache (64-bit line × 32 entries)

### 25.1.3 Manual Mode

- Read and write commands are available for serial flash memory.
- A write buffer is provided to improve the efficiency of data write operations.
- The cache is not usable for reading.

## 25.2 Block Diagram

**Figure 25.1** shows a block diagram of this module.

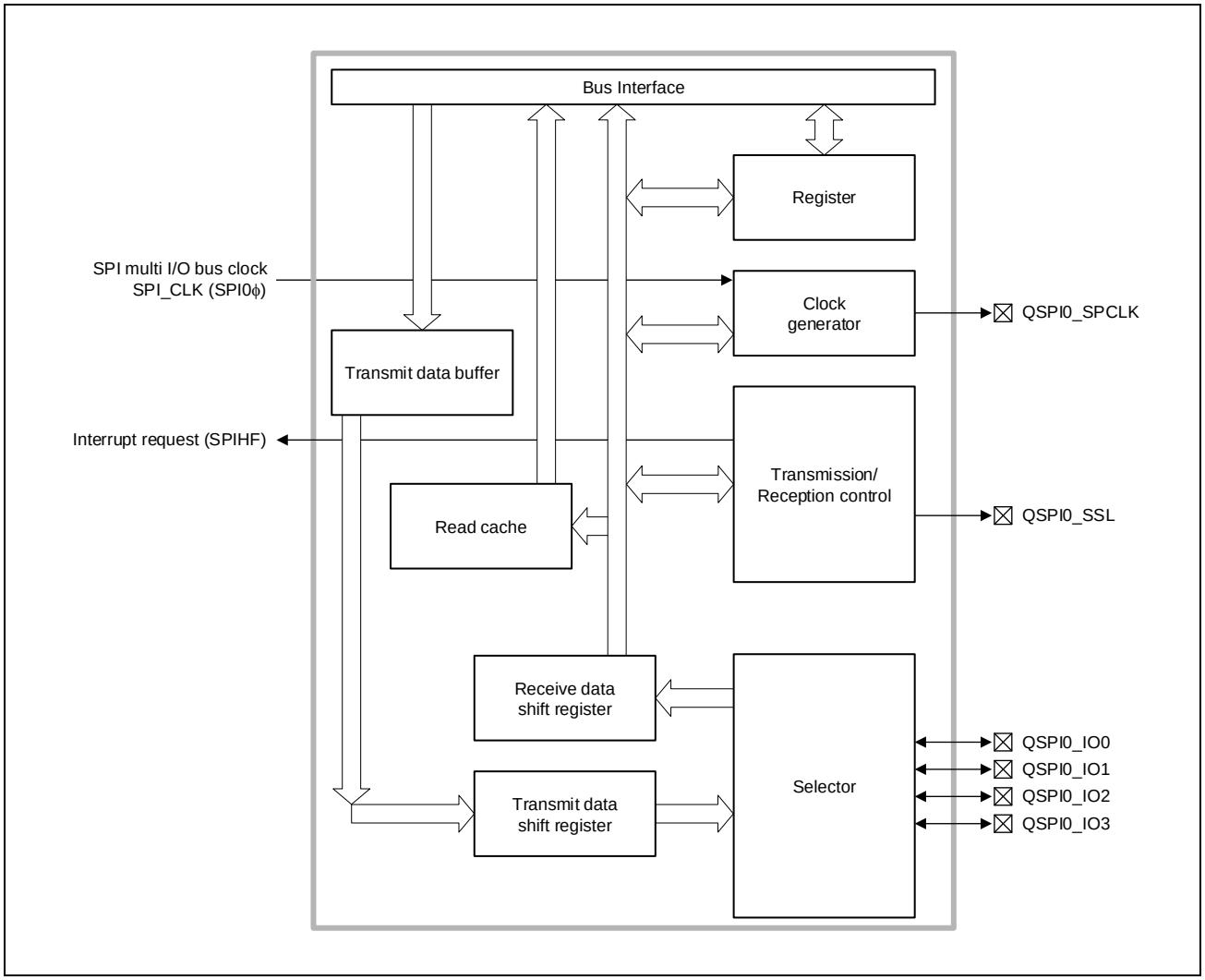



Figure 25.1 Block Diagram

## 25.3 Input/Output Pins

**Table 25.1** shows the pin configuration for one channel.

The operating voltage on pins of the SPI multi I/O bus controller is 3.3 V or 1.8 V.

Before using the pins of the SPI multi I/O bus controller, be sure to set up the QSPI IO Voltage Control Register(QSPI) and Driving Ability Control Register (IOLH).

For details, see **Section 41.3.15, QSPI IO Voltage Control Register(QSPI)** and **Section 41.3.7, Driving Ability Control Register(IOLH)**.

Table 25.1 Pin Configuration

| Pin Name | Symbol | I/O | One 4-bit Serial Flash connected.<br>CMNCR.BSZ = 00 |
|---|---|---|---|
| Clock*1 | QSPI0_SPCLK*3*2 | Output | SCK |
| Data 0*1 | QSPI0_IO0 | I/O | SI/IO0 |
| Data 1*1 | QSPI0_IO1 | I/O | SO/IO1 |
| Data 2*1 | QSPI0_IO2 | I/O | IO2 |
| Data 3*1 | QSPI0_IO3 | I/O | IO3 |
| Slave select*1 | QSPI0_SSL | Output | CS# |

Note 1. When connecting a serial flash memory device with the BSZ[1:0] bits set to 00, connect the serial flash memory to the QSPI0 pin group of this LSI.

Note 2. QSPI0_SPCLK is one-half frequency of SPI0ϕ.

## 25.4 Register Descriptions

**Table 25.2** shows the register configuration.

Base Address: H'0_1006_0000 (Cortex-A55 Address Space)

Table 25.2 Register Configuration

| Register Name | Abbreviation | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|
| Common control register | CMNCR | R/W | H'01557301 | H'0000 | 32 |
| SSL delay register | SSLDR | R/W | H'00000000 | H'0004 | 32 |
| Data read control register | DRCR | R/W | H'001F0100 | H'000C | 32 |
| Data read command setting register | DRCMR | R/W | H'00A00000 | H'0010 | 32 |
| Data read extended address setting register | DREAR | R/W | H'00000000 | H'0014 | 32 |
| Data read option setting register | DROPR | R/W | H'00000000 | H'0018 | 32 |
| Data read enable setting register | DRENR | R/W | H'A222D400 | H'001C | 32 |
| Manual mode control register | SMCR | R/W | H'00000000 | H'0020 | 32 |
| Manual mode command setting register | SMCMR | R/W | H'00000000 | H'0024 | 32 |
| Manual mode address setting register | SMADR | R/W | H'00000000 | H'0028 | 32 |
| Manual mode option setting register | SMOPR | R/W | H'00000000 | H'002C | 32 |
| Manual mode enable setting register | SMENR | R/W | H'00004000 | H'0030 | 32 |
| Manual mode read data register 0 | SMRDR0 | R | Undefined | H'0038 | 8, 16, 32 |
| Manual mode write data register 0 | SMWDR0 | R/W | H'00000000 | H'0040 | 8, 16, 32 |
| Common status register | CMNSR | R | H'00000001 | H'0048 | 32 |
| Data read dummy cycle setting register | DRDMCR | R/W | H'0000000B | H'0058 | 32 |
| Data read DDR enable register | DRDRENR | R/W | H'00005101 | H'005C | 32 |
| Manual mode dummy cycle setting register | SMDMCR | R/W | H'00000000 | H'0060 | 32 |
| Manual mode DDR enable register | SMDRENR | R/W | H'00000000 | H'0064 | 32 |
| PHY control register | PHYCNT | R/W | H'00000263 | H'007C | 32 |
| PHY offset register 1 | PHYOFFSET1 | R/W | H'21511144 | H'0080 | 32 |
| PHY offset register 2 | PHYOFFSET2 | R/W | H'00000431 | H'0084 | 32 |
| PHY adjustment register 1 | PHYADJ1 | R/W | H'00000000 | H'0070 | 32 |
| PHY adjustment register 2 | PHYADJ2 | R/W | H'00000000 | H'0074 | 32 |

**Note:** Do not write to any address those listed in the table. Otherwise, correct operation cannot be guaranteed. Reading from a non-listed address returns an undefined value.

### 25.4.1 Common Control Register (CMNCR)

CMNCR is a 32-bit register that controls the SPI multi I/O bus controller. The settings of this register are reflected both in external address space read mode and manual mode.

The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MD | — | — | — | — | — | — | — | MOIIO3[1:0] | | MOIIO2[1:0] | | MOIIO1[1:0] | | MOIIO0[1:0] | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| R/W | 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 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | IO3FV[1:0] | | IO2FV[1:0] | | — | — | IO0FV[1:0] | | — | — | — | — | — | — | BSZ[1:0] | |
| Initial Value | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| R/W | R/W | R/W | R/W | R/W | R | R | R/W | R/W | R | R | R | R | R | R | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 | MD | 0 | R/W | Operating Mode Switch<br>Switches the operating modes.<br>0: External address space read mode<br>1: Manual mode |
| 30 to 25 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 24 | — | 1 | R | Reserved<br>This bit is always read as 1. The write value should always be 1. |
| 23, 22 | MOIIO3[1:0] | 01 | R/W | QSPIn_SSL Output Idle Value Fix QSPIn_IO3<br>Fixes the output value on QSPIn_IO3 while QSPIn_SSL is negated.<br>00: The output value is 0.<br>01: The output value is 1.<br>10: The output value is that of the last bit in the previous transfer (the pin is placed in the Hi-Z state if that was the case in the previous transfer).<br>11: The pin is placed in the Hi-Z state. |
| 21, 20 | MOIIO2[1:0] | 01 | R/W | QSPIn_SSL Output Idle Value Fix QSPIn_IO2<br>Fixes the output value on QSPIn_IO2 while QSPIn_SSL is negated.<br>00: The output value is 0.<br>01: The output value is 1.<br>10: The output value is that of the last bit in the previous transfer (the pin is placed in the Hi-Z state if that was the case in the previous transfer).<br>11: The pin is placed in the Hi-Z state. |
| 19, 18 | MOIIO1[1:0] | 01 | R/W | QSPIn_SSL Output Idle Value Fix QSPIn_IO1<br>Fixes the output value on QSPIn_IO1 while QSPIn_SSL is negated.<br>00: The output value is 0.<br>01: The output value is 1.<br>10: The output value is that of the last bit in the previous transfer (the pin is placed in the Hi-Z state if that was the case in the previous transfer).<br>11: The pin is placed in the Hi-Z state. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 17, 16 | MOIIO0[1:0] | 01 | R/W | QSPIn_SSL Output Idle Value Fix QSPIn_IO0<br>Fixes the output value on QSPIn_IO0 while QSPIn_SSL is negated.<br>00: The output value is 0.<br>01: The output value is 1.<br>10: The output value is that of the last bit in the previous transfer (the pin is placed in the Hi-Z state if that was the case in the previous transfer).<br>11: The pin is placed in the Hi-Z state. |
| 15, 14 | IO3FV[1:0] | 01 | R/W | QSPIn_IO3 Fixed Value for 1-bit Size<br>Fixes the output value of QSPIn_IO3 pin for 1-bit size.<br>00: The output value is 0.<br>01: The output value is 1.<br>10: The output value is that of the last bit in the previous transfer (the pin is placed in the Hi-Z state if that was the case in the previous transfer).<br>11: The pin is placed in the Hi-Z state. |
| 13, 12 | IO2FV[1:0] | 11 | R/W | QSPIn_IO2 Fixed Value for 1-bit Size<br>Fixes the output value of QSPIn_IO2 pin for 1-bit size.<br>00: The output value is 0.<br>01: The output value is 1.<br>10: The output value is that of the last bit in the previous transfer (the pin is placed in the Hi-Z state if that was the case in the previous transfer).<br>11: The pin is placed in the Hi-Z state. |
| 11, 10 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 9, 8 | IO0FV[1:0] | 11 | R/W | QSPIn_IO0 Fixed Value for 1-bit Size<br>Fixes the output value of QSPIn_IO0 pin for 1-bit size.<br>00: The output value is 0.<br>01: The output value is 1.<br>10: The output value is that of the last bit in the previous transfer (the pin is placed in the Hi-Z state if that was the case in the previous transfer).<br>11: The pin is placed in the Hi-Z state. |
| 7 to 2 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 1, 0 | BSZ[1:0] | 01 | R/W | Data Bus Size<br>Specifies the number of serial flash memories to be connected.<br>00: one serial flash memory connected<br>01: Setting prohibited<br>1X: Setting prohibited<br>*Note*: After changing (the value of) this bit field, all the entries in the read cache must be cleared by setting the RCF bit in DRCR to 1. |

**Note:** The settings of the MOIIO3, MOIIO2, MOIIO1, and MOIIO0 bits for control over fixing the output values are reflected at the time QSPIn_SSL is negated on completion of data transfer.

### 25.4.2 SSL Delay Register (SSLDR)

SSLDR is a 32-bit register that adjusts the timing between the QSPIn_SSL signal and the QSPIn_SPCLK signal. The settings of this register are reflected both in external address space read mode and manual mode.

The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | SPNDL[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 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | SLNDL[2:0] | | | — | — | — | — | — | SCKDL[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/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<br>These bits are always read as 0. The write value should always be 0. |
| 18 to 16 | SPNDL[2:0] | 000 | R/W | Next Access Delay<br>Sets the period from transfer end to next transfer start (next access).<br>000: 1 QSPIn_SPCLK (3 QSPIn_SPCLK when SSLE = 0)<br>001: 2 QSPIn_SPCLK<br>010: 3 QSPIn_SPCLK<br>011: 4 QSPIn_SPCLK<br>100: 5 QSPIn_SPCLK<br>101: 6 QSPIn_SPCLK<br>110: 7 QSPIn_SPCLK<br>111: 8 QSPIn_SPCLK |
| 15 to 11 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 10 to 8 | SLNDL[2:0] | 000 | R/W | QSPIn_SSL Negation Delay<br>Sets the period from the time the last QSPIn_SPCLK edge is sent of a transfer to QSPIn_SSL pin negation (QSPIn_SSL negation delay).<br>000: 1 QSPIn_SPCLK<br>001: 2 QSPIn_SPCLK<br>010: 3 QSPIn_SPCLK<br>011: 4 QSPIn_SPCLK<br>100: 5 QSPIn_SPCLK<br>101: 6 QSPIn_SPCLK<br>110: 7 QSPIn_SPCLK<br>111: 8 QSPIn_SPCLK |
| 7 to 3 | — | All 0 | R | Reserved<br>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 | SCKDL[2:0] | 000 | R/W | Clock Delay<br>Sets the period from QSPIn_SSL pin assertion to QSPIn_SPCLK oscillation (clock delay).<br>000: 1.5 QSPIn_SPCLK<br>001: 2.5 QSPIn_SPCLK<br>010: 3.5 QSPIn_SPCLK<br>011: 4.5 QSPIn_SPCLK<br>100: 5.5 QSPIn_SPCLK<br>101: 6.5 QSPIn_SPCLK<br>110: 7.5 QSPIn_SPCLK<br>111: 8.5 QSPIn_SPCLK |

### 25.4.3 Data Read Control Register (DRCR)

DRCR is a 32-bit register that sets the operation in external address space read mode.

The bits except the SSLN bit should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | SSLN | — | — | — | RBURST[4:0] | | | | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 |
| R/W | R | R | R | R | R | R | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | RCF | RBE | — | — | — | — | — | — | — | SSLE |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | 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 25 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 24 | SSLN | 0 | W | QSPIn_SSL Negation<br>Asserted QSPIn_SSL can be negated by writing 1 to this bit when both the RBE and SSLE bits are 1.<br>This bit is always read as 0.<br>*Note*: To start next access after QSPIn_SSL negation using this bit, read SSLF in CMNSR = 0 to confirm that the QSPIn_SSL has been negated. |
| 23 to 21 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 20 to 16 | RBURST[4:0] | 11111 | R/W | Read Data Burst Length<br>Sets the burst length (data unit count) when reading.<br>This bit field is enabled when the RBE bit is set to 1.<br>00000: 1 data unit<br>00001: 2 continuous data units<br>:<br>11110: 31 continuous data units<br>11111: 32 continuous data units<br>One data unit is 64 bits long. |
| 15 to 10 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 9 | RCF | 0 | W | Read Cache Flush<br>When 1 is written to this bit, all the entries in the read cache are cleared.<br>This bit is always read as 0.<br>After using the cache area as the write buffer, the cache area must be cleared by writing 1 to the RCF bit.<br>*Note*: After flushing the read cache by writing 1 to the RCF bit, read the DRCR register before proceeding to read from the external address space. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 8 | RBE | 1 | R/W | Read Burst<br>Turns burst ON or OFF when reading.<br>0: Data is read according to the access size.<br>1: Read cache is enabled, and as many data units as the burst count specified in RBURST[4:0] bits is read.<br>*Note*: When the cache is enabled and read access reaches the last address of the flash memory, the access address does not match the address held in the cache. Control the read address and access size so that read access does not reach the last address of the flash memory. |
| 7 to 1 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 0 | SSLE | 0 | R/W | QSPIn_SSL Negation<br>Sets the conditions for QSPIn_SSL negation during read burst.<br>QSPIn_SSL is negated for each access during normal read.<br>0: QSPIn_SSL is negated after transfer of data set in burst length.<br>1: QSPIn_SSL is negated when the accessed address is not continuous with the previously transferred address.<br>*Note 1.* Set up this bit when a serial flash memory device is connected.<br>*Note 2.* When this bit is set to 1, setting the RBURST[4:0] bits to H'01 is prohibited. |

### 25.4.4 Data Read Command Setting Register (DRCMR)

DRCMR is a 32-bit register that sets the commands issued in external address space read mode.

The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | CMD[7:0] | | | | | | | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 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/W | R/W | R/W | R/W |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | OCMD[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 24 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 23 to 16 | CMD[7:0] | H'A0 | R/W | Command<br>Sets the command. |
| 15 to 8 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 7 to 0 | OCMD[7:0] | H'00 | R/W | Optional Command<br>Sets the optional command. |

### 25.4.5 Data Read Extended Address Setting Register (DREAR)

DREAR is a 32-bit register that sets the address when the serial flash address is output in 32-bit mode.

The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | EAV[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 |

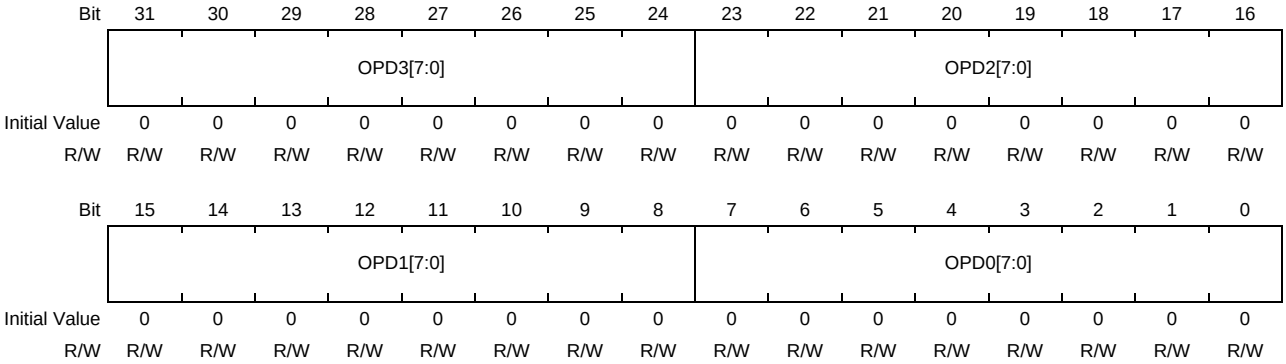
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | EAC[2:0] | | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 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 24 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 23 to 16 | EAV[7:0] | H'00 | R/W | 32-Bit Extended Upper Address Fixed Value<br>Sets the upper address bit values of the external address specified by the EAC[2:0] bits when the serial flash address is output in 32-bit mode.<br>Bit 0 corresponds to the serial flash address bit [25], and bit 7 corresponds to the bit [32].<br>This setting is valid when the ADE[3] bit in DRENR is 1.<br>When EAC[2:0] are 000, serial flash address [32:25] fixed values should set to EAV[7:0].<br>When EAC[2:0] are 001, serial flash address [32:26] fixed values should set to EAV[7:1].<br>When EAC[2:0] are 010, serial flash address [32:27] fixed values should set to EAV[7:2].<br>When EAC[2:0] are 011, serial flash address [32:28] fixed values should set to EAV[7:3].<br>When BSZ[1:0] in CMNCR = 00 (one serial flash memory connected) Serial flash addresses [31:0] are used for accessing. |
| 15 to 3 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 2 to 0 | EAC[2:0] | 000 | R/W | 32-Bit Extended External Address Valid Range<br>Sets the range of the external address to be used as serial flash address when the serial flash address is output in 32-bit mode.<br>This setting is valid when the ADE[3] bit in DRENR is 1.<br>000: External address bits [24:0] enabled<br>001: External address bits [25:0] enabled<br>010: External address bits [26:0] enabled<br>011: External address bits [27:0] enabled<br>Other than above: Setting prohibited |

### 25.4.6 Data Read Option Setting Register (DROPR)

DROPR is a 32-bit register that sets the option data in external address space read mode.

The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 24 | OPD3[7:0] | H'00 | R/W | Option Data 3<br>Sets the option data 3. |
| 23 to 16 | OPD2[7:0] | H'00 | R/W | Option Data 2<br>Sets the option data 2. |
| 15 to 8 | OPD1[7:0] | H'00 | R/W | Option Data 1<br>Sets the option data 1. |
| 7 to 0 | OPD0[7:0] | H'00 | R/W | Option Data 0<br>Sets the option data 0. |

**Note:** OPD3, OPD2, OPD1, and OPD0 are output in this order.

### 25.4.7 Data Read Enable Setting Register (DRENR)

DRENR is a 32-bit register that sets the bit size of the command, optional command, address, option data, and read data in external address space read mode and enables outputting them other than read data.

The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | CDB[1:0] | | OCDB[1:0] | | — | — | ADB[1:0] | | — | — | OPDB[1:0] | | — | — | DRDB[1:0] | |
| Initial Value | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 |
| R/W | R/W | R/W | R/W | R/W | R | R | R/W | R/W | R | R | R/W | R/W | R | R | R/W | R/W |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | DME | CDE | — | OCDE | ADE[3:0] | | | | OPDE[3:0] | | | | — | — | — | — |
| Initial Value | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 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 | R | R | R |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31, 30 | CDB[1:0] | 10 | R/W | Command Bit Size<br>Sets the command size in bit units.<br>00: 1 bit<br>10: 4 bits<br>Other than above: Setting prohibited |
| 29, 28 | OCDB[1:0] | 10 | R/W | Optional Command Bit Size<br>Sets the optional command size in bit units.<br>00: 1 bit<br>10: 4 bits<br>Other than above: Setting prohibited |
| 27, 26 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 25, 24 | ADB[1:0] | 10 | R/W | Address Bit Size<br>Sets the address size in bit units.<br>00: 1 bit<br>10: 4 bits<br>Other than above: Setting prohibited |
| 23, 22 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 21, 20 | OPDB[1:0] | 10 | R/W | Option Data Bit Size<br>Sets the option data size in bit units.<br>00: 1 bit<br>10: 4 bits<br>Other than above: Setting prohibited |
| 19, 18 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 17, 16 | DRDB[1:0] | 10 | R/W | Data Read Bit Size<br>Sets the data read size in bit units.<br>00: 1 bit<br>10: 4 bits<br>Other than above: Setting prohibited |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 | DME | 1 | R/W | Dummy Cycle Enable<br>Enables insertion of the dummy cycle before the read data.<br>*Note*: A setting is prohibited for a transfer starting with a dummy cycle.<br>0: Dummy cycle insertion disabled<br>1: Dummy cycle insertion enabled |
| 14 | CDE | 1 | R/W | Command Enable<br>Sets whether or not the command is to be output.<br>0: Command output disabled<br>1: Command output enabled |
| 13 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 12 | OCDE | 1 | R/W | Optional Command Enable<br>Sets whether or not the optional command is to be output.<br>0: Optional command output disabled<br>1: Optional command output enabled |
| 11 to 8 | ADE[3:0] | 0100 | R/W | Address Enable<br>Sets the address to be output.<br>Be sure to use the following setting; otherwise, the operation is not guaranteed.<br>When a serial flash memory device is connected<br>0000: Output disabled<br>0111: Address[23:0]<br>1111: Address[31:0]<br>Other than above: Setting prohibited |
| 7 to 4 | OPDE[3:0] | 0000 | R/W | Option Data Enable<br>Sets the option data to be output.<br>Use only the settings given below. Otherwise, the operation cannot be guaranteed.<br>0000: Output disabled<br>1000: OPD3<br>1100: OPD3, OPD2<br>1110: OPD3, OPD2, OPD1<br>1111: OPD3, OPD2, OPD1, OPD0<br>Other than above: Setting prohibited |
| 3 to 0 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |

### 25.4.8 Manual Mode Control Register (SMCR)

SMCR is a 32-bit register that sets the operation in Manual mode.

The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

| 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 |

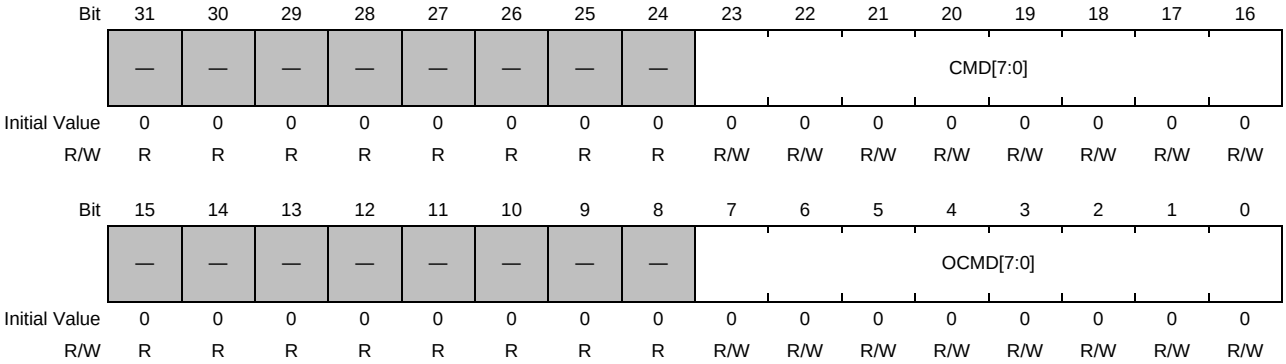
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | SSLKP | — | — | — | — | — | SPIRE | SPIWE | SPIE |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R/W | R | R | R | R | R | R/W | R/W | W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 9 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 8 | SSLKP | 0 | R/W | QSPIn_SSL Signal Level<br>Determines the QSPIn_SSL status after the end of transfer.<br>0: QSPIn_SSL signal is negated at the end of transfer.<br>1: QSPIn_SSL signal level is maintained from the end of transfer to the start of next access.<br>*Note*: The setting to start data transfer for reading is prohibited while the setting of SSLKP is 1. |
| 7 to 3 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 2 | SPIRE | 0 | R/W | Data Read Enable<br>Enables reading in manual mode.<br>0: Data reading disabled<br>1: Data reading enabled<br>*Note*: When the transfer data bit size is set to 4 bits with the SPIDB[1:0] bits, the SPIRE and SPIWE bits should not be set to 1 at the same time. |
| 1 | SPIWE | 0 | R/W | Data Write Enable<br>Enables writing in manual mode.<br>0: Data writing disabled<br>1: Data writing enabled<br>*Note*: When the transfer data bit size is set to 4 bits with the SPIDB[1:0] bits, the SPIRE and SPIWE bits should not be set to 1 at the same time. |
| 0 | SPIE | 0 | W | SPI Data Transfer Enable<br>Data is transferred by setting this bit to 1.<br>This bit is enabled only when the TEND bit in CMNSR is set to 1. The operation cannot be guaranteed when this bit is set to 1 with the TEND bit set to 0.<br>This bit is always read as 0.<br>*Note*: When the QSPIn_SSL pin is de-asserted, the command, optional command, address, and option data that are output enabled are output even if the SPIRE and SPIWE bits are set to 0. When the QSPIn_SSL pin is asserted, follow the notes described in **Section 25.6.3, Notes on Starting Transfer from the QSPIn_SSL Retained State in Manual Mode**. |

### 25.4.9 Manual Mode Command Setting Register (SMCMR)

SMCMR is a 32-bit register that sets the commands issued in Manual mode.

The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

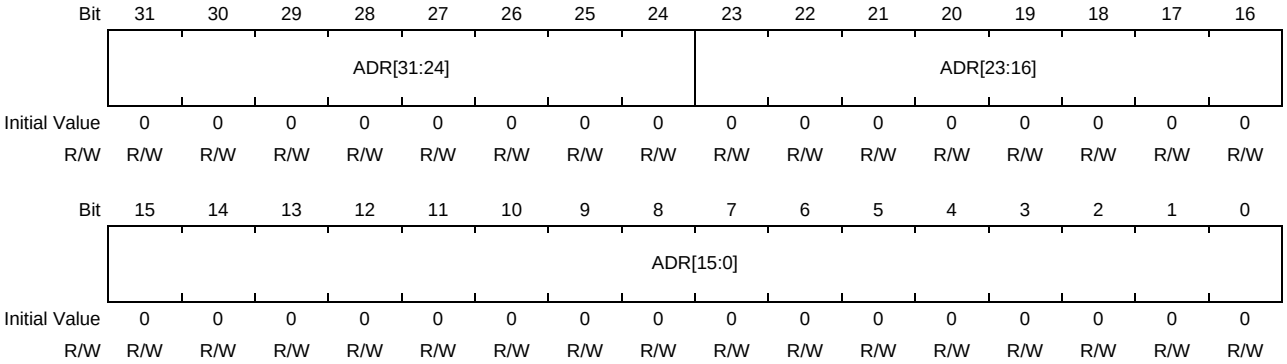



| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 24 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 23 to 16 | CMD[7:0] | H'00 | R/W | Command<br>Sets the command. |
| 15 to 8 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 7 to 0 | OCMD[7:0] | H'00 | R/W | Optional Command<br>Sets the optional command. |

## 25.4.10 Manual Mode Address Setting Register (SMADR)

SMADR is a 32-bit register that sets the addresses in Manual mode.

The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

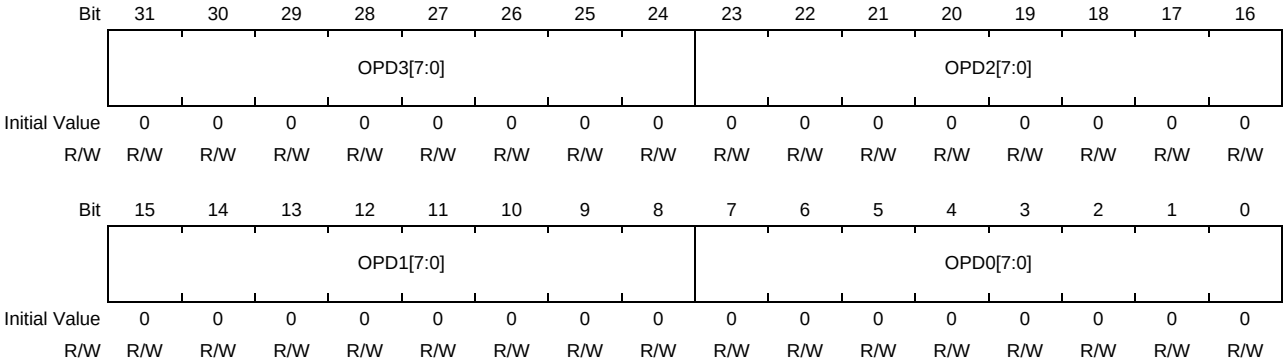



| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 24 | ADR[31:24] | H'00 | R/W | Address<br>Sets the value of bits 31 to 24 when the serial flash address is output in 32-bit units.<br>This setting is valid when ADE[3] in SMENR is 1. |
| 23 to 0 | ADR[23:0] | H'000000 | R/W | Address<br>Sets the address. |

### 25.4.11 Manual Mode Option Setting Register (SMOPR)

SMOPR is a 32-bit register that sets the option data in Manual mode.

The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 24 | OPD3[7:0] | H'00 | R/W | Option Data 3<br>Sets the option data 3. |
| 23 to 16 | OPD2[7:0] | H'00 | R/W | Option Data 2<br>Sets the option data 2. |
| 15 to 8 | OPD1[7:0] | H'00 | R/W | Option Data 1<br>Sets the option data 1. |
| 7 to 0 | OPD0[7:0] | H'00 | R/W | Option Data 0<br>Sets the option data 0. |

**Note:** OPD3, OPD2, OPD1, and OPD0 are output in this order.

## 25.4.12 Manual Mode Enable Setting Register (SMENR)

SMENR is a 32-bit register that sets the bit size of the command, optional command, address, option data, and transfer data in Manual mode and enables their output. SMENR also enables dummy cycle insertion. Disabling all of the command, optional command, address, option data, dummy cycle, and transfer data is prohibited. At least one of them except dummy cycle must be enabled.

The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | CDB[1:0] | | OCDB[1:0] | | — | — | ADB[1:0] | | — | — | OPDB[1:0] | | — | — | SPIDB[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 | R | R/W | R/W | R | R | R/W | R/W | R | R | R/W | R/W |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | DME | CDE | — | OCDE | ADE[3:0] | | | | OPDE[3:0] | | | | SPIDE[3:0] | | | |
| Initial Value | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 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 | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31, 30 | CDB[1:0] | 00 | R/W | Command Bit Size<br>Sets the command size in bit units.<br>00: 1 bit<br>10: 4 bits<br>Other than above: Setting prohibited |
| 29, 28 | OCDB[1:0] | 00 | R/W | Optional Command Bit Size<br>Sets the optional command size in bit units.<br>00: 1 bit<br>10: 4 bits<br>Other than above: Setting prohibited |
| 27, 26 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 25, 24 | ADB[1:0] | 00 | R/W | Address Bit Size<br>Sets the address size in bit units.<br>00: 1 bit<br>10: 4 bits<br>Other than above: Setting prohibited |
| 23, 22 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 21, 20 | OPDB[1:0] | 00 | R/W | Option Data Bit Size<br>Sets the option data size in bit units.<br>00: 1 bit<br>10: 4 bits<br>Other than above: Setting prohibited |
| 19, 18 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 17, 16 | SPIDB[1:0] | 00 | R/W | Transfer Data Bit Size<br>Sets the transfer data size in bit units.<br>00: 1 bit<br>10: 4 bits<br>Other than above: Setting prohibited |

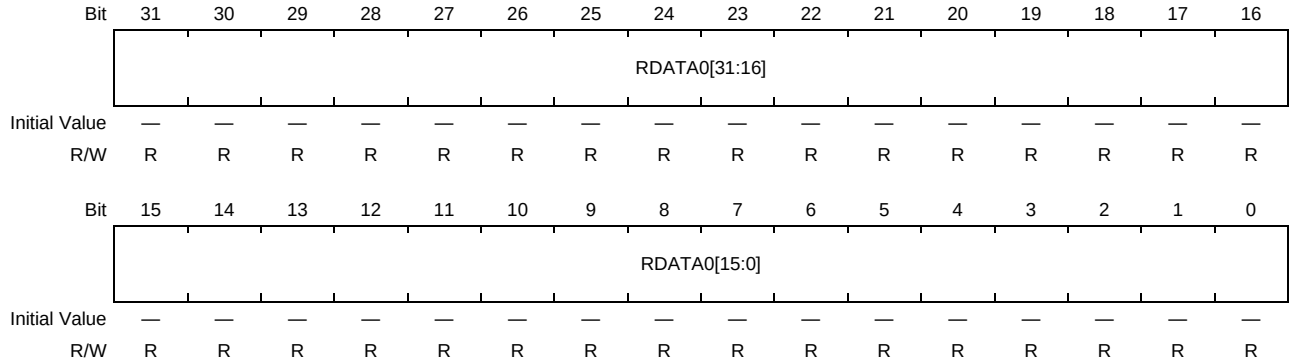
| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 | DME | 0 | R/W | Dummy Cycle Enable<br>Enables insertion of the dummy cycle before the read data.<br>*Note*: Dummy cycle insertion is prohibited for write in Manual mode including the case in which a transfer ends with a dummy cycle.<br>*Note*: A setting is prohibited for a transfer starting with a dummy cycle.<br>0: Dummy cycle insertion disabled<br>1: Dummy cycle insertion enabled |
| 14 | CDE | 1 | R/W | Command Enable<br>Sets the command to be output.<br>0: Command output disabled<br>1: Command output enabled |
| 13 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 12 | OCDE | 0 | R/W | Optional Command Enable<br>Sets the optional command to be output.<br>0: Optional command output disabled<br>1: Optional command output enabled |
| 11 to 8 | ADE[3:0] | 0000 | R/W | Address Enable<br>Sets the address to be output.<br>Use only the settings given below. Otherwise, the operation cannot be guaranteed.<br>When a serial flash memory device is connected<br>0000: Output disabled<br>0100: ADR[23:16]<br>0110: ADR[23:8]<br>0111: ADR[23:0]<br>1111: ADR[31:0]<br>Other than above: Setting prohibited<br>*Note*: The setting of these bits is required to be 0111 or 1111 when the write buffer is in use. |
| 7 to 4 | OPDE[3:0] | 0000 | R/W | Option Data Enable<br>Sets the option data to be output.<br>Use only the settings given below. Otherwise, the operation cannot be guaranteed.<br>0000: Output disabled<br>1000: OPD3<br>1100: OPD3, OPD2<br>1110: OPD3, OPD2, OPD1<br>1111: OPD3, OPD2, OPD1, OPD0<br>Other than above: Setting prohibited |
| 3 to 0 | SPIDE[3:0] | 0000 | R/W | Transfer Data Enable<br>Sets valid transfer data.<br>The following settings must be used. Otherwise, the operation is not guaranteed.<br>BSZ[1:0] bits in CMNCR = 00 (one serial flash memory connected)<br>0000: Not transferred<br>1000: 8 bits transferred (enables data at address 0 of the Manual mode read/write data registers 0)<br>1100: 16 bits transferred (enables data at addresses 0 and 1 of the Manual mode read/write data registers 0 )<br>1111: 32 bits transferred (enables data at addresses 0 to 3 of the Manual mode read/write data registers 0)<br>Other than above: Setting prohibited |

### 25.4.13 Manual Mode Read Data Register 0 (SMRDR0)

SMRDR0 is a 32-bit register that stores the read data in Manual mode.

Access to this register should be performed in the same size as the transfer size specified in the SPIDE[3:0] bits in the Manual mode enable setting register (SMENR). Be sure to access from LSB.

The settings of this register should be read when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 0 | RDATA0[31:0] | Undefined | R | Read Data<br>Holds the data read in Manual mode.<br>Data bits differ depending on the BSZ[1:0] bit setting in CMNCR. |

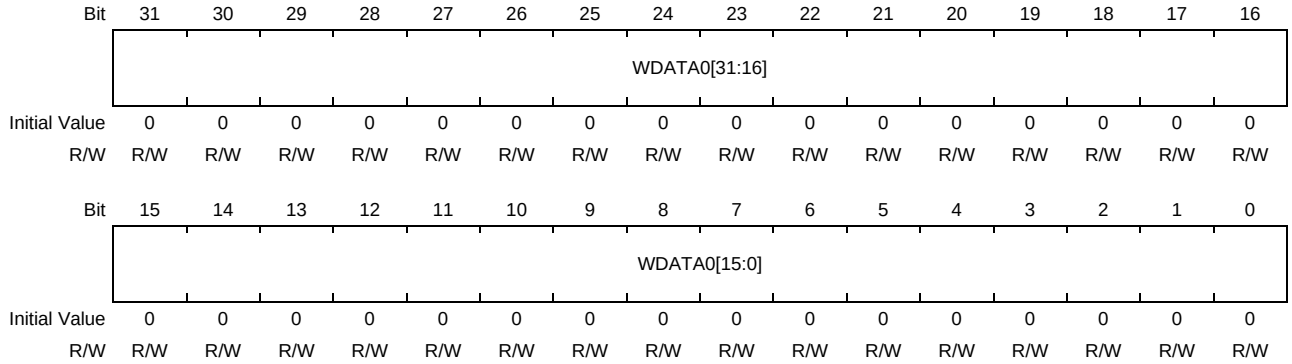
**Note:** The contents of this register and SMRDR1 are modified upon completion of reception in manual mode. Be sure to read data when reception in manual mode is completed.

### 25.4.14 Manual Mode Write Data Register 0 (SMWDR0)

SMWDR0 is a 32-bit register that sets the write data in Manual mode.

Access to this register should be performed in the same size as the transfer size specified in the SPIDE[3:0] bits in the Manual mode enable setting register (SMENR). Be sure to access from address 0.

The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 0 | WDATA0[31:0] | All 0 | R/W | Write Data<br>Holds the data to be written in Manual mode.<br>Data bits differ depending on the BSZ[1:0] bit setting in CMNCR. |

### 25.4.15 Common Status Register (CMNSR)

CMNSR is a 32-bit register that holds flags indicating the operating state.

The settings of this register are reflected both in external address space read mode and Manual 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | SSLF | TEND |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 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 2 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 1 | SSLF | 0 | R | QSPIn_SSL Pin Monitor<br>0: QSPIn_SSL pin is negated<br>1: QSPIn_SSL pin is asserted |
| 0 | TEND | 1 | R | Transfer End Flag<br>Indicates whether the data transfer has ended.<br>0: Indicates that data transfer is in progress<br>1: Indicates that data transfer has ended |

### 25.4.16 Data Read Dummy Cycle Setting Register (DRDMCR)

DRDMCR is a 32-bit register that sets the number of dummy cycles to be inserted in external address space read mode. The settings of this register are enabled when the DME bit in the data read enable setting register (DRENR) is 1.

The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | DMCYC[4:0] | | | | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 |
| 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 |
|---|---|---|---|---|
| 31 to 5 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 4 to 0 | DMCYC[4:0] | H'0B | R/W | Number of Dummy Cycles<br>Sets the number of dummy cycles to be inserted when the DME bit in the data read enable setting register (DRENR) is 1.<br>00001: 2 cycles<br>00010: 3 cycles<br>…<br>10010: 19 cycles<br>10011: 20 cycles<br>Other than above: Setting prohibited |

## 25.4.17 Data Read DDR Enable Register (DRDRENR)

DRDRENR is a 32-bit register that sets SDR or DDR transfer of the address, option data, and read data in external address space read mode.

The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | HYPE | | | — | — | — | ADDRE | — | — | — | OPDRE | — | — | — | DRDRE |
| Initial Value | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| R/W | R | R/W | R/W | R/W | R | R | R | R/W | R | R | R | R/W | R | R | R | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 15 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 14 to 12 | HYPE | 101 | R/W | Indicates the type of flash memory<br>000: SPI flash<br>Other than above: Setting prohibited<br>*Note*: Be sure to change the HYPE bits from the initial value to “000” |
| 11 to 9 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 8 | ADDRE | 1 | R/W | Address DDR Enable<br>Sets SDR or DDR transfer of the address.<br>0: SDR transfer<br>1: DDR transfer |
| 7 to 5 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 4 | OPDRE | 0 | R/W | Option Data DDR Enable<br>Sets SDR or DDR transfer of the option data.<br>0: SDR transfer<br>1: DDR transfer |
| 3 to 1 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 0 | DRDRE | 1 | R/W | Data Read DDR Enable<br>Sets SDR or DDR transfer of the read data.<br>0: SDR transfer<br>1: DDR transfer |

## 25.4.18 Manual Mode Dummy Cycle Setting Register (SMDMCR)

SMDMCR is a 32-bit register that sets the number of dummy cycles to be inserted in Manual mode.

The settings of this register are enabled when the DME bit in the Manual mode enable setting register (SMENR) is 1. The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

| 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/W | R/W |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | DMCYC[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/W | R/W | R/W | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 5 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 4 to 0 | DMCYC[4:0] | 00000 | R/W | Number of Dummy Cycles<br>Sets the number of dummy cycles to be inserted when the DME bit in the Manual mode enable setting register (SMENR) is 1.<br>00001: 2 cycles<br>00010: 3 cycles<br>....<br>10010: 19 cycles<br>10011: 20 cycles<br>Other than above: Setting prohibited |

### 25.4.19 Manual Mode DDR Enable Register (SMDRENR)

SMDRENR is a 32-bit register that sets SDR or DDR transfer of the address, option data, and transfer data in Manual mode.

The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | ADDRE | — | — | — | OPDRE | — | — | — | SPIDRE |
| 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/W | R | R | R | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 15 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 14 to 12 | — | 000 | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value.<br>Operation is not guaranteed if a value other than the initial value is written. |
| 11 to 9 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 8 | ADDRE | 0 | R/W | Address DDR Enable<br>Sets SDR or DDR transfer of the address.<br>0: SDR transfer<br>1: DDR transfer |
| 7 to 5 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 4 | OPDRE | 0 | R/W | Option Data DDR Enable<br>Sets SDR or DDR transfer of the option data.<br>0: SDR transfer<br>1: DDR transfer |
| 3 to 1 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 0 | SPIDRE | 0 | R/W | Transfer Data DDR Enable<br>Sets SDR or DDR transfer of the transfer data.<br>0: SDR transfer<br>1: DDR transfer |

## 25.4.20 PHY Control Register (PHYCNT)

PHYCNT is a 32-bit register that sets the PHY operation mode.

The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | CAL | — | — | — | — | — | — | — | — | — | — | — | — | HS | CKSEL[1:0] | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | W | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | WBUF2 | — | WBUF | PHYMEM[1:0] | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R/W | R | R/W | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 | CAL | 0 | W | PHY Calibration<br>Specifies whether to perform PHY calibration.<br>Be sure to perform calibration before access in Manual mode.<br>0: Calibration is not performed.<br>1: Calibration is performed. |
| 30 | — | 0 | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value.<br>Operation is not guaranteed if a value other than the initial value is written. |
| 29 to 24 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 23 to 20 | — | All 0 | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value.<br>Operation is not guaranteed if a value other than the initial value is written. |
| 19 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 18 | HS | 0 | R/W | High-Speed Response Mode<br>Specifies the high-speed response mode.<br>0: Data is read for the number of data units specified in the RBURST bits of the DRCR register and then output to the bus master.<br>1: The read data is output to bus master in parallel of device access when DRCR.RBE = 1.<br>*Note 1.* When this bit is set to 1, use DMA transfer. The transfer size in the RBURST[4:0] bits of the DRCR register should be fixed to H'1F.<br>*Note 2.* Do not access the register area during DMA transfer. |
| 17, 16 | CKSEL[1:0] | 00 | R/W | Clock Timing Switching<br>Adjusts the timing of the clock when serial flash is connected.<br>Always follow the procedure for setting shown in **Section 25.5.14, Timing Adjustment**. |
| 15 to 10 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 9 | — | 1 | R | Reserved<br>This bit is always read as 1. The write value should always be 1. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 8, 7 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 6, 5 | — | All 1 | R | Reserved<br>These bits are always read as 1. The write value should always be 1. |
| 4 | WBUF2 | 0 | R/W | Write Buffer Enable 2<br>Specifies whether to use the write buffer to write data to the flash memory.<br>For the usage of the write buffer, refer to **Section 25.5.12, Write Buffer Operation**.<br>0: The write buffer is not used.<br>1: The write buffer is used to write data to the flash memory. |
| 3 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 2 | WBUF | 0 | R/W | Write Buffer Enable<br>Specifies whether to use the write buffer to write data to the flash memory.<br>For the usage of the write buffer, refer to **Section 25.5.12, Write Buffer Operation**.<br>0: The write buffer is not used.<br>1: The write buffer is used to write data to the flash memory. |
| 1, 0 | PHYMEM[1:0] | 11 | R/W | Device Selection<br>Selects the device to be connected.<br>00: Serial flash memory in SDR mode<br>01: Serial flash memory in DDR mode<br>Other than above: Setting prohibited |

## 25.4.21 PHY Offset Register 1 (PHYOFFSET1)

PHYOFFSET1 is a 32-bit register that sets the timing adjustment in the DDR operation.

The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | DDRTMG | | — | — | — | — | — | — | — | — | — | — | — | — |
| Initial Value | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Initial Value | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 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, 30 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 29, 28 | DDRTMG | 10 | R/W | DDR Operation Timing<br>Adjusts the timing of reading in the DDR mode.<br>10: Specify this value for reading when SMDRENR.SPIDRE = 1 or DRDRENR.DRDRE = 1.<br>11: Specify this value for reading when SMDRENR.SPIDRE = 0 or DRDRENR.DRDRE = 0.<br>Other than above: Setting prohibited |
| 27 to 25 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 24 | — | 1 | R | Reserved<br>This bit is always read as 1. The write value should always be 1. |
| 23 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 22 | — | 1 | R | Reserved<br>This bit is always read as 1. The write value should always be 1. |
| 21 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 20 | — | 1 | R | Reserved<br>This bit is always read as 1. The write value should always be 1. |
| 19 to 17 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 16 | — | 1 | R | Reserved<br>This bit is always read as 1. The write value should always be 1. |
| 15 to 13 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 12 | — | 1 | R | Reserved<br>This bit is always read as 1. The write value should always be 1. |
| 11 to 9 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 8 | — | 1 | R | Reserved<br>This bit is always read as 1. The write value should always be 1. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. |
| 6 | — | 1 | R | Reserved<br>This bit is always read as 1. The write value should always be 1. |
| 5 to 3 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 2 | — | 1 | R | Reserved<br>This bit is always read as 1. The write value should always be 1. |
| 1, 0 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |

## 25.4.22 PHY Offset Register 2 (PHYOFFSET2)

PHYOFFSET2 is a 32-bit register that sets the timing adjustment in the DDR operation.

The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | OCTTMG | | | — | — | — | — | — | — | — | — |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 |
| R/W | R | R | R | R | R | R/W | R/W | R/W | R | R | R | R | R | R | R | R |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 11 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 10 to 8 | OCTTMG | 100 | R/W | Serial flash memory Operation Timing<br>Adjusts the timing of writing to the serial flash memory<br>000: Specify this value when using the serial flash memory with write buffer operation of the bit width 1-1-4 or 1-4-4 in **Table 25.10**.<br>100: Specify this value when the serial flash memory is used.<br>Other than above: Setting prohibited |
| 7, 6 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 5, 4 | — | All 1 | R | Reserved<br>These bits are always read as 1. The write value should always be 1. |
| 3 to 1 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. |
| 0 | — | 1 | R | Reserved<br>This bit is always read as 1. The write value should always be 1. |

## 25.4.23 PHY Adjustment Register 1 (PHYADJ1)

PHYADJ1 is a 32-bit register used to adjust the timing when serial flash memory is connected.

The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ADJ1[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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ADJ1[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 | ADJ1[31:0] | All 0 | R/W | Serial Flash Memory Operation Timing Adjustment<br>Before timing adjustment: Set these bits to H'80000000.<br>During timing adjustment: Set these bits to H'80000032.<br>*Note*: Do not set any value other than the specified ones. |

## 25.4.24 PHY Adjustment Register 2 (PHYADJ2)

PHYADJ2 is a 32-bit register used to adjust the timing when serial flash memory is connected.

The settings of this register should be changed when the TEND flag in CMNSR is 1; otherwise, the operation cannot be guaranteed.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ADJ2[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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ADJ2[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 |

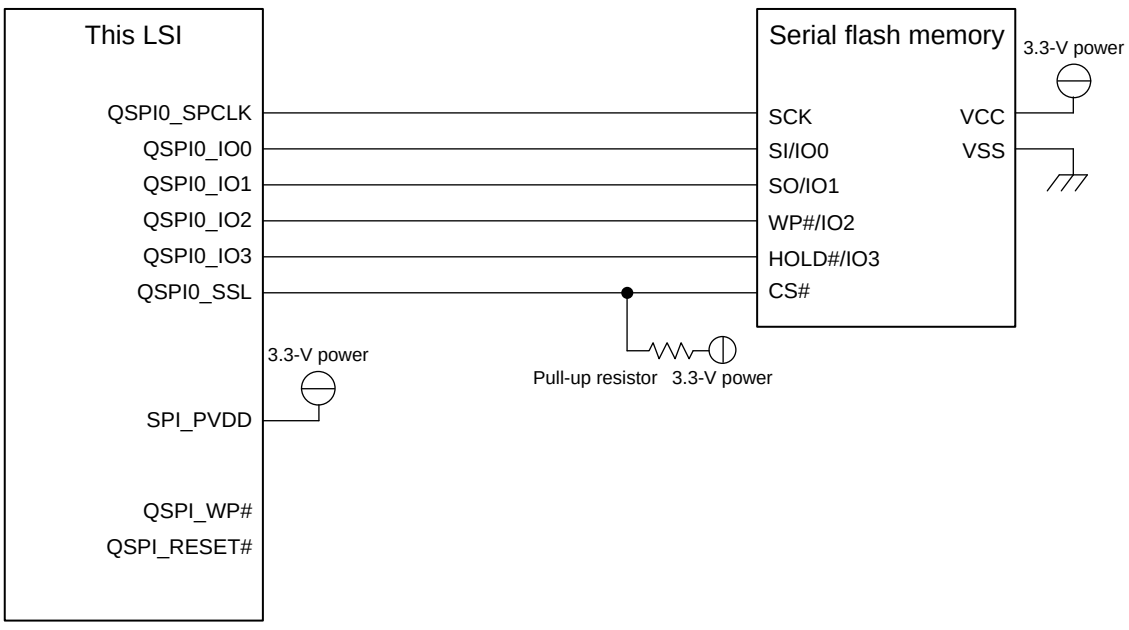
| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 0 | ADJ2[31:0] | All 0 | R/W | Serial Flash Memory Operation Timing Adjustment<br>Before timing adjustment: Set these bits to H'A5390000.<br>During timing adjustment: Set these bits by following the procedure shown in **Figure 25.19, Flow of Timing Adjustment when the Serial Flash Memory is Connected (SDR Mode)** and **Figure 25.20, Flow of Timing Adjustment when the Serial Flash Memory is Connected (DDR Mode)**.<br>*Note*: Do not set any value other than the specified ones. |

## 25.5 Operation

### 25.5.1 System Configuration

This module can connect one serial flash memory device.

Examples of system configuration are shown in **Figure 25.2**.




**Note:** When connecting this LSI to a single serial flash memory device, use the pins for channel 0. The pins for channel 1 cannot be used in this configuration.

Figure 25.2 System Configuration Example with 4-Bit Data Size and One Serial Flash Memory Connected

### 25.5.2 Address Map

In external address space read mode, the serial flash connected is assigned in the SPI multi I/O bus space. In combination with DREAR, a maximum of 4 Gbytes can be accessed when one serial flash memory is connected.

Table 25.3 Address Map

| Internal Address for Cortex-A55 | Max. Access Area |
|---|---|
| H'0_20000000 to H'0_2FFFFFFF | One serial flash memory connected: 4 G-byte |

**Note:** When access protection is set, not only this address space, but also the register area is protected.

### 25.5.3 32-bit Serial Flash Addresses

Since the SPI multi I/O bus space is 256 Mbytes, only a part of the 32-bit serial flash address area can be directly accessed. Here, the fixed value set in the pertinent register is used as the upper bit value of a 32-bit address.

To output serial flash addresses in 32 bits, set the ADE[3] bit in DRENR to 1, set the range of the external addresses used as the serial flash addresses to the EAC[2:0] bits in DREAR, and set the upper bit value of the 32-bit address as the fixed value to the EAV[7:0] bits in DREAR.

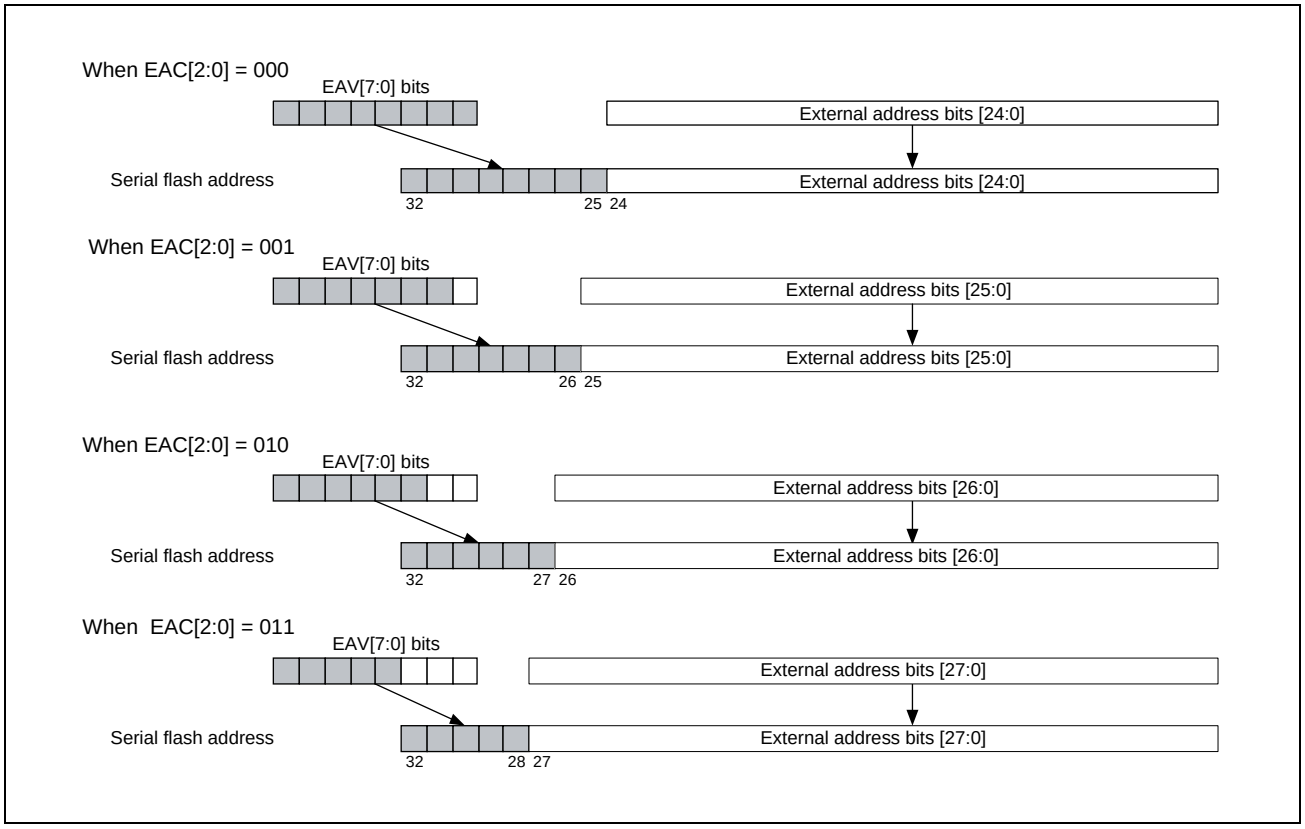



Figure 25.3 32-Bit Address Setting

Setting the ADE[3] bit in DRENR to 1 allows the serial flash address to be output using [31:0] bits.

When EAC[2:0] = 000, external address bits [24:0] are valid; set the value for [32:25] bits to EAV[7:0].
When EAC[2:0] = 001, external address bits [25:0] are valid; set the value for [32:26] bits to EAV[7:1].
When EAC[2:0] = 010, external address bits [26:0] are valid; set the value for [32:27] bits to EAV[7:2].
When EAC[2:0] = 011, external address bits [27:0] are valid; set the value for [32:28] bits to EAV[7:3].

The addresses actually used for external devices are as follows.

- Use the address [31: 0] when connecting one serial flash.

## 25.5.4 Operating Modes

This module has two operating modes: external address space read mode and Manual mode.

In external address space read mode, a read access to the SPI multi I/O bus space is converted into Data read protocol and data is received. After data acquisition, data is returned to the bus master that is the issuing source. For details, see **Section 25.5.5, External Address Space Read Mode**.

In Manual mode, arbitrary protocol is carried out using register settings. For details, see **Section 25.5.7, Manual mode**.

## 25.5.5 External Address Space Read Mode

A read access to the SPI multi I/O bus space can be converted into Data read protocol in external address space read mode. Further, the commands, optional commands, option data, and dummy cycle issued for reading can be modified using registers.

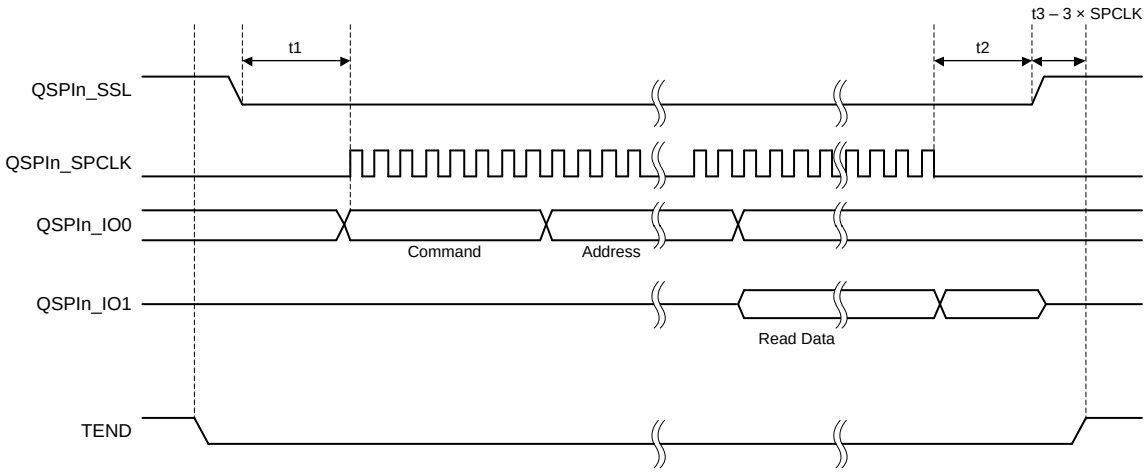
In external address space read mode, either normal read operation or burst read operation can be selected. The transfer format is determined based on the common control register (CMNCR), SSL delay register (SSLDR), data read control register (DRCR), data read command setting register (DRCMR), data read extended address setting register (DREAR), data read option setting register (DROPR), data read enable setting register (DRENR), data read dummy cycle setting register (DRDMCR), and data read DDR enable register (DRDRENR).

### (1) Normal Read Operation

When the RBE bit in the DRCR register is set to 0, normal read operation is performed. In the normal read operation, data is read from the external device when the bus master executes read access. After data is read, the QSPIn_SSL pins are negated.

The timing of normal read operation for the serial flash memory is shown in **Figure 25.4**.

t1 is the time period from QSPIn_SSL pin assertion to QSPIn_SPCLK oscillation (clock delay), t2 is the time period from transmission of the last QSPIn_CLK edge of a transfer to QSPIn_SSL pin negation (QSPIn_SSL negation delay), and t3 is the time period from one transfer end to the next transfer start (next access).




**Note:** t1, t2, and t3 are specified with SSLDR.SCKDL, SSLDR.SLNDL and SSLDR.SPNDL bits, respectively.

Figure 25.4 Normal Read Operation Timing

### (2) Burst Read Operation

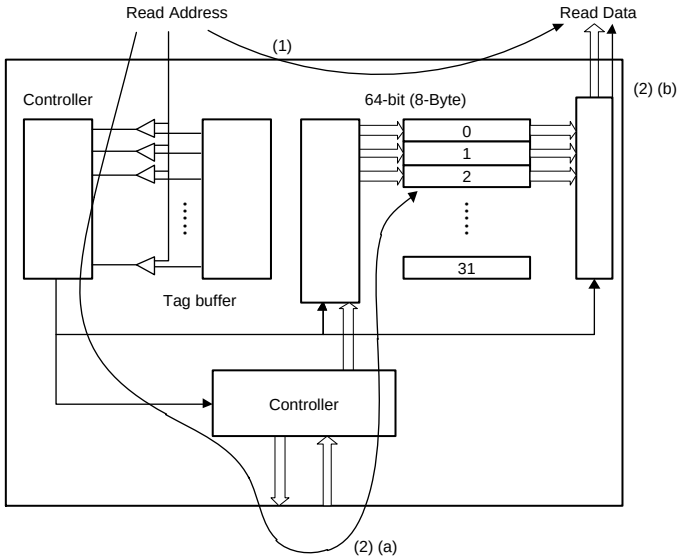
When the RBE bit in DRCR is set to 1, burst read operation is performed.

Read cache is enabled in the burst read operation. For read cache operation, see **Section 25.5.6, Read Cache**.

When the bus master executes read access, the read cache is first searched for the target data. When the read cache contains the data, it is read from the cache instead of accessing the external device. When the read cache does not contain the data, the external device is read in the burst mode and the read data is stored in the read cache and returned to the bus master. In this operation, the data transfer length is 64 bits × the value in the RBURST[4:0] bits, and data is always read from a 64-bit boundary address.

The QSPIn_SSL pin status after data transfer can be selected by using the SSLE bit in DRCR. When the SSLE bit is set to 0, the QSPIn_SSL pin is negated after data transfer. For an operation performed when the SSLE bit is set to 1, see **Section 25.5.5(3), Burst Read Operation with Automatic QSPIn_SSL Negation**, just below.

A pattern diagram of this operation and a burst read operation timing diagram when SSLE bit is set to 0 are shown in **Figure 25.5** and **Figure 25.6**.




(1) When the read cache contains the data
The data is read from the read cache without accessing external devices.

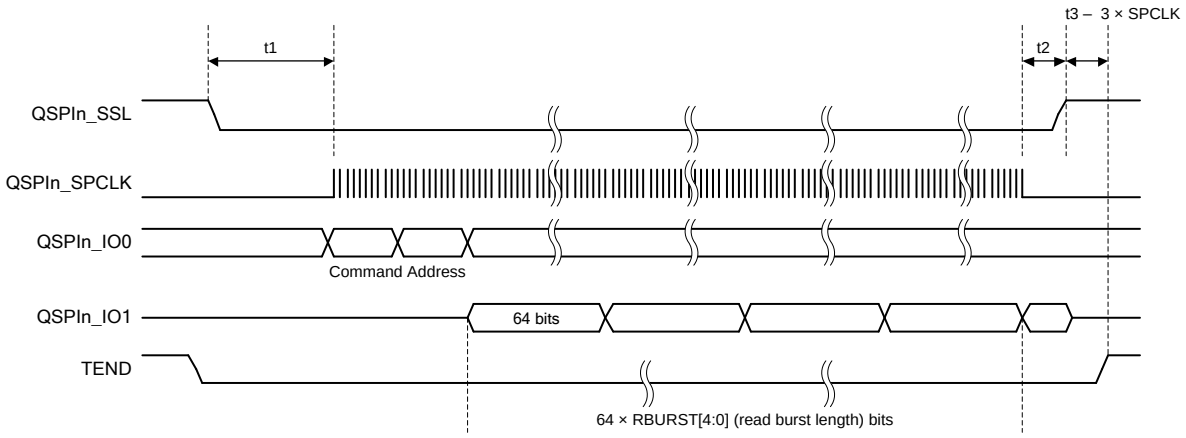
(2) When the read cache does not contain the data
(a) The data is read from the external device and stored in the read cache.
(b) The data is read from the read cache and returned to the bus master.

Figure 25.5 Burst Read Operation

NOTE

When connecting this LSI to a single serial flash memory device, use the pins for channel 0.

The pins for channel 1 cannot be used in this configuration.




**Note:** t1, t2, and t3 are specified with SSLDR.SCKDL, SSLDR.SLNDL and SSLDR.SPNDL bits, respectively.

Figure 25.6 Burst Read Operation Timing

### (3) Burst Read Operation with Automatic QSPIn_SSL Negation

When SSLE bit in DRCR is set to 1, this module does not negate the QSPIn_SSL pin after the burst read transfer. When accessing the next time, if the address is continuous with the previous read address, the burst read operation is performed without issuing the command, optional command, address, option data, or dummy cycle. If the address is not continuous with the previous read address, the QSPIn_SSL pin is once negated and the burst read operation is performed after issuing the command, optional command, address, option data, or dummy cycle.

Burst read timing diagrams for continuous address and non-continuous address are shown in **Figure 25.7** and **Figure 25.8**.

For the next access after negation of the QSPIn_SSL with the SSLN bit in DRCR with this operation, read SSLF = 0 in CMNSR to confirm that the QSPIn_SSL has been negated.

This function is available when serial flash memory is connected.

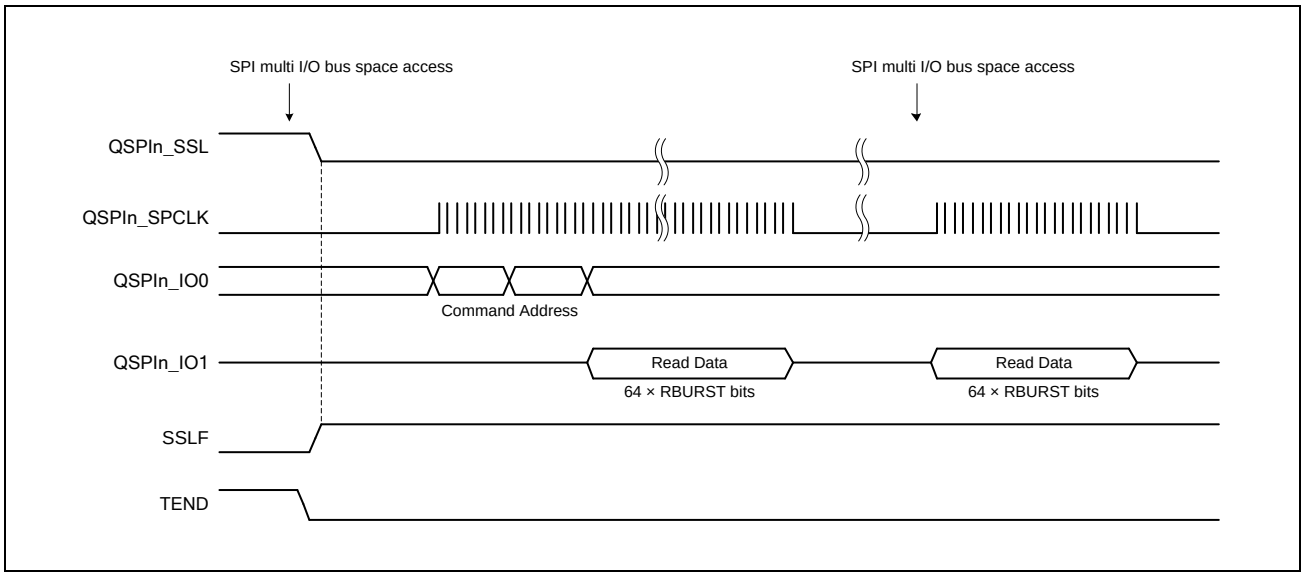



Figure 25.7 Burst Read Timing for Continuous Address (SSLE Bit = 1)

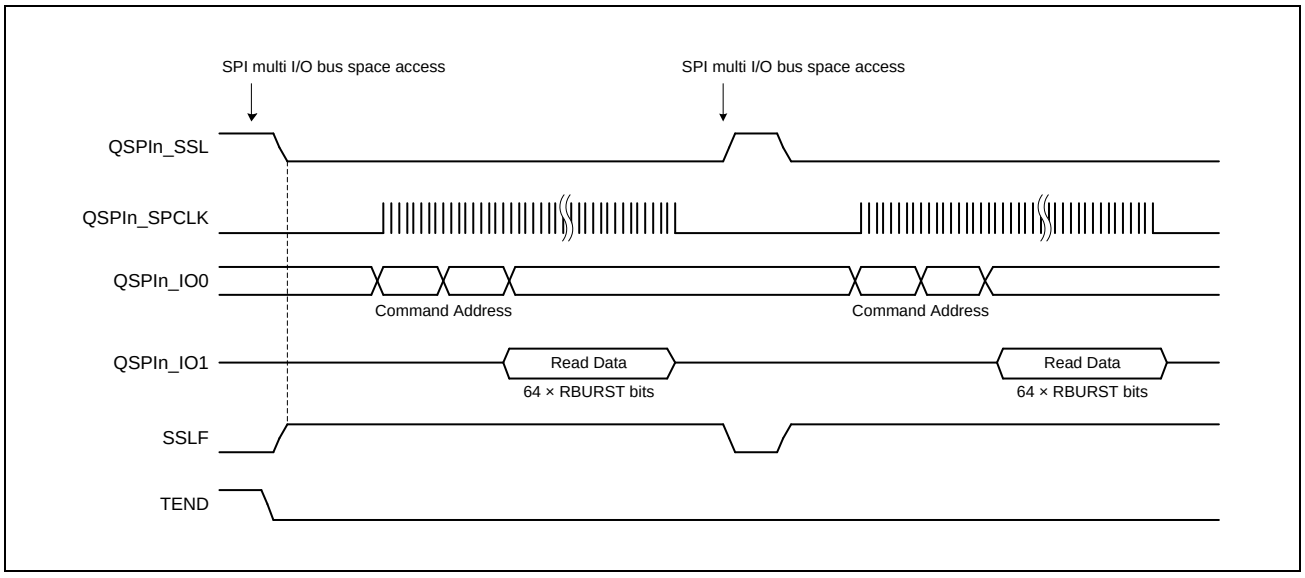



Figure 25.8 Burst Read Timing for Non-Continuous Address (SSLE Bit = 1)

### (4) Initial Setting Flow

An example of an initial setting flow in external address space read mode is shown in **Figure 25.9**.

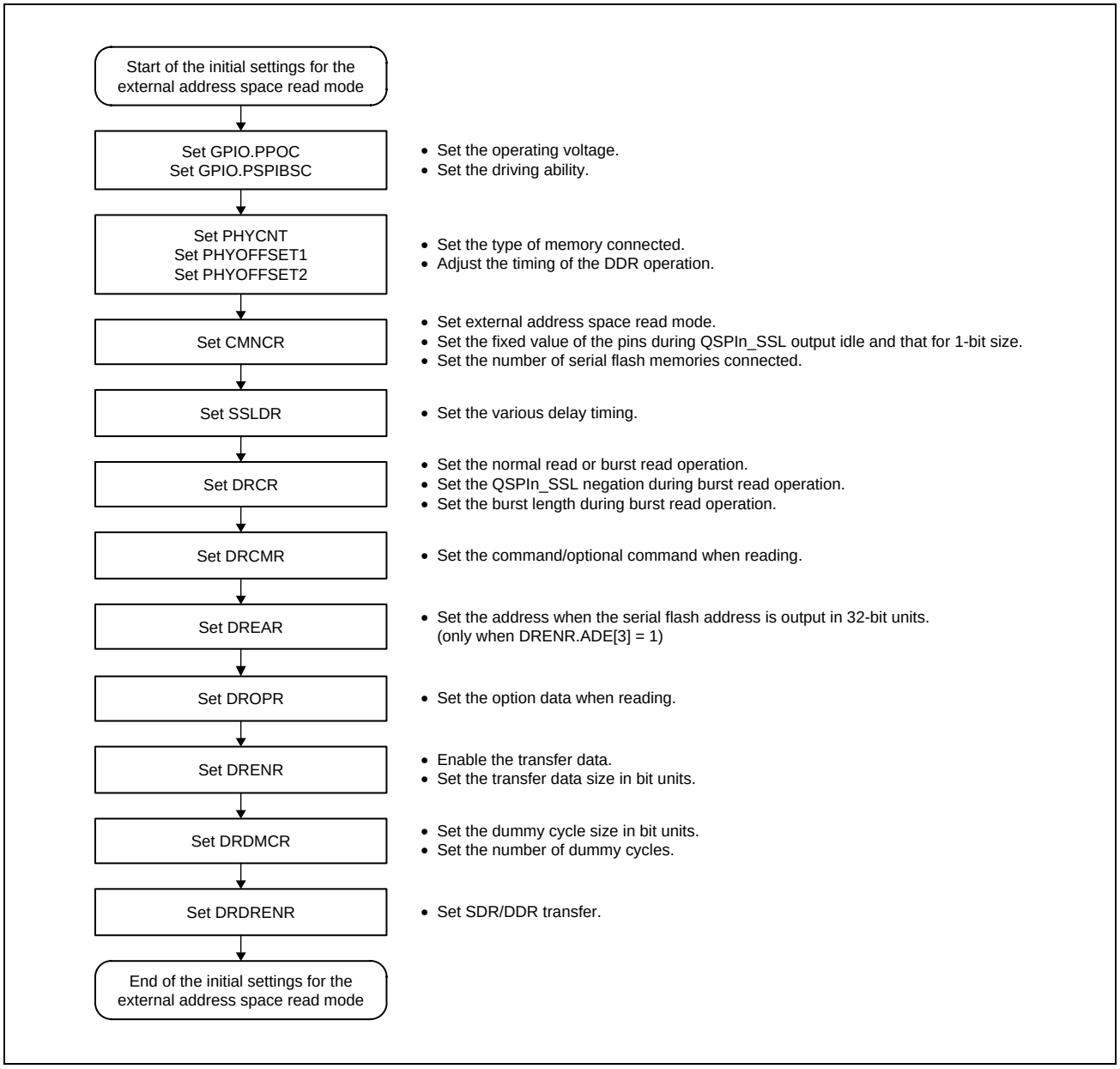



Figure 25.9 Example of Initial Setting Flow in External Address Space Read Mode

### 25.5.6 Read Cache

This module has a simple built-in read cache. The read cache can be used during external address space read mode and burst read operation. The read cache is configured with a line size of 64 bits and 32 entries.

Read cache configuration is shown in **Figure 25.10**.

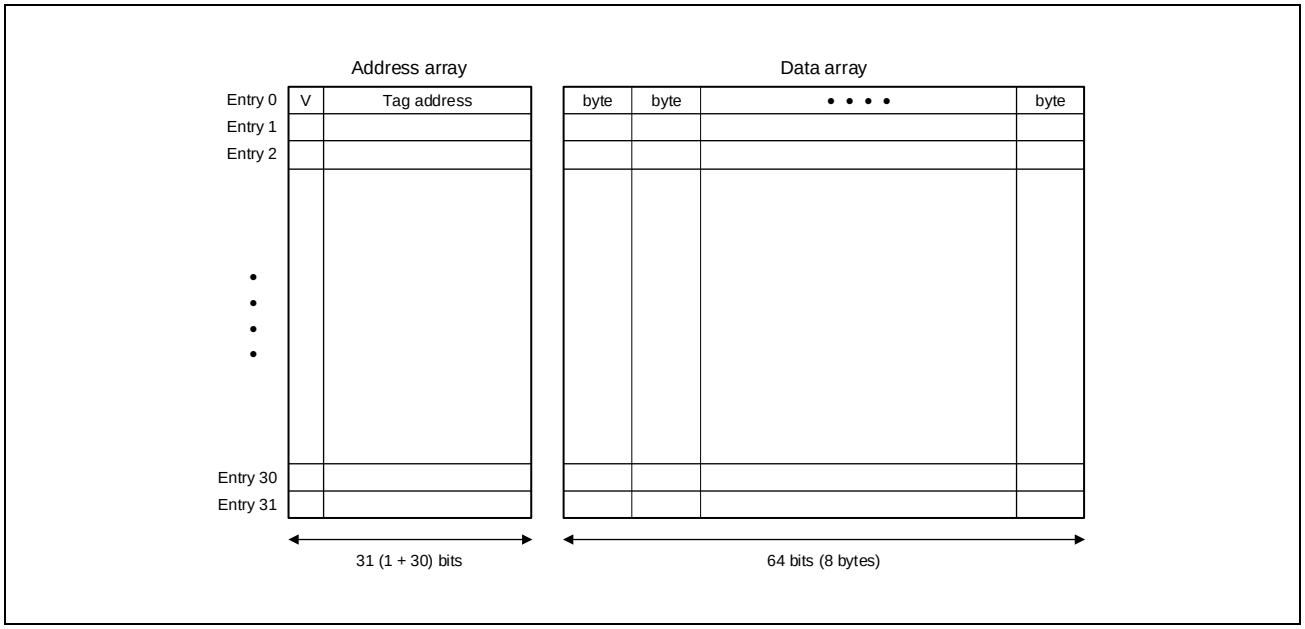



Figure 25.10 Read Cache Configuration

#### (1) Address Array

The V bit indicates whether the entry data is valid. When the V bit is 1, the data is valid and when V bit is 0, the data is invalid.

The tag address bits hold the address used for the serial flash memory. Address bits 32 to 3 are used for the purpose. Address bits 23 to 3 are enabled when address output is 24 bits and one serial flash memory is connected; and address bits 24 to 3 are enabled when two serial flash memories are connected.

Address bits 31 to 3 are enabled when address output is 32 bits and one serial flash memory is connected; and address bits 32 to 3 are enabled when two serial flash memories are connected.

#### (2) Data Array

It retains the 64-bit read data. Registration in the read cache is performed in line units.

#### (3) Read Operation

In case of read-hit, data is read from the read cache. In case of read-miss, after the 64 × RBURST (read burst length) data is read from the serial flash memory and the read cache is updated, the data is returned to the bus master.

#### (4) Data Replacement

The write pointer is used to update data. In case of read-miss, the RBURST (read burst length) portion data is replaced starting at the entry specified by the write pointer. In other words, the data is replaced in the storage order of the data. Whether data is referred to or not will not affect the replacement order of data.

## 25.5.7 Manual mode

This module can carry out an arbitrary serial transmission operation by using the register settings.

The transfer format is determined based on the common control register (CMNCR), SSL delay register (SSLDR), Manual mode control register (SMCR), Manual mode command setting register (SMCMR), Manual mode address setting register (SMADR), Manual mode option setting register (SMOPR), and Manual mode enable setting register (SMENR), Manual mode read data register (SMRDR), Manual mode write data register (SMWDR), Manual mode dummy cycle setting register (SMDMCR), and Manual mode DDR enable register (SMDRENR).

Manual mode can be used for reading the status of the serial flash memory and writing to the serial flash memory.

In this mode, one transfer refers to the operation from when the SPIE bit in SMCR is set to 1 to when the TEND bit is set to 1.

### (1) Transfer Start

The transfer of data is started in the set transfer format by setting the SPIE bit in SMCR to 1. When write operation is enabled, the Manual mode write data register is transmitted to the serial flash memory. When read operation is enabled, data read from the serial flash memory is stored into the Manual mode read data register.

The Manual operation timing is shown in **Figure 25.11**.

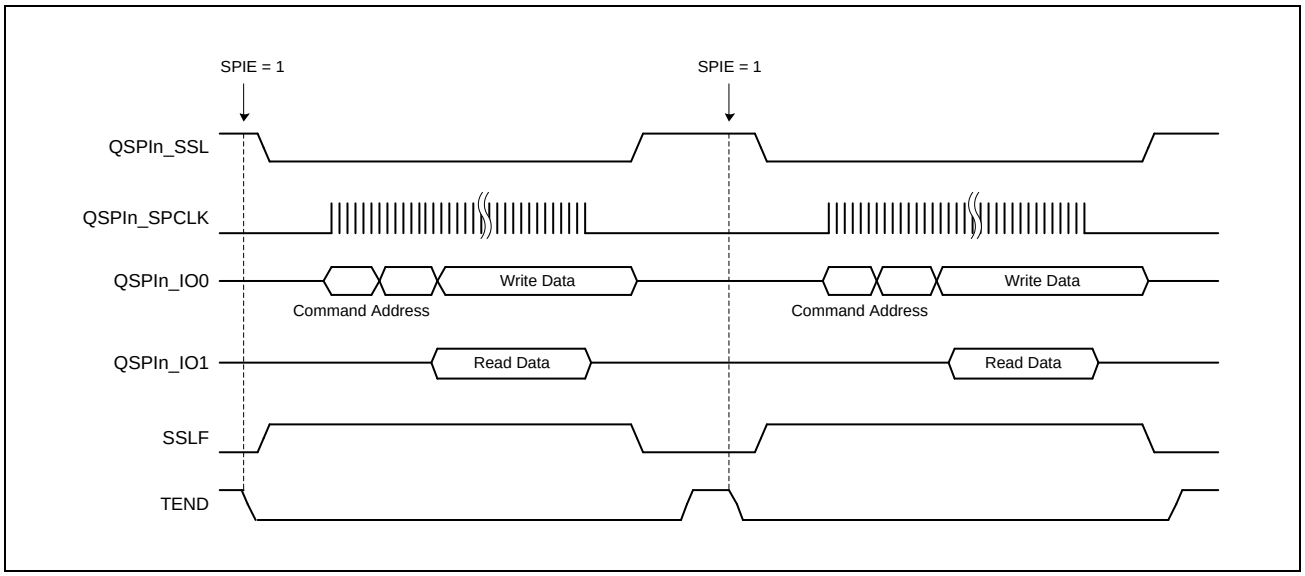



Figure 25.11 Serial transmission operation timing

### (2) Read/Write Enable

- Read operation: Data can be read by setting the SPIRE bit in SMCR to 1. The read data is stored into SMRDR.
- Write operation: Data can be written by setting the SPIWE bit in SMCR to 1. The data stored in SMWDR is output.

When the data size is set to 1 bit using the SPIDB[1:0] bits in SMENR, data can be transmitted and received by setting the SPIRE and SPIWE bits to 1. However, when the data size is set to 4 bits by using the SPIDB[1:0] bits, only one of the SPIRE and SPIWE bits should be enabled. The operation is not guaranteed if the both bits are enabled.

### (3) Retention of QSPIn_SSL Pin Assertion

By setting the SSLKP bit in SMCR to 1, assertion of the QSPIn_SSL pin can be continued till the next transfer. With this function, the transfer can be carried out continuously with the QSPIn_SSL kept in the asserted state.

The data transfer timing using the SSLKP bit is shown in **Figure 25.12**.

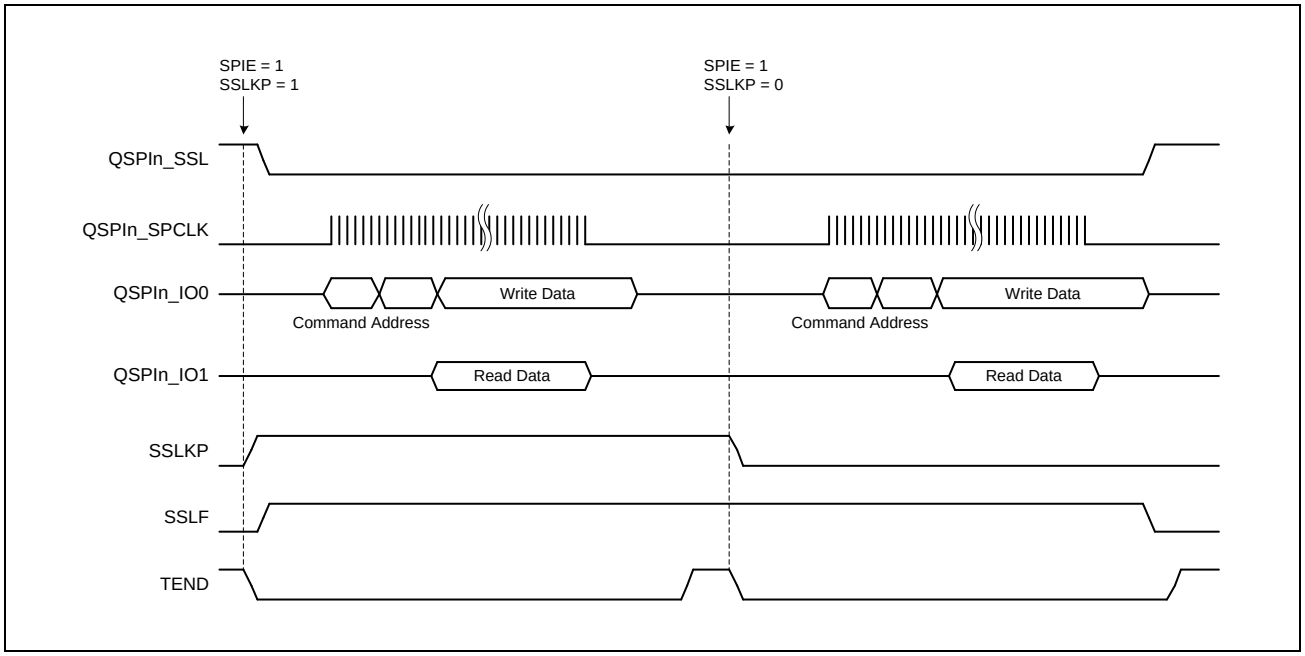



Figure 25.12 Data Transfer Timing using the SSLKP Bit

### (4) Initial Setting Flow

An example of an initial setting flow in Manual mode is shown in **Figure 25.13**.

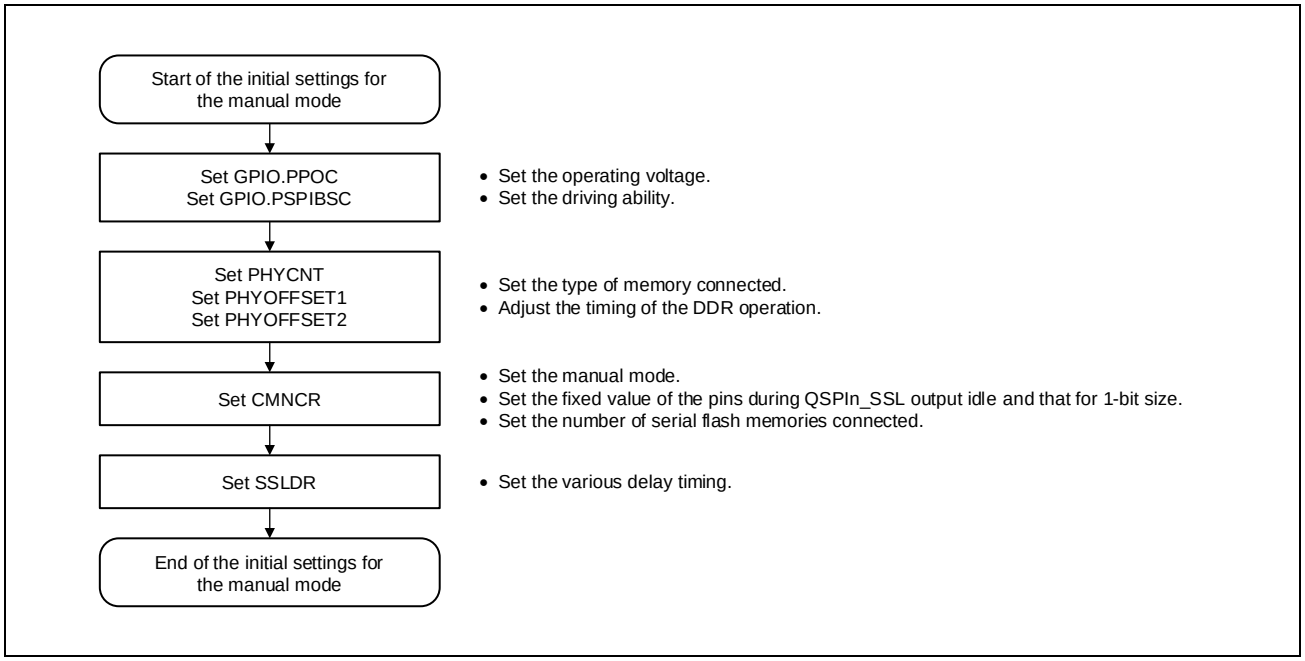



Figure 25.13 Example of Initial Setting Flow in Manual mode

### (5) Data Transfer Setting Flow

An example of a data transfer setting flow in manual mode is shown in **Figure 25.14**.

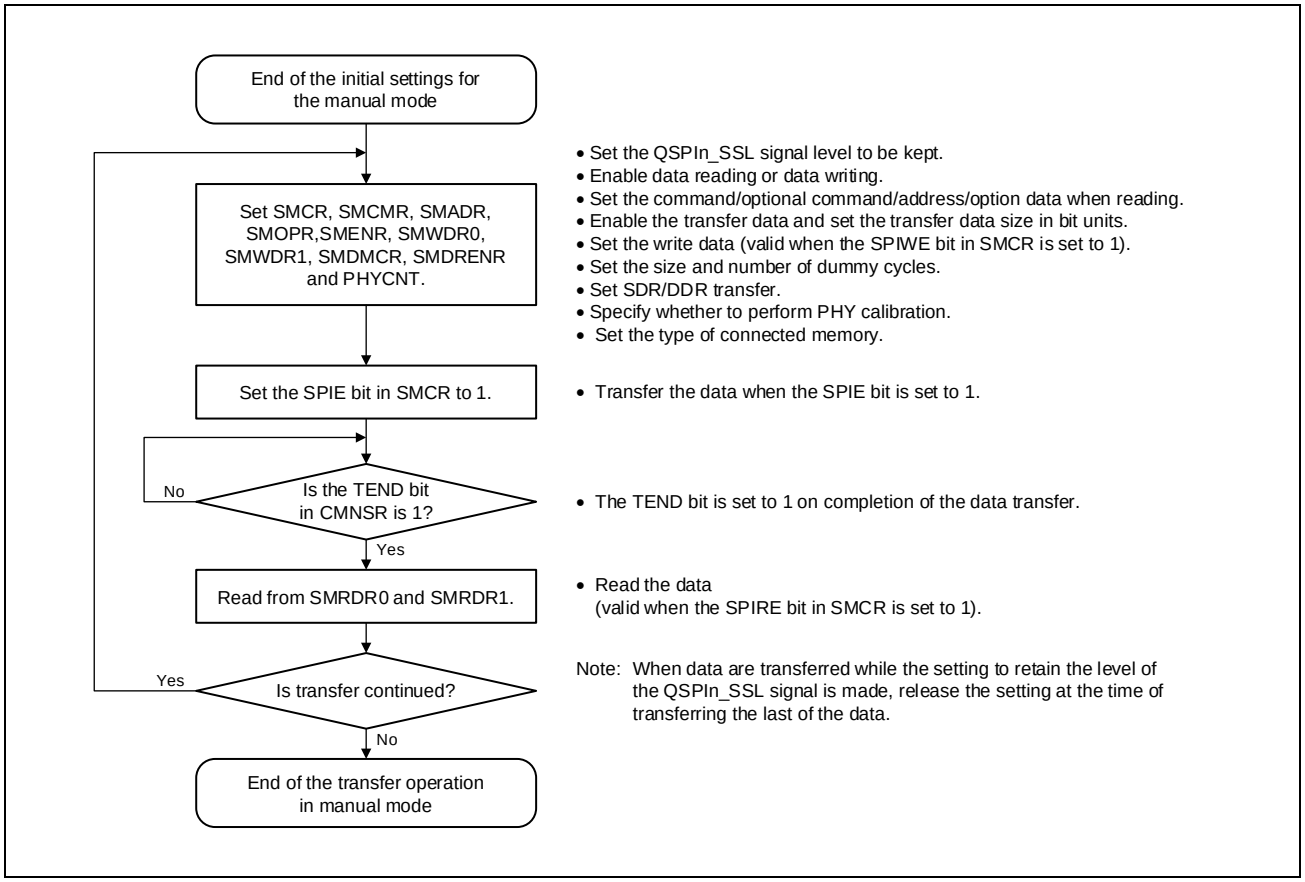



Figure 25.14 Example of a Data Transfer Setting Flow in Manual Mode

### 25.5.8 Command Sequence

This module can input and output data in the order of command, optional command, address, option data, dummy cycle, and data.

#### (1) Data Registers

**Table 25.4** shows the input and output data.

Table 25.4 Data Registers

| Data | | External Address Space Read Mode | Manual Mode |
|---|---|---|---|
| Command (8 bits) | | CMD[7:0] bits in DRCMR | CMD[7:0] bits in SMCMR |
| Optional command (8 bits) | | OCMD[7:0] bits in DRCMR | OCMD[7:0] bits in SMCMR |
| Address (32/24 bits) | BSZ[1:0] = 00 (one flash memory connected) | 32 bits: DREAR.EAV[6:3 to 0] bits + lower [27 to 24:0] bits of the read address.<br>24 bits: Lower [23:0] bits of the read address. | 32 bits: ADR[31:0] bits in SMADR<br>24 bits: ADR[23:0] bits in SMADR |
| Option data (8 bits × 4) | | DROPR | SMOPR |
| Dummy cycle (2 to 20 cycles) | | DRDMCR | SMDMCR (only when read) |
| Transfer data | | Normal read: 8, 16, and 32 bits<br>Burst read: 64 × RBURST bits | Read: SMRDR0, SMRDR1<br>Write: SMWDR0, SMWDR1 |

### (2) Data Enable

In external address space read mode, enabling or disabling transfer of the command, optional command, address, option data, and dummy cycle can be controlled with the CDE, OCDE, ADE[3:0], OPDE[3:0], and DME bits in DRENR, respectively. The number of dummy cycles can be controlled with the data read dummy cycle setting register (DRDMCR).

Similarly, in manual mode, enabling or disabling the command, optional command, address, option data, dummy cycle, and transfer data can be controlled with the CDE, OCDE, ADE[3:0], OPDE[3:0], DME, and SPIDE[3:0] bits in SMENR, respectively. However, disabling all the above parameters is prohibited in manual mode. At least one of them except dummy cycle must be enabled. The number of dummy cycles can be controlled with the Manual mode dummy cycle setting register (SMDMCR).

For the address and option data in external address space read mode and the address, option data, and transfer data in manual mode, the enable bit setting allowed is determined according to the transfer data size. For the allowed setting combinations of the enable bits and transfer data size, refer to the description of the pertinent register.

If data is disabled, that data is skipped, and input and output of the next data is carried out. The command, optional command, address, and option data are always output. During dummy cycles, the state of the used pins is Hi-Z. In external address space read mode, data is always input; and in manual mode, input and output of data is determined based on the settings of the SPIRE and SPIWE bits in SMCR.

There are some restrictions on dummy cycle insertion; refer to the description of the DME bits in DRENR and SMENR for details.

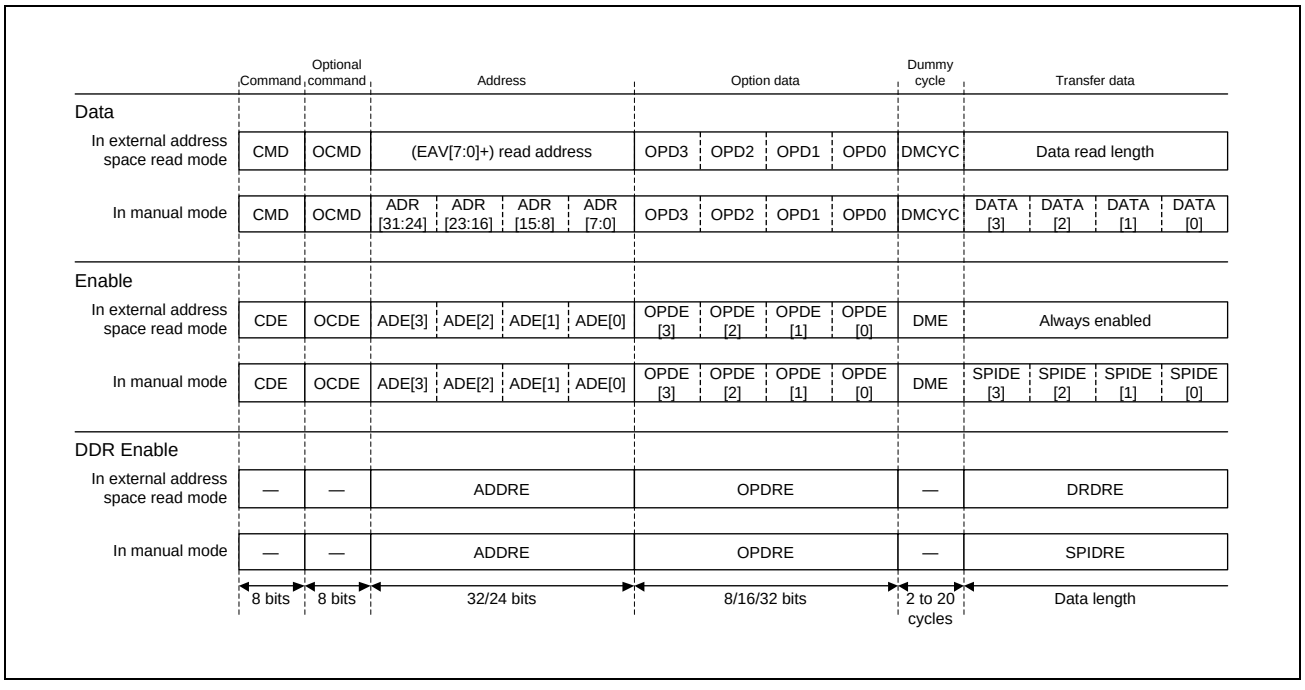



Figure 25.15 Data and Enable (Serial Flash)

### (3) Bit Size

In external address space read mode, the size of the command, optional command, address, option data, and the read data in bit units is respectively controlled with the CDB[1:0], OCDB[1:0], ADB[1:0], OPDB[1:0], and DRDB[1:0] bits in DRENR.

Similarly, in manual mode, the size of the command, optional command, address, option data, and read write data in bit units is controlled with the CDB[1:0], OCDB[1:0], ADB[1:0], OPDB[1:0], and SPIDB[1:0] bits in SMENR.

#### (a) 1-bit Size

When the size is set to 1 bit, QSPI0_IO1 pin will be the input pin and QSPI0_IO0 pin will be the output pin. QSPI0_IO2 and QSPI0_IO3 pins are not used.

**Figure 25.16** show the transfer format examples.

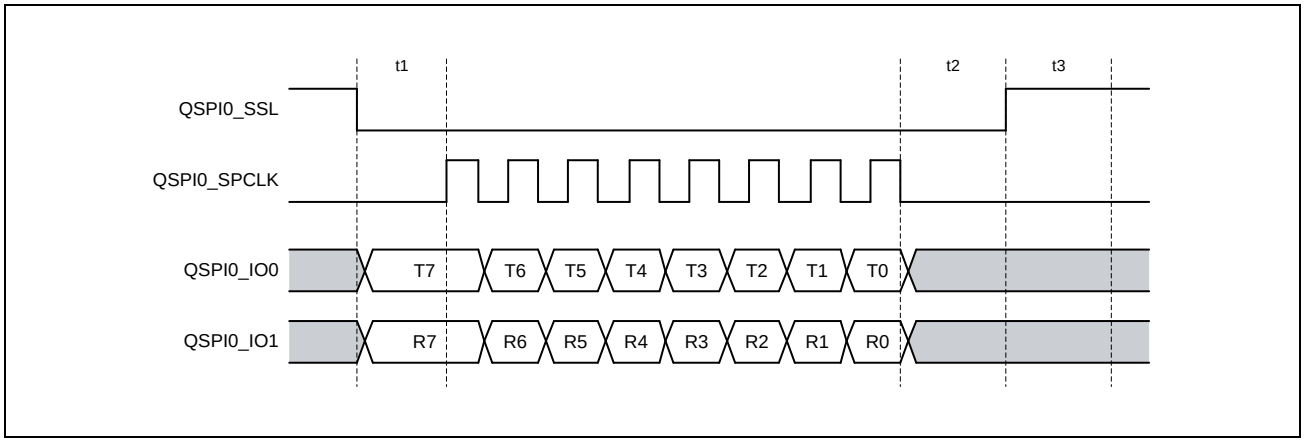



Figure 25.16 Transfer Format Example with 1-Bit Data Size and One Serial Flash Memory Connected

### (b) 4-bit Size (Serial Flash)

When the size is set to 4 bits, QSPI0_IO0, QSPI0_IO1, QSPI0_IO2, and QSPI0_IO3 pins will be either the input pins or the output pins.

**Figure 25.17** show the transfer format examples.

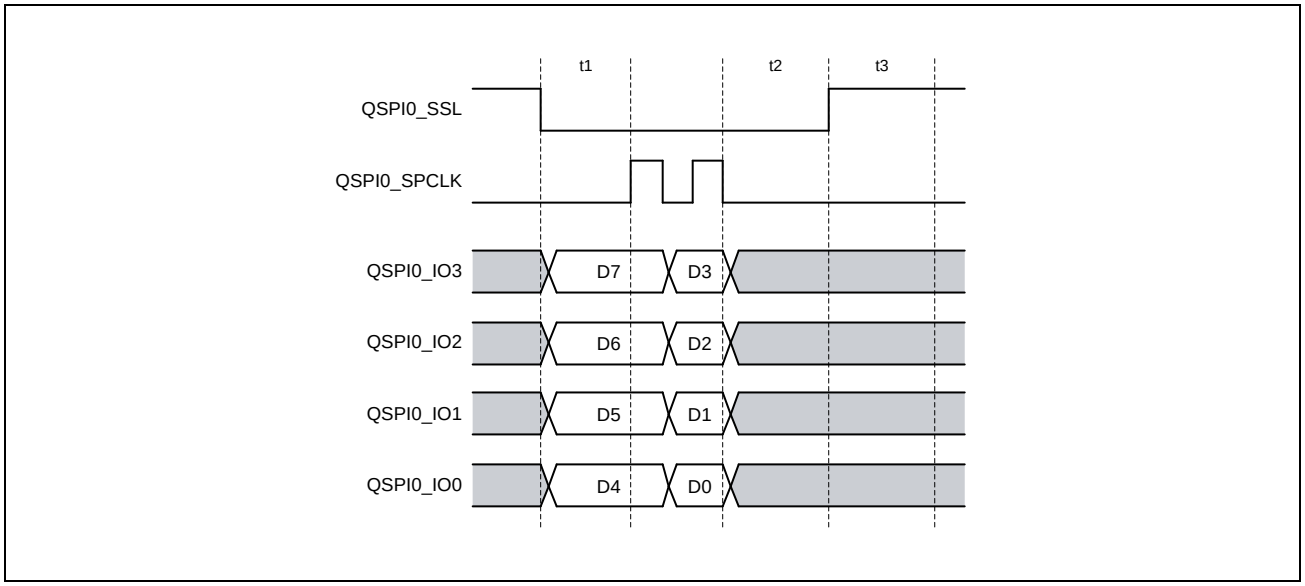



Figure 25.17 Transfer Format Example with 4-Bit Data Size and One Serial Flash Memory Connected

### 25.5.9 Data Pin Control

With this module, the status of pins can be automatically changed based on the data size to be used and the read/write settings. The pin status during the QSPIn_SSL negation can be set with the MOIIO3, MOIIO2, MOIIO1, and MOIIO0 bits in CMNCR. The QSPIn_SSL and QSPIn_SPCLK pins are always output pins.

The status of respective pins is specified in **Table 25.5** to **Table 25.8**.

Table 25.5 Pin Status (1)

| | | QSPIn_SSL Assertion | |
|---|---|---|---|
| | | Command, Optional Command, Address, Option Data | |
| Pin | QSPIn_SSL Negation | 1-bit Size | 4-bit Size |
| QSPI0_IO0 | MOIIO0 bit value | Output | Output |
| QSPI0_IO1 | MOIIO1 bit value | Hi-Z | Output |
| QSPI0_IO2 | MOIIO2 bit value | IO2FV bit value | Output |
| QSPI0_IO3 | MOIIO3 bit value | IO3FV bit value | Output |

Table 25.6 Pin Status (2)

| | Transfer Data | | | |
|---|---|---|---|---|
| | External Address Space Read Mode | | SPI Operation | |
| | | | SPIRE Bit = 1, SPIWE Bit = 0 | |
| Pin | 1-bit Size | 4-bit Size | 1-bit Size | 4-bit Size |
| QSPI0_IO0 | IO0FV bit value | Input | IO0FV bit value | Input |
| QSPI0_IO1 | Input | Input | Input | Input |
| QSPI0_IO2 | Hi-Z | Input | Hi-Z | Input |
| QSPI0_IO3 | Hi-Z | Input | Hi-Z | Input |

Table 25.7 Pin Status (3)

| | Transfer Data | | | |
|---|---|---|---|---|
| | SPI Operation | | | |
| | SPIRE Bit = 0, SPIWE Bit = 1 | | SPIRE Bit = 1, SPIWE Bit = 1 | |
| Pin | 1-bit Size | 4-bit Size | 1-bit Size | 4-bit Size |
| QSPI0_IO0 | Output | Output | Output | Setting prohibited |
| QSPI0_IO1 | Hi-Z | Output | Input | Setting prohibited |
| QSPI0_IO2 | IO2FV bit value | Output | IO2FV bit value | Setting prohibited |
| QSPI0_IO3 | IO3FV bit value | Output | IO3FV bit value | Setting prohibited |

Table 25.8 Pin Status (4)

| Pin | Dummy Cycle |
|---|---|
| QSPI0_IO0 | IO0FV bit value |
| QSPI0_IO1 | Hi-Z |
| QSPI0_IO2 | Hi-Z |
| QSPI0_IO3 | Hi-Z |

### 25.5.10 QSPIn_SSL Pin Control

Negation conditions of the QSPIn_SSL pin are as follows.

#### (1) External Address Space Read Mode

##### (a) Normal read operation (RBE bit in DRCR = 0)

QSPIn_SSL negated after completing the data transfer and t2 cycle.

##### (b) Burst read without automatic QSPIn_SSL negation (RBE bit in DRCR = 1, SSLE bit in DRCR = 0)

QSPIn_SSL negated after completing the data transfer and t2 cycle.

##### (c) Burst read with automatic QSPIn_SSL negation (RBE bit in DRCR = 1, SSLE bit in DRCR = 1)

- QSPIn_SSL negated after t2 cycle when the read address is not continuous with the previously read address
- QSPIn_SSL negated after the SSLN bit in DRCR is set to 1

#### (2) Manual Mode

##### (a) QSPIn_SSL pin assertion not retained (SSLKP bit in SMCR = 0)

QSPIn_SSL negated after completing the data transfer and t2 cycle.

##### (b) QSPIn_SSL pin assertion retained (SSLKP bit in SMCR = 1)

QSPIn_SSL not negated.

When to be negated, data should be transferred after setting the SSLKP bit to 0.

### 25.5.11 Flags

This module has two flag bits SSLF and TEND in CMNSR. These bits are read-only bits.

#### (1) SSLF Bit

This bit indicates the QSPIn_SSL pin status. The status is 1 when the QSPIn_SSL is asserted, and the status is 0 when the QSPIn_SSL is negated.

#### (2) TEND Bit

This bit indicates whether transfer of data is in progress or the transfer of data has ended.

During t1 time period, data transfer, t2 time period, t3 time period, and waiting for read access by burst read and QSPIn_SSL automatic negation, the TEND bit is read as 0 to indicate that the transfer of data is in progress.

Otherwise, the TEND bit is read as 1 to indicate that transfer of data has ended.

#### (3) Register Re-writing Timing

The status of the TEND bit determines the rewritable registers.

The registers which can be written to, except the SSLN bit in DRCR, should be modified when TEND = 1. Read SMRDR0 and SMRDR1 when TEND = 1. CMNSR can always be read.

### 25.5.12 Write Buffer Operation

This module uses the read cache space also as the write buffer space during write operation. The write buffer improves the performance of write access.

Table 25.9 Write Buffer Space

| Address for Cortex-A55 | Access Size |
|---|---|
| H'0_1007 0000 ~ H'0_1007 00FF | 4, 8, 16, 32, or 64 bytes |

**Note:** Be sure to sequentially access this space beginning from its start address with the transfer size set to eight bytes.
If access is non-sequential or random, correct operation is not guaranteed.
After an operation in Manual mode or with the use of the cache area as the write buffer, clearing of this cache area by writing 1 to the RCF bit is required.

**Figure 25.18** shows the procedure for using the write buffer. For the initial settings in Manual mode, see **Figure 25.13**.

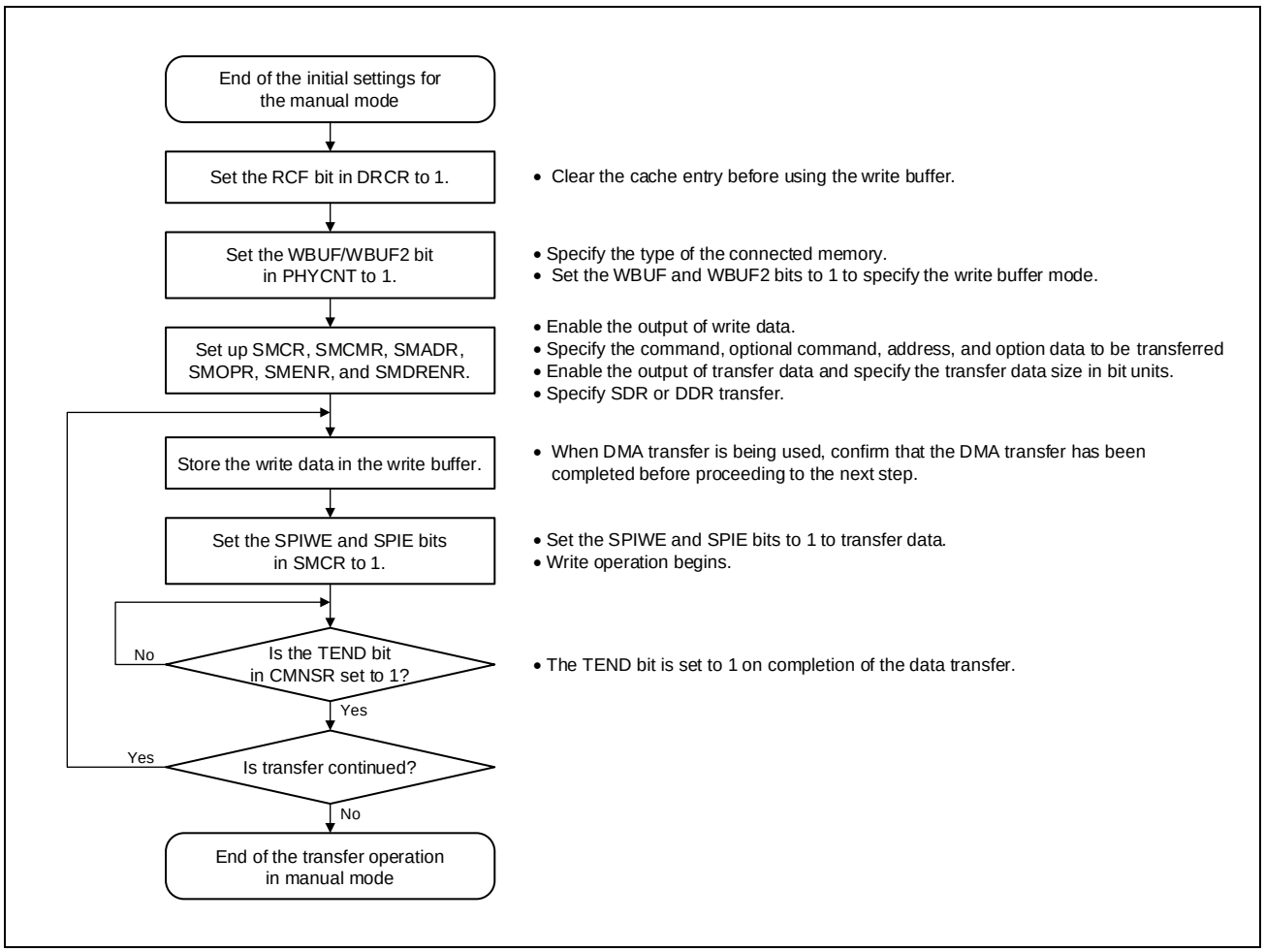



Figure 25.18 Procedure for Using the Write Buffer

### 25.5.13 Supported Protocol for Serial Flash Memory

The protocol for serial flash memory that this LSI supports is indicated in **Table 25.10**.

Table 25.10 Supported Protocol for Serial Flash Memory

| Bit Width (Command-Address-Data) | Command | Address | Data |
|---|---|---|---|
| 1-0-0 | SDR | NA | NA |
| 1-0-1 | SDR | NA | SDR |
| 1-1-1 | SDR | SDR | SDR |
| | SDR | DDR | DDR |
| 1-1-4 | SDR | SDR | SDR |
| 1-4-4 | SDR | SDR | SDR |
| | SDR | DDR | DDR |

### 25.5.14 Timing Adjustment

When data is transferred with serial flash memory connected, adjustment of the timing between the clock operation and the acquisition of input data is required.

**Figure 25.19** shows the timing adjustment flow for serial flash (SDR mode) connection.

Execute this flow before changing the setting of SCLKSEL.SPICR[1:0] from its initial value when the serial flash memory is connected (SDR mode). Execution of this flow at regular intervals is not required.

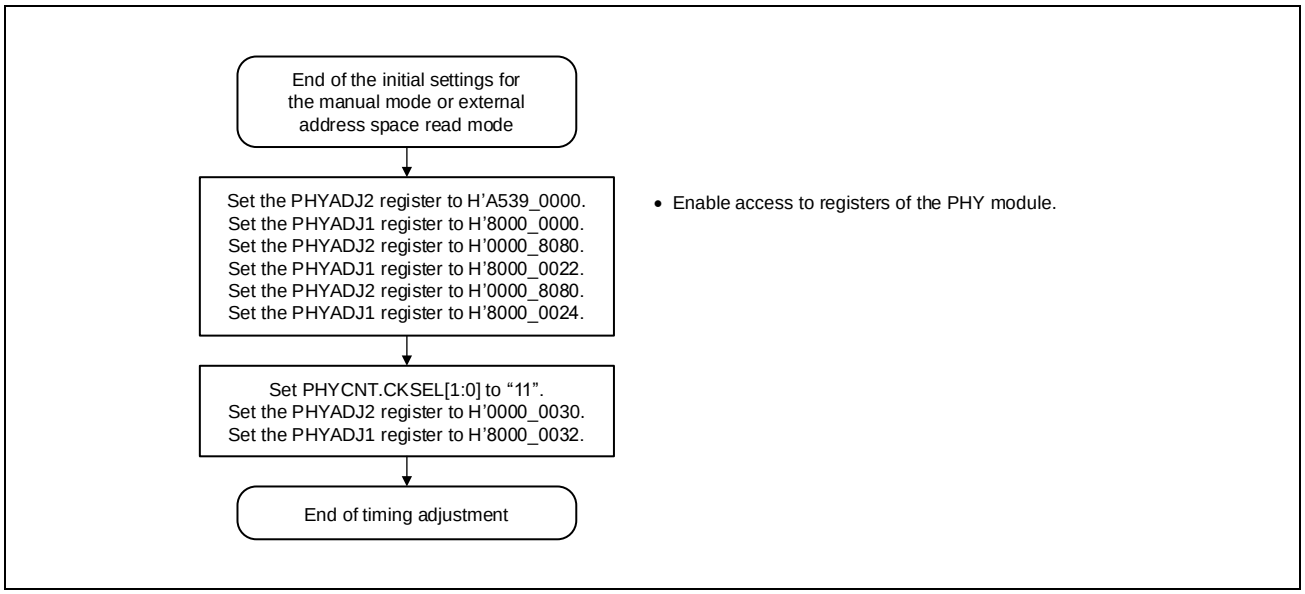



Figure 25.19 Flow of Timing Adjustment when the Serial Flash Memory is Connected (SDR Mode)

**Figure 25.20** shows the timing adjustment flow for serial flash (DDR mode) connection.

Since the set value acquired in this timing adjustment flow is affected by voltage and temperature variation, this flow should be performed periodically.

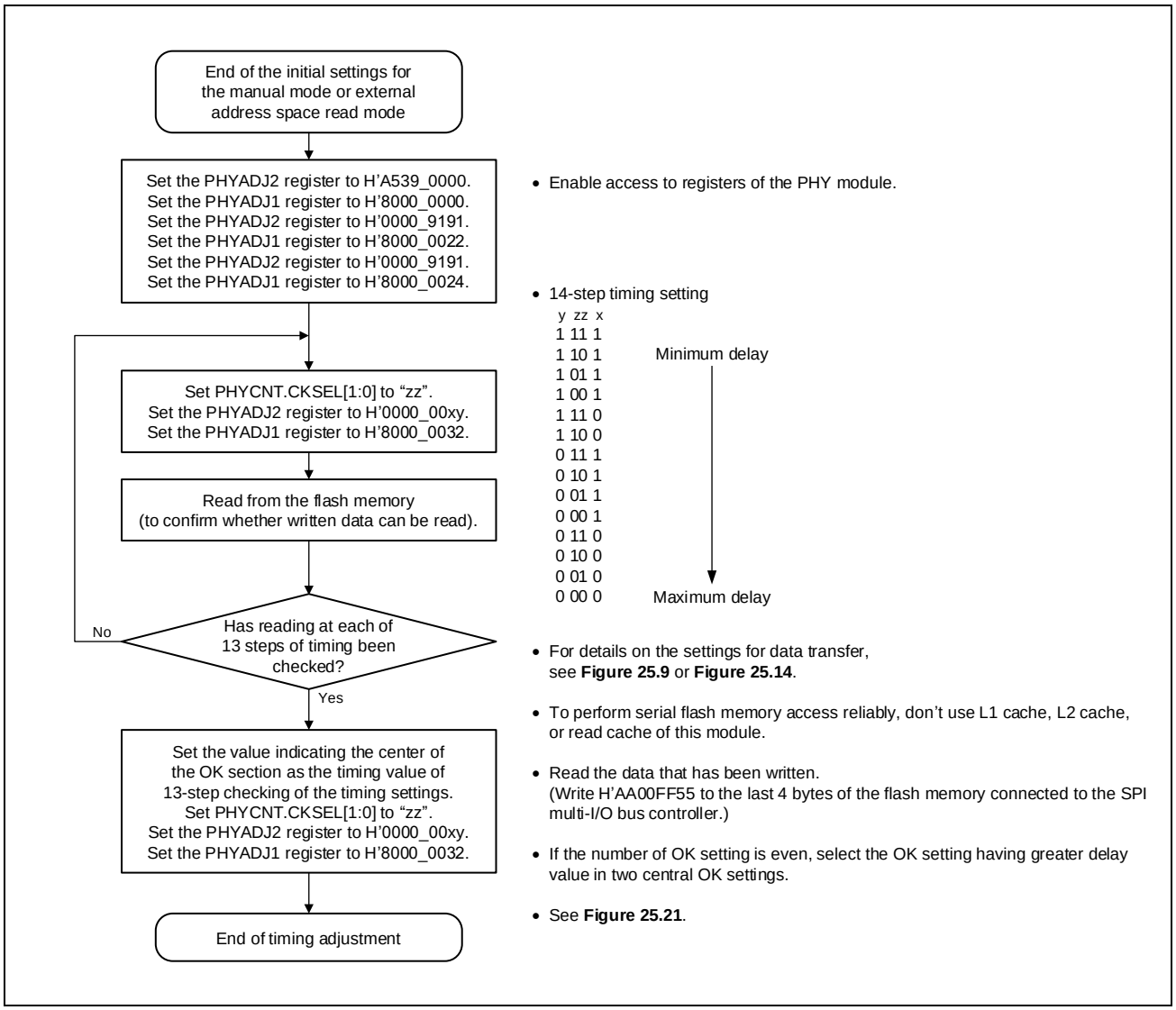
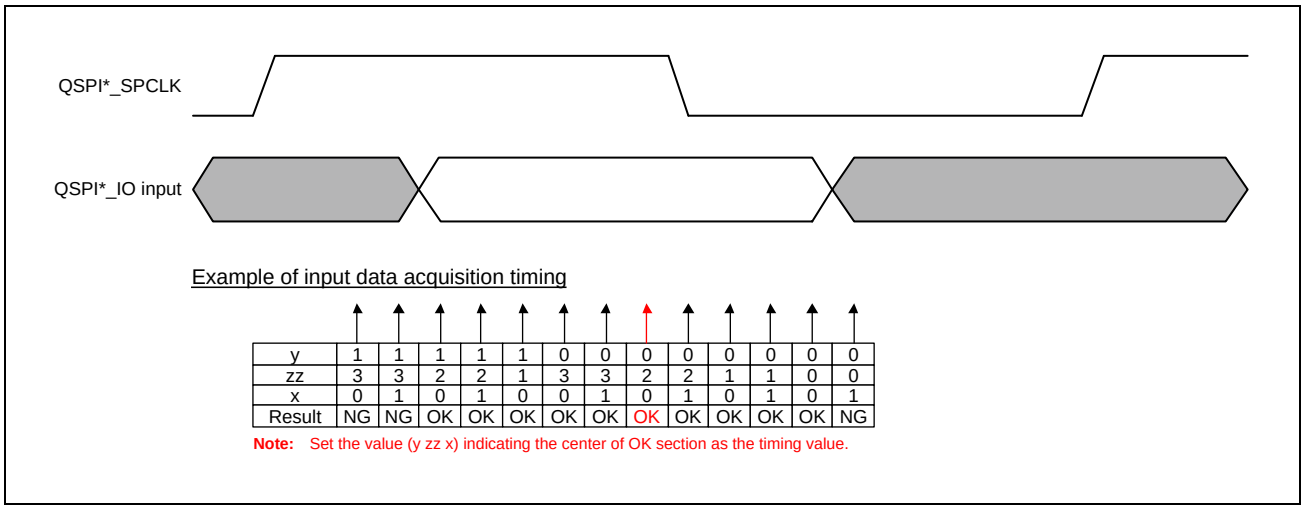



Figure 25.20 Flow of Timing Adjustment when the Serial Flash Memory is Connected (DDR Mode)

**Figure 25.21** shows the timing adjustment flow for serial flash (DDR mode) connection.




| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| y | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| zz | 3 | 3 | 2 | 2 | 1 | 3 | 3 | 2 | 2 | 1 | 1 | 0 | 0 |
| x | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| Result | NG | NG | OK | OK | OK | OK | OK | OK | OK | OK | OK | OK | NG |

**Note:** Set the value (y zz x) indicating the center of OK section as the timing value.

Figure 25.21 Example of Timing Adjustment when the Serial Flash Memory is Connected (DDR Mode)

## 25.5.15 Data Alignment

Data alignments in external address space read mode and Manual mode are shown in **Figure 25.22** and **Figure 25.23**, respectively.

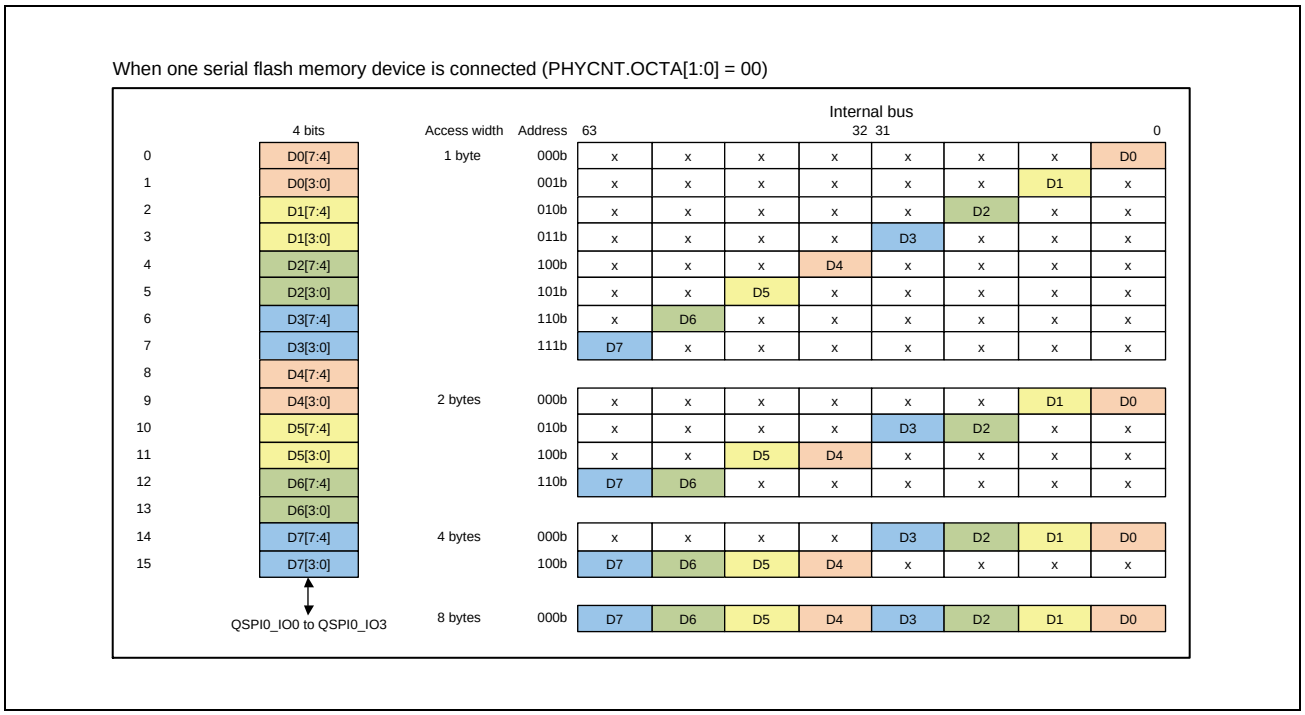



Figure 25.22 Data Alignment in External Address Space Read Mode

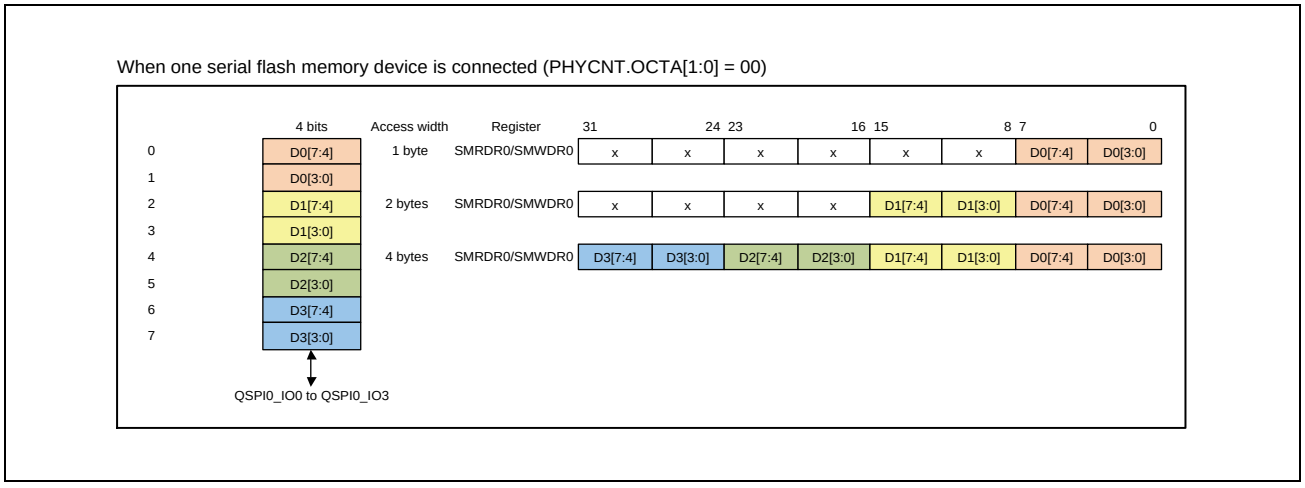



Figure 25.23 Data Alignment in Manual Mode

## 25.6 Usage Notes

### 25.6.1 Transfer to read data while the signal on the QSPIn_SSL pin is de-asserted

Set the SMENR.SPIDE[3:0] bits to 1100 or 1111 when transfer only for reading data is to proceed.

Transfer will not proceed normally if the setting of the SMENR.SPIDE[3:0] bits is 1000.

### 25.6.2 Transfer to read data while the signal on the QSPIn_SSL pin is asserted

When transfer only for reading data is to proceed, set the SMENR.SPIDE[3:0] bits to 1100 or 1111, or end the immediately preceding transfer with reading data.

When the immediately preceding transfer is of a command, optional command, address, or option data, or is transfer for writing data, the subsequent transfer only for reading data will not proceed normally if the setting of the SMENR.SPIDE[3:0] bits is 1000.

### 25.6.3 Notes on Starting Transfer from the QSPIn_SSL Retained State in Manual Mode

Be sure to set the SPIWE bit in the SMCR register to 1 when the transfer of a command, optional command, address, or option data is started while the QSPIn_SSL pin is being asserted in Manual mode.

In addition, while the QSPIn_SSL signal is being asserted in Manual mode, the setting to start data transfer for reading is prohibited.

# 26. I²C Bus Interface

This section gives an overall description of the I²C bus interface (RIIC).

The first section describes the features specific to this LSI, including the number of units and the register base addresses. The subsequent sections describe the RIIC's functions and registers.

## 26.1 Features

### 26.1.1 Channels

This LSI has the following number of channels of the I²C bus interface (RIIC).

Table 26.1 Channels of RIIC

| Item | Description |
|---|---|
| Number of channels | 2 |
| Name | RIICn (n = 0, 1) |

Table 26.2 Index

| Index | Description |
|---|---|
| n | Throughout this section, the individual channels of the I²C bus interface are identified by the index "n" (n = 0, 1); for example, RIICnCR1 for the I²C bus control register 1. |

### 26.1.2 Register Base Addresses

The base address <RIICn_base> of each RIICn is listed in the following table.

All RIICn register addresses are given as values obtained by adding offsets to the register base address <RIICn_base> for each channel.

Table 26.3 Register Base Address

| Channel | Base Address Name | Base Address |
|---|---|---|
| RIIC0 | <RIIC0_base> | H'0_1005_8000 (Cortex-A55 Address Space) |
| RIIC1 | <RIIC1_base> | H'0_1005_8400 (Cortex-A55 Address Space) |

### 26.1.3 External I/O Signals

The following table shows the external I/O signals of the RIIC.

Table 26.4 RIICn Pin Configuration

| Channel | Alternative Port Pin Name | Function |
|---|---|---|
| RIIC0 | RIIC0_SCL | RIIC0 serial clock I/O pin |
| | RIIC0_SDA | RIIC0 serial data I/O pin |
| RIIC1 | RIIC1_SCL | RIIC1 serial clock I/O pin |
| | RIIC1_SDA | RIIC1 serial data I/O pin |

Table 26.5 I2C Bus Interface

| Channel Number*3 | Pin Name*1 | I/O | Function | This LSI Output Buffer Type*2 |
|---|---|---|---|---|
| 0 | SCL | I/O | SCL: I2C serial clock input/output pin<br>SDA: I2C serial data input/output pin | OD |
| | SDA | | | |
| 1 | SCL | | | LVTTL |
| | SDA | | | |

Note 1. The actual pin names are RIICnSCL and RIICnSDA.

Note 2. Output buffer type: “OD” is open drain buffer and “LVTTL” is low level drive only LVTTL buffer.

Note 3. AC specification is different between channel 0 and channel 1.

## 26.2 Overview

### 26.2.1 Functional Overview

#### Communications format

- $I^2C$ bus format or SMBus format
- Master mode or slave mode selectable
- Automatic securing of the various set-up times, hold times, and bus-free times for the transfer rate

#### Transfer rate

Up to 1 Mbps

#### SCL clock

For master operation, the duty cycle of the SCL clock is selectable in the following range. 0% < Duty cycle < 100%.

#### Issuing and detecting conditions

- Start, restart, and stop conditions are automatically generated.
- Start conditions (including restart conditions) and stop conditions are detected.

#### Slave address

- Up to three slave-address settings can be made.
- Seven- and ten-bit address formats are supported (along with the use of both at once).
- General call addresses, device ID addresses, and SMBus host addresses are detected.

#### Acknowledgement

- For transmission, the acknowledge bit is automatically loaded
  - Transfer of the next data for transmission can be automatically suspended on detection of a not-acknowledge bit.
- For reception, the acknowledge bit is automatically transmitted
  - If a wait between the eighth and ninth clock cycles has been selected, software control of the value in the acknowledge field in response to the received value is possible.

#### Wait function

- In reception, the following periods of waiting can be obtained by holding the clock signal (SCL) at the low level:
  - Waiting between the eighth and ninth clock cycles
  - Waiting between the ninth clock cycle and the first clock cycle of the next transfer (WAIT function)

#### SDA output delay function

Timing of the output of transmitted data, including the acknowledge bit, can be delayed.

### Arbitration

- For multi-master operation
  - Operation to synchronize the SCL (clock) signal in cases of conflict with the SCL signal from another master is possible.
  - When issuing the start condition would create conflict on the bus, loss of arbitration is detected by testing for non-matching between the internal signal for the SDA line and the level on the SDA line.
  - In master operation, loss of arbitration is detected by testing for non-matching between the signal on the SDA line and the internal signal for the SDA line.
- Loss of arbitration due to detection of the start condition while the bus is busy is detectable (to prevent the issuing of double start conditions).
- Loss of arbitration in transfer of a not-acknowledge bit due to the internal signal for the SDA line and the level on the SDA line not matching is detectable.
- Loss of arbitration due to non-matching of internal and line levels for data is detectable in slave transmission.

### Timeout function

The internal time-out function is capable of detecting long-interval stop of the SCL (clock signal).

### Noise removal

The interface incorporates analog noise filters and digital noise filters for input on the RIICnSCL and RIICnSDA pins, and the width for noise cancellation by the digital noise filters is adjustable by software.

### Interrupt sources

- Eight sources:
  - Transmission complete
  - Receive-data-full
  - Transmit-data-empty
  - Detection of a stop condition
  - Detection of a start condition
  - Reception of a NACK
  - Arbitration lost
  - Timeout

### Low power consumption function

Module-stop state can be set.

## 26.2.2 Block Diagram

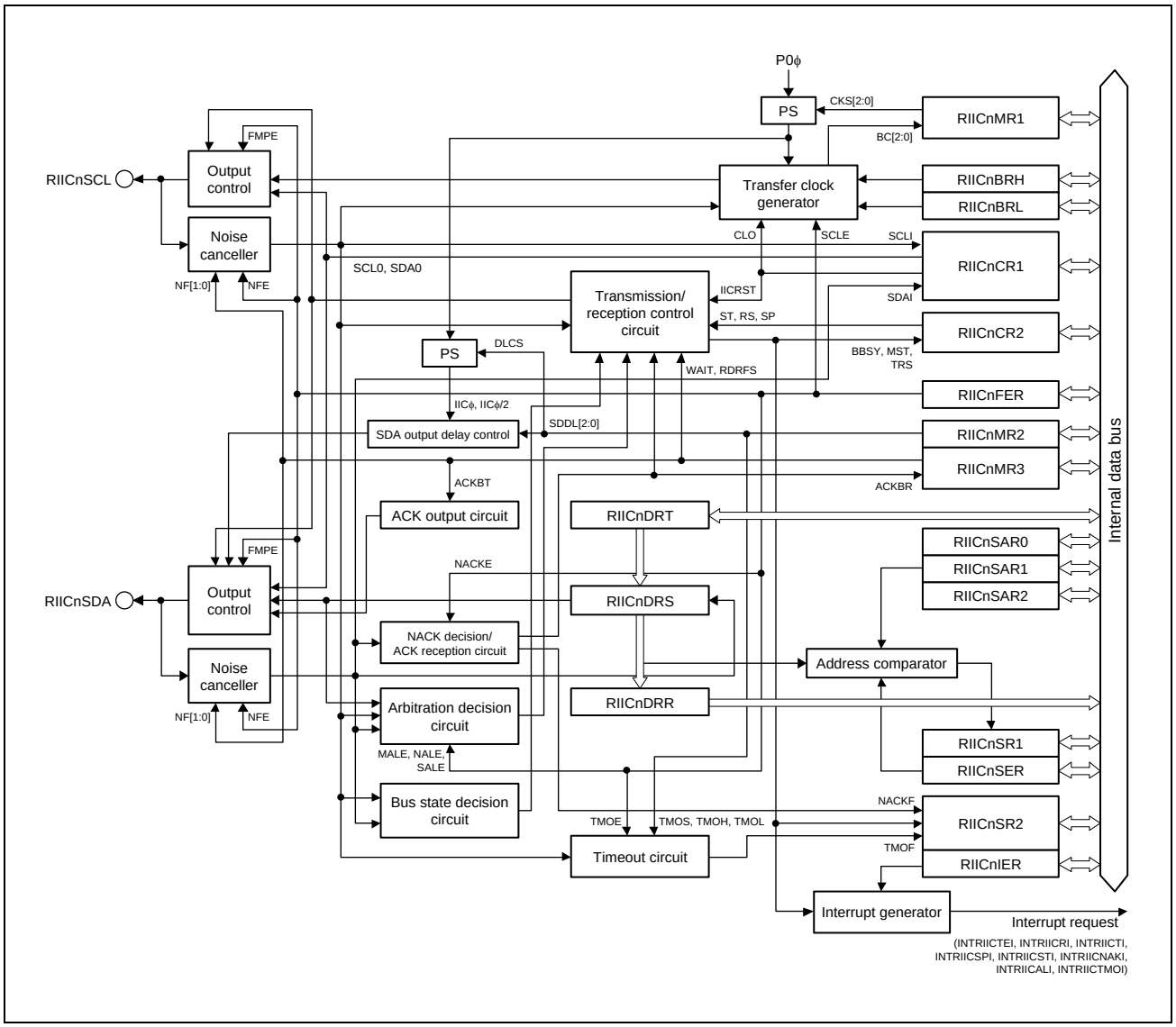



Figure 26.1 Block Diagram of RIIC

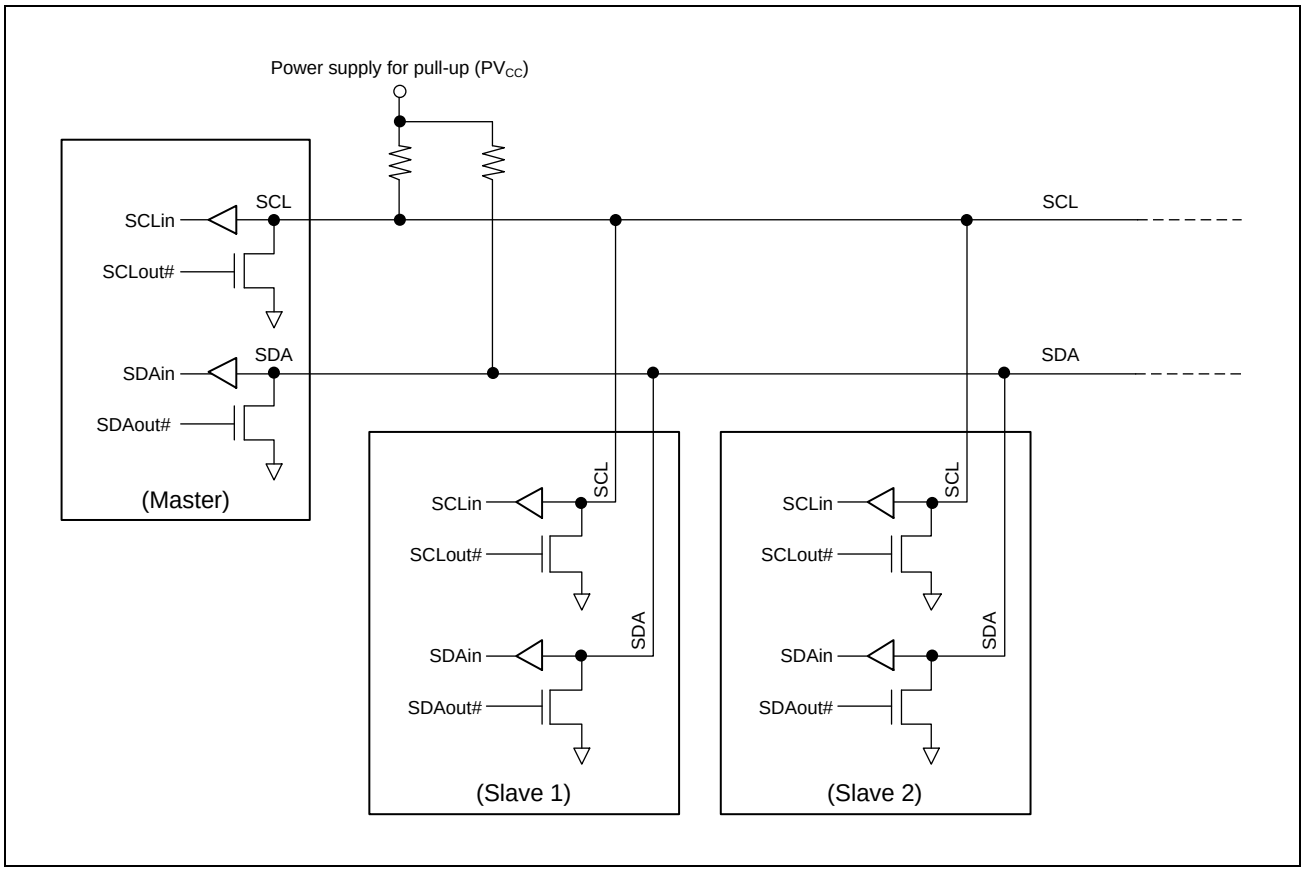



Figure 26.2 Connections to the External Circuit by the I/O Pins ($I^2C$ Bus Configuration Example)

RIICnSCL and RIICnSDA are Schmitt input/open-drain output pins for both master and slave operations. Because the output is open drain, an external pull-up resistor is required.

## 26.3 Registers

Table 26.6 Register Configuration

| Channel | Register Name | Abbreviation | Offset Address | Access Size |
|---|---|---|---|---|
| RIICn | $I^2C$ bus control register 1 | ICCR1 | H'00 | 8, 16, 32 |
| (n = 0, 1) | $I^2C$ bus control register 2 | ICCR2 | H'04 | 8, 16, 32 |
| | $I^2C$ bus mode register 1 | ICMR1 | H'08 | 8, 16, 32 |
| | $I^2C$ bus mode register 2 | ICMR2 | H'0C | 8, 16, 32 |
| | $I^2C$ bus mode register 3 | ICMR3 | H'10 | 8, 16, 32 |
| | $I^2C$ bus function enable register | ICFER | H'14 | 8, 16, 32 |
| | $I^2C$ bus status enable register | ICSER | H'18 | 8, 16, 32 |
| | $I^2C$ bus interrupt enable register | ICIER | H'1C | 8, 16, 32 |
| | $I^2C$ bus status register 1 | ICSR1 | H'20 | 8, 16, 32 |
| | $I^2C$ bus status register 2 | ICSR2 | H'24 | 8, 16, 32 |
| | $I^2C$ slave address register 0 | ICSAR0 | H'28 | 8, 16, 32 |
| | $I^2C$ slave address register 1 | ICSAR1 | H'2C | 8, 16, 32 |
| | $I^2C$ slave address register 2 | ICSAR2 | H'30 | 8, 16, 32 |
| | $I^2C$ bus bit rate low-level register | ICBRL | H'34 | 8, 16, 32 |
| | $I^2C$ bus bit rate high-level register | ICBRH | H'38 | 8, 16, 32 |
| | $I^2C$ bus transmit data register | ICDRT | H'3C | 8, 16, 32 |
| | $I^2C$ bus receive data register | ICDRR | H'40 | 8, 16, 32 |

## 26.3.1 RIICnCR1 — I²C Bus Control Register 1

**Access Size:** RIICnCR1 is a 32-bit readable/writable register.
RIICnCR1L and RIICnCR1H are 16-bit readable/writable registers.
RIICnCR1LL, RIICnCR1LH, RIICnCR1HL, and RIICnCR1HH are 8-bit readable/writable registers.

**Address(es):** RIICnCR1: <RIICn_base> + H'0000
RIICnCR1L: <RIICn_base> + H'0000, RIICnCR1H: <RIICn_base> + H'0002
RIICnCR1LL: <RIICn_base> + H'0000, RIICnCR1LH: <RIICn_base> + H'0001,
RIICnCR1HL: <RIICn_base> + H'0002, RIICnCR1HH: <RIICn_base> + H'0003

**Initial Value:** H'0000 001F. This register is initialized by any reset.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | ICE | IICRST | CLO | SOWP | SCLO | SDAO | SCLI | SDAI |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 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 | R |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 8 | — | All 0 | R | Reserved<br>These bits are read as 0. The write value should be 0. |
| 7 | ICE | 0 | R/W | I²C Bus Interface Enable<br>0: Output to the RIICnSCL and RIICnSDA pins is disabled. (Input to the RIICnSCL and RIICnSDA pins is enabled.)<br>1: Enabled (RIICnSCL and RIICnSDA pins are in the driving state)<br>(An RIIC reset or an internal reset is selected according to the combination of this bit and IICRST bit settings.) |
| 6 | IICRST | 0 | R/W | I²C Bus Interface Internal Reset<br>0: Clears the RIIC reset or internal reset.<br>1: Initiates the RIIC reset or internal reset.<br>(Clears the bit counter and the SCL/SDA output latch) |
| 5 | CLO | 0 | R/W | Extra SCL Clock Cycle Output<br>0: Does not output an extra SCL clock cycle (default).<br>1: Outputs an extra SCL clock cycle.<br>(The CLO bit is cleared automatically after one clock cycle is output.) |
| 4 | SOWP*2 | 1 | R/W | SCLO/SDAO Write Protect<br>0: Allows the SCLO and SDAO bits to be rewritten.<br>(This bit is read as 1.) |
| 3 | SCLO*1, *2 | 1 | R/W | SCL Output Control<br>• Read:<br>0: RIICnSCL pin output is at a low level.<br>1: RIICnSCL pin is in a high-impedance state.<br>• Write:<br>0: Changes the RIICnSCL pin output to a low level.<br>1: Changes the RIICnSCL pin in a high-impedance state.<br>(High level output is achieved through an external pull-up resistor.) |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 2 | SDAO*1,*2 | 1 | R/W | SDA Output Control<br>• Read:<br>0: RIICnSDA pin output is at a low level.<br>1: RIICnSDA pin is in a high-impedance state.<br>• Write:<br>0: Changes the RIICnSDA pin output to a low level.<br>1: Changes the RIICnSDA pin in a high-impedance state.<br>(High level output is achieved through an external pull-up resistor.) |
| 1 | SCLI | 1 | R | SCL Bus Input Monitor<br>0: RIICnSCL pin input is at a low level.<br>1: RIICnSCL pin input is at a high level. |
| 0 | SDAI | 1 | R | SDA Bus Input Monitor<br>0: RIICnSDA pin input is at a low level.<br>1: RIICnSDA pin input is at a high level. |

Note 1. Do not write to these bits during communication. Changing a value during communication may cause a transmission or reception failure or an AL error.

Note 2. To change the SDAO and SCLO bits, set the SOWP bit to 0 at the same timing to set the SDAO and SCLO bits to 0.

### CLO Bit (Extra SCL Clock Cycle Output)

This bit is used to output an extra SCL clock cycle for debugging or error processing.

Normally, set the bit to 0. Setting the bit to 1 in a normal communication state causes a communication error. For details on this function, see **Section 26.13.2, Extra SCL Clock Cycle Output Function**.

### IICRST Bit ($I^2C$ Bus Interface Internal Reset)

This bit is used to reset the internal states of the RIIC. Setting this bit to 1 initiates an RIIC reset or internal reset.

Whether an RIIC reset or internal reset is initiated is determined according to the combination with the ICE bit. **Table 26.7** lists the resets of the RIIC.

The RIIC reset resets all registers including the RIICnCR2.BBSY flag (except ICE and IICRST) and internal states of the RIIC, and the internal reset resets the bit counter (RIICnMR1.BC[2:0] bits), the $I^2C$ bus shift register (RIICnDRS), and the $I^2C$ bus status registers (RIICnSR1 and RIICnSR2) as well as the internal states of the RIIC. For the reset conditions for each register, see **Section 26.15, Reset Function of RIIC**.

An internal reset initiated with the IICRST bit set to 1 during operation (with the ICE bit set to 1) resets the internal states of the RIIC without initializing the port settings and the control and setting registers of the RIIC when the bus or RIIC hangs up due to a communication error.

If the RIIC hangs up in a low level output state, resetting the internal states cancels the low level output state and releases the bus with the RIICnSCL pin and RIICnSDA pin at a high impedance.

**CAUTION**

If an internal reset is initiated using the IICRST bit for a bus hang-up occurred during communication with the master device in slave mode, the states may become different between the slave device and the master device (due to the difference in the bit counter information). For this reason, do not initiate an internal reset in slave mode, but initiate restoration processing from the master device. If an internal reset is necessary because the RIIC hangs up with the SCL line in a low level output state in slave mode, initiate an internal reset and then issue a restart condition from the master device or resume communication from the start condition issuance after issuing a stop condition. If communication is restarted by initiating a reset solely in the slave device without issuing a start condition or restart condition from the master device, synchronization will be lost because the master and slave devices operate asynchronously.

Table 26.7 RIIC Resets

| IICRST | ICE | State | Specifications |
|---|---|---|---|
| 1 | 0 | RIIC reset | Resets all registers (except ICE and IICRST) and internal states of the RIIC. |
| | 1 | Internal reset | Reset the RIICnMR1.BC[2:0] bits, and the RIICnSR1, RIICnSR2, RIICnDRS registers and the internal states of the RIIC. |

### ICE Bit ($I^2C$ Bus Interface Enable)

The ICE bit selects driving or non-driving of the RIICnSCL and RIICnSDA pins. Moreover, this bit can perform two types of reset in combination with the IICRST bit. For the types of reset, see **Table 26.7**.

Set the ICE bit to 1 when using RIIC. Setting the ICE bit to 1 selects driving of the RIICnSCL and RIICnSDA pins. Set the ICE bit to 0 when RIIC is not to be used. Clearing the ICE bit to 0 disables output from the RIICnSCL and RIICnSDA pins.

## 26.3.2 RIICnCR2 — I²C Bus Control Register 2

**Access Size:** RIICnCR2 is a 32-bit readable/writable register.
RIICnCR2L and RIICnCR2H are 16-bit readable/writable registers.
RIICnCR2LL, RIICnCR2LH, RIICnCR2HL, and RIICnCR2HH are 8-bit readable/writable registers.

**Address(es):** RIICnCR2: <RIICn_base> + H'0004
RIICnCR2L: <RIICn_base> + H'0004, RIICnCR2H: <RIICn_base> + H'0006
RIICnCR2LL: <RIICn_base> + H'0004, RIICnCR2LH: <RIICn_base> + H'0005,
RIICnCR2HL: <RIICn_base> + H'0006, RIICnCR2HH: <RIICn_base> + H'0007

**Initial Value:** H'0000 0000 This register is initialized by any reset.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | BBSY | MST | TRS | — | SP | RS | ST | — |
| 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 |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 8 | — | All 0 | R | Reserved<br>These bits are read as 0. The write value should be 0. |
| 7 | BBSY | 0 | R | Bus Busy Detection Flag<br>0: The I²C bus is released (bus free state).<br>1: The I²C bus is occupied (bus busy state or in the bus free state). |
| 6 | MST | 0 | R | Master/Slave Mode<br>0: Slave mode<br>1: Transmit mode |
| 5 | TRS | 0 | R | Transmit/Receive Mode<br>0: Receive mode<br>1: Transmit mode |
| 4 | — | 0 | R | Reserved<br>These bits are read as 0. The write value should be 0. |
| 3 | SP | 0 | R/W | Stop Condition Issuance Request<br>0: Does not request to issue a stop condition.<br>1: Requests to issue a stop condition. |
| 2 | RS | 0 | R/W | Restart Condition Issuance Request<br>0: Does not request to issue a restart condition.<br>1: Requests to issue a restart condition. |
| 1 | ST | 0 | R/W | Start Condition Issuance Request<br>0: Does not request to issue a start condition.<br>1: Requests to issue a start condition. |
| 0 | — | 0 | R | Reserved<br>These bits are read as 0. The write value should be 0. |

### ST Bit (Start Condition Issuance Request)

This bit is used to request transition to master mode and issuance of a start condition.

When this bit is set to 1 to request to issue a start condition, a start condition is issued when the BBSY flag is set to 0 (bus free).

For details on the start condition issuance, see **Section 26.12, Start Condition/Restart Condition/Stop Condition Issuing Function**.

[Setting condition]

When 1 is written to the ST bit

[Clearing conditions]

- When 0 is written to the ST bit
- When a start condition has been issued
- When the RIICnSR2.AL (arbitration-lost) flag is set to 1
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

**CAUTION**

Set the ST bit to 1 (start condition issuance request) when the BBSY flag is set to 0 (bus free).

Note that arbitration may be lost if the ST bit is set to 1 (start condition issuance request) when the BBSY flag is set to 1 (bus busy).

### RS Bit (Restart Condition Issuance Request)

This bit is used to request that a restart condition be issued in master mode.

When this bit is set to 1 to request to issue a restart condition, a restart condition is issued when the BBSY flag is set to 1 (bus busy) and the MST bit is set to 1 (master mode).

For details on the restart condition issuance, see **Section 26.12, Start Condition/Restart Condition/Stop Condition Issuing Function**.

[Setting condition]

When 1 is written to the RS bit with the RIICnCR2.BBSY flag set to 1

[Clearing conditions]

- When 0 is written to the RS bit
- When a restart condition has been issued or a start condition is detected
- When a stop condition is detected
- When the RIICnCR2.AL (arbitration-lost) flag is set to 1
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

**CAUTIONS**

1. Do not set the RS bit to 1 while issuing a stop condition.
2. It is commended to issue a restart condition in master transmit mode. If the RS bit is set to 1 (restart condition issuance request) in mode other than master mode, the restart condition is not issued in this mode but the restart condition issuance request bit remains set. If the operating mode changes to master mode with the bit not being cleared, the restart condition may be issued.

### SP Bit (Stop Condition Issuance Request)

This bit is used to request that a stop condition be issued in master mode.

When this bit is set to 1 to request to issue a stop condition, a stop condition is issued when the BBSY flag is set to 1 (bus busy) and the MST bit is set to 1 (master mode).

For details on the stop condition issuance, see **Section 26.12, Start Condition/Restart Condition/Stop Condition Issuing Function**.

[Setting condition]

When 1 is written to the SP bit with both the RIICnCR2.BBSY flag and the RIICnCR2.MST bit set to 1

[Clearing conditions]

- When 0 is written to the SP bit
- When a stop condition has been issued or a stop condition is detected
- When the RIICnSR2.AL (arbitration-lost) flag is set to 1
- When a start condition and a restart condition are detected
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

**CAUTIONS**

1. Wring to the SP bit is not possible while the setting of the BBSY flag is 0 (bus free).
2. Do not set the SP bit to 1 while a restart condition is being issued.

### TRS Bit (Transmit/Receive Mode)

This bit indicates transmit or receive mode.

The RIIC is in receive mode when the TRS bit is set to 0 and is in transmit mode when the bit is set to 1. Combination of this bit and the MST bit indicates the operating mode of the RIIC.

The value of the TRS bit is automatically changed to the value for transmission mode or reception mode by detection or issuing of a start condition, setting or clearing of the R/W# bit, etc.

[Setting conditions]

- When a start condition is issued normally according to the start condition issuance request (when a start condition is detected with the ST bit set to 1)
- When the R/W# bit added to the slave address is set to 0 in master mode
- When the address received in slave mode matches the address enabled in RIICnSER, with the R/W# bit set to 1

[Clearing conditions]

- When a stop condition is detected
- The RIICnSR2.AL (arbitration-lost) flag being set to 1
- In master mode, reception of a slave address to which an R/W# bit with the value 1 is appended
- In slave mode, a match between the received address and the address enabled in RIICnSER when the value of the received R/W# bit is 0 (including cases where the received address is the general call address)
- In slave transmit mode, a restart condition is detected (a restart condition is detected with RIICnCR2.BBSY = 1 and RIICnCR2.MST = 0)
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

### MST Bit (Master/Slave Mode)

This bit indicates master or slave mode.

The RIIC is in slave mode when the MST bit is set to 0 and is in master mode when the bit is set to 1. Combination of this bit and the TRS bit indicates the operating mode of the RIIC.

The value of the MST bit is automatically changed to the value for master mode or slave mode by detection or issuing of a start condition, etc.

[Setting conditions]

- When a start condition is issued normally according to the start condition issuance request (when a start condition is detected with the ST bit set to 1)

[Clearing conditions]

- When a stop condition is detected
- When the RIICnSR2.AL (arbitration-lost) flag is set to 1
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

### BBSY Flag (Bus Busy Detection)

The BBSY flag indicates whether the $I^2C$ bus is occupied (bus busy) or released (bus free).

This bit is set to 1 when the SDA line changes from high to low under the condition of SCL = high, assuming that a start condition has been issued.

When the SDA line changes from low to high under the condition of SCL = high, this bit is cleared to 0 after the bus free time (specified in RIICnBRL) start condition is not detected, assuming that a stop condition has been issued.

[Setting condition]

- When a start condition is detected

[Clearing conditions]

- When the bus free time (specified in RIICnBRL) start condition is not detected after detecting a stop condition
- When 1 is written to the RIICnCR1.IICRST bit with the RIICnCR1.ICE bit set to 0 (RIIC reset)

## 26.3.3 RIICnMR1 — I²C Bus Mode Register 1

**Access Size:** RIICnMR1 is a 32-bit readable/writable register.
RIICnMR1L and RIICnMR1H are 16-bit readable/writable registers.
RIICnMR1LL, RIICnMR1LH, RIICnMR1HL, and RIICnMR1HH are 8-bit readable/writable registers.

**Address(es):** RIICnMR1: <RIICn_base> + H'0008
RIICnMR1L: <RIICn_base> + H'0008, RIICnMR1H: <RIICn_base> + H'000A
RIICnMR1LL: <RIICn_base> + H'0008, RIICnMR1LH: <RIICn_base> + H'0009,
RIICnMR1HL: <RIICn_base> + H'000A, RIICnMR1HH: <RIICn_base> + H'000B

**Initial Value:** H'0000 0008 This register is initialized by any reset.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | CKS[2:0] | | | BCWP | BC[2:0] | | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| R/W | R | R | R | R | 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 |
|---|---|---|---|---|
| 31 to 7 | — | All 0 | R | Reserved<br>These bits are read as 0. The write value should be 0. |
| 6 to 4 | CKS[2:0] | All 0 | R/W | Internal Reference Clock (IICϕ) Selection<br>b6 b4<br>0 0: IICϕ = P0ϕ/1<br>0 0: IICϕ = P0ϕ/2<br>0 1: IICϕ = P0ϕ/4<br>0 1: IICϕ = P0ϕ/8<br>1 0: IICϕ = P0ϕ/16<br>1 0: IICϕ = P0ϕ/32<br>1 1: IICϕ = P0ϕ/64<br>1 1: IICϕ = P0ϕ/128 |
| 3 | BCWP*1 | 1 | R/W | BC Write Protect<br>0: Enables a value to be written in the BC[2:0] bits.<br>(This bit is read as 1.) |
| 2 to 0 | BC[2:0] | All 0 | R/W | Bit Counter<br>b2 b0<br>0 0: 9 bits<br>0 0: 1: 2 bits<br>0 1: 0: 3 bits<br>0 1: 1: 4 bits<br>1 0: 0: 5 bits<br>1 0: 1: 6 bits<br>1 1: 0: 7 bits<br>1 1: 1: 8 bits |

Note 1. When rewriting the BC[2:0] bits, write 0 to the BCWP bit simultaneously.

### BC[2:0] Bits (Bit Counter)

These bits function as a counter that indicates the number of bits remaining to be transferred at the detection of a rising edge on the SCL line. Although these bits are writable and readable, it is not necessary to access these bits under normal conditions.

To write to these bits, specify the number of bits to be transferred plus one (data is transferred with an additional acknowledge bit) between transferred frames when the SCL line is at a low level.

The values of the BC[2:0] bits return to 000b at the end of a data transfer including the acknowledge bit or when a start condition including a restart condition is detected.

## 26.3.4 RIICnMR2 — I²C Bus Mode Register 2

**Access Size:** RIICnMR2 is a 32-bit readable/writable register.
RIICnMR2L and RIICnMR2H are 16-bit readable/writable registers.
RIICnMR2LL, RIICnMR2LH, RIICnMR2HL, and RIICnMR2HH are 8-bit readable/writable registers.

**Address(es):** RIICnMR2: <RIICn_base> + H'000C
RIICnMR2L: <RIICn_base> + H'000C, RIICnMR2H: <RIICn_base> + H'000E
RIICnMR2LL: <RIICn_base> + H'000C, RIICnMR2LH: <RIICn_base> + H'000D,
RIICnMR2HL: <RIICn_base> + H'000E, RIICnMR2HH: <RIICn_base> + H'000F

**Initial Value:** H'0000 0006 This register is initialized by any reset.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | DLCS | SDDL[2:0] | | | — | TMOH | TMOL | TMOS |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 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<br>These bits are read as 0. The write value should be 0. |
| 7 | DLCS | 0 | R/W | SDA Output Delay Clock Source Selection<br>0: The internal reference clock (IICϕ) is selected as the clock source of the SDA output delay counter.<br>1: The internal reference clock divided by 2 (IICϕ/2) is selected as the clock source of the SDA output delay counter.*1 |
| 6 to 4 | SDDL[2:0] | All 0 | R/W | SDA Output Delay Counter<br>• When RIICnMR2.DLCS = 0 (IICϕ)<br>b6 b4<br>0 0 0: No output delay<br>0 0 1: 1 IICϕ cycle<br>0 1 0: 2 IICϕ cycles<br>0 1 1: 3 IICϕ cycles<br>1 0 0: 4 IICϕ cycles<br>1 0 1: 5 IICϕ cycles<br>1 1 0: 6 IICϕ cycles<br>1 1 1: 7 IICϕ cycles<br>• When RIICnMR2.DLCS = 1 (IICϕ/2)<br>b6 b4<br>0 0 0: No output delay<br>0 0 1: 1 or 2 IICϕ cycles<br>0 1 0: 3 or 4 IICϕ cycles<br>0 1 1: 5 or 6 IICϕ cycles<br>1 0 0: 7 or 8 IICϕ cycles<br>1 0 1: 9 or 10 IICϕ cycles<br>1 1 0: 11 or 12 IICϕ cycles<br>1 1 1: 13 or 14 IICϕ cycles |
| 3 | — | 0 | R | Reserved<br>These bits are read as 0. The write value should be 0. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 2 | TMOH | 1 | R/W | Timeout H Count Control<br>0: Count is disabled while the SCL line is at a high level.<br>1: Count is enabled while the SCL line is at a high level. |
| 1 | TMOL | 1 | R/W | Timeout L Count Control<br>0: Count is disabled while the SCL line is at a low level.<br>1: Count is enabled while the SCL line is at a low level. |
| 0 | TMOS | 0 | R/W | Timeout Detection Time Selection<br>0: Long mode is selected.<br>1: Short mode is selected. |

Note 1. The setting DLCS = 1 (IICϕ/2) only becomes valid when SCL is at the low level. When SCL is at the high level, the setting DLCS = 1 becomes invalid and the clock source becomes the internal reference clock (IICϕ).

### TMOS Bit (Timeout Detection Time Selection)

This bit is used to select long mode or short mode for the timeout detection time when the timeout function is enabled (RIICnFER.TMOE bit = 1). When this bit is set to 0, long mode is selected. When this bit is set to 1, short mode is selected. In long mode, the timeout detection internal counter functions as a 16 bit-counter. In short mode, the counter functions as a 14 bit-counter. While the SCL line is in the state that enables this counter as specified by bits TMOH and TMOL, the counter counts up in synchronization with the internal reference clock (IICϕ) as a count source.

For details on the timeout function, see **Section 26.13.1, Timeout Function**.

### TMOL Bit (Timeout L Count Control)

This bit is used to enable or disable the internal counter of the timeout function to count up while the SCL line is held low when the timeout function is enabled (RIICnFER.TMOE bit = 1).

### TMOH Bit (Timeout H Count Control)

This bit is used to enable or disable the internal counter of the timeout function to count up while the SCL line is held high when the timeout function is enabled (RIICnFER.TMOE bit = 1).

### SDDL[2:0] Bits (SDA Output Delay Setup Counter)

The SDA output can be delayed by the SDDL[2:0] setting. This counter works with the clock source selected by the DLCS bit. The setting of this function can be used for all types of SDA output, including the transmission of the acknowledge bit.

For details on this function, see **Section 26.7, Facility for Delaying SDA Output**.

**CAUTION**

Set the SDA output delay time to meet the $I^2C$ bus standard (within the data enable time/acknowledge enable time*1) or the SMBus standard (within the data hold time: 300 [ns] or more, and SCL-clock low-level period - the data setup time: 250 [ns]). Note that, if a value outside the standard is set, communication with communication devices may malfunction or it may seemingly become a start condition or stop condition depending on the bus state.

**Note 1.** Data enable time/acknowledge enable time
3,450 [ns] (0 to 100 [kbps]: standard mode (Sm))
900 [ns] (0 to 400 [kbps]: fast mode (Fm))
450 [ns] (0 to 1 [Mbps]: fast mode plus (Fm+))

## 26.3.5 RIICnMR3 — I²C Bus Mode Register 3

**Access Size:** RIICnMR3 is a 32-bit readable/writable register.
RIICnMR3L and RIICnMR3H are 16-bit readable/writable registers.
RIICnMR3LL, RIICnMR3LH, RIICnMR3HL, and RIICnMR3HH are 8-bit readable/writable registers.

**Address(es):** RIICnMR3: <RIICn_base> + H'0010
RIICnMR3L: <RIICn_base> + H'0010, RIICnMR3H: <RIICn_base> + H'0012
RIICnMR3LL: <RIICn_base> + H'0010, RIICnMR3LH: <RIICn_base> + H'0011,
RIICnMR3HL: <RIICn_base> + H'0012, RIICnMR3HH: <RIICn_base> + H'0013

**Initial Value:** H'0000 0000 This register is initialized by any reset.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | SMBE | WAIT | RDRFS | ACKWP | ACKBT | ACKBR | NF[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 | W | R/W | R | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 8 | — | All 0 | R | Reserved<br>These bits are read as 0. The write value should be 0. |
| 7 | SMBE | 0 | R/W | SMBus/I²C Bus Selection<br>0: I²C bus is selected.<br>1: SMBus is selected. |
| 6 | WAIT*2 | 0 | R/W | WAIT<br>0: No WAIT<br>(The period between ninth clock cycle and first clock cycle is not held low.)<br>1: WAIT<br>(The period between ninth clock cycle and first clock cycle is held low.) Low-hold is released by reading RIICnDRR. |
| 5 | RDRFS*2 | 0 | R/W | RDRF Flag Set Timing Selection<br>0: The RDRF flag is set at the rising edge of the ninth SCL clock cycle.<br>(The SCL line is not held low at the falling edge of the eighth clock cycle.)<br>1: The RDRF flag is set at the rising edge of the eighth SCL clock cycle.<br>(The SCL line is held low at the falling edge of the eighth clock cycle.) Low-hold is released by writing a value to the ACKBT bit. |
| 4 | ACKWP*1 | 0 | W | ACKBT Write Protect<br>0: Modification of the ACKBT bit is disabled.<br>1: Modification of the ACKBT bit is enabled. |
| 3 | ACKBT*1 | 0 | R/W | Transmit Acknowledge<br>0: A 0 is sent as the acknowledge bit (ACK transmission).<br>1: A 1 is sent as the acknowledge bit (NACK transmission). |
| 2 | ACKBR | 0 | R | Receive Acknowledge<br>0: A 0 is received as the acknowledge bit (ACK reception).<br>1: A 1 is received as the acknowledge bit (NACK reception). |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 1, 0 | NF[1:0] | All 0 | R/W | Noise Filter Stage Selection<br>b1 b0<br>0 0: Noise of up to one IICϕ cycle is filtered out (single-stage filter).<br>0 1: Noise of up to two IICϕ cycles is filtered out (2-stage filter).<br>1 0: Noise of up to three IICϕ cycles is filtered out (3-stage filter).<br>1 1: Noise of up to four IICϕ cycles is filtered out (4-stage filter). |

Note 1. If it is attempted to write 1 to both ACKWP and ACKBT bits, the ACKBT bit cannot be set to 1.

Note 2. The WAIT and RDRFS bits are valid only in receive mode (invalid in transmit mode).

### NF[1:0] Bits (Noise Filter Stage Selection)

These bits are used to select the width of noise that can be removed from the signals input to RIICnSCL or RIICnSDA pin.

### ACKBR Bit (Receive Acknowledge)

This bit is used to store the acknowledge bit information received from the receive device in transmit mode.

[Setting condition]

When 1 is received as the acknowledge bit with the RIICnCR2.TRS bit set to 1

[Clearing conditions]

- When 0 is received as the acknowledge bit with the RIICnCR2.TRS bit set to 1
- When 1 is written to the RIICnCR1.IICRST bit while the RIICnCR1.ICE bit is 0 (RIIC reset)

### ACKBT Bit (Transmit Acknowledge)

This bit is used to set the bit to be sent at the acknowledge timing in receive mode.

[Setting condition]

When 1 is written to this bit with the ACKWP bit set to 1

[Clearing conditions]

- When 0 is written to this bit with the ACKWP bit set to 1
- When stop condition issuance is detected
- When 1 is written to the RIICnCR1.IICRST bit while the RIICnCR1.ICE bit is 0 (RIIC reset)

**CAUTION**

The ACKBT bit must be written to while the ACKWP bit is 1. If the ACKBT bit is written to with the ACKWP bit cleared to 0, writing to the ACKBT bit is disabled.

### ACKWP Bit (ACKBT Write Protect)

This bit is used to control the modification of the ACKBT bit.

### RDRFS Bit (RDRF Flag Set Timing Selection)

This bit is used to select the RDRF flag set timing in receive mode and also to select whether to hold the SCL line low at the falling edge of the eighth SCL clock cycle.

When the RDRFS bit is 0, the SCL line is not held low at the falling edge of the eighth SCL clock cycle, and the RDRF flag is set to 1 at the rising edge of the ninth SCL clock cycle.

When the RDRFS bit is 1, the RDRF flag is set to 1 at the rising edge of the eighth SCL clock cycle and the SCL line is held low at the falling edge of the eighth SCL clock cycle. The low-hold of the SCL line is released by writing a value to the ACKBT bit.

After data is received with this setting, the SCL line is automatically held low before the acknowledge bit is sent. This enables processing to send ACK (ACKBT = 0) or NACK (ACKBT = 1) according to receive data.

### WAIT Bit (WAIT)

This bit is used to control whether to hold the period between the ninth SCL clock cycle and the first SCL clock cycle low until the receive data buffer (RIICnDRR) is completely read each time single-byte data is received in receive mode. When the WAIT bit is 0, the receive operation is continued without holding the period between the ninth and the first SCL clock cycle low. When both the RDRFS and WAIT bits are 0, continuous receive operation is enabled with the double buffer.

When the WAIT bit is 1, the SCL line is held low from the falling edge of the ninth clock cycle until the RIICnDRR value is read each time single-byte data is received. This enables receive operation in byte units.

**CAUTION**

When the value of the WAIT bit is to be read, be sure to read the RIICnDRR beforehand.

### SMBE Bit (SMBus Select)

Setting this bit to 1 enables the RIICnSER.HOAE bit.

## 26.3.6 RIICnFER — I²C Bus Function Enable Register

**Access Size:** RIICnFER is a 32-bit readable/writable register.
RIICnFERL and RIICnFERH are 16-bit readable/writable registers.
RIICnFERLL, RIICnFERLH, RIICnFERHL, and RIICnFERHH are 8-bit readable/writable registers.

**Address(es):** RIICnFER: <RIICn_base> + H'0014
RIICnFERL: <RIICn_base> + H'0014, RIICnFERH: <RIICn_base> + H'0016
RIICnFERLL: <RIICn_base> + H'0014, RIICnFERLH: <RIICn_base> + H'0015,
RIICnFERHL: <RIICn_base> + H'0016, RIICnFERHH: <RIICn_base> + H'0017

**Initial Value:** H'0000 0072 This register is initialized by any reset.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | FMPE | SCLE | NFE | NACKE | SALE | NALE | MALE | TMOE |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 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<br>These bits are read as 0. The write value should be 0. |
| 7 | FMPE | 0 | R/W | Fast-mode Plus Enable<br>0: No fm+ slope control circuit is used for the SCLn pin and SDAn pin<br>1: An fm+ slope control circuit is used for the SCLn pin and SDAn pin |
| 6 | SCLE | 1 | R/W | SCL Synchronous Circuit Enable<br>0: No SCL synchronous circuit is used.<br>1: An SCL synchronous circuit is used. |
| 5 | NFE | 1 | R/W | Digital Noise Filter Circuit Enable<br>0: No digital noise filter circuit is used.<br>1: A digital noise filter circuit is used. |
| 4 | NACKE | 1 | R/W | NACK Reception Transfer Suspension Enable<br>0: Transfer operation is not suspended during NACK reception (transfer suspension disabled).<br>1: Transfer operation is suspended during NACK reception (transfer suspension enabled). |
| 3 | SALE | 0 | R/W | Slave Arbitration-Lost Detection Enable<br>0: Slave arbitration-lost detection is disabled.<br>1: Slave arbitration-lost detection is enabled. |
| 2 | NALE | 0 | R/W | NACK Transmission Arbitration-Lost Detection Enable<br>0: NACK transmission arbitration-lost detection is disabled.<br>1: NACK transmission arbitration-lost detection is enabled. |
| 1 | MALE | 1 | R/W | Master Arbitration-Lost Detection Enable<br>0: Master arbitration-lost detection is disabled.<br>(Disables the arbitration-lost detection function and does not clear the RIICnCR2.MST and TRS bits automatically when arbitration is lost.)<br>1: Master arbitration-lost detection is enabled.<br>(Enables the arbitration-lost detection function and clears the RIICnCR2.MST and TRS bits automatically when arbitration is lost.) |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 0 | TMOE | 0 | R/W | Timeout Function Enable<br>0: The timeout function is disabled.<br>1: The timeout function is enabled. |

### TMOE Bit (Timeout Function Enable)

This bit is used to enable or disable the timeout function.

For details on the timeout function, see **Section 26.13.1, Timeout Function**.

### MALE Bit (Master Arbitration-Lost Detection Enable)

This bit is used to specify whether to use the arbitration-lost detection function in master mode. Normally, set this bit to 1.

### NALE Bit (NACK Transmission Arbitration-Lost Detection Enable)

This bit is used to specify whether to cause arbitration to be lost when ACK is detected during transmission of NACK in receive mode (such as when slaves with the same address exist on the bus or when two or more masters select the same slave device simultaneously with different number of receive bytes).

### SALE Bit (Slave Arbitration-Lost Detection Enable)

This bit is used to specify whether to cause arbitration to be lost when a value different from the value being transmitted is detected on the bus in slave transmit mode (such as when slaves with the same address exist on the bus or when a mismatch with the transmit data occurs due to noise).

### NACKE Bit (NACK Reception Transfer Suspension Enable)

This bit is used to specify whether to continue or discontinue the transfer operation when NACK is received in transmit mode. Normally, set this bit to 1.

When NACK is received with the NACKE bit set to 1, the next transfer operation is suspended.

When the NACKE bit is 0, the next transfer operation is continued regardless of the received acknowledge content.

### SCLE Bit (SCL Synchronous Circuit Enable)

This bit is used to specify whether to synchronize the SCL clock with the SCL input clock. Normally, set this bit to 1. When the SCLE bit is cleared to 0 (SCL synchronous circuit not used), the RIIC does not synchronize the SCL clock with the SCL input clock. In this setting, the RIIC outputs the SCL clock with the transfer rate set in RIICnBRH and RIICnBRL regardless of the SCL line state. For this reason, if the bus load of the $I^2C$ bus line is much larger than the specification value or if the SCL clock output overlaps in multiple masters, the short-cycle SCL clock that does not meet the specification may be output. When no SCL synchronous circuit is used, it also affects the issuance of a start condition, restart condition, and stop condition, and the continuous output of extra SCL clock cycles.

This bit must not be cleared to 0 except for checking the output of the transfer rate.

### FMPE Bit (Fast-mode Plus Enable)

This bit is used to specify whether to use a slope control circuit for Fast-mode Plus [fm+].

When this bit is set to 1, a slope control circuit conforming to the Fast-mode Plus [fm+] slope control standard (tof) of the $I^2C$ bus is selected. When this bit is cleared to 0, a slope control circuit conforming to the Standard-mode [Sm] and Fast- mode [fm] slope control standard (tof) of the $I^2C$ bus is selected.

Set this bit to 1 when using the transmission rate within a range up to 1 Mbps (Fast-mode Plus[fm+]) of the $I^2C$ bus standard. Clear this bit to 0 when using the transmission rate at other rates (up to 100 kbps[Sm], up to 400 kbps[fm]) or for SMBus (10 to 100 kbps).

## 26.3.7 RIICnSER — I²C Bus Status Enable Register

**Access Size:** RIICnSER is a 32-bit readable/writable register.
RIICnSERL and RIICnSERH are 16-bit readable/writable registers.
RIICnSERLL, RIICnSERLH, RRIICnSERHL, and RIICnSERHH are 8-bit readable/writable registers.

**Address(es):** RIICnSER: <RIICn_base> + H'0018
RIICnSERL: <RIICn_base> + H'0018, RIICnSERH: <RIICn_base> + H'001A
RIICnSERLL: <RIICn_base> + H'0018, RIICnSERLH: <RIICn_base> + H'0019,
RIICnSERHL: <RIICn_base> + H'001A, RIICnSERHH: <RIICn_base> + H'001B

**Initial Value:** H'0000 0009 This register is initialized by any reset.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | HOAE | — | DIDE | — | GCE | SAR2 | SAR1 | SAR0 |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 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 |
|---|---|---|---|---|
| 31 to 8 | — | All 0 | R | Reserved<br>These bits are read as 0. The write value should be 0. |
| 7 | HOAE | 0 | R/W | Host Address Enable<br>0: Host address detection is disabled.<br>1: Host address detection is enabled. |
| 6 | — | 0 | R | Reserved<br>This bit is read as 0. The write value should be 0. |
| 5 | DIDE | 0 | R/W | Device-ID Address Detection Enable<br>0: Device-ID address detection is disabled.<br>1: Device-ID address detection is enabled. |
| 4 | — | 0 | R | Reserved<br>This bit is read as 0. The write value should be 0. |
| 3 | GCE | 1 | R/W | General Call Address Enable<br>0: General call address detection is disabled.<br>1: General call address detection is enabled. |
| 2 | SAR2 | 0 | R/W | Slave Address Register 2 Enable<br>0: Slave address in RIICnSAR2 is disabled.<br>1: Slave address in RIICnSAR2 is enabled. |
| 1 | SAR1 | 0 | R/W | Slave Address Register 1 Enable<br>0: Slave address in RIICnSAR1 is disabled.<br>1: Slave address in RIICnSAR1 is enabled. |
| 0 | SAR0 | 1 | R/W | Slave Address Register 0 Enable<br>0: Slave address in RIICnSAR0 is disabled.<br>1: Slave address in RIICnSAR0 is enabled. |

#### SARy Bit (Slave Address Register y Enable) (y = 0 to 2)

This bit is used to enable or disable the received slave address and the slave address set in RIICnSARy.

When this bit is set to 1, the slave address set in RIICnSARy is enabled and is compared with the received slave address. When this bit is cleared to 0, the slave address set in RIICnSARy is disabled and is ignored even if it matches the received slave address.

#### GCE Bit (General Call Address Enable)

This bit is used to specify whether to ignore the general call address (0000 000b + 0 [W]: All 0) when it is received. When this bit is set to 1, if the received slave address matches the general call address, the RIIC recognizes the received slave address as the general call address independently of the slave addresses set in RIICnSARy (y = 0 to 2) and performs data receive operation.

When this bit is cleared to 0, the received slave address is ignored even if it matches the general call address.

#### DIDE Bit (Device-ID Address Detection Enable)

This bit is used to specify whether to recognize and execute the Device-ID address when a device ID (1111 100b) is received in the first frame after a start condition or restart condition is detected.

When this bit is set to 1, if the received first frame matches the device ID, the RIIC recognizes that the Device-ID address has been received. When the following R/W# bit is 0 [W], the RIIC recognizes the second and the following frames as slave addresses and continues the receive operation.

When this bit is cleared to 0, the RIIC ignores the received first frame even if it matches the device ID address and recognizes the first frame as a normal slave address.

For details on the device-ID address detection, see **Section 26.9.3, Device-ID Address Detection**.

#### HOAE Bit (Host Address Enable)

This bit is used to specify whether to ignore received host address (0001 000b) when the RIICnMR3.SMBS bit is 1. When this bit is set to 1 while the RIICnMR3.SMBS bit is 1, if the received slave address matches the host address, the RIIC recognizes the received slave address as the host address independently of the slave addresses set in RIICnSARy (y = 0 to 2) and performs the receive operation.

When the RIICnMR3.SMBS bit or the HOAE bit is cleared to 0, the received slave address is ignored even if it matches the host address.

## 26.3.8 RIICnIER — I²C Bus Interrupt Enable Register

**Access Size:** RIICnIER is a 32-bit readable/writable register.
RIICnIERL and RIICnIERH are 16-bit readable/writable registers.
RIICnIERLL, RIICnIERLH, RIICnIERHL, and RIICnIERHH are 8-bit readable/writable registers.

**Address(es):** RIICnIER: <RIICn_base> + H'001C
RIICnIERL: <RIICn_base> + H'001C, RIICnIERH: <RIICn_base> + H'001E
RIICnIERLL: <RIICn_base> + H'001C, RIICnIERLH: <RIICn_base> + H'001D,
RIICnIERHL: <RIICn_base> + H'001E, RIICnIERHH: <RIICn_base> + H'001F

**Initial Value:** H'0000 0000 This register is initialized by any reset.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | TIE | TEIE | RIE | NAKIE | SPIE | STIE | ALIE | TMOIE |
| 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<br>These bits are read as 0. The write value should be 0. |
| 7 | TIE | 0 | R/W | Transmit Data Empty Interrupt Enable<br>0: Transmit data empty interrupt request (INTRIICTI) is disabled.<br>1: Transmit data empty interrupt request (INTRIICTI) is enabled. |
| 6 | TEIE | 0 | R/W | Transmit End Interrupt Enable<br>0: Transmit end interrupt request (INTRIICTEI) is disabled.<br>1: Transmit end interrupt request (INTRIICTEI) is enabled. |
| 5 | RIE | 0 | R/W | Receive Data Full Interrupt Enable<br>0: Receive data full interrupt request (INTRIICRI) is disabled.<br>1: Receive data full interrupt request (INTRIICRI) is enabled. |
| 4 | NAKIE | 0 | R/W | NACK Reception Interrupt Enable<br>0: NACK reception interrupt request (INTRIICNAKI) is disabled.<br>1: NACK reception interrupt request (INTRIICNAKI) is enabled. |
| 3 | SPIE | 0 | R/W | Stop Condition Detection Interrupt Enable<br>0: Stop condition detection interrupt request (INTRIICSPI) is disabled.<br>1: Stop condition detection interrupt request (INTRIICSPI) is enabled. |
| 2 | STIE | 0 | R/W | Start Condition Detection Interrupt Enable<br>0: Start condition detection interrupt request (INTRIICSTI) is disabled.<br>1: Start condition detection interrupt request (INTRIICSTI) is enabled. |
| 1 | ALIE | 0 | R/W | Arbitration-Lost Interrupt Enable<br>0: Arbitration-lost interrupt request (INTRIICALI) is disabled.<br>1: Arbitration-lost interrupt request (INTRIICALI) is enabled. |
| 0 | TMOIE | 0 | R/W | Timeout Interrupt Enable<br>0: Timeout interrupt request (INTRIICTMOI) is disabled.<br>1: Timeout interrupt request (INTRIICTMOI) is enabled. |

#### TMOIE Bit (Timeout Interrupt Enable)

This bit is used to enable or disable timeout interrupt requests (INTRIICTMOI) when the RIICnSR2.TMOF flag is set to 1. An INTRIICTMOI interrupt request is canceled by clearing the TMOF flag or the TMOIE bit to 0.

#### ALIE Bit (Arbitration-Lost Interrupt Enable)

This bit is used to enable or disable arbitration-lost interrupt requests (INTRIICALII) when the RIICnSR2.AL flag is set to 1. An INTRIICALII interrupt request is canceled by clearing the AL flag or the ALIE bit to 0.

#### STIE Bit (Start Condition Detection Interrupt Enable)

This bit is used to enable or disable start condition detection interrupt requests (INTRIICSTI) when the RIICnSR2.START flag is set to 1. An INTRIICSTI interrupt request is canceled by clearing the START flag or the STIE bit to 0.

#### SPIE Bit (Stop Condition Detection Interrupt Enable)

This bit is used to enable or disable stop condition detection interrupt requests (INTRIICSPI) when the RIICnSR2.STOP flag is set to 1. An INTRIICSPI interrupt request is canceled by clearing the STOP flag or the SPIE bit to 0.

#### NAKIE Bit (NACK Reception Interrupt Enable)

This bit is used to enable or disable NACK reception interrupt requests (INTRIICNAKI) when the RIICnSR2.NACKF flag is set to 1. An INTRIICNAKI interrupt request is canceled by clearing the NACKF flag or the NAKIE bit to 0.

#### RIE Bit (Receive Data Full Interrupt Enable)

This bit is used to enable or disable receive data full interrupt requests (INTRIICRI) when the RIICnSR2.RDRF flag in ICSR2 is set to 1. An INTRIICRI interrupt request is canceled by clearing the RDRF flag or the RIE bit to 0.

#### TEIE Bit (Transmit End Interrupt Enable)

This bit is used to enable or disable transmit end interrupts (INTRIICTEI) when the RIICnSR2.TEND flag is set to 1. An INTRIICTEI interrupt request is canceled by clearing the TEND flag or the TEIE bit to 0.

#### TIE Bit (Transmit Data Empty Interrupt Enable)

This bit is used to enable or disable transmit data empty interrupts (INTRIICTI) when the RIICnSR2.TDRE flag is set to 1.

## 26.3.9 RIICnSR1 — I²C Bus Status Register 1

**Access Size:** RIICnSR1 is a 32-bit readable/writable register.
RIICnSR1L and RIICnSR1H are 16-bit readable/writable registers.
RIICnSR1LL, RIICnSR1LH, RIICnSR1HL, and RIICnSR1HH are 8/1-bit readable/writable registers.

**Address(es):** RIICnSR1: <RIICn_base> + H'0020
RIICnSR1L: <RIICn_base> + H'0020, RIICnSR1H: <RIICn_base> + H'0022
RIICnSR1LL: <RIICn_base> + H'0020, RIICnSR1LH: <RIICn_base> + H'0021,
RIICnSR1HL: <RIICn_base> + H'0022, RIICnSR1HH: <RIICn_base> + H'0023

**Initial Value:** H'0000 0000 This register is initialized by any reset.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | HOA | — | DID | — | — | — | — | — |
| 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)*1 | R | R(/W)*1 | R | R/W*1 | R(/W)*1 | R(/W)*1 | R(/W)*1 |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 8 | — | All 0 | R | Reserved<br>These bits are read as 0. The write value should be 0. |
| 7 | HOA | 0 | R(/W) *1 | Host Address Detection Flag<br>0: Host address is not detected.<br>1: Host address is detected. |
| 6 | — | 0 | R | Reserved<br>This bit is read as 0. The write value should be 0. |
| 5 | DID | 0 | R(/W) *1 | Device-ID Address Detection Flag<br>0: Device-ID command is not detected.<br>1: Device-ID command is detected. |
| 4 | — | 0 | R | Reserved<br>This bit is read as 0. The write value should be 0. |
| 3 | GCA | 0 | R/W *1 | General Call Address Detection Flag<br>0: General call address is not detected.<br>1: General call address is detected. |
| 2 | AAS2 | 0 | R(/W) *1 | Slave Address 2 Detection Flag<br>0: Slave address 2 is not detected.<br>1: Slave address 2 is detected. |
| 1 | AAS1 | 0 | R(/W) *1 | Slave Address 1 Detection Flag<br>0: Slave address 1 is not detected.<br>1: Slave address 1 is detected. |
| 0 | AAS0 | 0 | R(/W) *1 | Slave Address 0 Detection Flag<br>0: Slave address 0 is not detected.<br>1: Slave address 0 is detected. |

Note 1. Only 0 can be written to this bit.

### AASy Flag (Slave Address y Detection) (y = 0 to 2)

[Setting conditions]

<For 7-bit address format: RIICnSARy.FSy = 0>

When the received slave address matches the RIICnSARy.SVA[7:1] value with the RIICnSER.SARy bit set to 1 (slave address y detection enabled)

This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the frame.

<For 10-bit address format: RIICnSARy.FSy = 1>

When the received slave address matches a value of (1111 0b + RIICnSARy.SVA[9:8]) and the following address matches the RIICnSARy.SVA[7:0] value with the RIICnSER.SARy bit set to 1 (slave address y detection enabled)
This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the AASy bit after reading AASy = 1
- When a stop condition is detected
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

<For 7-bit address format: RIICnSARy.FSy = 0>

- When the received slave address does not match the RIICnSARy.SVA[7:1] value with the RIICnSER.SARy bit set to 1 (slave address y detection enabled)
  This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.

<For 10-bit address format: RIICnSARy.FSy = 1>

- When the received slave address does not match a value of (1111 0b + RIICnSARy.SVA[9:8]) with the RIICnSER.SARy bit set to 1 (slave address y detection enabled)
  This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.
- When the received slave address matches a value of (1111 0b + RIICnSARy.SVA[9:8]) and the following address does not match the RIICnSARy.SVA[7:0] value with the RIICnSER.SARy bit set to 1 (slave address y detection enabled)
  This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.

### GCA Flag (General Call Address Detection)

[Setting condition]

When the received slave address matches the general call address (0000 000b + 0 [W]) with the RIICnSER.GCE bit set to 1 (general call address detection enabled)

This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the GCA bit after reading GCA = 1
- When a stop condition is detected
- When the received slave address does not match the general call address (0000 000b + 0 [W]) with the RIICnSER.GCE bit set to 1 (general call address detection enabled)
  This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

### DID Flag (Device-ID Address Detection)

[Setting condition]

- When the first frame received immediately after a start condition or restart condition is detected matches a value of (device ID (1111 100b) + 0 [W]) with the RIICnSER.DIDE bit set to 1 (Device-ID address detection enabled) This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the frame.
- When a restart condition is detected after a match with the device ID address and the device ID address (1111 100b) + 1 [R] has matched with the RIICnSER.DIDE bit set to 1 (Device-ID address detection enabled)
  This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the DID bit after reading DID = 1
- When a stop condition is detected
- When the first frame received immediately after a start condition or restart condition is detected does not match a value of (device ID (1111 100b)) with the RIICnSER.DIDE bit set to 1 (Device-ID address detection enabled) This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.
- When the first frame received immediately after a start condition or restart condition is detected matches a value of (device ID (1111 100b) + 0 [W]) and the second frame does not match any of slave addresses 0 to 2 with the RIICnSER.DIDE bit set to 1 (Device-ID address detection enabled)
  This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

### HOA Flag (Host Address Detection)

[Setting condition]

When the received slave address matches the host address (0001 000b) while the RIICnMR3.SMBE bit and RIICnSER.HOAE bit are set to 1 (host address detection enabled)

This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the HOA bit after reading HOA = 1
- When a stop condition is detected
- When 0 is written to the RIICnMR3.SMBS bit in ICMR3 or the RIICnSER.HOAE bit
- When the received slave address does not match the host address (0001 000b) with the RIICnSER.HOAE bit set to 1 (host address detection enabled)
  This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.
- When 1 is written to the RIICnCR1.IICRST bit 1 to apply an RIIC reset or an internal reset

## 26.3.10 RIICnSR2 — I²C Bus Status Register 2

**Access Size:** RIICnSR2 is a 32-bit readable/writable register.
RIICnSR2L and RIICnSR2H are 16-bit readable/writable registers.
RIICnSR2LL, RIICnSR2LH, RIICnSR2HL, and RIICnSR2HH are 8/1-bit readable/writable registers.

**Address(es):** RIICnSR2: <RIICn_base> + H'0024
RIICnSR2L: <RIICn_base> + H'0024, RIICnSR2H: <RIICn_base> + H'0026
RIICnSR2LL: <RIICn_base> + H'0024, RIICnSR2LH: <RIICn_base> + H'0025,
RIICnSR2HL: <RIICn_base> + H'0026, RIICnSR2HH: <RIICn_base> + H'0027

**Initial Value:** H'0000 0000 This register is initialized by any reset.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | TDRE | TEND | RDRF | NACKF | STOP | STAT | AL | TMOF |
| 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(/W)*1 | R(/W)*1 | R(/W)*1 | R(/W)*1 | R(/W)*1 | R(/W)*1 | R(/W)*1 |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 8 | — | All 0 | R | Reserved<br>These bits are read as 0. The write value should be 0. |
| 7 | TDRE | 0 | R | Transmit Data Empty Flag<br>0: RIICnDRT contains transmit data.<br>1: RIICnDRT contains no transmit data. |
| 6 | TEND | 0 | R(/W) *1 | Transmit End Flag<br>0: Data is being transmitted.<br>1: Data has been transmitted. |
| 5 | RDRF | 0 | R(/W) *1 | Receive Data Full Flag<br>0: RIICnDRR contains no receive data.<br>1: RIICnDRR contains receive data. |
| 4 | NACKF | 0 | R(/W) *1 | NACK Reception Flag<br>0: NACK is not received.<br>1: NACK is received. |
| 3 | STOP | 0 | R(/W) *1 | Stop Condition Detection Flag<br>0: Stop condition is not detected.<br>1: Stop condition is detected. |
| 2 | START | 0 | R(/W) *1 | Start Condition Detection Flag<br>0: Start condition is not detected.<br>1: Start condition is detected. |
| 1 | AL | 0 | R(/W) *1 | Arbitration-Lost Flag<br>0: Arbitration is not lost.<br>1: Arbitration is lost. |
| 0 | TMOF | 0 | R(/W) *1 | Timeout Flag<br>0: No timeout has occurred.<br>1: Timeout has occurred. |

Note 1. Only 0 can be written to this bit.

### TMOF Flag (Timeout)

This flag is set to 1 when the RIIC recognizes timeout after the SCL line state remains unchanged for a certain period.
[Setting condition]

The timeout function is enabled when the RIICnFER.TMOE bit is 1. It detects an abnormal bus state that the SCL line is held low or high during the following conditions:

- The bus is busy (RIICnCR2.BBSY = 1) in master mode (RIICnCR2.MST = 1).
- The slave address matches that of this module (RIICnSR1 register is not H'00) and the bus is busy (RIICnCR2.BBSY = 1) in slave mode (RIICnCR2.MST = 0).
- Issuing of a start condition is being requested (RIICnCR2.ST = 1) and the bus is free (RIICnCR2.BBSY = 0).

[Clearing conditions]

- When 0 is written to the TMOF bit after reading TMOF = 1
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

### AL Flag (Arbitration-Lost)

This flag shows that bus mastership has been lost (loss in arbitration) due to a bus conflict or some other reason when a start condition is issued or an address and data are transmitted. The RIIC monitors the level on the SDA line during transmission and, if the level on the line does not match the value of the bit being output, sets the value of the AL bit to 1 to indicate that the bus is occupied by another device.

The RIIC can also set the flag to indicate the detection of loss of arbitration during NACK transmission in receive mode or during data transmission in slave mode.

[Setting conditions]

<When master arbitration-lost detection is enabled: RIICnFER.MALE = 1>

- When the internal SDA output state does not match the SDA line level at the rising edge of SCL clock except for the ACK period during data (including slave address) transmission in master transmit mode (when the SDA line is driven low while the internal SDA output is at a high level (the SDA pin is in the high-impedance state))
- When a start condition is detected while the RIICnCR2.ST bit is 1 (start condition issuance request) or the internal SDA output state does not match the SDA line level
- When the RIICnCR2.ST bit is set to 1 (start condition issuance request) with the RIICnCR2.BBSY flag set to 1.

<When NACK arbitration-lost detection is enabled: RIICnFER.NALE = 1>

When the internal SDA output state does not match the SDA line level at the rising edge of SCL clock in the ACK period during NACK transmission in receive mode

<When slave arbitration-lost detection is enabled: RIICnFER.SALE = 1>

When the internal SDA output state does not match the SDA line level at the rising edge of SCL clock except for the ACK period during data transmission in slave transmit mode

[Clearing conditions]

- When 0 is written to the AL bit after reading AL = 1
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

Table 26.8 Relationship between Arbitration-Lost Generation Sources and Arbitration-Lost Enable Functions

| RIICnFER | | | RIICnSR2 | | |
|---|---|---|---|---|---|
| MALE | NALE | SALE | AL | Error | Arbitration-Lost Generation Source |
| 1 | x | x | 1 | Start condition issuance error | When internal SDA output state does not match SDA line level when a start condition is detected while the RIICnCR2.ST bit is 1 |
| | | | | | When RIICnCR2.ST is set to 1 with RIICnCR2.BBSY set to 1 |
| | | | 1 | Transmit data mismatch | When transmit data (including slave address) does not match the bus state in master transmit mode |
| x | 1 | x | 1 | NACK transmission mismatch | When ACK is detected during transmission of NACK in master receive mode or slave receive mode |
| x | x | 1 | 1 | Transmit data mismatch | When transmit data does not match the bus state in slave transmit mode |

**Note:** x: Don't care

### START Flag (Start Condition Detection)

[Setting condition]

When a start condition (or a restart condition) is detected

[Clearing conditions]

- When 0 is written to the START bit after reading START = 1
- When a stop condition is detected
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

### STOP Flag (Stop Condition Detection)

[Setting condition]

When a stop condition is detected

[Clearing conditions]

- When 0 is written to the STOP bit after reading STOP = 1
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

### NACKF Flag (NACK Reception)

[Setting condition]

When acknowledge is not received (NACK is received) from the receive device in transmit mode with the RIICnFER.NACKE bit set to 1 (transfer suspension enabled)

[Clearing conditions]

- When 0 is written to the NACKF bit after reading NACKF = 1
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

**CAUTION**

When the NACKF flag is set to 1, the RIIC suspends data transmission/reception. Writing to RIICnDRT in transmit mode or reading from RIICnDRR in receive mode with the NACKF flag set to 1 does not enable data transmit/ receive operation. To restart data transmission/reception, clear the NACKF flag to 0.

RDRF Flag (Receive Data Full)

[Setting conditions]

- Slave receive mode
  - When the received slave address matches and the RIICnCR2.TRS bit is cleared to 0 after a start condition (or a restart condition) is detected
  - At the rising edge of the eighth or ninth SCL clock cycle (selected by the RIICnMR3.RDRFS bit) after receive data is transferred from RIICnDRS to RIICnDRR
- Master receive mode
  - When the slave address and the data direction are transmitted and the receive mode is entered (the RIICnCR2.TRS bit is set to 1) after a start condition (or a restart condition) is issued
  - At the rising edge of the eighth or ninth SCL clock cycle (selected by the RIICnMR3.RDRFS bit) after receive data is transferred from RIICnDRS to RIICnDRR

[Clearing conditions]

- When 0 is written to the RDRF bit after reading RDRF = 1
- When data is read from RIICnDRR
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

#### TEND Flag (Transmit End)

[Setting condition]

At the rising edge of the ninth SCL clock cycle while the TDRE flag is 1

[Clearing conditions]

- When 0 is written to the TEND bit after reading TEND = 1
- When data is written to RIICnDRT
- When a stop condition is detected
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

#### TDRE Flag (Transmit Data Empty)

[Setting conditions]

- When data has been transferred from RIICnDRT to RIICnDRS and RIICnDRT becomes empty
- When the RIICnCR2.TRS bit is set to 1
  - When the RIICnCR2.MST bit is set to 1 after a start condition (or a restart condition) is detected
  - When the RIIC enters transmit mode from receive mode
- When the received slave address matches while the TRS bit is 1

[Clearing conditions]

- When data is written to RIICnDRT
- When the RIICnCR2.TRS bit is cleared to 0
  - When a stop condition is detected
  - When the RIIC enters receive mode from transmit mode
- When 1 is written to the RIICnCR1.IICRST bit to apply an RIIC reset or an internal reset

**CAUTION**

When the NACKF flag is set to 1 while the RIICnFER.NACKE bit is 1, the RIIC suspends data transmission/ reception. Here, if the TDRE flag is 0 (next transmit data has been written), data is transferred to the RIICnDRS register and the RIICnDRT register becomes empty at the rising edge of the ninth clock cycle, but the TDRE flag is not set to 1.

## 26.3.11 RIICnSARy — I²C Slave Address Register y (y = 0 to 2)

**Access Size:** RIICnSARy is a 32-bit readable/writable register.
RIICnSARyL and RIICnSARyH are 16-bit readable/writable registers.
RIICnSARyLL, RIICnSARyLH, RIICnSARyHL, and RIICnSARyHH are 8-bit readable/writable registers.

**Address(es):** RIICnSAR0: <RIICn_base> + H'0028
RIICnSAR0L: <RIICn_base> + H'0028, RIICnSAR0H: <RIICn_base> + H'002A
RIICnSAR0LL: <RIICn_base> + H'0028, RIICnSAR0LH: <RIICn_base> + H'0029,
RIICnSAR0HL: <RIICn_base> + H'002A, RIICnSAR0HH: <RIICn_base> + H'002B
RIICnSAR1: <RIICn_base> + H'002C
RIICnSAR1L: <RIICn_base> + H'002C, RIICnSAR1H: <RIICn_base> + H'002E
RIICnSAR1LL: <RIICn_base> + H'002C, RIICnSAR1LH: <RIICn_base> + H'002D,
RIICnSAR1HL: <RIICn_base> + H'002E, RIICnSAR1HH: <RIICn_base> + H'002F
RIICnSAR2: <RIICn_base> + H'0030
RIICnSAR2L: <RIICn_base> + H'0030, RIICnSAR2H: <RIICn_base> + H'0032
RIICnSAR2LL: <RIICn_base> + H'0030, RIICnSAR2LH: <RIICn_base> + H'0031,
RIICnSAR2HL: <RIICn_base> + H'0032, RIICnSAR2HH: <RIICn_base> + H'0033

**Initial Value:** H'0000 0000 This register is initialized by any reset.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | FSy | — | — | — | — | — | SVA[9:1] | | | | | | | | | SVA0 |
| 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 |
|---|---|---|---|---|
| 31 to 16 | — | All 0 | R | Reserved<br>These bits are read as 0. The write value should be 0. |
| 15 | FSy | 0 | R/W | 7-Bit/10-Bit Address Format Selection<br>0: The 7-bit address format is selected.<br>1: The 10-bit address format is selected. |
| 14 to 10 | — | All 0 | R | Reserved<br>These bits are read as 0. The write value should be 0. |
| 9 to 1 | SVA[9:1] | All 0 | R/W | 7-Bit Address/10-Bit Address Upper Bits A slave address is set.<br>• When the FSy bit is 0 (7-bit address format), the SVA[7:1] bits are Valid and form a 7-bit slave address.<br>• When the FSy bit is 1 (10-bit address format), SVA[9:1] bits form a 10-bit slave address (combined with the SVA0 bit). |
| 0 | SVA0 | 0 | R/W | 10-Bit Address LSB<br>The least significant bit (LSB) of a 10-bit slave address is set.<br>• When the FSy bit is 0 (7-bit address format), this bit is invalid.<br>• When the FSy bit is 1 (10-bit address format), this bit is a 10-bit slave address (combined with the SVA[9:1] bits). |

#### SVA0 Bit (10-Bit Address LSB)

When the 10-bit address format is selected (RIICnSARy.FSy = 1), this bit functions as the LSB of a 10-bit address and forms a 10-bit address in combination with the SVA[9:1] bits.

When the RIICnSER.SARy bit is set to 1 (RIICnSARy enabled) and the RIICnSARy.FSy bit is 1, this bit is valid. While the RIICnSARy.FSy bit or SARy bit is 0, the setting of this bit is ignored.

#### SVA[9:1] Bits (7-Bit Address/10-Bit Address Upper Bits)

When the 7-bit address format is selected (RIICnSARy.FSy = 0), these bits function as a 7-bit address. When the 10-bit address format is selected (RIICnSARy.FSy = 1), these bits function as a 10-bit address in combination with the SVA0 bit.

While the RIICnSER.SARy bit is 0, the setting of these bits is ignored.

#### FSy Bit (7-Bit/10-Bit Address Format Selection)

This bit is used to select 7-bit address or 10-bit address for slave address y (in RIICnSARy).

When the RIICnSER.SARy bit is set to 1 (RIICnSARy enabled) and the RIICnSARy.FSy bit is 0, the 7-bit address format is selected for slave address y, the RIICnSARy.SVA[7:1] setting is valid, and the settings of the SVA[9:8] bits and the RIICnSARy.SVA0 bit are ignored.

When the RIICnSER.SARy bit is set to 1 (RIICnSARy enabled) and the RIICnSARy.FSy bit is 1, the 10-bit address format is selected for slave address y and the settings of the SVA[9:1] bits and the SVA0 bit are valid.

While the RIICnSER.SARy bit is 0 (RIICnSARy disabled), the setting of the RIICnSARy.FSy bit is invalid.

## 26.3.12 RIICnBRL — I²C Bus Bit Rate Low-Level Register

**Access Size:** RIICnBRL is a 32-bit readable/writable register.
RIICnBRLL and RIICnBRLH are 16-bit readable/writable registers.
RIICnBRLLL, RIICnBRLLH, RIICnBRLHL, and RIICnBRLHH are 8-bit readable/writable registers.

**Address(es):** RIICnBRL: <RIICn_base> + H'0034
RIICnBRLL: <RIICn_base> + H'0034, RIICnBRLH: <RIICn_base> + H'0036
RIICnBRLLL: <RIICn_base> + H'0034, RIICnBRLLH: <RIICn_base> + H'0035,
RIICnBRLHL: <RIICn_base> + H'0036, RIICnBRLHH: <RIICn_base> + H'0037

**Initial Value:** H'0000 00FF This register is initialized by any reset.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | BRL[4:0] | | | | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 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 |
|---|---|---|---|---|
| 31 to 8 | — | All 0 | R | Reserved<br>These bits are read as 0. The write value should be 0. |
| 7 to 5 | — | All 1 | R | Reserved<br>These bits are read as 1. The write value should be 1. |
| 4 to 0 | BRL[4:0] | All 1 | R/W | Bit Rate Low-Level Period Low-level period of SCL clock |

The RIICnBRL register is a 5-bit register that is used to set the width at low level for the SCL clock.

It also works to generate the data setup time for automatic SCL low-hold operation (see **Section 26.10, Automatically Low-Hold Function for SCL**); when the RIIC is used only in slave mode, this register needs to be set to a value at least the same as the data setup time*[1].

RIICnBRL counts the low-level period with the internal reference clock source (IICϕ) specified by the RIICnMR1.CKS[2:0] bits.

**Note 1.** Data setup time (tSU: DAT)
250 [ns] (0 to 100 [kbps]: standard mode (Sm))
100 [ns] (0 to 400 [kbps]: fast mode (Fm))
50 [ns] (0 to 1 [Mbps]: fast mode plus (Fm+))

## 26.3.13 RIICnBRH — I²C Bus Bit Rate High-Level Register

**Access Size:** RIICnBRH is a 32-bit readable/writable register.
RIICnBRHL and RIICnBRHH are 16-bit readable/writable registers.
RIICnBRHLL, RIICnBRHLH, RIICnBRHHL, and RIICnBRHHH are 8-bit readable/writable registers.

**Address(es):** RIICnBRH: <RIICn_base> + H'0038
RIICnBRHL: <RIICn_base> + H'0038, RIICnBRHH: <RIICn_base> + H'003A
RIICnBRHLL: <RIICn_base> + H'0038, RIICnBRHLH: <RIICn_base> + H'0039,
RIICnBRHHL: <RIICn_base> + H'003A, RIICn- BRHHH:<RIICn_base> + H'003B

**Initial Value:** H'0000 00FF This register is initialized by any reset.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | BRH[4:0] | | | | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 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 |
|---|---|---|---|---|
| 31 to 8 | — | All 0 | R | Reserved<br>These bits are read as 0. The write value should be 0. |
| 7 to 5 | — | All 1 | R | Reserved<br>These bits are read as 1. The write value should be 1. |
| 4 to 0 | BRH[4:0] | All 1 | R/W | Bit Rate High-Level Period High-level period of SCL clock |

RIICnBRH is a 5-bit register to set the high-level period of SCL clock. RIICnBRH is valid in master mode. If the RIIC is used only in slave mode, this register need not to set the high-level period.

RIICnBRH counts the high-level period with the internal reference clock source (IICϕ) specified by the RIICnMR1.CKS[2:0] bits in ICMR1.

The frequency and duty cycle are calculated using one of the following expressions (1) to (5) according to the register settings.

**CAUTION**

The minimum value that can be specified in RIICnBRL and RIICnBRH is determined according to the values of the SCLE and NFE bits in RIICnFER and the NF bit in RIICnMR3. For details of the minimum specifiable value, see **Table 26.9**.

(1) When SCLE = 0
Frequency = 1 / {[ (BRH + 1) + (BRL + 1)] / IICϕ + tr + tf}
Duty cycle = {tr + (BRH + 1) / IICϕ} / {tr + tf + [(BRH + 1) + (BRL + 1)] / IICϕ}

(2) When SCLE = 1, NFE = 0, CKS = 000 (IICϕ = P0ϕ) Frequency = 1 / {[(BRH + 3) + (BRL + 3)] / IICϕ + tr + tf}
Duty cycle = {tr + (BRH + 3) / IICϕ} / {tr + tf + [(BRH + 3) + (BRL + 3) ] / IICϕ}

(3) When SCLE = 1, NFE = 1, CKS = 000 (IICϕ = P0ϕ)
Frequency = 1 / {[(BRH + 3 + nf) + (BRL + 3 + nf)] / IICϕ + tr + tf}
Duty cycle = {tr + (BRH + 3 + nf) / IICϕ} / {tr + tf + [(BRH + 3 + nf) + (BRL + 3 + nf)] / IICϕ}

(4) When SCLE = 1, NFE = 0, CKS [1] 000 (IICϕ < P0ϕ)
Frequency = 1 / {[(BRH + 2) + (BRL + 2)] / IICϕ + tr + tf}
Duty cycle = {tr + (BRH + 2) / IICϕ} / {tr + tf + [(BRH + 2) + (BRL + 2) ] / IICϕ}

(5) When SCLE = 1, NFE = 1, CKS [1] 000 (IICϕ < P0ϕ)
Frequency = 1 / {[(BRH + 2 + nf) + (BRL + 2 + nf)] / IICϕ + tr + tf}
Duty cycle = {tr + (BRH + 2 + nf) / IICϕ} / {tr + tf + [(BRH + 2 + nf) + (BRL + 2 + nf) ] / IICϕ}

Symbols in the expressions
SCLE: RIICnFER.SCLE bit
BRH: RIICnBRH.BRH[4:0] bits
BRL: RIICnBRL.BRL[4:0] bits
CKS: RIICnMR1.CKS bits
NFE: RIICnFER.NFE bit
IICϕ: Internal reference clock selected by the CKS bits
tf: SCL signal falling time [s]*[1]
tr: SCL signal rising time [s]* [1]
nf: Number of digital noise filter stages specified in the RIICnMR3.NF[1:0] bits

**Note 1.** The rising time (tr) and falling time (tf) of the SCL signal depend on the total capacitance of the bus line (Cb) and pull-up resistor (Rp). For details, see I²C Bus Standard from NXP Semiconductors.

Table 26.9 Minimum Specifiable Value for RIICnBRL and RIICnBRH

| SCLE | NFE | nf | Minimum Pulse Width that Passes through Digital Filter | Minimum Specifiable Value for BRH and BRL | Pulse Width when Minimum Value is Specified |
|---|---|---|---|---|---|
| 0 | 0 | — | 1 × IICϕ | 1 | 2 × IICϕ |
| 0 | 1 | 1 | 2 × IICϕ | 2 | 3 × IICϕ |
| 0 | 1 | 2 | 3 × IICϕ | 3 | 4 × IICϕ |
| 0 | 1 | 3 | 4 × IICϕ | 4 | 5 × IICϕ |
| 0 | 1 | 4 | 5 × IICϕ | 5 | 6 × IICϕ |
| IICϕ cycle > P0ϕ cycle (CKS ≠ 000) | | | | | |
| 1 | 0 | — | 1 × IICϕ | 0 | 2 × IICϕ |
| 1 | 1 | 1 | 2 × IICϕ | 1 | 4 × IICϕ |
| 1 | 1 | 2 | 3 × IICϕ | 2 | 6 × IICϕ |
| 1 | 1 | 3 | 4 × IICϕ | 3 | 8 × IICϕ |
| 1 | 1 | 4 | 5 × IICϕ | 4 | 10 × IICϕ |
| IICϕ cycle = P0ϕ cycle (CKS = 000) | | | | | |
| 1 | 0 | — | 2 × P0ϕ | 0 | 3 × IICϕ |
| 1 | 1 | 1 | 3 × P0ϕ | 1 | 5 × IICϕ |
| 1 | 1 | 2 | 4 × P0ϕ | 2 | 7 × IICϕ |
| 1 | 1 | 3 | 5 × P0ϕ | 3 | 9 × IICϕ |
| 1 | 1 | 4 | 6 × P0ϕ | 4 | 11 × IICϕ |

**Table 26.10** and **Table 26.11** list examples of RIICnBRH/RIICnBRL settings.

Table 26.10 Examples of RIICnBRH/RIICnBRL Settings for Transfer Rate (when RIICnFER.SCLE = 1 and RIICnFER.NFE= 0)

| Transfer Rate [kbps] | Peripheral Clock Operating Frequency P0ϕ [MHz] | | | |
|---|---|---|---|---|
| | 100 | | | |
| | RIICnMR1.CKS[2:0] | RIICnBRH.BRH | RIICnBRL.BRL | Actual transfer rate [kbps] |
| 12 | 111b | 30 (H'FE) | 31 (H'FF) | 12 |
| 50 | 101b | 28 (H'FC) | 30 (H'FE) | 50.4 |
| 100 | 100b | 28 (H'FC) | 31 (H'FF) | 99.2 |
| 400 | 010b | 29 (H'FD) | 30 (H'FE) | 396.8 |
| 1000 | 001b | 20 (H'F4) | 25 (H'F9) | 1020.4 |

Table 26.11 Examples of RIICnBRH/RIICnBRL Settings for Transfer Rate (when RIICnFER.SCLE = 1, RIICnFER.NFE= 1, and Number of NF Stages = 4)

| Transfer Rate [kbps] | Peripheral Clock Operating Frequency P0ϕ [MHz] | | | |
|---|---|---|---|---|
| | 100 | | | |
| | RIICnMR1.CKS[2:0] | RIICnBRH.BRH | RIICnBRL.BRL | Actual transfer rate [kbps] |
| 12 | 111b | 25 (H'F9) | 28 (H'FC) | 12 |
| 50 | 101b | 23 (H'F7) | 28 (H'FC) | 49.6 |
| 100 | 100b | 24 (H'F8) | 27 (H'FB) | 99.2 |
| 400 | 010b | 25 (H'F9) | 26 (H'FA) | 396.8 |
| 1000 | 001b | 14 (H'EE) | 24 (H'F8) | 1000 |

## 26.3.14 RIICnDRT — I²C Bus Transmit Data Register

**Access Size:** RIICnDRT is a 32-bit readable/writable register.
RIICnDRTL and RIICnDRTH are 16-bit readable/writable registers.
RIICnDRTLL, RIICnDRTLH, RIICnDRTHL, and RIICnDRTHH are 8-bit readable/writable registers.

**Address(es):** RIICnDRT: <RIICn_base> + H'003C
RIICnDRTL: <RIICn_base> + H'003C, RIICnDRTH: <RIICn_base> + H'003E
RIICnDRTLL: <RIICn_base> + H'003C, RIICnDRTLH: <RIICn_base> + H'003D,
RIICnDRTHL: <RIICn_base> + H'003E, RIICnDRTHH: <RIICn_base> + H'003F

**Initial Value:** H'0000 00FF This register is initialized by any reset.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | DRT[7:0] | | | | | | | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 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 |

When RIICnDRT detects a space in the I²C bus shift register (RIICnDRS), it transfers the transmit data that has been written to RIICnDRT to RIICnDRS and starts transmitting data in transmit mode.

The double-buffer structure of RIICnDRT and RIICnDRS allows continuous transmit operation if the next transmit data has been written to RIICnDRT while the RIICnDRS data is being transmitted.

RIICnDRT can always be read and written. Write transmit data to RIICnDRT once when a transmit data empty interrupt (INTRIICTI) request is generated. When writing to bits 8 to 15, be sure to write 0 to these bits.

## 26.3.15 RIICnDRR — I²C Bus Receive Data Register

**Access Size:** RIICnDRR is a 32-bit readable/writable register.
RIICnDRRL and RIICnDRRH are 16-bit readable/writable registers.
RIICnDRRLL, RIICnDRRLH, RIICnDRRHL, and RIICnDRRHH are 8-bit readable/writable registers.

**Address(es):** RIICnDRR: <RIICn_base> + H'0040
RIICnDRRL: <RIICn_base> + H'0040, RIICnDRRH: <RIICn_base> + H'0042
RIICnDRRLL: <RIICn_base> + H'0040, RIICnDRRLH: <RIICn_base> + H'0041,
RIICnDRRHL: <RIICn_base> + H'0042, RIICnDRRHH: <RIICn_base> + H'0043

**Initial Value:** H'0000 0000 This register is initialized by any reset.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | DRR[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 |

When 1 byte of data has been received, the received data is transferred from the I²C bus shift register (RIICnDRS) to RIICnDRR to enable the next data to be received.

The double-buffer structure of RIICnDRS and RIICnDRR allows continuous receive operation if the received data has been read from RIICnDRR while RIICnDRS is receiving data.

RIICnDRR cannot be written. Read data from RIICnDRR once when a receive data full interrupt (INTRIICRI) request is generated.

If DRR receives the next receive data before the current data is read from RIICnDRR (while the RIICnSR2.RDRF flag is 1), the RIIC automatically holds the SCL clock low one cycle before the RDRF flag is set to 1 next.

## 26.3.16 RIICnDRS — I²C Bus Shift Register

**Access Size:** This register is not accessible.

**Address(es):** —

**Initial Value:** H'0000 00FF

| 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 | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | DRS[7:0] | | | | | | | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| R/W | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |

RIICnDRS is an 8-bit shift register to transmit and receive data.

During transmission, transmit data is transferred from RIICnDRT to RIICnDRS and is sent from the SDA pin. During reception, data is transferred from RIICnDRS to RIICnDRR after 1 byte of data has been received.

RIICnDRS cannot be accessed directly.

## 26.4 Interrupt Sources

The RIIC issues eight types of interrupt request: transmit end, receive data full, transmit data empty, stop condition detection, start condition detection, NACK reception, arbitration-lost, and timeout.

**Table 26.12** lists details of the several interrupt requests. The receive data full and transmit data empty sources are both capable of launching data transfer by the DMAC.

Table 26.12 Interrupt Sources

| Symbol | Interrupt Source | Interrupt Flag | DMAC Launching | Priority*1 | Interrupt Condition |
|---|---|---|---|---|---|
| INTRIICRI | Receive-data-full | RDRF | Possible | High | RDRF = 1 • RIE = 1 |
| INTRIICTI | Transmit-data-empty | TDRE | Possible | ↑ | TDRE = 1 • TIE = 1 |
| INTRIICTEI | Transmission complete | TEND | Not possible | ↑ | TEND = 1 • TEIE = 1 |
| INTRIICNAKI | Reception of a NACK | NACKF | Not possible | ↑ | NACKF = 1 • NAKIE = 1 |
| INTRIICSPI | Detection of a stop condition | STOP | Not possible | ↑ | STOP = 1 • SPIE = 1 |
| INTRIICSTI | Detection of a start condition | START | Not possible | ↑ | START = 1 • STIE = 1 |
| INTRIICALI | Arbitration lost | AL | Not possible | ↑ | AL = 1 • ALIE = 1 |
| INTRIICTMOI | Timeout | TMOF | Not possible | Low | TMOF = 1 • TMOIE = 1 |

Note 1. When the interrupt priority register (ICDIPRn) setting is the same

Clear or mask the each flag during interrupt handling.

**CAUTIONS**

1. There is a latency (delay) between the execution of a write instruction for a peripheral module by the CPU and actual writing to the module. Thus, when an interrupt flag has been cleared or masked, read the relevant flag again to check whether clearing or masking has been completed, and then return from interrupt processing. Returning from interrupt processing without checking that writing to the module has been completed creates a possibility of repeated processing of the same interrupt.
2. Since INTRIICRI and INTRIICTI are edge-detected interrupts, they do not require clearing.
3. When using the INTRIICTEI interrupt, clear the RIICnSR2.TEND flag in the INTRIICTEI interrupt processing.
4. When using the INTRIICSPI interrupt, clear the RIICnSR2.STOP flag in the INTRIICSPI interrupt processing.
5. When using the INTRIICSTI interrupt, clear the RIICnSR2.START flag in the INTRIICSTI interrupt processing.
6. When using the INTRIICNAKI interrupt, clear the RIICnSR2.NACKF flag in the INTRIICNAKI interrupt processing.
7. When using the INTRIICALI interrupt, clear the RIICnSR2.AL flag in the INTRIICALI interrupt processing.
8. When using the INTRIICTMOI interrupt, clear the RIICnSR2.TMOF flag in the INTRIICTMOI interrupt processing.

# 26.5 Operation

## 26.5.1 Communication Data Format

The I²C bus format consists of 8-bit data and 1-bit acknowledge (one frame). After a start condition or restart condition is issued, the master device sends the slave address and data direction in the first frame. The specified slave is valid until a stop condition is issued or a new slave is specified by a restart condition.

**Figure 26.3** shows the I²C bus format, and **Figure 26.4** shows the I²C bus timing.

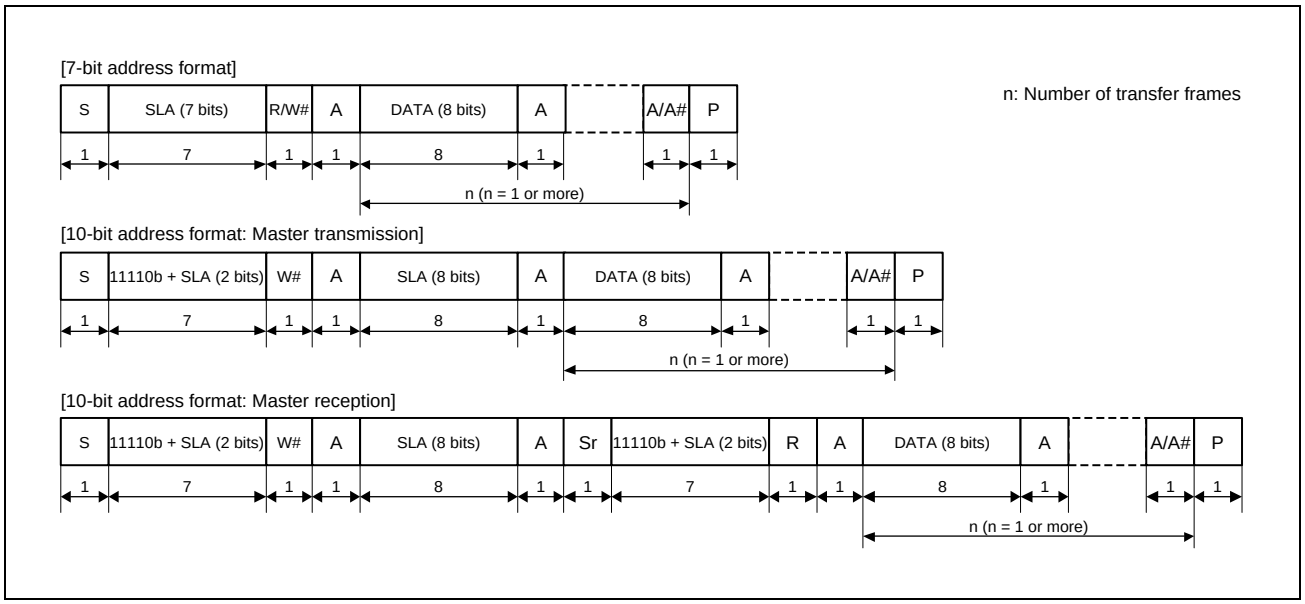
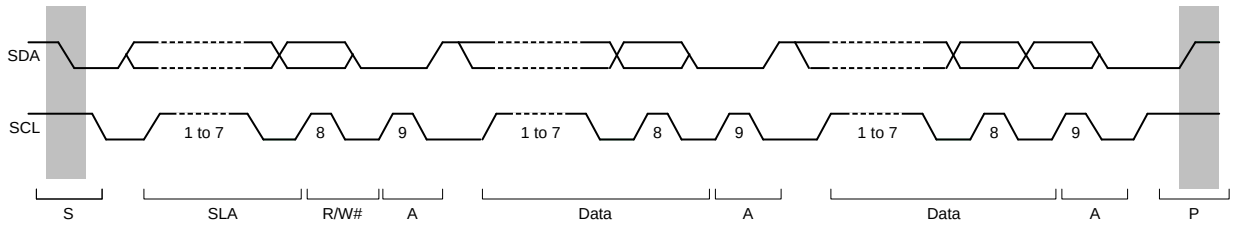



Figure 26.3 I²C Bus Format




S: Start condition. The master device drives the SDA line low from high level while the SCL line is at a high level.

SLA: Slave address, by which the master device selects a slave device.

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 the SDA line low. (In master transmit mode, the slave device returns acknowledge. In master receive mode, the master device returns acknowledge.)

A#: Not-acknowledge. The receiving device has not returned a response or is not present so the SDA line has remained at the high level.

Sr: Restart condition. The master device drives the SDA line low from the high level after the setup time has elapsed with the SCL line at the high level.

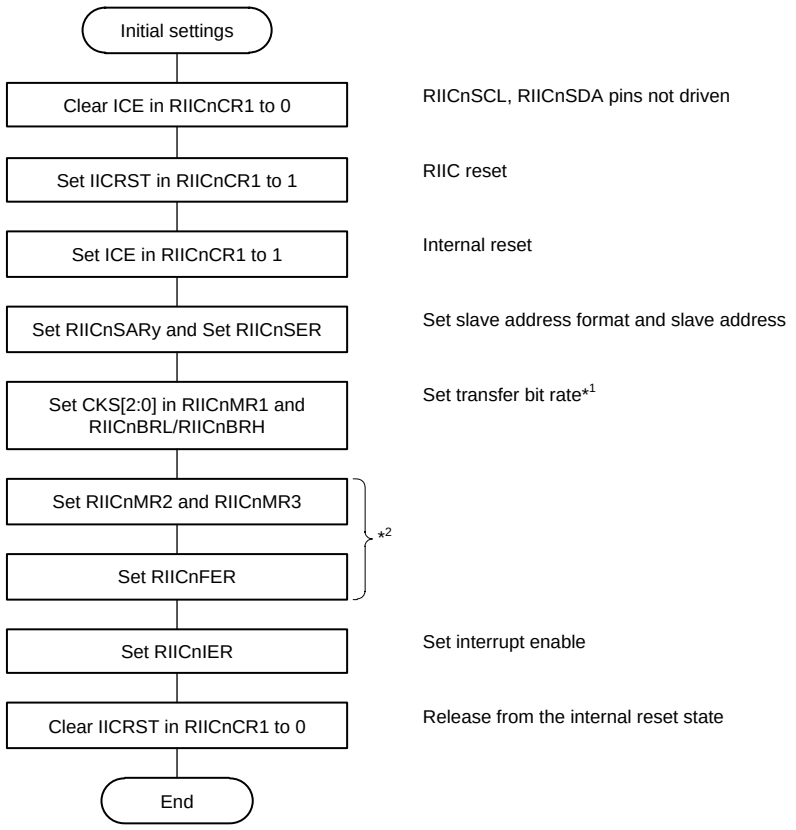
DATA: Transmitted or received data

P: Stop condition. The master device drives the SDA line high from low level while the SCL line is at a high level.

Figure 26.4 $I^2C$ Bus Timing (SLA = 7 Bits)

### 26.5.2 Initial Settings

Before starting data transmission and reception, initialize the RIIC according to the procedure in **Figure 26.5**.




**Note:** y = 0 to 2

Note 1. When the RIIC is used only in slave mode, set the RIICnBRL register to a value at least the same as the data setup time.

Note 2. Set these registers as necessary.

Figure 26.5 Example of RIIC Initialization Flowchart

### 26.5.3 Master Transmit Operation

In master transmit operation, the RIIC outputs the SCL (clock) and transmitted data signals as the master device, and the slave device returns acknowledgements. **Figure 26.6** shows an example of usage of master transmission and **Figure 26.7** to **Figure 26.9** show the timing of operations in master transmission.

The following describes the procedure and operations for master transmission.

(1) Set the RIICnCR1.IICRST bit 1 to 1 (RIIC reset) and then set the RIICnCR1.ICE bit to 1 (internal reset) with the RIICnCR1.ICE bit cleared to 0 (RIICnSCL and RIICnSDA pins not driven). This initializes the internal state and the various flags of RIICnSR1. After that, set registers RIICnSARy, RIICnSER, RIICnMR1, RIICnBRH, and RIICnBRL (y = 0 to 2), and set the other registers as necessary (for initial settings of the RIIC, see **Figure 26.5**). When the necessary register settings have been completed, set the RIICnCR1.IICRST bit to 0 (for release from the reset state). This step is not necessary if initialization of the RIIC has already been completed.

(2) Read the RIICnCR2.BBSY flag to check that the bus is open, and then set the RIICnCR2.ST bit to 1 (start condition issuance request). Upon receiving the request, the RIIC issues a start condition. At the same time, the BBSY flag and the RIICnSR2.START flag are automatically set to 1 and the ST bit is automatically cleared to 0. At this time, if the start condition is detected and the internal levels for the SDA output state and the levels on the SDA line have matched while the ST bit is 1, the RIIC recognizes that issuing of the start condition as requested by the ST bit has been successfully completed, and the RIICnCR2.MST and TRS bits are automatically set to 1, placing the RIIC in master transmit mode. The RIICnSR2.TDRE flag is also automatically set to 1 in response to setting of the TRS bit to 1.

(3) Check that the RIICnSR2.TDRE flag is 1, and then write the value for transmission (the slave address and the R/W# bit) to RIICnDRT. Once the data for transmission are written to RIICnDRT, the TDRE flag is automatically cleared to 0, the data are transferred from RIICnDRT to RIICnDRS, and the TDRE flag is again set to 1. After the byte containing the slave address and R/W# bit has been transmitted, the value of the TRS bit is automatically updated to select master transmit or master receive mode in accord with the value of the transmitted R/W# bit. If the value of the R/W# bit was 0, the RIIC continues in master transmit mode.

Since the RIICnSR2.NACKF flag being 1 at this time indicates that no slave device recognized the address or there was an error in communications, write 1 to the RIICnCR2.SP bit to issue a stop condition.

For data transmission with an address in the 10-bit format, start by writing 1111 0b, the two higher-order bits of the slave address, and W# to RIICnDRT as the first address transmission. Then, as the second address transmission, write the eight lower-order bits of the slave address to RIICnDRT.

(4) After confirming that the RIICnSR2.TDRE flag is 1, write the data for transmission to the RIICnDRT register. The RIIC automatically holds the SCL line low until the data for transmission are ready or a stop condition is issued.

(5) After the last byte of the data to be transmitted is written to the RIICnDRT register, wait until the value of the RIICnSR2.TEND flag returns to 1, and then set the RIICnCR2.SP bit to 1 (stop condition issuance request). Upon receiving a stop condition issuance request, the RIIC issues the stop condition.

(6) Upon detecting the stop condition, the RIIC automatically clears the RIICnCR2.MST and TRS bits to 00b and enters slave receive mode. Furthermore, it automatically clears the RIICnSR2.TDRE and TEND flags to 0, and sets the RIICnSR2.STOP flag in to 1.

(7) Clear the RIICnSR2.NACKF and STOP flags to 0.

**CAUTION**

Operations for transfer start if the RIICnSR2.NACKF flag is cleared to 0 before RIICnSR2.STOP is set to 1. Be sure to confirm that RIICnSR2.STOP is set to 1 before clearing RIICnSR2.NACKF to 0. In particular, when the NACK receive interrupt (INTRIICNAKI) is in use, take care not to clear the NACKF flag to 0 before the STOP flag is set to 1 during interrupt processing.

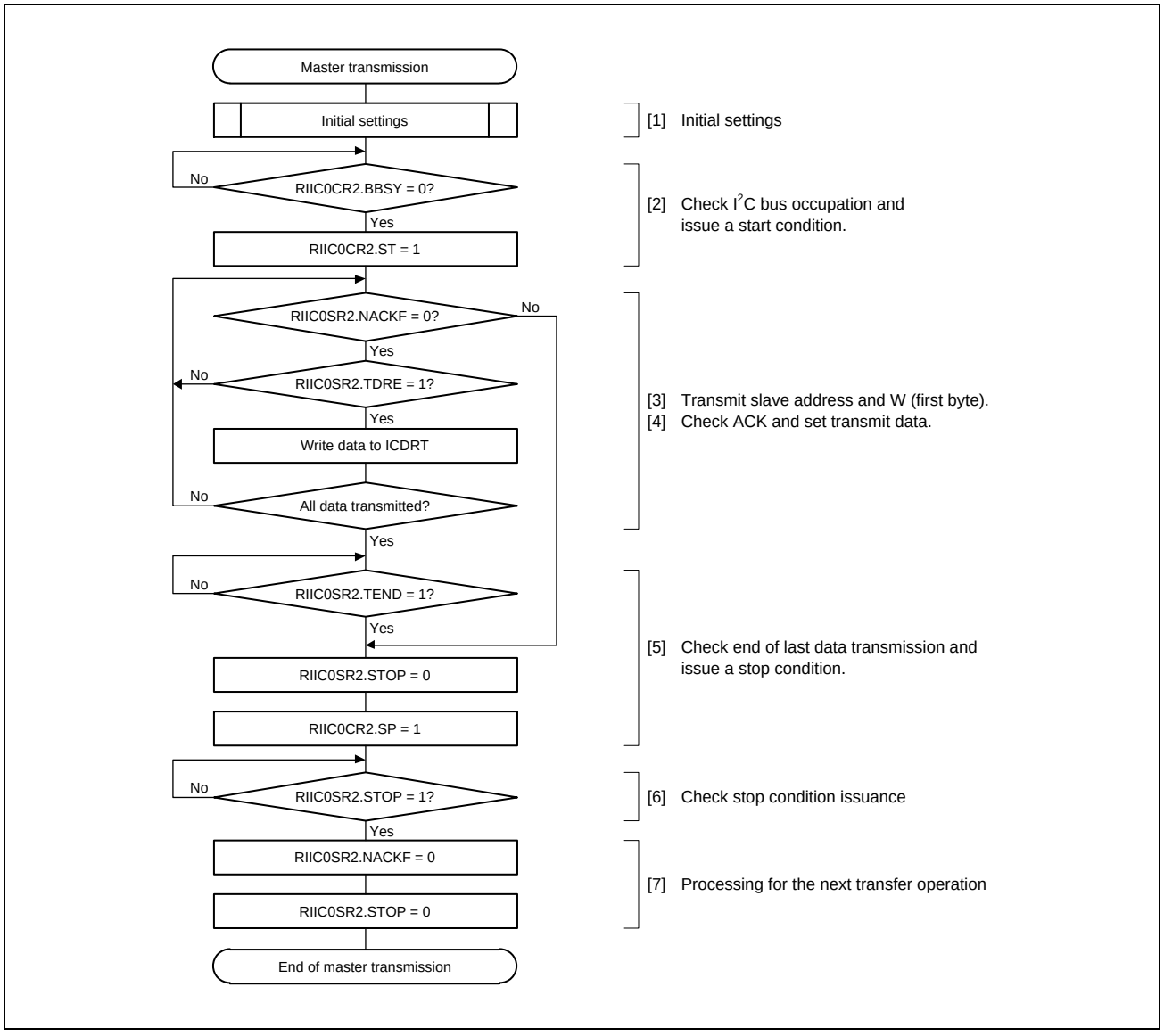



Figure 26.6 Example of Master Transmission Flowchart

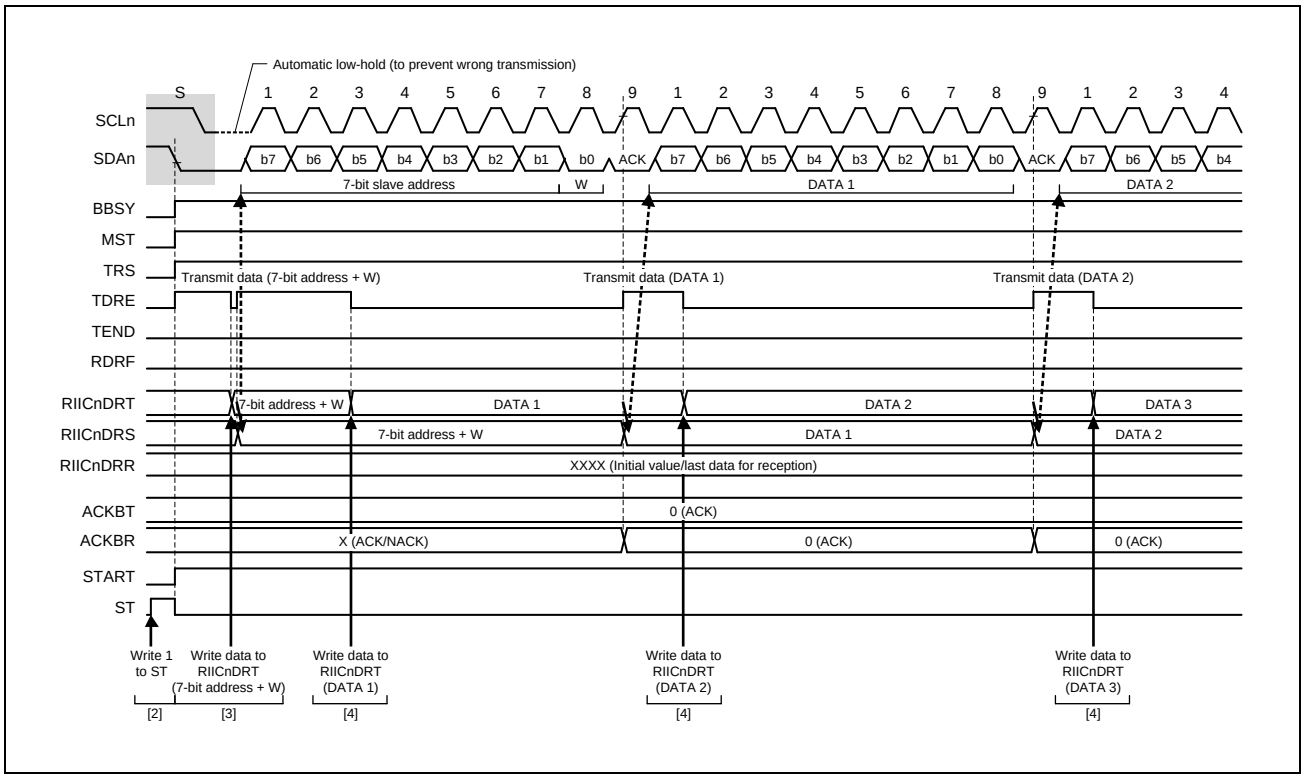



Figure 26.7 Master Transmit Operation Timing (1) (7-Bit Address Format)

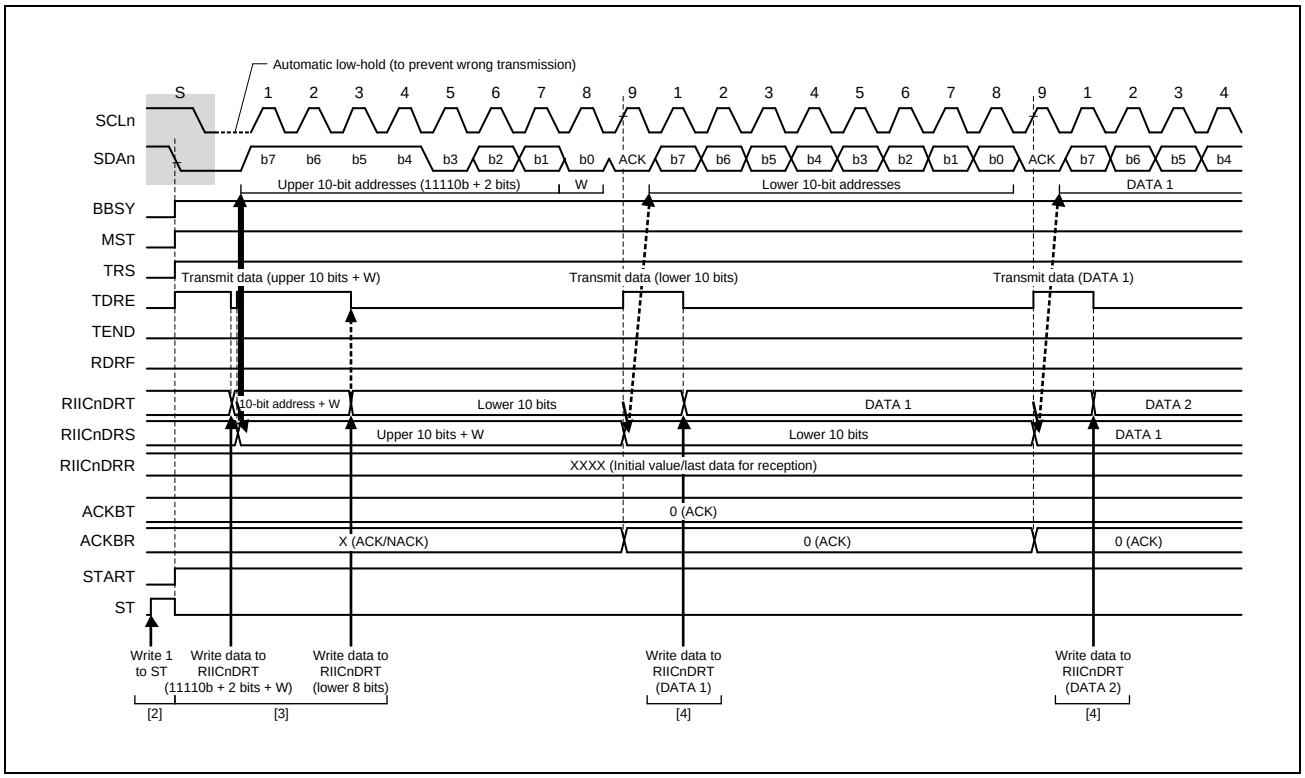



Figure 26.8 Master Transmit Operation Timing (2) (10-Bit Address Format)

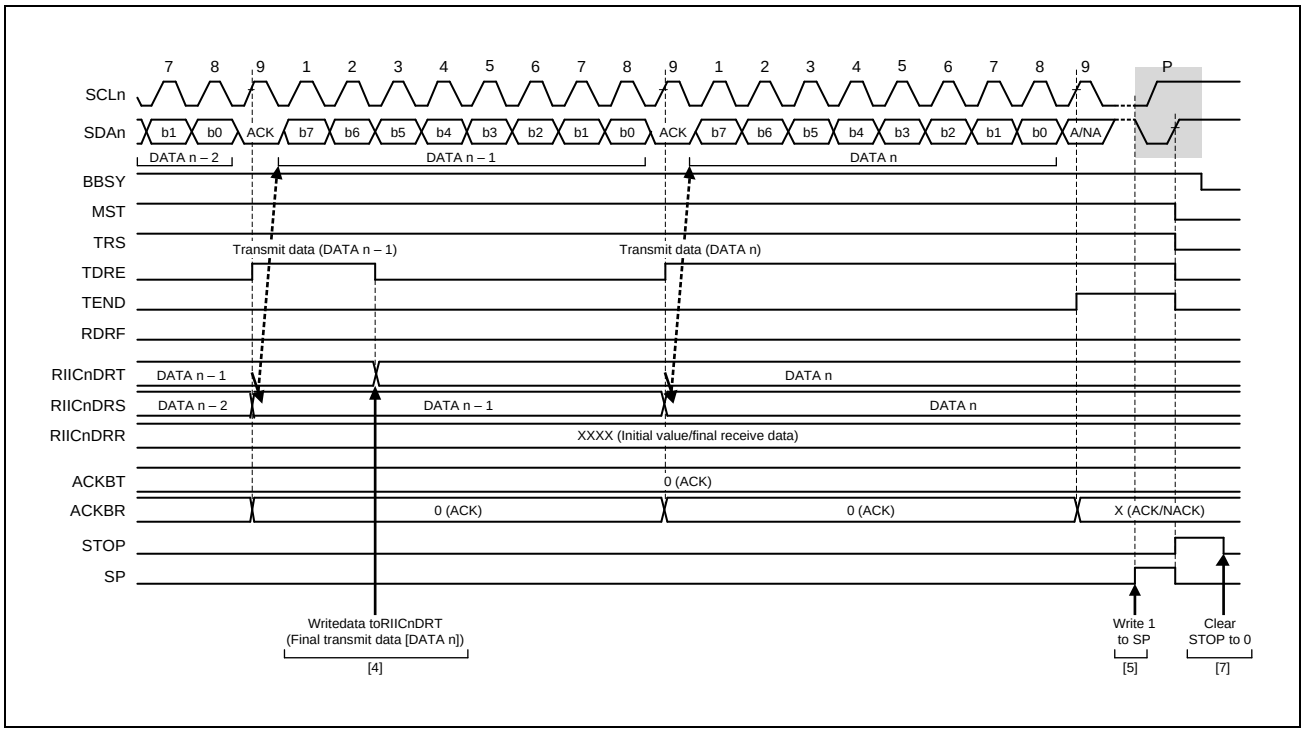



Figure 26.9 Master Transmit Operation Timing (3)

### 26.5.4 Master Receive Operation

In master receive operation, the RIIC as a master device outputs the SCL (clock) signal, receives data from the slave device, and returns acknowledgements. Since the RIIC must start by sending a slave address to the corresponding slave device, this part of the procedure is performed in master transmit mode, but the subsequent steps are in master receive mode.

**Figure 26.10** shows an example of usage for the master reception of 3 or more bytes (7-bit address format), **Figure 26.14** shows an example of usage for the master reception of 1 or 2 bytes (7-bit address format), and **Figure 26.11** to **Figure 26.13** show the timing of operations in master reception.

The following describes the procedure and operations for master reception.

(1) Set the RIICnCR1.IICRST bit to 1 (RIIC reset) and then set the RIICnCR1.ICE bit to 1 (internal reset) with the RIICnCR1.ICE bit cleared to 0 (RIICnSCL and RIICnSDA pins not driven). This initializes the internal state and the various flags of RIICnSR1. After that, set registers RIICnSARy, RIICnSER, RIICnMR1, RIICnBRH, and RIICnBRL (y = 0 to 2), and set the other registers as necessary (for initial settings of the RIIC, see **Figure 26.5**). When the necessary register settings have been completed, set the RIICnCR1.IICRST bit to 0 (for release from the reset state). This step is not necessary if initialization of the RIIC has already been completed.

(2) Read the RIICnCR2.BBSY flag to check that the bus is open, and then set the RIICnCR2.ST bit to 1 (start condition issuance request). Upon receiving the request, the RIIC issues a start condition. When the RIIC detects the start condition, the BBSY flag and the RIICnSR2.START flag are automatically set to 1 and the ST bit is automatically cleared to 0. At this time, if the start condition is detected and the levels for the SDA output and the levels on the SDA line have matched while the ST bit is 1, the RIIC recognizes that issuing of the start condition as requested by the ST bit has been successfully completed, and the RIICnCR2.MST and TRS bits are automatically set to 1, placing the RIIC in master transmit mode. The RIICnSR2.TDRE flag is also automatically set to 1 in response to setting of the TRS bit to 1.

(3) Check that the RIICnSR2.TDRE flag is 1, and then write the value for transmission (the first byte indicates the slave address and value of the R/W# bit) to RIICnDRT. Once the data for transmission are written to RIICnDRT, the TDRE flag is automatically cleared to 0, the data are transferred from RIICnDRT to RIICnDRS, and the TDRE flag is again set to 1. Once the byte containing the slave address and R/W# bit has been transmitted, the value of the RIICnCR2.TRS bit is automatically updated to select transmit or receive mode in accord with the value of the transmitted R/W# bit. If the value of the R/W# bit was 1, the RIICnCR2.TRS bit is cleared to 0 on the rising edge of the ninth cycle of SCL (the clock signal), placing the RIIC in master receive mode. At this time, the TDRE flag is automatically cleared to 0 and the RIICnSR2.RDRF flag is automatically set to 1.

Since the RIICnSR2.NACKF flag being 1 at this time indicates that no slave device recognized the address or there was an error in communications, write 1 to the RIICnCR2.SP bit to issue a stop condition.

For master reception from a device with a 10-bit address, start by using master transmission to issue the 10-bit address, and then issue a restart condition. After that, transmitting 1111 0b, the two higher-order bits of the slave address, and the R bit places the RIIC in master receive mode.

(4) Dummy read RIICnDRR after confirming that the RIICnSR2.RDRF flag is 1; this makes the RIIC start output of the SCL (clock) signal and start data reception.

(5) After 1 byte of data has been received, the RIICnSR2.RDRF flag is set to 1 on the rising edge of the eighth or ninth cycle of SCL clock (the clock signal) as selected by the RIICnMR3.RDRFS bit. Reading out RIICnDRR at this time will produce the received data, and the RDRF flag is automatically cleared to 0 at the same time. Furthermore, the value of the acknowledgement field received during the ninth cycle of SCL clock is returned as the value set in the RIICnMR3.ACKBT bit. Furthermore, if the next byte to be received is the next to last byte, set the RIICnMR3.WAIT bit to 1 (for wait insertion) before reading the RIICnDRR (containing the second byte from last). As well as enabling NACK output even in the case of delays in processing to set the RIICnMR3.ACKBT bit

to 1 (NACK) in step (6), due to other interrupts, etc., this fixes the SCL line to the low level on the rising edge of the ninth clock cycle in reception of the last byte, so the state is such that issuing a stop condition is possible.

(6) When the RIICnMR3.RDRFS bit is 0 and the slave device must be notified that it is to end transfer for data reception after transfer of the next (final) byte, set the RIICnMR3.ACKBT bit to 1 (NACK).

(7) After reading out the byte before last from the RIICnDRR register, if the value of the RIICnSR2.RDRF flag is confirmed to be 1, write 1 to the RIICnCR2.SP bit (stop condition issuance request) and then read the last byte from RIICnDRR. When RIICnDRR is read, the RIIC is released from the wait state and issues the stop condition after low-level output in the ninth clock cycle is completed or the SCL line is released from the low-hold state.

(8) Upon detecting the stop condition, the RIIC automatically clears the RIICnCR2.MST and TRS bits to 00b and enters slave receive mode. Furthermore, detection of the stop condition leads to setting of the RIICnSR2.STOP flag to 1.

(9) After checking that the RIICnSR2.STOP flag is 1, clear the RIICnSR2.NACKF and STOP flags to 0 for the next transfer operation.

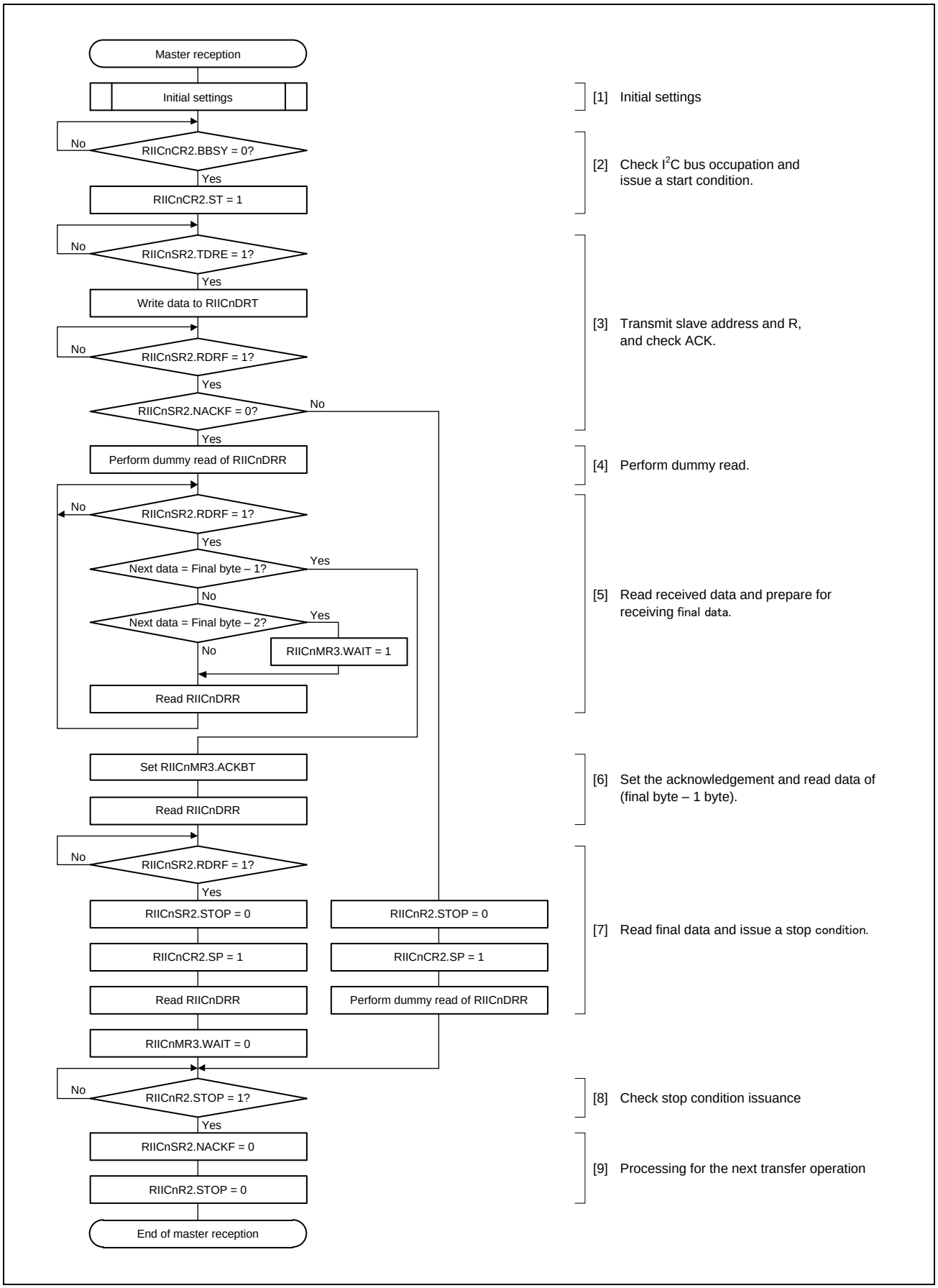



Figure 26.10 Example Flowchart for the Master Reception of 3 or More Bytes (7-Bit Address Format)

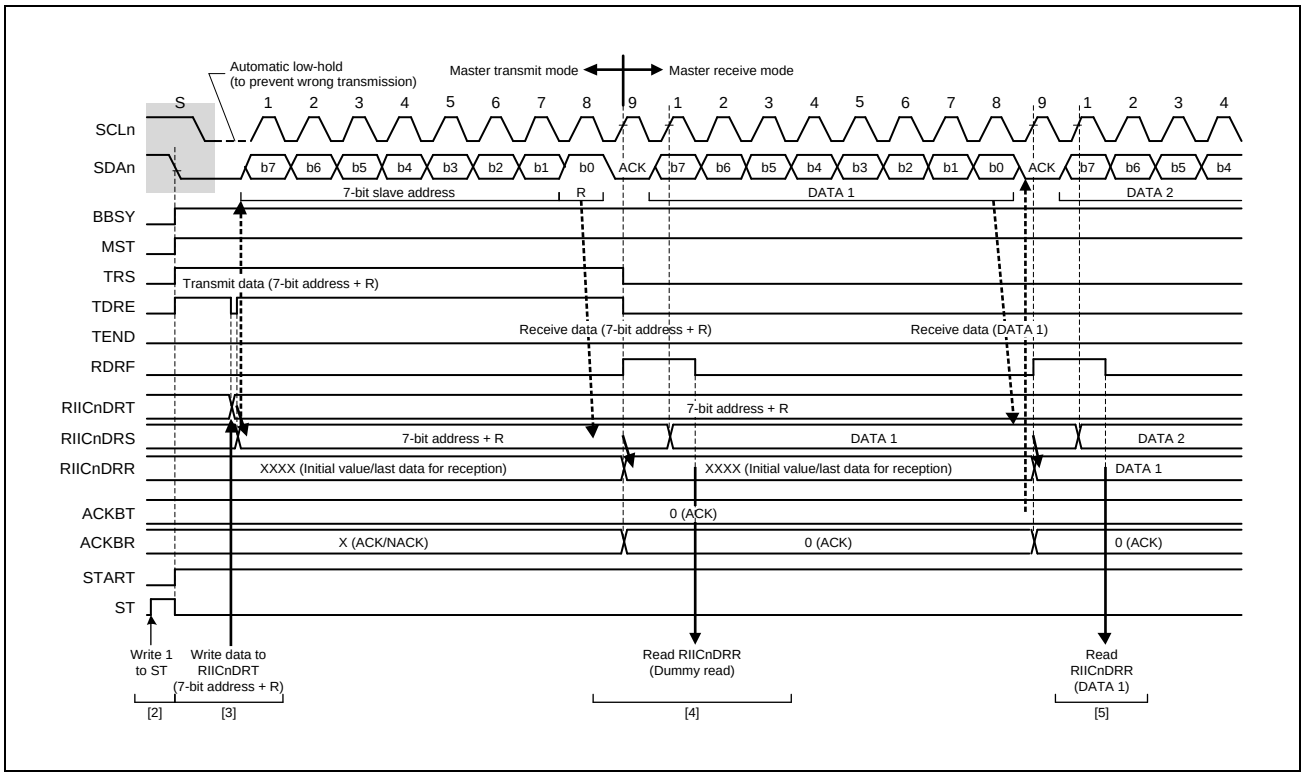



Figure 26.11 Master Receive Operation Timing (1) (7-Bit Address Format, when RDRFS = 0)

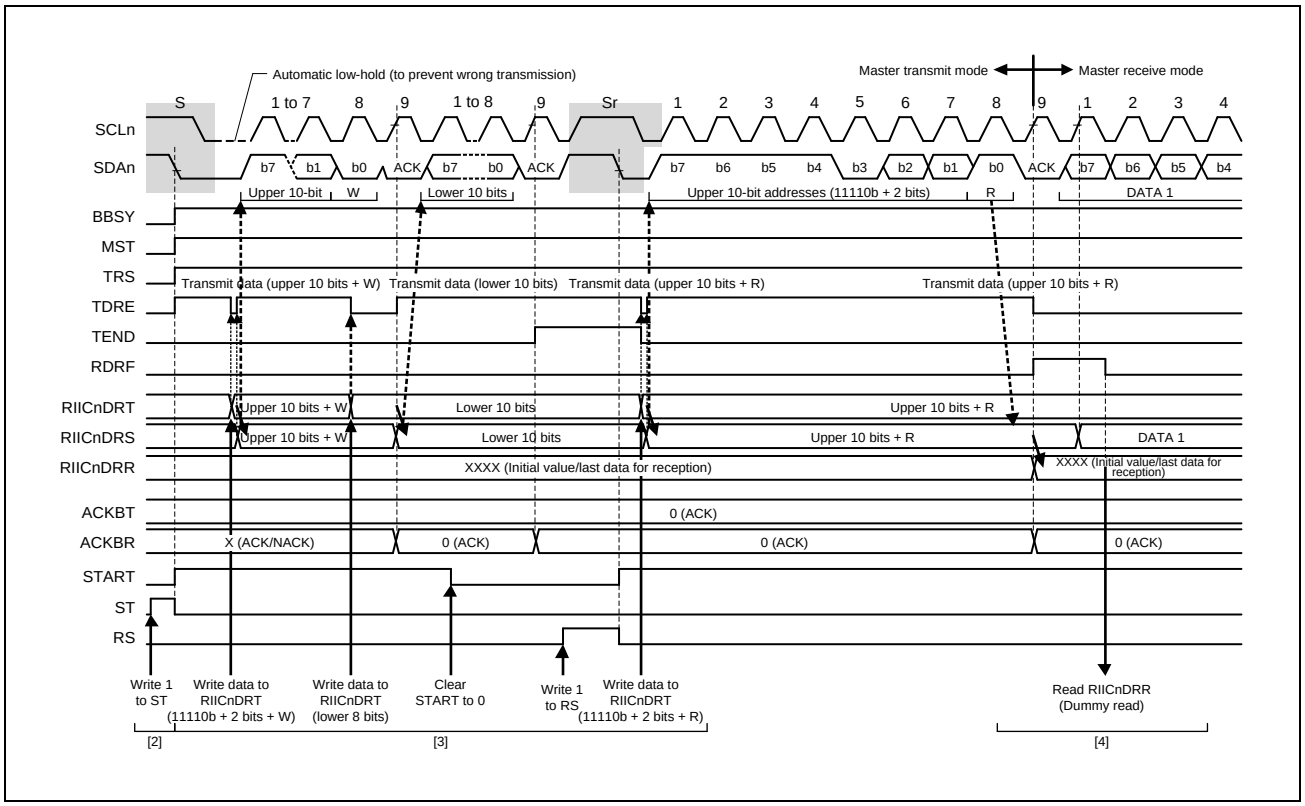



Figure 26.12 Master Receive Operation Timing (2) (10-Bit Address Format, when RDRFS = 0)

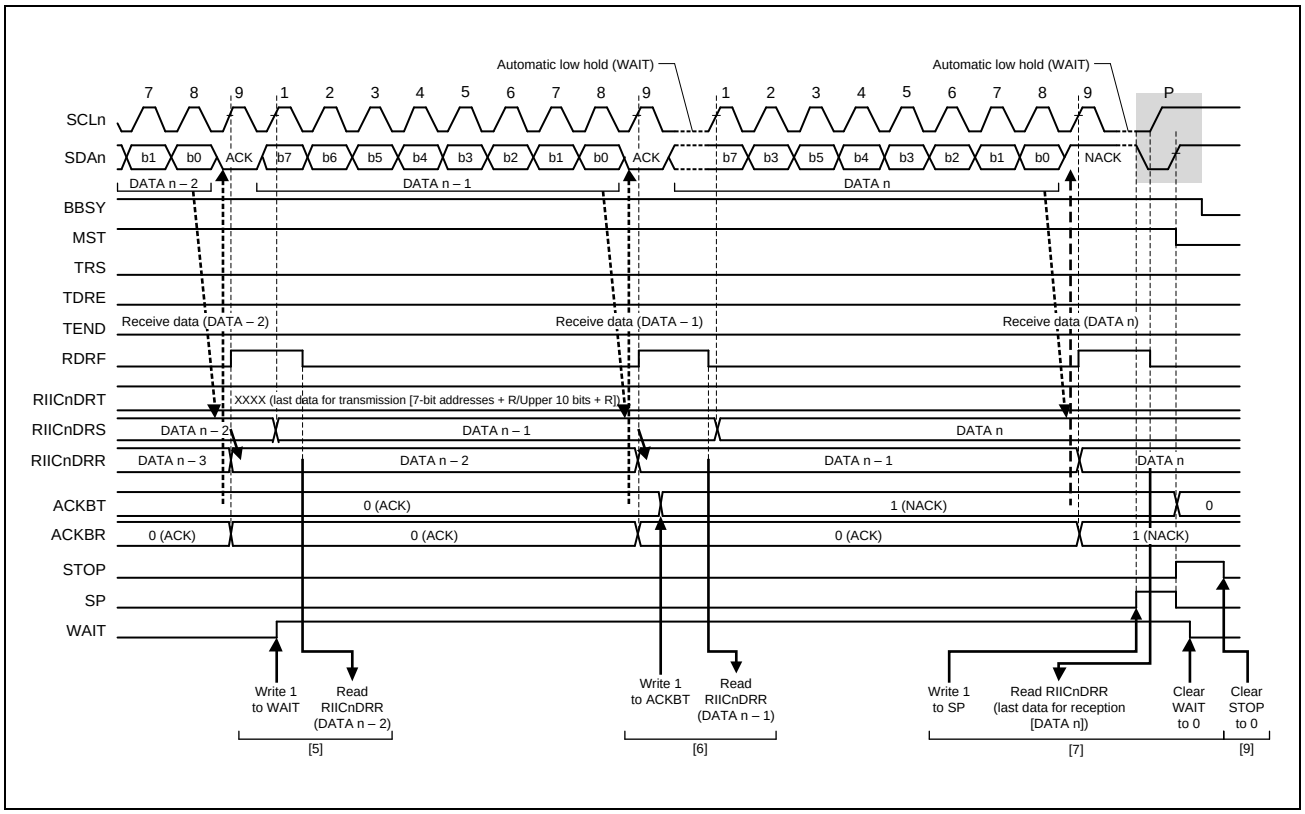



Figure 26.13 Master Receive Operation Timing (3) (when RDRFS = 0)

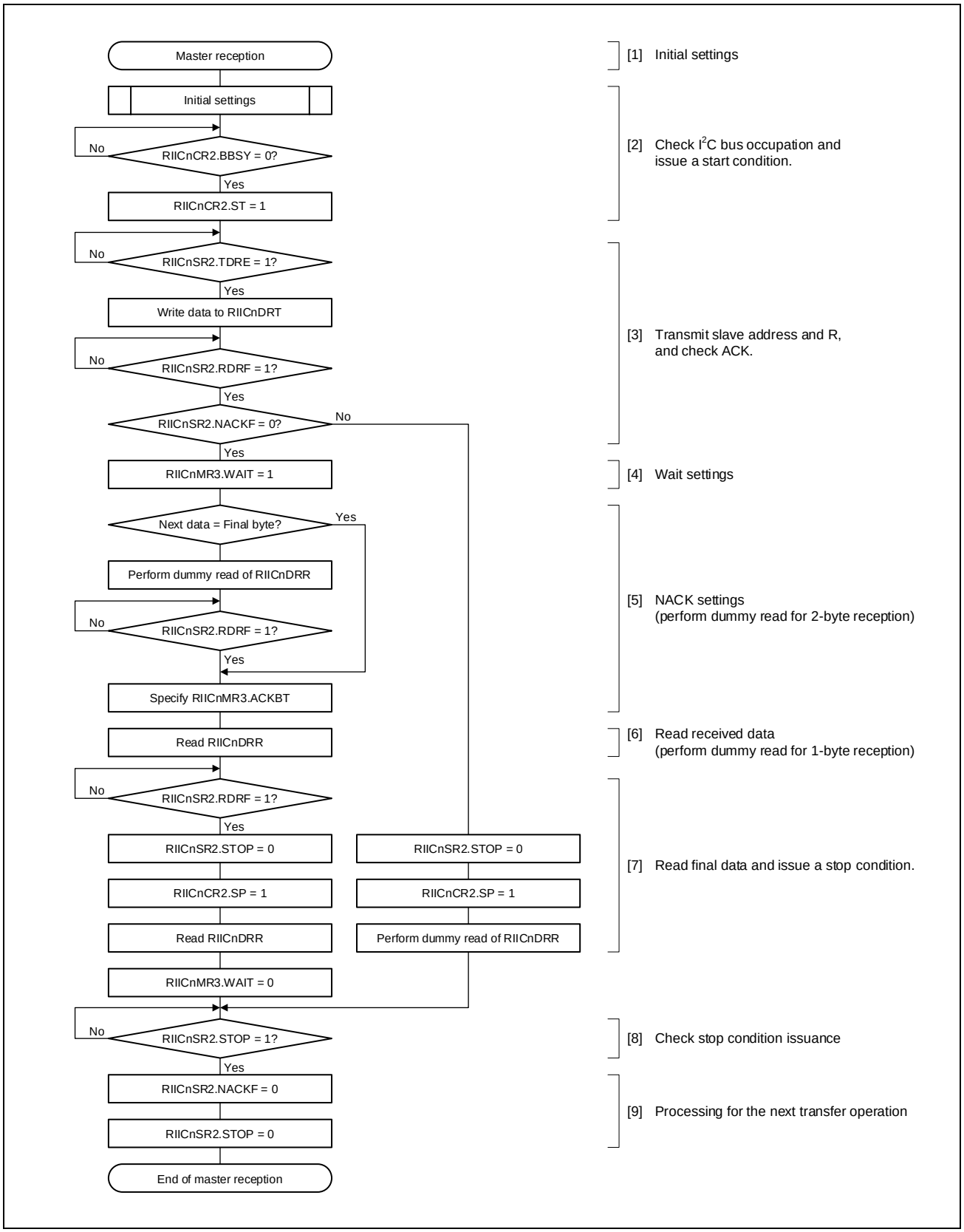



Figure 26.14 Example Flowchart for the Master Reception of 1 or 2 Bytes (7-Bit Address Format)

### 26.5.5 Slave Transmit Operation

In slave transmit operation, the master device outputs the SCL (clock) signal, the RIIC transmits data as a slave device, and the master device returns acknowledgements.

**Figure 26.15** shows an example of usage of slave transmission and **Figure 26.16** and **Figure 26.17** show the timing of operations in slave transmission.

The following describes the procedure and operations for slave transmission.

(1) Follow the procedure in **Figure 26.5** to make initial settings for the RIIC. This step is not necessary if initialization of the RIIC has already been completed. After initial settings, the RIIC will stay in the standby state until it receives a slave address that it matches.

(2) After receiving a matching slave address, the RIIC sets one of the corresponding bits RIICnSR1.HOA, GCA, and AASy (y = 0 to 2) to 1 on the rising edge of the ninth cycle of SCL clock (the clock signal) and outputs the value set in the RIICnMR3.ACKBT bit to the acknowledge bit on the ninth cycle of SCL clock. If the value of the R/W# bit that was also received at this time is 1, the RIIC automatically places itself in slave transmit mode by setting both the RIICnCR2.TRS bit and the RIICnSR2.TDRE flag to 1.

(3) After the RIICnSR2.TEND flag is confirmed to be 1, write the data for transmission to the RIICnDRT register. At this time, if the RIIC receives no acknowledge from the master device (receives an NACK signal) while the RIICnFER.NACKE bit is 1, the RIIC suspends transfer of the next data.

(4) Wait unit the RIICnSR2.TEND flag is set to 1 while the RIICnSR2.TDRE flag is 1, after the RIICnSR2.NACKF flag is set to 1 or the last byte for transmission is written to the RIICnDRT register. When the RIICnSR2.NACKF flag or the TEND flag is 1, the RIIC drives the SCL line low on the ninth falling edge of SCL clock.

(5) When the RIICnSR2.NACKF flag or the RIICnSR2.TEND flag is 1, dummy read RIICnDRR to complete the processing. This releases the SCL line.

(6) Upon detecting the stop condition, the RIIC automatically clears bits RIICnSR1.HOA, GCA, and AASy (y = 0 to 2), flags RIICnSR2.TDRE and TEND, and the RIICnCR2.TRS bit to 0, and enters slave receive mode.

(7) Clear the RIICnSR2.NACKF and STOP flags to 0.

**CAUTION**

Operations for transfer start if the RIICnSR2.NACKF flag is cleared to 0 before RIICnSR2.STOP is set to 1. Be sure to confirm that RIICnSR2.STOP is set to 1 before clearing RIICnSR2.NACKF to 0. In particular, when the NACK receive interrupt (INTRIICNAKI) is in use, take care not to clear the NACKF flag to 0 before the STOP flag is set to 1 during interrupt processing.

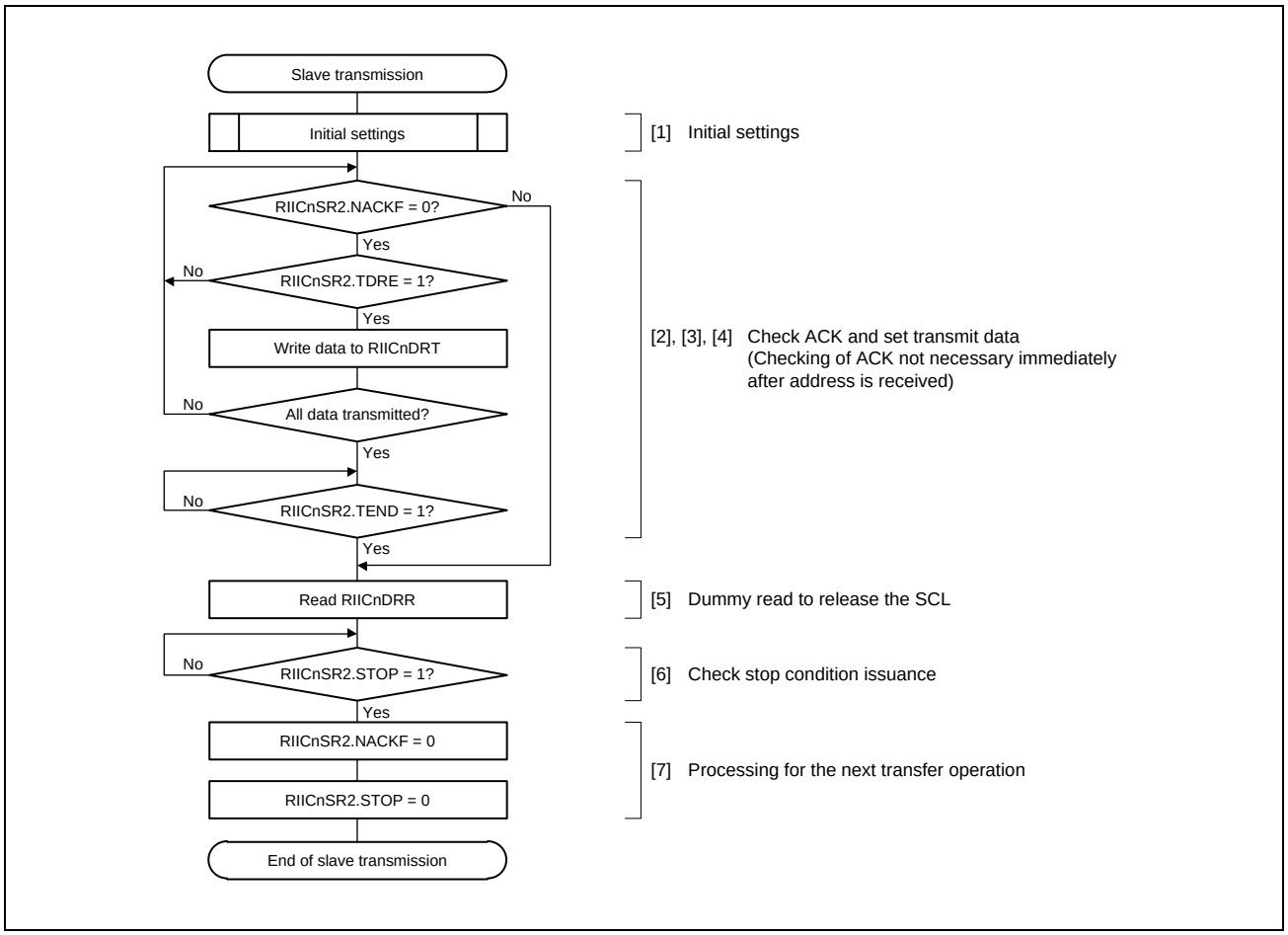



Figure 26.15 Example of Slave Transmission Flowchart

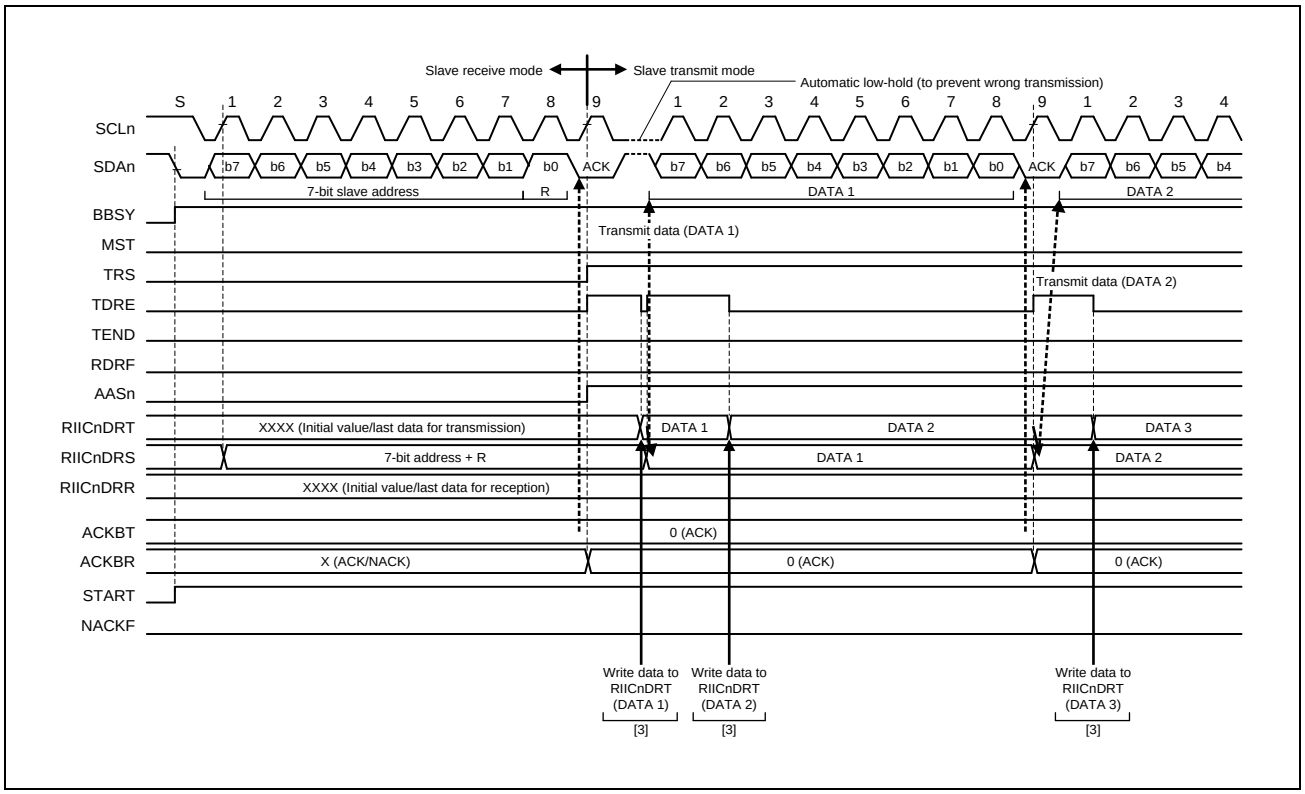



Figure 26.16 Slave Transmit Operation Timing (1) (7-Bit Address Format)

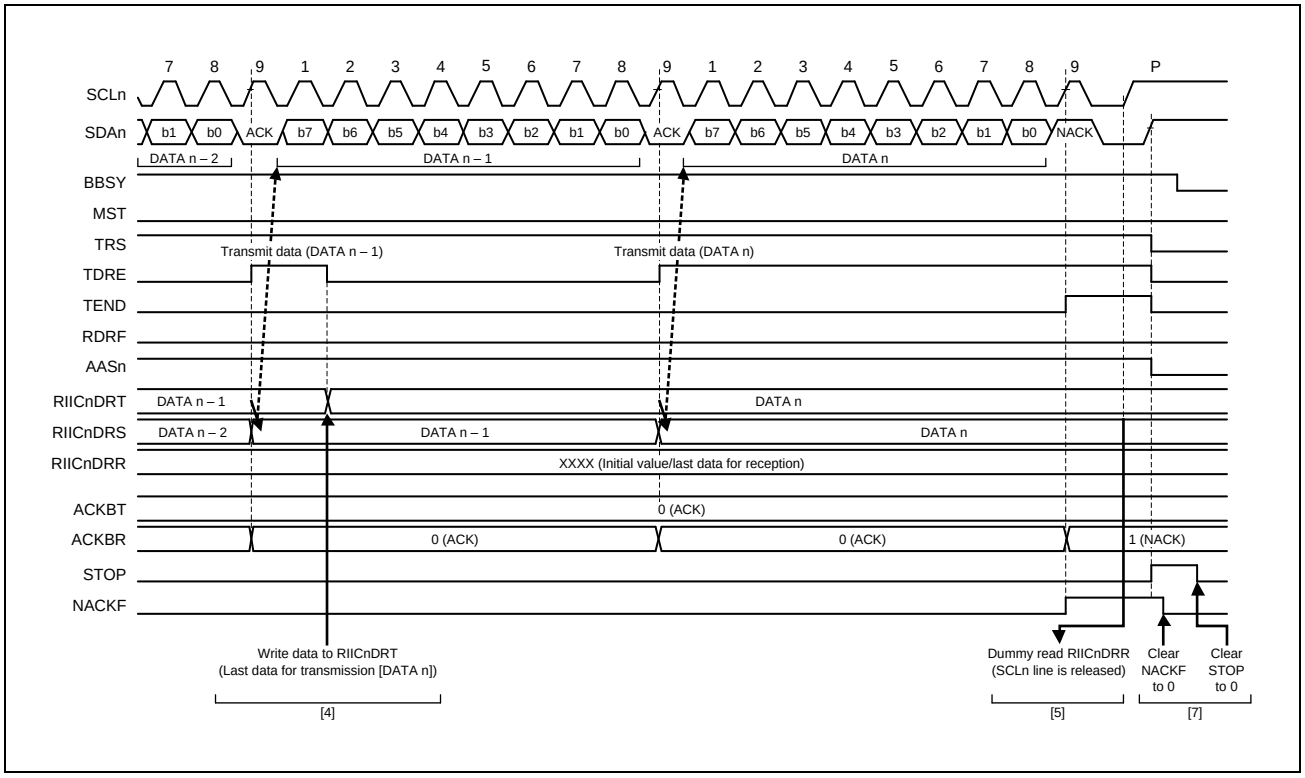



Figure 26.17 Slave Transmit Operation Timing (2)

### 26.5.6 Slave Receive Operation

In slave receive operation, the master device outputs the SCL clock and transmit data, and the RIIC returns acknowledgements as a slave device.

**Figure 26.18** shows an example of usage of slave reception and **Figure 26.19** and **Figure 26.20** show the timing of operations in slave reception.

The following describes the procedure and operations for slave reception.

(1) Follow the procedure in **Figure 26.5** to make initial settings for the RIIC. This step is not necessary if initialization of the RIIC has already been completed. After initial settings, the RIIC will stay in the standby state until it receives a slave address that it matches.

(2) After receiving a matching slave address, the RIIC sets one of the corresponding bits RIICnSR1.HOA, GCA, and AASy (y = 0 to 2) to 1 on the rising edge of the ninth cycle of SCL clock (the clock signal) and outputs the value set in the RIICnMR3.ACKBT bit to the acknowledge bit on the ninth cycle of SCL clock. If the value of the R/W# bit that was also received at this time is 0, the RIIC continues to place itself in slave receive mode and sets the RIICnSR2.RDRF flag to 1.

(3) After the RIICnSR2.STOP flag is confirmed to be 0 and the RIICnSR2.RDRF flag to be 1, dummy read RIICnDRR (the dummy value consists of the slave address and R/W# bit when the 7-bit address format is selected, or the lower eight bits when the 10-bit address format is selected).

(4) When RIICnDRR is read, the RIIC automatically clears the RIICnSR2.RDRF flag to 0. If reading of RIICnDRR is delayed and a next byte is received while the RDRF flag is still set to 1, the RIIC holds the SCL line low from one SCL cycle before the timing with which RDRF should be set. In this case, reading RIICnDRR releases the SCL line from being held at the low level.

When the RIICnSR2.STOP flag is 1 and the RIICnSR2.RDRF flag is also 1, read RIICnDRR until all the data is completely received.

(5) Upon detecting the stop condition, the RIIC automatically clears bits RIICnSR1.HOA, GCA, and AASy (y = 0 to 2) to 0.

(6) After checking that the RIICnSR2.STOP flag is 1, clear the RIICnSR2.STOP flag to 0 for the next transfer operation.

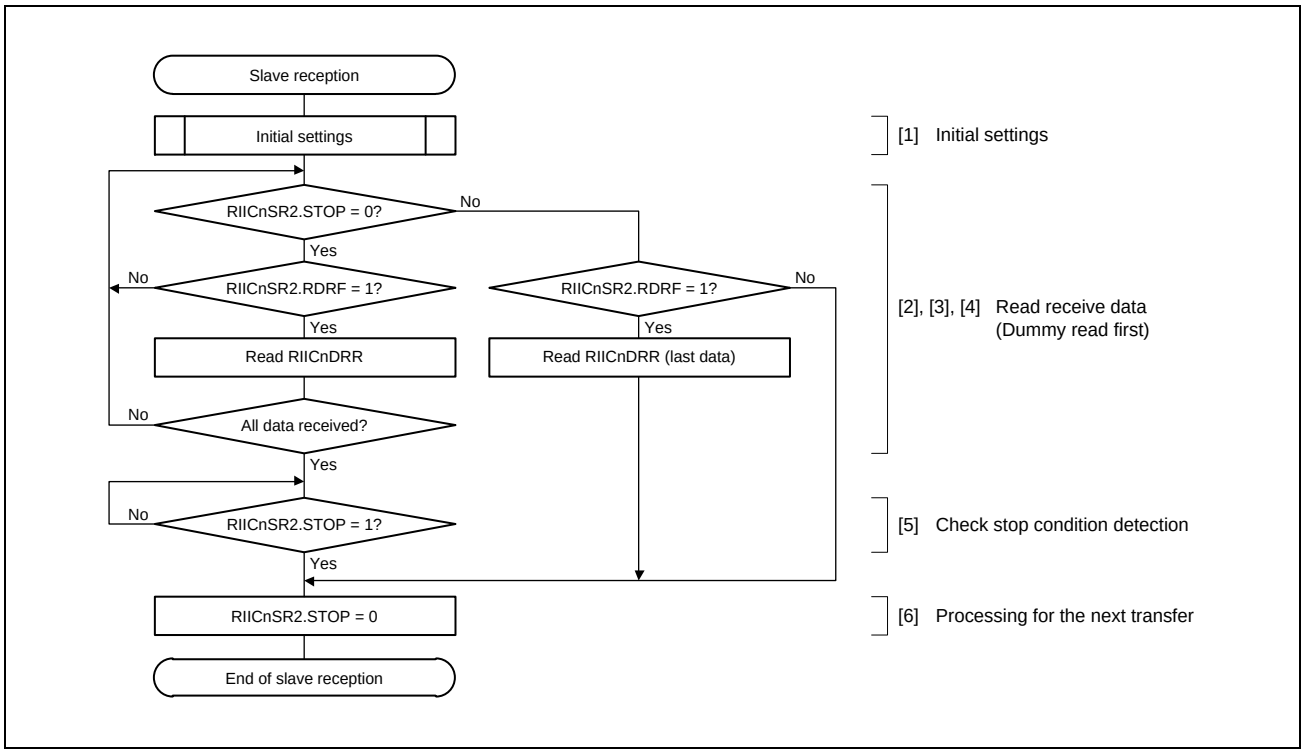



Figure 26.18 Example of Slave Reception Flowchart

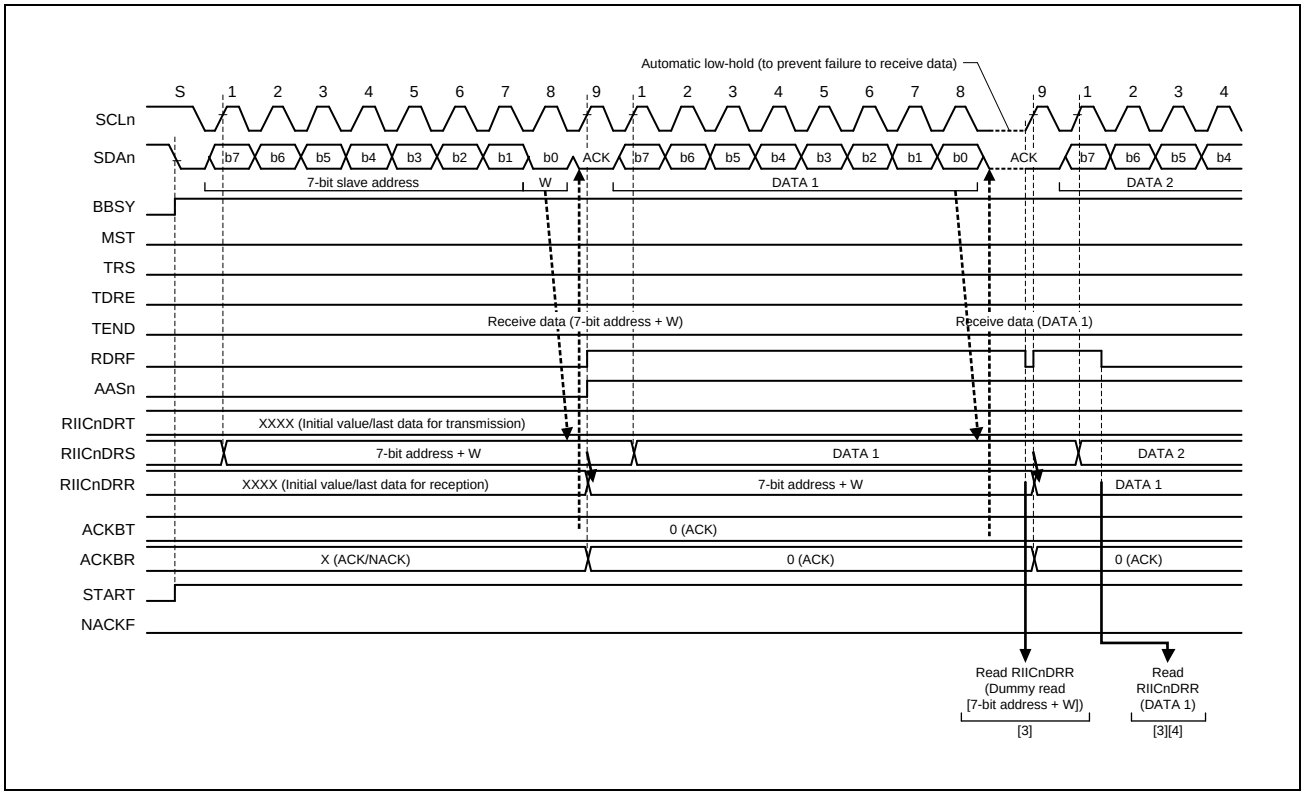



Figure 26.19 Slave Receive Operation Timing (1) (7-Bit Address Format, when RDRFS = 0)

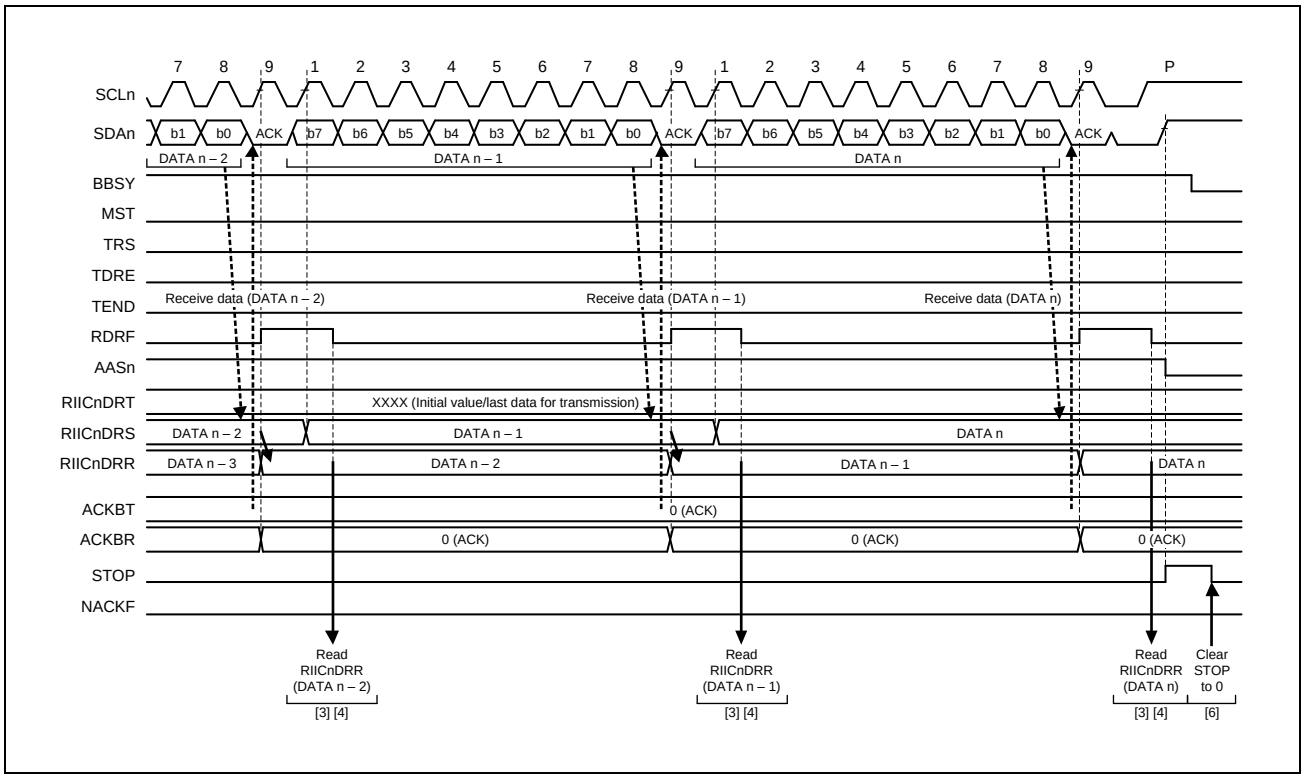


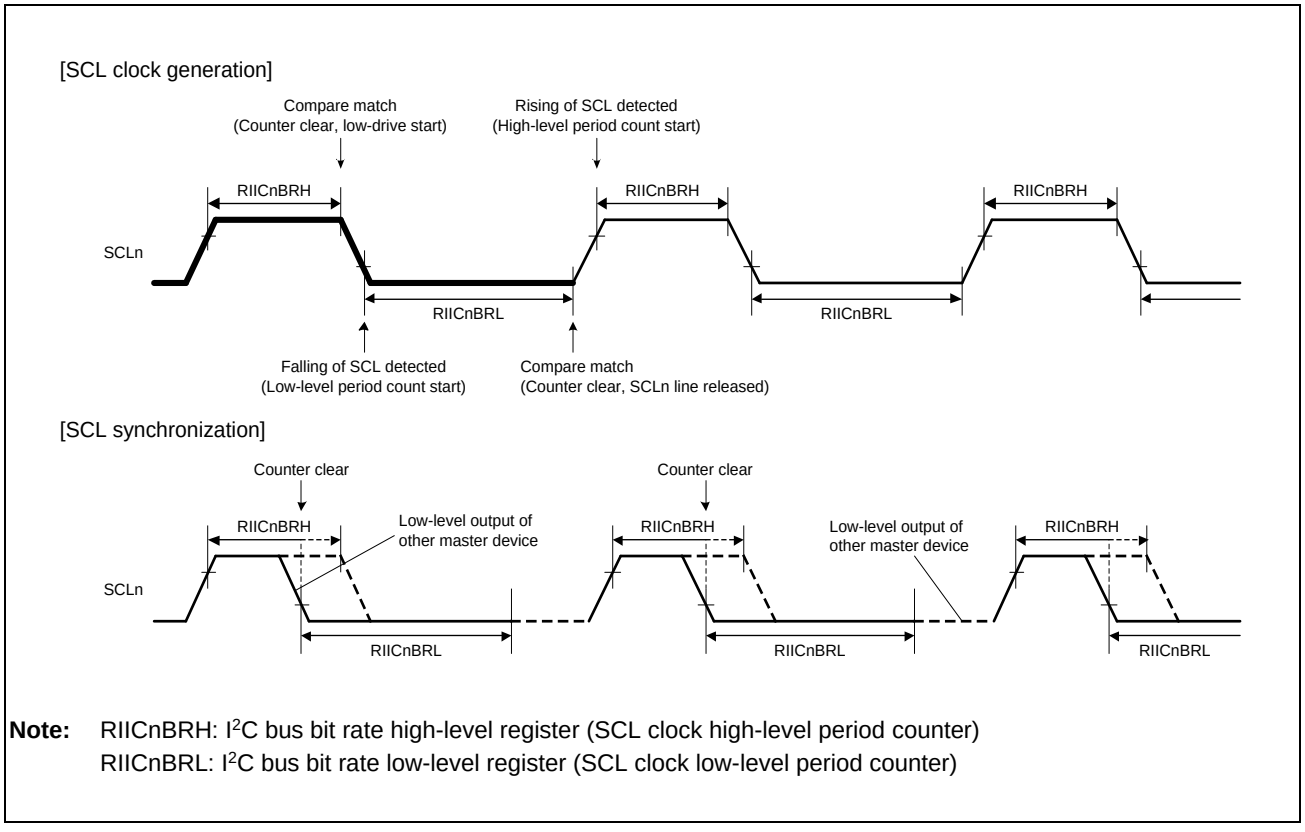

Figure 26.20 Slave Receive Operation Timing (2) (when RDRFS = 0)

## 26.6 SCL Synchronization Circuit

In generation of the SCL (clock) signal, the RIIC starts counting out the value for width at high level specified in RIICnBRH when it detects a rising edge on the SCL line and drives the SCL line low once counting of the width at high level is complete. When the RIIC detects the falling edge of the SCL line, it starts counting out the width at low level period specified in RIICnBRL, and then stops driving the SCL line (releases the line) once counting of the width at low level is complete. The SCL (clock) signal is thus generated.

If multiple master devices are connected to the $I^2C$ bus, a collision of SCL signals may arise due to contention with another master device. In such cases, the master devices have to synchronize their SCL signals. Since this synchronization of SCL signals must be bit by bit, the RIIC is equipped with a facility (the SCL synchronization circuit) to obtain bit-by-bit synchronization of the SCL clock signals by monitoring the SCL line during communication.

When the RIIC has detected a rising edge on the SCL line and thus started counting out the width at high level specified in RIICnBRH, and the level on the SCL line falls because an SCL signal is being generated by another master device, the RIIC stops counting when it detects the falling edge, drives the level on the SCL line low, and starts counting out the width at low level specified in RIICnBRL. When the RIIC finishes counting out the width at low level, it stops driving the SCL line to the low level (i.e. releases the line). At this time, if the width at low level of the SCL clock signal from the other master device is longer than the width at low level set in the RIIC, the width at low level of the SCL signal will be extended. Once the width at low level for the other master device has ended, the SCL signal rises because the SCL line has been released. When the RIIC finishes outputting the low-level period of the SCL clock, the SCL line is released and the SCL clock rises. That is, in cases of contention of SCL signals from more than one master, the width at high level of the SCL signal is synchronized with that of the clock having the narrower width, and the width at low level of the SCL signal is synchronized with that of the clock having the broader width. However, such synchronization of the SCL signal is only enabled when the RIICnFER.SCLE bit is set to 1.




**Note:** RIICnBRH: $I^2C$ bus bit rate high-level register (SCL clock high-level period counter)
RIICnBRL: $I^2C$ bus bit rate low-level register (SCL clock low-level period counter)

Figure 26.21 Generation and Synchronization of the SCL Signal from the RIIC

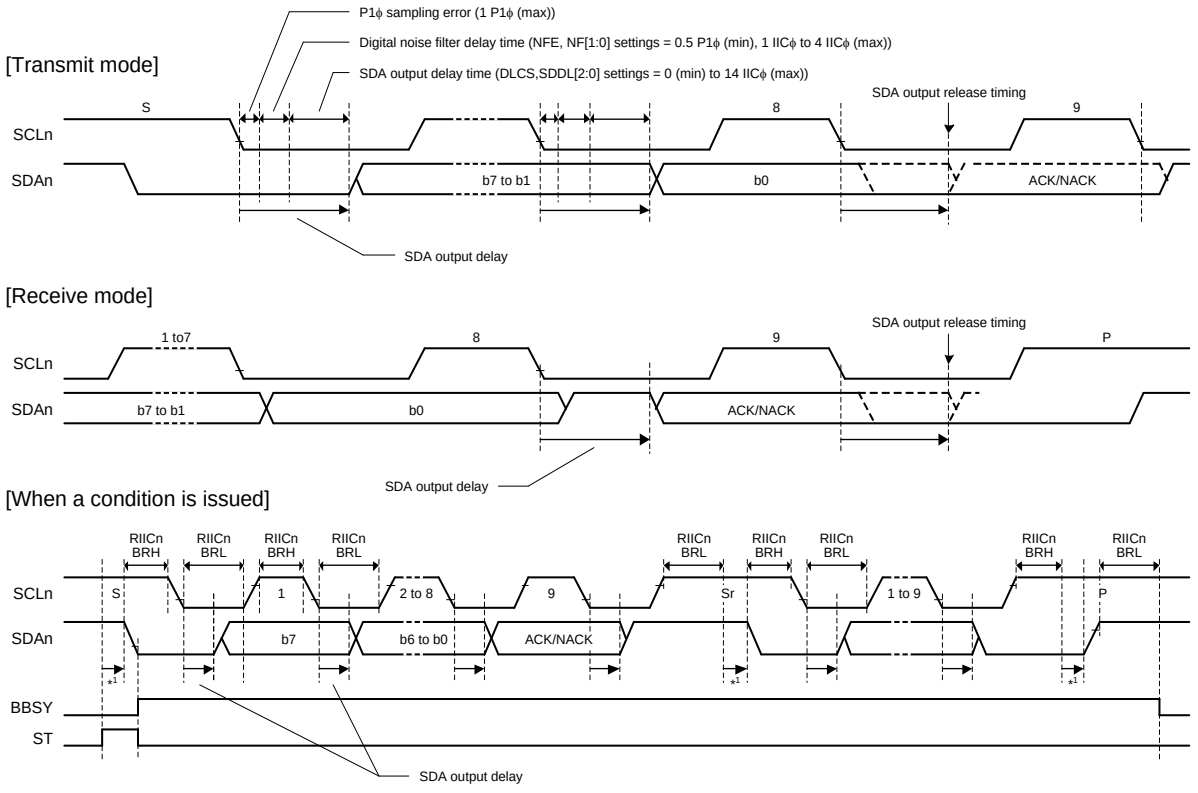
## 26.7 Facility for Delaying SDA Output

The RIIC module incorporates a facility for delaying output on the SDA line. The delay can be applied to all output (issuing of the start, restart, and stop conditions, data, and the ACK and NACK signals) on the SDA line.

With the SDA output delay facility, SDA output is delayed from detection of a falling edge of the SCL signal to ensure that the SDA signal is output within the interval over which the SCL (clock) signal is at the low level. Doing this leads to usage with the aim of preventing erroneous operation of communications devices, with the aim of satisfying the 300-ns (min.) data-hold time requirement of the SMBus specification.

The output delay facility is enabled by setting the RIICnMR2.SDDL[2:0] bits to any value other than 000b, and disabled by setting the same bits to 000b.

While the SDA output delay facility is enabled (i.e. while the SDDL[2:0] bits in IMCR2 are set to any value other than 000b), the RIICnMR2.DLCS bit selects the clock source for counting by the SDA output delay counter as the internal base clock (IICϕ) for the RIIC module or as a clock signal derived by dividing the frequency of the internal base clock by two (IICϕ/2). The counter counts the number of cycles set in the SDDL[2:0] bits in IMCR2. After counting of the set number of cycles of delay is completed, the RIIC module places the required output (start, restart, or stop condition, data, or an ACK or NACK signal) on the SDA line.




Note 1. The output is delayed by the number of cycles set by the SDDL [2:0] bits when a start (S), restart (Sr), or stop (P) condition is issued.

Figure 26.22 SDA Output Delay Facility

## 26.8 Digital Noise-Filter Circuits

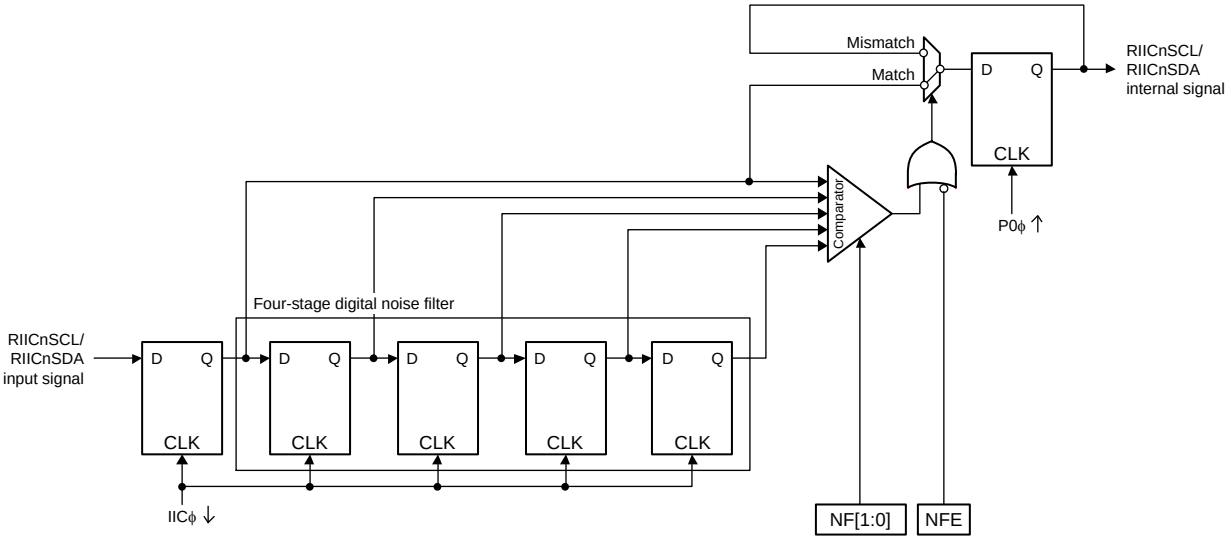
**Figure 26.23** is a block diagram of the digital noise-filter circuit. When the NFE bit in the RIICnFER register is set to 1, input to the RIICnSCL and RIICnSDA pins are conveyed to the internal circuitry through digital noise-filter circuits.

The on-chip digital noise-filter circuit of the RIIC consists of four flip-flop circuit stages connected in series and a match- detection circuit.

The number of effective stages in the digital noise filter is selected by the RIICnMR3.NF[1:0] bits. The selected number of effective stages determines the noise-filtering capability as a period from one to four IICϕ cycles.

The input signal to the RIICnSCL pin (or RIICnSDA pin) is sampled on falling edges of the IICϕ signal. When the input signal level matches the output level of the number of effective flip-flop circuit stages as selected by the RIICnMR3.NF[1:0] bits, the signal level is conveyed to the subsequent stage. If the signal levels do not match, the previous value is retained.

If the ratio between P0ϕ and IICϕ is small when the RIICnMR1.CKS[2:0] bits are set to 000b, note that the characteristics of the digital noise filter may lead to the elimination of needed signals as noise.




**Note:** NFE: Digital noise filter circuit enable bit
NF[1:0]: Digital noise filter stage selection bits

Figure 26.23 Block Diagram of Digital Noise Filter Circuit

## 26.9 Address Match Detection

The RIIC can set three unique slave addresses in addition to the general call address and host address, and also can set 7- bit or 10-bit slave addresses.

### 26.9.1 Slave-Address Match Detection

The RIIC can set three unique slave addresses, and has a slave address detection function for each unique slave address. When the RIICnSER.SARy bit (y = 0 to 2) is set to 1, the slave addresses set in RIICnSARy (y = 0 to 2) can be detected. When the RIIC detects a match of the set slave address, the corresponding RIICnSR1.AASy flag (y = 0 to 2) is set to 1 at the rising edge of the ninth SCL clock cycle, and the RIICnSR2.RDRF flag or the RIICnSR2.TDRE flag is set to 1 by the following R/W# bit. This causes a receive data full interrupt (INTRIICRI) or transmit data empty interrupt (INTRIICTI) to be generated. The AASy flag is used to identify which slave address has been specified.

**Figure 26.24** to **Figure 26.26** show the AASy flag set timing in three cases.

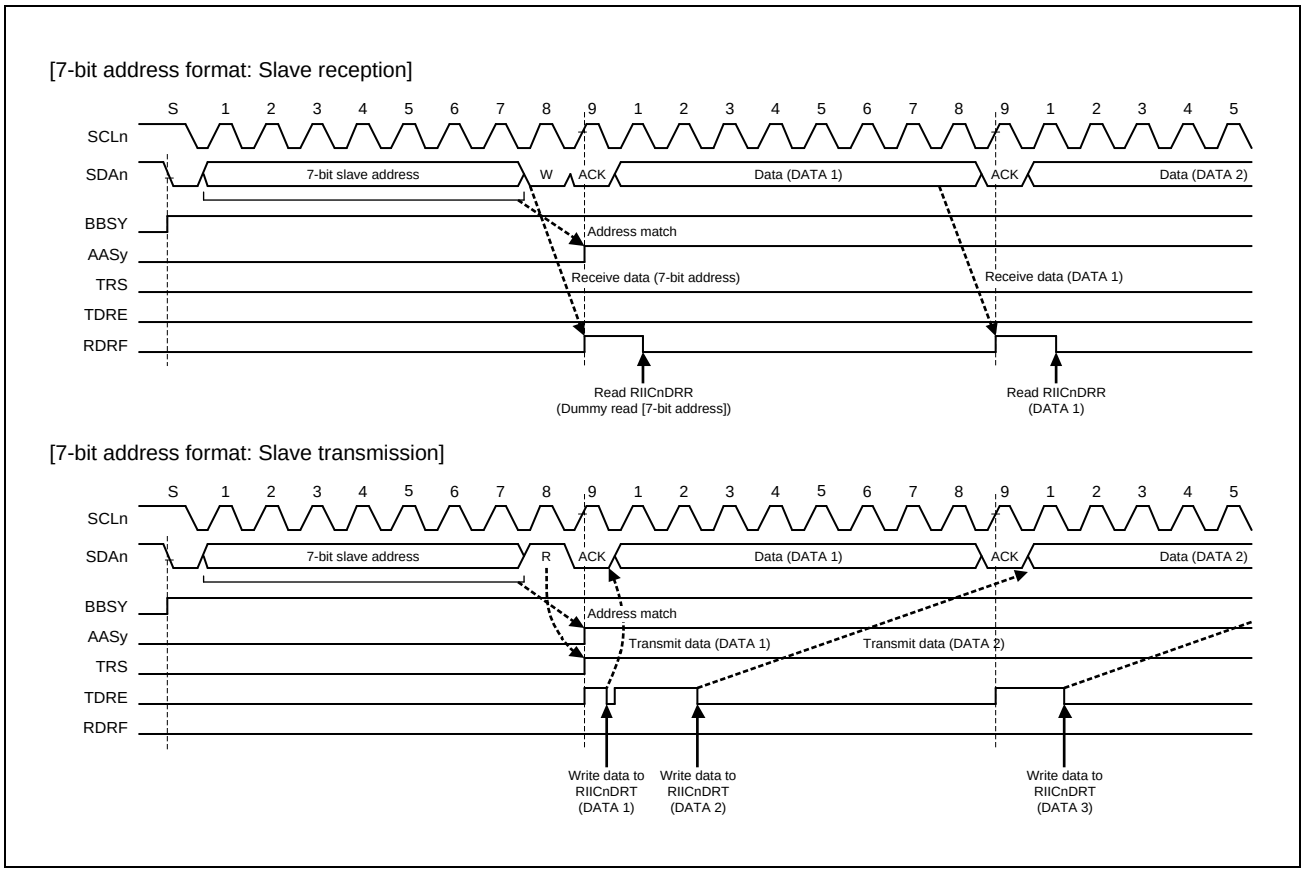



Figure 26.24 AASy Flag Set Timing with 7-Bit Address Format Selected

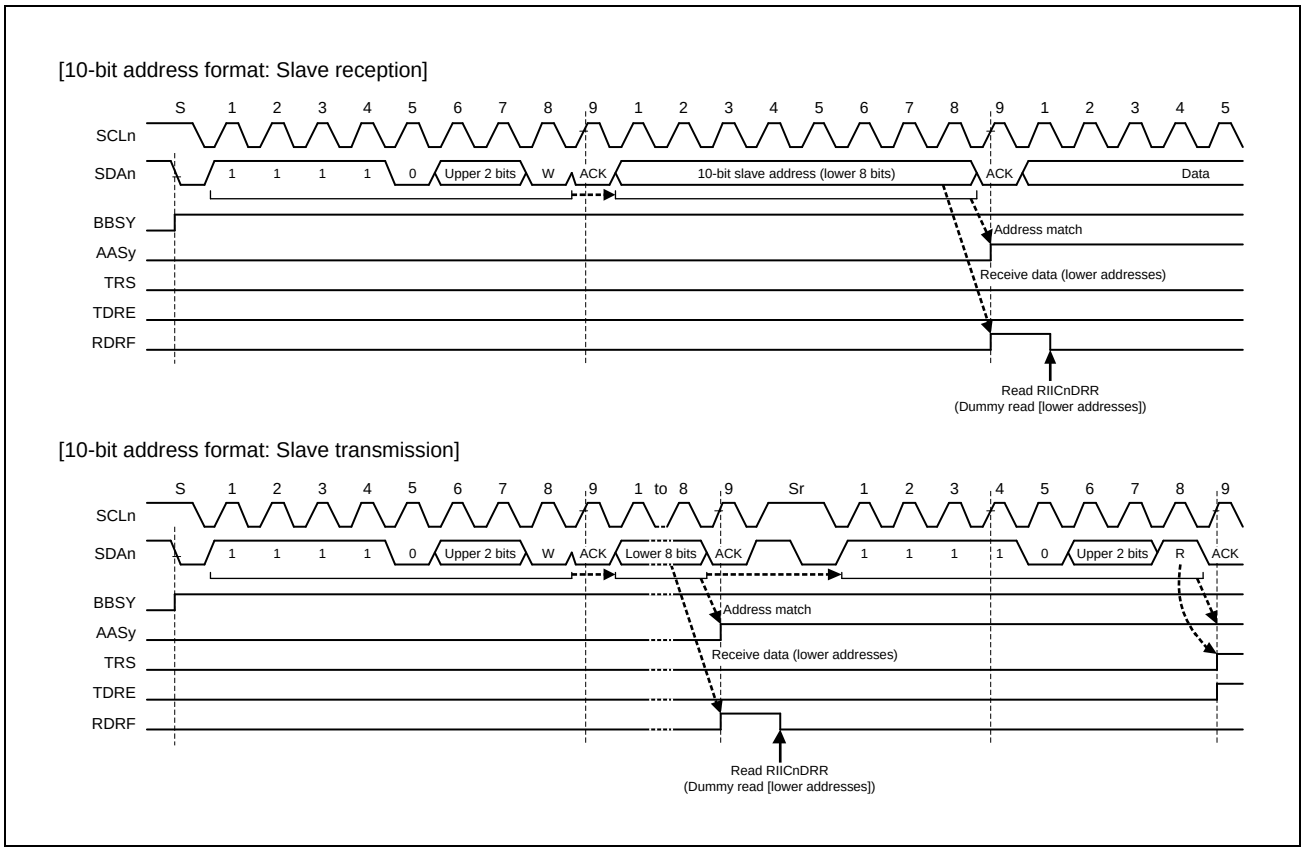



Figure 26.25 AASy Flag Set Timing with 10-Bit Address Format Selected

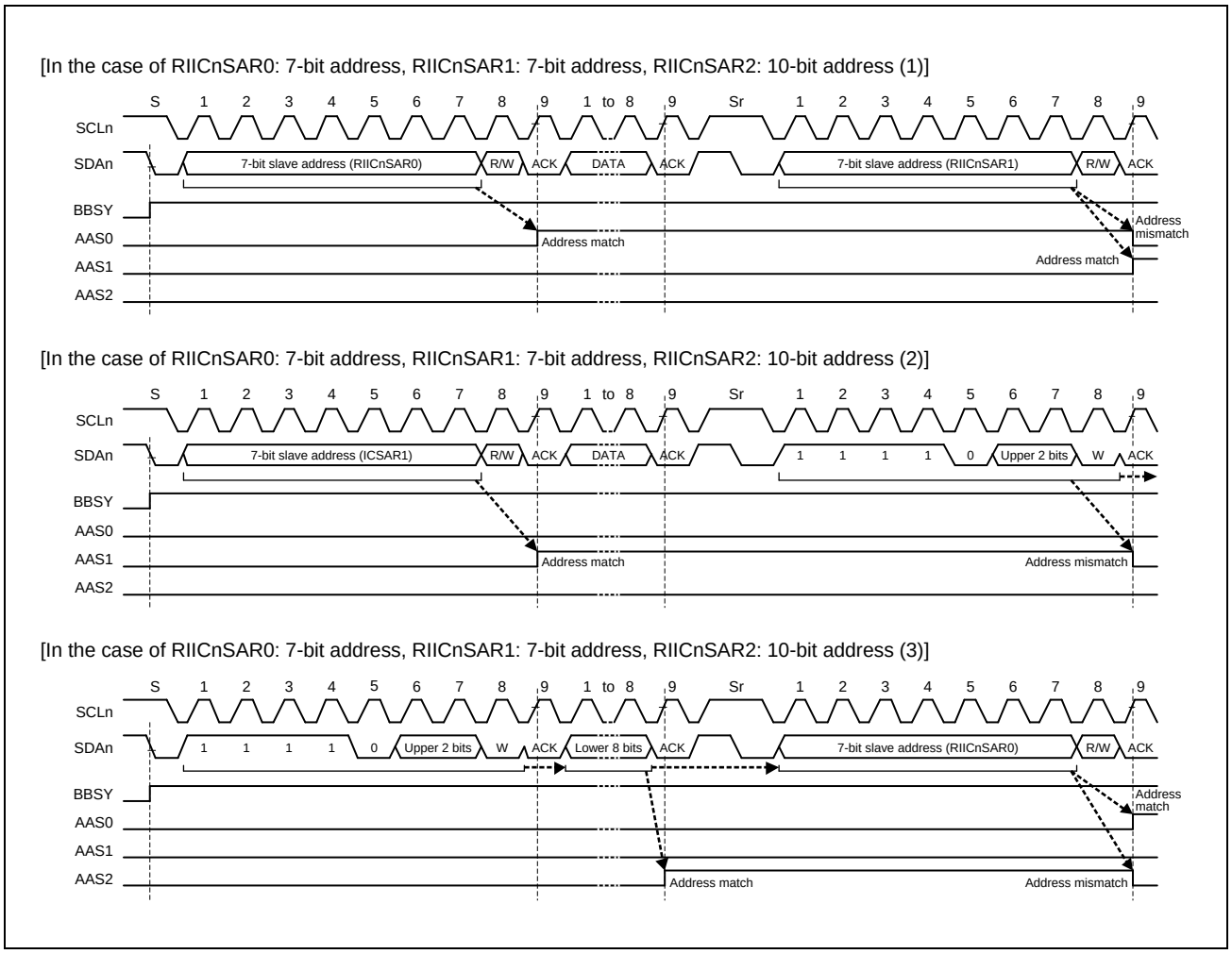



Figure 26.26 AASy Flag Set/Clear Timing with 7-Bit/10-Bit Address Formats Mixed

### 26.9.2 Detection of the General Call Address

The RIIC has a facility for detecting the general call address (0000 000b + 0 [W]). This is enabled by setting the RIICnSER.GCE bit to 1.

If the address received after a start or restart condition is issued is 0000 000b + 1[R] (start byte), the RIIC recognizes this as the address of a slave device with an “all-zero” address but not as the general call address.

When the RIIC detects the general call address, both the RIICnSR1.GCA flag and the RIICnSR2.RDRF flag are set to 1 on the rising edge of the ninth cycle of SCL clock. This leads to the generation of a receive data full interrupt (INTRIICRI). The value of the GCA flag can be confirmed to recognize that the general call address has been transmitted.

Operation after detection of the general call address is the same as normal slave receive operation.

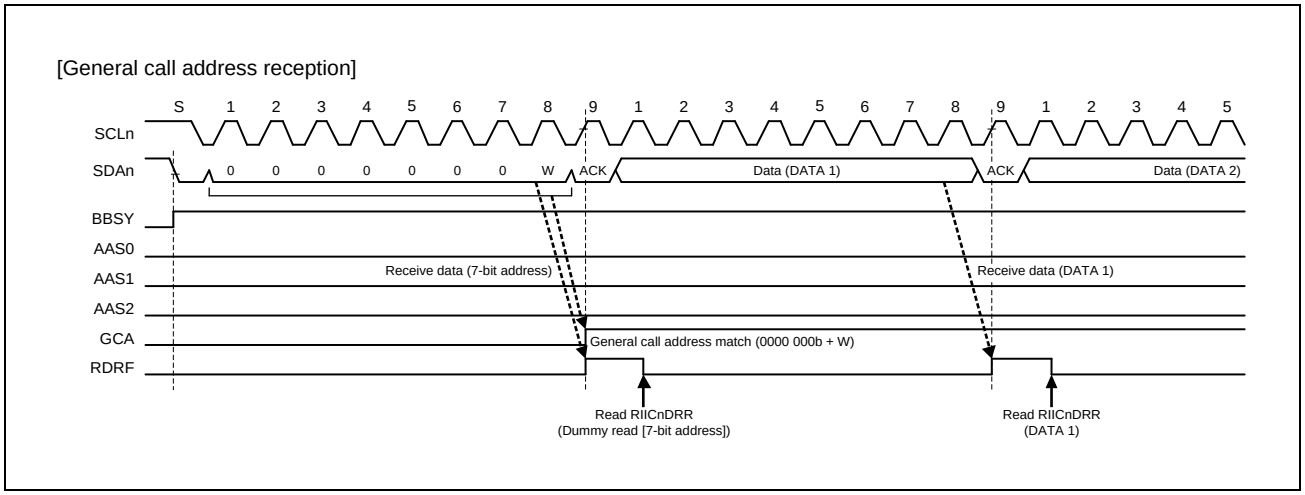



Figure 26.27 Timing of GCA Flag Setting during Reception of General Call Address

### 26.9.3 Device-ID Address Detection

The RIIC module has a facility for detecting device-ID addresses conformant with the I²C bus specification (Rev. 03). When the RIIC receives 1111 100b as the first byte after a start condition or restart condition was issued with the RIICnSER.DIDE bit set to 1, the RIIC recognizes the address as a device ID, sets the RIICnSR1.DID flag to 1 on the rising edge of the ninth SCL clock cycle when the following R/W# bit is 0, and then compares the second and subsequent bytes with its own slave address. If the address matches the value in the slave address register, the RIIC sets the corresponding RIICnSR1.AASy flag (y = 0 to 2) to 1.

After that, when the first byte received after a start or restart condition is issued matches the device ID address (1111 100b) again and the following R/W# bit is 1, the RIIC does not compare the second and subsequent bytes and sets the RIICnSR2.TDRE flag to 1.

In the device-ID address detection function, the RIIC clears the DID flag to 0 if a match with the RIIC's own slave address is not obtained or a match with the device ID address is not obtained after a match with the RIIC's own slave address and the detection of a restart condition. If the first byte after detection of a start or restart condition matches the device ID address (1111 100b) and the R/W# bit is 0, the RIIC sets the DID flag to 1 and compares the second and subsequent bytes with the RIIC's slave address. If the R/W# bit is 1, the DID flag holds the previous value and the RIIC does not compare the second and subsequent bytes. Therefore, the reception of a device-ID address can be checked by reading the DID flag after confirming that TDRE = 1.

Furthermore, prepare the device-ID fields (three bytes: 12 bits indicating the manufacturer + 9 bits identifying the part + 3 bits indicating the revision) that must be sent to the host after reception of a continuous device-ID field as normal data for transmission. For details, see I²C Bus Standard from NXP Semiconductors.

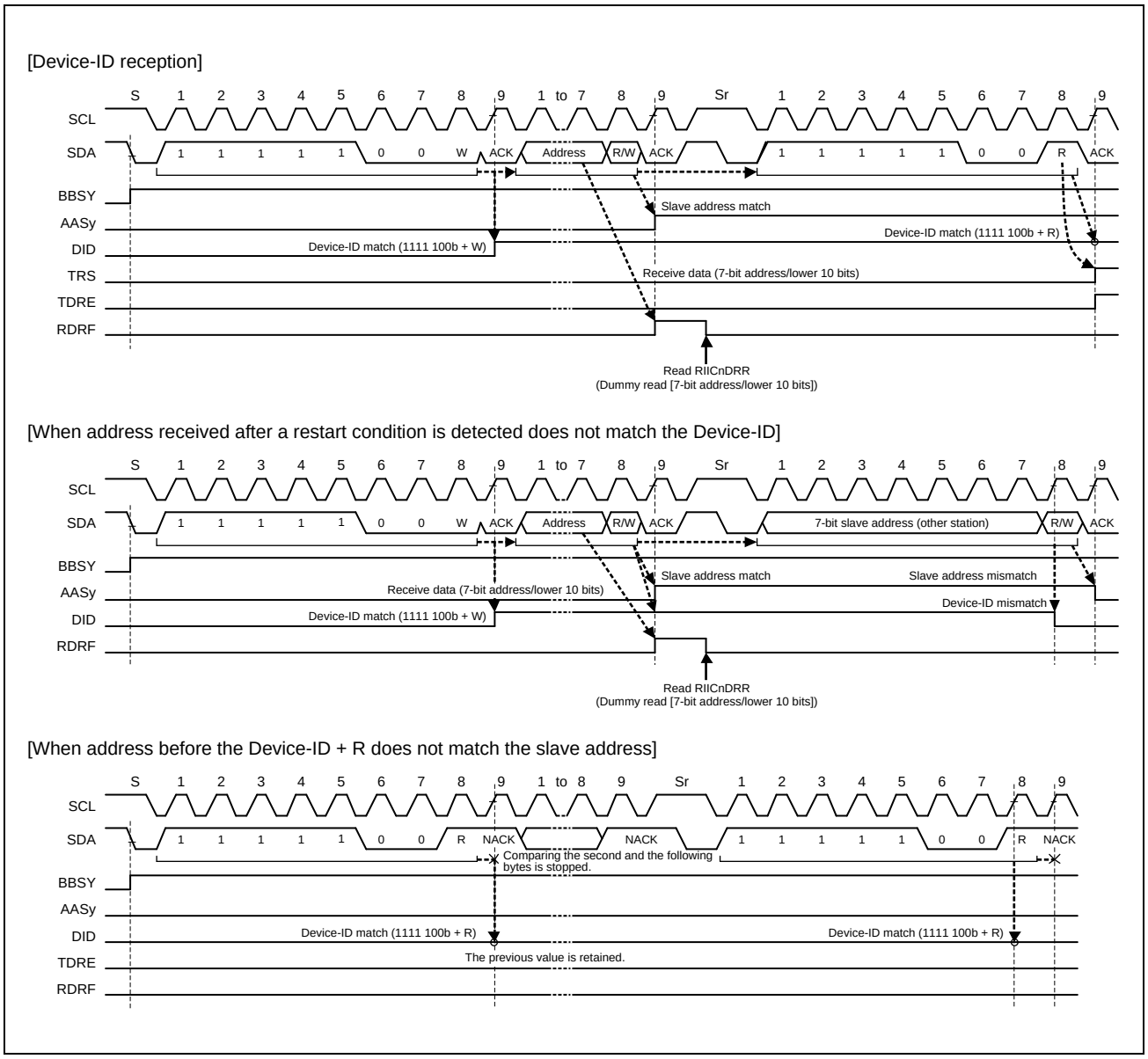



Figure 26.28 AASy/DID Flag Set/Clear Timing during Reception of Device-ID

### 26.9.4 Host Address Detection

The RIIC has a function to detect the host address while the SMBus is operating. When the RIICnSER.HOAE bit is set to 1 while the RIICnMR3.SMBS bit is 1, the RIIC can detect the host address (0001 000b) in slave receive mode (RIICnCR2.MST and TRS bits = 00b).

When the RIIC detects the host address, the RIICnSR1.HOA flag is set to 1 at the rising edge of the ninth SCL clock cycle, and at the same time, the RIICnSR2.RDRF flag is set to 1 when the R/W# bit is 0 (Wr bit). This causes a receive data full interrupt (INTRIICRI) to be generated. The HOA flag is used to recognize that the host address was sent from the smart battery or other devices.

If the bit following the host address (0001 000b) is an Rd bit (R/W# bit = 1), the RIIC can also detect the host address. After the host address is detected, the RIIC operates in the same manner as normal slave operation.

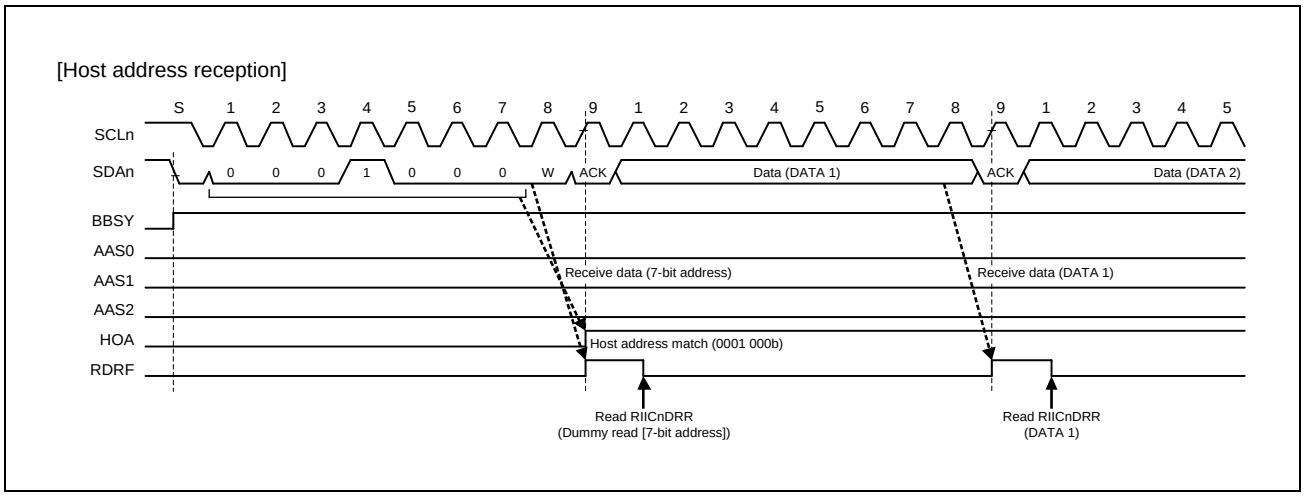



Figure 26.29 HOA Flag Set Timing during Reception of Host Address

# 26.10 Automatically Low-Hold Function for SCL

## 26.10.1 Function to Prevent Wrong Transmission of Transmit Data

If the shift register (RIICnDRS) is empty when data have not been written to the transmit data register (RIICnDRT) with the RIIC in transmission mode (RIICnCR2.TRS bit = 1), the SCL signal is automatically held at the low level over the intervals shown below. This low-hold period is extended until data for transmission have been written, which prevents the unintended transmission of erroneous data.

**<Master transmit mode>**

- Low-level interval after a start condition or restart condition is issued
- Low-level interval of one clock cycle between the ninth clock cycle of one transfer and the first clock cycle of the next

**<Slave transmit mode>**

- Low-level interval between the ninth clock cycle of one transfer and the first clock cycle of the next

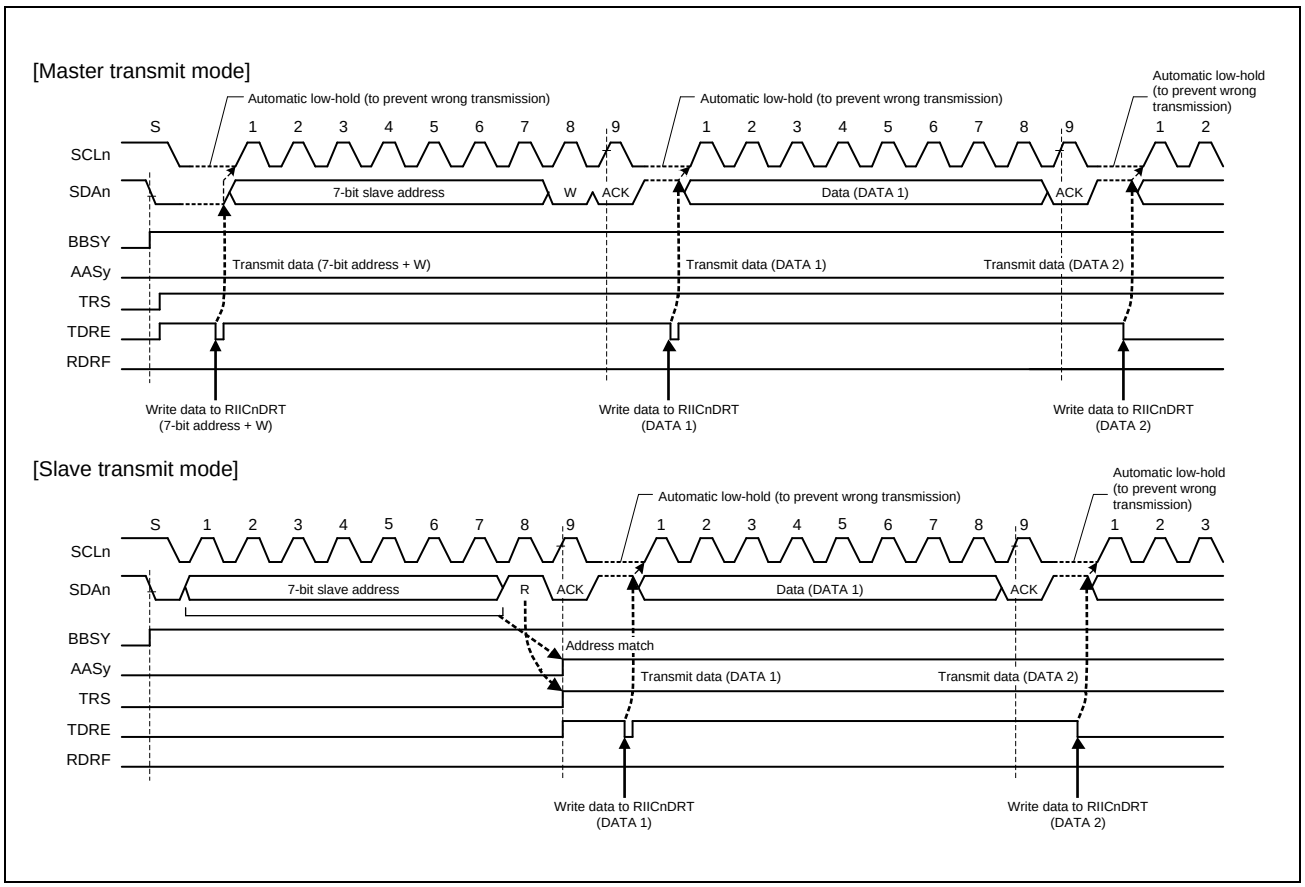



Figure 26.30 Automatic Low-Hold Operation in Transmit Mode

## 26.10.2 NACK Reception Transfer Suspension Function

The RIIC has a function to suspend transfer operation when NACK is received in transmit mode (RIICnCR2.TRS bit = 1). This function is enabled when the RIICnFER.NACKE bit is set to 1 (transfer suspension enabled). If the next transmit data has already been written (RIICnSR2.TDRE flag = 0) when NACK is received, next data transmission at the falling edge of the ninth SCL clock cycle is automatically suspended. This prevents the SDA line output level from being held low when the MSB of the next transmit data is 0.

If the transfer operation is suspended by this function (RIICnSR2.NACKF flag = 1), transmit operation and receive operation are discontinued. To restore transmit/receive operation, be sure to clear the NACKF flag to 0. In master transmit mode, clear the NACKF flag to 0 after issuing a restart condition or clear the NACKF and STOP flags to 0 after confirming that a stop condition has been issued, and then issue a start condition.

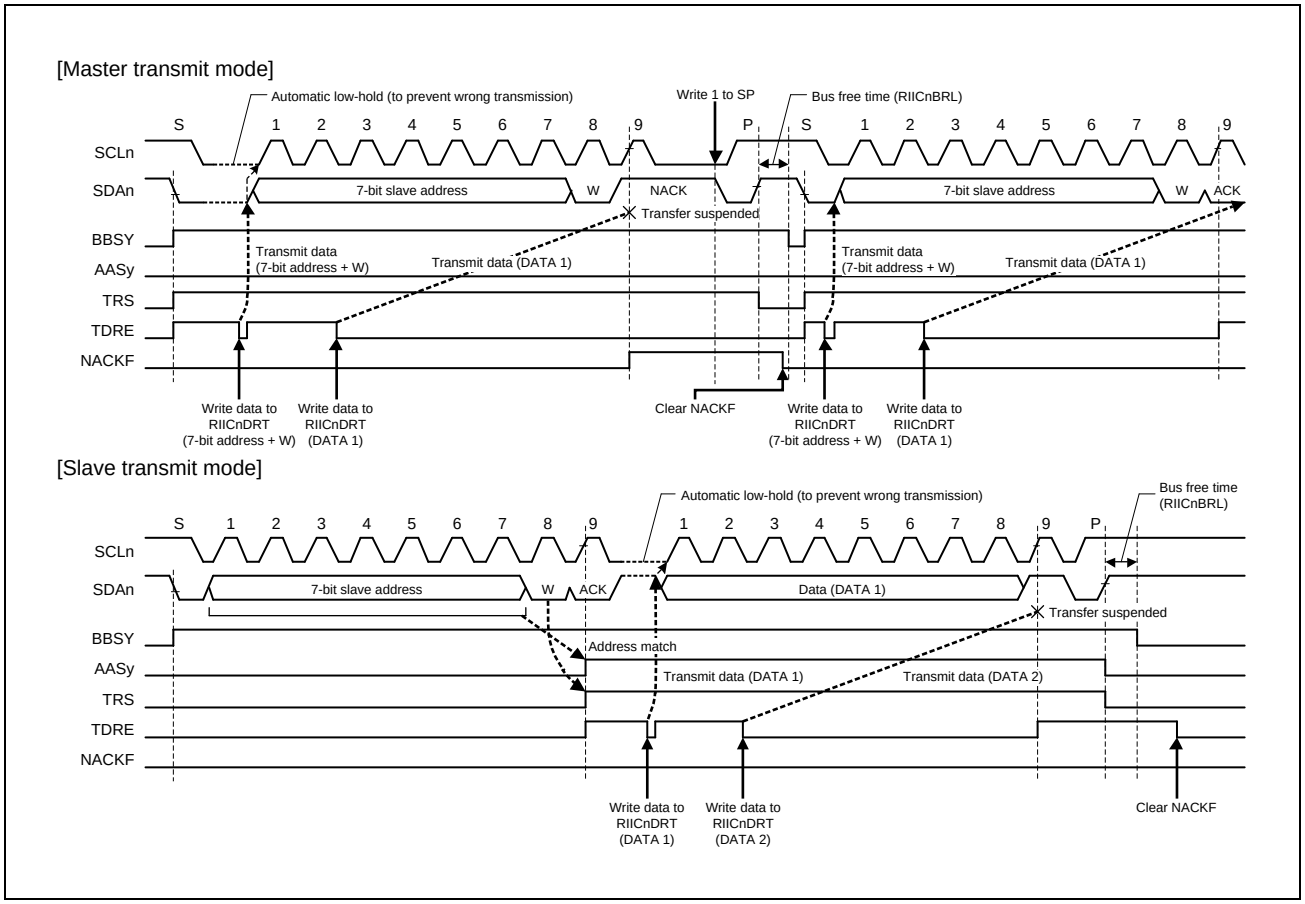



Figure 26.31 Suspension of Data Transfer when NACK is Received (NACKE = 1)

### 26.10.3 Function to Prevent Failure to Receive Data

If response processing is delayed when receive data (RIICnDRR) read is delayed for a period of one transfer frame or more with receive data full (RIICnSR2.RDRF flag = 1) in receive mode (RIICnCR2.TRS = 0), the RIIC holds the SCL line low automatically immediately before the next data is received to prevent failure to receive data.

This function to prevent failure to receive data using the automatic low-hold function is also enabled even if the read processing of the final receive data is delayed and, in the meantime, the RIIC's own slave address is designated after a stop condition is issued. This function does not disturb other communication because the RIIC does not hold the SCL line low when a mismatch with its own slave address occurs after a stop condition is issued.

Sections in which the SCL line is held low can be selected with a combination of the RIICnMR3.WAIT and RDRFS bits.

#### (1) One-Byte Receive Operation and Automatic Low-Hold Function Using the WAIT Bit

When the RIICnMR3.WAIT bit is set to 1, the RIIC performs one-byte receive operation using the WAIT bit function. Furthermore, when the RIICnMR3.RDRFS bit is 0, the RIIC automatically sends the RIICnMR3.ACKBT bit value for the acknowledge bit in the period from the falling edge of the eighth SCL clock cycle to the falling edge of the ninth SCL clock cycle, and automatically holds the SCL line low at the falling edge of the ninth SCL clock cycle using the WAIT bit function. This low-hold is released by reading data from RIICnDRR, which enables bytewise receive operation.

The WAIT bit function is enabled for receive frames after a match with the RIIC's own slave address (including the general call address and host address) is obtained in master receive mode or slave receive mode.

#### (2) One-Byte Receive Operation (ACK/NACK Transmission Control) and Automatic Low-Hold Function Using the RDRFS Bit

When the RIICnMR3.RDRFS bit is set to 1, the RIIC performs one-byte receive operation using the RDRFS bit function. When the RIICnSR2.RDRFS bit is set to 1, the RDRF flag (receive data full) in RIICnSR2 is set to 1 at the rising edge of the eighth SCL clock cycle, and the SCL line is automatically held low at the falling edge of the eighth SCL clock cycle. This lowhold is released by writing a value to the RIICnMR3.ACKBT bit, but cannot be released by reading data from RIICnDRR, which enables receive operation by the ACK/NACK transmission control according to the data received in byte units.

The RDRFS bit function is enabled for receive frames after a match with the RIIC's own slave address (including the general call address and host address) is obtained in master receive mode or slave receive mode.

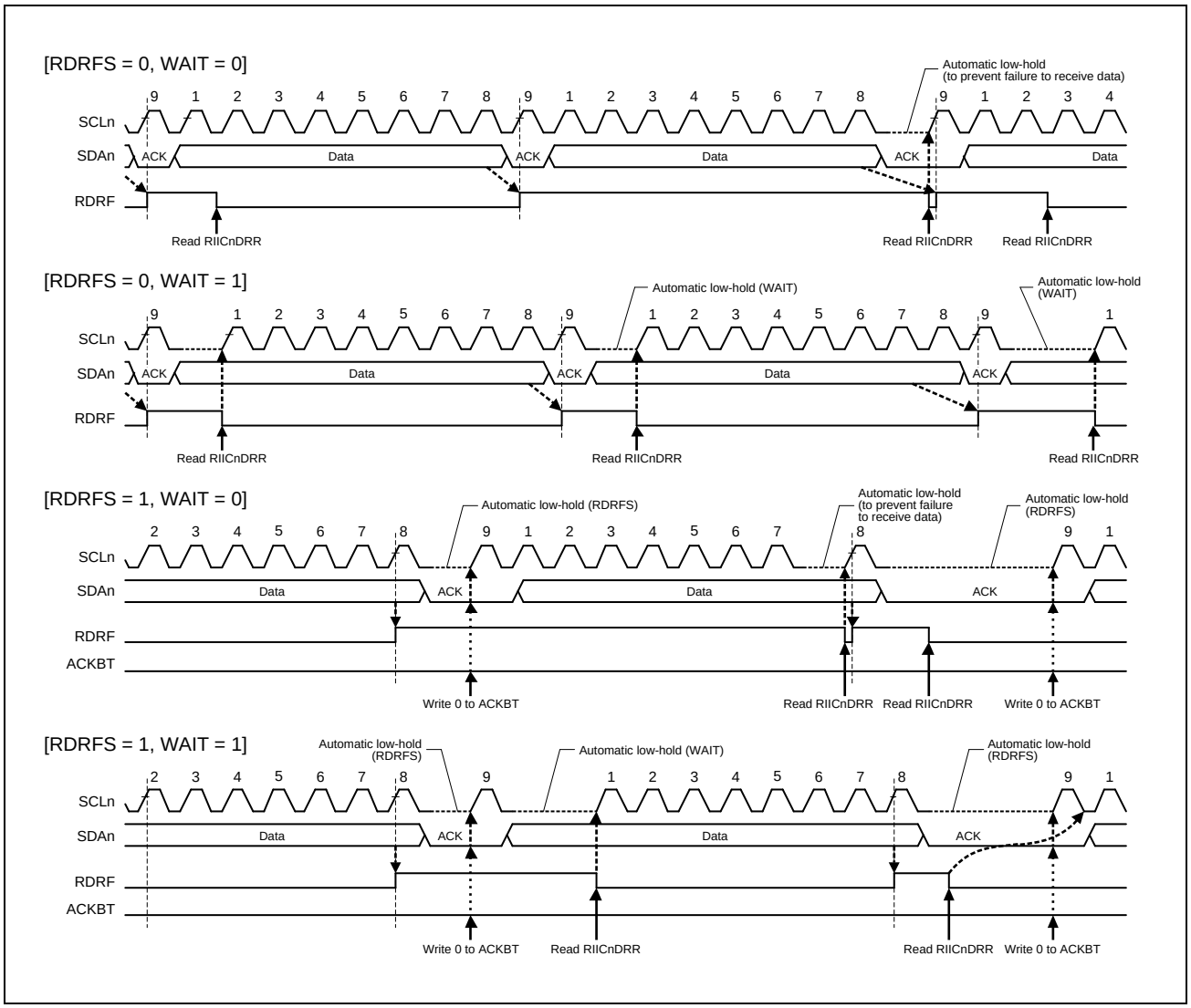



Figure 26.32 Automatic Low-Hold Operation in Receive Mode (Using RDRFS and WAIT Bits)

## 26.11 Arbitration-Lost Detection Functions

In addition to the normal arbitration-lost detection function defined by the $I^2C$ bus standard, the RIIC has functions to prevent double-issue of a start condition, to detect arbitration-lost during transmission of NACK, and to detect arbitration-lost in slave transmit mode.

### 26.11.1 Master Arbitration-Lost Detection (MALE Bit)

The RIIC drives the SDA line low to issue a start condition. However, if the SDA line has already been driven low by another master device issuing a start condition, the RIIC regards its own issuing of a start condition as an error and considers this a loss in arbitration, so priority is given to transfer by the other master device. Similarly, if a request to issue a start condition is made by setting the RIICnCR2.ST bit to 1 while the bus is busy (RIICnCR2.BBSY flag = 1), the RIIC regards this as a double-issuing-of-start-condition error and considers itself to have lost in arbitration, thus preventing a failure of transfer due to issuing of a start condition while transfer is in progress.

When a start condition is issued successfully, if the data for transmission including the address bits (i.e. the internal SDA output level) and the level on the SDA line do not match (the high output as the internal SDA output; i.e. the SDA pin is in the high-impedance state) and the low level is detected on the SDA line, the RIIC loses in arbitration.

After a loss in arbitration of mastership, the RIIC immediately enters slave receive mode. If a slave address (including the general call address) matches its own address at this time, the RIIC continues in slave operation.

A loss in arbitration of mastership is detected when the following conditions are met while the RIICnFER.MALE bit is 1 (master arbitration-lost detection enabled).

[Master arbitration-lost conditions]

- Non-matching of the internal level for output on SDA and the level on the SDA line after a start condition was issued by setting the RIICnCR2.ST bit to 1 while the RIICnCR2.BBSY flag was cleared to 0 (erroneous issuing of a start condition)
- Setting of the RIICnCR2.ST bit to 1 (start condition double-issue error) while the RIICnCR2.BBSY flag is set to 1
- When the transmit data excluding acknowledge (internal SDA output level) does not match the level on the SDA line in master transmit mode (RIICnCR2.MST and TRS bits = 11b)

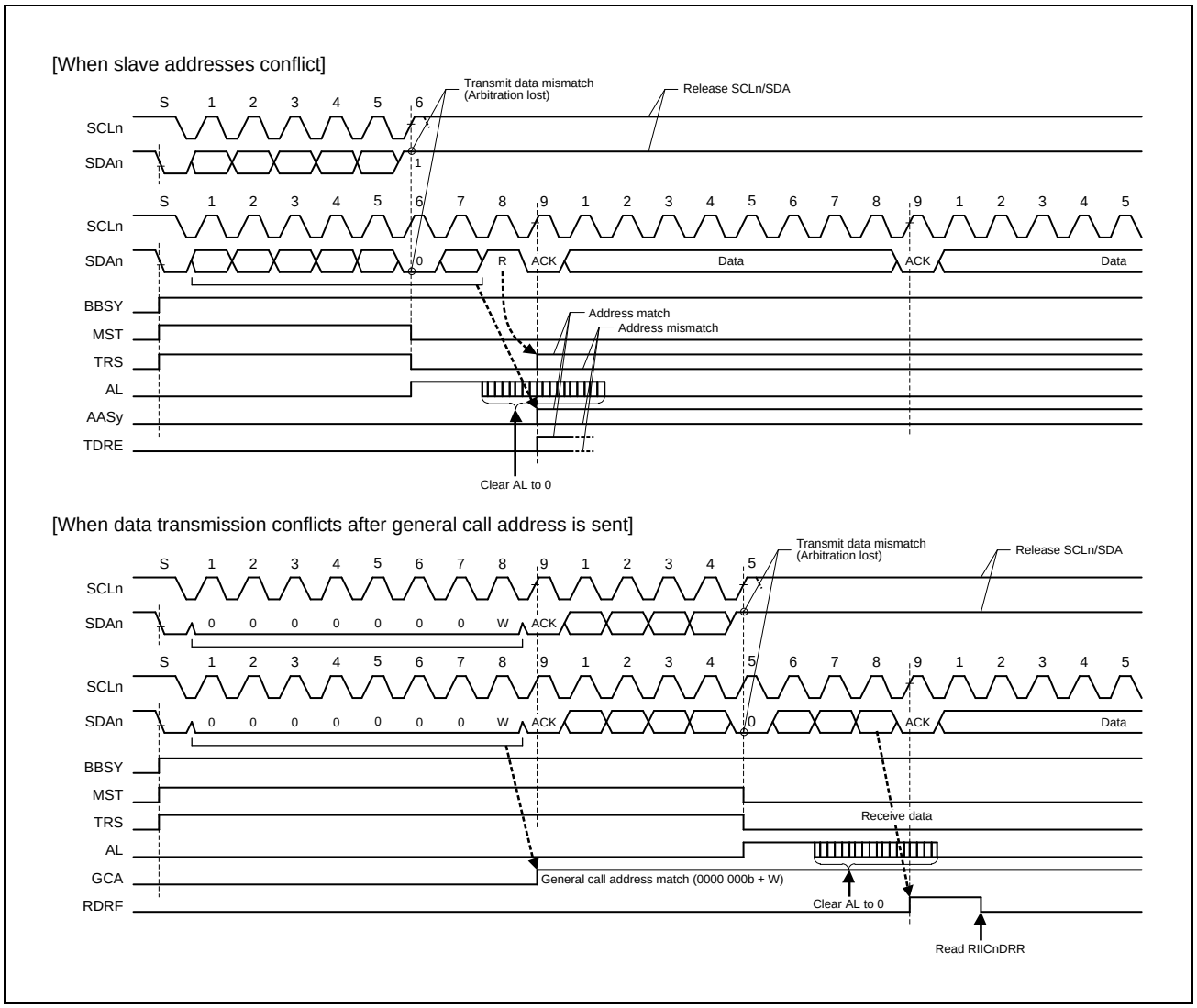



Figure 26.33 Examples of Master Arbitration-Lost Detection (MALE = 1)

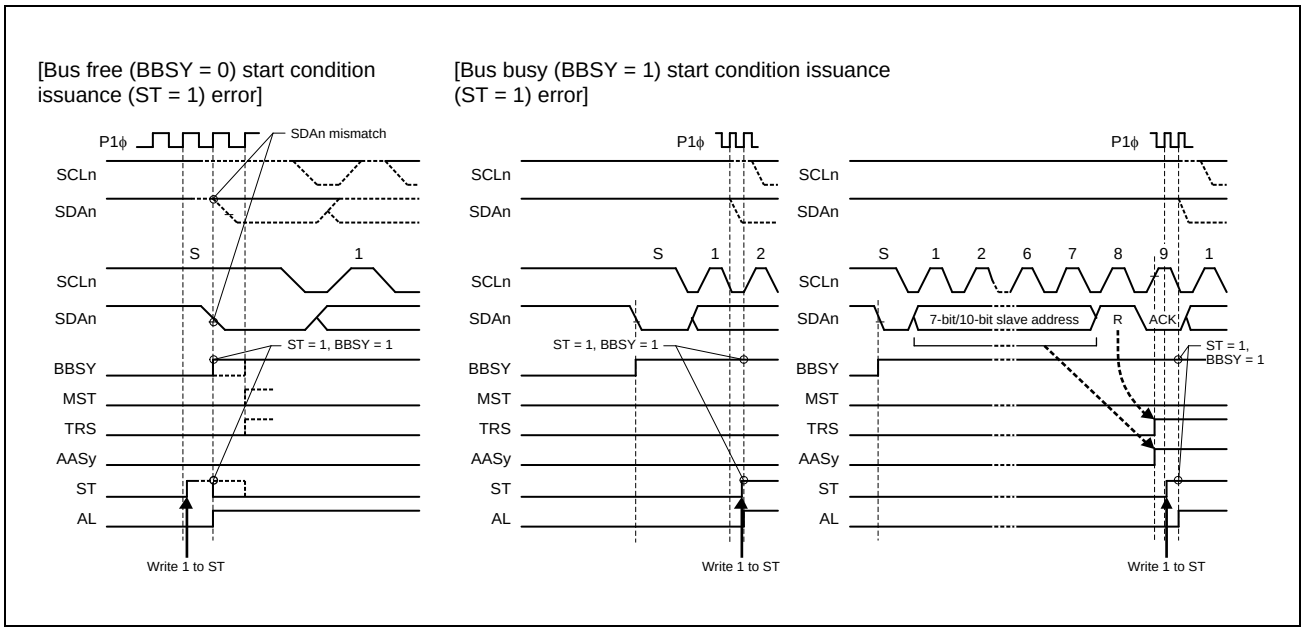



Figure 26.34 Arbitration-Lost when a Start Condition is Issued (MALE = 1)

### 26.11.2 Function to Detect Loss of Arbitration during NACK Transmission (NALE Bit)

The RIIC has a function to cause arbitration to be lost if the internal SDA output level does not match the level on the SDA line (the high output as the internal SDA output; i.e. the SDA pin is in the high-impedance state) and the low level is detected on the SDA line during transmission of NACK in receive mode. Arbitration is lost due to a conflict of NACK transmission and ACK transmission when two or more master devices receive data from the same slave device simultaneously in a multi-master system. Such conflict occurs when multiple master devices send/receive the same information through a single slave device. **Figure 26.35** shows an example of arbitration-lost detection during transmission of NACK.

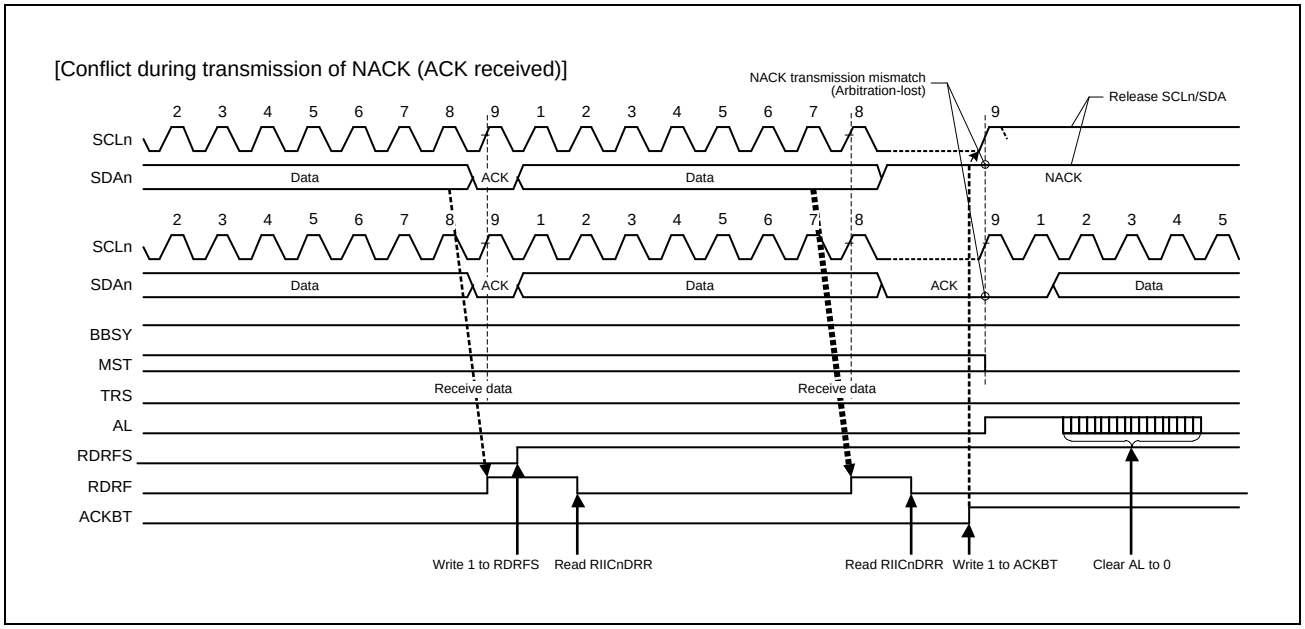



Figure 26.35 Example of Arbitration-Lost Detection during Transmission of NACK (NALE = 1)

The following explains arbitration-lost detection using an example where two master devices (master A and master B) and a single slave device are connected through the bus. In this example, master A receives two bytes of data from the slave device, and master B receives four bytes of data from the slave device.

If master A and master B access the slave device simultaneously, because the slave address is identical, arbitration is not lost in both master A and master B during access to the slave device. Therefore, both master A and master B recognize that they have obtained the bus mastership and operate as such. Here, master A sends NACK when it has received two final bytes of data from the slave device. Meanwhile, master B sends ACK because it has not received necessary four bytes of data. At this time, the NACK transmission from master A and the ACK transmission from master B conflict. In general, if a conflict like this occurs, master A cannot detect ACK transmitted by master B and issues a stop condition. Therefore, the issuance of the stop condition conflicts with the SCL clock output of master B, which disturbs communication.

When the RIIC receives ACK during transmission of NACK, it detects a defeat in conflict with other master devices and causes arbitration to be lost.

If arbitration is lost during transmission of NACK, the RIIC enters slave receive mode. This prevents a stop condition from being issued, preventing a communication failure on the bus.

Similarly, in the ARP command processing of SMBus, the function to detect loss of arbitration during transmission of NACK is also available for eliminating the extra clock cycle processing (such as H’FF transmission processing)

necessary if the UDID (Unique Device Identifier) of assign address does not match in the Get UDID (general) processing after the Assign address command.

The RIIC detects arbitration-lost during transmission of NACK when the following condition is met with the RIICnFER.NALE bit set to 1 (arbitration-lost detection during NACK transmission enabled).

[Condition for arbitration-lost during NACK transmission]

When the internal SDA output level does not match the SDA line (ACK is received) during transmission of NACK (RIICnMR3.ACKBT bit = 1).

### 26.11.3 Slave Arbitration-Lost Detection (SALE Bit)

The RIIC has a function to cause arbitration to be lost if the data for transmission (i.e. the internal SDA output level) and the level on the SDA line do not match (the high output as the internal SDA output; i.e. the SDA pin is in the highimpedance state) and the low level is detected on the SDA line in slave transmit mode. This arbitration-lost detection function is mainly used when transmitting a UDID (Unique Device Identifier) over an SMBus.

When it loses slave arbitration, the RIIC enters slave receive mode. This function can detect conflicts of data during transmission of UDIDs over an SMBus and eliminates subsequent redundant processing (processing for the transmission of H'FF).

The RIIC detects slave arbitration-lost when the following condition is met with the RIICnFER.SALE bit set to 1 (slave arbitration-lost detection enabled). [Condition for slave arbitration-lost]

When transmit data excluding acknowledge (internal SDA output level) does not match the SDA line in slave transmit mode (RIICnCR2.MST and TRS bits = 01b)

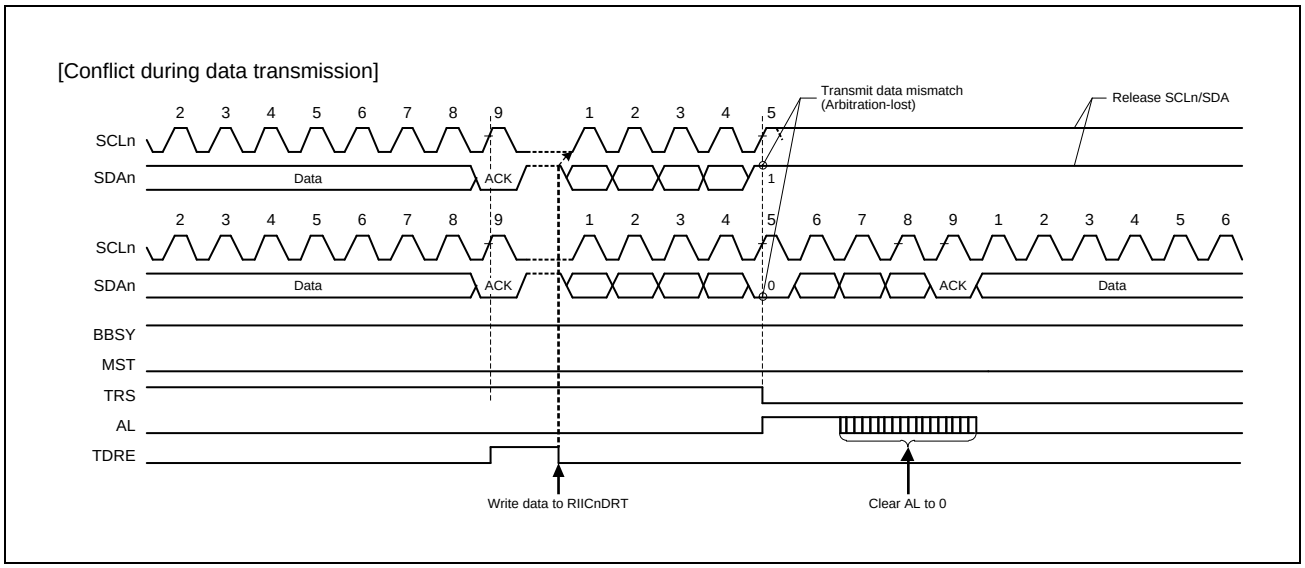



Figure 26.36 Example of Slave Arbitration-Lost Detection (SALE = 1)

## 26.12 Start Condition/Restart Condition/Stop Condition Issuing Function

### 26.12.1 Issuing a Start Condition

The RIIC issues a start condition when the RIICnCR2.ST bit is set to 1.

When the ST bit is set to 1, a start condition issuance request is made and the RIIC issues a start condition when the RIICnCR2.BBSY flag is 0 (bus free). When a start condition is issued normally, the RIIC automatically shifts to the master transmit mode.

A start condition is issued in the following sequence.

[Start condition issuance]

- Drive the SDA line low (high level to low level).
- Ensure the time set in RIICnBRH and the start condition hold time.
- Drive the SCL line low (high level to low level).
- Detect low level of the SCL line and ensure the low-level period of SCL line set in RIICnBRL.

### 26.12.2 Issuing a Restart Condition

The RIIC issues a restart condition when the RIICnCR2.RS bit is set to 1.

When the RS bit is set to 1, a restart condition issuance request is made and the RIIC issues a restart condition when the RIICnCR2.BBSY flag is 1 (bus busy) and the RIICnCR2.MST bit is 1 (master mode).

A restart condition is issued in the following sequence.

[Restart condition issuance]

- Release the SDA line.
- Ensure the low-level period of SCL line set in RIICnBRL.
- Release the SCL line (low level to high level).
- Detect a high level of the SCL line and ensure the time set in RIICnBRL and the restart condition setup time.
- Drive the SDA line low (high level to low level).
- Ensure the time set in RIICnBRH and the restart condition hold time.
- Drive the SCL line low (high level to low level).
- Detect a low level of the SCL line and ensure the low-level period of SCL line set in RIICnBRL.

**CAUTION**

To issue a restart condition request, set RIICnSR2.RS to 1 after setting RIICnSR2.START to 0. After checking that RIICnSR2.START is 1, set RIICnSR2.START to 0, and then write the slave address to RIICnDRT.

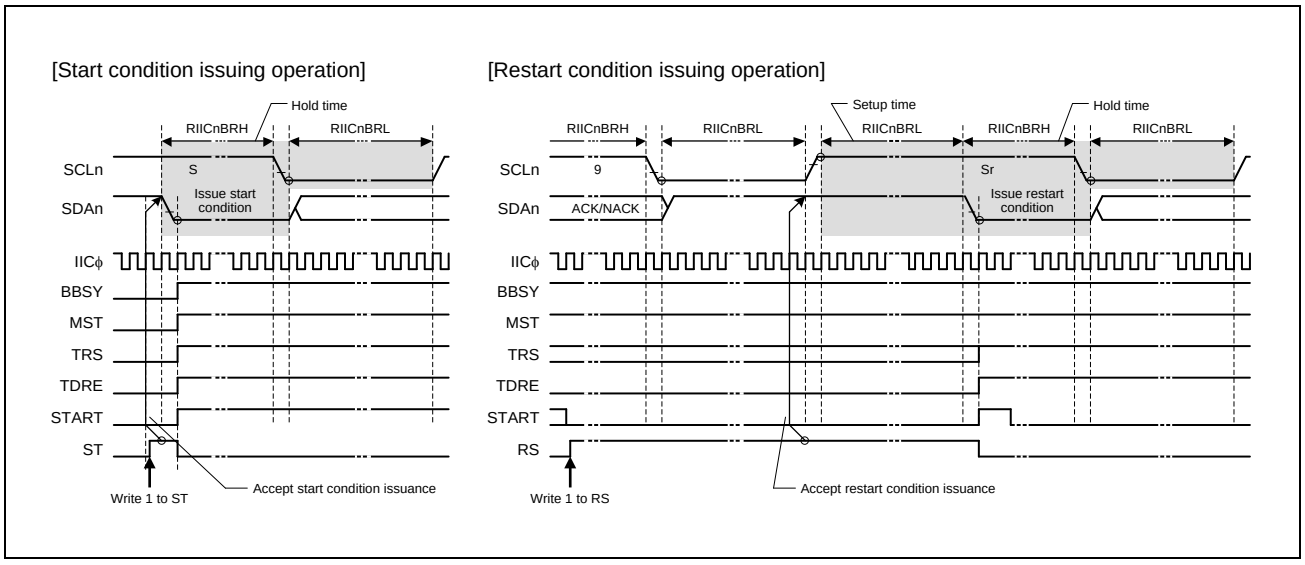



Figure 26.37 Start Condition/Restart Condition Issue Timing (ST and RS Bits)

### 26.12.3 Issuing a Stop Condition

The RIIC issues a stop condition when the RIICnCR2.SP bit is set to 1.

When the SP bit is set to 1, a stop condition issuance request is made and the RIIC issues a stop condition when the RIICnCR2.BBSY flag is 1 (bus busy) and the RIICnCR2.MST bit is 1 (master mode).

A stop condition is issued in the following sequence.

[Stop condition issuance]

- Drive the SDA line low (high level to low level).
- Ensure the low-level period of SCL line set in RIICnBRL.
- Release the SCL line (low level to high level).
- Detect a high level of the SCL line and ensure the time set in RIICnBRH and the stop condition setup time.
- Release the SDA line (low level to high level).
- Ensure the time set in RIICnBRL and the bus free time.
- Clear the BBSY flag to 0 (to release the bus mastership).

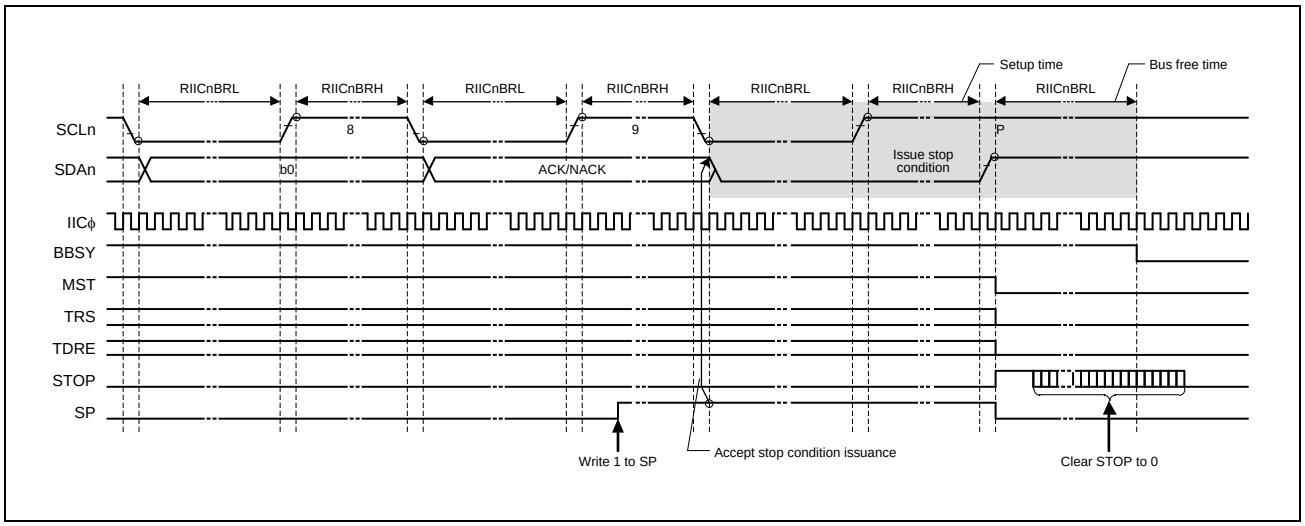



Figure 26.38 Stop Condition Issue Timing (SP Bit)

## 26.13 Bus Hanging

If the clock signals from the master and slave devices go out of synchronization due to noise or other factors, the $I^2C$ bus might hang with a fixed level on the SCL line and/or SDA line.

As measures against the bus hanging, the RIIC has a timeout function to detect hanging by monitoring the SCL line, a function for the output of an extra SCL clock cycle to release the bus from a hung state due to clock signals being out of synchronization, and the RIIC/internal reset function.

By checking the RIICnCR1.SCLO, SDAO, SCLI, and SDAI bits, it is possible to see whether the RIIC or its partner in communications is placing the low level on the SCL or SDA lines.

### 26.13.1 Timeout Function

The RIIC has the timeout function to detect an abnormality that the SCL line is held for a certain period of time. The RIIC can detect an abnormal bus state by monitoring that the SCL line is held low or high for a predetermined time. The timeout function monitors the SCL line state and counts the low-level period or high-level period using the internal counter. The timeout function resets the internal counter each time the SCL line changes (rising or falling), but continues to count unless the SCL line changes. If the internal counter overflows due to no SCL line change, the RIIC can detect the timeout and report the bus abnormality.

This timeout function is enabled when the RIICnFER.TMOE bit is 1. It detects an abnormal bus state in which the SCL line is held low or high in the following cases.

1. When the bus is busy (RIICnCR2.BBSY = 1) in master mode (RIICnCR2.MST = 1)
2. When the bus is busy (RIICnCR2.BBSY = 1) and the RIIC’s own slave address matches (RIICnSR1 is not H’00) in slave mode (RIICnCR2.MST = 0)
3. While the bus is free (RIICnCR2.BBSY = 0) and issuing of a start condition is being requested (RIICnCR2.ST = 1).

The internal counter of the timeout function works using the internal reference clock (IICϕ) set by the RIICnMR1.CKS[2:0] bits as a count source. It functions as a 16-bit counter when long mode is selected (RIICnMR2.TMOS bit = 0) or a 14-bit counter when short mode is selected (TMOS bit = 1).

The SCL line level (low/high or both levels) during which this counter is activated can be selected by the setting of the RIICnMR2.TMOH and TMOL bits. If both TMOL and TMOH bits are cleared to 0, the internal counter does not work.

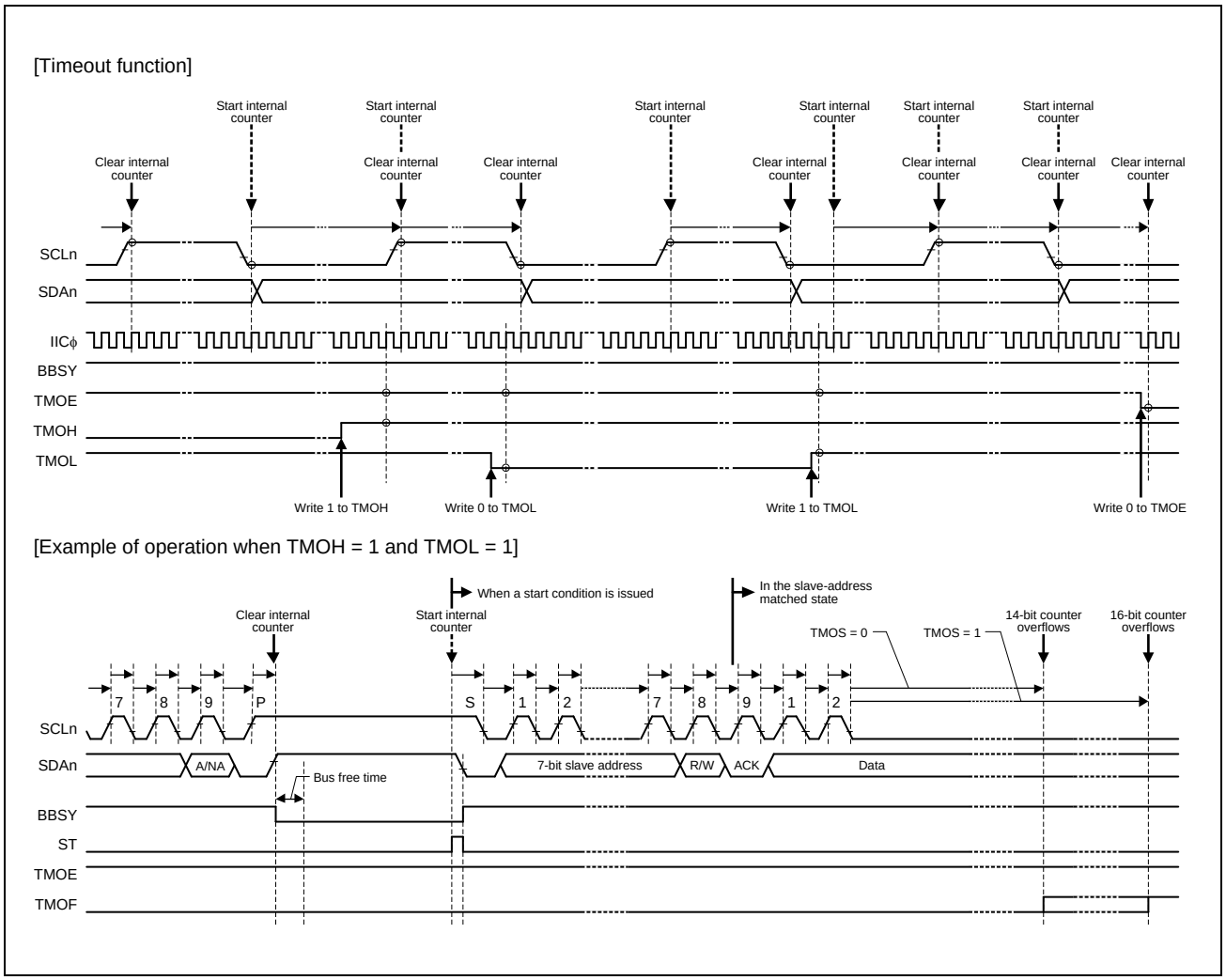



Figure 26.39 Timeout Function (TMOE, TMOS, TMOH, and TMOL Bits)

### 26.13.2 Extra SCL Clock Cycle Output Function

In master mode, the RIIC module has a facility for the output of extra SCL (clock) cycles to release the SDA line of the slave device from being held at the low level due to the master being out of synchronization with the slave device.

This function is mainly used in master mode to release the SDA line of the slave device from the state of being fixed to the low level by including extra cycles of SCL output from the RIIC with single cycles of the SCL (clock) signal as the unit in the case of a bus error where the RIIC cannot issue a stop condition because the slave device is holding the SDA line at the low level. Do not use this facility in normal situations. Using it when communications are proceeding correctly will lead to malfunctions.

When the RIICnCR1.CLO bit is set to 1 in master mode, a single cycle of the SCL clock at the frequency corresponding to the transfer rate settings (settings of the RIICnMR1.CKS[2:0] bits, and of the RIICnBRH and RIICnBRL registers) is output as an extra clock cycle. After output of this single cycle of the SCL clock, the CLO bit is automatically cleared to 0. At this point, the SCL pin output is held at a low level if the BBSY flag is 1, or the SCL pin outputs at high level if the BBSY flag is 0. Further clock cycles can consecutively be output by writing 1 to the CLO bit after having read CLO is 0 by software.

When the RIIC module is in master mode and the slave device is holding the SDA line at the low level because synchronization with the slave device has been lost due to the effects of noise, etc., the output of a stop condition is not possible. The facility for output of an extra cycle of the SCL (clock) signal can be used to output extra cycles of SCL one by one to make the slave device release the SDA line from being held at the low level, thus recovering the bus from an unusable state. Release of the SDA line by the slave device can be monitored by reading the RIICnCR1.SDAI bit. After confirming release of the SDA line by the slave device, complete communications by reissuing the stop condition.

[Output conditions for using the RIICnCR1.CLO bit]

- When the bus is free (RIICnCR2.BBSY flag = 0) or in master mode (RIICnCR2.MST bit = 1 and BBSY flag = 1)
- When the communication device does not hold the SCL line low

**Figure 26.40** shows the operation timing of the extra SCL clock cycle output function (CLO bit).

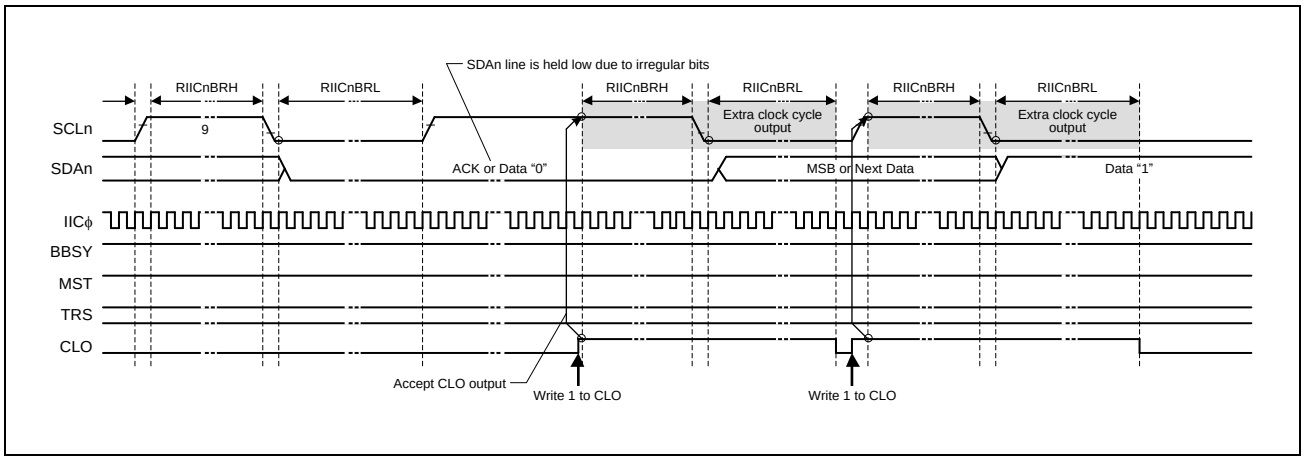



Figure 26.40 Extra SCL Clock Cycle Output Function (CLO Bit)

### 26.13.3 RIIC Reset and Internal Reset

The RIIC module incorporates a function for resetting itself. There are two types of reset. One is referred to as an RIIC reset; this initializes all registers including the RIICnCR2.BBSY flag. The other is referred to as an internal reset; this releases the RIIC from the slave-address matched state and initializes the internal counter while retaining other settings. After issuing a reset, be sure to clear the RIICnCR1.IICRST bit to 0.

Both types of reset are effective for release from bus-hung states since both restore the output state of the SCL and SDA pins to the high impedance state.

Issuing a reset during slave operation may lead to a loss of synchronization between the master device clock and the slave device clock, so avoided this where possible. Note that monitoring of the bus state, such as for the presence of a start condition, is not possible during an RIIC reset (RIICnCR1.ICE and IICRST bits = 01b).

For a detailed description of the RIIC and internal resets, see **Section 26.15, Reset Function of RIIC**.

## 26.14 SMBus Operation

The RIIC is available for data communication conforming to the SMBus (Version 2.0). To perform SMBus communication, set the RIICnMR3.SMBS bit to 1. To use the transfer rate within a range of 10 kbps to 100 kbps of the SMBus standard, set the RIICnMR1.CKS[2:0] bits, RIICnCBRH, and RIICnBRL. In addition, determine the values of the RIICnMR2.DLCS bit and the RIICnMR2.SDDL[2:0] bits to meet the data hold time specification of 300 ns or more. If the RIIC is used only as a slave device, the transfer rate setting is not necessary. When the RIIC is used only as a slave device, the transfer rate setting is not necessary, whereas the RIICnBRL needs to be set to a value at least the same as the data setup time (250 ns).

For the SMBus device default address (1100 001b), use one of the slave address registers L0 to L2 (RIICnSAR0, RIICnSAR1, and RIICnSAR2), and set the corresponding RIICnSARy.FSy bit (7-bit/10-bit address format select) (y = 0 to 2) to 0 (7-bit address format).

When transmitting the UDID (Unique Device Identifier), set the RIICnFER.SALE bit to 1 to enable the slave arbitration lost detection function.

### 26.14.1 SMBus Timeout Measurement

#### (1) Measuring timeout of slave device

The following period (timeout interval: $T_{LOW:SEXT}$) must be measured for slave devices in SMBus communication.

- From start condition to stop condition

  To measure timeout for slave devices, measure the period from start condition detection to stop condition detection with the internal timer using a start condition detection interrupt (INTRIICSTI) and stop condition detection interrupt (INTRIICSPI) of the RIIC. The measured timeout period must be within the total clock low-level period [slave device] $T_{LOW:SEXT}$: 25 ms (max.) of the SMBus standard.

  If the time measured with the internal timer exceeds the clock low-level detection timeout TTIMEOUT: 25 ms (min.) of the SMBus standard, the slave device must release the bus by writing 1 to the RIICnCR1.IICRST bit to issue an internal reset of the RIIC. When an internal reset is issued, the RIIC stops driving the bus for the SCL pin and SDA pin and make the SCL/SDA pin outputs high impedance, which releases the bus.

#### (2) Measuring timeout of master device

The following periods (timeout interval: $T_{LOW:MEXT}$) must be measured for master devices in SMBus communication.

- From start condition to acknowledge bit
- Between acknowledge bits
- From acknowledge bit to stop condition

To measure timeout for master devices, measure these periods with the internal timer using a start condition detection interrupt (INTRIICSTI), stop condition detection interrupt (INTRIICSPI), and transmit end interrupt (INTRIICTEI) or receive data full interrupt (INTRIICRI) of the RIIC. The measured timeout period must be within the total clock low-level extended period [master device] $T_{LOW:MEXT}$: 10 ms (max.) of the SMBus standard, and the total of all $T_{LOW:MEXT}$ from start condition to stop condition must be within $T_{LOW:SEXT}$: 25 ms (max.).

For the ACK receive timing (rising edge of the ninth SMBCLK clock cycle), monitor the RIICnSR2.TEND flag in master transmit mode (master transmitter) and the RIICnSR2.RDRF flag in master receive mode (master receiver). For this reason, perform bytewise transmit operation in master transmit mode, and hold the RIICnMR3.RDRFS bit 0 until the byte just before reception of the final byte in master receive mode. While the RDRFS bit is 0, the RDRF flag is set to 1 at the rising edge of the ninth SMBCLK clock cycle.

If the period measured with the internal timer exceeds the total clock low-level extended period [master device] $T_{LOW:MEXT}$: 10 ms (max.) of the SMBus standard or the total of measured periods exceeds the clock low-level detection timeout TTIMEOUT: 25 ms (min.) of the SMBus standard, the master device must stop the transaction by issuing a stop condition. In master transmit mode, immediately stop the transmit operation (writing data to RIICnDRT).

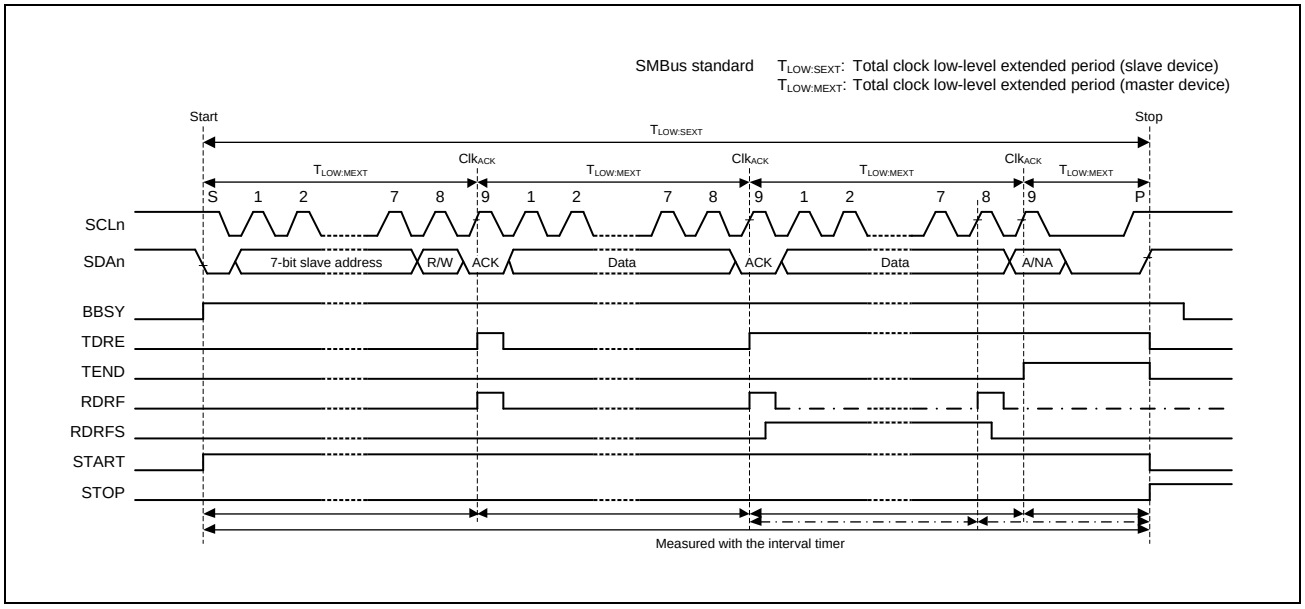



Figure 26.41 SMBus Timeout Measurement

## 26.14.2 SMBus Host Notification Protocol/Notify ARP Master

In communications over an SMBus, a slave device can temporarily act as a master device to notify the SMBus host (or ARP master) of (or request the SMBus host for) its own slave address or to request its own slave address from the SMBus host.

For this LSI to operate as an SMBus host (or ARP master), the host address (0001 000b) sent from the slave device must be detected as a slave address, so the RIIC has a function for detecting the host address. To detect the host address as a slave address, set the RIICnMR3.SMBS bit and the RIICnSER.HOAE bit to 1. Operation after the host address has been detected is the same as normal slave operation.

## 26.15 Reset Function of RIIC

The RIIC has chip reset, RIIC reset, and internal reset functions. **Table 26.13** lists the scope of each reset and reset conditions.

Table 26.13 RIIC Reset Functions (1/2)

| Register | | RIIC Reset (ICE = 0, IICRST = 1) | Internal Reset (ICE = 1, IICRST = 1) | Start Condition/Restart Condition Detection | Stop Condition Detection |
|---|---|---|---|---|---|
| RIICnCR1 | ICE | 0 | 1 | Retained | Retained |
| | IICRST | 1 | 1 | Retained | Retained |
| | CLO | Initialized | Retained | Retained | Retained |
| | SOWP | Initialized | Retained | Retained | Retained |
| | SCLO | Initialized | Initialized | Retained | Retained |
| | SDAO | Initialized | Initialized | Retained | Retained |
| | SCLI | Initialized | Retained | Retained | Retained |
| | SDAI | Initialized | Retained | Retained | Retained |
| RIICnCR2 | BBSY | Initialized | Initialized*1 | Operation | Retained |
| | MST | Initialized | Initialized | Operation (retained) | Initialized |
| | TRS | Initialized | Initialized | Operation (retained) | Initialized |
| | SP | Initialized | Initialized | Initialized | Initialized |
| | RS | Initialized | Initialized | Initialized | Initialized |
| | ST | Initialized | Initialized | Initialized | Retained |
| RIICnMR1 | CKS[2:0] | Initialized | Retained | Retained | Retained |
| | BCWP | Initialized | Retained | Retained | Retained |
| | BC[2:0] | Initialized | Initialized | Initialized | Retained |
| RIICnMR2 | | Initialized | Retained | Retained | Retained |
| RIICnMR3 | WAIT | Initialized | Retained | Retained | Retained |
| | RDRFS | Initialized | Retained | Retained | Retained |
| | ACKWP | Initialized | Retained | Retained | Retained |
| | ACKBT | Initialized | Retained | Retained | Initialized |
| | ACKBR | Initialized | Retained | Retained | Retained |
| | NF[1:0] | Initialized | Retained | Retained | Retained |
| RIICnFER | | Initialized | Retained | Retained | Retained |
| RIICnSER | | Initialized | Retained | Retained | Retained |
| RIICnIER | | Initialized | Retained | Retained | Retained |
| RIICnSR1 | DID | Initialized | Initialized | Retained | Initialized |
| | GCA | Initialized | Initialized | Retained | Initialized |
| | AAS2 | Initialized | Initialized | Retained | Initialized |
| | AAS1 | Initialized | Initialized | Retained | Initialized |
| | AAS0 | Initialized | Initialized | Retained | Initialized |

Table 26.13 RIIC Reset Functions (2/2)

| Register | | RIIC Reset (ICE = 0, IICRST = 1) | Internal Reset (ICE = 1, IICRST = 1) | Start Condition/Restart Condition Detection | Stop Condition Detection |
|---|---|---|---|---|---|
| RIICnSR2 | TDRE | Initialized | Initialized | Retained | Initialized |
| | TEND | Initialized | Initialized | Retained | Initialized |
| | RDRF | Initialized | Initialized | Retained | Retained |
| | NACKF | Initialized | Initialized | Retained | Retained |
| | STOP | Initialized | Initialized | Retained | Operation |
| | START | Initialized | Initialized | Operation | Initialized |
| | AL | Initialized | Initialized | Retained | Retained |
| | TMOF | Initialized | Initialized | Retained | Retained |
| RIICnSAR0, RIICnSAR1, RIICnSAR2 | | Initialized | Retained | Retained | Retained |
| RIICnBRH, RIICnBRL | | Initialized | Retained | Retained | Retained |
| RIICnDRT | | Initialized | Retained | Retained | Retained |
| RIICnDRR | | Initialized | Retained | Retained | Retained |
| RIICnDRS | | Initialized | Initialized | Retained | Retained |

Note 1. When an internal reset is applied while the bus is free after detection of a stop condition, the setting of the BBSY flag is 0 while the bus is free following de-assertion of the internal reset signal.
When an internal reset is applied while the bus is not free, the BBSY flag is not cleared.

# 27. Serial Sound Interface (SSIF-2)

## 27.1 Overview

The serial sound interface (SSIF-2) transmits and receives audio data to and from various devices that are compatible with $I^2S$ format, monaural format and TDM format.

### 27.1.1 Features

Table 27.1 Features of SSIF-2

| Item | | Description |
|---|---|---|
| Number of interfaces | | • 1 |
| Communication mode | | • Master/slave<br>• Full-duplex communication is available. |
| Communication format | | • $I^2S$ format<br>• Monaural format<br>• TDM format |
| Serial data | | • MSB first<br>• Data can be left-justified or right-justified.<br>• Data delay (one clock cycle) or no delay selectable for the period from SSI_RCK to SSI_TXD/SSI_RXD<br>• System word length: 8, 16, 24, 32, 48, 64, 128, or 256 bits<br>• Data word length: 8, 16, 18, 20, 22, 24, or 32 bits<br>• Padding polarity: Low or high |
| Bit clock (SSIBCK) | In master mode | • Clock source division ratio: 1/1, 1/2, 1/4, 1/6, 1/8, 1/12, 1/16, 1/24, 1/32, 1/48, 1/64, 1/96, and 1/128.<br>• Supply/stop of SSI_BCK is selectable while communication is halted. |
| | In slave mode | • SSI_BCK input method:Direct input or via the noise canceller |
| | In master/slave mode | • Polarity: Rising/falling edge canceler |
| Left and right clock/frame synchronization (SSILRCK/SSIFS) | In master mode | • Polarity: Low/high level selectable<br>• Supply/stop of SSI_RCK is selectable while communication is halted. |
| | In slave mode | • Two input methods: Direct input or via the noise canceller |
| Transmit data (SSITxD) and receive data (SSIRxD) | Transmission | • Mute: Transmission of transmit FIFO data or fixed value 0 selectable |
| | Reception (in slave mode) | • Two input methods: Direct input or via the noise canceller |
| FIFO | Capacity | • Transmit FIFO/receive FIFO: 4 bytes × 32 stages,<br>• Receive FIFO: 4 bytes × 32 stages |
| | Data alignment | • Method of data transfer between FIFO and shift register ( Left-justification and right-justification ) selectable. |
| Interrupt | Interrupt output | • Communication error/idle mode error (level)<br>• Receive data full interrupt (edge)<br>• Transmit data empty interrupt (edge)<br>• Receive data full/transmit data empty interrupt (edge) |
| | Interrupt capture spilling solution function | • Edge Interrupt supports inhibition function of capture spilling. |
| Low power consumption function | | • Supply/stop of audio clock is selectable in master-mode communication. |

The definition of words and phrases used by the format of the communication of SSIF-2 is as follows.

Table 27.2 Word Definition List

| Word | Definition |
|---|---|
| Start trigger | The first edge that reaches value that set to LRCKP in SSI_RCK pin after SSIF-2 permit communication. |
| Frame boundary | When SSIF-2 begins to transmit the first data of one frame.<br>When SSIF-2 forwarding ends final data of one frame. |
| Frame word number | Sound channel number in one frame |
| System word length | Bit length in one frame. |
| Data word length | Effective bit length in one frame. |
| Control bit of communication format | • SSICR register: FRM,DWL,SWL,LRCKP,SPDP,SDTA,PDTA,DEL<br>• SSIFCR register: BSW<br>• SSIOFR register: OMOD<br>• SSISCR register: TDES, RDFS |

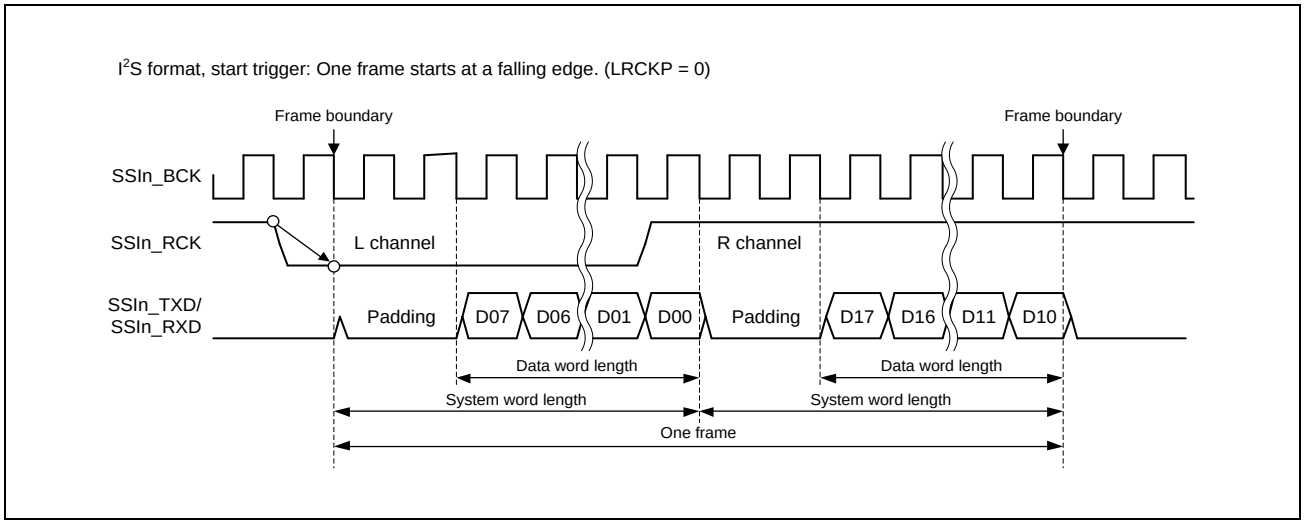
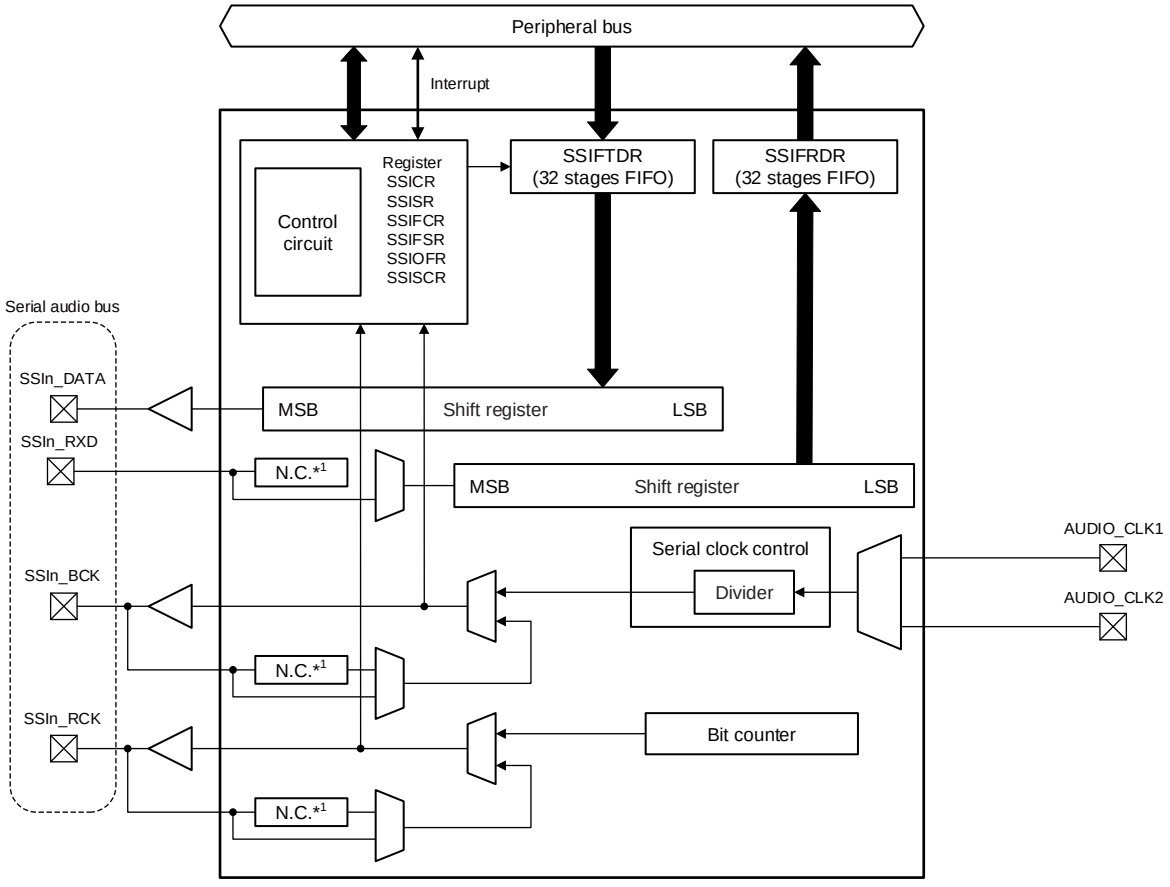



Figure 27.1 Communication Format Definition

### 27.1.2 Block Diagram

**Figure 27.2** shows the block diagram of SSIF-2.



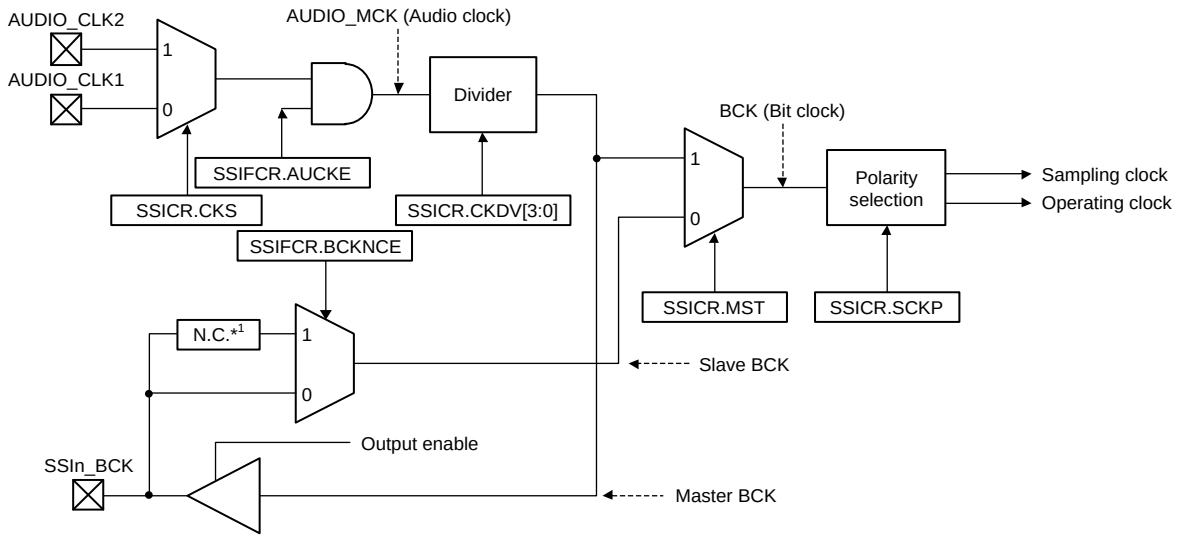

[Legend]

| | |
|---|---|
| SSICR: | Control register |
| SSISR: | Status register |
| SSIFCR: | FIFO control register |
| SSIFSR: | FIFO status register |
| SSIOFR: | Audio format register |
| SSIFTDR: | Transmit FIFO data register |
| SSIFRDR: | Receive FIFO data register |
| SSISCR: | Status Control register |

Note 1. N.C.: Noise canceller

Figure 27.2 SSIF-2 Block Diagram

**Figure 27.3** shows the clock configuration of SSIF-2.




[Legend]

SSICR: Control register

**Note:** Switch the above SFRs while communication is halted.

Note 1. N.C. (Noise canceller)

Figure 27.3 SSIF-2 Clock Configuration

## 27.2 Input/Output Pins

**Table 27.3** shows the pin configuration

Table 27.3 Pin Configuration

| Interface | name | I/O | Function |
|---|---|---|---|
| 0, (n = 0) | SSIn_BCK | I/O | Bit clock |
| | SSIn_RCK | I/O | LR clock/frame synchronization |
| | SSIn_TXD | Output | Serial data output |
| | SSIn_RXD | Input | Serial data input |
| Common | AUDIO_CLK1 | Input | External clock1 for audio (The oversampling clock signal is input.) |
| | AUDIO_CLK2 | Input | External clock2 for audio (The oversampling clock signal is input.) |

## 27.3 List of Registers

**Table 27.4** shows the register configuration of SSIF-2. The address of the SSIF-2 register is represented by the offset address from the base address. SSIF-2 base address is as follows:

SSIF-2 base address: H'0_1004_9C00 (ch0) (Overall Address Space)

Table 27.4 List of Control Registers

| Channel | Register Name | Symbol | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|---|
| 0 | Control register | SSICR | RW | H'0000_0000 | H'00 | 32 bits |
| | Status register<br>*Note*: Bits 31, 30, and 25 to 0 are read only. | SSISR | RW | H'0200_0000 | H'04 | 32 bits |
| | FIFO control register | SSIFCR | RW | H'0000_0000 | H'10 | 32 bits |
| | FIFO status register<br>*Note*: Bits 31 to 17 and 15 to 1 are read only. | SSIFSR | RW | H'0001_0000 | H'14 | 32 bits |
| | Transmit FIFO data register | SSIFTDR | W | H'0000_0000 | H'18 | 32 bits,<br>16 bits,<br>8 bits |
| | Receive FIFO data register | SSIFRDR | R | H'0000_0000 | H'1C | 32 bits,<br>16 bits,<br>8 bits |
| | Audio format register | SSIOFR | RW | H'0000_0000 | H'20 | 32 bits |
| | Status control register | SSISCR | RW | H'0000_0000 | H'24 | 32 bits |

**Note:** Access to the addresses other than those listed above is prohibited.

# 27.4 Function Details

## 27.4.1 Register Descriptions

### 27.4.1.1 Control Register (SSICR)

This is a 32-bit readable/writable register. With this register, select an audio clock, control IRQ, select data formats, and set an operation mode.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | CKS | TUIEN | TOIEN | RUIEN | ROIEN | IIEN | — | FRM[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 | R/W | 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 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | MST | BCKP | LRCKP | SPDP | SDTA | PDTA | DEL | CKDV[3:0] | | | | MUEN | — | TEN | REN |
| 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/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 | — | 0 | R | Reserved. Write 0. The read value is 0. |
| 30 | CKS | 0 | RW | Selects an audio clock for master-mode communication*1<br>0: Selects the AUDIO_CLK2 input<br>1: Selects the AUDIO_CLK1 input |
| 29 | TUIEN | 0 | RW | Transmit underflow interrupt output enable<br>0: Disables transmit underflow interrupt output<br>1: Enables transmit underflow interrupt output |
| 28 | TOIEN | 0 | RW | Transmit overflow interrupt output enable<br>0: Disables transmit overflow interrupt output<br>1: Enables transmit overflow interrupt output |
| 27 | RUIEN | 0 | RW | Receive underflow interrupt output enable<br>0: Disables receive underflow interrupt output<br>1: Enables receive underflow interrupt output |
| 26 | ROIEN | 0 | RW | Receive overflow interrupt output enable<br>0: Disables receive overflow interrupt output<br>1: Enables receive overflow interrupt output |
| 25 | IIEN | 0 | RW | Idle mode interrupt output enable<br>0: Disables idle mode interrupt output<br>1: Enables idle mode interrupt output |
| 24 | — | 0 | R | Reserved. Write 0. The read value is 0. |
| 23, 22 | FRM[1:0] | 00b | RW | Select frame word number*1 |

Communication format (SSIOFR.OMOD[1:0])

| FRM[1:0] | I²S (00b) | Monaural (10b) | TDM (01b) |
|---|---|---|---|
| 00b | 2 | 1 | Set prohibition |
| 01b | Set | Set prohibition | 4 |
| 10b | prohibition | | 6 |
| 11b | | | 8 |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 21 to 19 | DWL[2:0] | 000b | R/W | Selects data word length*1<br>000: 8 bits<br>001: 16 bits<br>010: 18 bits<br>011: 20 bits<br>100: 22 bits<br>101: 24 bits<br>110: 32 bits<br>111: Setting prohibited |
| 18 to 16 | SWL[2:0] | 000b | RW | Selects system word length*1<br>000: 8 bits<br>001: 16 bits<br>010: 24 bits<br>011: 32 bits<br>100: 48 bits<br>101: 64 bits<br>110: 128 bits<br>111: 256 bits |
| 15 | — | 0 | R | Reserved. Write 0. The read value is 0. |
| 14 | MST | 0 | RW | Master enable*1<br>0: Slave-mode communication<br>1: Master-mode communication |
| 13 | BCKP | 0 | RW | Selects bit clock polarity*1<br>0: SSILRCK/SSIFS and SSIFTxD/SSIFRxD change at a falling edge (sampling SSILRCK/SSIFS and SSIFRxD at a rising edge of SSIBCK).<br>1: SSILRCK/SSIFS and SSIFTxD/SSIFRxD change at a rising edge (sampling SSILRCK/SSIFS and SSIFRxD at a falling edge of SSIBCK). |
| 12 | LRCKP | 0 | RW | Selects the initial value and polarity of LRCK/FS*1<br>0: The initial value is at a high level<br>For SSILRCK/SSIFS, one frame start trigger is falling edge of SSILRCK/SSIFS.<br>1: The initial value is at a low level<br>For SSILRCK/SSIFS, one frame start trigger is rising edge of SSILRCK/SSIFS. |
| 11 | SPDP | 0 | RW | Selects serial padding polarity*1<br>0: Padding data is at a low level<br>1: Padding data is at a high level |
| 10 | SDTA | 0 | RW | Selects serial data alignment*1<br>0: Transmits and receives serial data first and then padding bits.<br>1: Transmit and receives padding bits first and then serial data. |
| 9 | PDTA | 0 | RW | Selects placement data alignment*1<br>0: Left-justifies placement data (SSIFTDR, SSIFRDR)<br>1: Right-justifies placement data (SSIFTDR, SSIFRDR) |
| 8 | DEL | 0 | RW | Selects serial data delay*1<br>0: Delay of one cycle of SSIBCK between SSILRCK/SSIFS and SSIFTxD/SSIFRxD.<br>1: No delay between SSILRCK/SSIFS and SSIFTxD/SSIFRxD.<br>This bit control SSILRCK/SSIFS in monaural format (Refer to **Section 27.4.2.2, Monaural Format** for details). |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 to 4 | CKDV[3:0] | H'0 | RW | Selects bit clock division ratio*1<br>CKDV[3:0]<br>0000: AUDIO_MCK<br>0001: AUDIO_MCK/2<br>0010: AUDIO_MCK/4<br>0011: AUDIO_MCK/8<br>0100: AUDIO_MCK/16<br>0101: AUDIO_MCK/32<br>0110: AUDIO_MCK/64<br>0111: AUDIO_MCK/128<br>1000: AUDIO_MCK/6<br>1001: AUDIO_MCK/12<br>1010: AUDIO_MCK/24<br>1011: AUDIO_MCK/48<br>1100: AUDIO_MCK/96<br>1101: Setting prohibited<br>1110: Setting prohibited<br>1111: Setting prohibited |
| 3 | MUEN | 0 | RW | Mute (silent) enable<br>0: Disables muting on the next frame boundary.<br>1: Enables muting on the next frame boundary. |
| 2 | — | 0 | R | Reserved. Write 0. The read value is 0. |
| 1 | TEN | 0 | RW | Transmission and reception enable*2 |
| 0 | REN | 0 | RW | (TEN, REN): Operating<br>00: Disables transmission and reception<br>01: Enables reception (starts reception)<br>10: Enables transmission (starts transmission)<br>11: Enables transmission and reception (starts transmission and reception) |

Note 1. Writing to these bits while SSIF-2 is in a communication state (SSISR.IIRQ = 0) is prohibited. If written, the operation performed immediately after writing is not guaranteed.

Note 2. If the TEN bit or REN bit is rewritten, make sure that the SSISR.IIRQ bit is in the desired status. If written, the operation performed immediately after writing is not guaranteed. For example, after enabling operation, make sure that SSISR.IIRQ = 0 and after disabling operation, make sure that SSISR.IIRQ = 1.

### CKS Bit

This bit sets the audio clock in master-mode communication (MST = 1). In slave-mode communication (MST = 0), setting of this bit is invalid.

To write this bit, do so when AUDIO_MCK is not supplied. Refer to the AUCKE bit details explanation in **Section 27.4.1.3, FIFO Control Register (SSIFCR)** for detailed timing.

### TUIEN Bit

This bit enables/disables output of transmit underflow interrupts. By enabling this bit (set it to 1), an interrupt is output at a rising edge of SSISR.TUIRQ = 1. An interrupt is also output when this bit is changed from 0 to 1 while SSISR.TUIRQ = 1.

### TOIEN Bit

This bit enables/disables output of transmit overflow interrupts. By enabling this bit (set it to 1), an interrupt is output at a rising edge of SSISR.TOIRQ = 1. An interrupt is also output when this bit is changed from 0 to 1 while SSISR.TOIRQ = 1.

### RUIEN Bit

This bit enables/disables output of receive underflow interrupts. By enabling this bit (set it to 1), an interrupt is output at a rising edge of SSISR.RUIRQ = 1. An interrupt is also output when this bit is changed from 0 to 1 while SSISR.RUIRQ = 1.

### ROIEN Bit

This bit enables/disables output of receive overflow interrupts. By enabling this bit (set it to 1), an interrupt is output at a rising edge of SSISR.ROIRQ = 1. An interrupt is also output when this bit is changed from 0 to 1 while SSISR.ROIRQ = 1.

### IIEN Bit

This bit enables/disables output of idle mode interrupts. By enabling this bit (set it to 1), an interrupt is output at a rising edge of SSISR.IIRQ = 1. An interrupt is also output when this bit is changed from 0 to 1 while SSISR.IIRQ = 1.

### FRM[1:0] Bits

These bits set system word number in one frame of the communication formats.

Rewrite these bits when the LR clock supply to the terminal SSILRCK stops. Refer to the LRCONT bit details explanation in **Section 27.4.1.7, Audio Format Register (SSIOFR)** for the output operation of the LR clock.

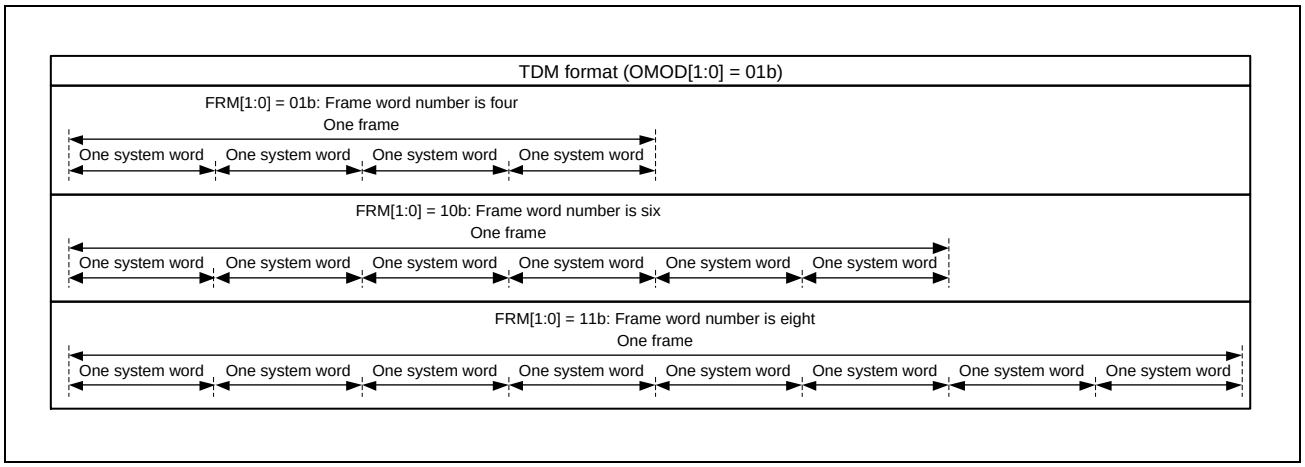



Figure 27.4 Frame word number

### DWL[2:0] Bits

These bits set the number of bits in one data word. The setting whose number of data of system word lengths is shorter than that of the data word length is a prohibition. Refer to **Table 27.13** in **Section 27.4.2, Communication Formats** for details.

### SWL[2:0] Bits

These bits set the number of bits in one system word. Padding bits are sent and received in relation with one data word set with DWL[2:0]. See **Table 27.13** in **Section 27.4.2, Communication Formats** for details.

Rewrite these bits when the LR clock supply to the terminal SSILRCK stops. Refer to the LRCONT bit details explanation in **Section 27.4.1.7, Audio Format Register (SSIOFR)** for the output operation of the LR clock.

### MST Bit

This bit sets master-/slave-mode communication.

To write this bit, do so when AUDIO_MCK is not supplied. Refer to the AUCKE bit details explanation in **Section 27.4.1.3, FIFO Control Register (SSIFCR)** for detailed timing.

### BCKP Bit

This bit sets the bit clock polarity (**Table 27.5**).

To write this bit, do so when AUDIO_MCK is not supplied. Refer to the AUCKE bit details explanation in **Section 27.4.1.3, FIFO Control Register (SSIFCR)** for detailed timing.

Table 27.5 Bit Clock Polarity

| Communication | Master/Slave | Timing | BCKP = 0 | BCKP = 1 |
|---|---|---|---|---|
| Reception | Slave | At SSILRCK/SSIFS sampling | SSIBCK rising edge | SSIBCK falling edge |
| | Master/slave | At SSIFRxD sampling | SSIBCK rising edge | SSIBCK falling edge |
| Transmission | Master | At change of SSILRCK/SSIFS output | SSIBCK falling edge | SSIBCK rising edge |
| | Master/slave | At change of SSIFTxD output | SSIBCK falling edge | SSIBCK rising edge |

### LRCKP Bit

This bit sets the initial value and polarity of SSILRCK/SSIFS. Set this bit according to the communication format to be used in SSIF-2 (**Table 27.6**). Only the start trigger is used at slave communication (MST=0).

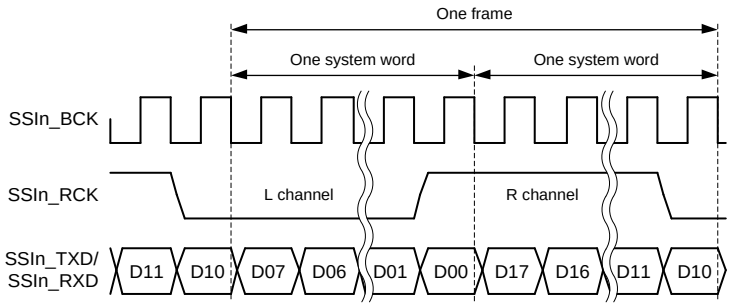
Rewrite this bit when the LR clock supply to the terminal SSILRCK stops. Refer to the LRCONT bit details explanation in **Section 27.4.1.7, Audio Format Register (SSIOFR)** for the output operation of the LR clock.

Table 27.6 Initial Output Value and Polarity of SSILRCK/SSIFS Pin

| Communication Format | Expected Initial State | Setting Value of LRCKP |
|---|---|---|
| $I^2S$ | H | 0 |
| Monaural | L | 1 |
| TDM | L | 1 |

**Note:** Do the setting that can be communicated respectively to $I^2S$, monaural, and TDM at the compatible format.

System word length: 8 bits (SWL = 000b), Data word length: 8 bits (DWL = 000b),
LR clock polarity: L channel = low, R channel = high (LRCKP = 0)
Other control bits of communication format are at their initial values.




System word length: 8 bits (SWL = 000b), Data word length: 8 bits (DWL = 000b),
LR clock polarity: L channel = high, R channel = low (LRCKP = 1)
Other control bits of communication format are at their initial values.

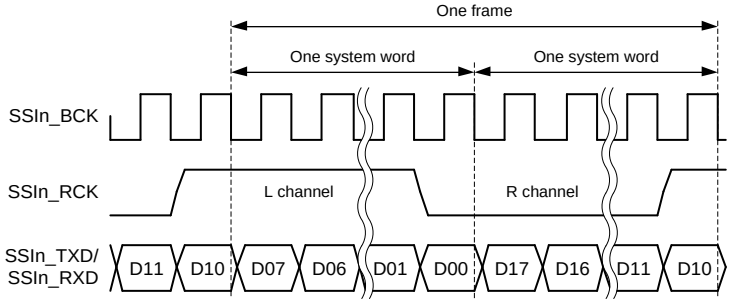



Figure 27.5 LRCK/FS Polarity Setting

### SPDP Bit

This bit sets polarity of padding bits.

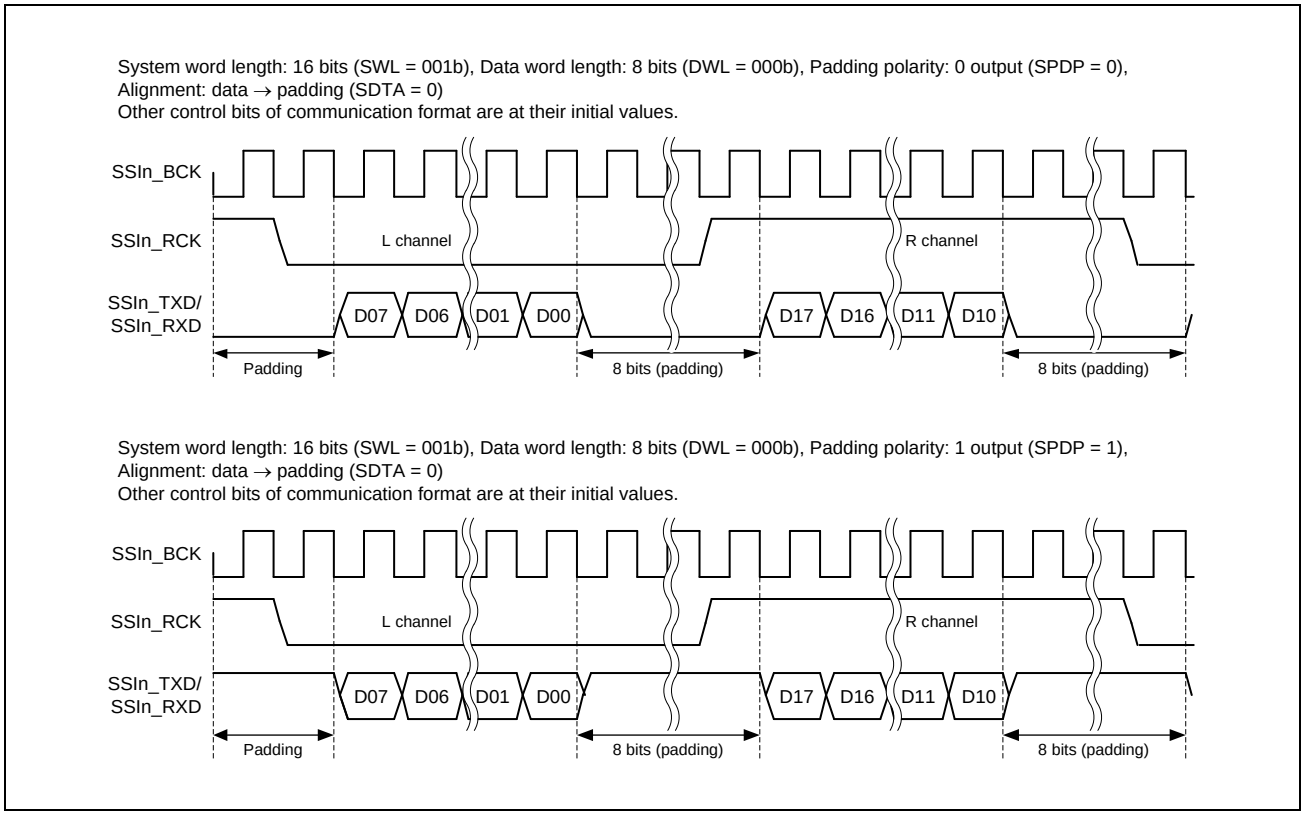



Figure 27.6 Padding Bit Polarity

### SDTA Bit

This bit sets alignment of serial data (data bits and padding bits). For communication without padding bits, this bit is invalid.

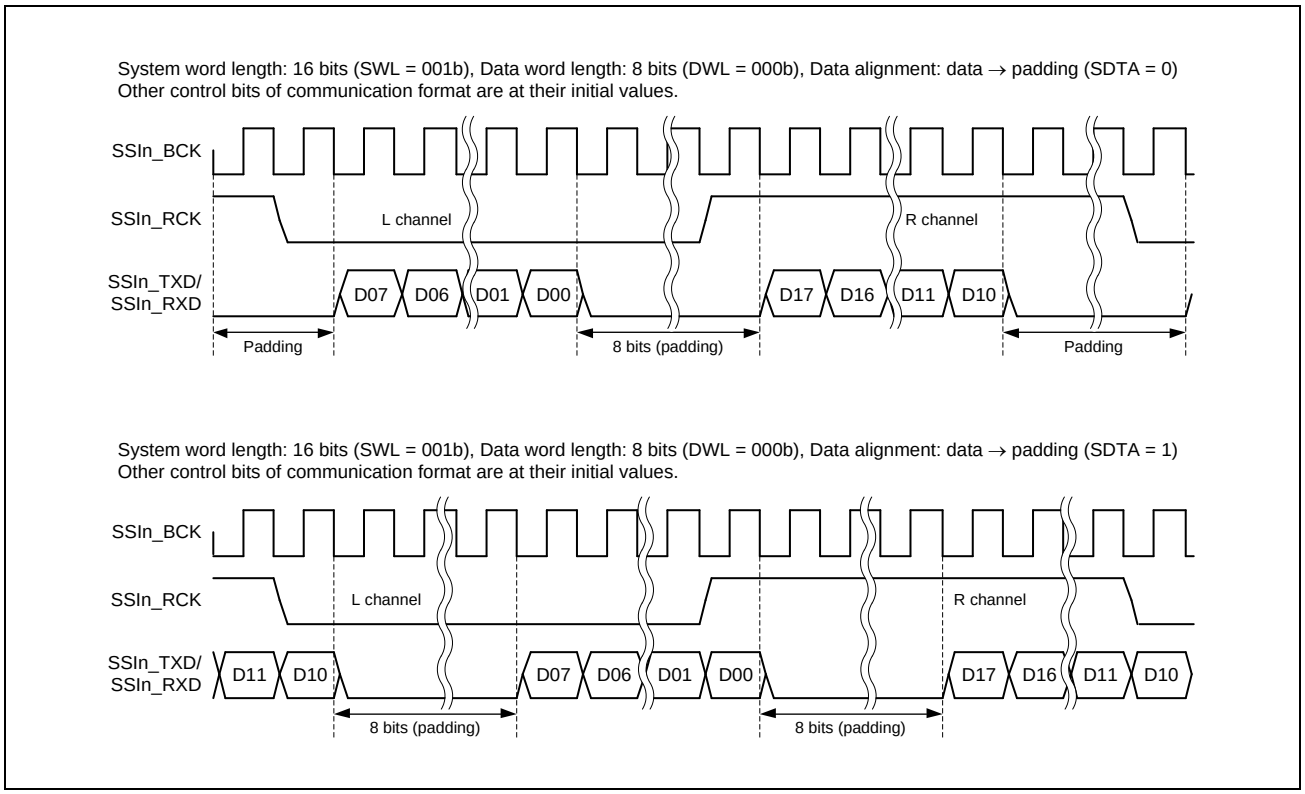



Figure 27.7 Alignment Setting of Serial Data with Padding Bits

### PDTA Bit

This bit sets alignment of placement data. With the setting of data word length as 32 bits (DWL[2:0] = 110b), this bit is invalid.

At transmission:

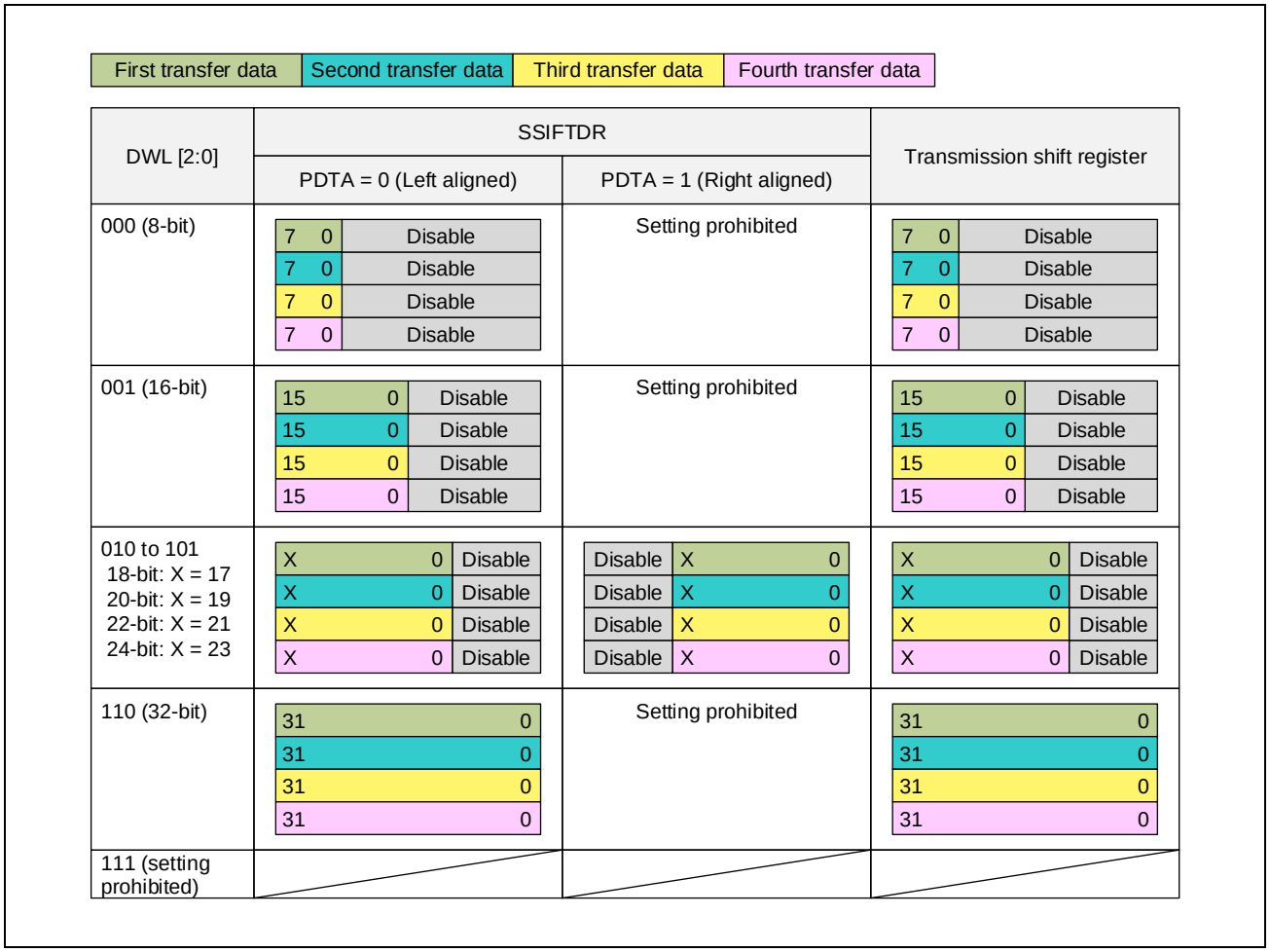



Figure 27.8 Alignment Setting of Placement Data in Transmission

At reception:

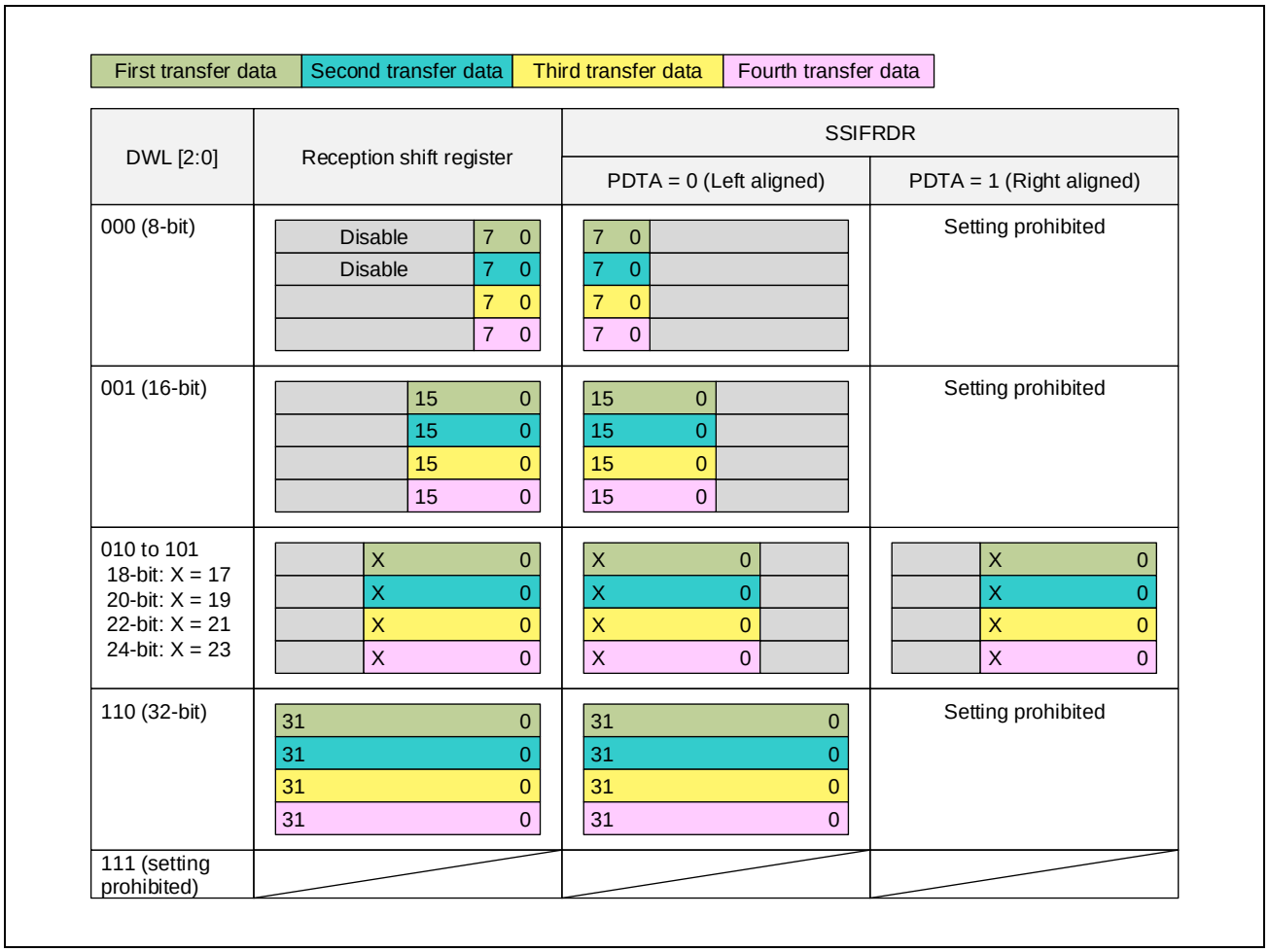



Figure 27.9 Alignment Setting of Placement Data in Reception

### DEL Bit

This bit sets whether or not there will be a delay between SSILRCK/SSIFS and SSITxD/SSIRxD.

With the I²S format and TDM format, set 0 to DEL (**Figure 27.10**). With the monaural format, this bit changes the high period width. Refer to **Section 27.4.2.2, Monaural Format** for details. Do the setting that can be communicated at the compatible communication format.

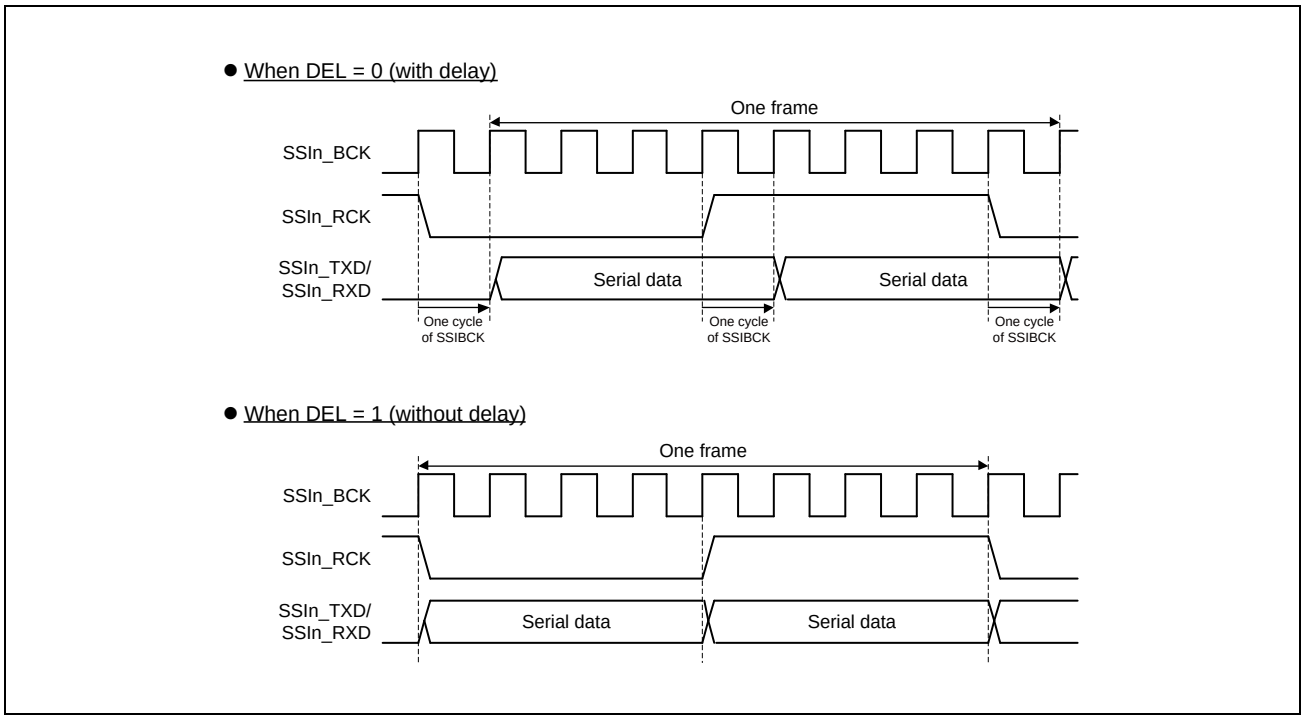



Figure 27.10 Setting of Delay in Serial Data

**CKDV[3:0] Bits**

These bits set the division ratio of the bit clock based on AUDIO_MCK in master-mode communication (MST = 1) (**Figure 27.11**). In slave-mode communication (MST = 0), setting of these bits are invalid.

To write this bit, do so when AUDIO_MCK is not supplied. Refer to the AUCKE bit details explanation in **Section 27.4.1.3, FIFO Control Register (SSIFCR)** for detailed timing.

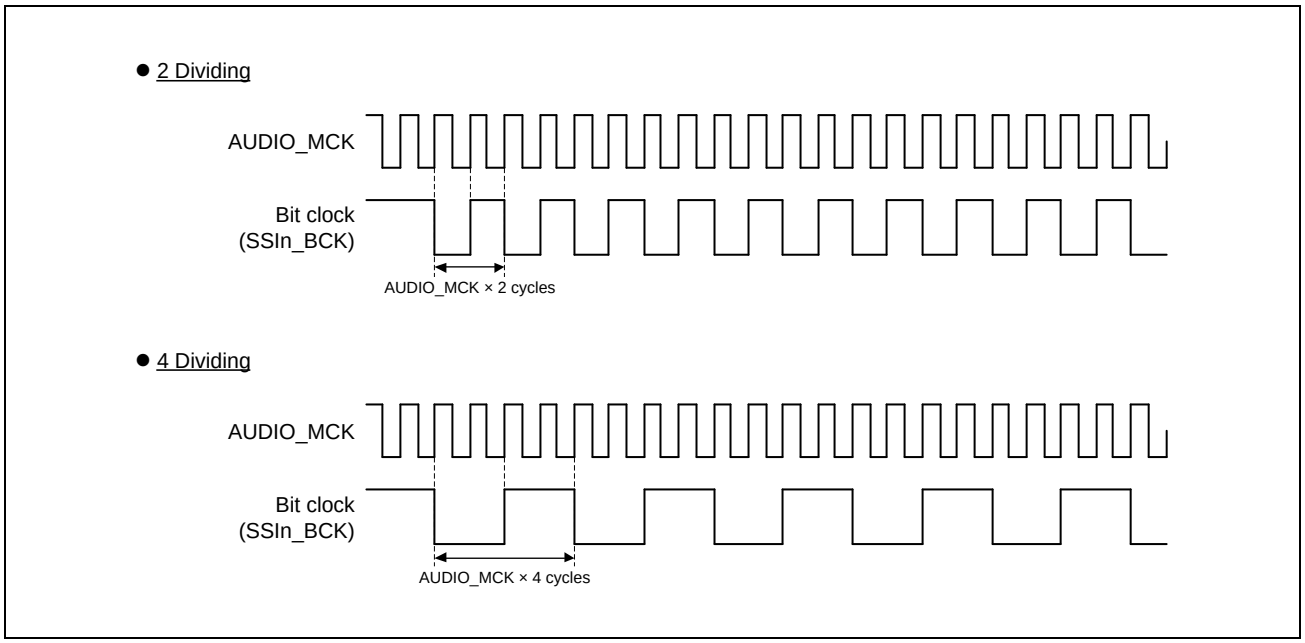



Figure 27.11 Sampling Frequencies in Master-mode Communication

### MUEN Bit

This bit sets/clears the mute function for the data output from the terminal SSITxD. When you set this bit to 1, the output value of SSITxD becomes 0 from the next frame boundary. When you set this bit to 0, the output value of SSITxD can be made the data of the FIFO transmission data register from the following frame boundary.

Note that this bit controls data only. Status flags and interrupt signals are normally generated. Set this bit after setting completing the communication format.

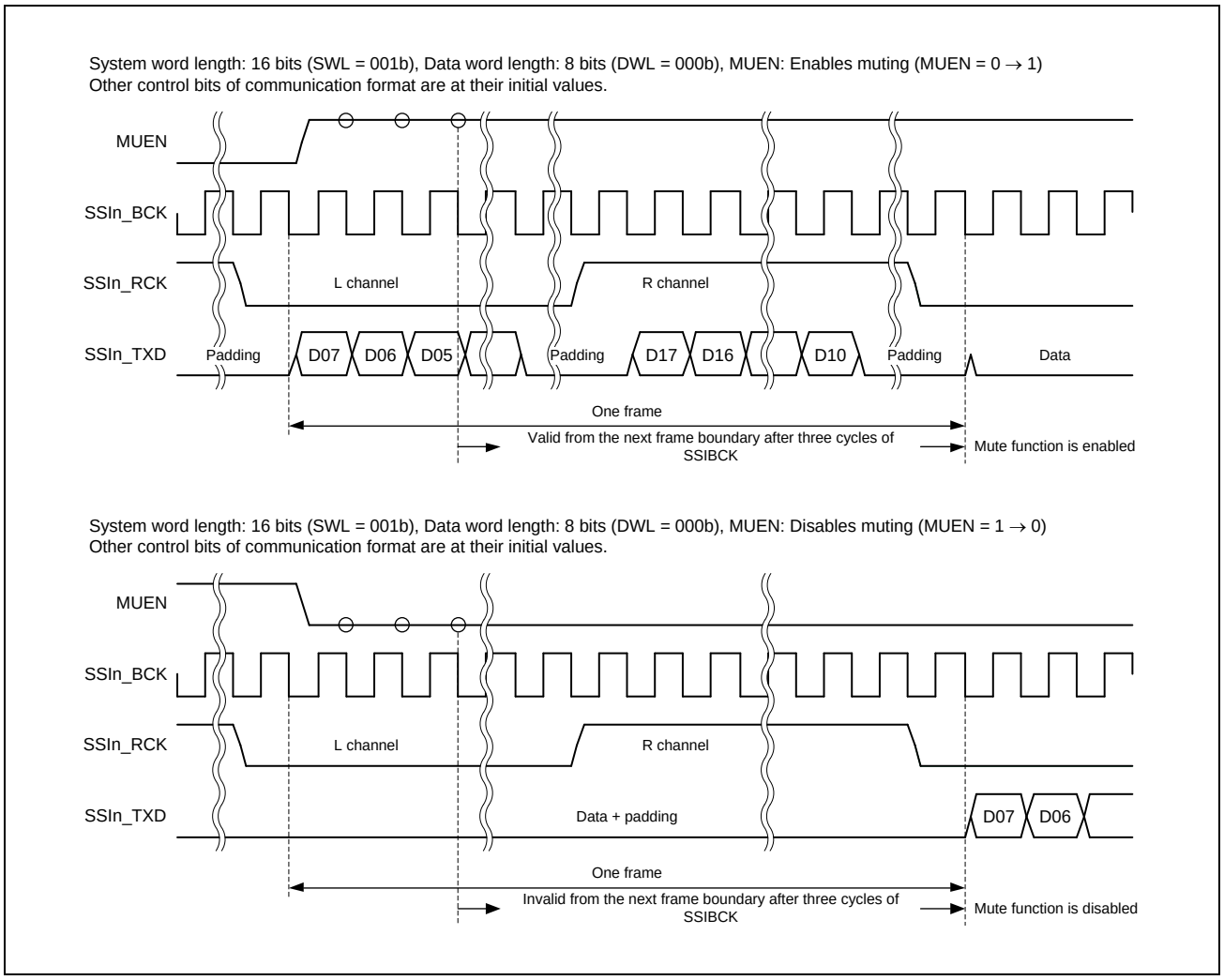



Figure 27.12 Transmit Data with the Mute Function Set

### TEN and REN Bits

These bits enable/disable transmission and reception. Writing 1 to these bits, transmission starts in synchronization with start trigger. Refer to **Section 27.5.2.2, Transmission**, **Section 27.5.2.3, Reception**, and **Section 27.5.2.4, Transmission and Reception** for details. Writing 0 to these bits, transmission stops in synchronization with next frame boundary. Set 1 at the same time when you use SSIF-2 as transmission and reception. To stop communication, make sure to stop transmission and reception (TEN = 0 and REN = 0).

To stop communication without waiting for the frame boundary, follow the software reset procedure.

### 27.4.1.2 Status Register (SSISR)

This is a readable/writable 32-bit register. It is configured with status flags that indicate SSIF-2 operational state.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | TUIRQ | TOIRQ | RUIRQ | ROIRQ | IIRQ | — | — | — | — | — | — | — | — | — |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R/W0 | R/W0 | R/W0 | R/W0 | 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, 30 | — | 00b | R | Reserved. Write 0. The read value is 0. |
| 29 | TUIRQ | 0 | RW0 | Transmit underflow error status flag<br>0: No transmit underflow error is generated.<br>1: A transmit underflow error is generated. |
| 28 | TOIRQ | 0 | RW0 | Transmit overflow error status flag<br>0: No transmit overflow error is generated.<br>1: A transmit overflow error is generated. |
| 27 | RUIRQ | 0 | RW0 | Receive underflow error status flag<br>0: No receive underflow error is generated.<br>1: A receive underflow error is generated. |
| 26 | ROIRQ | 0 | RW0 | Receive overflow error status flag<br>0: No receive overflow error is generated.<br>1: A receive overflow error is generated. |
| 25 | IIRQ | 1 | R | Idle mode status flag<br>0: In the communication state<br>1: In the idle state |
| 24 to 0 | — | H'0000000 | R | Reserved. Write 0. The read value is 0. |

#### TUIRQ Bit

This is a status flag that indicates a transmit underflow error. This flag is set by automatic determination but it must be cleared by register access. This flag indicates that writing serial data necessary for one frame to SSIFTDR was slower than one frame of the transmission operation. Even if this flag is cleared after this flag becomes '1', the output value of SSIFTxD is 0. Follow the communication stop procedure (**Figure 27.55**) and the error procedure (**Figure 27.56**) to output the data written in transmission FIFO data register (SSIFTDR) to the terminal SSIFTxD. See **Section 27.5.2.6, Error Handling** for the error recovery procedure. This flag is not cleared by transmission FIFO data register reset (SSIFCR.TFRST).

[Priority order for setting and clearing]

Setting is prioritized.*1

[Clearing condition]

When either of the following is approved.

(1) Writing 0 to this flag after reading this flag as 1.*2

(2) When you do the communication permission (SSICR.TEN is changed from 0 to 1)

[Clearing timing]

Timing according to the above-mentioned clear condition.

(1) At completion of writing 0 to this flag after reading this flag as 1 (**Figure 27.13**).

(2) Write 1 in SSICR.TEN, complete, and after one cycle of P0ϕ (**Figure 27.13**).

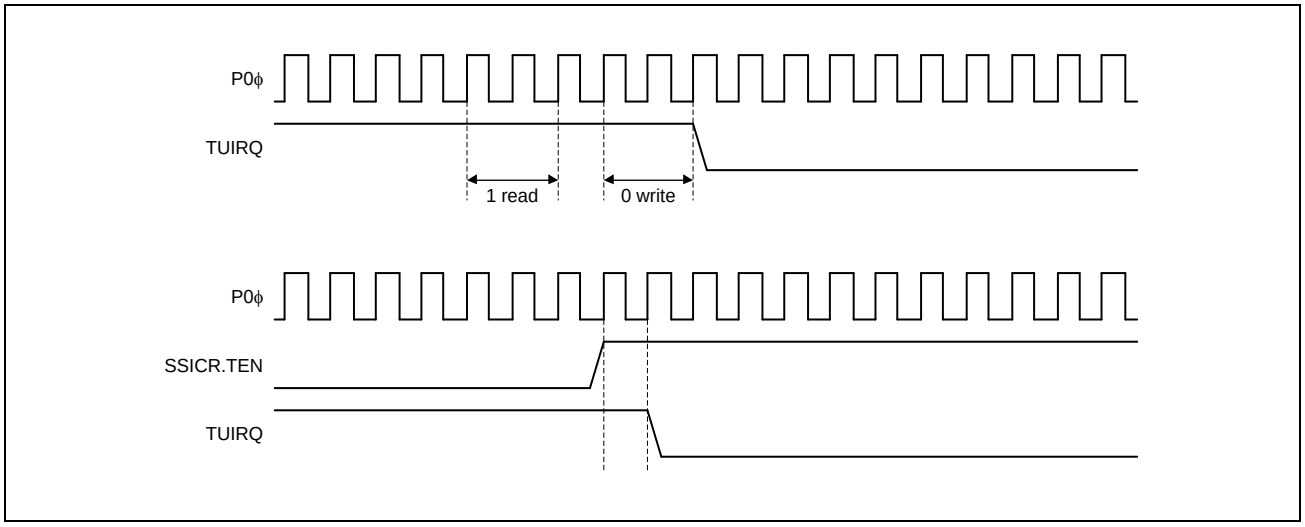



Figure 27.13 TUIRQ Clearing Timing

**Note 1.** This bit is also cleared by a software reset (SSIFCR.SSIRST = 1). The software reset has priority over all the clearing conditions described above.

**Note 2.** The state of reading this bit as one is cleared by any of the following three conditions being met:
- Software reset (SSIFCR.SSIRST=1).
- Completion of writing 0 to this flag after having read it as 1.
- One cycle of P0ϕ elapsing after the completion of writing 1 to SSICR.TEN.

NOTE

After the communication is permitted (SSICR.TEN is written and '1' is written from '0'), the transmission error flag (TOIRQ and TUIRQ of the SSISR register) is cleared. However, when the SSISR register is continuously read, the clearness of the transmission error flag might not be able to be read.

[Setting condition]

When data to be transmitted in the next frame is not fully written to SSIFTDR at the frame boundary of continuance communication.

[Setting timing]

After three cycles of P0ϕ from frame boundary (**Figure 27.14**).

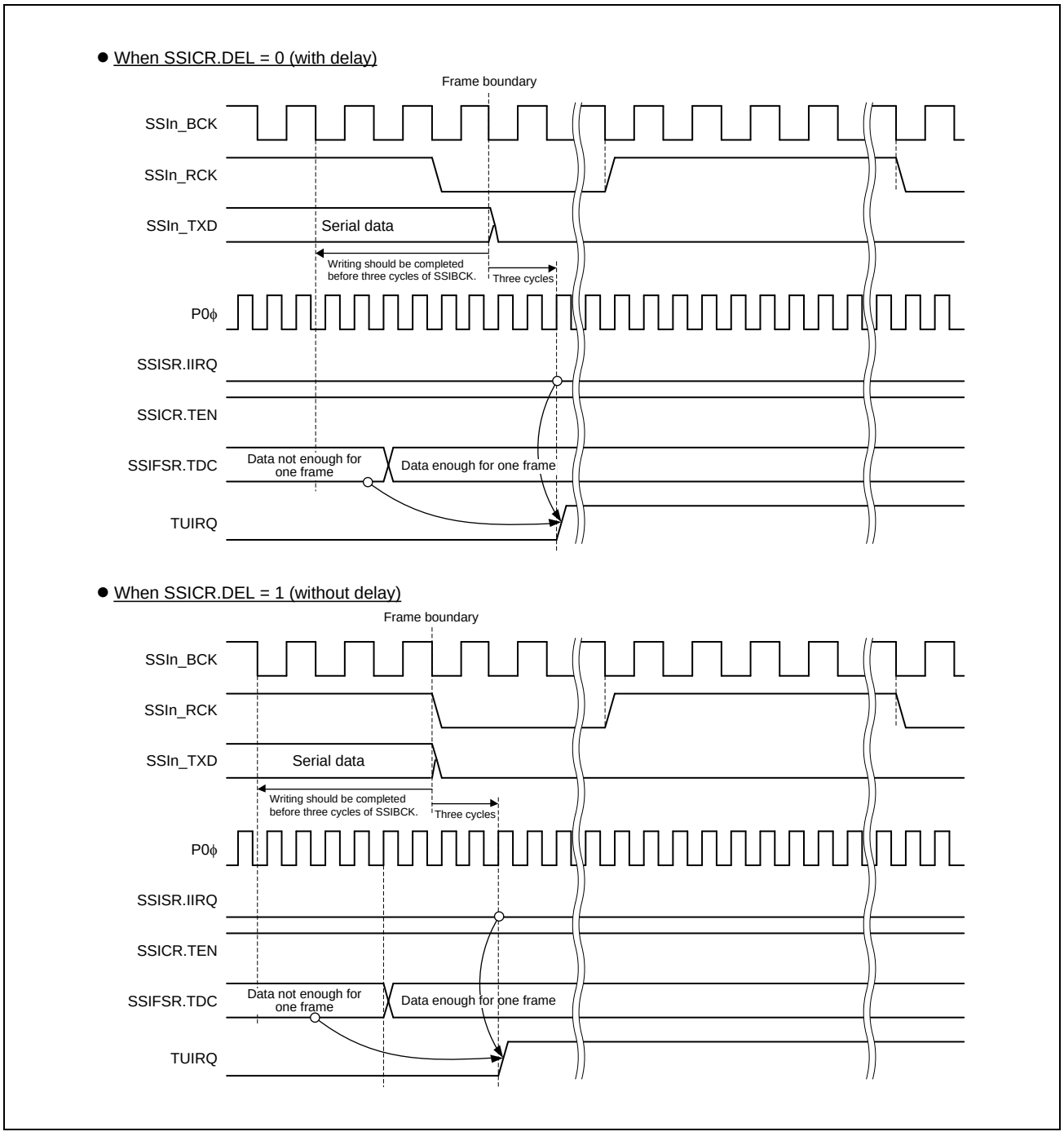



Figure 27.14 TUIRQ Setting Timing (When you continue communicating)

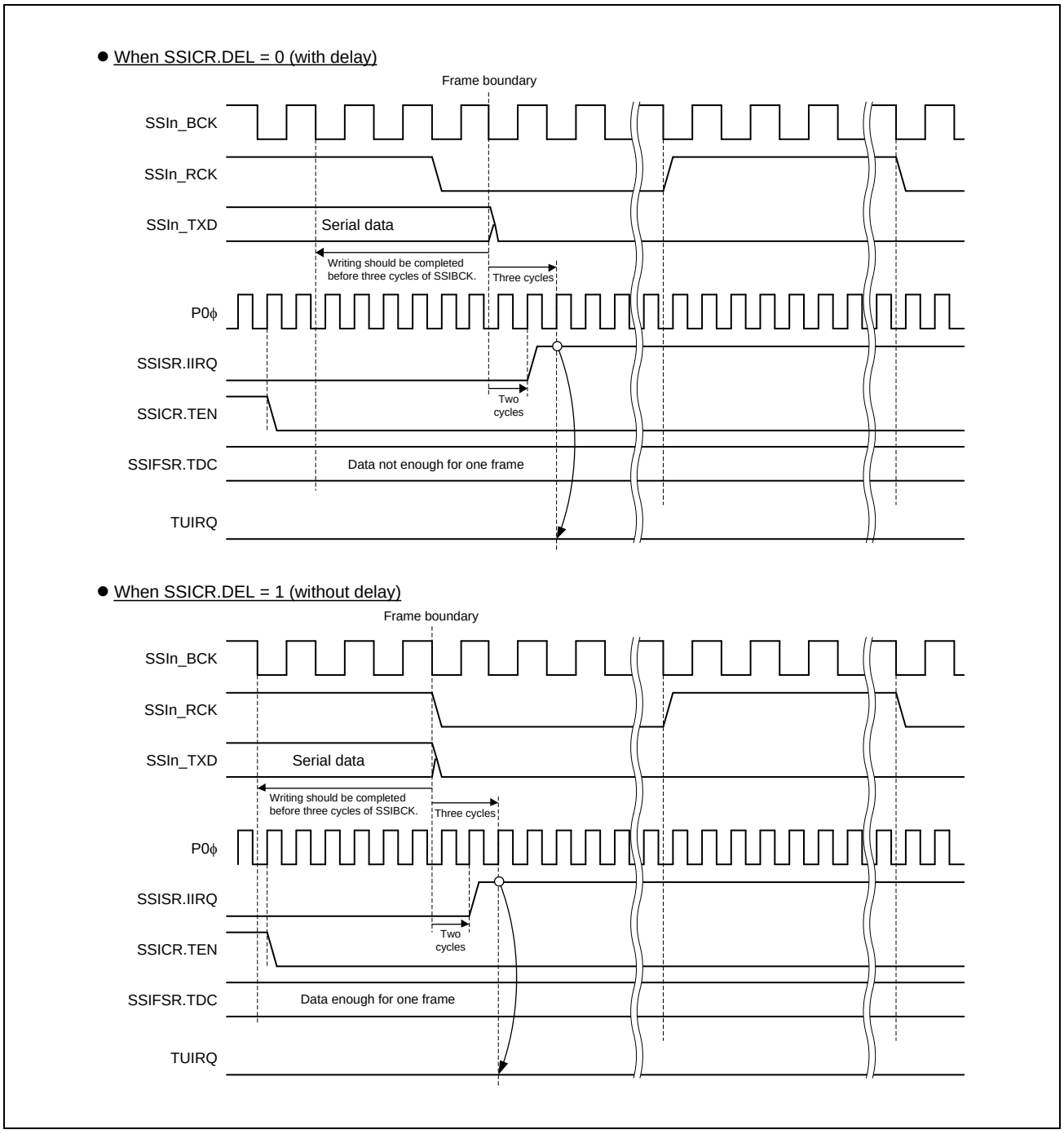



Figure 27.15 TUIRQ Setting Timing (When you stop communicating)

### TOIRQ Bit

This is a status flag that indicates a transmit overflow error. This flag is set by automatic determination but it must be cleared by register access. This flag indicates that serial data is written to SSIFTDR when it is full during transmission (SSICR.TEN=1). Data is not written to SSIFTDR where a transmit overflow error is generated. See **Section 27.5.2.6, Error Handling** for the error recovery procedure. This flag is not cleared by transmission FIFO data register reset (SSIFCR.TFRST).

[Priority order for setting and clearing]

Setting is prioritized.*1

[Clearing condition]

When either of the following is approved.

(1) Writing 0 to this flag after reading this flag as 1.*2

(2) When you do the communication permission (SSICR.TEN is changed from 0 to 1)

[Clearing timing]

Timing according to the above-mentioned clear condition.

(1) At completion of writing 0 to this flag after reading this flag as 1 (same as **Figure 27.13**).

(2) Write 1 in SSICR.TEN, complete, and after one cycle of P0ϕ (same as **Figure 27.13**).

**Note 1.** This bit is also cleared by a software reset (SSIFCR.SSIRST = 1). The software reset has priority over all the clearing conditions described above.

**Note 2.** The state of reading this bit as one is cleared by any of the following three conditions being met:
- Software reset (SSIFCR.SSIRST=1).
- Completion of writing 0 to this flag after having read it as 1.
- One cycle of P0ϕ elapsing after the completion of writing 1 to SSICR.TEN.

NOTE

After the communication is permitted (SSICR.TEN is written and '1' is written from '0'), the transmission error flag (TOIRQ and TUIRQ of the SSISR register) is cleared. However, when the SSISR register is continuously read, the clearness of the transmission error flag might not be able to be read.

[Setting condition]

During transmission operation (SSICR.TEN = 1), data is written to SSIFTDR while it is full.

[Setting timing]

At completion of writing to SSIFTDR (**Figure 27.16**)

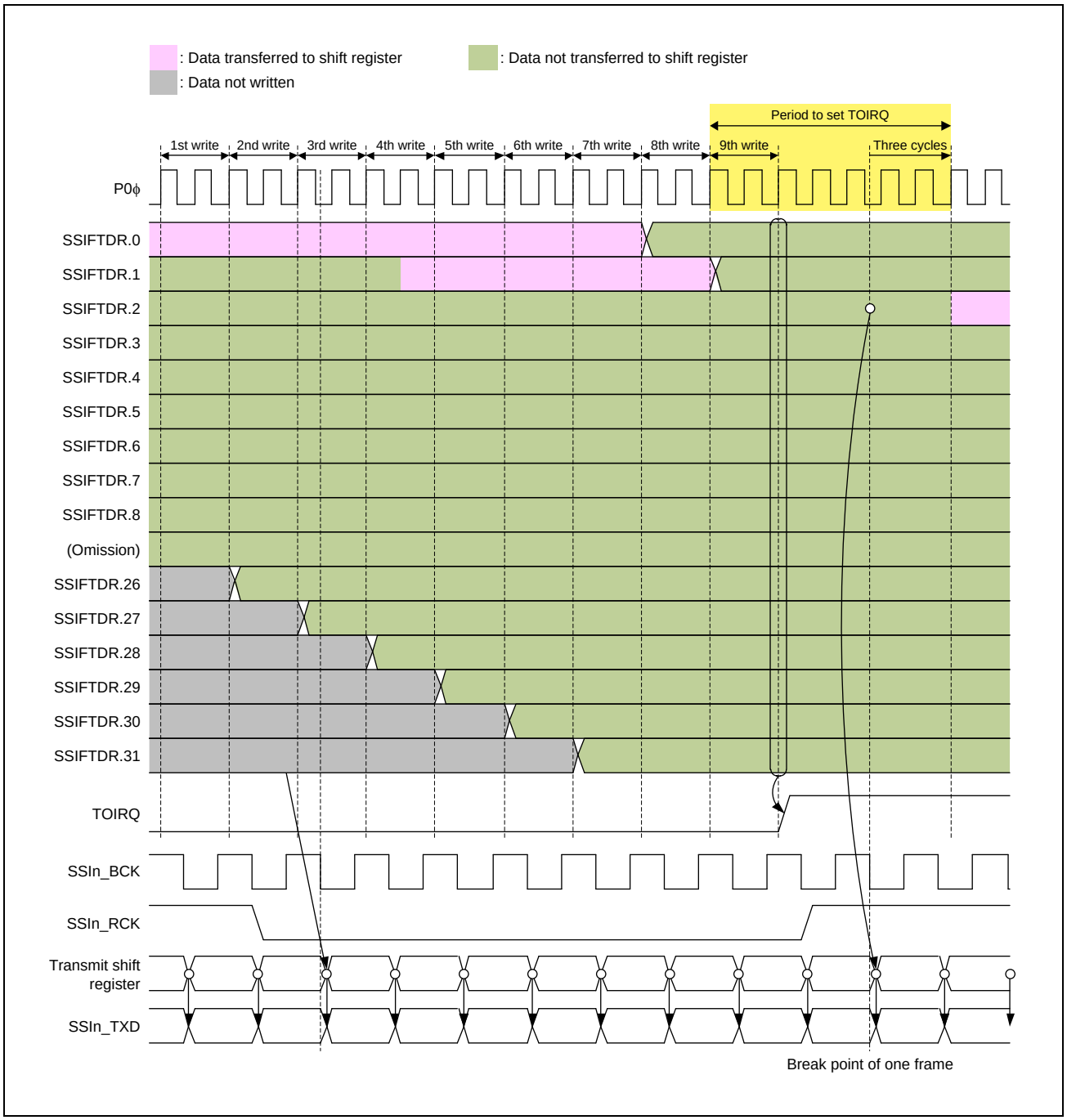



Figure 27.16 TOIRQ Setting Timing

**RUIRQ Bit**

This is a status flag that indicates a receive underflow error. This flag is set by automatic determination but it must be cleared by register access. This flag indicates that SSIFRDR is read while it is empty. Data read from SSIFRDR where a receive underflow error is generated is invalid. See **Section 27.5.2.6, Error Handling** for the error recovery procedure. This flag is not cleared by reception FIFO data register reset (SSIFCR.RFRST). Even if SSIFRDR is read to (SSIFCR.RFRST=1) while resetting reception FIFO data register, it doesn't set 1.

[Priority order for setting and clearing]

Setting is prioritized.*1

[Clearing condition]

When either of the following is approved.

(1) Writing 0 to this flag after reading this flag as 1.*2

(2) When you do the communication permission (SSICR.REN is changed from 0 to 1)

[Clearing timing]

Timing according to the above-mentioned clear condition.

(1) At completion of writing 0 to this flag after reading this flag as 1 (same as **Figure 27.13**).

(2) Write 1 in SSICR.REN, complete, and after one cycle of P0ϕ (same as **Figure 27.13**).

**Note 1.** This bit is also cleared by a software reset (SSIFCR.SSIRST = 1). The software reset has priority over all the clearing conditions described above.

**Note 2.** The state of reading this bit as one is cleared by any of the following three conditions being met:
- Software reset (SSIFCR.SSIRST=1).
- Completion of writing 0 to this flag after having read it as 1.
- One cycle of P0ϕ elapsing after the completion of writing 1 to SSICR.TEN.

NOTE

After the communication is permitted (SSICR.REN is written and '1' is written from '0'), the transmission error flag (ROIRQ and RUIRQ of the SSISR register) is cleared. However, when the SSISR register is continuously read, the clearness of the reception error flag might not be able to be read.

[Setting condition]

Reading from SSIFRDR while it is empty.

[Setting timing]

At completion of reading from SSIFRDR (**Figure 27.17**).

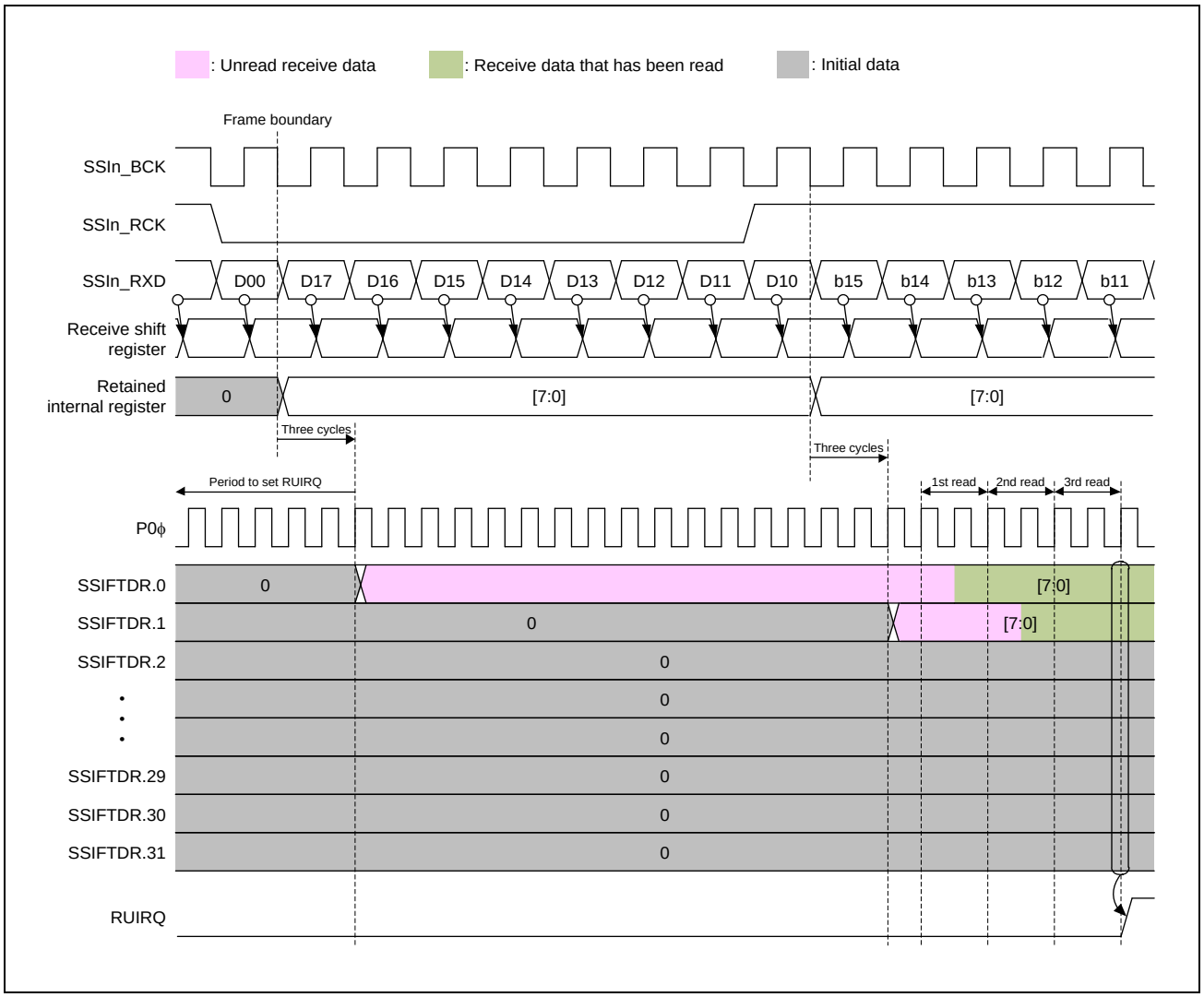



Figure 27.17 RUIRQ Setting Timing

**ROIRQ Bit**

This is a status flag that indicates a receive overflow error. This flag is set by automatic determination but it must be cleared by register access. This flag indicates that received data is supplied at a higher rate than requested. Data is not transferred from the receive shift register to SSIFRDR where a receive overflow error is generated. See **Section 27.5.2.6, Error Handling** for the error recovery procedure. This flag is not cleared by reception FIFO data register reset (SSIFCR.RFRST).

[Priority order for setting and clearing]

Setting is prioritized.

[Clearing condition]

Writing 0 to this flag after reading this flag as 1.

[Clearing timing]

At completion of writing 0 after reading 1 from an SFR (same as **Figure 27.13**).

[Priority order for setting and clearing]

Setting is prioritized.*1

[Clearing condition]

When either of the following is approved.

(1) Writing 0 to this flag after reading this flag as 1.*2

(2) When you do the communication permission (SSICR.REN is changed from 0 to 1)

[Clearing timing]

Timing according to the above-mentioned clear condition.

(1) At completion of writing 0 to this flag after reading this flag as 1 (same as **Figure 27.13**).

(2) Write 1 in SSICR.REN, complete, and after one cycle of P0ϕ (same as **Figure 27.13**).

**Note 1.** This bit is also cleared by a software reset (SSIFCR.SSIRST = 1). The software reset has priority over all the clearing conditions described above.

**Note 2.** The state of reading this bit as one is cleared by any of the following three conditions being met:
- Software reset (SSIFCR.SSIRST=1).
- Completion of writing 0 to this flag after having read it as 1.
- One cycle of P0ϕ elapsing after the completion of writing 1 to SSICR.TEN.

NOTE

After the communication is permitted (SSICR.REN is written and '1' is written from '0'), the transmission error flag (ROIRQ and RUIRQ of the SSISR register) is cleared. However, when the SSISR register is continuously read, the clearness of the reception error flag might not be able to be read.

[Setting condition]

At completion of receiving new data while SSIFRDR is full.

[Setting timing]

Three cycles of P0ϕ after reception is completed (**Figure 27.18**).

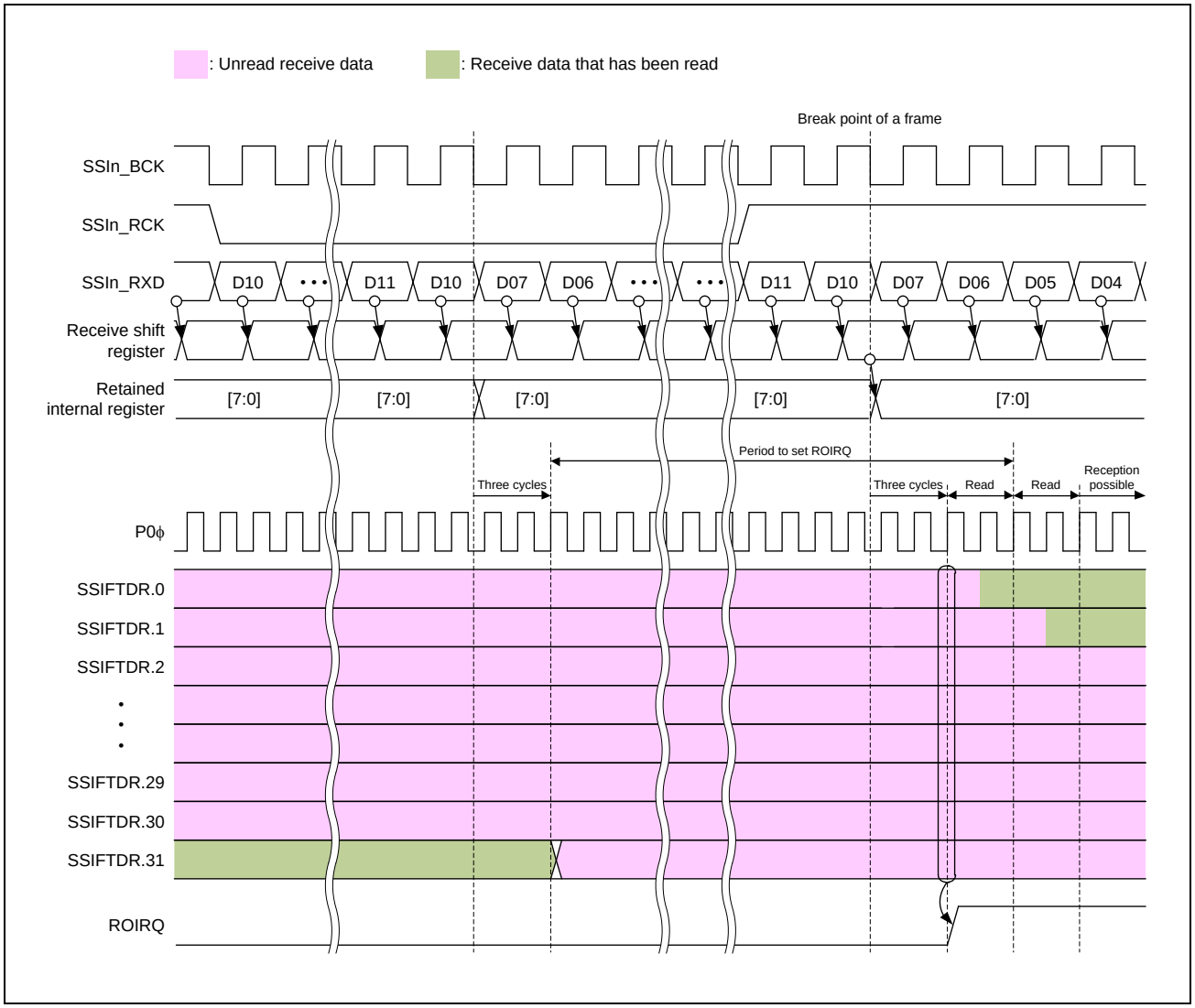



Figure 27.18 ROIRQ Setting Timing

### IIRQ Bit

This is a status flag that indicates the idle state. It indicates whether SSIF-2 is in the idle state or communication state (**Figure 27.19** and **Figure 27.20**).

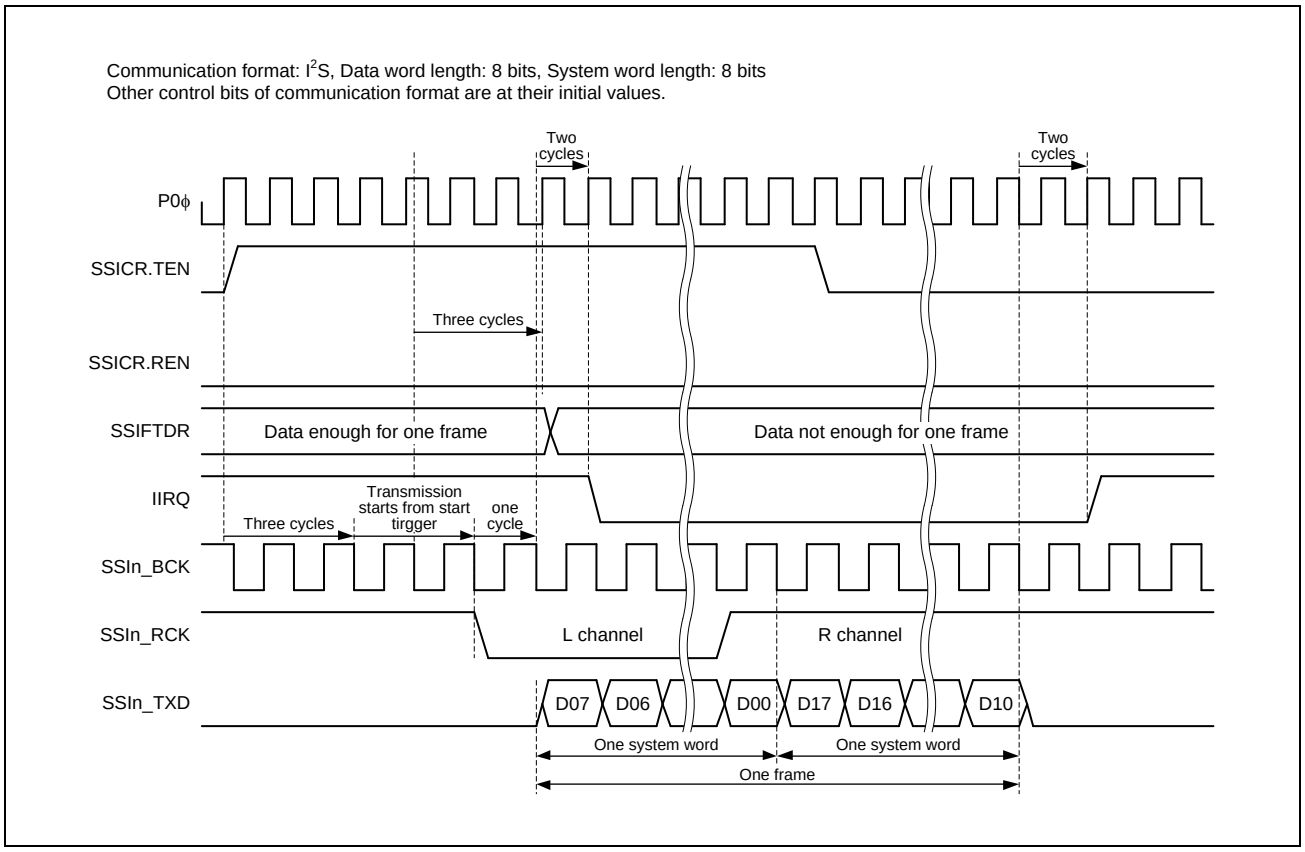



Figure 27.19 IIRQ Setting Timing (Transmission)

**Transmitter (dedicated to transmission)**

[Clearing condition]

While transmission is enabled (SSICR.TEN = 1 and SSICR.REN = 0), one frame of data to be transmitted has been written to SSIFTDR, and a start trigger is generated in SSILRCK/SSIFS.

[Clearing timing]

Two cycles of P0ϕ after one cycle of SSIBCK where a start trigger is generated.

[Setting condition]

One frame of data has been transmitted while transmission and reception are disabled (SSICR.TEN = 0 and SSICR.REN = 0).

[Setting timing]

Two cycles of P0ϕ after transmission is complete according to the setting condition (frame boundary).

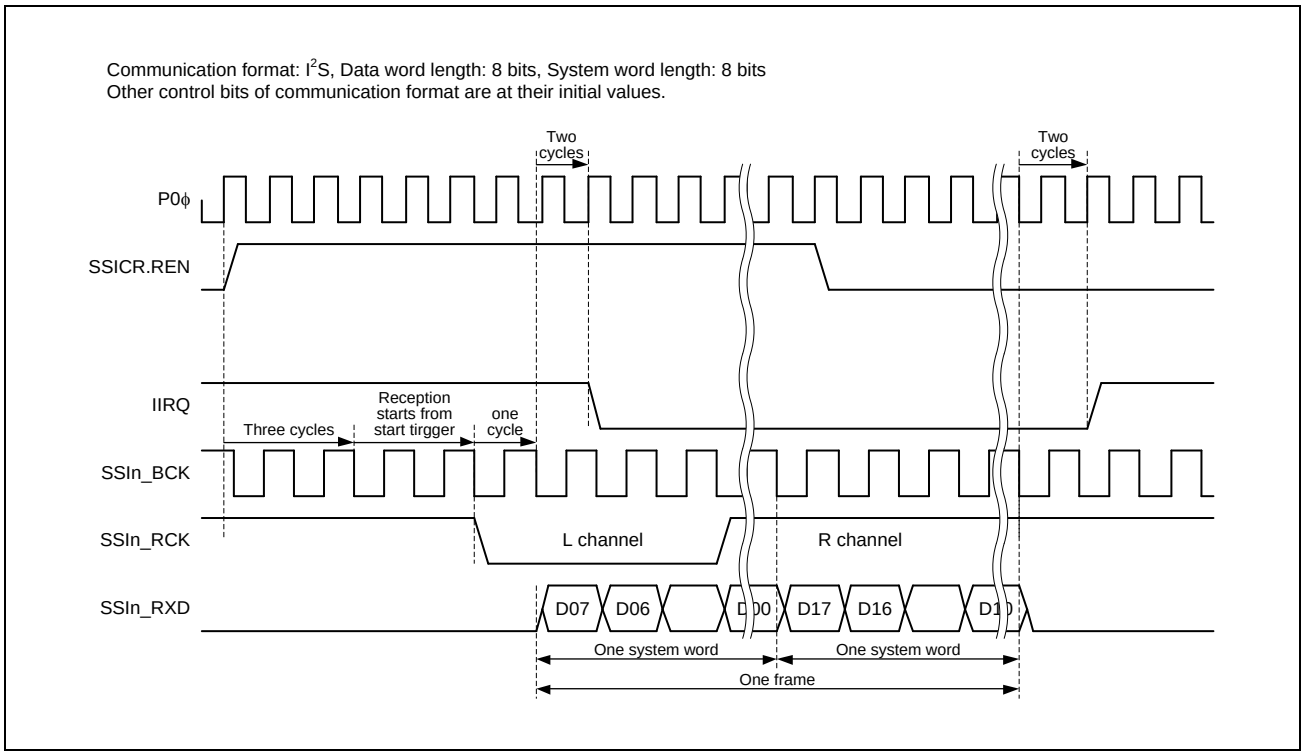



Figure 27.20 IIRQ Setting Timing (Reception)

**Receiver (dedicated to reception)**

[Clearing condition]

While reception is enabled (SSICR.TEN = 0 and SSICR.REN = 1), and a start trigger is generated in SSILRCK/SSIFS.

[Clearing timing]

Two cycles of P0ϕ after one cycle of SSIBCK when a start trigger of the clearing condition is generated.

[Setting condition]

One frame of data has been received while transmission and reception are disabled (SSICR.TEN = 0 and SSICR.REN = 0).

[Setting timing]

Two cycles of P0ϕ after reception of the setting condition is completed (frame boundary).

**Transceiver (transmission and reception)**

[Clearing condition]

While transmission and reception are enabled (SSICR.TEN = 1 and SSICR.REN = 1), one frame of data to be transmitted has been written to SSIFTDR, and a start trigger is generated in SSILRCK/SSIFS.

[Clearing timing]

Two cycles of P0ϕ after one cycle of SSIBCK when a start trigger of the clearing condition is generated.

[Setting condition]

One frame of data has been transmitted while transmission and reception are disabled (SSICR.TEN = 0 and SSICR.REN = 0).

[Setting timing]

Two cycles of P0ϕ after transmission is completed according to the setting condition (frame boundary).

### 27.4.1.3 FIFO Control Register (SSIFCR)

This is a readable/writable 32-bit register. It sets a software reset, byte swap, noise canceller, and enable/disable of interrupt requests.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | AUCKE | — | — | — | — | — | — | — | — | — | — | — | — | — | — | SSIRST |
| 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 | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | BSW | BCKNCE | LRCKNCE | RXDNCE | — | — | — | — | TIE | RIE | TFRST | RFRST |
| 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 | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 | AUCKE | 0 | RW | AUDIO_MCK Enable in Master-mode Communication*1<br>0: Disables supply of AUDIO_MCK<br>1: Enables supply of AUDIO_MCK |
| 30 to 17 | — | H'0000 | R | Reserved. Write 0. The read value is 0. |
| 16 | SSIRST | 0 | RW | Software Reset<br>0: Clears a software reset condition<br>1: Sets a software reset condition |
| 15 to 12 | — | H'00 | R | Reserved. Write 0. The read value is 0. |
| 11 | BSW | 0 | RW | Byte Swap Enable*1<br>0: Disables byte swap<br>1: Enables byte swap |
| 10 | BCKNCE | 0 | RW | Noise Canceller Enable in Slave-mode Communication (SSIBCK)*1,*2<br>0: Disables (bypasses) the noise canceller<br>1: Enables the noise canceller |
| 9 | LRCKNCE | 0 | RW | Noise Canceller Enable in Slave-mode Communication (SSILRCK/ SSIFS)*1,*2<br>0: Disables (bypasses) the noise canceller<br>1: Enables the noise canceller |
| 8 | RXDNCE | 0 | RW | Received Data Input Noise Canceller Enable in Slave-mode Communication (SSIRxD)*1,*2<br>0: Disables (bypasses) the noise canceller<br>1: Enables the noise canceller |
| 7 to 4 | — | H'00 | R | Reserved. Write 0. The read value is 0. |
| 3 | TIE | 0 | RW | Transmit Data Empty Interrupt Output Enable<br>0: Disables transmit data empty interrupts<br>1: Enables transmit data empty interrupts |
| 2 | RIE | 0 | RW | Receive Data Full Interrupt Output Enable<br>0: Disables receive data full interrupts<br>1: Enables receive data full interrupts |
| 1 | TFRST | 0 | RW | Transmit FIFO data register reset*1<br>0: Clears a transmit data FIFO reset condition<br>1: Sets a transmit data FIFO reset condition |
| 0 | RFRST | 0 | RW | Receive FIFO data register reset*1<br>0: Clears a receive data FIFO reset condition<br>1: Sets a receive data FIFO reset condition |

Note 1. Writing to these bits while SSIF-2 is in a communication state (SSISR.IIRQ = 0) is prohibited. If written, the operation performed immediately after writing is not guaranteed.

Note 2. This bit can only be set to 1 in slave-mode communication (SSICR.MST = '0') except in the case of transmission (SSICR.TEN = '1') while SSICR.DEL = '1'. Be sure to clear this bit to 0 in master-mode communication (SSICR.MST = '1') or during transmission in slave mode (SSICR.MST = '0', SSICR.TEN = '1') while SSICR.DEL = '1'. Set the BCKNCE, LRCKNCE, and RXDNCE bits to the same value.

### AUCKE Bit

This bit enables/disables supply to AUDIO_MCK while in master-mode communication (MST = 1). Select AUDIO_MCK (SSICR.CKS bit) before writing 1 to this bit.

Rewrite this bit after completing the setting (CKS, MST, BCKP, and CKDV of the SSICR register) that relates to AUDIO_MCK.

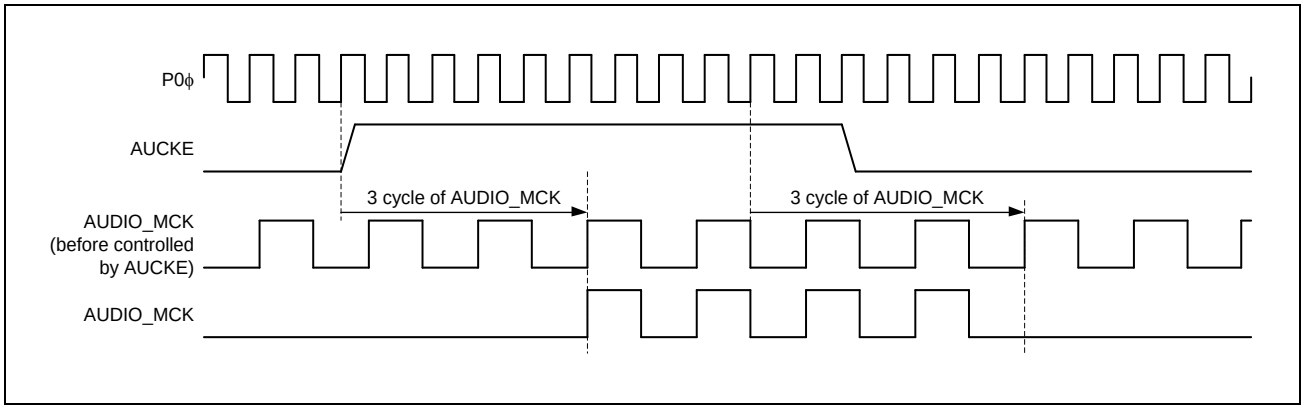



Figure 27.21 Stop/Resume of AUDIO_MCK

**CAUTION**

- In slave-mode communication (SSICR.MST = 0), SSIF-2 needs supply of SSIBCK. To stop BCK on the master side, make sure that SSIF-2 is in the idle state (SSISR.IIRQ = 1). If BCK is stopped before SSIF-2 becomes idle, take the procedure to start communication (**Figure 27.52**) or wait for an idle state by taking the procedure to resume communication (**Figure 27.57**).
- In master-mode communication (SSICR.MST = 1), SSIF-2 operates with the audio clock (AUDIO_MCK). To stop SSIF-2 completely, make sure that SSIF-2 is in the idle state (SSISR.IIRQ = 1) and then write 0 to SSIFCR.ADCKE. If 0 is written to SSIFCR.ADCKE before SSIF-2 becomes idle, take the procedure to start communication (**Figure 27.52**) or wait for the idle state by taking the procedure to resume communication (**Figure 27.57**).

**Figure 27.22** and **Figure 27.23** show timing until outputting it to the terminal SSIBCK after this bit is set to '1'.

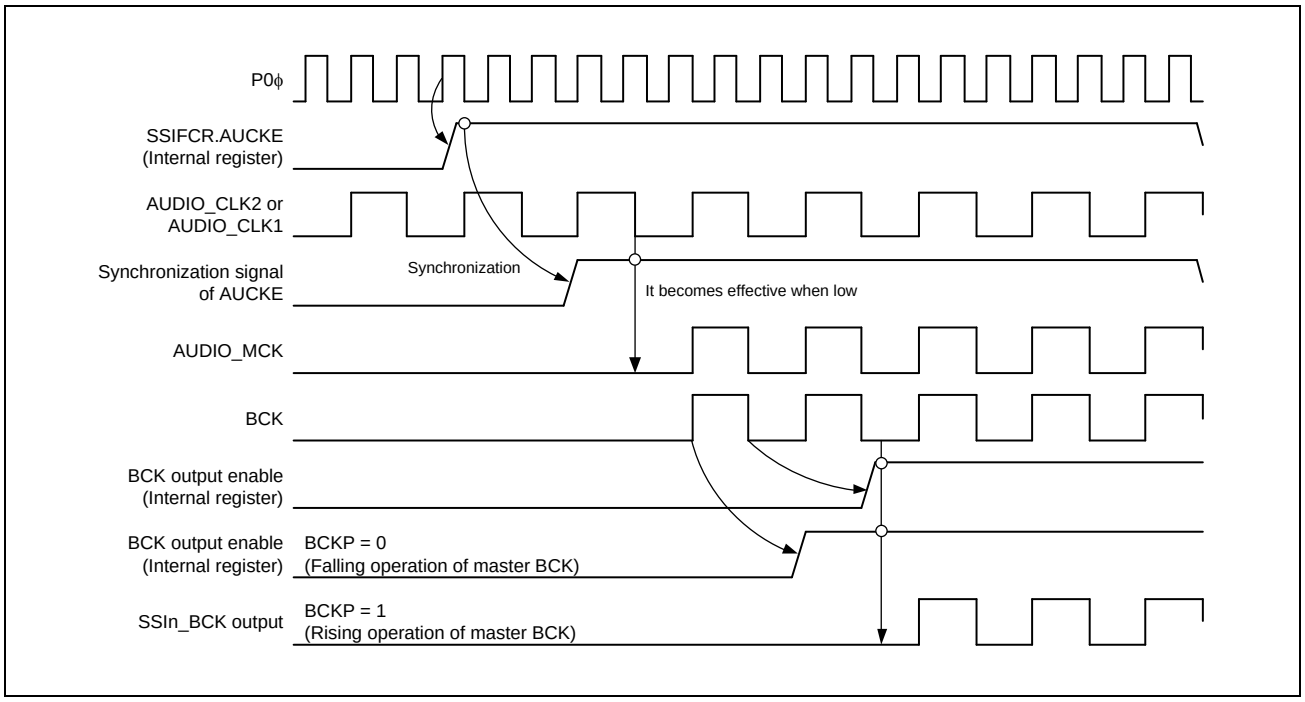



Figure 27.22 Timing chart when it begins to communicate master from system reset

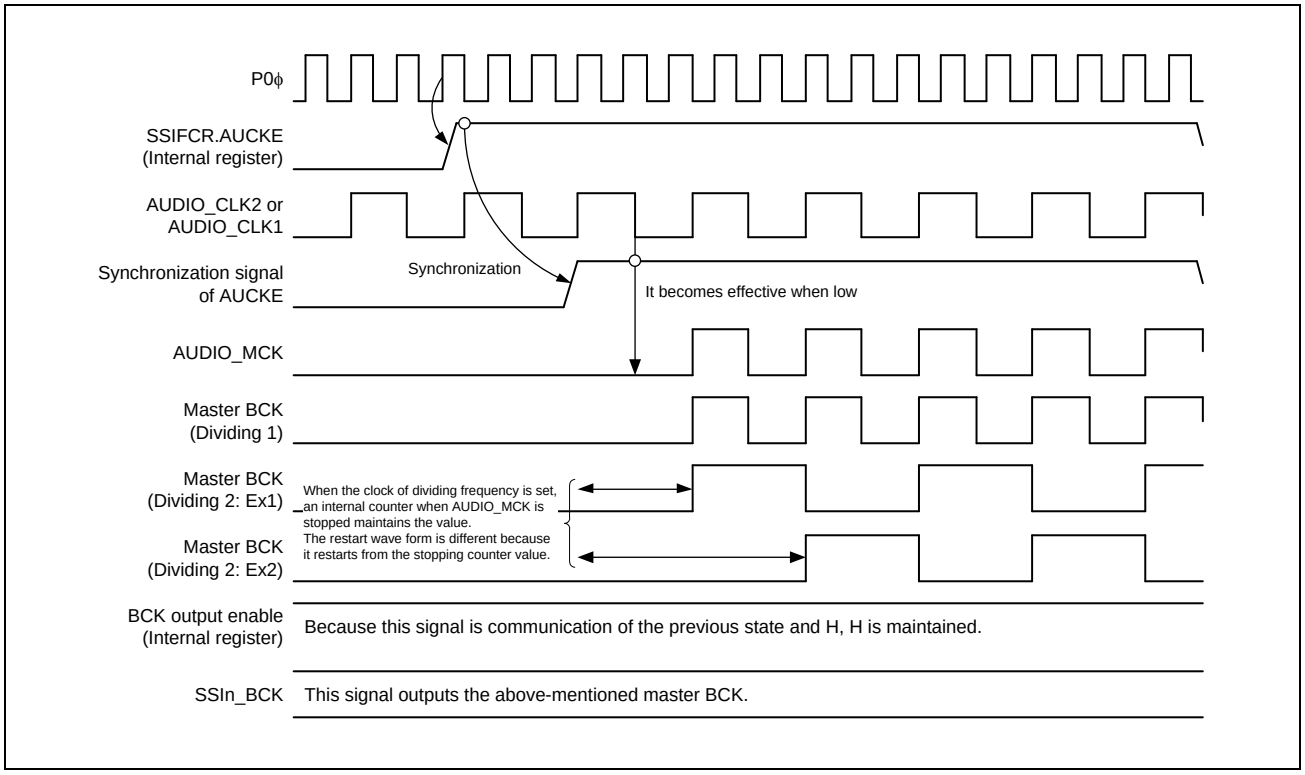



Figure 27.23 Timing chart when it begins to communicate master from communication stop

NOTE

When the supply of AUDIO_MCK is stopped, the terminal SSIBCK maintains the value. The terminal SSIBCK might stop in the state of “H”.

### SSIRST Bit

This bit sets a software reset of SSIF-2. Writing 1 to this bit initializes the internal state of SSIF-2. After a reset by writing 1, write 0 to release the reset because this bit is not automatically cleared to 0. After writing 0 to this bit, make sure that the bit becomes 0 before the next procedure.

To stop communication of SSIF-2 immediately, after turning off the peripheral functions, write 1 to this bit. Initialization by a software reset is performed without any relation with the bit clock.

Table 27.7 Bits Initialized by Software Reset of the SSIFCR.SSIRST Bit

| Symbol | Address (Base+) | | +0 | | | | | | | | +1 | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | | | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SSICR | H’00 | +0 | — | CKS | TUIEN | TOIEN | RUIEN | ROIEN | IIEN | — | FRM[1:0] | | DWL[2:0] | | | SWL[2:0] | | |
| | | +2 | — | MST | BCKP | LRCKP | SPDP | SDTA | PDTA | DEL | CKDV[3:0] | | | | MUEN | — | TEN | REN |
| SSISR | H’04 | +0 | — | — | TUIRQ | TOIRQ | RUIRQ | ROIRQ | IIRQ | — | — | — | — | — | — | — | — | — |
| | | +2 | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| SSIFCR | H’10 | +0 | AUCKE | — | — | — | — | — | — | — | — | — | — | — | — | — | — | SSIRST |
| | | +2 | — | — | — | — | BSW | BCKNCE | LRCKNCE | RxDNCE | — | — | — | — | TIE | RIE | TFRST | RFRST |
| SSIFSR | H’14 | +0 | — | — | TDC[5:0] | | | | | | — | — | — | — | — | — | — | TDE |
| | | +2 | — | — | RDC[5:0] | | | | | | — | — | — | — | — | — | — | RDF |
| SSIFTDR | H’18 | +0 | FTDR[31:16] | | | | | | | | | | | | | | | |
| | | +2 | FTDR[15:0] | | | | | | | | | | | | | | | |
| SSIFRDR | H’1C | +0 | FRDR[31:16] | | | | | | | | | | | | | | | |
| | | +2 | FRDR[15:0] | | | | | | | | | | | | | | | |
| SSIOFR | H’20 | +0 | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| | | +2 | — | — | — | — | — | — | BCKASTP | LRCONT | — | — | — | — | — | — | OMOD[1:0] | |
| SSISCR | H’24 | +0 | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| | | +2 | — | — | — | TDES[4:0] | | | | | — | — | — | RDFS[4:0] | | | | |

### BSW Bit

This bit enables/disables byte swap of register access for the transmit FIFO data register (SSIFTDR) and the receive FIFO data register (SSIFRDR). This bit is valid only with 16-bit access or 32-bit access to SSIFTDR and SSIFRDR (**Figure 27.24**).

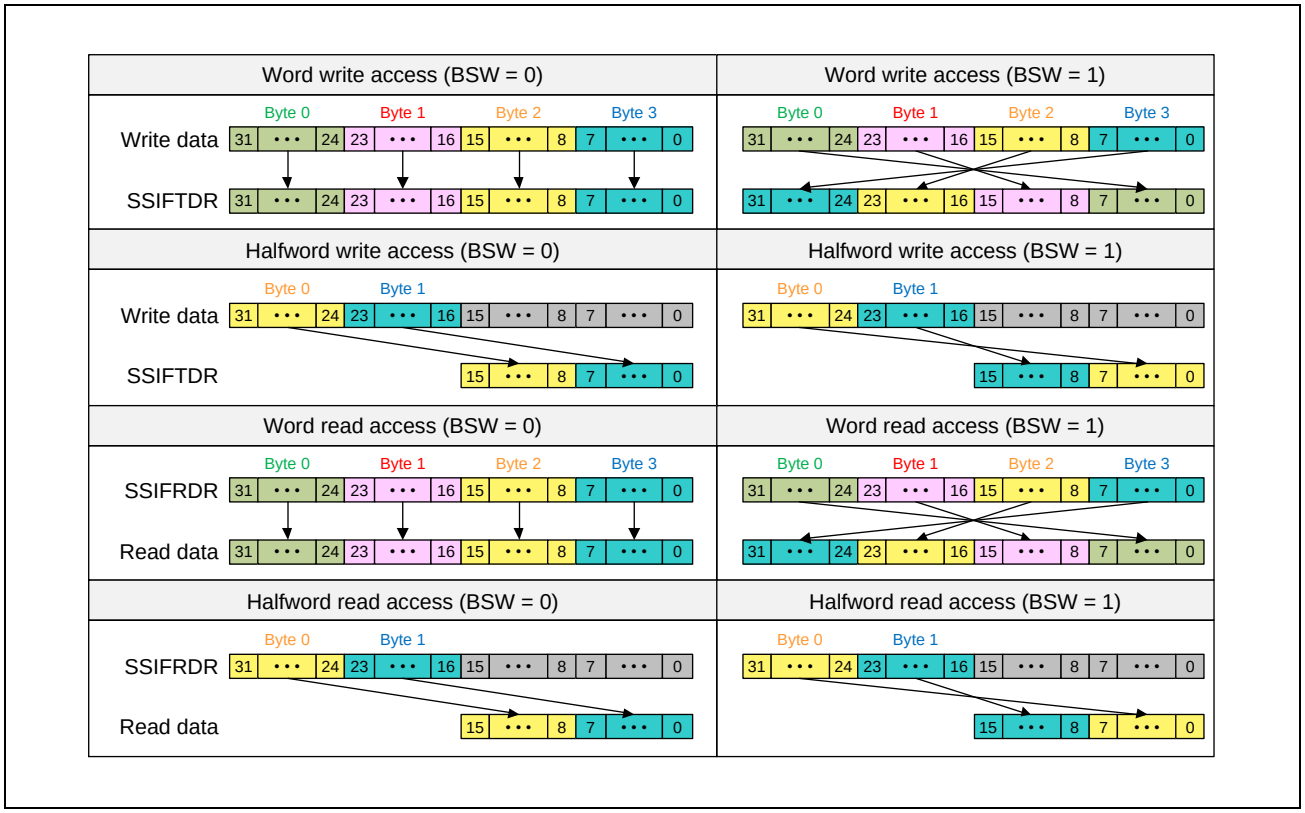



Figure 27.24 Operation Example of Byte Swap

### BCKNCE Bit

This bit enables or disables the noise cancellation function supporting the suppression of transients 15-ns in length (typ.) in input signals on the SSIBCK pin in slave-mode communication (SSICR.MST = ‘0’) except in the case of transmission (SSICR.TEN = ‘1’) while SSICR.DEL = ‘1’. The setting of this bit in master-mode communication (SSICR.MST = ‘1’) is disabled.

Be sure to clear this bit to 0 in master-mode communication or during transmission in slave mode (SSICR.MST = ‘0’, SSICR.TEN = ‘1’) while SSICR.DEL = ‘1’. Set the BCKNCE, LRCKNCE, and RXDNCE bits to the same value.

### LRCKNCE Bit

This bit enables or disables the noise cancellation function supporting the suppression of transients 15-ns in length (typ.) in input signals on the SSILRCK and SSIFS pins in slave-mode communication (SSICR.MST = ‘0’) except in the case of transmission (SSICR.TEN = ‘1’) while SSICR.DEL = ‘1’. The setting of this bit in master-mode communication (SSICR.MST = ‘1’) is disabled.

Be sure to clear this bit to 0 in master-mode communication or during transmission in slave mode (SSICR.MST = ‘0’, SSICR.TEN = ‘1’) while SSICR.DEL = ‘1’. Set the BCKNCE, LRCKNCE, and RXDNCE bits to the same value.

### RXDNCE Bit

This bit enables or disables the noise cancellation function supporting the suppression of transients 15-ns in length (typ.) in input signals on the SSIRxD pin in slave-mode communication (SSICR.MST = ‘0’) except in the case of transmission (SSICR.TEN = ‘1’) while SSICR.DEL = ‘1’ and while reception is enabled (SSICR.REN = ‘1’).

The setting of this bit in master-mode communication (SSICR.MST = ‘1’) or while reception is disabled (SSICR.REN = ‘0’) is disabled.

Be sure to clear this bit to 0 in master-mode communication or during transmission in slave mode (SSICR.MST = ‘0’, SSICR.TEN = ‘1’) while SSICR.DEL = ‘1’. Set the BCKNCE, LRCKNCE, and RXDNCE bits to the same value.

### TIE Bit

This bit enables/disables output of transmit data empty interrupts. Transmission data empty interrupt is used as an interrupt factor written in the FIFO transmission data register. Set this bit to 1 after setting set condition (SSISCR.TDES) of transmission data empty interrupt. See **Figure 27.25** for the generation timing of a transmit data empty interrupt.

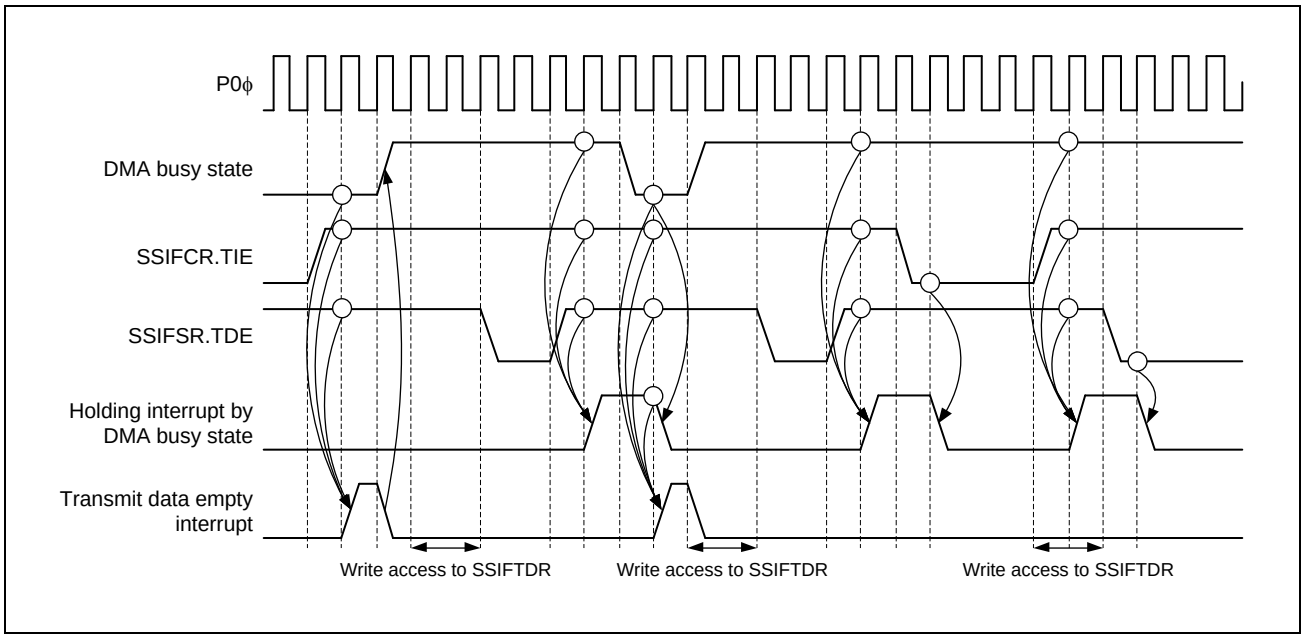



Figure 27.25 Generation timing of transmit data empty interrupt

### RIE Bit

This bit enables/disables output of receive data full interrupts. Receive data empty interrupt is used as an interrupt factor to read the FIFO receive data register. Set this bit to 1 after setting set condition (SSISCR.RDFS) of Receive data empty interrupt. See **Figure 27.26** for the generation timing of a receive data full interrupt.

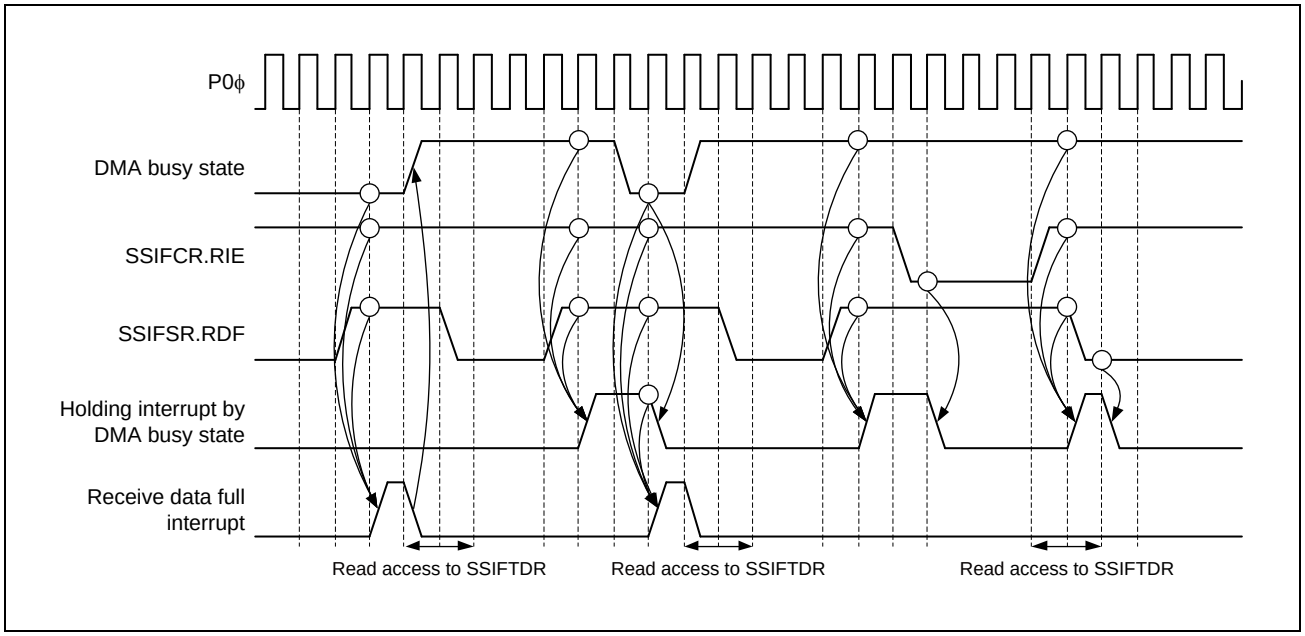



Figure 27.26 Generation timing of receive data empty interrupt

### TFRST Bit

This bit sets a software reset of the transmit FIFO data register (SSIFTDR). Writing 1 to this bit initializes the internal state related to SSIFTDR. See **Table 27.8** for the registers to which this software reset is applied. After a reset by writing 1, write 0 to clear the reset condition because this bit is not automatically cleared to 0. After writing 0 to this bit, make sure that the bit becomes 0 before the next procedure.

This bit is a bit of SSIRST for reset. Software reset by SSIRST becomes and writing this bit becomes invalid because of priority.

Table 27.8 Bits Initialized by Software Reset of the SSIFCR.TFRST Bit

| Symbol | Address (Base+) | | +0 | | | | | | | | +1 | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | | | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SSICR | H'00 | +0 | — | CKS | TUIEN | TOIEN | RUIEN | ROIEN | IIEN | — | FRM[1:0] | | DWL[2:0] | | | SWL[2:0] | | |
| | | +2 | — | MST | BCKP | LRCKP | SPDP | SDTA | PDTA | DEL | CKDV[3:0] | | | | MUEN | — | TEN | REN |
| SSISR | H'04 | +0 | — | — | TUIRQ | TOIRQ | RUIRQ | ROIRQ | IIRQ | — | — | — | — | — | — | — | — | — |
| | | +2 | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| SSIFCR | H'10 | +0 | AUCKE | — | — | — | — | — | — | — | — | — | — | — | — | — | — | SSIRST |
| | | +2 | — | — | — | — | BSW | BCKNCE | LRCKNCE | RxDNCE | — | — | — | — | TIE | RIE | TFRST | RFRST |
| SSIFSR | H'14 | +0 | — | — | TDC[5:0] | | | | | | — | — | — | — | — | — | — | TDE |
| | | +2 | — | — | RDC[5:0] | | | | | | — | — | — | — | — | — | — | RDF |
| SSIFTDR | H'18 | +0 | FTDR[31:16] | | | | | | | | | | | | | | | |
| | | +2 | FTDR[15:0] | | | | | | | | | | | | | | | |
| SSIFRDR | H'1C | +0 | FRDR[31:16] | | | | | | | | | | | | | | | |
| | | +2 | FRDR[15:0] | | | | | | | | | | | | | | | |
| SSIOFR | H'20 | +0 | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| | | +2 | — | — | — | — | — | — | BCKASTP | LRCONT | — | — | — | — | — | — | OMOD[1:0] | |
| SSISCR | H'24 | +0 | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| | | +2 | — | — | — | TDES[4:0] | | | | | — | — | — | RDFS[4:0] | | | | |

### RFRST Bit

This bit sets a software reset of the receive FIFO data register (SSIFRDR). Writing 1 to this bit initializes the internal state related to SSIFRDR. After a reset by writing 1, write 0 to clear the reset condition because this bit is not automatically cleared to 0. After writing 0 to this bit, make sure that the bit becomes 0 before the next procedure.

This bit is a bit of SSIRST for reset. Software reset by SSIRST becomes and writing this bit becomes invalid because of priority.

Table 27.9 Bits Initialized by Software Reset of the SSIFCR.TFRST Bit

| Symbol | Address (Base+) | | +0 | | | | | | | | +1 | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | | | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SSICR | H'00 | +0 | — | CKS | TUIEN | TOIEN | RUIEN | ROIEN | IIEN | — | FRM[1:0] | | DWL[2:0] | | | SWL[2:0] | | |
| | | +2 | — | MST | BCKP | LRCKP | SPDP | SDTA | PDTA | DEL | CKDV[3:0] | | | | MUEN | — | TEN | REN |
| SSISR | H'04 | +0 | — | — | TUIRQ | TOIRQ | RUIRQ | ROIRQ | IIRQ | — | — | — | — | — | — | — | — | — |
| | | +2 | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| SSIFCR | H'10 | +0 | AUCKE | — | — | — | — | — | — | — | — | — | — | — | — | — | — | SSIRST |
| | | +2 | — | — | — | — | BSW | BCKNCE | LRCKNCE | RxDNCE | — | — | — | — | TIE | RIE | TFRST | RFRST |
| SSIFSR | H'14 | +0 | — | — | TDC[5:0] | | | | | | — | — | — | — | — | — | — | TDE |
| | | +2 | — | — | RDC[5:0] | | | | | | — | — | — | — | — | — | — | RDF |
| SSIFTDR | H'18 | +0 | FTDR[31:16] | | | | | | | | | | | | | | | |
| | | +2 | FTDR[15:0] | | | | | | | | | | | | | | | |
| SSIFRDR | H'1C | +0 | FRDR[31:16] | | | | | | | | | | | | | | | |
| | | +2 | FRDR[15:0] | | | | | | | | | | | | | | | |
| SSIOFR | H'20 | +0 | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| | | +2 | — | — | — | — | — | — | BCKASTP | LRCONT | — | — | — | — | — | — | OMOD[1:0] | |
| SSISCR | H'24 | +0 | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| | | +2 | — | — | — | TDES[4:0] | | | | | — | — | — | RDFS[4:0] | | | | |

### 27.4.1.4 FIFO Status Register (SSIFSR)

This register is configured with status flags that indicate the status of the transmit FIFO data register and the receive FIFO data register.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | TDC[5:0] | | | | | | — | — | — | — | — | — | — | TDE |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W0 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | RDC[5:0] | | | | | | — | — | — | — | — | — | — | RDF |
| 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/W0 |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31, 30 | — | H'0 | R | Reserved. Write 0. The read value is 0. |
| 29 to 24 | TDC[5:0] | H'00 | R | Number of Transmit FIFO Data Indication Flag |
| 23 to 17 | — | H'00 | R | Reserved. Write 0. The read value is 0. |
| 16 | TDE | 1 | RW0 | Transmit Data Empty Flag<br>0: The free space of SSIFTDR is not more than the value of SSISCR.TDES<br>1: The free space of SSIFTDR is not less than the value of SSISCR.TDES plus one. |
| 15, 14 | — | H'0 | R | Reserved. Write 0. The read value is 0. |
| 13 to 8 | RDC[5:0] | H'00 | R | Number of Receive FIFO Data Indication Flag |
| 7 to 1 | — | H'00 | R | Reserved. Write 0. The read value is 0. |
| 0 | RDF | 0 | RW0 | Receive Data Full Flag<br>0: The size of received data in SSIFRDR is not more than the value of SSISCR.RDFS<br>1: The size of received data in SSIFRDR is not less than the value of SSISCR.RDFS plus one. |

### TDC[5:0] Bits

These bits indicate the number of valid data that are stored in the transmit FIFO data register (SSIFTDR). With this flag as H'0, there is no data to be transmitted. With H'20, there is no space to write data.

### TDE Bit

This bit indicates that the transmit FIFO data register (SSIFTDR) has free space not less than the amount set with the SSISCR.TDES bit plus one. This flag is set by automatic determination but it must be cleared by register access.
[Priority order for setting and clearing]

Clearing is prioritized.*1

[Clearing condition]

Either of the following two:

(1) Writing 0 to this bit after reading this bit as 1.*2

(2) When writing data to SSIFTDR.

[Clearing timing]

Timing according to the above-mentioned clear condition.

(1) At completion of writing 0 to this bit after reading this bit as 1 (**Figure 27.13**).

(2) When writing data to SSIFTDR.

**Note 1.** This bit is also cleared by a software reset (SSIFCR.SSIRST = 1) and transmit FIFO data register reset (SSIFCR.TFRST = 1). The software reset and transmit FIFO data register reset have priority over all the clearing conditions described above.

**Note 2.** The state of reading this bit as one is cleared by any of the following four conditions being met:
- Software reset (SSIFCR.SSIRST = 1)
- Resetting of the transmit FIFO data register (SSIFCR.TFRST = 1)
- Completion of writing 0 to this flag after having read it as 1.
- When writing data to SSIFTDR.

[Setting condition]

SSIFTDR has free space not less than the amount set with the SSISCR.TDES bit plus one.

[Setting timing]

While operating on P0ϕ, SSIFTDR is found to have free space not less than “size set in the SSISCR.TDES bits + 1.”

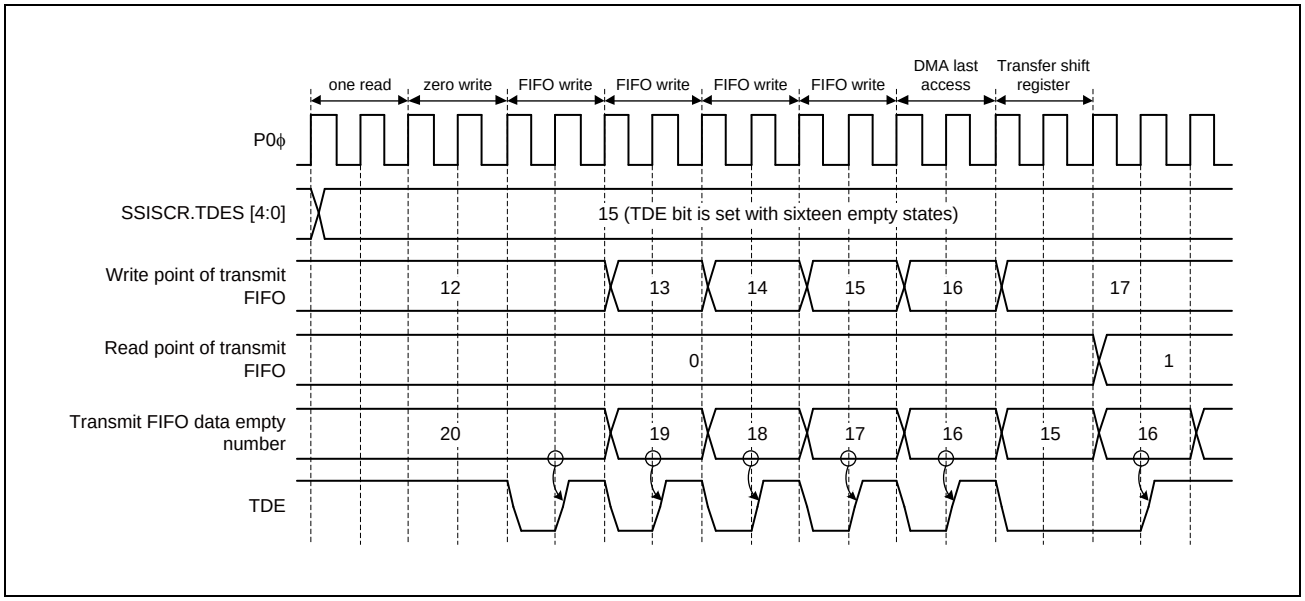



Figure 27.27 Set timing and clear timing of TDE

### RDC[5:0] Bits

These bits indicate the number of valid data that are stored in the receive FIFO data register (SSIFRDR). With this flag as H’0, there is no received data. With H’20, the register is filled with received data and there is no free space.

### RDF Bit

This bit indicates that the receive FIFO data register (SSIFRDR) has unread received data not less than the amount set with the SSISCR.RDFS bit plus one. This flag is set by automatic determination but it must be cleared by register access.

[Priority order for setting and clearing]

Clearing is prioritized.*1

[Clearing condition]

Either of the following two:

(1) Writing 0 to this bit after reading this bit as 1.*2

(2) When reading data from SSIFRDR.

[Clearing timing]

Timing according to the above-mentioned clear condition.

(1) At completion of writing 0 to this bit after reading this bit as 1 (**Figure 27.13**).

(2) After reading data from SSIFRDR.

**Note 1.** This bit is also cleared by a software reset (SSIFCR.SSIRST = 1) and receive FIFO data register reset (SSIFCR.RFRST = 1). Clearing conditions available for these bits are the software reset and receive FIFO data register reset as well as the clearing conditions described above.

**Note 2.** The state of reading this bit as one is cleared by any of the following four conditions being met:
- Software reset (SSIFCR.SSIRST = 1)
- Receive FIFO data register reset (SSIFCR.RFRST = 1)
- Completion of writing 0 to this flag after having read it as 1.
- When reading data from SSIFRDR.

[Setting condition]

SSIFRDR has data not less than the amount set with the SSISCR.RDFS bit plus one.

[Setting timing]

At completion of transfer from the shift register that results in SSIFRDR having data not less than the amount set with the SSISCR.RDFS bit plus one.

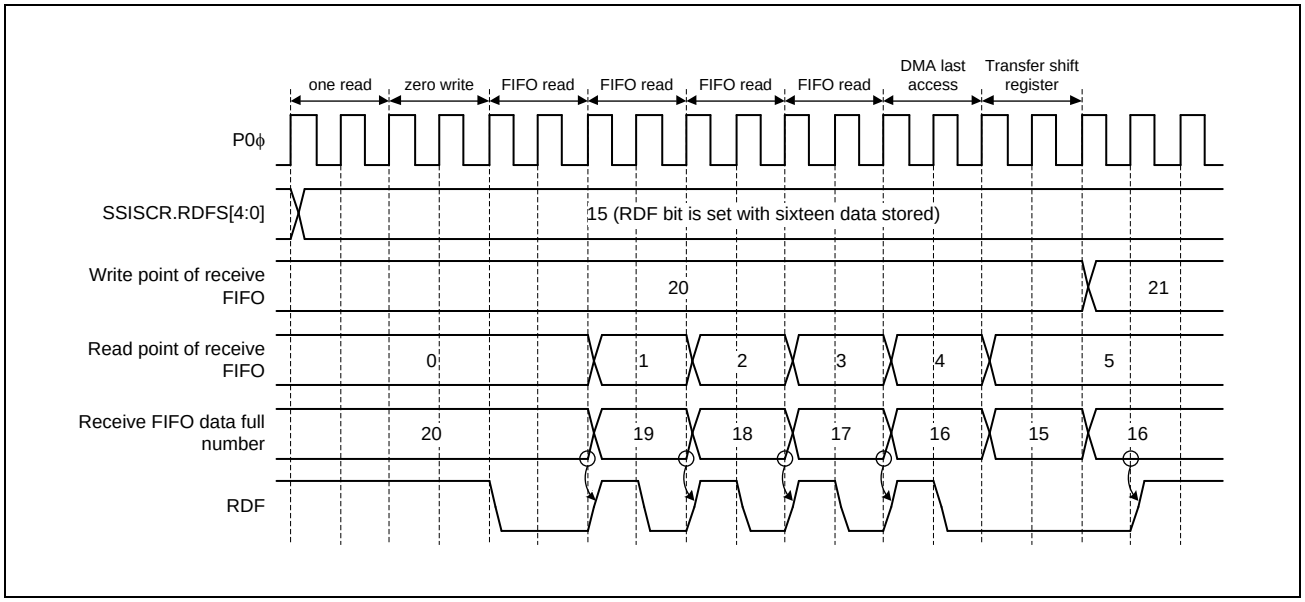
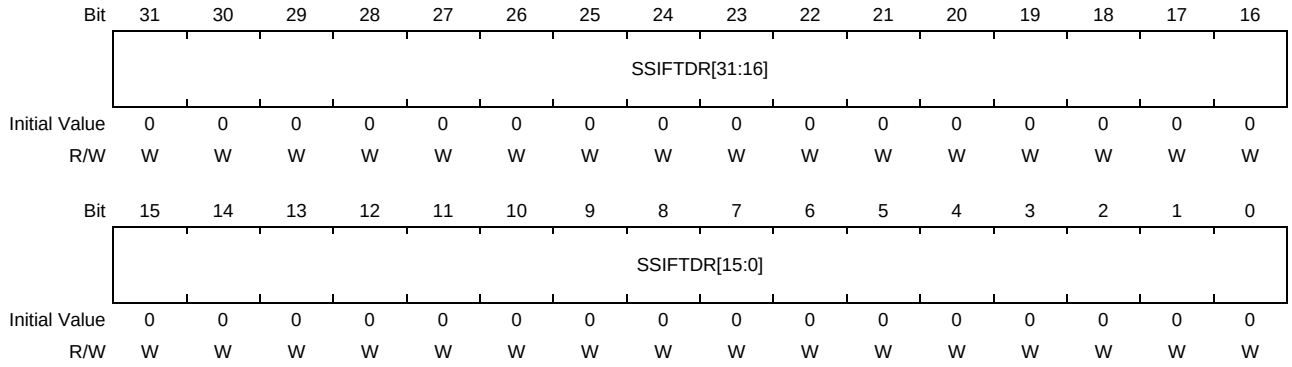



Figure 27.28 Set timing and clear timing of RDF

### 27.4.1.5 Transmit FIFO Data Register (SSIFTDR)

This is a 32-bit writable register. 0 is returned when this register is read. This register stores data to be serially transmitted.




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 0 | SSIFTDR[31:0] | H'00000000 | W | Transmit FIFO data |

To use this register for transmission, set DMA operation handling of a transmit data empty interrupt as writing to this register. Determine the access size to this register according to the data word length to be communicated (**Table 27.10**).

Table 27.10 Register Access Restriction to FIFOs

| Access Size | | | | |
|---|---|---|---|---|
| SSICR.DWL[2:0] | Data Word Length | Byte | Halfword | Word |
| 000b | 8 | ✓ | — | — |
| 001b | 16 | — | ✓ | — |
| 010b | 18 | — | — | ✓ |
| 011b | 20 | — | — | ✓ |
| 100b | 22 | — | — | ✓ |
| 101b | 24 | — | — | ✓ |
| 110b | 32 | — | — | ✓ |
| 111b | Setting prohibited | — | — | — |

**Figure 27.29** shows register access to the transmit FIFO data register.

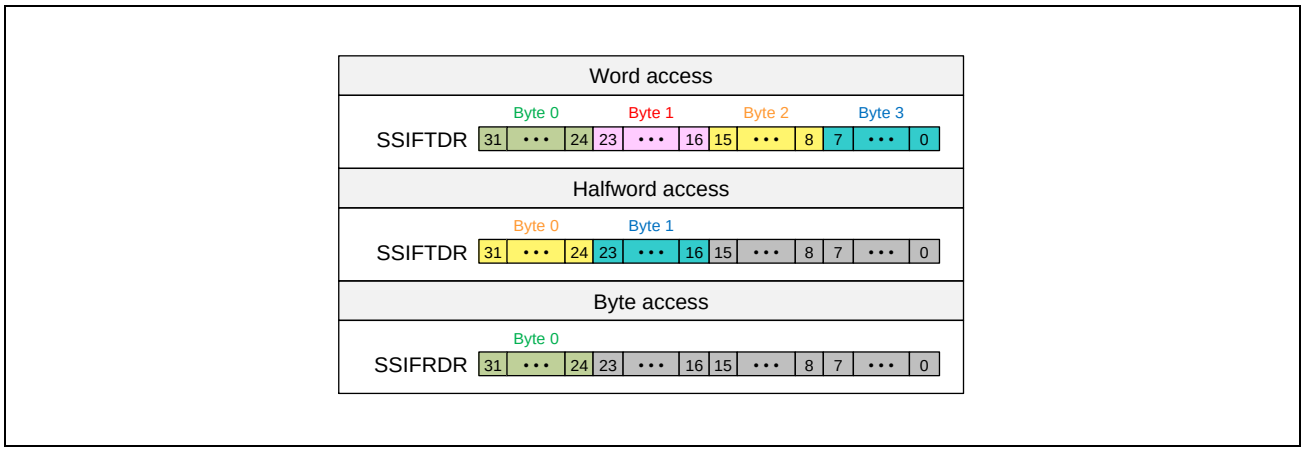



Figure 27.29 Example of Register Access to the Transmit FIFO Data Register

**Figure 27.30** shows the configurations and operation examples of the transmit FIFO data register and transmit shift register. The configurations are for storing data to FIFO and not related with communication.

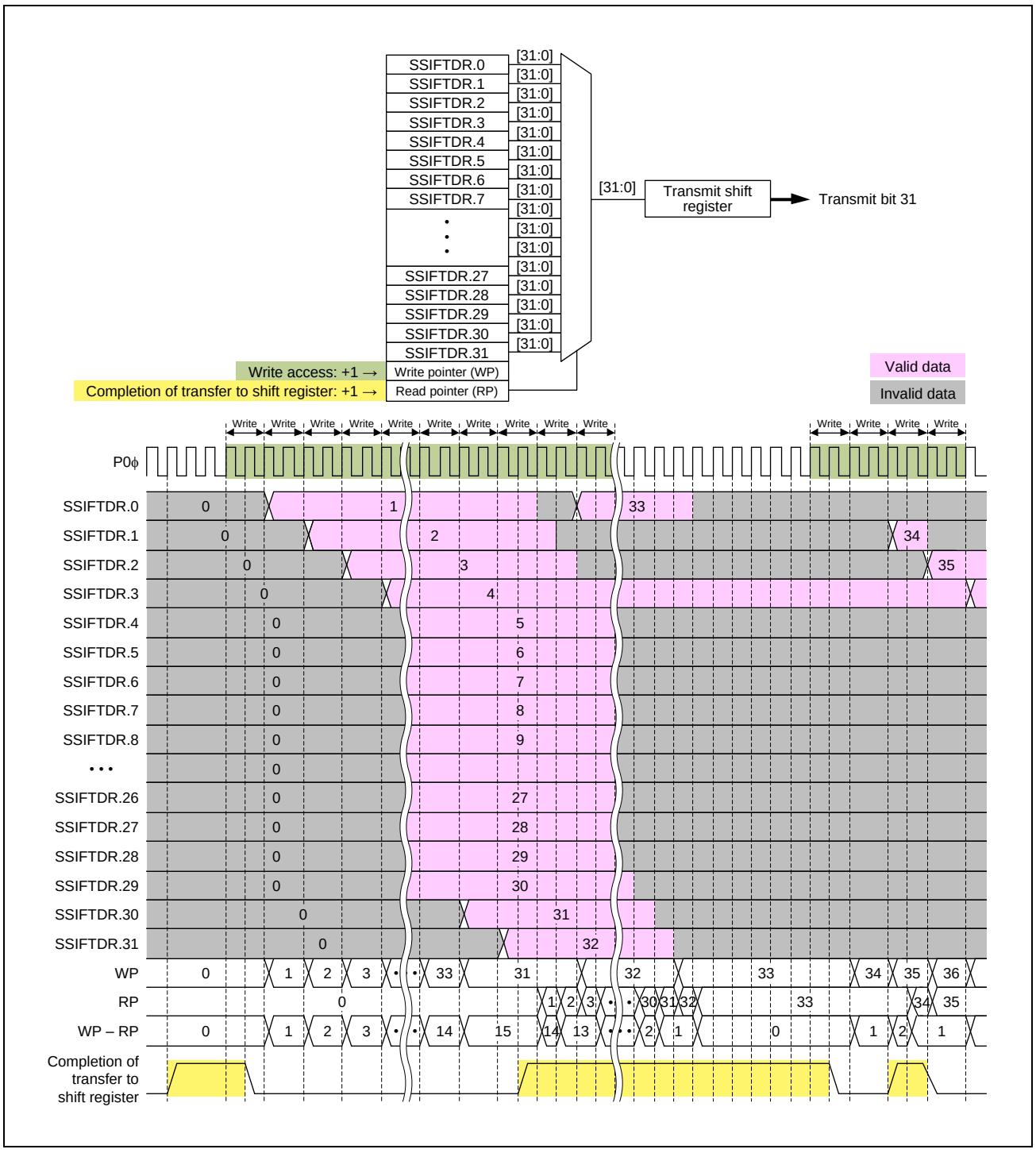



Figure 27.30 Configuration of the Transmit FIFO Data Register and Transmit Shift Register, and FIFO Operation Example

### 27.4.1.6 Receive FIFO Data Register (SSIFRDR)

This is a readable 32-bit register. This register stores received serial data.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | SSIFRDR[31:16] | | | | | | | | | | | | | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | SSIFRDR[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 0 | SSIFRDR[31:0] | H'00000000 | R | Receive FIFO data |

To use this register for reception, set DMA operation handling of a receive data full interrupt as reading from this register. Determine the access size to this register according to the data word length to be communicated (**Table 27.10**). Register access to the receive FIFO data register is same as for the transmit FIFO data register (**Figure 27.29**).

**Figure 27.31** shows the configurations and operation examples of the receive FIFO data register and receive shift register.

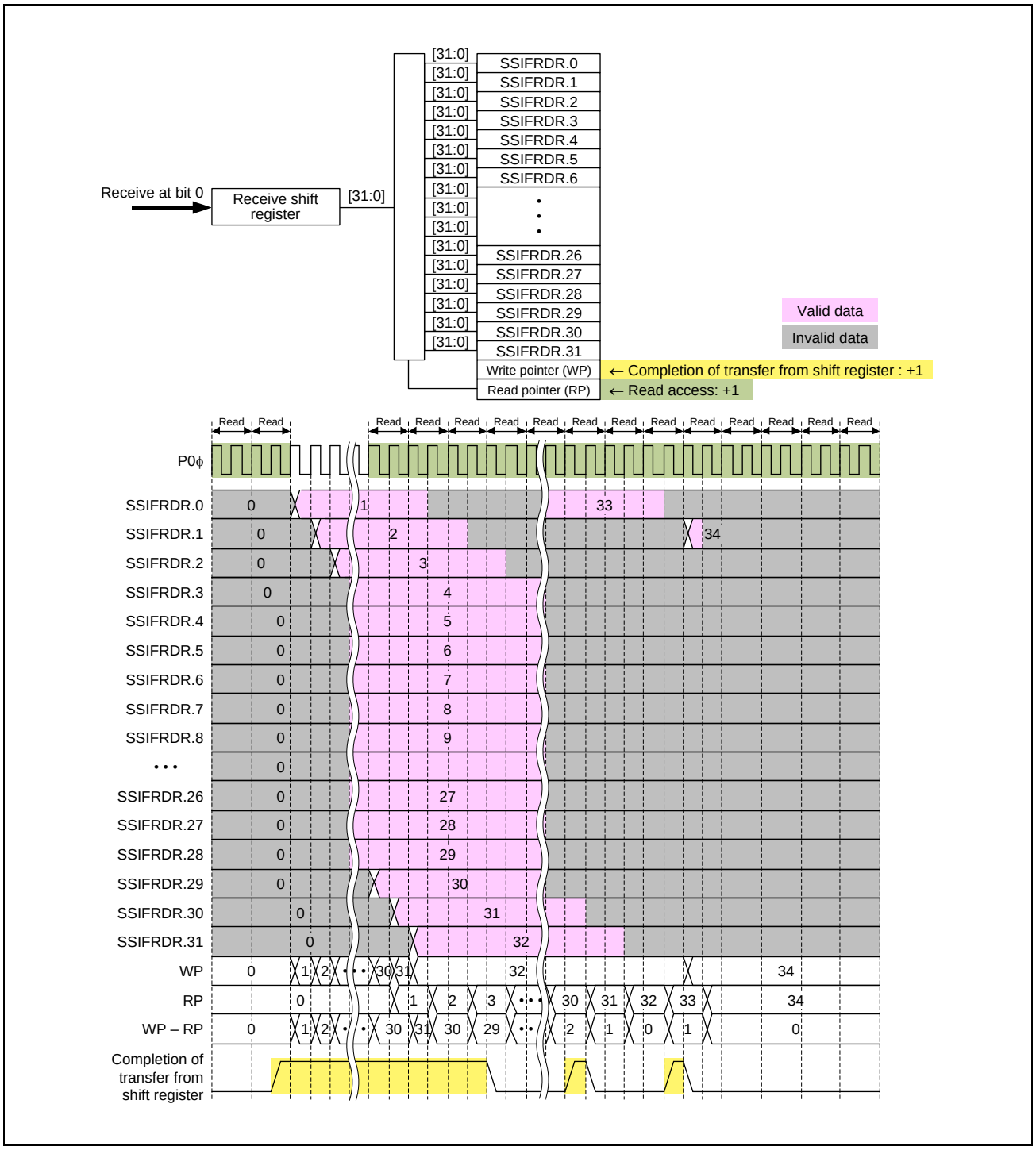



Figure 27.31 Configuration of the Receive FIFO Data Register and Receive Shift Register, and FIFO Operation Example

### 27.4.1.7 Audio Format Register (SSIOFR)

This is a readable/writable 32-bit register. It sets an audio format including the communication format, LRCK/FS continuation mode, and BCK output stop.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | BCKSTP | LRCONT | — | — | — | — | — | — | OMOD[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/W | R/W | R | R | R | R | R | R | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 10 | — | H'000000 | R | Reserved. Write 0. The read value is 0. |
| 9 | BCKASTP | 0 | RW | BCK Output Stop Enable*1,*2<br>0: Enables output of BCK<br>1: Disables output of BCK |
| 8 | LRCONT | 0 | RW | LRCK/FS Continuation Enable*1,*2<br>0: Disables LRCK continuation<br>1: Enables LRCK continuation |
| 7 to 2 | — | H'00 | R | Reserved. Write 0. The read value is 0. |
| 1, 0 | OMOD[1:0] | 0 | RW | Audio Format Select*3,*4<br>OMOD[1:0]: Format<br>00: $I^2S$ format<br>01: TDM format<br>10: Monaural format<br>11: Setting prohibited |

Note 1. This bit is valid only in master-mode communication (SSICR.MST = 1). The setting is invalid in slave-mode communication (SSICR.MST = 0).

Note 2. BCKASTP=1 and LRCONT=1 is prohibitions.

Note 3. Writing in this bit is a prohibition when SSIF-2 is communication state (SSISR.IIRQ=0). Operation is not guaranteed when rewriting it.

Note 4. Use the communication format setting that can be communicated when the format of the communication of the opposing device is compatible format of SSIF-2.

#### BCKASTP Bit

This bit turns on or off the function to output BCK to the SSIBCK pin according to the communication shown in **Figure 27.32** and **Figure 27.33** in master-mode communication (SSICR.MST = 1). Set this bit after setting completing the communication format.

This bit must defend the following usage. Start communication with BCKASTP = 0, and set 1 to BCKASTP while communicating. As a result, the bit clock output to the terminal SSIBCK stops by the automatic operation when the communication stop is done. When you will restart the communication, set 0 to BCKASTP with idle state (SSICR.IIRQ=1) and the AUDIO_MCK supply state (SSIFCR.AUCKE = 1)

When master communication (SSICR.MST=1) and idle state (SSICR.IIRQ=1).

Table 27.11 BCKASTP state, and Output state of SSIBCK pin.

| BCKASTP bit | output state of SSIBCK pin |
|---|---|
| 0 | Enable |
| 1 | Disable |

**Note:** When the opposing device who is the slave needs the clock of the terminal SSIBCK before the communication operates, it is not possible to use it. Use it to stop clocking after the communication ends (**Figure 27.33**). Refer to **Figure 27.32** for timing that the function becomes effective.

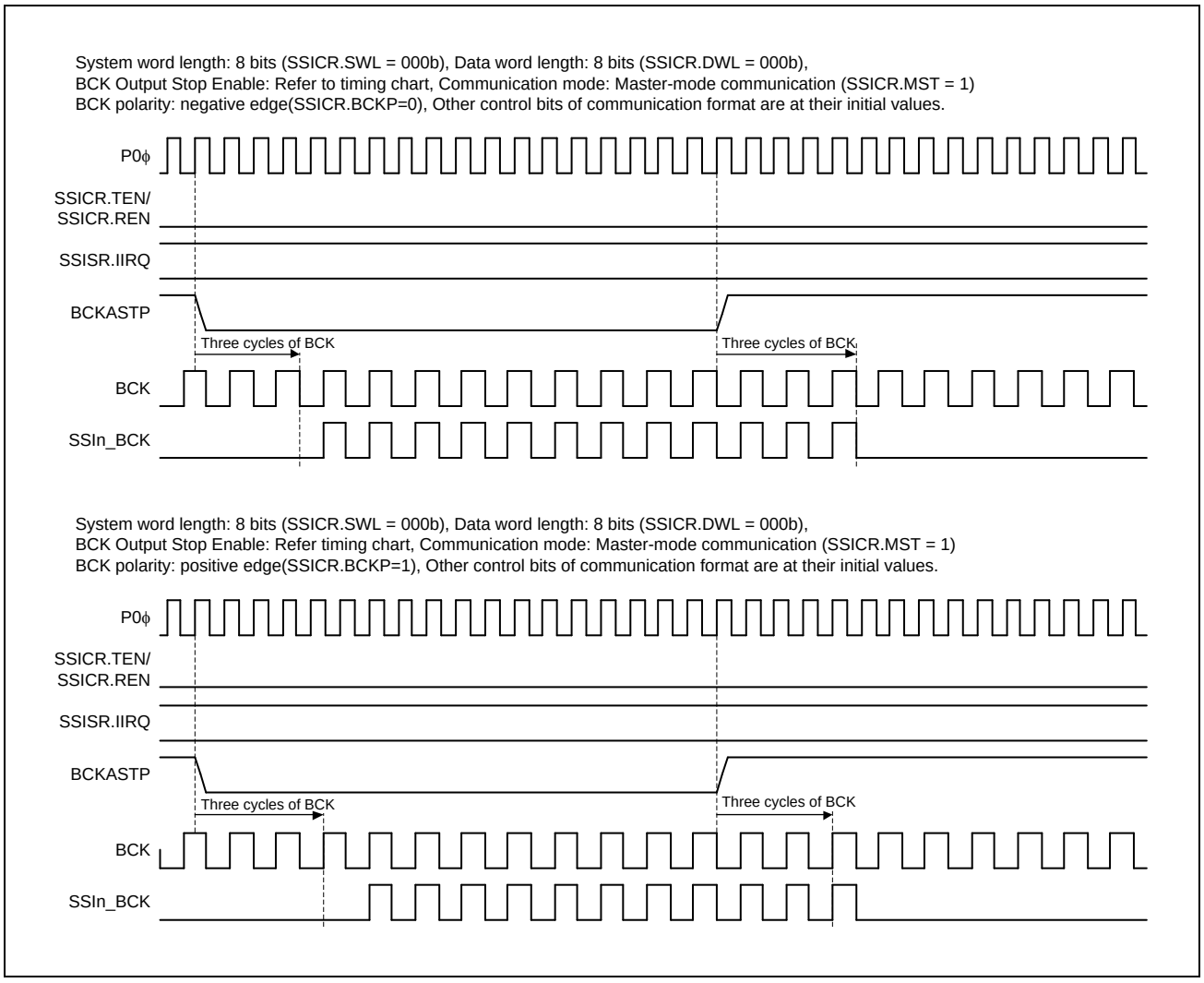



Figure 27.32 Example Operation of the BCKASTP Bit (in idle state)

When master communication (SSICR.MST=1) and the BCK output automatic operation stop function (BCKASTP = 1) (**Figure 27.33**).

Details of the BCK output to SSIBCK pin are as follows.

- Output start timing: To generate an effective edge in timing that LRCK/FS is changed into the valid value, BCK is output.
- Output stop timing: Frame boundary from 1 to 1.5 clock

Refer to the timing chart of **Figure 27.33** for detailed timing.

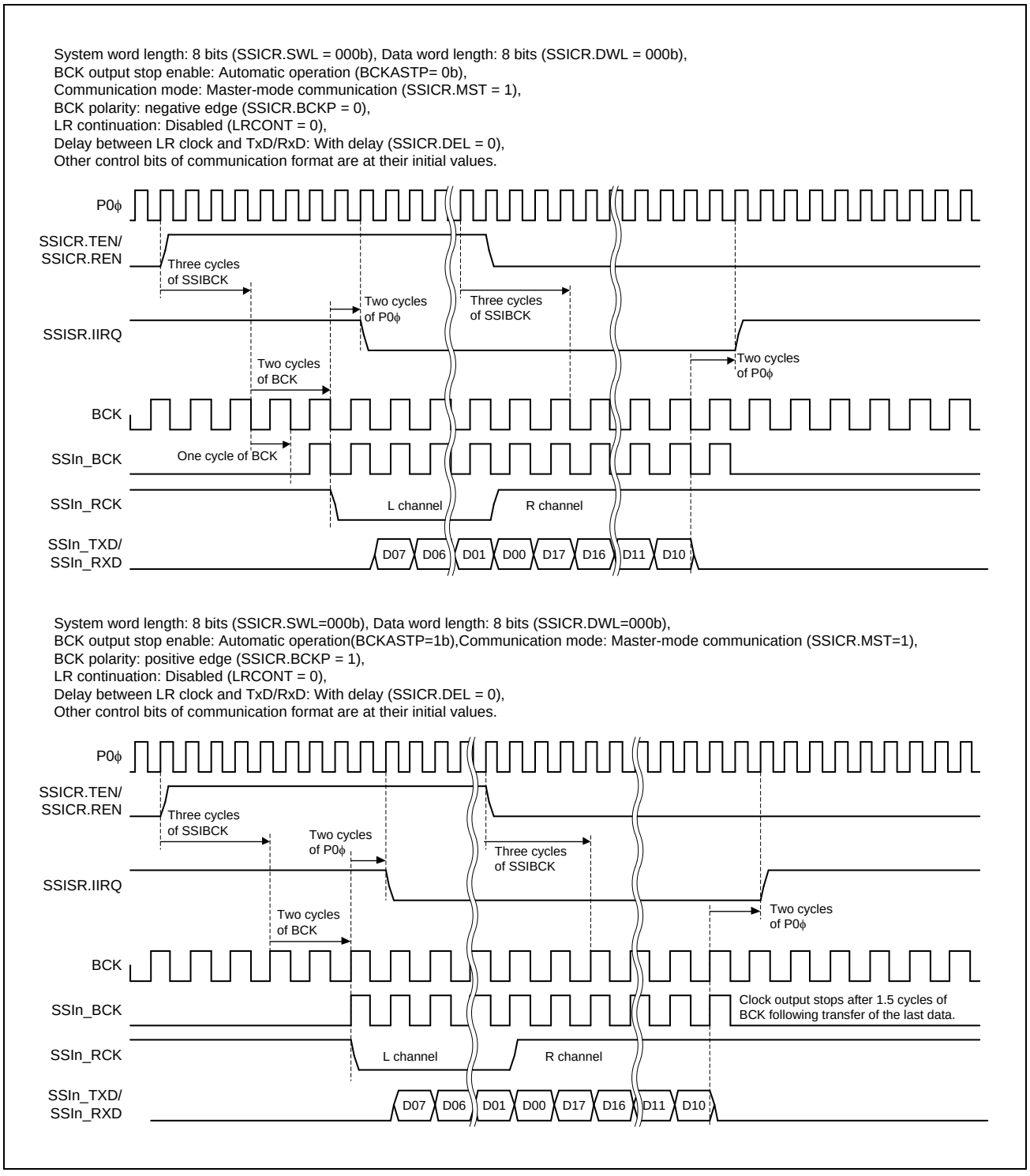



Figure 27.33 Example Operation of the BCKASTP Bit (Communication behavior in BCKASTP=1)

### LRCONT Bit

This bit enables/disables the output from the SSILRCK/SSIFS pin when SSIF-2 is master communication (SSICR.MST=1) and idle state (SSISR.IIRQ=1). Writing 1 to this bit (enables LRCK/FS continuation) in master mode (SSICR.MST = 1) enables continuation of the output from the SSILRCK/SSIFS pins even when idle state.

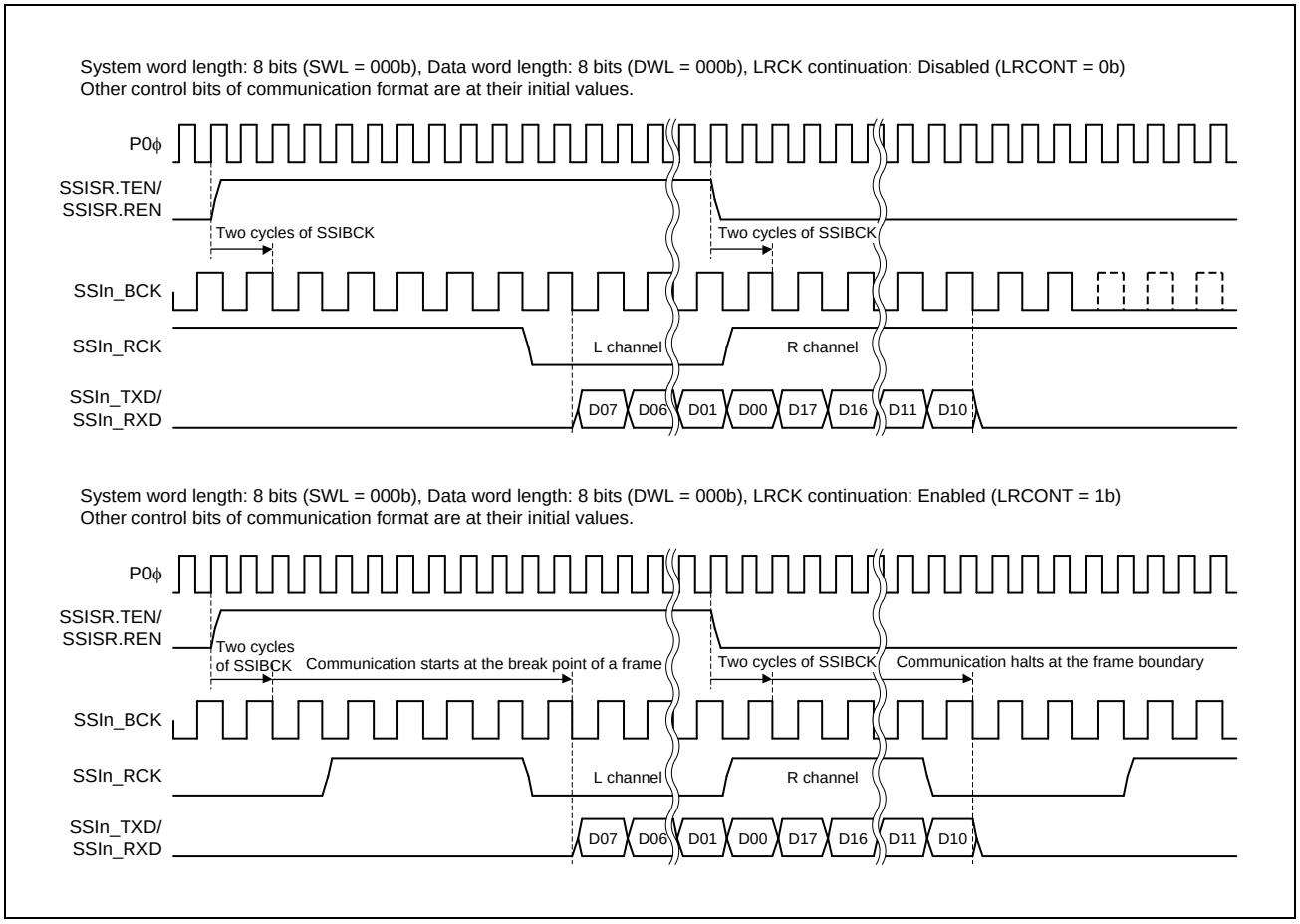



Figure 27.34 Operation Example of LRCK/FS Continuation (in idle state)

### OMOD[1:0] Bits

These bits set an audio format.

Rewrite this bit when the LR clock supply to the terminal SSILRCK stops. Refer to the LRCONT bit details explanation in **Section 27.4.1.7, Audio Format Register (SSIOFR)** for the output operation of the LR clock.

### 27.4.1.8 Status Control Register (SSISCR)

This is a readable/writable 32-bit register. It sets the operation of the TDE and RDF flags.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | TDES[4:0] | | | | | — | — | — | RDFS[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/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 13 | — | H'00000 | R | Reserved. Write 0. The read value is 0. |
| 12 to 8 | TDES[4:0] | H'0 | RW | TDE Setting Condition Select*1<br>TDES[4:0]: Operation<br>00000b: SSIFTDR has one stage or more free space<br>00001b: SSIFTDR has two stages or more free space<br>(snip)<br>11110b: SSIFTDR has thirty-one stages or more free space<br>11111b: SSIFTDR has thirty-two stages or more free space |
| 7 to 5 | — | H'0 | R | Reserved. Write 0. The read value is 0. |
| 4 to 0 | RDFS[4:0] | H'0 | RW | RDF Setting Condition Select*1<br>RDFS[4:0]: Operation<br>00000b: SSIFRDR has one stage or more data size<br>00001b: SSIFRDR has two stages or more data size<br>(snip)<br>01110b: SSIFRDR has thirty-one or more data size<br>01111b: SSIFRDR has thirty-two stages or more data size |

Note 1. Writing to these bits while SSIF-2 is in a communication state (SSISR.IIRQ = 0) is prohibited. If written, the operation performed immediately after writing is not guaranteed.

#### TDES[4:0] Bits

These bits set the setting condition of the transmit data empty flag (TDE). Set these bits in byte units.

#### RDFS[4:0] Bits

These bits set the setting condition of the receive data full flag (RDF). Set these bits in byte units.

### 27.4.2 Communication Formats

SSIF-2 supports three communication formats (**Table 27.12**).

Table 27.12 Supported Communication Formats

| Communication Format | SSIOFR.OMOD[1:0] |
|---|---|
| $I^2S$ format | 00 |
| TDM format | 01 |
| Monaural format | 10 |

The following figure shows the serial data configuration common to communication formats. Serial data is determined according to the system word length (SSICR.SWL[2:0]) and the data word length (SSICR.DWL[2:0]). When the system word length is longer than the data word length, padding bits are transferred (**Figure 27.35**).

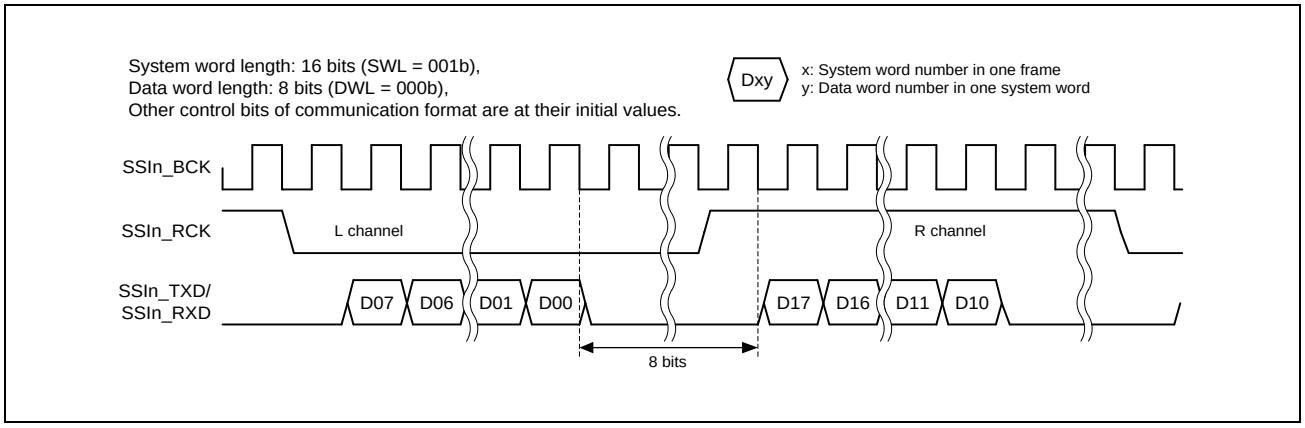



Figure 27.35 Example of Padding Bit Transfer ($I^2S$ Format: System Word Length > Data Word Length)

**Table 27.13** lists the number of padding bits to be transferred with each combination of system word length (SSICR.SWL[2:0]) and data word length (SSICR.DWL[2:0]). "-" indicates that the setting is prohibited.

Table 27.13 Number of Padding Bits

| | SSICR.DWL[2:0] | 000b | 001b | 010b | 011b | 100b | 101b | 110b | 111b |
|---|---|---|---|---|---|---|---|---|---|
| SSICR.SWL[2:0] | System Word Length | 8 | 16 | 18 | 20 | 22 | 24 | 32 | Set prohibition |
| 000b | 8 | 0 | — | — | — | — | — | — | — |
| 001b | 16 | 8 | 0 | — | — | — | — | — | — |
| 010b | 24 | 16 | 8 | 6 | 4 | 2 | 0 | — | — |
| 011b | 32 | 24 | 16 | 14 | 12 | 10 | 8 | 0 | — |
| 100b | 48 | 40 | 32 | 30 | 28 | 26 | 24 | 16 | — |
| 101b | 64 | 56 | 48 | 46 | 44 | 42 | 40 | 32 | — |
| 110b | 128 | 120 | 112 | 110 | 108 | 106 | 104 | 96 | — |
| 111b | 256 | 248 | 240 | 238 | 236 | 234 | 232 | 224 | — |

### 27.4.2.1 I²S Format

The I²S format is a communication format used for connection with I²S-compatible serial devices. With this format setting (SSIOFR.OMOD[1:0] = 00b), one frame is configured with two system words, one for the channel L and the other for channel R. The SSILRCK/SSIFS signals are at a low level for the channel L and at a high level for the channel R. Set the polarity of the signals with the SSICR.LRCKP bit. **Figure 27.36** shows the I²S format without padding. See **Figure 27.35** for the format with padding.

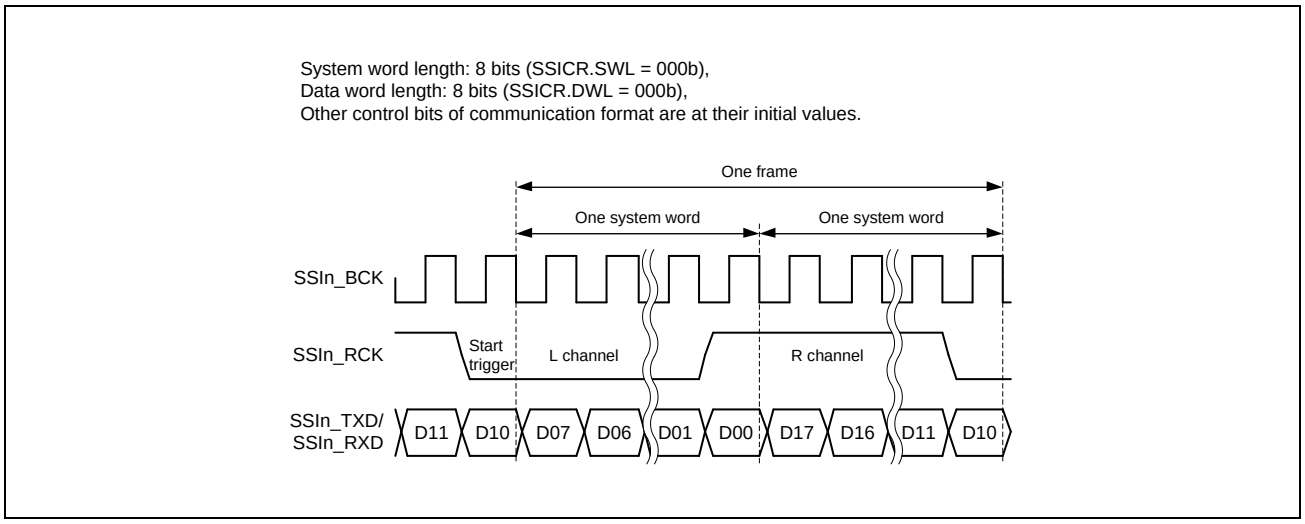



Figure 27.36 I²S Format (Without Padding: System Word Length = Data Word Length)

For the state of external pins when SSIF-2 is in the idle state, see **Section 27.5.1.1, Idle State**.

NOTE

SSIF-2 has the terminal SSILRCK/SSIFS that shows the synchronization of the communication. Match communication format of the opposing device and SSIF-2 when SSIF-2 is slave communication (SSICR.MST=0). SSIF-2 uses the signal of SSILRCK/ SSIFS only as a communication start trigger.

### 27.4.2.2 Monaural Format

The monaural format is a communication format used for connection with monaural-compatible serial devices. With this format setting (SSIOFR.OMOD[1:0] = 10b), one frame is configured with the number of system words set with the SSICR.FRM[1:0] bits (**Figure 27.37**). A rising of the SSILRCK/SSIFS signal means a start trigger. **Figure 27.37** and **Figure 27.38** respectively show the monaural formats without and with padding.

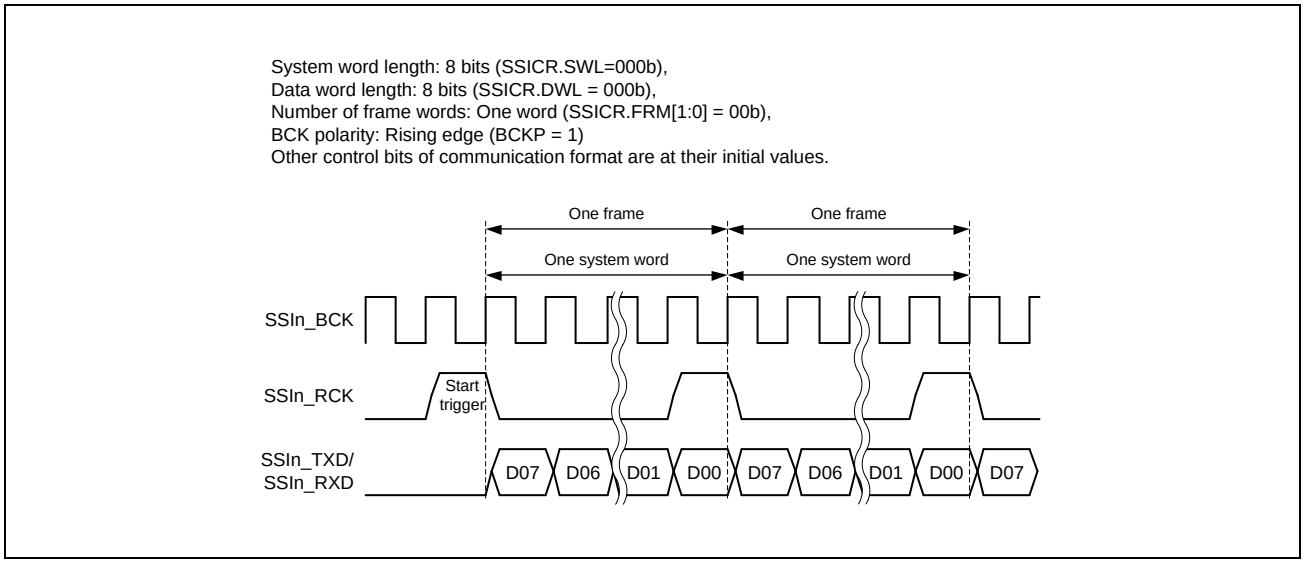



Figure 27.37 Short Frame in Monaural Format (Without Padding: System Word Length = Data Word Length)

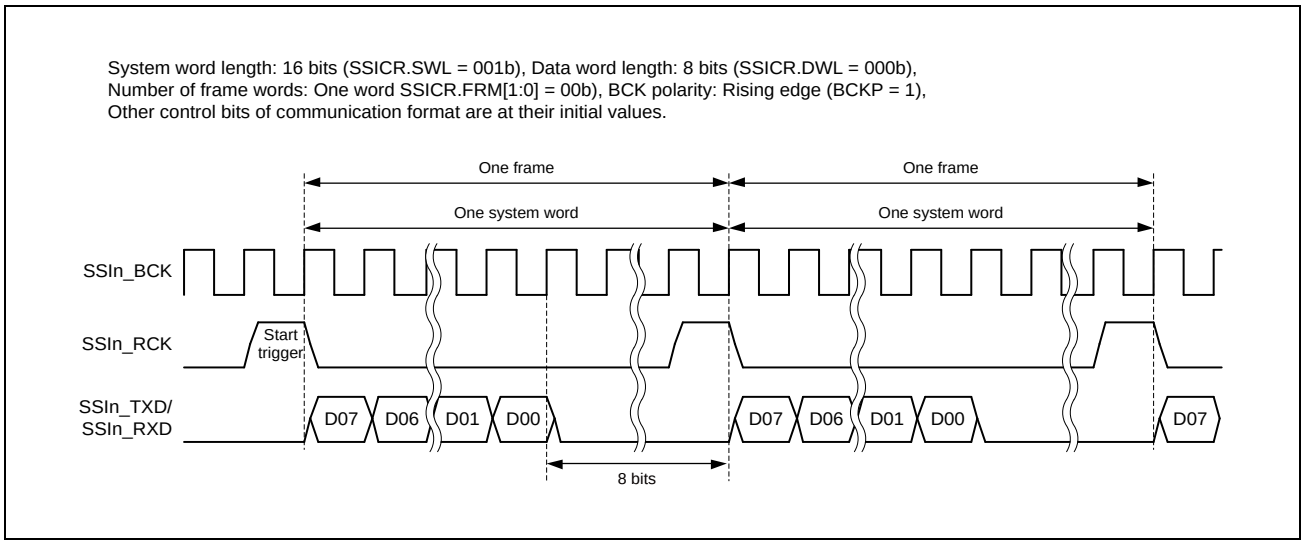



Figure 27.38 Short Frame in Monaural Format (With Padding: System Word Length > Data Word Length)

The monaural formats supported by SSIF-2 consist of short frames and long frames. See **Section 27.4.2.2(1), Short Frame** and **Section 27.4.2.2(2), Long Frame** for the difference between these two frames.

For the state of external pins state when SSIF-2 is in the idle state, see **Section 27.5.1.1, Idle State**.

NOTE

SSIF-2 has the terminal SSILRCK/SSIFS that shows the synchronization of the communication. Match communication format of the opposing device and SSIF-2 when SSIF-2 is slave communication (SSICR.MST=0). SSIF-2 uses the signal of SSILRCK/SSIFS only as a communication start trigger.

### (1) Short Frame

With the short frame (SSICR.DEL = 0), the SSILRCK/SSIFS signal that indicates a start of serial data is at a high level for one cycle of SSIBCK. Data transfer starts at the falling edge of the signal (**Figure 27.37** and **Figure 27.38**).

### (2) Long Frame

With the long frame (SSICR.DEL = 1), the SSILRCK/SSIFS signal that indicates a start of serial data is at a high level for two cycles of SSIBCK. Data transfer starts at the rising edge of the signal (**Figure 27.39**).

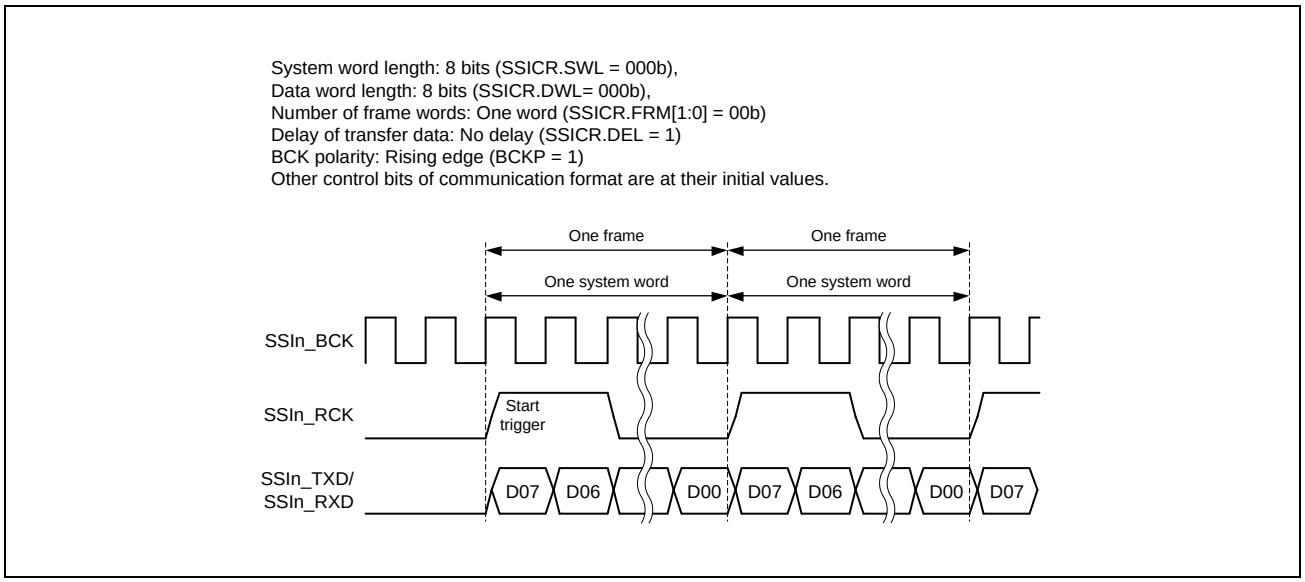



Figure 27.39 Long Frame in Monaural Format (Without Padding)

### 27.4.2.3 TDM Format

The TDM format is a communication format used for connection with TDM-compatible multi-channel devices. With this format setting (SSIOFR.OMOD[1:0] = 01b), one frame is configured with four to eight system words set with the SSICR.FRM[1:0] bits. With this format, the SSILRCK/SSIFS signal is at a high level for the first one system word and at a low level for the rest. The pulse generated on the SSILRCK/SSIFS signal is defined as the SYNC pulse and its rising edge means a start of one frame. **Figure 27.40** and **Figure 27.41** respectively show the TDM formats with and without padding.

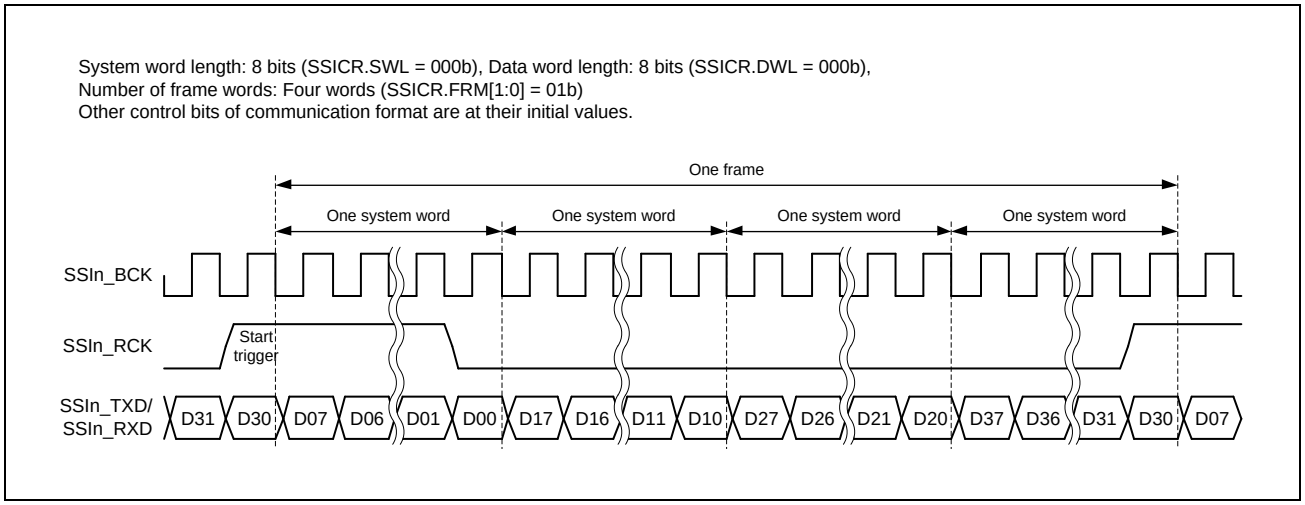



Figure 27.40 TDM Format (Without Padding: System Word Length = Data Word Length)

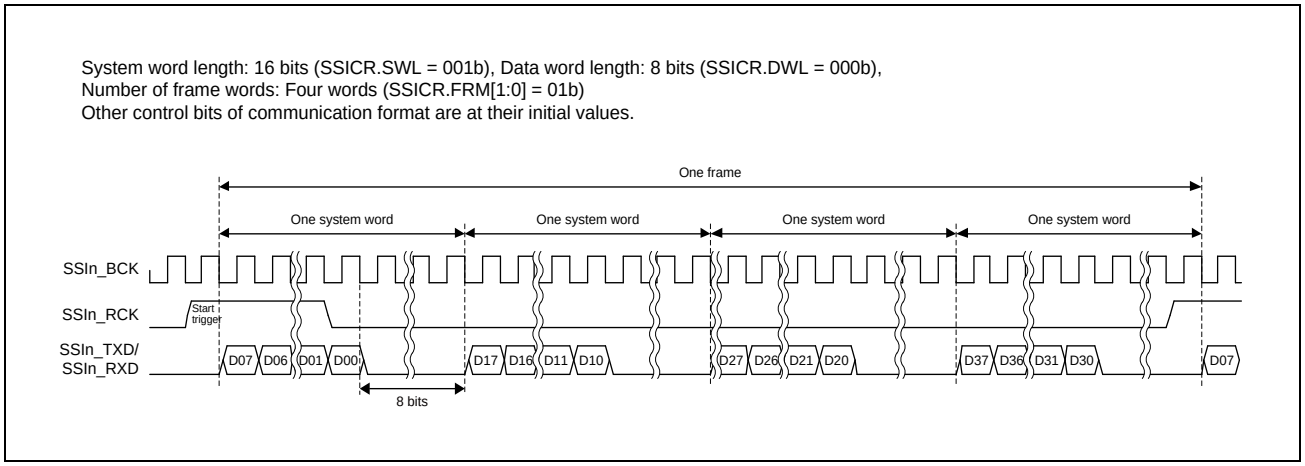



Figure 27.41 TDM Format (With Padding: System Word Length > Data Word Length)

For the state of external pins when SSIF-2 is in the idle state, see **Section 27.5.1.1, Idle State**.

NOTE

SSIF-2 has the terminal SSILRCK/SSIFS that shows the synchronization of the communication. Match communication format of the opposing device and SSIF-2 when SSIF-2 is slave communication (SSICR.MST=0). SSIF-2 uses the signal of SSILRCK/SSIFS only as a communication start trigger.

### 27.4.3 Communication Modes

SSIF-2 supports the following communication modes. **Table 27.15** lists the control bits that are not available with each communication mode. See **Section 27.4.3.1** to **Section 27.4.3.5** for details of these communication modes.

Table 27.14 Communication Modes

| Communication Mode | SSICR.MST Bit | SSICR.REN Bit | SSICR.TEN Bit |
|---|---|---|---|
| Slave-mode transmission | 0 | 0 | 1 |
| Slave-mode reception | 0 | 1 | 0 |
| Slave-mode transmission and reception | 0 | 1 | 1 |
| Master-mode transmission | 1 | 0 | 1 |
| Master-mode reception | 1 | 1 | 0 |
| Master-mode transmission and reception | 1 | 1 | 1 |

Table 27.15 Control Bits that Cannot be Used in Each Communication Mode

| Communication Mode Control Bit | Slave-mode Reception | Slave-mode Transmission | Slave-mode Transmission and Reception | Master-mode Reception | Master-mode Transmission | Master-mode Transmission and Reception |
|---|---|---|---|---|---|---|
| SSICR.CKS | Invalid | Invalid | Invalid | Available | Available | Available |
| SSICR.CKDV | Invalid | Invalid | Invalid | Available | Available | Available |
| SSICR.MUEN | Invalid | Available | Available | Invalid | Available | Available |
| SSICR.TEN | Invalid | Available | Available | Invalid | Available | Available |
| SSICR.REN | Available | Invalid | Available | Available | Invalid | Available |
| SSIFCR.AUCKEN | Invalid | Invalid | Invalid | Available | Available | Available |
| SSIFCR.BCKNCE | Available | Available*1 | Available*2 | Invalid | Invalid | Invalid |
| SSIFCR.LRCKNCE | Available | Available*1 | Available*2 | Invalid | Invalid | Invalid |
| SSIFCR.RXDNCE | Available | Invalid | Available | Invalid | Invalid | Invalid |
| SSIFCR.TIE | Invalid | Available | Available | Invalid | Available | Available |
| SSIFCR.RIE | Available | Invalid | Available | Available | Invalid | Available |
| SSIFCR.TFRST | Invalid | Available | Available | Invalid | Available | Available |
| SSIFCR.RFRST | Available | Invalid | Available | Available | Invalid | Available |
| SSIOFR.BCKASTP | Invalid | Invalid | Invalid | Available | Available | Available |
| SSIOFR.LRCONT | Invalid | Invalid | Invalid | Available | Available | Available |
| SSIOFR.OMOD | Available | Available | Available | Available | Available | Available |
| SSISCR.TDES | Invalid | Available | Available | Invalid | Available | Available |
| SSISCR.RDFS | Available | Invalid | Available | Available | Invalid | Available |

Note 1. Disabled when SSICR.DEL = ‘1’

Note 2. Note 2. Disabled during transmission (SSICR.TEN = ‘1’) when SSICR.DEL = ‘1’

#### 27.4.3.1 Slave-mode Communication

SSIF-2 operates in slave mode with SSICR.MST = 0. Supply the SSIBCK signal and the SSILRCK/SSIFS signal used for the serial data communication from external devices. If these signals do not match the communication format set for SSIF-2, operation is not guaranteed.

#### 27.4.3.2 Master-mode Communication

SSIF-2 operates in master mode with SSICR.MST = 1. The SSIBCK signal and SSILRCK/SSIFS signal used for the serial data communication are internally generated from the audio clock. These signals use the format according to the setting of SSIF-2. If the communication format of another device working as the slave device does not match the communication format set for SSIF-2, operation is not guaranteed.

#### 27.4.3.3 Transmission

SSIF-2 transmits serial data to opposing device when SSICR.TEN = 1 and SSICR.REN = 0. If the communication format of the opposing device does not match the communication format set for SSIF-2, operation is not guaranteed.

#### 27.4.3.4 Reception

SSIF-2 receives serial data from opposing device when SSICR.TEN = 0 and SSICR.REN = 1. If the communication format of the opposing device does not match the communication format set for SSIF-2, operation is not guaranteed.

#### 27.4.3.5 Transmission and Reception

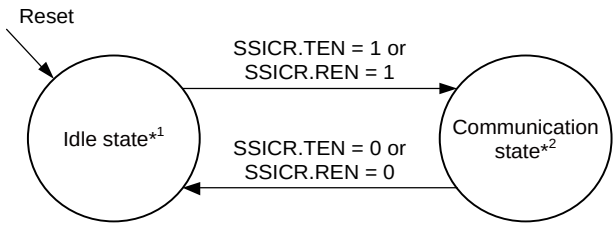
SSIF-2 transmits and receives serial data to and from opposing device when SSICR.TEN = 1, SSICR.REN = 1. If the communication format of opposing device does not match the communication format set for SSIF-2, operation is not guaranteed.

# 27.5 Operation

## 27.5.1 Operational State

SSIF-2 has the following two main operation states (**Figure 27.42**).

- Idle state (SSISR.IIRQ = 1)
- Communication state (SSISR.IIRQ = 0)




Note 1. See **Section 27.5.1.1** for details of the idle state.

Note 2. See **Section 27.5.1.2** for details of the communication state.

Figure 27.42 SSIF-2 State Transition

### 27.5.1.1 Idle State

In this state, communication of SSIF-2 is halted. However, when SSICR.MST = 1, output of BCK and LRCK/FS to external pins is enabled depending on the settings in the SSIOFR.BCKASTP bit and the SSIOFR.LRCONT bit (**Table 27.16**). This function is common to all formats.

Table 27.16 Output from External Pins in the Idle State

| | | | Output from Pins | | |
|---|---|---|---|---|---|
| SSICR.MST | SSIOFR.BCKASTP | SSIOFR.LRCONT | SSIBCK | SSILRCK/SSIFS | SSITxD |
| 0 | — | — | Stop | Stop | Stop |
| 1 | 0 | 0 | Supply | Stop | Stop |
| 1 | 0 | 1 | Supply | Supply | Stop |
| 1 | 1 | 0 | Stop | Stop | Stop |
| 1 | 1 | 1 | Stop | Supply | Stop |

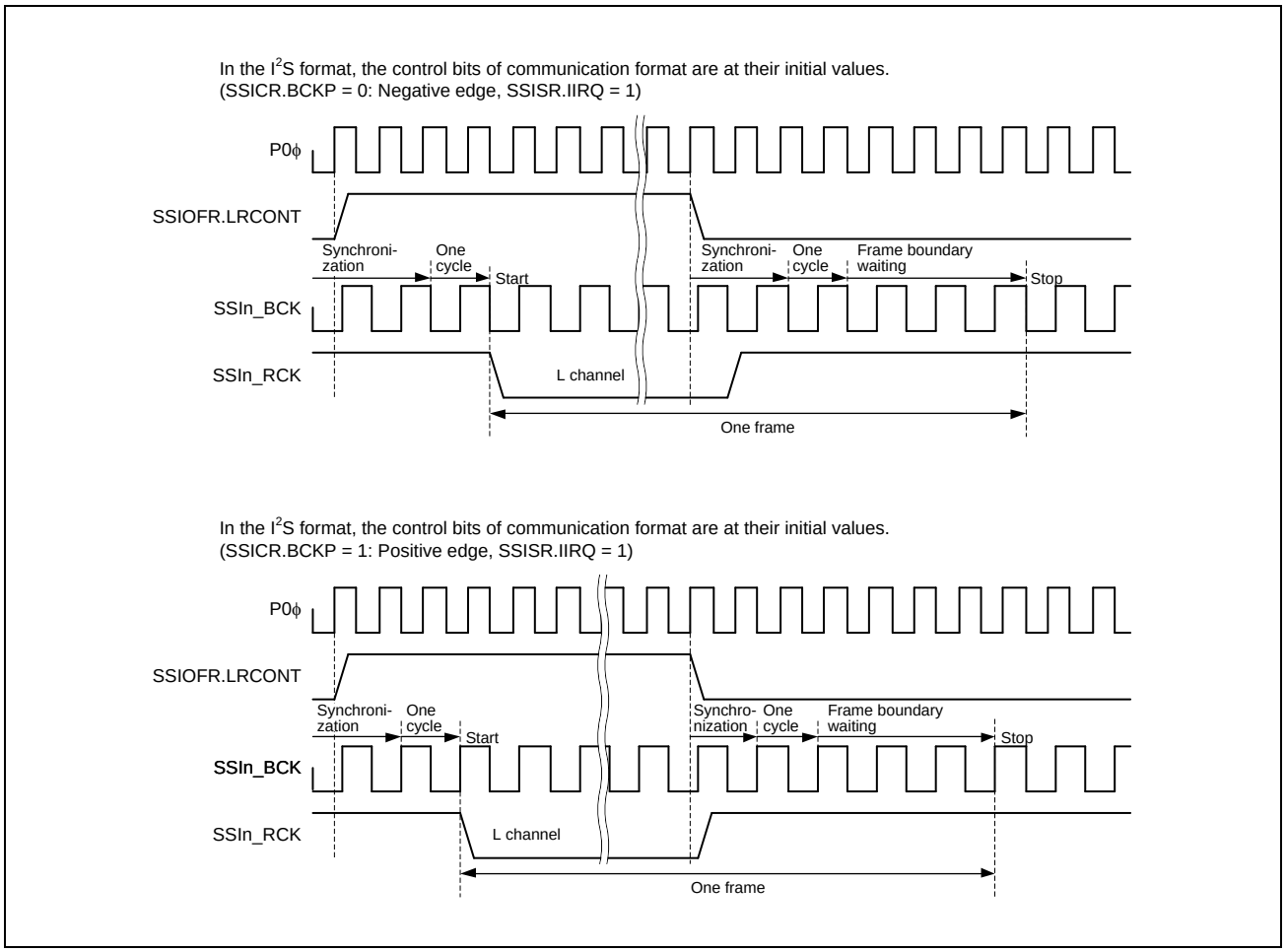



Figure 27.43 Example of LRCK/FS Continuation Release with SSIOFR.LRCONT

NOTE

To stop the output to the SSILRCK/SSIFS pin with SSIOFR.LRCONT when SSIF-2 is in the idle state in master- mode communication (SSICR.MST = 1), note the following: The output stops when 0 is written from 1 to SSIOFR.LRCONT. (**Figure 27.43**). Make sure that the remote device is not affected.

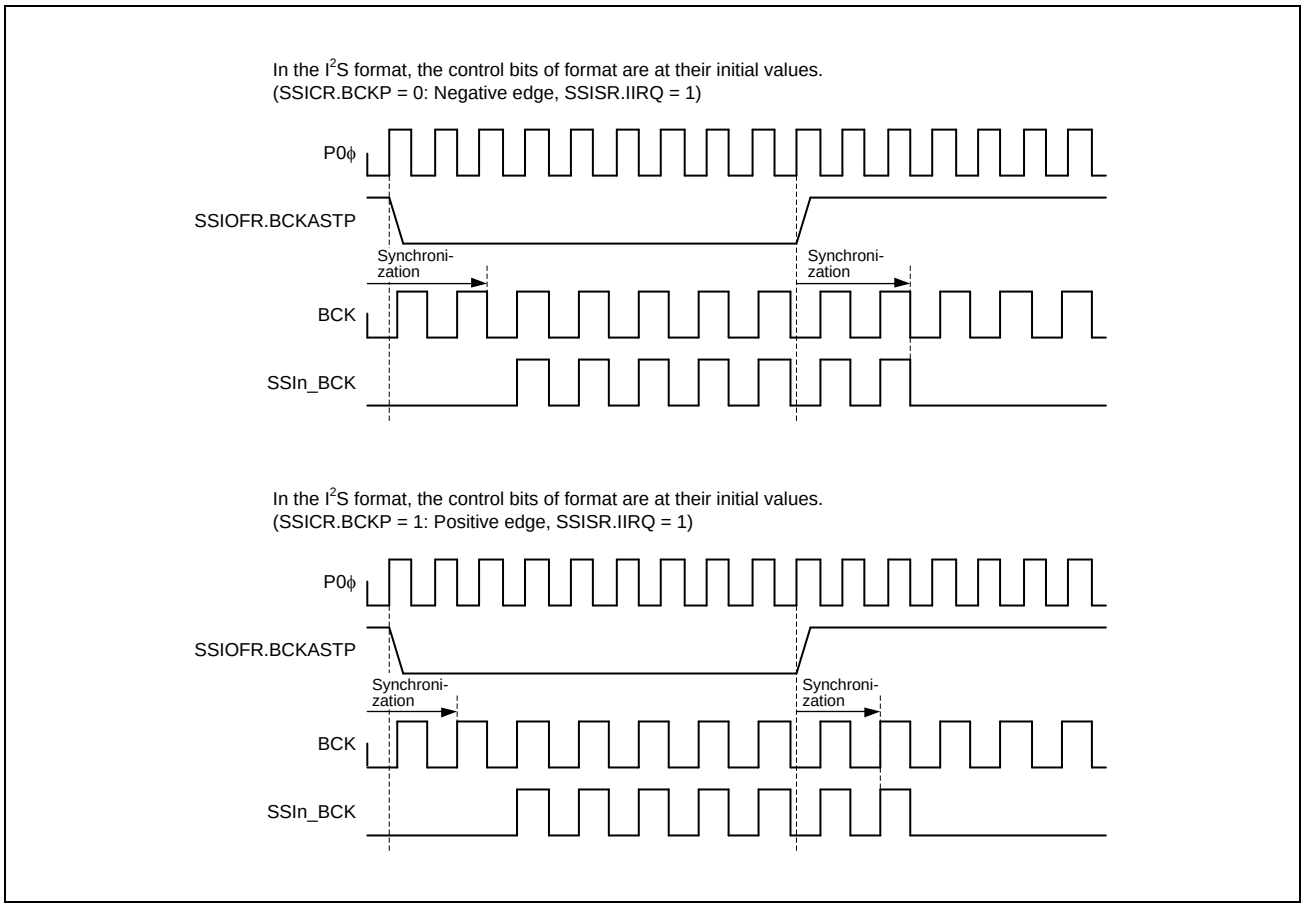



Figure 27.44 Example of Stopping SSIBCK with SSIOFR.BCKASTP

NOTE

To stop the output to the SSIBCK pin with SSIOFR.BCKASTP in master-mode communication (SSICR.MST = 1) and while SSIF-2 is in the idle state, note the following; By writing 0 to SSIOFR.BCKASTP while it is 1, the output stops immediately (**Figure 27.44**). So, make sure that the remote device is not affected.

### 27.5.1.2 Communication States

In this state, SSIF-2 is during communication. **Figure 27.45** shows transitions of communication states and **Table 27.17** lists the conditions for transition. If the transition condition is not satisfied, the state does not transit.

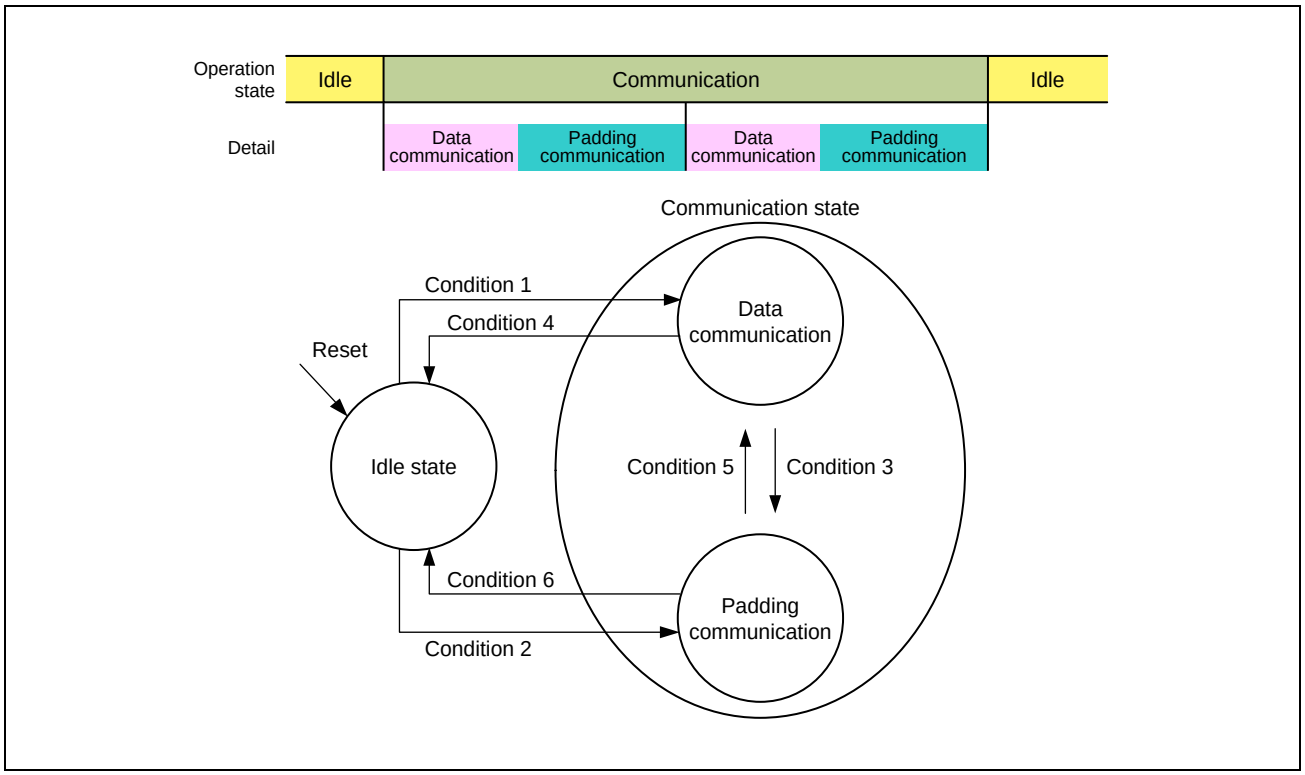



Figure 27.45 Communication State Transition

Table 27.17 Condition for Communication State Transition

| Condition No. | Condition for Transition |
|---|---|
| 1 | Writing SSICR.TEN = 1 or SSICR.REN = 1 while SSICR.SDTA = 0 or in the setting without padding bits. |
| 2 | Writing SSICR.TEN = 1 or SSICR.REN = 1 while SSICR.SDTA = 1 and in the setting with padding bits. |
| 3 | The following all of three conditions are satisfied:<br>• SSICR.TEN = 1 or SSICR.REN = 1<br>• In the setting without padding bits<br>• Transfer data words of the last data bit is completed. |
| 4 | The following all of two conditions are satisfied:<br>• SSICR.SDTA = 1 or without padding bits<br>• Transfer data words of the last data bit of one frame is completed while SSICR.TEN = 0 and SSICR.REN = 0 |
| 5 | Transfer of the last padding bit is completed while SSICR.TEN = 1 or SSICR.REN = 1 |
| 6 | The following all of two conditions are satisfied:<br>• SSICR.SDTA = 0 and with padding bits<br>• Transfer of the last data bit of one frame is completed while SSICR.TEN = 0 and SSICR.REN = 0. |

See **Table 27.13** for the setting with/without padding bits.

### (1) Data Communication State

In this state, SSIF-2 is during communication. Data of the data word length set with the SSICR.DWL[2:0] bits is transmitted, received, or transmitted and received.

#### State Transition in the Setting without Padding Bits

During communication (SSISR.IIRQ = 0), SSIF-2 is during data communication for all the time (**Figure 27.46**). By disabling transmission and reception (SSICR.TEN = 0 and SSICR.REN = 0), SSIF-2 transits to the idle state (**Figure 27.47**).

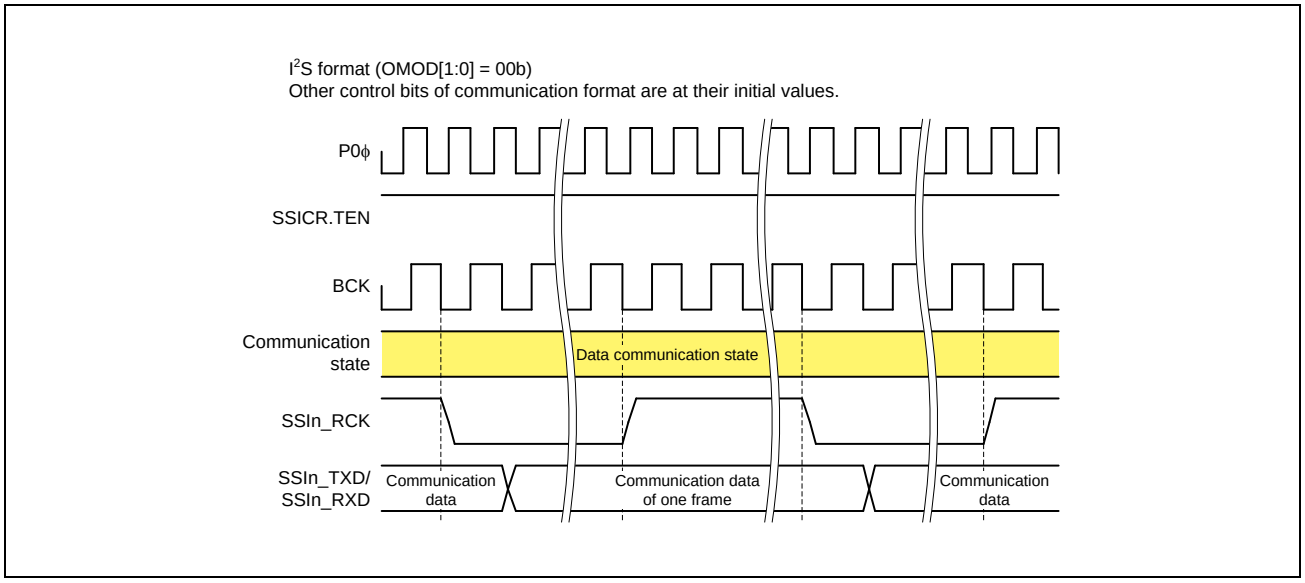



Figure 27.46 Continuation of the Data Communication State

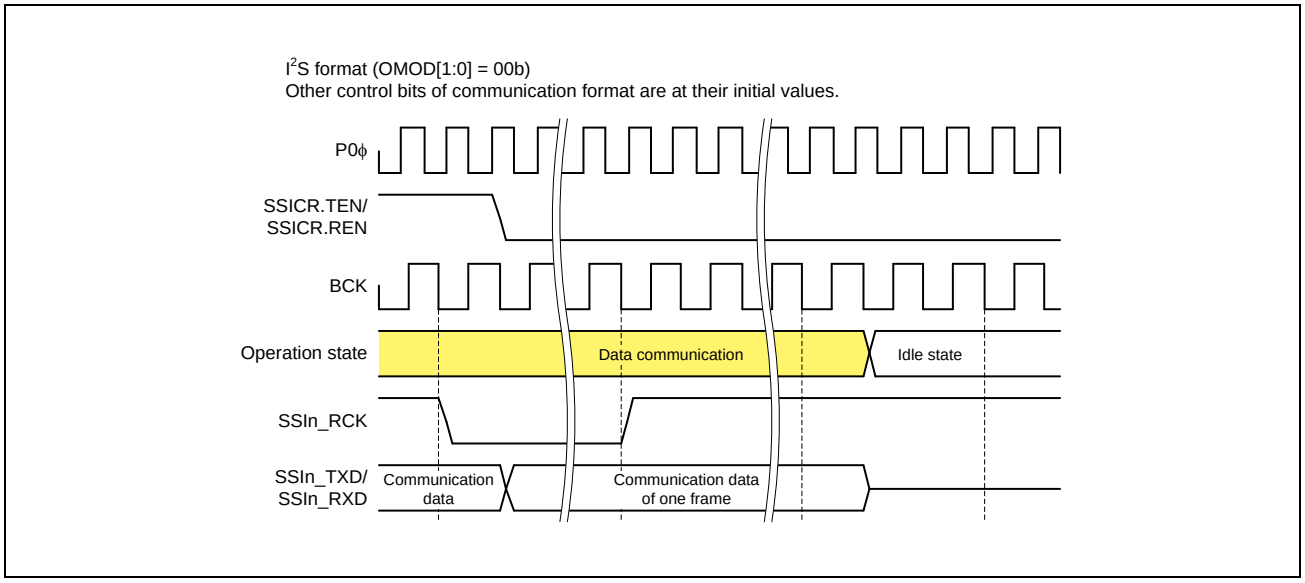



Figure 27.47 Halt from the Data Communication State (without Padding Bits)

### State Transition in the Setting with Padding Bits

During communication (SSISR.IIRQ = 0), after completing transfer of the last bit of a data word, SSIF-2 transits to the padding communication state from data communication state (**Figure 27.48**). By disabling transmission and reception (SSICR.TEN = 0 and SSICR.REN = 0) while SSICR.SDTA = 1, SSIF-2 transits to the idle state from data communication state when the communication stop is done. (**Figure 27.49**).

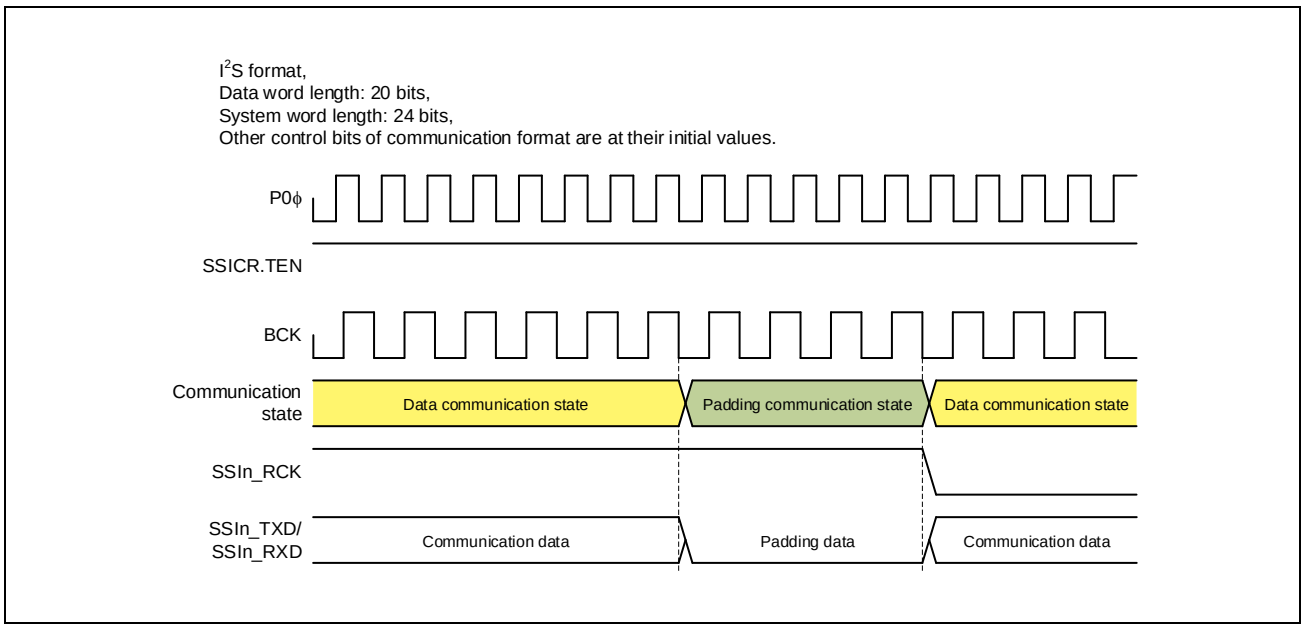



Figure 27.48 Transition from Data Communication to Padding Communication

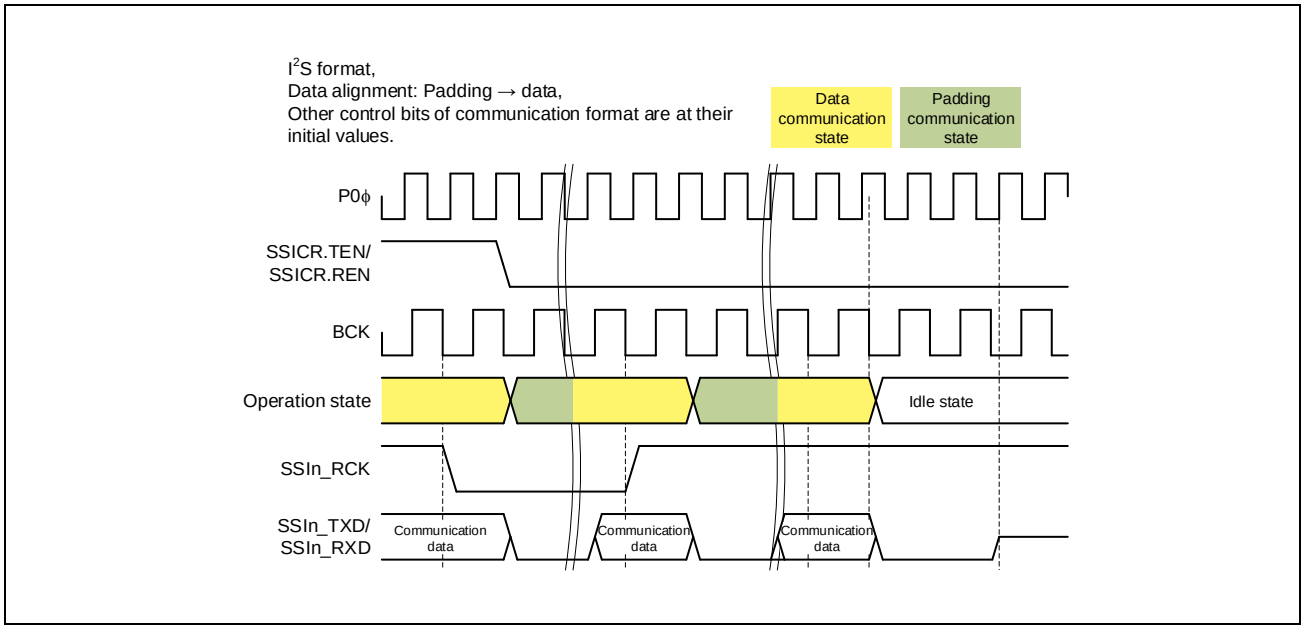



Figure 27.49 Halt from Data Communication (with Padding Bits)

### (2) Padding Communication State

In this state, SSIF-2 is during communication. The padding bits set with the SSICR.SWL[2:0] bits and SSICR.DWL[2:0] bits are transmitted, received, or transmitted and received.

#### State Transition in the Setting with Padding Bits

During communication (SSISR.IIRQ = 0), after completing transfer of the last padding bit, SSIF-2 transits to data communication state (**Figure 27.48**). By disabling transmission and reception (SSICR.TEN = 0 and SSICR.REN = 0) while SSICR.SDTA = 0, SSIF-2 transits to the idle state from padding communication when the communication stop is done. (**Figure 27.50**).

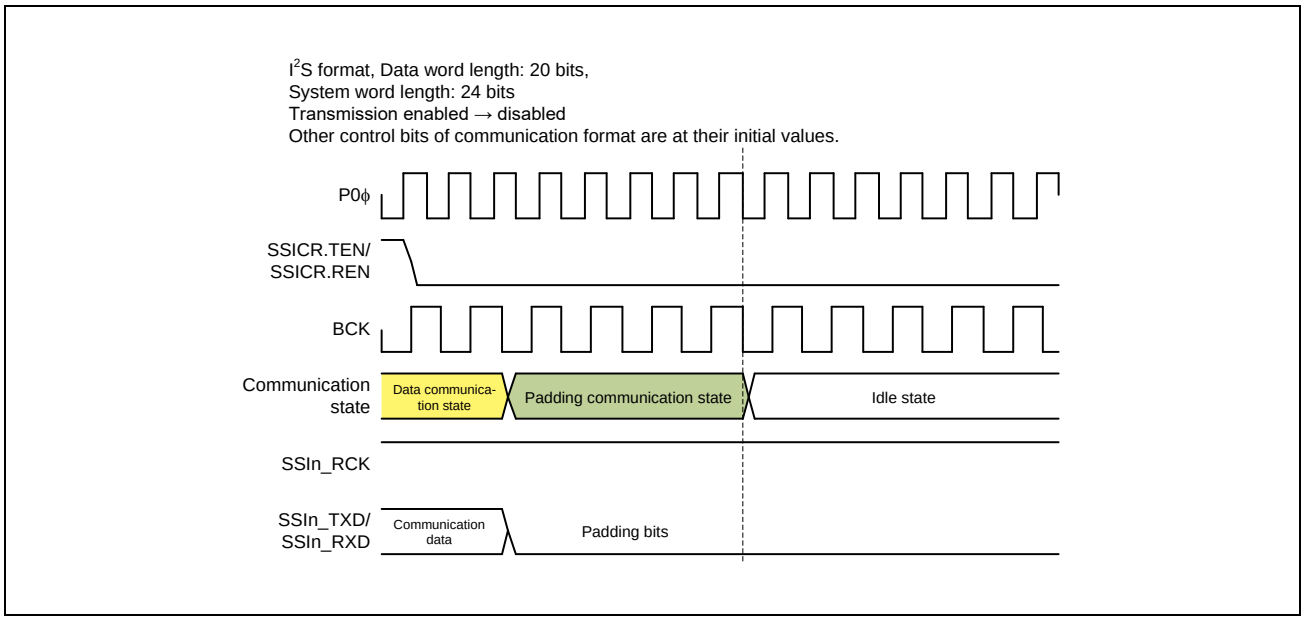



Figure 27.50 Halt from the Padding Communication State

### 27.5.2 Communication Operation

**Figure 27.51** shows the communication flow of SSIF-2.

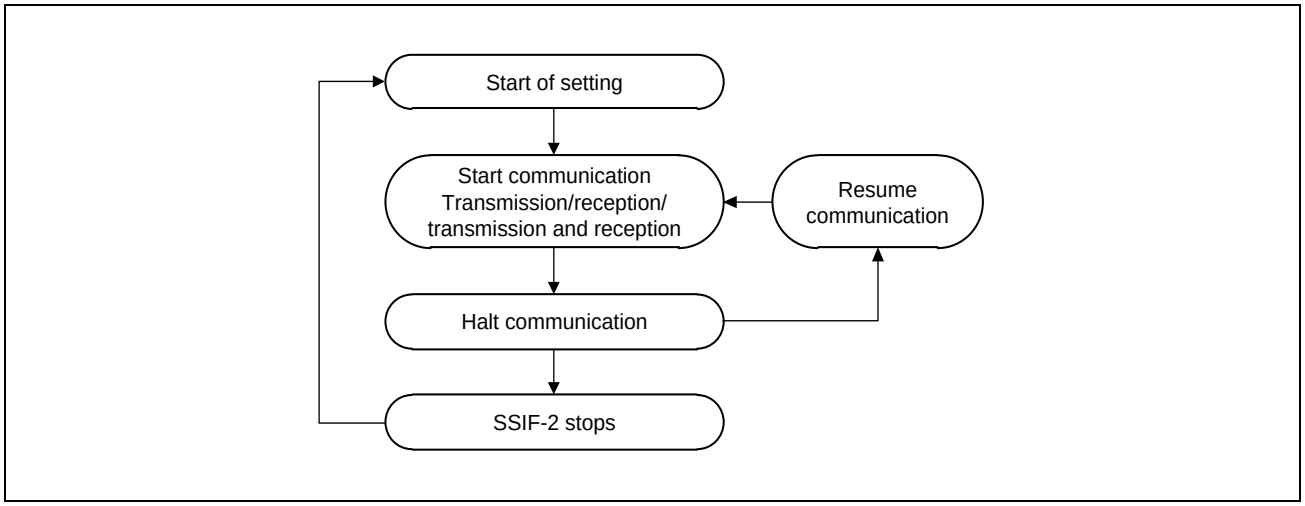
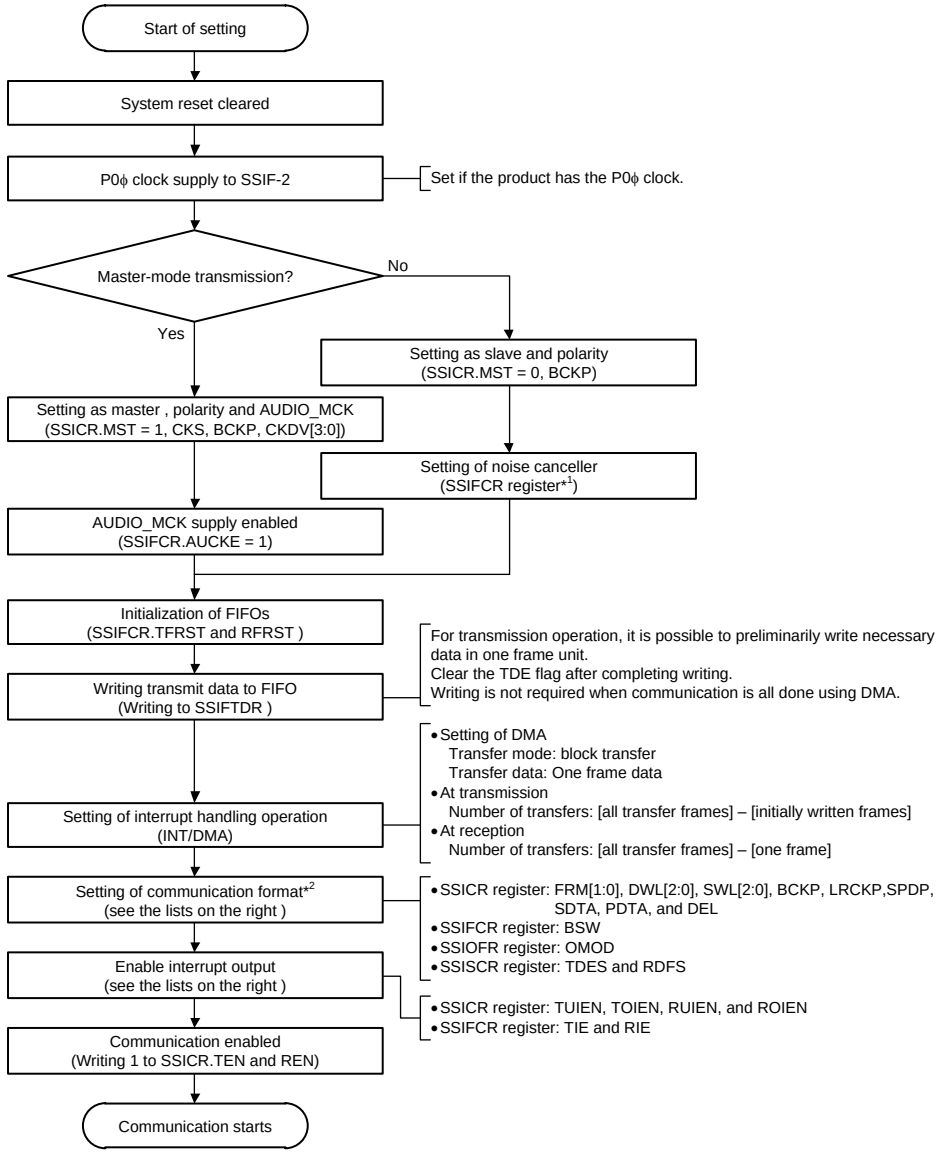



Figure 27.51 SSIF-2 Communication Operation

The procedure of each operation is described in **Section 27.5.2.1, Start Communication** to **Section 27.5.2.7, Resume Communication**.

### 27.5.2.1 Start Communication

This section describes how to start communication of SSIF-2. **Figure 27.52** shows the procedure to start communication. Be sure to follow the procedure. See **Section 27.5.2.2** for transmission operation and **Section 27.5.2.3** for reception operation.




Note 1. BCKNCE, LRCKNCE, LRCKNCE
Change all the settings of the noise cancellation to the same setting.

Note 2. Set SSICR.MUEN, SSIOFR.BCKASTP, and SSIOFR.LRCONT after completing the setting of the communication format.

Figure 27.52 Procedure to Start Communication (CPU Procedure)

For SSIF-2 communication, continuous communication is possible with DMA interrupt handling. For transmission, write 1 to SSIFCR.TIE, SSICR.TUIEN, and SSICR.TOIEN. For reception, write 1 to SSIFCR.RIE, SSICR.RUIEN, and SSICR.ROIEN.

#### 27.5.2.2 Transmission

Follow the transmission operation procedure (**Figure 27.53**) while the transmission is operating.

Transmission starts in synchronization with the start trigger of the SSILRCK/SSIFS signal while transmission is enabled (SSICR.TEN = 1 and SSICR.REN = 0) and while there is one or more frame of serial data in the transmit FIFO data register (SSIFTDR).

SSIF-2 outputs a transmit data empty interrupt to DMA according to the setting condition of TDE (SSISCR.TDES) and transmit data empty interrupt (SSIFCR.TIE) being enabled.

This interrupt requests writing to the transmit FIFO data register (SSIFTDR). By writing to SSIFTDR at each generation of an interrupt, serial data transmission is realized.

In the procedure to start communication, set DMA handling for transmit data empty interrupts as writing to the transmit FIFO data register (SSIFTDR). As a result, SSIF-2 is able not to mediate CPU and to transmit continuous data. When it becomes empty capacity of transmission FIFO data register set with SSISCR.TDES, transmission data empty interrupt is generated. Set the writing number of times according to empty capacity of transmission FIFO data register that transmission data empty interrupt shows. In a case of an error, stop receive procedure (**Section 27.5.2.5**) and then take the error procedure (**Section 27.5.2.6**).

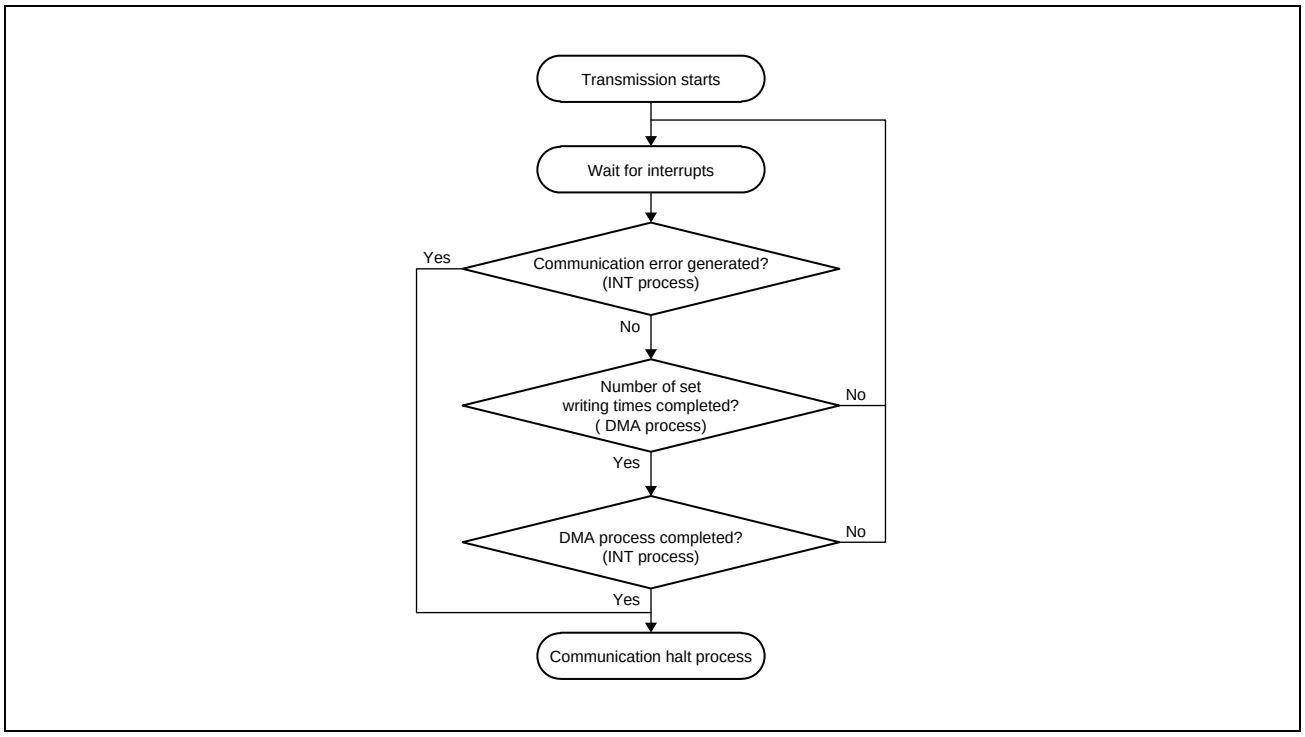



Figure 27.53 Transmission Procedure

NOTE

The communication flow defined in SSIF-2 uses the DMA. In the case the DMA is not used, observe one of SSIFSR.TDE, and write it in SSIFTDR. Follow empty capacity of transmission FIFO data register set with SSISCR.TDES about the number of times written in SSIFTDR detecting 1 of SSIFSR.TDE. After writing the transmission data corresponding to the free space to SSIFTDR, polling ‘1’ of SSIFSR.TDE and repeating this enables continuous transmission operation.

### 27.5.2.3 Reception

Follow the reception operation procedure (**Figure 27.54**) while the reception is operating.

Reception starts in synchronization with the start trigger of the SSILRCK/SSIFS signal while reception is enabled (SSICR.TEN = 0 and SSICR.REN = 1). SSIF-2 outputs a receive data full interrupt to DMA according to the setting of RDF (SSISCR.RDFS) and receive data empty interrupt (SSIFCR.RIE) being enabled. This interrupt requests reading from the receive FIFO data register (SSIFRDR). By reading from SSIFRDR at each generation of an interrupt, serial data reception is realized.

In the procedure to start communication, set DMA handling for receive data full interrupts as reading from receive FIFO data register (DDIFRDR). As a result, SSIF-2 is able not to mediate CPU and to receive continuous data. If the data capacity of reception FIFO data register set with SSISCR.RDFS is stored, receive data full interrupt is generated. Set the writing number of times according to the data capacity of reception FIFO data register that receive data full interrupt shows.

In the case of an error, stop the receive operation (**Section 27.5.2.6**) and then take the error procedure (**Section 27.5.2.6**).

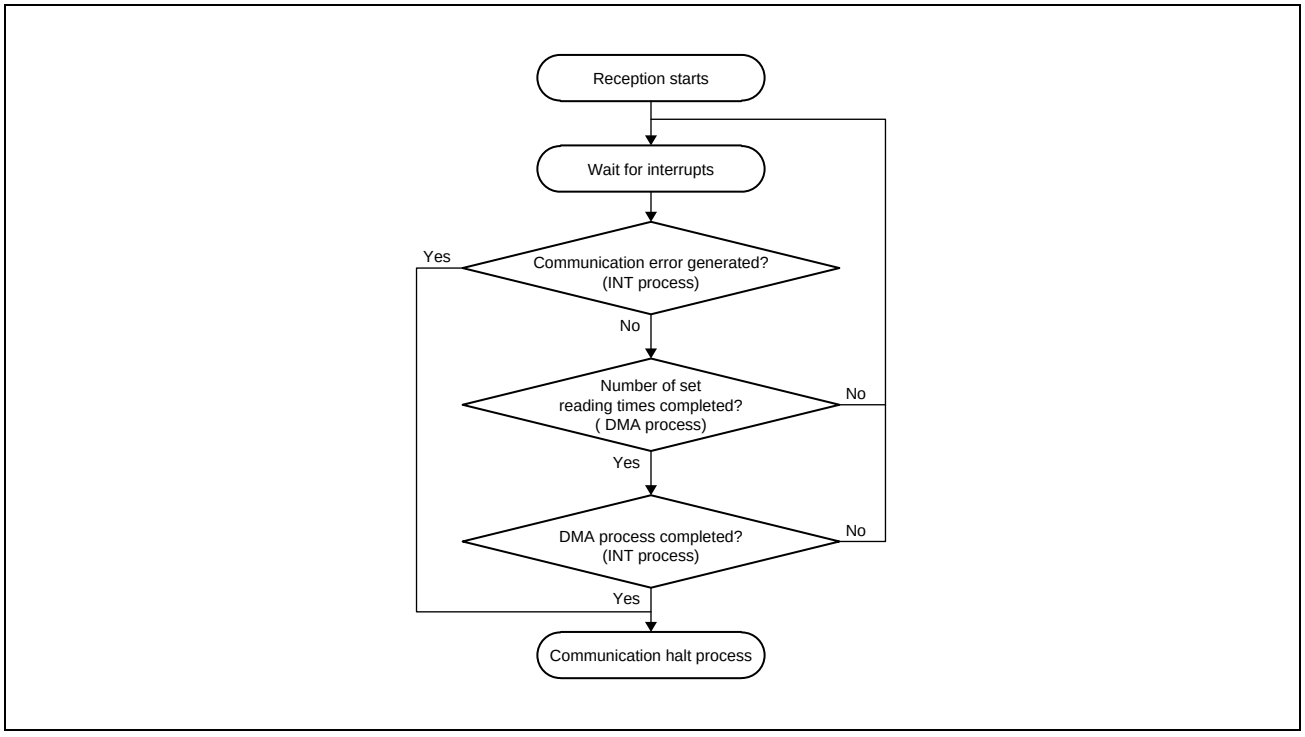



Figure 27.54 Reception Procedure

NOTE

The communication flow defined in SSIF-2 uses the DMA. In the case the DMA is not used, observe 1 of SSIFSR.RDF, and read SSIFRDR. Follow the receive data capacity of reception FIFO data register set with SSISCR.RDFS about the number of times in which SSIFRDR is read detecting 1 of SSIFSR.RDF. After reading the received data from SSIFRDR, Continuous reception operation is possible by polling ‘1’ of SSIFSR.RDF and repeating this.

#### 27.5.2.4 Transmission and Reception

Transmission and reception start in synchronization with the start trigger of the SSILRCK/SSIFS signal while transmission and reception are enabled (SSICR.TEN = 1 and SSICR.REN = 1) and while there is one or more frames of serial data in the transmit FIFO data register (SSIFTDR). The transmission and reception operation of SSIF-2 can be transmitted and received continuously by each doing the procedure for showing in the transmission operation (**Section 27.5.2.2**) and the reception operation (**Section 27.5.2.3**). As transmission and reception are independent in SSIF-2, see sections of transmission and reception respectively. See **Section 27.5.2.5, Halt Communication** to halt communication.

### 27.5.2.5 Halt Communication

This section describes how to halt communication of SSIF-2. **Figure 27.55** shows the procedure to halt communication. Be sure to follow the procedure.

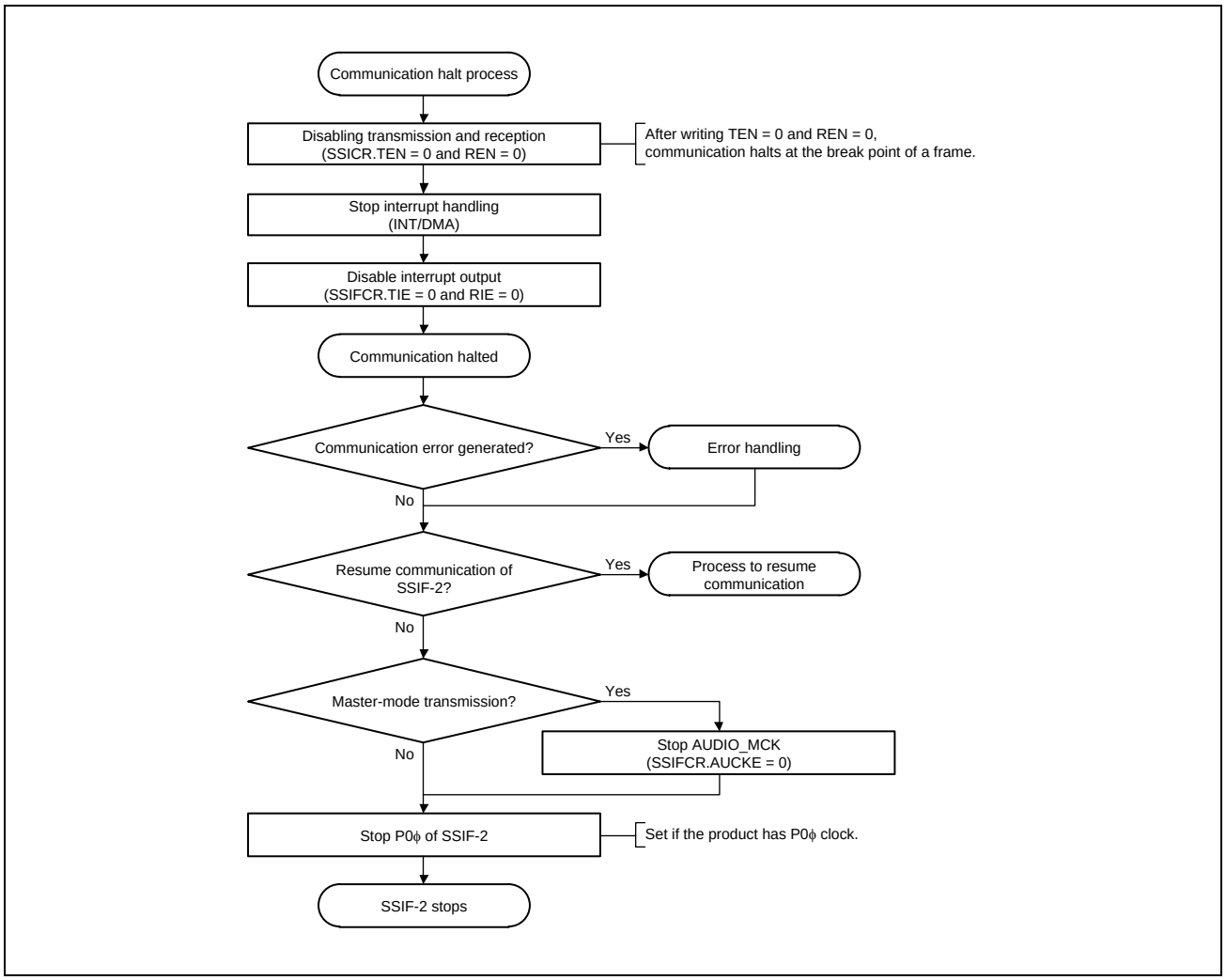



Figure 27.55 Procedure to Halt Communication (CPU Procedure)

To halt the communication of SSIF-2, supply of the following clocks are required until the SSISR.IIRQ bit indicates an idle state.

- Input clock from the SSIBCK pin when SSICR.MST = 0
- AUDIO_MCK when SSICR.MST = 1

To resume communication of SSIF-2 in the previous setting, see **Section 27.5.2.7, Resume Communication**.

NOTE

When communication of SSIF-2 is halted according to the procedure to halt communication in **Figure 27.55**, resume communication according to the procedure to start communication in **Figure 27.52**.

### 27.5.2.6 Error Handling

SSIF-2 has the following four errors.

- Transmit underflow error
- Transmit overflow error
- Receive underflow error
- Receive overflow error

When an underflow error or overflow error is generated, SSIF-2 need to be restarted. Take the error procedure (**Figure 27.56**) according to the procedure to halt communication (**Figure 27.55**).

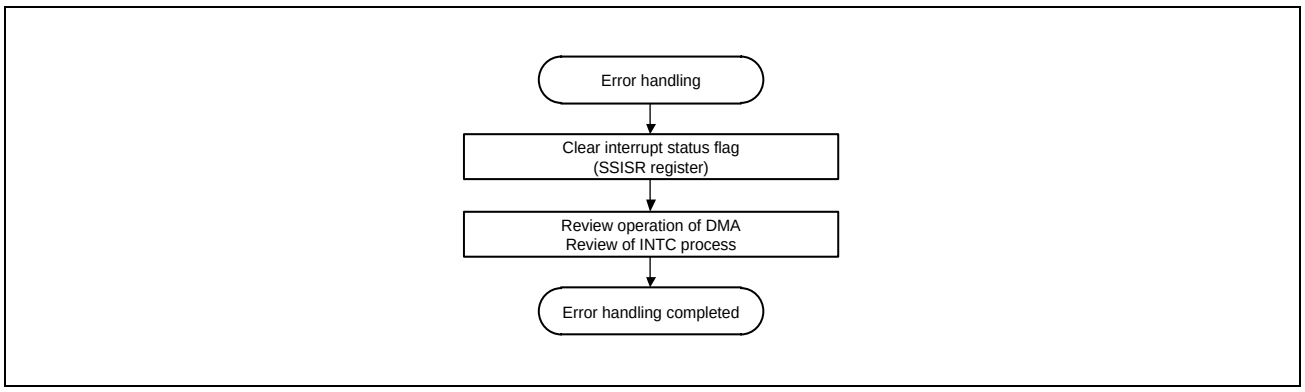



Figure 27.56 Error Handling Procedure

Four error operations are described as follows. When the interrupt output enable bit of the SSICR register is enabled and error flags are set, an error interrupt is generated. See descriptions of flags in **Section 27.4.1.2, Status Register (SSISR)** for the setting conditions of error flags.

#### (1) Transmit Underflow Error

In the case a transmit underflow error is generated during transmission (SSICR.TEN = 1, SSISR.IIRQ = 0), review the number of write operations to the transmit FIFO data register (SSIFTDR) according to serial data empty interrupts. After generation of a transmit underflow error, data to be transmitted from SSIF-2 are 0 output. To transmit data written to SSIFTxD pin normally, run the procedure to halt communication (**Figure 27.55**) and then take the error procedure (**Figure 27.56**). When this error is generated, serial data is consumed as usual. To resume transmission of SSIF-2, write data from the beginning of one frame.

#### (2) Transmit Overflow Error

In the case a transmit overflow error is generated, review the number of write operations to the transmit FIFO data register (SSIFTDR) according to transmit data empty interrupts. Data is not written to SSIFTDR where a transmit overflow error is generated. This error is generated regardless of while the transmission is operating. To resume transmission of SSIF-2, take the procedure to halt communication (**Figure 27.55**) and then take the error procedure (**Figure 27.56**). When communication is resumed, consider the lost serial data.

### (3) Receive Underflow Error

In the case a receive underflow error is generated, review the number of reading operations from the receive FIFO data register (SSIFRDR) according to receive data full interrupts. The value read from SSIFRDR is undefined where a receive underflow error is generated. This error is generated regardless of while the reception is operating. To resume reception of SSIF-2, take the procedure to halt communication (**Figure 27.55**) and then take the error procedure (**Figure 27.56**).

### (4) Receive Overflow Error

In the case a receive overflow error is generated during reception (SSICR.REN = 1, SSISR.IIRQ = 0), review the number of reading operations from the receive FIFO data register (SSIFRDR) according to receive data full interrupts. Data is not stored in SSIFRDR when a receive overflow error is generated. To resume reception of SSIF-2, take the procedure to halt communication (**Figure 27.55**) and then take the error handling procedure (**Figure 27.56**).

### 27.5.2.7 Resume Communication

To resume communication of SSIF-2, follow the procedure shown in **Figure 27.57**. After communication is halted, the same setting is applied when it is resumed. To change the settings of clocks and slave/master mode, follow the procedure to start communication (**Figure 27.52**). For communication after resume, see **Section 27.5.2.2** and **Section 27.5.2.3**.

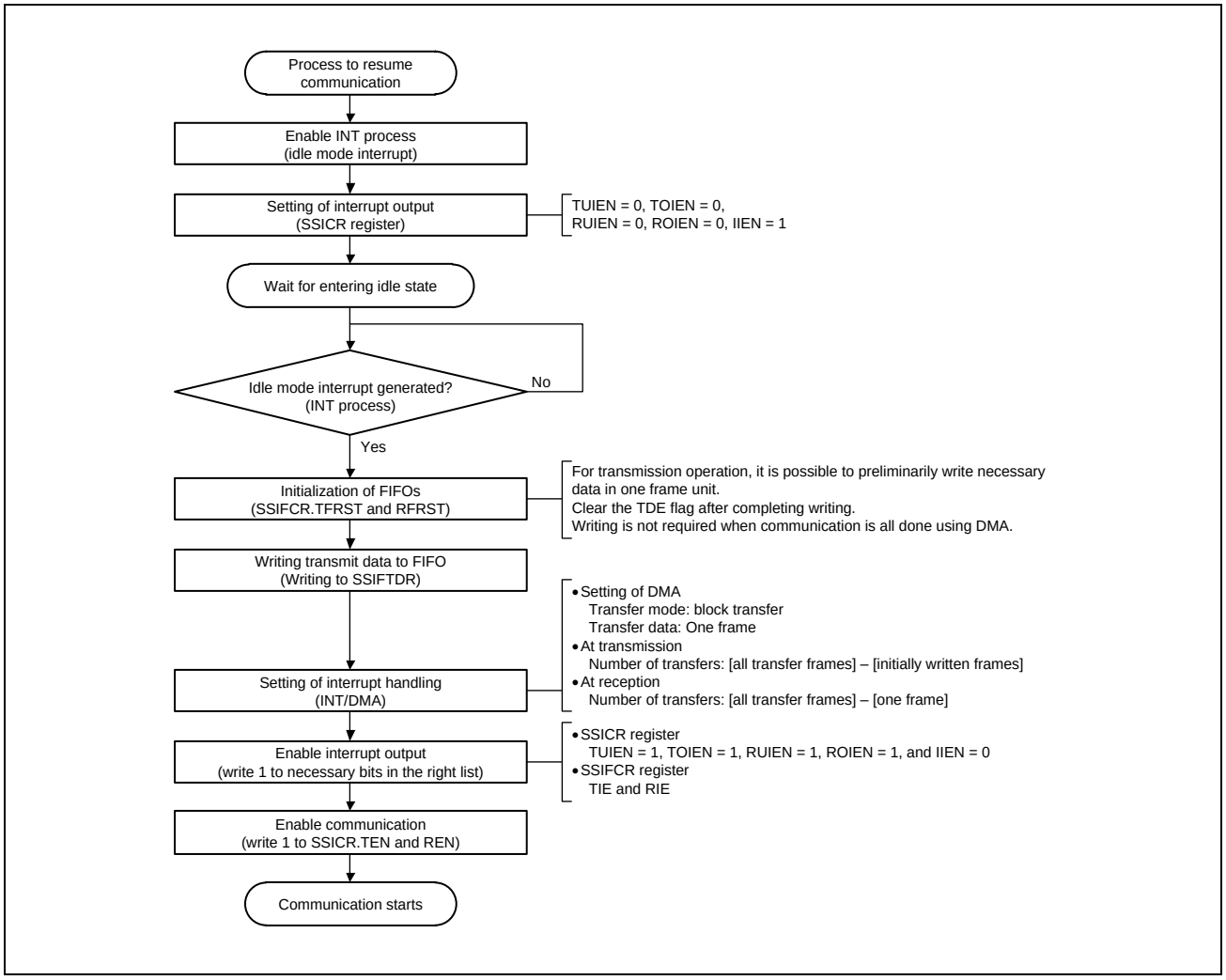



Figure 27.57 Procedure to Resume Communication (CPU Procedure)

### 27.5.3 Interrupt Sources

**Table 27.18** lists the interrupt sources. Set enable/disable of interrupt output of each source with the TUIEN, TOIEN, RUIEN, ROIEN, and IIEN bits in the SSICR register and the TIE and RIE bits in the SSIFCR register.

Table 27.18 Interrupt Sources

| Interrupt Source | Interrupt | Interrupt Flag | Interrupt Sense | DMA Activation |
|---|---|---|---|---|
| INT_ssif_int_req | Transmit underflow interrupt/ Transmit overflow interrupt/ Receive underflow interrupt/ Receive overflow interrupt/ Idle mode interrupt | SSISR.TUIRQ<br>SSISR.TOIRQ<br>SSISR.RUIRQ<br>SSISR.ROIRQ<br>SSISR.IIRQ | Level | Not available |
| INT_ssif_dma_tx | Transmit data empty interrupt | SSIFSR.TDE | Edge | Available |
| INT_ssif_dma_rx | Receive data full interrupt | SSIFSR.RDF | Edge | Available |

#### 27.5.3.1 INT_ssif_int_req Interrupt

This interrupt source combines five interrupts. Enable output of necessary interrupts before using SSIF-2. This interrupt moves from the flag and the output enable permission that five factors have respectively (**Figure 27.58**). To clear an interrupt, set the interrupt enable to 0 or clear the interrupt flag to 0.

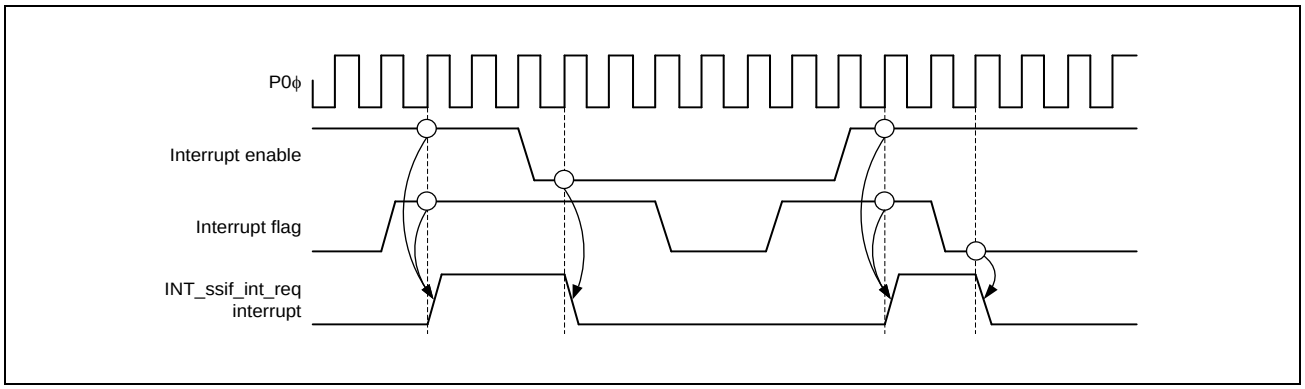



Figure 27.58 Timing Chart of the Common Interrupt Source, INT_ssif_int_req

**(1) Transmit underflow interrupt**

As the transmit underflow interrupt, SSISR.TUIRQ is output while SSICR.TUIEN = 1. To restart transmission (SSICR.TEN = 1), enable the output of this interrupt (SSICR.TUIRQ = 1). In the case this interrupt is generated, take the procedure to halt communication (**Figure 27.55**) and then take the error procedure (**Figure 27.56**).

**(2) Transmit overflow interrupt**

As the transmit overflow interrupt, SSISR.TOIRQ is output while SSICR.TOIRQ = 1. To restart transmission (SSICR.TEN = 1), enable the output of this interrupt (SSICR.TOIRQ = 1). In the case this interrupt is generated, take the procedure to halt communication (**Figure 27.55**) and then take the error procedure (**Figure 27.56**).

### (3) Receive underflow interrupt

As the receive underflow interrupt, SSISR.RUIRQ is output while SSICR.RUIRQ = 1. To restart reception (SSICR.REN = 1), enable the output of this interrupt (SSICR. RUIRQ = 1). In the case this interrupt is generated, take the procedure to halt communication (**Figure 27.55**) and then take the error procedure (**Figure 27.56**).

### (4) Receive overflow interrupt

As the receive overflow interrupt, SSISR.ROIRQ is output while SSICR.ROIRQ = 1. To restart reception (SSICR.REN = 1), enable the output of this interrupt (SSICR. ROIRQ = 1). In the case this interrupt is generated, take the procedure to halt communication (**Figure 27.55**) and then take the error procedure (**Figure 27.56**).

### (5) Idle mode interrupt

As the idle mode interrupt, SSISR.IIRQ is output while SSICR.IIEN = 1. When the thing that the communication has stopped completely is confirmed, this interrupt is used (**Figure 27.57**).

### 27.5.3.2 INT_ssif_dma_tx Interrupt [full-duplex communication]

As the transmit data empty interrupt, a pulse interrupt is output by the following condition.

- SSIFSR.TDE and SSIFCR.TIE becomes 1.
  SSIF-2 operation: SSIFSR.TDE becomes 1 from 0 while SSIFCR.TIE = 1
  CPU operation: SSIFCR.TIE is changed to 1 from 0 while SSIFSR.TDE = 1

An interrupt suppression function is available for this interrupt to prevent the thing that cannot be taken. When the DMA is busy state (Interrupt cannot be accepted), and the interruption condition occurs, output of this interrupt is held. This interrupt is output after the DMA becomes acceptable (**Figure 27.59**).

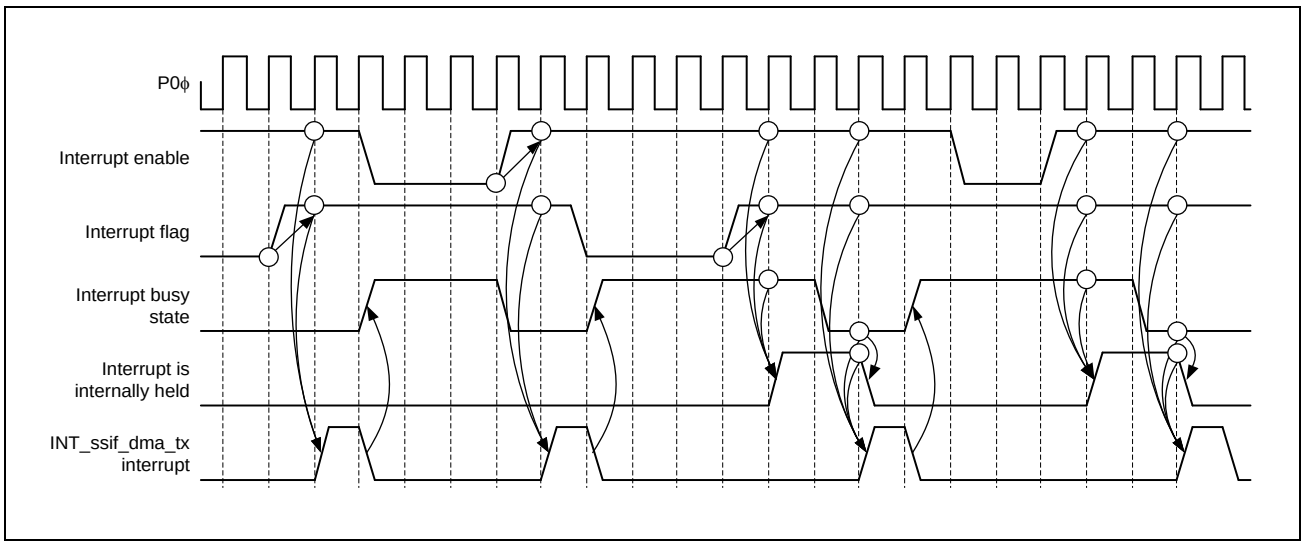



Figure 27.59 INT_ssif_dma_tx Interrupt Timing Chart

### 27.5.3.3 INT_ssif_dma_rx Interrupt [full-duplex communication]

As the receive data full interrupt, a pulse interrupt is output by the following condition.

- SSIFSR.RDF and SSIFCR.RIE becomes 1.
  SSIF-2 operation: SSIFSR.RDF becomes 1 from 0 while SSIFCR.RIE = 1.
  CPU operation: SSIFCR.RIE is changed to 1 from 0 while SSIFSR.RDF = 1.

An interrupt suppression function is available for this interrupt to prevent the thing that cannot be taken. When the DMA is busy state (Interrupt cannot be accepted), and the interruption condition occurs, output of this interrupt is held. This interrupt is output after the DMA becomes acceptable. It operates in the same way as in **Figure 27.59**.

## 27.5.4 Software Resets

SSIF-2 has three software reset bits to reset its states.

- SSIF-2 software reset (SSIFCR.SSIRST)
- Transmit FIFO data register reset (SSICR.TFRST)
- Receive FIFO data register reset (SSIFCR.RFRST)

It explains the procedure of three software reset.

### 27.5.4.1 Software Reset Procedure

#### (1) SSIF-2 Software Reset

For the SSIF-2 software reset bit (SSIFCR.SSIRST), follow the procedure shown in **Figure 27.60**. After a reset, the same setting is applied when it is resumed. To change the settings of clocks and slave/master mode, follow the procedure to start communication (**Figure 27.52**). See **Section 27.5.2.2, Transmission** and **Section 27.5.2.3, Reception** respectively for transmission and reception after communication is resumed.

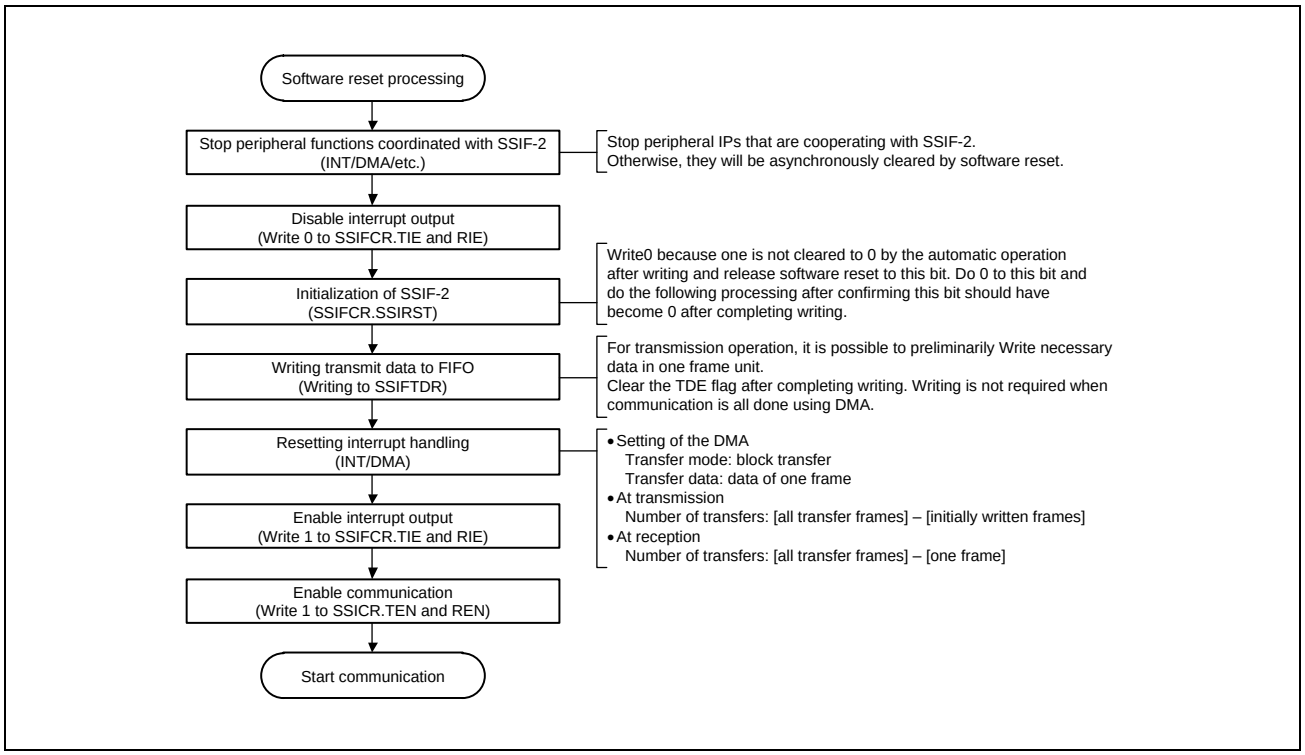



Figure 27.60 Software Reset Procedure (CPU Procedure)

### (2) Transmit FIFO Data Register Reset

To initiate a transmit FIFO data register reset, follow the procedure to start communication (**Figure 27.52**) and resume communication (**Figure 27.57**).

### (3) Receive FIFO Data Register Reset

To initiate a receive FIFO data register reset, follow the procedure to start communication (**Figure 27.52**) and resume communication (**Figure 27.57**).

# 27.6 Notes

## 27.6.1 Notes

### 27.6.1.1 Attention by communication using DMA

SSIF-2 assumes the DMA access as follows.

Table 27.19 Interrupt to expect DMA access

| Communication operation | Interrupt | DMA access | LSI specification | Remarks |
|---|---|---|---|---|
| Transmission | INT_ssif_dma_tx | Write to SSIFTDR | Full duplex transmission | |
| Reception | INT_ssif_dma_rx | Read SSIFRDR | Full duplex transmission | |
| Transmission and Reception | INT_ssif_dma_tx | Write to SSIFTDR | Full duplex transmission | |
| | INT_ssif_dma_rx | Read SSIFRDR | Full duplex transmission | |

### 27.6.1.2 Notes for Slave-mode Communication

#### (1) ADCKE Control

In slave-mode communication (SSICR.MST = 0), SSIF-2 needs supply of SSIBCK. To stop BCK on the master side, make sure that SSIF-2 is in the idle state (SSISR.IIRQ = 1). If BCK is stopped before SSIF-2 becomes idle, take the procedure to start communication (**Figure 27.52**) or wait for an idle state by taking the procedure to resume communication (**Figure 27.57**).

#### (2) SSI_RCK

SSIF-2 has the terminal SSI_RCK that shows the synchronization of the communication. Match communication format of the opposing device and SSIF-2 when SSIF-2 is slave communication (SSICR.MST=0). SSIF-2 uses the signal of SSI_RCK only as a communication start trigger.

### 27.6.1.3 Notes for Master-mode Communication

#### (1) ADCKE Control

In master-mode communication (SSICR.MST = 1), SSIF-2 operates with the audio clock (AUDIO_MCK). To stop SSIF- 2 completely, make sure that SSIF-2 is in the idle state (SSISR.IIRQ = 1) and then write 0 to SSIFCR.ADCKE. If 0 is written to SSIFCR.ADCKE before SSIF-2 becomes idle, take the procedure to start communication (**Figure 27.52**).

#### (2) LRCONT Control

To stop the output to the SSI_RCK pin with SSIOFR.LRCONT when SSIF-2 is in the idle state in master-mode communication (SSICR.MST = 1), note the following: The output stops when 0 is written from 1 to SSIOFR.LRCONT. (**Figure 27.43**). Make sure that the remote device is not affected.

#### (3) BCKASTP Control

To stop the output to the SSIBCK pin with SSIOFR.BCKASTP in master-mode communication (SSICR.MST = 1) and while SSIF-2 is in the idle state, note the following; By writing 0 to SSIOFR.BCKASTP while it is 1, the output stops immediately (**Figure 27.44**). So, make sure that the remote device is not affected.

NOTE

When the opposing device who is the slave needs the clock of the terminal SSI_BCK before the communication operates, it is not possible to use it.

### 27.6.1.4 Notes for Communication Flow

#### (1) When an Error Interrupt is Generated

SSIF-2 has the following four errors.

- Transmit underflow error
- Transmit overflow error
- Receive underflow error
- Receive overflow error

When an underflow error or overflow error is generated, SSIF-2 need to be restarted. Take the error procedure (**Figure 27.56**) according to the procedure to halt communication (**Figure 27.55**).

##### (a) Transmit Underflow Error

In the case a transmit underflow error is generated during transmission (SSICR.TEN = 1, SSISR.IIRQ = 0), review the number of write operations to the transmit FIFO data register (SSIFTDR) according to serial data empty interrupts. After generation of a transmit underflow error, data to be transmitted from SSIF-2 are 0 output. To transmit data written to SSIFTxD pin normally, run the procedure to halt communication (**Figure 27.55**) and then take the error procedure (**Figure 27.56**). When this error is generated, serial data is consumed as usual. To resume transmission of SSIF-2, write data from the beginning of one frame.

##### (b) Transmit Overflow Error

In the case a transmit overflow error is generated, review the number of write operations to the transmit FIFO data register (SSIFTDR) according to transmit data empty interrupts. Data is not written to SSIFTDR where a transmit

overflow error is generated. This error is generated regardless of while the transmission is operating. To resume transmission of SSIF-2, take the procedure to halt communication (**Figure 27.55**) and then take the error procedure (**Figure 27.56**). When communication is resumed, consider the lost serial data.

#### (c) Receive Underflow Error

In the case a receive underflow error is generated, review the number of reading operations from the receive FIFO data register (SSIFRDR) according to receive data full interrupts. The value read from SSIFRDR is undefined where a receive underflow error is generated. This error is generated regardless of while the reception is operating. To resume reception of SSIF-2, take the procedure to halt communication (**Figure 27.55**) and then take the error procedure (**Figure 27.56**).

#### (d) Receive Overflow Error

In the case a receive overflow error is generated during reception (SSICR.REN = 1, SSISR.IIRQ = 0), review the number of reading operations from the receive FIFO data register (SSIFRDR) according to receive data full interrupts. Data is not stored in SSIFRDR when a receive overflow error is generated. To resume reception of SSIF-2, take the procedure to halt communication (**Figure 27.55**) and then take the error handling procedure (**Figure 27.56**).

### (2) Transmit Data Empty Interrupt

The communication flow defined in SSIF-2 uses the DMA. In the case the DMA is not used, observe one of SSIFSR.TDE, and write it in SSIFTDR. Follow empty capacity of transmission FIFO data register set with SSISCR.TDES about the number of times written in SSIFTDR detecting 1 of SSIFSR.TDE. Clear the transmission data that corresponds to empty capacity and clear the SSIFSR.TDE flag to SSIFTDR after writing. Transmission operation consecutive by the repeated thing is possible. SSIFSR.TDE is not cleared when not clearing.

### (3) Receive Data Full Interrupt

The communication flow defined in SSIF-2 uses the DMA. In the case the DMA is not used, observe 1 of SSIFSR.RDF, and read SSIFRDR. Follow the receive data capacity of reception FIFO data register set with SSISCR.RDFS about the number of times in which SSIFRDR is read detecting 1 of SSIFSR.RDF. Clear data that does the reception completion and clear the SSIFSR.RDF flag from SSIFRDR after reading. Reception operation consecutive by the repeated thing is possible. SSIFSR.RDF is not cleared when not clearing.

### (4) Switching Transfer Modes

1. For state transition from transmission, reception, and transmission and reception, disable transmission and reception (SSICR.TEN = 0, SSICR.REN = 0).
2. Confirm it is in the idle state (SSISR.IIRQ = 1).
3. In the idle state, set the SSICR.TEN bit and the SSICR.REN bit again and resume transfer.

### (5) Resume Communication after Halting SSIF-2

When communication of SSIF-2 is halted according to the procedure to halt communication in **Figure 27.55**, resume communication according to the procedure to resume communication in **Figure 27.52**.

#### 27.6.1.5 Write Access Restriction

**(1) SSICR Register**

If the TEN bit or REN bit is rewritten, make sure that the SSISR.IIRQ bit is in the desired status. If written, the operation performed immediately after writing is not guaranteed. For example, after enabling operation, make sure that SSISR.IIRQ = 0 and after disabling operation, make sure that SSISR.IIRQ = 1.

**(a) TEN Bit and REN Bit**

Writing 1 to these bits, transmission starts in synchronization with start trigger. Refer to **Section 27.5.2.2, Transmission**, **Section 27.5.2.3, Reception**, and **Section 27.5.2.4, Transmission and Reception** for details. Writing 0 to these bits, transmission stops in synchronization with next frame boundary. Set 1 at the same time when you use SSIF-2 as transmission and reception. To stop communication, make sure to stop transmission and reception (TEN = 0 and REN = 0).

**(2) SSICR Register**

**(a) Clear operation of TUIRQ and TOIRQ**

After the communication is permitted (SSICR.TEN is written and ‘1’ is written from ‘0’), the transmission error flag (TOIRQ and TUIRQ of the SSISR register) is cleared. However, when the SSISR register is continuously read, the clearness of the transmission error flag might not be able to be read.

**(b) Clear operation of RUIRQ and ROIRQ**

After the communication is permitted (SSICR.REN is written and ‘1’ is written from ‘0’), the transmission error flag (ROIRQ and RUIRQ of the SSISR register) is cleared. However, when the SSISR register is continuously read, the clearness of the reception error flag might not be able to be read.

### (3) Communication State

Writing to the bits with shaded area in **Table 27.20** is prohibited. If written, the operation performed immediately after writing is not guaranteed.

Table 27.20 Bits Protected from Writing during Communication

| Symbol | Address (Base+) | | +0 | | | | | | | | +1 | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
| | | | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| SSICR | H'00 | +0 | — | CKS | TUIEN | TOIEN | RUIEN | ROIEN | IIEN | — | FRM[1:0] | | DWL[2:0] | | | SWL[2:0] | | |
| | | +2 | — | MST | BCKP | LRCKP | SPDP | SDTA | PDTA | DEL | CKDV[3:0] | | | | MUEN | — | TEN | REN |
| SSISR | H'04 | +0 | — | — | TUIRQ | TOIRQ | RUIRQ | ROIRQ | IIRQ | — | — | — | — | — | — | — | — | — |
| | | +2 | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| SSIFCR | H'10 | +0 | AUCKE | — | — | — | — | — | — | — | — | — | — | — | — | — | — | SSIRST |
| | | +2 | — | — | — | — | BSW | BCKNCE | LRCKNCE | RxDNCE | — | — | — | — | TIE | RIE | TFRST | RFRST |
| SSIFSR | H'14 | +0 | — | — | TDC[5:0] | | | | | | — | — | — | — | — | — | — | TDE |
| | | +2 | — | — | RDC[5:0] | | | | | | — | — | — | — | — | — | — | RDF |
| SSIFTDR | H'18 | +0 | FTDR[31:16] | | | | | | | | | | | | | | | |
| | | +2 | FTDR[15:0] | | | | | | | | | | | | | | | |
| SSIFRDR | H'1C | +0 | FRDR[31:16] | | | | | | | | | | | | | | | |
| | | +2 | FRDR[15:0] | | | | | | | | | | | | | | | |
| SSIOFR | | +0 | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| | | +2 | — | — | — | — | — | — | BCKASTP | LRCONT | — | — | — | — | — | — | OMOD[1:0] | |
| SSISCR | H'24 | +0 | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| | | +2 | — | — | TDES[5:0] | | | | | | — | — | RDFS[5:0] | | | | | |

# 32. USB2.0

This LSI includes 1 channel USB2.0 Host/Function interface.

This section describes the overview of USB2.0 and USB2.0 PHY control.

The detail function of USB2.0 Host controller and the common function both Host module and Function module are described in **Section 32A, USB 2.0 Host Module**.

The detail function of USB2.0 Function controller is described in **Section 32B, USB 2.0 Function Module**.

## 32.1 Features

This interface complies the following specifications.

- Universal Serial Bus Specification Revision 2.0
- Enhanced Host Controller Interface (EHCI) Specification for Universal Serial Bus Revision 1.0
- EHCI v1.1 Addendum*[1]
- Open Host Controller Interface (OHCI) Specification for USB Release 1.0a
- Battery Charging Specification Revision 1.2*[2]

**Note 1.** Some EHCI v1.1 feature are not supported. (Refer to **Section 32A, USB 2.0 Host Module** for detail)

**Note 2.** DCP (Dedicated Charging Port) is not supported as downstream port.

### 32.1.1 Ch0: USB2.0 (Host/Function) interface

The following table shows features of this interface.

Table 32.1 Ch0: USB2.0 (Host/Function) interface features

| Function | Description |
|---|---|
| Host mode | • Support mode: High-Speed(480Mbps)/Full-Speed(12Mbps)/Low-Speed(1.5Mbps)<br>• Support Isochronous/Interrupt/Control/Bulk transfer modes<br>• Support Isochroous/Interrupt high bandwidth transfer |
| Function mode | • Support mode: High-Speed(480Mbps)/Full-Speed(12Mbps)<br>• Isochronous/Interrupt/Control/Bulk transfer<br>• Up to 10 ch PIPE (includes default control PIPE) |
| Other functions | • Battery Charging function |

### 32.1.2 Block Diagram

The block diagram of USB interface is as follows.

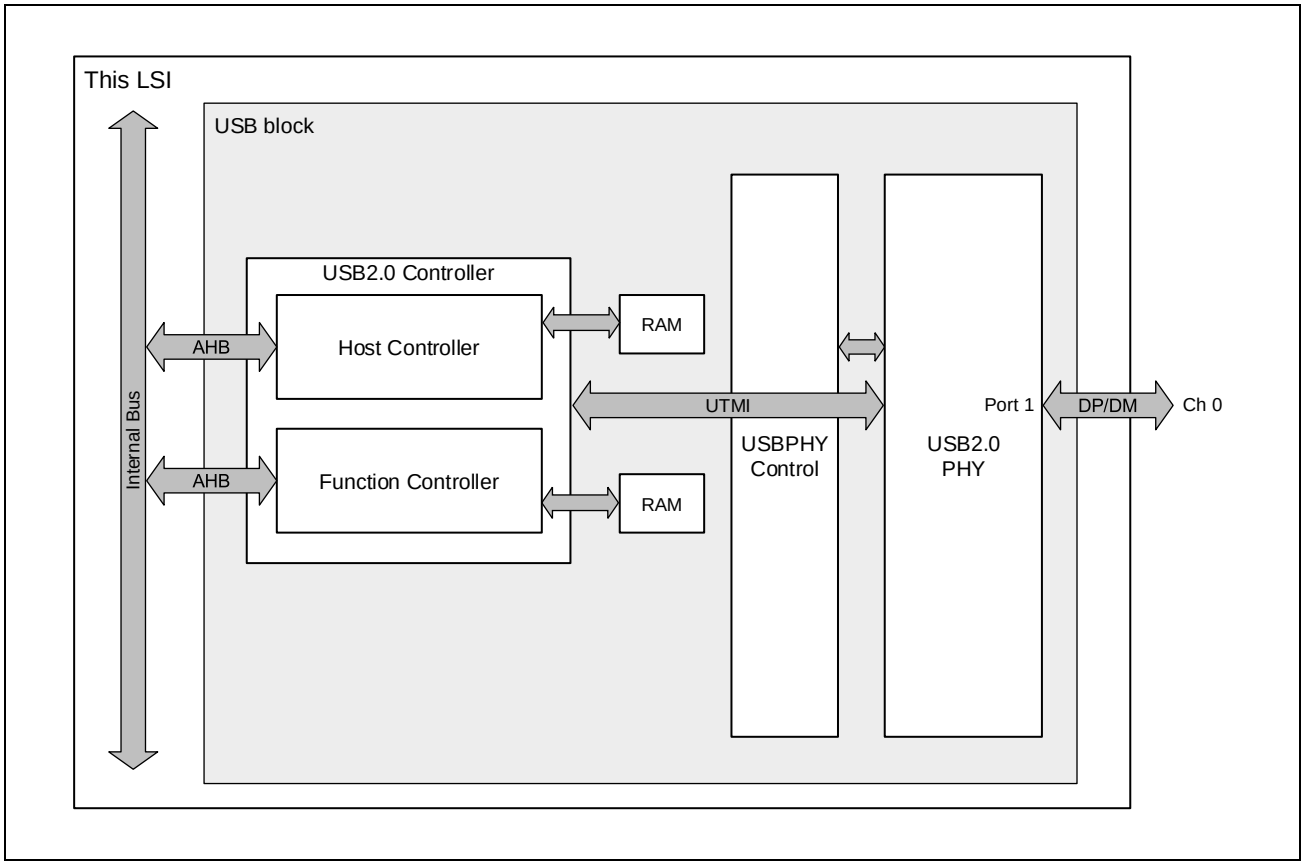



Figure 32.1 USB interface block diagram

### 32.1.3 External signal pins

The following table shows external signal pins for USB block.

Table 32.2 External signal pins

| Name | I/O | Description | Active Level | Initial value | Handling in not used |
|---|---|---|---|---|---|
| USB0_DP | — | USB data pin Data+ (Ch0) | — | — | Open*1 |
| USB0_DM | — | USB data pin Data- (Ch0) | — | — | Open*1 |
| USB0_OVRCUR | I | Over current detection input (Ch0) | L | — | Pull-Up |
| USB0_VBUSEN | O | VBUS output enable (Ch0) | H | 0b | Open |
| USB0_VBUSIN | I | Peripheral VBUS input (Ch0) | H | — | Pull-Down |

Note 1. When it is not used permanently, connect to GND via a resistor of 10kΩ.

### 32.1.4 Power and GND pins

The following table shows Power and GND pins for USB block.

Table 32.3 Power and GND pins

| Name | Description |
|---|---|
| USB_VDD18 | Internal Regulator Power (1.8V), HS receiver power (1.8V) |
| USB_VDD33 | IO power (3.3V) |
| USB_RREF | Reference voltage |
| VSS | GND |

### 32.1.5 Interrupt

The following table shows interrupt list for USB block.

Table 32.4 USB interrupt

| Name | Description | Active Level | Type |
|---|---|---|---|
| U2H0_INT | USB2.0 Host AHB Interrupt (Ch0) | H | Level-Sensitive |
| U2H0_OHCI_INT | USB2.0 Host OHCI Interrupt (Ch0) | H | Level-Sensitive |
| U2H0_EHCI_INT | USB2.0 Host EHCI Interrupt (Ch0) | H | Level-Sensitive |
| U2H0_WAKEON_INT | USB2.0 Host EHCI Wakeup Interrupt (Ch0) | H | Level-Sensitive |
| U2H0_OBINT | USB2.0 Host Battery Charging Interrupt (Ch0) | H | Level-Sensitive |
| U2P_IXL_INT | USB2.0 Function controller Interrupt (Ch0) | H | Edge-Triggered |
| U2P_INT_DMA[1:0] | USB2.0 Function controller DMA transaction complete interrupt (Ch0) | H | Level-Sensitive |
| U2P_INT_DMAERR | USB2.0 Function controller DMA error response interrupt (Ch0) | H | Level-Sensitive |

## 32.2 Register Configuration

### 32.2.1 Host controller Register

Refer to the **Section 32A, USB 2.0 Host Module**.

### 32.2.2 Function controller Register

Refer to the **Section 32B, USB 2.0 Function Module**.

### 32.2.3 USBPHY Control Register

USBPHY Control Register mainly controls reset and power down of the USB/PHY.

Base Address: H'0_11C4_0000 (Cortex-A55 Address Space)

The following table shows USBPHY Control Register list.

Prohibit to write undefined area.

Table 32.5 USBPHY Control Register List

| Register Name | Abbreviation | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|
| Reset register | RESET | R/W | H'0000_0133 | H'000 | 32 |
| Clock control register | UCLKCTL | R/W | H'0340_0303 | H'018 | 32 |
| Direct power down setting register | UDIRPD | R/W | H'0000_0000 | H'01C | 32 |
| VBUSEN control register | VBENCTL | R/W | H'0000_0000 | H'03C | 32 |
| Connection control register | CON_CTRL | R/W | H'0000_0000 | H'020 | 32 |
| Clock status register | CLK_STAT | R | H'0000_0000 | H'104 | 32 |

# 32.3 Register Descriptions

## 32.3.1 Reset register (RESET)

This register controls USB/PHY reset.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | sel_pll reset | — | — | — | pllreset | — | — | — | sel_p1 reset | — | — | — | phyrst_1 |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |
| R/W | R | R | R | R/W | R | R | R | R/W | R | R | R | R/W | R | R | R | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 13 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 12 | sel_pllreset | 0 | R/W | Select USB/PHY PLL reset<br>1: register (by bit8: pllreset)<br>0: control by host controller (Ch0) |
| 11 to 9 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 8 | pllreset | 1 | R/W | PLL reset<br>1: PLL reset<br>0: deassert PLL reset |
| 7, 6 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 5 | — | 1 | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 4 | sel_p1reset | 1 | R/W | Select USB/PHY Port1 reset<br>1: register(phyrst_1)<br>0: Control by host controller (Ch0) |
| 3, 2 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 1 | — | 1 | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 0 | phyrst_1 | 1 | R/W | USB/PHY reset (Port1)<br>1: USB/PHY reset<br>0: deassert USB/PHY reset<br>Only when sel_p1reset=1, this bit is valid. |

### 32.3.2 Clock control register (UCLKCTL)

This register controls USB/PHY clock.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | sel_sleepm | — | — | — | sleepm_1 | — | — | — | — | — | — | — | — |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R/W | R | R | R | R/W | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | sel_suspendm | — | — | — | suspendm_1 | — | — | — | — | — | — | — | — |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| R/W | R | R | R | R/W | R | R | R | R/W | R | R | R | R | R | R | R | R |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 29 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 28 | sel_sleepm | 0 | R/W | Select SLEEPM control<br>1: register (controlled by bit 24: sleepm_1)<br>0: Control by controller |
| 27, 26 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 25 | — | 1 | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 24 | sleepm_1 | 1 | R/W | Sleep control (Port1)<br>1: Normal mode<br>0: Sleep mode |
| 23 | — | 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 22 | — | 1 | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 21, 20 | — | All 0 | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 19, 18 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 17, 16 | — | All 0 | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 15, 14 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 13 | — | 0 | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 12 | sel_suspendm | 0 | R/W | Select SUSPENDM control<br>1: register (controlled by bit 8: suspendm_1)<br>0: control by controller |
| 11, 10 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 9 | — | 1 | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 8 | suspend_1 | 1 | R/W | Suspend control (Port1)<br>0: Suspend mode<br>1: Normal mode |
| 7 to 5 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 4 | — | 0 | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 3, 2 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 1, 0 | — | All 1 | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |

### 32.3.3 Direct power down setting register (UDIRPD)

This register controls USB/PHY direct power down.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | sel_udirpd | — | — | — | dirpd |
| 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 | R | R | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 5 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 4 | sel_udirpd | 0 | R/W | Select dirpd control<br>1: register (dirpd)<br>0: Control by host controller (Ch0) |
| 3 to 1 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 0 | dirpd | 0 | R/W | Power down control to reduce the power consumption of the USB/PHY when not using the USB function.<br>0: Normal operation<br>1: Power save |

### 32.3.4 VBUSEN control register (VBENCTL)

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | vbus_sel_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/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 2 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 1 | — | 0 | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 0 | vbus_sel_0 | 0 | R/W | Select VBUSEN control<br>1: VBOUT bit of the VBUS Control register.<br>0: Port Power bit of the EHCI/OHCI Operational register. |

### 32.3.5 Connection control register (CON_CTRL)

This register controls USB/PHY connection

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | sel_connect | — | — | — | connect_1 |
| 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 | R | R | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 5 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 4 | sel_connect | 0 | R/W | Select USB connection notification control<br>1: register (controlled by bit0: connect_1)<br>0: Control by host controller |
| 3, 2 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 1 | — | 0 | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 0 | connect_1 | 0 | R/W | Notice of USB connection (port1)<br>Used to notify UTM + core of USB connection status<br>0: USB not connected (single end receiver disabled)<br>1: USB not connected (single end receiver enabled) |

### 32.3.6 Clock status register (CLK_STAT)

This register shows USB/PHY clock status.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | frclk60 lock | frclk48 lock | phy lock | — | — | — | plllock_ 1 |
| 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 7 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 6 | frclk60_lock | 0 | R | FRCLK60 LOCK status<br>0: UNLOCK<br>1: LOCK |
| 5 | frclk48_lock | 0 | R | FRCLK48 LOCK status<br>0: UNLOCK<br>1: LOCK |
| 4 | phylock | 0 | R | PHY PLL LOCK Status (Transceiver)<br>0: UNLOCK<br>1: LOCK |
| 3, 2 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 1 | — | 0 | R | Reserved<br>When read, the initial value is read. The written value will be ignored. |
| 0 | plllock_1 | 0 | R | PLL LOCK status (port1)<br>0: UNLOCK<br>1: LOCK |

# 32.4 Operation

## 32.4.1 USB/PHY Initialization

In the initial state, the reset of USB/PHY Port1 is controlled by the register in the USBPHY Control Register, and the reset is asserted, so the control is switched to controller by the following initialization.

Also, in the initial state, the USB/PHY PLL Reset is controlled by the Host Controller, but it needs to be switched to controlled by the USBPHY Control Register.

Write H’0000_1133 to the reset register (RESET) of USBPHY Control Register, and then write H’0000_1022 to release the reset.

(By the above register settings, USBPHY Control Register controls the PLL Reset and the reset is released.

Port1 of USB/PHY is controlled by Host Controller.)

The common setting sequence both Host mode and Function mode including USB/PHY initialization is described in **Section 32A.9.1, Host/Peripheral Common Setting Sequence**.

### 32.4.2 Handling of permanently unused pin

Take the following measures when USB interface is not used permanently.

#### 32.4.2.1 USB/PHY related pins

- When Port1 is unused

Take the following measures for the USB pins and power supply.

- Connect the USB0_DP/DM to GND via a resistor of 10kΩ.
- Set the RREF to Open.
- Connect USB_VDD18 to a 1.8V power supply. However, there is no need to separate the digital and analog power supplies.
- Connect USB_VDD33 to GND.
- Connect VSS to GND.

Also, perform the following control with software.

- Set the direct power down (via UDIRPD register). See 34.3.3 Direct power down for more information.

#### 32.4.2.2 Handling of USB controller related pins

- When Host controller is unused

The USB0_VBUSEN pin leaves open and fix the USB0_OVRCUR pin to 1 when USB/PHY port1 is unused.

- Function controller is unused

Fix the USB0_VBUSIN pin to 0 when USB/PHY port1 is unused.

### 32.4.3 Direct power down (DIRPD)

By directly controlling the power down signal of the USB/PHY with the USBPHY Control register, the USB/PHY can be put into a low power state regardless of the state of the controller.

When set to the direct power down state, all USB/PHY functions including the PLL will be stopped. Direct power down can only be used when the USB/PHY is unused (means both ports are unused).

Follow the procedure below to switch to / return to the direct power down mode.

For details on the USBPHY Control Register reset register and direct power down setting register used in the procedure below, refer to **Section 32.3.1, Reset register (RESET)** and **Section 32.3.3, Direct power down setting register (UDIRPD)**, respectively.

#### 32.4.3.1 Enter to Direct power down mode

Follow the steps below to switch to direct power down mode.

(1) USB operation stop

Stop the operation of USB.

(2) Assert the direct power down signal.

It is asserted by setting H’0000_0010 in the direct power-down setting register (UDIRPD) and then setting H’0000_0011.

There are no timing restrictions when transitioning to direct power down. Immediately after setting, it shifts to power down mode.

#### 32.4.3.2 Resume from Direct power down mode

Follow the steps below to recover from the direct power down mode.

(1) Assert a USB/PHY PLL reset.

The PLL reset is asserted by setting 1 to sel_pllreset bit of the Reset register (RESET) and then setting 1 to pllreset bit.

The reset period should be 1us or more.

(2) De-assert direct power down signal

Negate the direct power down signal after continuing to assert the PLL reset for more than 1us. It is negated by setting H'0000_0010 in the direct power down setting register (UDIRPD) and then setting H'0000_0000.

After negating the direct power down signal, keep asserting the PLL reset for at least 1us.

(3) De-assert USB/PHY PLL reset

After negating the direct power down signal, continue to assert the PLL reset for 1us or more, and then negate the PLL reset.

The PLL reset is negated by clearing the pllreset bit in the reset register (RESET) and then clearing the sel_pllreset.

The timing chart when returning from the direct putter down is shown below.

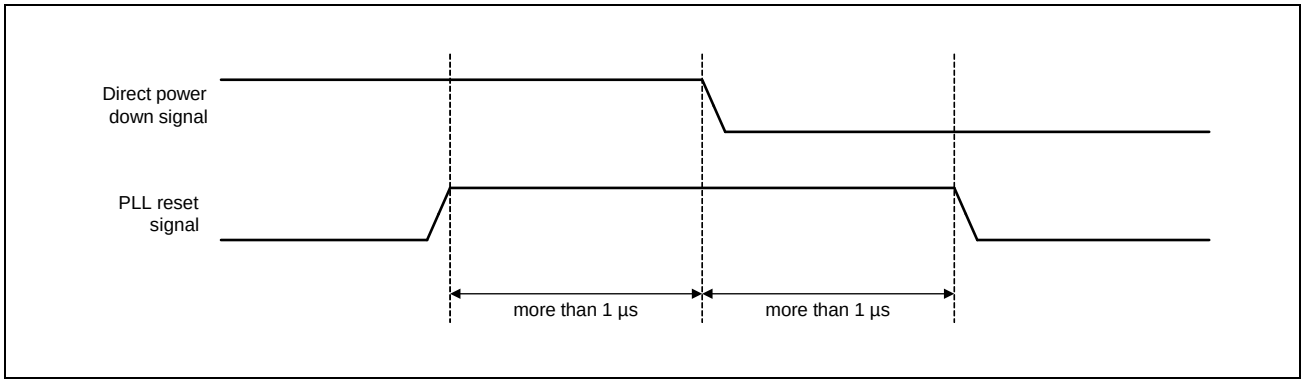



Figure 32.2 Resume from Direct power down mode

## 32.5 Usage Notes

### 32.5.1 Precautions in power down mode

When USB interface is unused, it can be set to power-down mode by register setting. For the setting, refer to **Section 32.3.3, Direct power down setting register (UDIRPD)**.

The precautions for power down mode are as follows.

- Once enter to power down, do not access the interface.
- Move to the power-down state after the USB is in unused state. If USB interface moves power-down state while it is operating, current may continue to flow to the USB side and it may damage device.

# 32A. USB 2.0 Host Module

NOTE

This LSI does not support the OTG function. Related description of the OTG Function in this section should be ignored.

## 32A.1 Overview

### 32A.1.1 Overview

This LSI has one USB 2.0 host/function module. For each module, you can switch between the host mode and the peripheral mode by specifying the UCOM register setting. This section describes the circuits that are common to both modes, and the host controller itself.

### 32A.1.2 Features

This module has the following features:

| Function | Description |
|---|---|
| Host function | • Supporting high-speed (480 Mbps), full-speed (12 Mbps), and low-speed (1.5 Mbps) transfers<br>• Compliant with Open Host Controller Interface (OHCI) Specification for USB Revision 1.0a<br>• Compliant with Enhanced Host Controller Interface (EHCI) Specification for USB Revision 1.1*1 |
| Other functions | • Battery charging (compliant with Battery Charging Specification Revision 1.2)*2<br>• Dual-role-device function (static switching between the USB host and USB peripheral functions) |

Note 1. Some functions (specifications) are not supported.

Note 2. The setting for battery charging is handled by the host controller even if the function controller is selected. This LSI does not support DCP (Dedicated Charging Port) as a Downstream Port.

### 32A.1.2.1 EHCI v1.1 functions

Conventional host controller modules comply with EHCI Specification Revision.1.0. Meanwhile, this module supports the additional functions listed below to comply with EHCI v1.1 Addendum.

Note, however, that this module supports only the functions marked with a circle "✓" in the table below.

| Function Name | Support |
|---|---|
| Per-Port Change Events | ✓ |
| Shorter Periodic Frame List | ✓ |
| Hardware Prefetching | — |
| Link Power Management (LPM) | ✓ |

Of the three features to support, the registers associated with "Per-Port Change Events" and "Shorter Periodic Frame List" are listed below. For more information on each register, refer to the appropriate register specification of the **Section 32A.2, Register Descriptions**.

With regard to "Link Power Management (LPM)", it is described in **Section 32A.1.2.2, Link Power Management (LPM) function**.

Table 32A.1 Per-Port Change Events

| Relevant register | Relevant bit | | Attribute | Function |
|---|---|---|---|---|
| HCCPARAMS (offset: H'108) | bit 18 | Per-Port Change Event Capability | R | This bit indicates to support this function.<br>This bit is fixed to 1b. |
| USBCMD (offset: H'120) | bit 15 | Per-Port Change Events Enable | R/W | Per-Port event notification setting.<br>Writing 1b to this bit enables the Per-Port event notification function. |
| USBSTS (offset: H'124) | bit 16 | Port-1 Change Detect | R/W(1) | If a port-1 change event is detected, 1b is written to this bit. |
| USBINTR (offset: H'128) | bit 16 | Port-1 Change Event Enable | R/W | Enable/Disable setting of the above "port-1 Change Detect" field.<br>To reflect the port event on the above field for the port, set corresponding bit to 1b. |

Table 32A.2 Shorter Periodic Frame List

| Relevant register | Relevant bit | | Attribute | Function |
|---|---|---|---|---|
| HCCPARAMS (offset: H'108) | bit 19 | 32-Frame Periodic List Capability | R | This bit indicates to support this function.<br>This bit is fixed to 1b. |
| USBCMD (offset: H'120) | bit[3:2] | Frame List Size | R/W | This bit determines the Frame List Size.<br>To determine 32-frames, set this field 11b. |

### 32A.1.2.2 Link Power Management (LPM) function

This module supports the power management function conforming to “USB 2.0 Link Power Management Addendum to the Universal Serial Bus v2.0 Specification” (abbreviated as LPM).

When the peripheral device which supports LPM function is connected to this module, the device can be moved to suspend state faster than conventional by using LPM function.

*In using LPM function: 8 to 10 µs

The relevant registers are listed in the following. Refer to the appropriate register specification of each register detail in **Section 32A.2, Register Descriptions**.

| Relevant register | Relevant bit | | Attribute | Function |
|---|---|---|---|---|
| HCCPARAMS (offset: H’108) | bit 17 | Link Power Management Capability | R | This bit indicates to support this function.<br>This bit is fixed to 1b. |
| USBCMD (offset: H’120) | bit[27:24] | Host-Initiated Resume Duration | R/W | The minimum duration time of K-state drive in resume from LPM state. |
| PORTSC1 (offset: H’164) | bit[31:25] | Device Address [7:0] | R/W | The device address of the destination of LPM Token. In using LPM function, this setting is needed. Set the address of the device connected to the corresponding port before sending LPM Token. |
| | bit[24:23] | Suspend Status [1:0] | R | The device response to LPM Token. |
| | bit 9 | Suspend using L1 | R/W | In using LPM function, set this bit 1b. |
| | bit 7 | Suspend | R/W | When this bit is set to 1b this module starts LPM Token transaction, in condition of “Suspend using L1 bit = 1b” and “Device address field = H’000”. |
| PORT_LPM_CTR1 (offset: H’320) | bit[7:4] | NYET_RETRY_CNT_P1[3:0] | R/W | The number of retry for NYET response of device in LPM transaction. |
| | bit 3 | REMOTEWAKE_EN_P1 | R/W | Setting of LPM RemoteWakeup permission.<br>0b: permitted (default)<br>1b: not permitted |
| | bit 2 | SLEEP_INT_EN_P1 | R/W | In LPM transaction, setting to generate an interrupt or not when received a response other than ACK.<br>0b: not to generate an interrupt (default)<br>1b: to generate an interrupt |
| | bit 1 | RETRY_ENABLE_NYET_P1 | R/W | In LPM transaction, setting to the host behavior when received NYET response from the device.<br>0b: not to Retry (default)<br>1b: to Retry |
| | bit 0 | HIRD_SEL_P1 | R/W | Setting the duration time of the K-state drive, in resume of LPM state. |

### 32A.1.2.3 OTG function

Detects changing of the USB0_OTG_ID pin by U2H0_OBINT interrupt and switches the Host/Function role. Confirm the IDCHG_STA bit of **Section 32A.2.4.5(2), OTG-BC Interrupt Status Register** and the IDMON bit of **Section 32A.2.4.5(5), Line Control Port 1 Register**.

Host/Function role switching is performed by the OTG_PERI bit of **Section 32A.2.4.5(1), Common Control Register**.

| OTG_PERI bit |
|---|
| 0 : Host mode |
| 1 : Function mode |

### 32A.1.2.4 Battery-charging function

The Control and monitoring of the Battery Charging I/F of the USBPHY is set in the UCOM register of this module. The detail of the UCOM register, refer to **Section 32A.2.4.5, UCOM Register**.

### 32A.1.2.5 Suspend extension function

This module implements the following two suspend extension functions to reduce power consumption by stopping PLL of USBPHY.

The detail of the register, refer to **Section 32A.2.4.4(6), Suspend Control Register**.

[Relevant register]

Suspend Control Register (offset: H’308)

[Functional specification]

| Function | | Relevant bit | |
|---|---|---|---|
| [1] | Function to assert USBPHY SUSPENDM by asserting the Suspend bit in the OHCI/EHCI Operational Register | bit 31 | SUSPENDM_ENABLE |
| [2] | Function to forcibly assert USBPHY SUSPENDM | bit 0 | GLOBAL_SUSPENDM_P1 |

### 32A.1.3 Support of USB-Related Specifications

| USB-Related Specification or Function | | | Support |
|---|---|---|---|
| Host function | High Speed | Bulk IN/OUT transfer | ✓ |
| | | Control IN/OUT transfer | ✓ |
| | | Isochronous IN/OUT transfer | ✓ |
| | | Isochronous Highband transfer | ✓ |
| | | Interrupt IN/OUT transfer | ✓ |
| | Full Speed | Bulk IN/OUT transfer | ✓ |
| | | Control IN/OUT transfer | ✓ |
| | | Isochronous IN/OUT transfer | ✓ |
| | | Interrupt IN/OUT transfer | ✓ |
| | Low Speed | Control IN/OUT transfer | ✓ |
| | | Interrupt IN/OUTtransfer | ✓ |
| | No. of hub connection stages | HS: 5 stages | ✓ |
| | | FS: 5 stages | ✓ |
| | Support of EHCI V1.1 | Hardware Prefetching | × |
| | | Link Power Management | ✓ |
| | | Per-Port Change Events | ✓ |
| | | Shorter Periodic Frame List | ✓ |
| Battery-charging function (hereafter called the “BC function”) | | | ✓ |
| Dual role device * Function to statically switch between the host and peripheral modes | | | ✓ |

**Remarks:** ✓: support
×: no support

## 32A.2 Register Descriptions

### 32A.2.1 Register Attributes

Table 32A.3 Register Attributes

| Register Attribute | Description |
|---|---|
| R/W | Register bits can be read and written. |
| R/W(1) | Register bits can be read. A clear bit may be set by writing “1”; writing 0 to R/W(1) bits has no effect. |
| R/W(0) | Register bits can be read. A clear bit may be set by writing “0”; writing 1 to R/W(0) bits has no effect. |
| R | Register bits can only be read. |
| W | Register bits can only be written. |
| Reserved | Reserved bits are Read Only field. |

### 32A.2.2 Base Address

Table 32A.4 Base Addresses for Channel of the USB Host Module

| Channel | Base Address | |
|---|---|---|
| 0 | H’0_11C5_0000 | (Cortex-A55 Address Space) |

### 32A.2.3 Register Overview

| Register Name | Abbreviated Name | R/W | Initial Value | Offset Address | Access Size | Remarks |
|---|---|---|---|---|---|---|
| HcRevision | HcRevision | R | H'0000_0000 | H'000 | 32 | OHCI Operation Registers |
| HcControl | HcControl | R/W | H'0000_0000 | H'004 | 32 | |
| HcCommandStatus | HcCommandStatus | R/W | H'0000_0000 | H'008 | 32 | |
| HcInterruptStatus | HcInterruptStatus | R/W | H'0000_0000 | H'00C | 32 | |
| HcInterruptEnable | HcInterruptEnable | R/W | H'0000_0000 | H'010 | 32 | |
| HcInterruptDisable | HcInterruptDisable | R/W | H'0000_0000 | H'014 | 32 | |
| HcHCCA | HcHCCA | R/W | H'0000_0000 | H'018 | 32 | |
| HcPeriodCurrentED | HcPeriodCurrentED | R | H'0000_0000 | H'01C | 32 | |
| HcControlHeadED | HcControlHeadED | R/W | H'0000_0000 | H'020 | 32 | |
| HcControlCurrentED | HcControlCurrentED | R/W | H'0000_0000 | H'024 | 32 | |
| HcBulkHeadED | HcBulkHeadED | R/W | H'0000_0000 | H'028 | 32 | |
| HcBulkCurrentED | HcBulkCurrentED | R/W | H'0000_0000 | H'02C | 32 | |
| HcDoneHead | HcDoneHead | R | H'0000_0000 | H'030 | 32 | |
| HcFmInterval | HcFmInterval | R/W | H'0000_2EDF | H'034 | 32 | |
| HcFmRemaining | HcFmRemaining | R | H'0000_2EDF | H'038 | 32 | |
| HcFmNumber | HcFmNumber | R | H'0000_0000 | H'03C | 32 | |
| HcPeriodicStart | HcPeriodicStart | R/W | H'0000_0000 | H'040 | 32 | |
| HcLSThreshold | HcLSThreshold | R/W | H'0000_0628 | H'044 | 32 | |
| HcRhDescriptorA | HcRhDescriptorA | R/W | H'0F00_0901 | H'048 | 32 | |
| HcRhDescriptorB | HcRhDescriptorB | R/W | H'0002_0000 | H'04C | 32 | |
| HcRhStatus | HcRhStatus | R/W | H'0000_0000 | H'050 | 32 | |
| HcRhPortStatus1 | HcRhPortStatus1 | R/W | H'0000_0000 | H'054 | 32 | |
| HCIVERSION / CAPLENGTH | CAPL_VERSION | R | H'0110_0020 | H'100 | 32 | EHCI Capability Registers |
| HCSPARAMS | HCSPARAMS | R | H'0000_1191 | H'104 | 32 | |
| HCCPARAMS | HCCPARAMS | R | H'000E_0006 | H'108 | 32 | |
| HCSP_PORTROUTE | HCSP_PORTROUTE | R | H'0000_0000 | H'10C | 32 | |
| USBCMD | USBCMD | R/W | H'0008_0B00 | H'120 | 32 | EHCI Operation Registers |
| USBSTS | USBSTS | R/W | H'0000_1000 | H'124 | 32 | |
| USBINTR | USBINTR | R/W | H'0000_0000 | H'128 | 32 | |
| FRINDEX | FRINDEX | R/W | H'0000_0000 | H'12C | 32 | |
| CTRLDSSEGMENT | CTRLDSSEGMENT | R | H'0000_0000 | H'130 | 32 | |
| PERIODICLISTBASE | PERIODICLISTBASE | R/W | H'0000_0000 | H'134 | 32 | |
| ASYNCLISTADDR | ASYNCLISTADDR | R/W | H'0000_0000 | H'138 | 32 | |
| CONFIGFLAG | CONFIGFLAG | R/W | H'0000_0000 | H'160 | 32 | |
| PORTSC1 | PORTSC1 | R/W | H'0000_2000 | H'164 | 32 | |
| INT_ENABLE | INT_ENABLE | R/W | H'0000_0000 | H'200 | 32 | AHB Registers |
| INT_STATUS | INT_STATUS | R/W | H'0000_0000 | H'204 | 32 | |
| AHB_BUS_CTR | AHB_BUS_CTR | R/W | H'0000_0000 | H'208 | 32 | |
| USBCTR | USBCTR | R/W | H'0000_0002 | H'20C | 32 | |

| Register Name | Abbreviated Name | R/W | Initial Value | Offset Address | Access Size | Remarks |
|---|---|---|---|---|---|---|
| Register Enable/Clock Gating Control | REGEN_CG_CTRL | R/W | H'0000_0000 | H'304 | 32 | Core Defined Registers |
| Suspend Control | SPD_CTRL | R/W | H'0000_0000 | H'308 | 32 | |
| Suspend/Resume Timer Setting | SPD_RSM_TIMSET | R/W | H'01F4_03E8 | H'30C | 32 | |
| Overcurrent Detection/Sleep Timer Setting | OC_SLP_TIMSET | R/W | H'0C83_0D40 | H'310 | 32 | |
| SBRN/FLADJ/PORTWAKECAP | SBRN_FLADJ_PW | R/W | H'0003_2020 | H'314 | 32 | |
| PORT_LPM_CTRL1 | PORT_LPM_CTRL1 | R/W | H'0000_0000 | H'320 | 32 | |
| Common Control | COMMCTRL | R/W | H'8000_0000 | H'800 | 32 | OTG/BC Module Control Register |
| OTG-BC Interrupt Status | OBINTSTA | R/W | H'0000_0001 | H'804 | 32 | OTG/BC Interrupt Status Register |
| OTG-BC Interrupt Enable | OBINTEN | R/W | H'0000_0000 | H'808 | 32 | OTG/BC Interrupt Enable Register |
| VBUS Control | VBCTRL | R/W | H'0001_0000 | H'80C | 32 | OTG VBUS Control Register |
| Line Control Port 1 | LINECTRL1 | R/W | H'0000_0000 | H'810 | 32 | OTG USB Bus Control Register (Port 1) |
| BC Control Port 1 | BCCTRL1 | R/W | H'0300_0000 | H'820 | 32 | Battery Charging Control Register (Port 1) |

## 32A.2.4 Description of Registers

### 32A.2.4.1 OHCI Operational Register

#### (1) HcRevision Register

Abbreviated name of register: HcRevision

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | Revision | | | | | | | |
| 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 8 | — | All 0 | R | Reserved<br>The write value should always be 0. |
| 7 to 0 | Revision | All 0 | R | This field indicates the version of HCI specifications implemented in this host controller module.<br>Because this module conforms to OHCI standard 1.0a, H'10 is indicated. |

### (2) HcControl Register

Abbreviated name of register: HcControl

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | RWC | — | HCFS | | BLE | CLE | IE | PLE | CBSR | |
| 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 | 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 | — | All 0 | R | Reserved<br>The write value should always be 0. |
| 9 | RWC | 0 | R/W | Remote Wakeup Connected<br>This bit indicates whether the host controller supports remote wakeup signaling.<br>Software should set this bit to 1b in the initialization sequence if it is required to suppor remote wakeup.<br>Note that this bit can be initialized at hardware reset only.<br>0b: The remote wakeup is not supported.<br>1b: The remote wakeup is supported |
| 8 | — | 0 | R | Reserved<br>The write value should always be 0. |
| 7, 6 | HCFS | All 0 | R/W | Host Controller Functional State<br>This field indicates the operating state of the host controller.<br>00b: USB Reset<br>01b: USB Resume 10b: USB Operational<br>11b: USB Suspend<br>When the state is changed to USB Operational, the host controller module starts SOF transmission at 1 ms boundary.<br>This field is basically controlled by software, but it can be controlled by the host controller in the state of USB Suspend only. If the host controller (in the state of USB Suspend) detects a Remote Wakeup signal from the connecting device, the state in this field is changed to USB Resume.<br>Note that the reset value of this field differs between hardware reset and software reset.<br>Hardware reset: 00b (USB Reset)<br>Software reset: 11b (USB Suspend) |
| 5 | BLE | 0 | R/W | BulkList Enable<br>This bit sets whether the bulk list processing is performed for the next frame. The setting value of this bit is enabled from the next frame.<br>When you correct the bulk list, this bit must be 0b.<br>0b: The processing of the Bulk list is disabled.<br>1b: The processing of the Bulk list is enabled |
| 4 | CLE | 0 | R/W | ControlList Enable<br>This bit sets whether the control list processing is performed for the next frame. The setting value of this bit is enabled from the next frame.<br>When you correct the control list, this bit must be 0b.<br>0b: The processing of the Control list is disabled.<br>1b: The processing of the Control list is enabled. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 3 | IE | 0 | R/W | Isochronous Enable<br>This bit sets whether the isochronous ED processing is performed. The setting value of this bit is enabled from the next frame.<br>If the host controller module detects isochronous ED (F = 1) during the periodic list processing, it checks the bit and determines whether to perform isochronous ED processing.<br>1b: The processing of the isochronous ED is continued.<br>0b: The periodic list processing is stopped, and the bulk/control list processing is started.<br><br>0b: The processing of the isochronous ED is disabled.<br>1b: The processing of the isochronous ED is enabled. |
| 2 | PLE | 0 | R/W | Periodic List Enable<br>This bit indicates whether the periodic list processing is performed for the next frame. The setting value of this bit is enabled from the next frame.<br>The host controller module checks this bit before starting the periodic list processing.<br>0b: The processing of the periodic list is disabled.<br>1b: The processing of the periodic list is enabled. |
| 1, 0 | CBSR | All 0 | R/W | Control Bulk Service Ratio<br>This field defines the service ratio of the control transfer and bulk transfer.<br>When the periodic list is processed, the service ratio defined in this field is used for transfer. |

| CBSR | No. of Control EDs Over Bulk EDs Served |
|---|---|
| 00b | 1 : 1 |
| 01b | 2 : 1 |
| 10b | 3 : 1 |
| 11b | 4 : 1 |

### (3) HcCommandStatus Register

Abbreviated name of register: HcCommandStatus

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | SOC | |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | BLF | CLF | HCR |
| 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 | W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 18 | — | All 0 | R | Reserved<br>The write value should always be 0. |
| 17, 16 | SOC | All 0 | R | Scheduling Overrun Count<br>This field counts the number of scheduling overrun.<br>This field is initialized to 00b, and is counted up every time scheduling overrun is detected.<br>After the field is incremented to 11b, it returns to 00b. Even if the SO (Scheduling Overrun) bit in the HcInterrupt Status register is set, this field is counted up when scheduling overrun is detected. |
| 15 to 3 | — | All 0 | R | Reserved<br>The write value should always be 0. |
| 2 | BLF | 0 | R/W | BulkList Filled<br>This bit indicates whether TD exists in the bulk list.<br>To add TD to ED in the bulk list, set this bit to 1b.<br>The host controller checks this bit when it starts processing of the head ED in the bulk list. If this bit is set to 0b, the host controller does not start the list processing. If this bit is set to 1b, the host controller starts processing of the bulk list, and sets this bit to 0b.<br>When the host controller finds TD in the bulk list, it sets this bit to 1b again, and continues processing of the bulk list. When the host controller finishes the list processing, this bit is set to 0b.<br>However, if TD is not found in the bulk list, or if this bit is not set to 1b, this bit remains to be 0b, and the list processing stops.<br>To rebuild the list and start the list processing, before you set the BLE bit of the HcControl register and start the list processing, you need to set this bit.<br>0b: TD does not exist in the bulk list.<br>1b: TD exists in the bulk list. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 1 | CLF | 0 | R/W | ControlList Filled<br>This bit indicates whether TD exists in the control list.<br>To add TD to ED in the control list, set this bit to 1b.<br>The host controller checks this bit when it starts processing of the head ED in the control list.<br>If this bit is set to 0b, the host controller does not start the processing of the control list. If this bit is set to 1b, the host controller starts processing of the control list, and set this bit to 0b.<br>When the host controller finds TD in the control list, it sets this be to 1b again, and continues the list processing. When the host controller finishes the list processing, this bit is set to 0b.<br>However, if TD is not found in the control list, or if this bit is not set to 1b, this bit remains to be 0b, and the list processing stops.<br>To rebuild the list and start the list processing, before you set the CLE bit of the HcControl register and start the list processing, you need to set this bit.<br>0b: TD does not exist in the control list.<br>1b: TD exists in the control list. |
| 0 | HCR | 0 | W | Host Controller Reset<br>This bit is used to start OHCI software reset for the host controller.<br>When this bit is set to 1b, the operating status of the host controller is changed to USB Suspend regardless of the functional state of the host controller. Also, the most OHCI Operational registers and OHCI control circuits are initialized.<br>When the software reset finishes, the host controller clears this bit to 0b. |

### (4) HcInterruptStatus Register

Abbreviated name of register: HcInterruptStatus

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | RHSC | FNO | UE | RD | SF | WDH | SO |
| 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/W | R/W | R/W | R/W | R/W | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 7 | — | All 0 | R | Reserved<br>The write value should always be 0. |
| 6 | RHSC | 0 | R/W | Root Hub Status Change<br>Interrupt bit that indicates that the state of the HcRhStatus register or HcRhPortStatus1 register has changed.<br>When the Root Hub status is changed, the host controller sets this bit to 1b.<br>Writing 1b to this bit clears the interrupt.<br>0b: The status of root hub has not changed.<br>1b: The status of root hub has changed. |
| 5 | FNO | 0 | R/W | Frame Number Overflow<br>Interrupt bit that indicates that the MSB in the FrameNumber (FN) field of the HcFmNumber register has changed.<br>When the MSB in the Frame Number field is changed from 0 to 1, or from 1 to 0, this bit is set after HccaFrameNumber is updated.<br>Writing this bit to 1b clears the interrupt.<br>0b: The overflow of frame number has not occurred.<br>1b: The overflow of frame number has occurred. |
| 4 | UE | 0 | R/W | Unrecoverable Error<br>Interrupt bit that indicates that a system error that is not related to the USB (for example, an error on the system bus) has been detected.<br>Writing this bit to 1b clears the interrupt.<br>0b: The unrecoverable error has not occurred.<br>1b: The unrecoverable error has occurred. |
| 3 | RD | 0 | R/W | Resume Detected<br>Interrupt bit that indicates that Resume has been detected.<br>When the host controller detects the Resume signal (RemoteWakeup) from an USB device, it sets this bit to 1b.<br>This bit is not set when the Resume signal is sent by setting the HCFS field to USB Resume.<br>Writing this bit to 1b clears the interrupt.<br>0b: The host controller has not detected resume signaling (RemoteWakeup).<br>1b: The host controller has detected resume signaling (RemoteWakeup). |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 2 | SF | 0 | R/W | Start of Frame<br>Interrupt bit that indicates that Hcca Frame Number was updated when each frame started.<br>The host controller sends an SOF packet and updates HccaFrameNumber at the same time, and sets this bit to 1b.<br>Writing this bit to 1b clears the interrupt.<br>0b: The host controller has not started new frame.<br>1b: The host controller has started new frame. |
| 1 | WDH | 0 | R/W | Writeback DoneHead<br>Interrupt bit that indicates that the host controller has written the contents of HcDoneHead to HccDoneHead.<br>The host controller sets this bit to 1b immediately after it updates HccaDoneHead.<br>Note that HccaDoneHead is not updated until this bit is cleared.<br>Writing this bit to 1b clears the interrupt.<br>This bit must be cleared only after the contents of HccaDoneHead are saved.<br>0b: The write back to HccaDoneHead has not occurred.<br>1b: The write back to HccaDoneHead has occurred. |
| 0 | SO | 0 | R/W | Scheduling Overrun<br>Interrupt bit that indicates that overrun of the USB schedule occurred.<br>When USB scheduling overrun occurs, the host controller updates HccaFrameNumberl, and sets this bit to 1b. When this bit is set, the SchedulingOverrunCount field of the HcCommandStatus register is also incremented.<br>Writing this bit to 1b clears the interrupt.<br>0b: The scheduling overrun has not occurred.<br>1b: The scheduling overrun has occurred. |

### (5) HcInterruptEnable Register

Abbreviated name of register: HcInterruptEnable

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MIE | OCE | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| 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 | R | R | R | R | R | R | R |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | RHSCE | FNOE | UEE | RDE | SFE | WDHE | SOE |
| 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/W | R/W | R/W | R/W | R/W | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 | MIE | 0 | R/W | Master Interrupt Enable<br>This bit sets whether each Interrupt Enable setting (which is set in HcInterruptEnable [30:0]) is enabled.<br>If this bit is set to 0b, all OHCI interrupts are masked.<br>To clear this bit, set the MID bit (bit 31) of the HcInterruptDisable register to 1b.<br>0b: All interrupts are disabled.<br>1b: Interrupts that are set to 1b are enabled. |
| 30 | OCE | 0 | R/W | OC (Ownership Change) Interrupt Enable bit.<br>If this bit is set to 1b, the interrupt source becomes OC (OwnershipChange).<br>To clear this bit, set the OCD bit (bit 30) of the HcInterruptDisable register to 1b.<br>0b: OC (OwnershipChange) interrupt is disabled.<br>1b: OC (OwnershipChange) interrupt is enabled. |
| 29 to 7 | — | All 0 | R | Reserved<br>The write value should always be 0. |
| 6 | RHSCE | 0 | R/W | RHSC (RootHub Status Change) Interrupt Enable bit.<br>If this bit is set to 1b, the interrupt source becomes RHSC (RootHub Status Change).<br>To clear this bit, set the RHSCD bit (bit 6) of the HcInterruptDisable register to 1b.<br>0b: RHSC (RootHub Status Change) interrupt is disabled.<br>1b: RHSC (RootHub Status Change) interrupt is enabled. |
| 5 | FNOE | 0 | R/W | FNO (Frame Number Overflow) Interrupt Enable bit.<br>If this bit is set to 1b, the interrupt source becomes FNO (Frame Number Overflow).<br>To clear this bit, set the FNOD bit (bit 5) of the HcInterruptDisable register to 1b.<br>0b: FNO (Frame Number Overflow) interrupt is disabled.<br>1b: FNO (Frame Number Overflow) interrupt is enabled. |
| 4 | UEE | 0 | R/W | UE (Unrecoverable Error) Interrupt Enable bit.<br>If this bit is set to 1b, the interrupt source becomes UE (Unrecoverable Error).<br>To clear this bit, set the UED bit (bit 4) of the HcInterruptDisable register to 1b.<br>0b: UE (Unrecoverable Error) interrupt is disabled.<br>1b: UE (Unrecoverable Error) interrupt is enabled. |
| 3 | RDE | 0 | R/W | RD (Resume Detect) Interrupt Enable bit.<br>If this bit is set to 1b, the interrupt source becomes RD (Resume Detect).<br>To clear this bit, set the RDD bit (bit 3) of the HcInterruptDisable register to 1b.<br>0b: RD (Resume Detect) interrupt is disabled.<br>1b: RD (Resume Detect) interrupt is enabled. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 2 | SFE | 0 | R/W | SF (Start of Frame) Interrupt Enable bit.<br>If this bit is set to 1b, the interrupt source becomes SF (Start of Frame).<br>To clear this bit, set the SFD bit (bit 2) of the HcInterruptDisable register to 1b.<br>0b: SF (Start of Frame) interrupt is disabled.<br>1b: SF (Start of Frame) interrupt is enabled. |
| 1 | WDHE | 0 | R/W | WDH (Writeback DoneHead) Interrupt Enable bit.<br>If this bit is set to 1b, the interrupt source becomes WDH (Writeback DoneHead).<br>To clear this bit, set the WDHDD bit (bit 1) of the HcInterruptDisable register to 1b.<br>0b: WDH (Writeback DoneHead) interrupt is disabled.<br>1b: WDH (Writeback DoneHead) interrupt is enabled. |
| 0 | SOE | 0 | R/W | SO (Scheduling Overrun) Interrupt Enable bit.<br>If this bit is set to 1b, the interrupt source becomes SO (Scheduling Overrun).<br>To clear this bit, set the SOD bit (bit 0) of the HcInterruptDisable register to 1b.<br>0b: SO (Scheduling Overrun) interrupt is disabled.<br>1b: SO (Scheduling Overrun) interrupt is enabled. |

### (6) HcInterruptDisable Register

Abbreviated name of register: HcInterruptDisable

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MID | OCD | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W(1) | R/W(1) | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

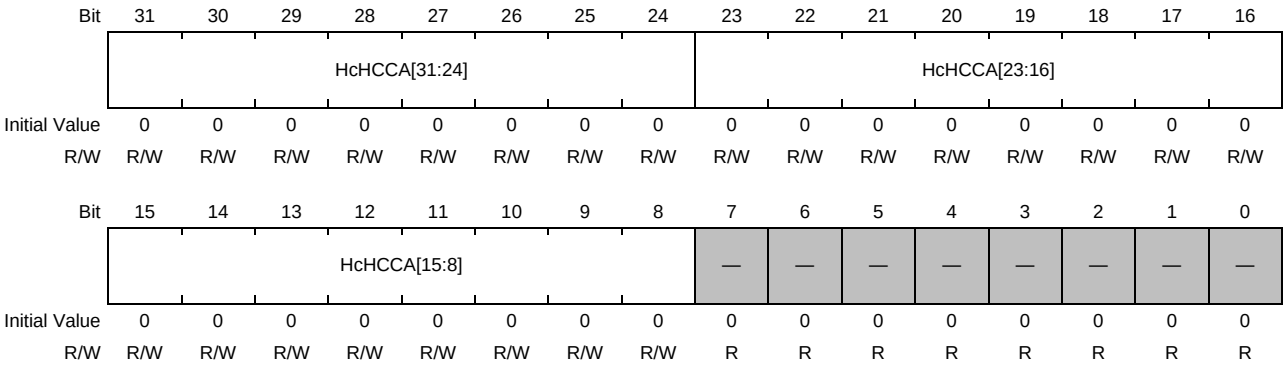
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | RHSCD | FNOD | UED | RDD | SFD | WDHD | SOD |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R/W(1) | R/W(1) | R/W(1) | R/W(1) | R/W(1) | R/W(1) | R/W(1) |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 | MID | 0 | R/W (1) | Master Interrupt Disable<br>This bit sets whether the Enable setting (for each interrupt) that is set by HcInterruptEnable[30:0] is disabled.<br>If this bit is set to 1b, the MIE bit (bit 31) of the HcInterruptEnable register is cleared to 0b, and all OHCI interrupts are masked.<br>Writing 0b to this bit is ignored. |
| 30 | OCD | 0 | R/W (1) | OC (Ownership Change) Interrupt Disable bit.<br>If this bit is set to 1b, the OCE bit (bit 30) of the HcInterruptEnable register is cleared to 0b, and OC (Ownership Change) is excluded from the interrupt source.<br>Writing 0b to this bit is ignored. |
| 29 to 7 | — | All 0 | R | Reserved<br>The write value should always be 0. |
| 6 | RHSCD | 0 | R/W (1) | RHSC (RootHub Status Change) Interrupt Disable bit.<br>If this bit is set to 1b, the RHSCE bit (bit 6) of the HcInterruptEnable register is cleared to 0b, and RHSC (RootHub Status Change) is excluded from the interrupt source.<br>Writing 0b to this bit is ignored. |
| 5 | FNOD | 0 | R/W (1) | FNO (Frame Number Overflow) Interrupt Disable bit.<br>If this bit is set to 1b, the FNOE bit (bit 5) of the HcInterruptEnable register is cleared to 0b, and FNO (Frame Number Overflow) is excluded from the interrupt source.<br>Writing 0b to this bit is ignored. |
| 4 | UED | 0 | R/W (1) | UE (Unrecoverable Error) Interrupt Disable bit.<br>If this bit is set to 1b, the UEE bit (bit 4) of the HcInterruptEnable register is cleared to 0b, and UE (Unrecoverable Error) is excluded from the interrupt source.<br>Writing 0b to this bit is ignored. |
| 3 | RDD | 0 | R/W (1) | RD (Resume Detected) Interrupt Disable bit.<br>If this bit is set to 1b, the RDE bit (bit 3) of the HcInterruptEnable register is cleared to 0b, and RD (Resume Detected) is excluded from the interrupt source.<br>Writing 0b to this bit is ignored. |
| 2 | SFD | 0 | R/W (1) | SF (Start of Frame) Interrupt Disable bit.<br>If this bit is set to 1b, the SFE bit (bit 2) of the HcInterruptEnable register is cleared to 0b, and SF (Start of Frame) is excluded from the interrupt source.<br>Writing 0b to this bit is ignored. |
| 1 | WDHD | 0 | R/W (1) | WDH (Writeback DoneHead) Interrupt Disable bit.<br>If this bit is set to 1b, the WDHE bit (bit 1) of the HcInterruptEnable register is cleared to 0b, and WDH (Writeback DoneHead) is excluded from the interrupt source.<br>Writing 0b to this bit is ignored. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 0 | SOD | 0 | R/W (1) | SO (Scheduling Overrun) Interrupt Disable bit.<br>If this bit is set to 1b, the SOE bit (bit 1) of the HcInterruptEnable register is cleared to 0b, and SO (Scheduling Overrun) is excluded from the interrupt source.<br>Writing 0b to this bit is ignored. |

### (7) HcHCCA Register

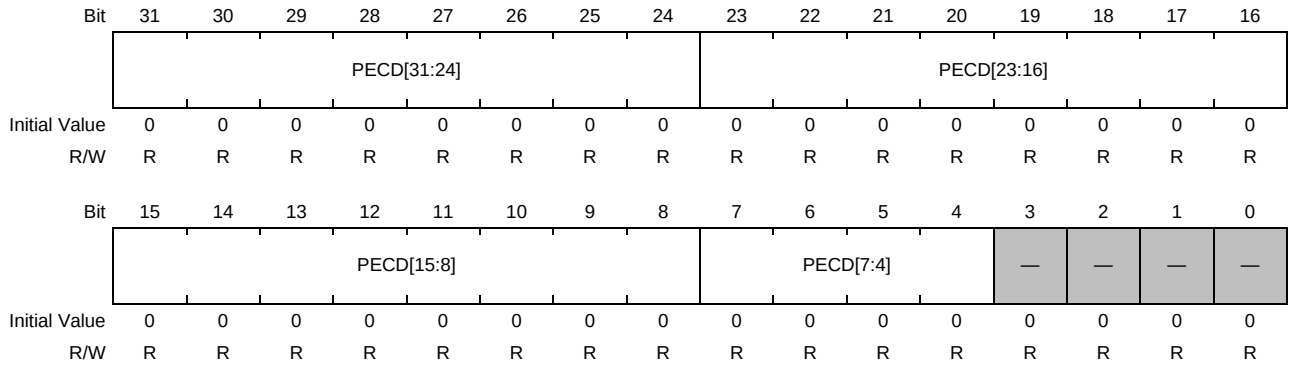
Abbreviated name of register: HcHCCA




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 8 | HcHCCA | All 0 | R/W | This field sets the base address (of the RAM) that is assigned as the Host Controller Communication Area.<br>This field must be set at initialization. The host controller requests (as HCCA) 256-byte area from the base address specified in this field. |
| 7 to 0 | — | All 0 | R | Reserved<br>The write value should always be 0. |

### (8) HcPeriodicCurrentED Register

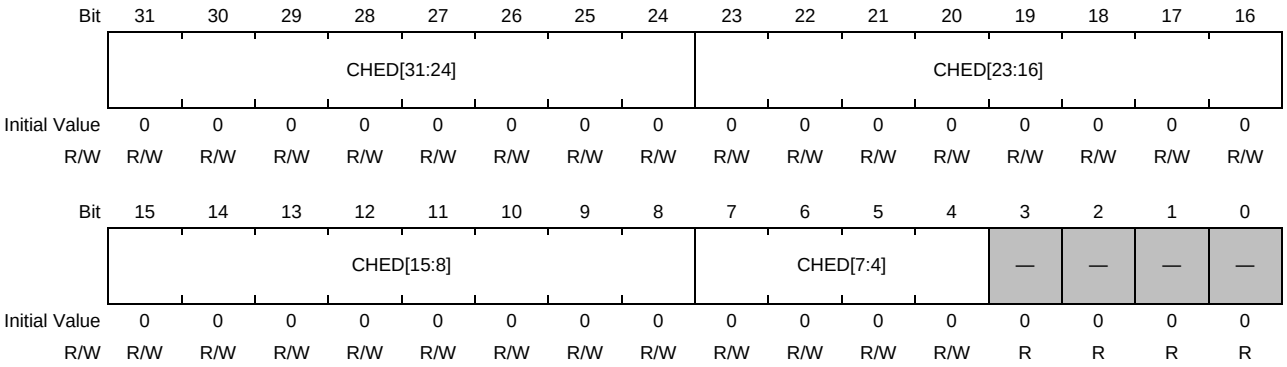
Abbreviated name of register: HcPeriodCurrentED




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 4 | PECD | All 0 | R | Period Current ED<br>This pointer indicates the physical address of ED in the periodic list that is currently processed.<br>The host controller updates this pointer when the list processing of the periodic ED finishes. |
| 3 to 0 | — | All 0 | R | Reserved<br>The write value should always be 0. |

### (9) HcControlHeadED Register

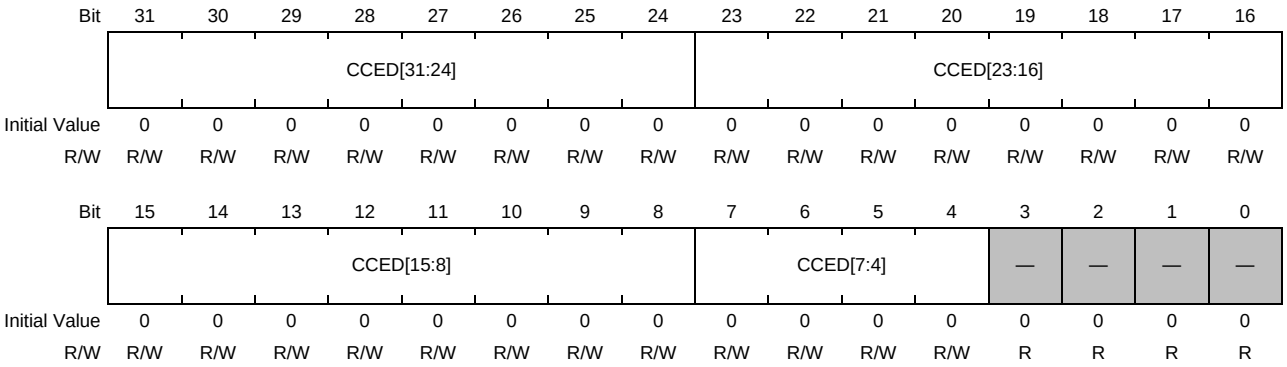
Abbreviated name of register: HcControlHeadED




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 4 | CHED | All 0 | R/W | Control Head ED<br>This field specifies the physical address of the head ED of the control list.<br>These bits must be set for control transfer before the CLE bit of the HcControl register is set.<br>The host controller starts processing of the control list from the HcBulkHeadED pointer. |
| 3 to 0 | — | All 0 | R | Reserved<br>The write value should always be 0. |

### (10) HcControlCurrentED Register

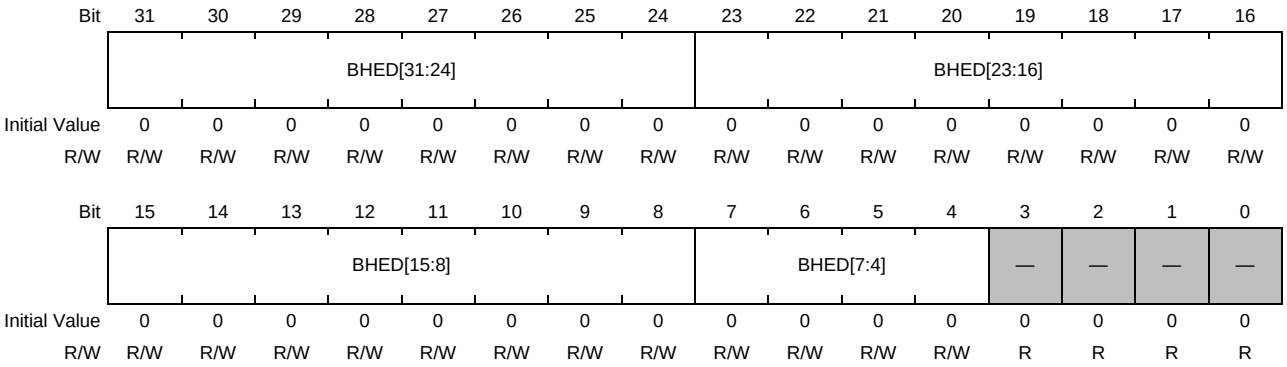
Abbreviated name of register: HcControlCurrentED




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 4 | CCED | All 0 | R/W | Control Current ED<br>This pointer indicates the physical address of the ED that is currently processed in the control list. After the current ED processing finishes, this pointer proceeds to the next ED.<br>The host controller continues the list processing until the end of the frame. When the end of the control list is reached, the host controller checks the ControlListFilled bit of the HcCommandStatus register. If the corresponding bit is set to 1b, the contents of the HcControlHeadED field are copied to the HcControlCurrentED field, and the ControlListFilled bit is cleared. If the corresponding bit is set to 0b, nothing is performed.<br>Update of this register is allowed only when the ControlListEnable bit of the HcControl register is cleared. If the ControlListEnable bit is set to 1b, the value of this register is only read. This register is initially set to H'0 to indicate the end of the control list. |
| 3 to 0 | — | All 0 | R | Reserved<br>The write value should always be 0. |

### (11) HcBulkHeadED Register

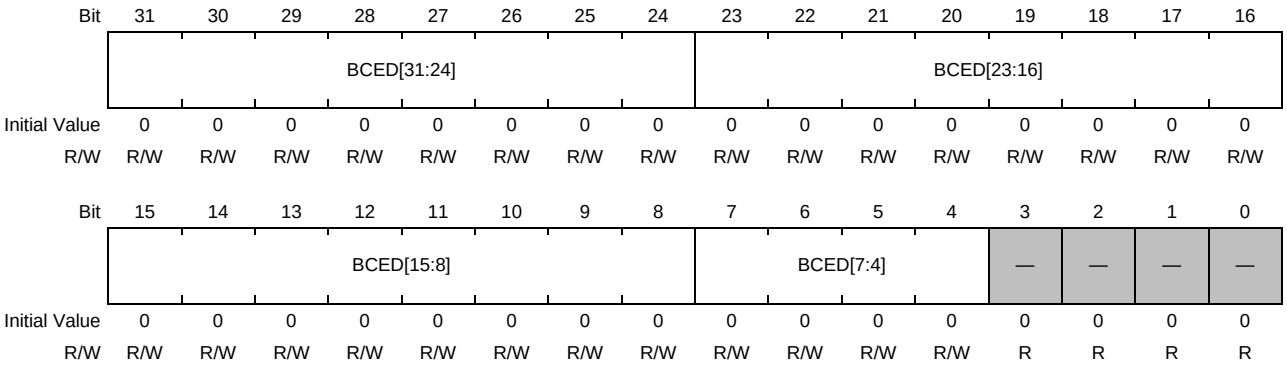
Abbreviated name of register: HcBulkHeadED




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 4 | BHED | All 0 | R/W | Bulk Head ED<br>This field specifies the physical address of the head ED of the bulk list.<br>This field must be set for bulk transfer before the BLE bit of the HcControl register is set.<br>The host controller starts the processing of the control list from the HcBulkHeadED pointer. |
| 3 to 0 | — | All 0 | R | Reserved<br>The write value should always be 0. |

### (12) HcBulkCurrentED Register

Abbreviated name of register: HcBulkCurrentED




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 4 | BCED | All 0 | R/W | Bulk Current ED<br>This pointer indicates the physical address of the ED that is currently processed in the bulk list. After the current ED processing finishes, this pointer proceeds to the next ED.<br>The host controller continues the list processing until the end of the frame. When the end of the bulk list is reached, the host controller checks the ControlListFilled bit of the HcCommandStatus register. If the corresponding bit is set to 1b, the contents of the HcBulkHeadED field are copied to the HcBulklCurrentED field, and the ControlListFilled bit is cleared. If the corresponding bit is set to 0b, nothing is performed.<br>Update of this register is allowed only when the ControlListEnable bit of the HcControl register is cleared. If the ControlListEnable bit is set to 1b, the value of this register is only read. This register is initially set to H'0 to indicate the end of the bulk list. |
| 3 to 0 | — | All 0 | R | Reserved<br>The write value should always be 0. |

### (13) HcDoneHead Register

Abbreviated name of register: HcDoneHead

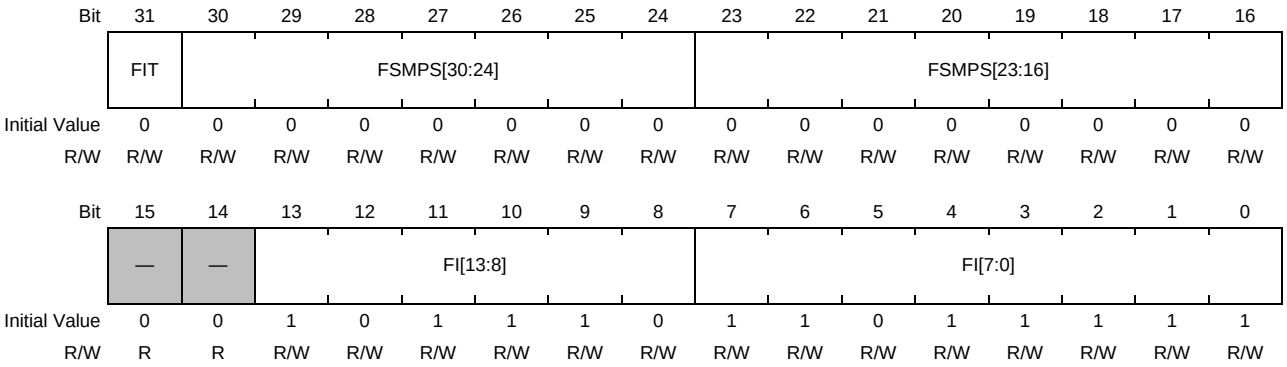
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | DH[31:24] | | | | | | | | DH[23:16] | | | | | | | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | DH[15:8] | | | | | | | | DH[7:4] | | | | — | — | — | — |
| 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 4 | DH | All 0 | R | Done Head<br>This field indicates the physical address of the HcDoneHead of the host controller.<br>The physical address of the TD that lately finished and is added to the Done queue.<br>After the TD processing finishes, the host controller writes the contents of the HcDoneHead to the NextTD field of the TD.<br>At the same time, the host controller overwrites the contents of HcDoneHead with the TD address.<br>After the host controller writes the contents of this register into HCCA, it sets 0b to this register.<br>Then, the WritebackDoneHead bit of the HcInterruptStatus register is set to 1b. |
| 3 to 0 | — | All 0 | R | Reserved<br>The write value should always be 0. |

### (14) HcFmInterval Register

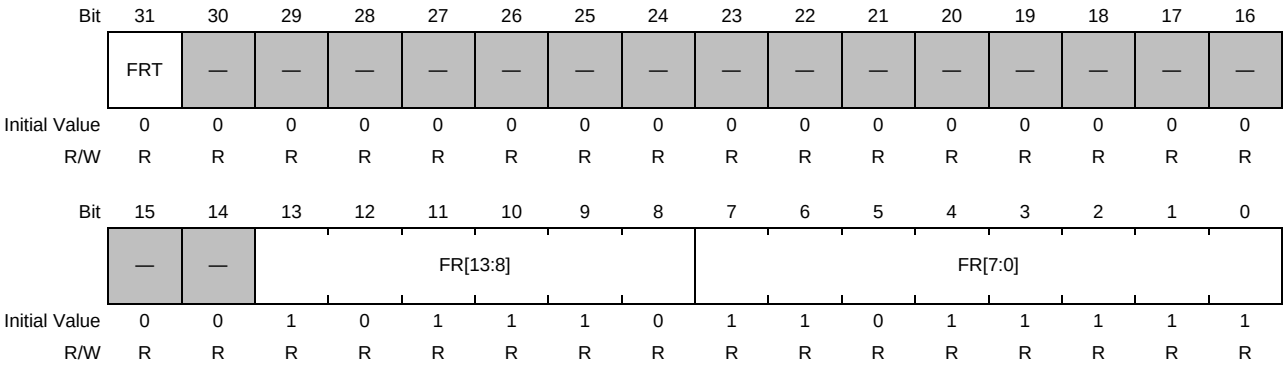
Register abbreviation: HcFmInterval




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 | FIT | 0 | R/W | Frame Interval Toggle<br>This bit is used to synchronize the frame setting value with the hardware (host controller). When the FI field is updated, the toggle value is written to this bit. When the FI field value is applied to the FR field of the HcFmRemaining register, the value of this bit is also applied to the FRT bit of the HcFmRemaining register.<br>By checking the toggle value in the FRT bit of the cFmRemaining register, the software can check whether the value in the FI field has been applied to the FR field of the HcFmRemaining register. |
| 30 to 16 | FSMPS | All 0 | R/W | FS Largest Data Packet<br>This field sets the maximum amount of data (bits) that the host controller can send and receive without the fear of schedule overrun.<br>The host controller compares the current frame position and the setting value in this field to determine the length (of the frame) that is ready to be transferred.<br>This value differs depending on the capacity of the system bus and other reasons, estimate the value and set it to this field.<br>*Note*: The maximum setting value for this field is H'2778. Do not set any value that is larger than H'2778. |
| 15, 14 | — | All 0 | R | Reserved<br>The write value should always be 0. |
| 13 to 0 | FI | H'2EDF | R/W | Frame Interval<br>This field is used to set the length of the frame (bit time) used for Full Speed.<br>Set the value of this field to "H'2EDF" so that 1 frame (= 1 ms) of USB standard is satisfied. |

### (15) HcFmRemaining Register

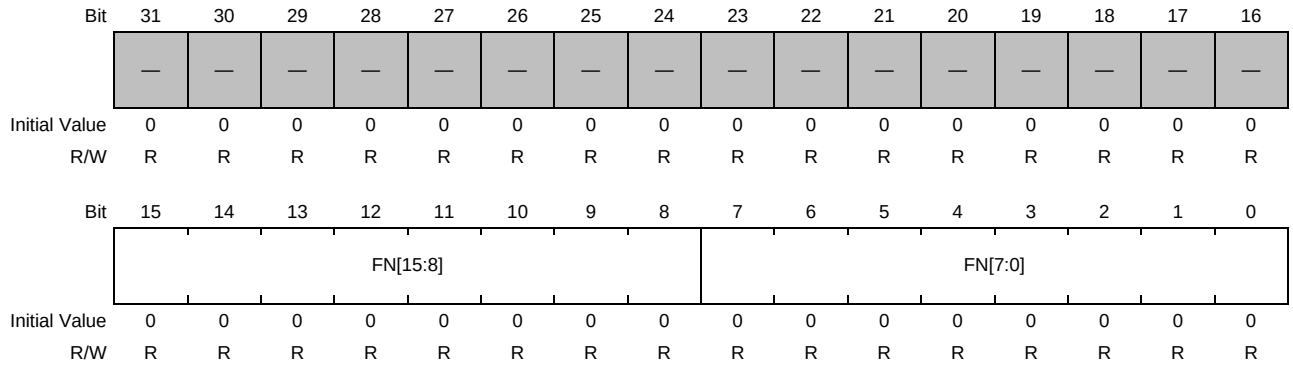
Register abbreviation: HcFmRemaining




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 | FRT | 0 | R | Frame Remaining Toggle<br>This bit is used to synchronize the frame setting value with the hardware (host controller). When the FR (Frame Remaining) field is set to H'0000, the host controller copies the FI (Frame Interval) field value to the FR field, and copies the FIT (Frame Interval Toggle) bit value to this bit. This bit can be used to check that the FI field of the HcFmInterval register has been correctly copied to the FR field. |
| 30 to 14 | — | All 0 | R | Reserved<br>The write value should always be 0. |
| 13 to 0 | FR | H'2EDF | R | Frame Remaining<br>This field indicates the current frame value for 14-bit down counter. The value in this field counts down as time passes. When the value becomes H'0000, the value of FI (Frame Interval) of the HcFmInterval register is loaded.<br>When the state of the host controller is changed to the USB Operational state, the host controller reloads the value in the FI (Frame Interval) field of the HcFmInterval register, and the new value is used from the next SOF. |

### (16) HcFmNumber Register

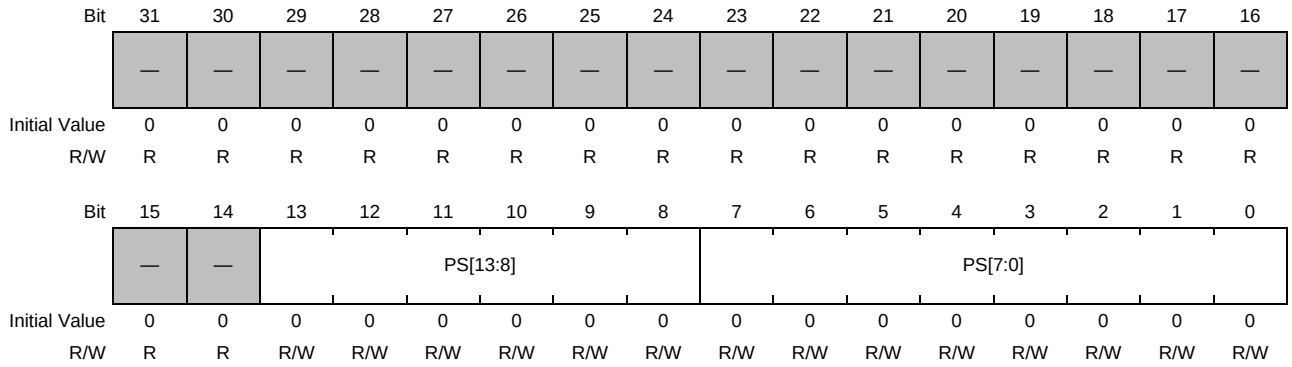
Register abbreviation: HcFmNumber




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 16 | — | All 0 | R | Reserved<br>The write value should always be 0. |
| 15 to 0 | FN | All 0 | R | Frame Number<br>This field indicates the number of passed frames.<br>When the HcFmRemaining register is reloaded, this field is incremented. If this field is reached to H'FFFF, the value is rolled over to H'0000. When the state of the host controller is changed to the USB Operational state, this field is automatically incremented. After the host controller increments the frame number at the frame boundary and sends SOF, the contents of this field are written to HCCA. This is performed before the host controller reads the first ED of the frame.<br>After writing to HCCA, the host controller sets the SF bit of the HcInterruptStatus register. |

### (17) HcPeriodicStart Register

Register abbreviation: HcPeriodicStart




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 14 | — | All 0 | R | Reserved<br>The write value should always be 0. |
| 13 to 0 | PS | All 0 | R/W | Periodic Start<br>This field indicates the time the host controller starts the periodic list processing in the frame. Estimate an appropriate value, and set the value to this field at the initial setting of the host controller. OHCI standard recommends that you set this setting value to about 90% of the FI field value of the HcFmInterval register.<br>The recommended value is H'2A2F.<br>When the value in the FR field of the HcFmRemaining register reaches the value set to this field, the periodic list processing is given priority over the control/bulk list processing.<br>Therefore, after the currently running control or bulk transfer finishes, the host controller starts the Interrupt list processing. |

### (18) HcLSThreshold Register

Register abbreviation: HcLSThreshold

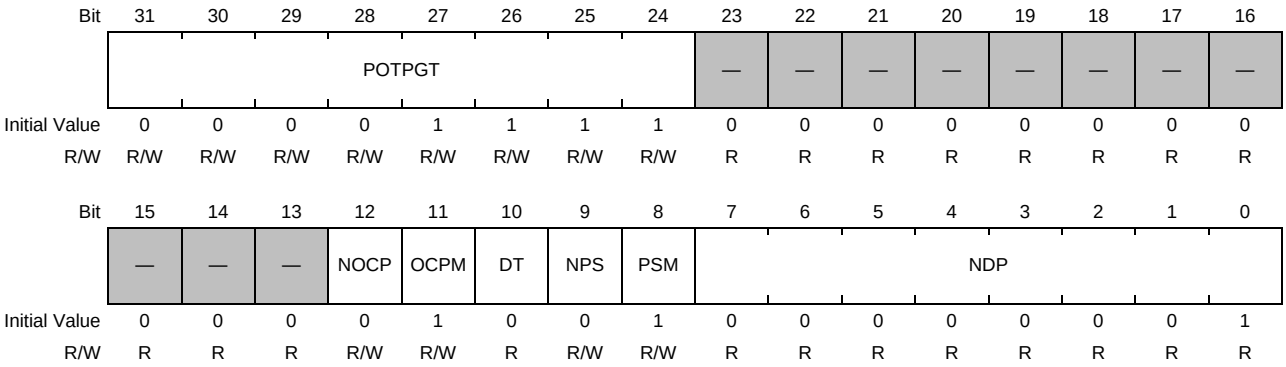
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | LST[11:8] | | | | LST[7:0] | | | | | | | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 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 | — | All 0 | R | Reserved<br>The write value should always be 0. |
| 11 to 0 | LST | H'628 | R/W | LS Threshold<br>This field indicates the threshold value of whether transfer is available for the remaining time of the LS transfer frame.<br>If the value of the FR field of the FmRemaining register is larger than the value set to this field, the host controller can start LS transfer. |

### (19) HcRhDescriptorA Register

Register abbreviation: HcRhDescriptorA




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 24 | POTPGT | H'0F | R/W | PowerOn To PowerGood Time<br>This field indicates the time for waiting until software can access the root hub port after the power is supplied to the port.<br>The unit of time is 2 ms. Therefore, the wait time is POTPGT × 2 ms. |
| 23 to 13 | — | All 0 | R | Reserved<br>The write value should always be 0. |
| 12 | NOCP | 0 | R/W | No OverCurrent Protection<br>This bit sets whether the overcurrent function of the root hub is supported.<br>If this bit is set to 0b, the overcurrent state is reported depending on the setting of the OCPM bit.<br>0b: The overcurrent state is supported.<br>1b: The overcurrent state is not supported.<br>If you do not require the overcurrent function, set this bit to 1b once the module is released from module standby. For details, see **Section 32A.9.1, Host/Peripheral Common Setting Sequence**. |
| 11 | OCPM | 1 | R/W | OverCurrent Protection Mode<br>This bit sets how to report the overcurrent state of the root hub.<br>If this bit is reset, this bit must indicate the same mode as the PSM (Power Switching Mode) bit.<br>This bit is valid only when the NOCP (NoOverCurrent Protection) bit is cleared (0b).<br>0b: The overcurrent state is collectively reported for all ports.<br>1b: The overcurrent state is reported for each port. |
| 10 | DT | 0 | R | Device Type<br>This bit indicates that the root hub is not a composite device.<br>This bit is always 0b because the root hub is not allowed to be a composite device. |
| 9 | NPS | 0 | R/W | No Power Switching<br>This bit sets how to control the port power.<br>If this bit is set to 0b, the PSM bit is used to set whether the power control is collectively performed for all ports or is performed for each port.<br>0b: The port power can be switched between on and off.<br>1b: The power is always on while the host controller is running. |
| 8 | PSM | 1 | R/W | Power Switching Mode<br>This bit sets how to control the power switch for each port of the root hub. This bit is valid only when the NPS bit is 0b.<br>0b: The power of all ports is collectively controlled.<br>1b: The power of ports is controlled for each port.<br>If the PPCM (PortPower Control Mask) bit of the HcRhDescriptorB register is set, each port responds only to the Set/ClearPortPower command. If the PPCM bit is cleared, each port is controlled by the Set/ClearGlobalPower command. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 to 0 | NDP | H'01 | R | Number Down stream Ports<br>This field indicates the number of downstream ports supported by the root hub. |

### (20) HcRhDescriptorB Register

Register abbreviation: HcRhDescriptorB

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | PPCM | — |
| 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 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | DR | — |
| 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 |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 18 | — | All 0 | R | Reserved<br>The write value should always be 0. |
| 17 | PPCM | 1 | R/W | PortPower Control Mask<br>This bit sets the port power control command when the PSM (Power Switching Mode) bit of the HcRhDescriptorA register is set.<br>If this bit is 0b, the Global Power Control command (Set/ClearGlobalPower) is used for control.<br>If this bit is 1b, the Port Power Control command (Set/ClearPortPower) is used for control.<br>If the PSM bit is 0b, this bit is ignored. |
| 16 to 2 | — | All 0 | R | Reserved<br>The write value should always be 0. |
| 1 | DR | 0 | R/W | Device Removable<br>This bit indicates whether each port of the root hub is removable.<br>If this bit is 0b, the connected device is removable.<br>If this bit is 1b, the connected device is not removable. |
| 0 | — | 0 | R | Reserved<br>The write value should always be 0. |

### (21) HcRhStatus Register

Register abbreviation: HcRhStatus

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | CRWE | — | — | — | — | — | — | — | — | — | — | — | — | — | OCIC | LPSC |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | W | R | R | R | R | R | R | R | R | R | R | R | R | R | R/W(1) | R/W |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | DRWE | — | — | — | — | — | — | — | — | — | — | — | — | — | OCI | LPS |
| 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 | R | R | R | R | R | R | R | R | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 | CRWE | 0 | W | Clear RemoteWakeup Enable<br>This bit is used to clear the DRWE bit.<br>If this bit is set to 1b, DRWE (Device RemoteWakeup Enable) bit can be cleared.<br>Writing 0b to this bit has no effect. |
| 30 to 18 | — | All 0 | R | Reserved<br>The write value should always be 0. |
| 17 | OCIC | 0 | R/W (1) | OverCurrent Indicator Change<br>This bit is used to report the change in the OCI bit.<br>If there is any change in the OCI bit, the host controller sets this bit to 1b.<br>If 1b is written to this bit while this bit is set to 1b, this bit can be cleared.<br>Writing 0b to this bit has no effect.<br>0b: There is no change in the Overcurrent state.<br>1b: There is a change in the Overcurrent state. |
| 16 | LPSC | 0 | R/W | The meaning of this bit differs depending on whether the operation is read or write.<br>[Read] Local Power Status Change<br>This bit is always read as 0b because the Local Power Status is not supported.<br>[Write] Set Global Power<br>If this bit is set to 1b, the power to the ports is turned on.<br>The ports whose power is turned on are determined by the settings of the PSM (Power Switching Mode) bit and the PPCM (Port Power Control Mask) bit of the HcRhDescriptorA register. |

| written to this bit | PSM | PPCM[N] | Description |
|---|---|---|---|
| 0 | — | — | Setting ignored |
| 1 | 0 | — | 1b is set to the PPS bit. |
| | 1 | 0 | 1b is set to the PPS bit. |
| | | 1 | Setting ignored |

| | | | | |
|---|---|---|---|---|
| | | | | Writing 0b to this bit has no effect. |
| 15 | DRWE | 0 | R/W | [Read] Device RemoteWakeup Enable<br>This bit sets whether the RemoteWakeup event includes the CSC (Connect Status Change) bit.<br>If this bit is set to 1b, the CSC bit of the HcRhPortStatus register becomes valid as the Resume event. If the CSC bit is changed to 1b, the state is changed from USB Suspend to USB Resume, and the Resume detection interrupt occurs.<br>0b: Connect Status Change is not the source of RemoteWakeup.<br>1b: Connect Status Change is the source of RemoteWakeup.<br>[Write] Set RemoteWakeup Enable<br>This bit is used to set the DRWE bit.<br>Writing 0b to this bit has no effect. |

<table>
<tr><th>Bit</th><th>Bit Name</th><th>Initial Value</th><th>R/W</th><th>Description</th></tr>
<tr><td>14 to 2</td><td>—</td><td>All 0</td><td>R</td><td>Reserved<br>The write value should always be 0.</td></tr>
<tr><td>1</td><td>OCI</td><td>0</td><td>R</td><td>Over Current Indicator<br>This bit is used to report the overcurrent state in the global overcurrent detection mode (OCPM bit = 0b).<br>This bit always indicates 0b when overcurrent for each port is reported (when OPCM bit = 1b).<br>0b: The port state is normal.<br>1b: The port is in the overcurrent state.</td></tr>
<tr><td>0</td><td>LPS</td><td>0</td><td>R/W</td><td>The meaning of this bit differs depending on whether the operation is read or write.<br>[Read] Local Power Status<br>This bit is always read as 0b because the Local Power Status is not supported.<br>[Write] Clear Global Power<br>If this bit is set to 1b, the power to the ports is turned off.<br>The ports whose power is turned off are determined by the settings of the PSM (Power Switching Mode) bit and PPCM (Port Power Control Mask) bit of the HcRhDescriptorA register.
<table>
<tr><th>written to this bit</th><th>PSM</th><th>PPCM[N]</th><th>Description</th></tr>
<tr><td>0</td><td>—</td><td>—</td><td>Setting ignored.</td></tr>
<tr><td rowspan="3">1</td><td>0</td><td>—</td><td>The PPS bit is cleared to 0b.</td></tr>
<tr><td rowspan="2">1</td><td>0</td><td>The PPS bit is cleared to 0b.</td></tr>
<tr><td>1</td><td>Setting ignored.</td></tr>
</table>
Writing 0b to this bit has no effect.</td></tr>
</table>

### (22) HcRhPortStatus[1:NDP] Register

Register abbreviation: HcRhPortStatus1

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | PRSC | OCIC | PSSC | PESC | CSC |
| 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(1) | R/W(1) | R/W(1) | R/W(1) | R/W(1) |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | LSDA | PPS | — | — | — | PRS | POCI | PSS | PES | CCS |
| 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/W | R/W | R/W | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 21 | — | All 0 | R | Reserved<br>The write value should always be 0. |
| 20 | PRSC | 0 | R/W (1) | Port Reset Status Change<br>This bit indicates that port reset (bus reset) has finished.<br>The host controller sets this bit when 10 ms port reset (bus reset) finishes.<br>0b: Port reset has not finished, or the PRS (Port Reset Status) bit is not changed.<br>1b: Port reset has finished. |
| 19 | OCIC | 0 | R/W (1) | Port OverCurrent Indicator Change<br>This bit is set when the overcurrent state of the port is detected.<br>This bit is set when the host controller changed the POCI bit value.<br>This bit is valid only under the setting that the overcurrent state is reported for each port (OCPM bit = 1b).<br>If this bit is set to 1b, this bit is cleared.<br>Writing 0b to this bit has no effect.<br>0b: The overcurrent state has not changed.<br>1b: The overcurrent state has changed (POCI bit changed). |
| 18 | PSSC | 0 | R/W (1) | Port Suspend Status Change<br>This bit indicates that the Resume sequence finished.<br>This sequence includes 20 ms of the Resume signal, LS EOP, and 3 ms of resynchronization delay.<br>If this bit is set to 1b, this bit is cleared.<br>Writing 0b to this bit has no effect.<br>If the PRSC (Port Reset Status Change) bit is set, this bit is cleared.<br>0b: The Resume sequence has not finished.<br>1b: The Resume sequence has finished. |
| 17 | PESC | 0 | R/W (1) | Port Enable Status Change<br>This bit indicates that the PES (Port Enable Status) bit was changed.<br>If a hardware event clears the PES bit, this bit is set to 1b.<br>If this bit is set to 1b, this bit is cleared.<br>Writing 0b to this bit has no effect.<br>0b: The PES (Port Enable Status) bit has not changed.<br>1b: The PES (Port Enable Status) bit has changed. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 16 | CSC | 0 | R/W (1) | Connect Status Change<br>This bit indicates that the CCS (Current Connect Status) bit was changed.<br>If the Connect/Disconnect event occurs, this bit is set to 1b.<br>If this bit is set to 1b, this bit is cleared.<br>Writing 0b to this bit has no effect.<br>If a request (Port Reset/Port Enable/Port Suspend) is received during the Disconnect status, this bit is set for reevaluation of device connection confirmation.<br>0b: The CCS (Current Connect Status) bit has not changed.<br>1b: The CCS (Current Connect Status) bit has changed. |
| 15 to 10 | — | All 0 | R | Reserved<br>The write value should always be 0. |
| 9 | LSDA | 0 | R/W | The meaning of this bit differs depending on whether the operation is read or write.<br>[Read] Low Speed Device Attached<br>This bit indicates the speed of the device connected to the port.<br>This bit is valid only when the CCS (Current Connect Status) bit is set.<br>0b: Full Speed device is connected.<br>1b: Low Speed device is connected.<br>[Write] Clear Port Power<br>This bit is used to turn off the power of the port when the port is power controlled.<br>Writing 1b to this bit turns off the port power.<br>Writing 0b to this bit has no effect. |
| 8 | PPS | 0 | R/W | The meaning of this bit differs depending on whether the operation is read or write.<br>[Read] Port Power Status<br>This bit indicates the power status of the port.<br>This bit is cleared when the overcurrent is detected.<br>0b: Port power is off.<br>1b: Port power is on.<br>[Write] Set Port Power<br>This bit is used to turn on the power of the port when the port is power controlled.<br>Writing 1b to this bit turns on the port power.<br>Writing 0b has no effect. |
| 7 to 5 | — | All 0 | R | Reserved<br>The write value should always be 0. |
| 4 | PRS | 0 | R/W | The meaning of this bit differs depending on whether the operation is read or write.<br>[Read] Port Reset Status<br>This bit indicates the port reset (bus reset) status.<br>When 10 ms of port reset finishes, the PRSC (Port Reset Status Change) bit is set and this bit is cleared. When the CCS bit is cleared (when no device is connected), this bit cannot be set.<br>0b: Not during port reset<br>1b: During port reset<br>[Write] Set Port Reset<br>This bit is used to issue a port reset (bus reset) to the downstream port.<br>Writing 1b to this bit starts 10 ms of port reset.<br>Writing 0b to this bit has no effect.<br>If CCS is cleared, this bit cannot be set. Instead, CSC is set to 1b.<br>This is performed to report to ports to which no device is connected, that port reset was performed. |

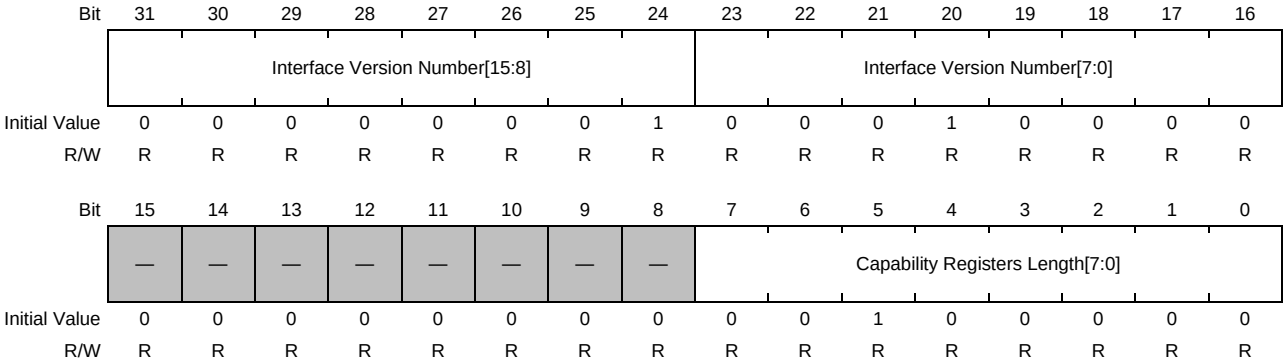
| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 3 | POCI | 0 | R/W | The meaning of this bit differs depending on whether the operation is read or write.<br>[Read] Port OverCurrent Indicator<br>This bit indicates that the downstream port is in the overcurrent status.<br>This bit is valid only when the setting is specified that the overcurrent status is reported for each port (OCPM bit = 1b).<br>On the other hand, if the setting is specified that the overcurrent status is reported for all ports, this bit is always read as 0b.<br>0b: The port is in normal status.<br>1b: The port is in overcurrent status.<br>[Write] Clear Suspend Status<br>This bit is used to finish the Suspend status and start the Resume sequence.<br>Writing 1b to this bit starts the Resume sequence.<br>Writing 0b has no effect.<br>The Resume sequence starts only when PSS (Port Suspend Status) is set. |
| 2 | PSS | 0 | R/W | The meaning of this bit differs depending on whether the operation is read or write.<br>[Read] Port Suspend Status<br>This bit indicates that the port status is in Suspend or Resume sequence.<br>0b: The port is not in Suspend status.<br>1b: The port is in Suspend status.<br>When the CCS bit is cleared (when no device is connected), this bit cannot be set.<br>This bit is cleared under the following conditions:<br>• When the Resume sequence finished, and the PSSC (Port Suspend Status Change) bit is set<br>• When the port reset finished, and the PRSC (Port Reset Status Change) bit is set<br>• When the host controller is in the USB Resume state<br>[Write] Set Port Suspend<br>This bit is used to change the port status to Suspend.<br>Writing 1b to this bit changes the status to Suspend.<br>Writing 0b to this bit has no effect.<br>If CCS is cleared, this bit cannot be set. Instead, CSC is set to 1b.<br>This is performed to report to the ports to which no device is connected that the Suspend command was issued. |
| 1 | PES | 0 | R/W | The meaning of this bit differs depending on whether the operation is read or write.<br>[Read] Port Enable Status<br>This bit indicates whether the port status is Enable or Disable.<br>If the host controller detects a bus error (for example, overcurrent state, disconnect, port power off, and babble error), it clears this bit. Then, the PESC (PortEnableStatusChange) bit is set.<br>When the CCS bit is cleared (when no device is connected), this bit cannot be set.<br>This bit is set “when port reset finished and the PRSC bit is set” or “when the port status becomes Suspend and the PRSC bit is set”.<br>0b: The port status is Disable.<br>1b: The port status is Enable.<br>[Write] Set Port Enable<br>This bit is used to set the PES bit.<br>Writing 1b to this bit changes the port status to Enable.<br>Writing 0b to this bit has no effect.<br>If CCS is cleared, this bit cannot be set. Instead, CSC is set to 1b.<br>This is performed to report that the status of the ports to which no device is connected was tried to be changed to Enable. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 0 | CCS | 0 | R/W | The meaning of this bit differs depending on whether the operation is read or write.<br>[Read] Current Connect Status<br>The current connection status of the downstream port is applied to this bit.<br>0b: No device is connected.<br>1b: A device is connected.<br>[Write] Clear Port Enable<br>This bit is used to clear the PES (Port Enable Status) bit.<br>Writing 1b to this bit changes the port status to Disable. Writing 0b to this bit has no effect. |

### 32A.2.4.2 EHCI Controller Capability Register

#### (1) HCIVERSION/CAPLENGTH Register

Register abbreviation: CAPL_VERSION




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 16 | Interface Version Number | H'0110 | R | This field indicates the EHCI version supported by the host controller.<br>H'0110 is indicated because this host controller supports EHCI Rev1.1. |
| 15 to 8 | — | All 0 | R | Reserved<br>The write value should always be 0. |
| 7 to 0 | Capability Registers Length | H'20 | R | This field is used as an offset that is added to the base address to find the start address of the EHCI Operational register.<br>H'20 is indicated because the EHCI Operation register of this module starts from H'20. |

### (2) HCSPARAMS Register

Register abbreviation: HCSPARAMS

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | Debug Port Number [3:0] | | | | — | — | — | P_INDICATOR |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | N_CC | | | | N_PCC | | | | Port Routing Rules | — | — | PPC | N_PORTS[3:0] | | | |
| Initial Value | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 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 24 | — | All 0 | R | Reserved The write value should always be 0. |
| 23 to 20 | Debug Port Number | All 0 | R | This field indicates that the host controller port is a debug port.<br>0000b is indicated because this module does not have a debug port. |
| 19 to 17 | — | All 0 | R | Reserved<br>The write value should always be 0. |
| 16 | P_INDICATOR | 0 | R | Port Indicators<br>This bit indicates whether the host controller supports port indicator control.<br>0b is indicated because this module does not support port indicator control. |
| 15 to 12 | N_CC | H'1 | R | Number of Companion Controller<br>This field indicates the number of OHCI host controllers implemented in this module.<br>H'1 is indicated because this module has one OHCI host controller. |
| 11 to 8 | N_PCC | H'1 | R | Number of Ports per Companion Controller<br>This field indicates the number of ports supported by an OHCI host controller.<br>The setting value of the Port_no field of the PCI Configuration EXT1 register is applied to this field. |
| 7 | Port Routing Rules | 1 | R | This bit indicates how individual ports are mapped to the OHCI host controller.<br>This bit indicates 1b because, in this module, the contents of the HCSP_PORTROUTE register show the mapping method. |
| 6, 5 | — | All 0 | R | Reserved<br>The write value should always be 0. |
| 4 | PPC | 1 | R | Port Power Control<br>This bit indicates how the port power of this module is controlled.<br>0b: The port power is always on.<br>1b: The PP bit of the PORTSC register controls the port power. |
| 3 to 0 | N_PORTS | H'1 | R | This field indicates the number of physical downstream ports used by this module.<br>0001b: 1 Port |

### (3) HCCPARAMS Register

Register abbreviation: HCCPARAMS

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | 32-Frame Periodic List Capability | Per-Port Change Event Capability | Link Power Management Capability | Hardware Prefetch Capability |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R | R |

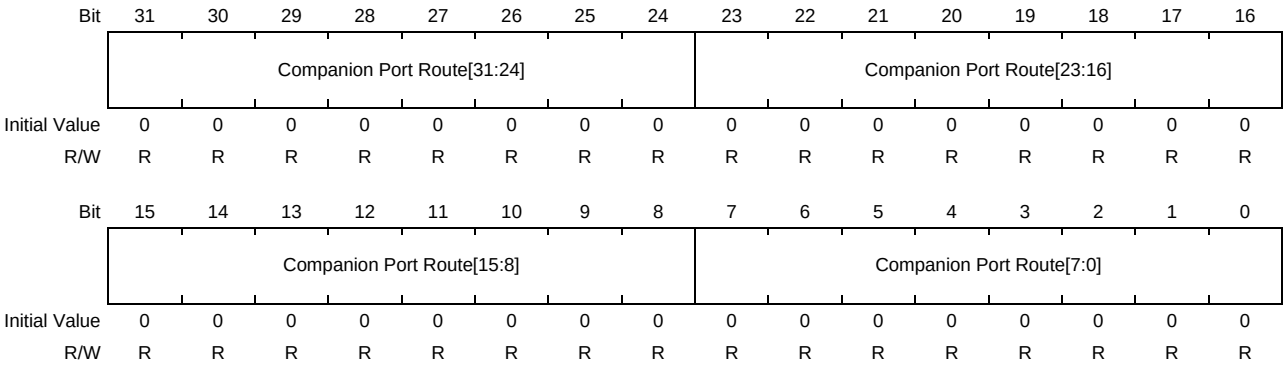
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | EECP | | | | | | | | Isochronous Scheduling Threshold | | | | — | Asynchronous Schedule Park Capability | Programmable Frame List Flag | 64-bit Addressing Capability |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 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 20 | — | All 0 | R | Reserved<br>The write value should always be 0. |
| 19 | 32-Frame Periodic List Capability | 1 | R | This bit indicates 1b because this module supports 32-Frame Periodic List defined in EHCI V1.1.<br>This means that the Frame List Size field of the USBCMD register is set to 11b, so this module supports 32-Frame Periodic List. |
| 18 | Per-Port Change Event Capability | 1 | R | This bit indicates 1b because this module supports event detection function for ports that are defined in EHCI V1.1.<br>This means that this module supports event detection function for ports, and is associated with the Pre-Port Change Event Enable field of the USBCMD register, Port-1 Change Detect field of the USBSTS register, and Port-1 Change Interrupt Enable field of the USBINT register.<br>If this bit is 0b, the corresponding fields of the above registers are treated as Reserved. |
| 17 | Link Power Management Capability | 1 | R | This bit indicates 1b because this module supports LPM (Link Power Management) defined in EHCI V1.1.<br>This means that this module supports LPM L1 state, and is controlled by the Suspend using L1 bit, Suspend Status bit, and Device Address field of the PORTSC register.<br>If this bit is 0b, the corresponding bits and field of the above PORTC register are treated as Reserved. |
| 16 | Hardware Prefetch Capability | 0 | R | This bit indicates 0b because this module does not support the hardware prefetch function defined in EHCI V1.1. |
| 15 to 8 | EECP | All 0 | R | This field indicates the offset address of the EHCI Extend Capabilities Registers.<br>This field indicates H'00 because this module does not use EHCI Extend Capabilities Registers. |
| 7 to 4 | Isochronous Scheduling Threshold | All 0 | R | This field indicates H'0 because this module does not support caches with isochronous data structure for the entire frame. |
| 3 | — | 0 | R | Reserved<br>The write value should always be 0. |
| 2 | Asynchronous Schedule Park Capability | 1 | R | This bit indicates whether the Park mode is supported for High Speed QH (Queue Head) in an asynchronous schedule.<br>This bit indicates 1b because this module supports the above function. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 1 | Programmable Frame List Flag | 1 | R | This bit indicates the setting for the available frame list size.<br>This bit indicates 1b for this module.<br>If this bit is set to 1b, bit [3:2] (Frame List Size) of the USBCMD register can be used to set the available frame list size, and the frame list size smaller than 4 Kbyte is configurable. |
| 0 | 64-bit addressing Capability | 0 | R | This bit indicates which type of memory pointer the data structure uses (32 bit address memory pointer or 64 bit address memory pointer).<br>This bit indicates 0b for this module because this module has the data structure that uses 32 bit address memory pointer. 64 bit address is not supported. |

### (4) HCSP-PORTROUTE Register

Register abbreviation: HCSP_PORTROUTE




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 0 | Companion Port Route[31:0] | All 0 | R | This field indicates the ports controlled by the OHCI host controller.<br>This field indicates 0b for this module because this module has one OHCI host controller. |

### 32A.2.4.3 HCI Operational Register

#### (1) USBCMD Register

Register abbreviation: USBCMD

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | Host-Initiated Resume Duration | | | | Interrupt Threshold Control | | | | | | | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| R/W | R | R | R | R | R/W | R/W | R/W | R/W | R | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Per-Port Change Events Enable | — | — | — | Asynchronous Schedule Park Mode Enable | — | Asynchronous Schedule Park Mode Count | | — | Interrupt on Async Advance Doorbell | Asynchronous Schedule Enable | Periodic Schedule Enable | Frame List Size | | HCRESET | RS |
| Initial Value | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R | R | R | R/W | R | R/W | R/W | R | R/W | R/W | R/W | R/W | R/W | W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 28 | — | All 0 | R | Reserved<br>The write value should always be 0. |
| 27 to 24 | Host-Initiated Resume Duration | All 0 | R/W | This field indicates the minimum time of the K-state drive while the host controller resumes from the LPM (L1) state.<br>The value in this field is sent to the connected device that has the LPM function via the HIDR field in the bmAttributes field of the LPM token.<br>The encoded value of this field is defined as the name of the HIRD field in the LPM Token.<br>Specifically, H'0 means 50 µs, and if the value is incremented by 1, 75 µs is incremented. For example, H'1 means 125 µs, and H'4 means 1175 µs. |
| 23 to 16 | Interrupt Threshold Control | H'08 | R, R/W | This field indicates the maximum rate until the host controller issues an interrupt. Note that the setting values other than the following values are not guaranteed.<br>Value / Maximum Interrupt Interval<br>H'00 Reserved<br>H'01 1 micro-frame<br>H'02 2 micro-frames<br>H'04 4 micro-frames<br>H'08 8 micro-frames (default, equals to 1 ms)<br>H'10 16 micro-frames (2 ms)<br>H'20 32 micro-frames (4 ms)<br>H'40 64 micro-frames (8 ms)<br>Do not set H'00 to this field while the Halted bit is 0b. |
| 15 | Per-Port Change Events Enable | 0 | R/W | This field is used to enable the event report function of the ports defined by the Port-1 Change Detect field of the USBSTS register and the Port-1 Change Detect Enable field of the USBINTR register.<br>0b: The event report function of the ports is disabled.<br>1b: The event report function of the ports is enabled. |
| 14 to 12 | — | All 0 | R | Reserved<br>The write value should always be 0. |
| 11 | Asynchronous Schedule Park Mode Enable | 1 | R/W | This bit enables or disables the Asynchronous Schedule Park mode.<br>0b: The Park mode is disabled.<br>1b: The Park mode is enabled. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 10 | — | 0 | R | Reserved<br>The write value should always be 0. |
| 9, 8 | Asynchronous Schedule Park Mode Count | 11b | R/W | This field sets the number of transactions that the host controller can serially execute for one QH (Queue Head) fetch in an asynchronous schedule.<br>The valid value range is H'1 to H'3.<br>This field is valid when bit 11 (Asynchronous Schedule Park Mode Enable) is 1b.<br>Do not set H'0 to this field. |
| 7 | — | 0 | R | Reserved<br>The write value should always be 0. |
| 6 | Interrupt on Async Advance Doorbell | 0 | R/W | This bit is used as the doorbell.<br>In an asynchronous schedule processing, if you want an interrupt to occur before proceeding to the next QH (Queue), set this bit to 1b.<br>After a QH processing normally finishes, the host controller clears this bit to 0b, and sets bit 5 (Interrupt on Async Advance bit) of the USBSTS register to 1b.<br>If bit 5 (Interrupt on Async Advance Enable bit) of the UBINTR register is set to 1b, an interrupt occurs at the next interrupt timing.<br>If the asynchronous schedule is disabled, do not write 1b to this bit. |
| 5 | Asynchronous Schedule Enable | 0 | R/W | This bit sets whether the host controller proceeds to the asynchronous list processing or skip the processing.<br>0b: The asynchronous list processing is skipped.<br>1b: Use the ASYNCLISTADDR register to proceed the asynchronous list processing. |
| 4 | Periodic Schedule Enable | 0 | R/W | This bit sets whether the host controller proceeds or skips the periodic list processing.<br>0b: The periodic list processing is skipped.<br>1b: Use the PERIODICLISTBASE register to proceed the periodic list processing. |
| 3, 2 | Frame List Size | 0 | R/W | This field specifies the frame list size.<br>The setting value of this field determines the size of the Frame List Current index of the FRINDEX register.<br>Value: 00b — the number of frames in Frame List: 1024 frames (default)<br>01b: 512 frames<br>10b: 256 frames<br>11b: 32 frames |
| 1 | HCRESET | 0 | W | Host Controller Reset<br>This bit is used to initialize the EHCI circuit of the host controller.<br>If this bit is set to 1b, the host controller initializes the internal pipelines, counters, and state machines, and communications on the USB immediately stop. At this time, port reset is not issued to the downstream ports.<br>Reset by this bit has no effect on the registers other than the EHCI Operational register. The EHCI Operational register is initialized, and the port owner returns to OHCI.<br>Software must be reset to return the host controller to the operational state.<br>When this reset processing finishes, the host controller sets this bit to 0b. Writing 0b to this bit cannot stop the reset processing. If the HCHalted bit of the USBSTS register is set to 0b, do not set 1b to this bit. |
| 0 | RS | 0 | R/W | Run/Stop<br>This bit is used to run or stop the EHCI host controller.<br>If this bit is set to 1b, the host controller starts operation.<br>As long as this bit is set to 1b, the host controller continues running.<br>If this bit is set to 0b, the host controller finishes the currently executing transaction and some other transactions, and then is changed to the Halt status. The HCHalted bit of the USBSTS register indicates that the host controller finished the transaction processing and entered into the stop status.<br>If the host controller is in a status other than Halt (the HCHalted bit of the USBSTS register is 1b), do not write 1b to this bit. |

### (2) USBSTS Register

Register abbreviation: USBSTS

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | Port-1 Change Detect |
| 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(1) |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Asynchronous Schedule Status | Periodic Schedule Status | Reclamation | HCHalted | — | — | — | — | — | — | Interrupt on Async Advance | Host System Error | Frame List Rollover | Port Change Detect | USBERRINT | USBINT |
| Initial Value | 0 | 0 | 0 | 1 | 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(1) | R/W(1) | R/W(1) | R/W(1) | R/W(1) | R/W(1) |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 17 | — | All 0 | R | Reserved<br>The write value should always be 0. |
| 16 | Port-1Change Detect | 0 | R/W (1) | If this bit is set to 1b, it indicates that a change in the port status was detected.<br>This bit is used only when the Per-Port Change Events Enable bit of the USBCMD register is set to 1b. |
| 15 | Asynchronous Schedule Status | 0 | R | This bit indicates the current status of the asynchronous schedule.<br>0b: The asynchronous schedule is disabled.<br>1b: The asynchronous schedule is enabled.<br>If this bit and bit 5 (Asynchronous Schedule Enable) of the USBCMD register have the same value, the asynchronous schedule is enabled (1b) or disabled (0b). |
| 14 | Periodic Schedule Status | 0 | R | This bit indicates the current status of the periodic schedule.<br>0b: The periodic schedule is disabled.<br>1b: The periodic schedule is enabled.<br>If this bit and bit 4 (Periodic Schedule Enable) of the USBCMD register have the same value, the periodic schedule is enabled (1b) or disabled (0b). |
| 13 | Reclamation | 0 | R | This bit is used to detect an empty asynchronous schedule.<br>If this bit is 1b, the asynchronous schedule is empty.<br>After reset or when a QH (H = 1) is fetched, the host controller clears this bit to 0b.<br>Also, when the host controller executes an asynchronous transaction or detects a start event, it sets this bit to 1b.<br>If this bit is 0b and a QH (H = 1) is fetched, the host controller is entered into the Async Sched Sleeping mode. |
| 12 | HCHalted | 1 | R | If the Run/Stop bit of the USBCMD register is 1b, this bit indicates 0b.<br>If the software or host controller sets the Run/Stop bit to 0b, the host controller stops operation, and sets 1b to this bit.<br>0b: The EHCI host controller is running.<br>1b: The EHCI host controller is stopped. |
| 11 to 6 | — | All 0 | R | Reserved<br>The write value should always be 0. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 5 | Interrupt on Async Advance | 0 | R/W (1) | This bit indicates the Async Advance Interrupt status.<br>After the host controller fetches the QH, it checks bit 6 (Interrupt on Async Advance Doorbell [IAAD] bit) of the USBCMD Register. If the IAAD bit is set to 1b, the host controller clears the IAAD bit after the QH processing normally finishes, and sets this bit.<br>If bit 5 (Interrupt on Async Advance Enable bit) of the UBINTR register is set to 1b, an interrupt due to this source will occur at the next interrupt timing after 1b is set to this bit.<br>If HCD writes 1b to this bit, this bit can be cleared. Writing 0b to this bit has no effect.<br>0b: Async Advance Interrupt not occurred.<br>1b: Async Advance Interrupt status is detected. |
| 4 | Host System Error | 0 | R/W (1) | This bit is set to 1b if a serious error occurs in the host controller.<br>If this error occurs, the host controller clears the Run/Stop bit of the USBCMD register to 0b so that the subsequent schedules are not executed.<br>Writing 1b to this bit can clear this bit. Writing 0b to this bit has no effect.<br>0b: No system error occurred.<br>1b: A system error occurred. |
| 3 | Frame List Rollover | 0 | R/W (1) | If a frame list rollover occurs, the host controller sets this bit to 1b.<br>Specifically, when the Frame Index field of the FRINDEX register is returned to H'000 from the maximum value (rollover), the host controller sets this bit to 1b. The maximum value (the value at which rollover occurs) depends on the Frame List Size field of the USBCMD register.<br>For example, if the Frame List Size is 1024 frame, rollover occurs every time FRIMDEX [13] toggles. If the Frame List Size is 512 frame, rollover occurs every time FRINDEX [12] toggles.<br>If the Frame List Size is 256 frame, rollover occurs every time FRINDEX [11] toggles.<br>Writing 1b to this bit can clears this bit. Writing 0b to this bit has no effect.<br>0b: The frame list is not returned to H'000.<br>1b: The frame list is returned to H'000 (rollover occurred). |
| 2 | Port Change Detect | 0 | R/W (1) | This bit indicates that the port status has changed.<br>Among the ports for which the Port Owner bit of the PORTSC1 register is set to 0b, if any port satisfies one of the following conditions, the host controller sets this bit to 1b:<br>• Connect or Disconnect status of a device is detected, and the Connect Status Change bit of the PORTSC1 register is changed from 0 to 1.<br>• A change of the Enable status of the port is detected, and the Port Enable/Disable Change bit of the PORTSC1 register is changed from 0 to 1.<br>• The overcurrent state is detected, and the Over-current Change bit of the PORTSC1 register is changed from 0 to 1.<br>• J-K transition is detected on a port in the Suspend status, and the Force Port Resume bit of the PORTSC1 register is changed from 0 to 1.<br>Writing 1b to this bit can clear this bit. Writing 0b to this bit has no effect. |
| 1 | USBERRINT | 0 | R/W (1) | USB Error Interrupt<br>This bit indicates that a USB transaction finished with an error.<br>When a USB transaction finishes with an error, the host controller sets this bit to 1b.<br>If 1b is set to the IOC bit of qTD at which an error interrupt occurred, 1b is set to both of this bit and USBINT bit.<br>Writing 1b to this bit can clear this bit. Writing 0b to this bit has no effect.<br>0b: USB transaction is normal.<br>1b: USB transaction finished with an error. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 0 | USBINT | 0 | R/W (1) | USB Interrupt<br>This bit indicates that a USB transfer has finished.<br>The host controller sets this bit to 1b if one of the following conditions is satisfied:<br>• A USB transfer has finished.<br>• A short packet is received.<br>Even if a USB transfer finished with an error, if IOC (Interrupt On Complete) of the TD is set to 1b, this bit is set to 1b.<br>Writing 1b to this bit can clear this bit. Writing 0b to this bit has no effect.<br>0b: A USB transfer has not finished.<br>1b: A USB transfer has finished. |

### (3) USBINTR Register

Register abbreviation: USBINTR

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | Port-1 Change Event Enable |
| 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 |

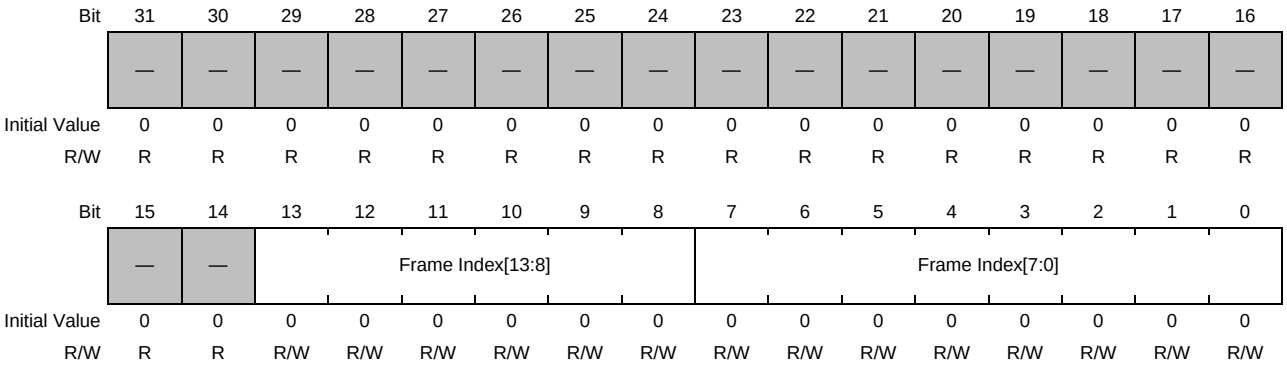
| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | Interrupt on Async Advance Enable | Host System Error Enable | Frame List Rollover Enable | Port Change Detect Enable | USB Error Interrupt Enable | USB Interrupt Enable |
| 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 |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 17 | — | All 0 | R | Reserved<br>The write value should always be 0. |
| 16 | Port-1 Change EventEnable | 0 | R/W | If this bit is 1b and the Port Change Detect bit of the USBSTS register is set to 1b, the host controller issues an interrupt. |
| 15 to 6 | — | All 0 | R | Reserved<br>The write value should always be 0. |
| 5 | Interrupt on Async Advance Enable | 0 | R/W | This bit sets whether bit 5 (Interrupt on Async Advance [IAA] bit) of the USBSTS register is enabled or disabled.<br>If the IAA bit is set to 1b while this bit is set to 1b, the host controller issues an interrupt at the next interrupt timing.<br>0b: Disabled.<br>1b: Enabled (an interrupt occurs via the IAA bit). |
| 4 | Host System Error Enable | 0 | R/W | This bit sets whether bit 4 (Host System Error [HSE] bit) of the USBSTS register is enabled or disabled. If the HSE bit is set to 1b while this bit is set to 1b, the host controller issues an interrupt at the next interrupt timing.<br>0b: Disabled.<br>1b: Enabled (an interrupt occurs via the HSE bit). |
| 3 | Frame List Rollover Enable | 0 | R/W | This bit sets whether the bit 3 (Frame List Rollover [FLR] bit) of the USBSTS register is enabled or disabled.<br>If the FLR bit is set to 1b while this bit is set to 1b, the host controller issues an interrupt at the next interrupt timing.<br>0b: Disabled.<br>1b: Enabled (an interrupt occurs via the FLR bit). |
| 2 | Port Change Detect Enable | 0 | R/W | This bit sets whether bit 2 (Port Change Detect [PCD] bit) of the USBSTS register is enabled or disabled.<br>If the USBERRINT bit is set to 1b while this bit is set to 1b, the host controller issues an interrupt at the next interrupt timing.<br>0b: Disabled.<br>1b: Enabled (an interrupt occurs via the PCD bit). |
| 1 | USB Error Interrupt Enable | 0 | R/W | This bit sets whether bit 1 (USBERRINT bit) of the USBSTS register is enabled or disabled. If the USBERRINT bit is set to 1b while this bit is set to 1b, the host controller issues an interrupt at the next interrupt timing.<br>0b: Disabled.<br>1b: Enabled (an interrupt occurs via the USBERRINT bit). |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 0 | USB Interrupt Enable | 0 | R/W | This bit sets whether bit 0 (IUSBINT bit) of the USBSTS register is enabled or disabled.<br>If the USBINT bit is set to 1b while this bit is set to 1b, the host controller issues an interrupt at the next interrupt timing.<br>0b: Disabled.<br>1b: Enabled (an interrupt occurs via the USBINT bit). |

### (4) FRINDEX Register

Register abbreviation: FRINDEX



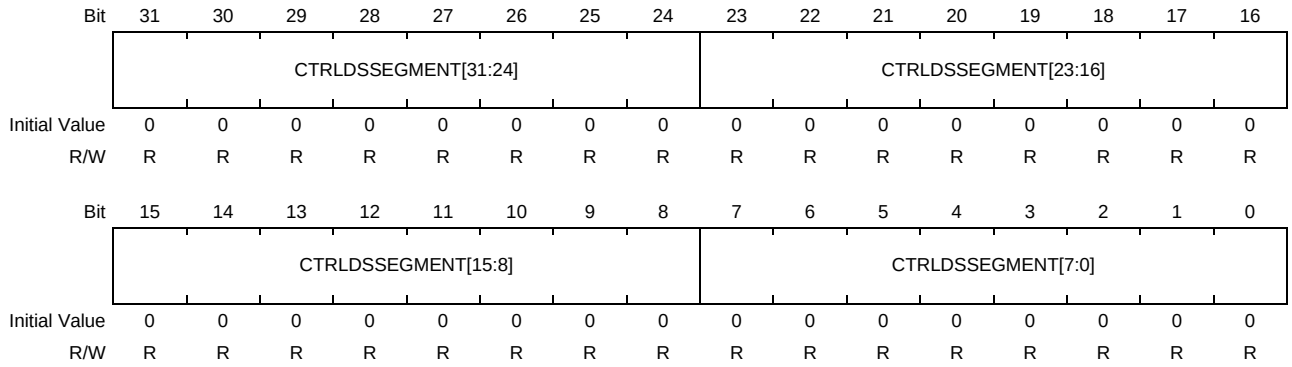

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 14 | — | All 0 | R | Reserved<br>The write value should always be 0. |
| 13 to 0 | Frame Index[13:0] | All 0 | R/W | This field is used by the host controller to add an index to the periodic frame list. The value in this field is incremented at the end of a micro frame.<br>Bit [N:3] of this field is used as the Frame List Current index. This means that, before the next index arrives, the current frame list is accessed 8 times. The value for N is determined, as follows, by the setting value of bit [3:2] (Frame List Size field) of the USBCMD register. |

| Frame List Size | Number of Frames | N |
|---|---|---|
| 00b | 1024 | 12 |
| 01b | 512 | 11 |
| 10b | 256 | 10 |
| 11b | 32 | 12 |

Access this register only when the host controller is in stop status (bit 12 [HCHalted] = 1b). The setting value of this field is applied to the SOF frame number of the SOF token.

### (5) CTRLDSSEGMENT Register

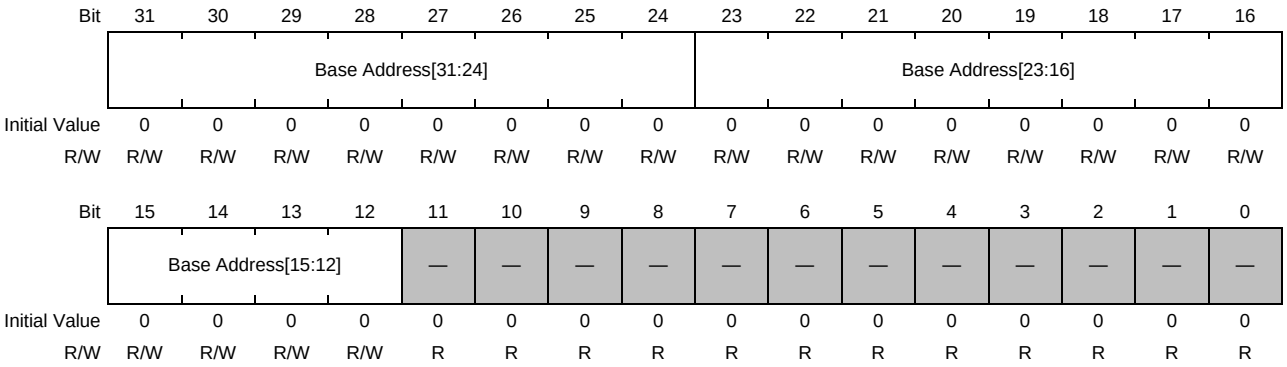
Register abbreviation: CTRLDSSEGMENT




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 0 | CTRLDSSEGMENT | All 0 | R | This register is not used because this module does not support 64-bit address method.<br>Therefore, HCD must not access this register. |

### (6) PERIODICLISTBASE Register

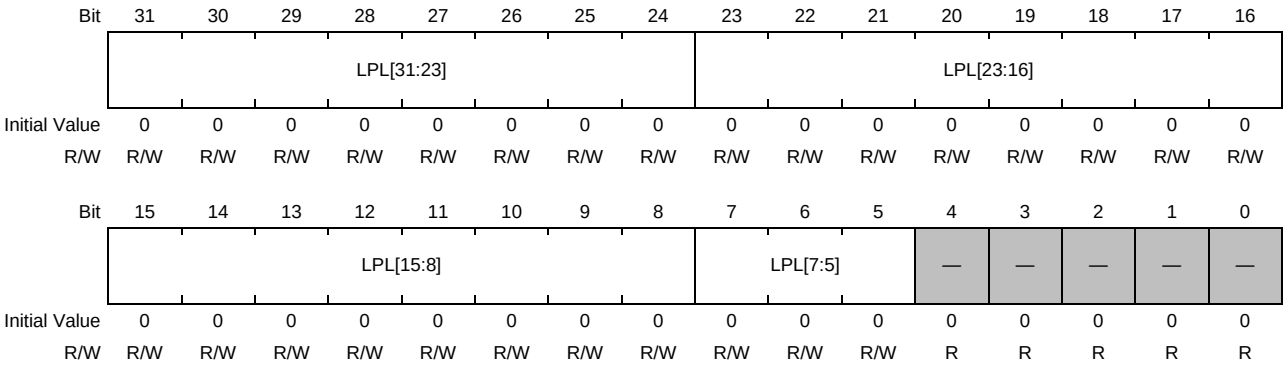
Register abbreviation: PERIODICLISTBASE




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 12 | Base Address | All 0 | R/W | This field indicates the head address of the periodic frame list on the system memory.<br>The host controller loads the contents of this register before starting the list processing.<br>The host controller determines the frame list to be processed by using this field and Frame Index of the FRINDEX register.<br>Align the address of the periodic frame list by 4 Kbyte.<br>Normal operation is not guaranteed if any of these bits are changed during operation. |
| 11 to 0 | — | All 0 | R | Reserved<br>The write value should always be 0. |

### (7) ASYNCLISTADDR Register

Register abbreviation: ASYNCLISTADDR




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 5 | LPL | All 0 | R/W | Link Pointer Low<br>This field indicates the address (on the system memory) of the Asynchronous Queue Head to be processed next time.<br>Align the address of Asynchronous Queue Head by 32 byte. |
| 4 to 0 | — | All 0 | R | Reserved<br>The write value should always be 0. |

### (8) CONFIGFLAG Register

Register abbreviation: CONFIGFLAG

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CF |
| 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<br>The write value should always be 0. |
| 0 | CF | 0 | R/W | Configuration Flag<br>This bit controls which of OHCI or EHCI is routed by the port routing control circuit by default.<br>At the end of the host controller configuration, this bit is set to 1b.<br>0b: The port routing control circuit routes each port to the OHCI host controller by default.<br>1b: The port routing control circuit routes each port to the EHCI host controller by default. |

### (9) PORTSC1Register

Register abbreviation: PORTSC1

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Device Address | | | | | | | Suspend Status[1] | Suspend Status[0] | WKOC_E | WKDSCNNT_E | WKCNNT_E | Port Test Control | | | |
| 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 | R | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | Port Owner | PP | Line Status | | Suspend using L1 | Port Reset | Suspend | Force Port Resume | Over current Active Change | Over-current Active | Port Enabled /Disabled Change | Port Enabled /Disabled | Connect Status Change | Current Connect Status |
| Initial Value | 0 | 0 | 1 | 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/W | R/W | R/W(1) | R | R/W(1) | R/W | R/W(1) | R |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 25 | Device Address | All 0 | R/W | This field indicates the USB device address (7 bits) of the device connected to the down port.<br>This address is used when the LPM Token is sent.<br>If the value of this filed is H'00, it means that no device that needs to use this field has been connected. |
| 24, 23 | Suspend Status | All 0 | R | This field indicates the response from the connected device to the LPM Token (L1 transition request).<br>00b: The device succeeded transition to the L1 state (ACK received from the device).<br>01b: The device has not changed to the L1 state (NYET received from the device).<br>10b: The device does not support the L1 state transition (STALL received from the device).<br>11b: Other response (for example, Timeout error)<br>Change this field only when the Suspend bit is 0b. |
| 22 | WKOC_E | 0 | R/W | Wake on Overcurrent Enable<br>By writing 1b to this bit, the overcurrent state can be detected as an EHCI Wakeup event.<br>If bit 12 (PP [Port Power] bit) is 0b, this bit becomes 0b. |
| 21 | WKDSCNNT_E | 0 | R/W | Wake on Disconnect Enable<br>By writing 1b to this bit, device disconnection can be detected as an EHCI Wakeup event.<br>If bit 12 (PP [Port Power] bit) is 0b, this bit becomes 0b. |
| 20 | WKCNNT_E | 0 | R/W | Wake on Connect Enable<br>By writing 1b to this bit, device connection can be detected as an EHCI Wakeup event.<br>If bit 12 (PP [Port Power] bit) is 0b, this bit becomes 0b. |

<table>
<tr><th>Bit</th><th>Bit Name</th><th>Initial Value</th><th>R/W</th><th>Description</th></tr>
<tr><td>19 to 16</td><td>Port Test Control</td><td>All 0</td><td>R/W</td><td>This field is controlled by the test mode.<br>If the value of this field is other than 0000b, it indicates that this module is running in the test mode.
<table>
<tr><th>Value</th><th>Test mode</th></tr>
<tr><td>0000b</td><td>Test mode not enabled</td></tr>
<tr><td>0001b</td><td>Test J_STATE</td></tr>
<tr><td>0010b</td><td>Test K_STATE</td></tr>
<tr><td>0011b</td><td>Test SE0_NAK</td></tr>
<tr><td>0100b</td><td>Test Packet</td></tr>
<tr><td>0101b</td><td>Test FORCE_ENABLE</td></tr>
<tr><td>Other</td><td>Reserved</td></tr>
</table></td></tr>
<tr><td>15, 14</td><td>—</td><td>All 0</td><td>R</td><td>Reserved<br>The write value should always be 0.</td></tr>
<tr><td>13</td><td>Port Owner</td><td>1</td><td>R/W</td><td>This bit indicates which of OHCI or EHCI has the port ownership.<br>0b: EHCI has the port ownership.<br>1b: OHCI has the port ownership.<br>When bit 0 (Configure Flag bit) of the CONFIGFLA register is changed from 0b to 1b, this bit becomes 0b. If bit 0 (Configure Flag bit) of the CONFIGFLA register is 0b, this bit becomes 1b.<br>If the connected device is not a High Speed device, this bit is set to 1b to transfer the port ownership to OHCI.</td></tr>
<tr><td>12</td><td>PP</td><td>0</td><td>R/W</td><td>Port Power<br>This bit controls power supply to the port.<br>If this bit is 0b, power is not supplied to the port. Therefore, the port does not function, and does not recognize connection and disconnection.<br>If overcurrent is detected while this bit is set to 1b, the host controller clears this bit to 0b, and the power supplied to the port is stopped.<br>Note: As described later, if the PPC bit is 0b, this bit is fixed to 1b, so the power supplied to the port is not stopped.<br>The function of this bit differs depending on the value of bit 4 (PPC [Port Power Control] bit) of the HCSPARAMS register.
<table>
<tr><th>PPC</th><th>PP</th><th colspan="2">Operation</th></tr>
<tr><td>0b</td><td>1b</td><td>R</td><td colspan="2">This bit is fixed to 1b, and the power is always supplied to the port.</td></tr>
<tr><td>1b</td><td>0b/1b</td><td>R/W</td><td colspan="2">Whether or not power is supplied to the port depends on the setting of this bit.</td></tr>
<tr><td></td><td></td><td></td><td>0b</td><td>Power is not supplied to the port.</td></tr>
<tr><td></td><td></td><td></td><td>1b</td><td>Power is supplied to the port.</td></tr>
</table></td></tr>
<tr><td>11, 10</td><td>Line Status</td><td>All 0</td><td>R</td><td>This field indicates the logical level of D+/D- lines of the current USB bus. (bit 11: DP / bit 10: DM)<br>This field is used to detect an LS device before starting a sequence for port reset and port enable.<br>Therefore, this bit is valid only when bit 3 (Port Enable/Disable bit) is 0b and bit 0 (Current Connect Status bit) is 1b.
<table>
<tr><th>bit 11</th><th>bit 10</th><th>USB bus status</th><th>Description</th></tr>
<tr><td>0b</td><td>0b</td><td>SE0</td><td>The device is not an LS device.</td></tr>
<tr><td></td><td></td><td></td><td>EHCI port reset is executing.</td></tr>
<tr><td>1b</td><td>0b</td><td>J-state</td><td>The device is not an LS device.</td></tr>
<tr><td></td><td></td><td></td><td>EHCI port reset is executing.</td></tr>
<tr><td>0b</td><td>1b</td><td>K-state</td><td>An LS device was connected.</td></tr>
<tr><td></td><td></td><td></td><td>The port ownership is transferred from EHCI to OHCI.</td></tr>
<tr><td>1b</td><td>1b</td><td>Undefined</td><td>The device is not an LS device.</td></tr>
<tr><td></td><td></td><td></td><td>EHCI port reset is executing.</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Bit</th><th>Bit Name</th><th>Initial Value</th><th>R/W</th><th>Description</th></tr>
<tr><td>9</td><td>Suspend using L1</td><td>0</td><td>R/W</td><td>Suspend using L1 (LPM) control bit.<br>If 1b is written to the Suspend bit (bit 7) while this bit is 1b, the state of this module is changed to the LPM state.<br>Writing to this bit is possible only when the Suspend bit (bit 7) is 0b.<br>0b: Suspend using L2<br>1b: Suspend using L1 (LPM)<br>If this bit is 1b and the Device Address field is other than H’0000, when the Suspend bit is set to 1b, the host controller generates an LPM Token to change the state to the L1 state.<br>If this bit is 0b, the host controller operates as L2 Suspend.</td></tr>
<tr><td>8</td><td>Port Reset</td><td>0</td><td>R/W</td><td>This bit indicates the reset status of the port.<br>0b: The port is not being reset.<br>1b: The port is being reset.<br>If 1b is written to this bit while this bit is 0b, the bus reset sequence defined in USB 2.0 standard starts. To finish the bus reset sequence, 0b must be written to this bit. Note that this bit must remain 1b long time enough to guarantee that the bus reset sequence defined in USB 2.0 standard will be complete.<br>If bit 12 (HCHalted) of the USBSTS register is 1b, do not set this bit to 1b.<br>If any of the PP (Port Power) bit, Port Owner bit, and Current Connect Status bit is in the following status, this bit becomes 0b.<br>Note: Even if 1b is written to this bit, the bus reset sequence does not start.<br>• PP (Port Power) bit = 0b<br>• Port Owner bit = 1b<br>• Current Connect Status bit = 0b</td></tr>
<tr><td>7</td><td>Suspend</td><td>0</td><td>R/W</td><td>This bit indicates the Suspend control and status of the port.<br>This bit and bit 2 (Port Enabled/Disabled bit) indicate the following port status.
<table>
<tr><th>Port Enabled</th><th>Suspend</th><th>Port Status</th></tr>
<tr><td>0b</td><td>—</td><td>Disable</td></tr>
<tr><td>1b</td><td>0b</td><td>Enable</td></tr>
<tr><td></td><td>1b</td><td>Suspend</td></tr>
</table>
To change the port state to L1 or L2 Suspend, set this bit to 1b.<br>Whether the host controller supports L1 Suspend state or L2 Suspend state depends on the value in the Suspend Using L1 bit.<br>In the Suspend state, data transfer to the downstream port is blocked by this port (except for port reset). If this bit is set to 1b during data transfer, blocking of transfer data does not occur until the current data transfer finishes.<br>Writing 0b to this bit has no effect.<br>This bit can be set to 1b only when all the following conditions are satisfied: “PP (Port Power) bit = 1b”, “Port Owner bit = 0b”, and “Current Connect Status bit = 1b”. If any of the following conditions is satisfied, the host controller clears this bit:<br>• “Resume is complete” is detected.<br>• The PR bit is set to 1b when PR (Port Reset) bit is 0b.<br>• The port owner is 1b (OHCI).<br>• The PP (Port Power) bit is set to 0b.<br>• The Port Enabled/Disabled bit is set to 0b.</td></tr>
</table>

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 6 | Force Port Resume | 0 | R/W | This bit indicates that the Resume state of the port is detected.<br>0b: Resume signal is not detected or output.<br>1b: Resume (K-state) is detected or output.<br>When the port is in Suspend state, if the host controller detects transition of the state from J to K (if RemoteWakeup is detected from the connected device), it sets this bit to 1b. The host controller also sets the Port Change Detect bit or Port-1 Changes Detect bit of the USBSTS register to 1b.<br>If this bit is set to 1b, the host controller does not set 1b to the Port Change Detect bit and Port-1 Changes Detect bit of the USBSTS register. While this bit is 1b, the Resume signal (FS K) is driven onto the USB bus.<br>For L2 transition, this bit must be cleared to 0b after an appropriate time has passed. By writing 0b to this bit while this bit is 1b, the port status is recovered to be the HS Idle status. This bit remains 1b until the port is recovered. The host controller must finish transition to the HS Idle state within 2msec since this bit is cleared to 0b.<br>On the other hand, for L1 transition, the host controller sends a Resume signal at necessary timing, and this bit is cleared to 0b at Resume recovery. Note that the software sets the length of the Resume signal driven by the host controller, by using the Host-Initiated Resume Duration field of the USBCMD register.<br>If the Port Power (PP) bit is 0b, this bit becomes 0b. |
| 5 | Over-current Change | 0 | R/W (1) | This bit indicates that bit 4 (Over-current Active bit) has changed.<br>Writing 1b to this bit can clear this bit. Writing 0b to this bit has no effect.<br>0b: Over-current Active bit has not changed.<br>1b: Over-current Active bit has changed. |
| 4 | Over-current Active | 0 | R | This bit indicates the overcurrent status of the port.<br>If the host controller detects overcurrent, it disables the port, and set this bit to 1b.<br>After the overcurrent state is released, the host controller automatically clears this bit from 1b to 0b.<br>0b: The port is not in overcurrent state.<br>1b: The port is in overcurrent state. |
| 3 | Port Enable/ Disable Change | 0 | R/W (1) | This bit indicates that the host controller detected frame babble.<br>If the Host Controller detects frame babble, it disables the port and sets this bit to 1b.<br>Writing 1b to this bit can clear this bit. Writing 0b to this bit has no effect. If the Port Power (PP) bit is 0b, this bit becomes 0b.<br>0b: Frame babble has not occurred.<br>1b: Frame babble is detected. |
| 2 | Port Enabled/ Disabled | 0 | R/W | This bit indicates the Enable/Disable status of the port.<br>The host controller resets the port, and enables the port if the connected device is recognized as an HS device, and sets this bit to 1b. The software cannot set this bit to 1b.<br>If the host controller detects disconnection of a device or other errors, it disables the port, and clears this bit to 0b. The port also becomes disabled when 0b is written to this bit.<br>If the port is disabled, data transfer to the downstream port is blocked except for port reset.<br>If Port Test Control [3:0] = 0101b (Test FORCE_ENABLE), the port becomes enabled, and this bit is set to 1b.<br>If the Port Power (PP) bit is 0b, this bit becomes 0b.<br>0b: The port is disabled.<br>1b: The port is enabled. |
| 1 | Connect Status Change | 0 | R/W (1) | This bit indicates that bit 0 (Current Connect Status bit) has changed.<br>Writing 1b to this bit can clear this bit. Writing 0b to this bit has no effect.<br>If the Port Power (PP) bit is 0b, this bit becomes 0b.<br>0b: The Current Connect Status bit has no change.<br>1b: The Current Connect Status bit has changed. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 0 | Current Connect Status | 0 | R | This bit indicates the connection status of the port.<br>If the host controller detects connection of a device, it sets this bit to 1b. Also, if Port Test Control [3:0] = 0101b (Test FORCE_ENABLE), the Host Controller sets this bit to 1b even if no device is connected.<br>On the other hand, if the host controller detects disconnection of a device, it sets this bit to 0b.<br>If the Port Power (PP) bit is 0b, or the Port Owner (PO) bit is 0b, this bit becomes 0b.<br>0b: No device is connected to the port.<br>1b: A device is connected to the port. |

### 32A.2.4.4 AHB Bridge Register

#### (1) INT_ENABLE Register

Abbreviated name of register: INT_ENABLE

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | WAKEON_INTEN | UCOM_INTEN | USBH_INTBEN | USBH_INTAEN | AHB_INTEN |
| 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 |
|---|---|---|---|---|
| 31 to 5 | — | All 0 | R | Reserved<br>The write value should always be 0. |
| 4 | WAKEON_INTEN | 0 | R/W | This bit enables or disables bit 4 (WAKEON_INT) of the INT_STATUS register.<br>0b: disable<br>1b: enable |
| 3 | UCOM_INTEN | 0 | R/W | This bit enables or disables bit 3 (UCOM_INT) of the INT_STATUS register.<br>0b: disable<br>1b: enable |
| 2 | USBH_INTBEN | 0 | R/W | This bit enables or disables bit 2 (USBH_INTB) of the INT_STATUS register.<br>0b: disable<br>1b: enable |
| 1 | USBH_INTAEN | 0 | R/W | This bit enables or disables bit 1 (USBH_INTA) of the INT_STATUS register.<br>0b: disable<br>1b: enable |
| 0 | AHB_INTEN | 0 | R/W | This bit enables or disables bit 0 (AHB_INT) of the INT_STATUS register.<br>0b: disable<br>1b: enable |

### (2) INT_STATUS Register

Abbreviated name of register: INT_STATUS

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | WAKEON_INT | UCOM_INT | USBH_INTB | USBH_INTA | AHB_INT |
| 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(1) | R | R | R | R/W(1) |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 5 | — | All 0 | R | Reserved<br>The write value should always be 0. |
| 4 | WAKEON_INT | 0 | R/W (1) | This bit indicates the state of WAKEON interrupt from the HOST module.<br>Writing 1b to this bit can clear this bit.<br>0b: No WAKEON interrupt<br>1b: WAKEON interrupt |
| 3 | UCOM_INT | 0 | R | This bit indicates the state of interrupt from the UCOM register.<br>To clear the interrupt, use the UCOM2 Register.<br>0b: No UCOM register interrupt<br>1b: UCOM register interrupt |
| 2 | USBH_INTB | 0 | R | This bit indicates the state of EHCI interrupt.<br>To clear the interrupt, use the USBSTS Register (of the EHCI Operational Register).<br>0b: No INTB interrupt<br>1b: INTB interrupt |
| 1 | USBH_INTA | 0 | R | This bit indicates the state of OHCI interrupt.<br>To clear the interrupt, use the HcInterruptStatus Register (of the OHCI Operational Register).<br>0b: No INTA interrupt<br>1b: INTA interrupt |
| 0 | AHB_INT | 0 | R/W (1) | This bit indicates that a BUS Master error occurred.<br>Writing 1b to this bit can clear this bit.<br>0b: No bus error occurred.<br>1b: A bus error occurred. |

### (3) AHB_BUS_CTR Register

Abbreviated name of register: AHB_BUS_CTR

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PROT_TYPE | | | | — | — | — | PROT_MODE | — | — | ALIGN_ADDRESS | | — | — | MAX_BURST_LEN | |
| 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/W | R | R | R/W | R/W | R | R | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 16 | — | All 0 | R | Reserved<br>The write value should always be 0. |
| 15 to 12 | PROT_TYPE | All 0 | R/W | This field sets MHPROT [3:0] used when the BUS Master interface issues a transfer request.<br>bit 15 0b: Cache disabled.<br>1b: Cache enabled.<br>bit 14 0b: Buffer disabled.<br>1b: Buffer enabled.<br>bit 13 0b: User access<br>1b: Privileged access<br>bit 12 0b: Operation code<br>1b: Data |
| 11 to 9 | — | All 0 | R | Reserved<br>The write value should always be 0. |
| 8 | PROT_MODE | 0 | R/W | This bit selects the mode of MHPROT [3:0] used when the Master interface issues a transfer request.<br>0b: The value of PROT_TYPE is output as MHPROT [3:0].<br>1b: When a DMA transfer is performed, MHPROT [3:0] is set to 0000b if the final burst is performed, or MHPROT [3:0] is set to the PROT_TYPE value if another burst transfer is performed. |
| 7, 6 | — | All 0 | R | Reserved<br>The write value should always be 0. |
| 5, 4 | ALIGN_ADDRESS | All 0 | R/W | This field sets the address boundary used when the BUS Master interface issues a burst transfer.<br>00b: A burst transfer is issued so that not to exceed 1-Kbyte boundary.<br>01b: A burst transfer is issued so that not to exceed 64-byte boundary.<br>10b: A burst transfer is issued so that not to exceed 32-byte boundary. (The maximum burst length is INCR8. This is because, if INCR16 is used, 32-byte boundary is exceeded.)<br>11b: A burst transfer is issued so that not to exceed16 byte boundary. (The maximum burst length is INCR4. This is because, if the length is at least INCR8, 16-byte boundary is exceeded.) |
| 3,2 | — | All 0 | R | Reserved<br>The write value should always be 0. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 1, 0 | MAX_BURST_LEN | All 0 | R/W | This field selects the maximum burst length used when the BUS Master interface issues a transfer request.<br>00b: INCR16<br>01b: INCR8<br>10b: INCR4<br>11b: SINGLE |

### (4) USBCTR Register

Abbreviated name of register: USBCTR

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | DIRPD | PLL_RST | USBH_RST |
| 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/W | R/W | W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 3 | — | All 0 | R | Reserved<br>The write value should always be 0. |
| 2 | DIRPD | 0 | R/W | USBPHY Standby Mode Control<br>0b: USBPHY normal operating mode<br>1b: USBPHY standby mode<br>*Note*: When the USB module is not in use, setting this bit to 1 reduces power consumption by the USBPHY module.<br>Only set it to 1 when the USB module is not in use.<br>In transitions from USBPHY standby mode to USBPHY normal operating mode, assert the reset signal for the USBPHY module for at least 1 µs before the transition. |
| 1 | PLL_RST | 1 | R/W | This bit controls resetting of the USBPHY module.<br>0b: The USBPHY reset is released.<br>1b: The USBPHY module is reset. |
| 0 | USBH_RST | 0 | W | Software reset to this module.<br>Setting this bit to 1b resets this module entirely.<br>This bit is always read as 0b.<br>*Note*: Set this bit only when the BUS Master interface of this module is not running.<br>Access to this module becomes valid 10 CLK (internal bus clock [P1ϕ]) after this bit is written.<br>0b: Nothing occurs.<br>1b: Reset is issued to this module. |

### (5) Register Enable/Clock Gating Control Register

Abbreviated name of register: REGEN_CG_CTRL

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | NONUSE_CLK_MSK | — | HOST_CLK_MSK | PERI_CLK_MSK | — | — | — | — | — | — | — | — | — | — | — | — |
| 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 | 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 | NONUSE_CLK_MSK | 0 | R/W | This bit is used to mask the clock for the unused host or peripheral controller, depending on the setting value of the OTG_PERI bit of the COMMCTRL register.<br>0b: Do not mask the clock.<br>1b: Mask the clock.<br>For details, see **Section 32A.3.1.2, Specifications of NONUSE_CLK_MSK operation**. |
| 30 | — | 0 | R | Reserved<br>The write value should always be 0. |
| 29 | HOST_CLK_MSK | 0 | R/W | This bit is used to forcibly mask clock supply to the host controller.<br>0b: Do not mask the clock supply.<br>1b: Mask the clock supply.<br>For details, see **Section 32A.3.1.3, Specifications of PERI_CLK_MSK and HOST_CLK_MSK operations**. |
| 28 | PERI_CLK_MSK | 0 | R/W | This bit is used to forcibly mask the clock supply to the peripheral controller.<br>0b: Do not mask the clock supply.<br>1b: Mask the clock supply.<br>For details, see **Section 32A.3.1.3, Specifications of PERI_CLK_MSK and HOST_CLK_MSK operations**. |
| 27 to 24 | — | All 0 | R | Reserved<br>The write value should always be 0. |
| 23 to 0 | — | All 0 | R | Reserved<br>The write value should always be 0. |

### (6) Suspend Control Register

Abbreviated name of register: SPD_CTRL

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | SUSPENDM_ENABLE | SLEEPM_ENABLE | — | — | — | — | — | — | WKCNNT_ENABLE | — | — | — | — | — | — | — |
| 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 | R | R | R | R | R | R |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | GLOBAL_SUSPENDM_P1 |
| 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 | SUSPENDM_ENABLE | 0 | R/W | The setting of this bit is valid only in the Host mode (when the OTG_PERI bit of the COMMCTRL register is 0b).<br>This bit is used to place USBPHY into the suspend status (the state in which PHY built-in PLL is stopped) when this module is changed to the USB Suspend mode.<br>If this bit is set to 1b, the Suspend related bits of the OHCI/EHCI Operational registers below are set. If this module is changed to the USB Suspend mode, USBPHY is placed into the suspend status.<br>[This function is valid for the following OHCI/EHCI Operational registers]<br>EHCI: bit [7] (Suspend bit) of the PORTSC1 register<br>OHCI: bit [2] (PSS bit) of the HcRhPortStatus register<br>OHCI: bit [7:6] (HCFS field) of the HcControl register |
| 30 | SLEEPM_ENABLE | 0 | R/W | The setting of this bit is valid only in the Host mode (when the OTG_PERI bit of the COMMCTRL register is 0b).<br>This bit is used to place USBPHY into the sleep status when the LPM function is used to change to the L1 Suspend mode. (In the sleep status, the PHY built-in PLL is running, but the 60MHz clock from USBPHY is gated.)<br>This bit is valid when "Suspend using L1" of the POTSC register is 1b. If this bit is set to 1b, an L1 transition request is issued to the device. If the request is accepted, USBPHY is placed into the sleep status after the following EHCI register bit is set to 1b.<br>[This function is valid for the following OHCI/EHCI Operational register]<br>EHCI: bit [7] (Suspend bit) of the PORTSC1 register |
| 29 to 24 | — | All 0 | R | Reserved<br>The write value should always be 0. |
| 23 | WKCNNT_ENABLE | 0 | R/W | The setting of this bit is valid only in the Host mode (the OTG_PERI bit of the COMMCTRL register is 0b).<br>If this bit is set to 1b, when a device disconnect occurs while USBPHY is in the suspend or sleep status, the suspend or sleep status is released.<br>This bit is valid only when the SUSPENDM_ENABLE bit (bit 31) or SLEEP_ENABLE bit (bit 30) is 1b.<br>*Note*: To set 1b to the SUSPENDM_ENABLE bit (bit 31) or SLEEPM_ENABLE bit (bit 30), as the general rule, set this bit to 1b.<br>If, in the above case, this bit is not set to 1b, the suspend or sleep status cannot be released even if a device disconnect occurs. |
| 22 to 1 | — | All 0 | R | Reserved<br>The write value should always be 0. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 0 | GLOBAL_SUSPENDM_P1 | 0 | R/W | The setting of this bit is valid regardless of the value of the OTG_PERI bit of the COMMCTRL register.<br>This bit is used to forcibly place USBPHY into the suspend status (in which the PHY built-in PLL is stopped).<br>If this bit is set to 1b, USBPHY is placed into the suspend status, regardless of the operating status and port status of the host controller.<br>*Note*:<br>– Do not set this bit to 1b during data transfer. We recommend that you set this bit to 1b after “stopping the EHCI/OHCI list processing” and “placing the port in the Disable status”.<br>– If the SUSPENDM_ENABLE bit (bit 31) is 1b, do not set this bit to 1b. |

### (7) Suspend/Resume Timer Setting Register

Abbreviated name of register: SPD_RSM_TIMSET

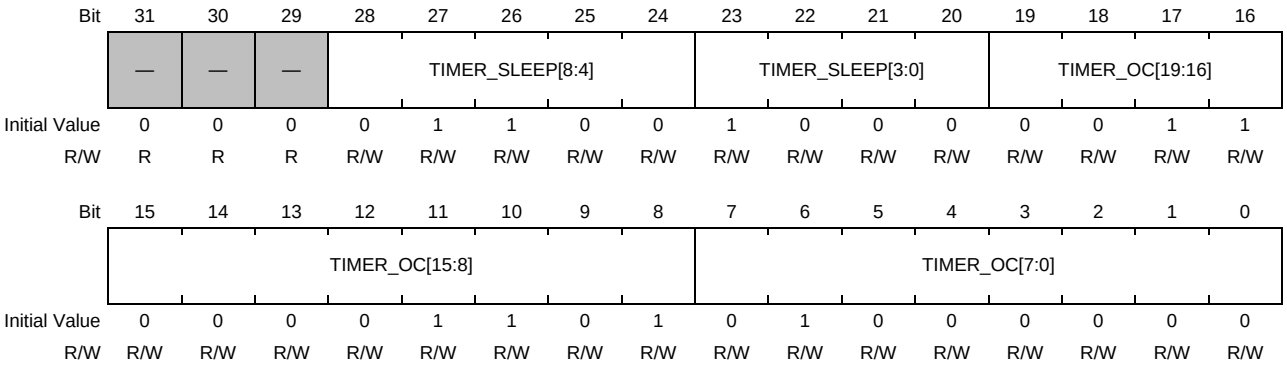
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TIMER_CONNECT[15:8] | | | | | | | | TIMER_CONNECT[7:0] | | | | | | | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TIMER_RESUME[15:8] | | | | | | | | TIMER_RESUME[7:0] | | | | | | | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 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 | TIMER_CONNECT[15:0] | H'01F4 | R/W | This field indicates the timer value used by USBPHY to detect Device Connect/Disconnect of the device in the suspend status (in which the PHY built-in PLL is stopped) when the SUSPENDM_ENABLE bit of the SPD_CTRL register is set to 1b.<br>When USBPHY is in suspend status, whether Connect/Disconnect occurs is judged by the internal bus clock (P1ϕ).<br>According to the internal bus clock (P1ϕ) frequency, specify the setting so that this timer value becomes at least 2.5 µs.<br>1 bit = 1 cycle (µs)<br>(Setting guideline)<br>For 100 MHz: At least H'FA |
| 15 to 0 | TIMER_RESUME[15:0] | H'03E8 | R/W | This field indicates the timer value used by USBPHY to detect RemoteWakeup signal from the device in suspend status (in which the PHY built-in PLL is stopped) when the SUSPENDM_ENABLE bit of the SPD_CTRL register is set to 1b.<br>When USBPHY is in suspend status, whether the RemoteWakeup signal or not is judged by the internal bus clock (P1ϕ).<br>According to the internal bus clock (P1ϕ) frequency, specify the setting so that this timer value becomes at least 5 µs.<br>1 bit = 1 cycle (µs)<br>(Setting guideline)<br>For 100 MHz: At least H'1F4 |

### (8) Overcurrent Detection/Sleep Timer Setting Register

Abbreviated name of register: OC_SLP_TIMSET




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 29 | — | All 0 | R | Reserved<br>The write value should always be 0. |
| 28 to 20 | TIMER_SLEEP[8:0] | H'0C8 | R/W | This field indicates the timer value used by USBPHY when the SLEEPM_ENABLE bit of the SPD_CTRL register is set to 1b. This timer value is used to measure the time for detecting RemoteWakeup reception and Resume-K drive time during sleep status (in which the PHY built-in PLL is running, but 60MHz clock from the USBPHY is gated).<br>When USBPHY is in sleep status, whether the RemoteWakeup signal or not is judged by the internal bus clock (P1ϕ).<br>According to the internal bus clock (P1ϕ) frequency, specify the setting so that this timer value becomes 1 µs.<br>1 bit = 1 cycle (µs)<br>(Setting guideline)<br>For 100 MHz: H'064 |
| 19 to 0 | TIMER_OC[19:0] | H'30D40 | R/W | This field indicates the timer value used for overcurrent detection.<br>If the overcurrent input (OVRCUR) set in this field is continuously asserted (0b) for the duration set in this register, this module determines that overcurrent occurred.<br>According to the internal bus clock (P1ϕ) frequency, specify the setting so that this timer value becomes at least 1 ms. 1 bit is 1 cycle (µs).<br>(Setting guideline)<br>For 100 MHz and 1 ms: At least H'1_86A0 |

### (9) SBRN_FLADJ_PORTWAKECAP Register

Abbreviated name of register: SBRN_FLADJ_PW

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PORTWAKECAP[15:8] | | | | | | | | PORTWAKECAP[7:0] | | | | | | | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 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 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | FLADJ[7:0] | | | | | | | | SBRN[7:0] | | | | | | | |
| Initial Value | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 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 | R | R |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 16 | PORTWAKECAP[15:0] | H'0003 | R/W | This field is used to mask the ports (of the connected device) that are used for Wakeup event.<br>Operation on this field has no effect on the operation of the HOST module. |
| 15 to 8 | FLADJ[7:0] | H'20 | R,<br>R/W | This field adjusts the length of one micro frame by 16HS bit time unit.<br>The initial value indicates H'20 (60000d HS bit time). |
| 7 to 0 | SBRN[7:0] | H'20 | R | This field indicates the Serial Bus Release Number.<br>The fixed value "H'20" is indicated. |

### (10) PORT_LPM_CTR1 Register

Abbreviated name of register: PORT_LPM_CTRL1

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | NYET_RETRY_CNT_P1[3:0] | | | | REMOTEWAKE_EN_P1 | SLEEP_INT_EN_P1 | RETRY_ENABLE_NYET_P1 | HIRD_SEL_P1 |
| 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<br>The write value should always be 0. |
| 7 to 4 | NYET_RETRY_CNT_P1[3:0] | All 0 | R/W | This field sets the number of retries that are allowed when the response from the device in the LPM transaction was NYET. The setting value of this bit is valid if the RETRY_ENABLE_NYET_P1 bit (bit 1) is 1b.<br>0000b: No retry.<br>0001b to 1111b: Retries the set number of times. (MAX: 15 retries) |
| 3 | REMOTEWAKE_EN_P1 | 0 | R/W | This bit is used to indicate the value of the RemoteWakeup bit of the LPM Token.<br>0b: RemoteWakeup is supported.<br>1b: RemoteWakeup is not supported. |
| 2 | SLEEP_INT_EN_P1 | 0 | R/W | This bit is used to enable an interrupt to occur when a status other than ACK is received in the LPM transaction.<br>The Per-Port Change interrupt can occur.<br>0b: No interrupt occurs.<br>1b: An interrupt occurs. |
| 1 | RETRY_ENABLE_NYET_P1 | 0 | R/W | This bit is used to set the behavior of the host controller when a response from the device is NYET in an LPM transaction.<br>0b: No retry.<br>1b: Retries are made. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 0 | HIRD_SEL_P1 | 0 | R/W | This bit sets the time for K drive for when recovered from the Sleep state.<br>Based on the setting values of this bit and EHCI USBCMD Register bit [27:24], the K drive time is determined as follows. |

| USBCMD Register | HIRD_SEL_P1 (Setting value of this bit) | |
|---|---|---|
| bit [27:24] | 0b | 1 |
| 0000b | 75 µs | 50 µs |
| 0001b | 100 µs | 125 µs |
| 0010b | 150 µs | 200 µs |
| 0011b | 250 µs | 275 µs |
| 0100b | 350 µs | 350 µs |
| 0101b | 450 µs | 425 µs |
| 0110b | 950 µs | 500 µs |
| 0111b | 1950 µs | 575 µs |
| 1000b | 2950 µs | 650 µs |
| 1001b | 3950 µs | 725 µs |
| 1010b | 4950 µs | 800 µs |
| 1011b | 5950 µs | 875 µs |
| 1100b | 6950 µs | 950 µs |
| 1101b | 7950 µs | 1025 µs |
| 1110b | 8950 µs | 1100 µs |
| 1111b | 9950 µs | 1175 µs |

### 32A.2.4.5 UCOM Register

#### (1) Common Control Register

Abbreviated name of register: COMMCTRL

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | OTG_PERI | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Initial Value | 1 | 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 | 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 | OTG_PERI | 1 | R/W | This bit specifies whether this module is set to the host mode or function mode.<br>0b: Host mode<br>1b: Function mode |
| 30 to 0 | — | All 0 | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. |

### (2) OTG-BC Interrupt Status Register

Abbreviated name of register: OBINTSTA

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | DPMONCHG_STA | DMMONCHG_STA |
| 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(1) | R/W(1) |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | CHGDETCHG1_STA | — | PDDETCHG1_STA | — | — | OCINT_STA | IDCHG_STA |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| R/W | R | R | R | R | R | R | R | R | R | R/W(1) | R | R/W(1) | R | R | R/W(1) | R/W(1) |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 18 | — | All 0 | R | Reserved The write value should always be 0. |
| 17 | DPMONCHG_STA | 0 | R/W (1) | This bit is set if the DPMON bit of the LINECTRL1 register has changed.<br>0b: The DPMON bit of the LINECTRL1 register has not changed.<br>1b: The DPMON bit of the LINECTRL1 register has changed. |
| 16 | DMMONCHG_STA | 0 | R/W (1) | This bit is set if the DMMON bit of the LINECTRL1 register has changed.<br>0b: The DMMON bit of the LINECTRL1 register has not changed.<br>1b: The DMMON bit of the LINECTRL1 register has changed. |
| 15 to 7 | — | All 0 | R | Reserved<br>The write value should always be 0. |
| 6 | CHGDETCHG1_STA | 0 | R/W (1) | This bit is set if the CHGDETSTS bit of the BCCTRL1 register has changed from 0b to 1b.<br>0b: The CHGDETSTS bit of the BCCTRL1 register has not changed from 0b to 1b<br>1b: The CHGDETSTS bit of the BCCTRL1 register has changed from 0b to 1b. |
| 5 | — | 0 | R | Reserved<br>The write value should always be 0. |
| 4 | PDDETCHG1_STA | 0 | R/W (1) | This bit is set if the PDDETSTS bit of the BCCTRL1 register has changed from 0b to 1b.<br>0b: The PDDETSTS bit of the BCCTRL1 register has not changed from 0b to 1b.<br>1b: The PDDETSTS bit of the BCCTRL1 register has changed from 0b to 1b. |
| 3 | — | 0 | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value.<br>Operation is not guaranteed if a value other than the initial value is written. |
| 2 | — | 0 | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value.<br>Operation is not guaranteed if a value other than the initial value is written. |
| 1 | OCINT_STA | 0 | R/W (1) | This bit is set if the OVRCUR pin is asserted.<br>0b: The OVRCUR pin is not asserted (and remains 1b).<br>1b: The OVRCUR pin is asserted (and became 0b). |
| 0 | IDCHG_STA | 1 | R/W (1) | This bit is set if the input value from the OTG_ID pin has changed.<br>0b: There is no change in the OTG_ID pin.<br>1b: There is a change in the OTG_ID pin.<br>*Note*: The initial value is 1. Before using this bit, clear the status. |

### (3) OTG-BC Interrupt Enable Register

Abbreviated name of register: OBINTEN

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | DPMONCHG_EN | DMMONCHG_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 | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | CHGDETCHG1_EN | — | PDDETCHG1_EN | — | — | OCINT_EN | IDCHG_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 | R | R/W | R | R/W | R | R | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 18 | — | All 0 | R | Reserved<br>The write value should always be 0. |
| 17 | DPMONCHG_EN | 0 | R/W | DPMONCHG_STA bit interrupt enable<br>0b: Interrupt via the DPMONCHG_STA bit is disabled.<br>1b: Interrupt via the DPMONCHG_STA bit is enabled. |
| 16 | DMMONCHG_EN | 0 | R/W | DMMONCHG_STA bit interrupt enable<br>0b: Interrupt via the DMMONCHG_STA bit is disabled.<br>1b: Interrupt via the DMMONCHG_STA bit is enabled. |
| 15 to 7 | — | All 0 | R | Reserved<br>The write value should always be 0. |
| 6 | CHGDETCHG1_EN | 0 | R/W | CHGDETCHG1_STA bit interrupt enable<br>0b: Interrupt via the CHGDETCHG1_STA bit is disabled.<br>1b: Interrupt via the CHGDETCHG1_STA bit is enabled. |
| 5 | — | 0 | R | Reserved<br>The write value should always be 0. |
| 4 | PDDETCHG1_EN | 0 | R/W | PDDETCHG1_STA bit interrupt enable<br>0b: Interrupt via the PDDETCHG1_STA bit is disabled.<br>1b: Interrupt via the PDDETCHG1_STA bit is enabled. |
| 3 | — | 0 | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value.<br>Operation is not guaranteed if a value other than the initial value is written. |
| 2 | — | 0 | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value.<br>Operation is not guaranteed if a value other than the initial value is written. |
| 1 | OCINT_EN | 0 | R/W | OCINT_STA bit interrupt enable<br>0b: Interrupt via the OCINT_STA bit is disabled.<br>1b: Interrupt via the OCINT_STA bit is enabled. |
| 0 | IDCHG_EN | 0 | R/W | IDCHG_STA bit interrupt enable<br>0b: Interrupt via the IDCHG_STA bit is disabled.<br>1b: Interrupt via the IDCHG_STA bit is enabled. |

### (4) VBUS Control Register

Abbreviated name of register: VBCTRL

| 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 | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | VGPUO | — | — | — | VBOUT |
| 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 | R | R | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31, 30 | — | All 0 | R | Reserved<br>The write value should always be 0. |
| 29 | — | 0 | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. |
| 28 to 22 | — | All 0 | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. |
| 21 | — | 0 | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. |
| 20 to 17 | — | All 0 | R | Reserved<br>The write value should always be 0. |
| 16 | — | 1 | R | Reserved<br>The write value should always be 1. |
| 15 to 5 | — | All 0 | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. |
| 4 | VGPUO | 0 | R/W | The level corresponding to the inverse of this bit (in terms of positive logic) is output from the OTG_EXICEN pin.<br>This bit is used, for example, for control of the external power IC. |
| 3 to 1 | — | All 0 | R | Reserved<br>The write value should always be 0. |
| 0 | VBOUT | 0 | R/W | This bit is one of the VBUSEN control bits.<br>This bit is used to assert VBUS by controlling the external power IC.<br>0b: VBUS output disable<br>1b: VBUS output enable<br>If overcurrent occurs, this bit is automatically cleared to 0. |

### (5) Line Control Port 1 Register

Abbreviated name of register: LINECTRL1

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | DPRPU_EN | DP_RPU | DPRPD_EN | DP_RPD | DMRPD_EN | DM_RPD |
| 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 |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | DPMON | DMMON | — | IDMON |
| 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 26 | — | All 0 | R | Reserved |
| 25, 24 | — | All 0 | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value.<br>Operation is not guaranteed if a value other than the initial value is written. |
| 23, 22 | — | All 0 | R | Reserved |
| 21 | DPRPU_EN | 0 | R/W | This bit enables the DP_RPU (bit 20) setting for pullup control of D+ in port 1.<br>0: DP_RPU (bit 20) setting for pullup control of D+ in port 1 is disabled.<br>1: DP_RPU (bit 20) setting for pullup control of D+ in port 1 is enabled. |
| 20 | DP_RPU | 0 | R/W | This bit controls pullup of D+ in port 1.<br>This control is enabled only when DPRPU_EN (bit 21) = 1.<br>Note that if this bit is cleared to 0 when DPRPU_EN (bit 21) = 1,<br>D+ is in the floating state while no device drives the USB bus.<br>Note also that setting DPRPD_EN (bit 19) and DP_RPD (bit 18) to enabled is prohibited while this bit is set to enabled - that is, enabling both pullup and pulldown of D+ together is prohibited.<br>0: Pullup of D+ in port 1 is disabled.<br>1: Pullup of D+ in port 1 is enabled. |
| 19 | DPRPD_EN | 0 | R/W | This bit enables DP_RPD (bit 18) to control USB bus (DP) 15 kΩ Pulldown resister.<br>0b: Control of DP-side 15 kΩ Pulldown resister by DP_RPD (bit 18) is disabled.<br>1b: Control of DP-side 15 kΩ Pulldown resister by DP_RPD (bit 18) is enabled. |
| 18 | DP_RPD | 0 | R/W | This bit controls USB bus (DP) 15 kΩ Pulldown resister when DPRPD_EN (bit 19) = 1b.<br>0b: DP-side 15 kΩ Pulldown resister is OFF.<br>1b: DP-side 15 kΩ Pulldown resister is ON. |
| 17 | DMRPD_EN | 0 | R/W | This bit enables DM_RPD (bit 16) to control USB bus (DM) 15 kΩ Pulldown resister.<br>0b: Control of DM-side 15 kΩ Pulldown resister by DM_RPD (bit 16) is disabled.<br>1b: Control of DM-side 15 kΩ Pulldown resister by DM_RPD (bit 16) is enabled. |
| 16 | DM_RPD | 0 | R/W | This bit controls USB bus (DM) 15 kΩ Pulldown resister when DMRPD_EN (bit 17) = 1b.<br>0b: DM-side 15 kΩ Pulldown resister is OFF.<br>1b: DM-side 15 kΩ Pulldown register is ON. |
| 15 to 4 | — | All 0 | R | Reserved<br>The write value should always be 0. |
| 3 | DPMON | 0 | R | This bit indicates the value of USB bus DP. |
| 2 | DMMON | 0 | R | This bit indicates the value of USB bus DM. |
| 1 | — | 0 | R | Reserved<br>The write value should always be 0. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 0 | IDMON | 0 | R | This bit indicates the value of the OTG_ID pin. |

### (6) BC Control Port 1 Register

Abbreviated name of register: BCCTRL1

| 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 | 1 | 1 | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | PDDETSTS | CHGDETSTS | — | — | DCPMODE | VDMSRCE | IDPSINKE | VDPSRCE | IDMSINKE | IDPSRCE |
| 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 |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 10 | — | H’00_C000 | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. |
| 9 | PDDETSTS | 0 | R | This bit indicates the USBPHY Portable Device Detect signal state. |
| 8 | CHGDETSTS | 0 | R | This bit indicates the USBPHY Charging Downstream Port Detect signal state. |
| 7, 6 | — | All 0 | R | Reserved<br>The write value should always be 0. |
| 5 | DCPMODE | 0 | R/W | If USBPHY is used as DCP (Dedicated Charging Port), this bit is set to 1b. |
| 4 | VDMSRCE | 0 | R/W | This bit controls the USBPHY built-in VDM_SRC circuit.<br>If this bit is set to 1b, VDM_SRC goes ON, and the DM pin is driven. |
| 3 | IDPSINKE | 0 | R/W | This bit controls the USBPHY built-in Portable Device Detect circuit.<br>If this bit is set to 1b, Portable Device detection is enabled. |
| 2 | VDPSRCE | 0 | R/W | This bit controls the USBPHY built-in VDP_SRC circuit.<br>If this bit is set to 1b, VDP_SRC goes ON, and the DP pin is driven. |
| 1 | IDMSINKE | 0 | R/W | This bit controls the USBPHY built-in Charging Downstream Port Detect circuit.<br>If this bit is set to 1b, Charging Downstream Port detection is enabled. |
| 0 | IDPSRCE | 0 | R/W | This bit controls the USBPHY built-in IDP_SRC circuit.<br>If this bit is set to 1b, IDP_SRC goes ON, and the DP pin is driven. |

# 32A.3 Clock Signals

## 32A.3.1 Clock Gating Specifications

### 32A.3.1.1 Overview of clock gating

Because this module has a feature of switching the host controller and peripheral controller, clock supply to the unused controller might not be necessary.

Therefore, three clock gating control bits are allocated in the Register Enable/Clock Gating Control Register for the purpose of reducing power consumption by implementing clock gating for the “circuits that do not require clock supply temporarily.”

The clock gating control bits can be controlled to gate the clock that is supplied to the host controller or the peripheral controller.

[Target Register]

Register Enable/Clock Gating Control Register (offset:H’304)

[Functional specification]

| bit | Symbol | Functional specification |
|---|---|---|
| 31 | NONUSE_CLK_MSK | Gating clocks for unused host controller or peripheral controller. |
| 29 | HOST_CLK_MSK | Gating clocks for host controller. |
| 28 | PERI_CLK_MSK | Gating clocks for peripheral controller. |

### 32A.3.1.2 Specifications of NONUSE_CLK_MSK operation

This function automatically gates the clock to the unused host controller or the unused peripheral controller.

This function is enabled if the NONUSE_CLK_MSK bit (bit 31) is set to 1b.

If it is no problem whether the clock supply to the unselected function is stopped, use this function.

The following table shows the operating specifications of this function, based on the setting of the OTG_PERI bit (bit 31) of the Common Control Register (offset: H’800).

| Clock Gating Register Setting | | | Host/Peripheral Switching Setting | Gating Target | |
|---|---|---|---|---|---|
| | | | Register | | |
| NONUSE_CLK_MSK | HOST_CLK_MSK | PERI_CLK_MSK | OTG_PERI | Host Controller | Function Controller |
| 1 | 0 | 0 | 0 | | ✓ |
| | | | 1 | ✓ | |

[Note on using the NONUSE_CLK_MSK bit]

If the NONUSE_CLK_MSK bit is used, as the general rule, set the HOST_CLK_MSK/PERI_CLK_MSK bit to 0b.

If the HOST_CLK_MSK/PERI_CLK_MSK bit is 1b, the effect of clock gating becomes logical OR of each bit.

#### 32A.3.1.3 Specifications of PERI_CLK_MSK and HOST_CLK_MSK operations

This function forcibly gates the clock to the host controller and function controller.

The following table shows the operating specifications of this function.

| Clock Gating Register Setting | | | Host/Peripheral Switching Setting | Gating Target | |
|---|---|---|---|---|---|
| | | | Register | | |
| NONUSE_CLK_MSK | HOST_CLK_MSK | PERI_CLK_MSK | OTG_PERI | Host Controller | Function Controller |
| 0 | 1 | 0 | — | ✓ | |
| | 0 | 1 | — | | ✓ |
| | 1 | 1 | 0 | ✓ | ✓ |

**Note:** If the HOST_CLK_MSK/PERI_CLK_MSK bit is used, as the general rule, set the NONUSE_CLK_MSK bit to 0b.
If the NONUSE_CLK_MSK bit is set t0 1b, the effect of clock gating becomes logical OR of each bit.

# 32A.4 Interrupt Sources

## 32A.4.1 Interrupt Signals

This module has the five interrupt signals listed below.

| Interrupt Source Name | Interrupt Type | Pulse/ Level | Active Level |
|---|---|---|---|
| U2H_INT | BUS Master interrupt signal.<br>This signal is asserted when a bus error occurs in the BUS Master.<br>Interrupt control is performed by the AHB Bridge Register. | Level | H |
| U2H_OHCI_INT | OHCI interrupt signal<br>This signal is asserted during FS/LS transfer, when data transfer finishes or when the change of the USB bus state is detected.<br>Interrupt control is performed by the OHCI Operational Register. | Level | H |
| U2H_EHCI_INT | EHCI interrupt signal.<br>This signal is asserted during HS transfer, when data transfer finishes or when the change of the USB bus state is detected.<br>Interrupt control is performed by the EHCI Operational Register. | Level | H |
| U2H_WAKEON_INT | EHCI Wakeup interrupt signal.<br>This signal is asserted by an EHCI Wakeup event.<br>Interrupt control is performed by the EHCI Operational Register. | Level | H |
| U2H_OBINT | OTG/Battery Charging interrupt signal.<br>This signal is asserted by OTG or Battery Charging related event.<br>Interrupt control is performed by the UCOM2 Register. | Level | H |

## 32A.4.2 Interrupt Sources and Control

### 32A.4.2.1 U2H_INT assertion source and control

[Interrupt enable control by register]

Assert the interrupt enable bit in the register below. The interrupt signal is asserted when an interrupt source occurs.

AHB Bridge Register INT_ENABLE Register (offset:H’200) bit[0] (AHB_INTEN)

[Interrupt source]

A bus error (MHRESP = 1b) occurs in the AHB master.

[Clearing the interrupt]

To clear the interrupt, write 1b to the relevant bit in the register below.

AHB Bridge Register INT_STATUS Register (offset:H’204) bit[0] (AHB_INT)

### 32A.4.2.2 U2H_OHCI_INT assertion source and control

[Interrupt enable control by register]

Assert the interrupt enable bit in the registers below. The interrupt signal is asserted when an interrupt source occurs.

AHB Bridge Register INT_ENABLE Register (offset:H'200) bit[1] (USBH_INTAEN)

OHCI Operational Register HcInterruptEnable Register (offset:H'010) bit[31], bit[6:0]*[1]

**Note 1.** Enable the bit required as the assertion source.

[Interrupt source]

| Interrupt Source | | Registers That Require Interrupt Enable Setting | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | HcInterruptEnable | | | | | | |
| | | USBH_INTAEN | bit [31] MIE | bit [6] RHSCE | bit [5] FNOE | bit [3] ROE | bit [2] SFE | bit [1] WDHE | bit [0] SOE |
| 1 | Device connection is detected. | ✓ | ✓ | ✓ | | | | | |
| 2 | Device disconnection is detected. | ✓ | ✓ | ✓ | | | | | |
| 3 | Port power is OFF (excluding overcurrent detection). | ✓ | ✓ | ✓ | | | | | |
| 4 | Babble error is detected during a USB transfer. | ✓ | ✓ | ✓ | | | | | |
| 5 | Resume is complete. | ✓ | ✓ | ✓ | | | | | |
| 6 | Overcurrent is detected. | ✓ | ✓ | ✓ | | | | | |
| 7 | Bus reset is complete. | ✓ | ✓ | ✓ | | | | | |
| 8 | When the HcRhDescritorB Register DR bit is 1b, OHCI becomes "USB Operational" (HCFS[1:0] bit = 10b) or "USB Suspend" (HCFS[1:0] bit = 11b). | ✓ | ✓ | ✓ | | | | | |
| 9 | When no device is connected (CCS bit = 0b), 1 is written to bit [0] (Clear Port Enable) of the OHCI HcRhPort Status register. | ✓ | ✓ | ✓ | | | | | |
| 10 | When no device is connected (CCS bit = 0b), 1 is written to bit [1] (Set Port Enable) of the OHCI HcRhPort Status register. | ✓ | ✓ | ✓ | | | | | |
| 11 | When no device is connected (CCS bit = 0b), 1 is written to bit [2] (Set Port Suspend) of the OHCI HcRhPort Status register. | ✓ | ✓ | ✓ | | | | | |
| 12 | When no device is connected (CCS bit = 0b), 1 is written to bit [3] (Clear Suspend Status) of the OHCI HcRhPort Status register. | ✓ | ✓ | ✓ | | | | | |
| 13 | When no device is connected (CCS bit = 0b), 1 is written to bit [4] (Set Port Reset) of the OHCI HcRhPort Status register. | ✓ | ✓ | ✓ | | | | | |
| 14 | When no device is connected (CCS bit = 1b), the port power is turned off. | ✓ | ✓ | ✓ | | | | | |
| 15 | The MSB of bit [15:0] (Frame Number) of the HcFmNumber register has changed. | ✓ | ✓ | | ✓ | | | | |
| 16 | RemoteWakeup signal (Resume signal) is detected from a device. | ✓ | ✓ | | | ✓ | | | |
| 17 | HccaFrameNumber is updated. (Almost the same meaning as SOF is sent.) | ✓ | ✓ | | | | ✓ | | |
| 18 | A transfer finishes (including an error), and the host module updated HccaDoneHead. | ✓ | ✓ | | | | | ✓ | |
| 19 | USB schedule overrun occurred for the frame. | ✓ | ✓ | | | | | | ✓ |

[Clearing the interrupt]

To clear the interrupt, write 1b to the bit corresponding to the interrupt source in the register below to clear the interrupt.

OHCI Operational Register HcInterruptStatus Register (offset:H'00C) bit[6:0]

### 32A.4.2.3 U2H_EHCI_INT assertion source and control

[Interrupt enable control by register]

Assert the interrupt enable bit in the registers below. The interrupt signal is asserted when an interrupt source occurs.
AHB Bridge Register INT_ENABLE Register (offset:H'200) bit[2] (USBH_INTBEN)

EHCI Operational Register USBINTR Register (offset:H'128) bit[16], bit[5:0]*1

**Note 1.** Enable the bit required as the assertion source.

Also, control the relevant bit(s) in the register below as required.

EHCI Operational Register USBCMD Register (offset:H'120) bit[15], bit[6]

[Interrupt source]

| Interrupt Source | | Registers That Require Interrupt Enable Setting | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | USBCMD | | USBINTR | | | | | |
| | | | bit [15] | bit [6] | bit [16] | bit [5] | bit [3] | bit [2] | bit [1] | bit [0] |
| | | USBH_INTBEN | Per-Port Change Event | Doorbell | Port-1 Change Event | Async Advance | Frame List Rollover | Port Change Detect | USB ERRINT | USB INT |
| [1] | Device connection is detected. | ✓ | | | | | | ✓ | | |
| [2] | Device disconnection is detected. | ✓ | | | | | | ✓ | | |
| [3] | Overcurrent is detected. | ✓ | | | | | | ✓ | | |
| [4] | RemoteWakeup signal (Resume Signal) is detected from a device. | ✓ | | | | | | | | |
| [5] | Babble status of the USB bus is detected. | ✓ | | | | | | ✓ | | |
| [6] | USB transfer with "qTD IOC = 1b" normally finishes. | ✓ | | | | | | | | ✓ |
| [7] | Short packet is received. | ✓ | | | | | | | | ✓ |
| [8] | USB transfer finished with an error. (Retry transfer failed three times. A bubble error was detected. STALL was received.) | ✓ | | | | | | | ✓ | |
| [9] | The QH processing normally finished while the USBCMD Register bit [6] (Interrupt on Async Advance Doorbell) is 1b. | ✓ | | ✓ | | ✓ | | | | |
| [10] | The FRINDEX Register FrameIndex bit returned from the maximum value to H'000 (rollover detected). | ✓ | | | | | ✓ | | | |
| [11] | The Port Change Detect event (interrupt source 1 to 5) was detected. | ✓ | ✓ | | ✓ | | | | | |

[Clearing the interrupt]

To clear the interrupt, write 1b to the bit corresponding to the interrupt source in the register below to clear the interrupt.

EHCI Operational Register USBSTS Register (offset:H'124) bit[16], bit[5:0]

#### 32A.4.2.4 U2H_WAKEON_INT assertion source and control

[Interrupt enable control by register]

Assert the interrupt enable bit in the registers below. The interrupt signal is asserted when an interrupt source occurs.

AHB Bridge Register INT_ENABLE Register (offset:H'200) bit[4] (WAKEON_INTEN)

EHCI Operational Register PORTSC[1:2] Register (offset:H'164/H'168) bit[22:20]*1

**Note 1.** Enable the bit required as the assertion source.

[Interrupt source]

| Interrupt Source | | Registers That Require Interrupt Enable Setting | | | |
|---|---|---|---|---|---|
| | | | PORTSC | | |
| | | WAKEON_INTEN | bit [22] WKOC_E | bit [21] WKDSCNNT_E | bit [20] WKCNNT_E |
| 1 | Device connection is detected. | ✓ | | | ✓ |
| 2 | Device disconnection is detected. | ✓ | | ✓ | |
| 3 | Overcurrent is detected. | ✓ | ✓ | | |
| 4 | RemoteWakeup signal (Resume Signal) is detected from a device. | ✓ | | | |

[Clearing the interrupt]

To clear the interrupt, write 1b to the relevant bit in the register below.

AHB Bridge Register INT_STATUS Register (offset:H'204) bit[4] (WAKEON_INT)

### 32A.4.2.5 U2H_OBINT assertion source and control

[Interrupt enable control by register]

Assert the interrupt enable bit in the registers below. The interrupt signal is asserted when an interrupt source occurs.

AHB Bridge Register INT_ENABLE Register (offset:H'200) bit[3] (UCOM_INTEN)

UCOM Register OTG-BC Interrupt Enable Register (offset:H'808) bit[17:16], bit[6], bit[4], bit[1:0]*[1]

**Note 1.** Enable the bit required as the assertion source.

[Interrupt source]

| | | Registers That Require Interrupt Enable Setting | | |
|---|---|---|---|---|
| | | | OTG-BC Interrupt Enable | |
| | | | bit [17] | bit [16] |
| | Interrupt Source | UCOM_INTEN | DPMONCHG_EN | DMMONCHG_EN |
| 1 | The DP pin has changed. | ✓ | ✓ | |
| 2 | The DM pin has changed. | ✓ | | ✓ |

| | | Registers That Require Interrupt Enable Setting | | |
|---|---|---|---|---|
| | | | OTG-BC Interrupt Enable | |
| | | | bit [6] | bit [4] |
| | Interrupt Source | UCOM_INTEN | CHGDETCHG1_EN | PDDETCHG1_EN |
| 1 | The portable device detection signal has changed. | ✓ | | ✓ |
| 2 | The charging port detection signal has changed. | ✓ | ✓ | |

| | | Registers That Require Interrupt Enable Setting | | |
|---|---|---|---|---|
| | | | OTG-BC Interrupt Enable | |
| | | | bit [1] | bit [0] |
| | Interrupt Source | UCOM_INTEN | OCINT_EN | IDCHG_EN |
| 1 | Overcurrent is detected. (The change of the OVRCUR pin from 1 to 0b is detected.) | ✓ | ✓ | |
| 2 | The OTG_ID pin has changed. | ✓ | | ✓ |

[Clearing the interrupt]

To clear the interrupt, write 1b to the bit corresponding to the interrupt source in the register below to clear the interrupt.

UCOM Register OTG-BC Interrupt Status Register (offset:H'804) bit[17:16], bit[6], bit[4], bit[1:0]

### 32A.4.3 Timing of De-asserting Interrupt Signals

After the register access to clear an interrupt source, it may take time to begin clearing the interrupt triggered by the interrupt source. Therefore, take a measure to prevent the false recognition of interrupts during the period from the end of the register clear access until the next interrupt is recognized.

## 32A.5 Power-Saving Function

This module controls power consumption by using the following two methods:

1. Controlling the SUSPENDM/SLEEPM pin of USBPHY
2. Using clock gating to stop the clock to the host controller or peripheral controller.

### 32A.5.1 Controlling the SUSPENDM and SLEEPM Pins of the USBPHY

You can expect the following power-saving effects by asserting the SUSPENDM and SLEEPM pins of the USBPHY:

- Reducing the power consumption by the USBPHY
- Reducing the power consumption by the host core by stopping the clocks from the USBPHY

As described at **Section 32A.1.2.5, Suspend extension function**, by default, the SUSPENDM and SLEEPM pins of the USBPHY are not asserted even when EHCI and OHCI are put into the Suspended state.

See **Section 32A.2.4.4(6), Suspend Control Register** and control the relevant registers appropriately.

### 32A.5.2 Controlling the Clock-Gating Function

See **Section 32A.2.4.4(5), Register Enable/Clock Gating Control Register** and **Section 32A.3.1, Clock Gating Specifications**, then control the relevant registers appropriately.

# 32A.6 Battery Charging

## 32A.6.1 Support of charging port

The charging port refers to a port that supplies power in compliance with the Battery Charging Specification. The charging port is usually installed on the host controller.

Charging ports can be generally classified by function as described below.

| Type | Function |
|---|---|
| CDP<br>(Charging Downstream Port) | Downstream port that can supply power in compliance with the Battery Charging Specification.<br>This type of charging port detects a portable device, and after the handshake (for Battery Charging) finishes, it proceeds to the usual device connect sequence (and operates as the usual host). |
| DCP<br>(Dedicated Charging Port) | Port that provides only power supply function in compliance with the Battery Charging Specification.<br>This type of charging port does not operate as the usual host. |
| SDP<br>(Standard Downstream Port) | Standard Downstream Port that is not compliant with the Battery Charging Specification.<br>This type of charging port supplies power in the range conforming to the conventional USB 2.0 standard, and operates as the usual host. |

## 32A.6.2 Support of portable device

A portable device refers to a device that is supplied power (or that requests power supply) in compliance with the Battery Charging Specification. The portable port is usually installed in the peripheral controller.

# 32A.7 Bus Master

## 32A.7.1 Functional specifications of the bus master

### 32A.7.1.1 Supported bus master functions

| Function as the BUS MASTER | Status |
|---|---|
| Residual burst after an error response is received | The transfer is not stopped. |
| 1 Kbyte boundary processing | The fixed length burst (INCRx) across 1 Kbyte boundary is not performed. |

### 32A.7.1.2 Issuing requests for different types of bus transfer

| MHTRANS[1:0] | MHSIZE[2:0] | MHWRITE | MHBURST[2:0] | Reply | Remarks |
|---|---|---|---|---|---|
| IDLE (00b) | — | — | — | — | — |
| BUSY (01b) | — | — | — | — | Not issued. |
| NONSEQ (10b) | 32-bit (010b) | WRITE | SINGLE<br>INCR4<br>INCR8<br>INCR16 | OKAY/ERROR | 32-bit transfer is issued.<br>A response error is reported by an interrupt, and the transfer is not stopped. |
| | | READ | | | |
| | 8-bit (000b)<br>16-bit (001b) | WRITE | SINGLE | OKAY/ERROR | 8-bit/16-bit transfer is issued (SINGLE only). A response error is reported by an interrupt. |
| | other than the above | — | — | — | Not issued. |
| SEQ (11b) | 32-bit (010b) | WRITE | INCR4<br>INCR8<br>INCR16 | OKAY/ERROR | 32-bit transfer is issued.<br>A response error is reported by an interrupt, and the transfer is not stopped. |
| | | READ | | OKAY/ERROR | |
| | other than the above | — | — | — | Not issued. |

### 32A.7.1.3 Supported responses

| Response type | Response | Remarks |
|---|---|---|
| OKAY | Enable | Supported. |
| ERROR | Enable | A response error is reported by an interrupt, and termination of transfer (Early Burst Termination) is not performed. |

### 32A.7.1.4 Protection control information

The value of MHPROT[3:0] can be set in the PROT_TYPE bits (bits [15:12]) in the AHB_BUS_CTR Register (offset: H’208).

Also, when the PROT_MODE bit (bit 8) in the AHB_BUS_CTR Register is set, only the last data transfer in an EHCI/OCHI DMA transfer can be handled as a non-buffered transfer, and other transfers can be handled as buffered transfers.

#### 32A.7.1.5 Maximum burst length

The maximum burst length can be selected from SINGLE, INCR4, INCR8, and INCR16 by using the MAX_BURST_LEN bits (bit[1:0]) in the AHB_BUS_CTR Register (offset: H’208). The maximum burst length is common to reading and writing.

#### 32A.7.1.6 Boundary of transfer data

The value of MHADDR[31:0] bits does not exceed 1 KB boundary during a burst transfer. Also, you can change the address boundary for burst transfer to 16, 32, or 64 bytes by writing a value to the ALIGN_ADDRESS bits (bit[5:4]) in the AHB_BUS_CTR Register (offset: H’208).

### 32A.7.1.7 Start address of fixed-length INCR burst transfer

The following table lists the values of the lower bits of MHADDR to be applied when a fixed-length INCR burst transfer starts.

| ALIGN_ADDRESS Setting | Fixed length Start Address of INCR Burst | | |
|---|---|---|---|
| | INCR4 | INCR8 | INCR16 |
| 00b<br>(Aligned at the 1-Kbyte boundary) | MHADDR[9:0] = H'000<br>H'004<br>H'008<br>H'00C<br>H'010<br>:<br>H'3D0<br>H'3D4<br>H'3D8<br>H'3DC<br>H'3E0<br>H'3E4<br>H'3E8<br>H'3EC<br>H'3F0 | MHADDR[9:0] = H'000<br>H'004<br>H'008<br>H'00C<br>H'010<br>:<br>H'3C8<br>H'3CC<br>H'3D0<br>H'3D4<br>H'3D8<br>H'3DC<br>H'3E0 | MHADDR[9:0] = H'000<br>H'004<br>H'008<br>H'00C<br>H'010<br>:<br>H'3B8<br>H'3BC<br>H'3C0 |
| 01b<br>(Aligned at the 16-byte boundary) | MHADDR[3:0] = H'0 | -- (Not issued) | -- (Not issued) |
| 10b<br>(Aligned at the 32-byte boundary) | MHADDR[4:0] = H'00<br>H'04<br>H'08<br>H'0C<br>H'10 | MHADDR[4:0] = H'00 | -- (Not issued) |
| 11b<br>(Aligned at the 64-byte boundary) | MHADDR[5:0] = H'00<br>H'04<br>H'08<br>H'0C<br>H'10<br>H'14<br>H'18<br>H'1C<br>H'20<br>H'24<br>H'28<br>H'2C<br>H'30 | MHADDR[5:0] = H'00<br>H'04<br>H'08<br>H'0C<br>H'10<br>H'14<br>H'18<br>H'1C<br>H'20 | MHADDR[5:0] = H'00 |

# 32A.8 Overcurrent Control and VBUS Control

## 32A.8.1 OVRCUR/VBUSEN pin

Overcurrent detection on the USB port and port power (VBUS) control are performed by the external power IC connected to this module.

| This module pin | Input/Output | Level | Description |
|---|---|---|---|
| OVRCURI | Input | L | Overcurrent status was detected. |
| | | H | Overcurrent status was not detected. |
| VBUSEN | Output | L | Port Power (VBUS) OFF |
| | | H | Port Power (VBUS) ON |

## 32A.8.2 Overcurrent detection timer setting

This module detects overcurrent when the OVRCUR pin remains asserted (0b) for a set period.

The period over which assertion of the OVRCUR pin is required (“overcurrent detection time”) can be set in the following register.

| Register | Overcurrent detection/sleep timer setting register (offset: H’310) |
|---|---|
| Bits | TIMER_OC[19:0] |
| Initial value | H’3_0D40 |

The overcurrent detection time can be converted from the setting value of the above register, taking that one bit equal to the internal bus clock (P1ϕ) cycle.

Therefore, set the value of the above register at initial configuration, considering “the internal bus clock (P1ϕ) frequency to be used” and “the overcurrent detection time you want to specify”.

### 32A.8.3 Port Power (VBUS) control specifications

VBUSEN can be controlled by the Port Power bit of the EHCI/OHCI Operational register or by the VBOUT bit of the VBUS Control Register. Which of the above bit controls VBUSEN is determined by the VBUSEN control (VBENCTL) register in USBPHY Control.

Refer to 32.3.4.

When using as Function, VBUSEN should be 0..

The following table describes control examples by the Port Power bit of the EHCI/OHCI Operational register.

| Situation | | Register | | bit |
|---|---|---|---|---|
| EHCI control | | PORTSC1 (offset: H'164) | | bit 12 (PP) |
| OHCI control | Global control*1 | ON setting | HcRhStatus Register (offset: H'050) | bit 16 (Set Global Power) |
| | | OFF setting | | bit 0 (Clear Port Status) |
| | Selective control*2 | ON setting | HcRhPortStatus1 Register (offset: H'054) | bit 8 (Set Port Power) |
| | | OFF setting | | bit 9 (Clear Port Power) |

Note 1. Global control refers to the status in which the register settings are as follows:
HcRhDescriptorA Register (offset: H'048) bit 8 (PSM) = 0b
or
HcRhDescriptorA Register (offset: H'048) bit 8 (PSM) = 1b
and
HcRhDescriptorB Register (offset: H'04C) bit 17 (PPCM[1]) = 0b

Note 2. Selective control refers to the status in which the register settings are as follows:
HcRhDescriptorA Register (offset: H'048) bit 8 (PSM) = 1b
and
HcRhDescriptorB Register (offset: H'04C) bit 17 (PPCM[1]) = 1b

However, if the register settings are as follows, the VBUSEN pin is always asserted (1b), and the Port Power (VBUS) becomes ON, regardless of the OVRCUR pin's status.

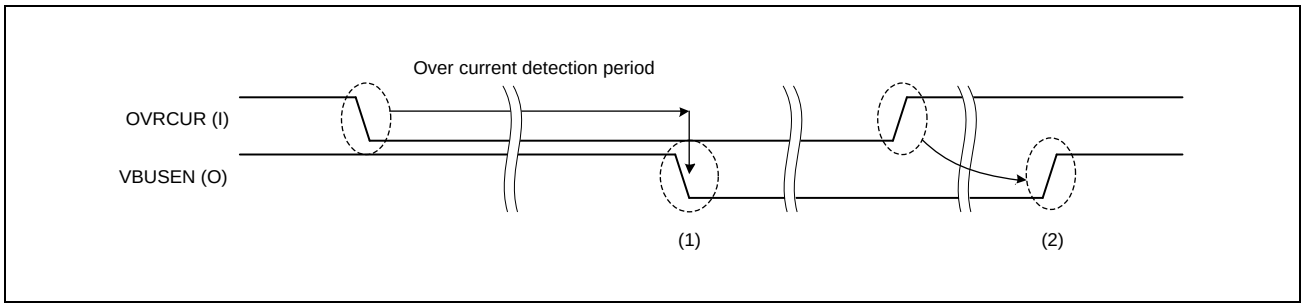
Specify the settings as necessary (for example, your system needs that VBUS is always ON).

| EHCI Operational | OHCI Operational | | | | |
|---|---|---|---|---|---|
| HCSPARAMS (offset: H'104) | HcRhDescriptorA (offset: H'048) | | | HcRhDescriptorB (offset: H'04C) | Pin Operation |
| PPC (bit 4) | NOCP (bit 12) | NPS (bit 9) | PSM (bit 8) | PPCM[1] (bit 17) | When the OVRCUR pin is asserted (0b) |
| 0 | — | — | — | — | Fixed to 1b |
| — | 1 | — | — | — | Fixed to 1b |
| — | — | 1 | — | — | Fixed to 1b |
| 1 | 0 | 0 | 0 | — | 0b |
| | | | 1 | 0 | |
| | | | | 1 | |

### 32A.8.4 Timing Chart for Overcurrent Detection and Recovery

The figure below shows the assertion/de-assertion timings of the OVRCUR and VBUSEN pins signals at overcurrent detection and recovery.

Note that this timing chart is on the assumption that the changes of register settings to fix the Port Power bit to asserted state (see **Section 32A.8.3, Port Power (VBUS) control specifications**) have not been made.




1. When the OVRCUR pin is kept asserted (0b) for the overcurrent detection time, this module determines the occurrence of overcurrent, and then de-asserts the VBUSEN pin (0b).
2. After the overcurrent status has been resolved, and de-assertion of the OVRCUR pin (1b) is confirmed, 1b is written to the Port Power bit described in **Section 32A.8.3, Port Power (VBUS) control specifications** to turn on the Port Power (Vbus).

NOTE

Before the Port Power bit is set by firmware, be sure to check that the OVRCUR pin has been deasserted.

# 32A.9 Procedure for Setting this Module

## 32A.9.1 Host/Peripheral Common Setting Sequence

The following shows the necessary sequence common to both host and peripheral modes.

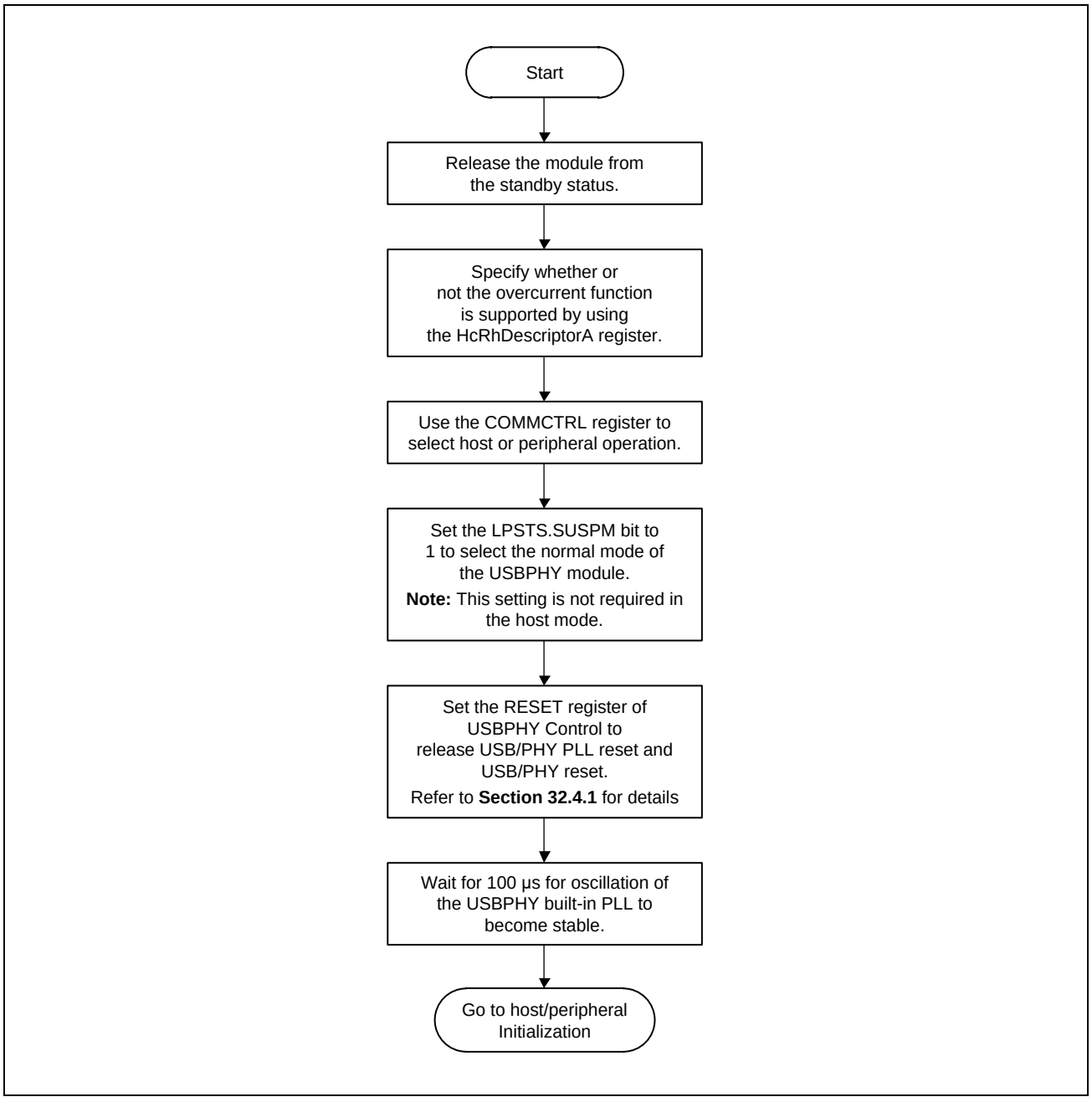



Figure 32A.1 Sequence Common to Both Host and Peripheral Modes

### 32A.9.2 Initialization Sequence

The following shows the initialization sequence in the Host mode

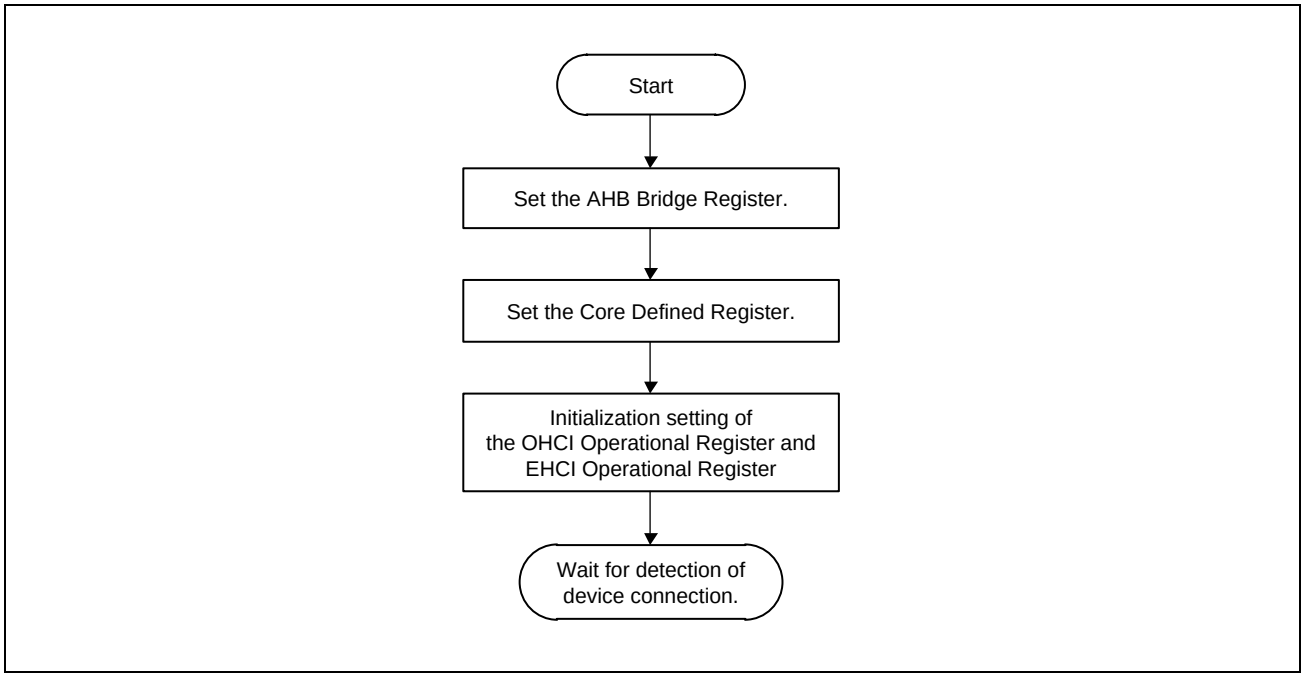



Figure 32A.2 Initialization Sequence

### 32A.9.3 Flow of Error Handling

While operating this module, if the operation falls into an abnormal state and recovery to the normal flow seems difficult, perform the following steps for reset.

[When an abnormality occurs while EHCI is running]

1. Write 1b to the HCRESET bit (bit 1) of the EHCI Operational Register USBCMD Register (offset: H’120) to execute EHCI software reset.
2. Re-initialize the EHCI Operational Register.

[When an abnormality occurs while OHCI is running]

1. Write 1b to the HCR bit (bit 0) of the OHCI Operational Register HcCommandStatus Register (offset: H’008) to execute OHCI software reset.
2. Re-initialize the OHCI Operational Register.

# 32B. USB 2.0 Function Module

NOTE

This LSI does not support the OTG function. Related description of the OTG Function in this section should be ignored.

## 32B.1 Overview

### 32B.1.1 Overview

This LSI has USB 2.0 host/function module. Switching between the host and peripheral functions is possible by setting the UCOM register.

The setting for battery charging is handled by the host controller even if the peripheral controller is selected.

This chapter describes the peripheral controller. For details on the host/peripheral common circuit and battery charging, see **Section 32A, USB 2.0 Host Module**.

This module is a universal serial bus (USB) controller that has peripheral functions.

This module supports high-speed and full-speed transfer defined by the Universal Serial Bus Specification Revision 2.0.

This module supports all transfer types defined in the USB Specification. This module incorporates 8 Kbytes of buffer memory for data transfer, and can use a maximum of 10 pipes. You can assign any endpoint number to any pipe other than pipe 0 in conformity to the peripheral equipment or system to communicate with.

### 32B.1.2 Features

#### 32B.1.2.1 Peripheral controller supporting high-speed USB

(1) On-chip peripheral USB controller

#### 32B.1.2.2 Support of all types of USB transfer

(1) Supporting all types of USB transfer, including isochronous transfer

(2) Control transfer

(3) Bulk transfer

(4) Interrupt transfer (high-bandwidth transfers not supported)

(5) Isochronous transfer (high-bandwidth transfers not supported)

#### 32B.1.2.3 Bus interface

(1) Includes a two-channel DMA interface

### 32B.1.2.4 Pipe configuration

(1) 8 Kbytes of buffer memory for USB communications

(2) Up to 10 pipes (including the default control pipe) selectable

(3) Programmable pipe configuration

(4) Any endpoint number assignable to pipes other than pipe 0

Table 32B.1 Pipe Settings

| PIPE | Transfer Type | Double buffer | Continuous transfer mode | Buffer size |
|---|---|---|---|---|
| PIPE0 | Control | — | — | Fixed to 64 bytes/<br>256 bytes (CNTMD = 1) |
| PIPE1 | Iso/Bulk | ✓ | ✓ (Bulk only) | Up to 2 Kbytes |
| PIPE2 | Iso/Bulk | ✓ | ✓ (Bulk only) | Up to 2 Kbytes |
| PIPE3 | Bulk | ✓ | ✓ | Up to 2 Kbytes |
| PIPE4 | Bulk | ✓ | ✓ | Up to 2 Kbytes |
| PIPE5 | Bulk | ✓ | ✓ | Up to 2 Kbytes |
| PIPE6 | Int | — | — | Fixed to 64 bytes |
| PIPE7 | Int | — | — | Fixed to 64 bytes |
| PIPE8 | Int | — | — | Fixed to 64 bytes |
| PIPE9 | Int | — | — | Fixed to 64 bytes |

### 32B.1.2.5 Features of peripheral functions

(1) Support of high-speed transfer (at 480 Mbps) and full-speed transfer (at 12 Mbps)

(2) Automatic recognition of high-speed or full-speed operation based on automatic response to the reset handshake

(3) Control transfer stage monitoring function

(4) Device state monitoring function

(5) Automatic response to SET_ADDRESS request

(6) NAK response interrupt (NRDY)

(7) SOF interpolation

### 32B.1.2.6 Features of DMA transfer

DMA transaction mode: Fetching in both register and link modes are supported.

Interrupt: Level is supported.

Transfer size: A transfer size from 1 to 128 bytes can be selected separately for the transfer source and transfer destination.

Skip (scatter/gather) function: The access size and skip size can be specified separately for the transfer source and destination.

Suspend function: A running DMA transaction can be suspended temporarily.

Interval function: The interval of DMA transfers can be specified to control the bus occupancy.

#### 32B.1.2.7 Other functions

(1) Byte endian swap function to support both big endian and little endian as data formats (when using only CFIFO)

(2) Transfer ending function using a transaction counter

(3) SOF pulse output function

(4) BRDY interrupt event notification timing change function (BFRE)

(5) Function (SHTNAK) to set NAK in the response PID when transfer ends

(6) Support of the Link Power Management (LPM) ECN, making available a new low-power-consumption state (L1 state)

## 32B.1.3 Overview of Functions

### 32B.1.3.1 Automatic recognition of USB transfer speed

This module automatically recognizes USB transfer speed.

#### (1) Methods of FIFO buffer memory access

This module supports the two types of access described below to the FIFO buffer memory for USB data transfer.

##### (a) Access from the CPU

Specify a FIFO port address, and then write data to or read data from the FIFO buffer memory.

##### (b) Direct memory access (DMA)

Selecting a pipe window and setting the DMA control registers enables writing data to or reading data from the FIFO buffer memory.

### 32B.1.3.2 USB event

This module notifies the event in USB operation by issuing an interrupt.

You can specify whether to enable notification by interrupt for individual interrupt types and sources through software settings.

### 32B.1.3.3 USB data transfer

This module performs all types of USB data transfer: control transfer, bulk transfer, interrupt transfer, and isochronous transfer. The following pipe resources are available for individual transfer types:

(1) One pipe dedicated to control transfer

(2) Four pipes dedicated to interrupt transfer

(3) Three pipes dedicated to bulk transfer

(4) Two pipes selectively used for bulk or isochronous transfer

For each pipe, specify the settings, including transfer type, endpoint number, and maximum packet size, required for USB transfer according to the system.

This module can incorporate up to 8 Kbytes of buffer memory. For the pipes dedicated to bulk transfer and those selectively used for bulk transfer or isochronous transfer, allocate buffer memory and specify a buffer operating mode and other necessary settings according to the system. Setting the buffer operating mode enables high-speed data transfers with fewer interrupts to be performed by using double-buffering and continuous transfer of data packets.

### 32B.1.3.4 SOF pulse output function

This module has a function to output an SOF pulse to indicate the timing of SOF packet transmission. This module asserts a SOF pulse output signal when an SOF packet is received. This module outputs pulses at regular intervals based on an SOF interpolation timer even when an SOF packet is damaged.

## 32B.1.4 Restriction matter and Notes

### 32B.1.4.1 Restriction matter

#### (1) Restrictions on the USB specifications

It's no support About below of USB 2.0 specification

- HighBandWidth transfer is no support

#### (2) Restrictions on DMA Master

- DAD = 1 (destination address fixed) and can not use skip transfer of destination side. When forwarding by such setting, movement is unsettled. Should not do such transmission.
- SAD = 1 (source address fixed), and can not use skip transfer of source side. When forwarding by such setting, movement is unsettled. Should not do such transmission.
- DAD = 1 (destination address fixed), and can not use the beat unalign transfer of destination side. When forwarding by such setting, movement is unsettled. Should not do such transmission.
- SAD = 1 (source address fixed), and can not use beat unalign transfer of source side. When forwarding by such setting, movement is unsettled. Should not do such transmission.
- When REQD =1, SBE = 1 (sweep mode) and compulsion discharge function can not use.

### 32B.1.4.2 Notes

#### (1) DMA transfer and passing problem of interrupt signal in DMA Master construction.

##### (a) Overview

A USBFDMAmn interrupt (m, n = 0, 1) might occur before the last data of DMA transaction is written to the write-target device.

##### (b) Plan to avoid

A plan to avoid the above problem is indicated below.

Plan to avoid 1) HPROT is set as non bufferable.

According to the DMA mode, set 0 in the DPR[2] bit in the CHEXT_n register or the LDPR[2] bit in the DCTRL register, set the HPROT signal as non bufferable, and then perform the DMA transaction.

If all transfers are set to non bufferable, transfer efficiency might fall.

In such cases, set the most part of transaction as bufferable and perform a transfer, and then perform the last transfer by setting a register set or descriptor as non bufferable.

#### (2) Setting for battery charging when the peripheral controller is selected

The setting for battery charging is handled by the host controller even if the peripheral controller is selected.

## 32B.2 Registers

How to read the register table

(1) Bit number:

(2) State after reset: Initial state of the register that occurs immediately after a reset “Power on Reset” indicates the initial state at power-on reset.
The state after USB reset is the initial state of the register that occurs when this module detects a USB bus reset.
Significant points regarding reset operation are indicated in notes.
“—” indicates that a user’s setting is retained without this module operation having been performed. “X” indicates that the value is undefined.

(4) Access condition: The condition to be met when this module accesses the register for an operation.

R: Reading only
W: Writing only
R/W: Reading or writing
R(0): 0-reading only
W(1): 1-writing only

(5) Name: Bit symbol and bit name

(6) Function: Description of functions.

**<Example of description>**

| (1) Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | A bit | B bit | C bit | — | — | — | — | — | — | — | — | — | — | — | — |
| (2) Initial Value | X | 0 | 0 | 0 | | | | | | | | | | | | |
| R/W | X | 0 | — | — | | | | | | | | | | | | |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 | — | | | Nothing is assigned. Fix this bit to 0. |
| 14 | A bit | | R/W | AAA enable<br>0: Disables operation<br>1: Enables operation |
| 13 | B bit | | R | BBB operation<br>0: Outputs low-level signal<br>1: Outputs high-level signal |
| 12 | C bit | | R(0)/ W(1) | CCC control<br>0: .....<br>1: ..... |
| | (5) | | (4) | (6) |

### 32B.2.1 Base Address

Table 32B.2 Base addresses of the USB function module

| Channel number | Base address |
|---|---|
| 0 | H'0_11C6_0000 (Cortex-A55 Address Space) |

### 32B.2.2 List of Registers

**Table 32B.3** lists the registers of this module.

Table 32B.3 List of Registers (1/3)

| Offset Address | Symbol | Name | Access unit |
|---|---|---|---|
| 000 | SYSCFG0 | System Configuration Control Register 0 | 16-bit |
| 002 | SYSCFG1 | System Configuration Control Register 1 | 16-bit |
| 004 | SYSSTS0 | System Configuration Status Register | 16-bit |
| 008 | DVSTCTR0 | Device Control Register 0 | 16-bit |
| 00C | TESTMODE | Test Mode Register | 16-bit |
| 014 | CFIFO | CFIFO Port Register | 8/16/32-bit |
| 016 | | | |
| 020 | CFIFOSEL | CFIFO Port Select Register | 16-bit |
| 022 | CFIFOCTR | CFIFO Port Control Register | 16-bit |
| 028 | D0FIFOSEL | D0FIFO Port Select Register | 16-bit |
| 02A | D0FIFOCTR | D0FIFO Port Control Register | 16-bit |
| 02C | D1FIFOSEL | D1FIFO Port Select Register | 16-bit |
| 02E | D1FIFOCTR | D1FIFO Port Control Register | 16-bit |
| 030 | INTENB0 | Interrupt Enable Register 0 | 16-bit |
| 036 | BRDYENB | BRDY Interrupt Enable Register | 16-bit |
| 038 | NRDYENB | NRDY Interrupt Enable Register | 16-bit |
| 03A | BEMPENB | BEMP Interrupt Enable Register | 16-bit |
| 03C | SOFCFG | SOF Output Configuration Register | 16-bit |
| 040 | INTSTS0 | Interrupt Status Register 0 | 16-bit |
| 046 | BRDYSTS | BRDY Interrupt Status Register | 16-bit |
| 048 | NRDYSTS | NRDY Interrupt Status Register | 16-bit |
| 04A | BEMPSTS | BEMP Interrupt Status Register | 16-bit |
| 04C | FRMNUM | Frame Number Register | 16-bit |
| 04E | UFRMNUM | Micro Frame Number Register | 16-bit |
| 050 | USBADDR | USB Address Register | 16-bit |
| 054 | USBREQ | USB Request Type Register | 16-bit |
| 056 | USBVAL | USB Request Value Register | 16-bit |
| 058 | USBINDX | USB Request Index Register | 16-bit |
| 05A | USBLENG | USB Request Length Register | 16-bit |
| 05C | DCPCFG | DCP Configuration Register | 16-bit |
| 05E | DCPMAXP | DCP Max. Packet Size Register | 16-bit |

Table 32B.3 List of Registers (2/3)

| Offset Address | Symbol | Name | Access unit |
|---|---|---|---|
| 060 | DCPCTR | DCP Control Register | 16-bit |
| 064 | PIPESEL | Pipe Window Select Register | 16-bit |
| 068 | PIPECFG | Pipe Configuration Register | 16-bit |
| 06A | PIPEBUF | Pipe Buffer Setting Register | 16-bit |
| 06C | PIPEMAXP | Pipe Max. Packet Size Register | 16-bit |
| 06E | PIPEPERI | Pipe Cycle Control Register | 16-bit |
| 070 | PIPE1CTR | PIPE1 Control Register | 16-bit |
| 072 | PIPE2CTR | PIPE2 Control Register | 16-bit |
| 074 | PIPE3CTR | PIPE3 Control Register | 16-bit |
| 076 | PIPE4CTR | PIPE4 Control Register | 16-bit |
| 078 | PIPE5CTR | PIPE5 Control Register | 16-bit |
| 07A | PIPE6CTR | PIPE6 Control Register | 16-bit |
| 07C | PIPE7CTR | PIPE7 Control Register | 16-bit |
| 07E | PIPE8CTR | PIPE8 Control Register | 16-bit |
| 080 | PIPE9CTR | PIPE9 Control Register | 16-bit |
| 090 | PIPE1TRE | PIPE1 Transaction Counter Enable Register | 16-bit |
| 092 | PIPE1TRN | PIPE1 Transaction Counter Register | 16-bit |
| 094 | PIPE2TRE | PIPE2 Transaction Counter Enable Register | 16-bit |
| 096 | PIPE2TRN | PIPE2 Transaction Counter Register | 16-bit |
| 098 | PIPE3TRE | PIPE3 Transaction Counter Enable Register | 16-bit |
| 09A | PIPE3TRN | PIPE3 Transaction Counter Register | 16-bit |
| 09C | PIPE4TRE | PIPE4 Transaction Counter Enable Register | 16-bit |
| 09E | PIPE4TRN | PIPE4 Transaction Counter Register | 16-bit |
| 0A0 | PIPE5TRE | PIPE5 Transaction Counter Enable Register | 16-bit |
| 0A2 | PIPE5TRN | PIPE5 Transaction Counter Register | 16-bit |
| 0D0 | DEVADD0 | Device Address 0 Configuration Register | 16-bit |
| 0D2 | DEVADD1 | Device Address 1 Configuration Register | 16-bit |
| 0D4 | DEVADD2 | Device Address 2 Configuration Register | 16-bit |
| 0D6 | DEVADD3 | Device Address 3 Configuration Register | 16-bit |
| 0D8 | DEVADD4 | Device Address 4 Configuration Register | 16-bit |
| 0DA | DEVADD5 | Device Address 5 Configuration Register | 16-bit |
| 0DC | DEVADD6 | Device Address 6 Configuration Register | 16-bit |
| 0DE | DEVADD7 | Device Address 7 Configuration Register | 16-bit |
| 0E0 | DEVADD8 | Device Address 8 Configuration Register | 16-bit |
| 0E2 | DEVADD9 | Device Address 9 Configuration Register | 16-bit |
| 0E4 | DEVADDA | Device Address A Configuration Register | 16-bit |
| 100 | LPCTRL | Low Power Control Register | 16-bit |
| 102 | LPSTS | Low Power Status Register | 16-bit |
| 104 | PHYFUNCTR | PHY Function Control Register | 16-bit |
| 10A | PHYOTGCTR | PHY OTG Control Register | 16-bit |
| 144 | PL1CTRL1 | Peripheral L1 Control Register 1 | 16-bit |
| 146 | PL1CTRL2 | Peripheral L1 Control Register 2 | 16-bit |
| 400 | N0SA_0 | Next0 Source Address Register 0 | 32-bit |
| 404 | N0DA_0 | Next0 Destination Address Register 0 | 32-bit |
| 408 | N0TB_0 | Next0 Transaction Byte Register 0 | 32-bit |

Table 32B.3 List of Registers (3/3)

| Offset Address | Symbol | Name | Access unit |
|---|---|---|---|
| 40C | N1SA_0 | Next1 Source Address Register 0 | 32-bit |
| 410 | N1DA_0 | Next1 Destination Address Register 0 | 32-bit |
| 414 | N1TB_0 | Next1 Transaction Byte Register 0 | 32-bit |
| 418 | CRSA_0 | Current Source Address Register 0 | 32-bit |
| 41C | CRDA_0 | Current Destination Address Register 0 | 32-bit |
| 420 | CRTB_0 | Current Transaction Byte Register 0 | 32-bit |
| 424 | CHSTAT_0 | Channel Status Register 0 | 32-bit |
| 428 | CHCTRL_0 | Channel Control Register 0 | 32-bit |
| 42C | CHCFG_0 | Channel Configuration Register 0 | 32-bit |
| 430 | CHITVL_0 | Channel Interval Register 0 | 32-bit |
| 434 | CHEXT_0 | Channel Extension Register 0 | 32-bit |
| 438 | NXLA_0 | Next Link Address Register 0 | 32-bit |
| 43C | CRLA_0 | Current Link Address Register 0 | 32-bit |
| 440 | N0SA_1 | Next0 Source Address Register 1 | 32-bit |
| 444 | N0DA_1 | Next0 Destination Address Register 1 | 32-bit |
| 448 | N0TB_1 | Next0 Transaction Byte Register 1 | 32-bit |
| 44C | N1SA_1 | Next1 Source Address Register 1 | 32-bit |
| 450 | N1DA_1 | Next1 Destination Address Register 1 | 32-bit |
| 454 | N1TB_1 | Next1 Transaction Byte Register 1 | 32-bit |
| 458 | CRSA_1 | Current Source Address Register | 32-bit |
| 45C | CRDA_1 | Current Destination Address Register 1 | 32-bit |
| 460 | CRTB_1 | Current Transaction Byte Register 1 | 32-bit |
| 464 | CHSTAT_1 | Channel Status Register 1 | 32-bit |
| 468 | CHCTRL_1 | Channel Control Register 1 | 32-bit |
| 46C | CHCFG_1 | Channel Configuration Register 1 | 32-bit |
| 470 | CHITVL_1 | Channel Interval Register 1 | 32-bit |
| 474 | CHEXT_1 | Channel Extension Register 1 | 32-bit |
| 478 | NXLA_1 | Next Link Address Register 1 | 32-bit |
| 47C | CRLA_1 | Current Link Address Register 1 | 32-bit |
| 600 | SCNT_0 | Source Continuous Register 0 | 32-bit |
| 604 | SSKP_0 | Source Skip Register 0 | 32-bit |
| 608 | DCNT_0 | Destination Continuous Register 0 | 32-bit |
| 60C | DSKP_0 | Destination Skip Register 0 | 32-bit |
| 620 | SCNT_1 | Source Continuous Register 1 | 32-bit |
| 624 | SSKP_1 | Source Skip Register 1 | 32-bit |
| 628 | DCNT_1 | Destination Continuous Register 1 | 32-bit |
| 62C | DSKP_1 | Destination Skip Register 1 | 32-bit |
| 700 | DCTRL | DMA Control Register | 32-bit |
| 704 | DSCITVL | Descriptor Interval | 32-bit |
| 710 | DST_EN | DMA Status EN Register | 32-bit |
| 714 | DST_ER | DMA Status ER Register | 32-bit |
| 718 | DST_END | DMA Status END Register | 32-bit |
| 71C | DST_TC | DMA Status TC Register | 32-bit |
| 720 | DST_SUS | DMA Status SUS Register | 32-bit |

## 32B.2.3 System Configuration Control Registers

### 32B.2.3.1 System Configuration Control Register 0 [SYSCFG0] <Address: H'000>

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | CNEN | HSE | — | DRPD | DPRPU | — | — | — | USBE |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| R/W | R | R | R | R | R | R | R | R/W | R/W | R | R/W | R/W | R | R | R | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 to 9 | — | All 0 | R | Nothing is assigned to these bits. Fix these bits to 0. |
| 8 | CNEN | 0 | R/W | This bit prohibits or enables single-ended receiver operation.<br>0: Single-ended receiver operation prohibited<br>1: Single-ended receiver operation enabled |
| 7 | HSE | 0 | R/W | This bit prohibits or enables High-Speed operation.<br>0: High-Speed operation prohibited (Full-Speed)<br>1: High-Speed operation enable (The controller detects the communication speed.) |
| 6 | — | 0 | R | Nothing is assigned to this bit. Fix this bit to 0. |
| 5 | DRPD | 1 | R/W | D+/D- line resistor control<br>Set this bit to 0 to use this module.<br>For details, see Control of USB Data Bus Resistors. |
| 4 | DPRPU | 0 | R/W | D+ line resistor control<br>This bit prohibits or enables D+ line pull-up for the peripheral controller function.<br>For details, see Control of USB Data Bus Resistors.<br>0: Pull Up prohibited<br>1: Pull Up enabled |
| 3 to 1 | — | All 0 | R | Nothing is assigned to these bits. Fix these bits to 0. |
| 0 | USBE | 0 | R/W | USB block operation prohibited<br>This bit prohibits or enables USB block operation.<br>0: USB block operation prohibited<br>1: USB block operation enabled |

**Note:** Data can be written to and read from this register even while the UTMI+PHY clock is stopped. However, if a value is set while the UTMI+PHY clock is stopped, the corresponding function takes effect after the oscillation of UTMI+PHY clock starts.

#### (1) Single-ended receiver operation enable (CNEN) bit

Setting this bit enables the single-ended receiver to operate. This bit is intended to prevent damage by inrush current that can be caused when the single-ended receiver in unattached status goes floating. This bit also allows the LNST bit to be referenced.

Set this bit when VBUS is detected as the result of a VBUS interrupt. Clear this bit when VBUS is removed.

### (2) High-speed operation enable (HSE) bit

Setting this bit enables the high-speed operation. When this bit is 1, this module performs a high-speed or full-speed operation according to the result of reset handshake.

- When the HSE bit is 0, this module performs a full-speed operation.
- When the HSE bit is 1, this module executes the reset handshake protocol, and then automatically performs a high-speed or full-speed operation according to the result of reset handshake.

Rewriting the value of this bit must be done when the DPRPU bit is 0.

### (3) D+/D- line resistor control (DRPD or DPRPU) bit

**Table 32B.4** shows available settings of the resistors for the USB data bus. Use the DPRPU bit to select the USB data bus resistors.

Table 32B.4 Control of USB Data Bus Resistors

| Setting | | Control of USB Data Bus Resistors | | |
|---|---|---|---|---|
| DRPD | DPRPU | D- Line | D+ Line | Remarks |
| 0 | 0 | Open | Open | |
| 0 | 1 | Open | Pull-Up | Specify the settings as shown in the left. |
| 1 | 0 | Pull-Down | Pull-Down | Initial state (When power on reset is canceled) |
| 1 | 1 | Pull-Down | Pull-Up | Setting prohibited |

#### (a) D+ pull-up resistor control (DPRPU) bit

Setting this bit enables this module to notify the USB host of attaching by pulling up the D+ line voltage to 3.3 V. Clearing this bit enables this module to let the USB host know that the device has been detached by stopping pulling up the D+ line voltage.

### (4) USB block operation enable (USBE) bit

This bit enables or disables the operation of the USB block of this module.

When this bit is changed from 1 to 0, this module initializes the bits shown in **Table 32B.5**.

Table 32B.5 Register Bits That Are Initialized by Writing 0 to the USBE Bit

| Register Name | Bit Name |
|---|---|
| SYSSTS0 | LNST |
| DVSTCTR0 | RHST |
| INTSTS0 | DVSQ |
| USBADDR | USBADDR |
| USBREQ | bRequest<br>bmRequestType |
| USBVAL | wValue |
| USBINDX | wIndex |
| USBLENG | wLength |

**Note:** Changing the value of this bit must be done when the SUSPENDM bit is 1 and after the oscillation of UTMI+PHY clock starts.

### 32B.2.3.2 System Configuration Control Register 1 [SYSCFG1] <Address: H'002>

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | BWAIT | | | | | |
| Initial Value | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 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 | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 to 12 | — | All 0 | R | Nothing is assigned to these bits. Fix these bits to 0. |
| 11 to 8 | — | H'F | R | Nothing is assigned to these bits. Fix these bits to 0. |
| 7, 6 | — | All 0 | R | Nothing is assigned to these bits. Fix these bits to 0. |
| 5 to 0 | BWAIT | H'0F | R/W | CPU bus access wait specification<br>These bits specify the number of wait cycles for the access to this module.<br>000000: 0 wait cycles (2 access cycles)<br>:<br>000010: 2 wait cycles (4 access cycles)<br>:<br>000100: 4 wait cycles (6 access cycles)<br>:<br>001111: 15 wait cycles (17 access cycles) (default)<br>:<br>111111: 63 wait cycles (65 access cycles) |

#### (1) CPU bus access wait specification (BWAIT) bits

These bits specify the wait cycles for the access to the HPB.

The following restriction is placed on the cycle of the access to the registers at address H'04 or after of this module: Restriction on wait cycle: The cycle of continuous accesses to registers of this module must be at least 67 ns.

To comply with this restriction, you must control the number of wait cycles with the internal bus clock (P1ϕ) frequency.

The default of wait cycles is 17 clock cycles (maximum limit). Select an optimum setting.

This setting is also applied to accesses to FIFO port registers. The maximum speeds of accesses to FIFO ports are as follows:

MBW = 10 (32-bit access width): Max 60 MBytes/sec
MBW = 01 (16-bit access width): Max 30 MBytes/sec
MBW = 00 (8-bit access width): Max 15 MBytes/sec

## 32B.2.4 System Configuration Status

### 32B.2.4.1 System Configuration Status Register (SYSSTS0) <Address: H'004>

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | LNST | |
| 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 | R | R |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 to 2 | — | See above | R | Nothing is assigned to these bits. Fix these bits to 0. |
| 1, 0 | LNST | All 0 | R | USB Line status monitor<br>The USB line status is displayed.<br>*Note*: See the detailed description. |

#### (1) Line status monitor (LNST) bits

**Table 32B.6** shows the line status of the USB data bus of this module. This module monitors the line status (status of the D+ and D- lines) of the USB data bus in the LNST bits of the SYSSTS0 register.

Referencing the LNST bits must be done only after the USBE bit is set and attaching is performed (the DPRPU bit is set).

Table 32B.6 Line Status of USB Data Bus

| LNST [1] | LNST [0] | Full-Speed operation | High-Speed operation | Chirp operation |
|---|---|---|---|---|
| 0 | 0 | SE0 | Squelch | Squelch |
| 0 | 1 | J State | Unsquelch | Chirp J |
| 1 | 0 | K State | Invalid | Chirp K |
| 1 | 1 | SE1 | Invalid | Invalid |

**Note:** Chirp: State in which high-speed operation is enabled (HSE = 1) and the reset handshake protocol is being executed
Squelch: SE0 or idle state
Unsquelch: High-speed J or high-speed K state
Chirp J: Chirp J State
Chirp K: Chirp K State

## 32B.2.5 USB Signal Control Registers

### 32B.2.5.1 Device State Control Register 0 [DVSTCTR0] <Address: H'008>

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | WKUP | — | — | — | — | — | RHST | | |
| Initial Value | 0 | 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(1) | R | R | R | R | R | R | R | R |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 | — | 0 | R | Reserved. When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 14 to 9 | — | All 0 | R | Nothing is assigned to these bits. Fix these bits to 0. |
| 8 | WKUP | 0 | R/W (1) | Remote wakeup output<br>This bit prohibits or enables Remote wakeup (resume signal output).<br>0: Remote wakeup signal is not output.<br>1: Remote wakeup signal is output. |
| 7 to 3 | — | All 0 | R | Nothing is assigned to these bits. Fix these bits to 0. |
| 2 to 0 | RHST | All 0 | R | Reset handshake<br>This bit indicates the reset handshake status.<br>*Note*: See the detailed description. |

#### (1) Remote wakeup (resume signal output) enable (WKUP) bit

When this bit is set, this module outputs the remote wakeup signal to the USB.

This module manages the time of remote wakeup signal output. When the WKUP bit is set, this module outputs the K state for 10 ms, and then clears the WKUP bit.

The USB Specification requires the USB idle state to be retained for at least 5 ms before the remote wakeup signal is sent. Therefore, even if the WKUP bit is set immediately after the suspended state is detected, this module waits for 2 ms, and then outputs the K state.

Writing 1 to the WKUP bit must be done only when the device is in the suspended state (DVSQ = 1xx) and remote wakeup is allowed by the USB host.

When setting the WKUP bit, do not stop the internal clock even if the device is in the suspended state. (Write 1 to the WKUP bit when the SUSPM bit is 1.)

When the WKUP bit is set at a transition to the L1 state, this module outputs the K state for 50 µs, and then clears the WKUP bit. In the L1 state, setting the WKUP bit must be done only when the DVSQ[4] bit is 1.

### (2) Reset handshake status (RHST) bits

This module outputs the result of reset handshake to this bit. **Table 32B.7** lists the results of reset handshake.

Table 32B.7 Reset Handshake Status

| Bus State | Value of RHST Bit |
|---|---|
| Powered or disconnected state | 000 |
| Reset handshake in process | 100 |
| Full-speed connection | 010 |
| High-speed connection | 011 |

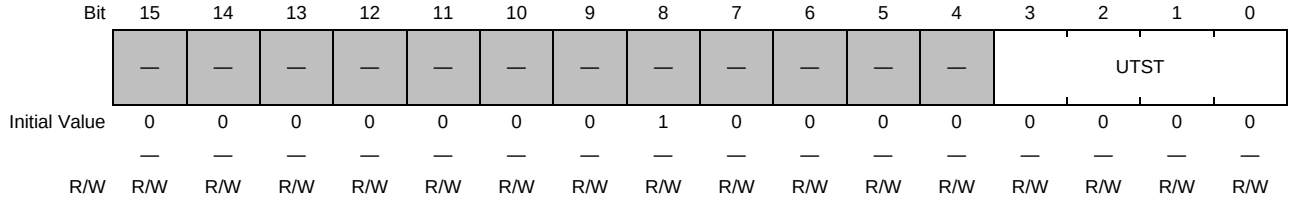
If the HSE bit is 1, the RHST bits indicate 100 when this module detects a USB bus reset. Then, after this module has output Chirp K, these bits indicate 011 when this module detects Chirp JK from the USB host three times. If the status is not fixed to High-Speed within 2.5 ms after Chirp K is output, these bits indicate 010.

If the HSE bit is 0, the RHST bits indicate 010 when this module detects a bus reset.

After this module has detected a USB reset, a DVST interrupt occurs when the value of the RHST bits is fixed to 010 or 011.

## 32B.2.6 Test Mode Register

### 32B.2.6.1 USB Test Mode Register [TESTMODE] <Address: H'00C>




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 | — | 0 | R | Nothing is assigned to this bit. Fix this bit to 0. |
| 14 | — | 0 | R/W | Reserved. When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 13, 12 | — | All 0 | R | Nothing is assigned to these bits. Fix these bits to 0. |
| 11 | — | 0 | R/W | Reserved. When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 10 to 4 | — | H'10 | R | Nothing is assigned to these bits. Fix these bits to 0. |
| 3 to 0 | UTST | All 0 | R/W | Test mode<br>See the detailed description. |

#### (1) Test mode (UTST) bits

When a value is written to these bits, this module outputs a USB test signal during high-speed operation.

**Table 32B.8** lists the test modes of this module.

Table 32B.8 List of test mode operation

| Test mode | Value of UTST Bits |
|---|---|
| Normal operation | 0000 |
| Test_J | 0001 |
| Test_K | 0010 |
| Test_SE0_NAK | 0011 |
| Test_Packet | 0100 |
| Test_Force_Enable | — |
| Reserved | 0101 to 0111 |

Write a value to these bits according to the SetFeature request sent from the USB host during high-speed communication. When these bits contain a value from 0001 to 0100, this module does not enter the suspended state.

To perform a normal USB communication after setting a test mode, perform Power on reset.

## 32B.2.7 FIFO Port Registers

### 32B.2.7.1 CFIFO Port Register [CFIFO] <Address: H'014>

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | FIFOPORT (Low) | | | | | | | | | | | | | | | |
| 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 |

### 32B.2.7.2 CFIFO Port Register [CFIFO] <Address: H'016>

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | FIFOPORT (High) | | | | | | | | | | | | | | | |
| 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 | All 0 | R/W | FIFO port<br>These bits are accessed to read received data from the FIFO buffer or to write send data to the FIFO buffer. |

#### (1) FIFO port control bits (for FIFOPORT)

The send/receive buffer memory of this module has a FIFO structure (FIFO buffer). Use FIFO port registers to access the FIFO buffer. The FIFO port consists of the port register (CFIFO) to read data from and write data to the FIFO buffer, the register (CFIFOSEL) to select the pipe to be allocated to the FIFO port, and the control register (CFIFOCTR).

Individual FIFO ports have the following features:

1. The CFIFO port must be used to access the FIFO buffer through the DCP.
2. When functions specific to FIFO ports are used, the pipe number (selected pipe) specified in the CURPIPE bits cannot be changed.
3. The registers configured for a FIFO port do not affect any other FIFO ports.
4. The FIFO buffer memory can be accessed by either the CPU or SIE. Access by the CPU is not possible while the SIE has the right of access to the FIFO buffer memory.

### (2) FIFO port bits (CFIFO)

When one of these registers is accessed, this module accesses the FIFO buffer allocated to the pipe number specified in the CURPIPE bits in the corresponding pipe select register (CFIFOSEL).

These registers can be accessed only when the FRDY bit of the respective control registers (CFIFOCTR) is 1 (or when the UCL_Dx_DREQ output is asserted by this module).

The valid bits of these registers vary depending on the values of the NBW and BIGEND bits. The valid bits are shown in **Table 32B.9** to **Table 32B.11**.

Table 32B.9 Endian Operation in 32-Bit Access (When MBW = 10)

| BIGEND | b31 to b24 | b23 to b16 | b15 to b8 | b7 to b0 |
|---|---|---|---|---|
| 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 32B.10 Endian Operation in 16-Bit Access (When MBW = 01)

| BIGEND | b31 to b24 | b23 to b16 | b15 to b8 | b7 to b0 |
|---|---|---|---|---|
| 0 | Writing: invalid<br>Reading: prohibited*1 | | even-numbered address | odd-numbered address |
| 1 | even-numbered address | odd-numbered address | Writing: invalid<br>Reading: prohibited*1 | |

Table 32B.11 Endian Operation in 8-Bit Access (When MBW = 00)

| BIGEND | b31 to b24 | b23 to b16 | b15 to b8 | b7 to b0 |
|---|---|---|---|---|
| 0 | Writing: invalid<br>Reading: prohibited*1 | | | Writing: valid<br>Reading: valid |
| 1 | Writing: valid<br>Reading: valid | Writing: invalid<br>Reading: prohibited*1 | | |

Note 1. Reading words or bytes from an invalid register is prohibited.

### 32B.2.7.3 CFIFO Port Select Register [CFIFOSEL] <Address: H'020>

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RCNT | REW | — | — | MBW | | — | BIGEND | — | — | ISEL | — | CURPIPE | | | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| R/W | R/W | R(0)/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<br>This bit specifies DTLN read mode for the CFIFOCTR register.<br>0: Clears the DTLN bits when all received data is read.<br>1: Decrements the value of the DTLN bits each time received data is read. |
| 14 | REW | 0 | R(0)/W | Buffer pointer rewind<br>Set this bit to 1 to rewind the buffer pointer.<br>0: Does not rewind the buffer pointer.<br>1: Rewinds the buffer pointer. |
| 13, 12 | — | All 0 | R | Nothing is assigned to these bits. Fix these bits to 0. |
| 11, 10 | MBW | 0 | R/W | CFIFO port access bit width<br>This bit specifies the bit width for access to the CFIFO port.<br>00: 8-bit width<br>01: 16-bit width<br>10: 32-bit width<br>11: Setting prohibited |
| 9 | — | 0 | R | Nothing is assigned to this bit. Fix this bit to 0. |
| 8 | BIGEND | 0 | R/W | FIFO port byte endian control<br>This bit specifies the byte endian of the CFIFO port.<br>0: Little endian<br>1: Big endian |
| 7, 6 | — | All 0 | R | Nothing is assigned to these bits. Fix these bits to 0. |
| 5 | ISEL | 0 | R/W | FIFO port access direction with DCP selected<br>This bit specifies the FIFO port access direction when DCP is selected for the CURPIPE bits.<br>0: Selects reading of buffer memory.<br>1: Selects writing of buffer memory. |
| 4 | — | 0 | R | Nothing is assigned to this bit. Fix this bit to 0. |
| 3 to 0 | CURPIPE | All 0 | R/W | FIFO port access pipe specification<br>This bit specifies the pipe number used for accessing the CFIFO port.<br>0000: DCP<br>0001: PIPE1<br>0010: PIPE2<br>↓<br>1000: PIPE8<br>1001: PIPE9<br>↓<br>1110: PIPE14<br>1111: PIPE15 |

#### (1) Read count mode (RCNT) bit

When this bit is 0, this module clears the DTLN bits in the CFIFOCTR register when all received data has been read from the FIFO buffer allocated to the pipe (selected pipe) specified in the CURPIPE bits (or, in the case of double-buffer configuration, when all relieved data has been read from one buffer).

With this bit is 1, this module decrements the value of the DTLN bits in the CFIFOCTR register each time received data is read from the FIFO buffer allocated to the specified pipe.

#### (2) Buffer pointer rewind (REW) bit

When this bit is set during data reading from the FIFO buffer while the selected pipe is in the receiving direction, reading can be restarted from the first data in the FIFO buffer (or, in the case of a double-buffer configuration, rereading can be started from the first data in the FIFO buffer being read).

Do not set this bit at the same time as changing the value of the CURPIPE bits. Before setting this bit, always check that the FRDY bit is 1.

If you want to redo writing to the FIFO buffer from the first data in the FIFO buffer when the selected PIPE is in the sending direction, use the BCLR bit.

#### (3) CFIFO port access bit width (MBW) bits

These bits are used to specify the bit width for the access to the CFIFO port.

If you start reading after setting a value in these bits when the pipe specified in the CURPIPE bits is in the receiving direction, do not change the value of these bits until data is all read.

Also, to set a value in these bits when the pipe specified in the CURPIPE bits is in the receiving direction, temporarily change the original value of the CURPIPE bits to a different value, and then set the values of the CURPIPE and MBW bits at the same time.

For how to change the value of the CURPIPE bits, see the description of the CURPIPE bits.

When the pipe specified in the CURPIPE bits is in the sending direction, you cannot change the bit width from 8 bits to 16 bits or 32 bits or from 16 bits to 32 bits while writing to the buffer memory is in process.

Even with the 16-bit width or 32-bit width setting, you can write data also to odd bytes by using byte access control.

#### (4) FIFO port byte endian control (BIGEND) bit

This bit is used to specify the byte endian of the CFIFO port.

For details see **Section 32B.2.7.2(1), FIFO port control bits (for FIFOPORT)**.

#### (5) FIFO port access direction with DCP selected (ISEL) bit

To change the value of this bit when the specified pipe is DCP, write a value to this bit, read the bit, and then check that the written value is the same as the read value before proceeding to the next processing.

If the value of the bit is changed in the middle of an access to the FIFO buffer, the access is held. Therefore, the same access can be resumed after these bits are restored to the original value.

Set this bit simultaneously with setting of the CURPIPE bits.

### (6) FIFO port access pipe specification (CURPIPE) bits

These bits are used to specify the pipe number of the pipe through which to read or write data via the CFIFO port.

If you change the value of these bits, write a desired value to these bits, read these bits, and then check that the written value is the same as the read value before proceeding to the next processing.

If the value of these bits is changed in the middle of an access to the FIFO buffer, the access is held. Therefore, the same access can be resumed after these bits are restored to the original value.

### 32B.2.7.4 D0FIFO Port Select Register [D0FIFOSEL] <Address:H'028>
### D1FIFO Port Select Register [D1FIFOSEL] <Address:H'02C>

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RCNT | REW | DCLRM | DREQE | MBW | | — | — | — | — | — | — | CURPIPE | | | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| R/W | R/W | R(0)/W | R/W | R/W | 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 | RCNT | 0 | R/W | Read count mode<br>This bit specifies Dx_FIFOCTR DTLN read mode.<br>0: Clears the DTLN bits when all received data is read.<br>1: Decrements the value of the DTLN bits each time received data is read. |
| 14 | REW | 0 | R(0)/W | Buffer pointer rewind<br>Set this bit to 1 to rewind the buffer pointer.<br>0: Does not rewind the buffer pointer.<br>1: Rewinds the buffer pointer. |
| 13 | DCLRM | 0 | R/W | Automatic buffer memory clear mode after reading data through the specified pipe<br>This bit prohibits or enables automatic buffer memory clear after data is read through the specified pipe.<br>0: Automatic FIFO buffer clear prohibited<br>1: Automatic FIFO buffer clear enabled |
| 12 | DREQE | 0 | R/W | UCL_Dx_DREQ output enable<br>This bit prohibits or enables the output of the UCL_Dx_DREQ signal.<br>0: Output prohibited<br>1: Output enabled |
| 11, 10 | MBW | All 0 | R/W | DxFIFO port access bit width<br>This bit specifies the bit width for access to the DxFIFO port.<br>00: 8-bit width<br>01: 16-bit width<br>10: 32-bit width<br>11: Setting prohibited |
| 9 to 4 | — | All 0 | R | Nothing is assigned to these bits. Fix these bits to 0. |
| 3 to 0 | CURPIPE | All 0 | R/W | FIFO port access pipe specification<br>0000: No specification<br>0001: PIPE1<br>0010: PIPE2<br>↓<br>1000: PIPE8<br>1001: PIPE9 |

#### (1) Read count mode (RCNT) bit

When this bit is 0, this module clears the DTLN bits in the DxFIFOCTR register when all received data has been read from the FIFO buffer allocated to the pipe (selected pipe) specified in the CURPIPE bits (or, in the case of double-buffer configuration, when all relieved data has been read from one buffer).

With this bit is 1, this module decrements the value of the DTLN bits in the DxFIFOCTR register each time received data is read from the FIFO buffer allocated to the specified pipe.

#### (2) Buffer pointer rewind (REW) bit

When this bit is set during data reading from the FIFO buffer while the selected pipe is in the receiving direction, reading can be restarted from the first data in the FIFO buffer (or, in the case of a double-buffer configuration, rereading can be started from the first data in the FIFO buffer being read).

Do not set this bit at the same time as changing the value of the CURPIPE bits. Before setting this bit, always check that the FRDY bit is 1.

If you want to redo writing to the FIFO buffer from the first data in the FIFO buffer when the selected PIPE is in the sending direction, use the BCLR bit.

#### (3) Automatic FIFO buffer clear enable (DCLRM) bit

This bit is used to enable or disable the mode to automatically clear the FIFO buffer memory after reading data from the specified pipe. With this bit set, this module performs the “BCLR = 1” processing on the FIFO buffer, if a zero-length packet is received when the FIFO buffer allocated to the specified pipe is empty or if data reading ends because a short packet is received when the BFRE bit is 1.

Always clears this bit when you use this module with the BRDYM bit set.

#### (4) UCL_Dx_DREQ output enable (DREQE) bit

This bit is used to enable or disable the output of the UCL_Dx_DREQ signal.

When enabling the UCL_Dx_DREQ signal, set this bit always after setting a value in the CURPIPE bits. When changing the value of the CURPIPE bits, change the value always after clearing this bit.

#### (5) DxFIFO port access bit width (MBW) bits

These bits are used to specify the bit width for the access to the DxFIFO port. For details, see **Section 32B.2.7.3(3), CFIFO port access bit width (MBW) bits**.

#### (6) FIFO port access pipe specification (CURPIPE) bits

These bits are used to specify the pipe number through which to read or write data via the DxFIFO port.

If you change the value of these bits, write a desired value to these bits, read these bits, and then check that the written value is the same as the read value before proceeding to the next processing.

Do not specify the same pipe number for the CURPIPE bits in the CFIFOSEL, D0FIFOSEL, and D1FIFOSEL registers. If the value of these bits is changed in the middle of an access to the FIFO buffer, the access is held. Therefore, the same access can be resumed after these bits are restored to the original value.

### 32B.2.7.5 CFIFO Port Control Register [CFIFOCTR] <Address: H'022>
### D0FIFO Port Control Register [D0FIFOCTR] <Address: H'02A>
### D1FIFO Port Control Register [D1FIFOCTR] <Address: H'02E>

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | BVAL | BCLR | FRDY | — | DTLN | | | | | | | | | | | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| R/W | R/W(1) | R(0)/W(1) | 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<br>Specify 1 for this bit when writing to the FIFO buffer on the CPU side through the pipe specified in CURPIPE (current pipe) ends.<br>0: Invalid<br>1: Writing complete |
| 14 | BCLR | 0 | R(0)/W(1) | CPU buffer clear<br>Specify 1 for this bit to clear the FIFO buffer on the CPU side of the current pipe.<br>0: Invalid<br>1: CPU buffer memory clear |
| 13 | FRDY | 0 | R | FIFO port ready<br>This bit indicates whether the FIFO port can be accessed.<br>0: The FIFO port cannot be accessed.<br>1: The FIFO port can be accessed. |
| 12 | — | 0 | R | Nothing is assigned to this bit. Fix this bit to 0. |
| 11 to 0 | DTLN | All 0 | R | Receive data length<br>These bits indicate the length of receive data. |

#### (1) Buffer memory valid flag (BVAL)

When the pipe (selected pipe) specified in the CURPIPE bits is in the sending direction, this bit must be set in the cases descried below. This module switches the FIFO buffer from the CPU side to the SIE side to enable data sending.

(1) To send short packets, set this bit when data writing ends.

(2) To send zero length packets, set this bit before writing data to the FIFO buffer.

(3) Set this bit after writing, to the pipe in continuous transfer mode, the data of which the size is a positive integral multiple of the maximum packet size and less than the buffer size.

When the maximum packet size of data is written to the pipe in non-continuous transfer mode, this module sets this bit to switch the FIFO buffer from the CPU side to the SIE side and enable data sending.

When this bit and the BCLR bit are set at the same time when the specified pipe is in the sending direction, this module clears the data that has been written so far and enables zero-length packets to be sent.

Setting this bit must be done only when the FRDY bit in the corresponding port control register is 1. If you want to check the value of the FRDY bit after setting this bit, wait at least 80 ns after setting this bit, and then reference the FRDY bit. Do not set this bit when the specified pipe is in the receiving direction.

### (2) CPU buffer clear (BCLR) bit

When this bit is set, this module clears the FIFO buffer on the CPU side among the FIFO buffers allocated to the specified pipe.

Even if the two FIFO buffers in a double-buffer configuration are allocated to the specified pipe and the both buffers can be read, this module clears only one of the two buffers.

If this bit is set when the specified pipe is the DCP, this module clears the FIFO buffer regardless of whether the FIFO buffer is on the CPU side or it is on the SIE side. To clear the buffer on the SIE side, set this bit always after setting the PID bits for DCP to “NAK”.

When the specified pipe is other than the DCP, setting this bit must be done only when the FRDY bit in the corresponding port control register is 1. If you want to check the value of the FRDY bit after setting this bit, wait at least 80 ns after setting this bit, and then reference the FRDY bit.

### (3) FIFO port ready (FRDY) bit

This bit indicates whether the FIFO port can be accessed from the CPU. This bit is operated by this module.

In the cases described below, even when this module sets this bit, data cannot be read from the FIFO port because the FIFO buffer does not contain data to be read. In these cases, set the BCLR bit to clear the FIFO buffer to enable the next sending and receiving of data.

(1) A zero-length packet has been received while the FIFO buffer allocated to the specified pipe is empty.

(2) A short packet has been received and data reading has ended while the BFRE bit is 1.

### (4) Receive data length (DTLN) bits

These bits indicate the length of receive data. These bits are operated by this module. During reading of the FIFO buffer, the value of these bits varies depending on the value of the RCNT bit as described below.

(1) When the RCNT bit is 0:
This module indicates a receive data length by using these bits until the CPU ends reading all received data from one FIFO buffer.
When the BFRE bit is 1, this module retains the receive data length until the BCLR bit is set even if reading of received data has ended.

(2) When the RCNT bit is 1:
This module decrements the value of these bits each time the CPU reads data.
(The value is decremented by 1 when the MBW bits are 00, by 2 when the MBW bits are 01 or by 4 when the MBW bits are 10.)

When the CPU ends reading from one FIFO buffer, this module clears these bits. If reading of one of the FIFO buffers in a double-buffer configuration, however, ends before reading of received data from the other FIFO buffer ends, these bits indicate the receive data length for the other FIFO buffer at the end of reading from the FIFO buffer of which reading ends earlier.

When these bits are read during reading of the FIFO buffer when the RCNT bit is 1, this module updates the value of these bits within 150 ns after a cycle of read access to the corresponding FIFO port.

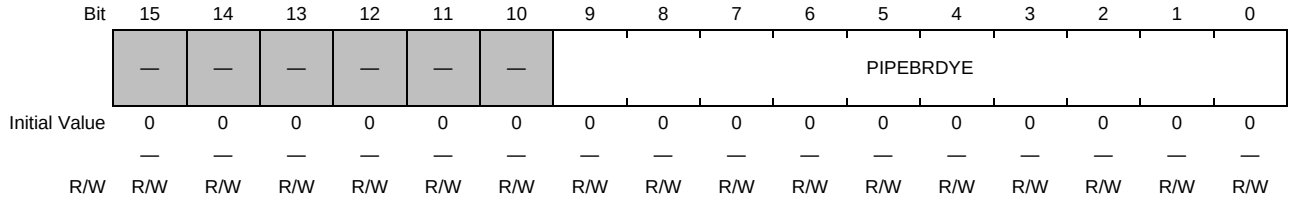
## 32B.2.8 Interrupt Enable Registers (INTENBx, BRDYENB, NRDYENB, BEMPENB)

### 32B.2.8.1 Interrupt Enable Register 0 [INTENB0] <Address: H'030>

| 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 | 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 Interrupt Enable<br>This bit prohibits or enables a USB interrupt when a VBINT interrupt is detected.<br>0: Interrupt output prohibited<br>1: Interrupt output enabled |
| 14 | RSME | 0 | R/W | Frame number update interrupt enable<br>Resume interrupt enable<br>This bit prohibits or enables a USB interrupt when a RESM interrupt is detected.<br>0: Interrupt output prohibited<br>1: Interrupt output enabled |
| 13 | SOFE | 0 | R/W | This bit prohibits or enables a USB interrupt when a SOF interrupt is detected.<br>0: Interrupt output prohibited<br>1: Interrupt output enabled |
| 12 | DVSE | 0 | R/W | Device state transition interrupt enable<br>This bit prohibits or enables a USB interrupt when a DVST interrupt is detected.<br>0: Interrupt output prohibited<br>1: Interrupt output enabled |
| 11 | CTRE | 0 | R/W | Control transfer stage transition interrupt enable<br>This bit prohibits or enables a USB interrupt when a CTRT interrupt is detected.<br>0: Interrupt output prohibited<br>1: Interrupt output enabled |
| 10 | BEMPE | 0 | R/W | Buffer empty interrupt enable<br>This bit prohibits or enables a USB interrupt when a BEMP interrupt is detected.<br>0: Interrupt output prohibited<br>1: Interrupt output enabled |
| 9 | NRDYE | 0 | R/W | Buffer not ready response interrupt enable<br>This bit prohibits or enables a USB interrupt when an NRDY interrupt is detected.<br>0: Interrupt output prohibited<br>1: Interrupt output enabled |
| 8 | BRDYE | 0 | R/W | Buffer ready interrupt enable<br>This bit prohibits or enables a USB interrupt when a BRDY interrupt is detected.<br>0: Interrupt output prohibited<br>1: Interrupt output enabled |
| 7 to 0 | — | All 0 | R | Nothing is assigned to these bits. Fix these bits to 0. |

### 32B.2.8.2 BRDY Interrupt Enable Register [BRDYENB] <Address: H'036>



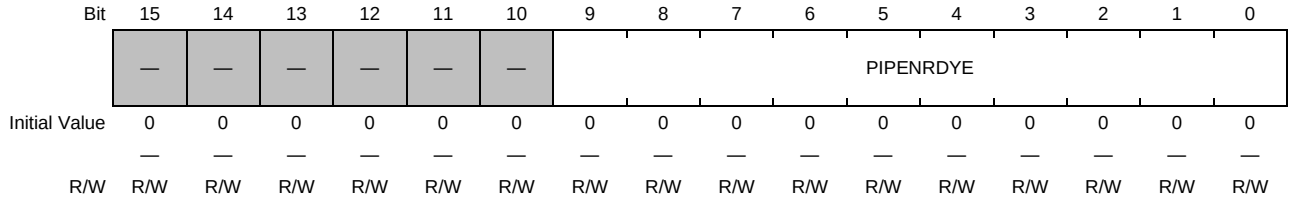

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 to 10 | — | All 0 | R/W | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 9 to 0 | PIPEBRDYE | All 0 | R/W | Pipe BRDY interrupt enable<br>These bits prohibit or enable the BRDY bit to be set when a BRDY interrupt to a pipe is detected.<br>0: Interrupt output prohibited<br>1: Interrupt output enabled |

#### (1) Pipe BRDY interrupt enable (PIPEBRDYE) bits

When this module detects a BRDY interrupt to the pipe for which is set 1 in this register, this module sets the corresponding PIPEBRDY bit in the BRDYSTS register, sets the BRDY bit in the INTSTS0 register, and asserts the interrupt.

When at least one of the PIPEBRDY bits in the BRDYSTS register is 1 and software changes the corresponding interrupts enable bit in this register changes from 0 to 1, this module asserts the interrupt.

### 32B.2.8.3 NRDY Interrupt Enable Register [NRDYENB] <Address: H'038>



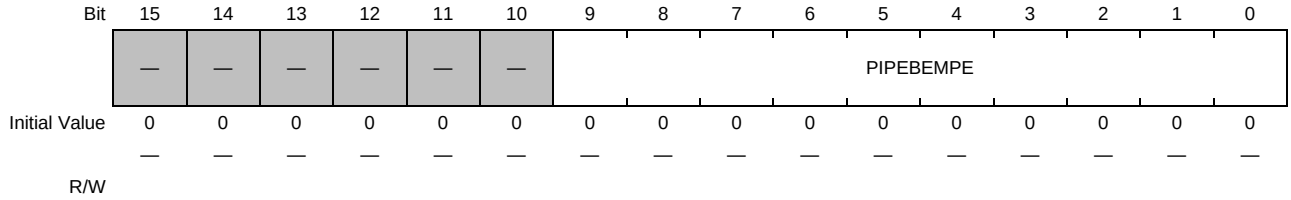

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 to 10 | — | All 0 | R/W | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 9 to 0 | PIPENRDYE | All 0 | R/W | Pipe NRDY interrupt enable<br>These bits prohibit or enable the NRDY bit to be set when a BRDY interrupt to a pipe is detected.<br>0: Interrupt output prohibited<br>1: Interrupt output enabled |

#### (1) Pipe NRDY interrupt enable (PIPENRDYE) bits

When this module detects a BRDY interrupt to the pipe for which is set 1 in this register, this module sets the corresponding PIPENRDY bit in the NRDYSTS register, sets the NRDY bit in the INTSTS0 register, and asserts the interrupt.

When at least one of the PIPENRDY bits in the NRDYSTS register is 1 and the corresponding interrupts enable bit in this register changes from 0 to 1, this module asserts the interrupt.

### 32B.2.8.4 BEMP Interrupt Enable Register [BEMPENB] <Address: H'03A>




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 to 10 | — | All 0 | R/W | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 9 to 0 | PIPEBEMPE | All 0 | R/W | Pipe BEMP interrupt enable<br>These bits prohibit or enable the BEMP bit to be set when a BRDY interrupt to a pipe is detected<br>0: Interrupt output prohibited<br>1: Interrupt output enabled |

**Note:** Bit numbers correspond to pipe numbers.

#### (1) Pipe BEMP interrupt enable (PIPEBEMPE) bits

When this module detects a BEMP interrupt to the pipe for which is set 1 in this register, this module sets the corresponding PIPEBEMP bit in the BEMPSTS register, sets the BEMP bit in the INTSTS0 register, and asserts the interrupt.

When at least one of the PIPEBEMP bits in the BEMPSTS register is 1 and the corresponding interrupts enable bit in this register changes from 0 to 1, this module asserts the interrupt.

## 32B.2.9 SOF Control Register

### 32B.2.9.1 SOF Pin Configuration Register [SOFCFG] <Address: H'03C>

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | BRDYM | — | — | SOFM | | — | — |
| 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/W | R | R | R/W | R/W | R | R |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 to 7 | — | All 0 | R | Nothing is assigned to these bits. Fix these bits to 0. |
| 6 | BRDYM | 0 | R/W | PIPEBRDY interrupt status clear timing<br>This bit specifies the timing at which the PIPEBRDY interrupt is to be cleared.<br>0: Software clears the status.<br>1: Hardware clears the status by reading from or writing to the FIFO buffer.<br>This bit can be set only during initialization (before communication).<br>The setting cannot be changed after communication. |
| 5, 4 | — | All 0 | R | Nothing is assigned to these bits. Fix these bits to 0. |
| 3, 2 | SOFM | All 0 | R/W | SOF function setting<br>These bits are used to select SOF pulse output mode.<br>00: SOF output disabled<br>01: SOF output in units of 1 ms<br>10: µSOF output in units of 125 µs<br>11: Reserved |
| 1, 0 | — | All 0 | R | Nothing is assigned to these bits. Fix these bits to 0. |

## 32B.2.10 Interrupt Status

### 32B.2.10.1 Interrupt Status Register 0 [INTSTS0] <Address: H'040>

| 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 | | | VALID | CTSQ | | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | — | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | — | — | — | 1 | — | — | — | — | — | 0 | 0 | 1 | — | — | — | — |
| R/W | R/W(0) | R/W(0) | R/W(0) | R/W(0) | R/W(0) | R | R | R | R | R | R | R | R/W(0) | R | R | R |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 | VBINT | 0 | R/W(0) | Change detection interrupt status<br>This bit indicates the VBUS change detection interrupt status.<br>0: No VBUS interrupt is generated.<br>1: A VBUS interrupt is generated. |
| 14 | RESM | 0 | R/W(0) | Resume interrupt status<br>This bit indicates the resume detection interrupt status.<br>0: No resume interrupt is generated.<br>1: A resume interrupt is generated. |
| 13 | SOFR | 0 | R/W(0) | Frame number update interrupt status<br>This bit indicates the frame number update interrupt status.<br>0: No SOF interrupt is generated.<br>1: A SOF interrupt is generated. |
| 12 | DVST | 0 | R/W(0) | Device state transition interrupt status<br>This bit indicates the device state transition interrupt.<br>0: No device state transition interrupt is generated.<br>1: A device state transition interrupt is generated. |
| 11 | CTRT | 0 | R/W(0) | Control transfer stage transition interrupt status<br>This bit indicates the status of a control transfer stage transition interrupt.<br>0: No control transfer stage transition interrupt is generated.<br>1: A control transfer stage transition interrupt is generated. |
| 10 | BEMP | 0 | R | BEMP interrupt status<br>This bit indicates the BEMP interrupt status.<br>0: No BEMP interrupt is generated.<br>1: A BEMP interrupt is generated. |
| 9 | NRDY | 0 | R | NRDY interrupt status<br>This bit indicates the NRDY interrupt status.<br>0: No NRDY interrupt is generated.<br>1: An NRDY interrupt is generated. |
| 8 | BRDY | 0 | R | BRDY interrupt status<br>This bit indicates the BRDY interrupt status.<br>0: No BRDY interrupt is generated.<br>1: A BRDY interrupt is generated. |
| 7 | VBSTS | 0 | R | VBUS input status<br>This bit indicates the VBUS pin input status.<br>0: The VBUS pin is at the low level.<br>1: The VBUS pin is at the high level. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 6 to 4 | DVSQ | All 0 | R | Device state<br>These bits indicate the device state.<br>000: Powered state<br>001: Default state<br>010: Address state<br>011: Configured state<br>1xx: Suspended state |
| 3 | VALID | 0 | R/W(0) | USB request reception<br>This bit indicates whether USB request reception is detected.<br>0: Not detected<br>1: A setup packet is received. |
| 2 to 0 | CTSQ | All 0 | R | Control transfer stage<br>These bits indicate the control transfer stage.<br>000: Idle or setup stage<br>001: Control reading data stage<br>010: Control reading status stage<br>011: Control writing data stage<br>100: Control writing status stage<br>101: Control writing (No Data) status stage<br>110: Control transfer sequence error<br>111: Reserved |

**Note:** When you want to clear the status indicated by the VBINT, RESM, SOFR, DVST, or CTRT bit, write 0 to only the bit to be cleared and 1 to other bits. Do not write 0 to any status bit that is currently 0.

**Note:** This module detects a status change indicated by the VBINT or RESM bit in this register even while the clock is stopped (the SUSPM bit is 0), and reports an interrupt corresponding to the status bit if the interrupt is enabled. Clearing the interrupt status must be done after the clock enabled.

### (1) VBUS change interrupt status (VBINT) bit

This module sets this bit when it detects a change of the level of the VBUS pin input (from the high level to low level, or vice versa). This module indicates the level of the VBUS pin input by the VBSTS bit. When a VBINT interrupt occurs, read the VBSTS bit several times to check for consistency and remove chattering.

### (2) Resume interrupt status (RESM) bit

This module sets this bit when it is in the suspended state (DVSQ bits are 1XXb) and detects a falling edge of the signal at the DP pin.

### (3) Frame number update interrupt status (SOFR) bit

This module sets this bit under the following conditions:

This module sets this bit when the frame number is updated. (The frame number update interrupt is monitored at intervals of 1 ms.)

This module detects an SOFR interrupt based on SOF interpolation even when an SOF packet from the USB host is damaged.

### (4) Device state transition interrupt status (DVST) bit

When this module detects a change of the device state, this module updates the value of the DVSQ bits, and sets this bit. When a device state transition interrupt occurs, clear the interrupt status before this module detects the next device state transition.

#### (5) Control transfer stage transition interrupt status (CTRT) bit

When this module detects a transition of control transfer stage, this module updates the value of the CTSQ bits and sets this bit.

When a control transfer stage transition interrupt occurs, clear the interrupt status before this module detects the next transition of control transfer stage.

#### (6) Buffer empty interrupt status (BEMP) bit

This module sets this bit when at least one of the PIPEBEMP bits in the BEMPSTS register corresponding to the pipes for which the PIPEBEMPE bit in the BEMPENB register is set (that is, when this module detects a BEMP interrupt to at least one of the pipes for which is enabled BEMP interrupt notification).

For the conditions to assert the PIPEBEMP status signal, see the description of the BEMPSTS register.

This module clears this bit when writes 0 to all the PIPEBEMP bits corresponding to the pipes for which a BEMP interrupt has been enabled by setting the PIPEBEMPE bit.

Cannot clear this bit even by writing 0.

#### (7) Buffer not-ready interrupt status (NRDY) bit

This module sets this bit when at least one of the PIPENRDY bits in the BNRDYSTS register corresponding to the pipes for which the PIPENRDYE bit in the NRDYENB register is set (that is, when this module detects an NRDY interrupt to at least one of the pipes for which is enabled NRDY interrupt notification).

For the conditions to assert the PIPENRDY status signal, see the description of the NRDYSTS register.

This module clears this bit when writes 0 to all the PIPENRDY bits corresponding to the pipes for which a NRDY interrupt has been enabled by setting the PIPENRDYE bit.

Software cannot clear this bit even by writing 0.

#### (8) Buffer ready interrupt status (BRDY) bit

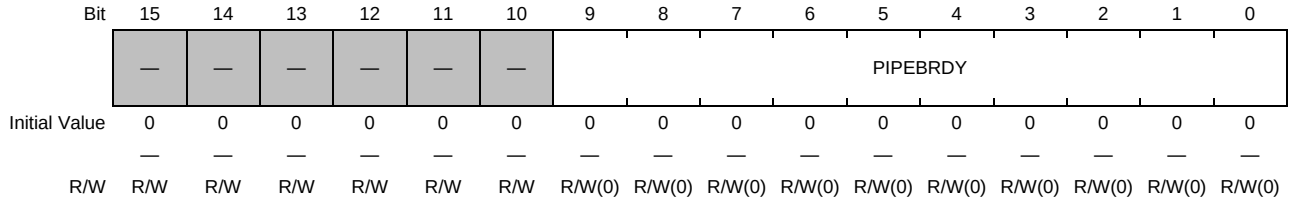
This module sets this bit when at least one of the PIPEBRDY bits in the BRDYSTS register corresponding to the pipes for which the PIPEBRDYE bit in the BRDYENB register is set (that is, when this module detects an BRDY interrupt to at least one of the pipes for which is enabled BRDY interrupt notification).

For the conditions to assert the PIPEBRDY status signal, see the description of the BRDYSTS register.

This module clears this bit when writes 0 to all the PIPEBRDY bits corresponding to the pipes for which a BRDY interrupt has been enabled by setting the PIPEBRDYE bit.

Software cannot clear this bit even by writing 0.

### 32B.2.10.2 BRDY Interrupt Status Register [BRDYSTS] <Address: H'046>




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 to 10 | — | All 0 | R/W | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 9 to 0 | PIPEBRDY | All 0 | R/W(0) | Pipe BRDY interrupt status<br>These bits indicate the BRDY interrupt status of each pipe.<br>0: No interrupt is generated.<br>1: An interrupt is generated. |

**Note:** Bit numbers correspond to pipe numbers.

**Note:** To clear the interrupt status indicated by a bit in this register when the BRDYM bit is 0, write 0 to only the bit to be cleared and 1 to other bit.

**Note:** When the BRDYM bit is 0, clearing the BRDY interrupt status must be done always before the next access to the FIFO.

#### (1) Pipe BRDY interrupt status (PIPEBRDY) bit

When this module detects a BRDY interrupt to a pipe, this module sets the corresponding PIPEBRDY bit in the BRDYSTS register. At that time, if already set the corresponding bit in the BRDYENB register, this module sets the BRDY bit in the INTSTS0 register, and asserts the interrupt.

Conditions to generate and clear a BRDY interrupt vary depending on the values of the BRDYM bit, and the BFRE bit for each pipe.

##### (a) BRDYM = 0 and BFRE = 0

When the BRDYM and BFRE bits are 0, the BRDY interrupt is generated to indicate that the FIFO port has become ready for access.

Under the conditions described below, this module generates an internal BRDY interrupt request trigger, and sets the PIPEBRDY bit corresponding to the pipe for which the request trigger is generated.

1. For the pipe in the sending direction

(a) When the DIR bit changes from 0 to 1

(b) When this module has ended sending packets through a pipe when data writing by the CPU to the FIFO buffer allocated to that pipe is disabled (when the value read from the BSTS bit is 0)
In continuous transfer mode, a request trigger is generated when data has all been sent from one FIFO buffer.

(c) When, in a double-buffer configuration, one FIFO buffer is empty when writing to the other FIFO buffer has ended
If sending to one FIFO buffer has ended during writing to the other FIFO buffer, no request trigger is generated until the ongoing writing to the other FIFO buffer ends.

(d) When this module flushes the FIFO buffer allocated to the pipe of which the transfer type is isochronous.

(e) When the state of the FIFO buffer is changed from the write-disabled state to the write-enabled state by writing 1 to the ACLRM bit

No request trigger is generated when the pipe is the DCP (in other words, when data is sent by a control transfer).

2. For the pipe in the receiving direction

(a) When this module has ended receiving packets through a pipe and reading from the FIFO buffer is enabled when data reading by the CPU from the FIFO buffer allocated to that pipe is disabled (when the value read from the BSTS bit is 0)
No request trigger is generated for the transaction that involves a data PID mismatch.
In continuous transmission/reception mode, no request trigger is generated when the data size is the maximum packet size and the FIFO buffer still has a free space.
If a short packet is received, a request trigger is generated even when the FIFO buffer has a free space.
When the transaction counter is used, a request trigger is generated when the specified number of packets have been received.
In that case, the request trigger is generated even if the FIFO buffer has a free space.

(b) When, in a double-buffer configuration, one FIFO buffer is in the read-enabled state when reading from the other FIFO buffer has ended
If receiving from one FIFO buffer has ended during reading from the other FIFO buffer, no request trigger is generated until the ongoing reading from the other FIFO buffer ends.

This interrupt does not occur during the communication at the status stage of a control transfer.

The PIPEBRDY interrupt status of the corresponding pipe can be cleared to "0" by writing "0" to the corresponding bit. When clearing a PIPEBRDY bit by writing 0, write 1 to all the PIPEBRDY bits corresponding to other pipes.

Clearing the pipe BRDT interrupt status must be done always before the next access to the FIFO buffer.

### (b) When BRDYM = 0 and BFRE = 1

When the BRDYM bit is 0 and the BFRE bit is 1, this module determines that a BRDY interrupt occurs when all the data for a transfer has been read through a receiving pipe, and sets the PIPEBRDY bit corresponding to the pipe.

This module determines that the last data in a transfer has been received when one of the following conditions is met:

1. A short packet or a zero-length packet has been received.
2. The transaction counter (TRNCNT bits) is used, and as many packets as the value of the TRNCNT bits have been received.

When one of the above conditions is met and reading of the relevant data has ended, this module determines that all the data in a transfer has been read.

If a zero-length packet is received when the FIFO buffer is empty, this module determines that all the data in a transfer has been read when the FRDY bit is set and the DTLN bits are cleared in the corresponding FIFO Port Control Register. To start the next transfer in that case, write 1 to the BCLR bit in the corresponding FIFOCTR.

When the BRDYM bit is 0 and the BFRE bit is 1, this module does not detect any BRDY interrupt to the pipe in the sending direction.

The PIPEBRDY interrupt status of the corresponding pipe can be cleared to “0” by writing “0” to the corresponding bit. When clearing a PIPEBRDY bit by writing 0, write 1 to the PIPEBRDY bits corresponding to other pipes.

In this mode, do not change the value of the BFRE bit until all the processing for a transfer ends.

If you need to change the value of the BFRE bit during the transfer, set the ACLRM bit to clear all the FIFO buffers for the specified pipe.

### (c) When BRDYM = 1 and BFRE = 0

When the BRDYM bit is 1 and the BFRE bit is 0, the values of individual PIPEBRDY bits interlock with the values of the BSTS bits for individual pipes. In other words, this module sets or clears the BRDY interrupt status of a pipe according to the state of the FIFO buffer allocated to the pipe.

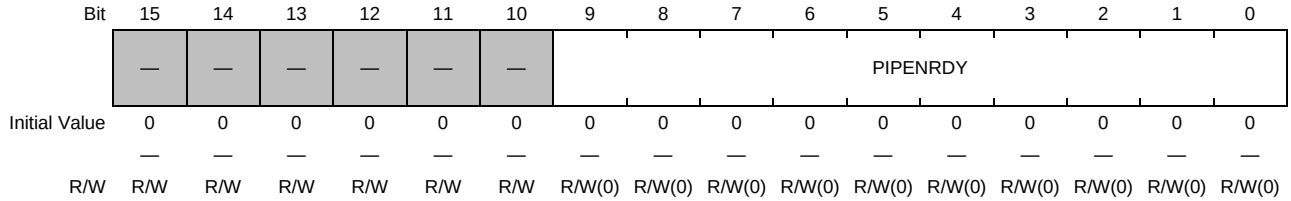
(a) For the pipe in the sending direction
This module sets the PIPEBRDY bit for the pipe when data can be written to the FIFO port or clears the PIPEBRDY bit when data cannot be written to the FIFO port.
The BRDY interrupt signal, however, is not asserted even when the sending pipe is write-enabled if the pipe is the DCP.

(b) For the pipe in the receiving direction
This module sets the PIPEBRDY bit for the pipe when data can be read from the FIFO port or clears the PIPEBRDY bit when all data has been read (that is, when data reading from the FIFO port is disabled).
If a zero-length packet is received when the FIFO buffer is empty, the PIPEBRDY bit corresponding to the specified pipe is kept being set and the BRDY interrupt signal is kept being asserted until sets the BCLR bit.

When the BRDYM bit is 1 and the BFRE bit is 0, this module cannot clear any PIPEBRDY bit.

When the BRDYM bit is 1, all the BFRE bits (for all pipes) must be 0.

### 32B.2.10.3 NRDY Interrupt Status Register [NRDYSTS] <Address: H'048>




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 to 10 | — | All 0 | R/W | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 9 to 0 | PIPENRDY | All 0 | R/W(0) | Pipe NRDY interrupt status<br>These bits indicate the NRDY interrupt status of each pipe.<br>0: No interrupt is generated.<br>1: An interrupt is generated. |

**Note:** Bit numbers correspond to pipe numbers.

**Note:** To clear the interrupt status indicated by a bit in this register, write 0 only to the bit and 1 to all other bit.

#### (1) Pipe NRDY interrupt status (PIPENRDY) bit

When this module issues an internal NRDY interrupt request for a pipe of which the PID is set to BUF, this module sets the PIPENRDY bit corresponding to the pipe in the NRDYSTS register. At that time, if the NRDYENB register bit corresponding to the pipe is set, this module sets the NRDY bit in the INTSTS0 register, and asserts the interrupt.

This module issues an internal NRDY interrupt request for individual pipes under the conditions described below. This module does not issue any interrupt request at the status stage of a control transfer.

##### (a) For the pipe in the sending direction

(1) When an IN token is received in a situation where the PID bits corresponding to the specified pipe are 01 (BUF) and the FIFO buffer does not contain send data
When an IN token is received, this module issues an NRDY interrupt request, and sets the PIPENRDY bit.
If the transfer type of the pipe for which the interrupt is generated is isochronous transfer, this module sends a zero- length packet, and sets the OVRN bit.

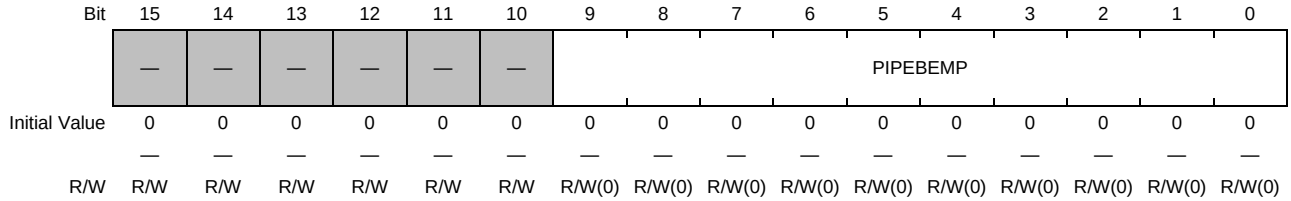
##### (b) For the pipe in the receiving direction

(1) When an OUT token is received in a situation where the PID bits corresponding to the specified pipe are 01 (BUF) and the FIFO buffer is full
If the transfer type of the pipe for which the interrupt is generated is isochronous transfer, this module issues an NRDY interrupt request, sets the PIPENRDY bit, and sets the OVRN bit.
If the transfer type of the pipe for which the interrupt is generated is not isochronous transfer, this module issues an NRDY interrupt request when sending an NAK Handshake signal after receiving the data that follows the OUT token, and sets the PIPENRDY bit.
Note, however, that this module does not issue an NRDY interrupt request when resending data (when a DATA-PID mismatch has occurred).
This module does not issue an NRDY interrupt request also when an error has occurred in a data packet.

(2) When a PING token is received in a situation where the PID bits corresponding to the specified pipe are 01 (BUF) and the FIFO buffer is full
When a PING token is received, this module issues an NRDY interrupt request, and sets the PIPENRDY bit.

(3) When the transfer type of the specified pipe is isochronous transfer, the PID bits corresponding to the specified pipe are 01 (BUF), and data has not been received normally within an interval frame
When an SOF token is received, this module issues an NRDY interrupt request, and sets the PIPENRDY bit for the specified pipe.

### 32B.2.10.4 BEMP Interrupt Status Register [BEMPSTS] <Address: H'04A>




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 to 10 | — | All 0 | R/W | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 9 to 0 | PIPEBEMP | All 0 | R/W(0) | Pipe BEMP interrupt status<br>These bits indicate the BEMP interrupt status of each pipe.<br>0: No interrupt is generated.<br>1: An interrupt is generated. |

**Note:** Bit numbers correspond to pipe numbers.

**Note:** To clear the interrupt status indicated by a bit in this register, write 0 only to the bit and 1 to all other bit.

#### (1) Pipe BEMP interrupt status (PIPEBEMP) bit

When this module detects a BEMP interrupt to a pipe of which the PID is set to BUF, this module sets the PIPEBEMP bit corresponding to the pipe in the BEMPSTS register. At that time, if the BEMPENB register bit corresponding to the pipe is set, this module sets the BEMP bit in the INTSTS0 register, and asserts the interrupt.

This module issues an internal BEMP interrupt request for individual pipes under the conditions described below.

1. When data sending (including sending of zero-length packets) to a pipe in the sending direction has ended and the FIFO buffer allocated to the pipe is empty
   In a single-buffer configuration, this module generates a BRDY interrupt at the same time as issuing an internal BEMP interrupt request for the pipes other than the DCP.
   Note, however, that this module does not issue the internal BEMP interrupt request in the following cases:

(a) When software (DMAC) has already started writing to the CPU-side FIFO buffer, in a double-buffer configuration, when sending data for one buffer ends

(b) When the buffer is cleared (to empty the buffer) by writing 1 to the ACLRM or BCLR bit

(c) During an IN transfer (sending zero-length packets) at the status stage of a control transfer

2. For the pipe in the receiving direction
   When the data larger than the specified maximum packet size has been received normally
   In that case, this module issues a BEMP interrupt request, sets the PIPEBEMP bit for the specified pipe, discards the received data, and changes the value of the PID bits for the specified pipe to 11 (STALL).
   Then, this module returns a STALL packet.
   Note, however, that this module does not issue the internal BEMP interrupt request in the following cases:

(a) When a CRC or bit stuff error has been detected in the received data

(b) When a SETUP transaction is being executed

Writing 0 to this bit clears the interrupt status.

Writing 1 to this bit causes no effect.

## 32B.2.11 Frame Number Registers (FRMNUM, UFRMNUM)

### 32B.2.11.1 Frame Number Register [FRMNUM] <Address: H'04C>

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | OVRN | CRCE | — | — | — | FRNM | | | | | | | | | | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| R/W | R/W(0) | R/W(0) | 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(0) | Overrun/underrun detection status<br>This bit indicates whether an overrun or underrun is detected in the pipe being used to perform an isochronous transfer.<br>0: No error<br>1: An error occurred. |
| 14 | CRCE | 0 | R/W(0) | CRC error detection status<br>This bit indicates the CRC error detection status for the pipe being used to perform an isochronous transfer.<br>0: No error<br>1: An error occurred. |
| 13 to 11 | — | All 0 | R | Nothing is assigned to these bits. Fix these bits to 0. |
| 10 to 0 | FRNM | All 0 | R | Frame number<br>These bits indicate the latest frame number. |

**Note:** The OVRN bit is intended for debugging. When designing a system, design transfer timings appropriately to prevent buffer overruns and underruns.

#### (1) Overrun/underrun detection status (OVRN) bit

This module sets this bit when this module detects an overrun or underrun in a pipe of which the transfer type is isochronous transfer.

When this module detects an overrun or underrun, this module issues an internal NRDY interrupt request. For details, see **Section 32B.2.10.3(1), Pipe NRDY interrupt status (PIPENRDY) bit**.

Software can clear this bit by writing 0 to this bit. If the CRCE bit should not be cleared together when this bit is cleared, write H'40.

When the peripheral controller function is selected

This module sets this bit in the following cases:

1. When an IN token is received although writing send data to the FIFO has not ended, in the case of a pipe that is in the sending direction and performs an isochronous transfer
2. When an OUT token is received although the free space of FIFO buffer is less than the size of one FIFO buffer, in the case of a pipe that is in the sending direction and performs an isochronous transfer

### (2) CRC error detection status (CRCE) bit

This module sets this bit when this module detects a CRC or bit stuff error in a pipe of which the transfer type is isochronous transfer.

Software can clear this bit by writing 0 to this bit. If the OVRN bit should not be cleared together when this bit is cleared, write H’80.

When this module detects a CRC error, this module issues an internal NRDY interrupt request. For details, see **Section 32B.2.10.3(1), Pipe NRDY interrupt status (PIPENRDY) bit**.

### (3) Frame number (FRNM) bits

This module updates the value of these bits each time an SOF packet is received (once per 1 ms), and indicates the latest frame number in these bits.

When reading these bits, read them twice and check for consistency.

### 32B.2.11.2 Micro Frame Number Register [UFRMNUM] <Address: H'04E>

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | UFRNM | | |
| 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 | — | 0 | R | Nothing is assigned to these bits. Fix these bits to 0. |
| 14 to 3 | — | All 0 | R | Nothing is assigned to these bits. Fix these bits to 0. |
| 2 to 0 | UFRNM | All 0 | R | Micro frame<br>These bits indicate the micro frame number. |

#### (1) Micro frame number (UFRNM) bit

In high-speed transfer mode, this module writes the micro frame number to these bits. In a mode other than high-speed transfer mode, this module writes H'00 to these bits.

When reading these bits, read them twice and check for consistency.

## 32B.2.12 USB Address

### 32B.2.12.1 USB Address Register [USBADDR] <Address: H'050>

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | USBADDR | | | | | | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 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 11 | — | All 0 | R | Nothing is assigned to these bits. Fix these bits to 0. |
| 10 to 8 | — | All 0 | R | Nothing is assigned to these bits. Fix these bits to 0. |
| 7 | — | 0 | R | Nothing is assigned to this bit. Fix this bit to 0. |
| 6 to 0 | USBADDR | All 0 | R | USB address<br>These bits indicate the USB address allocated by the host. |

#### (1) USB address (USBADDR) bits

When this module has received and normally processed a SetAddress request, this module writes the received USB address to these bits.

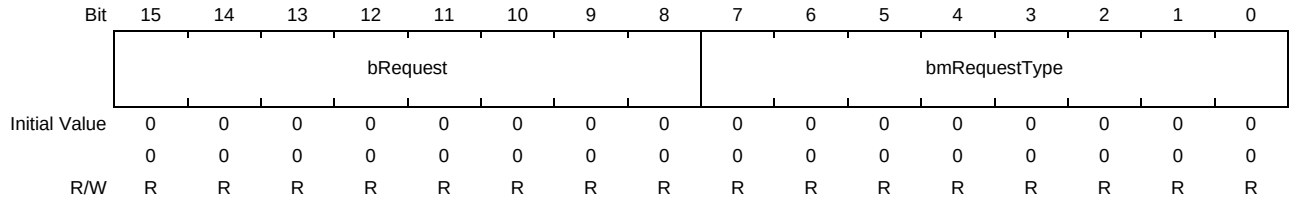
When this module detects a USB bus reset, this module writes H'00 to these bits.

## 32B.2.13 USB Request Registers

USB request registers are used to store control transfer setup requests.

These registers store the values set in the received USB requests.

### 32B.2.13.1 USB Request Type Register [USBREQ] <Address: H'054>




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 to 8 | bRequest | All 0 | R | Request<br>Value of USBRequestbRequest |
| 7 to 0 | bmRequestType | All 0 | R | Request type<br>Value of USBRequestbmRequestType |

#### (1) USB request (bRequest) bits

These bits indicate the value of the USB request data this module has received in a SETUP transaction. Writing to these bits is ignored.

#### (2) USB request type (bRmRequestType) bits

These bits indicate the value of the USB request data this module has received in a SETUP transaction. Writing to these bits is ignored.

### 32B.2.13.2 USB Request Value Register [USBVAL] <Address: H'056>

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | wValue | | | | | | | | | | | | | | | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 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 0 | wValue | All 0 | R | Value<br>Value of the wValue field in a USB request |

#### (1) Value (wValue) bits

These bits indicate the value of the wValue field in a USB request.

These bits indicate the value of wValue field in the USB request data this module has received in a SETUP transaction. Writing to these bits is ignored.

### 32B.2.13.3 USB Request Index Register [USBINDX] <Address: H'058>

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | wIndex | | | | | | | | | | | | | | | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 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 |

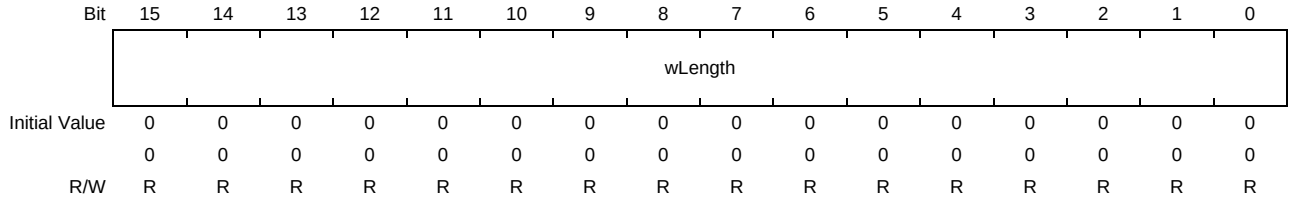
| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 to 0 | wIndex | All 0 | R | Index<br>Value of the wIndex field in a USB request |

#### (1) Index (wIndex) bits

These bits indicate the value of the wIndex field in a USB request.

These bits indicate the value of wIndex field in the USB request data this module has received in a SETUP transaction. Writing to these bits is ignored.

### 32B.2.13.4 USB Request Length Register [USBLENG] <Address: H'05A>




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 to 0 | wLength | All 0 | R | Length<br>Value of the wLength field in a USB request |

#### (1) Length (wLength) bits

These bits indicate the value of the wLength field in a USB request.

These bits indicate the value of wLength field in the USB request data this module has received in a SETUP transaction. Writing to these bits is ignored.

## 32B.2.14 DCP Configuration

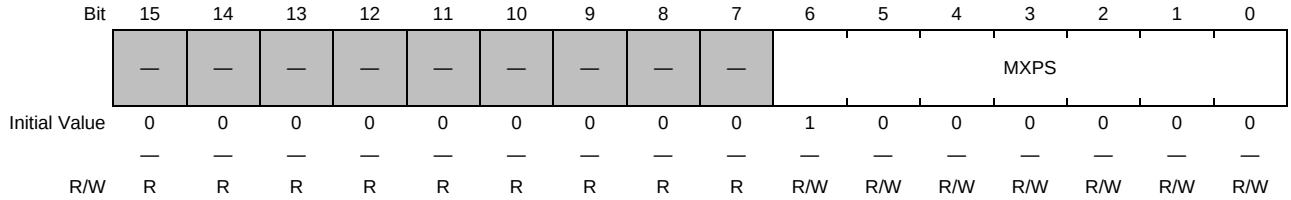
When performing a data communication by a control transfer, use the default control pipe (DCP).

### 32B.2.14.1 DCP Configuration Register [DCPCFG] <Address: H'05C>

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | CNTMD | SHTNAK | — | — | — | — | — | — | — |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | — | — | — | — | — | — | — | R/W | R/W | — | — | — | — | — | — | — |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 to 9 | — | All 0 | | Nothing is assigned to these bits. Fix these bits to 0. |
| 8 | CNTMD | 0 | R/W | Continuous transfer mode<br>This bit specifies whether to use the default control pipe to perform communication in continuous transfer mode.<br>0: Non-continuous transfer mode<br>1: Continuous transfer mode |
| 7 | SHTNAK | 0 | R/W | Pipe disable at transfer end<br>When the default control pipe is in the receiving direction, this bit specifies whether to change the PID setting to NAK when the transfer ends.<br>0: Continues the pipe when the transfer ends<br>1: Disables the pipe when the transfer ends |
| 6 to 0 | — | All 0 | | Nothing is assigned to these bits. Fix these bits to 0. |

### 32B.2.14.2 DCP Max. Packet Size Register [DCPMAXP] <Address: H'05E>




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 to 7 | — | All 0 | R | Nothing is assigned to these bits. Fix these bits to 0. |
| 6 to 0 | MXPS | H'40 | R/W | Maximum packet size<br>These bits specify the maximum data payload size (maximum packet size) for the DCP. |

#### (1) Maximum packet size (MXPS) bits

These bits are used to specify the maximum data payload size (maximum packet size) for the DCP.

The default is H'40 (64 bytes).

The value of the MXPS bits must conform to the USB Specification.

Writing a value to the MXPS bits must be done when the PID is set to NAK and no value is set in the CURPIPE bits. If you need to change the value of these bits after changing the PID setting for the specified pipe from BUF to NAK, check the PBUSY bit is 0 before changing the value of these bits. If, however, this module has changed the PID setting from BUF to NAK, the value of the PBUSY bit need not be checked.

When the MXPS bits are all 0, data must not be written to the FIFO buffer, and the PID setting must not be changed to BUF.

### 32B.2.14.3 DCP Control Register [DCPCTR] <Address: H'060>

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | BSTS | — | — | — | — | — | — | SQCLR | SQSET | SQMON | PBUSY | — | — | CCPL | PID | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R(0)/W(1) | R(0)/W(1) | R | R | R | R | R/W | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 | BSTS | 0 | R | Buffer status<br>This bit indicates the accessibility status of the DCP FIFO buffer.<br>0: Buffer access is not possible.<br>1: Buffer access is possible. |
| 14 to 9 | — | All 0 | R | Nothing is assigned to these bits. Fix these bits to 0. |
| 8 | SQCLR | 0 | R(0)/W(1) | Toggle bit clear<br>This bit can set DATA0 as the expected value of the sequence toggle bit for the next transaction in DCP transfer.<br>0: Writing disabled<br>1: DATA0 specified |
| 7 | SQSET | 0 | R(0)/W(1) | Toggle bit set<br>This bit can set DATA1 as the expected value of the sequence toggle bit for the next transaction in DCP transfer.<br>0: Writing disabled<br>1: DATA1 specified |
| 6 | SQMON | 1 | R | Sequence toggle bit monitor<br>This bit indicates the expected value of the sequence toggle bit for the next transaction in DCP transfer.<br>0: DATA0<br>1: DATA1 |
| 5 | PBUSY | 0 | R | Pipe busy<br>This bit indicates whether the specified pipe is being used in the USB bus.<br>0: The specified pipe is not used in the USB bus.<br>1: The specified pipe is being used in the USB bus. |
| 4 to 3 | — | All 0 | R | Nothing is assigned to these bits. Fix these bits to 0. |
| 2 | CCPL | 0 | R/W | Control transfer end enable<br>Setting this bit permits the status stage of the control transfer to end.<br>0: Does not permit the control transfer to end.<br>1: Permits the control transfer to end. |
| 1, 0 | PID | All 0 | R/W | Response PID<br>These bits control responses from this module in control transfer.<br>00: NAK response<br>01: BUF response (according to the buffer status)<br>10: STALL response<br>11: STALL response |

### (1) Buffer status (BSTS) bit

This bit indicates whether the CPU can access the FIFO buffer allocated to the DCP. This bit is operated by this module. The meaning of this bit varies as follows depending on the value of the ISEL bit:

1. When ISEL = 0: This bit indicates whether receive data can be read from the FIFO buffer.
2. When ISEL = 1: This bit indicates whether send data can be written to the FIFO buffer.

### (2) Sequence toggle bit clear (SQCLR) bit

When software sets this bit, this module specifies DATA0 as the expected value of the sequence toggle bit for the specified pipe. This module always clears this bit.

Do not set the SQCLR and SQSET bits at the same time.

Setting this bit must be done when the PID is set to NAK and no value is set in the CURPIPE bits.

If you need to set this bit after changing the PID setting for the specified pipe from BUF to NAK, check, that the PBUSY bit is 0 before setting this bit. If, however, this module has changed the PID setting from BUF to NAK, the value of the PBUSY bit need not be checked.

### (3) Sequence toggle bit set (SQSET) bit

When software sets this bit, this module specifies DATA1 as the expected value of the sequence toggle bit for the specified pipe. This module always clears this bit.

Do not set the SQCLR and SQSET bits at the same time.

Setting this bit must be done when the PID is set to NAK and no value is set in the CURPIPE bits.

If you need to set this bit after changing the PID setting for the specified pipe from BUF to NAK, check, that the PBUSY bit is 0 before setting this bit. If, however, this module has changed the PID setting from BUF to NAK, the value of the PBUSY bit need not be checked.

### (4) Sequence toggle bit monitor (SQMON) bit

This bit indicates the expected value of the sequence toggle bit for the specified pipe. This bit is operated by this module. When a transaction ends normally, this module toggles this bit.

This module, however, does not toggle this bit if a DATA-PID mismatch occurs during a transfer in the receiving direction.

When a SETUP packet is received normally, this module sets this bit (to specify DATA1 as the expected value).

This module does not reference this bit in an IN or OUT transaction at the status stage. Also, this module does not toggle this bit even when the transaction ends normally.

### (5) Pipe busy (PBUSY) bit

This module changes this bit from 0 to 1 when a USB transaction using the specified pipe starts. This module changes this bit from 1 to 0 when the transaction ends.

Reading this bit after software sets the PID to NAK enables you to check whether you can change pipe settings.

### (6) Control transfer end enable (CCPL) bit

When software sets this bit when the PID of the specified pipe is BUF, this module ends the status stage of the ongoing control transfer.

In other words, in a control read transfer, this module sends an ACK handshake in response to an OUT transaction request from the USB host, and, in a control write or no-data control transfer, this module sends a zero-length packet in response to an IN transaction request from the USB host. If, however, a SetAddress request is detected, this module performs automatic response throughout the period from the setup stage to the end of the status stage regardless of the value of this bit.

When a new SETUP packet is received, this module changes this bit from 1 to 0.

When the VALID bit is 1, software cannot set this bit.

### (7) Response PID (PID) bits

The setting of these bits must be changed from NAK to BUF when the data stage or status stage of a control transfer is executed.

This module changes the value of these bits in the following cases:

1. This module changes the value of these bits to 00 (NAK) when it receives a SETUP packet. At that time this module sets the VALID bit. Software cannot change the value of these bits until it clears the VALID bit.
2. When set these bits to 01 (BUF), this module changes the value of these bits to 11 (STALL) when it receives the data exceeding the specified maximum packet size.
3. This module changes the value of these bits to 1x (STALL) when it detects a sequence error in a control transfer.
4. This module changes the value of these bits to 00 (NAK) when it detects a USB reset.

This module does not reference these bits during SetAddress request processing (automatic processing).

## 32B.2.15 Pipe Configuration Registers (PIPESEL, PIPECFG, PIPEBUF, PIPEMAXP, PIPEPERI)

To configure pipes 1 to 15, use the PIPESEL, PIPECFG, PIPEBUF, PIPEMAXP, PIPEPERI, PIPExCTR, PIPExTRE, and PIPExTRN registers.

Select the pipes to be used by using the PIPESEL register, and then configure functions of individual pipes by using the PIPECFG, PIPEBUF, PIPEMAXP, and PIPEPERI registers. Note that you can use the PIPExCTR, PIPExTRE, and PIPExTRN registers for setting regardless of the pipe selection by the PIPESEL register.

### 32B.2.15.1 Pipe Window Select Register [PIPESEL] <Address: H'064>

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | PIPESEL | | | |
| 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 | Nothing is assigned to these bits. Fix these bits to 0. |
| 3 to 0 | PIPESEL | All 0 | R/W | Pipe window select<br>These bits specify a pipe for registers at addresses H'68 to H'6E.<br>0000: No selection<br>0001: PIPE1<br>0010: PIPE2<br>0011: PIPE3<br>0100: PIPE4<br>0101: PIPE5<br>0110: PIPE6<br>0111: PIPE7<br>1000: PIPE8<br>1001: PIPE9 |

**Note:** When the PIPESEL bits are 0000, all bits of the PIPECFG, PIPEBUF, PIPEMAXP, and PIPEPERI registers are cleared. When the PIPESEL bits are 0000, writing to the registers at addresses H'68 to H'6E is ignored.

#### (1) Pipe Window select (PIPESEL) bits

When software a value from 0001 to 1111 to these bits, this module indicate the pipe information and settings corresponding to the registers at addresses H'68 to H'6E. After a pipe is selected by these bits, the values set in the areas at addresses H'68 to H'6E are applied, by this module, to the transfer operation using the selected pipe.

When writes 0000 to these bits, this module writes 0 to all bits of the registers at addresses H'68 to H'6E. Then, writing to the areas at addresses H'68 to H'6E is ignored.

### 32B.2.15.2 Pipe Configuration Register [PIPECFG] <Address: H'068>

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TYPE | | — | — | — | BFRE | DBLB | CNTMD | SHTNAK | — | — | DIR | EPNUM | | | |
| 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/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 | All 0 | R/W | Transfer type<br>These bits specify the transfer type of the pipe specified in the PIPESEL bit.<br>00: The pipe cannot be used.<br>01: Bulk transfer<br>10: Interrupt transfer<br>11: Isochronous transfer |
| 13 to 11 | — | All 0 | R | Nothing is assigned to these bits. Fix these bits to 0. |
| 10 | BFRE | 0 | R/W | BRDY interrupt operation specification<br>This bit specifies the timing at which this module notifies a BRDY interrupt relating to the specified pipe.<br>0: A BRDY interrupt is notified when data is sent or received.<br>1: A BRDY interrupt is notified when reading of data is completed. |
| 9 | DBLB | 0 | R/W | Double-buffer mode<br>This bit specifies a single or double FIFO buffer to be used by the specified pipe.<br>0: Single buffer<br>1: Double buffer |
| 8 | CNTMD | 0 | R/W | Continuous transfer mode<br>This bit specifies whether to use the specified pipe to perform communication in continuous transfer mode.<br>0: Non-continuous transfer mode<br>1: Continuous transfer mode |
| 7 | SHTNAK | 0 | R/W | Pipe disable at transfer end<br>When the specified pipe is in the receiving direction, this bit specifies whether to change the PID setting to NAK when the transfer ends.<br>0: Continues the pipe when the transfer ends.<br>1: Disables the pipe when the transfer ends. |
| 6, 5 | — | All 0 | R | Nothing is assigned to these bits. Fix these bits to 0. |
| 4 | DIR | 0 | R/W | Transfer direction<br>This bit specifies the transfer direction of the specified pipe.<br>0: Receiving direction<br>1: Sending direction |
| 3 to 0 | EPNUM | All 0 | R/W | Endpoint number<br>These bits specify the endpoint number of the specified pipe. |

### (1) Transfer type (TYPE) bits

These bits are used to specify the USB transfer type of the pipe (selected pipe) specified in the PIPESEL bits.

**Table 32B.12** lists pipes and the transfer types specifiable in these bits.

Table 32B.12 Selected Pipes and the Transfer Types Specifiable in the TYPE Bits

| Selected Pipe | TYPE Bits | USB Transfer Type |
|---|---|---|
| PIPE1 or PIPE2 | 01 or 11 | Bulk or isochronous transfer |
| PIPE3 to PIPE5 | 01 | Bulk transfer |
| PIPE6 to PIPE9 | 10 | Interrupt transfer |

Always specify a value other than 00 in these bits for a selected pipe before setting the PID of the selected pipe to BUF (to start USB communication using the selected pipe).

The value of these bits for a selected pipe must be done while the PID of the selected pipe is NAK. If you change the value of these bits for a selected pipe after having changed the PID of the selected pipe from BUF to NAK, check, that, and the PBUSY bit is 0 before changing the value of these bits. Note, however, that, if this module has changed the PID of the selected pipe to NAK, the value of the PBUSY bit need not be checked.

### (2) BRDY interrupt operation specification (BFRE) bit

This bit is valid when the selected PIPE is PIPE1 to PIPE5.

When set this bit and been using the selected pipe in the receiving direction (in other words, the DIR bit is 0), this module detects the end of transfer (when it occurs) and generates a BRDY interrupt when reading of the last packet ends.

If a BRDY interrupt occurs with the above settings, software must write 1 to the BCLR bit. The FIFO buffer allocated to the selected pipe remains unready for reception until 1 is written to the BCLR bit.

When set this bit and been using the selected pipe in the sending direction (in other words, the DIR bit is 1), this module does not generate any BRDY interrupt.

For details, see the description of the PIPEBRDY interrupt status bit.

Changing the value of this bit for a selected pipe must be done while the PID of the selected pipe is NAK and the selected pipe is not specified in the CURPIPE bits.

If you change the value of this bit for a selected pipe after the USB communication using the selected pipe, not only check the values of above three kinds of register bits but also write 1 and 0 successively to the ACLRM bit to clear the FIFO buffer allocated to the selected pipe.

If you change the value of this bit for a selected pipe after having changed the PID of the selected pipe from BUF to NAK, check, that, and the PBUSY bit is 0 before changing the value of this bit. Note, however, that, if this module has changed the PID of the selected pipe to NAK, the value of the PBUSY bit need not be checked.

### (3) Double-buffer mode (DBLB) bit

This bit is valid when the selected pipe is PIPE1 to PIPE5.

When set this bit for a selected pipe, this module allocates, to the selected pipe, two FIFO buffers, each of which has the FIFO buffer size specified in the BUFSIZE bits in the PIPEBUF register.

The size of the FIFO buffer this module allocates to the selected pipe is as follows:

(BUFSIZE + 1) × 64 × (DBLB + 1) [bytes]

Changing the value of this bit for a selected pipe must be done while the PID of the selected pipe is NAK and the selected pipe is not specified in the CURPIPE bits.

If you change the value of this bit for a selected pipe after the USB communication using the selected pipe, not only check the values of above three kinds of register bits but also write 1 and 0 successively to the ACLRM bit to clear the FIFO buffer allocated to the selected pipe.

If you change the value of this bit for a selected pipe after having changed the PID of the selected pipe from BUF to NAK, check, that, and the PBUSY bit is 0 before changing the value of this bit. Note, however, that, if this module has changed the PID of the selected pipe to NAK, the value of the PBUSY bit need not be checked.

### (4) Continuous transfer mode (CNTMD) bit

This bit is valid when the selected pipe is PIPE1 to PIPE5, and the transfer type of the selected pipe is bulk transfer.

This module determines whether data sending from or receiving in the FIFO buffer allocated to the selected pipe has ended according to the value of this bit in the way described in **Table 32B.13**.

Table 32B.13 How to Determine the End of Data Sending from or Receiving in the FIFO Buffer According to the Value of the CNTMD Bit

| CNTMD Bit Setting Value | How to Determine Whether Reading or Sending is Enabled |
|---|---|
| 0 | Condition for enabling reading from the FIFO buffer when the receiving direction is set ("DIR = 0"):<br>This module receives one packet |
| | Condition for enabling sending from the FIFO buffer when the sending direction is set ("DIR = 1"):<br>Either of the following conditions (1) and (2) is met:<br>(1) Data for the maximum packet size is written to the FIFO buffer.<br>(2) Data for the short packet (including the case of zero-byte data) is written to the FIFO buffer and then 1 is written in the BVAL bit. |
| 1 | Conditions for enabling reading from the FIFO buffer when the receiving direction is set ("DIR = 0"):<br>(1) The number of bytes in data received in the FIFO buffer allocated to the selected pipe becomes equal to the number of allocated bytes ((BUFSIZE + 1) * 64).<br>(2) This module receives a short packet other than a zero-length packet.<br>(3) This controller receives a zero-length packet when the FIFO buffer allocated to the selected pipe already contains data.<br>(4) Packets are received as many times as the value of the transaction counter set for the selected pipe. |
| | Condition for enabling sending from the FIFO buffer when the sending direction is set ("DIR = 1"):<br>One of the following conditions (1) to (3) is met:<br>(1) The amount of written data becomes equal to the size of one FIFO buffer allocated to the selected pipe.<br>(2) Data for less than the size of one FIFO buffer allocated to the selected pipe (including the case of zero-byte data) is written to the FIFO buffer and then 1 is written in the BVAL bit.<br>(3) Data for less than the size of one FIFO buffer allocated to the selected pipe (including the case of zero-byte data) is written to the FIFO buffer, and a transfer end signal is asserted at the same time of the last writing. |

Changing the value of this bit for a selected pipe must be done while the PID of the selected pipe is NAK and the selected pipe is not specified in the CURPIPE bits.

If you change the value of this bit for a selected pipe after the USB communication using the selected pipe, not only check the values of above three kinds of register bits but also write 1 and 0 successively to the ACLRM bit to clear the FIFO buffer allocated to the selected pipe.

If you change the value of this bit for a selected pipe after having changed the PID of the selected pipe from BUF to NAK, check, that, and the PBUSY bit is 0 before changing the value of this bit. Note, however, that, if this module has changed the PID of the selected pipe to NAK, the value of the PBUSY bit need not be checked.

### (5) PIPE disable at transfer end (SHTNAK) bit

This bit is valid when the selected pipe is PIPE1 to PIPE5 and is in the receiving direction.

When set this bit for a selected pipe in the receiving direction, this module changes the PID of the selected pipe to NAK when this module determines the end of data transfer to the selected pipe. This module determines the end of transfer when one of the following conditions (1) and (2) is met:

(1) This module has normally received short packet data (including zero-length packets).

(2) When using a transaction counter, this module has normally received as many packets as the value set in the transaction counter.

Changing the value of this bit for a selected pipe must be done while the PID of the selected pipe is NAK.

If you change the value of this bit for a selected pipe after having changed the PID of the selected pipe from BUF to NAK, check, that, and the PBUSY bit is 0 before changing the value of this bit. Note, however, that, if this module has changed the PID of the selected pipe to NAK, the value of the PBUSY bit need not be checked.

For the pipes in the sending direction, this bit must be cleared.

### (6) Transfer direction (DIR) bit

When writes 0 to this bit for a selected pipe, this module uses the selected pipe in the receiving direction. When software writes 1 to this bit for the selected pipe, this module uses the selected pipe in the sending direction.

Changing the value of this bit for a selected pipe must be done while the PID of the selected pipe is NAK and the selected pipe is not specified in the CURPIPE bits.

If you change the value of this bit for a selected pipe after the USB communication using the selected pipe, not only check the values of above three kinds of register bits but also write 1 and 0 successively to the ACLRM bit to clear the FIFO buffer allocated to the selected pipe.

If you change the value of this bit for a selected pipe after having changed the PID of the selected pipe from BUF to NAK, check, that, and the PBUSY bit is 0 before changing the value of this bit. Note, however, that, if this module has changed the PID of the selected pipe to NAK, the value of the PBUSY bit need not be checked.

### (7) Endpoint number (EPNUM) bits

These bits are used to specify the endpoint number of the endpoint of a selected pipe. Note that specifying 0000 in these bits for a pipe means that the pipe is not used.

Changing the value of these bits for a selected pipe must be done while the PID of the selected pipe is NAK and the selected pipe is not specified in the CURPIPE bits.

If you change the value of these bits for a selected pipe after having changed the PID of the selected pipe from BUF to NAK, check, that, and the PBUSY bit is 0 before changing the value of these bits. Note, however, that, if this module has changed the PID of the selected pipe to NAK, the value of the PBUSY bit need not be checked.

The combination of the values of the DIR bit and EPNUM bits for a pipe must be unique among those for all pipes. (The setting "EPNUM = 000" [the selected pipe is not used] can be duplicated for multiple pipes.)

### 32B.2.15.3 Pipe Buffer Setting Register [PIPEBUF] <Address: H'06A>

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | BUFSIZE | | | | | — | — | BUFNMB | | | | | | | |
| 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/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 | — | 0 | R | Nothing is assigned to this bit. Fix this bit to 0. |
| 14 to 10 | BUFSIZE | All 0 | R/W | Buffer size<br>These bits specify the size of the FIFO buffer for the pipe specified in the PIPESEL bit.<br>H'00: 64 bytes<br>H'01: 128 bytes<br>...<br>(H'1F: 2 Kbytes) |
| 9, 8 | — | All 0 | R | Nothing is assigned to these bits. Fix these bits to 0. |
| 7 to 0 | BUFNMB | All 0 | R/W | Buffer number<br>These bits specify the FIFO buffer number of the specified pipe.<br>(H'4 to H'7F) |

**Note:** Changing values of these register bits for a selected pipe must be done when sets the PID of the selected pipe to NAK, and specifies no pipe in the CURPIPE bits.

**Note:** If you change values of these register bits for a selected pipe after having changed the PID of the selected pipe from BUF to NAK, check, that, and the PBUSY bit is 0 before changing the values of these bits. Note, however, that, if this module has changed the PID of the selected pipe to NAK, the value of the PBUSY bit need not be checked.

#### (1) Buffer size (BUFSIZE) bits

These bits are used to specify the size of the FIFO buffer to be allocated to the selected pipe. Specify the FIFO buffer size in units of blocks. One block has 64 bytes.

When set the DBLB bit for a selected pipe, this module allocates, to the selected pipe, two FIFO buffers, each of which has the FIFO buffer size specified in these bits.

The size of the FIFO buffer this module allocates to the selected pipe is as follows:

(BUFSIZE + 1) × 64 × (DBLB + 1) [bytes]

The following value can be specified in these bits:

(1) Any value from H'0 to H'1F when the selected pipe is PIPE1 to PIPE5.

(2) H'0 only when the selected pipe is PIPE6 to PIPE9.

In continuous transfer mode (CNTMD = 1), specify an integral multiple of the maximum packet size in the BUFSIZE bits.

### (2) Buffer number (BUFNMB) bits

These bits are used to specify the block number of the first block in the FIFO buffer to be allocated to the selected pipe. This module allocates the following blocks of FIFO buffer to the selected pipe:

Block with block number “BUFNMB” to the block with block number “BUFNMB + (BUFSIZE + 1) × (DBLB + 1) - 1”

The value of these bits must be H’04 to H’7F. Note, however, that the following rules must be observed: Value " H’00" is exclusively used for DCP.

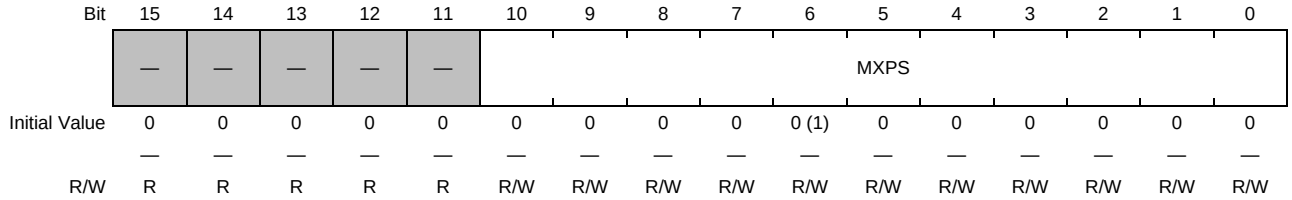
Value “H’04” is exclusively used for PIPE6. When, however, PIPE6 is not used, this value can be used for another pipe. If PIPE6 is selected, writing to the BUFNMB bits is ignored. Then, this module aromatically sets H’04 in the BUFNMB bits for PIPE6.

Value “H’05” is exclusively used for PIPE7. When, however, PIPE7 is not used, this value can be used for another pipe. If PIPE7 is selected, writing to the BUFNMB bits is ignored. Then, this module aromatically sets H’05 in the BUFNMB bits for PIPE7.

Value “H’06” is exclusively used for PIPE8. When, however, PIPE8 is not used, this value can be used for another pipe. If PIPE8 is selected, writing to the BUFNMB bits is ignored. Then, this module aromatically sets H’06 in the BUFNMB bits for PIPE8.

Value “H’07” is exclusively used for PIPE9. When, however, PIPE9 is not used, this value can be used for another pipe. If PIPE9 is selected, writing to the BUFNMB bits is ignored. Then, this module aromatically sets H’07 in the BUFNMB bits for PIPE9.

### 32B.2.15.4 Pipe Maximum Packet Size Register [PIPEMAXP] <Address: H'06C>




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 to 11 | — | All 0 | R | Nothing is assigned to these bits. Fix these bits to 0. |
| 10 to 0 | MXPS | *Note | R/W | Maximum packet size<br>These bits specify the maximum data payload size (maximum packet size) for the specified pipe.<br>For PIPE6 to PIPE8, a value from H'1 to H'40 (bytes) can be set. |

**Note:** The initial value of the MXPS bits is H'000 when no pipe is specified in the PIPESEL bits in the PIPESEL register or H'040 when a pipe is specified in the PIPESEL bits.

#### (1) Maximum packet size (MXPS) bits

These bits are used to specify the maximum data payload size (maximum packet size) for the selected pipe.

The initial value of these bits is H'40 (64 bytes).

- For PIPE1 and PIPE2, a value from H'1 (1 byte) to H'400 (1024 bytes) can be specified.
- For PIPE3 to PIPE5, H'8 (8 bytes), H'10 (16 bytes), H'20 (32 bytes), H'40 (64 bytes), or H'200 (512 bytes) can be specified. (Bits [2:0] are excluded.)
- For PIPE6 to PIPE9, a value from H'1 (1 byte) to H'40 (64 bytes) can be specified.

The value of the MXPS bits for individual transfer type must conform to the USB Specification.

Setting a value in these bits for a selected pipe must be done while the PID of the selected pipe is NAK and the selected pipe is not specified in the CURPIPE bits.

If you change the value of these bits for a selected pipe after having changed the PID of the selected pipe from BUF to NAK, check the PBUSY bit is 0 before changing the value of these bits. Note, however, that, if this module has changed the PID of the selected pipe to NAK, the value of the PBUSY bit need not be checked.

When the MXPS bits are all 0, data must not be written to the FIFO buffer, and the PID setting must not be changed to BUF.

### 32B.2.15.5 Pipe Cycle Control Register [PIPEPERI] <Address: H'06E>

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | IFIS | — | — | — | — | — | — | — | — | — | IITV | | |
| 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 | 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 | — | All 0 | R | Nothing is assigned to these bits. Fix these bits to 0. |
| 12 | IFIS | 0 | R/W | Isochronous IN buffer flush<br>This bit specifies whether to perform a buffer flush when the pipe specified in the PIPESEL bit is used for isochronous IN transfer.<br>0: Does not perform a buffer flush.<br>1: Performs a buffer flush. |
| 11 to 3 | — | All 0 | R | Nothing is assigned to these bits. Fix these bits to 0. |
| 2 to 0 | IITV | All 0 | R/W | These bits specify the transfer interval of the specified pipe. The value to be specified is the frame timing multiplied by an n-th power of 2. |

#### (1) Isochronous IN buffer flush (IFIS) bit

When the selected pipe is used for isochronous IN transfer, this bit is used to specify this module automatically clears the FIFO buffer if this module fails to receive the IN token from the USB host in a (micro) frame sent at intervals specified in the IITV bits.

In double-buffer mode (DBLB = 1), this module clears only the data in one buffer used earlier than the other.

This module clears the FIFO buffer when it receives an SOF packet immediately after the (micro) frame in which the IN token hag to be received. Even if the SOF packet is corrupted, this module clears the FIFO buffer in the same timing to receive the SOF packet by the use of the internal interpolation function.

#### (2) Interval error detection interval (IITV) bits

These bits specify the interval of interval error detection for the selected pipe. The value to be specified is the frame timing multiplied by an n-th power of 2.

Setting a value in these bits for a selected pipe must be done while the PID of the selected pipe is NAK and the selected pipe is not specified in the CURPIPE bits.

If you change the value of these bits for a selected pipe after having changed the PID of the selected pipe from BUF to NAK, check, that, and the PBUSY bit is 0 before changing the value of these bits. Note, however, that, if this module has changed the PID of the selected pipe to NAK, the value of the PBUSY bit need not be checked.

If you change the value specified in these bits to another after a USB communication, change the PID to NAK, and then set the ACLRM bit to initialize the interval timer before changing the value.

For PIPE3 to PIPE5 these bits are ignored. Write 0 to all these bits corresponding to PIPE3 to PIPE5.

You can specify a value in these bits when the transfer type of the selected pipe is isochronous.

### (a) When the selected pipe is used for isochronous OUT transfer

If this module does not receive any data packet in the (micro) frame sent at intervals specified in the IITV bits, this module generates an NRDT interrupt.

This module generates an NRDY interrupt also if this module cannot receive data because an error, e.g., CRC error, has occurred in a data packet or because the FIFO buffer is full (such a situation might result if, for example, software [DMAC] delays in reading data from the FIFO buffer).

This module generates the NRDY interrupt when it receives an SOF packet. Even if the SOF packet is corrupted, this module generates the NRDY interrupt in the same timing to receive the SOF packet by the use of the internal interpolation function.

When, however, the value of the IITV bits is not 0, this module generates the NRDY interrupt every time an SOF packet is received at the specified intervals after interval counting starts.

If the PID of the selected pipe is changed to NAK after the interval timer starts, this module does not generate the NRDY interrupt even when it receives an SOF packet.

The condition for starting interval counting varies by the value of the IITV bits.

(a) When IITV = 0: Interval counting starts when the PID of the selected pipe is changed to BUF.

| (Micro) frame | SOF | SOF | SOF | OUT | DATA0 | SOF | OUT | DATA0 |
|---|---|---|---|---|---|---|---|---|
| Setting of PID bits | NAK | NAK | BUF | | | BUF | | |
| Whether token reception is expected<br>(0: reception expected<br>—: non-reception expected) | — | — | 0 | | | 0 | | |
| Start of interval counting | | | ↑ | | | | | |

Figure 32B.1 Relationship between (Micro) Frames and Whether Token Reception Is Expected When IITV = 0

(b) When IITV is not 0: Interval counting starts at the end of the first normal reception of data packet after the PID of the selected pipe is changed to BUF.

| (Micro) frame | SOF | SOF | SOF | OUT | DATA0 | SOF | SOF | OUT | DATA0 | SOF | SOF | OUT | DATA0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Setting of PID bits | NAK | BUF | BUF | | | BUF | BUF | | | BUF | BUF | | |
| Whether token reception is expected<br>(0: reception expected<br>—: non-reception expected) | — | — | 0 | | | — | 0 | | | — | 0 | | |
| Start of interval counting | | | ↑ | | | | | | | | | | |

Figure 32B.2 Relationship between (Micro) Frames and Whether Token Reception Is Expected When IITV = 1

### (b) When the selected pipe is used for isochronous IN transfer

The IITV bits are used in combination with the setting of the IFIS bit to 1. When the IFIS bit is 0, this module sends a data packet in response to a received token regardless of the value of the IITV bits.

When the IFIS bit is 1, if this module does not receive any IN token in the (micro) frame sent at intervals specified in the IITV bits although the FIFO buffer has sendable data, this module clears the FIFO buffer.

This module clears the FIFO buffer also when it cannot receive an IN token normally because of a bus error, e.g., CRC error.

This module clears the FIFO buffer when it receives an SOF packet. Even if the SOF packet is corrupted, this module clears the FIFO buffer in the same timing to receive the SOF packet by the use of the internal interpolation function.

The condition for starting interval counting varies by the value of the IITV bits. (The condition is the same as that for isochronous OUT transfer.)

The interval counter is cleared when one of the following conditions (1) to (3) is met:

(1) This module is reset (then, also the IITV bits are cleared).

(2) The ACLRM bit is set.

(3) This module detects a USB bus reset.

## 32B.2.16 Pipe Control Registers (PIPExCTR)

### 32B.2.16.1 PIPE1 Control Register [PIPE1CTR] <Address: H'070>
### PIPE2 Control Register [PIPE2CTR] <Address: H'072>
### PIPE3 Control Register [PIPE3CTR] <Address: H'074>
### PIPE4 Control Register [PIPE4CTR] <Address: H'076>
### PIPE5 Control Register [PIPE5CTR] <Address: H'078>

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | BSTS | INBUFM | — | — | — | ATREPM | ACLRM | SQCLR | SQSET | SQMON | PBUSY | — | — | — | PID | |
| Initial Value | 0 | 0 | 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(0)/W(1) | R(0)/W(1) | R | R | R | R | R | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 | BSTS | 0 | R | Buffer status<br>This bit indicates the FIFO buffer status of the specified pipe.<br>0: Buffer access is not possible.<br>1: Buffer access is possible. |
| 14 | INBUFM | 0 | R | Transmit buffer monitor<br>When the specified pipe is in the sending direction, this bit indicates the FIFO buffer status of the specified pipe.<br>0: The FIFO buffer does not contain data that can be sent.<br>1: The FIFO buffer contains data that can be sent. |
| 13 to 11 | — | All 0 | R | Nothing is assigned to these bits. Fix these bits to 0. |
| 10 | ATREPM | 0 | R/W | Automatic response mode<br>This bit prohibits or enables automatic response of the specified pipe.<br>0: Automatic response prohibited<br>1: Automatic response enabled. (A zero-length packet response is sent during transmission. For reception, an NAK response is sent and an NRDY interrupt is generated.) |
| 9 | ACLRM | 0 | R/W | Automatic buffer clear mode<br>This bit prohibits or enables automatic buffer clear mode for the specified pipe.<br>0: Prohibited<br>1: Enabled (all buffers are initialized) |
| 8 | SQCLR | 0 | R(0)/W(1) | Toggle bit clear<br>Specify 1 in this bit to clear the expected value of the sequence toggle bit for the next transaction in the specified pipe to DATA0.<br>0: Writing disabled<br>1: DATA0 specified |
| 7 | SQSET | 0 | R(0)/W(1) | Toggle bit set<br>Specify 1 in this bit to set the expected value of the sequence toggle bit for the next transaction in the specified pipe to DATA1.<br>0: Writing disabled<br>1: DATA1 specified |
| 6 | SQMON | 0 | R | Sequence toggle bit monitor<br>This bit indicates the expected value of the sequence toggle bit for the next transaction in the specified pipe.<br>0: DATA0<br>1: DATA1 |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 5 | PBUSY | 0 | R | Pipe busy<br>This bit indicates whether the specified pipe is being used in the USB bus.<br>0: The specified pipe is not used in the USB bus.<br>1: The specified pipe is being used in the USB bus. |
| 4 to 2 | — | All 0 | R | Nothing is assigned to these bits. Fix these bits to 0. |
| 1, 0 | PID | All 0 | R/W | Response PID<br>These bits specify the response method for the next transaction in the specified pipe.<br>00: NAK response<br>01: BUF response (according to the buffer status)<br>10: STALL response<br>11: STALL response |

### (1) Buffer status (BSTS) bit

This bit indicates whether the CPU can access the FIFO buffer allocated to the selected pipe. This bit is operated by this module.

The meaning of this bit varies as follows depending on the value of the values of the DIR, BFRE, and DCLRM bits:

Table 32B.14 BSTS Bit Operations

| DIR bit Setting Value | BFRE bit Setting Value | DCLRM bit Setting Value | Meaning of the BSTS Bit |
|---|---|---|---|
| 0 | 0 | 0 | This bit indicates 1 when reading of received data from the FIFO buffer becomes possible, and indicates 0 when reading data has finished. |
| | | 1 | Setting prohibited |
| | 1 | 0 | This bit indicates 1 when reading of received data from the FIFO buffer becomes possible. This bit indicates 0 when 1 is written in the BCLF bit after reading data has finished. |
| | | 1 | This bit indicates 1 when reading of received data from the FIFO buffer becomes possible, and indicates 0 when reading data has finished. |
| 1 | 0 | 0 | This bit indicates 1 when writing of send data in the FIFO buffer becomes possible, and indicates 0 when writing data has finished. |
| | | 1 | Setting prohibited |
| | 1 | 0 | Setting prohibited |
| | | 1 | Setting prohibited |

### (2) Transmit buffer monitor (INBUFM) bit

When the selected pipe is in the sending direction (DIR = 1), this module sets this bit when software (or the DMAC) has finished writing data to at least one FIFO buffer.

This module clears this bit when this module finishes sending all data from the FIFO buffer to which the data has been written. In double-buffer mode (DBLB = 1), this module clears this bit when this module has finished sending all data from the two FIFO buffers and software (or the DMAC) has not yet finished writing data to one FIFO buffer.

When the selected pipe is in the receiving direction (DIR = 0), this bit indicates the same value as that of the BSTS bit.

### (3) Automatic response mode (ATREPM) bit

This bit can be set when the transfer type of the selected pipe is bulk transfer.

When this bit is 1, this module responds to tokens sent from the USB host as described below.

1. When the selected pipe is used for bulk IN transfer (TYPE = 01 and DIR = 1)
   When the ATREPM bit is 1 and the PID of the selected pipe is BUF, this module responds to an IN token by sending a zero-length packet.
   Each time this module receives ACK from the USB host (the sequence of one transaction is receiving an IN token, sending a zero-length packet, and then receiving ACK), this module updates (toggles) the sequence toggle bit (DATA- PID).
   This module does not generate BRDY and BEMP interrupts.

2. When the selected pipe is used for bulk OUT transfer (TYPE = 01 and DIR = 0)
   When the ATREPM bit is 1 and the PID of the selected pipe is BUF, this module responds to an OUT token (or a PING token) by sending an NAK response and generates an NRDY interrupt.

Changing the value of this bit for a selected pipe must be done while the PID of the selected pipe is NAK.

If you change the value of this bit for a selected pipe after having changed the PID of the selected pipe from BUF to NAK, check, that the PBUSY bit is 0 before changing the value of this bit. Note, however, that, if this module has changed the PID of the selected pipe to NAK, the value of the PBUSY bit need not be checked.

To perform a USB communication with this bit set, the FIFO buffer must be empty. No data must be written to the FIFO buffer during the USB communication with this bit set.

When the transfer type of the selected pipe is isochronous transfer, this bit must always be 0.

### (4) Automatic buffer clear mode (ACLRM) bit

When you need to clear the whole FIFO buffer allocated to the selected pipe, write 1 and 0 successively to the ACLRM bit.

**Table 32B.15** shows the buffer contents this module clears when 1 and 0 are written successively to the ACLRM bit.

**Table 32B.16** shows the cases that require this processing.

Table 32B.15 Buffer Contents This Core Clears When the ACLRM Bit is Set

| No. | Contents to be Cleared by Setting the ACLRM Bit |
|---|---|
| (1) | Whole contents of the FIFO buffer allocated to the specified pipe (if the double buffer is set, both FIFO buffers are cleared) |
| (2) | If the transfer type of the specified pipe is Isochronous transfer, the interval count value is cleared. |

Table 32B.16 Cases Requiring the ACLRM Bit to be Set

| No. | Cases when Data Clear is Required |
|---|---|
| (1) | The whole contents of the FIFO buffer allocated to the specified pipe needs to be cleared. |
| (2) | The interval count value needs to be reset. |
| (3) | The value of the BFRE bit is changed. |
| (4) | The value of the DBLB bit is changed. |
| (5) | Forced termination of the transaction count function is performed. |

Changing the value of this bit for a selected pipe must be done while the PID of the selected pipe is NAK and the selected pipe is not specified in the CURPIPE bits.

If you change the value of this bit for a selected pipe after having changed the PID of the selected pipe from BUF to NAK, check, that, and the PBUSY bit is 0 before changing the value of this bit. Note, however, that, if this module has changed the PID of the selected pipe to NAK, the value of the PBUSY bit need not be checked.

### (5) Sequence toggle bit clear (SQCLR) bit

When software sets this bit, this module specifies DATA0 as the expected value of the sequence toggle bit for the selected pipe. This module always clears this bit.

Writing 1 to the SQCLR bit must be done when the PID of the selected pipe is NAK.

If you write 1 to this bit for a selected pipe after having changed the PID of the selected pipe from BUF to NAK, check, that the PBUSY bit is 0 before changing the value of this bit. Note, however, that, if this module has changed the PID of the selected pipe to NAK, the value of the PBUSY bit need not be checked.

### (6) Sequence toggle bit set (SQSET) bit

When software sets this bit, this module specifies DATA1 as the expected value of the sequence toggle bit for the selected pipe. This module always clears this bit.

Writing 1 to the SQSET bit must be done when the PID of the selected pipe is NAK.

If you write 1 to this bit for a selected pipe after having changed the PID of the selected pipe from BUF to NAK, check, that the PBUSY bit is 0 before changing the value of this bit. Note, however, that, if this module has changed the PID of the selected pipe to NAK, the value of the PBUSY bit need not be checked.

### (7) Sequence toggle bit monitor (SQMON) bit

This bit indicates the expected value of the sequence toggle bit for the selected pipe. This bit is operated by this module.

When the transfer type of the selected pipe is other than isochronous transfer, this module toggles this bit when a transaction ends normally. This module, however, does not toggle this bit if a DATA-PID mismatch occurs during a transfer in the receiving direction.

### (8) Pipe busy (PBUSY) bit

This module changes this bit from 0 to 1 when a USB transaction using the selected pipe starts. This module changes this bit from 1 to 0 when the transaction ends normally.

Reading this bit after software sets the PID to NAK enables you to check whether you can change pipe settings.

### (9) Response PID (PID) bits

These bits are used to specify, the type of response of this module for individual pipes.

The default of PID is NAK. When the selected pipe is used for USB transfers, the PID setting muse be changed to BUF. For the basic operations (without communication packet errors involved) of this module depending on the value of the PID bits, see **Table 32B.17**.

If you have changed the PID of a selected pipe from BUF to NAK while the selected pipe is performing a USB communication, check, that the PBUSY bit is 0 to confirm that the USB transfer through the selected pipe has actually changed to the NAK status. Note, however, that, if this module has changed the PID of the selected pipe to NAK, the value of the PBUSY bit need not be checked.

This module changes the value of the PID bits in the following cases:

1. When the selected pipe is in the receiving direction and set the SHTNAK bit for the selected pipe, this module sets the PID to NAK when this module recognizes the end of a transfer.
2. When this module has received a data packet of which the payload size is larger than the maximum packet size, this module sets the PID to STALL (PID = 11).
3. If this module detects a USB bus reset, this module sets the PID to NAK.

To change the PID from NAK (PID = 00) to STALL, write 10 to the PID bits. To change the PID from BUF (PID = 01) to STALL, write 11 to the PID bits.

To change the PID from STALL (PID = 11) to NAK, write 10 to the PID bits once, and then write 00 to the PID bits. To change the PID from STALL to BUF, change the PID to NAK once, and then change it to BUF.

Table 32B.17 Core Operations Depending on the PID Setting

| PID bit Setting Value | Transfer Type (TYPE Bit Setting Value) | Transfer Direction (DIR Bit Setting Value) | Operation of This Core |
|---|---|---|---|
| 00 (NAK) | Bulk ("TYPE = 01"), or Interrupt ("TYPE = 10") | Independent of the setting value | Sends a NAK response for a token from the USB host. |
| | Isochronous ("TYPE = 11") | Receiving direction ("DIR = 0") | Does not respond to a token from the USB host. |
| | | Sending direction ("DIR = 1") | Sends a zero-length packet for a token from the USB host. |
| 01 (BUF) | Bulk ("TYPE = 01") | Receiving direction ("DIR = 0") | For an OUT token from the USB host, if the FIFO buffer corresponding to the specified PIPE can receive data, this module receives data and then sends an ACK or NYET response. If receiving data is not possible, this module sends an NAK response.<br>For a PING Token from the USB host, if the FIFO buffer corresponding to the specified PIPE can receive data, this module sends an ACK response. If receiving data is not possible, this module sends an NAK response. |
| | Interrupt ("TYPE = 10") | Receiving direction ("DIR = 0") | For an OUT token from the USB host, if the FIFO buffer corresponding to the specified PIPE can receive data, this module receives data and then sends an ACK response. If receiving data is not possible, this module sends an NAK response. |
| | Bulk ("TYPE = 01") or Interrupt ("TYPE = 10") | Sending direction ("DIR = 1") | If the corresponding FIFO buffer is available for sending data, this module sends data in response to a token from the USB host. If sending data is not possible, this module sends an NAK response. |
| | Isochronous ("TYPE = 11") | Receiving direction ("DIR = 0") | For an OUT token from the USB host, if the FIFO buffer corresponding to the specified PIPE can receive data, this module receives data. If receiving data is not possible, this module discards data. |
| | | Sending direction ("DIR = 1") | If the corresponding FIFO buffer is available for sending data, this module sends data in response to a token from the USB host. If sending data is not possible, this module sends a zero-length packet. |
| 10 (STALL) or 11 (STALL) | Bulk ("TYPE = 01") or Interrupt ("TYPE = 10") | Independent of the setting value | Sends a STALL response for a token from the USB host. |
| | Isochronous ("TYPE = 11") | Independent of the setting value | Does not respond to a token from the USB host. |

### 32B.2.16.2 PIPE6 Control Register [PIPE6CTR] <Address: H'07A>
### PIPE7 Control Register [PIPE7CTR] <Address: H'07C>
### PIPE8 Control Register [PIPE8CTR] <Address: H'07E>
### PIPE9 Control Register [PIPE9CTR] <Address: H'080>

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | BSTS | — | — | — | — | — | ACLRM | SQCLR | SQSET | SQMON | PBUSY | — | — | — | PID | |
| Initial Value | 0 | 0 | 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(0)/W(1) | R(0)/W(1) | R | R | R | R | R | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 | BSTS | 0 | R | Buffer status<br>This bit indicates the FIFO buffer status of the specified pipe.<br>0: Buffer access is not possible.<br>1: Buffer access is possible. |
| 14 to 10 | — | All 0 | R | Nothing is assigned to these bits. Fix these bits to 0. |
| 9 | ACLRM | 0 | R/W | Automatic buffer clear mode<br>This bit prohibits or enables automatic buffer clear mode for the specified pipe.<br>0: Automatic buffer clear mode prohibited<br>1: Automatic buffer clear mode enabled (all buffers are initialized) |
| 8 | SQCLR | 0 | R(0)/W(1) | Toggle bit clear<br>Specify 1 in this bit to clear the expected value of the sequence toggle bit for the next transaction in the specified pipe to DATA0.<br>0: Disabled<br>1: DATA0 specified |
| 7 | SQSET | 0 | R(0)/W(1) | Toggle bit set<br>Specify 1 in this bit to set the expected value of the sequence toggle bit for the next transaction in the specified pipe to DATA1<br>0: Disabled<br>1: DATA1 specified |
| 6 | SQMON | 0 | R | Toggle bit monitor<br>This bit indicates the expected value of the sequence toggle bit for the next transaction in the specified pipe.<br>0: DATA0<br>1: DATA1 |
| 5 | PBUSY | 0 | R | Pipe busy<br>This bit indicates whether the specified pipe is being used in the USB bus.<br>0: The specified pipe is not used in the USB bus.<br>1: The specified pipe is being used in the USB bus. |
| 4 to 2 | — | All 0 | R | Nothing is assigned to these bits. Fix these bits to 0. |
| 1, 0 | PID | All 0 | R/W | Response PID<br>These bits specify the response method for the next transaction in the specified pipe<br>00: NAK response<br>01: BUF response (according to the buffer status)<br>10: STALL response<br>11: STALL response |

#### (1) Buffer status (BSTS) bit

See **Section 32B.2.16.1(2), Transmit buffer monitor (INBUFM) bit**.

#### (2) Automatic buffer clear mode (ACLRM) bit

When you need to clear the whole FIFO buffer allocated to the selected pipe, write 1 and 0 successively to the ACLRM bit.

**Table 32B.18** shows the buffer contents this module clears when 1 and 0 are written successively to the ACLRM bit.

**Table 32B.19** shows the cases that require this processing.

Table 32B.18 Buffer Contents This Core Clears when the ACLRM Bit is Set

| No. | Contents Cleared by ACLRM Bit Operation |
|---|---|
| (1) | All contents of the FIFO buffer allocated to the selected pipe |

Table 32B.19 Cases Requiring the ACLRM Bit to be Set

| No. | Contents Cleared by ACLRM Bit Operation |
|---|---|
| (1) | When clearing contents of the FIFO buffer allocated to the selected pipe |
| (2) | When resetting the interval counter |
| (3) | When the value of the BFRE bit is changed |
| (4) | When the transaction count function is terminated forcibly |

Changing the value of this bit for a selected pipe must be done while the PID of the selected pipe is NAK and the selected pipe is not specified in the CURPIPE bits.

If you change the value of this bit for a selected pipe after having changed the PID of the selected pipe from BUF to NAK, check, that, and the PBUSY bit is 0 before changing the value of this bit. Note, however, that, if this module has changed the PID of the selected pipe to NAK, the value of the PBUSY bit need not be checked.

#### (3) Sequence toggle bit clear (SQCLR) bit

See **Section 32B.2.16.1(5), Sequence toggle bit clear (SQCLR) bit**.

#### (4) Sequence toggle bit set (SQSET) bit

See **Section 32B.2.16.1(6), Sequence toggle bit set (SQSET) bit**.

#### (5) Sequence toggle bit monitor (SQMON) bit

See **Section 32B.2.16.1(7), Sequence toggle bit monitor (SQMON) bit**.

#### (6) Pipe busy (PBUSY) bit

See **Section 32B.2.16.1(8), Pipe busy (PBUSY) bit**.

#### (7) Response PID (PID) bits

See **Section 32B.2.16.1(9), Response PID (PID) bits**.

## 32B.2.17 Transaction Counters (PIPExTRE)

### 32B.2.17.1 PIPE1 Transaction Counter Enable Register [PIPE1TRE] <Address: H'090>
### PIPE2 Transaction Counter Enable Register [PIPE2TRE] <Address: H'094>
### PIPE3 Transaction Counter Enable Register [PIPE3TRE] <Address: H'098>
### PIPE4 Transaction Counter Enable Register [PIPE4TRE] <Address: H'09C>
### PIPE5 Transaction Counter Enable Register [PIPE5TRE] <Address: H'0A0>

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | TRENB | TRCLR | — | — | — | — | — | — | — | — |
| 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(0)/W(1) | R | R | R | R | R | R | R | R |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 to 10 | — | All 0 | R | Nothing is assigned to these bits. Fix these bits to 0. |
| 9 | TRENB | 0 | R/W | Transactio n counter enable<br>This bit enables or disables the transaction counter.<br>0: Disables the transaction counter.<br>1: Enables the transaction counter. |
| 8 | TRCLR | 0 | R(0)/W(1) | Transactio n counter clear<br>This bit clears the transaction counter to 0. To clear the counter, write 1 to this bit.<br>0: Invalid<br>1: Clears the current transaction counter. |
| 7 to 0 | — | All 0 | R | Nothing is assigned to these bits. Fix these bits to 0. |

**Note:** Changing values of these register bits for a selected pipe must be done when the PID of the selected pipe is NAK.
If you change the value of a bit for a selected pipe after having changed the PID of the selected pipe from BUF to NAK, check, that the PBUSY bit is 0 before changing the value of the bit. Note, however, that, if this module has changed the PID of the selected pipe to NAK, the value of the PBUSY bit need not be checked.

### (1) Transaction counter enable (TRENB) bit

When software sets this bit for the selected pipe in the receiving direction after specifying a total number of packets in the TRNCNT bits, this module performs the following control when it finishes receiving the same number of packets as the total number specified in the TRNCNT bits:

1. When the continuous transfer mode is used (CNTMD = 1), this module switches the FIFO buffer to the CPU side at the end of reception even if the FIFO buffer is not full.
2. When the SHTNAK bit is 1, this module changes the PID of the selected pipe to NAK when it finishes receiving the same number of packets as the total number specified in the TRNCNT bits.
3. When the DENDE bit is 1 and the PKTMD bit is 0, this module asserts the DEND signal when reading the last data after having received the same number of packets as the total number specified in the TRNCNT bits.
4. When the BFRE bit is 1, this module asserts the BRDY interrupt signal when it finishes reading the last data after having received the same number of packets as the total number specified in the TRNCNT bits.

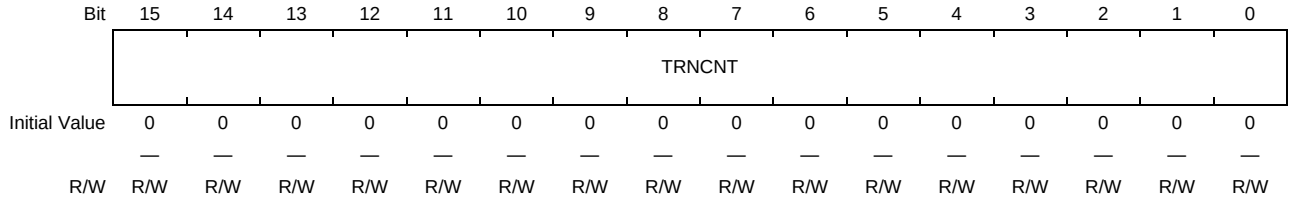
For the pipe in the sending direction, write 0 to this bit (TRENB bit). When not using the transaction count function, write 0 to this bit.

When using the transaction count function, specify a value in the TRNCNT bits before writing 1 to this bit. Also, write 1 to this bit before receiving the first packet among those to be counted by the transaction count function.

### (2) Transaction counter clear (TRCLR) bit

When software sets this bit for a selected pipe, this module clears the current value of the transaction counter corresponding to the selected pipe, and then clears this bit.

### 32B.2.17.2 PIPE1 Transaction Counter Register [PIPE1TRN] <Address: H'092>
### PIPE2 Transaction Counter Register [PIPE2TRN] <Address: H'096>
### PIPE3 Transaction Counter Register [PIPE3TRN] <Address: H'09A>
### PIPE4 Transaction Counter Register [PIPE4TRN] <Address: H'09E>
### PIPE5 Transaction Counter Register [PIPE5TRN] <Address: H'0A2>




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 to 0 | TRNCNT | All 0 | R/W | When writing:<br>Specifies the total number of packets to be received by the pertinent pipe (number of transactions).<br>When reading:<br>Indicates the specified number of transactions if TRENB is 0.<br>Indicates the number of the currently counted transaction if TRENB is 1. |

#### (1) Transaction counter (TRNCNT) bits

When software writes 1 to the TRENB bit after setting the total number of packets to be received for the selected pipe in the receiving direction, this module performs the control described in **Section 32B.2.17.1(1), Transaction counter enable (TRENB) bit**.

When the TRENB bit is 0, this module indicates, by these bits, the number of transactions set.

When the TRENB bit is 1, this module indicates, by these bits, the current number of transactions counted.

This module increments the value of the TRNCNT bits by 1 when the status of reception meets all the following conditions (a) to (c):

(a) The TRENB bit is 1.

(b) When a packet is received, the value of the TRCNT bits is not equal to "current count + 1."

(c) The payload size of received packets has reached the value of the MXPS bits.

This module clears the TRNCNT bits to 0 when any of the following conditions (1) to (3) is met:

(1) All the following conditions (a) to (c) are met:

(a) The TRENB bit is 1.

(b) When a packet is received, the value of the TRCNT bits is equal to "current count + 1. "

(c) The payload size of received packets has reached the value of the MXPS bits.

(2) Both of the following conditions (a) and (b) are met:

(a) The TRENB bit is 1.

(b) A short packet has been received.

(3) The following condition is met:

(a) Written 1 to the TRCLR bit.

For the pipe in the sending direction, write 0 to these bits (TRNCNT bits). When not using the transaction count function, write 0 to these bits.

Changing the value of these bits for a selected pipe must be done while the PID of the selected pipe is NAK and the TRENB bit is 0.

If you change the value of this bit for a selected pipe after having changed the PID of the selected pipe from BUF to NAK, check, that, and the PBUSY bit is 0 before changing the value of this bit. Note, however, that, if this module has changed the PID of the selected pipe to NAK, the value of the PBUSY bit need not be checked.

If you change the value of these bits, write 1 to the TRCLR bit before writing 1 to the TRENB bit.

## 32B.2.18 Low Power Control Register

### 32B.2.18.1 Low Power Control Register [LPCTRL] <Address: H'100>

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | HWUPM | — | — | — | — | — | — | — |
| 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 |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 to 8 | — | All 0 | R | Reserved<br>Nothing is assigned to these bits. Fix these bits to 0. |
| 7 | HWUPM | 0 | R/W | 0: Resumes the PHY from the low-power mode while the internal bus clock (P1ϕ) is operating.<br>1: Enables resume from the low-power mode while the internal bus clock (P1ϕ) is stopped. |
| 6 to 0 | — | All 0 | R | Nothing is assigned to these bits. Fix these bits to 0. |

#### (1) HWUPM

This bit is used to specify whether to enable resumption from the low-power mode even while the internal bus clock (P1ϕ) is stopped.

0: Disables resumption while the internal bus clock (P1ϕ) is stopped.

1: Enables resumption while the internal bus clock (P1ϕ) is stopped.

This bit specifies whether to detect resume signaling while the internal bus clock (P1ϕ) is stopped. Whether to resume is controlled by the L1EXTMD bit. To resume from the low-power mode (LPM L1 state) while the internal bus clock (P1ϕ) is stopped, set both this bit and the L1EXTMD bit.

## 32B.2.19 Low Power Status Register

### 32B.2.19.1 Low Power Status Register [LPSTS] <Address: H'102>

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | SUSPM | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| 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 | — R | — R | — R | — R | — R |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 | — | 0 | R | Nothing is assigned to this bit. Fix this bit to 0. |
| 14 | SUSPM | 0 | R/W | USBPHY Suspend M control<br>This bit controls the Suspend M signal to the USBPHY.<br>0: USBPHY suspend mode<br>1: USBPHY normal mode |
| 13 to 0 | — | All 0 | R | Nothing is assigned to these bits. Fix these bits to 0. |

#### (1) USBPHY SuspendM control (SUSPM) bit

This bit is used to control the SuspendM signal to the USBPHY. By default, the value of this bit is 0 and the USBPHY is in suspend mode. To operate this module, write 1 to this bit.

When the SUSPM bit is 0 (that is, when the UTMI clock is stopped), data cannot be written to this module, but can only be read from this module. Note, however, that data can be written to the registers listed in **Table 32B.20**.

Table 32B.20 Registers That Allow Writing when the SUSPM Bit is 0

| Address | Register Name |
|---|---|
| H'000 | SYSCFG0 |
| H'002 | BUSWAIT |
| H'100 | LPCTRL |
| H'102 | SUSPMODE |

Note that the values written to the SYSCFG0 register while the USBPHY clock is stopped (SUSPM = 0) will be applied after the USBPHY clock starts (SUSPM = 1).

When the L1EXTMD bit is 0, this bit (SUSPM bit) is controlled (set or cleared) by software. When L1EXTMD bit is 1, this bit is controlled by software for the transition to the L1 or L2 state, and controlled by hardware for resumption from the L1 or L2 state, regardless of the level (L1 or L2).

## 32B.2.20 PHY Function Control Register

### 32B.2.20.1 PHY Function Control Register [PHYFUNCTR] <Address: H'104>

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | SusMon | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| 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 | — | 0 | R | Nothing is assigned to this bit. Fix this bit to 0. |
| 14 | SusMon | 0 | R | This bit allows reading of the status of the Suspend M signal. |
| 13 to 0 | — | All 0 | R | Nothing is assigned to these bits. Fix these bits to 0. |

#### (1) SusMon

SuspendM monitor bit (read only)

The status of the Suspend M signal can be read.

## 32B.2.21 PHY_OTG Control Register (PHYOTGCTR)

### 32B.2.21.1 PHY_OTG Control Register [PHYOTGCTR] <Address: H'10A>

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | DmPuDwn | DpPuDwn | — | — | — | — | — | — | — | — | — |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 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 11 | — | All 0 | R | Nothing is assigned to these bits. Fix these bits to 0. |
| 10 | DmPuDwn | 1 | R | Dm Pulldown monitor bit |
| 9 | DpPuDwn | 1 | R | Dp Pulldown monitor bit |
| 8 to 0 | — | All 0 | R | Nothing is assigned to these bits. Fix these bits to 0. |

#### (1) DmPuDwn

DmPulldown monitor bit (read only)

0: 15-kΩ Pulldown resistor control on the DM side is disabled.

1: 15-kΩ Pulldown resistor control on the DM side is enabled.

#### (2) DpPuDwn

DpPulldown monitor bit (read only)

0: 15-kΩ Pulldown resistor control on the DP side is disabled.

1: 15-kΩ Pulldown resistor control on the DP side is enabled.

## 32B.2.22 Peripheral L1 Control Register 1 (PL1CTRL)

### 32B.2.22.1 Peripheral L1 Control Register 1 [PL1CTRL1] <Address: H'144>

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | L1EXTMD | — | — | HIRDTHR[3:0] | | | | DVSQ[3:0] | | | | L1NEGOMD | L1RESPMD[1:0] | | L1RESPEN |
| 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 | R | R | R | R/W | R/W | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 | — | 0 | R | Nothing is assigned to this bit. Fix this bit to 0. |
| 14 | L1EXTMD | 0 | R/W | USBPHY control mode on resumption from the L1 state<br>This bit controls the USBPHY resume operation on resumption from the L1 state.<br>0: Does not set the Suspend M bit when the Host K signal is received.<br>1: Sets the Suspend M bit when the Host K signal is received |
| 13, 12 | — | All 0 | R | Nothing is assigned to these bits. Fix these bits to 0. |
| 11 to 8 | HIRDTHR[3:0] | All 0 | R/W | L1 response negotiation threshold<br>HIRD threshold to be used for the L1NEGOMD bit<br>The format is the same as the HIRD field of the HL1CTRL register. |
| 7 | DVSQ[3] | 0 | R | DVSQ extension bit<br>This bit combined with the device state (DVSQ[2:0]) bit indicates the L1 state.<br>0000b: Powered state<br>0001b: Default state<br>0010b: Address state<br>0011b: Configured state<br>01xxb: Suspended state<br>10xxb: L1 state |
| 6 to 4 | DVSQ[2:0] | All 0 | R | These bits mirror the DVSQ[2:0] bits in INTSTS0. |
| 3 | L1NEGOMD | 0 | R/W | L1 response negotiation control<br>This bit sets the negotiation function using for the HIRD value.<br>0: Returns an ACK response if the received HIRD value is larger than the value of the HIRDTHR[3:0] bits.<br>In other cases (including same values), an NYET response is returned.<br>1: Returns an ACK response if the received HIRD value is smaller than the value of the HIRDTHR[3:0] bits.<br>In other cases (including same values), an NYET response is returned.<br>This bit is valid only when the value of the L1RESPMD[1:0] bit is 11b. |
| 2, 1 | L1RESPMD[1:0] | All 0 | R/W | L1 response mode<br>These bits specify how to respond to an LPM token.<br>00b: NYET<br>01b: ACK<br>10b: STALL<br>11b: Response according to the value of the L1NEGOMD bit |
| 0 | L1RESPEN | 0 | R/W | L1 response enable<br>This bit enables an L1 response.<br>0: Does not support LPM.<br>1: Supports LPM. |

### (1) L1 EXT mode (L1EXTMD) bit

This bit specifies how to control the SuspendM bit upon receiving the Host K signal when the USBPHY is stopped by setting the SuspendM bit in the L1 state.

0: Does not set the SuspendM bit when this module is resumed from the L1 state.

1: Sets the SuspendM bit when this module is resumed from the L1 state.

NOTES

1. The Host K period lasts a minimum of 50 µs. Therefore, the USBPHY might be unable to be resumed within the Host K period if the software settings for resume same as those for the suspend state are applied. Because the initial value of this bit is controlled by software, set this bit at initialization when the L1 state is to be supported.
2. For the transition to the L1 state, the SuspendM bit is controlled by software regardless of the value of this bit.
3. When this bit is set, the SuspendM bit will be set also at resumption from the L2 state.

### (2) HIRD negotiation threshold (HIRDTHR[3:0]) bits

These bits specify the value of HIRD threshold to be used for the negotiation specified by the L1NEGOMD bit.

The format of the value is the same as that of the HIRD field in the HL1CTRL register.

### (3) Device state extension (DVSQ[3]) bit

This bit is used as the fourth bit for the device state (DVSQ) bits.

0000b: Powered state

0001b: Default state

0010b: Address state

0011b: Configured state

01xxb: Suspended state

10xxb: L1 state

### (4) Device status (DVSQ[2:0]) bits

These bits mirror the DVSQ[2:0] bits in the Interrupt Status Register (INTSTS0).

### (5) L1 negotiation mode (L1NEGOMD) bit

This bit is used to specify the negotiation function using the HIRD value.

0: Returns an ACK response when the received HIRD value is larger than the value in HIRDTHR[3:0] bits, or returns an NYET response in other cases.

1: Returns an ACK response when the received HIRD value is smaller than the value in HIRDTHR[3:0] bits, or returns an NYET response in other cases.

This bit is valid only when the value of L1RESPMD[1:0] bits is 11b.

### (6) L1 response mode (L1RSPMD[1:0]) bits

When the L1RSPED bit is set, this module responds to an LPM token according to the value of these bits. These bits specify how to respond to the LPM token.

00b: NYET

01b: ACK

10b: STALL

11b: Response according to the value of L1NEGOMD bit

### (7) L1 response enable (L1RSPEN) bit

When this bit is 0, this module does not respond to the LPM token it receives. When this bit is 1, this module responds to the LPM token (it receives) according to the value of the LPMRESPMD[1:0] bits.

## 32B.2.23 Peripheral L1 Control Register 2

### 32B.2.23.1 Peripheral L1 Control Register 2 [PL1CTRL2] <Address: H'146>

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | RWEMON | HIRDMON[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/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 to 13 | — | All 0 | R | Nothing is assigned to these bits. Fix these bits to 0. |
| 12 | RWEMON | 0 | R/W | This bit reflects the value of the RWE bit in the LPM token received last. |
| 11 to 8 | HIRDMON[3:0] | All 0 | R/W | These bits reflect the value of the HIRD field in the LPM token received last. |
| 7 to 0 | — | All 0 | R | Nothing is assigned to these bits. Fix these bits to 0. |

#### (1) RWE value monitor (RWEMON) bit

This bit is referenced to monitor the value of the RWE bit in a received LPM token.

This bit reflects the value of the RWE bit in the LPM token received last.

#### (2) HIRD value monitor (HIRDMON) bits

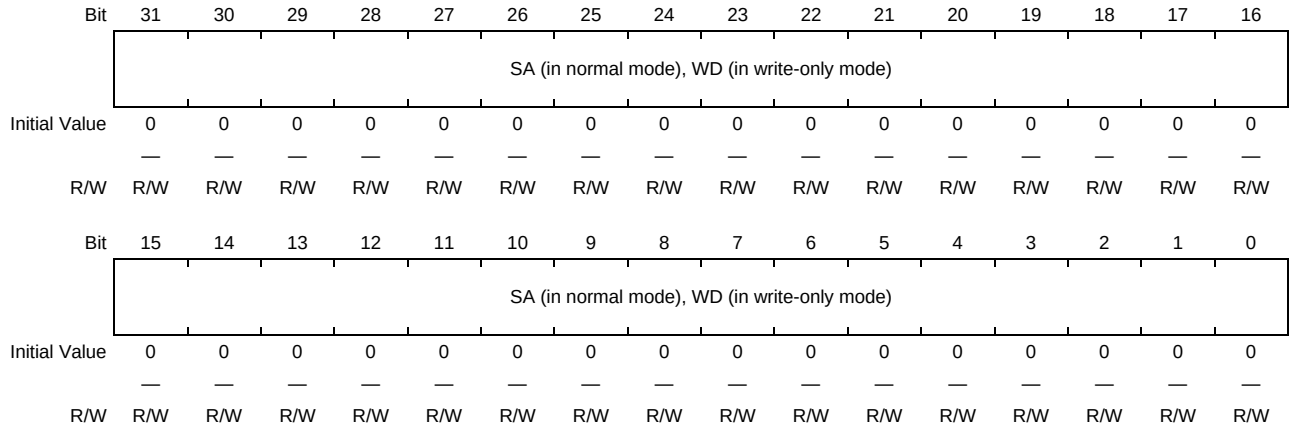
These bits are referenced to monitor the value of the HIRD field in a received LPM token.

These bits reflect the value of the HIRD field in the LPM token received last.

# 32B.3 Next Register Set

## 32B.3.1 Next Source Address Register n

### 32B.3.1.1 Next0 Source Address Register ch0 [N0SA_0] <Address: H'400>
### Next1 Source Address Register ch0 [N1SA_0] <Address: H'40C>
### Next0 Source Address Register ch1 [N0SA_1] <Address: H'440>
### Next1 Source Address Register ch1 [N1SA_1] <Address: H'44C>

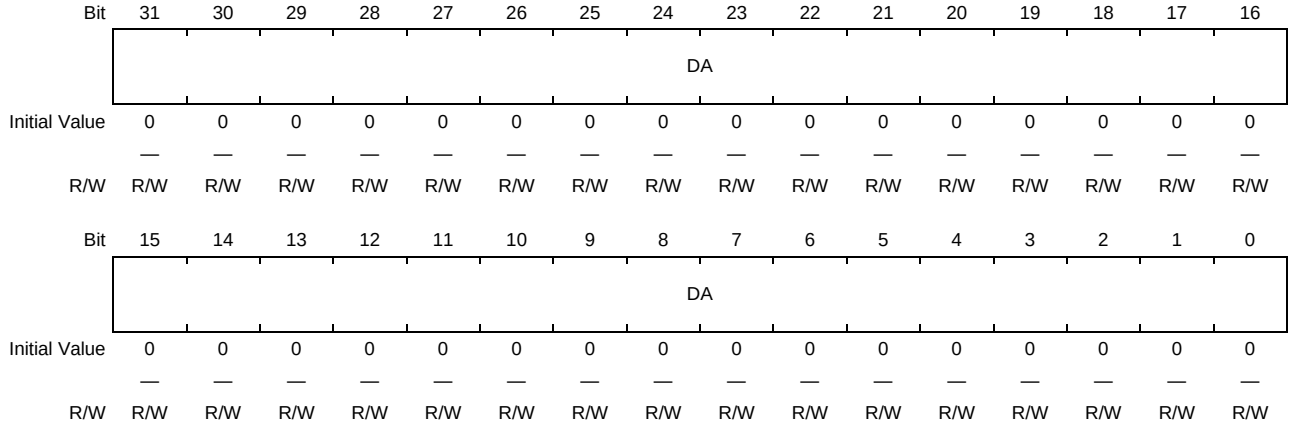



| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 0 | SA (in normal mode) | All 0 | R/W | Source Address<br>These bits specify the start address of the DMA transfer source. |
| | WD (in write-only mode) | All 0 | R/W | Write Data<br>These bits specify the write data in write-only mode. |

**Note:** In a transfer in link mode, the data in the N0SA_n register is overwritten with descriptor read data.

## 32B.3.2 Next Destination Address Register n

### 32B.3.2.1 Next0 Destination Address Register ch0 [N0DA_0] <Address: H'404>
### Next1 Destination Address Register ch0 [N1DA_0] <Address: H'410>
### Next0 Destination Address Register ch1 [N0DA_1] <Address: H'444>
### Next1 Destination Address Register ch1 [N1DA_1] <Address: H'450>




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 0 | DA | All 0 | R/W | Destination Address<br>These bits specify the start address of the DMA transfer destination. |

**Note:** In a transfer in link mode, the data in the N0DA_n register is overwritten with descriptor read data.

## 32B.3.3 Next Transaction Byte Register n

### 32B.3.3.1 Next0 Transaction Byte Register ch0 [N0TB_0] <Address: H'408>
### Next1 Transaction Byte Register ch0 [N1TB_0] <Address: H'414>
### Next0 Transaction Byte Register ch1 [N0TB_1] <Address: H'448>
### Next1 Transaction Byte Register ch1 [N1TB_1] <Address: H'454>

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TB | | | | | | | | | | | | | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TB | | | | | | | | | | | | | | | |
| 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 | TB | All 0 | R/W | Transaction Byte<br>These bits specify the total number of transfer bytes.<br>*Note*: Do not start a DMA transaction with 0 set in these bits. |

**Note:** The N0TB_n register is overwritten by the descriptor read data during link mode transfer.

# 32B.4 Current Register Set

## 32B.4.1 Current Source Address Register

### 32B.4.1.1 Current Source Address Register ch0 [CRSA_0] <Address: H'418>
### Current Source Address Register ch1 [CRSA_1] <Address: H'458>

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | CRSA | | | | | | | | | | | | | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | CRSA | | | | | | | | | | | | | | | |
| 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 0 | CRSA | All 0 | R | Current Source Address Register<br>This register indicates the read address for the next DMA transaction. The value of this register is incremented automatically while DMA transactions are in process. (When the SAD bit in the CHCFG_n register is 1, the value of this register is fixed. When the WONLY bit in the CHCFG_n register is 1, the value of this register is undefined.)<br>The initial value of this register is loaded from the following register:<br>In register mode:<br>A transfer source address is loaded from the Next0/1 Register Set.<br>In link mode:<br>A transfer source address is loaded from the descriptor. (The descriptor read data is input to the N0SA_n register, and is loaded into the CRSA_n register when a transfer starts.)<br>The value of this register is incremented when a read transfer ends.<br>Read this register after DMA stops (that is, after the TACT bit in the CHSTAT_n register is cleared). (Handle the value read during DMA only as a reference value.) |

## 32B.4.2 Current Destination Address Register

### 32B.4.2.1 Current Destination Address Register ch0 [CRDA_0] <Address: H'41C>
### Current Destination Address Register ch1 [CRDA_1] <Address: H'45C>

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | CRDA | | | | | | | | | | | | | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | CRDA | | | | | | | | | | | | | | | |
| 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 0 | CRDA | All 0 | R | Current Destination Address Register<br>This register indicates the write address for the next DMA transaction. The value of this register is incremented automatically while DMA transactions are in process. (When the DAD bit in the CHCFG_n register is 1, the value of this register is fixed.)<br>The initial value of this register is loaded from the following register:<br>In register mode:<br>A transfer destination address is loaded from the Next0/1 Register Set.<br>In link mode:<br>A transfer destination address is loaded from the descriptor. (The descriptor read data is input to the N0DA_n register, and is loaded into the CRDA_n register when a transfer starts.)<br>The value of this register is incremented when a write transfer ends.<br>Read this register after DMA stops (that is, after the TACT bit in the CHSTAT_n register is cleared). (Handle the value read during DMA only as a reference value.) |

## 32B.4.3 Current Transaction Byte Register

### 32B.4.3.1 Current Transaction Byte Register ch0 [CRTB_0] <Address: H'420>
### Current Transaction Byte Register ch1 [CRTB_1] <Address: H'460>

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | CRTB | | | | | | | | | | | | | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | CRTB | | | | | | | | | | | | | | | |
| 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 0 | CRTB | All 0 | R | Current Transaction Byte Register<br>This register indicates the remaining number of transfer bytes in the ongoing DMA transaction.<br>The value of this register is decremented automatically while the DMA transaction is in process.<br>The initial value of this register is loaded from the following register:<br>In register mode:<br>The number of transfer bytes is loaded from the Next0/1 Register Set.<br>In link mode:<br>The number of transfer bytes is loaded from the descriptor. (The descriptor read data is input to the N0TB_n register, and is loaded into the CRTB_n register when a transfer starts.)<br>The value of this register is decremented when a write transfer ends.<br>Read this register after DMA stops (that is, after the TACT bit in the CHSTAT_n register is cleared). (Handle the value read during DMA only as a reference value.) |

# 32B.5 Channel Register Set

## 32B.5.1 Channel Status Register n

### 32B.5.1.1 Channel Status Register ch0 [CHSTAT_0] <Address: H'424>
### Channel Status Register ch1 [CHSTAT_1] <Address: H'464>

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | DNUM | | | | | | | | — | — | — | — | — | SWPRQ | DMARQM | INTM |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | MODE | DER | DW | DL | SR | TC | END | ER | SUS | TACT | RQST | 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 | R | R | R | R | R | R | R | R |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 24 | DNUM | All 0 | R | Data Number<br>These bits indicate the amount of valid data in the buffer.<br>The indicated amount of data is the amount of the data that was read from the source but has not yet been written to the destination. (Unit: byte)<br>Incrementing condition:<br>● A DMA read transfer ends.<br>Decrementing condition:<br>● A DMA write transfer ends.<br>Clearing conditions:<br>● A condition for clearing the EN bit is met.<br>● "1" is written to the SWRST bit in the CHCTRL_n register. |
| 23 to 19 | — | All 0 | R | Reserved area. Write 0 to these bits. When these bits are read, 0 is read. |
| 18 | SWPRQ | 0 | R | Sweep Request<br>This bit indicates the state of sweep request.<br>This bit indicates the state of software sweep request (which has been activated by the SETSSWPRQ bit in the CHCTRL_n register).<br>1: The sweep request signal has been asserted.<br>0: The sweep request signal has not been asserted.<br>Setting condition:<br>● The SETSSWPRQ bit in the CHCTRL_n register is asserted.<br>Clearing conditions:<br>● The buffer becomes empty because of sweep.<br>● "1" is written to the CLRHSWPRQM bit in the CHCTRL_n register.<br>● "1" is written to the SWRST bit in the CHCTRL_n register. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 17 | DMARQM | 0 | R | DMAREQ Mask<br>This bit indicates the state of temporary masking of the DMA transfer request from the USB control.<br>1: The request is temporarily masked.<br>0: The request is released from temporary masking.<br>Setting condition:<br>● The SETDMARQM bit in the CHCTRL_n register is set.<br>Clearing conditions:<br>● “1” is written to the CLRDMARQM bit in the CHCTRL_n register.<br>● “1” is written to the SWRST bit in the CHCTRL_n register. |
| 16 | INTM | 0 | R | Interrupt Mask<br>This bit indicates the state of temporary masking of the output from the USBFDMAmn interrupt.<br>1: The output is temporarily masked.<br>0: The output is released from temporary masking.<br>Setting condition:<br>● “1” is written to the SETINTM bit in the CHCTRL_n register.<br>Clearing conditions:<br>● “1” is written to the CLRINTM bit in the CHCTRL_n register.<br>● “1” is written to the SWRST bit in the CHCTRL_n register. |
| 15 to 12 | — | All 0 | R | Reserved area. Write 0 to these bits. When these bits are read, 0 is read. |
| 11 | MODE | 0 | R | DMA Mode<br>This bit indicates the DMA mode. The indicated value is the value of the DMS bit in the CHCFG_n register.<br>0: Register mode<br>1: Link mode |
| 10 | DER | 0 | R | Descriptor Error<br>This bit indicates whether the data read from the descriptor is invalid (LV = 0) (regardless of the value of the DIM bit in the CHCFG_n register).<br>0: No descriptor error has occurred.<br>1: A descriptor error has occurred.<br>Setting condition:<br>● In link mode, when the DRRP bit in the CHCFG_n register is 0, the LV bit value read from the descriptor is 0.<br>Clearing conditions:<br>● “1” is written to the CLRDER bit in the CHCTRL_n register.<br>● “1” is written to the SWRST bit in the CHCTRL_n register. |
| 9 | DW | 0 | R | Descriptor WriteBack<br>This bit indicates whether data is being written back to the descriptor. If a bus error occurs during write-back to the descriptor, this bit retains 1.<br>0: Status other than write-back to the header in link mode<br>1: (When the ER bit in the CHSTAT_n register is 0)<br>Data is being written back to the header in link mode.<br>(When the ER bit in the CHSTAT_n register is 1)<br>A bus error has occurred during write-back to the header in link mode.<br>Setting condition:<br>● Write-back to the header is started in link mode.<br>Clearing conditions:<br>● Write-back to the header in link mode ends with an OK response.<br>● “1” is written to the SWRST bit in the CHCTRL_n register. If the bit retains 1 because of an error response, this bit can be cleared only by setting the SWRST bit. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 8 | DL | 0 | R | Descriptor Load<br>This bit indicates whether data is being read from the descriptor. If a bus error occurs during descriptor reading, this bit retains 1.<br>0: Status other than descriptor reading<br>1: (When the ER bit is 0)<br>Descriptor reading is in process in link mode.<br>(When the ER bit is 1)<br>A bus error has occurred during descriptor reading in link mode.<br>Setting condition:<br>● Descriptor reading is started in link mode.<br>Clearing conditions:<br>● Descriptor reading in link mode ends with an OK response.<br>● “1” is written to the SWRST bit in the CHCTRL_n register. If the bit retains 1 because of an error response, this bit can be cleared only by setting the SWRST bit. |
| 7 | SR | 0 | R | Selected Register Set<br>In register mode, this bit indicates the register set that is selected.<br>0: Next0 Register Set<br>1: Next1 Register Set<br>Setting condition:<br>● The RSEL bit in the CHCFG_n register is set.<br>Clearing condition:<br>● The RSEL bit in the CHCFG_n register is cleared. |
| 6 | TC | 0 | R | Terminal Count<br>This status bit indicates whether the DMA transaction has ended. This bit is set only when the TCM bit in the CHCFG_n register is 0.<br>0: The DMA transfer has not ended.<br>1: The DMA transfer has ended.<br>Setting conditions:<br>● In register mode, transfer of the total number of transfer bytes specified in the CRTB bits ends.<br>● In link mode, when the WBD bit in the header of the descriptor is 1, transfer of the total number of transfer bytes specified in the CRTB bits ends.<br>● In link mode, when the WBD bit in the header of the descriptor is 0, write-back to the descriptor ends.<br>Clearing conditions:<br>● “1” is written to the CLRTC bit in the CHCTRL_n register.<br>● “1” is written to the SWRST bit in the CHCTRL_n register. |
| 5 | END | 0 | R | USBFDMAmn Interrupted<br>This bit indicates whether the DMA transaction has ended and an USBFDMAmn interrupt has occurred.<br>0: The DMA transfer has not ended.<br>1: The DMA transfer has ended.<br>Setting conditions:<br>● The condition for setting the TC bit is met, and the DEM bit in the CHCFG_n register is 0.<br>● In link mode, when the descriptor is read, the LV bit in the header is 0, and the DRRP and DIM bits in the CHCFG_n register are 0.<br>Clearing conditions:<br>● “1” is written to the CLREND bit in the CHCTRL_n register.<br>● “1” is written to the SWRST bit in the CHCTRL_n register. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 4 | ER | 0 | R | Error<br>This bit indicates whether an error response has been received and a DMAERR interrupt has occurred during the DMA transfer.<br>0: No error response has been received.<br>1: An error response has been received.<br>Setting condition:<br>● An error response is received in a bus cycle.<br>Clearing condition:<br>● “1” is written to the SWRST bit in the CHCTRL_n register. |
| 3 | SUS | 0 | R | Suspend<br>This bit indicates whether the channel is suspended. For details, see **Section 32B.9.13.8(2), Suspension**.<br>0: Channel_n is not suspended.<br>1: Channel_n is suspended.<br>Setting condition:<br>● “1” is written to the SETSUS bit in the CHCTRL_n register during the DMA transfer using channel_n, and, thereby, the inside of the channel is suspended.<br>Clearing conditions:<br>● “1” is written to the CLRSUS bit in the CHCTRL_n register.<br>● “1” is written to the CLREN bit in the CHCTRL_n register.<br>● A condition for clearing the EN bit in the CHSTAT_n register. |
| 2 | TACT | 0 | R | Transaction Active<br>This bit indicates whether the DMAC is operating. This bit is used to check whether the channel is stopped fully. For details, see **Section 32B.9.13.8, Transfer state**.<br>0: The DMA in channel_n is stopped.<br>1: The DMA in channel_n is operating.<br>Setting condition:<br>● “1” is written to the SETEN bit in the CHCTRL_n register (to start descriptor reading or wait for a DMA request).<br>Clearing condition:<br>● The internal state is the idle state (the EN bit in the register has been cleared, and all transfers have ended). |
| 1 | RQST | 0 | R | Request<br>This bit indicates whether a transfer request has been received.<br>0: No DMA transfer request has been received.<br>1: A DMA transfer request has been received.<br>Setting conditions:<br>● “1” is written to the STG bit in the CHCTRL_n register.<br>● A DMA transfer request is received from the USB control.<br>Clearing conditions:<br>● “1” is written to the SWRST bit in the CHCTRL_n register.<br>● “1” is written to the CLRRQ bit in the CHCTRL_n register.<br>● In single transfer mode (the TM bit in the CHCFG_n register is 0), a transfer is executed on the side specified by the REQD bit in the CHCFG_n register<br>● In register mode, all DMA transactions are complete (the REN bit in the CHCFG_n register is 0).<br>● In link mode, the DMA transfer of the last descriptor (LE = 1) ends.<br>● In link mode, a DMA transfer is stopped during descriptor reading (the LV bit is 0 and the DRRP bit in the CHCFG_n register is 0).<br>● In link mode, when the DEM bit in the CHCFG_n register is 0, a DMA transaction ends.<br>● The master interface receives a bus error signal. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 0 | EN | 0 | R | Enable<br>This bit indicates whether the operation of DMA channel n is enabled or stopped.<br>0: Operation is stopped.<br>1: Operation is enabled.<br>Setting conditions:<br>• “1” is written to the SETEN bit in the CHCTRL_n register.<br>Clearing conditions:<br>• “1” is written to the SWRST bit in the CHCTRL_n register.<br>• “1” is written to the CLREN bit in the CHCTRL_n register.<br>• An error response is received during transfer.<br>• In register mode, all DMA transactions are completed (the REN bit in the CHCFG_n register is 0).<br>• In link mode, DMA transfer of the last descriptor (the LE bit is 1) ends (if the WBD bit is 0, write- back to the descriptor ends).<br>• In link mode, reading of a descriptor is stopped (the LV bit is 0 and the DRRP bit in the CHCFG_n register is 0). |

Note 1. When the ER bit in the CHSTAT_n register is set, treat the corresponding series of transfers as invalid transactions.

Note 2. To interrupt a DMA transaction, mask or clear transfer requests or clear the EN bit in the CHSTAT_n register. (For the procedure to interrupt, see **Section 32B.9.13.8(3), Transfer suspension**.)

Note 3. If the DMA transfer request from the USB control and the transfer request by software (setting the STG bit in the CHCFG_n register) are used together, the cause of activating the request that takes effect cannot be identified. Therefore, design the system so that only one type of transfer requests is used.

Note 4. When using the transfer request by software, operate the STG bit for a new transfer request only after the DMA transfer requested last ends (after checking the end of the last DMA transfer by referencing the Current Register Set or another method).

## 32B.5.2 Channel Control Register n

### 32B.5.2.1 Channel Control Register ch0 [CHCTRL_0] <Address: H'428>
### Channel Control Register ch1 [CHCTRL_1] <Address: H'468>

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | CLRDMARQM | SETDMARQM | CLRINTM | SETINTM |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | SETSSWPRQ | — | SETREN | — | — | CLRSUS | SETUSS | CLRDER | CLRTC | CLRENDD | CLRRQ | SWRST | STG | CLREN | SETEN |
| 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/W | 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 20 | — | All 0 | R | Reserved area. Write 0 to these bits. When these bits are read, 0 is read |
| 19 | CLRDMARQM | 0 | R/W | Clear DMAREQ Mask<br>Writing 1 to this bit releases the DMA transfer requests from the USB control from temporary masking. Writing 1 to this bit also clears the DMARQM bit in the CHSTATn register.<br>When this bit is read, 0 is read.<br>1: Releases the DNA transfer requests from masking set by using the SETDMARQM bit.<br>0: Has no effect on operation. |
| 18 | SETDMARQM | 0 | R/W | SET DMAREQ Mask<br>Writing 1 to this bit temporarily masks the DMA transfer requests from the USB control. Writing 1 to this bit also sets the DMARQM bit in the CHSTATn register.<br>When this bit is read, 0 is read.<br>1: Masks the DMA transfer requests from the USB control.<br>0: Has no effect on operation. |
| 17 | CLRINTM | 0 | R/W | Clear Interrupt Mask<br>Writing 1 to this bit releases the USBFDMAmn interrupt from masking. Writing 1 to this bit also clears the INTM bit in the CHSTATn register.<br>Releasing the INT_DMA[n] pin output from masking when the LVINT bit in the DCTRL register and the END bit in the CHSTAT_n register are 1 activates the INT_DMA[n] pin output. (The pin output is not activated if the LVINT bit is 0.)<br>When this bit is read, 0 is read.<br>1: Releases the pin output from masking set by using the SETINTM bit.<br>0: Has no effect on operation. |
| 16 | SETINTM | 0 | R/W | SETINTMSet Interrupt Mask<br>Writing 1 to this bit temporarily masks the USBFDMAmn interrupt. Writing 1 to this bit also sets the INTM bit in the CHSTATn register.<br>When this bit is read, 0 is read.<br>1: Masks the USBFDMAmn interrupt.<br>0: Has no effect on operation. |
| 15 | — | 0 | R | Reserved area. Write 0 to this bit. When this bit is read, 0 is read. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 14 | SETSSWPRQ | 0 | R/W | Set Software Sweep Request<br>Writing 1 to this bit sweeps out the data stored in the buffer to the destination (see **Section 32B.9.13.3(1), Forced software sweeping request**).<br>When this bit is read, 0 is read.<br>1: Writes, to the destination, the data that is stored in the buffer and has not yet been written to the destination.<br>0: Has no effect on operation.<br>If the destination asserts a hardware request (REQD = 1), the sweep operation cannot be used. |
| 13 | — | 0 | R | Reserved area. Write 0 to this bit. When this bit is read, 0 is read. |
| 12 | SETREN | 0 | R/W | Set Register Set Enable<br>Writing 1 to this bit sets the REN bit in the CHCFG_n register.<br>When this bit is read, 0 is read.<br>1: Sets the REN bit in the CHCFG_n register.<br>0: Has no effect on operation. |
| 11, 10 | — | All 0 | R | Reserved area. Write 0 to these bits. When these bits are read, 0 is read. |
| 9 | CLRSUS | 0 | R/W | Clear Suspend<br>Writing 1 to this bit when the SUS bit in the CHSTAT_n register is 1 releases the ongoing DMA transfer from the suspended state.<br>When this bit is read, 0 is read.<br>1: Releases the ongoing DMA transfer from the suspended state.<br>0: Has no effect on operation. |
| 8 | SETSUS | 0 | R/W | Set Suspend<br>Writing 1 to this bit when the EN bit in the CHSTAT_n register is 1 suspends the ongoing DMA transfer.<br>When this bit is read, 0 is read.<br>1: Suspends the ongoing DMA transfer.<br>0: Has no effect on operation. |
| 7 | CLRDER | 0 | R/W | Clear DER<br>Writing 1 to this bit clears the DER bit in the CHSTAT_n register. Writing 1 to this bit also clear the USBFDMAmn interrupt.<br>When this bit is read, 0 is read.<br>1: Clears the DER bit.<br>0: Has no effect on operation. |
| 6 | CLRTC | 0 | R/W | Clear TC<br>Writing 1 to this bit clears the TC bit in the CHSTAT_n register.<br>When this bit is read, 0 is read.<br>1: Clears the TC bit.<br>0: Has no effect on operation. |
| 5 | CLREND | 0 | R/W | Clear End<br>Writing 1 to this bit clears the END bit in the CHSTAT_n register. Writing 1 to this bit also clear the USBFDMAmn interrupt.<br>When this bit is read, 0 is read.<br>1: Clears the END bit.<br>0: Has no effect on operation. |
| 4 | CLRRQ | 0 | R/W | Clear Request<br>Writing 1 to this bit clears the RQST bit in the CHSTAT_n register.<br>When this bit is read, 0 is read.<br>1: Clears the RQST bit in the CHSTAT_n register.<br>0: Has no effect on operation. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 3 | SWRST | 0 | R/W | Software Reset<br>Writing 1 to this bit clears individual bits in the CHSTAT_n register (for the bits to be cleared, see the description of each bit). Setting this bit must be done when the EN and TACT bits are 0.<br>When this bit is read, 0 is read.<br>1: Clears individual bits in the CHSTAT_n register.<br>0: Has no effect on operation. |
| 2 | STG | 0 | R/W | Software Trigger<br>Writing 1 to this bit makes software set an internal transfer request. If this bit and the SWRST bits are set at the same time, clearing by the SWRST bit takes priority.<br>When this bit is read, 0 is read.<br>1: Makes software set a transfer request (set the RQST bit in the CHSTAT_n register).<br>0: Has no effect on operation. |
| 1 | CLREN | 0 | R/W | Clear Enable<br>Writing 1 to this bit clears the EN bit in the CHSTAT_n register (for details, see **Section 32B.9.13.8(3) Transfer suspension**).<br>When this bit is read, 0 is read.<br>1: Disables DMA transfers (clears the EN bit in the CHSTAT_n register).<br>0: Has no effect on operation. |
| 0 | SETEN | 0 | R/W | Set Enable<br>Writing 1 to this bit enables DMA transfers in DMA channel n. If this bit and the SWRST bits are set at the same time, clearing by the SWRST bit takes priority, and DMA transfers do not start.<br>When this bit is read, 0 is read.<br>1: Enables DMA transfers (sets the EN bit in the CHSTAT_n register).<br>0: Has no effect on operation. |

**Note:** Temporary masking (using the CLRDMARQM, and SETDMARQM bits) of the DMA transfer requests from the USB control applies to only the resources for channel n. Setting the SETDMARQM bit for channel n does not affect the operation of channel m.

## 32B.5.3 Channel Configuration Register

### 32B.5.3.1 Channel Configuration Register ch0 [CHCFG_0] <Address: H'42C>
### Channel Configuration Register ch1 [CHCFG_1] <Address: H'46C>

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | DMS | REN | RSW | RSEL | SBE | DIM | TCM | DEM | WONLY | — | DAD | SAD | DDS | | | |
| 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 | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | SDS | | | | DRRP | — | — | — | — | — | — | — | REQD | — | — | SEL |
| 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 | R | R | R | R | R | R | R/W | R | R | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 | DMS | 0 | R/W | DMA Mode Select<br>This bit specifies the DMA mode to be used.<br>0: Register mode (default)<br>1: Link mode |
| 30 | REN | 0 | R/W | Register Set Enable<br>This bit specifies whether to successively perform, after a DMA transaction ends, another DMA transaction using the Next Register Set selected by the RSEL bit. This bit is valid only in register mode.<br>0: Does not perform the DMA transaction successively.<br>1: Performs the DMA transaction successively.<br>Setting conditions:<br>• "1" is written to this bit.<br>• "1" is written to the SETREN bit in the CHCTRL_n register.<br>Clearing conditions:<br>• "0" is written to this bit.<br>• The REN bit is 1, and a DMA transaction ends.<br>To re-set the REN bit during a transaction, we recommend you to use the SETREN bit in the CHCTRL_n register. |
| 29 | RSW | 0 | R/W | Register Select Switch<br>This bit specifies whether to automatically invert the RSEL bit after a DMA transaction ends. This bit is valid only in register mode.<br>0: Does not invert the RSEL bit after a DMA transaction ends. (Default)<br>1: Inverts the RSEL bit after a DMA transaction ends. |
| 28 | RSEL | 0 | R/W | Register Set Select<br>This bit is used to select the Next Register Set to be used for the next DMA transaction. This bit is valid only in register mode.<br>When the RSW bit is 1, this bit is inverted automatically at the end of a DMA transaction.<br>0: Uses the Next0 Register Set. (Default)<br>1: Uses the Next1 Register Set.<br>Transition condition:<br>A DMA transaction ends with the RSW bit set. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 27 | SBE | 0 | R/W | Sweep Buffer Enable<br>This bit specifies whether to sweep (write) the data already read and stored in the buffer and stop transfer when the EN bit in the CHSTAT_n register is cleared during a DMA transaction.<br>The sweep mode can be used only when the REQD bit is 0.<br>0: Stops transfer without sweeping out the buffer. (Default)<br>1: Stops transfer after sweeping out the buffer. |
| 26 | DIM | 0 | R/W | Descriptor Interrupt Mask<br>This bit specifies whether to mask the USBFDMAmn interrupt when the LV bit value read from descriptor header is 0.<br>0: Does not mask the USBFDMAmn interrupt. (Default)<br>1: Mask the USBFDMAmn interrupt. |
| 25 | TCM | 0 | R/W | DMATC Mask<br>This bit is used to mask the DMATC signal, which is sent from the DMAC to USB control.<br>When this bit is 1 at the time the DMATC signal is to be output, the DMATC signal is not asserted.<br>Also, the TC bit in the CHSTAT_n register is not asserted in that case. In register mode, this bit is automatically cleared. In link mode, this bit is not cleared automatically.<br>Use this bit when you control DMA transfers by software.<br>0: Does not mask the DMATC signal. (Default)<br>1: Masks the DMATC signal.<br>Clearing condition:<br>A DMA transaction ends with the TCM bit set. |
| 24 | DEM | 0 | R/W | USBFDMAmn Mask<br>When this bit is 1 at the time the USBFDMAmn interrupt is not asserted. Also, the END bit in the CHSTAT_n register is not asserted in that case. In register mode, this bit is not cleared automatically. In link mode, this bit is automatically cleared.<br>0: Does not mask the USBFDMAmn interrupt. (Default)<br>1: Masks the USBFDMAmn interrupt.<br>Clearing condition:<br>A DMA transaction ends with the DEM bit set. |
| 23 | WONLY | 0 | R/W | Write Only Mode<br>This bit is used to switch the transfer operation mode to the write-only mode (see **Section 32B.9.12.2, Write only mode**).<br>0: Normal operation (default)<br>1: Write-only mode |
| 22 | — | 0 | R | Nothing is assigned to this bit. Fix this bit to 0. |
| 21 | DAD | 0 | R/W | Destination Address Direction<br>This bit specifies the direction of counting the transfer-destination address in DMA channel n. If the transfer destination is on the USB control side, write 1 (fixed) to this bit.<br>0: Incrementing (default)<br>1: Fixed<br>When the transfer destination uses the skip mode or is beat-unaligned, do not specify 1 (fixed) in this bit. |
| 20 | SAD | 0 | R/W | Source Address Direction<br>This bit specifies the direction of counting the transfer-source address in DMA channel n. If the transfer source is on the USB control side, write 1 (fixed) to this bit.<br>0: Incrementing (default)<br>1: Fixed<br>When the transfer source uses the skip mode or is beat-unaligned, do not specify 1 (fixed) in this bit. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 19 to 16 | DDS[3:0] | All 0 | R/W | Destination Data Size<br>These bits specify the size of DMA transfer data. When the transfer destination is on the USB control side, select the normal mode.<br>Use bit 3 to switch between the normal and skip modes.<br>0: Normal mode (default)<br>1: Skip mode<br>Use bits 2 to 0 to specify the size of transfer data. (For specifiable values, see **Table 32B.21**.)<br>000: 8 bits (default)<br>001: 16 bits<br>010: 32 bits<br>011: 64 bits<br>100: 128 bits<br>101: 256 bits<br>110: 512 bits<br>111: Setting prohibited |
| 15 to 12 | SDS[3:0] | All 0 | R/W | Source Data Size<br>These bits specify the size of DMA transfer data.<br>Use bit 3 to switch between the normal and skip modes.<br>0: Normal mode (default)<br>1: Skip mode<br>Use bits 2 to 0 to specify the size of transfer data. (For specifiable values, see **Table 32B.21**.)<br>000: 8 bits (default)<br>001: 16 bits<br>010: 32 bits<br>011: 64 bits<br>100: 128 bits<br>101: 256 bits<br>110: 512 bits<br>111: Setting prohibited |
| 11 | DRRP | 0 | R/W | Descriptor Read Repeat<br>This bit switches the operation to be performed when the LV value in the header read from the descriptor is 0. (See **section 32B.9.12.1(2)(a), Operation flow of link mode**.)<br>• 0: This module sets the DER bit in the CHSTAT_n register, and then stops descriptor reading. (Default)<br>• 1: This module keeps reading the same descriptor until the LV value changes to 1, and, when the LV value becomes 1, starts the DMA transfer using the values in the descriptor. The interval of descriptor reading is controlled by using the DSCITVL register. |
| 10 to 7 | — | All 0 | R | Nothing is assigned to these bits. Fix these bits to 0. |
| 6, 5 | — | All 0 | R | Nothing is assigned to these bits. Fix these bits to 1. |
| 4 | — | 0 | R | Nothing is assigned to this bit. Fix this bit to 0. |
| 3 | REQD | 0 | R/W | Request Direction<br>This bit specifies whether the USB control is on the transfer source side or it is on the transfer destination side.<br>0: The USB control is on the transfer source side. (Default)<br>1: The USB control is on the transfer destination side. |
| 2, 1 | — | All 0 | R | Nothing is assigned to these bits. Fix these bits to 0. |
| 0 | SEL | 0 | R/W | Terminal Select<br>This bit selects the FIFO channel to be used on the USB control side.<br>0: D0FIFO<br>1: D1FIFO |

The range of values specifiable in the SDS[2:0] and DDS[2:0] bits depends on the data bus width, number of implemented buffer stages, and whether the address to be accessed is beat-aligned or not (beat-unaligned). The table below shows the range of specifiable values.

Table 32B.21 Range of Sizes That Can Be Specified in SDS and DDS Bits

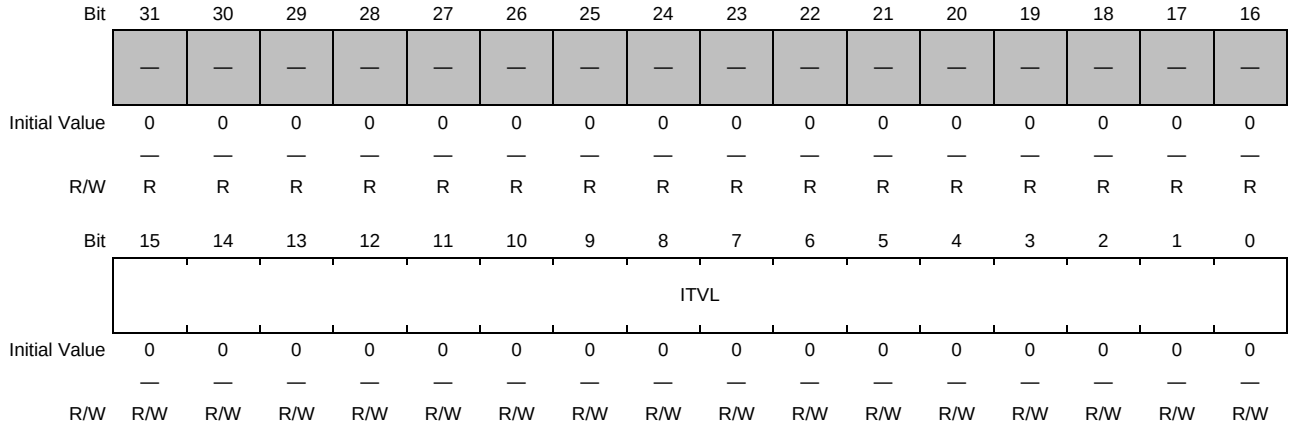
| Transfer Address | REQD | SDS[2:0] | DDS[2:0] |
|---|---|---|---|
| Beat aligned | 0 | This data size should be equal to that of the MBW bits in the DxFIFOSEL register. | 8 to 32 bits (000 to 010)<br>128 to 512 bits (100 to 110) |
| | 1 | 8 to 32 bits (000 to 010)<br>128 to 512 bits (100 to 110) | This data size should be equal to that of the MBW bits in the DxFIFOSEL register. |
| Beat unaligned | 0 | This data size should be equal to that of the MBW bits in the DxFIFOSEL register. | 8 to 32 bits (000 to 010)<br>128 to 256 bits (100 to 101) |
| | 1 | 8 to 32 bits (000 to 010)<br>128 to 256 bits (100 to 101) | This data size should be equal to that of the MBW bits in the DxFIFOSEL register. |

**Note:** When the destination is beat-unaligned with REQD = 0 or the source is beat-unaligned with REQD = 1, specify values in the range of specifiable values for beat-unaligned transfer addresses in both of the SDS[2:0] and DDS[2:0] bits.
Even if the transfer source and destination are beat-aligned when a DMA transaction starts, they might become beat-unaligned in the middle of the transaction when a skip transfer is used. If this might occur, perform settings in the first place on the assumption that the transfer source and destination are beat-unaligned.
If software cannot determine whether the transfer source and/or destination is beat-aligned, use values in the range of specifiable values for beat-unaligned transfer addresses.

## 32B.5.4 Channel Interval Register n

For details, see **Section 32B.9.13.6, Interval Count Function**.

### 32B.5.4.1 Channel Interval Register ch0 [CHITVL_0] <Address: H'430>
### Channel Interval Register ch1 [CHITVL_1] <Address: H'470>




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 16 | — | All 0 | R | Nothing is assigned to these bits. Fix these bits to 0. |
| 15 to 0 | ITVL | All 0 | R/W | Interval<br>These bits specify the DMA transfer interval. |

## 32B.5.5 Channel Extension Register n

### 32B.5.5.1 Channel Extension Register ch0 [CHEXT_0] <Address: H'434>
### Channel Extension Register ch1 [CHEXT_1] <Address: H'474>

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | DPR | | | | — | — | — | — | SPR | | | |
| 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 12 | — | All 0 | R | Nothing is assigned to these bits. Fix these bits to 0. |
| 11 to 8 | DPR | All 0 | R/W | Destination PROT<br>These bits specify the value to be output to the MHPROT[3:0] pin in a DMA write transfer.<br>The initial value of these bits is H'0. |
| 7 to 4 | — | All 0 | R/W | Nothing is assigned to these bits. Fix these bits to 0. |
| 3 to 0 | SPR | All 0 | R/W | Source PROT<br>These bits specify the value to be output to the MHPROT[3:0] pin in a DMA read transfer.<br>The initial value of these bits is H'0. |

## 32B.6 Link Register Set

When software sets a descriptor address in the NXLA_n register and starts the DMAC, hardware loads the value set in the NXLA_n register into the CRLA_n register. Then, the descriptor is read, and the DMAC starts a DMA transaction according to the values read from the descriptor. The value in the NXLA_n register is automatically updated to the Next Link Address value read from the descriptor, and the updated value is used as the descriptor address in the next DMA transaction.

### 32B.6.1 Next Link Address Register n (NXLA_n)

#### 32B.6.1.1 Next Link Address Register ch0 [NXLA_0] <Address: H'438>
#### Next Link Address Register ch1 [NXLA_1] <Address: H'478>

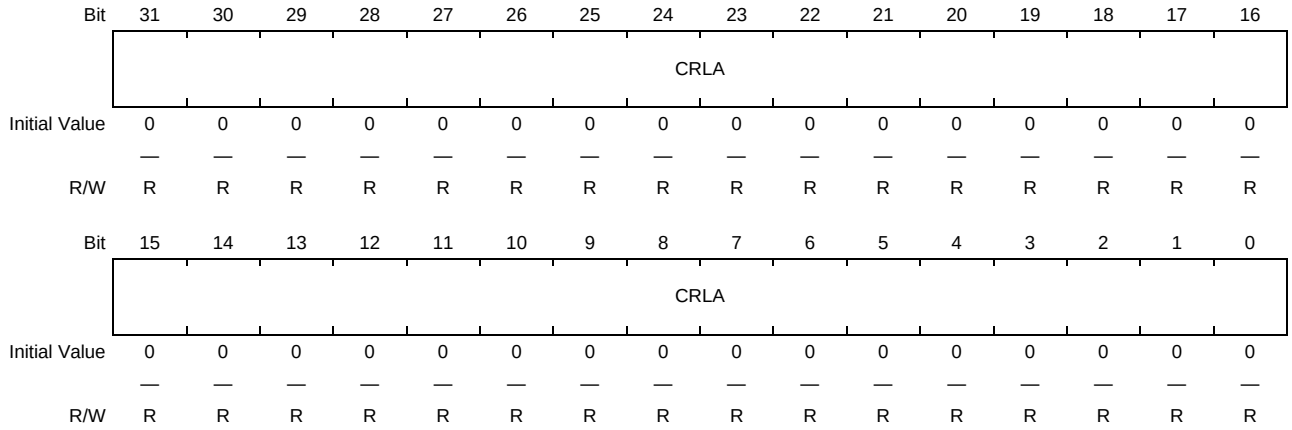
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | NXLA | | | | | | | | | | | | | | | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | R/W | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | NXLA | | | | | | | | | | | | | | | |
| 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 | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 2 | NXLA | All 0 | R/W | Next Link Address |
| 1, 0 | | All 0 | R | These bits specify the link-destination address. Upper two bits are fixed to 0, and only the address aligned with a word boundary can be set. |

## 32B.6.2 Current Link Address Register n (CRLA_n)

### 32B.6.2.1 Current Link Address Register ch0 [CRLA_0] <Address: H'43C>
### Current Link Address Register ch1 [CRLA_1] <Address: H'47C>



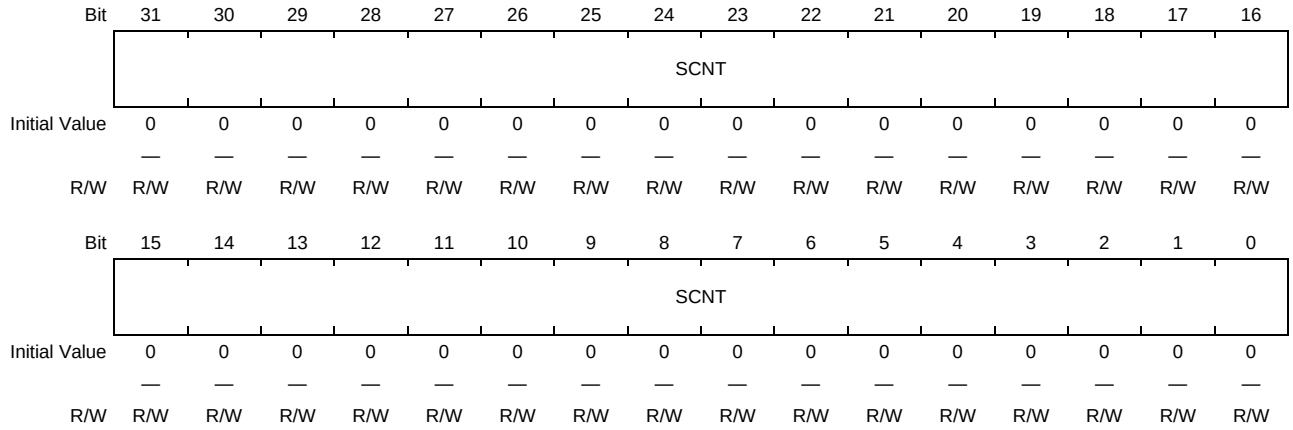

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 0 | CRLA | All 0 | R | Current Link Address<br>These bits indicate the address of the descriptor being used for current transaction. |

## 32B.7 Skip Register Set

This register set is used to specify settings for a skip (scatter/gather) transfer.

### 32B.7.1 Source Continuous Register n

#### 32B.7.1.1 Source Continuous Register ch0 [SCNT_0] <Address: H'600>
#### Source Continuous Register ch1 [SCNT_1] <Address: H'620>

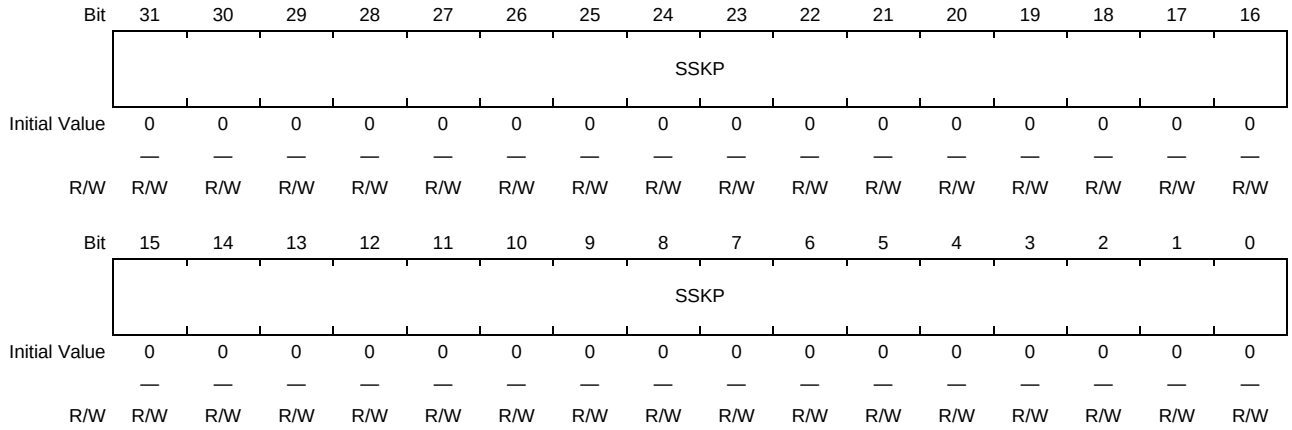



| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 0 | SCNT | All 0 | R/W | Source Continuous<br>These bits specify the size of the space to be accessed continuously by source address access. (Unit: byte) |

**Note:** This register is used in pair with the SSKP_n register (see **Figure 32B.3**). To use this mode, set the SDS[3] bits in the CHCFG_n register to 1. To perform a skip transfer on the transfer source side, the SAD bit in the CHCFG_n register must not be set to 1 (Fixed). Do not perform a skip transfer with the SCNT bits set to 0.

## 32B.7.2 Source Skip Register n

### 32B.7.2.1 Source Skip Register ch0 [SSKP_0] <Address: H'604>
### Source Skip Register ch1 [SSKP_1] <Address: H'624>




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 0 | SSKP | All 0 | R/W | Source Skip<br>These bits specify the size of area to be skipped in a source address access. (Unit: byte) |

**Note:** This register is used in pair with the SCNT_n register (see **Figure 32B.3**). To use this mode, set the SDS[3] bits in the CHCFG_n register to 1. To perform a skip transfer on the transfer source side, the SAD bit in the CHCFG_n register must not set to 1 (Fixed).

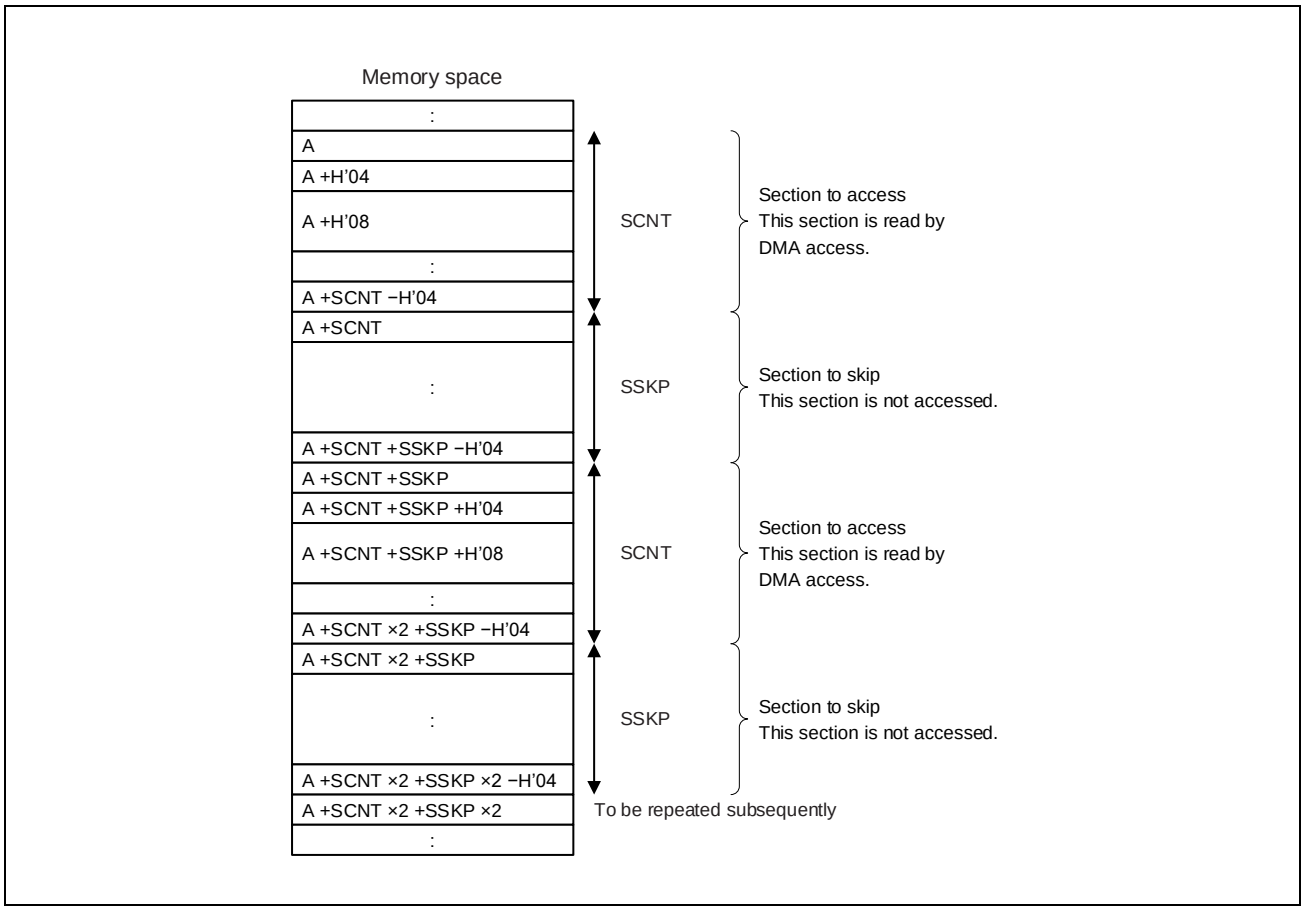
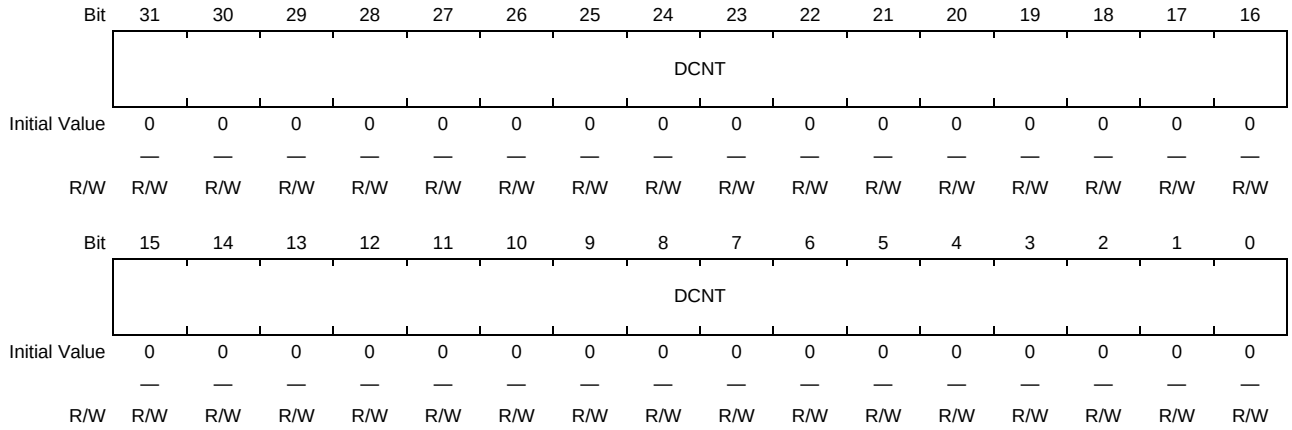



Figure 32B.3 Relationship between SSKP and SCNT

You can specify values of the SCNT and SSKP bits regardless of the source address and the value of the SDS field in the CHCFG_n register. The DMAC performs access based on the size specified in the SDS field, and fetches only valid data into the buffer (see **Section 32B.9.14.1(1), Read access**).

## 32B.7.3 Destination Continuous Register n

### 32B.7.3.1 Destination Continuous Register ch0 [DCNT_0] <Address: H'608>
### Destination Continuous Register ch1 [DCNT_1] <Address: H'628>

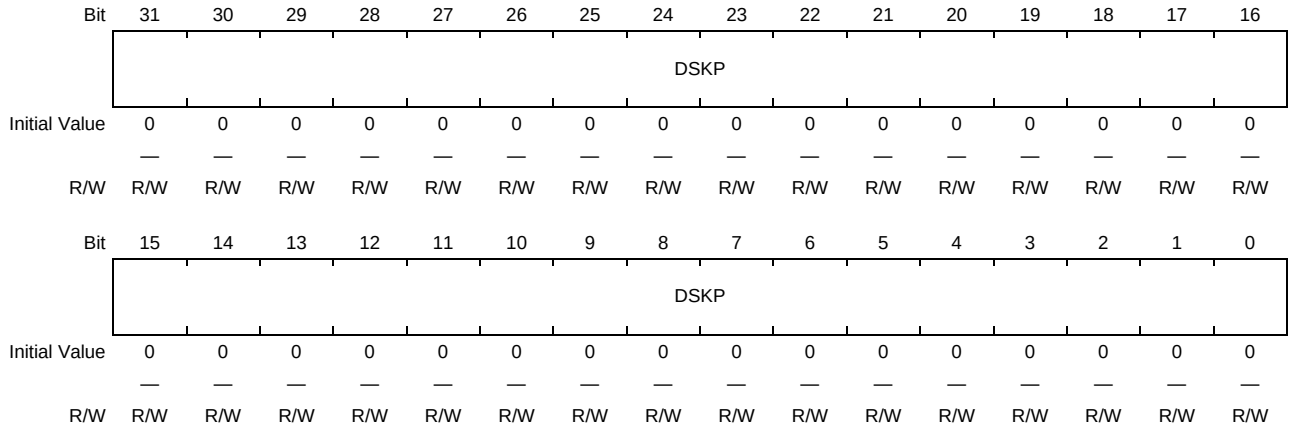



| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 0 | DCNT | All 0 | R/W | Destination Continuous<br>These bits specify the size of the space to be accessed continuously by destination address access.<br>(Unit: byte) |

**Note:** This register is used in pair with the DSKP_n register (see **Figure 32B.4**). To use this mode, set the DDS[3] bits in the CHCFG_n register to 1. To perform a skip transfer on the transfer destination side, the DAD bit in the CHCFG_n register must not be set to 1 (Fixed). Do not perform a skip transfer with the DCNT bits set to 0.

## 32B.7.4 Destination Skip Register n

### 32B.7.4.1 Destination Skip Register ch0 [DSKP_0] <Address: H'60C>
### Destination Skip Register ch1 [DSKP_1] <Address: H'62C>




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 0 | DSKP | All 0 | R/W | Destination Skip<br>These bits specify the size of area to be skipped in a destination address access.<br>(Unit: byte) |

**Note:** This register is used in pair with the DCNT_n register (see **Figure 32B.4**). To use this mode, set the DDS[3] bits in the CHCFG_n register to 1. To perform a skip transfer on the transfer destination side, the DAD bit in the CHCFG_n register must not be set to 1 (Fixed).

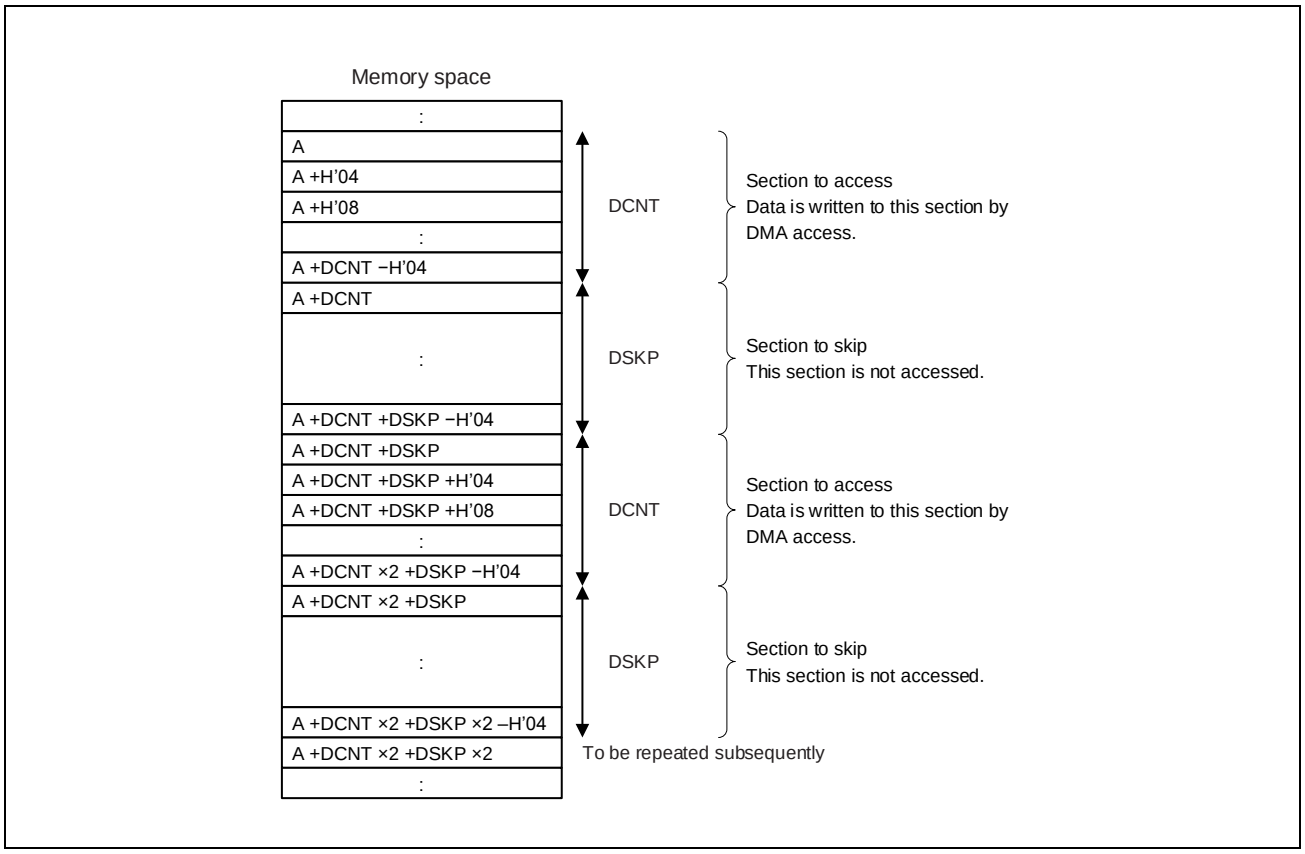



Figure 32B.4 Relationship between DSKP and DCNT

You can specify values of the DCNT and DSKP bits regardless of the destination address and the value of the DDS field in the CHCFG_n register. The DMAC performs write access to only the specified space that has a combined size not more than the size specified in the DDS field (see **Section 32B.9.14.1(1), Read access**).

# 32B.8 DMA Register Set

The registers described below are shared by all channels.

## 32B.8.1 DMA Control Register

### 32B.8.1.1 DMA Control Register [DCTRL] <Address: H'700>

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | LWPR | | | | — | — | — | — | LDPR | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | LVINT | PR |
| 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 |
|---|---|---|---|---|
| 31 to 28 | — | All 0 | R | Nothing is assigned to these bits. Fix these bits to 0. |
| 27 to 24 | LWPR | All 0 | R/W | Link WriteBack PROT<br>These bits specify the value to be output to the MHPROT[3:0] pin at write-back to the descriptor in link mode. |
| 23 to 20 | — | All 0 | R | Nothing is assigned to these bits. Fix these bits to 0. |
| 19 to 16 | LDPR | All 0 | R/W | Link Descriptor PROT<br>These bits specify the value to be output to the MHPROT[3:0] pin at reading of the descriptor in link mode. |
| 15 to 2 | — | All 0 | R | Nothing is assigned to these bits. Fix these bits to 0. |
| 1 | LVINT | 0 | R/W | Level Interrupt<br>To use this module, be sure to set this bit to 1. |
| 0 | PR | 0 | R/W | Priority<br>This bit specifies the transfer priority control mode (see **Section 32B.9.13.2, DMA channel priority control**).<br>0: Fixed priority mode<br>1: Round-robin mode |

## 32B.8.2 Descriptor Interval Register n

### 32B.8.2.1 Descriptor Interval Register [DSCITVL] <Address: H'704>

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | DITVL | | | | | | | | — | — | — | — | — | — | — | — |
| 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 |
|---|---|---|---|---|
| 31 to 16 | — | All 0 | R | Nothing is assigned to these bits. Fix these bits to 0. |
| 15 to 8 | DITVL | All 0 | R/W | Descriptor Interval<br>These bits specify the interval of descriptor read operation. The descriptor will be re-read at intervals of "value of DITVL x 256." |
| 7 to 0 | — | All 0 | R | Nothing is assigned to these bits. Fix these bits to 0. |

## 32B.8.3 DMA Control Register

### 32B.8.3.1 DMA Status EN Register [DSTAT_EN] <Address: H'710>

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | EN1 | EN0 |
| 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 2 | — | All 0 | R | Nothing is assigned to these bits. Fix these bits to 0. |
| 1 | EN1 | 0 | R | This bit indicates the state of the EN bit for DMA channel 1. |
| 0 | EN0 | 0 | R | This bit indicates the state of the EN bit for DMA channel 0. |

## 32B.8.4 DMA Status ER Register

### 32B.8.4.1 DMA Status ER Register [DSTAT_ER] <Address: H'714>

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | ER1 | ER0 |
| 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 2 | — | All 0 | R | Nothing is assigned to these bits. Fix these bits to 0. |
| 1 | ER1 | 0 | R | This bit indicates the state of the ER bit for DMA channel 1. |
| 0 | ER0 | 0 | R | This bit indicates the state of the ER bit for DMA channel 0. |

## 32B.8.5 DMA Status END Register

### 32B.8.5.1 DMA Status END Register [DSTAT_END] <Address: H'718>

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | END1 | END0 |
| 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 2 | — | All 0 | R | Nothing is assigned to these bits. Fix these bits to 0. |
| 1 | END1 | 0 | R | This bit indicates the state of the END bit for DMA channel 1. |
| 0 | END0 | 0 | R | This bit indicates the state of the END bit for DMA channel 0. |

## 32B.8.6 DMA Status TC Register

### 32B.8.6.1 DMA Status TC Register [DSTAT_TC] <Address: H'71C>

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | TC1 | TC0 |
| 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 2 | — | All 0 | R | Nothing is assigned to these bits. Fix these bits to 0. |
| 1 | TC1 | 0 | R | This bit indicates the state of the TC bit for DMA channel 1. |
| 0 | TC0 | 0 | R | This bit indicates the state of the TC bit for DMA channel 0. |

## 32B.8.7 DMA Status SUS Register

### 32B.8.7.1 DMA Status SUS Register [DSTAT_SUS] <Address: H'720>

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | SUS1 | SUS0 |
| 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 2 | — | All 0 | R | Nothing is assigned to these bits. Fix these bits to 0. |
| 1 | SUS1 | 0 | R | This bit indicates the state of the SUS bit for DMA channel 1. |
| 0 | SUS0 | 0 | R | This bit indicates the state of the SUS bit for DMA channel 0. |

# 32B.9 Functions

## 32B.9.1 System Control and Oscillation Control

This chapter describes register manipulations required to perform initial setup of this module. This chapter also describes the registers required to control power consumption.

For the parts of the sequence that are required in both host and peripheral modes, see **Section 32A.9.1, Host/Peripheral Common Setting Sequence**.

### 32B.9.1.1 USB data bus resistor control

This module controls switchover between the pull-up resistors for the D+ signal and the pull-down resistors for the D- signal of the USBPHY. Use the DPRPU bit and DRPD bit in the SYSCFG0 register to set pull-up or pull-down of each signal.

Recognize that a connection to the USB host is established, and then set 1 for the DPRPU bit in the SYSCFG0 register to pull up the D+ signal.

After connected to the host, this module automatically switches the resistor when the state changes to reset handshake, suspend, or resume.

If 0 is set for the DPRPU bit in the SYSCFG0 register during communication with the host, the pull-up resistor (or termination resistor) for the USB data line is disabled. This can notify the host controller of disconnection of a device.

## 32B.9.2 Interrupt Function

### 32B.9.2.1 Overview of Interrupt Function (other than DMA Master)

The following shows a list of interrupt functions of this module. An interrupt is notified as U2P_IXL_INT. Check the status register to identify the interrupt factor. The interrupt from this module is not asserted when the supply of the internal bus clock (P1ϕ) is stopped.

Table 32B.22 List of Interrupt functions

| Bit | Interrupt name | Interrupt factor | Related status |
|---|---|---|---|
| VBINT | VBUS interrupt | The status change of the VBUS input pin is detected.<br>(Changes from L to H and from H to L are detected.) | VBSTS |
| RESM | Resume interrupt | In the suspended state, a change of the USB bus status is detected (from J-State to K-State or from J-State to SE0). | — |
| SOFR | Frame number update interrupt | If SOFRM is 0: An SOF packet with a different frame number is received.<br>If SOFRM is 1: An SOF with µframe number 0 cannot be received due to a problem such as packet corruption. | — |
| DVST | Device state transition interrupt | A transition of a device state is detected.<br>USB bus reset detected<br>Suspended state detected Set Address request received<br>Set Configuration request received | DVSQ |
| CTRT | Control transfer stage transition interrupt | A transition of a control transfer stage is detected.<br>Setup stage completed<br>ControlWrite transfer status stage transition<br>ControlRead transfer status stage transition<br>Control transfer completed<br>Control transfer sequence error | CTSQ |
| BEMP | Buffer empty interrupt | All data in the buffer memory is sent and the buffer becomes empty.<br>A packet exceeding the maximum packet size is received. | PIPEBEMP |
| NRDY | Buffer not ready interrupt | A token is received when the PID setting is BUF and the buffer memory is not available for sending and receiving data.<br>A CRC error or bit stuff error occurs when data is received in isochronous transfer.<br>An interval error occurs when data is received in isochronous transfer. | PIPENRDY |
| BRDY | Buffer ready interrupt | The buffer becomes ready (available for reading or writing data). | PIPEBRDY |

The following shows the relationship between interrupts of this module.

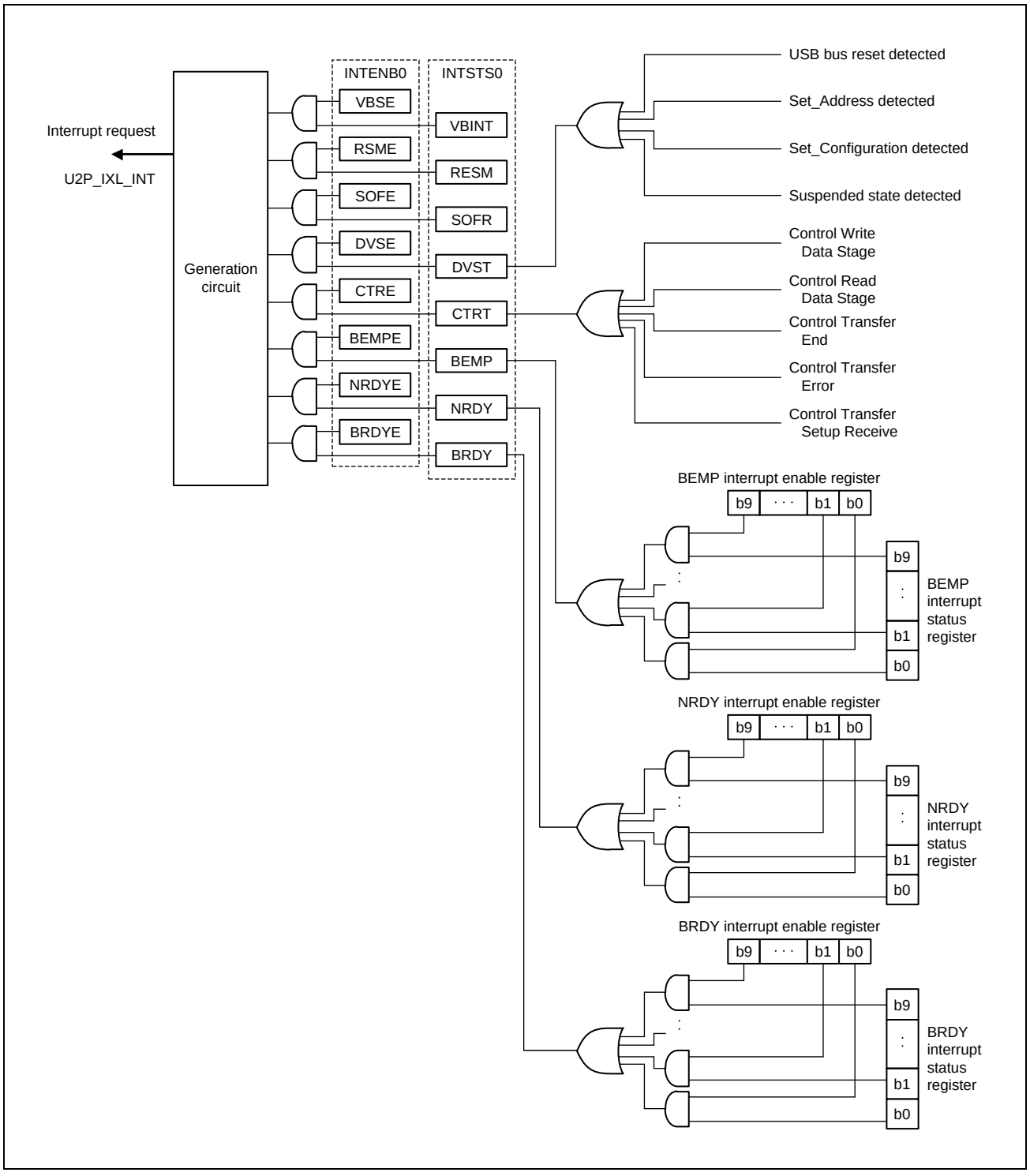



Figure 32B.5 Interrupt Association Diagram

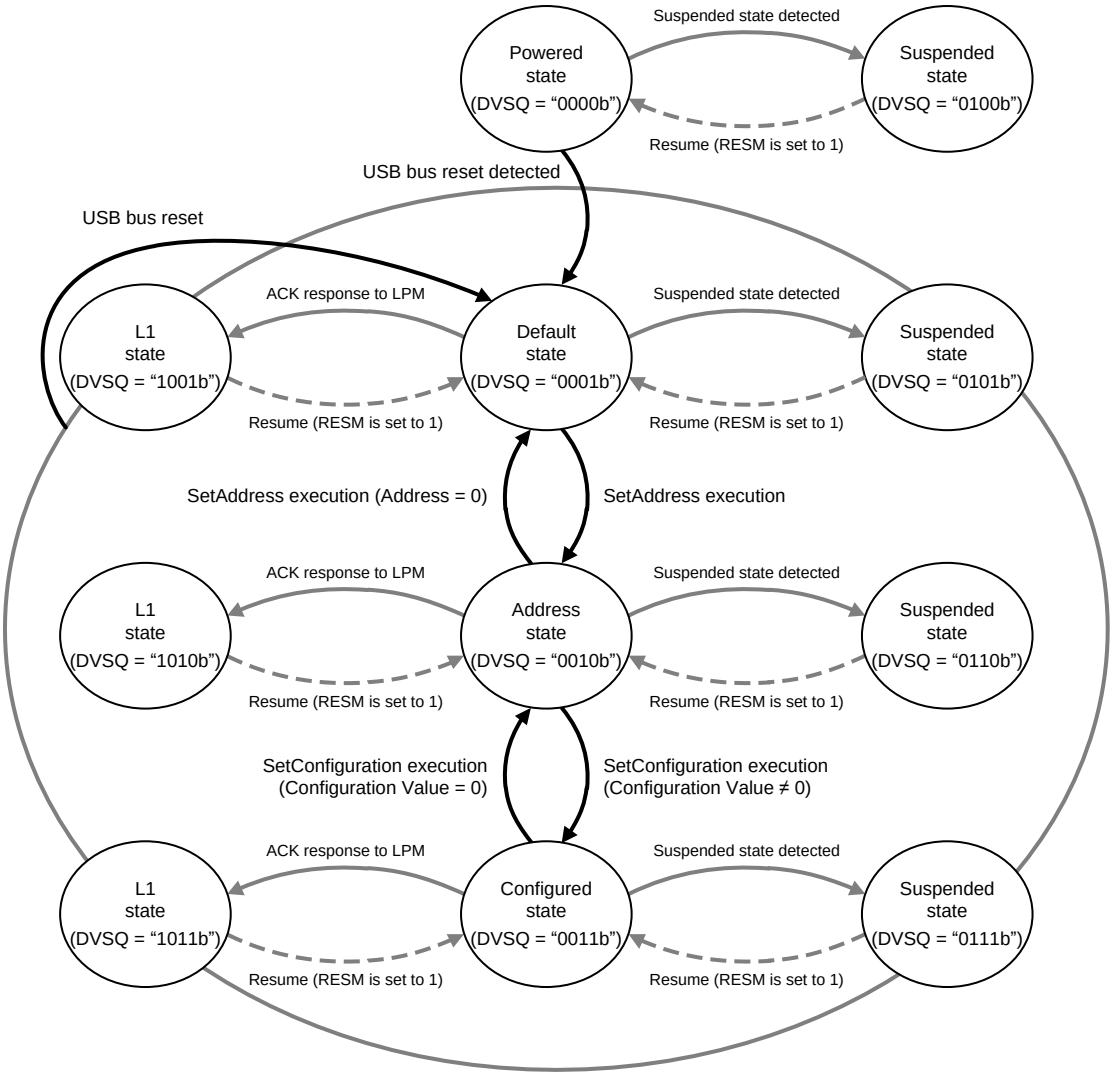
### 32B.9.2.2 Device State Transition Interrupts

**Figure 32B.6** shows the device state transition diagram of this module.

This module manages device states and generates device state transition interrupts. However, resumption from the suspended state (resume signal detection) is detected by a resume interrupt.

Device state transition interrupts can be enabled and prohibited by using the INTENB0 register. A device state for which a transition has occurred can be checked in the DVSQ bits in the INTSTS0 register.

To trigger a transition to the default state, a device state transition interrupt is generated after a reset hand-shake protocol ends.




**Note:** When a transition indicated by a solid line occurs, the DVST bits are set to 1.
When a transition indicated by a broken line occurs, the RESM bit is set to 1.

Figure 32B.6 Device State Transition Diagram

#### 32B.9.2.3 Control Transfer Stage Transition Interrupts

**Figure 32B.7** shows the control transfer stage transition diagram of this module. This module manages a sequence of control transfers, and generates a control transfer stage transition interrupt. Control transfer stage transition interrupts can be enabled and prohibited by using the INTENB0 register. A transfer stage for which a transition has occurred can be checked in the CTSQ bits in the INTSTS0 register.

The following describes sequence errors that can occur during control transfer. If an error occurs, the PID bits in the DCPCTR register are set to 1X (STALL).

**(1) For Control Read transfer**

(a) An OUT or PING token is received in a situation where data has not been transferred yet for an IN token of the data stage.

(b) An IN token is received in the status stage.

(c) A data packet "DATAPID = DATA0" is received in the status stage.

**(2) For Control Write transfer**

(a) An IN token is received in a situation where an ACK response has not been sent yet for an OUT token of the data stage.

(b) The first data packet ”DATAPID = DATA0” is received in the data stage.

(c) An OUT or PING token is received in the status stage.

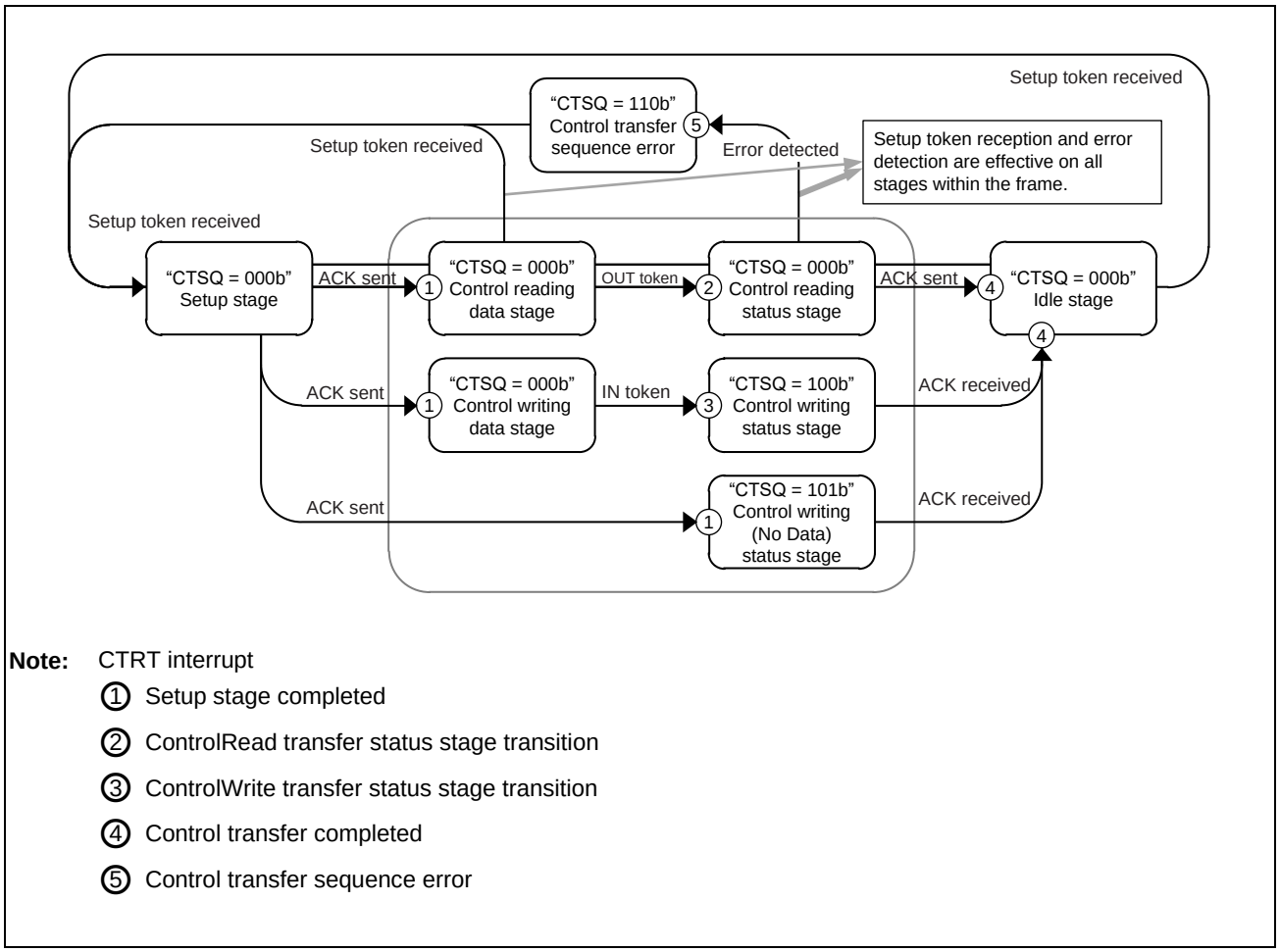
**(3) For Nodata Control transfer**

(a) An OUT or PING token is received in the status stage.

If the amount of received data exceeds the value of the wLength field in a USB request in the Control Write transfer data stage, this module cannot identify this situation as a control transfer sequence error. If a packet other than a zero-length packet is received in the Control Read transfer status stage, this module sends an ACK response and then terminates processing normally.

If a CTRT interrupt is generated (SERR bit is set to 1) due to a sequence error, the CTSQ bits retain 110 until the system writes 0 to the CTRT bits to clear the interrupt status.

Therefore, as long as the CTSQ bits retain 110, even if a new USB request is received, a CTRT interrupt that reports completion of a setup stage is not generated. (Information on completion of the setup stage is retained by this module, and a CTRT interrupt is generated after the interrupt status is cleared.)




**Note:** CTRT interrupt

① Setup stage completed

② ControlRead transfer status stage transition

③ ControlWrite transfer status stage transition

④ Control transfer completed

⑤ Control transfer sequence error

Figure 32B.7 Control Transfer Stage Transition Diagram

#### 32B.9.2.4 Interrupts Relating to DMA Master

Table 32B.23 List of Interrupts

| Interrupt Name | Interrupt Type |
|---|---|
| USBFDMAmn (m, n = 0, 1) | A DMA transaction ends. |
| | An invalid descriptor is read in link mode. |
| USBFDMAERRm (m = 0, 1) | An error response is returned for a transfer issued by the master interface. |

This module has two types of DMA interrupts USBFDMAmn and USBFDMAERRm.

(1) USBFDMAmn interrupt (m, n = 0, 1)
This interrupt occurs when a DMA transaction ends or when an invalid descriptor is read in link mode.
An interrupt is divided for each DMA channel. USBFDMAm0 corresponds to the interrupt for DMA 0ch, and USBFDMAm1 corresponds to the interrupt for DMA 1ch.

(2) USBFDMAERRm interrupt
This interrupt occurs when an error response is returned for a transfer issued by the master interface. This interrupt applies to all DMA channels.

USBFDMAmn interrupt output can be temporarily masked by setting a register.

Interrupt detection can also be masked by setting the register. If interrupt detection is masked, the status register that indicates generation of an interrupt does not change.

On the other hand, a USBFDMAERRm interrupt signal does not have the masking function.

### 32B.9.3 Pipe Control

**Table 32B.24** lists the pipe settings for this module. For USB data transfer, logical pipes called endpoints are used to enable data communication. This module provides 16 pipes for data transfer. Set each pipe according to the specification of the system.

Table 32B.24 PIPE Settings

| Register Name | Bit Name | Setting | Comment |
|---|---|---|---|
| DCPCFG | TYPE | Specifies the transfer type. | Pipes 1 to 9: Settable |
| PIPECFG | BFRE | Selects BRDY interrupt mode. | Pipes 1 to 5: Settable |
| | DBLB | Selects double-buffer configuration. | Pipes 1 to 5: Settable |
| | CNTMD | Selects continuous transfer or non-continuous transfer. | DCP: Settable<br>Pipes 1, 2: Settable only in bulk transfers<br>Pipes 3 to 5: Settable |
| | DIR | Selects transfer direction. | Settable to IN or OUT |
| | EPNUM | Endpoint number | Pipes 1 to 9: Settable<br>Set to a value other than “0000” when a pipe is in use. |
| | SHTNAK | Disables pipes when transfer is completed. | DCP: Settable<br>Pipes 1, 2: Settable only in bulk transfers<br>Pipes 3 to 5 Settable |
| PIPEBUF | BUFSIZE | Buffer memory size | DCP: Unsettable (fixed to 64/256 (CNTMD = 1) bytes)<br>Pipes 1 to 5: Settable (up to 2 Kbytes specifiable)<br>Pipes 6 to 9: Unsettable (fixed to 64 bytes) |
| | BUFNMB | Buffer memory number | DCP: Unsettable (fixed to area 0 to 3 hex)<br>Pipes 1 to 5 Settable (area 8 to 87 hex specifiable)<br>Pipes 6 to 9: Unsettable (fixed to area 4 to 7 hex) |
| DCPMAXP<br>PIPEMAXP | MXPS | Maximum packet size | Setting conforming to the USB specification. |
| PIPEPERI | IFIS | Buffer flush | Pipes 1, 2: Settable only in isochronous transfers<br>Pipes 3 to 5: Unsettable<br>Pipes 6 to 9: Unsettable |
| | IITV | Interval counter | Pipes 1, 2: Settable only in isochronous transfers<br>Pipes 3 to 5: Unsettable<br>Pipes 6 to 9: Unsettable |
| DCPCTR<br>PIPExCTR | BSTS | Buffer status | DCP state switched between receive and transmit buffer by ISEL bit |
| | INBUFM | IN buffer monitor | Available only for pipes 3 to 5 |
| | ACLRM | Auto buffer clear | Pipes 1 to 9: Settable |
| | SQCLR | Sequence clear | Clears data toggle bit. |
| | SQSET | Sequence set | Sets data toggle bit. |
| | SQMON | Sequence monitor | Monitors data toggle bit. |
| | PBUSY | Pipe busy monitor | |
| | PID | Response PID | |
| DCPCTR<br>PIPExCTR | ATREPM | Auto response mode | Pipes 1 to 5: Settable |
| PIPExTRE | TRENB | Transaction count enable | Pipes 1 to 5: Settable |
| | TRCLR | Current transaction counter clear | Pipes 1 to 5: Settable |
| PIPExTRN | TRNCNT | Transaction counter | Pipes 1 to 5: Settable |

#### 32B.9.3.1 Maximum packet size setting

Use the MXPS bits in the DCPMAXP and PIPEMAXP registers to specify the maximum packet size for each pipe. The default control pipe (DCP) and pipes 1 to 5 can be set to any of the maximum packet sizes defined by the USB specification. For pipes 6 to 9, 64 bytes are the upper limit of the maximum packet size. Set the maximum packet size before starting transfer (by setting “PID = BUF”).

DCP: Set 64 for high-speed operation.

DCP: Set 8, 16, 32, or 64 for full-speed operation. Pipes 1 to 5: Set 512 for high-speed bulk transfer.

Pipes 1 to 5: Set 8, 16, 32, or 64 for full-speed bulk transfer.

Pipes 1, 2: Set a value from 1 to 1024 for high-speed isochronous transfer.

Pipes 1, 2: Set a value from 1 to 1023 for full-speed isochronous transfer.

Pipes 6 to 9: Set 64.

High-bandwidth transfers used for interrupt transfers and isochronous transfers are not supported.

#### 32B.9.3.2 Response PID

Set the response PID for each pipe with the PID bits in the DCPCTR and PIPExCTR registers.

(1) Response PID setting
The response PID specifies the response to a transaction from the host.

a) NAK: Always sends a NAK response to a generated transaction.

b) BUF: Responds to a transaction in accordance with the buffer memory state.

c) STALL: Always sends a STALL response to a generated transaction.

Regardless of the value set in the PID bits, an ACK is always sent as a response to a setup transaction and a USB request is stored in corresponding registers.

This module might write data to the PID bits depending on the transaction result. This module writes data to the PID bits in the following cases:

a) NAK:

(A) The SETUP token is received normally (for the DCP only).

(B) If 1 is set for the SHTNAK bit in the PIPECFG register during bulk transfer, a short packet is received.

(C) If 1 is set for the SHTNAK bit during bulk transfer, the transaction counter finishes.

b) BUF:

This module does not write "BUF".

c) STALL:

(A) When a maximum packet size over error is detected in a received data packet

(B) When a control transfer sequence error is detected

#### 32B.9.3.3 Pipe control register switching procedure

The following bits in the pipe control registers can be modified only when USB transmission is disabled (PID = NAK).

**Figure 32B.8** shows the procedure for changing the pipe control register state from the USB transmission enabled (PID = BUF) state.

The registered that cannot be manipulated in the USB transmission enabled (PID = BUF) state are as follows:

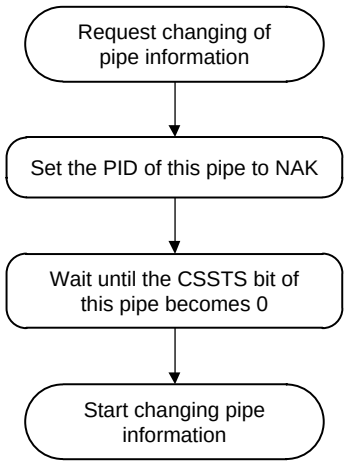
(1) All bits in the DCPCFG and DCPMAXP registers

(2) SQCLR and SQSET bits in the DCPCTR register

(3) All bits in the PIPECFG, PIPEBUF, PIPEMAXP, and PIPEPERI registers

(4) ATREPM, ACLRM, SQCLR, and SQSET bits in the PIPExCTR register

(5) All bits in the PIPExTRE and PIPExTRN registers




**Note:** If the pipe is detached during a USB transaction, the PBUSY bit might remain 1.

Figure 32B.8 Procedure for Changing Pipe Information from USB Transmission Enabled (PID = BUF) State

In addition, for the settings of the following bits in pipe control registers, only the pipe information that is not set for the CURPIPE bit in the CPU, DMA0, or DMA1 FIFO port can be changed.

Registers that cannot be set with pipe information that is set for the CURPIPE bit in a FIFO port:

(6) All bits in the DCPCFG and DCPMAXP registers

(7) All bits in the PIPECFG, PIPEBUF, PIPEMAXP, and PIPEPERI registers

(8) ACLRM bit in the PIPExCTR register

If you change the pipe information, make sure that the setting of CURPIPE is different from the new pipe number. For the default control pipe (DCP), after modifying the pipe information, use the BCLR bit to clear the buffer.

#### 32B.9.3.4 Data PID sequence bit

When a normal data transfer is performed during bulk transfer or interrupt transfer, or in the data stage of control transfer, this module automatically toggles the sequence bit of a data PID. The sequence bit of the next data PID to be transferred can be confirmed with the SQMON bit in the DCPCTR or PIPExCTR registers. For data transmission, the sequence bit is switched when an ACK handshake is received. For data reception, the sequence bit is switched when an ACK handshake is sent. The data PID sequence bit can be changed by using the SQCLR and SQSET bits in the DCPCTR and PIPExCTR registers.

During control transfer, this module automatically sets the sequence bit when the stage changes. When the setup stage finishes, the data PID is set to DATA1. In the status stage, this module responds with “PID = DATA1” without referencing the sequence bit.

Note that, the data PID sequence bit must set, when a clear feature request is sent or received.

Also note that for isochronous transfer setting pipes, you cannot use the SQSET bit to manipulate the sequence bit.

### 32B.9.4 FIFO Buffer

This section describes the processing related to the FIFO buffer of this module.

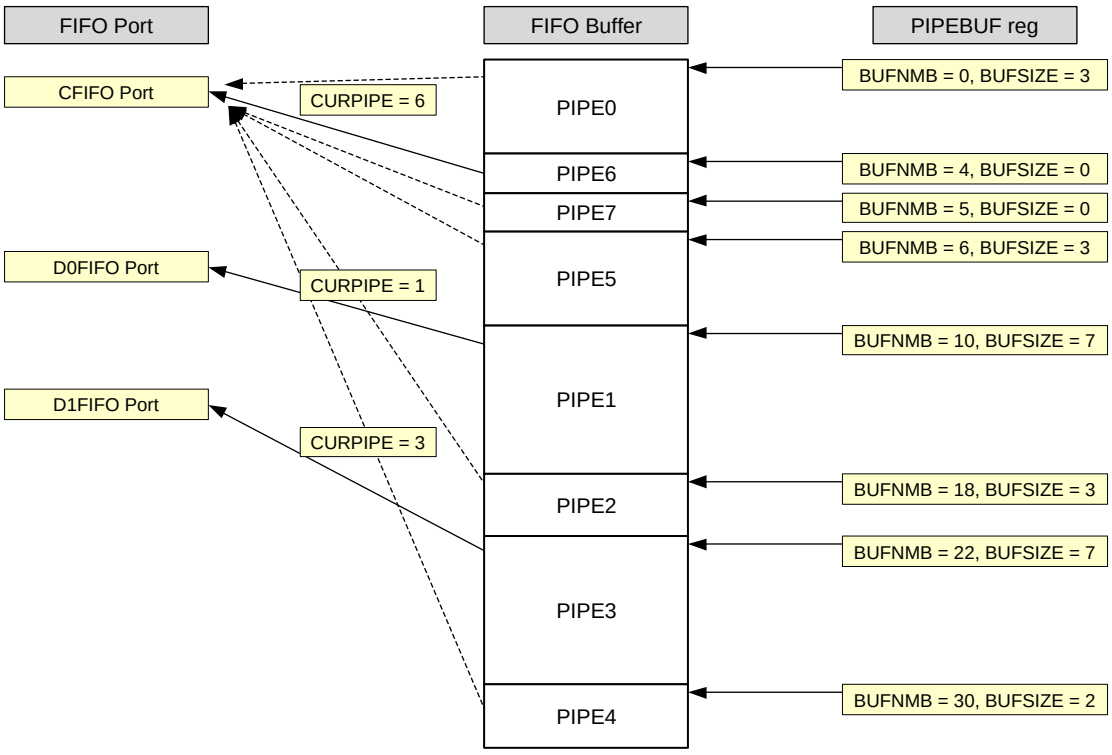
#### 32B.9.4.1 FIFO buffer allocation

**Figure 32B.9** shows an example of FIFO buffer memory mapping of this module. The FIFO buffer area is shared by the CPU and this module. The FIFO buffers can be accessed by either the system (CPU) or this module (SIE).

An independent FIFO buffer area is allocated for each pipe. The memory area is made up of memory blocks (1 block = 64 bytes) and is specified by the starting block number and the number of blocks (specified by the BUFNMB and BUFSIZE bits in the PIPEBUF register). When the CNTMD bit in the PIPExCFG register is used to set "continuous transfer mode", the value set with the BUFSIZE bit must be an integral multiple of the maximum packet size. If the double-buffer configuration is selected by the DBLB bit in the PIPExCFG register, two memory areas the size of which is specified by the BUFSIZE bit in the PIPEBUF register are allocated to a single pipe.

FIFO ports are used to access the FIFO buffer (data read/write). The pipe number of a pipe to be assigned to a FIFO port is specified by the CURPIPE bits in the CFIFOSEL/DxFIFOSEL register.

The FIFO buffer status of each pipe can be confirmed by using the BSTS and INBUFM bits in the DCPCTR and PIPExCTR registers. The access right of a FIFO port can be confirmed by using the FRDY bit in the CFIFOCTR/ DxFIFOCTR register.




**Note:** If pipes 8 and 9 are not in use: Settings such as BUFSIZE are unspecified.

Figure 32B.9 Example of FIFO Buffer Memory Mapping

### 32B.9.4.2 Clearing FIFO buffers

**Table 32B.25** lists the modes in which this module can clear the FIFO buffer. Clearing of the FIFO buffer is controlled by the following bits.

Table 32B.25 List of FIFO Buffer Clearing Modes

| Bit Name | BCLR | DCLRM | ACLRM |
|---|---|---|---|
| Register | CFIFOCTR register<br>DxFIFOCTR register | DxFIFOSEL register | PIPExCTR register |
| Function | The FIFO buffer on the CPU side is cleared. | The FIFO buffer is automatically cleared after the data is read from the specified pipe. | The buffer is automatically cleared to discard all the received packets. |
| Clearing method | Write “1” to clear. | 1: Mode enabled<br>0: Mode disabled | 1: Mode enabled<br>0: Mode disabled |

### 32B.9.5 FIFO Port Functions

This section describes FIFO port functions. **Table 32B.26** shows the FIFO port function settings for this module.

If data write access is enabled and data is written up to buffer full state (in non-continuous transfer: maximum packet size), the port automatically goes to the USB bus transmittable state. To enable transmission of data smaller than the buffer full (in non-continuous transfer: less than the maximum packet size), the BVAL bit in the CFIFOCTR/ DxFIFOCTR register must be used to set write end.

To send a zero-length packet, the BCLR bit of that register must be used to clear the buffer before the BVAL bit is used to write end.

When all the data is read in a read access, the port automatically enters the state in which new packets can be received. However, when a zero-length packet is received (DTLN = 0), no data can be read, and therefore the buffer must be cleared by using the BCLR bit.

The receive data length is confirmed with the DTLN bits in the CFIFOCTR/DxFIFOCTR register.

Table 32B.26 FIFO Port Function Settings

| Register Name | Bit Name | Function | Note |
|---|---|---|---|
| C/DxFIFOSEL | RCNT | Selects DTLN read mode | |
| | REW | Rewinds buffer memory (re-read, re-write) | |
| | DCLRM | Reads received data of the specified pipe, and then automatically clears the received data | DxFIFO only |
| | DREQE | Asserts DREQ signal | DxFIFO only |
| | MBW | Specifies FIFO port access bit width | |
| | BIGEND | Selects FIFO port endian | CFIFO only |
| | ISEL | Specifies FIFO port access direction | DCP only |
| | CURPIPE | Selects current pipe | |
| C/DxFIFOCTR | BVAL | Finishes buffer memory write | |
| | BCLR | Clears CPU-side buffer memory | |
| | FRDY | Monitors FIFO port ready | |
| | DTLN | Confirms received data length | |

#### 32B.9.5.1 FIFO port selection

**Table 32B.27** lists pipes that can be selected for each FIFO port.

Use the CURPIPE bits in the C/DxFIFOSEL register to select the pipe to be accessed. After selecting the pipe, confirm that the value written to the CURPIPE bits can be read correctly (if the previous pipe number is read out, this module is currently changing the pipe), confirm that FRDY = 1, and then access the FIFO port.

**Figure 32B.10** shows the procedure for switching the pipe when accessing a FIFO port.

In addition, use the MBW bit to select the bus width with which to access the FIFO port. If the target pipe is the default control pile (DCP), the ISEL bit determines the buffer memory access direction. If the target pipe is not the DCP, the DIR bit in the PIPExCFG register determines the buffer memory access direction.

Table 32B.27 FIFO Port Access for Each Pipe

| Pipe | Access Method | Usable Port |
|---|---|---|
| DCP | CPU access | CFIFO port register |
| Pipes 1 to 9 | CPU access | CFIFO port register |
| | DMA access | DxFIFO port register |

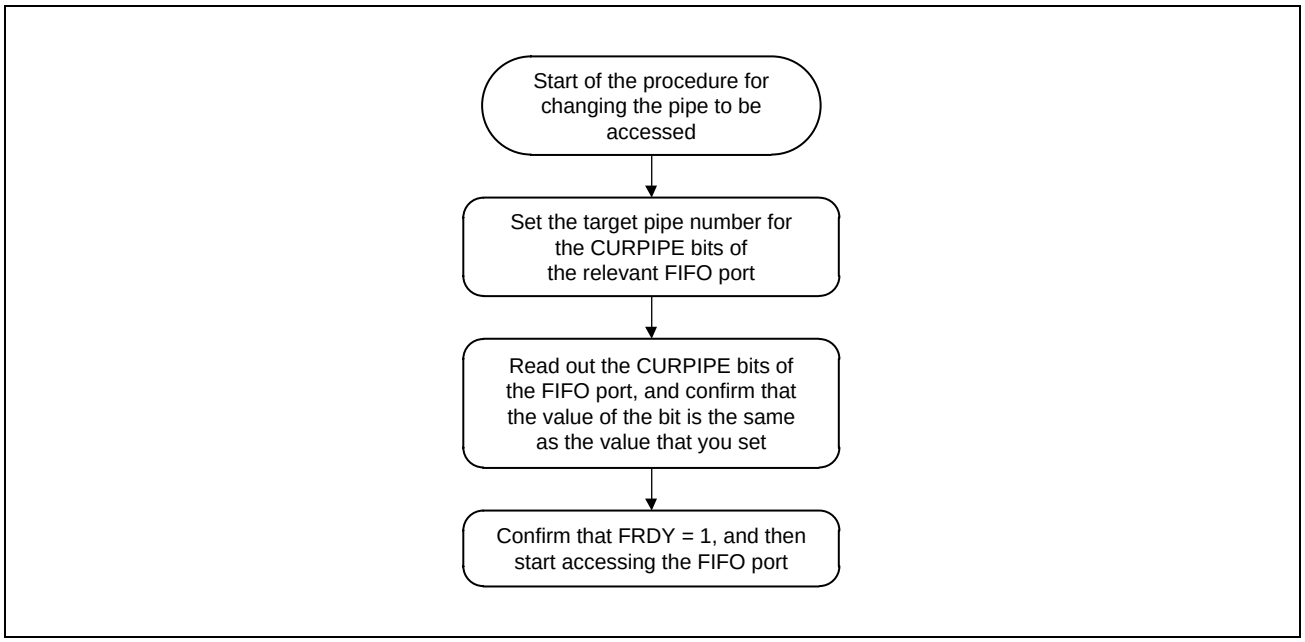



Figure 32B.10 Pipe Switching Procedure for FIFO Port Access

### 32B.9.5.2 DxFIFO automatic clear mode (DxFIFO port read direction)

This module automatically clears the buffer memory for a pipe when data is read out from the buffer memory if the DCLRM bit in the DxFIFOSEL register is set to 1.

**Table 32B.28** shows the correspondence between packet reception and buffer memory clear processing by the software for each setting.

As shown in **Table 32B.28**, the buffer clearing conditions vary with the value that is set for the BFRE bit. However, using the DCLRM bit eliminates the need for buffer clear by the software even in states where clearing is required, which enables DMA transfers without using the software.

Note that for this function, only the buffer memory read direction can be set.

Table 32B.28 Relationship between Packet Reception and Buffer Memory Clear Processing by the Software

| | Register Setting | | | |
|---|---|---|---|---|
| | DCLRM = 0 | | DCLRM = 1 | |
| Buffer State during Packet Reception | BFRE = 0 | BFRE = 1 | BFRE = 0 | BFRE = 1 |
| Buffer full | Clearing unnecessary | | Clearing unnecessary | |
| Zero-length packet received | Clearing necessary | | | |
| Normal short packet received | Clearing unnecessary | Clearing necessary | | |
| Transaction count end | | | | |

### 32B.9.5.3 BRDY interrupt timing selection function

The BFRE bit in the PIPECFG register can be set so that a BRDY interrupt is not generated when a data packet of maximum packet size is received.

When using a DMA transfer, this function enables an interrupt to be generated only when the last data is received. The “last data” here indicates either a short packet reception or the transaction count end. If the BFRE bit is set to 1, a BRDY interrupt is generated after the received data is read. By reading the DTLN bits in the DxFIFOCTR register, the receive data length of last data packet received just before the BRDY interrupt is generated can be confirmed.

**Table 32B.29** shows when this module generates a BRDY interrupt.

Table 32B.29 BRDY Interrupt Generation Timing

| | Register Setting | |
|---|---|---|
| Buffer State during Packet Reception | BFRE = 0 | BFRE = 1 |
| Buffer full (normal packet received) | When packet is received | No interrupt generated |
| Zero-length packet received | When packet is received | When packet is received |
| Normal short packet received | When packet is received | When received data has been read from the buffer memory |
| Transaction count end | When packet is received | When received data has been read from the buffer memory |

The BFRE bit function is valid only in the reading direction from the buffer memory. For the writing direction, fix the BFRE bit to 0.

### 32B.9.6 Control Transfer (DCP)

In the data stage of control transfer, the default control pipe (DCP) is used to transfer data. For the DCP, a single 64-byte buffer is allocated as a fixed area that is used for both control reading and writing (in continuous transfer mode (CNTMD = 1), the size of this area is fixed to 256 bytes). The buffer memory can be accessed through the CFIFO port only.

#### 32B.9.6.1 Control transfer

##### (1) Setup stage

This module always responds with ACK to any normal setup packet for this module. The following shows the behavior of this module in the setup stage:

1. When this module receives a new setup packet, this module sets the following bits:
   (a) VALID bit in the INTSTS0 register: 1
   (b) PID bit in the DCPCTR register: NAK
   (c) CCPL bit in the DCPCTR register: 0

2. When this module receives a data packet after receiving a setup packet, this module stores USB request parameters in the USBREQ, USBVAL, USBINDX, and USBLENG registers.

Response processing for control transfer must be performed after VALID is set to 0. While VALID is 1, PID cannot be set to BUF, and therefore the data stage cannot end.

By using a function of the VALID bit, when this module receives a new USB request during control transfer, this module can respond to the newest request, canceling the request that is being processed.

This module also automatically recognizes the type of transfer (ControlRead, ControlWrite, or NodataControl) from the direction bit (bit 8 of bmRequestType) and request data length (wLength) of the received USB request to manage stage transition. For an incorrect sequence, a control transfer stage transition interrupt occurs to report a sequence error to software. For details about stage management by this module, see **Figure 32B.7**.

##### (2) Data stage

Use the DCP to transfer data in response to the received USB request. Before the DCP buffer memory is accessed, use the ISEL bit in the CFIFOSEL register to specify the access direction.

A transaction is executed by setting the PID bit in the DCPCTR register to BUF.

The end of data transfer is detected with a BRDY or BEMP interrupt. For ControlWrite transfer, use a BRDY interrupt. For ControlRead transfer, use a BEMP interrupt.

For ControlWrite transfer in high-speed operation mode, an NYET handshake response is performed in accordance with the buffer memory state.

### (3) Status stage

If the PID bit in the DCPCTR register is BUF, setting the CCPL bit to 1 terminates control transfer.

After control transfer is terminated by the above setting, this module automatically executes the status stage according to the data transfer direction determined in the setup stage, as shown below:

#### (a) For ControlRead transfer

Upon receiving a zero-length packet from the USB Host Controller, this module sends an ACK response.

#### (b) For ControlWrite or NodataControl transfer

This module sends a zero-length packet, and then receives an ACK response from the USB Host Controller.

### (4) Control transfer automatic response function

This module automatically responds to any normal SET_ADDRESS request. However, if a SET_ADDRESS request has any of the following errors, software must respond, instead of this module:

(a) bmRequestType ≠ “H’00”

(b) wIndex ≠ “H’00”

(c) wLength ≠ “H’00”

(d) wValue > “H’7F”

(e) DVSQ = “011 (Configured) ”

Software must respond to all requests other than SET_ADDRESS.

### 32B.9.7 Bulk Transfer (Pipes 1 to 5)

The user can select the buffer memory usage method (single/double buffer, continuous/non-continuous transfer mode) for bulk transfers. The buffer memory size can be set up to 2 KB. The controller manages the buffer memory state and automatically responds to PING packets and NYET handshakes.

#### 32B.9.7.1 NYET handshake control

**Table 32B.30** lists responses to tokens received in a bulk or control transfer.

When an OUT token is received in a bulk or control transfer and there is only an open space for one packet in the buffer memory, this module sends a NYET response. However, when a short packet is received, this module sends an ACK response instead of a NYET response even under these conditions.

Table 32B.30 List of Responses to Received Tokens

| PID bit value | Buffer memory state*1 | Received token | Response | Note |
|---|---|---|---|---|
| NAK/STALL | — | SETUP | ACK | — |
| | — | IN/OUT/PING | NAK/STALL | — |
| BUF | — | SETUP | ACK | — |
| | RCV-BRDY | OUT/PING | ACK | Receives data packet when OUT token is received*1 |
| | RCV-BRDY | OUT | NYET | Receives data packet*2 |
| | RCV-BRDY | OUT (Short) | ACK | Receives data packet*2 |
| | RCV-BRDY | PING | ACK | *2 |
| | RCV-NRDY | OUT/PING | NAK | |
| | TRN-BRDY | IN | DATA0/1 | Sends data packet |
| | TRN-NRDY | IN | NAK | |

**Note:** Details are described below.
RCV-BRDY*1: Buffer memory has a space for 2 packets or more when an OUT or PING token is received.
RCV-BRDY*2: Buffer memory has only a space for one packet when an OUT token is received.
RCV-NRDY: Buffer memory has no space for any packet when a PING token is received.
TRN-BRDY: Buffer memory has sent data when an IN token is received.
TRN-NRDY: Buffer memory has no send data when an IN token is received.

### 32B.9.8 Interrupt Transfer (Pipes 6 to 9)

This module performs an interrupt transfer in accordance with the period managed by the host controller. This module ignores (no response) PING packets in interrupt transfers. In addition, this module does not send a NYET handshake, but sends an ACK, NAK or STALL response.

Note that this module does not support high-bandwidth interrupt transfers.

### 32B.9.9 Isochronous Transfer (Pipes 1 and 2)

This module is provided with the following functions for isochronous transfers:

1. Notification of error information about isochronous transfers
2. Interval counter (IITV bit)
3. Data setup control for isochronous IN transfers (IDLY function)
4. Buffer flush function for isochronous IN transfers (IFIS bit)
5. SOF pulse output function

This module does not support high-bandwidth Isochronous transfers.

#### 32B.9.9.1 Isochronous transfer error detection

This module has the following error information detection functions for the software to manage errors that occur during isochronous transfer.

**Table 32B.31** and **Table 32B.32** describe the error checking procedure and interrupts that are generated.

1. PID error
   The PID of the received packet is invalid.
2. CRC error and bit stuffing error
   The received packet has a CRC error or invalid bit stuffing.
3. Max packet size over
   The data size of the received packet is larger than the preset maximum packet size.
4. Overrun error and underrun error
   (a) The buffer memory has no data when an IN token is received during IN-direction (send) transfer.
   (b) The buffer memory has no space although an OUT token is received during OUT-direction (receive) transfer.
5. Interval error
   An interval error occurs in the following cases:
   (a) An IN token cannot be received within the interval frame during isochronous IN transfer.
   (b) An OUT token cannot be received within the interval frame during isochronous OUT transfer.

Table 32B.31 Error Detection during Transmission/Reception of Token

| Error Detection Priority | Error Type | Interrupt Generated at Error Detection and Status |
|---|---|---|
| 1 | PID error | No interrupt is generated (ignored as a corrupted packet). |
| 2 | CRC error, bit stuffing error | No interrupt is generated (ignored as a corrupted packet). |
| 3 | Overrun error, underrun error | An NRDY interrupt is generated and the OVRN bit is set.<br>A zero-length packet is sent in response to an IN token. A data packet is not received in response to an OUT token. |
| 4 | Interval error | An NRDY interrupt is not generated. |

Table 32B.32 Error Detection during Reception of Data Packet

| Error Detection Priority | Error Type | Interrupt Generated at Error Detection and Status |
|---|---|---|
| 1 | PID error | No interrupt is generated (ignored as a corrupted packet). |
| 2 | CRC error, bit stuffing error | An NRDY interrupt is generated and the CRCE bit is set. |
| 3 | Packet size error (too large packet) | A BEMP interrupt is generated and the PID bit is set to STALL. |

#### 32B.9.9.2 DATA-PID

This module does not support high-bandwidth transfers.

The following shows actions that can be taken in response to a received PID.

1. IN direction:

(a) DATA0: Used to send packets.

(b) DATA1: Not used to send packets.

(c) DATA2: Not used to send packets.

(d) mDATA: Not used to send packets.

2. OUT direction (in full-speed operation):

(a) DATA0: Packets are received normally.

(b) DATA1: Packets are received normally.

(c) DATA2: Packets are ignored.

(d) mDATA: Packets are ignored.

3. OUT direction (in high-speed operation):

(a) DATA0: Packets are received normally.

(b) DATA1: Packets are received normally.

(c) DATA2: Packets are received normally.

(d) mDATA: Packets are received normally.

### 32B.9.9.3 Interval counter

#### (1) Outline of operation

The IITV bit in the PIPEPERI register can be used to set the interval of isochronous transfer. The interval counter enables the functions listed in **Table 32B.33**.

Table 32B.33 Functions of the Interval Counter

| Transfer Direction | Function | Detecting Condition |
|---|---|---|
| IN | Transmit buffer flush function | An IN token cannot successfully be received within the interval frame during isochronous IN transfer. |
| OUT | Notification of unreceived token | An OUT token cannot successfully be received within the interval frame during isochronous OUT transfer. |

Counting of intervals is based on received SOF packets or interpolated SOFs. Therefore, even if SOF packets are damaged, the isochronism can still be maintained. Frame intervals are set as 2n (micro) frames, where n is the value of the IITV bit.

#### (2) Interval counter initialization

This module initializes the interval counter under the following conditions:

(a) Power on reset
The IITV bit is initialized.

(b) Clearing of the buffer memory by the ACLRM bit
The IITV bit is not initialized but the counter is initialized.

(c) USB bus reset

After the interval counter is initialized and a packet is successfully transferred, counting of intervals starts under the following conditions:

1) An SOF packet is received after data is sent in response to an IN token when PID = BUF.

2) An SOF packet is received after data is received in response to an OUT token when PID = BUF.

Note that the interval counter is not initialized in the following conditions:

(a) The PID bit is set to NAK or STALL.
The interval timer is not stopped at this interval. The transaction will be attempted at the next interval.

(b) USB bus reset or USB suspension
The IITV bit is not initialized. When an SOF packet is received, counting starts from the value existing before reception.

### 32B.9.9.4 Send data setup for isochronous transfer

This module becomes able to send data packets by isochronous transfer from the next frame after data is written to the buffer memory and then an SOF packet is detected. This is called “send data setup for isochronous transfer”.

This function can identify the frame with which data sending started.

If the buffer memory is in a double-buffer configuration and writing to both buffers has been completed, only the buffer to which writing finished earlier can transfer data. Therefore, even when several IN tokens are received within the same frame, only one packet of data is sent from the buffer memory.

When an IN token is received, if the buffer memory is ready for sending data, the data is transferred and a normal response is returned. However, if the buffer memory is not ready for sending data, a zero-length packet is sent and an underrun error occurs.

**Figure 32B.11** shows examples of sending using the send data setup function for isochronous transfer by setting “IITV = 0” (each frame) in this module.

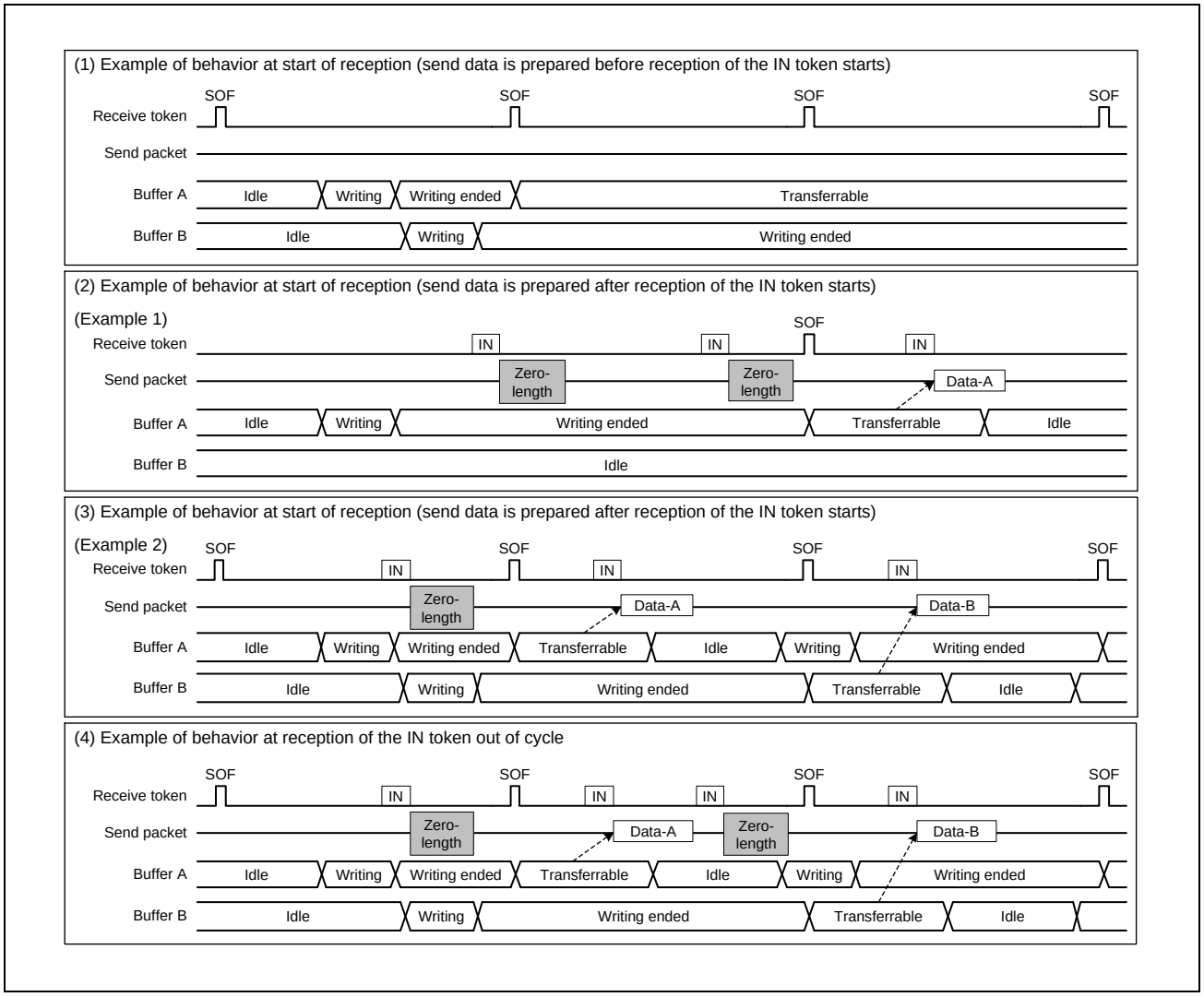



Figure 32B.11 Examples of Data Setup Function Behavior

#### 32B.9.9.5 Transmit buffer flush for isochronous transfer

If this module does not receive an IN token in an interval frame and receives a (micro) SOF packet in the next frame during isochronous data transfer, this module handles the IN token as a corrupted token and clears the buffer that can send data to make the buffer writable.

At this time, if the buffer memory is in a double-buffer configuration and writing to both buffers has been completed, this module assumes the discarded buffer memory to be sent within the same interval frame. As a result, the buffer memory that is not discarded by reception of a (micro) SOF packet becomes to be able to transfer data.

The operation start timing of the buffer flush function varies with the value of the IITV bit.

1. If IITV is 0
   Buffer flush operation is performed from the first frame after the pipe is enabled.
2. If IITV is not 0
   Buffer flush operation is performed after the first successful transaction.

**Figure 32B.12** shows an example of how the buffer flush function of this module behaves. For a token outside the set interval (token prior to the interval frame), however, this module sends the written data or a zero-length packet as an underrun error according to the data setup state.

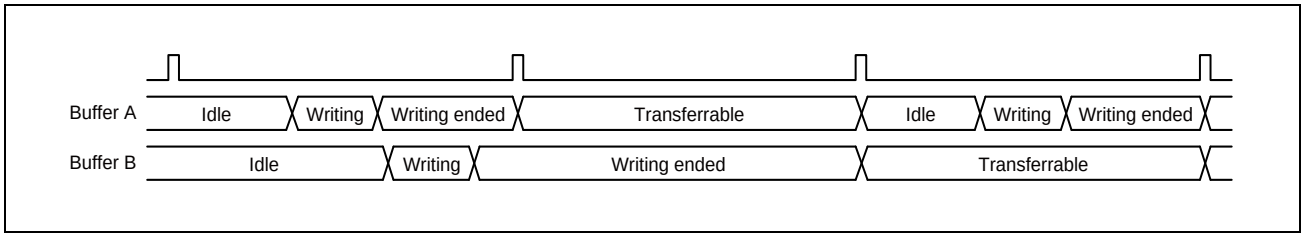



Figure 32B.12 Example of Buffer Flush Function Behavior

**Figure 32B.13** shows an example of an interval error that occurs in this module.

There are five types of interval errors, as listed below. At timing (1) in the figure, an interval error occurs and the buffer flush function operates.

If an interval error occurs during IN transfer, the buffer flush function starts. If an interval error occurs during OUT transfer, an NRDY interrupt occurs.

Use the OVRN bit to determine whether an error is an NRDY interrupt (such as a receive packet error) or an overrun error.

In the figure, responses to tokens indicated as shaded boxes are made in accordance with the buffer memory state.

1. IN direction:

(a) If the buffer is ready to transfer data, data is transferred and a normal response is returned.

(b) If the buffer is not ready to transfer data, a zero-length packet is sent and an underrun error occurs.

2. OUT direction:

(a) If the buffer is ready to receive data, data is received and a normal response is returned.

(b) If the buffer is not ready to receive data, data is discarded and an overrun error occurs.

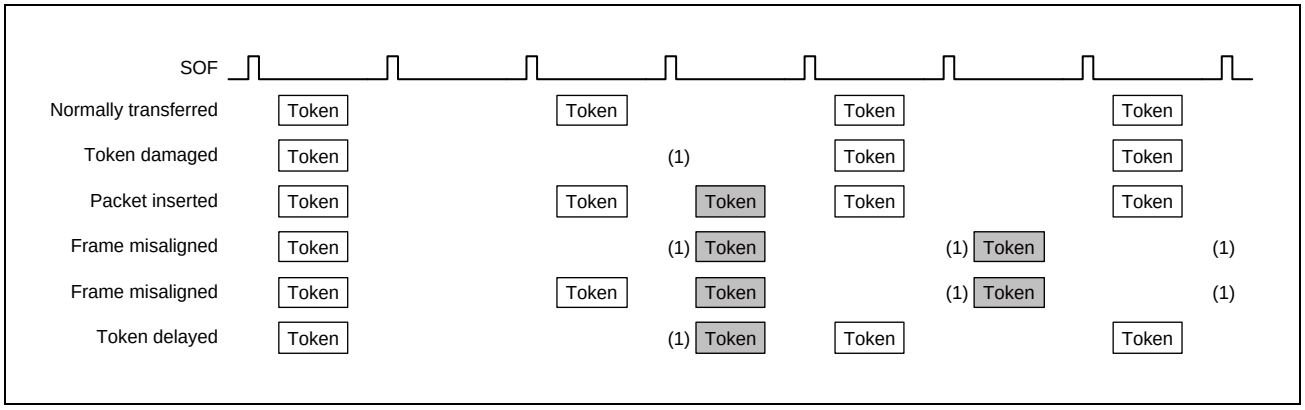



Figure 32B.13 Example of Occurrence of Interval Error (when IITV is 1)

### 32B.9.10 SOF Interpolation Function

If the controller cannot receive data at intervals of 1 ms (in full-speed operation) or 125 µs (in high-speed operation) due to corruption or missing of an SOF packet, the controller internally interpolates the SOF. The controller starts SOF interpolation upon receiving an SOF packet when both the USBE bit and SUSPM bit are set to 1.

The interpolation function is initialized under the following conditions:

(1) Power on reset

(2) USB bus reset

(3) Suspended state detected

The SOF interpolation operates according to the following specifications:

(1) Frame interval (125 µs or 1 ms) is based on the results of the reset handshake protocol.

(2) The interpolation function does not operate until an SOF packet is received.

(3) After receiving the first SOF packet, this module interpolates the SOF by using the 60-MHz internal clock to measure 125 µs or 1 ms.

(4) After receiving the second or a subsequent SOF packet, this module interpolates the SOF by using the previous reception interval.

(5) Interpolation is not performed in the suspended state or while a USB bus reset is being received.
(If this module enters the suspended state in high-speed operation, interpolation continues for 3 ms after receiving the last packet.)

The SOF interpolation function operates with the following functions:

(1) Updating of frame number or micro-frame number

(2) SOFR interrupt and micro SOF lock

(3) SOF pulse output

(4) Counting of isochronous transfer intervals

If an SOF packet is lost in full-speed operation, the FRNM bit in the FRMNUM register is not updated.

If a micro SOF packet is lost in high-speed operation, the UFRNM bit in the uFRMNUM register is updated.

However, if a micro SOF packet for which “micro-FRNM = 000” is set is lost, the FRNM bit is not updated. In this case, the FRNM bit is not updated even if subsequent micro SOF packets for which “micro-FRNM = 000” is not set are successfully received.

### 32B.9.11 Link Power Management Processing

According to the Link Power Management specification, the existing suspend state is redefined as the L2 state and a new L1 state is defined as a state where transition and resumption at a lower latency than L2 (suspend) are possible.

The table below compares the features of the L2 (suspend) state and L1 state.

Table 32B.34 Comparison Between Suspend (L2) State and L1 State

| Item | L1 | Suspend (L2) |
|---|---|---|
| Transition | LPM Transaction | 3-ms idle period |
| Host-activated resumption | (Host)<br>Minimum drive period specifiable by the host. Specified between 75 µs to 1.175 ms<br>(Device)<br>10-µs K drive | (Host)<br>Min. 20-ms K drive<br><br>(Device)<br>10-ms K drive |
| Device-activated resumption | (Device)<br>50-µs K drive<br>(Host)<br>60- to 990-µs K drive<br>(Device)<br>10-µs K drive | (Device)<br>1-ms to 15-ms K drive<br>(Host)<br>Min. 20-ms K drive<br>(Device)<br>10-ms K drive |
| Signaling | Low and Full Speed Idle | Low and Full Speed Idle |

The following describes the processing for transition to and resumption from the L1 state.

#### 32B.9.11.1 Descriptor

This module must return its own descriptor when receiving the GetDescriptor command.

Whether the contents of the descriptor to be returned need to be modified is dependent on whether this module responds to the transition to and resumption from the L1 state with an LPM transaction. The details are summarized in the table below.

Table 32B.35 Relationship between LPM Response and Descriptor

| LPM Response | bcdUSB | Presence of USB 2.0 Ex.Desc | USB 2.0 Ex.Desc LPM | Response When LPM Is Received | Remarks |
|---|---|---|---|---|---|
| Not respond | 0200 | Not present | — | Not Respond | Standard action in the case where this module does not respond to LPM |
| | 0201 | Present | LPM = 0 | STALL | This is when rejection of response to LPM is explicitly declared. In this case, it is necessary to send a STALL response instead of making no response. |
| Respond | 0201 | Present | LPM = 1 | ACK or NYET | Standard action in the case where this module responds to LPM |

Whether to respond to transition to and resumption from L1 is declared by the LPM bit of the USB 2.0 extension descriptor. To provide this module with the USB 2.0 extension descriptor, it is necessary to set the bcdUSB field of the device descriptor to 0201 or greater.

When not responding to LPM, set the bcdUSB value to 0200 without providing this module with the USB 2.0 extension descriptor. In this case, it is necessary to ignore any LPM token received.

When not responding to LPM, it is also possible to set bcdUSB to 0201 and set the LPM bit of the USB 2.0 extension descriptor to 0 (noncompliant). In this case, however, it is not allowed to ignore LPM and is necessary to send a STALL response.

When responding to LPM, set bcdUSB to 0201 and the LPM bit of the USB 2.0 extension descriptor to 1 (compliant). This grants this module to send an NYET or ACK in response to an LPM token.

#### 32B.9.11.2 Basic processing

This module needs to execute the following processing.

1. Responds to the LPM token received from the host with “No response”, “ACK”, “NYET”, or “STALL” according to this module's own state.
2. Transitions to the L1 state if it fails to detect the retransmission of an LPM token for 8 μs after making an ACK response.
3. Detects a K drive of the host and performs resume processing to the idle state.
4. Performs resume processing to the idle state based on the Remote Wake signal.

For 1, the software specifies the response method according to the values of the L1RESPEN, L1RESPMD, and L1NEGOMD bits in the PL1CTRL register. The hardware makes the response that is designated by the software upon receiving an LPM token.

For 2, both retransmission control and transition to the L1 state are processed by the hardware. Transition to the L1 state can be identified through a DVST interrupt.

For 3, a RESM interrupt occurs on detection of host K in the L1 state.

For 4, starting the Remote Wake processing can be instructed to the hardware by setting the WKUP bit by the software. The specification stipulates that the software clears this bit on resumption from the L2 state. On the other hand, the hardware clears this bit on resumption from the L1 state.

#### 32B.9.11.3 HIRD value negotiation

The HIRD value contained in the LPM token is the K period of the host on resumption from the L1 state.

This module can respond with ACK if the received HIRD value falls within the desired range as specified by the L1NEGOMD and HIRDTHR bits in the L1CTRL register; otherwise, this module can respond with NYET and request the host to modify the HIRD value.

NOTE

This HIRD value negotiation function must also be supported on the host side.

## 32B.9.12 DMA Mode

### 32B.9.12.1 Register mode/link mode

The DMS bit in the CHCFG_n register can be used to switch between register mode and link mode.

Table 32B.36 DMA mode settings

| DMS (CHCFG) | Mode | Description |
|---|---|---|
| 0 | Register mode | Performs DMA transfer based on the values set by Next Register Set. |
| 1 | Link mode | Accesses the descriptor area, and executes DMA transfer based on the values set by descriptors. This module repeats descriptor reading and DMA transfer unless the descriptor settings are changed or the control register is used to stop the processing. |

### (1) Register mode

In register mode, this module executes DMA transfer based on the values set in internal registers.

Two sets of the transfer-source address, transfer-destination address, and number of bytes to be transferred can be held (in Next0 Register Set and Next1 Register Set registers). One of these Next registers can be used to execute transfer, and both Next registers can be used to execute continuous transfer.

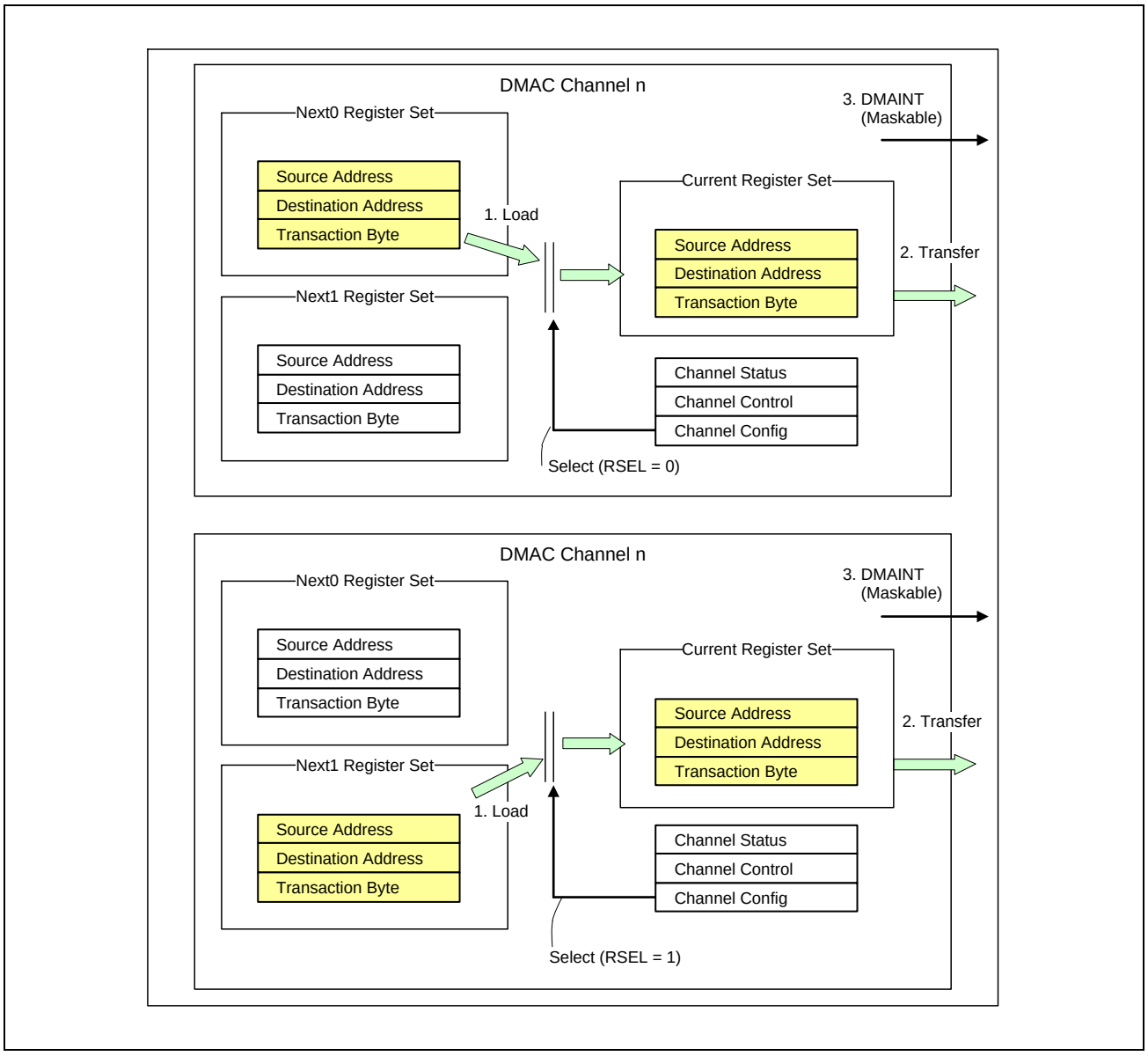



Figure 32B.14 Overview of Normal Behavior of Register Mode

The upper part of the above figure indicates a case when Next0 Register Set is processed. The lower part of the above figure indicates a case when Next1 Register Set is processed.

### (a) Operation flow of register mode

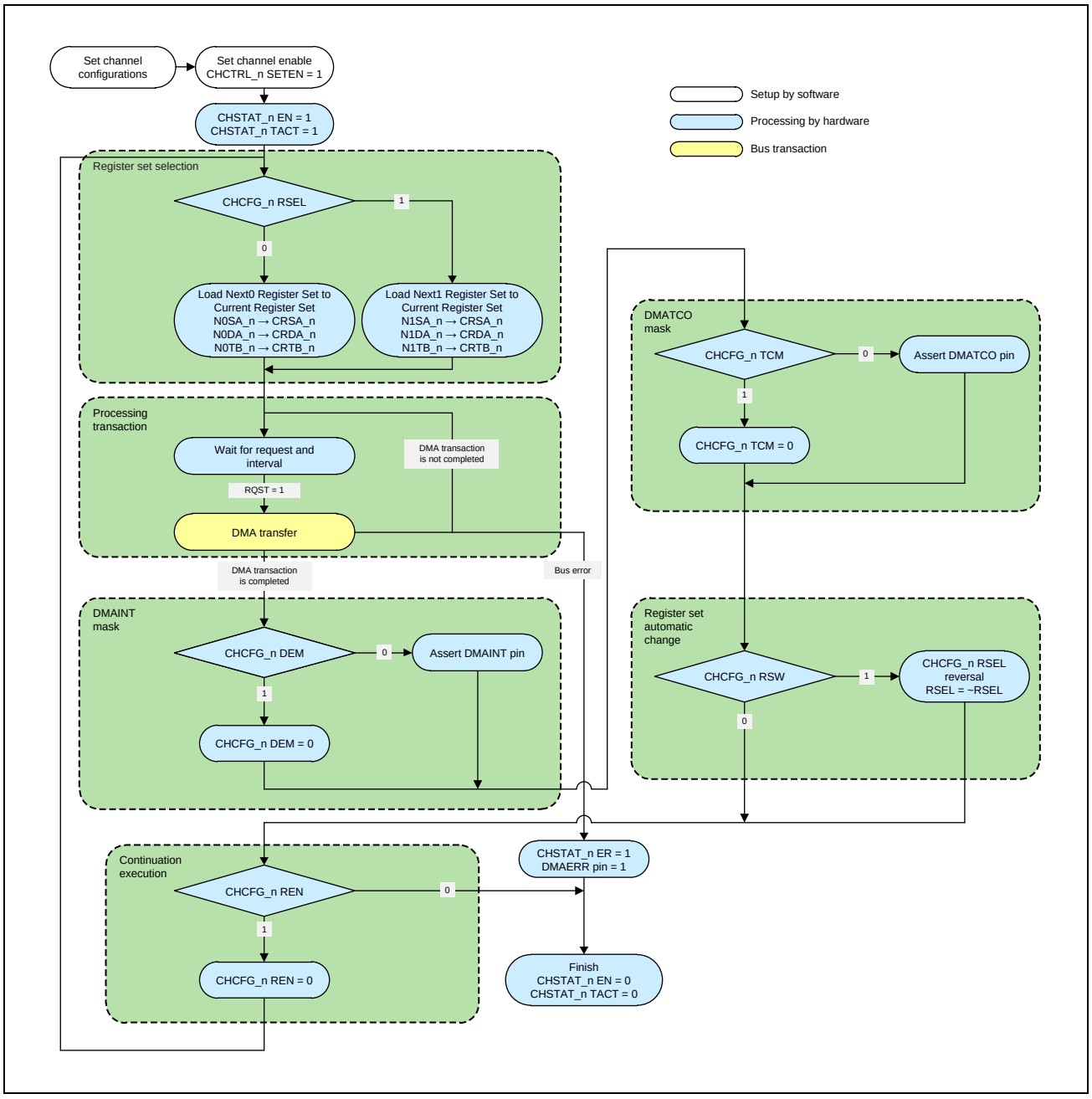



Figure 32B.15 Register Mode Operation Flow

**Description of the register mode operation flow:**

1. Channel setting
   The Next0 Register Set or Next1 Register Set register is set (transfer-destination address, transfer-source address, and total number of bytes to be transferred). In addition, the FIFO channel, volume of transfer data, and other items are set for USB control that is used for the Channel Register Set register.
   No software settings are required in OUT transfer (reception in peripheral mode) or in IN transfer (transmission in peripheral mode) because the hardware sets the transfer source or destination address according to the setting of the CURPIPE bits in the DxFIFOSEL register.
   (see **Section 32B.9.13, DMA Transfer**).

2. Register set selection
   When 1 is written to the SETEN bit in the CHCTRL_n register, the EN and TACT bits in the CHSTAT_n register are set to 1. As a result, the values set by the Next Register Set register selected by the RSEL bit in the CHCFG_n register are loaded to the Current Register Set register.

3. DMA transaction
   A DMA transaction is executed based on the values that are set. For details about transfer, **Section 32B.9.13, DMA Transfer**.

4. The USBFDMAmn masking
   The USBFDMAmn interrupt is masked depending on the value of the DEM bit in the CHCFG_n register. If DEM = 1, the USBFDMAmn interrupt is masked, and the DEM bit is automatically cleared to 0.

5. DMATC masking
   DMATC from DMAC control to USB control is masked depending on the value of the TCM bit in the CHCFG_n register. If TCM = 1, DMATC is masked, and the TCM bit is automatically cleared to 0.

6. Automatic register set switchover
   Whether the current Next register set is to be switched to the other Next register set is determined by the value of the RSW bit in the CHCFG_n register.

7. Continuation of execution
   Whether to continue DMA transfer is determined by the value of the REN bit in the CHCFG_n register. If REN = 0, the EN and TACT bits in the CHSTAT_n register are cleared to 0, and DMAC operation stops. If REN = 1, DMAC operation continues, and the REN bit is automatically cleared to 0.

### (b) Register mode setting

- Register mode setting

  The register set to be processed is selected.

Table 32B.37 Register Mode Setting

| DMS (CHCFG_n) | RSEL (CHCFG_n) | Description |
|---|---|---|
| 0 | 0 | Processes Next0 Register Set. |
| | 1 | Processes Next1 Register Set. |

- USBFDMAmn masking

  The USBFDMAmn interrupt can be masked.

Table 32B.38 USBFDMAmn Mask Setting

| DEM (CHCFG_n) | Description |
|---|---|
| 0 | Asserts the USBFDMAmn interrupt when the DMA transaction is completed. |
| 1 | Does not assert the USBFDMAmn interrupt even when the DMA transaction is completed.<br>After the DMA transaction is completed, the DEM bit is cleared to 0. |

- DMATC mask setting

  DMATC from DMAC control to USB control can be masked.

Table 32B.39 DMATC Mask Setting

| DEM (CHCFG_n) | Description |
|---|---|
| 0 | Asserts DMATC when the DMA transaction is completed. |
| 1 | Does not assert DMATC even when the DMA transaction is completed.<br>After the DMA transaction is completed, the TCM bit is cleared to 0. |

- Automatic transaction execution for a register set

  After a DMA transaction finishes, another DMA transaction can be executed.

Table 32B.40 Automatic Execution Setting for a Register Set

| REN (CHCFG_n) | Behavior | Remarks |
|---|---|---|
| 0 | The EN bit is cleared and DMA operation is terminated when the DMA transaction for the register set that is set by RSEL finishes. | Use this setting to execute a DMA transaction only once. |
| 1 | After a DMA transaction finishes, DMA transfer of the contents of the next register set continues. The REN bit is cleared to 0 when continuous transfer is successful. | Use this setting to continue processing of register set contents. |

- Automatic register set switchover setting

  After a DMA transaction finishes, the next register set to be processed can be switched.

Table 32B.41 Automatic Execution Setting for a Register Set

| RSW (CHCFG_n) | Behavior | Remarks |
|---|---|---|
| 0 | The register set is not switched when a DMA transaction finishes. | Use this setting to use only one register set. |
| 1 | When REN = 1 and a DMA transaction finishes, the RSEL setting is automatically reversed to select the other register set. | Use this setting to switch the register set. |

### (c) Example of setting the register mode

- Example of setting the register mode when using only the Next0 register set

Table 32B.42 Register Mode Setting Example

| DMS (CHCFG_n) | RSEL (CHCFG_n) | DEM (CHCFG_n) | TCM (CHCFG_n) | RSW (CHCFG_n) | REN (CHCFG_n) |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 |
| (register mode) | (Next0) | (not masked) | (not masked) | (not switched) | (continuous execution disabled) |

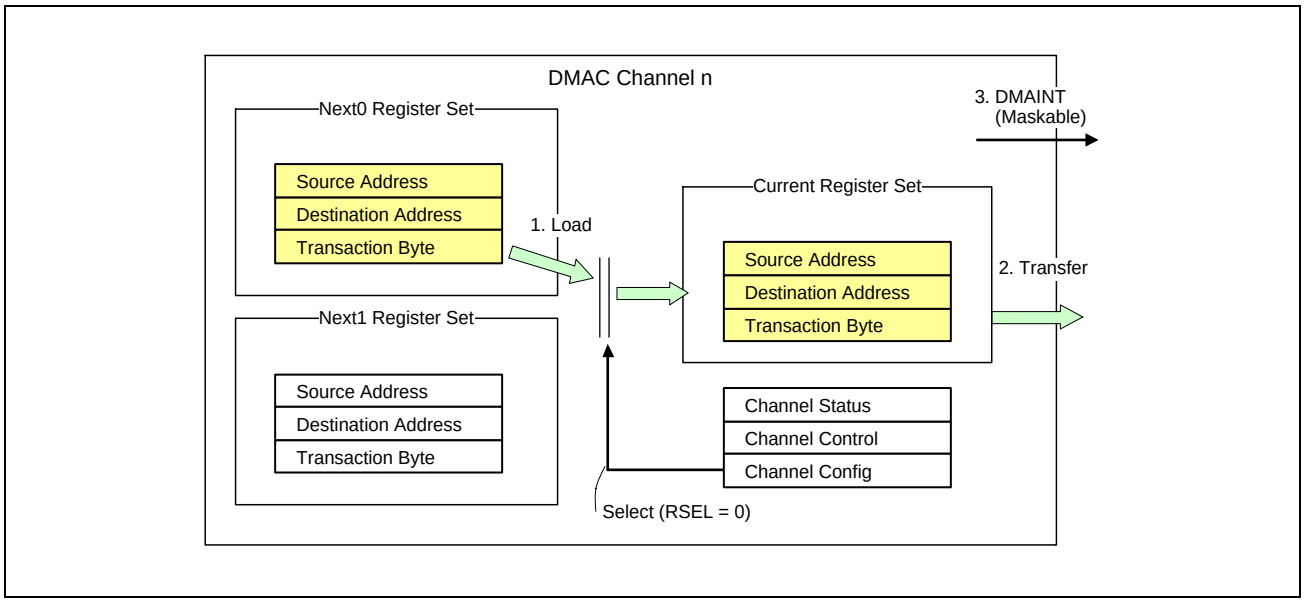



Figure 32B.16 Register Mode Setting Example 1

1. By writing 1 to the SETEN bit in the CHCTRL_n register, the EN bit in the CHSTAT_n register is set to 1. As a result, the contents of Next0 Register Set are loaded to Current Register Set.
2. A DMA transaction is executed based on the values of Current Register Set and Channel Register Set.
3. Because the DEM bit in the CHCFG_n register is 0, the USBFDMAmn interrupt is asserted after the DMA transaction finishes.
4. Because the TCM bit in the CHCFG_n register is 0, DMATC is asserted after the DMA transaction finishes.
5. Because the REN bit in the CHCFG_n register is 0, the EN bit in the CHSTAT_n register is cleared to 0, and the operation ends.

- Example of setting the register mode when using two register sets continuously

Table 32B.43 Automatic Register Set processing Setting

| DMS (CHCFG_n) | RSEL (CHCFG_n) | DEM (CHCFG_n) | TCM (CHCFG_n) | RSW (CHCFG_n) | REN (CHCFG_n) |
|---|---|---|---|---|---|
| 0 (register mode) | 0 (Next0) | 1 (masked) | 0 (not masked) | 1 (switched) | 1 (continuous execution enabled) |

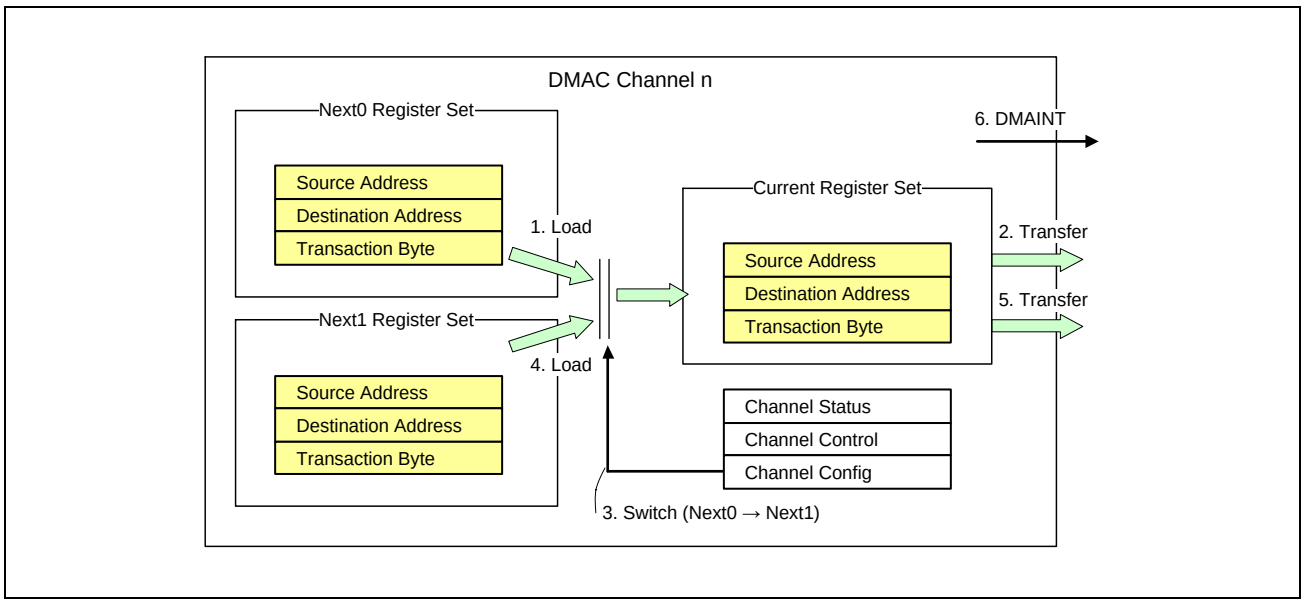



Figure 32B.17 Register Mode Setting Example 2

1. By writing 1 to the SETEN bit in the CHCTRL_n register, the EN bit in the CHSTAT_n register is set to 1. As a result, the contents of Next0 Register Set are loaded to Current Register Set.
2. A DMA transaction is executed based on the values of Current Register Set and Channel Register Set.
3. Because the DEM bit in the CHCFG_n register is 1, the USBFDMAmn interrupt is not asserted after the DMA transaction finishes.
4. Because the REN bit in the CHCFG_n register is 1, operation continues. The REN bit is automatically cleared to 0.
5. Because the RSW bit in the CHCFG_n register is 1, the next register set to be processed is switched (RSEL = 0 to 1).
6. The contents of Next1 Register Set are loaded to Current Register Set.
7. A DMA transaction is executed based on the values of Current Register Set and Channel Register Set.
8. Because the DEM bit in the CHCFG_n register is 0, the USBFDMAmn interrupt is asserted after the DMA transaction finishes.
9. Because the TCM bit in the CHCFG_n register is 0, DMATC is asserted after the DMA transaction finishes.
10. Because the REN bit in the CHCFG_n register is 0, the EN bit in the CHSTAT_n register is cleared to 0, and the operation ends.

### (2) Link mode

In link mode, this module reads the value set in a descriptor placed in an external storage area to execute a DMA transaction. In DMAC, there are Next Link Address (NXLA_n) and Current Link Address (CRLA_n) registers for each channel. The Next Link Address (NXLA_n) register is used to set the address of the descriptor to be read the next time. The Current Link Address (CRLA_n) register is used to display the descriptor address for the current DMA transaction.

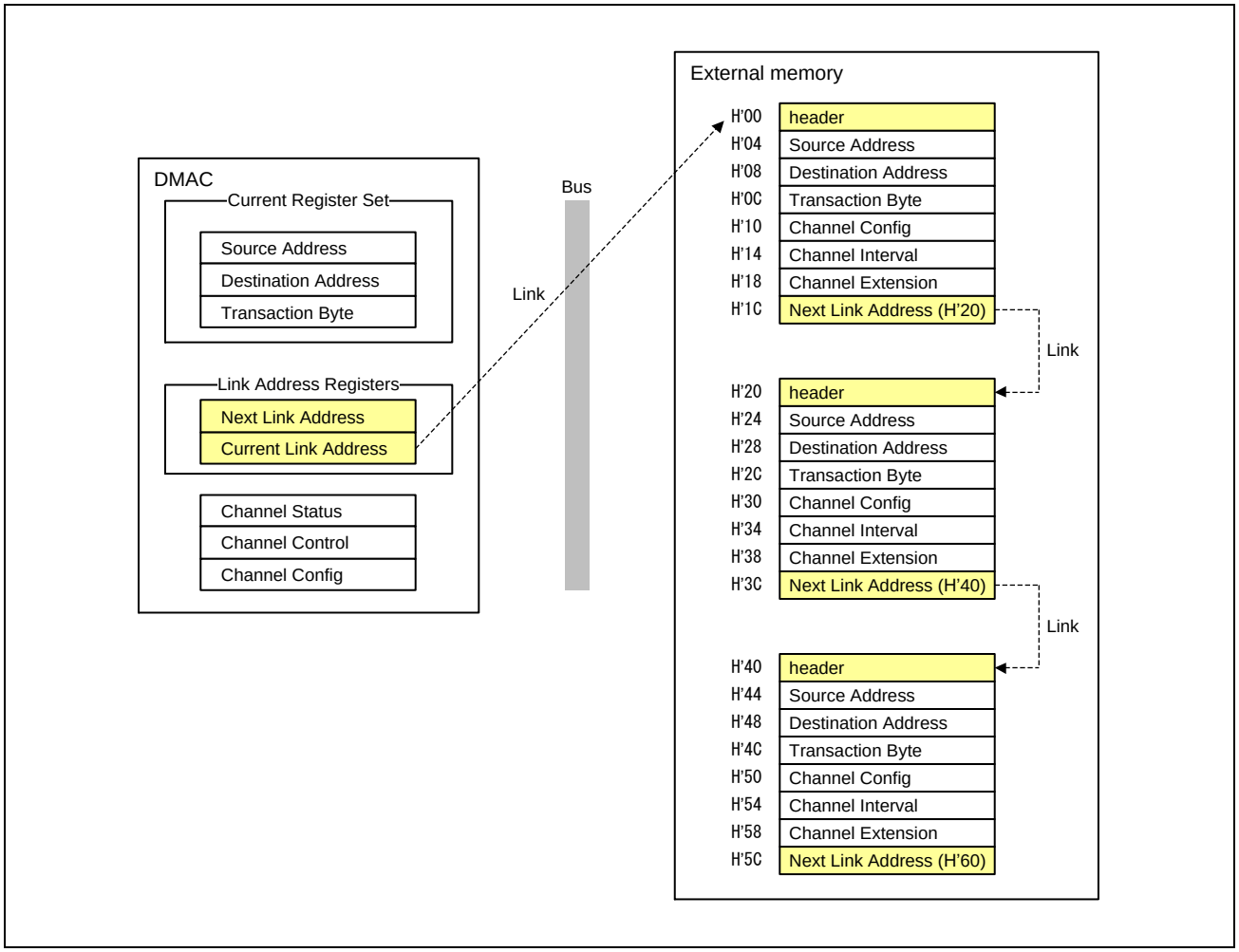



Figure 32B.18 Overview of Link Mode

### (a) Operation flow of link mode

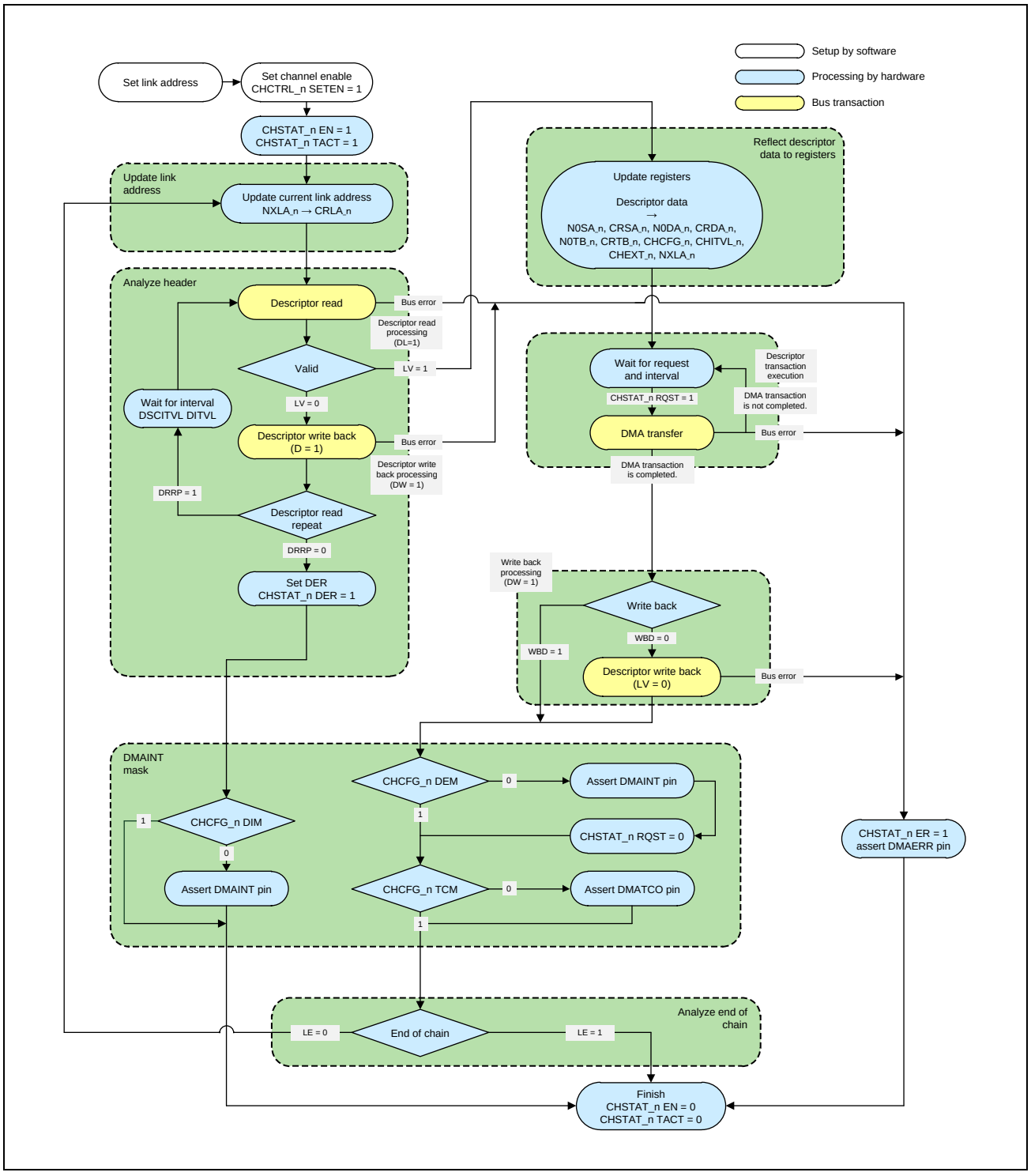



Figure 32B.19 Link Mode Operation Flow

**Description of the link mode operation flow:**

1. Channel setting
   The beginning address of the link destination is set in the NXLA_n register.

2. Link address updating
   If 1 is written to the SETEN bit in the CHCTRL_n register, the EN and TACT bits of the CHSTAT_n register are set to 1. As a result, the address set in the NXLA_n register is loaded to the CRLA_n register.

3. Descriptor reading and header judgment
   A read of the descriptor starts, and DMAC checks the contents of header. If LV = 0, this module writes 1 back to the D bit of header. After that, if the DRRP bit in the CHCFG_n register is 1, this module waits for the time intervals set by the DSCITVL register, and then reads the same descriptor again. If DRRP = 0, the DER bit in the CHSTAT_n register is set to 1, and this module is placed in the end state (both EN and TACT bits in the CHSTAT_n register are 0). At this time, if the DIM bit in the CHCFG_n register is 0, this module asserts the USBFDMAmn interrupt.

4. Descriptor setting
   If LV = 1, the data read from the descriptor is loaded to Current Register Set and Channel Register Set. In addition, the next link target is loaded to the NXLA_n register.

5. DMA transaction
   A DMA transaction is executed based on the values that are set. For details about transfer, see **Section 32B.9.13, DMA Transfer**.

6. Header write-back
   If WBD of header is 0, DMAC writes LV = 0 to the header area.

7. USBFDMAmn masking
   If the DEM bit in the CHCFG_n register is 0, this module asserts the USBFDMAmn interrupt.

8. DMATC masking
   If the TCM bit in the CHCFG_n register is 0, this module asserts DMATC.

9. Link end judgment
   If LE of header is 1, the EN and TACT bits in the CHSTAT_n register are cleared to 0, and DMAC terminates operation. If LE is 0, this module updates Current Register Set, and then restarts reading the next descriptor.

### (b) Register setting

- Link mode setting

  To use link mode, set the DMS bit in the CHCFG_n register to 1.

Table 32B.44 Link Mode Setting

| DMS (CHCFG_n) | Description |
|---|---|
| 1 | This module operates in link mode. The setting of this bit cannot be changed by using a descriptor. |

- Link address setting

  The Next Link Address (NXLA_n) and Current Link Address (CRLA_n) registers are used to indicate a link target. Before starting link mode, set the link target in the NXLA_n register.
  After reading a descriptor, this module updates the NXLA_n register to the next link. Note that the CRLA_n register indicates the address of the link target that is being executed.

Table 32B.45 Link Address Register Set

| Register | Description |
|---|---|
| Next Link Address Register (NXLA_n) | This register is used to set and display the next link target. Before starting link mode, set the address of the link target in this register. |
| Current Link Address Register (CRLA_n) | This register is used to display the link target that is being executed. This is a read-only register. |

### (c) Descriptor setting

DMAC supports multiple descriptor formats.

A switchover between formats is specified by using the DSCFM field of bits [31:28] of the 1st word (header) of the descriptor.

The following table shows the relationship between DSCFM values and descriptor formats.

Table 32B.46 Descriptor Formats

| DSCFM | Descriptor Size | Next Link Address | Channel Extension | Channel Interval | Channel Config | Transaction Size | Destination Address | Source Address | header |
|---|---|---|---|---|---|---|---|---|---|
| 3 | 4 words | ✓ | — (reload) | — (reload) | — (reload) | — (header) | ✓ | ✓ | ✓ (with STS) |
| 1 | 8 words | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ (no STS) |
| Other than the above | If DSCFM is set to a value that is not 1 or 3, operation cannot be guaranteed. Make sure that DSCFM is set to 1 or 3. | | | | | | | | |

Table 32B.47 Explanation of the Marks in **Table 32B.46**

| Field | Mark | Description | Remarks |
|---|---|---|---|
| Header | ✓ (with STS) | The STS field of bits [15:0] in the header is valid.<br>The value set in the STS field is used as the total number of transfer bytes (Transaction Size). | — |
| | ✓ (no STS) | The STS field of bits [15:0] in the header is invalid.<br>The value of "Transaction Size" in the descriptor is used as the total number of bytes. | |
| Source Address | ✓ | Specify the source address. | — |
| Destination Address | ✓ | Specify the destination address. | — |
| Transaction Size | ✓ | Specify the transaction size. | — |
| | — (header) | Omit the transaction size.<br>The value set in the STS field is used as the total number of transfer bytes (Transaction Size) | Because the STS field is of 16 bits, a maximum of 65,535 bytes can be set. |
| Channel Config<br>Channel Interval<br>Channel Extension | ✓ | Specify Channel Config, Channel Interval, and Channel Extension. | — |
| | — (reload) | Omit Channel Config, Channel Interval, and Channel Extension.<br>The previous settings (the values of the CHCFG_n, CHITVL_n, and CHEXT_n registers of the last time) are inherited. | — |
| Next Link Address | ✓ | Specify the next descriptor address (Next Link Address) to be read after DMA transfer of the descriptor. | — |

DMAC sequentially interprets data read from descriptors. If the number of words specified for DSCFM is less than 8, place the data of descriptors that are indicated by "✓" in **Table 32B.46** on memory.

No software settings are required in OUT transfer (reception in peripheral mode) or in IN transfer (transmission in peripheral mode) because the hardware sets the transfer source or destination address according to the setting of the CURPIPE bits in the DxFIFOSEL register.

Table 32B.48 Example of Placing Descriptors

| DSCFM | Address | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Link Address + H'1C | Link Address + H'18 | Link Address + H'14 | Link Address + H'10 | Link Address + H'0C | Link Address + H'08 | Link Address + H'04 | Link Address + H'00 |
| H'3 | — | — | — | — | Next Link Address | Destination Address | Source Address | header |
| H'1 | Next Link Address | Extension | Interval | Config | Transaction Byte | Destination Address | Source Address | header |

- header

The header area provides the descriptor status and other information as shown below.

DMAC reads this area before DMA transfer in link mode starts. After a DMA transaction terminates, DMAC writes the transfer status back to this area.

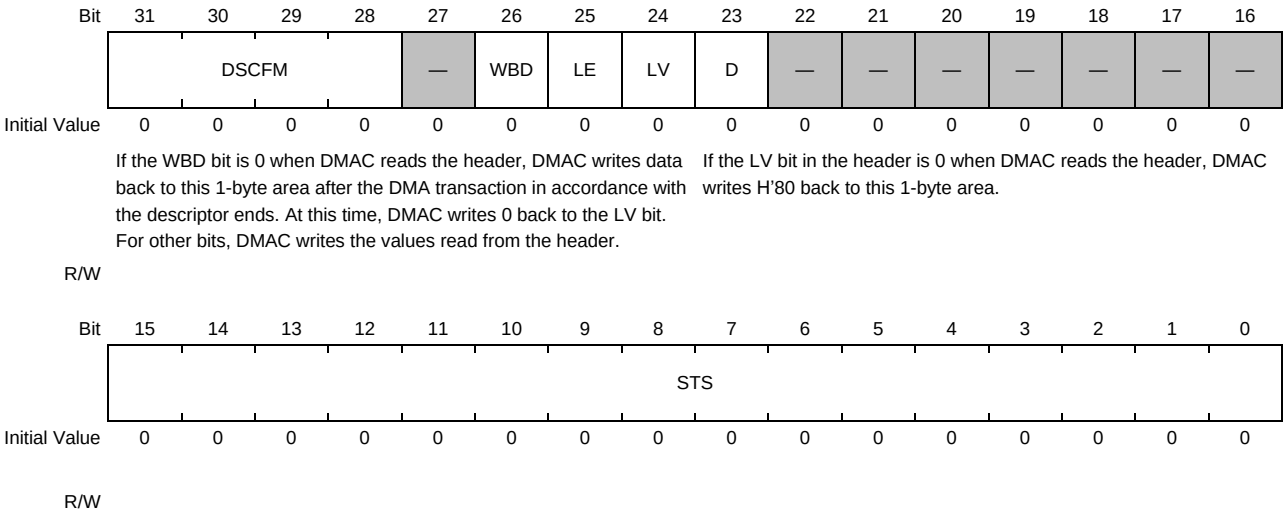



Figure 32B.20 Header Area

Table 32B.49 Header Area (1/2)

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 28 | DSCFM | | | Descriptor Format<br>Specifies the descriptor format (length and combination of descriptors).<br>For details see **Table 32B.46**. |
| 27 | — | | | Reserved area. Set 0. |
| 26 | WBD | | | Wrazite Back Disable<br>Masks a write-back operation for the LV bit. If this bit is 1, DMAC does not perform a write-back operation.<br>0: Writes 0 back to the LV bit.<br>1: Does not perform a write-back operation for the LV bit. |
| 25 | LE | | | Link End<br>Indicates that the link will end with the DMA transaction for this descriptor.<br>Set this bit to 1 to indicate the end of link.<br>0: The link continues.<br>1: The link ends. |

Table 32B.49 Header Area (2/2)

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 24 | LV | | | Link Valid<br>Indicates that this descriptor is valid.<br>If WBD = 0, after DMAC executes the DMA transaction written in the descriptor, DMAC writes 0 to this bit. When header is set, set 1 to this bit.<br>0: This descriptor is invalid.<br>1: This descriptor is enable. |
| 23 | D | | | Descriptor Error<br>Indicates a descriptor access error. If LV is 0 when the descriptor is read, DMAC writes 1 back to this bit.<br>0: A descriptor error has not occurred.<br>1: LV was 0 when the descriptor was read. |
| 22 to 16 | — | | | Reserved area. Set 0. |
| 15 to 0 | STS | | | Short Transaction Size<br>If DSCFM is 3, the transaction size is set (in bytes). The maximum number of transfer bytes that can be set is 65,535.<br>If DSCFM is 3, Do not set 0 for STS. If 0 is set, operation cannot be guaranteed. |

If descriptors are added sequentially while DMAC is operating, the access that the CPU sets 0 to the LV bit and the access that DMAC writes 1 back to the D bit might contend with each other. If this contention occurs, prior-written data is overwritten by latter-written data.

To prevent this problem from occurring, DMAC performs a write-back operation for the D bit in a byte-write manner. Therefore, the CPU must also set LV to 1 in a byte-write manner. Because the byte lanes for the D and LV bits are different, by writing data to different areas, occurrence of this problem can be prevented.

- Settings of descriptors other than header

  The specifications of data of the descriptors other than header are the same as the specifications of internal registers.

- Settings specified when descriptors are accessed

  The MHPROT pin output can be set for the LWPR and LDPR fields of the DCTRL register when descriptors are accessed. Set it according to the access target in which descriptors are deployed.

- Descriptor areas and DMA transfer areas

  The following provides an overview of the descriptor areas and DMA transfer areas that are accessed by DMAC.

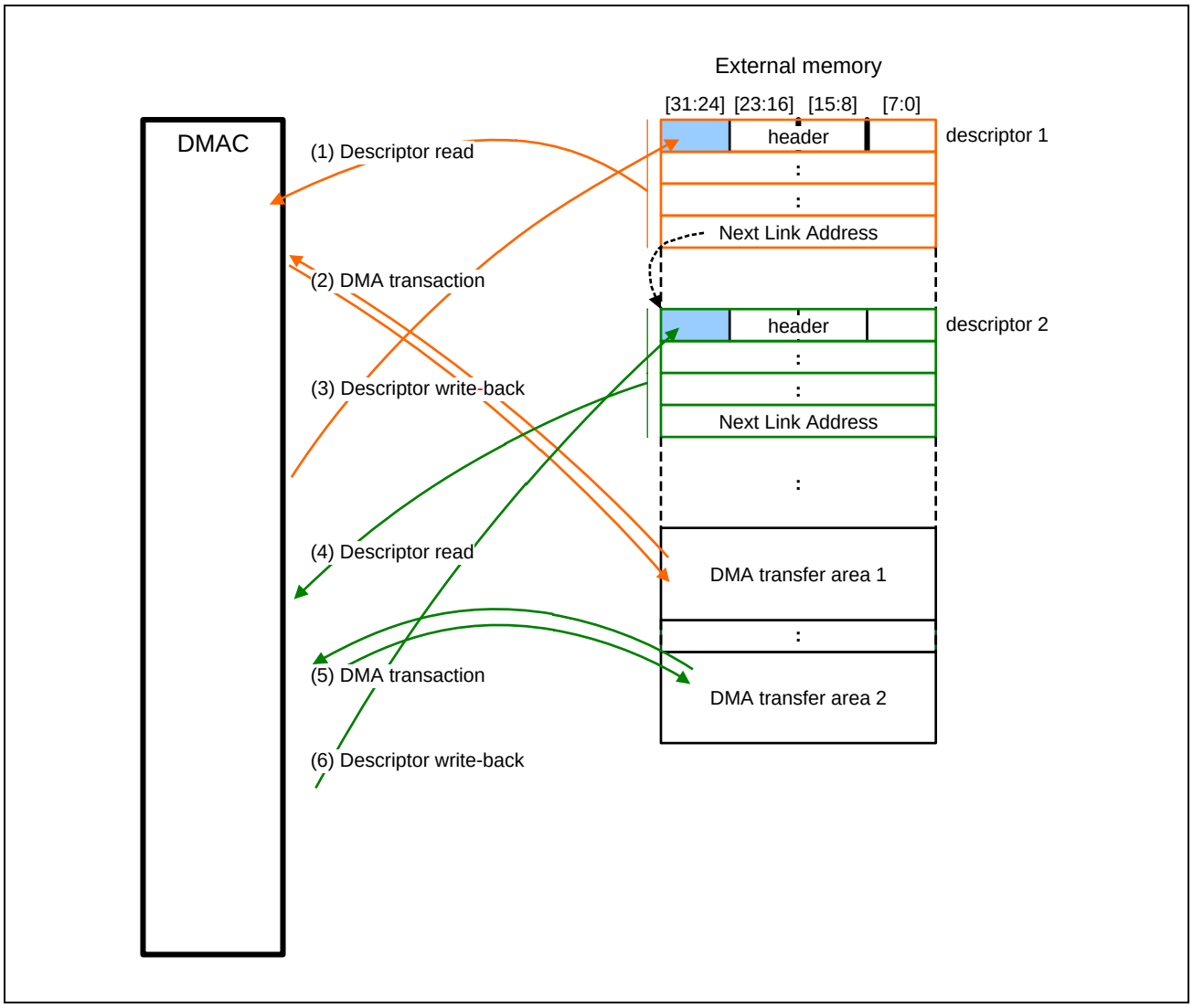



Figure 32B.21 Overview of Descriptor Areas and DMA Transfer Areas

(1) Descriptor read
DMAC loads a value from the internal NXLA_n register to the CRLA_n register, and then reads a descriptor from the external memory space indicated by the CRLA_n register (descriptor 1).

(2) DMA transfer
If the LV bit of the header descriptor is 1, DMAC performs a DMA transfer in accordance with the descriptor information.

(3) Descriptor write-back
After performing a DMA transfer of data by the number of bytes that are set, if the WBD bit in header is 0, DMAC performs write-back for bits [31:24] in header of Descriptor 1. For the LV field, 0 is written back. For other fields, the values read in (1) are written back on a byte-size basis.

(4) Descriptor read
If the value of the LE bit in the header descriptor that was read previously (in (1)) is 0, DMAC reads the next descriptor from the address (descriptor2) indicated by Next Link Address in the current descriptor.

(5) DMA transfer
If the LV bit in the header descriptor is 1, DMAC performs a DMA transfer in accordance with the descriptor information.

(6) Descriptor write-back
After performing a DMA transfer of data by the number of bytes that are set, if the WBD bit in header is 0, DMAC performs write-back for bits [31:24] in header of Descriptor 2. For the LV field, 0 is written back. For other fields, the values read in (4) are written back on a byte-size basis.

(Steps (4) to (6) are repeated.)

If LE = 1 and WBD = 0 in header, DMAC performs a DMA transfer with the descriptor settings, writes 0 back to the LV bit in header, and then terminates processing.

If LE = 1 and WBD = 1 in header, DMAC performs a DMA transfer with the descriptor settings, and then terminates processing (without performing write-back).

If LV = 0 in header, DMAC writes 1 back to the D bit in header, and then, if the DRRP bit in the CHCFG_n register is 1, DMAC waits for the number of intervals specified by the DITVL field of the DSCITVL_n register, and then reads the descriptor again. If DRRP = 0, DMAC terminates processing.

- Notes on descriptors
  - In link mode, settings can be changed by reading descriptors. However, it is impossible to synchronize the setting change times and hardware requests. Therefore, to use hardware requests, before setting the SETEN bit in the CHCTRL_n register, set the AM, LVL, HIEN, LOEN, and SEL bits in the CHCFG_n register. Note that these setting bits must not be changed in descriptors.
  - Descriptors cannot be used to change the DMS field in the CHCFG_n register (DMAC is always placed in link mode). Although descriptors can be used to change the settings of the REN, RSW, and RSEL fields in the CHCFG_n register, changes of those fields do not affect operation.
  - The descriptor can be initialized by overwriting the memory area corresponding to the descriptor you intend to initialize while the DMAC is not operating. The DMAC determines whether or not the descriptor is valid by referring to the DSCFM field and LV bit in the header. Accordingly, set the areas in memory corresponding to the DSCFM field and LV bit to 1 or 3, and to 1, respectively, before enabling DMAC operation.
  - To set the next descriptor on memory while DMAC is operating, make sure that 1 is written to the LV bit after the descriptors subsequent to header (Source Address, Destination Address, ..., Next Link Address) are set. If this is not performed and descriptor setting by the CPU and descriptor reading by DMAC contend, DMAC performs a DMA transfer using the previous values of those descriptors (Source Address, Destination Address, ..., Next Link Address).
  - To leave the write-back information for the D bit of header, make sure that 1 is written to the LV bit of header in a byte-access manner.

### (d) Link configuration example

In link mode, descriptors can be configured as shown below.

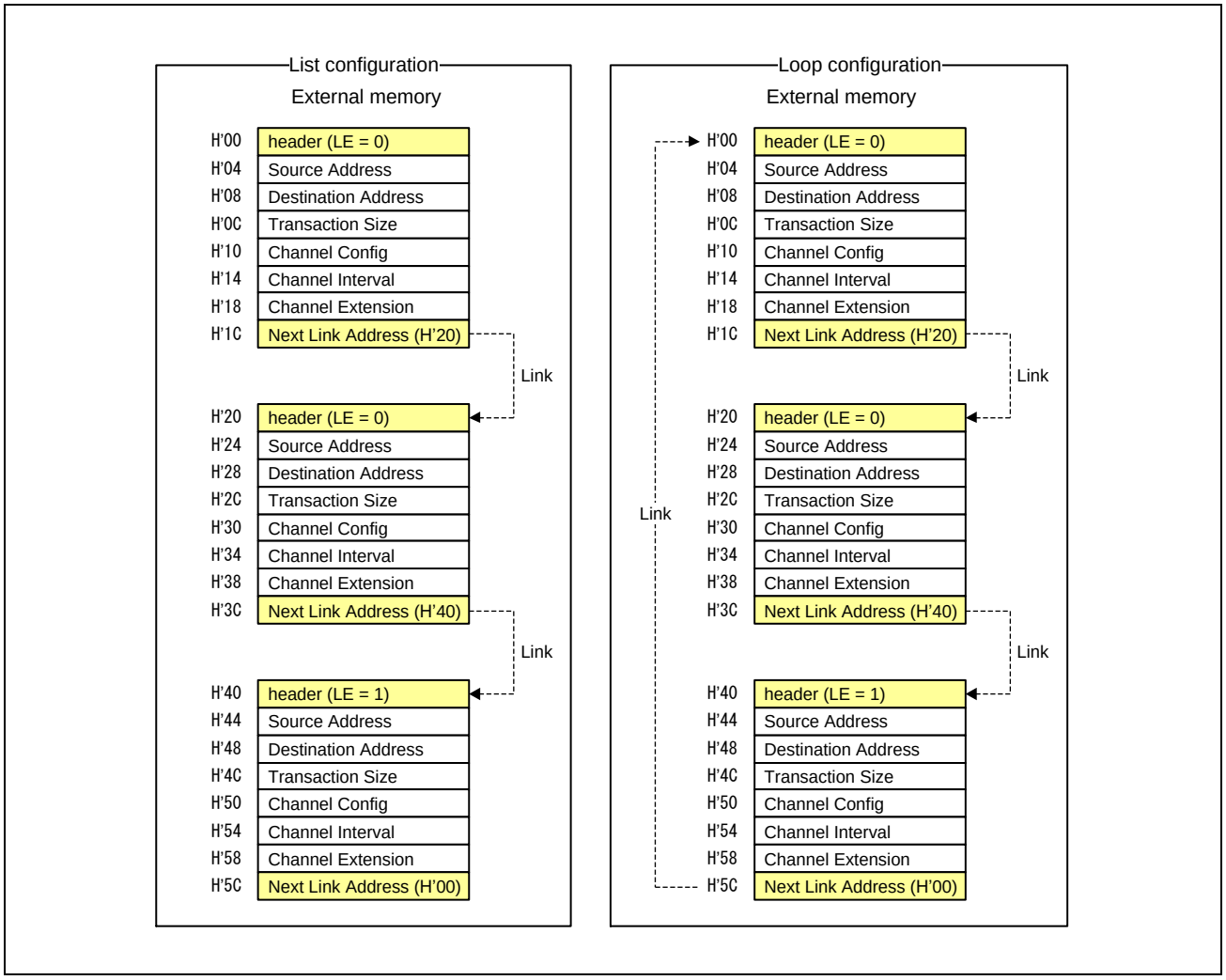



Figure 32B.22 Link Mode Configuration Example

- List configuration

  The link ends by setting 1 for the LE bit in the header of the last descriptor.

- Loop configuration

  A loop of descriptors can be created by setting the link target of the last descriptor to the address of the first descriptor. To end the loop, change the value of the LE bit of header to 1 or use the transfer interrupt procedure.

#### 32B.9.12.2 Write only mode

Write-only mode is enabled by setting 1 for the WONLY bit in the CHCFG_n register.

Table 32B.50 Write-Only Mode Setting

| WONLY (CHCFG) | Mode | Description |
|---|---|---|
| 0 | Normal mode | A DMA transfer is performed using the values set in Next Register Set. |
| 1 | Write-only mode | A DMA write transfer is performed without performing a DMA read transfer. |

In write-only mode, no DMA read transfer is performed (note that descriptors are read in the same way as in normal mode). In register mode, the values set in the NxSA_n register (if RSEL = 0, x = 0; if RSEL = 1, x = 1) are used as write data. In link mode, the values of the SA fields of descriptors are used as write data.

Use this mode to, for example, initialize the memory area.

## 32B.9.13 DMA Transfer

This chapter describes the basic operation of DMA transfer.

### 32B.9.13.1 Transfer modes

DMAC supports only single transfer mode.

Upon receiving a DMA transfer request from USB control, DMAC executes a single DMA transfer on the side (source or destination) indicated by the REQD bit in the CHCFG_n register. DMAC then asserts internal DMA permission from internal USB control to DMAC control. DMAC performs a single transfer each time a transfer is received. DMAC continues this operation by the transfer size loaded to the CRTB_n register (arbitration between channels is performed for each DMA transfer).

The timing of internal DMA permission from internal USB control to DMAC control differs depending on the setting of the REQD bit in the CHCFG_n register and the setting of the transfer size (DDS[2:0] and SDS[2:0] in the CHCFG_n register). For details, see **Section 32B.9.13.7, Operational difference depending on the transfer size**.

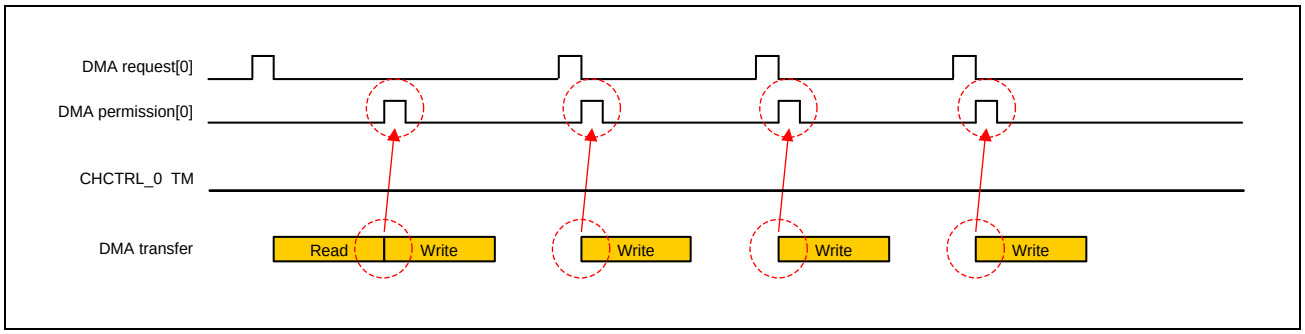



Figure 32B.23 Single Transfer Mode (REQD = 1, SDS > DDS)

### 32B.9.13.2 DMA channel priority control

DMAC supports fixed-priority mode and round-robin mode as methods of arbitration between channels. The mode is selected by using the PR bit in the DCTRL register. If the PR bit is 0, fixed-priority mode is selected. If the PR bit is 1, round-robin mode is selected.

Table 32B.51 Priority Control Setting

| Mode | PR (DCTRL) | Description | Remarks |
|---|---|---|---|
| Fixed-priority | 0 | Controls requests based on fixed priority (CH0 > CH1 > ...). | Use this mode if channels have priority. |
| Round-robin | 1 | Controls requests in a round-robin manner. | Use this mode to execute requests equally. |

#### (1) Fixed-priority mode

In fixed-priority mode, a fixed priority level is assigned to each channel as shown below.

(High) CH0 > CH1 (Low)

If DMA transfer requests simultaneously occur over multiple channels, the DMA transfer over the channel whose number is smallest is performed first.

The transfer over channel 0 is performed first. However, while a transfer switches to another transfer over channel 0, a transfer over the channel with the next highest priority level is performed in order to increase the bus usage rate.

### (2) Round-Robin Mode

In round-robin mode, each time a transfer over a channel is received, the priority level of the channel that was used for the previous transfer is changed to the lowest level.

In the status immediately after the mode is reset, channels are assigned priority levels in the same way as fixed-priority mode as shown below.

(High) CH0 > CH1 (Low)

In this status, if a transfer request for DMA channel 0 does not occur and a transfer request for DMA channel 1 occurs, the transfer over DMA channel 1 is performed. When the transfer finishes, the channel priority is changed as follows.

(High) CH1 > CH0 (Low)

The following shows an example of DMA transfer in round-robin mode.

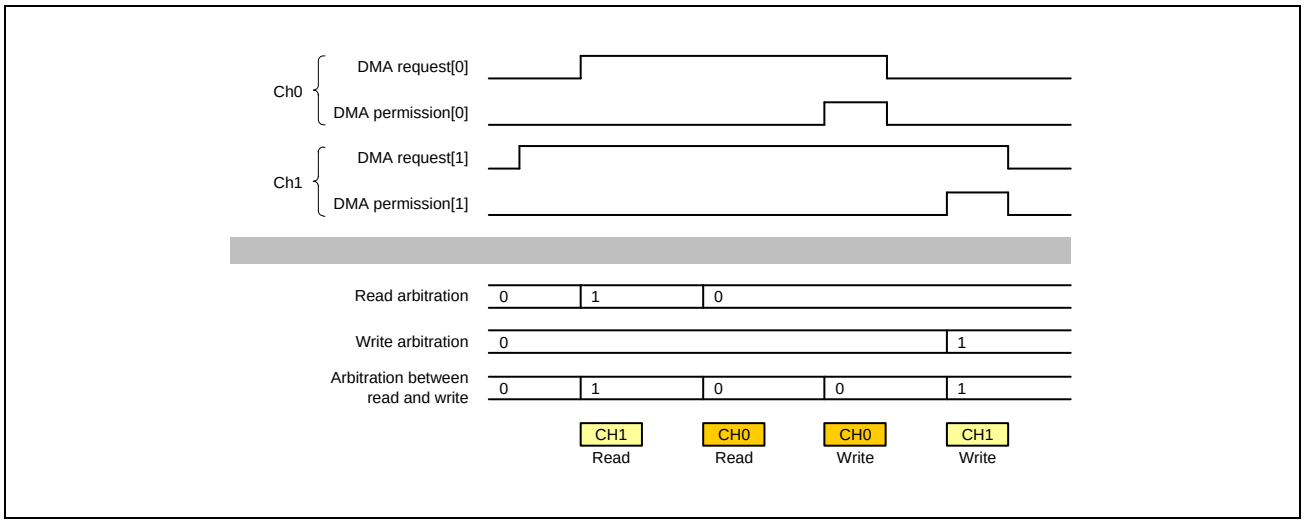



Figure 32B.24 Example of Operation in Round-Robin Mode (with 4 channels, REQD = 1)

DMAC internally performs arbitration between read channels and arbitration between write channels, further performs arbitration between the arbitration results, and then issues bus access.

### 32B.9.13.3 Forced sweeping request

When a forced sweeping request is entered, DMAC transfers the data that is left untransferred in the buffer to the destination address. After the sweep processing finishes, DMAC continues DMA transfer.

The following provides notes on forced sweeping requests:

(1) If a forced sweeping request and a transfer request from USB control contend, DMAC performs forced sweeping first, and then performs a DMA transfer.

(2) If the USB-control-side system is the destination (REQD bit in the CHCFG_n register is 1) register REQD = 1, buffer overflow or another error might occur on the destination unit because data is transferred although no DMA transfer request is made on the USB control side. Therefore, the specifications physically prohibit DMAC from using forced sweeping if REQD = 1.

(3) The difference from ordinary sweeping mode described in **Section 32B.9.13.8(3)(b), Transfer suspension (buffer sweeping enabled: SBE = 1)** (EN is cleared by setting 1 for the SBE bit in the CHCFG_n register) is as follows: DMAC stops operation after writing data in the buffer in ordinary sweeping mode, whereas DMAC can continue DMA transfer after sweeping the buffer in forced sweeping mode.

#### (1) Forced software sweeping request

The SETSSWPRQ bit in the CHCTRL_n register determines whether software-based forced sweeping requests can be used. To perform a forced sweeping request, write 1 to the SETSSWPRQ bit. DMAC then outputs the data in the buffer to the destination.

### 32B.9.13.4 DMA transfer completion interrupt (USBFDMAmn)

USBFDMAmn (m, n = 0, 1) is an interrupt signal that indicates termination of a DMA transaction.

If a transfer for the total number of transfer bytes loaded to the CRTB_n register is completed by an OKAY response, the END bit in the CHSTAT_n register is set to 1. At this time, if the DEM bit in the CHCFG_n register is 0, DMAC generates a USBFDMAmn (m, n = 0, 1) interrupt.
(If a write-back operation is performed in link mode, the interrupt is generated after the write-back operation finishes.)

In link mode, when the DRRP bit in the CHCFG_n register is 0, if the LV bit of the header of the descriptor that is read is 0, the DER bit in the CHSTAT_n register is set to 1. At this time, if the DIM bit in the CHCFG_n register is 0, DMAC generates a USBFDMAmn interrupt.

Use this signal to detect a transfer completion interrupt performed by the interrupt controller.

Table 32B.52 USBFDMAmn Assertion Conditions

| Cause | Condition | INT_DMA[n] mask signal |
|---|---|---|
| DMA transaction ended | A transfer of data by the number of transfer bytes loaded to the CRTB_n register is completed by an OKAY response (if a write-back operation is performed in link mode, after the operation finishes) | DEM bit in the CHCFG_n register |
| Descriptor was invalid | LV of header of the descriptor that is read is 0 when DRRP and DIM in the CHCFG_n register are both 0 in link mode | DIM bit in the CHCFG_n register |

### 32B.9.13.5 DMA error interrupt (USBFDMAERRm)

If an error response is received for DMA transfer or descriptor access, this module stops transfer, assuming that an error occurred. When an error response is received, the EN bit in the CHSTAT_n register for channel n that is being used for transfer is cleared to 0, and the ER bit is set to 1 (n = 1, 0). Also, the USBFDMAERRm interrupt is asserted.

The USBFDMAERRm signal cannot be masked.

For a sequence of transfers for which an error occurred, data integrity cannot be guaranteed. Always use the following procedure to restart the transfer sequence from the beginning.

1. Set the SWRST bit in the CHCTRL_n register to 1.
2. Reset each register.

### 32B.9.13.6 Interval Count Function

The execution interval of a DMA transfer can be adjusted by using the ITVL field in the CHITVL_n register. This function prevents DMAC from continuously occupying the bus. If this function is enabled, DMAC does not perform a DMA transfer for the next request until the counter value becomes 0.

The following shows an operation example.

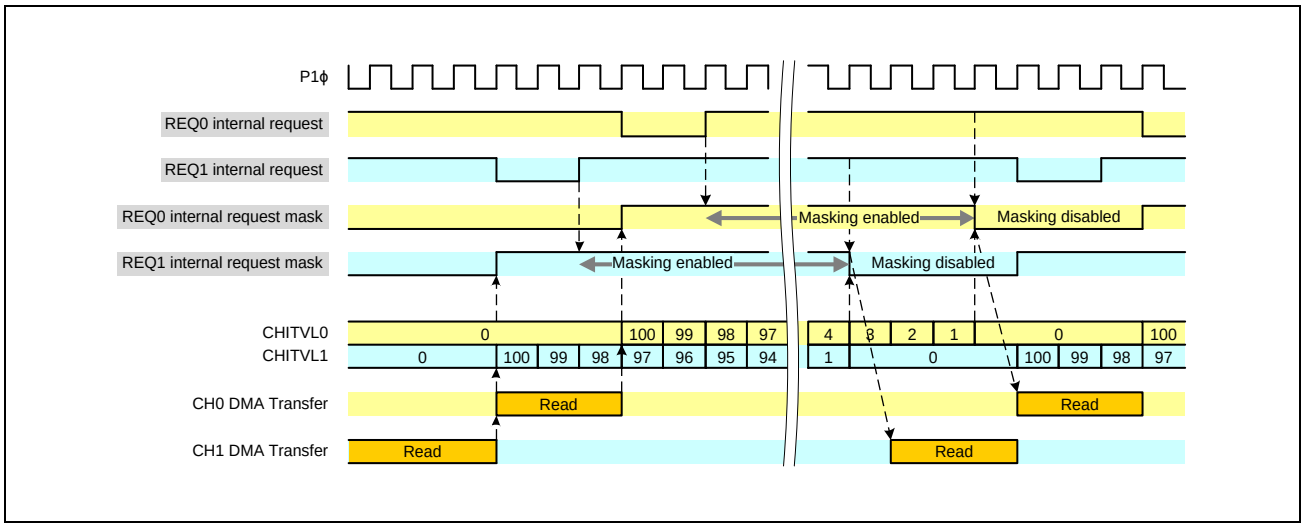



Figure 32B.25 Example of Counting Intervals (REQD = 0, SDS < DDS)

An interval is inserted after a transfer is performed on the side specified by the REQD bit in the CHCFG_n register. The following shows how the REQD, SDS, and DDS values of the CHCFG_n register are related with the interval.

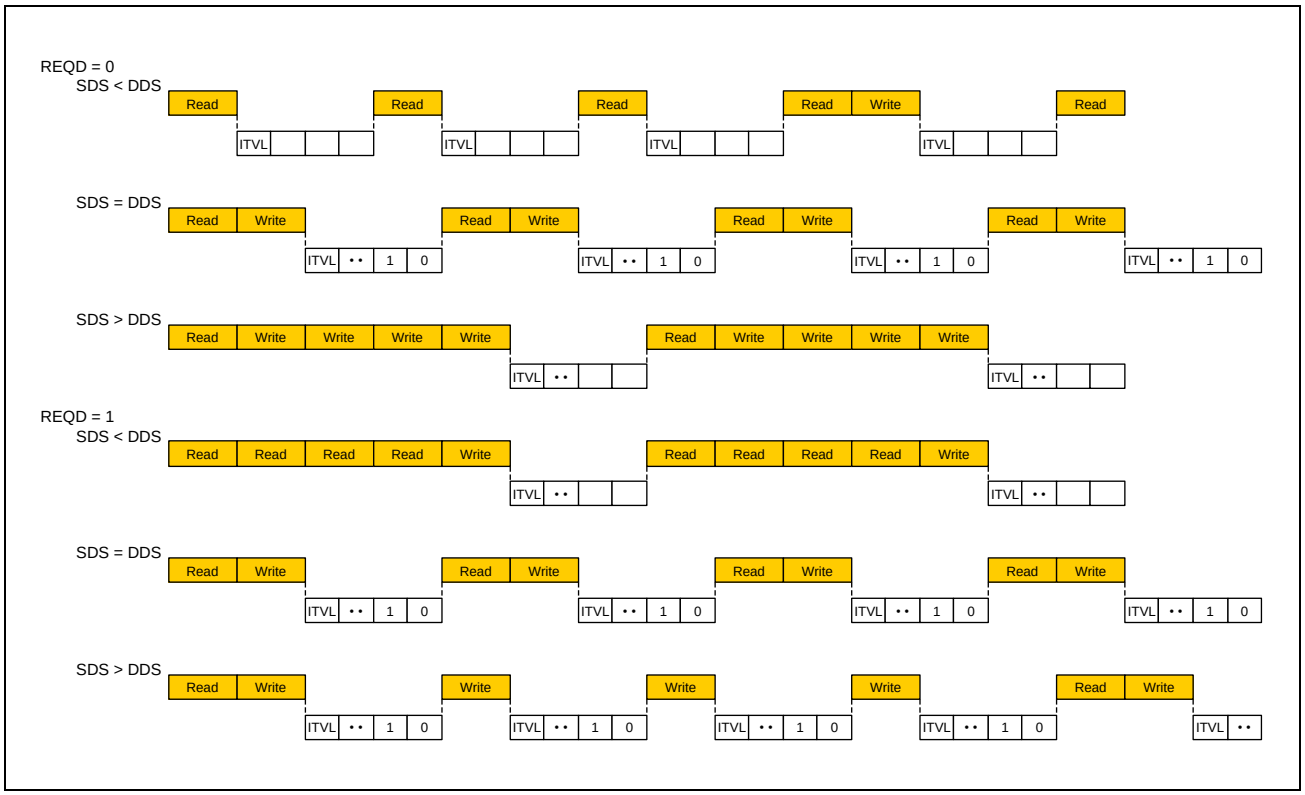



Figure 32B.26 DMA Transfer Settings and Interval Count

### 32B.9.13.7 Operational difference depending on the transfer size

#### (1) If the transfer size on the source side is smaller

When reading of as much data as the destination data size finishes, a write to the destination starts.

The following figure is an example of the timing chart in the case where the source is an 8-bit field and the destination is a 32-bit field (SDS = 0 and DDS = 2 in the CHCFG_n register) (when the rising edge is detected).

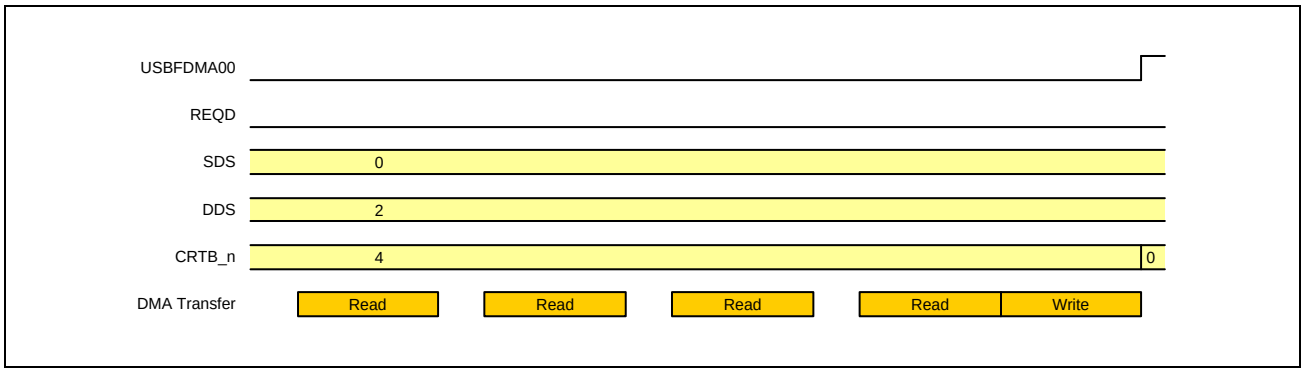



Figure 32B.27 Example of Timing Chart in the Case Where the Source is Smaller
(LVL = 0, HIEN = 1, REQD = 0, and SDS < DDS)

#### (2) If the transfer size on the destination side is smaller

Because the source side is larger than the destination side, two or more destination write operations occur for one source read operation. The following is an example of the timing chart in the case where the source is a 64-bit field and the destination is a 16-bit field (SDS = 3 and DDS = 1 in the CHCFG_n register) (when the rising edge is detected).

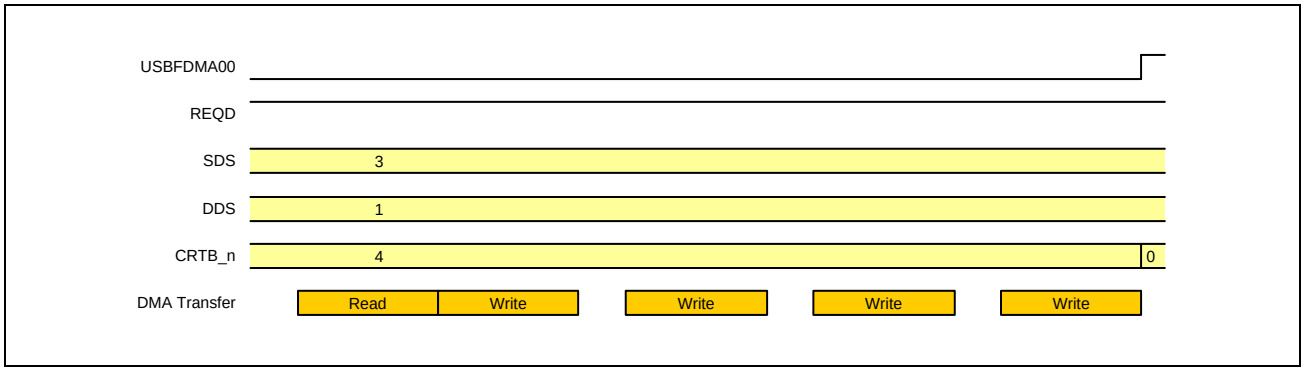



Figure 32B.28 Example of Timing Chart in the Case Where the Destination Is Smaller
(LVL = 0, HIEN = 1, REQD = 1, and SDS > DDS in the CHCFG_n Register)

### (3) If the Source and Destination Transfer Sizes Are the Same

Each time a DMA transfer request is detected, a source read operation and a destination write operation occur.

The following is an example of the timing chart in the case where the source and destination are 8-bit fields (SDS = 0 and DDS = 0 in the CHCFG_n register) (when the rising edge is detected).

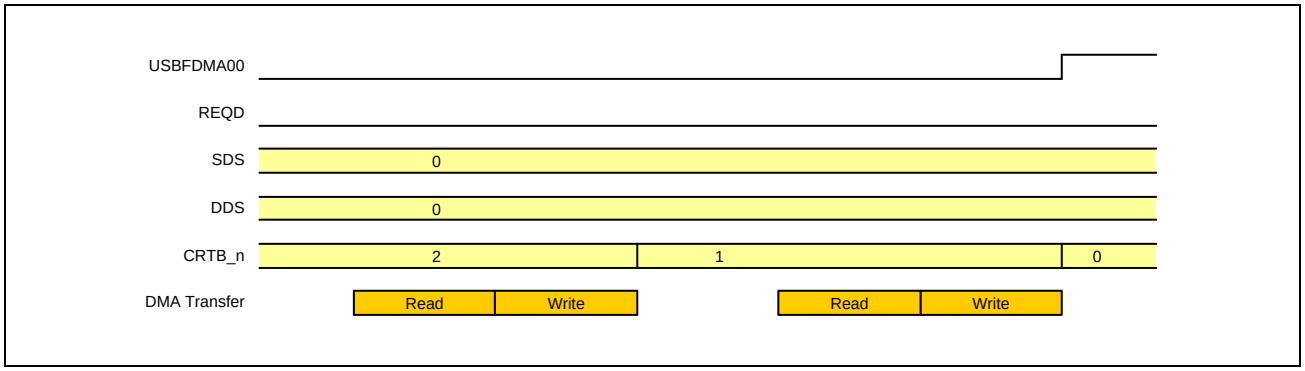



Figure 32B.29 Example of Timing Chart in the Case where the Source and Destination Sizes are the Same (LVL = 0, HIEN = 1, REQD = 0, and SDS = DDS in the CHCFG_n Register)

### 32B.9.13.8 Transfer state

The CHSTAT_n register indicates the transfer state of a channel.

#### (1) Transfer state

The TACT bit in the CHSTAT_n register indicates that channel n is operating. When 1 is written to the SETEN bit in the CHCTRL_n register, the TACT bit is set to 1. The TACT bit continues to be 1 while DMAC is accessing a descriptor or waiting for a DMA request.

The TACT bit is cleared when the EN bit in the CHSTAT_n register is cleared (for details about the conditions in which the EN bit is cleared, see **Section 32B.4.1, Current Source Address Register**) and the DMA transfer ends.

If the EN bit is not cleared when the DMA transaction ends (for example, when the REN bit of the CHCFG_n register is 1 in register mode or when DMAC accesses the next descriptor in link mode), the TACT bit is not cleared.

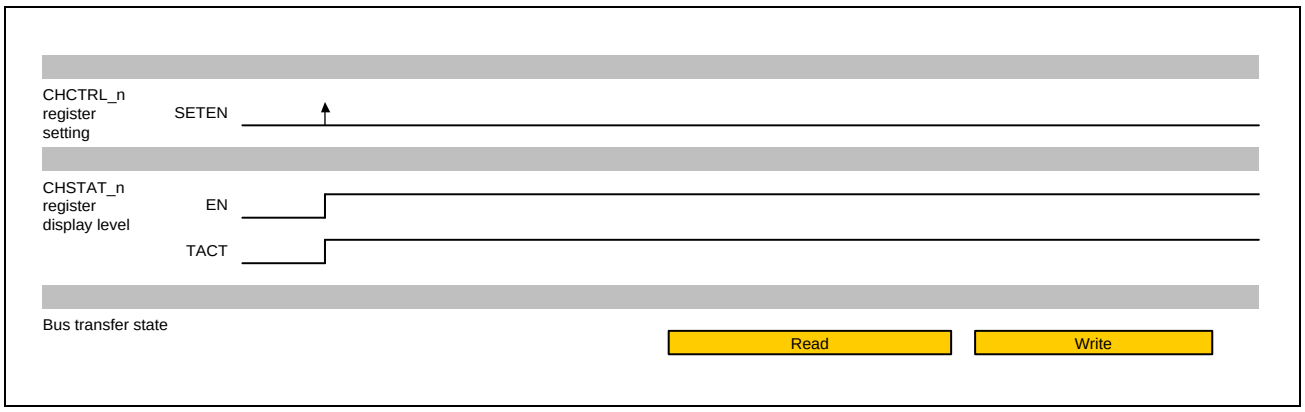



Figure 32B.30 DMAC State Example 1 (Hardware Request)

#### (2) Suspension

The SETSUS bit in the CHCTRL_n register can be used to suspend a DMA transfer. If suspension of a DMA transfer is attempted when a bus cycle is running, the DMA transfer is suspended after the bus cycle finishes. The suspended transfer can be resumed by writing 1 to the CLRSUS bit in the CHCTRL_n register.

To check whether a DMA transfer is suspended, after setting the SETSUS bit in the CHCTRL_n register, check the SUS bit in the CHSTAT_n register or the SUS bit for the relevant channel in the DSTAT_SUS register. If The SUS bit is 1, the DMA transfer is currently suspended.

### (3) Transfer suspension

By writing 1 to the CLREN bit in the CHCTRL_n register during a DMA transaction, the DMA transaction of the channel can be suspended. As the post-processing for suspension, the SBE bit in the CHCFG_n register can be used to select whether to sweep the data remaining in the buffer when the transaction is suspended. The default is SBE = 0 (do not sweep remaining data).

If the sweep mode is enabled and a transfer is suspended by setting 1 for the CLREN bit in the CHCTRL_n register, DMAC sweeps any data remaining in the buffer and stops operation.

#### (a) Transfer suspension (buffer sweeping disabled: SBE = 0)

If 1 is written to the CLREN bit in the CHCTRL_n register during DMA transfer, DMAC suspends DMA transfer and then stops. The timing of stoppage depends on the value set for the REQD bit. After DMAC stops, write 1 to the SWRST bit in the CHCTRL_n register to clear the internal status of DMAC, and then specify the next transfer settings.

Complete deactivation of the channel can be confirmed when the value of the TACT bit in the CHSTAT_n register changes from 1 to 0.

If DMA transfer is suspended before it is completed, the USBFDMAmn interrupt is not asserted.

If the REQD bit in the CHCFG_n register is 0, DMAC stops when the next read operation is completed. However, if data that can be written exists in the buffer, DMAC writes the data and then stops.

If the REQD bit in the CHCFG_n register is 1, DMAC stops when the next read operation is completed.

#### (b) Transfer suspension (buffer sweeping enabled: SBE = 1)

If 1 is written to the CLREN bit in the CHCTRL_n register during DMA transfer, DMAC suspends DMA transfer. If the REQD bit in the CHCFG_n register is 0, DMAC sweeps (writes) the already read data, and then stops DMA transfer. If the REQD bit is 1, sweep mode cannot be used physically.

After DMAC stops, set the SWRST bit in the CHCTRL_n register to clear the internal status of DMAC, and then specify the next transfer settings.

Complete deactivation of the channel can be confirmed when the value of the TACT bit in the CHSTAT_n register changes from 1 to 0.

#### (c) How to confirm deactivation of the channel

Even when the EN bit of the CHSTAT_n register is cleared to 0 by writing 1 to the CLREN bit in the CHCTRL_n register, if a transfer has already been executed over the bus, DMAC cannot immediately stop. To check whether DMAC has stopped completely, check the EN and TACT bits in the CHSTAT_n register. If both bits are 0, DMAC has stopped completely.

### (d) Procedure for suspending transfer

To suspend transfer:

1. Write 1 to the CLREN bit in the CHCTRL_n register.
2. If the SBE bit in the CHCFG_n register is 0, DMAC stops according to the value of the REQD bit in the CHCFG_n register. If the SBE bit is 1, DMAC is placed in sweep mode.
3. Read the CHSTAT_n register to check whether the TACT bit is 0. If the TACT bit is 0, DMAC has stopped completely. If the TACT bit is 1, continue polling until the bit changes to 0.
4. To perform the next DMA transfer after it is suspended, make sure that the SWRST (software reset) bit in the CHCTRL_n register is turned on before the next DMA transfer starts.

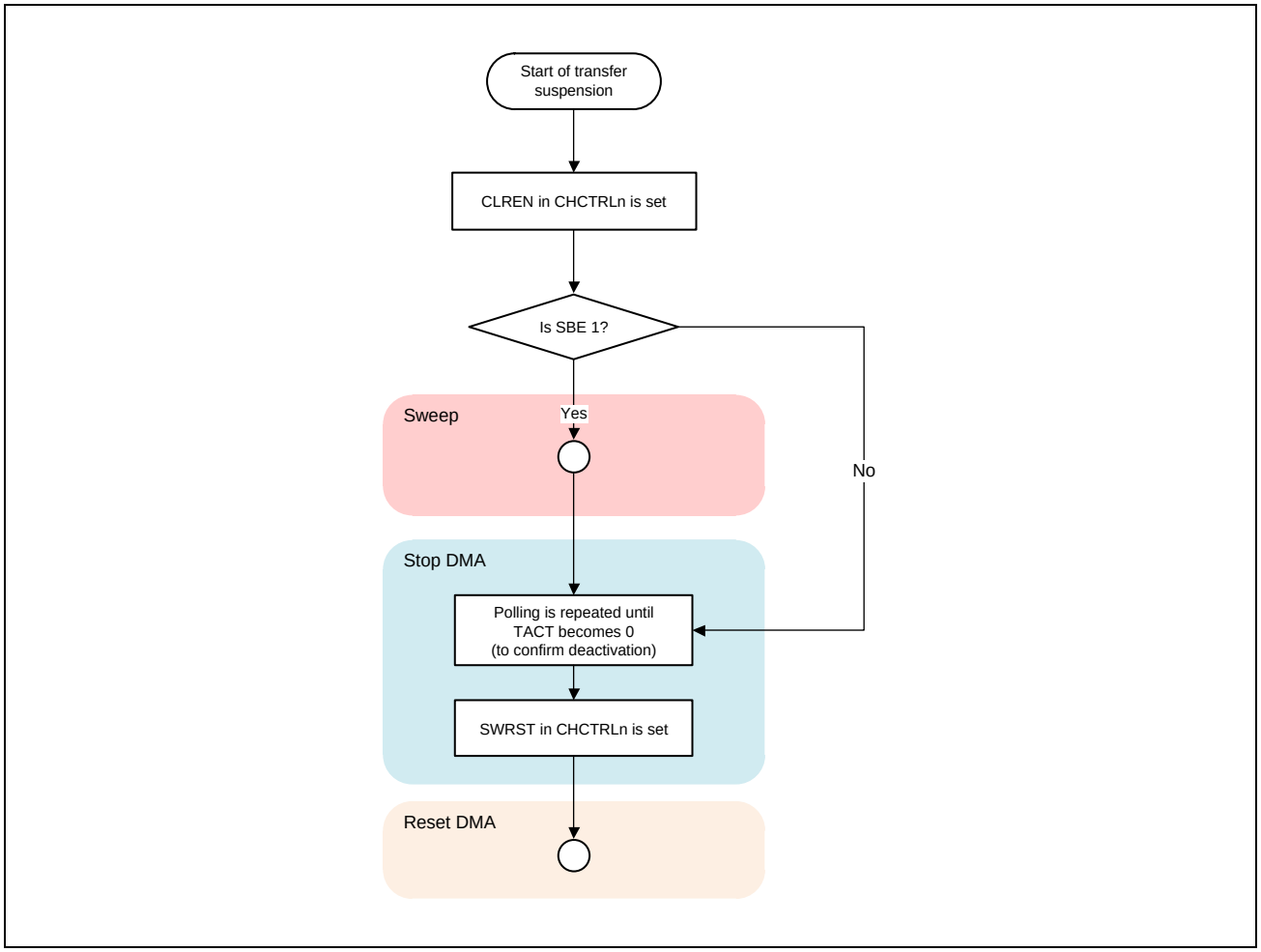



Figure 32B.31 Transfer Suspension Flowchart

## 32B.9.14 Access Type

### 32B.9.14.1 DMA master transfer combination list

#### (1) Read access

The following describes characteristics of the issuance type for DMA read access.

- An access is performed for a beat align space whose size is set by SDS[2:0] in the CHCFG_n register, including the source address indicated in the CRSA_n register. A beat unaligned transfer for the bus is not performed. An excess area is sometimes read depending on CRSA_n or SKIP settings. In this case, the necessary data is imported into the buffer from the read data.
- The size and burst length are determined based on the value set in the SDS[2:0] field.
  If the value set in the SDS[2:0] field is equal to or less than the bus width
  Size: Value set in SDS[2:0]
  Burst type: SINGLE
  If the value set in the SDS[2:0] field is larger than the bus width
  Size: Bus width
  Burst type: Fixed-length burst (burst length = SDS[2:0] value/bus width)

The following indicates the access types for the bus.

Table 32B.53 DMA Read Transfer Combination List

| | | AHB transfer | | | | | |
|---|---|---|---|---|---|---|---|
| | | First transfer | | | Second transfer | | |
| SDS | Source Address | Address | Data Size | Burst Type | Address | Data Size | Burst Type |
| 0 (8 bits) | - | addr | 8 | SINGLE | | | |
| 1 (16 bits) | 2 byte align | {addr[31:1], 0b} | 16 | SINGLE | | | |
| | 2 byte unalign | | | | {addr[31:1], 0b} + H'2 | 16 | SINGLE |
| 2 (32 bits) | 4 byte align | {addr[31:2], 00b} | 32 | SINGLE | | | |
| | 4 byte unalign | | | | {addr[31:2], 00b} + H'4 | 32 | SINGLE |
| 4 (128 bits) | 16 byte align | {addr[31:4], H'0} | 32 | INCR4 | | | |
| | 16 byte unalign | | | | {addr[31:4], H'0} + H'10 | 32 | INCR4 |
| 5 (256 bits) | 32 byte align | {addr[31:5], H'00} | 32 | INCR8 | | | |
| | 32 byte unalign | | | | {addr[31:5], H'00} + H'20 | 32 | INCR8 |
| 6 (512 bits) | 64 byte align | {addr[31:6], H'00} | 32 | INCR16 | | | |
| | 64 byte unalign | | | | {addr[31:6], H'00} + H'40 | 32 | INCR16 |

**Note:** If EBT is detected in the middle of burst, 32-bit INCR burst is used to transfer the remaining data.

### (2) Write access

The following describes characteristics of the issuance type for DMA write access.

- An access is performed from the destination address indicated by the CRDA_n register to the beat align boundary whose size is set by DDS[2:0] in the CHCFG_n register.
- The size and burst length are determined based on the value set in the DDS[2:0] field.
  If the value set in the DDS[2:0] field is equal to or less than the bus width
  Size: Value set in DDS[2:0]
  Burst type: SINGLE
  If the value set in the DDS[2:0] field is greater than the bus width
  Size: Bus width
  Burst type: Fixed-length burst (burst length = DDS[2:0] value/bus width)
- In write access, only the specified space is accessed. In the following cases, the combination of values smaller than the value set in the DDS[2:0] field is used for access.
  - The destination address is beat-unaligned for the value set in the DDS[2:0] field.
  - An access specified in the DDS field will be across the SKIP boundary.
  - The size specified in the DDS[2:0] field is too large for the number of remaining bytes to be transferred.

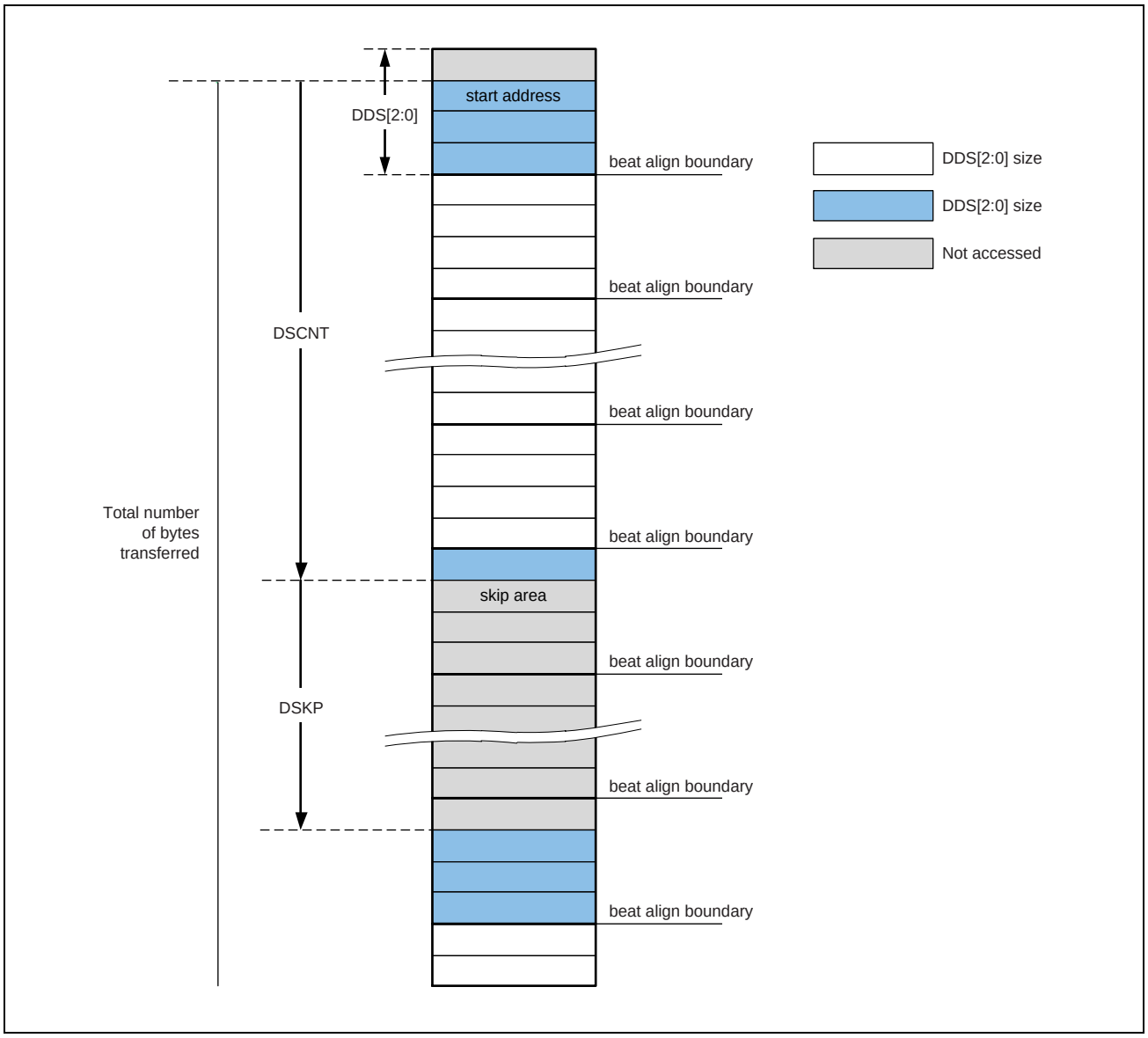



Figure 32B.32 Example of DMA Write Access Space and Access Types

The following shows the access types for the bus in the case of beat-aligned.

Table 32B.54 DMA Write Transfer Combination List

| | AHB transfer | | |
|---|---|---|---|
| | First transfer | | |
| DDS[2:0] | Address | Data Size | Burst Type |
| 0 (8 bits) | addr | 8 | SINGLE |
| 1 (16 bits) | {addr[31:1], 0b} | 16 | SINGLE |
| 2 (32 bits) | {addr[31:2], 00b} | 32 | SINGLE |
| 4 (128 bits) | {addr[31:4], H'0} | 32 | INCR4 |
| 5 (256 bits) | {addr[31:5], H'00} | 32 | INCR8 |
| 6 (512 bits) | {addr[31:6], H'00} | 32 | INCR16 |

### 32B.9.14.2 DMA master descriptor combination list

#### (1) Read access

The following describes characteristics of a descriptor access.

- Accesses are performed for 8-word beat align space including the LINK address (the address indicated in the CRLA_n register). A beat unaligned transfer is not performed.
- The size and burst length set by INCR8 are used.
- The descriptor format (DSCFM) of the read header is analyzed, and then the descriptor data is set in an internal register.
- If the descriptor extends over the 8-word boundary, an additional read is performed on the succeeding eight words.

The following indicates the types of access to the bus.

Table 32B.55 Descriptor Read Transfer Combination List

| Descriptor Format | Address | AHB Transfer | | | | | |
|---|---|---|---|---|---|---|---|
| | | First Transfer | | | Second Transfer | | |
| | addr[4:0] | Address | Size | Burst | Address | Size | Burst |
| 4 words | H'00 | {addr[31:4], 0000b} | 32 | INCR8 | | | |
| | H'04 | | | | | | |
| | H'08 | | | | | | |
| | H'0C | | | | | | |
| | H'10 | | | | | | |
| | H'14 | | | | {addr[31:4], 0000b} + H'20 | 32 | INCR8 |
| | H'18 | | | | | | |
| | H'1C | | | | | | |
| 8 words | H'00 | | | | | | |
| | H'04 | | | | {addr[31:4], 0000b} + H'20 | 32 | INCR8 |
| | H'08 | | | | | | |
| | H'0C | | | | | | |
| | H'10 | | | | | | |
| | H'14 | | | | | | |
| | H'18 | | | | | | |
| | H'1C | | | | | | |

**Note:** If EBT is detected in the middle of burst, 32-bit INCR burst is used to transfer the remaining data.

#### (2) Write access

Use single transfer as the issuance type for writing data back to a descriptor. The following indicates the types of access to the bus.

Table 32B.56 Descriptor Write Transfer Combination List

| Type | AHB transfer | | |
|---|---|---|---|
| | Address | Size | Burst |
| Write-back in normal mode | {addr[31:2], 00b} + H'3 | 8 | SINGLE |
| Write-back in the case of an error | {addr[31:2], 00b} + H'2 | 8 | SINGLE |

### 32B.9.15 Arbitration between DMACs

Arbitration between internal DMACs is performed in round-robin mode.

In round-robin mode, the highest priority is given to the DMAC whose DMAC number is the DMC number being used for transfer + 1. Immediately after a reset, DMAC0 has the highest priority.

Table 32B.57 Priority of a transfer request for DMACs that are performing transfer

| | Next DMAC | |
|---|---|---|
| Current DMAC | DMAC0 | DMAC1 |
| DMAC0 | 2 | 1 |
| DMAC1 | 1 | 2 |

**Note:** Priority: 1 (high), 2 (low)

## 32B.9.16 Notes

### 32B.9.16.1 Access

During read access, a beat align area is accessed by one transfer. Therefore, if beat unaligned is set, the beat align area including the specified area is accessed.

For example, if the Source Address is H’0000_1038 and SDS is 5 (256-bit), read starts from area H’0000_1020, not from address H’0000_1038. At this time, if the area from H’0000_1020 to H’0000_1037 contains a register whose value changes by read, the operation might be disrupted.

To prevent problems, use the beat align setting to access a register whose value changes by read access or access to an area adjacent to FIFO.

### 32B.9.16.2 Level Interrupt bit

This is a DMA interrupt output enable bit. Always set this bit to 1 irrespective of whether peripheral module interrupt USBFDMAmn (m, n = 0, 1) or USBFDMAERRm (m = 0, 1) is used.

To use peripheral module interrupt USBFDMAmn or USBFDMAERRm, set this bit to 1, and then set up the interrupt controller.

# 33. LCDC

## 33.1 Overview

This chapter describes the features of the LCDC unit of this LSI.

This unit is LCD controller, it is composed of Frame Compression Processor (FCPVD), Video Signal Processor (VSPD), and Display Unit (DU)

### 33.1.1 Features

The following is key features of this unit.

- FCPVD
  - Support out-of-order for the whole outstanding transactions
  - Read linear addressing image data
  - Read display list data
  - Write image data
- VSPD
  - Supports various data formats and conversion
    - Supports YCbCr444/422/420, RGB, α RGB, α plane
    - Color space conversion and changes to the number of colors by dithering
    - Color keying
    - Supports combination between pixel alpha and global alpha
    - Supports generating pre multiplied alpha
  - Video processing
    - Blending of two picture layers and raster operations (ROPs)
    - Clipping
    - 1D look up table
    - Vertical flipping in case of output to memory
  - Direct connection to display module
    - Supporting 1280 pixels in horizontal direction
    - Writing back image data which is transferred to Display Unit (DU) to memory
- DU
  - Supporting Display Parallel Interface (DPI) and MIPI LINK Video Interface
  - Display timing master
    - Generating video timings (Front porch, Back porch, Sync active, Active video area)
    - Selecting the polarity of output DCLK, HSYNC, VSYNC, and DE
  - Supporting Progressive (Non-interlace)
  - Not supports Interlace

- Input data format (from VSPD): RGB888, RGB666 (not supports dithering of RGB565)
- Output data format: same as Input data format
- Supporting WXGA (1280 pixels × 800 lines ) for MIPI-DSI Output
- Supporting WXGA (1280 pixels × 800 lines) for Parallel Output

### 33.1.2 Block Diagram

**Figure 33.1** shows a block diagram.

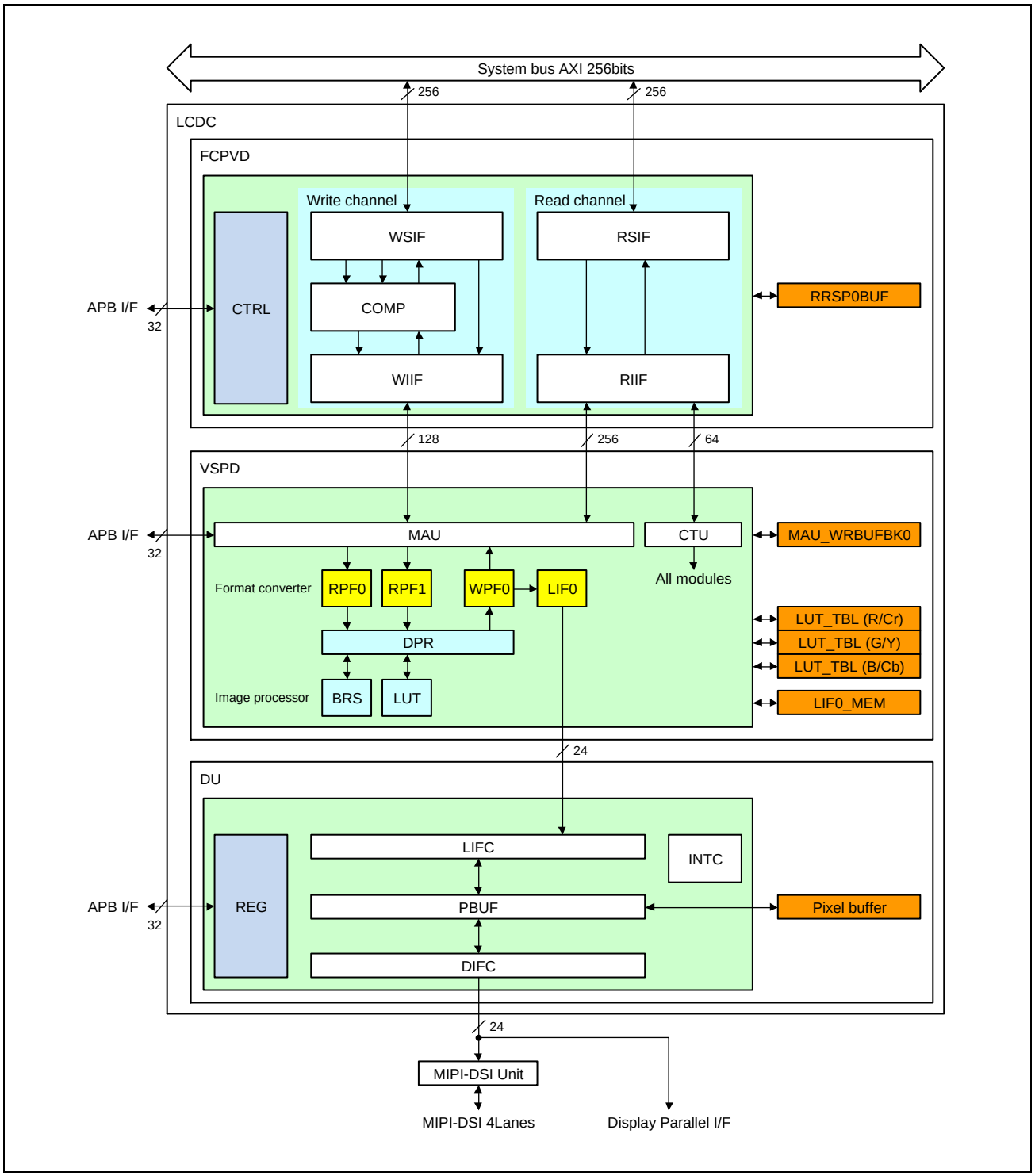



Figure 33.1 Block Diagram

#### 33.1.2.1 FCPVD

FCPVD has the following sub modules.

Table 33.1 Sub modules in FCPVD

| Abbreviation | Description |
|---|---|
| CTRL | FCPVD controller |
| WIIF | VSPD write channel interface |
| WSIF | System AXI write channel interface |
| RIIF | VSPD read channel interface |
| RSIF | System AXI read channel interface |
| COMP | Compression module (This is a dummy module) |

#### 33.1.2.2 VSPD

VSPD has the following sub modules.

##### (1) Memory Access Unit (MAU)

The VSPD applies processing to the image data stored in the external memory and writes the resultant data back to the external memory. The data transfer between the external memory and VSPD necessary for this operation is done by the MAU, which works as the bus master, according to the register settings. The MAU executes this data transfer between the external memory and VSPD.

##### (2) Command Transfer Unit (CTU)

The VSPD can directly read register parameters for image processing by display lists stored in external memory. The CTU module is a bus interface and controls display lists when the CTU reads display lists as a bus master.

### (3) Read Pixel Formatter (RPF)

The RPF reads image data from the external memory through the MAU, unpacks data according to the specified format, converts the color space, converts the number of colors, executes color keying, ROP operation, Multiply-alpha and OSD processing, and outputs the resultant data to the DPR. The RPF has an input format unpacking unit, a 1-bit mask generator, a raster operation unit (ROP unit), a color keying unit, a color space converter and multiply-alpha. **Figure 33.2** show the processing flow in the RPF.

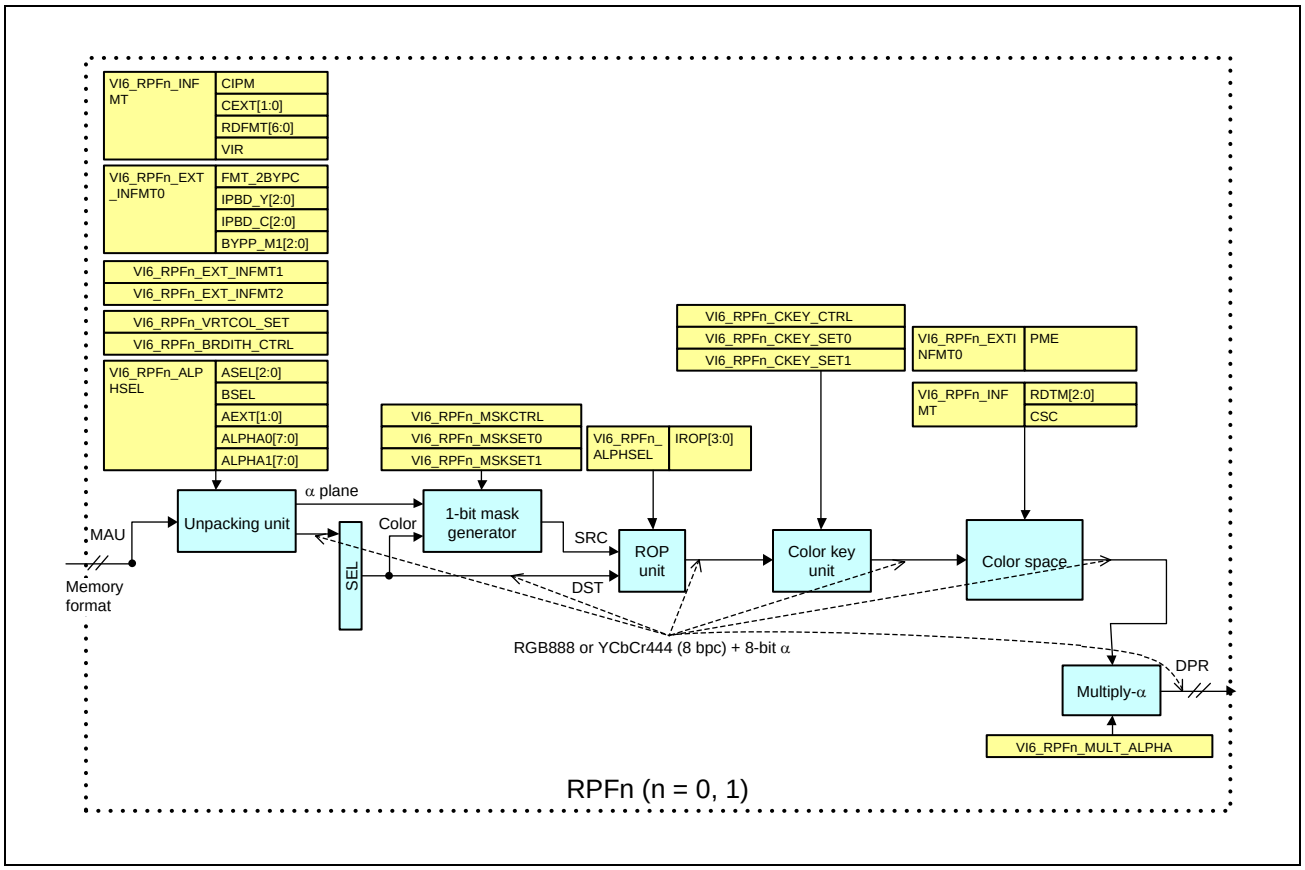



Figure 33.2 RPF Processing Flow

The input format unpacking unit expand the image data input from the MAU into the image format for internal processing (YCbCr444 8bpc or RGB888 8bpc), and the 1-bit mask generator generates a 1-bit image mask from the image data expanded through the unpacking unit. Alternately, a 1-bit image mask can be generated from the alpha plane that is different from the picture plane read through the MAU.

The raster operation unit (ROP unit) executes raster operation between the data from the 1-bit mask generator and the image data expanded from the input format, and the color keying unit applies color replacement and specifies the transparent color for the image data input from the ROP unit. The color space converter converts the color space (RGB-YCbCr) of the image data input from the color keying unit as necessary. The multiply-alpha unit multiplies pixel-alpha by specified alpha or/and multiplies image data by pixel-alpha or/and specified alpha.

The VSPD provides maximum two RPF modules (RPF0 to RPF1).

### (4) Data Path Router (DPR)

The DPR controls the data paths among RPFs, function modules, and WPFs. The DPR selects one of the images input from RPFs, outputs it to a function module (BRS or LUT), and selects one of WPFs as the destination where the image data processed in the function module will be output. Before output to the WPF, the output from each function module can be input to another function module, which enables multiple image processing functions to be executed continuously without involving the external memory.

### (5) Look Up Table (LUT)

This is a 1D-LUT that converts each of three color components by using a lookup table. The LUT is connected to the DPR and can be used for gamma correction, negative-positive conversion, posterization, and binarization through desired tone curve settings.

The LUT supports local area processing for Y component. The LUT can apply 1D look up table for Y component per area where an image is divided into multiple area specified by registers and command list.

### (6) Blend ROP Sub Unit (BRS)

The BRS is a module connected to the DPR, which executes the image blending processing and ROP operation. The BRS has two blend/ROP operation units (blend/ROP unit m, m = A to B), a blend/ROP input switch (SEL) for selecting the input to these operation units, and a divider for normalization (div unit).

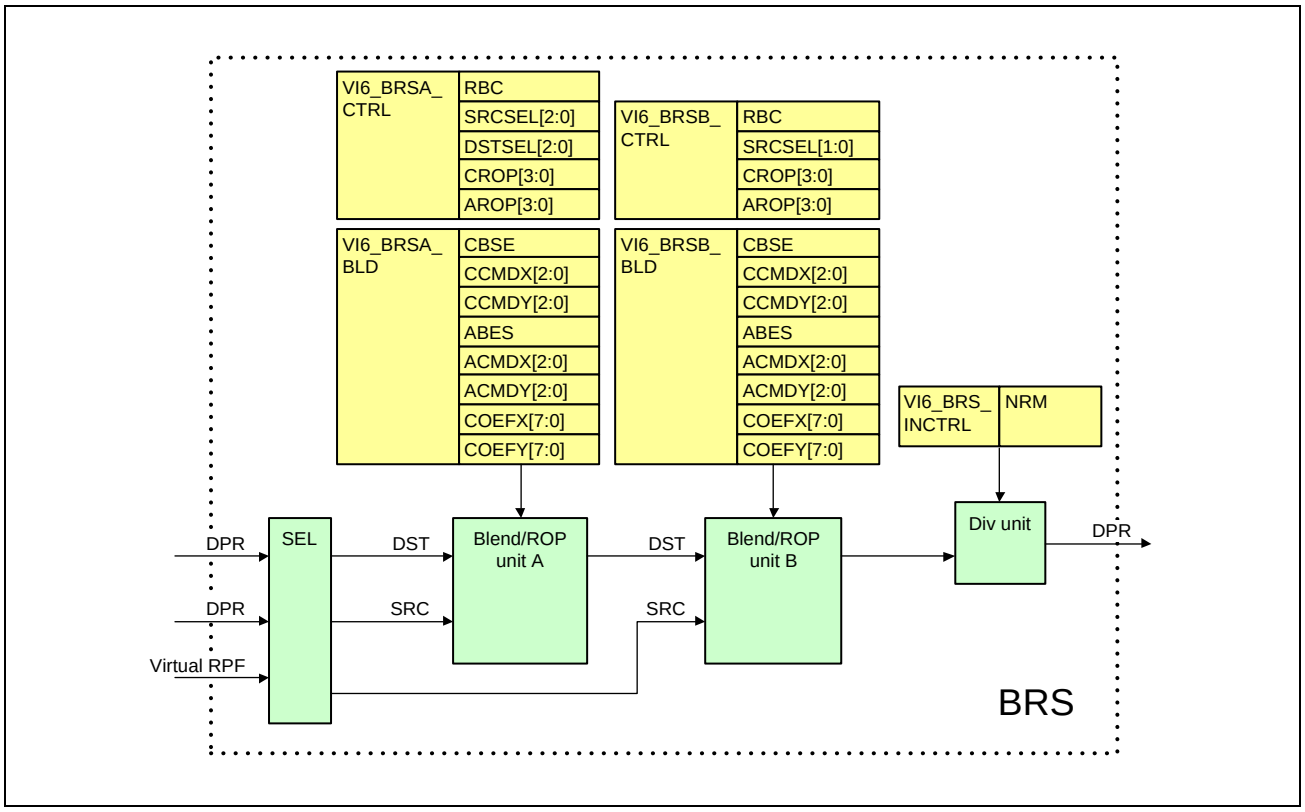



Figure 33.3 BRS Processing Flow

Each of the two blend-ROP operation units (blend/ROP unit m) receives the output from the SEL or blend/ROP Unit m, and executes blending or raster operation (ROP) of images.

The divider for normalization (div unit) divides the pixel value by the α value.

### (7) Write Pixel Formatter (WPF)

The WPF is an output module that receives 32bits image data (YCbCr444 or RGB888 + 8-bitα) from the DPR, converts the color space, number of colors, and format of the data, and outputs the results of VSPD image processing to external memory through the MAU. The WPF is mainly configured from a color space converter and an output format converter (the packing unit).

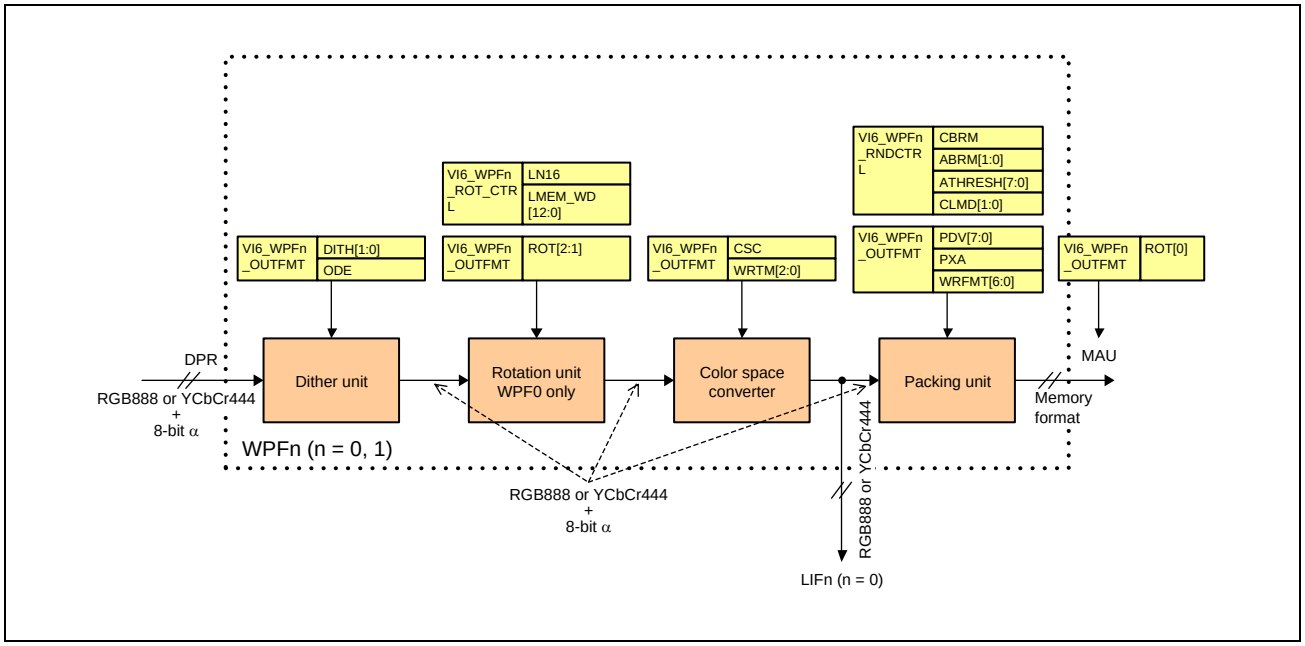



Figure 33.4 WPF Processing Flow

The color space converter converts the color space between RGB and YCbCr, and the packing unit converts the format into the picture plane storing format. The VSPD provides one WPF (WPF0). The WPF has the output to LIF module after the color space conversion for transferring the video data to the display module. Image data to LIF is 24bits (YCbCr444 or RGB888) format, and its' color space is after color space conversion.

### (8) Display Unit InterFace (LIF)

The LIF module is used for transferring image data to the display module. The input port of the LIFn (n = 0) module is connected to WPFn (n = 0), and the output port of the LIF module is connected to DU (Display Unit). Data flow in LIF is shown in **Figure 33.30**.

### (9) Detail Function

Table 33.2 Detail Function of VSPD (1/5)

| | | | |
|---|---|---|---|
| • Image Data Transfer Function | | | |
| Bus interface | Protocol | | AXI 256 bits through FCPVD |
| | Data alignment | Conversion method | Byte, Word, LW, or LLW data swapping |
| | | Channel | Data alignment can be specified separately for each input/output channel |
| Image memory | Input | Address setting | 1-byte units |
| | Output | Address setting | 1-byte units |
| | | Memory area | Images can be written to the same memory area where the master layer is stored.<br>Note the restrictions shown in **Table 33.38** |
| Tile transfer mode | | | Supported by RPF0 to RPF1 |
| YCbCr memory storage format | | | Interleaved, planar, or semi-planar |
| • Display list/Extended Display List Transfer Function | | | |
| Bus interface | Protocol | | AXI 256 bits through FCPVD |
| | Data alignment | Conversion method | Byte, Word, LW data swapping |
| | | | Data alignment can be specified separately for display list and each entry of extended display list |
| Data memory | Load | Address setting | 8-byte units |
| | Store | — | Not available |

Table 33.2 Detail Function of VSPD (2/5)

| | | | |
|---|---|---|---|
| • Read Pixel Formatter (RPF) | | | |
| Number of channels | | | Two channels (RPF0 to RPF1) |
| Input color bit-depth | | Color | 2 to 8bit |
| | | Alpha | 1/2/4/8 bit |
| Operation bit-depth | | Color | 8bit |
| | | Alpha | 8bit |
| Image format | Input | RGB | RGB888, RGB565, RGB666,<br>αRGB8888, αRGB4444, αRGB1555<br>α plane (8 bpp, 1 bpp) |
| | | YCbCr | YCbCr4:4:4 (8 bpc) Planar/Semi planar/Interleaved<br>YCbCr4:2:2 (8 bpc) Planar/Semi planar/Interleaved<br>YCbCr4:2:0 (8 bpc) Planar/Semi planar/Interleaved<br>α plane (8 bpp, 1 bpp) |
| | | Maximum size | 1280 × 800 pixels<br>The internal data path modules have separate restrictions on the maximum image size. For details, refer to **Section 33.5.3**. |
| | | Minimum size | 1 × 1 pixel<br>The internal data path modules have separate restrictions on the minimum image size. For details, refer to **Section 33.5.3**. |
| | | Size setting unit | YCbCr420: 2-pixel units both horizontally and vertically.<br>YCbCr422: 2-pixel units horizontally and 1-pixel units vertically.<br>In other formats, the size can be set in 1-pixel units. |
| Color keying | Color replacement | Compared data | RGB or Y (8 bpc) |
| | | Replaced data | αRGB or αYCbCr (8 bpc for R, G, B, Y, Cb, Cr, α) |
| | | Comparison Mode | Matched color mode |
| | | Input source | RPF0 to RPF1 |
| | Transparent color | Compared data | RGB or Y (8 bpc) |
| | | Replaced data | α 8bpp |
| | | Comparison Mode | Matched color mode, Luma threshold mode |
| | | Input source | RPF0 to RPF1 |
| Raster operation | ROP2 (within input channels) | Operator | 16 types (OpenGL2.0 is supported) |
| | | Sources of operation | ROP2 operation between the 1-bpp α plane and RGB/YCbCr data in RPF0 to RPF1.<br>Note that 1-bpp α is converted to αRGB or αYCbCr4:4:4. (8 bpc) |
| Color space conversion | RGB to YCbCr | Conversion expression | RGB (0, $2^n$-1) to BT601 ($2^{n-4}$, 235*$2^{n-8}$/240*$2^{n-8}$)<br>RGB (0, $2^n$-1) to BT709 ($2^{n-4}$, 235*$2^{n-8}$/240*$2^{n-8}$)<br>RGB (0, $2^n$-1) to BT601 (0, $2^n$-1)<br>RGB ($2^{n-4}$, 235*$2^{n-8}$) to BT709 ($2^{n-4}$, 235*$2^{n-8}$/240*$2^{n-8}$)<br>(n = 8) |
| | | Target of conversion | RPF0 to RPF1 |
| | YCbCr to RGB | Conversion expression | BT.601 ($2^{n-4}$, 235*$2^{n-8}$/240*$2^{n-8}$) to RGB (0, $2^n$-1)<br>BT.709 ($2^{n-4}$, 235*$2^{n-8}$/240*$2^{n-8}$) to RGB (0, $2^n$-1)<br>BT.601 (0, $2^n$-1) to RGB (0, $2^n$-1)<br>BT.709 ($2^{n-4}$, 235*$2^{n-8}$/240*$2^{n-8}$) to RGB ($2^{n-4}$, 235*$2^{n-8}$)<br>(n = 8) |
| | | Target of conversion | RPF0 to RPF1 |

Table 33.2 Detail Function of VSPD (3/5)

| | | | |
|---|---|---|---|
| • Read Pixel Formatter (RPF) | | | |
| Changing number of colors | Bit extension | Extended bits | Input bitdepth (2bpc to 6bpc) to operation bitdepth 8bpc |
| | | Target format | RGB666, RGB565, RGB555, RGB444, or RGB332 |
| | | Bit reduction method | Padded with 0.<br>Copied from the most significant bits. |
| | | Target of conversion | RPF0 to RPF1 |
| | YCbCr444 generation | Vertical | CbCr copying |
| | | Horizontal | copying or interpolation. |
| α bit count conversion | Bit extension | Extended bits | Input bitdepth 1bpc to operation bitdepth 8bpc |
| | | Method | Padded with 0.<br>Copied from the most significant bits. |
| Multiply-alpha function | Fade-alpha | | RPF0 to RPF1<br>Available for straight pixel and pre-multiplied pixel. |
| | Generate pre-multiplied alpha | | RPF0 to RPF1 |
| Virtual display | RPF0 to RPF1 | Color format | αRGB8888 or αYCbCr4:4:4 single-color |
| | | Display size | Same as the size of the input channel |
| | Virtual RPF | Color format | αRGB8888 or αYCbCr4:4:4 single-color |
| | | Display size | Maximum: 1280 ×800 pixels<br>Minimum: 4 × 4 pixels |

Table 33.2 Detail Function of VSPD (4/5)

| | | | |
|---|---|---|---|
| • Write Pixel Formatter (WPF) | | | |
| Number of channels | | | One channel (WPF0) |
| Image format | Output | RGB | RGB332, RGB444, RGB565, RGB666, RGB888, αRGB8666, αRGB8888,<br>αRGB4444, αRGB1555 |
| | | YCbCr | YCbCr4:4:4, YCbCr4:2:2, YCbCr4:2:0 |
| | | Maximum size | 1280 × 800 pixels<br>The internal data path modules have separate restrictions on the maximum image size. For details, refer to **Section 33.5.4**. |
| | | Minimum size | 1 × 1 pixel<br>The internal data path modules have separate restrictions on the minimum image size. For details, refer to **Section 33.5.4**. |
| | | Size setting unit | YCbCr420: 2-pixel units both horizontally and vertically.<br>YCbCr422: 2-pixel units horizontally and 1-pixel units vertically.<br>In other formats, the size can be set in 1-pixel units. |
| Color space conversion | RGB to YCbCr | Conversion expression | RGB (0, 255) to BT.601 (16, 235/240)<br>RGB (0, 255) to BT.709 (16, 235/240)<br>RGB (0, 255) to BT.601 (0, 255)<br>RGB (16, 235) to BT.709 (16, 235/240) |
| | | Target of conversion | WPF0 |
| | YCbCr to RGB | Conversion expression | BT.601 (16, 235/240) to RGB (0, 255)<br>BT.709 (16, 235/240) to RGB (0, 255)<br>BT.601 (0, 255) to RGB (0, 255)<br>BT.709 (16, 235/240) to RGB (16, 235) |
| | | Target of conversion | WPF0 |
| Changing number of colors | Output | Reducing RGB color depth | Dithering, lower-order bit truncation, or rounding |
| | | YCbCr422/420 | CbCr skipping or CbCr vertical skipping and horizontal skipping |
| α bit count conversion | Output | Bit reduction | Truncation, rounding, or comparison with threshold (for 1 bpp) |

Table 33.2 Detail Function of VSPD (5/5)

| | | | |
|---|---|---|---|
| • Image Compositing | | | |
| α blending | Input α value selection | RGB | Pixel α, fixed α value, α plane, or<br>1-bit α converted from the color specified for pixels |
| | | YCbCr | α plane, fixed α value, or<br>1-bit α converted from the color specified for pixels |
| | α blending expression | Layer A: Upper layer<br>Layer B: Lower layer | xA + yB, xA - yB<br>Coefficients x and y should be selected from the following.<br>Fixed α value,<br>(α for layer A), (1 - α for layer A),<br>(α for layer B), (1 - α for layer B) |
| | Output α value selection | RGB | Fixed α value,<br>x (α for layer A) + y (α for layer B),<br>x (α for layer A) – y (α for layer B)<br>Coefficients x and y should be selected from the following.<br>Fixed α value,<br>(α for layer A), (1 - α for layer A),<br>(α for layer B), (1 - α for layer B) |
| | Blending layers | Number of layers | Two layers selected from RPF0 to RPF1 and video processing function output, and virtual RPF; three layers in total |
| | | Order of layers | The order of three layers selected from RPF0 to RPF1, virtual RPF, and video processing function output can be changed as desired. |
| | α plane | Format | 8 bpp or 1 bpp (α value can be specified through register) |
| | | Input source | RPF0 to RPF1 |
| | Fixed α value | Format | 8 bpp |
| | | Input source | RPF0 to RPF1, virtual RPF, or video processing function output |
| Raster operation | ROP2 (between input channels) | Operator | 16 types (OpenGL2.0 is supported) |
| | | Sources of operation | RPF0 to RPF1, virtual RPF, and video processing function output |
| | | Operation control | RGB/YCbCr and α are operated separately. |
| | ROP3 (between input channels) | Operator | 256-type ROP3 is available by combining ROP2 operations |
| | | Sources of operation | RPF0 to RPF1, virtual RPF, and video processing function output |
| | | Operation control | RGB/YCbCr and α are operated separately. |
| • Color Adjustment Function | | | |
| 1D-LUT | LUT configuration | Independent R/Y, G/Cb, and B/Cr. 256 entries each | |
| Direct connection to Display module | | | |
| Display I/F | Destination | | DU |
| | Direct connection | | Can transfer data without going through external memory to DU |
| Line padding | Padding cycles | | 1 to 32 cycles |
| | Padding pattern | | Arbitrarily setting up to 32 cycles |

#### 33.1.2.3 DU

DU has the following sub modules.

##### (1) REG

The REG controls registers for DU. The DU has APB4 interface, its transaction is transfer with no wait states.

##### (2) LIFC

The LIFC is the interface block from/to VSPD. The LIFC receives image data from VSPD, sends to PBUF in a pixel unit. The LIFC continues to receive the image data from VSPD unless FIFO will be full.

##### (3) PBUF

The PBUF is the asynchronous FIFO which convert clock domain of image data from System Bus Clock to Video Clock. The PBUF watches over FIFO underflow.

##### (4) DIFC

The DIFC is the display timing master. The DIFC controls the polarity and the timing of the control signals (DISP_CLK, DISP_HSYNC, DISP_VSYNC, and DISP_DE). Controllable timings are the followings.

- Hfront: Horizontal front porch (pixels)
- Hback: Horizontal back porch (pixels)
- Hsync: Horizontal sync active (pixels)
- Hactive: Horizontal active area (pixels)
- Vfront: Vertical front porch (lines)
- Vback: Vertical back porch (lines)
- Vsync: Vertical sync active (lines)
- Vactive: Vertical active area (lines)

##### (5) INTC

The INTC generates the interrupt when PBUF will detect FIFO underflow. The interrupt is level signal.

### 33.1.3 External Pins

**Table 33.3** shows the pin configuration.

Table 33.3 LCDC Display Parallel Interface (DPI) External Pins

| Pin Name | Input/Output | Function |
|---|---|---|
| DISP_CLK | Output | Display Parallel Interface pixel clock |
| DISP_HSYNC | Output | Display Parallel Interface Horizontal sync pulse |
| DISP_VSYNC | Output | Display Parallel Interface Vertical sync pulse |
| DISP_DE | Output | Display Parallel Interface data enable |
| DISP_DATA0 | Output | Display Parallel Interface pixel data output Red0 |
| DISP_DATA1 | Output | Display Parallel Interface pixel data output Red1 |
| DISP_DATA2 | Output | Display Parallel Interface pixel data output Red2 |
| DISP_DATA3 | Output | Display Parallel Interface pixel data output Red3 |
| DISP_DATA4 | Output | Display Parallel Interface pixel data output Red4 |
| DISP_DATA5 | Output | Display Parallel Interface pixel data output Red5 |
| DISP_DATA6 | Output | Display Parallel Interface pixel data output Red6 |
| DISP_DATA7 | Output | Display Parallel Interface pixel data output Red7 |
| DISP_DATA8 | Output | Display Parallel Interface pixel data output Green0 |
| DISP_DATA9 | Output | Display Parallel Interface pixel data output Green1 |
| DISP_DATA10 | Output | Display Parallel Interface pixel data output Green2 |
| DISP_DATA11 | Output | Display Parallel Interface pixel data output Green3 |
| DISP_DATA12 | Output | Display Parallel Interface pixel data output Green4 |
| DISP_DATA13 | Output | Display Parallel Interface pixel data output Green5 |
| DISP_DATA14 | Output | Display Parallel Interface pixel data output Green6 |
| DISP_DATA15 | Output | Display Parallel Interface pixel data output Green7 |
| DISP_DATA16 | Output | Display Parallel Interface pixel data output Blue0 |
| DISP_DATA17 | Output | Display Parallel Interface pixel data output Blue1 |
| DISP_DATA18 | Output | Display Parallel Interface pixel data output Blue2 |
| DISP_DATA19 | Output | Display Parallel Interface pixel data output Blue3 |
| DISP_DATA20 | Output | Display Parallel Interface pixel data output Blue4 |
| DISP_DATA21 | Output | Display Parallel Interface pixel data output Blue5 |
| DISP_DATA22 | Output | Display Parallel Interface pixel data output Blue6 |
| DISP_DATA23 | Output | Display Parallel Interface pixel data output Blue7 |

### 33.1.4 Clock and Interrupt signal

The following clocks are supplied to the LCDC.

Table 33.4 Clock List

| Clock Name | Clock Symbol | Description | Frequency (MHz) |
|---|---|---|---|
| LCDC_CLK_A | M0ϕ | AXI clock.<br>This clock should be a synchronous clock of LCDC_CLK_P | 200 |
| LCDC_CLK_P | ZTϕ | APB clock.<br>This clock should be a synchronous clock of LCDC_CLK_A | 100 |
| LCDC_CLK_D | M3ϕ | Video clock.<br>The duty of this clock is 50% when the display parallel interface | up to 148.5 |

The LCDC transmits the following interrupt signals to the interrupt controller.

Table 33.5 Interrupt List

| Signal Name | Interrupt Condition | Interrupt Source Register |
|---|---|---|
| VSPD_INT | Frame End | VI6_WPF0_IRQ_STA.FRE |
| | Display List Frame End | VI6_WPF0_IRQ_STA.DFE |
| | DU Connection UnderRun Error | VI6_WPF0_IRQ_STA.UND |
| DU_INT | RBUF FIFO Underflow | DU_MSR0.ST_PB_RUF |

**Note:** Refer to **Section 33.3.2.4(13)**. WPFn Interrupt Status Registers and **Section 33.3.3.1**. DU Module Control Register 0 for details.

## 33.2 Register Configuration

### 33.2.1 FCPVD Register Configuration

The FCPVD contains the registers which can directly be read and written to by the host CPU. The host can access any long word (32 bits) in the registers. Therefore, read-modify-write is needed to change partial bits in 32bits register.

Never access the registers when the FCPVD is in the module standby mode (with the operation clock placed in hold status).

Base Address: H'0_1088_0000 (Cortex-A55 Address Space)

Table 33.6 FCPVD Register Configuration

| Register Name | Abbreviation | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|
| FCPVD version control register | FCP_VCR | R | H'0000 0109 | H'0000 | 32 |

### 33.2.2 VCPD Register Configuration

Base Address: H'0_1087_0000 (Cortex-A55 Address Space)

Lowercase "n" in a register name or a offset address indicates an integer and the range of value n is defined when necessary. For RPFn, and WPFn, when the range of value n is not defined, RPFn (n = 0, 1), WPFn (n = 0) are assumed.

Table 33.7 VCPD Register Configuration (1/3)

| Register Name | Abbreviation | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|
| VSPD Start Registers 0 | VI6_CMD0 | R/W*1 | H'0000 0000 | H'0000 | 32 |
| Clock Control Register 0 | VI6_CLK_CTRL0 | R/W | H'0000 0000 | H'0010 | 32 |
| Clock Control Register 1 | VI6_CLK_CTRL1 | R/W | H'0000 0000 | H'0014 | 32 |
| Dynamic Clock Stop Control Register | VI6_CLK_DCSWT | R/W | H'0000 0000 | H'0018 | 32 |
| Dynamic Clock Stop Disable Register 0 | VI6_CLK_DCSM0 | R/W | H'0000 0000 | H'001c | 32 |
| Dynamic Clock Stop Disable Register 1 | VI6_CLK_DCSM1 | R/W | H'0000 0000 | H'0020 | 32 |
| Software Reset Register | VI6_SRESET | R/W | H'0000 0000 | H'0028 | 32 |
| Module Reset Enable Register 0 | VI6_MRESET_ENB0 | R/W | H'0000 0000 | H'002c | 32 |
| Module Reset Enable Register 1 | VI6_MRESET_ENB1 | R/W | H'0000 0000 | H'0030 | 32 |
| Module Reset Issuing Register | VI6_MRESET | R/W | H'0000 0000 | H'0034 | 32 |
| Operating Status Register | VI6_STATUS | R | H'0000 0000 | H'0038 | 32 |
| WPF0 Interrupt Enable Register | VI6_WPF0_IRQ_ENB | R/W | H'0000 0000 | H'0048 | 32 |
| WPF0 Interrupt Status Register | VI6_WPF0_IRQ_STA | R/W | H'0000 0000 | H'004c | 32 |
| Display-0 Interrupt Enable Register | VI6_DISP0_IRQ_ENB | R/W | H'0000 0000 | H'0078 | 32 |
| Display-0 Interrupt Status Register | VI6_DISP0_IRQ_STA | R/W | H'0000 0000 | H'007c | 32 |
| Display List Control Register | VI6_DL_CTRL | R/W | H'0000 0000 | H'0100 | 32 |
| Display List-0 Header Address Register | VI6_DL_HDR_ADDR0 | R/W | H'0000 0000 | H'0104 | 32 |
| Display List-0 Data Swapping Register | VI6_DL_SWAP0 | R/W | H'0000 0000 | H'0114 | 32 |
| Extended Display List-0 Control Register | VI6_DL_EXT_CTRL0 | R/W | H'0000 0000 | H'011c | 32 |
| Display List Body Size Register 0 | VI6_DL_BODY_SIZE0 | R/W | H'0000 0000 | H'0120 | 32 |
| Display List-0 Header Reference Address Register | VI6_DL_HDR_REF_ADDR0 | R | H'0000 0000 | H'0130 | 32 |
| Display List-0 Wake Up Counter Register | VI6_DL_WUPCNT0 | R/W | H'0000 0000 | H'0158 | 32 |
| RPFn Basic Read Size Registers | VI6_RPFn_SRC_BSIZE | R/W | H'0000 0000 | H'0300 + H'0100*n | 32 |
| RPFn Extended Read Size Registers | VI6_RPFn_SRC_ESIZE | R/W | H'0000 0000 | H'0304 + H'0100*n | 32 |

Table 33.7 VCPD Register Configuration (2/3)

| Register Name | Abbreviation | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|
| RPFn Input Format Registers | VI6_RPFn_INFMT | R/W | H'0000 0000 | H'0308 + H'0100*n | 32 |
| RPFn Data Swapping Registers | VI6_RPFn_DSWAP | R/W | H'0000 0000 | H'030c + H'0100*n | 32 |
| RPFn Display Location Registers | VI6_RPFn_LOC | R/W | H'0000 0000 | H'0310 + H'0100*n | 32 |
| RPFn α Plane Selection Control Registers | VI6_RPFn_ALPH_SEL | R/W | H'0000 0000 | H'0314 + H'0100*n | 32 |
| RPFn Virtual Plane Color Information Registers | VI6_RPFn_VRTCOL_SET | R/W | H'0000 0000 | H'0318 + H'0100*n | 32 |
| RPFn Mask Control Registers | VI6_RPFn_MSKCTRL | R/W | H'0000 0000 | H'031c + H'0100*n | 32 |
| RPFn IROP-SRC Input Value Registers 0 | VI6_RPFn_MSKSET0 | R/W | H'0000 0000 | H'0320 + H'0100*n | 32 |
| RPFn IROP-SRC Input Value Registers 1 | VI6_RPFn_MSKSET1 | R/W | H'0000 0000 | H'0324 + H'0100*n | 32 |
| RPFn Color Keying Control Registers | VI6_RPFn_CKEY_CTRL | R/W | H'0000 0000 | H'0328 + H'0100*n | 32 |
| RPFn Color Keying Color Setting Registers 0 | VI6_RPFn_CKEY_SET0 | R/W | H'0000 0000 | H'032c + H'0100*n | 32 |
| RPFn Color Keying Color Setting Registers 1 | VI6_RPFn_CKEY_SET1 | R/W | H'0000 0000 | H'0330 + H'0100*n | 32 |
| RPFn Source Picture Memory Stride Setting Registers | VI6_RPFn_SRCM_PSTRIDE | R/W | H'0000 0000 | H'0334 + H'0100*n | 32 |
| RPFn Source α Memory Stride Setting Registers | VI6_RPFn_SRCM_ASTRIDE | R/W | H'0000 0000 | H'0338 + H'0100*n | 32 |
| RPFn Source Y/RGB Address Registers | VI6_RPFn_SRCM_ADDR_Y | R/W | H'0000 0000 | H'033c + H'0100*n | 32 |
| RPFn Source Chroma Address Registers 0 | VI6_RPFn_SRCM_ADDR_C0 | R/W | H'0000 0000 | H'0340 + H'0100*n | 32 |
| RPFn Source Chroma Address Registers 1 | VI6_RPFn_SRCM_ADDR_C1 | R/W | H'0000 0000 | H'0344 + H'0100*n | 32 |
| RPFn Source α Address Registers | VI6_RPFn_SRCM_ADDR_AI | R/W | H'0000 0000 | H'0348 + H'0100*n | 32 |
| RPFn Bus Access Control Registers | VI6_RPFn_BAC | R/W | H'0000 0000 | H'0350 + H'0100*n | 32 |
| RPFn Multiple Alpha Control Registers | VI6_RPFn_MULT_ALPH | R/W | H'0000 0000 | H'036C+ H'0100*n | 32 |
| WPFn-Source-RPF Registers | VI6_WPFn_SRCRPF | R/W | H'0000 0000 | H'1000 + H'0100*n | 32 |
| WPFn Horizontal Input Size Clipping Registers | VI6_WPFn_HSZCLIP | R/W | H'0000 0000 | H'1004 + H'0100*n | 32 |
| WPFn Vertical Input Size Clipping Registers | VI6_WPFn_VSZCLIP | R/W | H'0000 0000 | H'1008 + H'0100*n | 32 |
| WPFn Output Format Registers | VI6_WPFn_OUTFMT | R/W | H'0000 0000 | H'100c + H'0100*n | 32 |
| WPFn Data Swapping Registers | VI6_WPFn_DSWAP | R/W | H'0000 0000 | H'1010 + H'0100*n | 32 |
| WPFn Rounding Control Registers | VI6_WPFn_RNDCTRL | R/W | H'0000 0000 | H'1014 + H'0100*n | 32 |
| WPFn Destination Y Plane Memory Stride Registers | VI6_WPFn_DSTM_STRIDE_Y | R/W | H'0000 0000 | H'101c + H'0100*n | 32 |
| WPFn Destination C Plane Memory Stride Registers | VI6_WPFn_DSTM_STRIDE_C | R/W | H'0000 0000 | H'1020 + H'0100*n | 32 |

Table 33.7 VCPD Register Configuration (3/3)

| Register Name | Abbreviation | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|
| WPFn Destination Y/RGB Address Registers | VI6_WPFn_DSTM_ADDR_Y | R/W | H'0000 0000 | H'1024 + H'0100*n | 32 |
| WPFn Destination Chroma Address Registers 0 | VI6_WPFn_DSTM_ADDR_C0 | R/W | H'0000 0000 | H'1028 + H'0100*n | 32 |
| WPFn Destination Chroma Address Registers 1 | VI6_WPFn_DSTM_ADDR_C1 | R/W | H'0000 0000 | H'102c + H'0100*n | 32 |
| WPF0 LIF Write Back Control Registers | VI6_WPF0_WRBCK_CTRL | R/W | H'0000 0000 | H'1034 | 32 |
| RPF0 Routing Register | VI6_DPR_RPF0_ROUTE | R/W | H'0000 0000 | H'2000 | 32 |
| RPF1 Routing Register | VI6_DPR_RPF1_ROUTE | R/W | H'0000 0000 | H'2004 | 32 |
| WPF0 Timing Control Register | VI6_DPR_WPF0_FPORCH | R/W | H'0000 0000 | H'2014 | 32 |
| LUT Routing Register | VI6_DPR_LUT_ROUTE | R/W | H'0000 0000 | H'203c | 32 |
| BRS Routing Register | VI6_DPR_BRS_ROUTE | R/W | H'0000 0000 | H'2050 | 32 |
| LUT Control Register | VI6_LUT_CTRL | R/W | H'0000 0000 | H'2800 | 32 |
| BRS Input Control Register | VI6_BRS_INCTRL | R/W | H'0000 0000 | H'3900 | 32 |
| Size Register of BRS Input Virtual RPF | VI6_BRS_VIRRPF_SIZE | R/W | H'0000 0000 | H'3904 | 32 |
| Display Location Register of BRS Input Virtual RPF | VI6_BRS_VIRRPF_LOC | R/W | H'0000 0000 | H'3908 | 32 |
| Color Information Register of BRS Input Virtual RPF | VI6_BRS_VIRRPF_COL | R/W | H'0000 0000 | H'390c | 32 |
| BRS Control Registers A | VI6_BRSA_CTRL | R/W | H'0000 0000 | H'3910 | 32 |
| BRS Blend Control Registers A | VI6_BRSA_BLD | R/W | H'0000 0000 | H'3914 | 32 |
| BRS Control Registers B | VI6_BRSB_CTRL | R/W | H'0000 0000 | H'3918 | 32 |
| BRS Blend Control Registers B | VI6_BRSB_BLD | R/W | H'0000 0000 | H'391c | 32 |
| LIF0 Control Registers | VI6_LIF0_CTRL | R/W | H'0000 0000 | H'3b00 | 32 |
| LIF0 Clock Stop Buffer Control Register | VI6_LIF0_CSBTH | R/W | H'0000 0000 | H'3b04 | 32 |
| LIF0 Buffer Attribute Register | VI6_LIF0_LBA | R/W | H'0000 0000 | H'3b0c | 32 |
| LIF0 Padding Line Cycle Register | VI6_LIF0_PADLN_CYC | R/W | H'0000 0000 | H'3b30 | 32 |
| LIF0 Padding Line Pattern Register | VI6_LIF0_PADLN_PT | R/W | H'0000 0000 | H'3b34 | 32 |
| LIF0 Padding Line Value Register | VI6_LIF0_PADLN_VAL | R/W | H'0000 0000 | H'3b38 | 32 |
| LIF0 Padding Line Size Register | VI6_LIF0_PADLN_SIZE | R/W | H'0000 0000 | H'3b3C | 32 |
| LUT table | VI6_LUT_TBL_0 To VI6_LUT_TBL_255 | R/W | Undefined | H'7000 To H'73fc | 32 |

Note 1. Read only bits or write only bits or read/write bits are mixed in the registers.

### 33.2.3 DU Register Configuration

The DU contains the registers which can directly be read and written to by the host CPU. The host can access any long word (32 bits), word (16 bits), and byte (8 bits) in the registers. Therefore, read-modify-write is needed to change partial bits not more than 8 bits in 32 bits register.

Never access the registers when the DU is in the module standby mode (with the operation clock placed in hold status).

Base Address: H'0_1089_0000 (Cortex-A55 Address Space)

Table 33.8 DU Register Configuration

| Register Name | Abbreviation | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|
| DU Module Control Register 0 | DU_MCR0 | R/W | H'00000000 | H'00 | 32/16/8 |
| DU Module Status Register 0 | DU_MSR0 | R | H'00540000 | H'04 | 32/16/8 |
| DU Module Status Register 1 | DU_MSR1 | R | H'00000000 | H'08 | 32/16/8 |
| DU Interrupt Mask Register 0 | DU_IMR0 | R/W | H'00000000 | H'0C | 32/16/8 |
| DU Display I/F Timing Register 0 | DU_DITR0 | R/W | H'00030300 | H'10 | 32/16/8 |
| DU Display I/F Timing Register 1 | DU_DITR1 | R/W | H'00000000 | H'14 | 32/16/8 |
| DU Display I/F Timing Register 2 | DU_DITR2 | R/W | H'00000000 | H'18 | 32/16/8 |
| DU Display I/F Timing Register 3 | DU_DITR3 | R/W | H'00000000 | H'1C | 32/16/8 |
| DU Display I/F Timing Register 4 | DU_DITR4 | R/W | H'00000000 | H'20 | 32/16/8 |
| DU Module Control Register 1 | DU_MCR1 | R/W | H'00000000 | H'40 | 32/16/8 |
| DU PBUF Control Register 0 | DU_PBCR0 | R/W | H'00000000 | H'4C | 32/16/8 |
| DU PBUF Control Register 1 | DU_PBCR1 | R/W | H'00000000 | H'50 | 32/16/8 |
| DU PBUF Control Register 2 | DU_PBCR2 | R/W | H'00000000 | H'54 | 32/16/8 |

# 33.3 Register Descriptions

## 33.3.1 FCPVD Registers

### 33.3.1.1 FCPVD version Control Register (FCP_VCR)

FCP_VCR is the version control register of the FCPVD. The value read is fixed.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | CATEGORY | | | | | | | | REVISION | | | | | | | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 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 16 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 15 to 8 | CATEGORY | H'01 | R | FCP use case category. |
| 7 to 0 | REVISION | H'09 | R | Version of LSI product<br>H'09: FCP revision is this LSI. |

## 33.3.2 VCPD Registers

### 33.3.2.1 Notational Conventions for Registers and Bit Fields

This section uses the following notational conventions for the VSPD registers and bit fields.

1. The names of registers and bits are written in uppercase.
2. A bit or bit field in a register is indicated as [register name.bit name]. For example, the STRCMD bit in the VI6_CMD0 register is indicated as VI6_CMD0.STRCMD.
3. Lowercase “n” in a register name or a bit name indicates an integer and the range of value n is defined when necessary. For RPFn, and WPFn, when the range of value n is not defined, RPFn (n = 0, 1), WPFn (n = 0) are assumed.
4. In each subsection for register description in **Section 33.3.2**, when only a bit name is written without showing its register name, the bit is in the register described in that subsection.
5. A wildcard (*) indicates any characters in a name and represents all registers or bits that match the specified first part of a name. For example, when there are two registers VI6_RPF_SRC_BSIZE and VI6_RPF_SRC_ESIZE, VI6_RPF_SRC_* indicates both registers.

### 33.3.2.2 Register Classification

The VSPD registers are arranged in the following order; the general control registers to display list control registers control operation of the entire VSPD, and the other registers control each image processing and specify parameters for the processing. The functions of the registers are described in this order starting from **Section 33.3.2.4**.

1. General control registers (VI6_CMDn (n = 0, 1), VI6_SRESET, VI6_STATUS, VI6_WPFn_IRQ_* (n = 0))
2. Display list control registers (VI6_DL_*)
3. RPF control registers (VI6_RPFn_* (n = 0, 1))
4. WPF control registers (VI6_WPFn_* (n = 0))
5. DPR control registers (VI6_DPR_*)
6. LUT control register (VI6_LUT_CTRL)
7. BRS control registers (VI6_BRS_*)
8. LIFn control registers (VI6_LIFn_* (n = 0))

The VSPD has two RPF channels and the register configuration is the same for all of RPFn (n = 0, 1). However, some bit fields have restrictions in certain RPFs. These restrictions are included in the description of the corresponding bit fields and registers. Likewise, the register configuration is the same for all WPFn (n = 0), but some bit fields have restrictions in certain WPFs; the restrictions are included in the description of the corresponding bit fields and registers.

For each register address, refer to **Table 33.7**.

#### 33.3.2.3 Restrictions on Access to Registers and Lookup Tables

The VSPD has control registers and lookup tables. All VSPD registers are writable and readable by only 32 bits unit. To write partial bits in each register, read-modify-write is needed. When accessing the addresses where these registers and lookup tables are allocated, the following restrictions should be satisfied. If any restriction is violated, the VSPD will not operate correctly.

1. For the read-only bits and reserved bits in all VSPD registers, writing 1 is prohibited unless otherwise specified.
2. Addresses undefined in **Table 33.7** are reserved areas and write access is prohibited in these areas.
3. For all registers and lookup tables, except VI6_CMDn, VI6_SRESET, VI6_*IRQ* and two plane registers such as VI6_DL_HDR_ADDRn, VI6_DL_BODY_SIZE0, modifying register values during operation of the module is prohibited. Modify registers while the corresponding module is stopped. For the operating status of the target module, refer to **Section 33.4.4**.
4. There are three types about General control registers (**Section 33.3.2.4**) and Display List Control Registers (**Section 33.3.2.5**) as below.
   - Controlling WPF0 (ex. VI6_CMD0, VI6_SRESET.SRST0)
   - Common setting of WPF0 (ex. VI6_DL_CTRL.AR_WAIT [15:0])

Table 33.9 Correspondence between Modules and Register Names

| Module Name | Register Name |
|---|---|
| RPFn (n = 0, 1) | VI6_RPFn_* |
| WPFn (n = 0) | VI6_WPFn_* |
| DPR | VI6_DPR_* |
| LUT | VI6_LUT_CTRL |
| BRS | VI6_BRS_* |
| LIFn (n = 0) | VI6_LIFn_* |

### 33.3.2.4 General Control Registers

#### (1) VSPD Start Registers n (VI6_CMDn: n = 0)

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | UPDHDR | — | — | — | STRCMD |
| 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 | R | R | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 5 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 4 | UPDHDR | 0 | R/W | Reserved state of updating Display List Header (DLH) address.<br>UPDHDR can be negated by software reset (VI6_SRESET.SRSTn).<br>[Write]<br>0: NOP<br>1: Don't set 1 to this bit<br>[Read]<br>0: Not reserved to update DLH address.<br>1: Reserved to update DLH address.<br>Set 0 to this bit when STRCMD bit is set to 1. |
| 3 to 1 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 0 | STRCMD | 0 | R/W | Start reservation of WPF<br>VI6_CMDn.STRCMD controls WPFn. Writing 1 to this bit starts WPFn in VSPD. Set this bit for activation only after all register settings in each output channel have been completed.<br>If WPFn is idle, WPFn starts right after this bit is set to 1.<br>If WPFn is active, writing 1 to this bit reserves starting WPFn operation. Wait to set 1 to VI6_CMDn.STRCMD until this bit is read as 0.<br>VI6_CMDn.STRCMD can be negated by software reset (VI6_SRESET.SRSTn).<br>[Write]<br>0: Start reservation of WPFn is canceled.<br>1: Start reservation of WPFn is set.<br>[Read]<br>0: Starting VSPD is not reserved.<br>1: Starting VSPD is reserved. |

The basic concept of image processing operation started by activating the VSPD is shown in **Figure 33.5**. The actual data input/output is executed by the MAU, which is the bus interface module, as described in **Section 33.1.2**, but conceptually the RPF works as the data entry point to the VSPD and the WPF works as the data exit point. To process images through the VSPD, the RPF (entrance) and WPF (exit) should be connected and a data path from the RPF to the WPF should be formed.

To connect RPFn to WPFn, specify RPFn as the source RPF for WPFn in VI6_WPFn_SRCRPF, which is a register for WPFn (refer to **Section 33.3.2.7(1)**). This setting determines that RPFn will be started when WPFn is started through VI6_CMDn.

A data path to execute desired image processing should then be formed between the RPF (entrance) and WPF (exit). To form a data path, connect the necessary function modules in the VSPD between RPF and WPF. This function is provided by the DPR; specify the information for each module connection in data path routing registers VI6_DPR_*_ROUTE (refer to **Section 33.3.2.8**).

After a data path is formed (RPFn → WPFn) as described above, starting output module WPFn in the VSPD through VI6_CMDn starts all function modules connected to WPFn and the desired image processing is executed. According to this design concept, starting a WPF module means starting the VSPD.

There are two types of data path configuration in the VSPD; one is “a single input module to a single output module” as shown in **Figure 33.5 (A)**, and the other is “multiple input modules to a single output module” as shown in **Figure 33.5 (B)**.

**Figure 33.5 (A)** shows an example of a configuration where modules with single input and single output are implemented through the DPR.

**Figure 33.5 (B)** shows another configuration example where the module with multiple input and single output is implemented through the DPR.

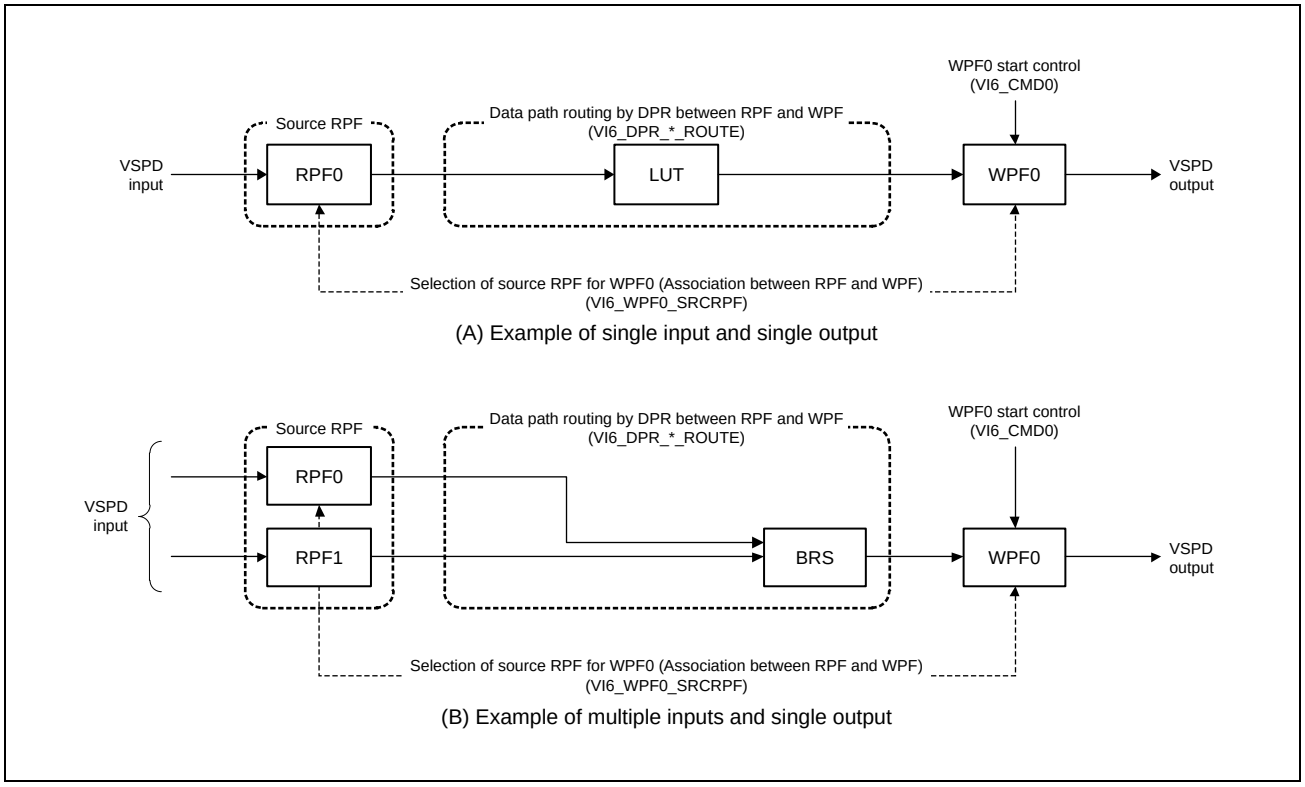



Figure 33.5 Basic Concept of VSPD Startup

### (2) Clock Control Register 0 (VI6_CLK_CTRL0)

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | GCS0 | — | — | — | — | — | — | — | — | — | — | — | GCS1 |
| 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 | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | GCS2 | | | | — | — | — | GCS3 | | | | |
| 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/W | R/W | R/W | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 29 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 28 | GCS0 | 0 | R/W | Clock Control Setting0 |
| 27 to 17 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 16 | GCS1 | 0 | R/W | Clock Control Setting1 |
| 15 to 12 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 11 to 8 | GCS2 | 0 | R/W | Clock Control Setting2 |
| 7 to 5 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 4 to 0 | GCS3 | 0 | R/W | Clock Control Setting3 |

VSPD can stop its operating clock for reducing power consumption.

To enable clock stop function, set following registers:

- VI6_CLK_CTRL0 = H’0000 0000
- VI6_CLK_CTRL1 = H’0000 0000
- VI6_CLK_DCSWT = H’0000 0808
- VI6_CLK_DCSM0 = H’0000 0000
- VI6_CLK_DCSM1 = H’0000 0000

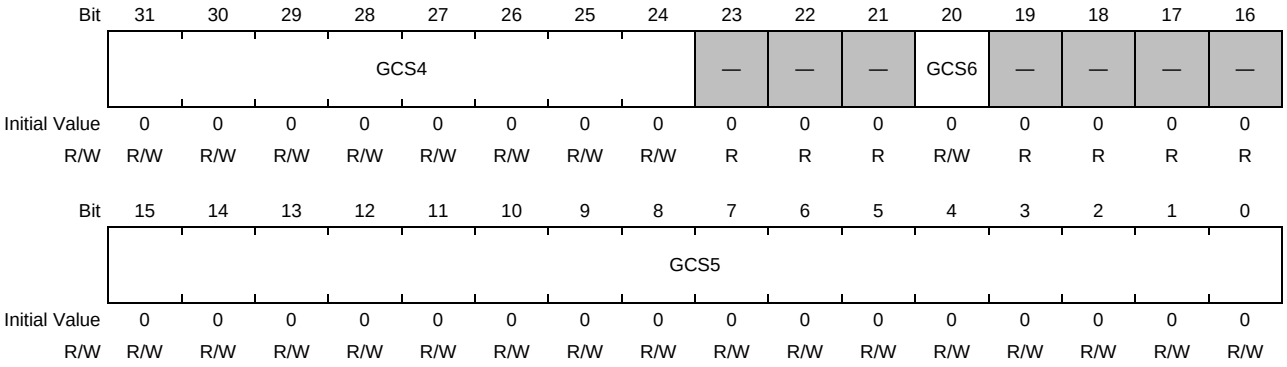
To disable clock stop function, set following registers:

- VI6_CLK_CTRL0 = H’1001 0F1F
- VI6_CLK_CTRL1 = H’FF10 FFFF
- VI6_CLK_DCSWT = H’0033 0808
- VI6_CLK_DCSM0 = H’1FFF 0F1F
- VI6_CLK_DCSM1 = H’FF10 FFFF

When the clock gating is enabled, the supply of the clock signal is stopped for the module which is not operated. If the bit field of this register is changed from 0 to 1, the supply of the clock signal for the module is immediately started. If

the bit field of the register is changed from 1 to 0, the clock supply is immediately stopped. Therefore, please do not change this register field from 1 to 0 in operation because the VSPD may be stalled.

### (3) Clock Control Register 1 (VI6_CLK_CTRL1)




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 24 | GCS4 | 0 | R/W | Clock Control Setting4 |
| 23 to 21 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 20 | GCS6 | 0 | R/W | Clock Control Setting6 |
| 19 to 16 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 15 to 0 | GCS5 | 0 | R/W | Clock Control Setting5 |

See **Section 33.3.2.4(2)** for detail.

### (4) Dynamic Clock Stop Control Register (VI6_CLK_DCSWT)

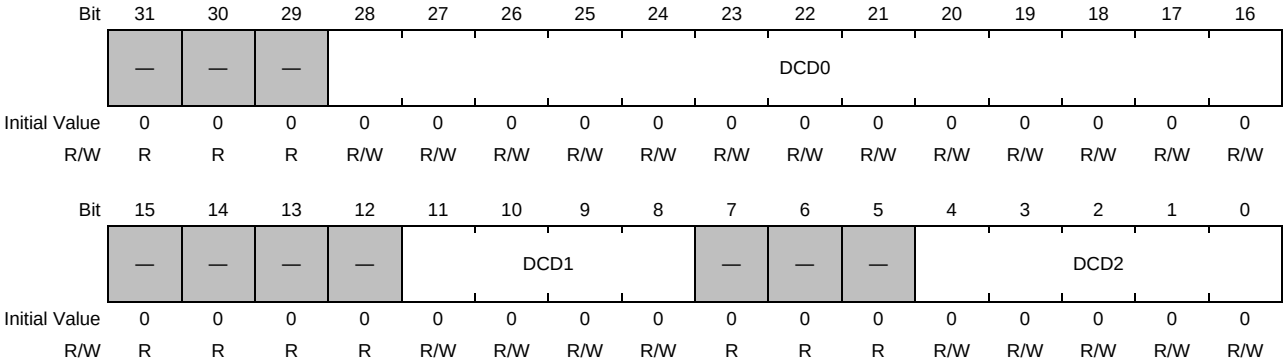
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | DCC0 | | — | — | DCC1 | DCC2 |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | R | R | R/W | R/W | R | R | R/W | R/W |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | CSTPW[7:0] | | | | | | | | CSTRW[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 22 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 21, 20 | DCC0 | 0 | R/W | Dynamic Clock Control Setting0 |
| 19, 18 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 17 | DCC1 | 0 | R/W | Dynamic Clock Control Setting1 |
| 16 | DCC2 | 0 | R/W | Dynamic Clock Control Setting2 |
| 15 to 8 | CSTPW[7:0] | 0 | R/W | Dynamic Clock Stop Control 1<br>Always specify 8. |
| 7 to 0 | CSTRW[7:0] | 0 | R/W | Dynamic Clock Stop Control 2<br>Always specify 8. |

See **Section 33.3.2.4(2)** for detail.

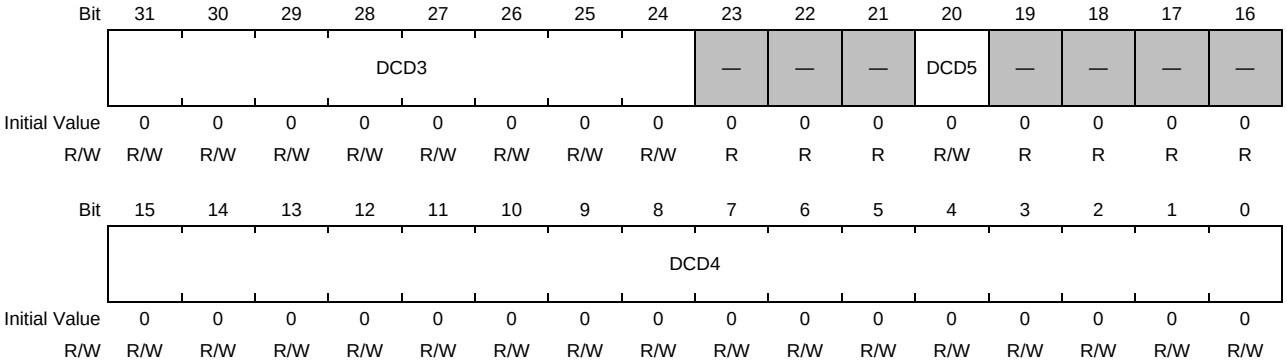
### (5) Dynamic Clock Stop Disable Register 0 (VI6_CLK_DCSM0)




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 29 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 28 to 16 | DCD0 | 0 | R/W | Dynamic Clock Stop Disable Setting0 |
| 15 to 12 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 11 to 8 | DCD1 | 0 | R/W | Dynamic Clock Stop Disable Setting1 |
| 7 to 5 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 4 to 0 | DCD2 | 0 | R/W | Dynamic Clock Stop Disable Setting2 |

See **Section 33.3.2.4(2)** for detail.

### (6) Dynamic Clock Stop Disable Register 1 (VI6_CLK_DCSM1)




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 24 | DCD3 | 0 | R/W | Dynamic Clock Stop Disable Setting3 |
| 23 to 21 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 20 | DCD5 | 0 | R/W | Dynamic Clock Stop Disable Setting5 |
| 19 to 16 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 15 to 0 | DCD4 | 0 | R/W | Dynamic Clock Stop Disable Setting4 |

See **Section 33.3.2.4(2)** for detail.

### (7) Software Reset Register (VI6_SRESET)

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | SRST0 |
| 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<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 0 | SRST0 | 0 | R/W | WPFn Software Reset (SRSTn, n = 0)<br>Writing 1 to this bit aborts the current processing in WPFn (the partially-completed image undergoing processing is output). The period until this software reset processing is completed depends on the bus state.<br>When this reset processing is completed, the VI6_WPFn_IRQ_STA.FRE interrupt source bit is set to 1; when the FRE interrupt is enabled, the FRE end interrupt is output to notify the end of the reset processing.<br>This bit is always read as 0.<br>0: NOP<br>1: WPFn software reset*1 |

Note 1. Applying a software reset to each WPF has the following restrictions.

1. A software reset can be applied to only one of WPF0 through single write access to VI6_SRESET.
2. After a software reset is issued, no more software reset can be issued to another WPF until the issued software reset processing is completed
3. The end of software reset processing is notified through the FRE bit in VI6_WPFn_IRQ_STA, but the software reset issued while WPF is stopped is ignored as NOP. As it takes a while until the reset is actually issued after the reset bit is set, the VSPD may complete operation before the reset is actually issued. In this case, no interrupt is output for the software reset that is issued after the VSPD completes operation.
4. If a software reset is issued during downloading of a display list, the downloading processing is not aborted. After the end of downloading that is in progress when a software reset is issued, a frame end interrupt is output.

### (8) Module Reset Enable Register 0 (VI6_MRESET_ENB0)

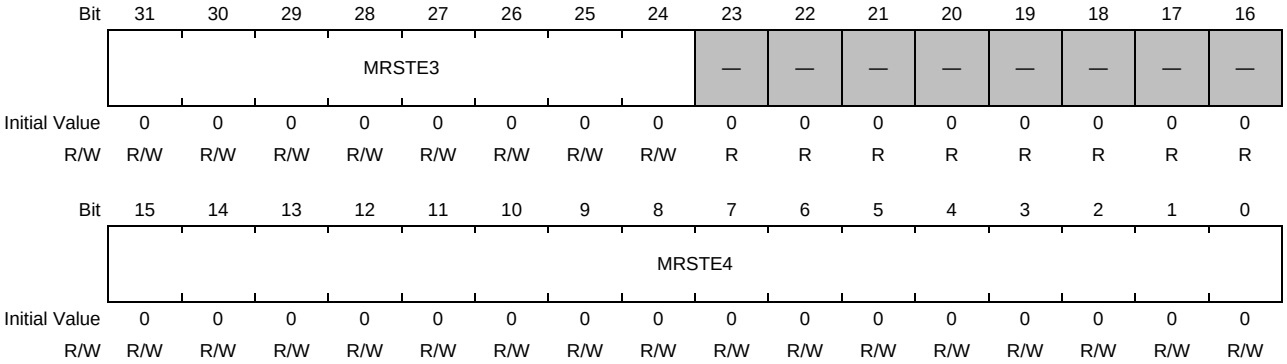
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | MRSTE0 | | — | — | — | — | — | — | — | — | — | — | — | — |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R/W | R/W | R | R | R | R | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | MRSTE1 | | | | — | — | — | MRSTE2 | | | | |
| 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/W | R/W | R/W | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31, 30 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 29, 28 | MRSTE0 | 0 | R/W | Module Reset Enable 0 |
| 27 to 12 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 11 to 8 | MRSTE1 | 0 | R/W | Module Reset Enable 1 |
| 7 to 5 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 4 to 0 | MRSTE2 | 0 | R/W | Module Reset Enable 2 |

This register is for purpose of h/w debugging.

### (9) Module Reset Enable Register 1 (VI6_MRESET_ENB1)




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 24 | MRSTE3 | 0 | R/W | Module Reset Enable 3 |
| 23 to 16 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 15 to 0 | MRSTE4 | 0 | R/W | Module Reset Enable 4 |

This register is for purpose of h/w debugging.

### (10) Module Reset Issuing Register (VI6_MRESET)

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | MRST |
| 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<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 0 | MRST | 0 | R/W | Module Reset Assertion |

This register is for purpose of h/w debugging.

### (11) Operating Status Register (VI6_STATUS)

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | FLDST0 | — | — | — | — | — | — | — | — | — | — | — | — |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | SYS0_ACT | — | — | — | — | — | — | — | — |
| 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 29 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 28 | FLDST0 | 0 | R | Field status of previous frame of WPF0.<br>0: Previous finished frame is TOP field.<br>1: Previous finished frame is BOT field.<br><br>This bit can be referred in case of AUTO-FLD or AUTO-DISP.<br>This bit is changed at the timing of VI6_WPF0_IRQ_STA.FRE. |
| 27 to 9 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 8 | SYS0_ACT | 0 | R | WPFn Operating Status (SYSn_ACT, n=0)<br>Each bit indicates the operating or stopped state of control channel n (WPFn).<br>0: WPFn is stopped.<br>1: WPFn is operating. |
| 7 to 0 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |

### (12) WPFn Interrupt Enable Registers (VI6_WPFn_IRQ_ENB: n = 0)

Each bit controls the interrupt enable of the corresponding interrupt source.

0: Interrupt Disabled

1: Interrupt Enabled

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | UNDE |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | DFEE | FREE |
| 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 |
|---|---|---|---|---|
| 31 to 17 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 16 | UNDE | 0 | R/W | Interrupt Enable for WPFn (n = 0) Underrun in case of DU connection. |
| 15 to 2 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 1 | DFEE | 0 | R/W | Interrupt Enable for WPFn (n = 0) Display List Frame End |
| 0 | FREE | 0 | R/W | Interrupt Enable for WPFn (n = 0) Frame End |

Each bit in VI6_WPFn_IRQ_STA is set to 1 when the corresponding interrupt source is generated. VI6_WPFn_IRQ_ENB specifies whether to output an interrupt signal for the generated source. When an interrupt is disabled in this register, no interrupt signal is generated even when the corresponding bit in VI6_WPFn_IRQ_STA is set to 1. When an interrupt is enabled in this register, an interrupt signal is output when the corresponding bit in VI6_WPFn_IRQ_STA is set to 1.

### (13) WPFn Interrupt Status Registers (VI6_WPFn_IRQ_STA: n = 0)

The read value from each bit is the status of the interrupt source, and the write access to each bit controls the interrupt status.

[Read Access] Interrupt Status

0: No interrupt

1: Interrupt activated

[Write Access] Interrupt Clear

0: The interrupt status is cleared to 0

1: Hold the interrupt status value

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | UND |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | DFE | FRE |
| 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 |
|---|---|---|---|---|
| 31 to 17 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 16 | UND | 0 | R/W | Interrupt Status and Clear for WPFn (n = 0) Underrun in case of DU connection.<br>This interrupt source bit is set to 1 when data underrun occurs in case of DU connection.<br>Timing that this bit is set to 1 is at end of 1 frame. |
| 15 to 2 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 1 | DFE | 0 | R/W | Interrupt Status and Clear for WPFn (n = 0) Display List Frame End<br>This interrupt source bit is set to 1 when VSPD completes one-frame processing while the current frame in enable value stored in the display list header is 1 (refer to **Section 33.4.4**).<br>When display lists are not used, this bit is not used. In this case, clear VI6_WPFn_IRQ_ENB.DFEE to 0 to mask the interrupt generation by this interrupt source.<br><br>This bit can hold the most recent of two times of interrupt status at the maximum. When writing 0 to this bit, the oldest interrupt status is cleared. And the status bit becomes 0 after all interrupt statuses are cleared. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 0 | FRE | 0 | R/W | Interrupt Status and Clear for WPFn (n = 0) Frame End<br>This interrupt source bit is set to 1 when VSPD completes one-frame processing. This bit is also set to 1 when one-frame processing using a display list is completed.<br>The interrupt status is set to 1 by any of the following conditions.<br>a) Processing one frame is finished normally<br>b) Software reset is issued during VSPD is processing.<br>c) One frame's data from DU is displayed while VSPD couldn't transfer one frame data to DU. (in case of Linked with DU)<br><br>This bit can hold the most recent of two times of interrupt status at the maximum. When writing 0 to this bit, the oldest interrupt status is cleared. And the status bit becomes 0 after all interrupt statuses are cleared. |

VI6_WPFn_IRQ_STA indicates the state of the interrupt sources generated in the VSPD. Whether to output a VSPD interrupt when an interrupt source is generated and the corresponding bit is set to 1 is determined by the corresponding bit setting in VI6_WPFn_IRQ_ENB. While an interrupt is disabled in VI6_WPFn_IRQ_ENB, the VSPD does not output an interrupt signal even when an interrupt source is generated, but the source flag in this register is set to 1.

Note that the interrupt source bits in this register cannot be cleared by write access using a display list.

### (14) Display-n Interrupt Enable Register (VI6_DISPn_IRQ_ENB: n = 0)

Each bit controls the interrupt enable of the corresponding interrupt source.

0: Interrupt Disabled

1: Interrupt Enabled

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | DSTE | — | — | MAEE | — | — | — | — | — |
| 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 | R | R | R | R |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 9 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 8 | DSTE | 0 | R/W | Interrupt Enable for Display Start |
| 7 to 6 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 5 | MAEE | 0 | R/W | Interrupt Enable for Display Read Data End |
| 4 to 0 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |

### (15) Display-n Interrupt Status Register (VI6_DISPn_IRQ_STA: n = 0)

The read value from each bit is the status of the interrupt source, and the write access to each bit controls the interrupt status.

[Read Access] Interrupt Status

0: No interrupt

1: Interrupt activated

[Write Access] Interrupt Clear

0: The interrupt status is cleared to 0

1: Hold the interrupt status value

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | DST | — | — | MAE | — | — | — | — | — |
| 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 | R | R | R | R |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 9 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 8 | DST | 0 | R/W | Interrupt Enable for Display Start |
| 7, 6 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 5 | MAE | 0 | R/W | Interrupt Enable for Display Read Data End |
| 4 to 0 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |

### 33.3.2.5 Display List Control Registers

#### (1) Display List Control Register (VI6_DL_CTRL)

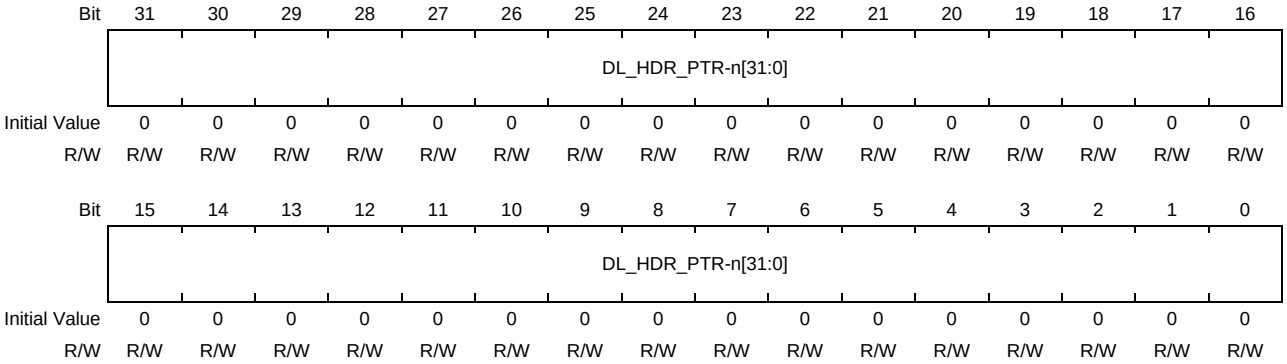
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | AR_WAIT[15:0] | | | | | | | | | | | | | | | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R/W | 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 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | DC2 | — | — | — | DC1 | — | — | — | DLE1 | RLM0 | CFM0 | NH0 | DLE0 |
| 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 | R | R | 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 16 | AR_WAIT [15:0] | All 0 | R/W | Display List Control Setting<br>Always specify 256. |
| 15 to 13 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 12 | DC2 | 0 | R/W | This bit doesn't affect anything to VSPD (NOP) |
| 11 to 9 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 8 | DC1 | 0 | R/W | This bit doesn't affect anything to VSPD (NOP) |
| 7 to 5 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 4 | DLE1 | 0 | R/W | This bit doesn't affect anything to VSPD (NOP) |
| 3 | RLM0 | 0 | R/W | Loading two plane Registers Mode for WPF0.<br>0: Reserved status (VI6_CMD0.UPDHDR) is accepted by next frame auto start. Two plane registers are downloaded by next_frame_start also.<br>1: Reserved status (VI6_CMD0.UPDHDR) is not accepted by next frame auto start. pNextHeader in previous frame's DLH is used for loading DisplayList. Two plane registers are not downloaded by next_frame_start also. |
| 2 | CFM0 | 0 | R/W | Continuous Frame Mode for Header-less Display List for WPF0<br>This bit determines whether the next frame is automatically started or not. When the updated flag of the display list, VI6_DL_BODY_SIZE0.UPD0, is not updated, the display list of the next frame is not transferred and the same register values are used for the next frame. When the value of VI6_DL_BODY_SIZE0.UPD0 is updated, the new display list is transferred.<br>0: Stopped at the end of every frame<br>1: The next frame is automatically started |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 1 | NH0 | 0 | R/W | Header-less Display List Mode<br>This bit is used for specifying the header-less display list mode. In case of header-less mode, the number of the display lists is 1. The address of the display body is set in VI6_DL_HDR_ADDR0 register, and the body size is set in VI6_DL_BODY_SIZE0 register.<br>When this bit is changed, make sure that VSPD is stopped. And also make sure the following before starting VSPD.<br>– Header Address (VI6_DL_HDR_ADDR0)<br>– Body Size (VI6_DL_BODY_SIZE0) in case of header-less mode<br>0: Use Display List Header (Normal DL Mode)<br>1: Don't use Display List Header (Header-less Mode)<br>*Note 1.* Only WPF0 supports header-less display list. WPF1 work as the normal display list mode even if the WPF0 is set to header-less display list mode.<br>*Note 2.* When DLE0 bit is 0, set 0 to NH0 bit. |
| 0 | DLE0 | 0 | R/W | Display List Enable/Disable for WPF0<br>Enables or disables the WPF0 display list function. When the display list function is enabled through this bit, all WPF processing channels work in display list mode.<br>When using display lists, note the restrictions in **Section 33.3.2.3**.<br>0: The display list function is disabled<br>1: The display list function is enabled |

### (2) Display List-n Header Address Register (VI6_DL_HDR_ADDRn: n = 0)




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 0 | DL_HDR_PTR-n[31:0] | All 0 | R/W | Display List-n Header Address<br>These bits specify the address of the display list header to be read for display list-n in 16-byte units (the lower-order four bits are read-only). When WPFn is first started in display list mode, the display list header is loaded from the address specified in this register. After loading of the header is completed, the register value of the display list address is updated to the next header address stored in the loaded header to prepare for loading of the next display list header. After that, this header address updating is repeated.<br><br>A value from H'0000 0000 to H'FFFF FFF0 can be specified. |

### (3) Display List-n Data Swapping Register (VI6_DL_SWAPn: n = 0)

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | IND | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| 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 | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | LWS | WDS | BTS |
| 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 | IND | 0 | R/W | Enabling independent swap setting per WPF.<br>This bit is available only for VI6_DL_SWAP1.<br>This bit is reserved for VI6_DL_SWAP0.<br>0: Display list swap for WPF1 is specified by LWS, WDS and BTS in VI6_DL_SWAP0.<br>1: Display list swap for WPF1 is specified by LWS, WDS and BTS in VI6_DL_SWAP1 |
| 30 to 3 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 2 | LWS | 0 | R/W | Display List Data Swapping in long word Units<br>The effect of this bit setting is defined in **Table 33.10**.<br>0: Data swapping in long word (32-bit) units is disabled<br>1: Data swapping in long word (32-bit) units is enabled |
| 1 | WDS | 0 | R/W | Display List Data Swapping in Word Units<br>The effect of this bit setting is defined in **Table 33.10**.<br>0: Data swapping in word (16-bit) units is disabled<br>1: Data swapping in word (16-bit) units is enabled |
| 0 | BTS | 0 | R/W | Display List Data Swapping in Byte Units<br>The effect of this bit setting is defined in **Table 33.10**.<br>0: Data swapping in byte (8-bit) units is disabled<br>1: Data swapping in byte (8-bit) units is enabled |

**Table 33.10** shows the data order before and after swapping according to the long word, word, and byte swapping settings.

When data order in memory for each format is the same as **Table 33.34**, set 111b to {LWS, WDS, BTS}. If data order is different from the definition, change data order within 8byte unit by these bits as shown in **Table 33.10**.

Table 33.10 Changing data order according to display list swap register

| Data order in memory | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Byte address | 8n+0 | 8n+1 | 8n+2 | 8n+3 | 8n+4 | 8n+5 | 8n+6 | 8n+7 | *_LWS | *_WDS | *_BTS |
| Data | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 1 | 1 | 1 |
| | 1 | 0 | 3 | 2 | 5 | 4 | 7 | 6 | 1 | 1 | 0 |
| | 2 | 3 | 0 | 1 | 6 | 7 | 4 | 5 | 1 | 0 | 1 |
| | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 1 | 0 | 0 |
| | 4 | 5 | 6 | 7 | 0 | 1 | 2 | 3 | 0 | 1 | 1 |
| | 5 | 4 | 7 | 6 | 1 | 0 | 3 | 2 | 0 | 1 | 0 |
| | 6 | 7 | 4 | 5 | 2 | 3 | 0 | 1 | 0 | 0 | 1 |
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 0 | 0 | 0 |

| Data order defined in **Table 33.34** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Byte address | 8n+0 | 8n+1 | 8n+2 | 8n+3 | 8n+4 | 8n+5 | 8n+6 | 8n+7 |
| Data | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |

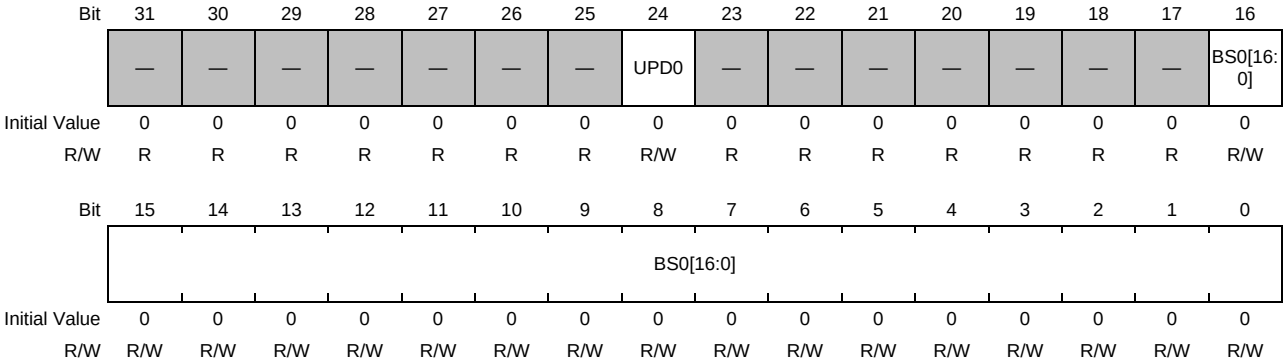
### (4) Extended Display List-n Control Register (VI6_DL_EXT_CTRLn: n = 0)

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | NWE |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | POLINT | | | | | | — | — | DLPRI | EXPRI | — | — | — | EXT |
| 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/W | R/W | R/W | R/W | R | R | R/W | R/W | R | R | R | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 17 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 16 | NWE | 0 | R/W | No Wait for Polling<br>When this bit is set to 1, the polling condition for extended display lists is always assumed to be true. |
| 15, 14 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 13 to 8 | POLINT | All 0 | R/W | Extended Display List Command Control<br>Always specify 2. |
| 7, 6 | — | 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 5 | DLPRI | 0 | R/W | Display List Control 0<br>Always specify 1. |
| 4 | EXPRI | 0 | R/W | Display List Control 1<br>Always specify 0. |
| 3 to 1 | — | 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 0 | EXT | 0 | R/W | Extended Display List for WPFn<br>Enables or disables the extended display list function. When extended display lists are used, the display list header size is 96 bytes; when they are not used, the header size is 80 bytes.<br>0: No extended display lists are used<br>1: Extended display lists are used<br>[Note]<br>When using extended display lists, be sure to also use normal display list mode (VI6_DL_CTRL.DLEn); executing only extended display lists is not possible.<br>When the header-less display list mode is activated, this bit should be set to 0. The extended display list cannot be used with the header-less display list mode.<br>Extended Display List is available only for VSPD/LIF0 or VSPDL/LIFn (n = 0, 1) to realized AUTO-FLD or AUTO-DISP. |

### (5) Display List Body Size Register n (VI6_DL_BODY_SIZEn: n = 0)




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 25 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 24 | UPD0 | 0 | R/W | Update Flag<br>This bit controls the download of the display list at the next downloading timing in case that the header-less display list mode is used. Set 1 to this bit when display list need to be downloaded for next frame. This bit is automatically cleared to 0 after VI6_DL_HDR_ADDRn and VI6_DL_BODY_SIZEn.BS0 (n = 0) is downloaded in H/W side. When this bit is set to 1, the value of VI6_DL_HDR_ADDRn and VI6_DL_BODY_SIZEn.BS0 (n = 0) should not be changed.<br>0: Updating display list for the next frame is not reserved<br>1: Updating display list for the next frame is reserved |
| 23 to 17 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 16 to 0 | BS0[16:0] | 0 | R/W | Header-less Display List Body Size (WPFn, n = 0)<br>These bits are used for specifying the body size of the display list in case of header-less display list mode. The unit of the size is byte. The value should be set in multiples of 8. |

See **Section 33.4.8.2** for detail.

### (6) Display List-n Header Reference Address Register (VI6_DL_HDR_REF_ADDRn: n = 0)

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | DL_HDR_REF_PTR-n [31:0] | | | | | | | | | | | | | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | DL_HDR_REF_PTR-n [31:0] | | | | | | | | | | | | | | | |
| 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 0 | DL_HDR_REF_PTR-n [31:0] | All 0 | R | Display List-n reference Header Address<br>Following value is read out from the VI6_DL_HDR_REF_ADDRn for each period.<br>(1) When H/W is reading display list from external memory, header address of the display list referred by VSPD -H/W is read out.<br>(2) When H/W is not reading display list from external memory, the value of VI6_DL_HDR_ADDRn is read out. |

### (7) Display List-0 Wake Up Counter Register (VI6_DL_WUPCNT0)

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| 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 0 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |

### 33.3.2.6 RPF Control Registers

#### (1) RPFn Basic Read Size Registers (VI6_RPFn_SRC_BSIZE)

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | BHSIZE[12:0] | | | | | | | | | | | | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R/W | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | BVSIZE[12:0] | | | | | | | | | | | | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R/W | 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 29 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 28 to 16 | BHSIZE[12:0] | All 0 | R/W | Horizontal Size of RPF Basic Read Area<br>These bits specify the horizontal size of the basic area to be read from the external RAM by the RPFn. When the input format is YCbCr4:2:2 or YCbCr4:2:0, specify the size in 2-pixel units. A value from 1 to 1280 can be specified.<br>Specify a value equal to or smaller than the extended read size (VI6_RPFn_SRC_ESIZE.EHSIZE). |
| 15 to 13 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 12 to 0 | BVSIZE[12:0] | All 0 | R/W | Vertical Size of RPF Basic Read Area<br>These bits specify the vertical size of the basic area to be read from the external RAM by the RPFn. When the input format is YCbCr4:2:0, specify the size in 2-pixel units. A value from 1 to 800 can be specified.<br>Specify a value equal to or smaller than the extended read size (VI6_RPFn_SRC_ESIZE.EVSIZE). |

**Figure 33.6** shows the relationship between the basic read size and extended read size. The RPF reads data from the source memory area specified by the basic read size. The RPF repeats reading the basic read area in the horizontal and vertical directions up to the extended read size and sends the read data to the processing modules in the VSPD.

For basic read size reading, the reading start address, called the RPFn source image storing address, should be specified in VI6_RPFn_SRCM_ADDR_*. In the memory area where the basic read area image is stored, the distance (number of bytes) between addresses for lines n and n + 1 of two-dimensional image data, called the memory stride, should be specified in VI6_RPFn_SRCM_PSTRIDE.

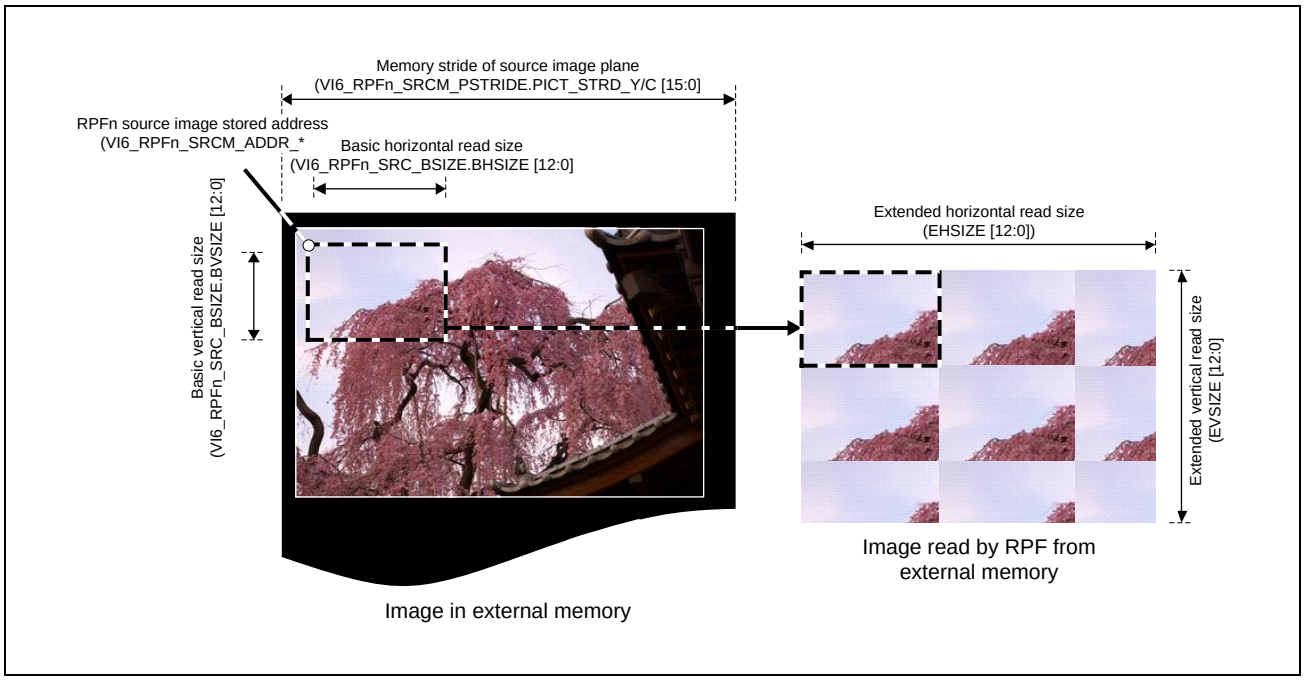
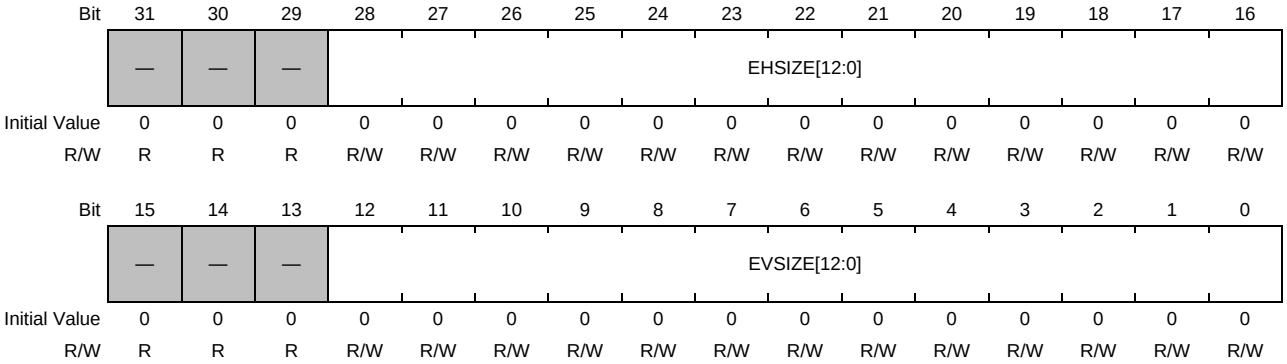



Figure 33.6 Relationship between Basic Read Size and Extended Read Size

Refer also to the following sections.

- **Section 33.3.2.6(2), RPFn Extended Read Size Registers (VI6_RPFn_SRC_ESIZE)**
- **Section 33.3.2.6(13), RPFn Source Picture Memory Stride Setting Registers (VI6_RPFn_SRCM_PSTRIDE)**
- **Section 33.3.2.6(15), RPFn Source Y/RGB Address Registers (VI6_RPFn_SRCM_ADDR_Y)**
- **Section 33.3.2.6(16), RPFn Source Chroma Address Registers 0 (VI6_RPFn_SRCM_ADDR_C0)**
- **Section 33.3.2.6(17), RPFn Source Chroma Address Registers 1 (VI6_RPFn_SRCM_ADDR_C1)**

### (2) RPFn Extended Read Size Registers (VI6_RPFn_SRC_ESIZE)




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 29 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 28 to 16 | BHSIZE[12:0] | All 0 | R/W | RPF Extended Horizontal Read Size<br>These bits specify the horizontal size of the extended read area to which the RPFn reads data from the external RAM. As shown in **Figure 33.6**, the basic read area image is repeatedly placed in the extended read area; in the EHSIZE bits, specify a value not smaller than the horizontal size of the basic read area. When the input format is YCbCr4:2:2 or YCbCr4:2:0, specify the size in 2-pixel units (an even value).<br>A value from 1 to 1280 can be specified. |
| 15 to 13 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 12 to 0 | EVSIZE[12:0] | All 0 | R/W | RPF Extended Vertical Read Size<br>These bits specify the vertical size of the extended read area to which the RPFn reads data from the external RAM. As shown in **Figure 33.6**, the basic read area image is repeatedly placed in the extended read area; in the EVSIZE bits, specify a value not smaller than the vertical size of the basic read area. When the input format is YCbCr4:2:0, specify the size in 2-pixel units (an even value).<br>A value from 1 to 800 can be specified. |

VI6_RPFn_SRC_ESIZE specifies the extended size for RPFn. The extended horizontal and vertical sizes should be equal to or greater than the basic sizes specified in VI6_RPFn_SRC_BSIZE. The RPF internal data processing described later and image processing described in **Section 33.3.2.8** and later sections are all applied to the image in the extended read size shown on the right side in **Figure 33.6**.

### (3) RPFn Input Format Registers (VI6_RPFn_INFMT)

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | VIR | — | — | — | — | — | — | — | — | — | — | — | CIPM |
| 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 | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | SPYCS | SPUVS | CEXT[1:0] | | RDTM[2:0] | | | CSC | — | RDFMT[6:0] | | | | | | |
| 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/W | R/W | R/W | R/W | R/W | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 29 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 28 | VIR | 0 | R/W | Virtual Input Enable<br>Enables or disables the virtual input function of the RPFn. The image to be processed by the RPFn is usually read from the external memory by the MAU. Instead of this input, the virtual input function generates a single-color image within the RPFn and sends it to the modules in VSPD. When the virtual input function is enabled, the fixed value specified in VI6_RPFn_VRTCOL_SET is used as the input to the RPFn.<br>While the virtual input function is enabled, data is not read from the external memory; that is, the α plane is not read and the IROP calculation thus cannot be executed. In this case, set VI6_RPFn_ALPH_SEL.ASEL to 4. Neither the color space conversion through CSC nor the color keying described in **Section 33.3.2.6(11)** can be used.<br>0: RPFn uses general input.<br>1: RPFn uses virtual input. |
| 27 to 17 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 16 | CIPM | 0 | R/W | Horizontal Chrominance Interpolation Method Setting<br>Image data is processed in the YCbCr444 format inside VSPD in case of YCbCr color space. When the chrominance format of the input image is YCbCr422 or YCbCr420, data is up-sampled as shown in **Figure 33.7** for internal processing. This bit specifies the method of up-sampling for this purpose.<br>0: The nearest-neighbor method is used for horizontal chrominance interpolation.<br>1: The bilinear method is used for horizontal chrominance interpolation. |
| 15 | SPYCS | 0 | R/W | RPF Input Mode Setting 1<br>When the input format is YUY2, set this bit to 1 and set the RDFMT bits to 71 (H'47).<br>When the input format is YVYU, set this bit and the SPUVS bit to 1 and set the RDFMT bits to 71 (H'47).<br>In other cases, set this bit to 0. |
| 14 | SPUVS | 0 | R/W | RPF Input Mode Setting 2<br>When the input format is NV61, set this bit to 1 and set the RDFMT bits to 65 (H'41).<br>When the input format is NV21, set this bit to 1 and set the RDFMT bits to 66 (H'42).<br>When the input format is YVYU, set this bit and the SPYCS bit to 1 and set the RDFMT bits to 71 (H'47).<br>In other cases, set this bit to 0. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 13, 12 | CEXT[1:0] | All 0 | R/W | Lower-Bit Color Data Extension Method Setting<br>When an RGB input format where each color component is expressed in less than 8 bits are selected from **Table 33.12** through the RDFMT bits, VSPD internally extends each color component to 8 bits before using the data. These bits select this extension method.<br>00b: Lower-order bits of color data are extended with 0.<br>01b: Upper-order bits of color data are copied to the lower -order bits.<br>10b: Lower-order bits of color data are extended with 0. The maximum value is limited to H'FF.<br>11b: Setting prohibited |
| 11 to 9 | RDTM[2:0] | All 0 | R/W | CSC Conversion Expression Setting<br>These bits select the expression used for color space conversion. The conversion direction is RGB → YCbCr when RGB is selected through the RDFMT bits; the direction is YCbCr → RGB when YCbCr is selected.<br>0: BT.601 YCbCr [16,235/240] ↔ RGB [0,255]<br>1: BT.601 YCbCr [0,255] ↔ RGB [0,255]<br>2: BT.709 YCbCr [16,235/240] ↔ RGB [0,255]<br>3: BT.709 YCbCr [16,235/240] ↔ RGB [16,235]<br>4 to 7: Setting prohibited |
| 8 | CSC | 0 | R/W | Color Space Conversion Enable<br>Enables or disables color space conversion between YCbCr and RGB to be executed in RPFn. The characteristics of color space conversion are determined by the RDTM bit setting.*1<br>When using the virtual input (VIR = 1), specify 0.<br>0: Color space conversion is disabled.<br>1: Color space conversion is enabled. |
| 7 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 6 to 0 | RDFMT[6:0] | All 0 | R/W | RPF Input Image Format Setting<br>These bits select the format of the image input from the external RAM to the RPFn. Select a value corresponding to the desired format from those shown in **Table 33.12**, and **Table 33.13**.<br>When the virtual input function is used (VIR = 1), the color information for the virtual input should be specified in VI6_RPFn_VRTCOL_SET. If this color information is in the RGB format, set the RDFMT bits to 19. If the color information is in the YCbCr format, set these bits to 64.<br><br>[Note 1] Number of input pixels<br>When YCbCr4:2:2 is selected through the RDFMT bits, the horizontal size of the input image should be specified in 2-pixel units. When YCbCr4:2:0 is selected, the vertical and horizontal sizes should be specified in 2-pixel units. Observe these restrictions when specifying the image size in VI6_RPFn_SRC_BSIZE and VI6_RPFn_SRC_ESIZE. |

Note 1. Note on color space settings

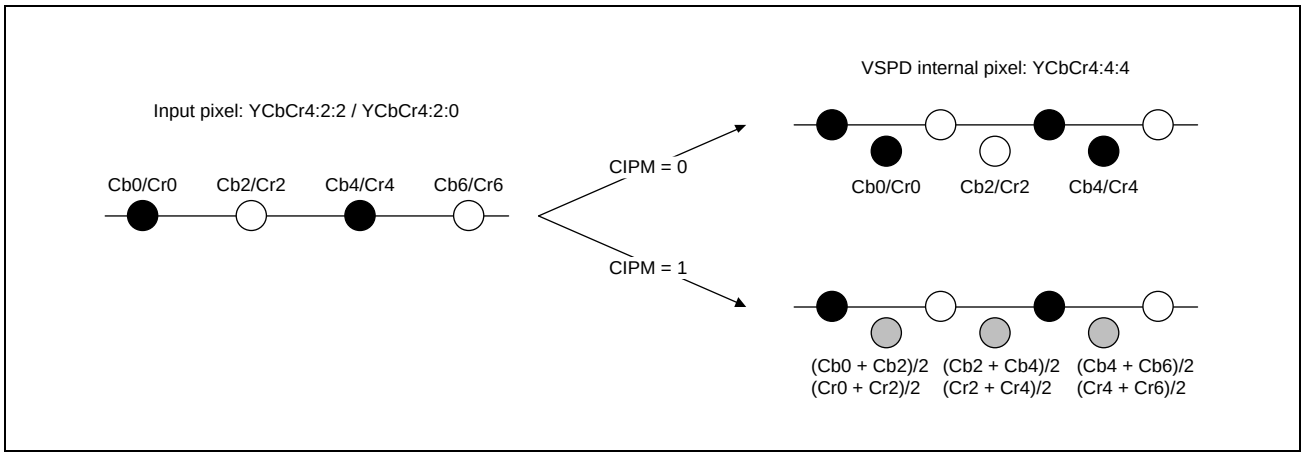



Figure 33.7 Chrominance Interpolation Methods Selectable through CIPM Setting

The color space for the image output from the RPF to VSPD internal modules is determined by the combination of the color space for the image input to the RPF, which is selected through the VI6_RPFn_INFMT.RDFMT setting, and the enabled or disabled state of the color space conversion function, which is selected through the VI6_RPFn_INFMT.CSC setting (**Table 33.11**). For example, when the image input to the RPF is in the YCbCr format, the RPF outputs data to VSPD internal modules in the YCbCr format if color space conversion is disabled through the CSC bit, and the RPF outputs data in the RGB format if color space conversion is enabled. When the image input to the RPF is in the RGB format, the relationship between the output format and the color space conversion setting is the opposite of the YCbCr case. For some VSPD internal modules, the YCbCr format is recommended for image processing because of the characteristics of the processing, or the same color space needs to be specified between multiple RPF outputs. In these cases, set VI6_RPFn_INFMT.RDFMT and VI6_RPFn_INFMT.CSC appropriately so that the RPFs can output the required color space according to the color space conditions described above.

Table 33.11 RPFn Input Color Space and Output Color Space

| RPFn Input Color Space (VI6_RPFn_INFMT. RDFMT) | | Color Space Conversion Setting (VI6_RPFn_INFMT.CSC) | | RPFn Output Color Space |
|---|---|---|---|---|
| RGB | (H'00 to H'3F)*1 | Disabled | (0) *1 | RGB |
| | | Enabled | (1) *1 | YCbCr |
| YCbCr | (H'40 to H'7F)*1 | Disabled | (0) *1 | YCbCr |
| | | Enabled | (1) *1 | RGB |

Note 1. Value specified in the register

A color space conversion function equivalent to that in the RPFn is also provided by the WPF. As shown in **Table 33.11**, the color space (YCbCr or RGB) output from the RPF becomes the input format for the WPF. Here, the color space of the output image obtained by the color space conversion function of the WPF must match the color space of the format specified through VI6_WPFn_OUTFMT.WRFMT.

**Figure 33.8** shows the relationship between the input/output format and color space. The input color space for the RPF is determined when the input image format for the RPF is specified through the RDFMT bits. The color space for the image output from the RPF to subsequent VSPD internal modules depends on the combination of the RPF input format and CSC (color space conversion function) enabled or disabled state in the RPF as shown in **Table 33.11** and **Figure 33.8**. The user should first determine whether the image processing in the VSPD is done in YCbCr or RGB, and then specify the RPF input format and CSC enabled or disabled state to obtain the desired color space. The color space of the RPF output image is also that of the WPF input image; the color space of the data output from the WPF to the outside of

VSPD depends on the combination of the WPF input color space and the enabled or disabled state of the CSC implemented in the WPF as shown in **Figure 33.8.** The color space of the WPF output image must match that of the WPF output format (determined by VI6_WPFn_OUTFMT.WRFMT). For example, in the flow shown in **Figure 33.8.**, YCbCr should not be specified as the WPF output format regardless of the fact that the color space of the WPF output image is in RGB format.

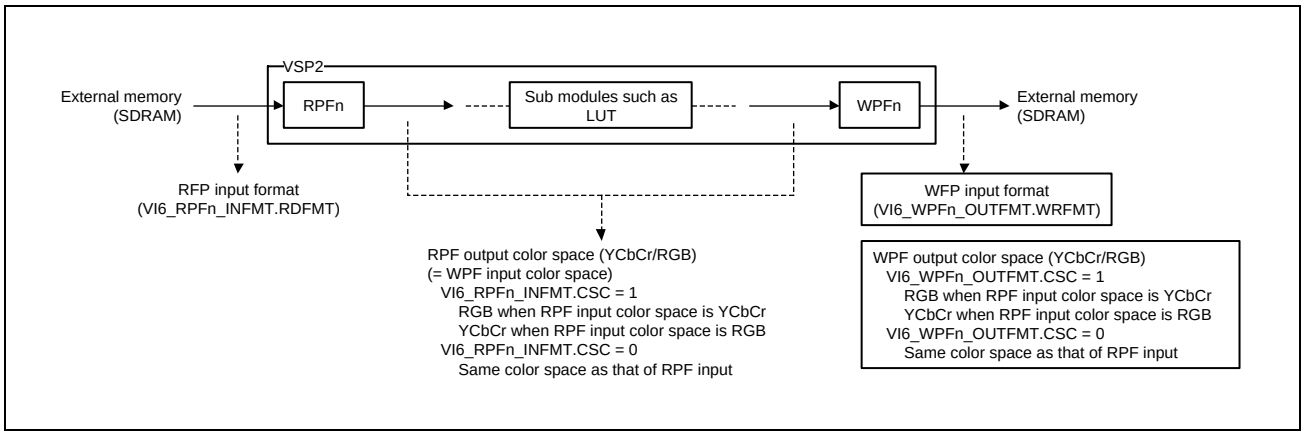



Figure 33.8 Relationship between Input/output Format and Color Space

Table 33.12 Packed Formats for RPF Input

| RDFMT[6:0] | Bit per pixel | Phase | upper row - address / bottom row - bit field | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | n | | | | | | | | n+1 | | | | | | | | n+2 | | | | | | | | n+3 | | | | | | | |
| | | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| H'00 | 8 | — | R0 | R0 | R0 | G0 | G0 | G0 | B0 | B0 | R1 | R1 | R1 | G1 | G1 | G1 | B1 | B1 | R2 | R2 | R2 | G2 | G2 | G2 | B2 | B2 | R3 | R3 | R3 | G3 | G3 | G3 | B3 | B3 |
| H'01 | 12 | — | | | | | R0 | R0 | R0 | R0 | G0 | G0 | G0 | G0 | B0 | B0 | B0 | B0 | | | | | R1 | R1 | R1 | R1 | G1 | G1 | G1 | G1 | B1 | B1 | B1 | B1 |
| H'02 | | | R0 | R0 | R0 | R0 | G0 | G0 | G0 | G0 | B0 | B0 | B0 | B0 | | | | | R1 | R1 | R1 | R1 | G1 | G1 | G1 | G1 | B1 | B1 | B1 | B1 | | | | |
| H'03 | — | — | Reserved | | | | | | | | Reserved | | | | | | | | Reserved | | | | | | | | Reserved | | | | | | | |
| H'04 | 15 | — | | R0 | R0 | R0 | R0 | R0 | G0 | G0 | G0 | G0 | G0 | B0 | B0 | B0 | B0 | B0 | | R1 | R1 | R1 | R1 | R1 | G1 | G1 | G1 | G1 | G1 | B1 | B1 | B1 | B1 | B1 |
| H'05 | | | R0 | R0 | R0 | R0 | R0 | G0 | G0 | G0 | G0 | G0 | B0 | B0 | B0 | B0 | B0 | | R1 | R1 | R1 | R1 | R1 | G1 | G1 | G1 | G1 | G1 | B1 | B1 | B1 | B1 | B1 | |
| H'06 | 16 | — | R0 | R0 | R0 | R0 | R0 | G0 | G0 | G0 | G0 | G0 | G0 | B0 | B0 | B0 | B0 | B0 | R1 | R1 | R1 | R1 | R1 | G1 | G1 | G1 | G1 | G1 | G1 | B1 | B1 | B1 | B1 | B1 |
| H'07 | 18 | — | A | A | A | A | A | A | A | A | | | | | | | R0 | R0 | R0 | R0 | R0 | R0 | G0 | G0 | G0 | G0 | G0 | G0 | B0 | B0 | B0 | B0 | B0 | B0 |
| H'08 | | | R0 | R0 | R0 | R0 | R0 | R0 | G0 | G0 | G0 | G0 | G0 | G0 | B0 | B0 | B0 | B0 | B0 | B0 | | | | | | | A | A | A | A | A | A | A | A |
| H'09 | | | | | | | | | R0 | R0 | R0 | R0 | R0 | R0 | G0 | G0 | G0 | G0 | G0 | G0 | B0 | B0 | B0 | B0 | B0 | B0 | A | A | A | A | A | A | A | A |
| H'0A | | | A | A | A | A | A | A | A | A | R0 | R0 | R0 | R0 | R0 | R0 | G0 | G0 | G0 | G0 | G0 | G0 | B0 | B0 | B0 | B0 | B0 | B0 | | | | | | |
| H'0B | | | A | A | A | A | A | A | A | A | | | R0 | R0 | R0 | R0 | R0 | R0 | | | G0 | G0 | G0 | G0 | G0 | G0 | | | B0 | B0 | B0 | B0 | B0 | B0 |
| H'0C | | | | | R0 | R0 | R0 | R0 | R0 | R0 | | | G0 | G0 | G0 | G0 | G0 | G0 | | | B0 | B0 | B0 | B0 | B0 | B0 | A | A | A | A | A | A | A | A |
| H'0D | | | A | A | A | A | A | A | A | A | R0 | R0 | R0 | R0 | R0 | R0 | | | G0 | G0 | G0 | G0 | G0 | G0 | | | B0 | B0 | B0 | B0 | B0 | B0 | | |
| H'0E | | | R0 | R0 | R0 | R0 | R0 | R0 | | | G0 | G0 | G0 | G0 | G0 | G0 | | | B0 | B0 | B0 | B0 | B0 | B0 | | | A | A | A | A | A | A | A | A |
| H'0F | | 0 | | | | | | | R0 | R0 | R0 | R0 | R0 | R0 | G0 | G0 | G0 | G0 | G0 | G0 | B0 | B0 | B0 | B0 | B0 | B0 | | | | | | | R1 | R1 |
| | | 1 | R1 | R1 | R1 | R1 | G1 | G1 | G1 | G1 | G1 | G1 | B1 | B1 | B1 | B1 | B1 | B1 | | | | | | | R2 | R2 | R2 | R2 | R2 | R2 | G2 | G2 | G2 | G2 |
| | | 2 | G2 | G2 | B2 | B2 | B2 | B2 | B2 | B2 | | | | | | | R3 | R3 | R3 | R3 | R3 | R3 | G3 | G3 | G3 | G3 | G3 | G3 | B3 | B3 | B3 | B3 | B3 | B3 |
| H'10 | | 0 | R0 | R0 | R0 | R0 | R0 | R0 | G0 | G0 | G0 | G0 | G0 | G0 | B0 | B0 | B0 | B0 | B0 | B0 | | | | | | | R1 | R1 | R1 | R1 | R1 | R1 | G1 | G1 |
| | | 1 | G1 | G1 | G1 | G1 | B1 | B1 | B1 | B1 | B1 | B1 | | | | | | | R2 | R2 | R2 | R2 | R2 | R2 | G2 | G2 | G2 | G2 | G2 | G2 | B2 | B2 | B2 | B2 |
| | | 2 | B2 | B2 | | | | | | | R3 | R3 | R3 | R3 | R3 | R3 | G3 | G3 | G3 | G3 | G3 | B3 | B3 | B3 | B3 | B3 | B3 | B3 | | | | | | |
| H'11 | | 0 | | | R0 | R0 | R0 | R0 | R0 | R0 | | | G0 | G0 | G0 | G0 | G0 | G0 | | | B0 | B0 | B0 | B0 | B0 | B0 | | | R1 | R1 | R1 | R1 | R1 | R1 |
| | | 1 | | | G1 | G1 | G1 | G1 | G1 | G1 | | | B1 | B1 | B1 | B1 | B1 | B1 | | | R2 | R2 | R2 | R2 | R2 | R2 | | | G2 | G2 | G2 | G2 | G2 | G2 |
| | | 2 | | | B2 | B2 | B2 | B2 | B2 | B2 | | | R3 | R3 | R3 | R3 | R3 | R3 | | | G3 | G3 | G3 | G3 | G3 | G3 | | | B3 | B3 | B3 | B3 | B3 | B3 |
| H'12 | | 0 | R0 | R0 | R0 | R0 | R0 | R0 | | | G0 | G0 | G0 | G0 | G0 | G0 | | | B0 | B0 | B0 | B0 | B0 | B0 | | | R1 | R1 | R1 | R1 | R1 | R1 | | |
| | | 1 | G1 | G1 | G1 | G1 | G1 | G1 | | | B1 | B1 | B1 | B1 | B1 | B1 | | | R2 | R2 | R2 | R2 | R2 | R2 | | | G2 | G2 | G2 | G2 | G2 | G2 | | |
| | | 2 | B2 | B2 | B2 | B2 | B2 | B2 | | | R3 | R3 | R3 | R3 | R3 | R3 | | | G3 | G3 | G3 | G3 | G3 | G3 | | | B3 | B3 | B3 | B3 | B3 | B3 | | |
| H'13 | 24 | — | A | A | A | A | A | A | A | A | R0 | R0 | R0 | R0 | R0 | R0 | R0 | R0 | G0 | G0 | G0 | G0 | G0 | G0 | G0 | G0 | B0 | B0 | B0 | B0 | B0 | B0 | B0 | B0 |
| H'14 | | | R0 | R0 | R0 | R0 | R0 | R0 | R0 | R0 | G0 | G0 | G0 | G0 | G0 | G0 | G0 | G0 | B0 | B0 | B0 | B0 | B0 | B0 | B0 | B0 | A | A | A | A | A | A | A | A |
| H'15 | | 0 | R0 | R0 | R0 | R0 | R0 | R0 | R0 | R0 | G0 | G0 | G0 | G0 | G0 | G0 | G0 | G0 | B0 | B0 | B0 | B0 | B0 | B0 | B0 | B0 | R1 | R1 | R1 | R1 | R1 | R1 | R1 | R1 |
| | | 1 | G1 | G1 | G1 | G1 | G1 | G1 | G1 | G1 | B1 | B1 | B1 | B1 | B1 | B1 | B1 | B1 | R2 | R2 | R2 | R2 | R2 | R2 | R2 | R2 | G2 | G2 | G2 | G2 | G2 | G2 | G2 | G2 |
| | | 2 | B2 | B2 | B2 | B2 | B2 | B2 | B2 | B2 | R3 | R3 | R3 | R3 | R3 | R3 | R3 | R3 | G3 | G3 | G3 | G3 | G3 | G3 | G3 | G3 | B3 | B3 | B3 | B3 | B3 | B3 | B3 | B3 |
| H'16 | 18 | — | | | | | | | | R0 | R0 | R0 | R0 | R0 | R0 | G0 | G0 | G0 | | | | | | | | G0 | G0 | G0 | B0 | B0 | B0 | B0 | B0 | B0 |
| H'17 | | | | | | | | | | | | | | | | | R0 | R0 | R0 | R0 | R0 | R0 | G0 | G0 | G0 | G0 | G0 | G0 | B0 | B0 | B0 | B0 | B0 | B0 |
| H'18 | 24 | 0 | B0 | B0 | B0 | B0 | B0 | B0 | B0 | B0 | G0 | G0 | G0 | G0 | G0 | G0 | G0 | G0 | R0 | R0 | R0 | R0 | R0 | R0 | R0 | R0 | B1 | B1 | B1 | B1 | B1 | B1 | B1 | B1 |
| | | 1 | G1 | G1 | G1 | G1 | G1 | G1 | G1 | G1 | R1 | R1 | R1 | R1 | R1 | R1 | R1 | R1 | B2 | B2 | B2 | B2 | B2 | B2 | B2 | B2 | G2 | G2 | G2 | G2 | G2 | G2 | G2 | G2 |
| | | 2 | R2 | R2 | R2 | R2 | R2 | R2 | R2 | R2 | B3 | B3 | B3 | B3 | B3 | B3 | B3 | B3 | G3 | G3 | G3 | G3 | G3 | G3 | G3 | G3 | R3 | R3 | R3 | R3 | R3 | R3 | R3 | R3 |
| H'19 | 12 | — | A | A | A | A | R0 | R0 | R0 | R0 | G0 | G0 | G0 | G0 | B0 | B0 | B0 | B0 | A | A | A | A | R1 | R1 | R1 | R1 | G1 | G1 | G1 | G1 | B1 | B1 | B1 | B1 |
| H'1A | | — | R0 | R0 | R0 | R0 | G0 | G0 | G0 | G0 | B0 | B0 | B0 | B0 | A | A | A | A | R1 | R1 | R1 | R1 | G1 | G1 | G1 | G1 | B1 | B1 | B1 | B1 | A | A | A | A |
| H'1B | 15 | — | A | R0 | R0 | R0 | R0 | R0 | G0 | G0 | G0 | G0 | G0 | B0 | B0 | B0 | B0 | B0 | A | R1 | R1 | R1 | R1 | R1 | G1 | G1 | G1 | G1 | G1 | B1 | B1 | B1 | B1 | B1 |
| H'1C | | — | R0 | R0 | R0 | R0 | R0 | G0 | G0 | G0 | G0 | G0 | B0 | B0 | B0 | B0 | B0 | A | R1 | R1 | R1 | R1 | R1 | G1 | G1 | G1 | G1 | G1 | B1 | B1 | B1 | B1 | B1 | A |
| H'1D | 12 | — | A | A | A | A | B0 | B0 | B0 | B0 | G0 | G0 | G0 | G0 | R0 | R0 | R0 | R0 | A | A | A | A | B1 | B1 | B1 | B1 | G1 | G1 | G1 | G1 | R1 | R1 | R1 | R1 |
| H'1E | | — | B0 | B0 | B0 | B0 | G0 | G0 | G0 | G0 | R0 | R0 | R0 | R0 | A | A | A | A | B1 | B1 | B1 | B1 | G1 | G1 | G1 | G1 | R1 | R1 | R1 | R1 | A | A | A | A |
| H'1F | 15 | — | A | B0 | B0 | B0 | B0 | B0 | G0 | G0 | G0 | G0 | G0 | R0 | R0 | R0 | R0 | R0 | A | B1 | B1 | B1 | B1 | B1 | G1 | G1 | G1 | G1 | G1 | R1 | R1 | R1 | R1 | R1 |
| H'20 | | — | B0 | B0 | B0 | B0 | B0 | G0 | G0 | G0 | G0 | G0 | R0 | R0 | R0 | R0 | R0 | A | B1 | B1 | B1 | B1 | B1 | G1 | G1 | G1 | G1 | G1 | R1 | R1 | R1 | R1 | R1 | A |
| H'21 | 18 | 0 | | | B0 | B0 | B0 | B0 | B0 | B0 | | | G0 | G0 | G0 | G0 | G0 | G0 | | | R0 | R0 | R0 | R0 | R0 | R0 | | | B1 | B1 | B1 | B1 | B1 | B1 |
| | | 1 | | | G1 | G1 | G1 | G1 | G1 | G1 | | | R1 | R1 | R1 | R1 | R1 | R1 | | | B2 | B2 | B2 | B2 | B2 | B2 | | | G2 | G2 | G2 | G2 | G2 | G2 |
| | | 2 | | | R2 | R2 | R2 | R2 | R2 | R2 | | | B3 | B3 | B3 | B3 | B3 | B3 | | | G3 | G3 | G3 | G3 | G3 | G3 | | | R3 | R3 | R3 | R3 | R3 | R3 |
| H'22 | 24 | — | A | A | A | A | A | A | A | A | B0 | B0 | B0 | B0 | B0 | B0 | B0 | B0 | G0 | G0 | G0 | G0 | G0 | G0 | G0 | G0 | R0 | R0 | R0 | R0 | R0 | R0 | R0 | R0 |
| H'23 | 16 | — | | | | | | | | | | | | | | | | | R0 | R0 | R0 | R0 | R0 | G0 | G0 | G0 | G0 | G0 | G0 | B0 | B0 | B0 | B0 | B0 |
| H'24 to H'3F | — | — | Reserved | | | | | | | | Reserved | | | | | | | | Reserved | | | | | | | | Reserved | | | | | | | |

Table 33.13 Packed YCbCr Formats for RPF Input

| RDFMT[6:0] | Packed YCbCr Input Format | Reference |
|---|---|---|
| H'40 | YCbCr4:4:4 semi-planar | **Figure 33.9***4 |
| H'41 | YCbCr4:2:2 semi-planar (NV16, NV61*1) | |
| H'42 | YCbCr4:2:0 semi-planar (NV12, NV21*1) *5 | |
| H'43 to H'45 | Reserved | — |
| H'46 | YCbCr4:4:4 interleaved | **Figure 33.10***4 |
| H'47 | YCbCr4:2:2 interleaved type 0 (UYVY, YUY2*2, YVYU*3) | |
| H'48 | YCbCr4:2:2 interleaved type 1 | |
| H'49 | YCbCr4:2:0 interleaved *6 | |
| H'4a | YCbCr4:4:4 planar | **Figure 33.11***4 |
| H'4b | YCbCr4:2:2 planar (YV16) | |
| H'4c | YCbCr4:2:0 planar (YV12, I420) *5 | |
| H'4d to H'7e | Reserved | — |
| H'7f | Reserved | — |

Note 1. When the input format is NV61 or NV21, set the SPUVS bit to 1.

Note 2. When the input format is YUY2, set the SPYCS bit to 1.

Note 3. When the input format is YVYU, set the SPUVS bit to 1 and SPYCS bit to 1.

Note 4. **Figure 33.12** shows the definition of memory address for each pixel in **Figure 33.9** to **Figure 33.11**.

Note 5. Each line of C plane is read twice, so byte/pixel of YCbCr420 is 2 byte/pixel (same as YCbCr422).

Note 6. Each line of plane is read twice, so byte/pixel of YCbCr420ITL is 3 byte/pixel (same as YCbCr444ITL).

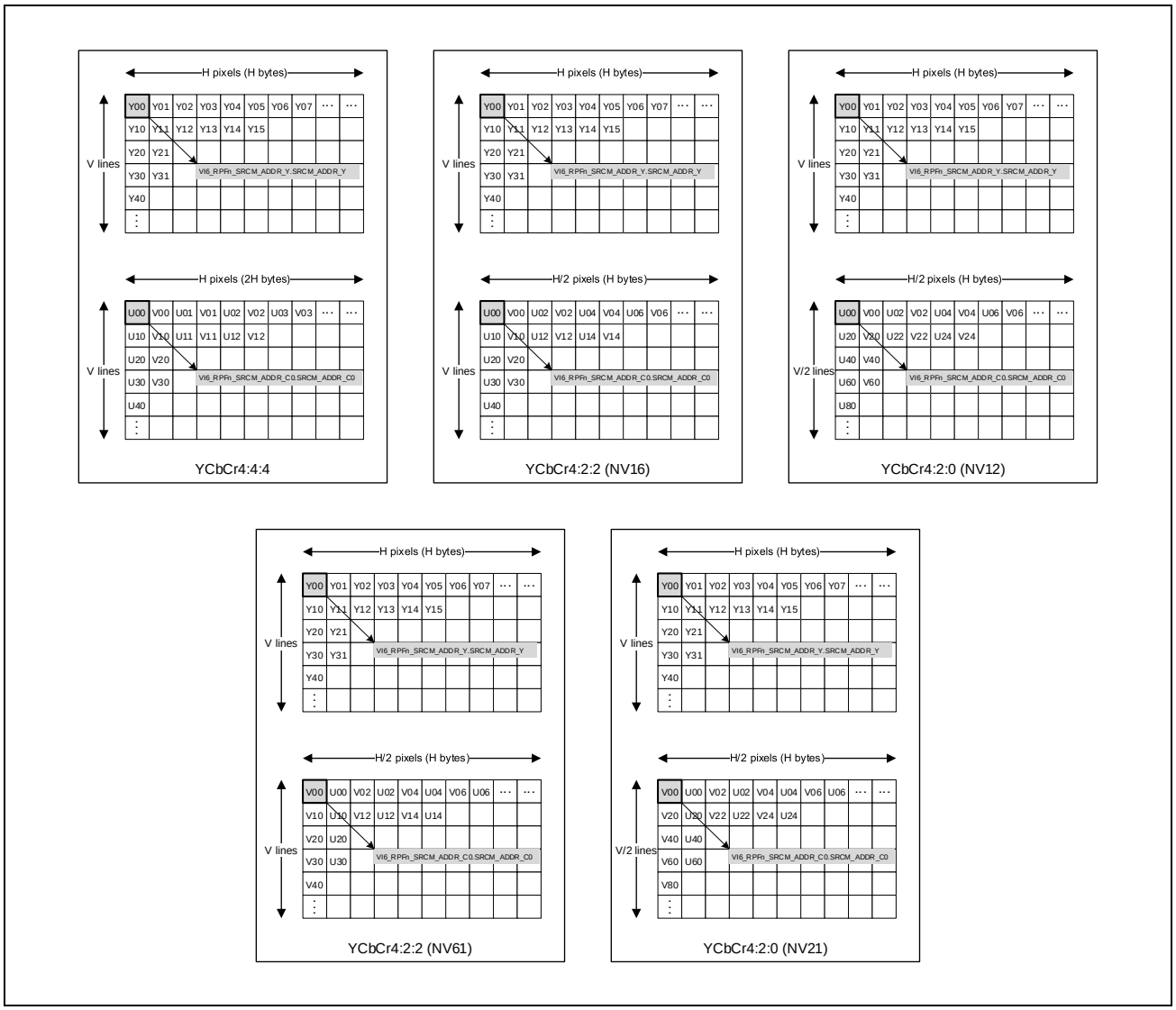



Figure 33.9 YCbCr Semi-Planar Formats*1

**Note 1.** This figure is for 8bpc.

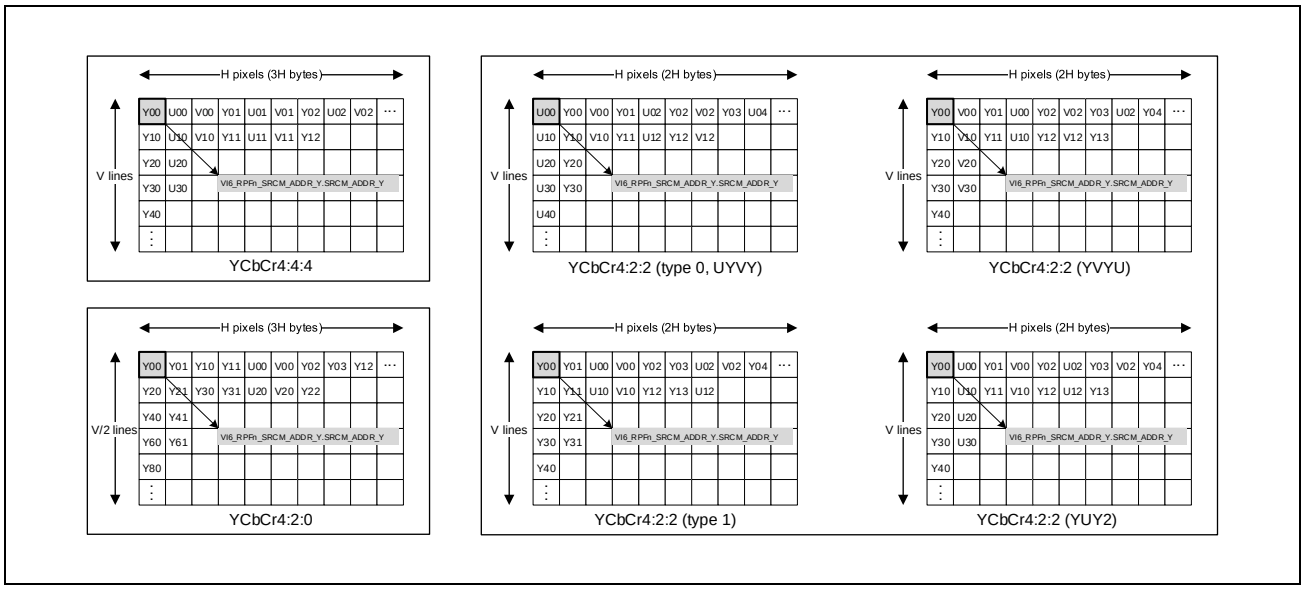



Figure 33.10 YCbCr Interleaved Formats*1

**Note 1.** This figure is for 8bpc.

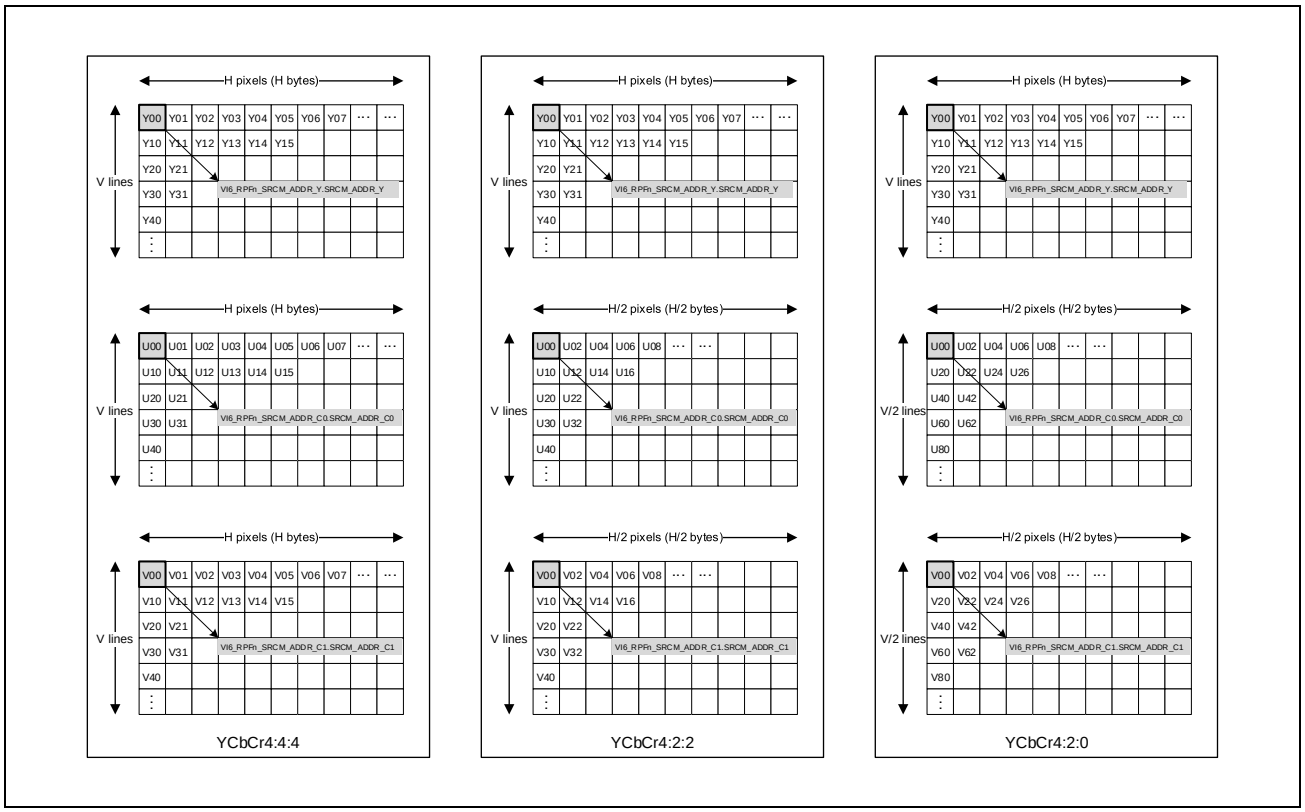



Figure 33.11 YCbCr Planar Formats*1

**Note 1.** This figure is for 8bpc.

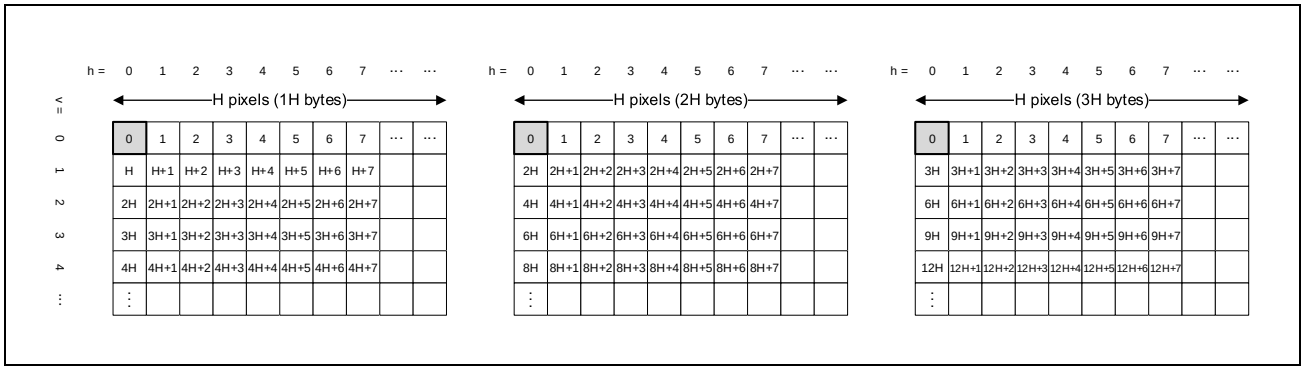



Figure 33.12 Memory address for the corresponding pixel's position

Legend of **Figure 33.12**

- H is horizontal image size in pixel unit.
- 1H, 2H and 3H is corresponding to the case 1 pixel has 1byte, 2bytes, 3bytes of data respectively.

### (4) RPFn Data Swapping Registers (VI6_RPFn_DSWAP)

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | A_LLS | A_LWS | A_WDS | A_BTS | — | — | — | — | P_LLS | P_LWS | P_WDS | P_BTS |
| 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 | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 12 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 11 | A_LLS | 0 | R/W | α Plane Data Swapping in LONG LWORD Units<br>The effect of this bit setting is defined in **Table 33.14**.<br>0: Data swapping in LONG LWORD (64-bit) units is disabled<br>1: Data swapping in LONG LWORD (64-bit) units is enabled |
| 10 | A_LWS | 0 | R/W | α Plane Data Swapping in long word Units<br>The effect of this bit setting is defined in **Table 33.14**.<br>0: Data swapping in long word (32-bit) units is disabled<br>1: Data swapping in long word (32-bit) units is enabled |
| 9 | A_WDS | 0 | R/W | α Plane Data Swapping in Word Units<br>The effect of this bit setting is defined in **Table 33.14**.<br>0: Data swapping in Word (16-bit) units is disabled<br>1: Data swapping in Word (16-bit) units is enabled |
| 8 | A_BTS | 0 | R/W | α Plane Data Swapping in Byte Units<br>The effect of this bit setting is defined in **Table 33.14**.<br>0: Data swapping in Byte (8-bit) units is disabled<br>1: Data swapping in Byte (8-bit) units is enabled |
| 7 to 4 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 3 | P_LLS | 0 | R/W | Picture Plane Data Swapping in LONG LWORD Units<br>The effect of this bit setting is defined in **Table 33.14**.<br>0: Data swapping in LONG LWORD (64-bit) units is disabled<br>1: Data swapping in LONG LWORD (64-bit) units is enabled<br>This register is available for only Luma plane (Y) in the case input data format is YUV Planar or Semi-Planar and input bit-depth is 2bypc (IPBD_Y != 0 or IPBD_C != 0), and available for both Luma plane (Y) and Chroma plane (CbCr/U/V) in other cases. |
| 2 | P_LWS | 0 | R/W | Picture Plane Data Swapping in long word Units<br>The effect of this bit setting is defined in **Table 33.14**.<br>0: Data swapping in long word (32-bit) units is disabled<br>1: Data swapping in long word (32-bit) units is enabled<br>This register is available for only Luma plane (Y) in the case input data format is YUV Planar or Semi-Planar and input bit-depth is 2bypc (IPBD_Y != 0 or IPBD_C != 0), and available for both Luma plane (Y) and Chroma plane (CbCr/U/V) in other cases. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 1 | P_WDS | 0 | R/W | Picture Plane Data Swapping in Word Units<br>The effect of this bit setting is defined in **Table 33.14**.<br>0: Data swapping in Word (16-bit) units is disabled<br>1: Data swapping in Word (16-bit) units is enabled<br>This register is available for only Luma plane (Y) in the case input data format is YUV Planar or Semi-Planar and input bit-depth is 2bypc (IPBD_Y != 0 or IPBD_C != 0), and available for both Luma plane (Y) and Chroma plane (CbCr/U/V) in other cases. |
| 0 | P_BTS | 0 | R/W | Picture Plane Data Swapping in Byte Units<br>The effect of this bit setting is defined in **Table 33.14**.<br>0: Data swapping in Byte (8-bit) units is disabled<br>1: Data swapping in Byte (8-bit) units is enabled<br>This register is available for only Luma plane (Y) in the case input data format is YUV Planar or Semi-Planar and input bit-depth is 2bypc (IPBD_Y != 0 or IPBD_C != 0), and available for both Luma plane (Y) and Chroma plane (CbCr/U/V) in other cases. |

When the virtual input function of the RPFn is used (VI6_RPFn_INFMT.VIR = 1), this register setting is ignored. Swapping of RPF input data can be specified separately for the α plane and picture plane.

**Table 33.14** shows the data order before and after swapping according to the long long word, long word, word, and byte swapping settings.

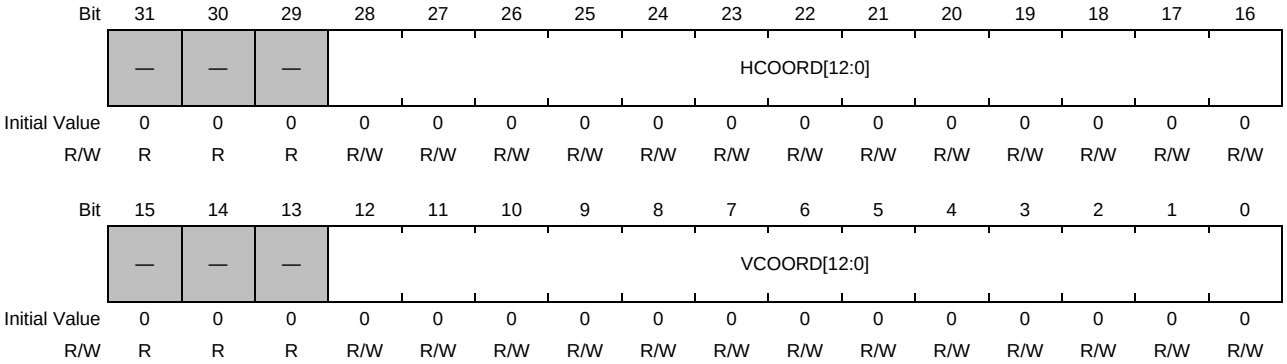
When data order in memory for each format is the same as **Table 33.12** for RGB format and **Figure 33.9** to **Figure 33.12** for YCbCr format, set 1111b to {*_LLS, *_LWS, *_WDS, *_BTS}. If data order is different from the definition, change data order within 16byte unit by these bits as shown in **Table 33.14**.

Table 33.14 Changing data order according to swap register

| Data order in memory | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Byte address | 16n+0 | 16n+1 | 16n+2 | 16n+3 | 16n+4 | 16n+5 | 16n+6 | 16n+7 | 16n+8 | 16n+9 | 16n+10 | 16n+11 | 16n+12 | 16n+13 | 16n+14 | 16n+15 | *_LLS | *_LWS | *_WDS | *_BTS |
| Data | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 1 | 1 | 1 | 1 |
| | 1 | 0 | 3 | 2 | 5 | 4 | 7 | 6 | 9 | 8 | 11 | 10 | 13 | 12 | 15 | 14 | 1 | 1 | 1 | 0 |
| | 2 | 3 | 0 | 1 | 6 | 7 | 4 | 5 | 10 | 11 | 8 | 9 | 14 | 15 | 12 | 13 | 1 | 1 | 0 | 1 |
| | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 11 | 10 | 9 | 8 | 15 | 14 | 13 | 12 | 1 | 1 | 0 | 0 |
| | 4 | 5 | 6 | 7 | 0 | 1 | 2 | 3 | 12 | 13 | 14 | 15 | 8 | 9 | 10 | 11 | 1 | 0 | 1 | 1 |
| | 5 | 4 | 7 | 6 | 1 | 0 | 3 | 2 | 13 | 12 | 15 | 14 | 9 | 8 | 11 | 10 | 1 | 0 | 1 | 0 |
| | 6 | 7 | 4 | 5 | 2 | 3 | 0 | 1 | 14 | 15 | 12 | 13 | 10 | 11 | 8 | 9 | 1 | 0 | 0 | 1 |
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 1 | 0 | 0 | 0 |
| | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 0 | 1 | 1 | 1 |
| | 9 | 8 | 11 | 10 | 13 | 12 | 15 | 14 | 1 | 0 | 3 | 2 | 5 | 4 | 7 | 6 | 0 | 1 | 1 | 0 |
| | 10 | 11 | 8 | 9 | 14 | 15 | 12 | 13 | 2 | 3 | 0 | 1 | 6 | 7 | 4 | 5 | 0 | 1 | 0 | 1 |
| | 11 | 10 | 9 | 8 | 15 | 14 | 13 | 12 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 0 | 1 | 0 | 0 |
| | 12 | 13 | 14 | 15 | 8 | 9 | 10 | 11 | 4 | 5 | 6 | 7 | 0 | 1 | 2 | 3 | 0 | 0 | 1 | 1 |
| | 13 | 12 | 15 | 14 | 9 | 8 | 11 | 10 | 5 | 4 | 7 | 6 | 1 | 0 | 3 | 2 | 0 | 0 | 1 | 0 |
| | 14 | 15 | 12 | 13 | 10 | 11 | 8 | 9 | 6 | 7 | 4 | 5 | 2 | 3 | 0 | 1 | 0 | 0 | 0 | 1 |
| | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 0 | 0 | 0 | 0 |

| **Table 33.12** for RGB format and **Figure 33.9** to **Figure 33.12** for YCbCr format | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Byte address | 16n+0 | 16n+1 | 16n+2 | 16n+3 | 16n+4 | 16n+5 | 16n+6 | 16n+7 | 16n+8 | 16n+9 | 16n+10 | 16n+11 | 16n+12 | 16n+13 | 16n+14 | 16n+15 |
| Data | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |

### (5) RPFn Display Location Registers (VI6_RPFn_LOC)




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 29 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 28 to 16 | HCOORD [12:0] | All 0 | R/W | Horizontal Coordinate of Sublayer Display Location on Master Layer<br>These bits specify the left-end location of the sublayer displayed by the RPFn and the subsequent module connected through the DPR. Specify the horizontal coordinate of the location in pixel units with the left-end pixel of the master layer set at coordinate 0. When the RPFn is the master layer, set these bits to 0.<br>If the sublayer extends beyond the master layer according to the HCOORD setting, the extended section is cut off at the right end of the master layer. Even in this case, however, a bus transfer that is unnecessary for output image generation is executed since the whole sublayer data is read from the external memory. Appropriate coordinate setting is required so that the sublayer does not extend beyond the right end of the master layer.<br>A value from 0 to 1919 can be specified. |
| 15 to 13 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 12 to 0 | VCOORD [12:0] | All 0 | R/W | Vertical Coordinate of Sublayer Display Location on Master Layer<br>These bits specify the top-end location of the sublayer displayed by the RPFn and the subsequent module connected through the DPR. Specify the vertical coordinate of the location in pixel units with the top-end pixel of the master layer set at coordinate 0. When the RPFn is the master layer, set these bits to 0.<br>If the sublayer extends beyond the master layer according to the VCOORD setting, the extended section is cut off at the bottom end of the master layer. Even in this case, however, a bus transfer that is unnecessary for output image generation is executed since the whole sublayer data is read from the external memory. Appropriate coordinate setting is required so that the sublayer does not extend beyond the bottom end of the master layer.<br>A value from 0 to 1079 can be specified. |

**Figure 33.13** shows an example of RPF1 offsets with respect to master layer RPF0. Although this figure only shows sublayers RPF1, specify offsets for all RPFs other than the master layer in the same way as shown in this example.

Whether an RPFn is the master layer or a sublayer is determined through the selection of the source RPF for WPFn (the VI6_WPFn_SRCRPF setting). For details, refer to **Section 33.3.2.7(1)**.

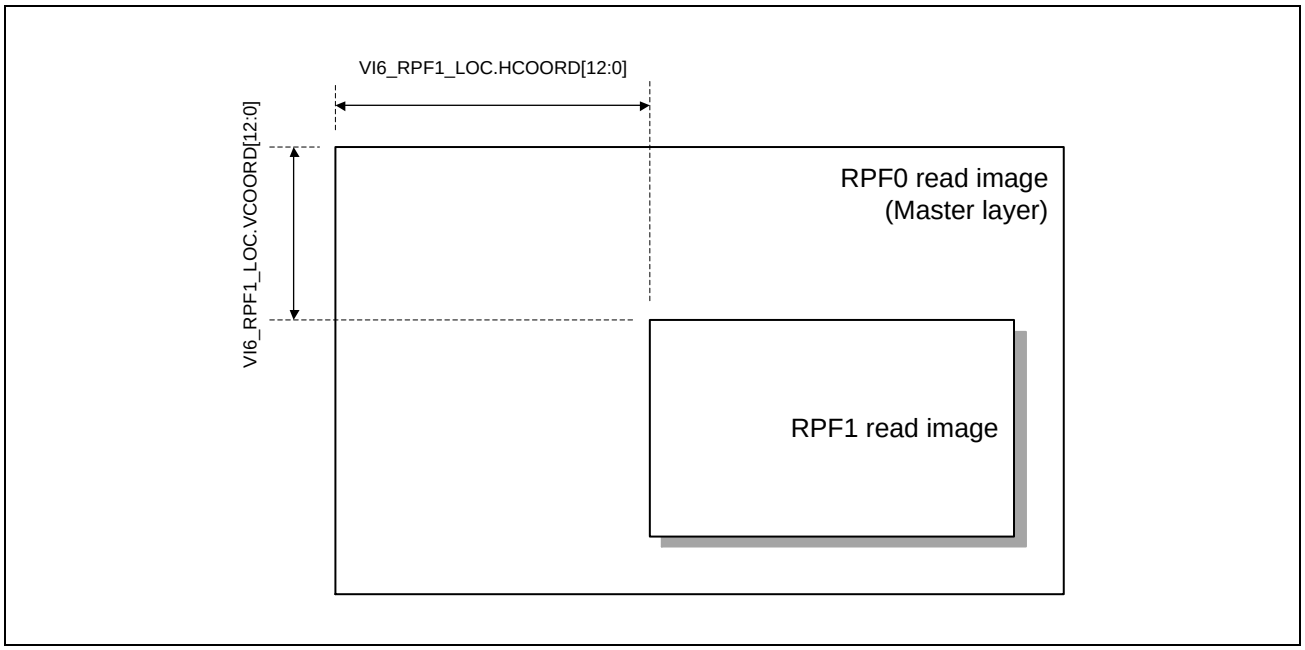
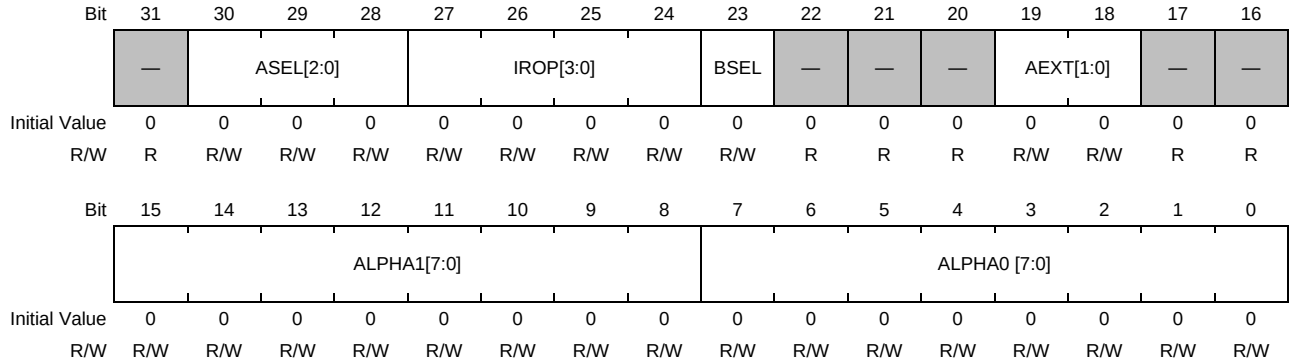



Figure 33.13 RPF1 Offsets from Master Layer

### (6) RPFn α Plane Selection Control Registers (VI6_RPFn_ALPH_SEL)




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 30 to 28 | ASEL[2:0] | All 0 | R/W | α Format and Processing Method Select<br>These bits select how to handle the α value to be used. The RPF handles two types of α value; 8-bit and 1-bit values. When a 1-bit α value is used, VSPD assumes that the 1-bpp α value for each pixel is stored in the order from MSB to LSB in each byte (big endian).<br>The α value is used as either transparency information or mask information. Transparency information is included in the α plane read from external memory when the ASEL bits are set to 1 or 3 and in the α value stored in the packed RGB bit field when these bits are set to 0 or 2. The α value as transparency information is sent as the destination value to the IROP as shown in **Figure 33.14** and then output to the subsequent modules. The output α value is used, for example, for blending in the BRS.<br>The α value as mask information is used for IROP operation in the RPF. The mask information is included in the α plane read from external RAM when the ASEL bits are set to 0 or 2 and the source value is used in IROP operation (IROP setting other than 0, 5, 10, or 15). This α value is sent as the source value to the IROP as shown in **Figure 33.14**.<br>Note that the α value selected through the ASEL bits has a lower priority than the VI6_RPFn_CKEY_SET*.AP* value replaced through the color keying function. When the color keying function is used, the α value may be replaced with the VI6_RPFn_CKEY_SET*.AP* value regardless of the ASEL bit setting.<br>When using the virtual input function (VI6_RPFn_INFMT.VIR = 1), specify 4.<br>0: 1, 4, or 8-bit packed α + plane α (IROP ! = 0, 5, 10, 15)<br>The α bit field in 1, 4, or 8-bit packed α is handled as transparency information. Be sure to specify the packed format that includes α through VI6_RPFn_INFMT.RDFMT.<br>When VI6_RPFn_MSKCTRL.MSK_EN is 0 and the IROP bit value is not 0, 5, 10, or 15, the α plane should be read as mask information. Specify the number of α data bits (BSEL) stored in the α plane and the α plane read start address (VI6_RPFn_SRCM_ADDR_AI). When the IROP bits are set to 0, 5, 10, or 15, the α plane is not read.<br>1: 8-bit plane α<br>The 8-bit α plane is read from external RAM as transparency information. When the packed RGB format has a bit field for α, the information in the α bit field is discarded. The α plane read start address (VI6_RPFn_SRCM_ADDR_AI) should be specified. The α value goes through the 8-bit transparent α generator shown in **Figure 33.14** without change.<br>When VI6_RPFn_MSKCTRL.MSK_EN is 0, IROP operation cannot be executed; set the IROP bits to 0 in this case. When VI6_RPFn_MSKCTRL.MSK_EN is 1. IROP operation can be executed.<br>2: 1-bit packed α + plane α (IROP ! = 0, 5, 10, 15)<br>The 1-bit packed α input is converted by the 8-bit transparent α generator shown in **Figure 33.14** according to the ALPHA0/1 setting into the 8-bit α value as transparency information. Select the packed input format that includes a 1-bit α field.<br>When VI6_RPFn_MSKCTRL.MSK_EN is 0 and the IROP value is not 0, 5, 10, or 15, the α plane should be read as mask information. Specify the number of α data bits (BSEL) stored in the α plane and the α plane read start address (VI6_RPFn_SRCM_ADDR_AI). When the IROP bits are set to 0, 5, 10, or 15, the α plane is not read.<br>3: 1-bit plane α<br>The 1-bit α plane is read from external RAM and converted by the 8-bit transparent α generator shown in **Figure 33.14**<br>according to the ALPHA0/1 setting into the 8-bit α value as transparency information. When the packed RGB format has a bit field for α, the information in the α bit field is discarded. The α plane read start address (VI6_RPFn_SRCM_ADDR_AI) should be specified.<br>When VI6_RPFn_MSKCTRL.MSK_EN is 0, IROP operation cannot be executed; set the IROP bits to 0 in this case. When VI6_RPFn_MSKCTRL.MSK_EN is 1. IROP operation can be executed.<br>4: Fixed α<br>The fixed α value (VI6_RPFn_VRTCOL_SET.LAYA value) is output from the RPF. IROP operation cannot be executed; set the IROP bits to 0 in this case.<br>5 to 7: Setting prohibited. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 27 to 24 | IROP [3:0] | All 0 | R/W | IROP Operation Setting<br>These bits specify the operator to be executed in the IROP operation unit shown in **Figure 33.14**. The source (S) for the IROP operation is the pixel data and α data specified in the VI6_RPFn_MSKSET0 or VI6_RPFn_MSKSET1 IROP input value register, which is selected according to the value (0 or 1) generated by the 1-bit mask generator. The destination (D) is the image data (RGB/YCbCr) and 8-bit α data output from the unpack/OSD processor. IROP operation is applied both for the image data and α data between the source and destination data.<br>If these bits are set to the operation that involves the source (S) (IROP setting other than 0, 5, 10, or 15) while VI6_RPFn_MSKCTRL.MSK_EN is 0, the α plane is read from the external RAM to be used for the α value for IROP operation; specify the α plane read start address (VI6_RPFn_SRCM_ADDR_A).<br>When the virtual input function is used (VI6_RPFn_INFMT.VIR = 1), IROP operation is not available; set these bits to 0000b.<br>0000b: NOP(D)<br>0001b: AND(S & D)<br>0010b: AND_REVERSE(S & ~D)<br>0011b: COPY(S)<br>0100b: AND_INVERTED(~S & D)<br>0101b: CLEAR(0)<br>0110b: XOR(S ^ D)<br>0111b: OR(S \| D)<br>1000b: NOR(~(S \| D))<br>1001b: EQUIV(~(S ^ D))<br>1010b: INVERT(~D)<br>1011b: OR_REVERSE(S \| ~D)<br>1100b: COPY_INVERTED(~S)<br>1101b: OR_INVERTED(~S \| D)<br>1110b: NAND(~(S & D))<br>1111b: SET(all 1) |
| 23 | BSEL | 0 | R/W | α Bit Count Conversion Selection for 1-Bit Mask Generator<br>Specifies the number of bits in the α plane to be read as mask information from the external RAM. The α value in mask information is used for the source (S) in IROP. When α plane data is 8 bits, it is converted to one bit through the 1-bit mask generator shown in **Figure 33.14**.<br>Note that this bit setting is valid when the ASEL bits are set to 0 or 2 and VI6_RPFn_MSKCTRL.MSK_EN is set to 0. In other cases, this bit setting has no effect.<br>0: 8-bit α is converted to 1-bit α through the 1-bit mask generator.<br>When the 8-bit α value input to the RPF is not 0, it is converted to 1b; when the value is 0, it is converted to 0b.<br>1: α value goes through the 1-bit mask generator.<br>The 1-bit α value input to the RPF is output through the 1-bit mask generator without change. |
| 22 to 20 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 19 to 18 | AEXT[1:0] | All 0 | R/W | Lower-Bit α Value Extension Method Set<br>These bits specify the method for extending the input α data to 8 bits through the unpack processing.<br>00b: The lower-order bits of α value are extended with 0.<br>01b: The upper-order bits of α value are copied to the lower-order bits.<br>10b: The lower-order bits of α value are extended with 0. The maximum value is limited to H'FF.<br>11b: Setting prohibited |
| 17, 16 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 to 8 | ALPHA1[7:0] | All 0 | R/W | 8-Bit α Value Output when 1-Bit α Value is 1<br>These bits specify the 8-bit α value to be output when 1-bit α data is input and the α value input to the 8-bit transparent α generator shown in **Figure 33.14** is 1b. This setting is valid when the ASEL bits are set to 010b or 011b.<br>A value from 0 to 255 can be specified. |
| 7 to 0 | ALPHA0 [7:0] | All 0 | R/W | 8-Bit α Value Output when 1-Bit α Value is 0<br>These bits specify the 8-bit α value to be output when 1-bit α data is input and the α value input to the 8-bit transparent α generator shown in **Figure 33.14** is 0b. This setting is valid when the ASEL bits are set to 010b or 011b.<br>A value from 0 to 255 can be specified. |

**Figure 33.14** shows the relationship between the α selector, IROP operation unit, color keying unit, and related registers. The IROP operation unit receives two inputs, source and destination. The image data input from the external memory is processed through the unpack processor and 8-bit transparent α generator and then input to the IROP operation unit as destination data. The α plane data input from the external memory is sent to the 8-bit transparent α generator when the ASEL bits are set to 1 or 3, or sent to the 1-bit mask α generator when the ASEL bits are set to 0 or 2. For the pixel data and 8-bit α value on the source side of the IROP operation unit, either the VI6_RPFn_MSKSET0 value or VI6_RPFn_MSKSET1 values will be selected according to the 1-bit α value output by the 1-bit mask α generator.

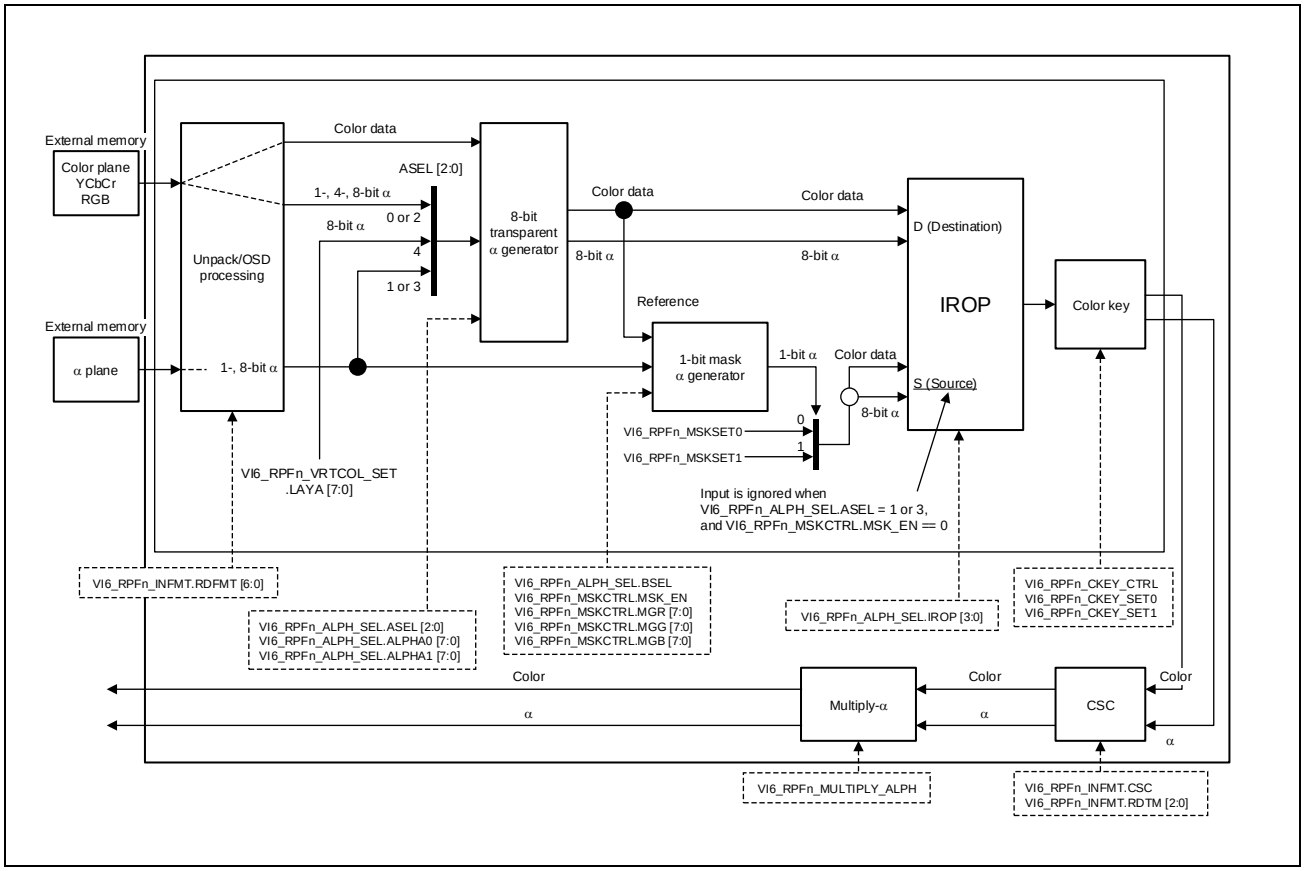



Figure 33.14 Configuration of alpha Selector and IROP Operation Unit in RPF

The following describes the function of each block shown in **Figure 33.14**. Read the following while referring to the figure as necessary.

**Unpack/OSD processor:**

Unpacks each component and α value of the image data according to the packed format specified in VI6_RPFn_INFMT.RDFMT.

**8-bit transparent α generator:**

Converts the input α value into 8-bit α when the input α is four bits or one bit.

When VI6_RPFn_ALPH_SEL.ASEL are set to 0 (8-, 4-, or 1-bit packed α), this generator outputs the input α value without change if the α bit field in the packed α data is 8 bits; if the α bit field is less than 8 bits, it is converted to an 8-bit α value by extending the LSB side according to the VI6_RPFn_ALPH_SEL.AEXT setting.

When VI6_RPFn_ALPH_SEL.ASEL are set to 1 (8-bit plane α) or 4 (fixed α), this generator outputs the input 8-bit plane α without change. If a packed α value is included in RGB data, it is discarded.

When VI6_RPFn_ALPH_SEL.ASEL are set to 2 (1-bit packed α) or 3 (8-bit α generated from 1-bit plane α), an 8-bit α value is generated by using the VI6_RPFn_ALPH_SEL.ALPHA0 [7:0] value when the input 1-bit α value is 0 or by using the VI6_RPFn_ALPH_SEL.ALPHA1 [7:0] value when the input 1-bit α value is 1. When VI6_RPFn_ALPH_SEL.ASEL is set to 3, a packed α value that is included in RGB data is discarded.

**1-bit mask α generator:**

Generates 1-bit α data from the input 8-bit α data or pixel data. When the input α data is one bit, this generator outputs it without change.

When VI6_RPFn_ALPH_SEL.ASEL are set to 0 (8-, 4-, or 1-bit packed α) or 2 (plane α) and VI6_RPFn_MSKCTRL.MSK_EN is set to 0, the α plane read from the external memory to be used in IROP is converted to 1-bit α data when necessary. When the α plane data read from the external RAM is 8 bits (BSEL = 0), if the value is 0, a 1-bit α value of 0b is generated; if the value is not 0, a 1-bit α value of 1b is generated. When the α plane data is one bit (BSEL = 1), this generator outputs it without change.

When the value of the 1-bit α generated by the 1-bit mask α generator is 0b, the 8-bit α and pixel data specified in VI6_RPFn_MSKSET0 are output as the source. When the generated 1-bit α value is 1b, the 8-bit α and pixel data specified in VI6_RPFn_MSKSET1 are output as the source.

As shown in **Figure 33.14**, when VI6_RPFn_ALPH_SEL.ASEL are set to 1 (8-bit plane α) or 3 (1-bit plane α), the α plane read from the external RAM is sent to the 8-bit transparent α generator as transparency information. When VI6_RPFn_MSKCTRL.MSK_EN is set to 0, the 1-bit α for masking is generated according to the input α plane (refer to **Section 33.3.2.6(8)**), but the 1-bit mask α generator does not refer to the input α plane because it is input to the 8-bit transparent α generator as transparency information. Accordingly, the 1-bit mask α generator does not generate a 1-bit α for masking and the data on the source side becomes invalid; that is, IROP operation cannot be executed. Set the IROP bits to 0 in this case. In contrast, when VI6_RPFn_MSKCTRL.MSK_EN is set to 1, the 1-bit mask α generator creates α data for masking according to the pixel data instead of the input α plane data, and IROP operation can be executed in this case.

**IROP operation unit:**

Executes ROP operation according to the opcode specified in VI6_RPFn_ALPH_SEL.IROP. For ROP operation (other than NOP), valid values should be input both for the source and destination. As described in the above (description of the 1-bit mask α generator), when VI6_RPFn_ALPH_SEL.ASEL are set to 1 or 3 and VI6_RPFn_MSKCTRL.MSK_EN is set to 0, the source data for the IROP operation unit is treated as invalid; set VI6_RPFn_ALPH_SEL.IROP to 0 (NOP). When VI6_RPFn_ALPH_SEL.ASEL are set to 4, a fixed α value is output from the RPF and IROP operation is not available. In the same way as the above case, set VI6_RPFn_ALPH_SEL.IROP to 0 (NOP).

To specify a valid source value for the IROP operation unit and execute IROP operation (specify an opcode other than NOP in the IROP bits), specify register values as shown in **Table 33.15**. Where the source input state is indicated as "Valid" in the table, IROP operation can be executed. In the cases where IROP operation is not available, set the IROP bits to 0 (NOP).

Table 33.15 Source Input State in IROP Operation Unit

| VI6_RPFn_ALPH_SEL.ASEL[2:0] | | VI6_RPFn_MSKCTRL.MSK_EN | | |
|---|---|---|---|---|
| | | 0 (Source data is generated according to input α plane) | | 1 (Source data is generated according to the destination-side pixel data) |
| 000b | (1-, 4-, or 8-bit packed α + plane α) | Valid | (α plane input) | Valid |
| 001b | (8-bit α plane) | Invalid | (IROP operation is not available; α plane is output to the subsequent modules behind RPF) | Valid |
| 010b | (8-bit α generated from 1-bit packed α + plane α) | Valid | (α plane input) | Valid |
| 011b | (8-bit α generated from 1-bit plane α) | Invalid | (IROP operation is not available; α plane is output to the subsequent modules behind RPF) | Valid |
| 100b | (Fixed α) | Invalid (IROP operation is not available; fixed α is output to the subsequent modules behind RPF) | | |

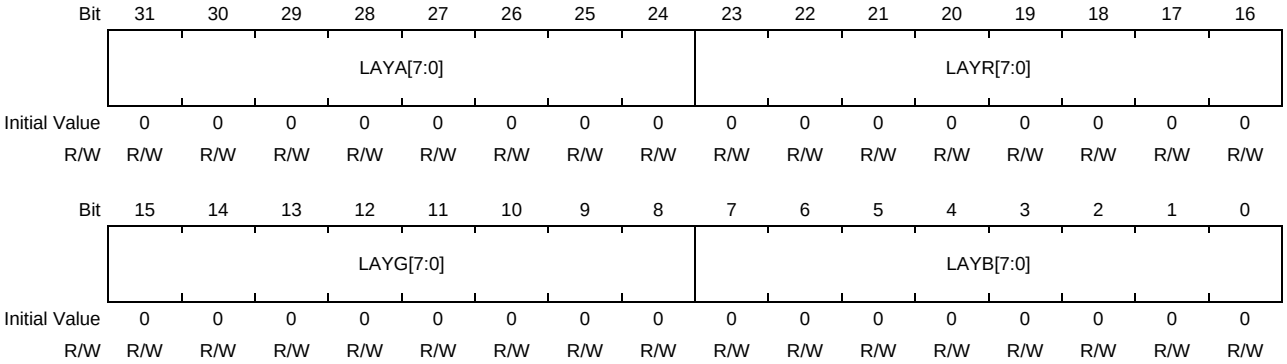
For the handling of the α values shown in **Figure 33.14** and **Table 33.15**, the relationship between the RPF input format and RPF output α value is shown in **Table 33.16**. Where only bit names are shown in the table, the bits are in VI6_RPFn_ALPH_SEL described in this section.

Table 33.16 α Value Selected and Output according to ASEL Bits in Each Input Format

| ASEL Setting | | α Value Output for Each Input Format | |
|---|---|---|---|
| | | RGB | YCbCr |
| 000b | (8-, 4-, or 1-bit packed α is input) | 1- ,4-, or 8-bit pixel α | H'FF*1 |
| 001b | (8-bit plane α is input) | 8-bit α plane | 8-bit α plane |
| 010b | (8-bit α is generated from the 1-bit packed α input) | ALPHA0 or ALPHA1 setting | H'FF*1 |
| 011b | (8-bit α is generated from the 1-bit plane α input) | ALPHA0 or ALPHA1 setting | ALPHA0 or ALPHA1 setting |
| 100b | (Fixed α is output) | VI6_RPFn_VRTCOL_SET.LAYA setting | |

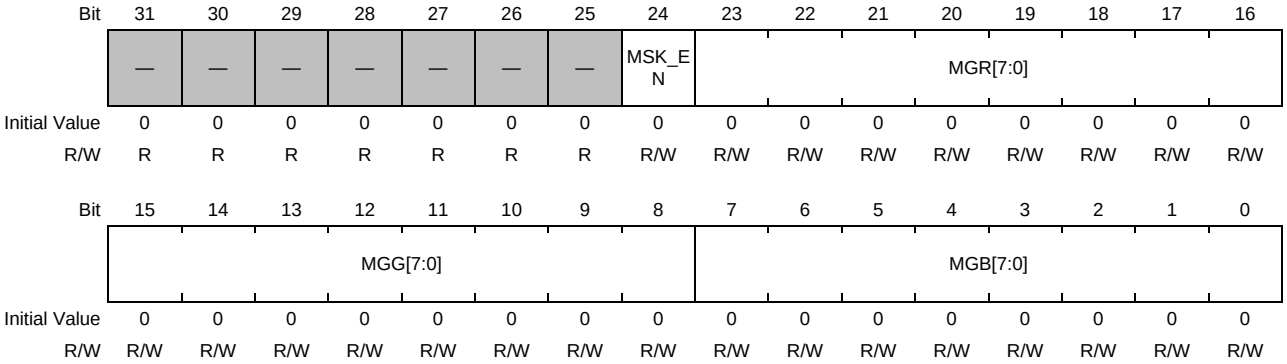
Note 1. Fixed value H'FF is output because packed α is not included in YCbCr.

### (7) RPFn Virtual Plane Color Information Registers (VI6_RPFn_VRTCOL_SET)



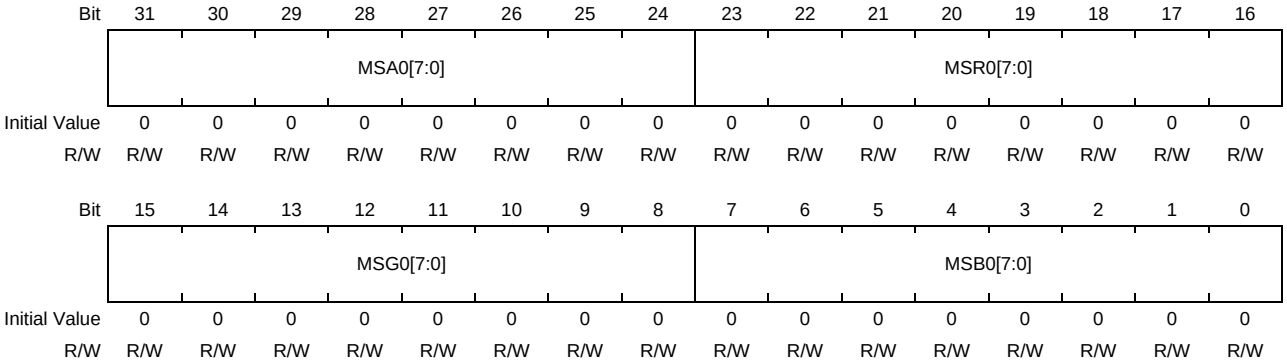

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 24 | LAYA[7:0] | All 0 | R/W | Virtual-Input Fixed α Value<br>These bits specify the fixed α value generated in the RPFn when the virtual input function is enabled through the VI6_RPFn_INFMT.VIR setting.<br>When the virtual input function is disabled (VI6_RPFn_INFMT.VIR = 0), these bits are used to specify the fixed α value to be output from the RPF while VI6_RPFn_ALPH_SEL.ASEL are set to 4. A value from 0 to 255 can be specified. |
| 23 to 16 | LAYR[7:0] | All 0 | R/W | Virtual-Input Fixed R/Cr Component Value<br>These bits specify the fixed R or Cr value generated in the RPFn when the virtual input function is enabled through the VI6_RPFn_INFMT.VIR setting. When VI6_RPFn_INFMT.RDFMT is set to 19, the value specified in these bits is used as the R value. When VI6_RPFn_INFMT.RDFMT is set to 64, the value specified in these bits is used as the Cr value. A value from 0 to 255 can be specified. |
| 15 to 8 | LAYG[7:0] | All 0 | R/W | Virtual-Input Fixed G/Y Component Value<br>These bits specify the fixed G or Y value generated in the RPFn when the virtual input function is enabled through the VI6_RPFn_INFMT.VIR setting. When VI6_RPFn_INFMT.RDFMT is set to 19, the value specified in these bits is used as the G value. When VI6_RPFn_INFMT.RDFMT is set to 64, the value specified in these bits is used as the Y value. A value from 0 to 255 can be specified. |
| 7 to 0 | LAYB[7:0] | All 0 | R/W | Virtual-Input Fixed B/Cb Component Value<br>These bits specify the fixed B or Cb value generated in the RPFn when the virtual input function is enabled through the VI6_RPFn_INFMT.VIR setting. When VI6_RPFn_INFMT.RDFMT is set to 19, the value specified in these bits is used as the B value. When VI6_RPFn_INFMT.RDFMT is set to 64, the value specified in these bits is used as the Cb value. A value from 0 to 255 can be specified. |

### (8) RPFn Mask Control Registers (VI6_RPFn_MSKCTRL)



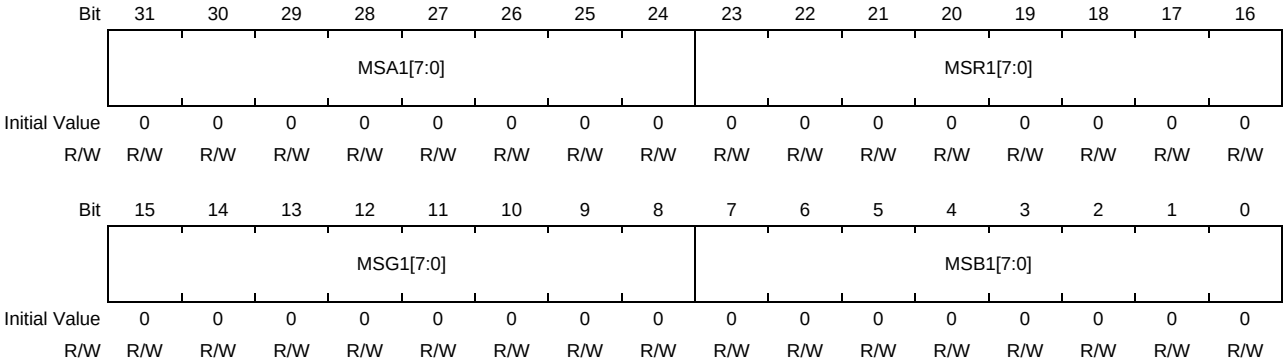

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 25 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 24 | MSK_EN | 0 | R/W | Mask Generation Specification<br>Specifies the method of α value generation in the 1-bit mask α generator shown in **Figure 33.14**.<br>0: A 1-bit mask value is generated according to the input α plane value. When the input α is in the 1-bit format (VI6_RPFn_ALPH_SEL.BSEL = 1), the 1-bit mask value is output without change. When the input α is in the 8-bit format (VI6_RPFn_ALPH_SEL.BSEL = 0), the 1-bit mask value is 0 if the α value is H'00; otherwise, the 1-bit mask value is 1.<br>1: The R/Cr, G/Y, and B/Cb components of the image input to the destination side of the IROP operation unit are compared with the values specified in the MGR, MGG, and MGB bits, respectively. When all values match, 1 is output as the 1-bit mask value, and in other cases, 0 is output. When the generated 1-bit mask data is not used, set VI6_RPFn_ALPH_SEL.IROP to 0. |
| 23 to 16 | MGR[7:0] | All 0 | R/W | R/Cr Comparison Value for 1-Bit α Generation<br>These bits specify the R/Cr value to be compared for 1-bit α generation by using the pixel data on the destination side. When RGB is specified as the RPFn input format through VI6_RPFn_INFMT.RDFMT, specify an R value for comparison. When YCbCr is specified, specify a Cr value for comparison.<br>This setting is ignored when the MSK_EN bit is set 0.<br>A value from 0 to 255 can be specified. |
| 15 to 8 | MGG[7:0] | All 0 | R/W | G/Y Comparison Value for 1-Bit α Generation<br>These bits specify the G/Y value to be compared for 1-bit α generation by using the pixel data on the destination side. When RGB is specified as the RPFn input format through VI6_RPFn_INFMT.RDFMT, specify a G value for comparison. When YCbCr is specified, specify a Y value for comparison.<br>This setting is ignored when MSK_EN is set to 0.<br>A value from 0 to 255 can be specified. |
| 7 to 0 | MGB[7:0] | All 0 | R/W | B/Cb Comparison Value for 1-Bit α Generation<br>These bits specify the B/Cb value to be compared for 1-bit α generation by using the pixel data on the destination side. When RGB is specified as the RPFn input format through VI6_RPFn_INFMT.RDFMT, specify a B value for comparison. When YCbCr is specified, specify a Cb value for comparison.<br>This setting is ignored when MSK_EN is set to 0.<br>A value from 0 to 255 can be specified. |

### (9) RPFn IROP-SRC Input Value Registers 0 (VI6_RPFn_MSKSET0)




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 24 | MSA0[7:0] | All 0 | R/W | IROP-Source Input α Value when 1-Bit α is 0<br>These bits specify the 8-bit α value to be input as the source to the IROP operation unit when the internal 1-bit α value generated through the 1-bit mask generator is 0 (**Figure 33.14**). A value from 0 to 255 can be specified. |
| 23 to 16 | MSR0[7:0] | All 0 | R/W | IROP-Source Input R/Cr Value when 1-Bit α is 0<br>These bits specify the R/Cr value to be input as the source to the IROP operation unit when the internal 1-bit α value generated through the 1-bit mask generator is 0 (**Figure 33.14**).<br>When RGB is specified as the RPFn input format through VI6_RPFn_INFMT.RDFMT, specify an R component value. When YCbCr is specified, specify a Cr component value. A value from 0 to 255 can be specified. |
| 15 to 8 | MSG0[7:0] | All 0 | R/W | IROP-Source Input G/Y Value when 1-Bit α is 0<br>These bits specify the G/Y value to be input as the source to the IROP operation unit when the internal 1-bit α value generated through the 1-bit mask generator is 0 (**Figure 33.14**).<br>When RGB is specified as the RPFn input format through VI6_RPFn_INFMT.RDFMT, specify a G component value. When YCbCr is specified, specify a Y component value. A value from 0 to 255 can be specified. |
| 7 to 0 | MSB0[7:0] | All 0 | R/W | IROP-Source Input B/Cb Value when 1-Bit α is 0<br>These bits specify the B/Cb value to be input as the source to the IROP operation unit when the internal 1-bit α value generated through the 1-bit mask generator is 0 (**Figure 33.14**).<br>When RGB is specified as the RPFn input format through VI6_RPFn_INFMT.RDFMT, specify a B component value. When YCbCr is specified, specify a Cb component value. A value from 0 to 255 can be specified. |

### (10) RPFn IROP-SRC Input Value Registers 1 (VI6_RPFn_MSKSET1)




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 24 | MSA1[7:0] | All 0 | R/W | IROP-Source Input α Value when 1-Bit α is 1<br>These bits specify the 8-bit α value to be input as the source to the IROP operation unit when the internal 1-bit α value generated through the 1-bit mask generator is 1. A value from 0 to 255 can be specified (**Figure 33.14**). |
| 23 to 16 | MSR1[7:0] | All 0 | R/W | IROP-Source Input R/Cr Value when 1-Bit α is 1<br>These bits specify the R/Cr value to be input as the source to the IROP operation unit when the internal 1-bit α value generated through the 1-bit mask generator is 1 (**Figure 33.14**).<br>When RGB is specified as the RPFn input format through VI6_RPFn_INFMT.RDFMT, specify an R component value. When YCbCr is specified, specify a Cr component value. A value from 0 to 255 can be specified. |
| 15 to 8 | MSG1[7:0] | All 0 | R/W | IROP-Source Input G/Y Value when 1-Bit α is 1<br>These bits specify the G/Y value to be input as the source to the IROP operation unit when the internal 1-bit α value generated through the 1-bit mask generator is 1 (**Figure 33.14**).<br>When RGB is specified as the RPFn input format through VI6_RPFn_INFMT.RDFMT, specify a G component value. When YCbCr is specified, specify a Y component value. A value from 0 to 255 can be specified. |
| 7 to 0 | MSB1[7:0] | All 0 | R/W | IROP-Source Input B/Cb Value when 1-Bit α is 1<br>These bits specify the B/Cb value to be input as the source to the IROP operation unit when the internal 1-bit α value generated through the 1-bit mask generator is 1 (**Figure 33.14**).<br>When RGB is specified as the RPFn input format through VI6_RPFn_INFMT.RDFMT, specify a B component value. When YCbCr is specified, specify a Cb component value. A value from 0 to 255 can be specified. |

### (11) RPFn Color Keying Control Registers (VI6_RPFn_CKEY_CTRL)

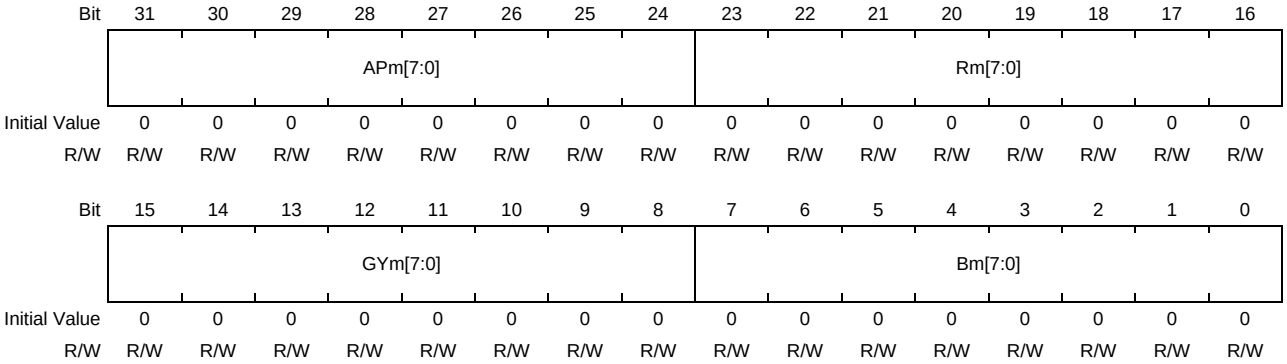
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | LTH | — | — | — | CV | — | — | SAPE1 | SAPE0 |
| 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/W | R | R | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 9 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 8 | LTH | 0 | R/W | Transparent color-luma Threshold Mode Enable/Disable<br>This bit enables or disables transparent color-luma threshold mode for the color keying module.<br>In transparent color-luma threshold mode, color information 0 (GY0) specified in VI6_RPFn_CKEY_SET0 register is compared with the input luma values.<br>When the input data is in YCbCr format, and if input Y value is equal to or smaller than specified color information (VI6_RPFn_CKEY_SET0.GY0), the input α value is replaced with the value specified in VI6_RPFn_CKEY_SET0.AP0. This bit is available only when the input data is in YCbCr format. When the input data is in an RGB format, set this bit to 0.<br>This bit setting is valid only when the CV bit is set to 0; it is ignored when the CV bit is set to 1. When using the virtual input function (VI6_RPFn_INFMT.VIR = 1), set this bit to 0.<br>0: Luma threshold mode is disabled (Matched color mode)<br>1: Luma threshold mode is enabled |
| 7 to 5 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 4 | CV | 0 | R/W | Color Replacement Control<br>This bit controls the color replacement function in the color keying module shown in **Figure 33.2**. When an RGB format is specified as the color space of the RPFn input data through VI6_RPFn_INFMT.RDFMT, and if all components of an input pixel match the color components specified in VI6_RPFn_CKEY_SET0, the color replacement function replaces the values of the input α and all RGB components with the α and color components specified in VI6_RPFn_CKEY_SET1. When a YCbCr format is specified as the color space of the RPFn input data through VI6_RPFn_INFMT.RDFMT, only the Y data is compared; if the luminance component of an input pixel matches the value specified in VI6_RPFn_CKEY_SET0.GY0, the color replacement function replaces the values of the input α and all YCbCr components with the α and color components specified in VI6_RPFn_CKEY_SET1.<br>When the CV bit is set to 1, the color replacement function is enabled. When using the virtual input function (VI6_RPFn_INFMT.VIR = 1), set this bit to 0.<br>0: Color replacement function is disabled (transparent color mode).<br>1: Color replacement function is enabled. |
| 3, 2 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 1 | SAPE1 | 0 | R/W | Comparison Color Data Setting 1 Enable/Disable<br>This bit enables or disables comparison color data setting 1 in the transparent color mode for the color keying module. This bit setting is valid only when the CV bit is set to 0; it is ignored when the CV bit is set to 1.<br>In transparent color mode, color information 1 (VI6_RPFn_CKEY_SET1.R1/GY1/B1) specified in VI6_RPFn_CKEY_SET1 is compared with the input component values. When the input data is in an RGB format, and if all input values match the specified color information, the input α value is replaced with the value specified in VI6_RPFn_CKEY_SET1.AP. When the input data is in YCbCr format, only the Y data is compared.<br>When using the virtual input function (VI6_RPFn_INFMT.VIR = 1), set this bit to 0.<br>0: Comparison color data setting 1 is disabled.<br>1: Comparison color data setting 1 is enabled. |
| 0 | SAPE0 | 0 | R/W | Comparison Color Data Setting 0 Enable/Disable<br>This bit enables or disables comparison color data setting 0 in the transparent color mode for the color keying module. This bit setting is valid only when the CV bit is set to 0; it is ignored when the CV bit is set to 1.<br>In transparent color mode, color information 0 (VI6_RPFn_CKEY_SET0.R0/GY0/B0) specified in VI6_RPFn_CKEY_SET0 is compared with the input component values. When the input data is in an RGB format, and if all input values match the specified color information, the input α value is replaced with the value specified in VI6_RPFn_CKEY_SET0.AP. When the input data is in YCbCr format, only the Y data is compared.<br>When using the virtual input function (VI6_RPFn_INFMT.VIR = 1), clear this bit to 0.<br>0: Comparison color data setting 0 is disabled.<br>1: Comparison color data setting 0 is enabled. |

### (12) RPFn Color Keying Color Setting Registers m (VI6_RPFn_CKEY_SETm : m = 0, 1)

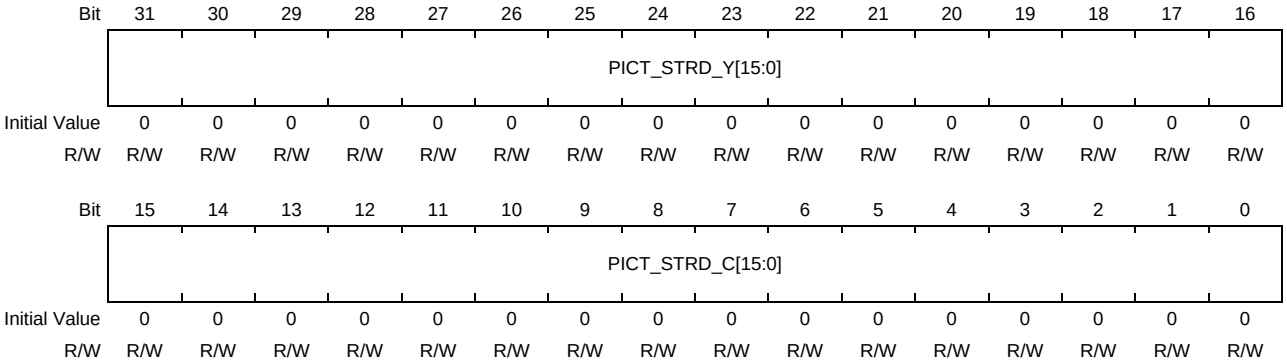



| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 24 | APm[7:0] | All 0 | R/W | α Data in Color Keying Color Information-m<br>● In transparent color-matched color mode for color keying<br>When the input data matches color setting-m (Rm, GYm, and Bm) for transparent color comparison in the color keying module, the input α value is replaced with the value specified in these bits. Specify the α value to replace the input value.<br>● In transparent color-luma threshold mode for color keying<br>When the input data is in YCbCr format, and if input Y value is equal to or smaller than GY0, the input α value is replaced with the value specified in this bit field (AP0).<br>Specify the α value in this bit field (AP0) to replace the input value.<br>AP1 is not used in this mode. Set AP1 to 0 value.<br>● In color replacement mode for color keying<br>These bits are not used in this mode. Clear them to 0.<br>α value replacement through these bits in transparent color mode for color keying takes priority over the α value selected through the VI6_RPFn_ALPH_SEL.ASEL setting.<br>A value from 0 to 255 can be specified. |
| 23 to 16 | Rm[7:0] | All 0 | R/W | R*1/Cr Component Data in Color Keying Color Information-m<br>● In transparent color-matched color mode for color keying<br>Specify the R component value for comparison enabled through the VI6_RPFn_CKEY_CTRL.SAPEm setting. When the RPFn input is in YCbCr format, the color keying module does not compare the Cr component, and the setting of these bits is ignored.<br>● In transparent color-luma threshold mode for color keying<br>The color keying module does not refer this bit field, and the setting of these bits is ignored.<br>● In color replacement mode for color keying<br>Specify the R component value to be compared with the input data in the color replacement function of the color keying module.<br>A value from 0 to 255 can be specified. |
| 15 to 8 | GYm[7:0] | All 0 | R/W | G*1/Y Component Data in Color Keying Color Information-m<br>● In transparent color-matched color mode for color keying<br>Specify the G/Y component value for comparison enabled through the VI6_RPFn_CKEY_CTRL.SAPEm setting.<br>● In transparent color-luma threshold mode for color keying<br>Specify the Y component value in the GY0 to be compared.<br>● In color replacement mode for color keying<br>Specify the G/Y component value to be compared with the input data in the color replacement function of the color keying module.<br>A value from 0 to 255 can be specified. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 to 0 | Bm[7:0] | All 0 | R/W | B*1/Cb Component Data in Color Keying Color Information-m<br>● In transparent color-matched color mode for color keying<br>Specify the B component value for comparison enabled through the VI6_RPFn_CKEY_CTRL.SAPEm setting. When the RPFn input is in YCbCr format, the color keying module does not compare the Cb component, and the setting of these bits is ignored.<br>● In transparent color-luma threshold mode for color keying<br>The color keying module does not refer this bit field, and the setting of these bits is ignored.<br>● In color replacement mode for color keying<br>Specify the B component value to be compared with the input data in the color replacement function of the color keying module.<br>A value from 0 to 255 can be specified. |

Note 1. When comparison data is specified in an RGB format, if a packed format is selected for RGB input and each of the RGB components is not 8 bits, the lower-order bits of input data are extended as specified through VI6_RPFn_INFMT.CEXT before comparison. The RGB components to be compared with the input should also be extended in the same way and the extended values should be specified in this register.

### (13) RPFn Source Picture Memory Stride Setting Registers (VI6_RPFn_SRCM_PSTRIDE)




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 16 | PICT_STRD_Y[15:0] | All 0 | R/W | Memory Stride of Source Picture Y/RGB Plane<br>These bits specify in 1-byte units the memory stride of the source picture Y/RGB plane read by the RPFn.<br>A value from 0 to 65535 can be specified. Refer to **Figure 33.15** for settings. |
| 15 to 0 | PICT_STRD_C[15:0] | All 0 | R/W | Memory Stride of Source Picture C Plane<br>These bits specify in 1-byte units the memory stride of the source picture C plane read by the RPFn. When an RGB-format picture is read, these bits do not need to be set.<br>A value from 0 to 65535 can be specified. Refer to **Figure 33.15** for settings.<br>In the YCbCr planar format, this setting is used as the memory stride of the Cb and Cr planes. |

This register specifies the memory stride of the picture planes in the source area as shown in **Figure 33.15**. The memory stride of the α plane should be specified through VI6_RPFn_SRCM_ASTRIDE.ALPH_STRD. When the RPF input is in an RGB format, only the RGB plane is read; when the input is in YCbCr format, the Y and C planes are read as shown in **Figure 33.15**. When the α plane is used, the α plane is also read. According to the image format and the necessity of the α plane, specify the necessary addresses where the source image is stored (VI6_RPFn_SRCM_ADDR_Y, VI6_RPFn_SRCM_ADDR_C, and VI6_RPFn_SRCM_ADDR_AI).

Whether the α plane needs to be read is determined according to the α plane selection method and IROP operation type. For details, refer to **Section 33.3.2.6(6)**.

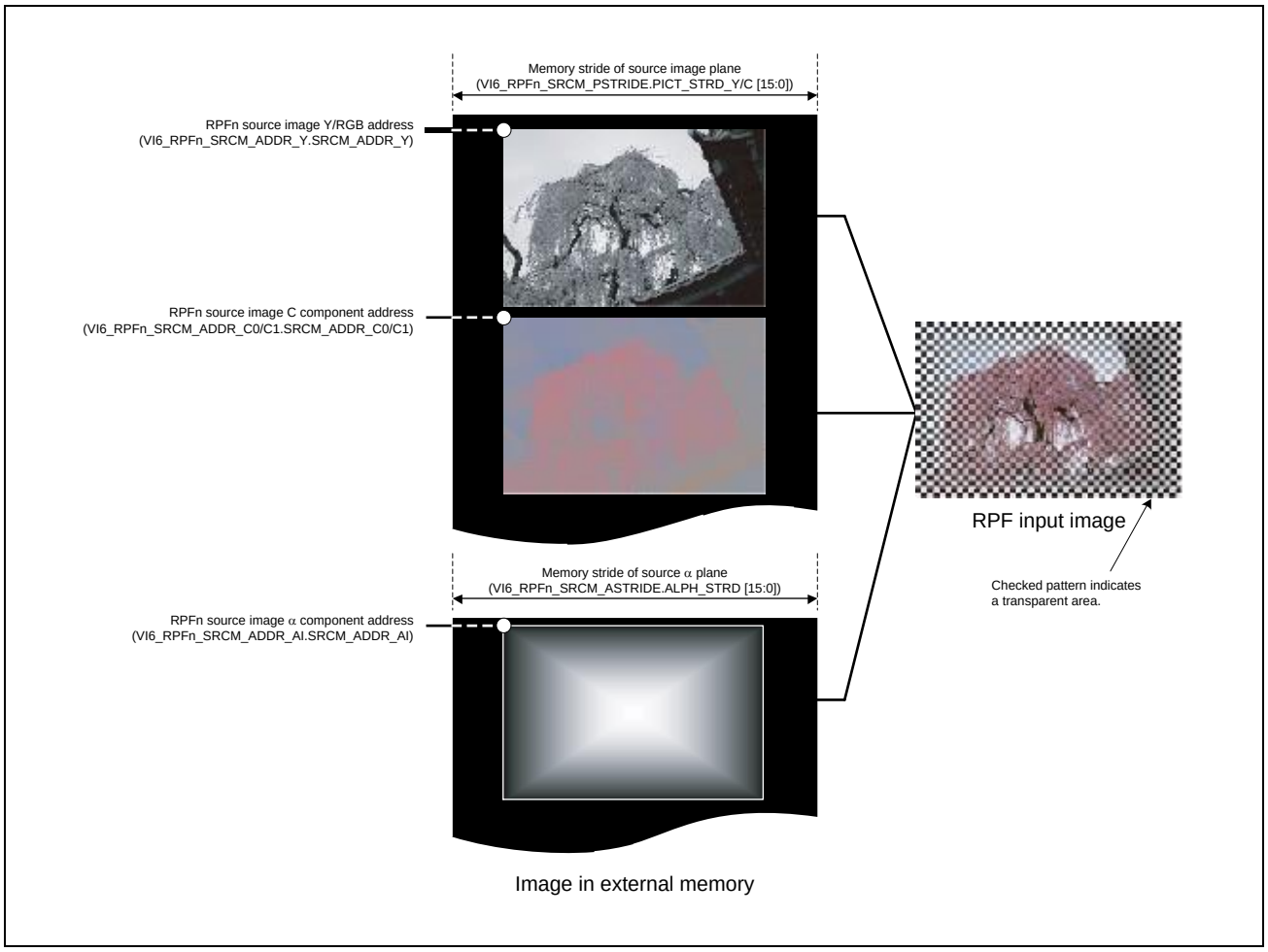



Figure 33.15 Reading an Image from RPFn Source Area

### (14) RPFn Source α Memory Stride Setting Registers (VI6_RPFn_SRCM_ASTRIDE)

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ALPH_STRD[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 16 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 15 to 0 | ALPH_STRD [15:0] | All 0 | R/W | Memory Stride of Source α Plane<br>These bits specify in 1-byte units the memory stride of the source α plane read by the RPFn. A value from 0 to 65535 can be specified. Refer to **Figure 33.15** for settings. |

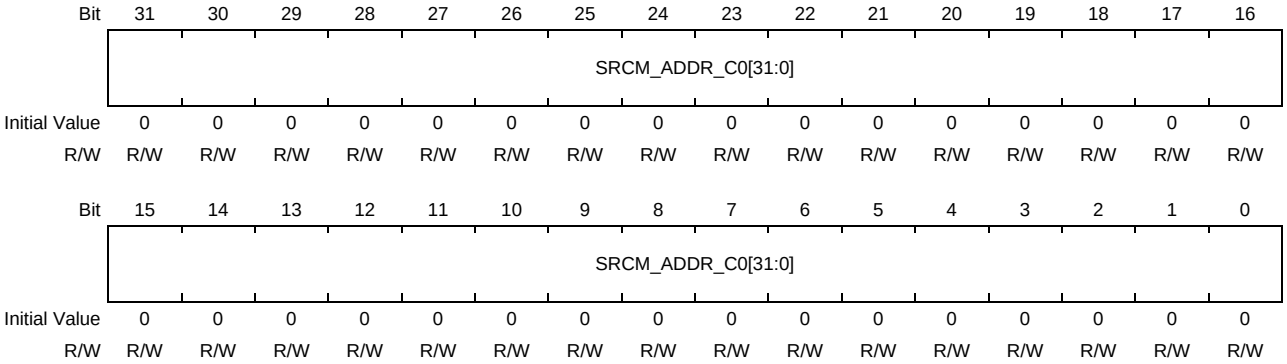
### (15) RPFn Source Y/RGB Address Registers (VI6_RPFn_SRCM_ADDR_Y)

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | SRCM_ADDR_Y[31:0] | | | | | | | | | | | | | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | SRCM_ADDR_Y[31:0] | | | | | | | | | | | | | | | |
| 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 |

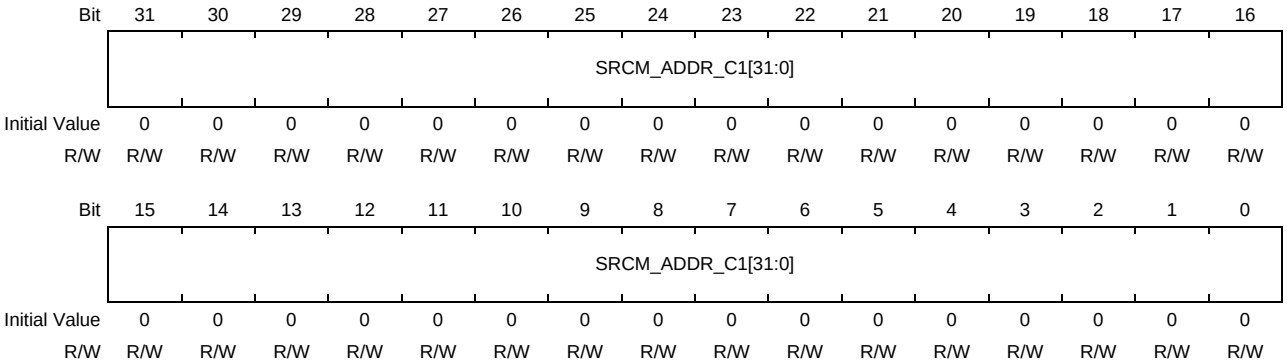
| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 0 | SRCM_ADDR _Y[31:0] | All 0 | R/W | Source Image Y/RGB Plane Storing Address<br>These bits specify in 1-byte units the start address of the source image Y plane and packed RGB plane read by the RPFn.<br>A value from H'0000 0000 to H'FFFF FFFF can be specified. Refer to **Figure 33.15** in **Section 33.3.2.6(13)** for settings. |

### (16) RPFn Source Chroma Address Registers 0 (VI6_RPFn_SRCM_ADDR_C0)



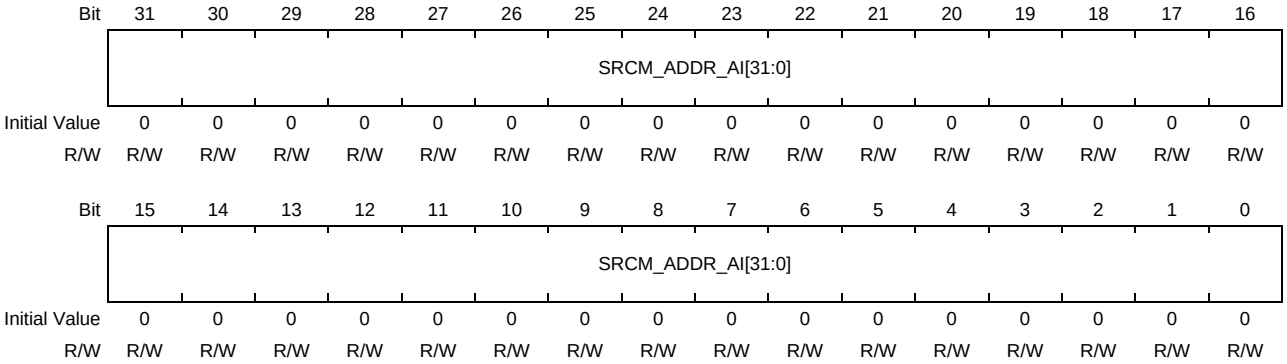

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 0 | SRCM_ADDR_C0[31:0] | All 0 | R/W | Source Image C Plane Storing Address 0<br>These bits specify in 1-byte units the start address of the source image C plane read by the RPFn. Here, the C plane indicates the combined CbCr plane when a semi-planar format is selected from the packed YCbCr formats shown in **Figure 33.15** or the Cb plane when a planar format is selected. When an interleaved format is selected or the RPF input is in an RGB format, this setting is not used.<br>A value from H'0000 0000 to H'FFFF FFFF can be specified. Refer to **Figure 33.15** in **Section 33.3.2.6(13)** for settings. |

### (17) RPFn Source Chroma Address Registers 1 (VI6_RPFn_SRCM_ADDR_C1)



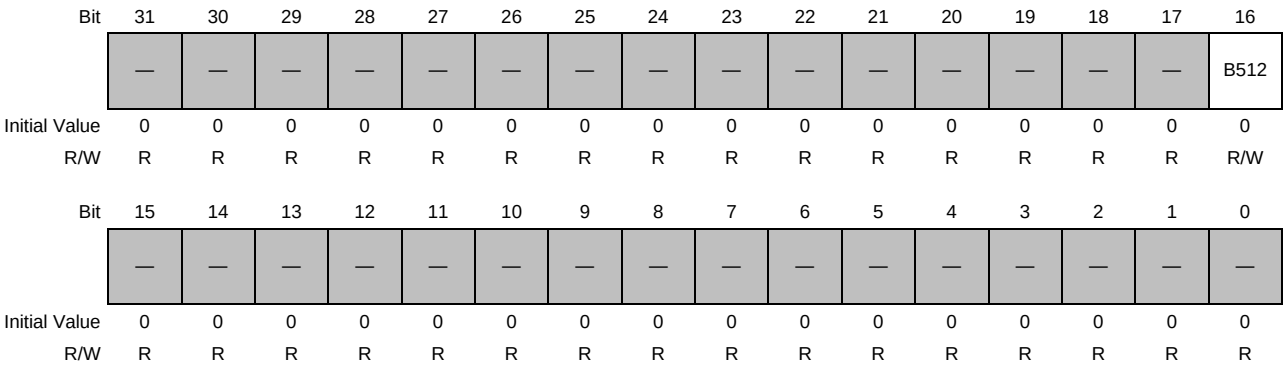

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 0 | SRCM_ADDR_C1[31:0] | All 0 | R/W | Source Image C Plane Storing Address 1<br>These bits specify in 1-byte units the start address of the Cr plane when a planar YCbCr format shown in **Figure 33.15** is read by the RPFn.<br>This setting is not used when the RPF input is in YCbCr format that is not a planar format or in an RGB format.<br>A value from H'0000 0000 to H'FFFF FFFF can be specified. Refer to **Figure 33.15** in **Section 33.3.2.6(13)** for settings. |

### (18) RPFn Source α Address Registers (VI6_RPFn_SRCM_ADDR_AI)




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 0 | SRCM_ADDR_AI[31:0] | All 0 | R/W | Source Image α Plane Storing Address<br>These bits specify in 1-byte units the start address of the α plane of the source image read by the RPFn. Specify in the same way as the start address of the picture plane. When the α plane is not read from the source area, these bits do not need to be set.<br>A value from H'0000 0000 to H'FFFF FFFF can be specified. Refer to **Figure 33.15** in **Section 33.3.2.6(13)** for settings. |

### (19) RPFn Bus Access Control Registers (VI6_RPFn_BAC)




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 17 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 16 | B512 | 0 | R/W | Burst access in 512 pixels' enable<br>0: burst access in 256 pixels<br>1: burst access in 512 pixels<br>Always set 0 to this bit |
| 15 to 0 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |

### (20) RPFn Multiple Alpha Control Registers (VI6_RPFn_MULT_ALPH)

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | A_MMD | — | — | P_MMD[1:0] | | ALPHA_RATIO[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/W | 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 13 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 12 | A_MMD | 0 | R/W | 0: Alpha data go through multiply-alpha unit.<br>1: Multiply-alpha unit multiplies alpha data by specified alpha (ALPHA_RATIO[7:0]).<br>When output format from CSC unit is YCbCr, set 0 to this bit. |
| 11, 10 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 9, 8 | P_MMD[1:0] | All 0 | R/W | 0: Image data go through multiply-alpha unit.<br>1: Multiply-alpha unit multiplies image data by specified alpha (ALPHA_RATIO[7:0]).<br>2: Multiply-alpha unit multiplies image data by alpha data.<br>3: Multiply-alpha unit multiplies image data by alpha data and specified alpha (ALPHA_RATIO[7:0]).<br>When output format from CSC unit is YCbCr, set 0 to this bit. |
| 7 to 0 | ALPHA_RATIO[7:0] | All 0 | R/W | Multiply alpha value |

ALPin: Input Alpha data to Multiply-alpha unit

PIXin(R): Input R data to Multiply-alpha unit

PIXin(G): Input G data to Multiply-alpha unit

PIXin(B): Input B data to Multiply-alpha unit

ALPout: Output Alpha data from Multiply-alpha unit

PIXout(R): Output R data from Multiply-alpha

PIXout(G): Output G data from Multiply-alpha

PIXout(B): Output B data from Multiply-alpha

Table 33.17 Expression of output alpha data from Multiply-alpha unit

| A_MMD | ALPHA_RATIO[7:0] | Expression |
|---|---|---|
| 0 | Don't care | ALPout = ALPin |
| 1 | Not 255 | ALPout = ALPin * ALPHA_RATIO / 256 |
| | 255 | ALPout = ALPin |

Table 33.18 Expression of output pixel data from Multiply-alpha unit

| P_MMD[1:0] | ALPHA_RATIO[7:0] | ALPin | Expression |
|---|---|---|---|
| 0 | Don't care | Don't care | PIXout(R) = PIXin(R)<br>PIXout(G) = PIXin(G)<br>PIXout(B) = PIXin(B) |
| 1 | Not 255 | Don't care | PIXout(R) = PIXin(R) * ALPHA_RATIO / 256<br>PIXout(G) = PIXin(G) * ALPHA_RATIO / 256<br>PIXout(B) = PIXin(B) * ALPHA_RATIO / 256 |
| | 255 | Don't care | PIXout(R) = PIXin(R)<br>PIXout(G) = PIXin(G)<br>PIXout(B) = PIXin(B) |
| 2 | Don't care | Not 255 | PIXout(R) = PIXin(R) * ALPin / 256<br>PIXout(G) = PIXin(G) * ALPin / 256<br>PIXout(B) = PIXin(B) * ALPin / 256 |
| | Don't care | 255 | PIXout(R) = PIXin(R)<br>PIXout(G) = PIXin(G)<br>PIXout(B) = PIXin(B) |
| 3 | Not 255 | Not 255 | PIXout(R) = PIXin(R) * ALPin * ALPHA_RATIO / 256 / 256<br>PIXout(G) = PIXin(G) * ALPin * ALPHA_RATIO / 256 / 256<br>PIXout(B) = PIXin(B) * ALPin * ALPHA_RATIO / 256 / 256 |
| | 255 | Not 255 | PIXout(R) = PIXin(R) * ALPin / 256<br>PIXout(G) = PIXin(G) * ALPin / 256<br>PIXout(B) = PIXin(B) * ALPin / 256 |
| | Not 255 | 255 | PIXout(R) = PIXin(R) * ALPHA_RATIO / 256<br>PIXout(G) = PIXin(G) * ALPHA_RATIO / 256<br>PIXout(B) = PIXin(B) * ALPHA_RATIO / 256 |
| | 255 | 255 | PIXout(R) = PIXin(R)<br>PIXout(G) = PIXin(G)<br>PIXout(B) = PIXin(B) |

### 33.3.2.7 WPF Control Registers

#### (1) WPFn-Source-RPF Registers (VI6_WPFn_SRCRPF)

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | VIR_ACT2[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/W | R/W | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | RPF1_ACT[1:0] | | RPF0_ACT[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/W | R/W | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 26 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 25, 24 | VIR_ACT2 [1:0] | All 0 | R/W | Virtual RPF Start Enable in BRS<br>These bits enable start of the virtual RPF in the BRS as the source RPF for the WPF0 when the WPF0 is started. For details of the virtual RPF, refer to the following.<br>**Section 33.3.2.10(1) BRS Input Control Register (VI6_BRS_INCTRL)**<br>**Section 33.3.2.10(2) Size Register of BRS Input Virtual RPF (VI6_BRS_VIRRPF_SIZE)**<br>**Section 33.3.2.10(3) Display Location Register of BRS Input Virtual RPF (VI6_BRS_VIRRPF_LOC)**<br>**Section 33.3.2.10(4) Color Information Register of BRS Input Virtual RPF (VI6_BRS_VIRRPF_COL)**<br>Note that the virtual RPF is in the BRS as shown in **Figure 33.28** and there are no register bits for DPR setting related to the virtual RPF.<br>0: The virtual RPF in the BRS is not started.<br>1: The virtual RPF in the BRS is started as a sublayer source RPF for the WPFn.<br>2: The virtual RPF in the BRS is started as the master-layer source RPF for the WPFn.<br>3: Setting prohibited |
| 23 to 4 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 3, 2 | RPF1_ACT [1:0] | All 0 | R/W | RPFn Start Enable (RPFn_ACT, n = 0 to 1)<br>These bits enable start of RPFn as the source RPF for the WPFn when the WPFn is started. When RPFn is not started by any of the WPFn (n = 0), set the VI6_DPR_RPFn_ROUTE.RT_RPFn bits to D'63.<br>0: RPFn is not started.<br>1: RPFn is started as a sublayer source RPF for the WPFn.<br>2: RPFn is started as the master-layer source RPF for the WPFn.<br>3: Setting prohibited |
| 1, 0 | RPF0_ACT [1:0] | All 0 | R/W | |

When the WPFn is started through the VSPD start register n (VI6_CMDn: n = 0), the RPF and virtual RPF in the BRS specified as the source RPF in this register are also started to supply data to the VSPD internal modules.

Note the following when specifying the source RPF.

(1) When blending or ROP operation is applied to multiple images through the BRS, multiple source RPFs are necessary for one WPF. When multiple source RPFs are used, images should be classified into a master layer and sublayers; assign one of the source RPFs as the master-layer source RPF and other RPFs as sublayer source RPFs.

Do not assign all RPFs as sublayer source RPFs (VI6_WPF1_SRCRPF = H’00000015) or two or more RPFs as the master-layer source RPF (VI6_WPF0_SRCRPF = H’0000025A) (such settings are prohibited).

(2) When the BRS is not used, there should be only one source RPF for one WPF. In this case, the source RPF should be assigned as the master-layer source RPF.

### (2) WPFn Horizontal Input Size Clipping Registers (VI6_WPFn_HSZCLIP)

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | HCEN | — | — | — | — | HCL_OFST[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/W | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | HCL_SIZE[12:0] | | | | | | | | | | | | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R/W | 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 29 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 28 | HCEN | 0 | R/W | Horizontal Size Clipping Enable/Disable<br>Enables or disables clipping of the horizontal size of the WPFn input image.<br>0: Horizontal size clipping is disabled<br>1: Horizontal size clipping is enabled |
| 27 to 24 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 23 to 16 | HCL_OFST [7:0] | All 0 | R/W | Horizontal Size Clipping Offset Value Setting<br>These bits specify the offset size (pixels) from the left end of the image in horizontal size clipping when the HCEN bit is 1 (**Figure 33.16**). The left side of the image input to the WPF is cut off for the size specified in these bits. When the HCEN bit is 0, this setting is ignored.<br>A value from 0 to 255 can be specified. (HCL_OFST + HCL_SIZE) should not exceed the horizontal size of the WPF input. If the setting shown in the bottom example in **Figure 33.16** is made, VSPD does not operate correctly. |
| 15 to 13 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 12 to 0 | HCL_SIZE [12:0] | All 0 | R/W | Horizontal Clipping Size Setting<br>When the HCEN bit is 1, these bits specify the clipping size for horizontal clipping processing. Through this processing, the area of the horizontal size specified through the HCL_SIZE bits starting from the offset position specified through the HCL_OFST bits is determined as the valid image area. Accordingly, the right-side pixels beyond the (HCL_OFST + HCL_SIZE) size in the WPFn input image are discarded. When the HCEN bit is 0, this setting is ignored.<br>A value from 1 to 1280 can be specified. (HCL_OFST + HCL_SIZE) should not exceed the horizontal size of the WPF input. If the setting shown in the bottom example in **Figure 33.16** is made, VSPD does not operate correctly.<br>(Note)<br>When the WPFn output format is YCbCr4:2:2 or YCbCr4:2:0, specify an even value in these bits. |

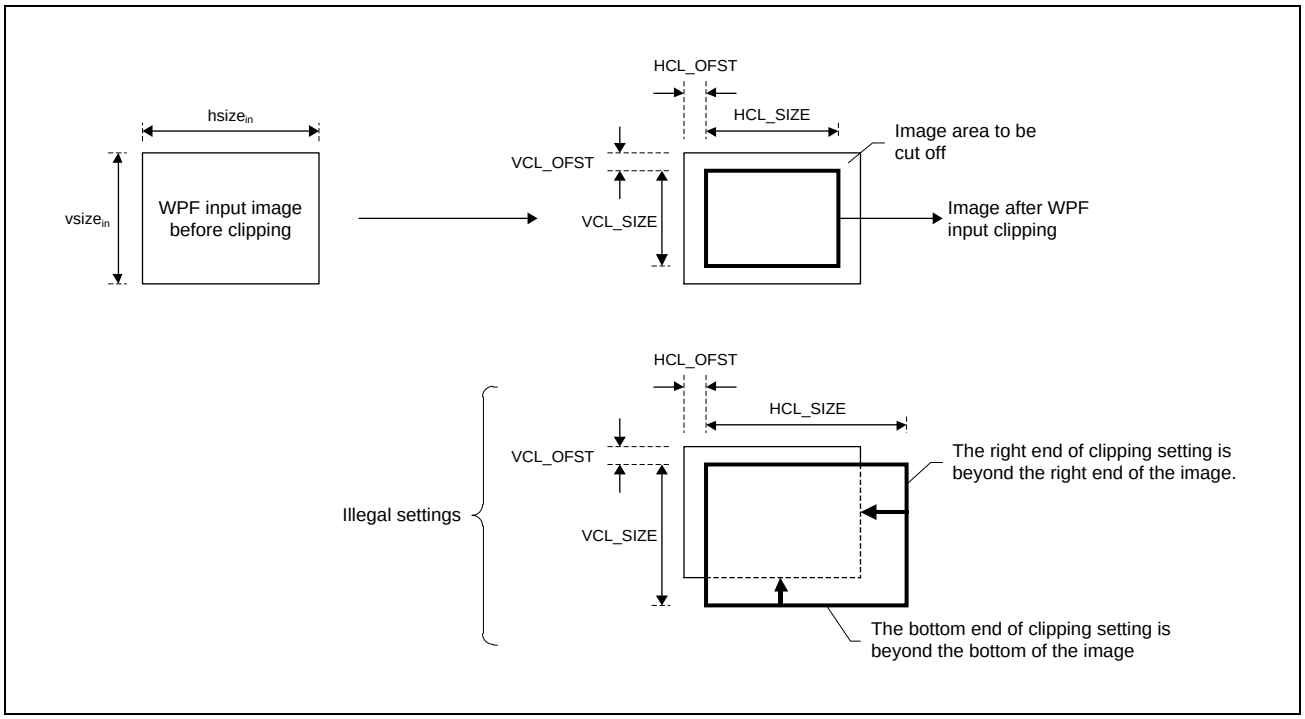
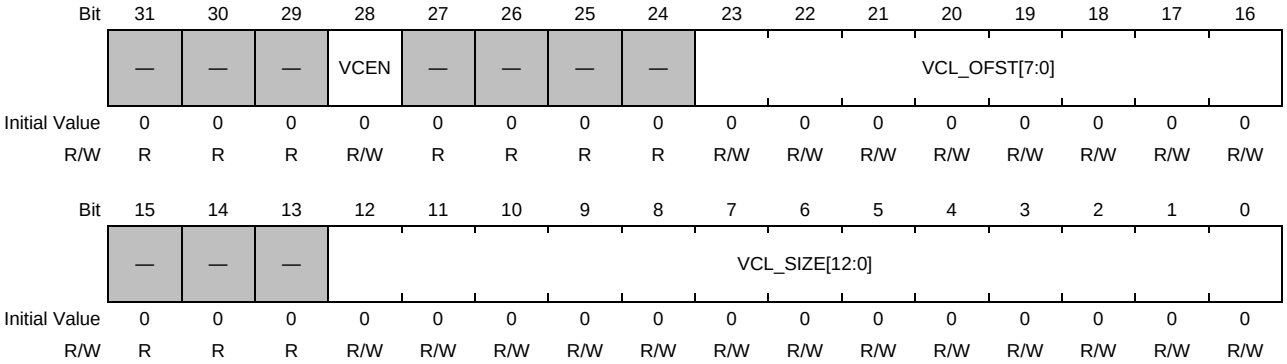



Figure 33.16 Image Clipping in WPF Input Section

### (3) WPFn Vertical Input Size Clipping Registers (VI6_WPFn_VSZCLIP)




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 29 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 28 | VCEN | 0 | R/W | Vertical Size Clipping Enable/Disable<br>Enables or disables clipping of the vertical size of the WPFn input image.<br>0: Vertical size clipping is disabled<br>1: Vertical size clipping is enabled |
| 27 to 24 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 23 to 16 | VCL_OFST [7:0] | All 0 | R/W | Vertical Size Clipping Offset Value Setting<br>These bits specify the offset size (pixels) from the top end of the image in vertical size clipping when the VCEN bit is 1 (**Figure 33.16**). The top of the image input to the WPF is cut off for the size specified in these bits. When the VCEN bit is 0, this setting is ignored.<br>A value from 0 to 255 can be specified. (VCL_OFST + VCL_SIZE) should not exceed the vertical size of the WPF input. If the setting shown in the bottom example in **Figure 33.16** is made, VSPD does not operate correctly. |
| 15 to 13 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 12 to 0 | VCL_SIZE [12:0] | All 0 | R/W | Vertical Clipping Size Setting<br>When the VCEN bit is 1, these bits specify the clipping size for vertical clipping processing. Through this processing, the area of the vertical size specified through the VCL_SIZE bits starting from the offset position specified through the VCL_OFST bits is determined as the valid image area. Accordingly, the bottom pixels beyond the (VCL_OFST + VCL_SIZE) size in the WPFn input image are discarded. When the VCEN bit is 0, this setting is ignored.<br>A value from 1 to 800 can be specified. (VCL_OFST + VCL_SIZE) should not exceed the vertical size of the WPF input. If the setting shown in the bottom example in **Figure 33.16** is made, VSPD does not operate correctly.<br>(Note)<br>When the WPFn output format is YCbCr4:2:0, specify an even value in these bits. |

### (4) WPFn Output Format Registers (VI6_WPFn_OUTFMT)

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PDV [7:0] | | | | | | | | PXA | ODE | — | — | — | ROT[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/W | R/W | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | SPYCS | SPUVS | DITH[1:0] | | WRTM[2:0] | | | CSC | — | WRFMT[6:0] | | | | | | |
| 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/W | R/W | R/W | R/W | R/W | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 24 | PDV [7:0] | All 0 | R/W | PAD Value in Output Packed Data<br>These bits specify the value to be stored in the bit field indicated as PAD or P in the output formats shown in **Table 33.19**. To store this value in PAD, specify 0 in the PXA bit. A value from 0 to 255 can be specified. |
| 23 | PXA | 0 | R/W | PAD Data Select<br>Selects the value to be stored in the bit field indicated as PAD or P in the packed RGB output formats shown in **Table 33.20**. Both the value specified in the PDV bits and the α data input from the DPR to WPF are 8 bits, but some of the PAD and P bit fields shown in **Table 33.19** are four bits or one bit. When the target bit field is not 8 bits, the number of bits in the PDV value and the α data input from the DPR to WPF is reduced according to the VI6_WPFn_RNDCTRL.ABRM setting. For bit count reduction, refer to **Figure 33.17** and the description of VI6_WPFn_RNDCTRL.ABRM.<br>0: The value specified in the PDV bits is stored in the PAD shown in **Table 33.19**.<br>1: The α value output from DPR in pixel units is stored in the PAD shown in **Table 33.19**. |
| 22 | ODE | 0 | R/W | Ordered Dither (mode A) Enable/Disable<br>0: Ordered dither (mode A) is disabled.<br>1: Ordered dither (mode A) is enabled.<br>When the output format specified through the WRFMT bits is YCbCr, specify 0 in this bit. And when VI6_WPFn_OUTFMT.CSC is set to 1, specify 0 in this bit even in the case that the output format specified through the WRFMT bits is RGB.<br>Ordered dither is available only for 18bpp. So, when ODE bit is equal to 1, set WRFMT at 18bpp format.<br>When ODE bit is equal to 0, WPF dither method is specified by DITH[1:0] in the register<br>Ordered dither (mode A) is recommended rather than ordered dither (mode B) in case of 18bpp. |
| 21 to 19 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 18 to 16 | ROT[2:0] | All 0 | R/W | Rotation Processing Select<br>These bits select the flipping processing to be applied to the WPFn output image. **Figure 33.18** shows the correspondence between the original image and the flipping result according to each setting.<br>0: No flipping<br>1: Vertical flipping<br>Note that the destination address setting should be changed according to the setting of these bits. For details, refer to **Section 33.3.2.7(9)**.<br>When the LIF module is used (VI6_LIF_CTRL.LIF_EN = 1), set 0 to ROT[2:0]. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 | SPYCS | 0 | R/W | WPF Output Mode Setting 1<br>When the output format is YUY2, set this bit to 1 and set the WRFMT bits to 71 (H'47).<br>When the output format is YVYU, set this bit and the SPUVS bit to 1 and set the WRFMT bits to 71 (H'47).<br>In other cases, set this bit to 0. |
| 14 | SPUVS | 0 | R/W | WPF Output Mode Setting 2<br>When the output format is NV61, set this bit to 1 and set the WRFMT bits to 65 (H'41).<br>When the output format is NV21, set this bit to 1 and set the WRFMT bits to 66 (H'42).<br>When the output format is YVYU, set this bit and the SPYCS bit to 1 and set the WRFMT bits to 71 (H'47).<br>In other cases, set this bit to 0. |
| 13, 12 | DITH[1:0] | All 0 | R/W | Ordered Dither (mode B) Enable/Disable<br>When the output format specified through the WRFMT bits is RGB with 18 bpp (260000 colors) or less, the color reduction processing is applied to match the number of colors. The color reduction processing may generate the artifacts of pseudo gradation, which can be suppressed through dithering. The DITH bits enable or disable dithering during color reduction.<br>When the output format specified through the WRFMT bits is YCbCr, specify 0 in these bits.<br>And when VI6_WPFn_OUTFMT.CSC is set to 1, specify 0 in these<br>bits even in the case that the output format specified through the<br>WRFMT bits is RGB.<br>0: Dithering (mode B) is disabled<br>3: Dithering (mode B) is enabled<br>1, 2: Setting prohibited<br>When ODE bit in the register is 1, set this bit to 0. |
| 11 to 9 | WRTM[2:0] | All 0 | R/W | CSC Conversion Expression Setting<br>These bits select the expression for color space conversion. The conversion direction is RGB to YCbCr when the format specified in the WRFMT bits is RGB, or YCbCr to RGB when the format is YCbCr.<br>0: BT.601 YCbCr [16,235/240] ↔ RGB [0,255]<br>1: BT.601 YCbCr [0,255] ↔ RGB [0,255]<br>2: BT.709 YCbCr [16,235/240] ↔ RGB [0,255]<br>3: BT.709 YCbCr [16,235/240] ↔ RGB [16,235]<br>4 to 7: Setting prohibited |
| 8 | CSC | 0 | R/W | Color Space Conversion Setting<br>Enables or disables YCbCr ↔ RGB color space conversion to be executed in the WPFn. The characteristics of color space conversion are determined by the WRTM setting.<br>There are some points to be noted about the relationship between the CSC setting and output format (WRFMT). For details refer to (*1) in **Section 33.3.2.6(3)**, RPFn Input Format Registers (VI6_RPFn_INFMT).<br>0: Color space is not converted.<br>1: Color space is converted. |
| 7 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 6 to 0 | WRFMT[6:0] | All 0 | R/W | WPF Output Image Format Setting<br>These bits select the format of the image output from the WPFn to the external memory from among those listed in **Table 33.19** and **Table 33.20**.<br>[Note 1] Number of output pixels<br>When YCbCr4:2:2 is specified through WRFMT, the horizontal size of the output image should be a multiple of 2 pixels. When YCbCr4:2:0 is specified, the vertical and horizontal sizes of the output image should be multiples of 2 pixels. Specify an appropriate data flow of the source RPF -> DPR -> target WPF so that the size of the image input to the target WPF satisfies the above restrictions. In particular, when the data flow includes a module or a function that modifies (up-scales, down-scales, or clips) the image size, take special care about the module or function settings.<br>[Note 2] Output lines in YCbCr4:2:0<br>In the YCbCr4:2:0 output format, the number of chrominance lines in the vertical direction is one-half the number of luminance lines. For this reason, the WPF outputs only even-numbered chrominance lines (lines 0, 2, 4, 6, ...) (conversion from (A) to (B) in. When vertical flipping is also specified through the ROT bits, the flipping processing is executed last and the chrominance line locations are inverted (lines 1, 3, 5, 7, …) in the output image ((C) in **Figure 33.19**).<br>[Note 3] Down sampling of CbCr in horizontal direction in YCbCr4:2:0 or YCbCr4:2:2<br>In the YCbCr4:2:2 or YCbCr4:2:0 output format, method of down sample of Cb/Cr in horizontal direction is average of neighbor two pixels. |

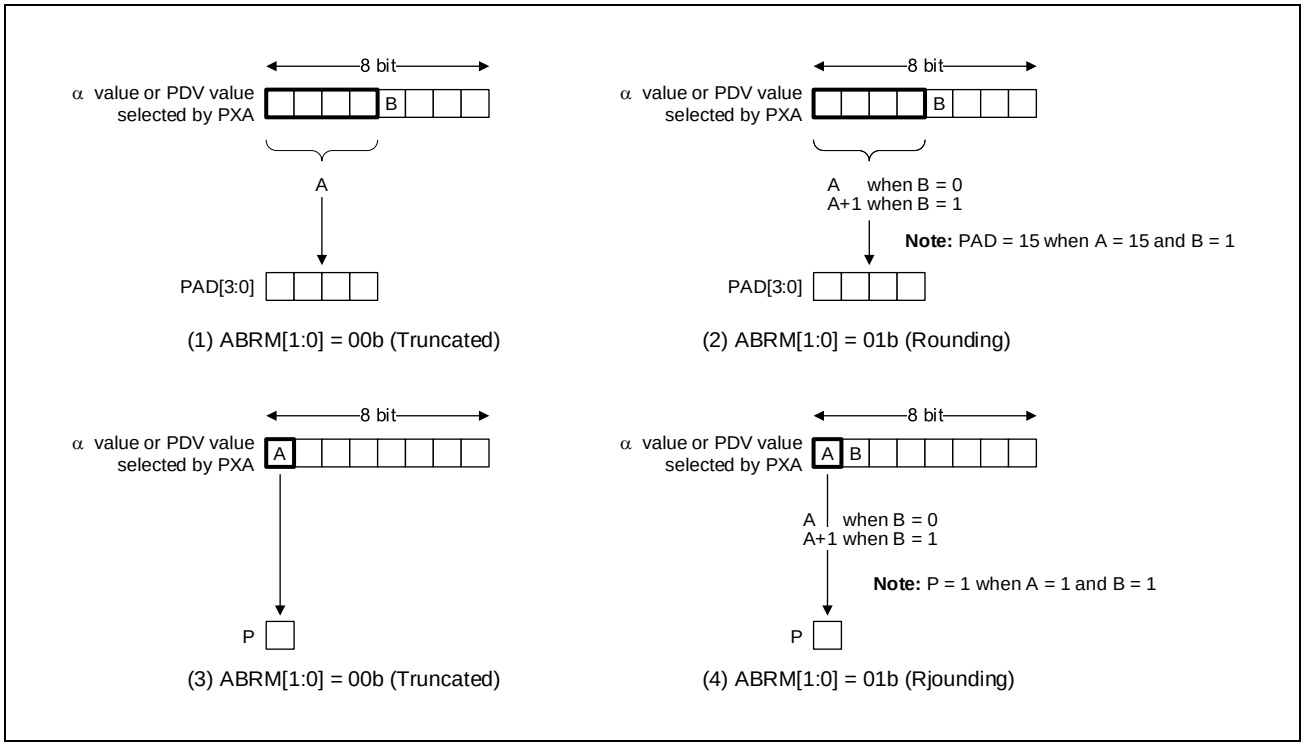
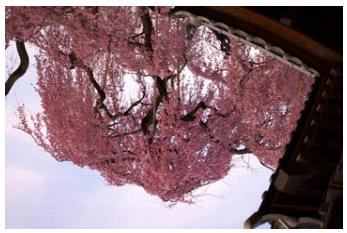



Figure 33.17 Selection of PAD Value and Reduction of Bit Count through PXA Setting



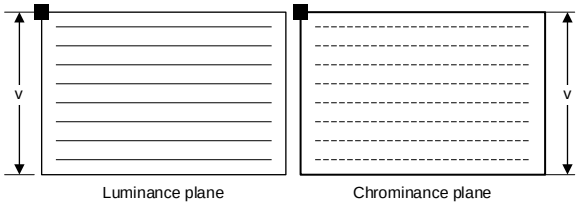
Without flipping (ROT = 0)

Vertical flipping

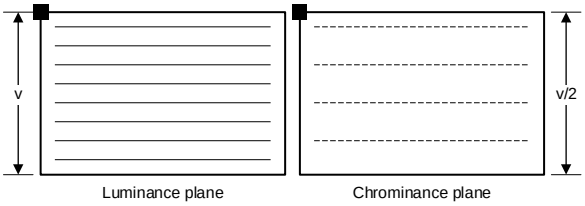


Vertical flipping (ROT = 1)

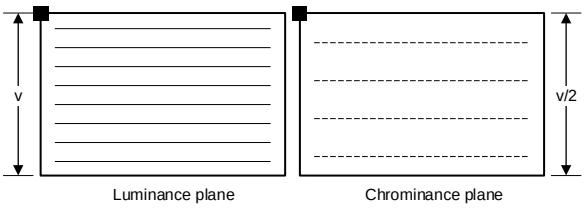
Figure 33.18 Correspondence between Original Image and Flipping Result according to ROT Setting




(A) Before conversion to YCbCr4:2:0




(B) After conversion to YCbCr4:2:0 without vertical flipping




(C) After conversion to YCbCr4:2:0 with vertical flipping

Figure 33.19 Chrominance Output Lines in YCbCr4:2:0 and Vertical Flipping Result

Table 33.19 Packed RGB Formats for WPF Output

| WRFMT[6:0] | Bit per pixel | Phase | upper row - address / bottom row - bit field | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | n | | | | | | | | n+1 | | | | | | | | n+2 | | | | | | | | n+3 | | | | | | | |
| | | | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| H'00 | 8 | — | R0 | R0 | R0 | G0 | G0 | G0 | B0 | B0 | R1 | R1 | R1 | G1 | G1 | G1 | B1 | B1 | R2 | R2 | R2 | G2 | G2 | G2 | B2 | B2 | R3 | R3 | R3 | G3 | G3 | G3 | B3 | B3 |
| H'01 | 12 | — | 0 | 0 | 0 | 0 | R0 | R0 | R0 | R0 | G0 | G0 | G0 | G0 | B0 | B0 | B0 | B0 | 0 | 0 | 0 | 0 | R1 | R1 | R1 | R1 | G1 | G1 | G1 | G1 | B1 | B1 | B1 | B1 |
| H'02 | | | R0 | R0 | R0 | R0 | G0 | G0 | G0 | G0 | B0 | B0 | B0 | B0 | 0 | 0 | 0 | 0 | R1 | R1 | R1 | R1 | G1 | G1 | G1 | G1 | B1 | B1 | B1 | B1 | 0 | 0 | 0 | 0 |
| H'03 | — | — | Reserved | | | | | | | | Reserved | | | | | | | | Reserved | | | | | | | | Reserved | | | | | | | |
| H'04 | 15 | — | 0 | R0 | R0 | R0 | R0 | R0 | G0 | G0 | G0 | G0 | G0 | B0 | B0 | B0 | B0 | B0 | 0 | R1 | R1 | R1 | R1 | R1 | G1 | G1 | G1 | G1 | G1 | B1 | B1 | B1 | B1 | B1 |
| H'05 | | | R0 | R0 | R0 | R0 | R0 | G0 | G0 | G0 | G0 | G0 | B0 | B0 | B0 | B0 | B0 | 0 | R1 | R1 | R1 | R1 | R1 | G1 | G1 | G1 | G1 | G1 | B1 | B1 | B1 | B1 | B1 | 0 |
| H'06 | 16 | — | R0 | R0 | R0 | R0 | R0 | G0 | G0 | G0 | G0 | G0 | G0 | B0 | B0 | B0 | B0 | B0 | R1 | R1 | R1 | R1 | R1 | G1 | G1 | G1 | G1 | G1 | G1 | B1 | B1 | B1 | B1 | B1 |
| H'07 | 18 | — | PAD | | | | | | | | 0 | 0 | 0 | 0 | 0 | 0 | R0 | R0 | R0 | R0 | R0 | R0 | G0 | G0 | G0 | G0 | G0 | G0 | B0 | B0 | B0 | B0 | B0 | B0 |
| H'08 | | | R0 | R0 | R0 | R0 | R0 | R0 | G0 | G0 | G0 | G0 | G0 | G0 | B0 | B0 | B0 | B0 | B0 | B0 | 0 | 0 | 0 | 0 | 0 | 0 | PAD | | | | | | | |
| H'09 | | | 0 | 0 | 0 | 0 | 0 | 0 | R0 | R0 | R0 | R0 | R0 | R0 | G0 | G0 | G0 | G0 | G0 | G0 | B0 | B0 | B0 | B0 | B0 | B0 | PAD | | | | | | | |
| H'0A | | | PAD | | | | | | | | R0 | R0 | R0 | R0 | R0 | R0 | G0 | G0 | G0 | G0 | G0 | G0 | B0 | B0 | B0 | B0 | B0 | B0 | 0 | 0 | 0 | 0 | 0 | 0 |
| H'0B | | | PAD | | | | | | | | 0 | 0 | R0 | R0 | R0 | R0 | R0 | R0 | 0 | 0 | G0 | G0 | G0 | G0 | G0 | G0 | 0 | 0 | B0 | B0 | B0 | B0 | B0 | B0 |
| H'0C | | | 0 | 0 | R0 | R0 | R0 | R0 | R0 | R0 | 0 | 0 | G0 | G0 | G0 | G0 | G0 | G0 | 0 | 0 | B0 | B0 | B0 | B0 | B0 | B0 | PAD | | | | | | | |
| H'0D | | | PAD | | | | | | | | R0 | R0 | R0 | R0 | R0 | R0 | 0 | 0 | G0 | G0 | G0 | G0 | G0 | G0 | 0 | 0 | B0 | B0 | B0 | B0 | B0 | B0 | 0 | 0 |
| H'0E | | | R0 | R0 | R0 | R0 | R0 | R0 | 0 | 0 | G0 | G0 | G0 | G0 | G0 | G0 | 0 | 0 | B0 | B0 | B0 | B0 | B0 | B0 | 0 | 0 | PAD | | | | | | | |
| H'0F | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | R0 | R0 | R0 | R0 | R0 | R0 | G0 | G0 | G0 | G0 | G0 | G0 | B0 | B0 | B0 | B0 | B0 | B0 | 0 | 0 | 0 | 0 | 0 | 0 | R1 | R1 |
| | | 1 | R1 | R1 | R1 | R1 | G1 | G1 | G1 | G1 | G1 | G1 | B1 | B1 | B1 | B1 | B1 | B1 | 0 | 0 | 0 | 0 | 0 | 0 | R2 | R2 | R2 | R2 | R2 | R2 | G2 | G2 | G2 | G2 |
| | | 2 | G2 | G2 | B2 | B2 | B2 | B2 | B2 | B2 | 0 | 0 | 0 | 0 | 0 | 0 | R3 | R3 | R3 | R3 | R3 | R3 | G3 | G3 | G3 | G3 | G3 | G3 | B3 | B3 | B3 | B3 | B3 | B3 |
| H'10 | | 0 | R0 | R0 | R0 | R0 | R0 | R0 | G0 | G0 | G0 | G0 | G0 | G0 | B0 | B0 | B0 | B0 | B0 | B0 | 0 | 0 | 0 | 0 | 0 | 0 | R1 | R1 | R1 | R1 | R1 | R1 | G1 | G1 |
| | | 1 | G1 | G1 | G1 | G1 | B1 | B1 | B1 | B1 | B1 | B1 | 0 | 0 | 0 | 0 | 0 | 0 | R2 | R2 | R2 | R2 | R2 | R2 | G2 | G2 | G2 | G2 | G2 | B2 | B2 | B2 | B2 | B2 |
| | | 2 | B2 | B2 | 0 | 0 | 0 | 0 | 0 | 0 | R3 | R3 | R3 | R3 | R3 | R3 | G3 | G3 | G3 | G3 | G3 | G3 | B3 | B3 | B3 | B3 | B3 | B3 | 0 | 0 | 0 | 0 | 0 | 0 |
| H'11 | | 0 | 0 | 0 | R0 | R0 | R0 | R0 | R0 | R0 | 0 | 0 | G0 | G0 | G0 | G0 | G0 | G0 | 0 | 0 | B0 | B0 | B0 | B0 | B0 | B0 | 0 | 0 | R1 | R1 | R1 | R1 | R1 | R1 |
| | | 1 | 0 | 0 | G1 | G1 | G1 | G1 | G1 | G1 | 0 | 0 | B1 | B1 | B1 | B1 | B1 | B1 | 0 | 0 | R2 | R2 | R2 | R2 | R2 | R2 | 0 | 0 | G2 | G2 | G2 | G2 | G2 | G2 |
| | | 2 | 0 | 0 | B2 | B2 | B2 | B2 | B2 | B2 | 0 | 0 | R3 | R3 | R3 | R3 | R3 | R3 | 0 | 0 | G3 | G3 | G3 | G3 | G3 | G3 | 0 | 0 | B3 | B3 | B3 | B3 | B3 | B3 |
| H'12 | | 0 | R0 | R0 | R0 | R0 | R0 | R0 | 0 | 0 | G0 | G0 | G0 | G0 | G0 | G0 | 0 | 0 | B0 | B0 | B0 | B0 | B0 | B0 | 0 | 0 | R1 | R1 | R1 | R1 | R1 | R1 | 0 | 0 |
| | | 1 | G1 | G1 | G1 | G1 | G1 | G1 | 0 | 0 | B1 | B1 | B1 | B1 | B1 | B1 | 0 | 0 | R2 | R2 | R2 | R2 | R2 | R2 | 0 | 0 | G2 | G2 | G2 | G2 | G2 | G2 | 0 | 0 |
| | | 2 | B2 | B2 | B2 | B2 | B2 | B2 | 0 | 0 | R3 | R3 | R3 | R3 | R3 | R3 | 0 | 0 | G3 | G3 | G3 | G3 | G3 | G3 | 0 | 0 | B3 | B3 | B3 | B3 | B3 | B3 | 0 | 0 |
| H'13 | 24 | — | PAD | | | | | | | | R0 | R0 | R0 | R0 | R0 | R0 | R0 | R0 | G0 | G0 | G0 | G0 | G0 | G0 | G0 | G0 | B0 | B0 | B0 | B0 | B0 | B0 | B0 | B0 |
| H'14 | | | R0 | R0 | R0 | R0 | R0 | R0 | R0 | R0 | G0 | G0 | G0 | G0 | G0 | G0 | G0 | G0 | B0 | B0 | B0 | B0 | B0 | B0 | B0 | B0 | PAD | | | | | | | |
| H'15 | | 0 | R0 | R0 | R0 | R0 | R0 | R0 | R0 | R0 | G0 | G0 | G0 | G0 | G0 | G0 | G0 | G0 | B0 | B0 | B0 | B0 | B0 | B0 | B0 | B0 | R1 | R1 | R1 | R1 | R1 | R1 | R1 | R1 |
| | | 1 | G1 | G1 | G1 | G1 | G1 | G1 | G1 | G1 | B1 | B1 | B1 | B1 | B1 | B1 | B1 | B1 | R2 | R2 | R2 | R2 | R2 | R2 | R2 | R2 | G2 | G2 | G2 | G2 | G2 | G2 | G2 | G2 |
| | | 2 | B2 | B2 | B2 | B2 | B2 | B2 | B2 | B2 | R3 | R3 | R3 | R3 | R3 | R3 | R3 | R3 | G3 | G3 | G3 | G3 | G3 | G3 | G3 | G3 | B3 | B3 | B3 | B3 | B3 | B3 | B3 | B3 |
| H'16 | 18 | — | 0 | 0 | 0 | 0 | 0 | 0 | 0 | R0 | R0 | R0 | R0 | R0 | R0 | G0 | G0 | G0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | G0 | G0 | G0 | B0 | B0 | B0 | B0 | B0 | B0 |
| H'17 | | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | R0 | R0 | R0 | R0 | R0 | R0 | G0 | G0 | G0 | G0 | G0 | G0 | B0 | B0 | B0 | B0 | B0 | B0 |
| H'18 | 24 | 0 | B0 | B0 | B0 | B0 | B0 | B0 | B0 | B0 | G0 | G0 | G0 | G0 | G0 | G0 | G0 | G0 | R0 | R0 | R0 | R0 | R0 | R0 | R0 | R0 | B1 | B1 | B1 | B1 | B1 | B1 | B1 | B1 |
| | | 1 | G1 | G1 | G1 | G1 | G1 | G1 | G1 | G1 | R1 | R1 | R1 | R1 | R1 | R1 | R1 | R1 | B2 | B2 | B2 | B2 | B2 | B2 | B2 | B2 | G2 | G2 | G2 | G2 | G2 | G2 | G2 | G2 |
| | | 2 | R2 | R2 | R2 | R2 | R2 | R2 | R2 | R2 | B3 | B3 | B3 | B3 | B3 | B3 | B3 | B3 | G3 | G3 | G3 | G3 | G3 | G3 | G3 | G3 | R3 | R3 | R3 | R3 | R3 | R3 | R3 | R3 |
| H'19 | 12 | — | PAD | | | | R0 | R0 | R0 | R0 | G0 | G0 | G0 | G0 | B0 | B0 | B0 | B0 | PAD | | | | R1 | R1 | R1 | R1 | G1 | G1 | G1 | G1 | B1 | B1 | B1 | B1 |
| H'1A | | — | R0 | R0 | R0 | R0 | G0 | G0 | G0 | G0 | B0 | B0 | B0 | B0 | PAD | | | | R1 | R1 | R1 | R1 | G1 | G1 | G1 | G1 | B1 | B1 | B1 | B1 | PAD | | | |
| H'1B | 15 | — | P | R0 | R0 | R0 | R0 | R0 | G0 | G0 | G0 | G0 | G0 | B0 | B0 | B0 | B0 | B0 | P | R1 | R1 | R1 | R1 | R1 | G1 | G1 | G1 | G1 | G1 | B1 | B1 | B1 | B1 | B1 |
| H'1C | | — | R0 | R0 | R0 | R0 | R0 | G0 | G0 | G0 | G0 | G0 | B0 | B0 | B0 | B0 | B0 | P | R1 | R1 | R1 | R1 | R1 | G1 | G1 | G1 | G1 | G1 | B1 | B1 | B1 | B1 | B1 | P |
| H'1D | 12 | — | PAD | | | | B0 | B0 | B0 | B0 | G0 | G0 | G0 | G0 | R0 | R0 | R0 | R0 | PAD | | | | B1 | B1 | B1 | B1 | G1 | G1 | G1 | G1 | R1 | R1 | R1 | R1 |
| H'1E | | — | B0 | B0 | B0 | B0 | G0 | G0 | G0 | G0 | R0 | R0 | R0 | R0 | PAD | | | | B1 | B1 | B1 | B1 | G1 | G1 | G1 | G1 | R1 | R1 | R1 | R1 | PAD | | | |
| H'1F | 15 | — | P | B0 | B0 | B0 | B0 | B0 | G0 | G0 | G0 | G0 | G0 | R0 | R0 | R0 | R0 | R0 | P | B1 | B1 | B1 | B1 | B1 | G1 | G1 | G1 | G1 | G1 | R1 | R1 | R1 | R1 | R1 |
| H'20 | | — | B0 | B0 | B0 | B0 | B0 | G0 | G0 | G0 | G0 | G0 | R0 | R0 | R0 | R0 | R0 | P | B1 | B1 | B1 | B1 | B1 | G1 | G1 | G1 | G1 | G1 | R1 | R1 | R1 | R1 | R1 | P |
| H'21 | 18 | 0 | 0 | 0 | B0 | B0 | B0 | B0 | B0 | B0 | 0 | 0 | G0 | G0 | G0 | G0 | G0 | G0 | 0 | 0 | R0 | R0 | R0 | R0 | R0 | R0 | 0 | 0 | B1 | B1 | B1 | B1 | B1 | B1 |
| | | 1 | 0 | 0 | G1 | G1 | G1 | G1 | G1 | G1 | 0 | 0 | R1 | R1 | R1 | R1 | R1 | R1 | 0 | 0 | B2 | B2 | B2 | B2 | B2 | B2 | 0 | 0 | G2 | G2 | G2 | G2 | G2 | G2 |
| | | 2 | 0 | 0 | R2 | R2 | R2 | R2 | R2 | R2 | 0 | 0 | B3 | B3 | B3 | B3 | B3 | B3 | 0 | 0 | G3 | G3 | G3 | G3 | G3 | G3 | 0 | 0 | R3 | R3 | R3 | R3 | R3 | R3 |
| H'22 | 24 | — | PAD | | | | | | | | B0 | B0 | B0 | B0 | B0 | B0 | B0 | B0 | G0 | G0 | G0 | G0 | G0 | G0 | G0 | G0 | R0 | R0 | R0 | R0 | R0 | R0 | R0 | R0 |
| H'23 | 16 | — | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | R0 | R0 | R0 | R0 | R0 | G0 | G0 | G0 | G0 | G0 | G0 | B0 | B0 | B0 | B0 | B0 |
| H'24 to H'3F | — | — | Reserved | | | | | | | | Reserved | | | | | | | | Reserved | | | | | | | | Reserved | | | | | | | |

Table 33.20 Packed YCbCr Formats for WPF Output

| WRFMT[6:0] | Packed YCbCr Output Format | Reference |
|---|---|---|
| H'40 | YCbCr4:4:4 semi-planar | **Figure 33.9***4 |
| H'41 | YCbCr4:2:2 semi-planar (NV16, NV61*1) | |
| H'42 | YCbCr4:2:0 semi-planar (NV12, NV21*1) | |
| H'43 to H'45 | Reserved | — |
| H'46 | YCbCr4:4:4 interleaved | **Figure 33.10***4 |
| H'47 | YCbCr4:2:2 interleaved type 0 (UYVY, YUY2*2, YVYU*3) | |
| H'48 | YCbCr4:2:2 interleaved type 1 | |
| H'49 | YCbCr4:2:0 interleaved *5 | |
| H'4A | YCbCr4:4:4 planar | **Figure 33.11***4 |
| H'4B | YCbCr4:2:2 planar (YV16) | |
| H'4C | YCbCr4:2:0 planar (YV12, I420) | |
| H'4D to H'7F | Reserved | — |

Note 1. When the output format is NV61 or NV21, set SPUVS (bit 14) to 1.

Note 2. When the output format is YUY2, set SPYCS (bit 15) to 1.

Note 3. When the output format is YVYU, set SPUVS (bit 14) to 1 and SPYCS (bit 15) to 1.

Note 4. **Figure 33.12** shows the definition of memory address for each pixel in **Figure 33.9** to **Figure 33.11**.

Note 5. Each line of plane is written twice, so byte/pixel of YCbCr420ITL is 3 byte/pixel (same as YCbCr444ITL).

For details of each YCbCr format, refer to **Figure 33.9**, **Figure 33.10**, **Figure 33.11**, **Figure 33.12**. In these figures, registers for the RPF are indicated; read them as registers for the WPF as follows.

| (RPF registers in the figures) | → | (Corresponding WPF registers) |
|---|---|---|
| VI6_RPFn_SRCM_ADDR_Y.SRCM_ADDR_Y | → | VI6_WPFn_DSTM_ADDR_Y.DSTM_ADDR_Y |
| VI6_RPFn_SRCM_ADDR_C0.SRCM_ADDR_C0 | → | VI6_WPFn_DSTM_ADDR_C0.DSTM_ADDR_C0 |
| VI6_RPFn_SRCM_ADDR_C1.SRCM_ADDR_C1 | → | VI6_WPFn_DSTM_ADDR_C1.DSTM_ADDR_C1 |

### (5) WPFn Data Swapping Registers (VI6_WPFn_DSWAP)

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | P_LLS | P_LWS | P_WDS | P_BTS |
| 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 |
|---|---|---|---|---|
| 31 to 4 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 3 | P_LLS | 0 | R/W | WPF Output Data Swapping in LONG LWORD Units<br>The effect of this bit setting is the same as data swapping in the RPF; refer to **Table 33.14**.<br>0: Data swapping in LONG LWORD (64-bit) units is disabled<br>1: Data swapping in LONG LWORD (64-bit) units is enabled |
| 2 | P_LWS | 0 | R/W | WPF Output Data Swapping in long word Units<br>The effect of this bit setting is the same as data swapping in the RPF; refer to **Table 33.14**.<br>0: Data swapping in long word (32-bit) units is disabled<br>1: Data swapping in long word (32-bit) units is enabled |
| 1 | P_WDS | 0 | R/W | WPF Output Data Swapping in Word Units<br>The effect of this bit setting is the same as data swapping in the RPF; refer to **Table 33.14**.<br>0: Data swapping in word (16-bit) units is disabled<br>1: Data swapping in word (16-bit) units is enabled |
| 0 | P_BTS | 0 | R/W | WPF Output Data Swapping in Byte Units<br>The effect of this bit setting is the same as data swapping in the RPF; refer to **Table 33.14**.<br>0: Data swapping in byte (8-bit) units is disabled<br>1: Data swapping in byte (8-bit) units is enabled |

**Table 33.14** shows the data order before and after swapping according to the long long word, long word, word, and byte swapping settings.

When data order in memory for each format is the same as in **Table 33.19** for RGB format and **Figure 33.9** to **Figure 33.12** for YCbCr format, set 1111b to {*_LLS, *_LWS, *_WDS, *_BTS}. If data order is not the same as the definition, change data order within 16byte unit by these bits as shown in **Table 33.14**.

### (6) WPFn Rounding Control Registers (VI6_WPFn_RNDCTRL)

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | CBRM | — | — | ABRM[1:0] | | ATHRESH[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/W | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | CLMD[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 | R | R | R | R | R | R | R | R |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 29 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 28 | CBRM | 0 | R/W | Bit Count Reduction Method Selection for Data Storage in Packed RGB<br>This bit specifies the method for reducing the number of bits when data is stored in the bit fields indicated as R, G, and B in **Table 33.19** and the target bit fields are not 8 bits.<br>0: Bit count conversion: The lower-order bits are truncated<br>1: Bit count conversion: Rounding (rounding off) |
| 27, 26 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 25, 24 | ABRM[1:0] | All 0 | R/W | Bit Count Reduction Method Selection for Data Storage in PAD<br>These bits specify the method for reducing the number of bits when the data selected through VI6_WPFn_OUTFMT.PXA is stored in the bit fields indicated as PAD or P in **Table 33.19** and the target bit field is four bits or one bit.<br>A value of 10b can be specified only when the packed RGB format specified through VI6_WPFn_OUTFMT.WRFMT includes a 1-bit P field. In this case, when the data selected through VI6_WPFn_OUTFMT.PXA is greater than the ATHRESH value, 1 is stored in the P field; when the selected data is not greater than the ATHRESH value, 0 is stored.<br>00b: Bit count conversion: The lower-order bits are truncated<br>01b: Bit count conversion: Rounding (rounding off)<br>10b: Bit count conversion: Comparison with the threshold value (this setting is allowed only when the storage field is one bit)<br>11b: Setting prohibited |
| 23 to 16 | ATHRESH [7:0] | All 0 | R/W | Threshold for Conversion to 1-Bit α Data<br>These bits specify the threshold value used for conversion from 8-bit α data to one bit when the ABRM bits are set to 10b. When the 8-bit α value before bit count reduction is equal to or smaller than the ATHRESH value, 0 is stored as the reduced 1-bit α data. In other cases, 1 is stored as the 1-bit α data.<br>A value from 0 to 255 can be specified. |
| 15, 14 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 13, 12 | CLMD[1:0] | All 0 | R/W | Color Data Clipping<br>These bits specify the method for clipping the YCbCr color data output from the WPF. When RGB color data is output from the WPF, specify 0 in these bits.<br>00b: Output value is not clipped (0-255)<br>01b: Output value is clipped: YCbCr mode 1 (16-235 (Y), 16-240 (Cb/Cr))<br>10b: Output value is clipped: YCbCr mode 2 (Y/Cb/Cr = 1-254)<br>11b: Setting prohibited |
| 11 to 0 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |

### (7) WPFn Destination Y Plane Memory Stride Registers (VI6_WPFn_DSTM_STRIDE_Y)

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PICT_STRD_Y[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 16 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 15 to 0 | PICT_STRD_Y[15:0] | All 0 | R/W | Memory Stride of Destination Picture Y/RGB Plane<br>These bits specify in 1-byte units the memory stride of the destination picture in the external memory to be written to by the WPFn as shown in **Figure 33.20**.<br>A value from H'0000 to H'FFFF can be specified. |

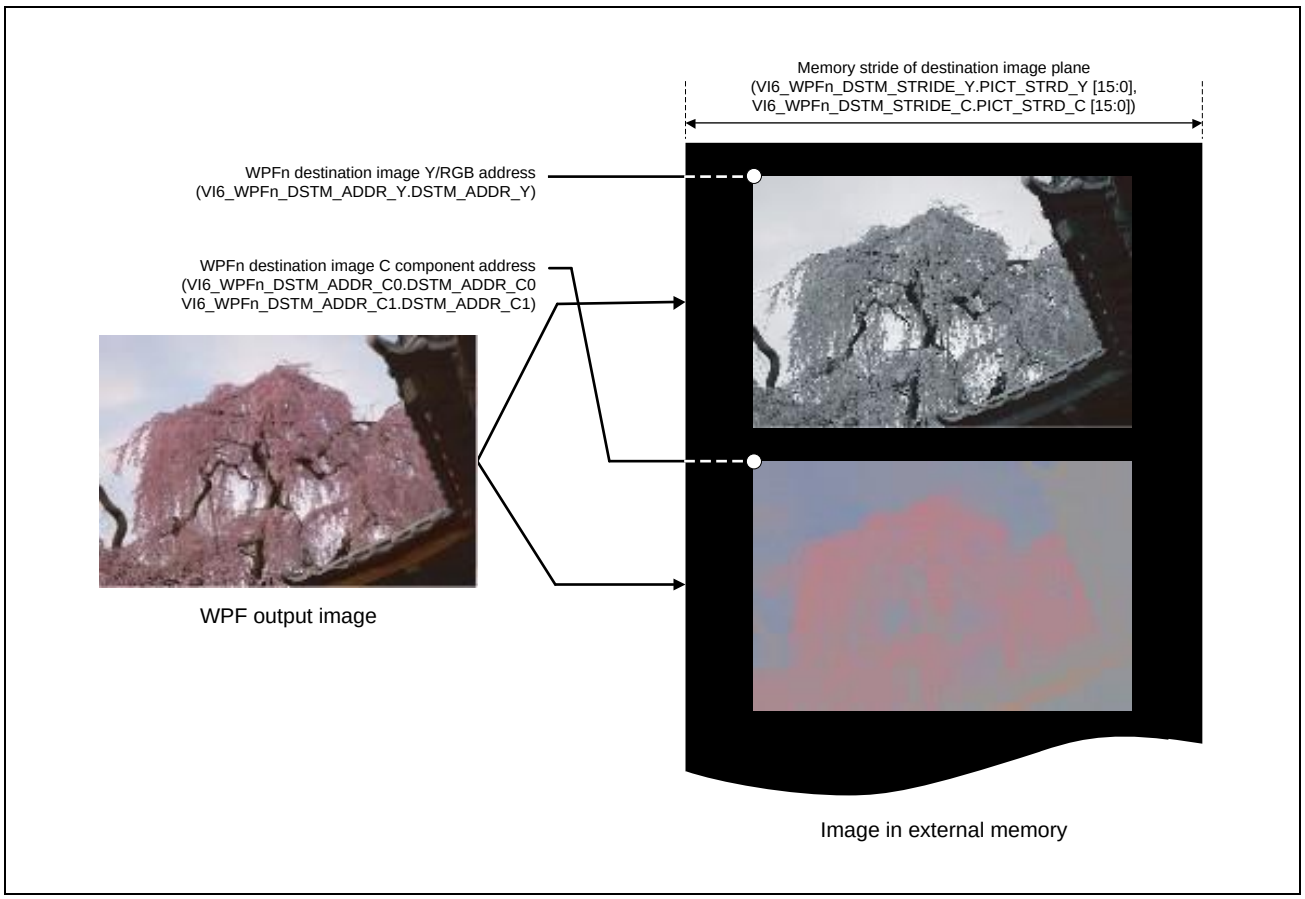
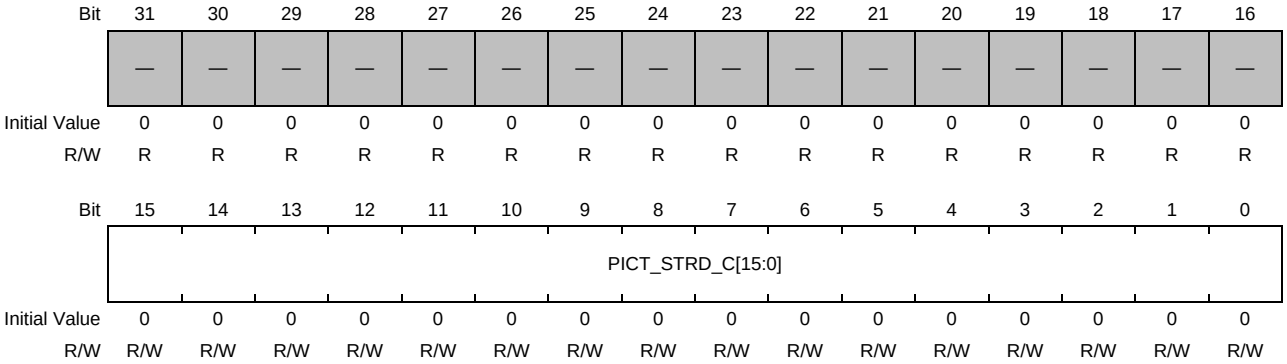


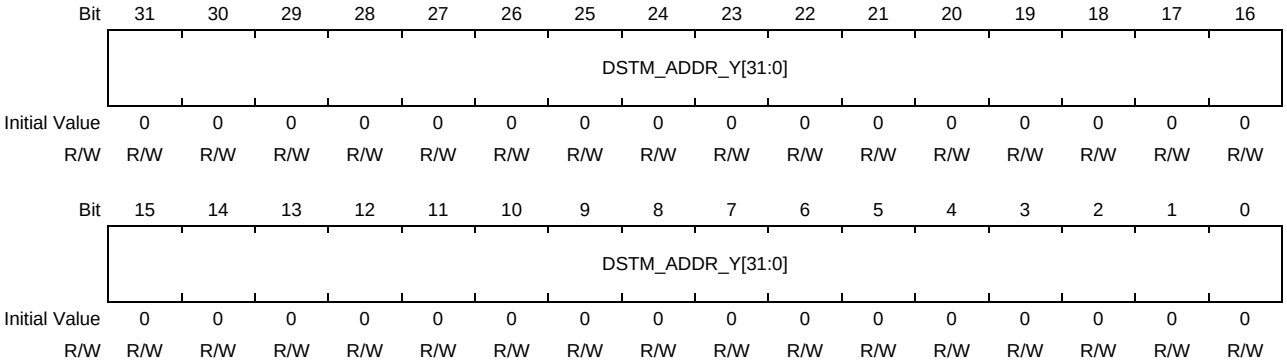

Figure 33.20 Writing Image Data to Destination Area in WPFn

### (8) WPFn Destination C Plane Memory Stride Registers (VI6_WPFn_DSTM_STRIDE_C)




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 16 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 15 to 0 | PICT_STRD_Y[15:0] | All 0 | R/W | Memory Stride of Destination Picture C Plane<br>These bits specify in 1-byte units the memory stride for the C plane of the destination picture in the external memory to be written to by the WPFn as shown in **Figure 33.20**. When the WPFn outputs images in an RGB format, this setting is not used. When the WPFn outputs images in YCbCr planar format, this setting is applied to both the Cb and Cr planes.<br>A value from H'0000 to H'FFFF can be specified. |

### (9) WPFn Destination Y/RGB Address Registers (VI6_WPFn_DSTM_ADDR_Y)




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 0 | DSTM_ADDR_Y[31:0] | All 0 | R/W | Destination Image Y/RGB Plane Storing Address<br>These bits specify in 1-byte units the address for storing the destination-image Y plane or packed RGB plane to be written to by the WPFn in the method described later.<br>A value from H'0000 0000 to H'FFFF FFFF can be specified. |

#### [Destination Address Specification Method]

When flipping is not used, the start address of a frame (address FHA shown in **Figure 33.21**) should be specified as the destination address. When flipping is used, the destination address is not the frame start address (FHA); one of addresses A0 and A2 shown in **Figure 33.21** should be selected according to the combination of desired flipping (VI6_WPFn_OUTFMT.ROT setting).

To strictly define locations A0 and A2, let the horizontal size of the output image be H, the vertical size of the output image be V, and the memory stride (VI6_WPFn_DSTM_STRIDE_Y/C setting) be S as shown in **Figure 33.21**. Calculate the destination address (one of A0 and A2) using the formula shown in **Table 33.21** and specify it in the destination address storing register.

The values of variables L in **Table 33.21** depend on the other register settings and luminance and chrominance components. These values should be obtained by referring to **Table 33.22** when calculating the address to be specified in VI6_WPFn_DSTM_ADDR_Y, or **Table 33.23** when calculating the address to be specified in VI6_WPFn_DSTM_ADDR_C0 or VI6_WPFn_DSTM_ADDR_C1.

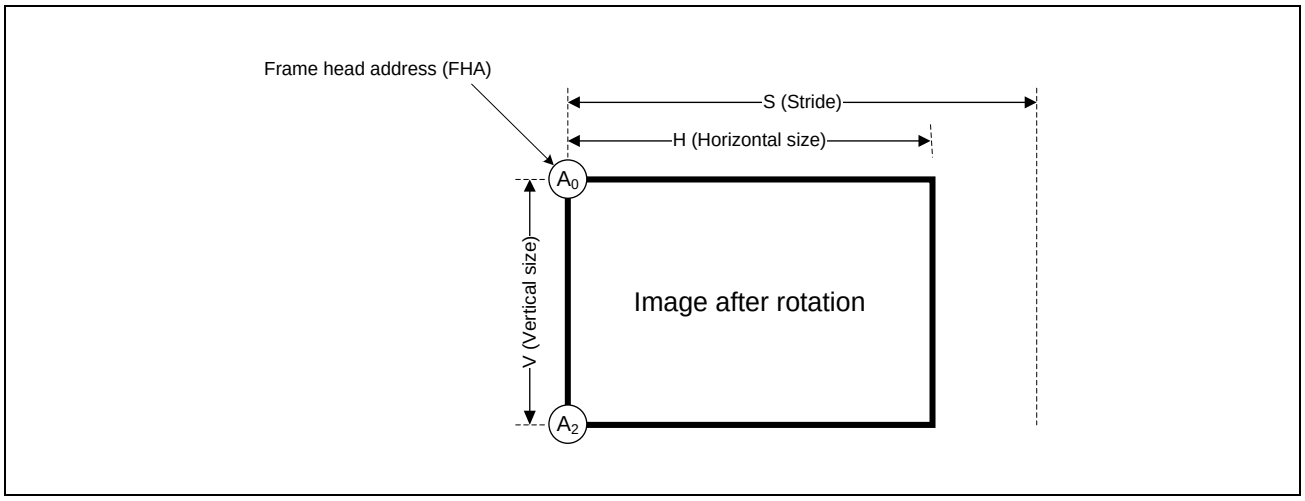



Figure 33.21 Location of Destination Address to be Specified for Flipping

Table 33.21 Destination Address $A_0$, $A_1$, $A_2$, and A3 Calculation Formulas

| VI6_WPFn_OUTFMT.ROT Setting | Formula for Calculating Address to be Set in VI6_WPFn_DSTM_ADDR_Y, VI6_WPFn_DSTM_ADDR_C0, and VI6_WPFn_DSTM_ADDR_C1 |
|---|---|
| 0 | $A_0$ = FHA |
| 1 | $A_2$ = FHA + (V×L – 1) × S |

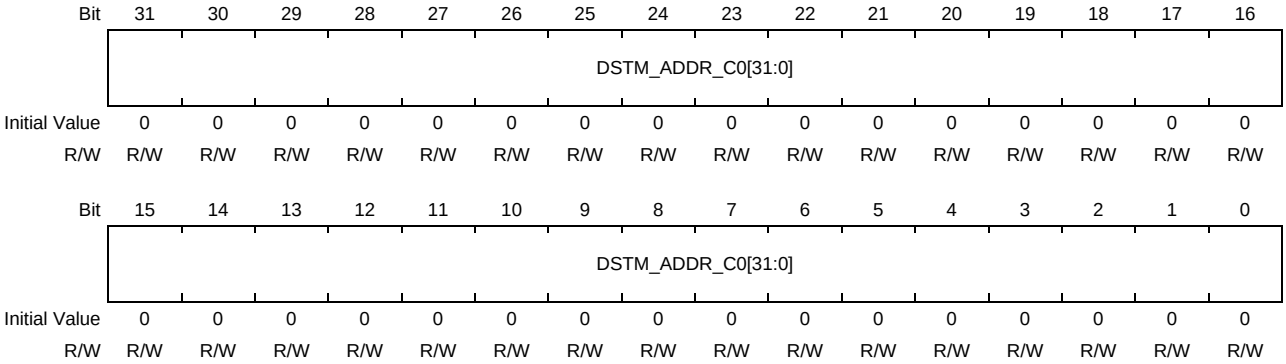
Table 33.22 Value of L according to VI6_WPFn_OUTFMT.WRFMT Setting (for RGB and Luminance Y Address Calculation)

| VI6_WPFn_OUTFMT.WRFMT | L |
|---|---|
| 73 | 0.5 |
| 0 to 2, 4 to 35, 64 to 66, 70 to 72, or 74 to 76 | 1 |

Table 33.23 Value of L according to VI6_WPFn_OUTFMT.WRFMT Setting (for Chrominance C0 and C1 Address Calculation)

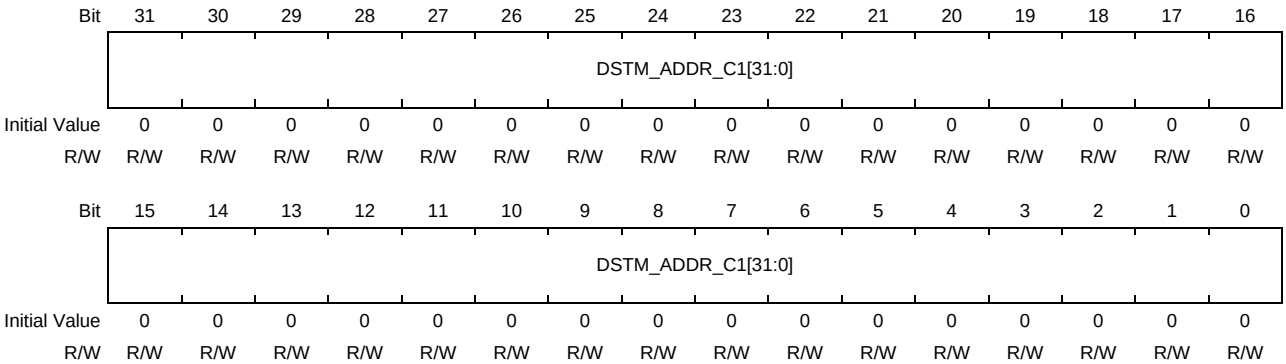
| VI6_WPFn_OUTFMT.WRFMT | L |
|---|---|
| 70 to 73 | Not defined |
| 66, 76 | 0.5 |
| 64 to 65, 74 to 75 | 1 |

### (10) WPFn Destination Chroma Address Registers 0 (VI6_WPFn_DSTM_ADDR_C0)




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 0 | DSTM_ADDR_C0[31:0] | All 0 | R/W | Destination Image C Plane Storing Address 0<br>These bits specify in 1-byte units the address for storing the destination-image C plane to be written to by the WPFn. Refer to the description of VI6_WPFn_DSTM_ADDR_Y for settings.<br>Here, the C plane indicates the combined CbCr plane when a semi-planar format is selected from the packed YCbCr formats shown in **Table 33.20** or the Cb plane when a planar format is selected. When an interleaved format is selected or the output is in an RGB format, this setting is not used.<br>A value from H'0000 0000 to H'FFFF FFFF can be specified. Refer to **Figure 33.21** in **Section 33.3.2.7(10)** for settings. |

### (11) WPFn Destination Chroma Address Registers 1 (VI6_WPFn_DSTM_ADDR_C1)




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 0 | DSTM_ADDR_C1[31:0] | All 0 | R/W | Destination Image C Plane Storing Address 1<br>These bits specify in 1-byte units the address for storing the Cr plane when the WPFn outputs images to the external memory in YCbCr planar format shown in **Table 33.20**. Refer to the description of VI6_WPFn_DSTM_ADDR_Y for settings.<br>This setting is not used when the WPF outputs in YCbCr format that is not a planar format or in an RGB format.<br>A value from H'0000 0000 to H'FFFF FFFF can be specified. Refer to **Figure 33.21** in **Section 33.3.2.7(10)** for settings. |

### (12) WPFn LIF Write Back Control Registers (VI6_WPFn_WRBCK_CTRL: n = 0)

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | WBMD[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 | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 2 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 1, 0 | WBMD[1:0] | All 0 | R/W | Display Data Write Back Control<br>This bit is used for selecting the write back mode when the value of VI6_LIFn_CTRL.LIF_EN bit is set to 1.<br>0: Write Back Disabled<br>1: Write Back Enabled |

### 33.3.2.8 DPR Control Registers

#### (1) Concept of DPR Settings

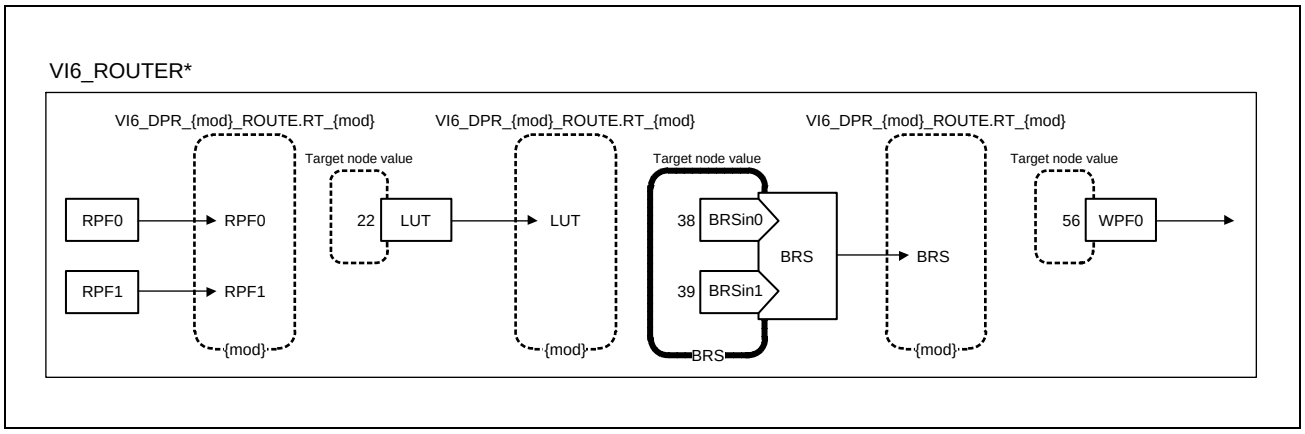



Figure 33.22 Node Register Names and Target Node Values on Data Path Router

In the VSPD internal data path, the order of processes can be specified as desired. The module for performing each process has a unique node value. Set each bit field in VI6_DPR_*_ROUTE to an appropriate node value shown in **Figure 33.22** to specify the target node to be connected behind each module.

For DPR settings, all of the following restrictions should be observed. If any of them is violated, even the WPF paths operating correctly at that time will be affected as well as the WPF paths connected through the DPR, and correct operation will not be guaranteed.

1. Specify 63 for the output node values of all RPFs and processing modules that are not used in the DPR. Here, make sure that no module is connected to a module for which 63 is specified as the node value. Note: VI6_DPR_{mod}_ROUTE.RT are not implemented in VSPD which doesn't have {mod}. See **Figure 33.1** whether VSPD has {mod} or not.
2. When specifying a value other than 63 for an output node value in the DPR, make sure that valid inputs (RPF0 to RPF1 or virtual RPF) and a target WPF are determined.
3. Only one module can be connected to each module; specifying the same target node value for two or more modules is prohibited.
4. Desired modules can be connected between each RPF and BRS input port, but all RPFs specified as the sources for a BRS input port should have the same target WPF.
5. Make appropriate routing or RPF register settings so that the color space formats (RGB/YCbCr) for all BRS input ports are the same.
6. Do not connect the output of any module as the input to the same module (in the BRS case, any input port) even when there is another module between the output and input (creating a loop is prohibited).
7. Each node can be used only once throughout all paths from RPFn to WPFn. When a module shown in **Figure 33.22** is assigned in one RPF to WPF path, it cannot be used in another RPF to WPF path.
8. While a WPF is operating, modifying the DPR connection settings in VI6_DPR_*_ROUTE is prohibited for modules used by the WPF but allowed for modules not used by the WPF. Be careful not to accidentally modify the settings of the modules included in the WPF path that is operating.
9. Be careful to do the following setting when BRS is used.
   VI6_DPR_ILV_BRS_ROUTE.BRSSEL = 1

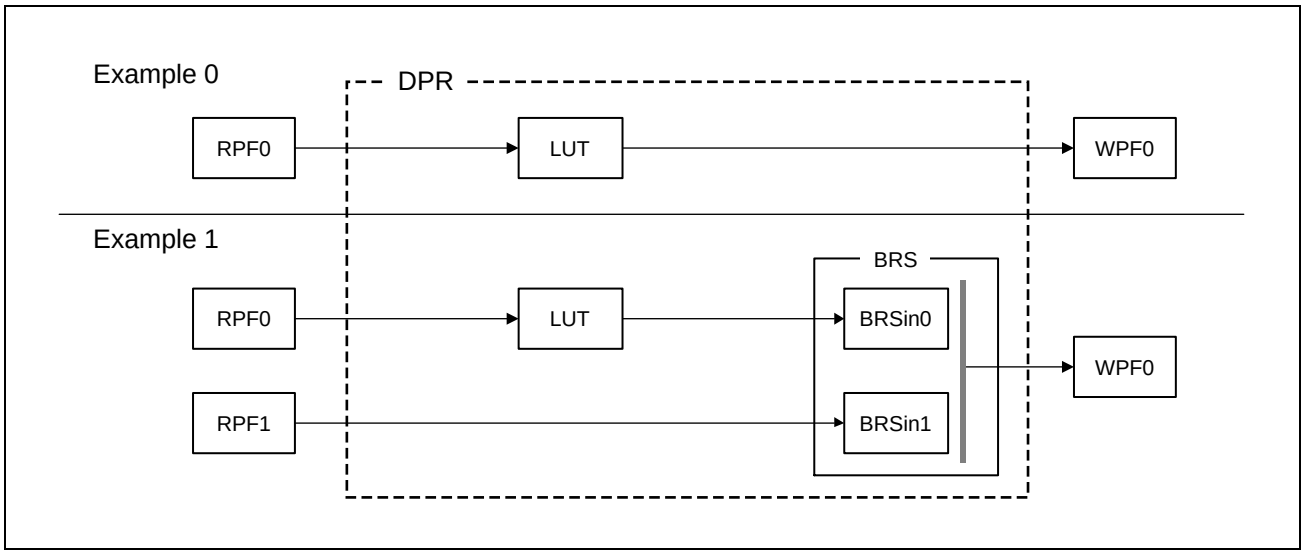



Figure 33.23 Examples of Internal Data Path Routing

**Figure 33.23** shows examples of internal data path routing. Example 0 is WPF0 processing (RPF0 is the source RPF), and example 1 is also WPF0 (RPF0 to RPF1 are the source RPFs). Each example has the configuration shown in **Figure 33.23**. Example 0 performs LUT (e.g. γ correction) processing. Example 1 performs LUT processing (e.g. γ correction), for input 0 (RPF0) and then applies blending or raster operation between the resultant data and input data 1 (RPF1). The VI6_ DPR_*_ROUTE settings for these examples are shown in **Table 33.24**. The bit fields for the modules that are not used in the examples should be set to 63.

**Table 33.24** VI6_DPR_*_ROUTE Register Settings in Connection Examples Shown in **Figure 33.23**. Although both examples show in **Figure 33.23** include operation in image processing modules, connect the RPF to the WPF directly when only image format conversion or packed format conversion is required.

Table 33.24 Examples of Internal Data Path Routing

| | Register Name | Setting | |
|---|---|---|---|
| Example 0 | VI6_DPR_RPF0_ROUTE.RT_RPF0 | 22 | (To LUT) |
| | VI6_DPR_RPF1_ROUTE.RT_RPF1 | 63 | (UNUSED) |
| | VI6_DPR_LUT_ROUTE.RT | 56 | (To WPF0) |
| | VI6_DPR_ILV_BRS_ROUTE.RT | 63 | (UNUSED) |
| Example 1 | VI6_DPR_RPF0_ROUTE.RT_RPF0 | 22 | (To LUT) |
| | VI6_DPR_RPF1_ROUTE.RT_RPF1 | 39 | (To BRSin1) |
| | VI6_DPR_LUT_ROUTE.RT | 38 | (To BRSin0) |
| | VI6_DPR_ILV_BRS_ROUTE.RT | 56 | (To WPF0) |

### (2) RPFn Routing Register (VI6_DPR_RPFn_ROUTE : n = 0, 1)

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | RT_RPFn[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 |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 6 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 5 to 0 | RT_RPFn [5:0] | All 0 | R/W | RPFn Target Node Value<br>These bits specify the target node value for RPFn. When using RPFn, refer to **Figure 33.22** for settings.<br>When RPFn is not started through the VI6_WPFn_SRCRPF setting, specify 63. |

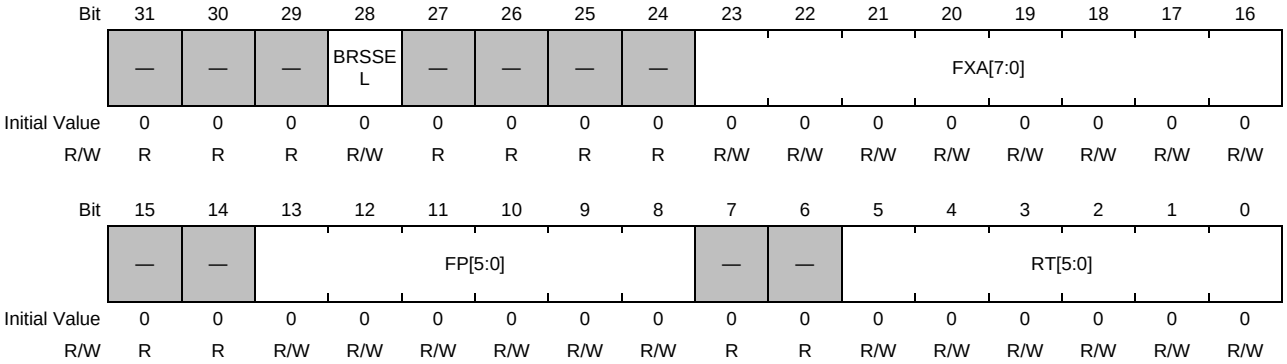
### (3) WPFn Timing Control Register (VI6_DPR_WPFn_FPORCH : n = 0)

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | FP_WPFn[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/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 |
|---|---|---|---|---|
| 31 to 14 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 13 to 8 | FP_WPFn [5:0] | All 0 | R/W | WPFn Internal Operation Timing Setting<br>Specify 5. |
| 7 to 0 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |

### (4) {mod} Routing Register (VI6_DPR_{mod}_ROUTE : {mod}=ILV_BRS, LUT)




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 29 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 28 | BRSSEL | 0 | R/W | This bit is valid only for VI6_DPR_ILV_BRS_ROUTE.<br>Always specify 1. |
| 27 to 24 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 23 to 16 | FXA[7:0] | All 0 | R/W | Fixed α Output Value for {mod}<br>The {mod} does not support input/output of the α value. The α value input to the {LUT} is discarded, and the fixed α value specified in these bits is always output from the {LUT}.<br>A value from 0 to 255 can be specified.<br>These bits are valid for LUT module. For BRS module, these bits are reserved. |
| 15, 14 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 13 to 8 | FP[5:0] | All 0 | R/W | {mod} Internal Operation Timing Setting<br>Specify 0. |
| 7, 6 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 5 to 0 | RT[5:0] | All 0 | R/W | {mod}Target Node Value<br>These bits specify the target node value for the {mod}. When using the {mod}, refer to **Figure 33.22** for settings. When not using the {mod}, specify 63. |

### 33.3.2.9 LUT Control Register

#### (1) LUT Control Register (VI6_LUT_CTRL)

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | LUT_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 | R | R | R | R | R | R | R | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 1 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 0 | LUT_EN | 0 | R/W | 1D-LUT Enable/Disable<br>Enables or disables the 1D-LUT function by the LUT. When the 1D-LUT is used, the color component curve information needs to be set separately in the LUT table. For the LUT table settings, refer to **Section 33.4.7.1**.<br>0: 1D-LUT function is disabled<br>1: 1D-LUT function is enabled |

In the LUT, various image processing, such as curves with high operation load (e.g., γ correction), negative-positive conversion, and gain adjustment of images, can be achieved by the data replacement processing by the 1D-LUT. As shown in **Figure 33.24**, the LUT replaces each component of the input pixel data using the set replacement table of 256 entries. For example, if the LUT is set as in **Figure 33.25**, when there is an input of 150, the data stored in address 150 of the 1D-LUT is read and output as the LUT output. **Figure 33.25** shows a case in which the input and output become equal for convenience in explaining.

In the LUT settings shown in **Figure 33.26**, the input bits are reversed. This has the effect of negative-positive flipping. In **Figure 33.27**, γ correction (γ = 1.8 is shown as an example) is possible. As described above, information to be set in the LUT indicates LUT processing characteristics. If the same value is set for each component in the LUT, an equal effect can be obtained for each component of the input image when it is processed. If the LUT is set with different characteristics for each component, the processing characteristics can be changed for each component.

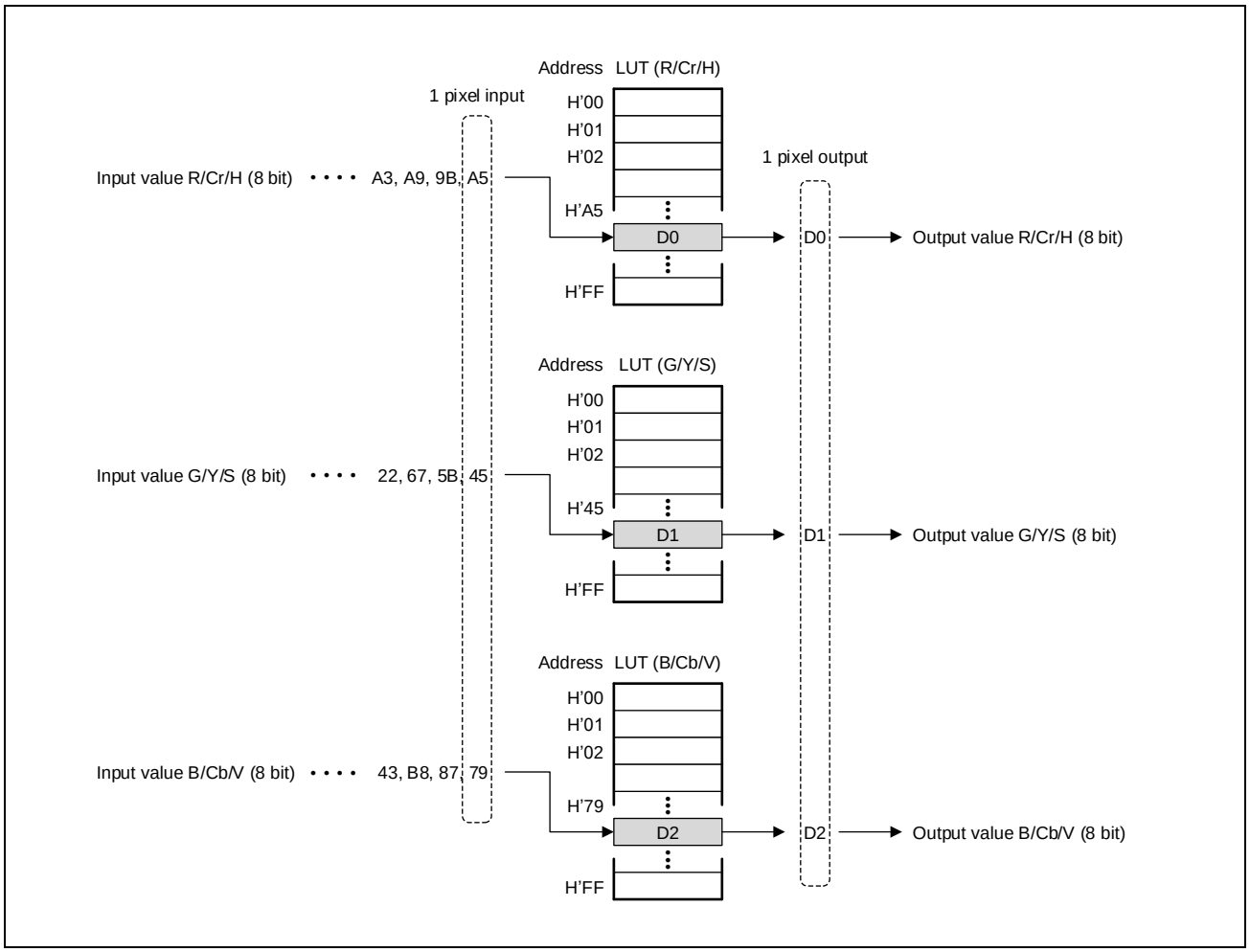



Figure 33.24 Relationship between Input and Output for 1D-LUT Table

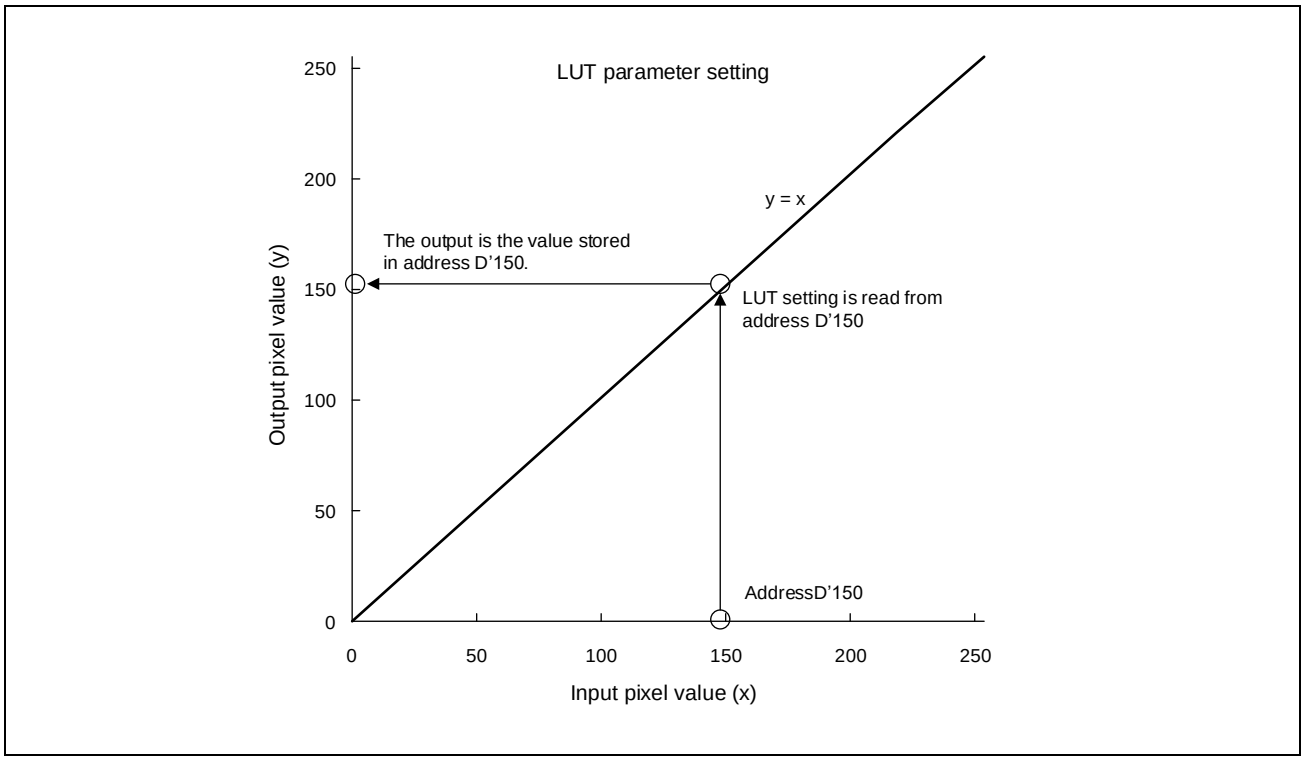



Figure 33.25 Setting Example in Which Output Becomes Equal to Input

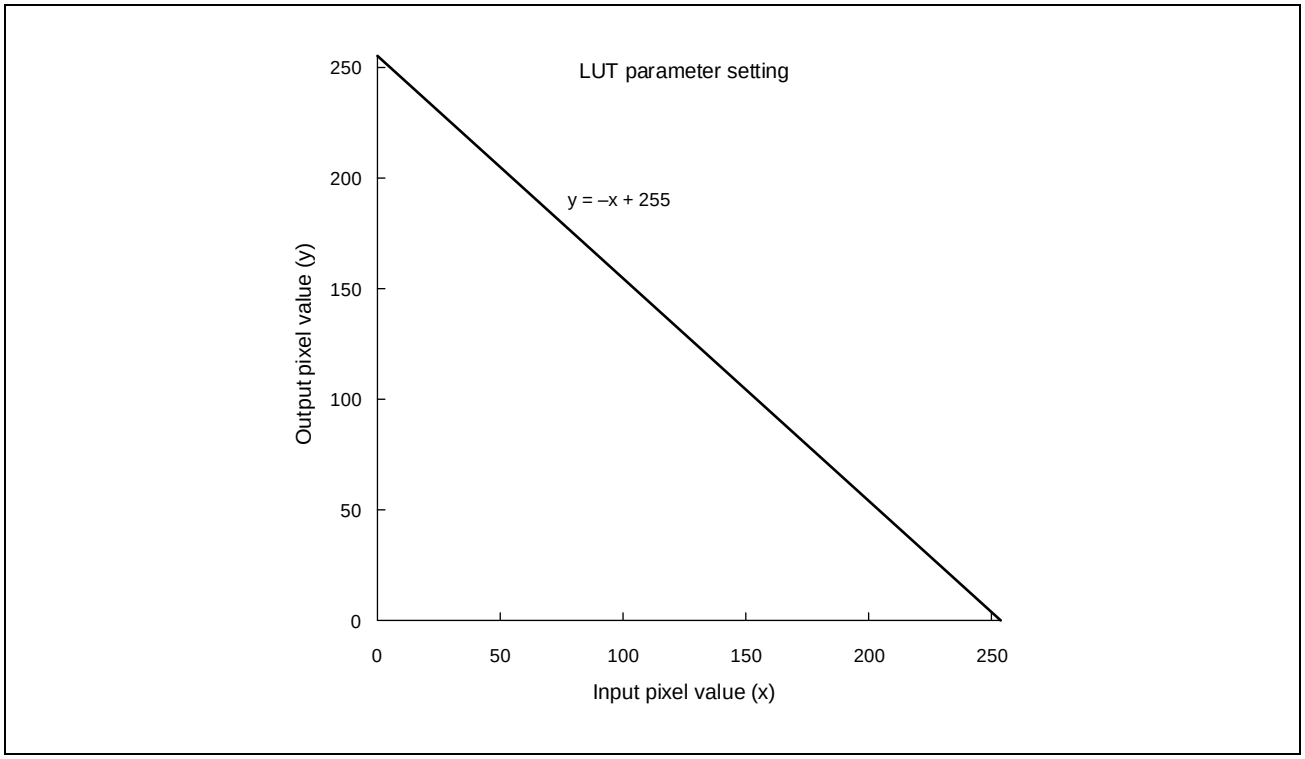


Figure 33.26 Setting Example of Negative-Positive Conversion

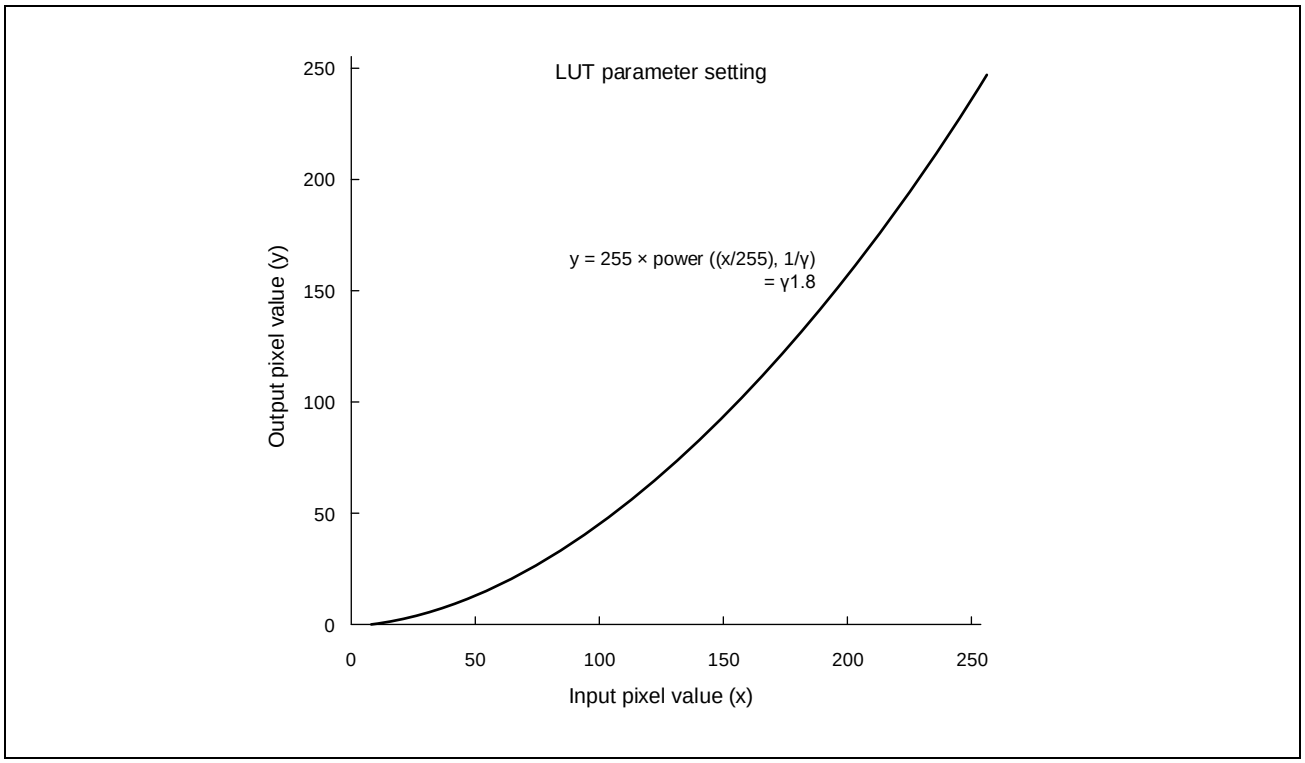



Figure 33.27 Setting Example of γ Correction

### 33.3.2.10 BRS Control Registers

#### (1) BRS Input Control Register (VI6_BRS_INCTRL)

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | NRM | — | — | — | — | — | — | — | — | — | — | D1ON | D0ON |
| 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 | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | ODE1 | DITH1[2:0] | | | ODE0 | DITH0 [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/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 29 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 28 | NRM | 0 | R/W | Color Data Normalization<br>Enables or disables division by the α value of the color data in BRS blending operation. This is used when converting the RGB color data format to which the α value is multiplied (pre-multiplied color) into the RGB color data format to which the α value is not multiplied (non pre-multiplied color). Do not use this for the YCbCr format.<br>0: Divider (DIV unit in **Figure 33.28**) does not divide the color value by α<br>1: Divider (DIV unit in **Figure 33.28**) divides the color value by α |
| 27 to 18 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 17 | D1ON | 0 | R/W | Ordered dither (mode B) Enable of BRS Input 1<br>Enables or disables dithering (color reduction) of BRS input 1 (BRSin1 in **Figure 33.28**).<br>0: Dithering (mode B) of BRSin1 is disabled<br>1: Dithering (mode B) of BRSin1 is enabled<br>When ODE1 in this bit is set to 1, set 0 to this bit. |
| 16 | D0ON | 0 | R/W | Ordered dither (mode B) Enable of BRS Input 0<br>Enables or disables dithering (color reduction) of BRS input 0 (BRSin0 in **Figure 33.28**).<br>0: Dithering (mode B) of BRSin0 is disabled<br>1: Dithering (mode B) of BRSin0 is enabled<br>When ODE0 in this bit is set to 1, set 0 to this bit. |
| 15 to 8 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 7 | ODE1 | 0 | R/W | Ordered Dither (mode A) of CH1 Input to BRS Enable/Disable<br>0: Ordered dither (mode A) of BRSin1 is disabled.<br>1: Ordered dither (mode A) of BRSin1 is enabled.<br>Ordered dither is available only for 18bpp. So, when ODE1 bit is equal to 1, set DITH1 = 1.<br>When ODE1 bit is equal to 0, BRSin1 dither method is specified by D1ON in the register<br>Ordered dither (mode A) is recommended rather than ordered dither (mode B) in case of 18bpp. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 6 to 4 | DITH1[2:0] | All 0 | R/W | Dithering of CH1 Input to BRS<br>These bits specify the number of colors for pixels after dithering (color reduction) when dithering (color reduction) for pixel information is enabled through the D1ON bit. When dithering (color reduction) for pixel information is disabled, specify 0 in these bits.<br>0: Dithering of BRSin1 input image is disabled<br>1: Dithering of BRSin1 input image at 18 bpp (RGB666: 260000 colors)<br>2: Dithering of BRSin1 input image at 16 bpp (RGB565: 65535 colors)<br>3: Dithering of BRSin1 input image at 15 bpp (RGB555: 32768 colors)<br>4: Dithering of BRSin1 input image at 12 bpp (RGB444: 4096 colors)<br>5: Dithering of BRSin1 input image at 8 bpp (RGB332: 256 colors)<br>6, 7: Setting prohibited |
| 3 | ODE0 | 0 | R/W | Ordered Dither (mode A) of CH0 Input to BRS Enable/Disable<br>0: Ordered dither (mode A) of BRSin0 is disabled.<br>1: Ordered dither (mode A) of BRSin0 is enabled.<br>Ordered dither is available only for 18bpp. So, when ODE0 bit is equal to 1, set DITH0 = 1.<br>When ODE0 bit is equal to 0, BRSin0 dither method is specified by D0ON in the register<br>Ordered dither (mode A) is recommended rather than ordered dither (mode B) in case of 18bpp. |
| 2 to 0 | DITH0 [2:0] | All 0 | R/W | Dithering of CH0 Input to BRS<br>These bits specify how to perform dithering of the CH0 input to the BRS. The setting method is the same as that for the DITH1 bits. Read the description of the DITH1 bits with BRSin0 and D0ON replacing BRSin1 and D1ON, respectively. |

**Figure 33.28** shows the configuration of the BRS. For the BRS inputs, there are two inputs from the DPR and one internal input as a virtual RPF. BRSin0 to BRSin1 are input ports that have the target node values shown in **Figure 33.22**, and they can be connected to any module on the DPR. The same color space (YCbCr or RGB) has to be used for the two inputs from the DPR to the BRS.

The virtual RPF inside the BRS is an input unit not connected to the DPR. It is called the “virtual RPF” because it outputs images internally created by the BRS. Starting of the virtual RPF is controlled by VI6_WPFn_SRCRPF.VIR_ACT2, and the single-color data created at the virtual RPF can be used for blending or raster operation (ROP) with data from the other input units BRSin0 to BRSin1. The color space for the single color to be set for the virtual RPF needs to match the color space of the two inputs from the DPR to the BRS. For this setting method, see **Section 33.3.2.10(4)**.

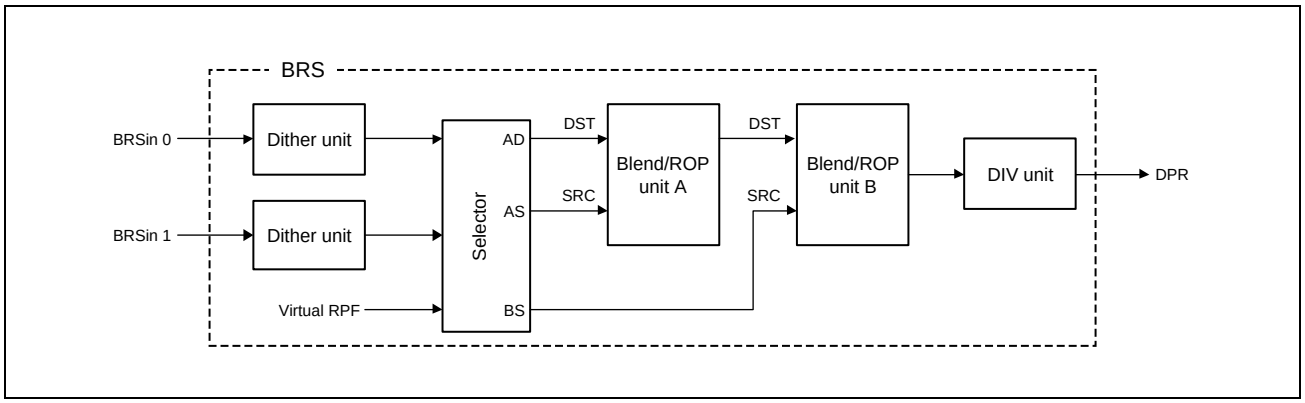


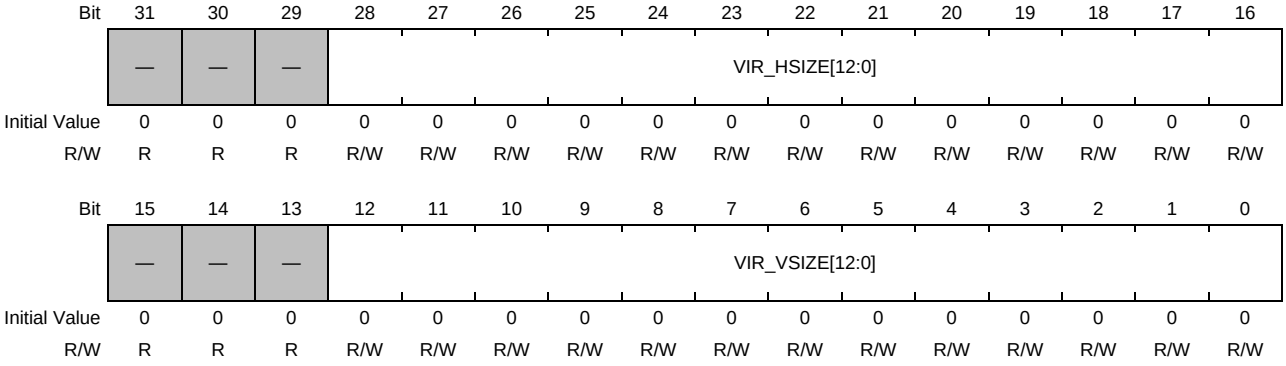

Figure 33.28 BRS Configuration

The selector in **Figure 33.28** is used to select the SRC and DST inputs to blending/ROP units A to B from BRSin0 to BRSin1 which are inputs from the DPR and the virtual RPF. The SRC and DST input sources for blending/ROP units A to B are either uniquely determined based on the configuration shown in **Figure 33.28** or selected as desired by registers. The input sources that can be arbitrarily selected by registers are AD, AS and BS, which correspond to the registers shown in **Table 33.25**.

Table 33.25 Correspondence between Selector Output Destinations and Register Bits for BRS

| Selector Output | Output Destination | Register Bits |
|---|---|---|
| AD | Blending/ROP unit A - DST | VI6_BRSA_CTRL.DSTSEL |
| AS | Blending/ROP unit A - SRC | VI6_BRSA_CTRL.SRCSEL |
| BS | Blending/ROP unit B - SRC | VI6_BRSB_CTRL.SRCSEL |

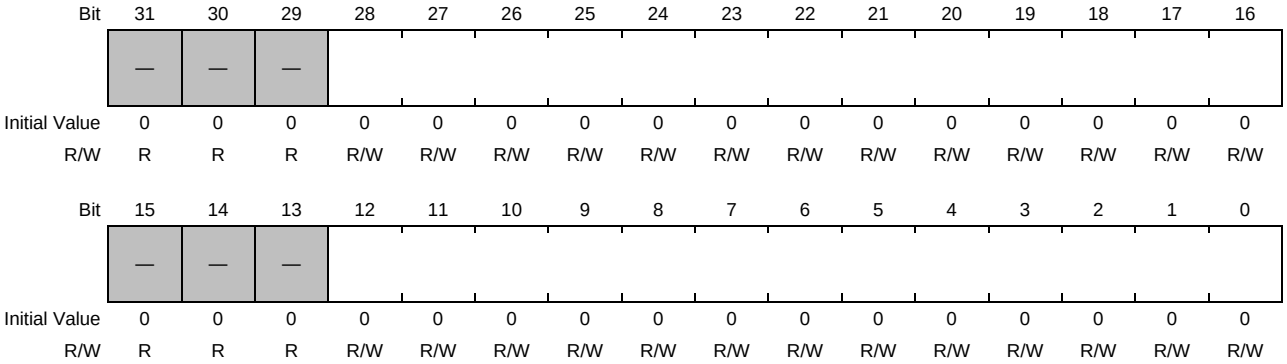
### (2) Size Register of BRS Input Virtual RPF (VI6_BRS_VIRRPF_SIZE)




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 29 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 28 to 16 | VIR_HSIZE [12:0] | All 0 | R/W | Virtual RPF Horizontal Size<br>These bits set the horizontal size of an image from the virtual RPF shown in **Figure 33.28**. A value from 1 to 1280 can be specified. |
| 15 to 13 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 12 to 0 | VIR_VSIZE [12:0] | All 0 | R/W | Virtual RPF Vertical Size<br>These bits set the vertical size of an image from the virtual RPF shown in **Figure 33.28**. A value from 1 to 800 can be specified. |

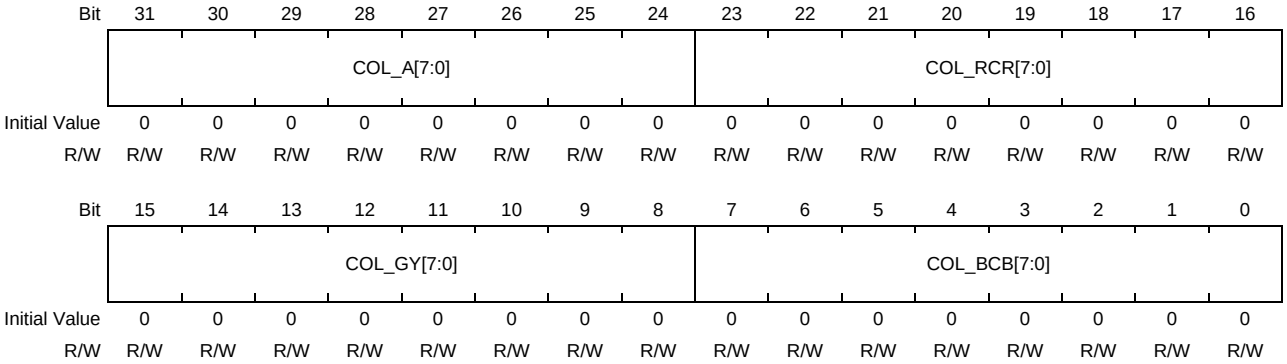
The virtual RPF has only a function to output a fixed α value and a fixed pixel value. The virtual RPF can internally create a single-color image without accessing external memory via the MAU. Same as images from the other BRS input ports, a sublayer can be blended on an image created in this manner with the image used as the background (master layer). In turn, when using the image as a sublayer, it can be drawn on the master layer as a window.

### (3) Display Location Register of BRS Input Virtual RPF (VI6_BRS_VIRRPF_LOC)




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 29 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 28 to 16 | HCOORD [12:0] | All 0 | R/W | Horizontal Coordinate of Virtual RPF Location on Master Layer<br>These bits specify the horizontal coordinate of where to locate the left-edge pixel of the virtual RPF's layer, with the left-edge pixel of the master layer set at coordinate 0. This setting should be made in pixel units. A value from 0 to 1919 can be specified.<br>When the virtual RPF is specified as the master layer by VI6_WPFn_SRCRPF.VIR_ACT2, set these bits to 0. |
| 15 to 13 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 12 to 0 | VCOORD [12:0] | All 0 | R/W | Vertical Coordinate of Virtual RPF Location on Master Layer<br>These bits specify the vertical coordinate of where to locate the top-edge pixel of the virtual RPF's layer, with the top-edge pixel of the master layer set at coordinate 0. This setting should be made in pixel units. A value from 0 to 1079 can be specified.<br>When the virtual RPF is specified as the master layer by VI6_WPFn_SRCRPF.VIR_ACT2, set these bits to 0. |

### (4) Color Information Register of BRS Input Virtual RPF (VI6_BRS_VIRRPF_COL)




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 24 | COL_A[7:0] | All 0 | R/W | Fixed α of Virtual RPF<br>These bits set the fixed α value when the virtual RPF creates a virtual plane. A value from 0 to 255 can be specified. |
| 23 to 16 | COL_RCR [7:0] | All 0 | R/W | Fixed R/Cr of Virtual RPF<br>These bits set the fixed R/Cr value when the virtual RPF creates a virtual plane. A value from 0 to 255 can be specified. |
| 15 to 8 | COL_GY[7:0] | All 0 | R/W | Fixed G/Y of Virtual RPF<br>These bits set the fixed G/Y value when the virtual RPF creates a virtual plane. A value from 0 to 255 can be specified. |
| 7 to 0 | COL_BCB [7:0] | All 0 | R/W | Fixed B/Cb of Virtual RPF<br>These bits set the fixed B/Cb value when the virtual RPF creates a virtual plane. A value from 0 to 255 can be specified. |

The transparency information and color information of the single color that is created by the virtual RPF are set in the bits of this register. As described earlier, the color information is set for the YCbCr or RGB color space. The color space to be set in this register depends on the register settings of the environment and other modules to which the BRS is connected by the DPR. Two cases can be considered. Since the α value (COL_A) is transparency information and irrelevant to the concept of color space, the same setting is made for either the YCbCr or RGB color space.

#### [Case 1: When an input other than the virtual RPF is used]

When the source RPF is connected to any one of the BRS input ports (BRSin0 to BRSin1) other than the virtual RPF and valid data is being supplied, the same color space data as the color space for the BRS inputs should be set in this register as the color space for the virtual RPF's color information. This is based on the restriction of “all BRS inputs must have the same color space”, as described in **Section 33.3.2.8(1)** or **Section 33.3.2.10(1)**.

#### [Case 2: When only the virtual RPF is used]

When only the virtual RPF is used as the source RPF of WPFn, RPFn is not connected to the BRS, as shown in **Figure 33.29**. Thus, there is no color space for another input that the color space for the virtual RPF has to follow, as in case 1. This means that the color space of the data output by the BRS is determined by the WPF setting.

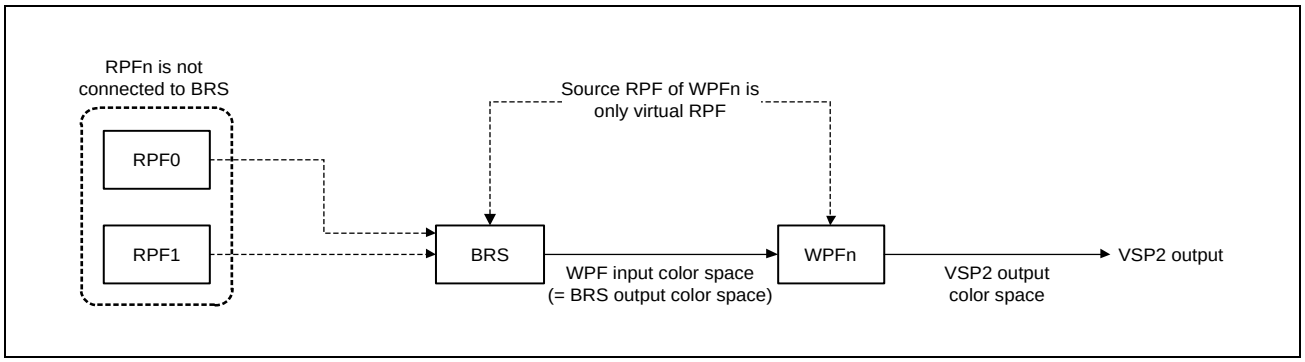



Figure 33.29 Relationship between DPR Connection and Color Space When Only Virtual RPF is Used

**Figure 33.29**, the output color space of the VSPD (= output color space of WPFn) is determined by VI6_WPFn_OUTFMT.WRFMT. When bit 6 in VI6_WPFn_OUTFMT.WRFMT (WRFMT [6]) is 0, the color space is RGB, while when it is 1, the color space is YCbCr. Next, the WPF input color space (= BRS output color space) is determined by the relationship between the WPF output color space and VI6_WPFn_OUTFMT.CSC. When VI6_WPFn_OUTFMT.CSC is 0, the WPF output color space and WPF input color space (= BRS output color space) are the same. When VI6_WPFn_OUTFMT.CSC is 1, the WPF output color space and WPF input color space (= BRS output color space) are the opposite. This relationship is summarized in **Table 33.26**.

The color space for the virtual RPF's color information should be set in this register according to the “BRS output color space” shown in **Table 33.26**.

Table 33.26 Relationship between WPF Output Color Space and BRS Output Color Space

| VI6_WPFn_OUTFMT Register Bit Settings | | | | BRS Output Color Space (= WPF Input Color Space) |
|---|---|---|---|---|
| Bit 6 in WRFMT | | CSC | | |
| 0 | (WPF output is RGB) | 0 | (YCbCr to RGB conversion is disabled) | RGB |
| 0 | (WPF output is RGB) | 1 | (YCbCr to RGB conversion is enabled) | YCbCr |
| 1 | (WPF output is YCbCr) | 0 | (RGB to YCbCr conversion is disabled) | YCbCr |
| 1 | (WPF output is YCbCr) | 1 | (RGB to YCbCr conversion is enabled) | RGB |

### (5) BRS Control Registers (VI6_BRSm_CTRL: m = A, B)

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RBC | — | — | — | — | — | — | — | — | DSTSEL[2:0] | | | — | SRCSEL[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 | R | R | R | R | R | R | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | LBA2[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 | R | R | R | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W |

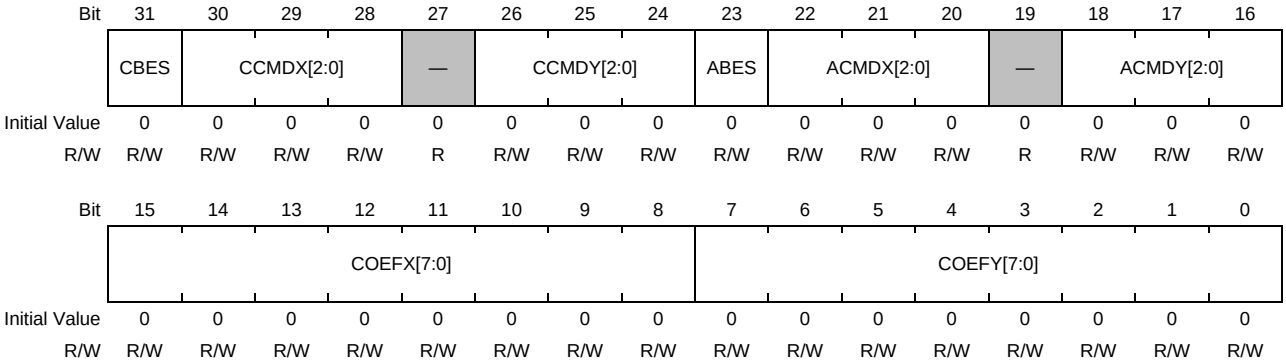
| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 | RBC | 0 | R/W | Operation Type of Blending/ROP Unit m (m = A, B)<br>Specifies the operation type for blending/ROP unit m (m = A, B) shown in **Figure 33.28**.<br>0: ROP (raster operation)<br>1: Blending operation |
| 30 to 23 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 22 to 20 | DSTSEL[2:0] | All 0 | R/W | Input Selection for DST Side of Blending/ROP Unit A<br>These bits select the input for the DST side of blending/ROP unit A shown in **Figure 33.28**. These bits specify the connection between the BRS input port and the DST separately from the setting of connections between other modules and the BRS input port through the DPR.<br>000b: BRS input 0 (BRSin0) is input to DST<br>001b: BRS input 1 (BRSin1) is input to DST<br>100b: Virtual RPF is input to DST<br>010b, 011b, 101b to 111b: Setting prohibited<br>[Note]<br>The DSTSEL bits for blending/ROP unit B is reserved. The write value should always be 0. |
| 19 | — | 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 18 to 16 | SRCSEL[2:0] | All 0 | R/W | Input Selection for SRC Side of Blending/ROP Unit m (m = A, B)<br>These bits select the input for the SRC side of blending/ROP unit m (m = A, B) shown in **Figure 33.28**.These bits specify the connection between the BRS input port and the SRC separately from the setting of connections between other modules and the BRS input port through the DPR.<br>000b: BRS input 0 (BRSin0) is input to SRC<br>001b: BRS input 1 (BRSin1) is input to SRC<br>100b: Virtual RPF is input to SRC<br>010b, 011b, 101b to 111b: Setting prohibited<br>[Note]<br>The SRCSEL bits for blending/ROP unit B are reserved. The write value should always be 0. |
| 15 to 8 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 7 to 4 | CROP[3:0] | All 0 | R/W | Color Data ROP Operator<br>These bits select the ROP operator of the color data in blending/ROP unit m (m = A, B). Select the opcode for ROP operation from **Table 33.27**. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 3 to 0 | AROP[3:0] | All 0 | R/W | α Data ROP Operator<br>These bits select the ROP operator of the α data in blending/ROP unit m (m = A, B). Select the opcode for ROP operation from **Table 33.27**. |

Table 33.27 ROP Operator of Blending/ROP Unit m (m = A, B)

| Opcode | Operator |
|---|---|
| 0000b | NOP(D) |
| 0001b | AND(S & D) |
| 0010b | AND_REVERSE(S & ~D) |
| 0011b | COPY(S) |
| 0100b | AND_INVERTED(~S & D) |
| 0101b | CLEAR(0) |
| 0110b | XOR(S ^ D) |
| 0111b | OR(S \| D) |
| 1000b | NOR(~(S \| D)) |
| 1001b | EQUIV(~(S ^ D)) |
| 1010b | INVERT(~D) |
| 1011b | OR_REVERSE(S \| ~D) |
| 1100b | COPY_INVERTED(~S) |
| 1101b | OR_INVERTED(~S \| D) |
| 1110b | NAND(~(S & D)) |
| 1111b | SET(all1) |

### (6) BRS Blend Control Registers (VI6_BRSm_BLD: m = A, B)




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 | CBES | 0 | R/W | Blending Expression Selection<br>Selects the blending expression of the color data in the BRS (VI6_BRSm_CTRL.RBC = 1). Blending coefficients are specified by the CCMDX and CCMDY bits.<br>0: CCMDX * (DST color data) + CCMDY * (SRC color data)<br>1: CCMDX * (DST color data) - CCMDY * (SRC color data) |
| 30 to 28 | CCMDX[2:0] | All 0 | R/W | Blending Coefficient X Selection<br>These bits specify coefficient X used in the blending expression determined by the CBES bit.<br>000b: DST α data is used as blending coefficient X<br>001b: 255 - (DST α data) is used as blending coefficient X<br>010b: SRC α data is used as blending coefficient X<br>011b: 255 - (SRC α data) is used as blending coefficient X<br>100b: Fixed α value 0 (COEFX setting) |
| 27 | — | 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 26 to 24 | CCMDY[2:0] | All 0 | R/W | Blending Coefficient Y Selection<br>These bits specify coefficient Y used in the blending expression determined by the CBES bit.<br>000b: DST α data is used as blending coefficient Y<br>001b: 255 - (DST α data) is used as blending coefficient Y<br>010b: SRC α data is used as blending coefficient Y<br>011b: 255 - (SRC α data) is used as blending coefficient Y<br>100b: Fixed α value 1 (COEFY setting) When the virtual RPF is specified as the master layer by VI6_WPFn_SRCRPF.VIR_ACT2, set these bits to 0. |
| 23 | ABES | 0 | R/W | Blending α Creation Expression<br>Specifies the expression for creating α data after blending by blending/ROP unit m (m = A, B, C, D, E). α creation coefficients are specified by the ACMDX and ACMDY bits.<br>0: ACMDX * (DST α data) + ACMDY * (SRC α data)<br>1: ACMDX * (DST α data) - ACMDY * (SRC α data) |
| 22 to 20 | ACMDX[2:0] | All 0 | R/W | α Creation Coefficient X<br>These bits specify α creation coefficient X used in the α creation expression determined by the ABES bit.<br>000b: (α creation coefficient X) = (DST α data)<br>001b: (α creation coefficient X) = 255 - (DST α data)<br>010b: (α creation coefficient X) = (SRC α data)<br>011b: (α creation coefficient X) = 255 - (SRC α data)<br>100b: (α creation coefficient X) = Fixed α value 0 (COEFX setting)<br>101b to 111b: Setting prohibited |
| 19 | — | 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 18 to 16 | ACMDY[2:0] | All 0 | R/W | α Creation Coefficient Y<br>These bits specify α creation coefficient Y used in the α creation expression determined by the ABES bit.<br>000b: (α creation coefficient Y) = (DST α data)<br>001b: (α creation coefficient Y) = 255 - (DST α data)<br>010b: (α creation coefficient Y) = (SRC α data)<br>011b: (α creation coefficient Y) = 255 - (SRC α data)<br>100b: (α creation coefficient Y) = Fixed α value 1 (COEFY setting)<br>101b to 111b: Setting prohibited |
| 15 to 8 | COEFX[7:0] | All 0 | R/W | Fixed α Value 0<br>These bits specify fixed α value 0 used when the CCMDX or ACMDX bits are set to 100b. A value from H'00 to H'FF can be specified. |
| 7 to 0 | COEFY[7:0] | All 0 | R/W | Fixed α Value 1<br>These bits specify fixed α value 1 used when the CCMDY or ACMDY bits are set to 100b. A value from H'00 to H'FF can be specified. |

### 33.3.2.11 LIF Control Registers

#### (1) LIF0 Control Register (VI6_LIF0_CTRL)

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | OBTH [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 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | PADL | — | — | — | CFMT | — | — | REQSEL | LIF_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 | R/W | R | R | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 28 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 27 to 16 | OBTH [11:0] | 0 | R/W | Buffer Threshold for Start Ready Notification to Display Module<br>Always specify 1500, when LIF_EN is set to 1. |
| 15 to 9 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 8 | PADL | 0 | R/W | Enable/Disable of padding with dummy lines to output-image to the DU. [LIF0 only]<br>Enables or disables of padding with dummy lines to output-image to the DU.<br>0: Padding with dummy lines to output image to DU is disabled.<br>1: Padding with dummy lines to output image to DU is enabled. |
| 7 to 5 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 4 | CFMT | 0 | R/W | Chroma Format<br>This bit selects the output format from the LIF module to the display module. When RGB format is used, this bit shall be set to 0.<br>0: YCbCr444 or RGB Format<br>1: YCbCr422 Format<br>[Note] The DU cannot receive YCbCr422 format. Therefore, when CFMT is 1, YCbCr444 data which information contents is equal to YCbCr422 is output to DU as shown in **Figure 33.30**. |
| 3, 2 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 1 | REQSEL | 0 | R/W | External Display Module Selection<br>0: This value is setting prohibited when 1 is set to LIF_EN<br>1: DU is selected as the destination external display module. |
| 0 | LIF_EN | 0 | R/W | Enable/Disable of Data Output to External Display Module<br>Enables or disables data output from the LIF to the external display module (DU).<br>0: Data output to the external display module is disabled.<br>1: Data output to the external display module is enabled. |

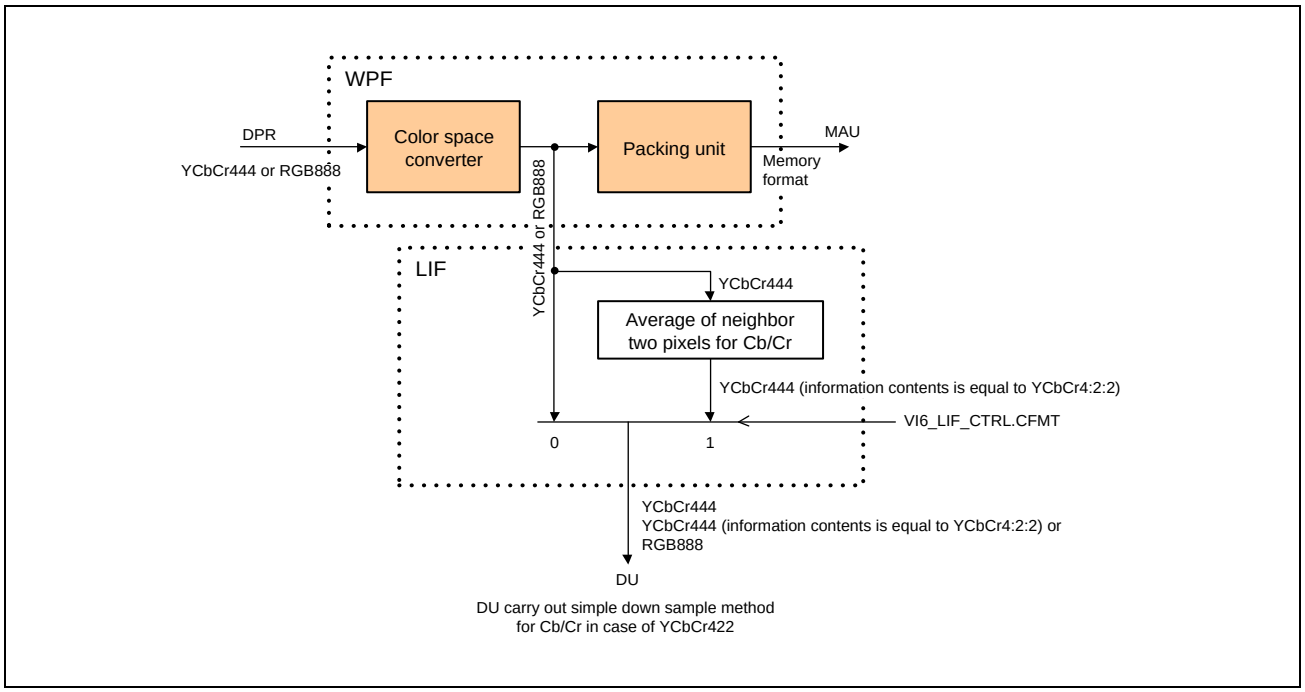



Figure 33.30 Data flow between LIF and DU

### (2) LIF0 Clock Stop Buffer Control Register (VI6_LIF0_CSBTH)

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | HBTH[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 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | LBTH[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 28 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 27 to 16 | HBTH[11:0] | 0 | R/W | Buffer Threshold for Start Ready Notification to Display Module<br>Always specify 1500, when LIF_EN is set to 1. |
| 15 to 12 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 11 to 0 | LBTH[11:0] | 0 | R/W | Buffer Threshold for Clock Start in Dynamic Clock Control<br>Set LBTH[11:0] = 0 (fixed value) |

### (3) LIF0 Buffer Attribute Register (VI6_LIF0_LBA)

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | LBA0 | — | — | — | LBA1[11:0] | | | | | | | | | | | |
| 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/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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | LBA2[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 | LBA0 | 0 | R/W | LIF Buffer Attribute Register0<br>Always specify 1, when LIF_EN is set to 1 |
| 30 to 28 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 27 to 16 | LBA1[11:0] | 0 | R/W | LIF Buffer Attribute Register1<br>Always specify 1536, when LIF_EN is set to 1 |
| 15 to 12 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 11 to 0 | LBA2[11:0] | 0 | R | LIF Buffer Attribute Register2<br>These bits are internal status for purpose of h/w debugging. |

### (4) LIF0 Padding Line Cycle Register (VI6_LIF0_PADLN_CYC)

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | PADLN_CYC[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 |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 6 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 5 to 0 | PADLN_CYC [5:0] | 0 | R/W | Cycle of padding pattern.<br>These bits specify the minimum number of cycles of the padding pattern of valid lines and dummy lines. For example, if PADLN_CYC[5:0] is 4, lower 4 bits of PADLN_PATTERN[3:0] indicates padding pattern. A value from 1 to 32 can be specified. |

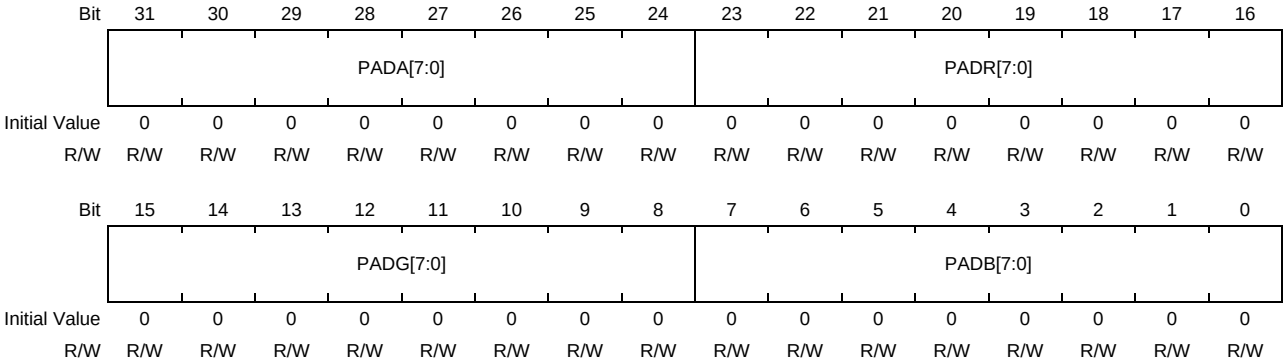
### (5) LIF0 Padding Line Pattern Register (VI6_LIF0_PADLN_PT)

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PADLN_PATTERN[31:0] | | | | | | | | | | | | | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PADLN_PATTERN[31:0] | | | | | | | | | | | | | | | |
| 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 | PADLN_PAT TERN[31:0] | 0 | R/W | Pattern of padding with dummy lines<br>These bits specify the padding pattern of valid lines and dummy lines. In each bit, 0b indicates an active line, 1b indicates a dummy line. It starts from the lower bit (Bit 0).<br>First line should be valid line (Always set 0 to PADLN_PATTERN[0]). |

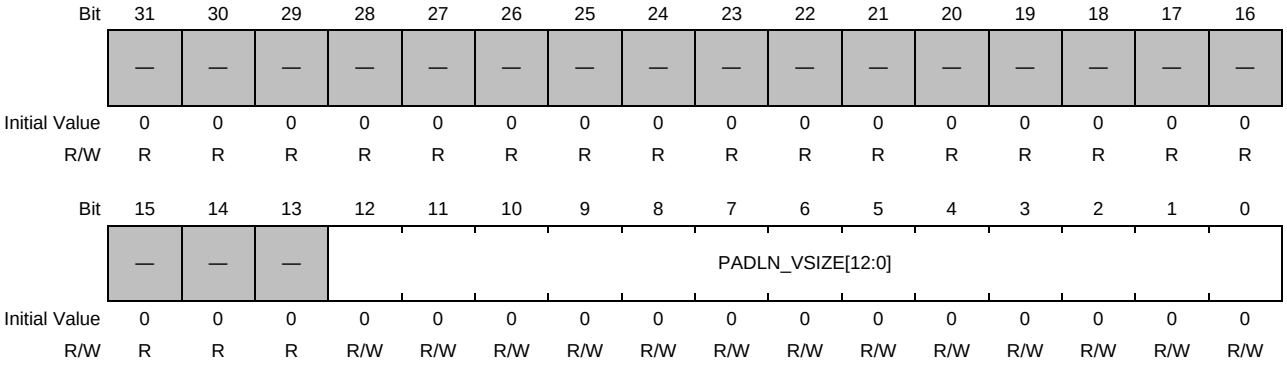
### (6) LIF0 Padding Line Value Register (VI6_LIF0_PADLN_VAL)




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 24 | PADA[7:0] | 0 | R/W | Padding data of EDC code for dummy line. |
| 23 to 16 | PADR[7:0] | 0 | R/W | Padding data of R/Cr for dummy line. |
| 15 to 8 | PADG[7:0] | 0 | R/W | Padding data of G/Y for dummy line. |
| 7 to 0 | PADB[7:0] | 0 | R/W | Padding data of B/Cb for dummy line. |

**Note:** Set H'24FFFFFF to the register, when output format is YUV422. Set H'2EFFFFFF to the register in other case.

### (7) LIF0 Padding Line Size Register (VI6_LIF0_PADLN_SIZE)




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 13 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 12 to 0 | PADLN_VSIZE[12:0] | 0 | R/W | Vertical Size of LIF output in case of enabling Padding line<br>When VI6_LIF0_CTRL.PADL is 1, dummy lines are included in the output data from LIF, so set the LIF output VSIZE including dummy lines to these bits. If the number of output lines does not reach the value of PADLN_VSIZE [12:0] even if all valid lines are output according to padding pattern, dummy lines are inserted until the number of output lines reaches PADLN_VSIZE [12:0].<br>• Output vsize of LIF is same with input vsize to LIF when VI6_LIF0_CTRL.PADL is 0<br>• Output vsize of LIF is PADLN_VSIZE [12:0] when VI6_LIF0_CTRL.PADL is 1 |

## 33.3.3 DU Registers

### 33.3.3.1 DU Module Control Register 0 (DU_MCR0)

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | PB_CLR |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | DI_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 | R | R | R | R | R |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 17 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 16 | PB_CLR | 0 | R/W | Clear PBUF pointers.<br>This parameter must be changed while VSPD is not running and DU_MSR0.ST_DIF_BSY is “0”.<br>Writing “1”. to this parameter starts clearing the PBUF pointers. After clearing the PBUF pointers is completed, DU_MSR0.ST_PB_WINIT and DU_MSR0.ST_PB_RINIT are read as “1”, this parameter must be written back to “0”. |
| 15 to 9 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 8 | DI_EN | 0 | R/W | Display enable.<br>1: Enable<br>0: Disable |
| 7 to 1 | — | All 0 | R | Reserved |
| 0 | — | 0 | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |

### 33.3.3.2 DU Module Status Register 0 (DU_MSR0)

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | ST_PB_RINIT | ST_PB_RUF | ST_PB_REMPTY | — | ST_PB_WINIT | — | ST_PB_WFULL |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | ST_DI_BSY | — | — | — | — | — | — | — | — |
| 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 23 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 22 | ST_PB_RINIT | 1 | R | PBUF FIFO read pointer cleared status.<br>1: Cleared (initial value)<br>0: Not cleared |
| 21 | ST_PB_RUF | 0 | R | PBUF read underflow status.<br>This parameter clears when PBUF pointers are cleared.<br>1: Occur PBUF underflow<br>0: Not occur PBUF underflow |
| 20 | ST_PB_REMPTY | 1 | R | PBUF read empty status.<br>1: Empty<br>0: Not empty |
| 19 | — | 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 18 | ST_PB_WINIT | 1 | R | PBUF FIFO write pointer cleared status.<br>1: Cleared (initial value)<br>0: Not cleared |
| 17 | — | 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 16 | ST_PB_WFULL | 0 | R | PBUF write full status.<br>1: Full<br>0: Not full |
| 15 to 9 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 8 | ST_DI_BSY | 0 | R | Display I/F status.<br>1: Busy<br>0: Idle |
| 7 to 0 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |

### 33.3.3.3 DU Module Status Register 1 (DU_MSR1)

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | UF_VACT | | | | | | | | | | | | |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | UF_HACT | | | | | | | | | | | | |
| 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 29 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 28 to 16 | UF_VACT | H'0000 | R | Vactive counter when the PBUF underflow occurs.<br>This parameter is value latched the down counter starting from VACTIVE.<br>This parameter is cleared when DU_MSR0.ST_PB_RUF is cleared. |
| 15 to 13 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 12 to 0 | UF_HACT | H'0000 | R | Hactive counter when the PBUF underflow occurs.<br>This parameter is value latched the down counter starting from HACTIVE.<br>This parameter is cleared when DU_MSR0.ST_PB_RUF is cleared. |

### 33.3.3.4 DU Interrupt Mask Register 0 (DU_IMR0)

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | IM_PB_RUF |
| 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<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 0 | IM_PB_RUF | 0 | R/W | Mask PBUF read underflow interrupt.<br>1: Mask interrupt<br>0: Not mask interrupt |

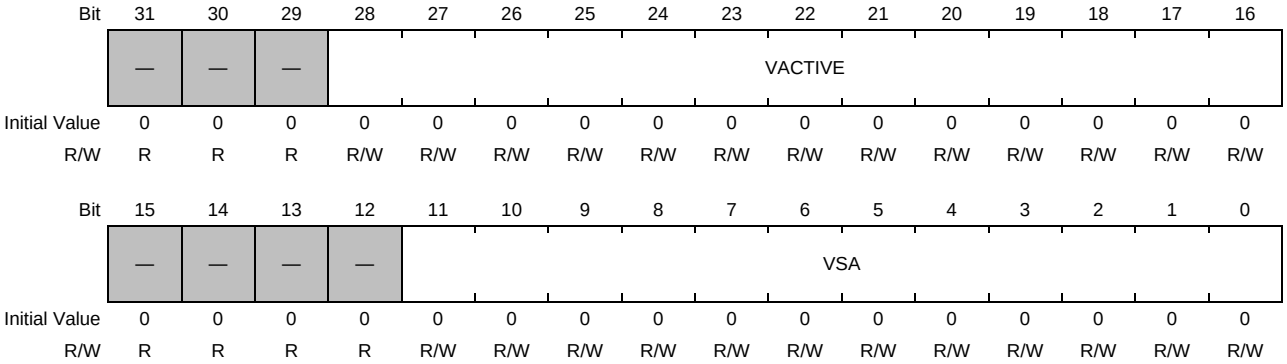
### 33.3.3.5 DU Display I/F Timing Register 0 (DU_DITR0)

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | HSPOL | VSPOL |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | DEMD | | — | — | — | — | — | — | — | DPI_CLKMD |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 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/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 18 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 17 | HSPOL | 1 | R/W | hsync polarity.<br>1: High active<br>0: Low active |
| 16 | VSPOL | 1 | R/W | vsync polarity.<br>1: High active<br>0: Low active |
| 15 to 10 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 9, 8 | DEMD | 11b | R/W | de output mode.<br>00: Fixed to Low<br>01: Reserved<br>10: Reserved<br>11: Data enable (High active) |
| 7 to 1 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 0 | DPI_CLKMD | 0 | R/W | Display parallel interface clock mode.<br>0: Through output<br>1: Inversion output |

### 33.3.3.6 DU Display I/F Timing Register 1 (DU_DITR1)




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 29 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 28 to 16 | VACTIVE | H'0000 | R/W | The number of lines in the Vactive period.<br>This parameter should be more than 0. |
| 15 to 12 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 11 to 0 | VSA | H'000 | R/W | The number of lines in the Vsync period.<br>This parameter should be 0 or more, and DU_DITR1.VSA + DU_DITR2.VBP should be more than 0.<br>If set 0, Vsync is not asserted to DU_DITR0.VSPOL. |

**Figure 33.31** shows the relationship between the parameters and the video output timings.

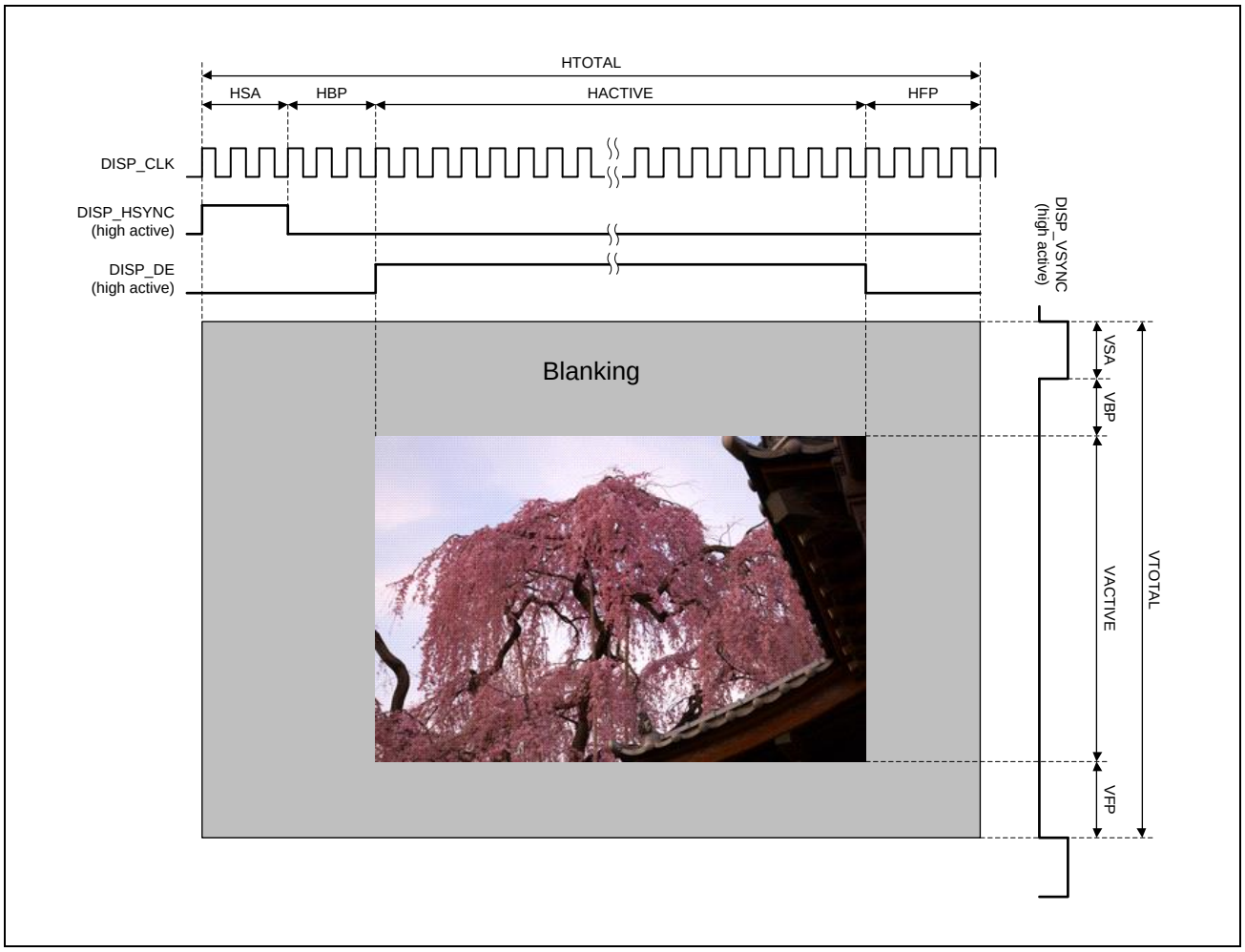



Figure 33.31 Video output timings

### 33.3.3.7 DU Display I/F Timing Register 2 (DU_DITR2)

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | VFP | | | | | | | | | | | | |
| 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 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | VBP | | | | | | | | | | | | |
| 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 |
|---|---|---|---|---|
| 31 to 29 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 28 to 16 | VFP | H'0000 | R/W | The number of lines in the Vfront period.<br>This parameter should be 0 or more. |
| 15 to 13 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 12 to 0 | VBP | H'0000 | R/W | The number of lines in the Vback period.<br>This parameter should be 0 or more, and DU_DITR1.VSA + DU_DITR2.VBP should be more than 0. |

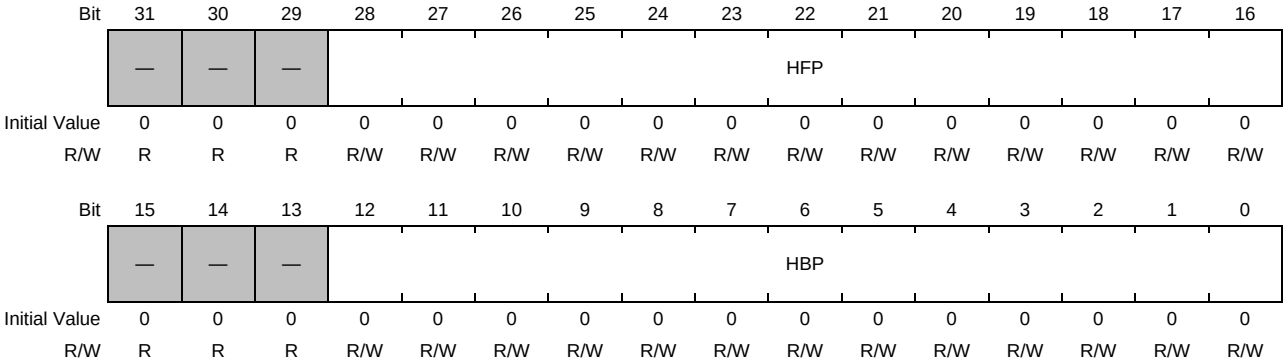
### 33.3.3.8 DU Display I/F Timing Register 3 (DU_DITR3)

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | HACTIVE | | | | | | | | | | | | |
| 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 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | HSA | | | | | | | | | | | |
| 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 29 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 28 to 16 | HACTIVE | H'0000 | R/W | The number of cycles (pixels) in the Hactive period.<br>This parameter should be more than 0. |
| 15 to 12 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 11 to 0 | HSA | H'000 | R/W | The number of cycles in the Hsync period.<br>This parameter should be 0 or more, and DU_DITR3.HSA + DU_DITR4.HBP + DU_DITR4.HFP (=Hblank) should be 3 or more.<br>If set 0, Hsync is not asserted to DU_DITR0.HSPOL. |

### 33.3.3.9 DU Display I/F Timing Register 4 (DU_DITR4)




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 29 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 28 to 16 | HFP | H'0000 | R/W | The number of cycles in the Hfront period.<br>This parameter should be 1 or more, and DU_DITR3.HSA + DU_DITR4.HBP + DU_DITR4.HFP (=Hblank) should be 3 or more. |
| 15 to 13 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 12 to 0 | HBP | H'0000 | R/W | The number of cycles in the Hback period.<br>This parameter should be 0 or more, and DU_DITR3.HSA + DU_DITR4.HBP + DU_DITR4.HFP (=Hblank) should be 3 or more. |

### 33.3.3.10 DU Module Control Register 1 (DU_MCR1)

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | PB_AUTOCLR |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | OPMD | |
| 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 |
|---|---|---|---|---|
| 31 to 17 | — | H'0000 | R | Reserved<br>When read, the initial value is read. The written value will be ignored. |
| 16 | PB_AUTOCLR | 0 | R/W | PBUF pointers auto clear enable.<br>Specify timing to clear the PBUF pointers.<br>0: Not clear until DU_MCR0.PB_CLR is asserted.<br>1: Clear during blanking (according DU_MCR1.OPMD). |
| 15 to 9 | — | H'00 | R | Reserved<br>When read, the initial value is read. The written value will be ignored. |
| 8 | — | 0 | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value.<br>Operation is not guaranteed if a value other than the initial value is written. |
| 7 to 2 | — | H'00 | R | Reserved<br>When read, the initial value is read. The written value will be ignored. |
| 1, 0 | OPMD | All 0 | R/W | Operation mode<br>Specify behavior of LIFC and PBUF during Vfront and Hfront period.<br>This parameter must be changed while VSPD is not running.<br>00: After frame end, LIFC wait, and PBUF pointers are cleared if DU_MCR1.PB_AUTOCLR is set to "1".<br>01: After line end, LIFC wait, and PBUF pointers are cleared if DU_MCR1.PB_AUTOCLR is set to "1".<br>10: Reserved<br>11: LIFC always request to VSPD.PBUF pointers are not cleared until DU_MCR0.PB_CLR is set to "1". |

### 33.3.3.11 DU PBUF Control Register 0 (DU_PBCR0)

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | PB_DEP | | | | |
| 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 |
|---|---|---|---|---|
| 31 to 5 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 4 to 0 | PB_DEP | H'00 | R/W | Valid PBUF depth.<br>This parameter should be set H'1F. |

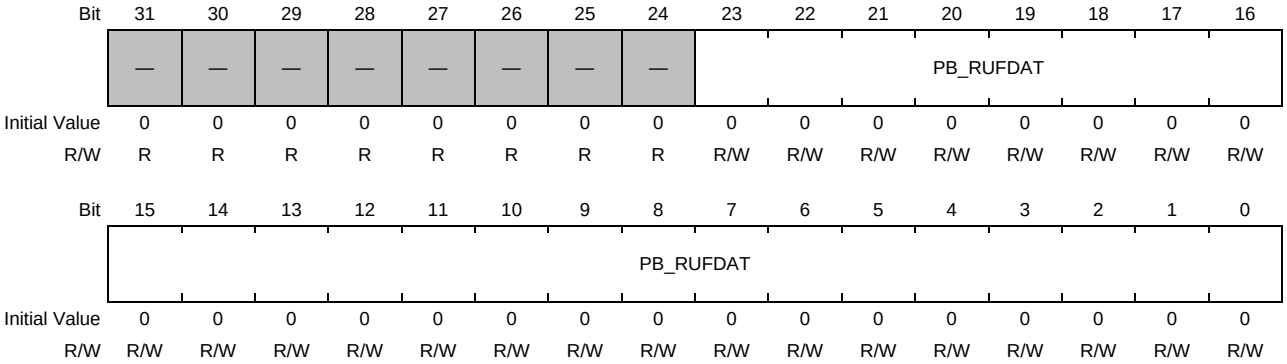
### 33.3.3.12 DU PBUF Control Register 1 (DU_PBCR1)

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | PB_RUFOP |
| 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 | — | H'0000_0000 | R | Reserved<br>When read, the initial value is read. The written value will be ignored. |
| 0 | PB_RUFOP | 0 | R/W | Specify behavior when the PBUF underflow occurs.<br>0: Continue outputting invalid data<br>1: Output DU_PBCR2.PB_RUFDAT |

### 33.3.3.13 DU PBUF Control Register 2 (DU_PBCR2)




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 24 | — | H’00 | R | Reserved<br>When read, the initial value is read. The written value will be ignored. |
| 23 to 0 | PB_RUFDAT | H’00_0000 | R/W | Read data when the PBUF underflow occurs. |

# 33.4 Operation

## 33.4.1 Operation Control Setting

### (a) To start image process

FCPVD registers do not need to be set when starting an image process. Therefore, refer the **Section 33.4.4** for the guidance of starting an image process.

## 33.4.2 Reset Operation

FCPVD reset can be conducted by power on reset. Power on reset initializes the whole FCPVD logic immediately.

### 33.4.2.1 Power-on Reset, Hardware reset

The power-on reset is controlled by an externally input signal called hardware reset. This reset initializes the whole internal logic of LCDC.

## 33.4.3 Registers to set fixed value

Set fixed value to following registers in any case.

[1] Always set H’0000 0808 to VI6_CLK_DCSWT

[2] Always set D’256 to VI6_DL_CTRL.AR_WAIT [7:0] in case of using display list.

[3] Always set D’2 to VI6_DL_EXT_CTRLn.POLINT [5:0] in case of using extended display list of WPFn.

[4] Always set D’1 to VI6_DL_EXT_CTRLn.DLPRI in case of using extended display list of WPFn.

[5] Always set D’0 to VI6_DL_EXT_CTRLn.EXPRI in case of using extended display list of WPFn.

[6] Always set H’0000 0500 toVI6_DPR_WPFn_FPORCH (n = 0, 1) in case of using WPFn.

[7] Always set D’1500 to VI6_LIF0_CTRL.OBTH [11:0] in case of using LIF (VI6_LIF0_CTRL.LIF_EN = 1).

[8] Always set H’8600 0000 to VI6_LIF0_LBA in case of using LIF0 (VI6_LIF0_CTRL.LIF_EN = 1).

[9] Always set D’1 to VI6_LIFn_CTRL.REQSEL in case of using LIFn (VI6_LIFn_CTRL.LIF_EN = 1).

[10] Always set D’0 to VI6_LIFn_CSBTH in case of using LIFn (VI6_LIFn_CTRL.LIF_EN = 1).

[11] Always set H'1F to DU_PBCR0 in case of using DU (DU_MCR0.DI_EN = 1).

### 33.4.4 Concept of VSPD Operation Starting and Stopping

The VSPD provides one channel of image processing. Each channel is started by setting the corresponding start register. Here, starting a processing channel means starting one of WPF0, which are output modules of the VSPD. Use the start registers shown in **Table 33.28** to start WPF modules.

After a WPF module is started and specified processing is completed, the corresponding channel stops operation and notifies the end of processing through an end interrupt. End interrupts are generated through the end interrupt source registers shown in **Table 33.28**; clearing a source register cancels the corresponding interrupt signal. Each of the operating status registers shown in the table indicates the busy state after the corresponding channel is started through the start register until processing is completed and operation stops. **Figure 33.32** shows these operation timings.

Table 33.28 Target Module and Corresponding Registers for Starting and Stopping Operation

| Target Module | Start Register | End Interrupt Source Register | Operating Status Register |
|---|---|---|---|
| WPF0 | VI6_CMD0.STRCMD | VI6_WPF0_IRQ_STA. FRE | VI6_STATUS.SYS0_ACT |

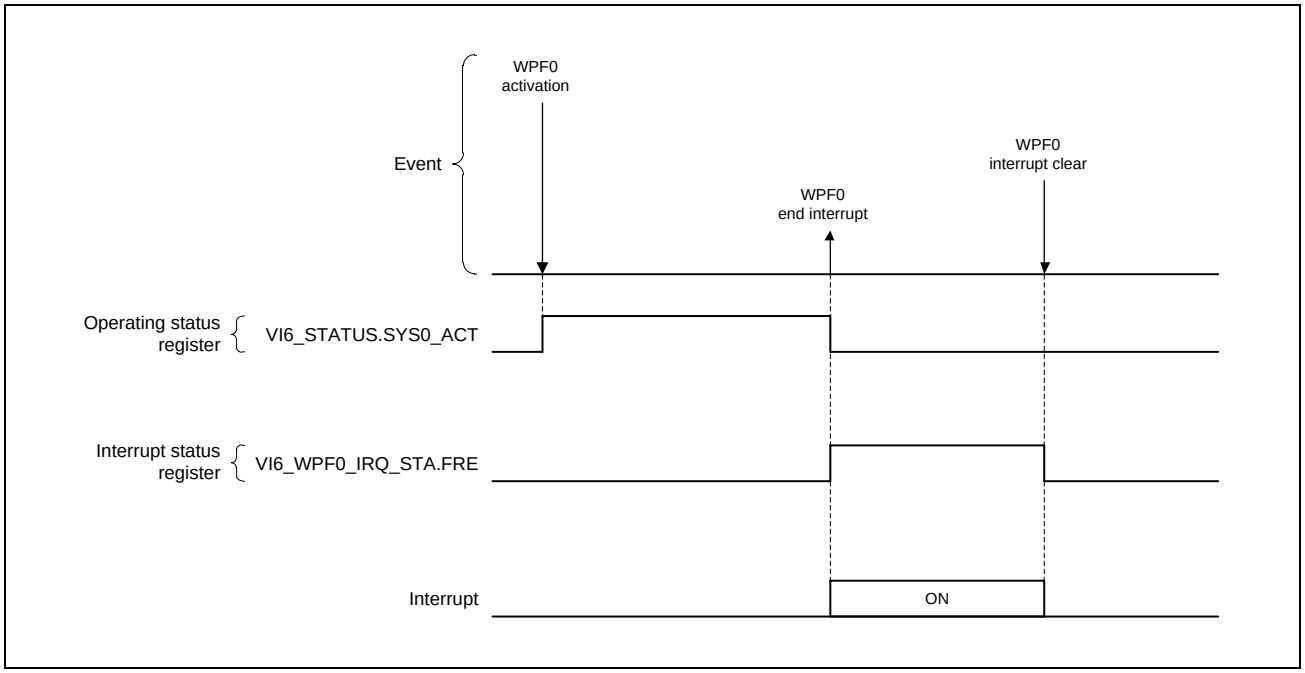



Figure 33.32 VSPD Startup and Status of Each Register and Interrupt

The following describes the operating states (operating or stopped) of VSPD internal modules. As described in **Section 33.3.2**, the VSPD has several image processing modules and the connections between modules are determined by the DPR. Accordingly, the operating state of a module is the same as that of the target WPF for that module. For example, when the target WPF for the BRS is WPF0, the BRS operating state is the same as that of WPF0; that is, the BRS operating state is indicated by the VI6_STATUS.SYS0_ACT as shown in **Table 33.28** and the status change timing is shown as VI6_STATUS. SYS0_ACT in **Figure 33.32**. Likewise, the operating states of all modules connected to WPF0 is indicated by VI6_STATUS. SYS0_ACT.

Connections should be changed through the DPR-related registers (described later) while all modules to be affected by any change in connections are stopped. If connections through the DPR are changed during operation, the VSPD will hang.

**Figure 33.33** shows relation between start reservation and operating status. When start is reserved while VSPD is operating, the reservation is accepted after VSPD status is moved from “operating” into “idle” as shown in **Figure 33.33**.

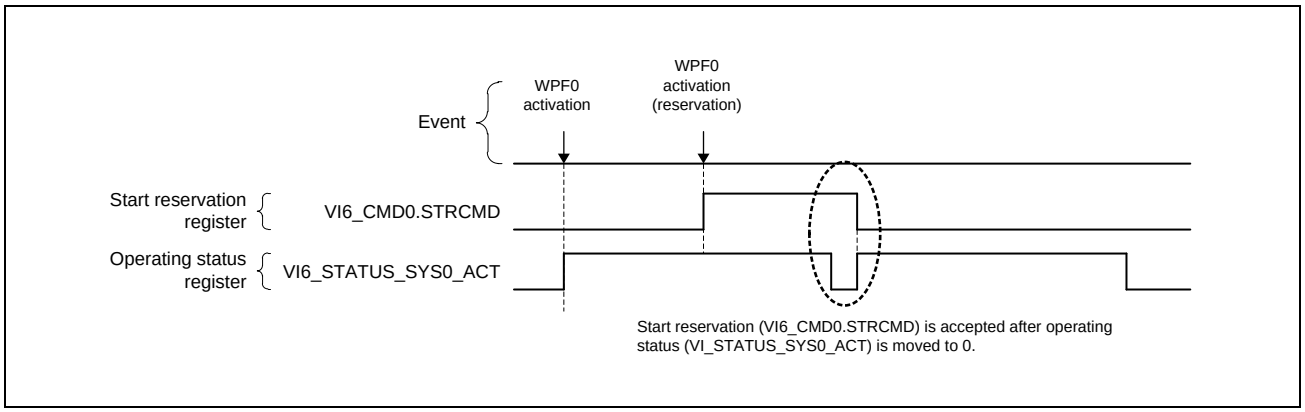



Figure 33.33 VSPD Start reservation and Operating status

## 33.4.5 Display List

### 33.4.5.1 Functional Description

The VSPD provides the display list function. As a display list, the VSPD automatically downloads the register settings except for the control registers (**Sections 33.3.2.4** and **Section 33.3.2.5**) from external memory and stores the settings in the VSPD registers. This function is advantageous in that the interrupt processing or register setting modification processing can be executed without CPU intervention during multiple-frame processing because the register settings used for VSPD processing are prepared in advance in external memory such as SDRAM.

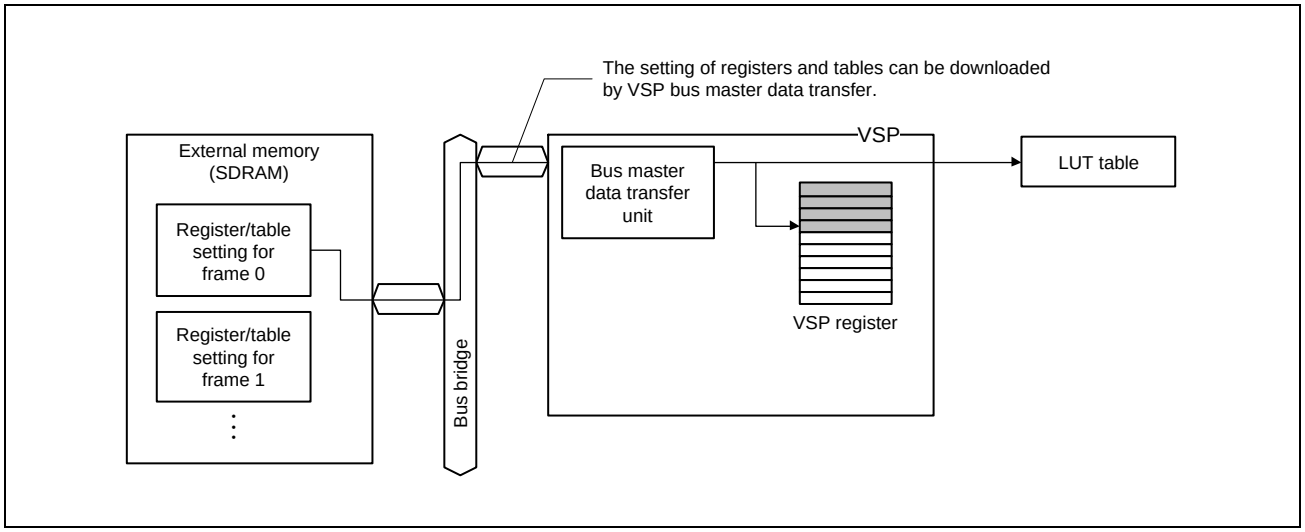



Figure 33.34 Concept of Display List

To use display lists, specify the external memory addresses to the display list control registers described in **Section 33.3.2.5**. The register settings or various types of information should be stored in external memory in the format described in **Section 33.4.5.2**. **Figure 33.35** shows the difference between VSPD operation through normal register settings and through display lists.

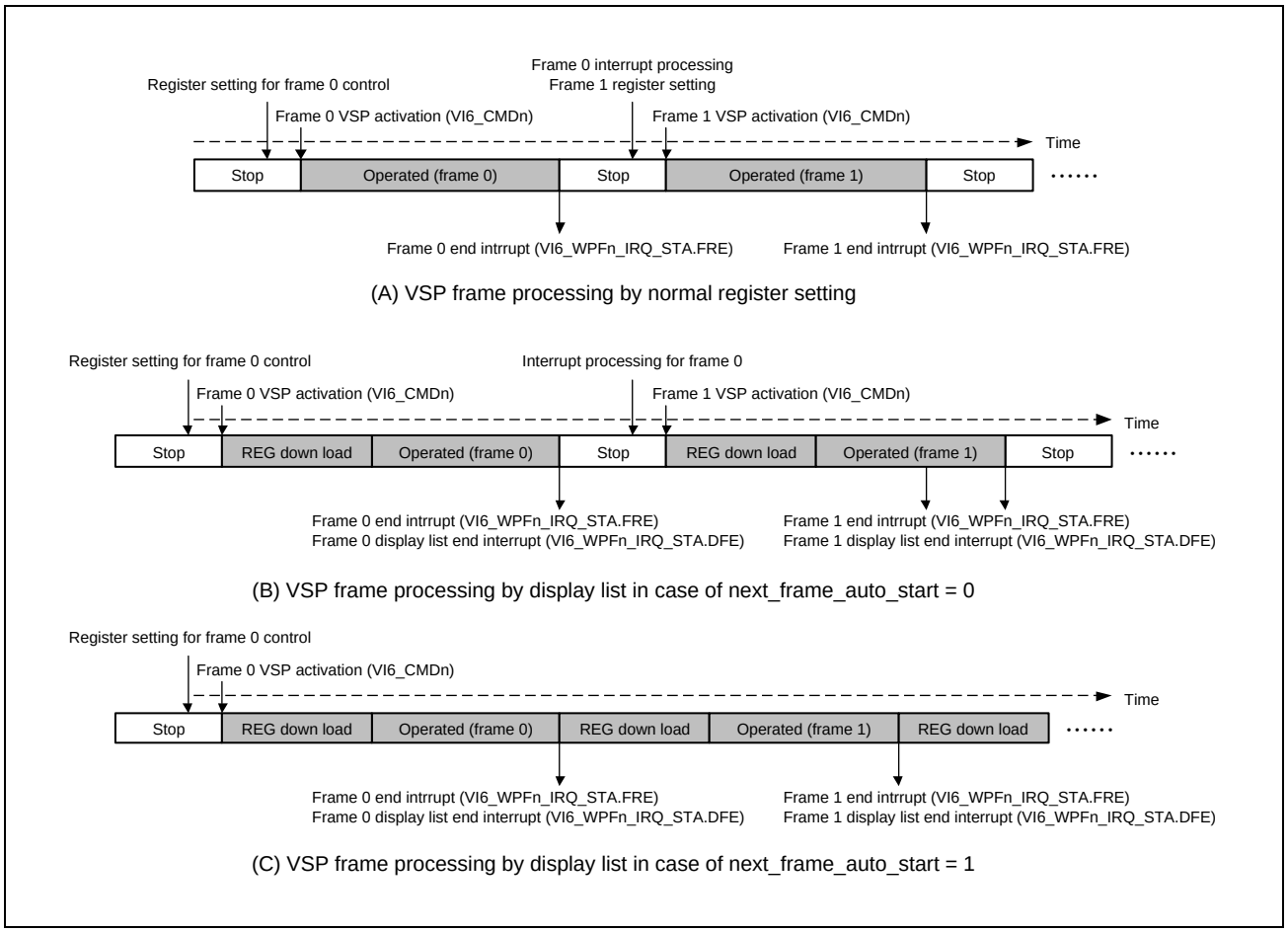



Figure 33.35 Comparison of VSPD Operation between Normal Register Settings and Display Lists

As shown in **Figure 33.35** (A), in the VSPD processing through normal register settings, all registers should be set up before the VSPD is started for each frame. After the VSPD processing is completed, the VSPD outputs a frame end interrupt (VI6_WPFn_IRQ_STA.FRE). This method requires a certain amount of time for register settings or interrupt processing by the CPU between frames. In contrast, when display lists are used, the VSPD automatically downloads register settings from external memory as shown in **Figure 33.35** (B) and (C), which reduces the load on the CPU between frames.

**Figure 33.35** (B) shows the display list usage where the VSPD stops at the end of every frame; only the VSPD start processing for each frame is done by the CPU. This is suitable for the cases when the CPU controls synchronization of frame processing in frame buffer management or when the amount of register values or table data to be set in the VSPD is large. In the case shown in **Figure 33.35** (C), as soon as the frame processing ends, the VSPD automatically begins next frame operation and starts downloading new register settings. This is the fastest operation using display lists.

**Table 33.29** shows the modes of the display list and the supported functions for each mode. The detail of each mode is described in the following sections.

Table 33.29 Display list mode and supported functions

| Mode | Extended Display List | Continuous Frames |
|---|---|---|
| Normal Display List Mode | Supported | Controlled by “next frame auto start” in the header of the display list. |
| Header-less Display List Mode | Not Supported | Controlled by VI6_DL_CTRL.CFM0 bit. |

### 33.4.5.2 Normal Display List Mode

The VSPD display lists include control information as well as simple register settings in order to control multiple-frame processing in an optimum way for each application. **Figure 33.36** shows the display list structure.

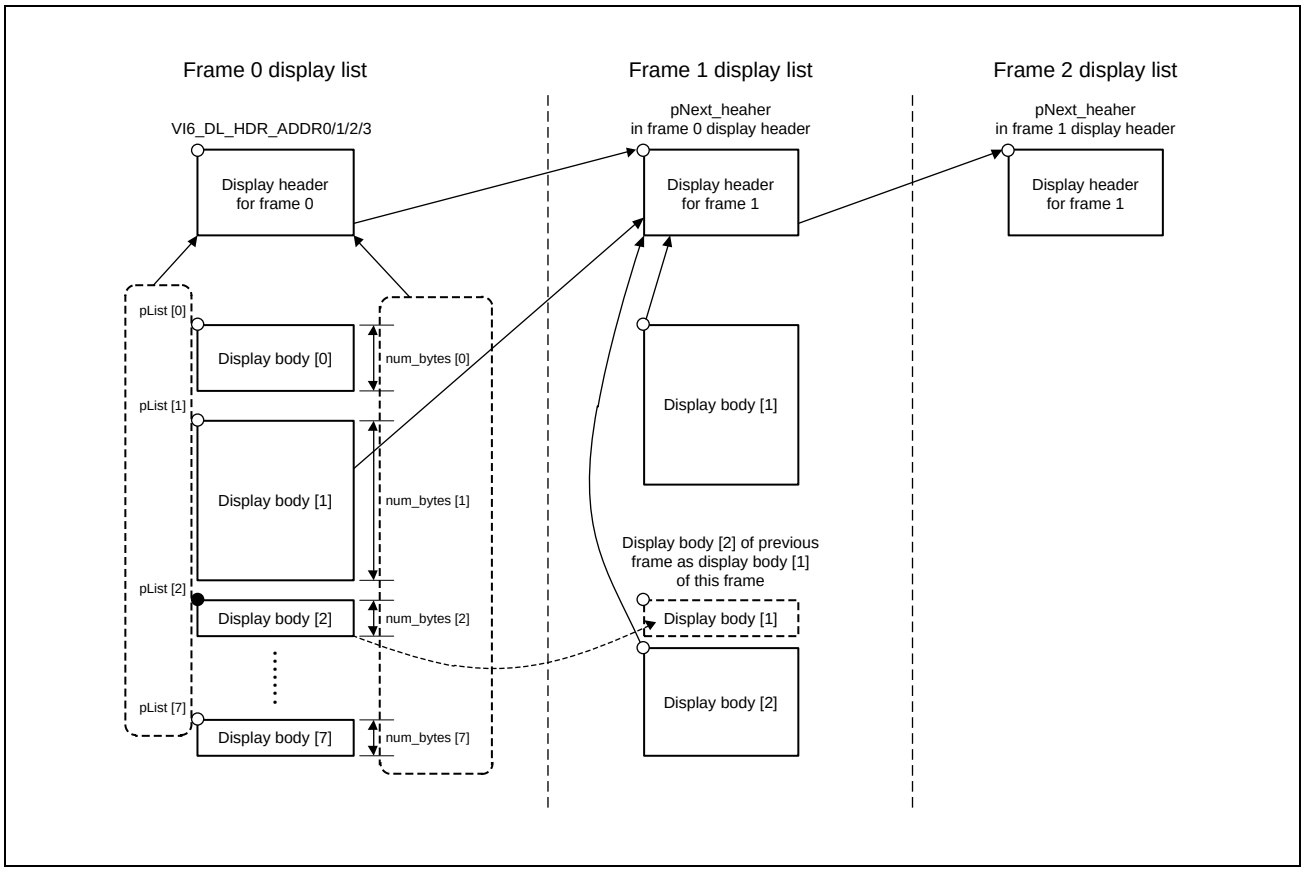



Figure 33.36 Structure and Concept of VSPD Display List

A VSPD display list consists of two sections; a header section for storing various information and control flags and a body section for storing register and table settings. A combination of these two sections is defined as a display list for a frame. The register and table settings can be divided and stored in up to eight separate bodies allocated in memory. Therefore, the separate bodies storing the register settings for one frame (for example, frame 0) can have non-sequential start addresses; that is, the bodies for one frame can be allocated to areas distant from each other in memory. To gather these bodies and configure the register settings for one frame, a header is used. A display header stores the number of bodies linked with the header and the start address and size of each body.

The VSPD analyzes the header to gather register and table settings stored in separate memory areas and reconfigures the complete register settings.

The addresses of display headers should be specified in the VSPD registers described in **Section 33.3.2.5**. When activated in a mode that uses display lists, the VSPD downloads display headers from the addresses specified in VI6_DL_HDR_ADDR0 (number 0 correspond to the WPF channel index numbers), analyzes the numbers of bodies and the address and data size of each body, downloads the bodies, and completes register and table settings. After display list downloading is completed, the VSPD becomes ready for frame image processing; the VSPD then starts the actual frame processing.

After processing of a frame ends, the VSPD proceeds to the next frame processing. Here, there are two modes for starting the next frame processing as shown in **Figure 33.35**. In one mode, the VSPD stops operation and waits for the next activation by the CPU in the same way as when display lists are not used. In this mode, the address used for

downloading the display header of the next frame is kept by the internal hardware. Therefore, if valid address information is not stored in the display header for the previous frame, a correct value should be specified in VI6_DL_HDR_ADDR0 while the VSPD is stopped. When the VSPD is started after a correct value is specified, the VSPD starts next frame processing with the same procedure for the previous frame. In contrast to this mode, which stops the VSPD after the end of one-frame processing, there is another mode for automatically starting next frame processing. In automatic start mode, the VSPD downloads the next display header as soon as the previous frame processing ends. After downloading ends, the VSPD starts image processing. The information regarding mode selection, that is, whether to automatically start next frame processing, should be stored in the display header downloaded for the previous frame.

In the automatic start mode, the VSPD continues processing until the display header for a frame specifies that the next frame should not be started automatically. To stop processing during automatic execution, use a software reset (VI6_SRESET).

To strictly define the display list format described above, the following shows the grammatical structure of a display list using pseudo-code. First, to simplify the description of the display list format in the following pages, **Table 33.30** defines a function. Function zero bits (num bits) generates a string of one-bit 0s for the number of bits specified by the parameter for the function. By using this function, **Table 33.31** defines the header section format of a display list and **Table 33.32** defines the body section format and **Table 33.33** defines the extended display list body section format.

Data order of Display List Header Section, Display List Body Section and Extended Display List Body Section are assumed as the data is stored in SDRAM by big endian. **Table 33.34** shows an example of data order. If data order is not same as the definition, data order within 8byte unit can be adjusted by setting VI6_DL_SWAP (see the **Section 33.3.2.5(3)**).

Table 33.30 Definition of a Function for Simple Description

| Syntax | Bit Count |
|---|---|
| zero_bits (num_bits) | |
| { | |
| for (i = 0; i<num_bits; i++) { | |
| **zero_bit** | 1 |
| } | |
| } | |

| Bit String | Contents |
|---|---|
| zero_bit | zero_bit indicates a 1-bit integer having a value of 0. |

Table 33.31 Format of Display List Header Section

| Syntax | Bit Count |
|---|---|
| display_header () /* Fixed length */ | |
| { | |
| zero_bits (29) | |
| **num_list_minus1** | 3 |
| | |
| for (i=0; i<8; i++) | |
| zero_bits (15) | |
| **num_bytes [i]** | 17 |
| **pList [i]** | 32 |
| } | |
| **pNext_header** | 32 |
| zero_bits (23) | |
| **wait_wup** | 1 |
| zero_bits (3) | |
| **ignore_upd_dl** | 1 |
| zero_bits (2) | |
| **current_frame_int_enable** | 1 |
| **next_frame_auto_start** /* 76 bytes from the beginning of this header*/ | 1 |
| | |
| if (VI6_DL_EXT_CTRL.EXT) { | |
| zero_bits (32) /* padding zero 4 bytes for alignment */ | |
| zero_bits (6) | |
| **pre_ext_dl_exec** | 1 |
| **post_ext_dl_exec** | 1 |
| zero_bits (8) | |
| **pre_ext_dl_num_cmd** | 16 |
| **pre_ext_dl_pList** | 32 |
| zero_bits (16) | |
| **post_ext_dl_num_cmd** | 16 |
| **post_ext_dl_pList** /* 96 bytes from the beginning of this header*/ | 32 |
| } | |
| } | |

| Bit String | Contents |
|---|---|
| num_list_minus1 | Specifies the value obtained by subtracting 1 from the total number of display list bodies linked with the display header. For example, when this bit field is set to 0, this display list uses one body. |
| num_bytes [i] | Specifies the number of bytes in the i-th display list body (indicated by index i). Be sure to specify a multiple of eight bytes. For the bodies that are not defined in num_list_minus1 (for example, i = 5 to 7 when num_list_minus1 is set to 4), specify 0. |
| pList [i] | Specifies the start address of the i-th display list body (indicated by index i). Be sure to specify an address aligned with an 8-byte boundary (the lower-order three bits are 0). For the bodies that are not defined in num_list_minus1 (for example, i = 5 to 7 when num_list_minus1 is set to 4), specify 0. |
| pNext_header | Specifies the address of the display list header for the next frame. After display list downloading ends, the VSPD keeps its value in the internal memory and uses it in the next display list header downloading.<br>Be sure to specify an address aligned with an 8-byte boundary (the lower-order three bits are 0). |
| wait_wup | When this bit is 1, VSPD starts reading image data from external memory after WUP (Wake Up) signal from any channel of VIN asserts. This bit is available for WPF0. |
| ignore_upd_dl | If this bit is set to 1, new display list pointed by VI6_DL_HDR_ADDR0 is not downloaded in VSPD H/W for the next frame, even if S/W sets VI6_DL_HDR_ADDR0 during the frame. New display list is downloaded in VSPD H/W at next frame of the processing which this bit is 0. This bit is available for WPF0. |
| current_frame_int_enable | This is a flag that indicates whether to set the display list end interrupt source (VI6_WPFn_IRQ_STA.DFE) to 1 when the current frame processing ends. If this flag is set to 0, the display list end interrupt source (VI6_WPFn_IRQ_STA.DFE) is not set to 1 when one-frame processing by this display header ends. In this state, even if the display list end interrupt is enabled (VI6_WPFn_IRQ_ENB.DFEE is set to 1), no interrupt will be generated.<br>If this flag is set to 1, the display list end interrupt source (VI6_WPFn_IRQ_STA.DFE) is set to 1 when one-frame processing by this display header ends. In this state, if the display list end interrupt is enabled (VI6_WPFn_IRQ_ENB.DFEE is set to 1), the VSPD generates an interrupt. |
| Next_frame_auto_start | Enables or disables automatic start of next frame processing when one-frame processing by this display header ends. If this bit is set to 1, the VSPD starts next frame processing as soon as one-frame processing by this display header ends, and starts downloading the next frame display header from the pNext_header address specified in this display header. If this bit is set to 0, the VSPD stops operation when one-frame processing by this display header ends. In this case, start the VSPD through VI6_CMDn to process the next frame. |
| pre_ext_dl_exec | Enables execution of the extended display list for frame preprocessing when VI6_DL_EXT_CTRL.EXT is 1. If this bit is set to 1, the VSPD executes the extended display list for frame preprocessing. The VSPD does not execute it if this bit is set to 0. This bit setting is ignored when VI6_DL_EXT_CTRL.EXT is 0. |
| post_ext_dl_exec | Set 0 to this bit. |
| pre_ext_dl_num_cmd | Specifies the number of commands in the extended display list body section for frame preprocessing when VI6_DL_EXT_CTRL.EXT is 1. The number of commands that can be specified is 1, and a command is 16 bytes.<br>When pre_ext_dl_exec is set to 0, the extended display list for frame preprocessing is not executed; specify 0 in this bit. This bit setting is ignored when VI6_DL_EXT_CTRL.EXT is 0. |
| pre_ext_dl_pList | Specifies the start address of the area where the extended display list body section for frame preprocessing is stored when VI6_DL_EXT_CTRL.EXT is 1. Be sure to specify an address aligned with a 16-byte boundary (lower-order four bits are 0). When pre_ext_dl_exec is set to 0, the extended display list for frame preprocessing is not executed; specify 0 in this bit. This bit setting is ignored when VI6_DL_EXT_CTRL.EXT is 0. |
| post_ext_dl_num_cmd | Set 0 to this bit. |
| post_ext_dl_pList | Set 0 to this bit. |

Table 33.32 Format of Display List Body Section

| Syntax | Bit Count |
|---|---|
| display_list (num_bytes) /* Variable length (num_bytes) */ | |
| { | |
| for (i =0; i<num_bytes; i+=8) { | |
| **set_address** | 32 |
| **set_data** | 32 |
| } | |
| } | |

| Bit String | Contents |
|---|---|
| set_address | Specifies the address where the value specified by set_data is to be stored. Specify a register address.<br>Note :<br>Register space of VSPD is 32Kbyte (H'0000 - H'7FFC). So, set 0 to upper 17bit of set_address in Display List Body Section. |
| set_data | Specifies the value to be stored in the address specified by set_address. Specify a value to be set in a register. |

Table 33.33 Format of Extended Display List Body Section

| Syntax | Bit Count |
|---|---|
| ext_dl_display_list (num_llw) /* Variable length (pre/post_ext_dl_num_bytes) */ | |
| { | |
| for (i=0; i<num_llw; i+=2) { | |
| **ext_dl_cmd** | 64 |
| **ext_dl_data** | 64 |
| } | |
| } | |

Table 33.34 Data order of display list body

| Address | Syntax |
|---|---|
| pList | set_address [0] (bit31-24) |
| pList + 1 | set_address [0] (bit23-16) |
| pList + 2 | set_address [0] (bit15-8) |
| pList + 3 | set_address [0] (bit7-0) |
| pList + 4 | set_data [0] (bit31-24) |
| pList + 5 | set_data [0] (bit23-16) |
| pList + 6 | set_data [0] (bit15-8) |
| pList + 7 | set_data [0] (bit7-0) |
| pList + 8 | set_address [1] (bit31-24) |
| pList + 9 | set_address [1] (bit23-16) |
| pList + 10 | set_address [1] (bit15-8) |
| pList + 11 | set_address [1] (bit7-0) |
| pList + 12 | set_data [1] (bit31-24) |
| pList + 13 | set_data [1] (bit23-16) |
| pList + 14 | set_data [1] (bit15-8) |
| pList + 15 | set_data [1] (bit7-0) |
| ~~~ | ~~~ |

#### 33.4.5.3 Header-less Display List Mode

The header-less display list does not have the display header listed in **Table 33.31**, and it has the simplest structure that has only single body. The extended display list function is not available in case of the header-less display list mode. Set the value 1 to VI6_DL_CTRL.NH0 (see the **Section 33.3.2.5(1)**) to use the header-less display list. The start address downloaded by the header-less display list should be set to VI6_DL_HDR_ADDR0. And the size of the display body which is originally defined in the display header should be set to VI6_DL_BODY_SIZE0. The header-less display list is available only in WPF0.

#### 33.4.5.4 Restrictions on Display List Usage

Access to the general control registers and display list control registers through a display list is prohibited. When using display lists, be sure to observe the following restrictions on register access by the CPU.

1. Do not execute write access to the same register (same address) from the CPU and through a display list at the same time. If such a conflict occurs, correct operation of the VSPD is not guaranteed.
2. Do not execute write access to the same LUT lookup table from the CPU and through a display list at the same time. If such a conflict occurs, correct operation of the VSPD is not guaranteed. Here, the same lookup table means the address space having the same space name shown in **Table 33.35**.
3. When read access by the CPU and write access through a display list to the same register (same address) occur at the same time, the read value returned to the CPU is not guaranteed.
4. When read access by the CPU and write access through a display list to the same LUT lookup table occur at the same time, the read value returned to the CPU is not guaranteed. Here, the same lookup table means the address space having the same space name shown in **Table 33.35**.
5. For other restrictions on values and timing of register setting through display lists, refer to the restrictions on normal register settings described in **Section 33.3.2.3**.

6. Manipulation and setting of the registers described in **Section 33.3.2.4** and **Section 33.3.2.5** through a display list is prohibited.
7. The extended display list function enabled by VI6_DL_EXT_CTRL.EXT is to control the registers of the ICB connected with the VSPD. Do not use extended display lists for any other purpose.

### 33.4.6 Interrupt Processing

When a WPF module generates an internal source that should be notified, an interrupt signal is output. As internal sources are generated in WPF independently, the VSPD has the following registers to control interrupts in WPF.

WPFn Interrupt Enable Registers (VI6_WPFn_IRQ_ENB: n = 0) (refer to **Section 33.3.2.4(12)**)

WPFn Interrupt Status Registers (VI6_WPFn_IRQ_STA: n = 0) (refer to **Section 33.3.2.4(13)**)

**CAUTION**

To use interrupts, enable them through WPF interrupt enable registers. If an interrupt source register has already been set to 1 for some reason, an unintended interrupt will occur as soon as the corresponding interrupt enable register is set as enabled. To avoid this, before enabling interrupts through WPF interrupt enable registers, be sure to clear all WPF interrupt sources to be enabled to 0. Be careful about this procedure when setting up registers before starting the VSPD.

## 33.4.7 Lookup Table Settings

### 33.4.7.1 LUT

For a single entry to the LUT space (see **Table 33.35**) of the VSPD, the LUT data is set by a write access in the format shown in **Figure 33.37**.

The entry address of each space is (start address of the space) + (entry number counting from the base point 0 × 4). For example, the address of entry 7 to the LUT space is (offset address) + H'7000 + 7 × 4 = (offset address) + H'701C.

**Table 33.35** shows the spaces for which entries can be made. Note that if the module that references that space is operating, write accesses to the relevant space are prohibited. For example, while RPF2 is operating, accesses to the entire space of VI6_CLUT2_TBL are prohibited. When a read access is made to the relevant space during operation of the referencing module, undefined values will be read out.

The operating/stopped state of each module in **Table 33.35** is the operating state of the WPF to which each module is connected. Determine whether the module is operating or stopped using each WPF operating status bit in the VI6_STATUS register.

Table 33.35 LUT Space Addresses

| Space Name | Space Addresses | Entry Count | Module that References the Space in the Left Column |
|---|---|---|---|
| VI6_LUT_TBL | (offset address) + H'7000 to (offset address) + H'73FC | 256 | LUT |

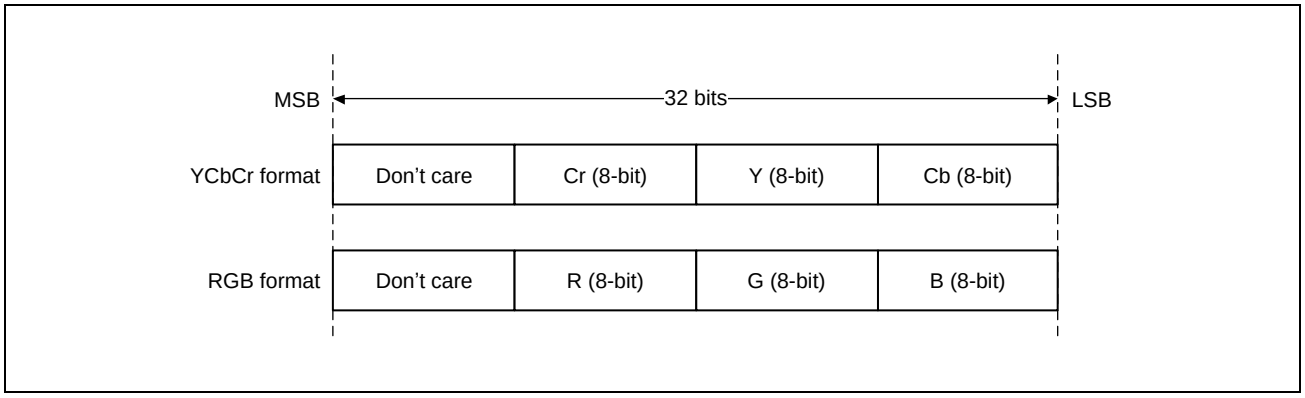



Figure 33.37 VI6_LUT_TBL Formats

## 33.4.8 Linked with DU

### 33.4.8.1 Operation flow of VSPD and DU

This section shows operation flow of VSPD by using Normal Display List Mode. For details of Normal Display List Mode, refer to **Section 33.4.5.2**.

**Figure 33.38** and **Figure 33.39** shows a procedure (register setting flow) to display image data to display panel.

**Figure 33.38** shows setting flow to start VSPD/LIF0 linked with DU. **Figure 33.39** shows setting flow to update display image.

Using display list with enabling next_frame_auto_start shown in **Section 33.4.5** is required to take the flow from **Figure 33.38** and **Figure 33.39**.

NOTE

If display sync operation ([1-8] step in **Figure 33.38**) is set before activating the VSPD, invalid data is displayed until image data read from external memory is ready to be transferred to DU. In this case, image data from external memory is displayed late by one vsync or more.

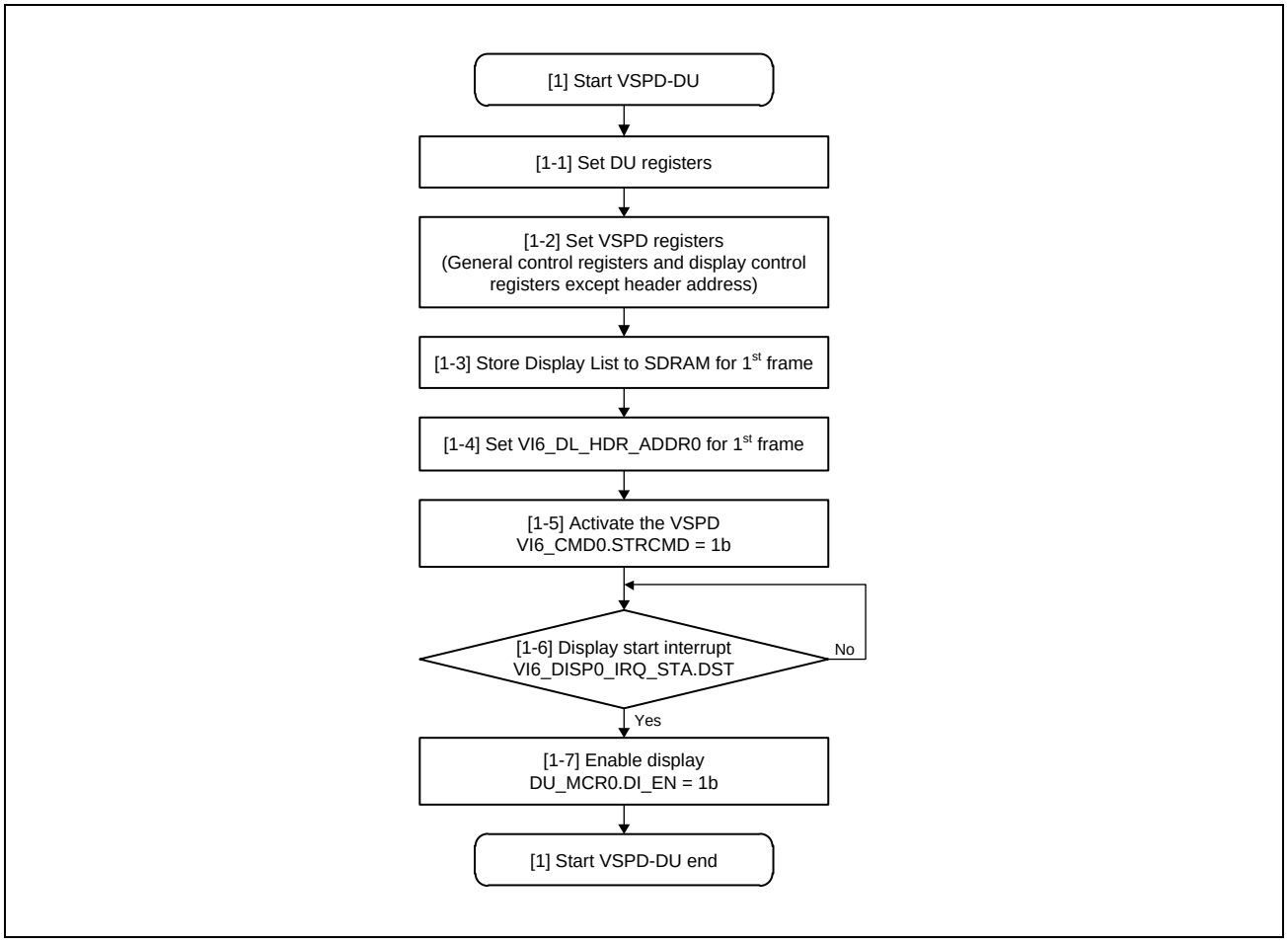



Figure 33.38 Setting flow to start VSPD linked with DU

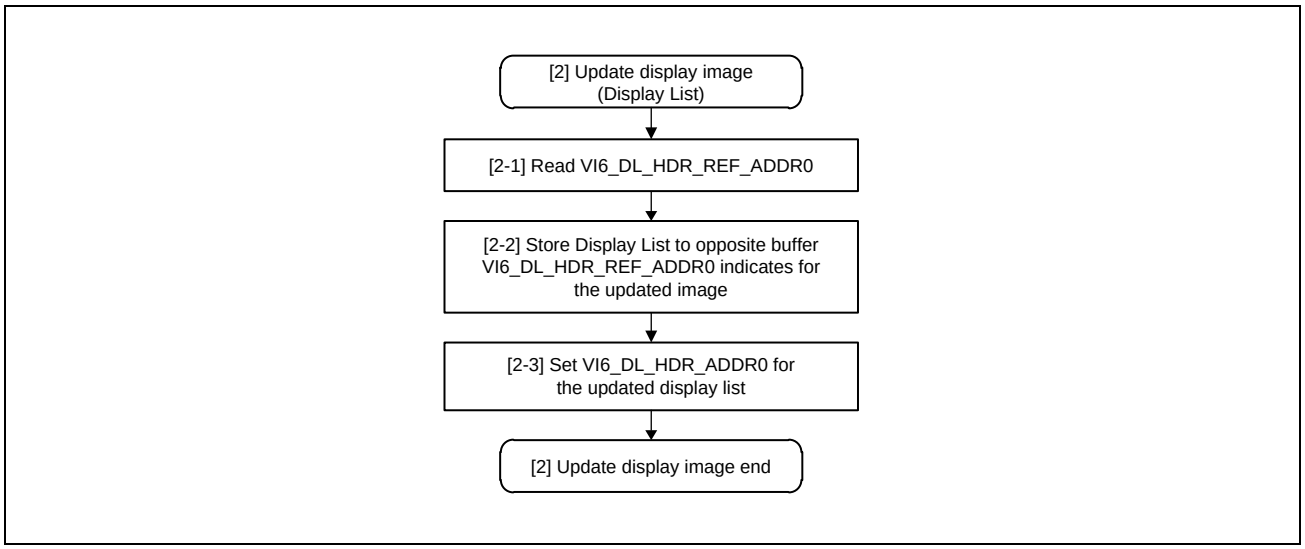



Figure 33.39 Setting flow to update display image

It is possible to update display list at any time (arbitrary timing) as shown in **Figure 33.39**. Don't overwrite display list in external memory pointed by VI6_DL_HDR_REF_ADDR. It is because VSPD is reading the display list pointed by VI6_DL_HDR_REF_ADDR or will be reading the display list soon.

Detail explanation is shown below.

As shown in **Section 33.3.2.5(6)**, when VSPD is reading display list from external memory (Period [A] in **Figure 33.40**), VI6_DL_HDR_REF_ADDRn indicates header address of the display list referred by VSPD. When VSPD is not reading display list from external memory (Period [B] in **Figure 33.40**), VI6_DL_HDR_REF_ADDRn indicates the value of VI6_DL_HDR_ADDRn, and there is possibility that the display list is being read soon by VSPD at next frame start timing. Therefore, keep display list in external memory pointed by VI6_DL_HDR_REF_ADDR. Detail software sequence to update display list and VSPD -H/W behavior are shown below.

[1] Store latest display list in external memory area different from the area pointed by VI6_DL_HDR_REF_ADDR, because the area pointed by VI6_DL_HDR_REF_ADDR is being read or is being read soon by VSPD.

[2] Set header address of new display list into VI6_DL_HDR_ADDRn.

[3] VSPD reads display list from the external memory area pointed by VI6_DL_HDR_ADDRn at frame start timing.

Refer also to **Section 33.3.2.5(2)** and **Section 33.3.2.5(6)**.

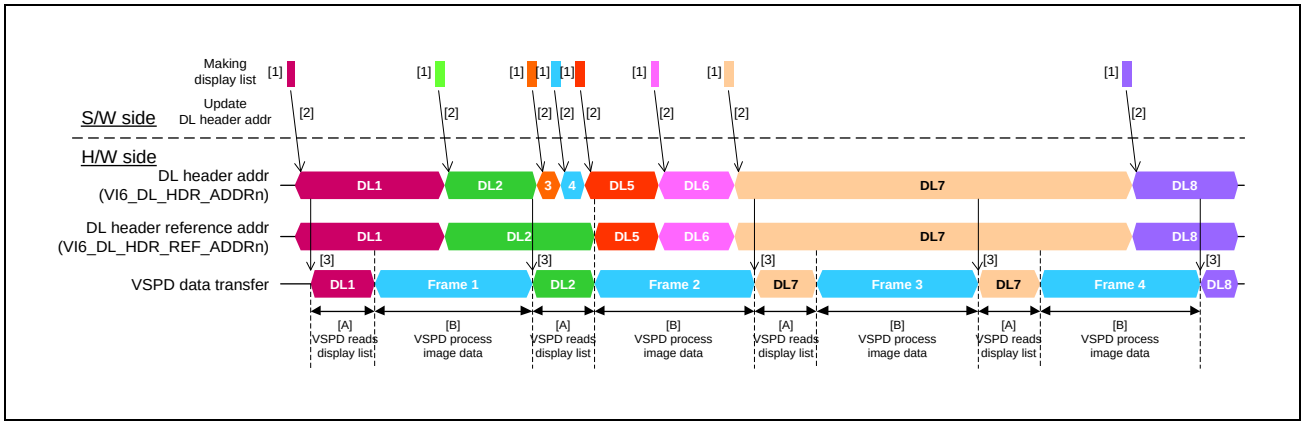
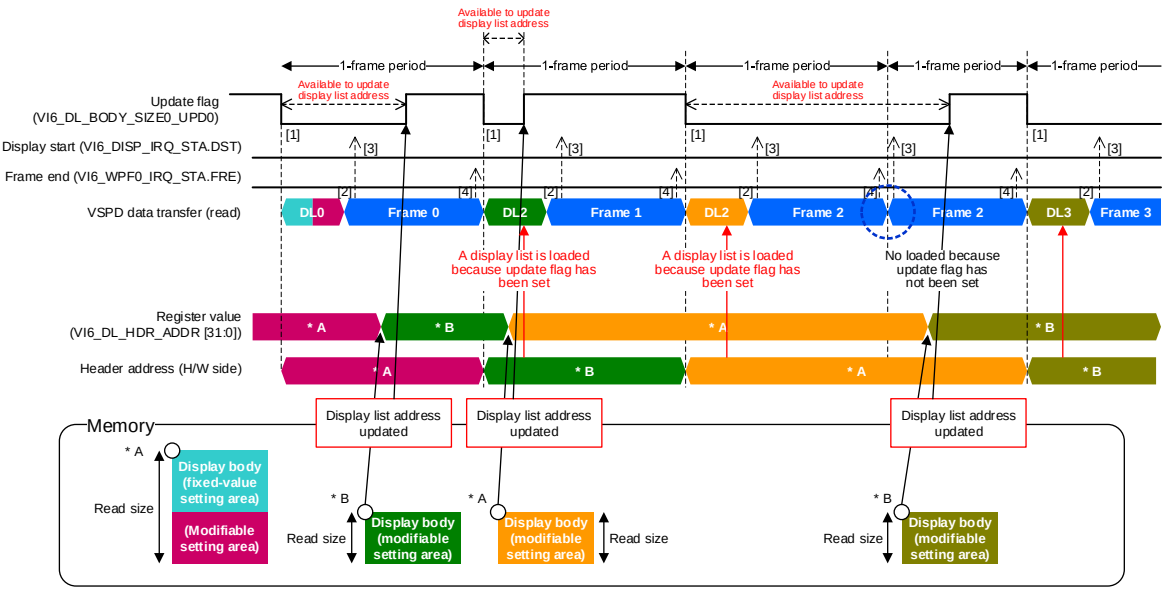



Figure 33.40 Updating display list at arbitrary timing

### 33.4.8.2 Controlling Two Register Planes Using Display Lists

This section shows operation flow of VSPD by using Header-less Display List Mode. Start procedure and stop procedure are same with Normal Display List Mode in **Section 33.4.8.1** except for the point that VI6_DL_BODY_SIZE0 should be set before step [1-5] in start procedure shown in **Figure 33.38**. The procedure to update display list is different with Normal Display List Mode, and is described later. **Figure 33.41** shows the control of two register planes using header-less display lists (see **Section 33.4.5.3**) and its timing. In the description hereafter, the use of header-less display lists is always assumed and they are simply called display lists.




Display list plane A: “Fixed-value settings” and “modifiable settings” are stored as a set in contiguous area A.
Display list plane B: Only the “modifiable settings” are stored in area B, which is separate from display list A.

To switch settings for every frame, modify two display list planes alternately. After every modification, update the display list address register (VI6_DL_HDR_ADDR0 [31:0]) and writing 1 to update flag (VI6_DL_BODY_SIZE0.UPD0) in the VSPD.
Update flag (VI6_DL_BODY_SIZE0.UPD0) is negated automatically after header address is loaded into H/W side.

Figure 33.41 Controlling Two Register Planes Using a Display List

The VSPD downloads a display list immediately after activation. In the start frame, download the display list that contains all necessary settings. From the next frame on, only the necessary register or table values should be specified in a display list.

When the update flag of VI6_DL_BODY_SIZE0.UPD0 is set to 1, VSPD downloads a new display list at the start of the next frame. When the update flag of VI6_DL_BODY_SIZE0.UPD0 is set to 0, the register settings acquired from the display list previously downloaded are retained and used for the next operation without downloading a new display list.

Order of each VSPD -H/W Events mentioned as [1], [2], [3] and [4] in **Figure 33.41** are shown below.

[1] VI6_DL_BODY_SIZE0.BS0 and VI6_DL_HDR_ADDR0 are downloaded in H/W side. And Update Flag (VI6_DL_BODY_SIZE0.UPD0) is negated automatically.

[2] Reading DisplayList from external memory into VSPD -H/W is finished through master access.

[3] Display start interrupt status (VI6_DISP_IRQ_STA.DST) asserts.

[4] Frame end interrupt status (VI6_WPF0_IRQ_STA.FRE) asserts.

Don’t overwrite the display list in external memory for a period from “[4] VI6_WPF0_IRQ_STA.FRE” to next “[3] VI6_DISP_IRQ_STA.DST”, because VSPD is reading the display list for the period.

## 33.5 Usage Note

### 33.5.1 Assignment in Memory Space

Make sure that VSPD memory space shall be mapped to Non-Cache region.

### 33.5.2 Limitations on Software Reset and Module Standby

Stop LCDC with the software reset sequence following to the guidance in the **Section 33.4.1**.

### 33.5.3 Input Image Size

**Table 33.36** is a list of input size specifications.

Table 33.36 List of Input Size Specifications

| Module | Min. Input Size | Max. Input Size | Restriction on Setting Unit |
|---|---|---|---|
| RPF | 1 (horizontal)<br>× 1 (vertical) pixel | 1280 (horizontal)<br>× 800 (vertical) pixels | YCbCr422: 2-pixel units horizontally and 1-pixel units vertically.<br>YCbCr420: 2-pixel units both horizontally and vertically.<br>Other formats: 1-pixel units both horizontally and vertically.<br>*Note 1.* When the 1-bpp alpha plane*1 is input, the size can always be specified in 8-pixel units both horizontally and vertically regardless of the input format.<br>*Note 2.* These restrictions including note 1 are applied to the following.<br>VI6_RPFn_SRC_BSIZE<br>VI6_RPFn_SRC_ESIZE |
| LUT | 1 (horizontal)<br>× 1 (vertical) pixel | 1280 (horizontal)<br>× 800 (vertical) pixels | 1-pixel units both horizontally and vertically. |
| BRS | 1 (horizontal)<br>× 1 (vertical) pixel | 1280 (horizontal)<br>× 800 (vertical) pixels | 1-pixel units both horizontally and vertically. |
| LIF | 1 (horizontal)<br>× 1 (vertical) pixel | 1280 (horizontal)<br>× 800 (vertical) pixels | 1-pixel units both horizontally and vertically. |
| WPF | 1 (horizontal)<br>× 1 (vertical) pixel | 1280 (horizontal)<br>× 800 (vertical) pixels | YCbCr422: 2-pixel units horizontally and 1-pixel units vertically.<br>YCbCr420: 2-pixel units both horizontally and vertically.<br>Other formats: 1-pixel units both horizontally and vertically.<br>*Note 1.* This restriction on the WPF only applies to the WPF output size. The WPF input size should be specified in 1-pixel units. |

Note 1. When VI6_RPFn_ALPH_SEL.ASEL is set to 011b.

The most important restriction shown in **Table 33.36** is the setting unit. Make appropriate register settings so that the size of the image input to each module does comply with setting units and does not exceed the limits shown in **Table 33.36**.

### 33.5.4 Output Image Size

The size of the output from each WPF is determined by the results of processing in the modules connected with the DPR. As shown in **Figure 33.1**, the data input to the VSPD is sent to the WPF output modules through the RPF and the modules connected with the DPR. When there is no processing that changes the image size through this data path, the WPF output size is the same as the RPF input size. **Table 33.37** is a list of the processing that changes image size.

Table 33.37 Image Processing that changes image size

| Module | Function*1 | Related Register | Size of Output from Module |
|---|---|---|---|
| WPF | Input size clipping | VI6_WPFn_HSZCLIP<br>VI6_WPFn_VSZCLIP | When this function is disabled, the input size and the output size are the same. When this function is enabled, the output is in the following size.<br>Horizontal output size: VI6_WPFn_HSZCLIP.HCL_SIZE x 2 setting<br>Vertical output size: VI6_WPFn_VSZCLIP.VCL_SIZE x 2 setting |
| LIF | Padding line | VI6_LIFn_CTRL | When padding with dummy lines is disabled, output size is same with Input size.<br>When padding with dummy lines is enabled:<br>Horizontal output size: input size<br>Vertical output size: VI6_LIF0_PADLN_SIZE setting |

Note 1. For details of each function, refer to the descriptions of the related registers.

Note 2. Refer to **Section 33.3.2.1** for explanation of register bit field.

The input image size can be changed only with the modules and functions shown in **Table 33.37**. With the other modules and functions, the input size and output size are the same. Accordingly, after module connections with the DPR are determined, the VSPD output size can also be determined through the following steps.

1. The image size (VI6_RPFn_SRC_ESIZE) read from the external memory and sent to the DPR by the RPFn is the initial value.
2. When the module connected with the DPR is not a module (function) shown in **Table 33.37**, the output size from the module is the same as the input size; the image size does not need to be updated.
3. When a module connected with the DPR is a module (function) shown in **Table 33.37**, the output image size should be updated to the size shown in the table, which should be used as the input image size for the module connected behind it.
4. When the final image size at the WPFn is determined, this size is the VSPD output size for that WPFn path.

**Figure 33.42** shows how to determine the image size in a sample DPR connection through the above steps. The related conditions that determine the image size are also shown.

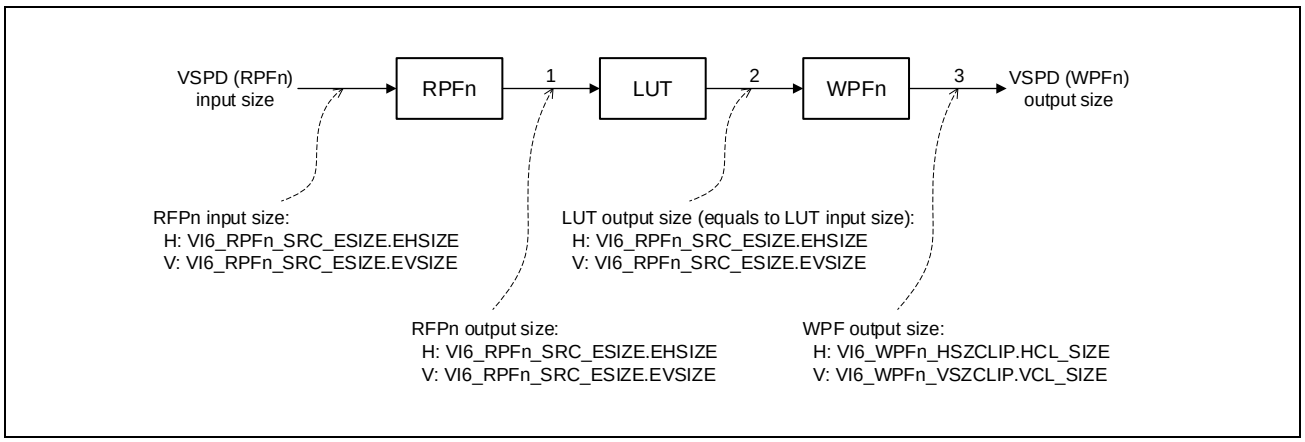



Figure 33.42 Input/output Size for Each Module in a Sample DPR Connection

Make appropriate register settings in each module so that the VSPD output image size determined as shown in the figure does not violate the restrictions shown in **Table 33.37**.

### 33.5.5 Restriction

When the data output from this unit is written back to the same memory area where the input data for this unit has been read, this unit has the following restrictions.

1. The access order and format on the frame memory are the same between the input pixels and output pixels.
2. Specifying a larger output image size than the input image size, either vertically, horizontally, or both vertically and horizontally, is prohibited.
3. When the YCbCr4:2:0 format is input, operation between color components is prohibited.

These restrictions are summarized in. Refer to the descriptions of the registers related to each restriction. In the table, RPFm indicates the RPFn that inputs the master layer, and WPFwb indicates the WPFn that writes back the output image to the source image area for the master layer.

Table 33.38 Restrictions on Use when Output Data is Written Back to Input Data Area

| No. | Restriction | Related Registers |
|---|---|---|
| Restriction 1 | The RPFm input format and the WPFwb output format should be the same. | VI6_RPFn_INFMT.RDFMT<br>VI6_RPFn_INFMT.VIR |
| | The RPFm source image storing address and the WPFwb destination address should be the same. | VI6_RPFn_SRCM_ADDR_* |
| | The RPFm source picture memory stride and the WPFwb destination memory stride should be the same. | VI6_RPFn_SRCM_PSTRIDE<br>VI6_RPFn_SRCM_ASTRIDE |
| | The RPFm and WPFwb data swapping settings should be the same. | VI6_WPFn_DSWAP |
| Restriction 2 | The RPFm basic read size and extended read size should be the same. | VI6_RPFn_SRC_BSIZE<br>VI6_RPFn_SRC_ESIZE |
| | Color space conversion is prohibited in RPFm and WPFwb. | VI6_RPFn_INFMT.CSC<br>VI6_WPFn_OUTFMT.CSC |
| Restriction 3*1 | NOP should be specified for IROP operation. | VI6_RPFn_ALPH_SEL.IROP |
| | Color keying is prohibited. | VI6_RPFn_CKEY_CTRL.CV |

Note 1. Note: When the input format is not YCbCr4:2:0, restriction 3 in not applied.

# 34. MIPI DSI

## 34.1 Overview

This chapter describes the features of the MIPI-DSI unit of this LSI.

This unit is the MIPI-DSI Tx module, it is composed of MIPI DSI-2 Host Controller (LINK), and MIPI D-PHY Tx (D-PHY).

This unit supports MIPI Alliance Specification for Display Serial Interface (DSI) Specification. This unit provides a solution for transmitting MIPI DSI compliant digital video and packets. Normative References are below.

- MIPI Alliance Specification for Display Serial Interface Version 1.3.1
- MIPI Alliance Specification for D-PHY Version 2.1

### 34.1.1 Features

The following is key features of this unit.

In this document the word “DSI” means “Display Serial Interface Version 1.3.1”.

■ **Overview**

- 1 channel
- The number of Lane: 4-lane
- Support up to WXGA (1280 × 800), 60 fps (RGB888)
- Maximum Bandwidth: 1.5 Gbps per lane
- Support Output Data Format: RGB666 / RGB888

■ **Protocol Interface**

- PPI Interface (8bit)
- Packet generation from Video Input signal for Video Mode (Video-Input Operation)
  - Supports RGB format (18bit, 18bit Loosely and 24bit)
  - Not supports RGB format (30bit and 36bit)
  - Not supports YCbCr 4:2:2 format (16bit, 20bit Loosely and 24bit)
  - Not supports YCbCr 4:2:0 format (12bit)
  - Not supports Compressed Pixel Stream
  - Supports Blanking Packet or LP-11 selection during each of HSA, HBP and HFP blanking interval
  - Supports “Non-Burst Mode with Sync Pulse”, “Non-Burst Mode with Sync Event” and “Burst Mode”
  - Not supports interlaced video
  - Not supports deskew pattern insertion after vertical video data timing
- LP only packet generation and LP packet reception from descriptor list (Sequence Operation channel 0)
  - Supports one sequence input channel
- HS or LP packet generation and LP packet reception from descriptor list (Sequence Operation channel 1)
  - Supports one sequence input channel

### ■ Link Layer

- Supports 1, 2, 3 and 4 lane configurations
- Supports unidirectional high-speed mode Tx
- Supports bidirectional LP mode Tx/Rx (Only Lane0)
- Supports ECC/Checksum generation for Tx packet
- Supports ECC/Checksum verification and ECC error correction for Rx packet
- Supports automated insertion of EoTp in HS mode
- Supports ULPS (Tx)
- Supports automated power change to LP mode and return to HS mode
- Supports automated clock stop and resume (non-continuous clock mode)
- Supports assignment for Virtual Channel for video input channel in Video-Input Operation
- Supports assignment for individual Virtual Channel for each packet in Sequence Operation
- Supports detection for PHY contention error and timeout error
- Supports generation of scrambled packets
- Not supports input of TE signal

### ■ PHY Layer

- Number of Lanes
  - Data Lane (DL): 4 Lanes
  - Clock Lane (CL): 1 Lane
- Transfer Rate
  - High Speed (HS) Mode: 80 – 1500Mbps
- Lane Module Functions
  - DL: CIL-MFAA (HS-TX, LP-TX, LP-RX, LP-CD)
  - CL: CIL-MCNN (HS-TX, LP-TX)
- Optional Functions
  - Escape Mode
    - LPDT
      - LP Transmit: Supported
      - LP Receive: Supported
      - ULPS: Supported
      - Reset Trigger: Supported
    - Bi-Direction
      - Bus Turn-Around: Supported
      - Contention Detection: Supported
    - Calibration
      - Deskew Calibration / Preamble: Not-Supported

- Pre/De-Emphasis: Not-Supported
- Equalization: Not-Supported

– Others

- HS-IDLE: Not-Supported
- 16bit, 32bit width of Tx Data: Not-Supported

- Power Supply Voltage (at Core Power Pins, Min/Typ/Max)
  - VDD: Same with the condition of this LSI
  - VCCQLV18: Min:1.65V – Typ:1.80V – Max:1.95V

### 34.1.2 Block Diagram

**Figure 34.1** shows a block diagram.

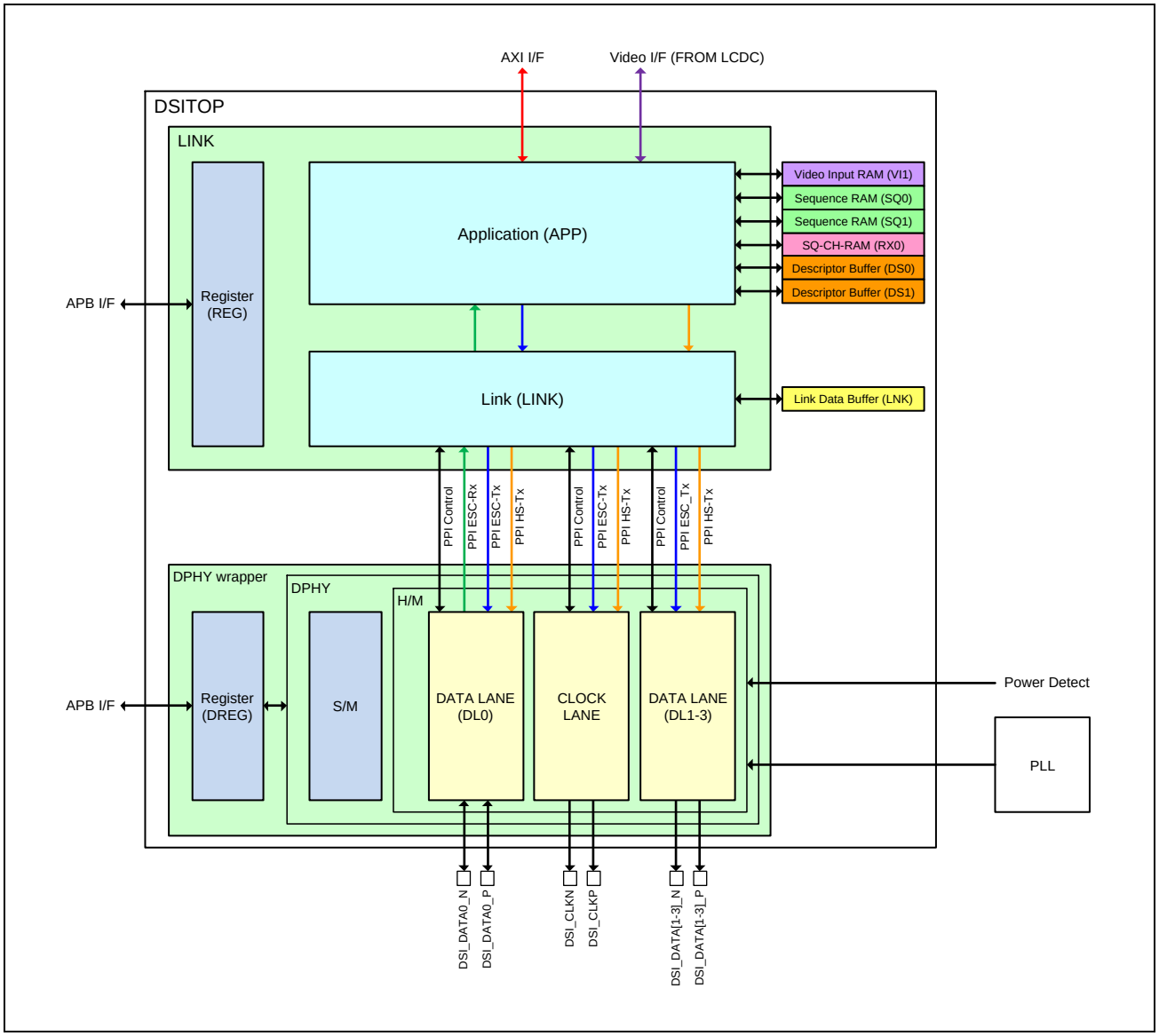



Figure 34.1 Block Diagram

### 34.1.3 Clock

**Table 34.1** shows a list of clocks for this unit.

- $T_{LPX}$: Transmitted length of any Low-Power state period (Min: 50ns)
- $T_{LPX(H)}$: TLPX of Host Processor (the DSI-2 TX module)
- $T_{LPX(P)}$: TLPX of Peripheral device (connected device)

Table 34.1 Clocks and the frequency range

| Clock name | I/O | Description | Frequency (MHz) |
|---|---|---|---|
| MIPI_DSI_PCLK | I | APB clock. | 100/50/25/12.5/3.125 |
| MIPI_DSI_ACLK | I | AXI clock. | 200/100/50/25/6.25 |
| MIPI_DSI_VCLK | I | Video clock. | 148.5~5.803 |
| MIPI_DSI_LPCLK | I | DSI Escape mode Transmit Clock (TxClkEsc).<br>Set D-PHY TxClkEsc ($1/T_{LPX(H)}$) frequency.<br>$T_{LPX(H)}/T_{LPX(P)}$ needs to set between 2/3 to 3/2. | 16.656/8.328/4.164/2.082 |
| MIPI_DSI_SYSCLK | I | System clock for D-PHY. | 133.25 |
| MIPI_DSI_PLLCLK | I | PLL Multiplied clock.<br>{Transfer rate x 2} MHz | 160~3000 |
| hsclk | — | DSI High-Speed Transmit Word Clock (TxWordClkHS).<br>Upper limit for D-PHY 1.5Gbps.<br>Lower limit for D-PHY 80Mbps.<br>This clock is supplied from D-PHY directly, and its frequency is 1/16 for MIPI_DSI_PLLCLK. | 10 to 187.5 |
| lppclk | — | DSI Escape mode Receive Clock (RxClkEsc).<br>This clock is supplied from D-PHY directly, and its frequency is half of ($1/T_{LPX(P)}$).<br>This clock is generated dividing in D-PHY. | 2 to 10 |

Video clock and DSI HS Byte clock must follow the relationship.

Video clock Frequency [Hz] × Video Pixel Bit Depth[bit]

<= DSI HS Byte clock Frequency [Hz] × 8 [bit] × Number of DSI HS Data Lane

### 34.1.4 External Pins

**Table 34.2** shows the pin configuration.

Table 34.2 External Pins

| Pin Name | Input/Output | Function |
|---|---|---|
| DSI_CLKP | Output | MIPI DSI clock lane differential clock (pos) |
| DSI_CLKN | Output | MIPI DSI clock lane differential clock (neg) |
| DSI_DATA0_P | Output | MIPI DSI data lane0 differential data (pos) |
| DSI_DATA0_N | Output | MIPI DSI data lane0 differential data (neg) |
| DSI_DATA1_P | Output | MIPI DSI data lane1 differential data (pos) |
| DSI_DATA1_N | Output | MIPI DSI data lane1 differential data (neg) |
| DSI_DATA2_P | Output | MIPI DSI data lane2 differential data (pos) |
| DSI_DATA2_N | Output | MIPI DSI data lane2 differential data (neg) |
| DSI_DATA3_P | Output | MIPI DSI data lane3 differential data (pos) |
| DSI_DATA3_N | Output | MIPI DSI data lane3 differential data (neg) |

## 34.2 Register Configuration

### 34.2.1 Registers in LINK

This section describes about registers in LINK. All registers can access by 8/16/32 bit.

Base Address: H'0_1086_0000 (Cortex-A55 Address Space)

Table 34.3 Resister list (1/3)

| Register Name | Abbreviation | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|
| Interrupt Status Register | ISR | *2 | H'0000_0000 | H'000 | 8/16/32 |
| Link Status Register | LINKSR | *2 | H'0000_0000 | H'010 | 8/16/32 |
| Tx Set Register | TXSETR | *2 | H'0003_0003 | H'100 | 8/16/32 |
| HS Clock Set Register | HSCLKSETR | *2 | H'0000_0000 | H'104 | 8/16/32 |
| ULPS Set Register | ULPSSETR | *2 | H'0000_00A0 | H'108 | 8/16/32 |
| ULPS Control Register | ULPSCR | *2 | H'0000_0000 | H'10C | 8/16/32 |
| Reset Control Register | RSTCR | *2 | H'0000_0000 | H'110 | 8/16/32 |
| Reset Status Register | RSTSR | *2 | H'0000_0000 | H'114 | 8/16/32 |
| DSI Set Register | DSISETR | *2 | H'80F1_0001 | H'120 | 8/16/32 |
| Receive Buffer Size Register | RXBUFSZR | *2 | H'0000_0000 | H'124 | 8/16/32 |
| Tx Packet Payload Data 0 Register | TXPPD0R | *2 | H'0000_0000 | H'160 | 8/16/32 |
| Tx Packet Payload Data 1 Register | TXPPD1R | *2 | H'0000_0000 | H'164 | 8/16/32 |
| Tx Packet Payload Data 2 Register | TXPPD2R | *2 | H'0000_0000 | H'168 | 8/16/32 |
| Tx Packet Payload Data 3 Register | TXPPD3R | *2 | H'0000_0000 | H'16C | 8/16/32 |
| Rx Status Register | RXSR | *2 | H'0000_0000 | H'200 | 8/16/32 |
| Rx Status Clear Register | RXSCR | *2 | H'0000_0000 | H'204 | 8/16/32 |
| Rx Interrupt Enable Register | RXIER | *2 | H'0000_0000 | H'208 | 8/16/32 |
| Peripheral Response Timeout BTA Set Register | PRESPTOBTASETR | *2 | H'0000_0000 | H'210 | 8/16/32 |
| Peripheral Response Timeout LP Set Register | PRESPTOLPSETR | *2 | H'0000_0000 | H'214 | 8/16/32 |
| Peripheral Response Timeout HS Set Register | PRESPTOHSSETR | *2 | H'0000_0000 | H'218 | 8/16/32 |
| Acknowledge and Error Report Packet Parameter Latest Info Register | AKEPLATIR | *2 | H'0000_0000 | H'220 | 8/16/32 |
| Acknowledge and Error Report Packet Parameter Accumulate Status Register | AKEPACMSR | *2 | H'0000_0000 | H'224 | 8/16/32 |
| Acknowledge and Error Report Packet Parameter Status Clear Register | AKEPSCR | *2 | H'0000_0000 | H'228 | 8/16/32 |
| Rx Result Saved Status Register | RXRSSR | *2 | H'0000_0000 | H'230 | 8/16/32 |
| Rx Result Saved Status Clear Register | RXRSSCR | *2 | H'0000_0000 | H'234 | 8/16/32 |
| Rx Result Info Overwrite Status Register | RXRINFOOWSR | *2 | H'0000_0000 | H'238 | 8/16/32 |
| Rx Result Info Overwrite Status Clear Register | RXRINFOOWSCR | *2 | H'0000_0000 | H'23C | 8/16/32 |
| Rx Result Save Slot 0 Register | RXRSS0R | *2 | H'0000_0000 | H'240 | 8/16/32 |
| Rx Result Save Slot 1 Register | RXRSS1R | *2 | H'0000_0000 | H'244 | 8/16/32 |
| Rx Result Save Slot 2 Register | RXRSS2R | *2 | H'0000_0000 | H'248 | 8/16/32 |
| Rx Result Save Slot 3 Register | RXRSS3R | *2 | H'0000_0000 | H'24C | 8/16/32 |
| Rx Packet Payload Data 0 Register | RXPPD0R | *2 | H'0000_0000 | H'2C0 | 8/16/32 |

Table 34.3 Resister list (2/3)

| Register Name | Abbreviation | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|
| Rx Packet Payload Data 1 Register | RXPPD1R | *2 | H'0000_0000 | H'2C4 | 8/16/32 |
| Rx Packet Payload Data 2 Register | RXPPD2R | *2 | H'0000_0000 | H'2C8 | 8/16/32 |
| Rx Packet Payload Data 3 Register | RXPPD3R | *2 | H'0000_0000 | H'2CC | 8/16/32 |
| HSTX Timeout Set Register | HSTXTOSETR | *2 | H'0000_0000 | H'2E0 | 8/16/32 |
| LRX-H Timeout Set Register | LRXHTOSETR | *2 | H'0000_0000 | H'2E4 | 8/16/32 |
| TA Timeout Set Register | TATOSETR | *2 | H'0000_0000 | H'2E8 | 8/16/32 |
| Fatal Error Status Register | FERRSR | *2 | H'0000_0000 | H'300 | 8/16/32 |
| Fatal Error Status Clear Register | FERRSCR | *2 | H'0000_0000 | H'304 | 8/16/32 |
| Fatal Error Interrupt Enable Register | FERRIER | *2 | H'0000_0000 | H'308 | 8/16/32 |
| Clock Lane Stop Time Set Register | CLSTPTSETR | *2 | H'0000_0000 | H'314 | 8/16/32 |
| LP Transition Time Set Register | LPTRNSTSETR | *2 | H'0000_0000 | H'318 | 8/16/32 |
| Physical Lane Status Register | PLSR | *2 | H'0000_00F1 | H'320 | 8/16/32 |
| Physical Lane Status Clear Register | PLSCR | *2 | H'0000_0000 | H'324 | 8/16/32 |
| Physical Lane Interrupt Enable Register | PLIER | *2 | H'0000_0000 | H'328 | 8/16/32 |
| Video-Input Channel 1 Set 0 Register | VICH1SET0R | *2 | H'0000_0000 | H'400 | 8/16/32 |
| Video-Input Channel 1 Set 1 Register | VICH1SET1R | *2 | H'1063_0000 | H'404 | 8/16/32 |
| Video-Input Channel 1 Status Register | VICH1SR | *2 | H'0000_0000 | H'410 | 8/16/32 |
| Video-Input Channel 1 Status Clear Register | VICH1SCR | *2 | H'0000_0000 | H'414 | 8/16/32 |
| Video-Input Channel 1 Interrupt Enable Register | VICH1IER | *2 | H'0000_0000 | H'418 | 8/16/32 |
| Video-Input Channel 1 Pixel Packet Set Register | VICH1PPSETR | *2 | H'000E_0000 | H'420 | 8/16/32 |
| Video-Input Channel 1 Vertical Size Set Register | VICH1VSSETR | *2 | H'0000_0000 | H'428 | 8/16/32 |
| Video-Input Channel 1 Vertical Porch Set Register | VICH1VPSETR | *2 | H'0000_0000 | H'42C | 8/16/32 |
| Video-Input Channel 1 Horizontal Size Set Register | VICH1HSSETR | *2 | H'0000_0000 | H'430 | 8/16/32 |
| Video-Input Channel 1 Horizontal Porch Set Register | VICH1HPSETR | *2 | H'0000_0000 | H'434 | 8/16/32 |
| Sequence Channel 0 Set 0 Register | SQCH0SET0R | *2 | H'0080_0000 | H'5C0 | 8/16/32 |
| Sequence Channel 0 Set 1 Register | SQCH0SET1R | *2 | H'0800_0000 | H'5C4 | 8/16/32 |
| Sequence Channel 0 Status Register | SQCH0SR | *2 | H'0000_0000 | H'5D0 | 8/16/32 |
| Sequence Channel 0 Status Clear Register | SQCH0SCR | *2 | H'0000_0000 | H'5D4 | 8/16/32 |
| Sequence Channel 0 Interrupt Enable Register | SQCH0IER | *2 | H'0000_0000 | H'5D8 | 8/16/32 |
| Sequence Channel 1 Set 0 Register | SQCH1SET0R | *2 | H'0080_0000 | H'600 | 8/16/32 |
| Sequence Channel 1 Set 1 Register | SQCH1SET1R | *2 | H'0830_0000 | H'604 | 8/16/32 |
| Sequence Channel 1 Status Register | SQCH1SR | *2 | H'0000_0000 | H'610 | 8/16/32 |
| Sequence Channel 1 Status Clear Register | SQCH1SCR | *2 | H'0000_0000 | H'614 | 8/16/32 |
| Sequence Channel 1 Interrupt Enable Register | SQCH1IER | *2 | H'0000_0000 | H'618 | 8/16/32 |
| Sequence Channel 0 Descriptor 00-A Register | SQCH0DSC00AR | *2 | *1 | H'780 | 32 |
| Sequence Channel 0 Descriptor 00-B Register | SQCH0DSC00BR | *2 | *1 | H'784 | 32 |
| Sequence Channel 0 Descriptor 00-C Register | SQCH0DSC00CR | *2 | *1 | H'788 | 32 |

Table 34.3 Resister list (3/3)

| Register Name | Abbreviation | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|
| Sequence Channel 0 Descriptor 00-D Register | SQCH0DSC00DR | *2 | *1 | H'78C | 32 |
| Sequence Channel 0 Descriptor 01-A Register | SQCH0DSC01AR | *2 | *1 | H'790 | 32 |
| Sequence Channel 0 Descriptor 01-B Register | SQCH0DSC01BR | *2 | *1 | H'794 | 32 |
| Sequence Channel 0 Descriptor 01-C Register | SQCH0DSC01CR | *2 | *1 | H'798 | 32 |
| Sequence Channel 0 Descriptor 01-D Register | SQCH0DSC01DR | *2 | *1 | H'79C | 32 |
| … | … | | … | | 32 |
| Sequence Channel 0 Descriptor 07-A Register | SQCH0DSC07AR | *2 | *1 | H'7F0 | 32 |
| Sequence Channel 0 Descriptor 07-B Register | SQCH0DSC07BR | *2 | *1 | H'7F4 | 32 |
| Sequence Channel 0 Descriptor 07-C Register | SQCH0DSC07CR | *2 | *1 | H'7F8 | 32 |
| Sequence Channel 0 Descriptor 07-D Register | SQCH0DSC07DR | *2 | *1 | H'7FC | 32 |
| Sequence Channel 1 Descriptor 00-A Register | SQCH1DSC00AR | *2 | *1 | H'800 | 32 |
| Sequence Channel 1 Descriptor 00-B Register | SQCH1DSC00BR | *2 | *1 | H'804 | 32 |
| Sequence Channel 1 Descriptor 00-C Register | SQCH1DSC00CR | *2 | *1 | H'808 | 32 |
| Sequence Channel 1 Descriptor 00-D Register | SQCH1DSC00DR | *2 | *1 | H'80C | 32 |
| … | … | | … | | 32 |
| Sequence Channel 1 Descriptor 07-A Register | SQCH1DSC07AR | *2 | *1 | H'870 | 32 |
| Sequence Channel 1 Descriptor 07-B Register | SQCH1DSC07BR | *2 | *1 | H'874 | 32 |
| Sequence Channel 1 Descriptor 07-C Register | SQCH1DSC07CR | *2 | *1 | H'878 | 32 |
| Sequence Channel 1 Descriptor 07-D Register | SQCH1DSC07DR | *2 | *1 | H'87C | 32 |

**Note:** Other addresses are Reserved. Do not Access to them.

Note 1. Initial Values are determined by initial value of Descriptor RAM.

Note 2. Refer to the respective register description and **Table 34.5**.

### 34.2.2 Registers in D-PHY

This section describes about registers in D-PHY. All registers can access by 8/16/32 bit.

Base Address: H'0_1085_0000 (Cortex-A55 Address Space)

Table 34.4 Resister list

| Register Name | Abbreviation | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|
| D-PHY Control Resister 0 | DSIDPHYCTRL0 | RW | H'00010104 | H'00 | 8/16/32 |
| D-PHY Timing Resister 0 | DSIDPHYTIM0 | RW | H'01002710 | H'04 | 8/16/32 |
| D-PHY Timing Resister 1 | DSIDPHYTIM1 | RW | H'04030909 | H'08 | 8/16/32 |
| D-PHY Timing Resister 2 | DSIDPHYTIM2 | RW | H'05100119 | H'0C | 8/16/32 |
| D-PHY Timing Resister 3 | DSIDPHYTIM3 | RW | H'0409060B | H'10 | 8/16/32 |

## 34.3 Register Descriptions

### 34.3.1 Resisters in LINK

Legend for Register Description.

Table 34.5 Resister R/W

| R/W | Write Access | Read Access |
|---|---|---|
| R | Ignored | Valid |
| RW | Valid | Valid |
| R0W | Valid | The read value is always 0. |

### 34.3.1.1 Interrupt Status Register (ISR)

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | PPI | — | — | — | FERR |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | RCV | — | — | — | VIN1 | — | — | — | SQ1 | — | — | — | SQ0 |
| 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 29 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 28 | — | 0 | R | Reserved<br>When read, the initial value is read. The written value will be ignored. |
| 27 to 21 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 20 | PPI | 0 | R | PPI Interrupt<br>When this value is “1”, an interrupt of DSI D-PHY PPI related exists.<br>Refer to the description of PLSR register for more information. |
| 19 to 17 | — | All 0 | R | Reserved |
| 16 | FERR | 0 | R | FERR Interrupt<br>When this value is “1”, an interrupt of DSI Fatal Error exists.<br>Refer to the description of FERRSR register for more information. |
| 15 to 13 | — | All 0 | R | Reserved |
| 12 | RCV | 0 | R | RCV Interrupt<br>When this value is “1”, an interrupt of DSI Packet Receive exists.<br>Refer to the description of RXSR register for more information. |
| 11 to 9 | — | All 0 | R | Reserved |
| 8 | VIN1 | 0 | R | VIN1 Interrupt<br>When this value is “1”, an interrupt of Video Input-Operation channel 1 exists.<br>Refer to the description of VICH1SR register for more information. |
| 7 to 5 | — | All 0 | R | Reserved |
| 4 | SQ1 | 0 | R | SQ1 Interrupt<br>When this value is “1”, an interrupt of Sequence Operation channel 1 exists.<br>Refer to the description of SQCH1SR register for more information. |
| 3 to 1 | — | All 0 | R | Reserved |
| 0 | SQ0 | 0 | R | SQ1 Interrupt<br>When this value is “1”, an interrupt of Sequence Operation channel 0 exists.<br>Refer to the description of SQCH0SR register for more information. |

### 34.3.1.2 Link Status Register (LINKSR)

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | LPBUSY | HSBUSY | — | — | — | VICHRUN1 | — | — | — | SQCHRUN1 | — | — | — | SQCHRUN0 |
| 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 14 | — | All 0 | R | Reserved<br>When read, the initial value is read. The written value will be ignored. |
| 13 | LPBUSY | 0 | R | LP Operation Busy<br>When this value is “1”, it indicates an operation related to LP is running. |
| 12 | HSBUSY | 0 | R | HS Operation Busy<br>When this value is “1”, it indicates an operation related to HS is running. |
| 11 to 9 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 8 | VICHRUN1 | 0 | R | Video-Input Channel 1 Running<br>This bit is copy of VICH1SR.RUNNING. |
| 7 to 5 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 4 | SQCHRUN1 | 0 | R | Sequence Channel 1 Running<br>This bit is copy of SQCH1SR.RUNNING. |
| 3 to 1 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 0 | SQCHRUN0 | 0 | R | Sequence Channel 0 Running<br>This bit is copy of SQCH0SR.RUNNING. |

### 34.3.1.3 Tx Set Register (TXSETR)

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | NUMLANECAP | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | DLEN | CLEN | — | — | — | — | — | — | NUMLANEUSE | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 |
| R/W | R | R | R | R | R | R | RW | RW | R | R | R | R | R | R | RW | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 18 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 17 to 16 | NUMLANECAP | H'3 | R | Number of Lanes Capability<br>H'3 (fixed value): Maximum Lane count is 4 (Lane0, Lane1, Lane2 and Lane3 can be used) |
| 15 to 10 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 9 | DLEN | 0 | RW | Data Lane Enable<br>When this bit is "1", Data Lane is enabled by PHY interface.<br>Be enabled lanes are determined by NUMLANEUSE. |
| 8 | CLEN | 0 | RW | Clock Lane Enable<br>When this bit is "1", Clock Lane is enabled by PHY interface. |
| 7 to 2 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 1 to 0 | NUMLANEUSE | H'3 | RW | Number of Lanes for Use<br>Set Lane count for use. Selected Data Lanes are Enabled if DLEN is "1".<br>H'0: 1Lane (Use of Lane0)<br>H'1: 2Lane (Use of Lane0 and Lane1)<br>H'2: 3Lane (Use of Lane0, Lane1 and Lane2)<br>H'3: 4Lane (Use of Lane0, Lane1, Lane2 and Lane3)<br>Setting value over NUMLANECAP is prohibited. |

### 34.3.1.4 HS Clock Set Register (HSCLKSETR)

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | HSCLK MODE | HSCLK RUN |
| 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 | RW | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 2 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 1 | HSCLKMODE | 0 | RW | HS Clock running Mode<br>0b: Request for clock lane HS transmission when HS Transmission scheduled. (non-continuous clock mode)<br>1b: Keep clock lane to HS transmission. (continuous clock mode)<br>If HSCLKRUN=0, this field has no meaning.<br>Prohibited to change this bit during HSCLKRUN=1. |
| 0 | HSCLKRUN | 0 | RW | Start HS Clock running on Clock Lane<br>0b: No HS transmission (keep LP)<br>1b: Allow HS transmission according to HSCLKMODE<br>Confirm stable TxWordClkHS(hsclk) before setting this field to 1.<br>Setting 1 to this field during TXSETR.CLEN=0 is prohibited. |

### 34.3.1.5 ULPS Set Register (ULPSSETR)

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | ULPSWKUP | | | | | | | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R | RW | RW | RW | RW | RW | RW | RW | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 8 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 7 to 0 | ULPSWKUP | H'A0 | RW | Twakeup period of ULPS<br>This period is specified by count value of MIPI_DSI_LPCLK *128.<br>This period should be more than 1ms.<br>For example;<br>• In case MIPI_DSI_LPCLK is 4/3MHz, MIPI_DSI_LPCLK *128=96us, Setting ULPSWKUP = H'0B, Twakeup period = 1.06ms (11*96us)<br>• In case MIPI_DSI_LPCLK is 20MHz, MIPI_DSI_LPCLK *128=6.4us, Setting ULPSWKUP = H'A0, Twakeup period = 1.02ms (160*6.4us)<br>Setting 1 to this field during TXSETR.CLEN=0 is prohibited. |

### 34.3.1.6 ULPS Control Register (ULPSCR)

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | DLULPSEXT | DLULPSENT | — | — | CLULPSEXT | CLULPSENT | — | — | — | — | — | — | — | — |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R0W | R0W | R | R | R0W | R0W | 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, 30 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 29 | DLULPSEXT | 0 | R0W | DL ULPS Exit<br>When write “1”, request to all enabled data lane exit from Ultra-low Power State (ULPS).<br>Write “1” when enabled data lanes are not in ULPS, is prohibited.<br>Writing “0” has no effect. |
| 28 | DLULPSENT | 0 | R0W | DL ULPS Enter<br>When write “1”, request to enabled data lane transition to Ultra-low Power State (ULPS).<br>Write “1” when enabled data lanes are in ULPS, is prohibited.<br>Writing “0” has no effect. |
| 27, 26 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 25 | CLULPSEXT | 0 | R0W | CL ULPS Exit<br>When write “1”, request to clock lane exit from Ultra-low Power State (ULPS).<br>Write “1” when clock lane is not in ULPS, is prohibited.<br>Writing “0” has no effect. |
| 24 | CLULPSENT | 0 | R0W | CL ULPS Enter<br>When write “1”, request to clock lane transition to Ultra-low Power State (ULPS).<br>Before setting this field, HSCLKSETR.HSCLKRUN must be set to 0.<br>Write “1” when clock lane is in ULPS, is prohibited.<br>Writing “0” has no effect. |
| 23 to 0 | — | 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |

### 34.3.1.7 Reset Control Register (RSTCR)

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | FCETXSTP |
| 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 | RW | R | R | R | R | R | R | R | RW |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | SWRST |
| 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 | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 25 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 24 | — | 0 | RW | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 23 to 17 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 16 | FCETXSTP | 0 | RW | Force Tx Stop Mode<br>When this value is “1”, force to change to Transmit mode and StopState by ForceTxStopmode on all PPI Data Lane.<br>Set “1” when SWRST=0 is prohibited. |
| 15 to 1 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 0 | SWRST | 0 | RW | Request of Software Reset<br>Set to “1” to request for Software Reset. After the request of Software Reset, wait to confirm the completion of the Reset procedure by RSTSR.SWRSTx=1. And set to “0” after that. |

### 34.3.1.8 Reset Status Register (RSTSR)

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | DL0DIR | — | — | — | DLSTPST | | | | — | — | — | SWRSTV1 | SWRSTIB | SWRSTAPB | SWRSTLP | SWRSTHS |
| 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 16 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 15 | DL0DIR | 0 | R | Direction (Tx or Rx) of PPI Data Lane0<br>0: PHY is in Tx Mode<br>1: PHY is in Rx Mode<br>This field is copy of PLSR.DL0DIR |
| 14 to 12 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 11 to 8 | DLSTPST | All 0 | R | Status of Stopstate on Data Lane<br>bit 3: Stopstate of Data Lane3<br>bit 2: Stopstate of Data Lane2<br>bit 1: Stopstate of Data Lane1<br>bit 0: Stopstate of Data Lane0<br>This field is copy of PLSR.DLSTPST |
| 7 to 5 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 4 | SWRSTV1 | 0 | R | Status of Software Reset by RSTCR.SWRST at MIPI_DSI_VCLK<br>0: Not on Reset procedure at MIPI_DSI_VCLK<br>1: Now on Reset procedure at MIPI_DSI_VCLK |
| 3 | SWRSTIB | 0 | R | Status of Software Reset by RSTCR.SWRST at MIPI_DSI_ACLK<br>0: Not on Reset procedure at MIPI_DSI_ACLK<br>1: Now on Reset procedure at MIPI_DSI_ACLK |
| 2 | SWRSTAPB | 0 | R | Status of Software Reset by RSTCR.SWRST at MIPI_DSI_PCLK<br>0: Not on Reset procedure at MIPI_DSI_PCLK<br>1: Now on Reset procedure at MIPI_DSI_PCLK |
| 1 | SWRSTLP | 0 | R | Status of Software Reset by RSTCR.SWRST at MIPI_DSI_LPCLK<br>0: Not on Reset procedure at MIPI_DSI_LPCLK<br>1: Now on Reset procedure at MIPI_DSI_LPCLK |
| 0 | SWRSTHS | 0 | R | Status of Software Reset by RSTCR.SWRST at hsclk<br>0: Not on Reset procedure at hsclk<br>1: Now on Reset procedure at hsclk |

### 34.3.1.9 DSI Set Register (DSISETR)

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | EOTPEN | TEIDIR | SCREN | — | — | — | — | — | CRCEN | | | | — | — | — | ECCEN |
| Initial Value | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| R/W | RW | RW | RW | R | R | R | R | R | RW | RW | RW | RW | R | R | R | RW |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MRPSZ | | | | | | | | | | | | | | | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| R/W | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 | EOTPEN | 0 | RW | HS Transfer EoTp Enable<br>0: Transfer EoTp disable<br>1: Transfer EoTp enable<br>EoTp is always disabled in LP Transfer. |
| 30 | TEIDIR | 0 | RW | Tearing Effect Input Direction<br>External Tearing Effect Setting<br>0: rise edge detect<br>1: fall edge detect |
| 29 | SCREN | 0 | RW | Scramble Enable<br>0: Scrambling data is disabled<br>1: Scrambling data is enabled<br>This bit should not set to “1”, if peer-device does not support data scrambling function. |
| 28 to 24 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 23 to 20 | CRCEN | H’F | RW | Force Tx Stop Mode<br>When this value is “1”, force to change to Transmit mode and StopState by ForceTxStopmode on all PPI Data Lane.<br>Set “1” when SWRST=0 is prohibited. |
| 19 to 17 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 16 | ECCEN | 0 | RW | Enable Received ECC check<br>0: Received ECC check is disabled<br>1: Received ECC check is enabled |
| 15 to 0 | MRPSZ | H’0001 | RW | Maximum Return Packet Size<br>If returned long packet’s WC is over this value, the DSI-Tx Module asserts RXSR.MAXRPSZERR error. In that case data will not save to memory space.<br>Setting 0 for this field is prohibited.<br>Set this field equal or more than peripheral setting. |

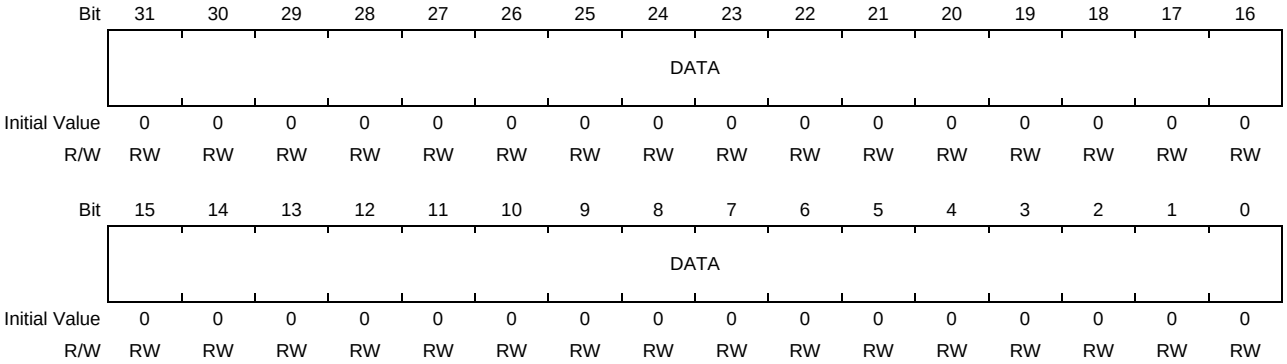
### 34.3.1.10 Receive Buffer Size Register (RXBUFSZR)

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | RXBUFSZ | | | | — | — | — | — |
| 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 |

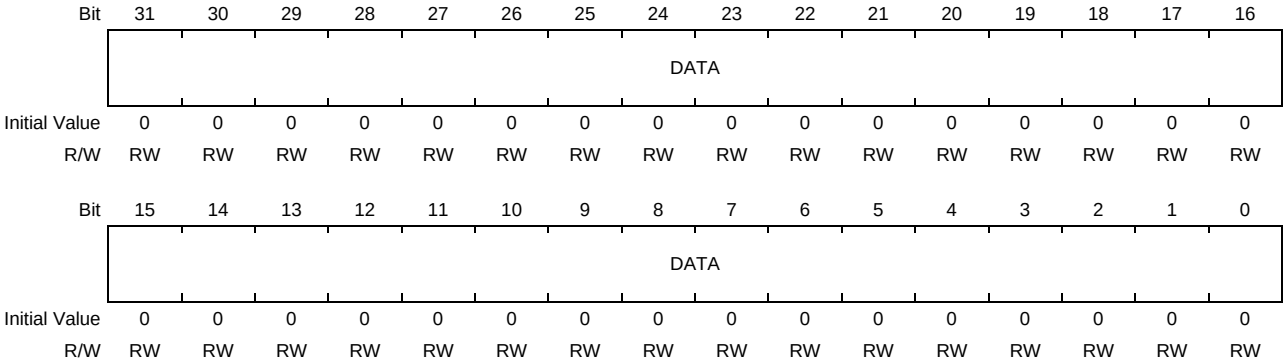
| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 24 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 23 to 20 | RXBUFSZ | H'0 | R | Receive Buffer Size<br>H'0: 128B |
| 19 to 0 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |

### 34.3.1.11 Tx Packet Payload Data 0 Register (TXPPD0R)



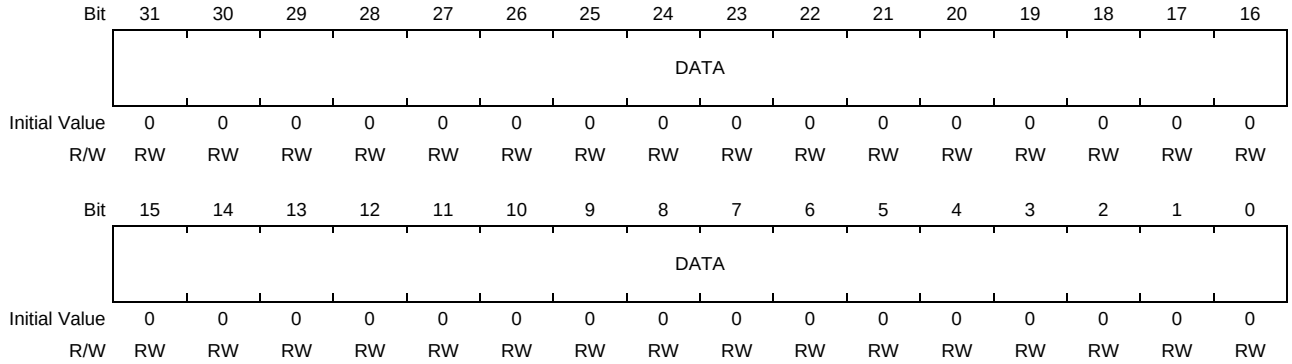

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 0 | DATA | All 0 | RW | Tx Packet Payload Data 0<br>When sequence operation Long packet is set (SQCHxDSCyAR.FMT=1) and Payload Data Access use Packet Payload Data register (SQCHxDSCyBR.DTSEL =H'0) , the values in this register is used for Long Packet Payload.<br>[7:0]: Data 0<br>[15:8]: Data 1<br>[23:16]: Data 2<br>[31:24]: Data 3 |

### 34.3.1.12 Tx Packet Payload Data 1 Register (TXPPD1R)



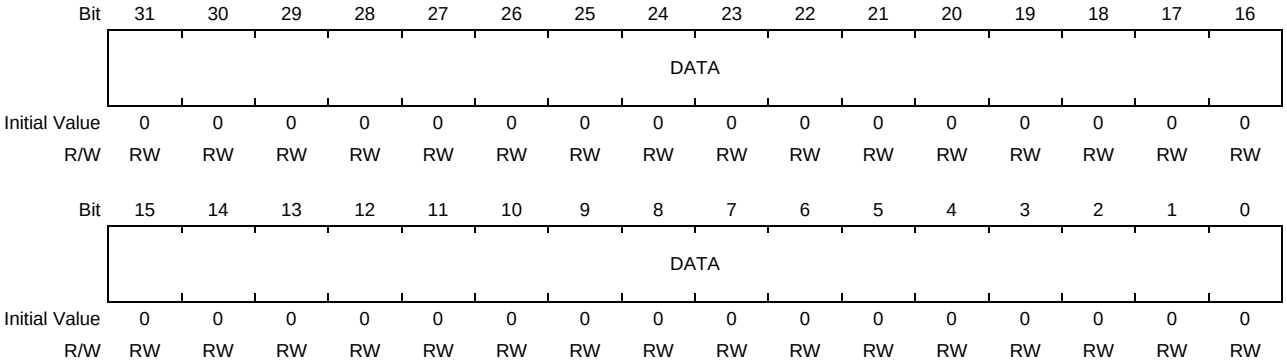

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 0 | DATA | All 0 | RW | Tx Packet Payload Data 1<br>When sequence operation Long packet is set (SQCHxDSCyAR.FMT=1) and Payload Data Access use Packet Payload Data register (SQCHxDSCyBR.DTSEL =H'0) , the values in this register is used for Long Packet Payload.<br>[7:0]: Data 4<br>[15:8]: Data 5<br>[23:16]: Data 6<br>[31:24]: Data 7 |

### 34.3.1.13 Tx Packet Payload Data 2 Register (TXPPD2R)




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 0 | DATA | All 0 | RW | Tx Packet Payload Data 2<br>When sequence operation Long packet is set (SQCHxDSCyAR.FMT=1) and Payload Data Access use Packet Payload Data register (SQCHxDSCyBR.DTSEL =H'0) , the values in this register is used for Long Packet Payload.<br>[7:0]: Data 8<br>[15:8]: Data 9<br>[23:16]: Data 10<br>[31:24]: Data 11 |

### 34.3.1.14 Tx Packet Payload Data 3 Register (TXPPD3R)




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 0 | DATA | All 0 | RW | Tx Packet Payload Data 3<br>When sequence operation Long packet is set (SQCHxDSCyAR.FMT=1) and Payload Data Access use Packet Payload Data register (SQCHxDSCyBR.DTSEL =H'0) , the values in this register is used for Long Packet Payload.<br>[7:0]: Data 12<br>[15:8]: Data 13<br>[23:16]: Data 14<br>[31:24]: Data 15 |

### 34.3.1.15 Rx Status Register (RXSR)

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | RXAKE | — | ECCERR1B | — | MAXRPSZERR | NORETERR | PRESPTOERR | RXOVFERR | IBERR | CRCERR | WCERR | — | UEXPKTERR | ECCERR | MLFERR |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | EXTTEDET | RXACK | RXTE | RXRTRG | RXUK5TRG | RXEOTP | — | RXRESP | — | — | — | — | — | TATO | LRXHTO | BTAREQEND |
| 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 | — | 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 30 | RXAKE | 0 | R | Rx Acknowledge and Error Report<br>When this value is “1”, the Acknowledge and Error Report Packet is received.<br>Write “1” to RXSCR.RXAKE register to clear this field. |
| 29 | — | 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 28 | ECCERR1B | 0 | R | ECC Error (1bit)<br>When this value is “1”, the Rx ECC Error (1bit) is detected.<br>Write “1” to RXSCR.ECCERR1B register to clear this field. |
| 27 | — | 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 26 | MAXRPSZERR | 0 | R | Maximum Return Packet Size Error<br>When this value is “1”, the WC value of Rx Long Packet is larger than DSISETR.MRPSZ.<br>Write “1” to RXSCR.MAXRPSZERR register to clear this field. |
| 25 | NORETERR | 0 | R | Nothing Return Error<br>When this value is “1”, no trigger or packet is returned in BTA period.<br>Write “1” to RXSCR.NORETERR register to clear this field. |
| 24 | PRESPTOERR | 0 | R | Peripheral Response Timeout Error<br>When this value is “1”, the Rx Peripheral Response Timeout is detected after changing to Rx mode along with Bus Turn-Around and within the time specified by PRESPTOBTASETR.PRESPTOBTA, PRESPTOLPSETR.PRESPTOLPR, PRESPTOLPSETR.PRESPTOLPW, PRESPTOLPSETR.PRESPTOHSR or PRESPTOLPSETR.PRESPTOHSW. The value is selected from kind of BTA.<br>Write “1” to RXSCR.PRESPTOERR register to clear this field. |
| 23 | RXOVFERR | 0 | R | Receive Buffer Overflow Error<br>When this value is “1”, the Buffer Overflow Error is detected at Rx Long Packet.<br>Write “1” to RXSCR.RXOVFERR register to clear this field. |
| 22 | IBERR | 0 | R | Internal Bus Error<br>When this value is “1”, the Internal Bus Write had failed response.<br>Write “1” to RXSCR.IBERR register to clear this field. |
| 21 | CRCERR | 0 | R | CRC Error<br>When this value is “1”, the Rx CRC Error is detected.<br>Write “1” to RXSCR.CRCERR register to clear this field. |
| 20 | WCERR | 0 | R | Word Count Error<br>When this value is “1”, the Rx Packet’s actual WC length is shorter than packet header’s WC.<br>Write “1” to RXSCR.WCERR register to clear this field. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 19 | — | 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 18 | UEXPKTERR | 0 | R | Unexpected Packet Error<br>When this value is “1”, the unexpected Packet is received.<br>“Unexpected Packet” means unexpected DT for receive or unexpected response for receive.<br>Write “1” to RXSCR.UEXPKTERR register to clear this field. |
| 17 | ECCERR | 0 | R | ECC Error (over 1bit)<br>When this value is “1”, the Rx ECC Error (over 1bit) is detected.<br>Write “1” to RXSCR.ECCERR register to clear this field. |
| 16 | MLFERR | 0 | R | Malform Error<br>When this value is “1”, the Rx Packet under 4 Byte length is received.<br>Write “1” to RXSCR.MLFERR register to clear this field. |
| 15 | EXTTEDET | 0 | R | External Tearing Effect Detect<br>When this value is “1”, External Tearing Effect is detected.<br>Write “1” to RXSCR.EXTTEDET register to clear this field. |
| 14 | RXACK | 0 | R | Rx ACK Trigger<br>When this value is “1”, the ACK trigger is received.<br>Write “1” to RXSCR.RXACK register to clear this field. |
| 13 | RXTE | 0 | R | Rx Tearing Effect Trigger<br>When this value is “1”, Tearing Effect Trigger is received.<br>Write “1” to RXSCR.RXTE register to clear this field. |
| 12 | RXRTRG | 0 | R | Rx Reset Trigger<br>When this value is “1”, Reset Trigger is received. Normally, the DSI-Tx Module does not receive Reset Trigger.<br>Write “1” to RXSCR.RXRTRG register to clear this field. |
| 11 | RXUK5TRG | 0 | R | Rx Unknown-5 Trigger<br>When this value is “1”, Unknown-5 Trigger is received. Normally, the DSI-Tx Module does not receive Unknown-5 Trigger.<br>Write “1” to RXSCR.RXUK5TRG register to clear this field. |
| 10 | RXEOTP | 0 | R | Rx EoTp<br>When this value is “1”, the EoTp is received.<br>Write “1” to RXSCR.RXEOTP register to clear this field. |
| 9 | — | 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 8 | RXRESP | 0 | R | Rx Response Packet<br>When this value is “1”, the Response Packet is received.<br>Write “1” to RXSCR.RXRESP register to clear this field. |
| 7 to 3 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 2 | TATO | 0 | R | Turnaround Acknowledge Timeout<br>When this value is “1”, the Turnaround Acknowledge Timeout (TA_TO) is detected.<br>Write “1” to RXSCR.TATO register to clear this field. |
| 1 | LRXHTO | 0 | R | LP-RX Host Processor Timeout<br>When this value is “1”, the LP-RX Host Processor Timeout (LRX-H_TO) is detected.<br>Write “1” to RXSCR.LRXHTO register to clear this field. |
| 0 | BTAREQEND | 0 | R | BTA Request End<br>When this value is “1”, the completion of Transmission request includes Rx is detected.<br>This bit is set if all BTA requests are finished including error.<br>Status of BTA request are indicated by RXSR[31:1].<br>Write “1” to RXSCR.BTAREQEND register to clear this field. |

### 34.3.1.16 Rx Status Clear Register (RXSCR)

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | RXAKE | — | ECCERR1B | — | MAXRPSZERR | NORETERR | PRESPTOERR | RXOVFERR | IBERR | CRCERR | WCERR | — | UEXPKTERR | ECCERR | MLFERR |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R0W | R | R0W | R | R0W | R0W | R0W | R0W | R0W | R0W | R0W | R | R0W | R0W | R0W |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | EXTTEDET | RXACK | RXTE | RXRTRG | RXUK5TRG | RXEOTP | — | RXRESP | — | — | — | — | — | TATO | LRXHTO | BTAREQEND |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R0W | R0W | R0W | R0W | R0W | R0W | R | R0W | R | R | R | R | R | R0W | R0W | R0W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 | — | 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 30 | RXAKE | 0 | R0W | RXSR.RXAKE Clear<br>Set to “1” to clear the RXSR.RXAKE<br>Writing “0” has no effect. |
| 29 | — | 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 28 | ECCERR1B | 0 | R0W | RXSR.ECCERR1B Clear<br>Set to “1” to clear the RXSR.ECCERR1B<br>Writing “0” has no effect. |
| 27 | — | 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 26 | MAXRPSZERR | 0 | R0W | RXSR.MAXRPSZERR Clear<br>Set to “1” to clear the RXSR.MAXRPSZERR<br>Writing “0” has no effect. |
| 25 | NORETERR | 0 | R0W | RXSR.NORETERR Clear<br>Set to “1” to clear the RXSR.NORETERR<br>Writing “0” has no effect. |
| 24 | PRESPTOERR | 0 | R0W | RXSR.PRESPTOERR Clear<br>Set to “1” to clear the RXSR.PRESPTOERR<br>Writing “0” has no effect. |
| 23 | RXOVFERR | 0 | R0W | RXSR.RXOVFERR Clear<br>Set to “1” to clear the RXSR.RXOVFERR<br>Writing “0” has no effect. |
| 22 | IBERR | 0 | R0W | RXSR.IBERR Clear<br>Set to “1” to clear the RXSR.IBERR<br>Writing “0” has no effect. |
| 21 | CRCERR | 0 | R0W | RXSR.CRCERR Clear<br>Set to “1” to clear the RXSR.CRCERR<br>Writing “0” has no effect. |
| 20 | WCERR | 0 | R0W | RXSR.WCERR Clear<br>Set to “1” to clear the RXSR.WCERR<br>Writing “0” has no effect. |
| 19 | — | 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 18 | UEXPKTERR | 0 | R0W | RXSR.UEXPKTERR Clear<br>Set to “1” to clear the RXSR.UEXPKTERR<br>Writing “0” has no effect. |
| 17 | ECCERR | 0 | R0W | RXSR.ECCERR Clear<br>Set to “1” to clear the RXSR.ECCERR<br>Writing “0” has no effect. |
| 16 | MLFERR | 0 | R0W | RXSR.MLFERR Clear<br>Set to “1” to clear the RXSR.MLFERR<br>Writing “0” has no effect. |
| 15 | EXTTEDET | 0 | R0W | RXSR.EXTTTEDET Clear<br>Set to “1” to clear the RXSR.EXTTTEDET<br>Writing “0” has no effect. |
| 14 | RXACK | 0 | R0W | RXSR.RXACK Clear<br>Set to “1” to clear the RXSR.RXACK<br>Writing “0” has no effect. |
| 13 | RXTE | 0 | R0W | RXSR.RXTE Clear<br>Set to “1” to clear the RXSR.RXTE<br>Writing “0” has no effect. |
| 12 | RXRTRG | 0 | R0W | RXSR.RXRTRG Clear<br>Set to “1” to clear the RXSR.RXRTRG<br>Writing “0” has no effect. |
| 11 | RXUK5TRG | 0 | R0W | RXSR.RXUK5TRG Clear<br>Set to “1” to clear the RXSR.RXUK5TRG<br>Writing “0” has no effect. |
| 10 | RXEOTP | 0 | R0W | RXSR.RXEOTP Clear<br>Set to “1” to clear the RXSR.RXEOTP<br>Writing “0” has no effect. |
| 9 | — | 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 8 | RXRESP | 0 | R0W | RXSR.RXRESP Clear<br>Set to “1” to clear the RXSR.RXRESP<br>Writing “0” has no effect. |
| 7 to 3 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 2 | TATO | 0 | R0W | RXSR.TATO Clear<br>Set to “1” to clear the RXSR.TATO.<br>Writing “0” has no effect. |
| 1 | LRXHTO | 0 | R0W | RXSR.LRXHTO Clear<br>Set to “1” to clear the RXSR.LRXHTO.<br>Writing “0” has no effect. |
| 0 | BTAREQEND | 0 | R0W | RXSR.BTAREQEND Clear<br>Set to “1” to clear the RXSR.BTAREQEND<br>Writing “0” has no effect. |

### 34.3.1.17 Rx Interrupt Enable Register (RXIER)

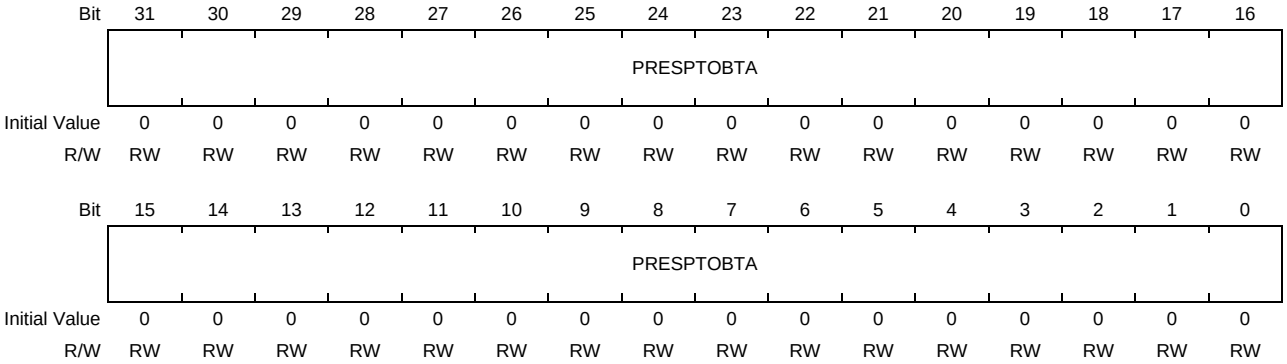
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | RXAKE | — | ECCERR1B | — | MAXRPSZERR | NORETERR | PRESPTOERR | RXOVFERR | IBERR | CRCERR | WCERR | — | UEXPKTERR | ECCERR | MLFERR |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | RW | R | RW | R | RW | RW | RW | RW | RW | RW | RW | R | RW | RW | RW |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | EXTTEDET | RXACK | RXTE | RXRTRG | RXUK5TRG | RXEOTP | — | RXRESP | — | — | — | — | — | TATO | LRXHTO | BTAREQEND |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | RW | RW | RW | RW | RW | RW | R | RW | R | R | R | R | R | RW | RW | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 | — | 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 30 | RXAKE | 0 | RW | Interrupt enable for RXSR.RXAKE<br>0: Disable interrupt of dsi_int_rcv when RXSR.RXAKE =1<br>1: Enable interrupt of dsi_int_rcv when RXSR.RXAKE =1 |
| 29 | — | 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 28 | ECCERR1B | 0 | RW | Interrupt enable for RXSR.ECCERR1B<br>0: Disable interrupt of dsi_int_rcv when RXSR.ECCERR1B=1<br>1: Enable interrupt of dsi_int_rcv when RXSR.ECCERR1B=1 |
| 27 | — | 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 26 | MAXRPSZERR | 0 | RW | Interrupt enable for RXSR.MAXRPSZERR<br>0: Disable interrupt of dsi_int_rcv when RXSR.MAXRPSZERR =1<br>1: Enable interrupt of dsi_int_rcv when RXSR.MAXRPSZERR =1 |
| 25 | NORETERR | 0 | RW | Interrupt enable for RXSR.NORETERR<br>0: Disable interrupt of dsi_int_rcv when RXSR.NORETERR =1<br>1: Enable interrupt of dsi_int_rcv when RXSR.NORETERR =1 |
| 24 | PRESPTOERR | 0 | RW | Interrupt enable for RXSR.PRESPTOERR<br>0: Disable interrupt of dsi_int_rcv when RXSR.PRESPTOERR=1<br>1: Enable interrupt of dsi_int_rcv when RXSR.PRESPTOERR=1 |
| 23 | RXOVFERR | 0 | RW | Interrupt enable for RXSR.RXOVFERR<br>0: Disable interrupt of dsi_int_rcv when RXSR.RXOVFERR =1<br>1: Enable interrupt of dsi_int_rcv when RXSR.RXOVFERR =1 |
| 22 | IBERR | 0 | RW | Interrupt enable for RXSR.IBERR<br>0: Disable interrupt of dsi_int_rcv when RXSR.IBERR =1<br>1: Enable interrupt of dsi_int_rcv when RXSR.IBERR =1 |
| 21 | CRCERR | 0 | RW | Interrupt enable for RXSR. CRCERR<br>0: Disable interrupt of dsi_int_rcv when RXSR.CRCERR =1<br>1: Enable interrupt of dsi_int_rcv when RXSR.CRCERR =1 |
| 20 | WCERR | 0 | RW | Interrupt enable for RXSR. WCERR<br>0: Disable interrupt of dsi_int_rcv when RXSR.WCERR =1<br>1: Enable interrupt of dsi_int_rcv when RXSR.WCERR =1 |
| 19 | — | 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |

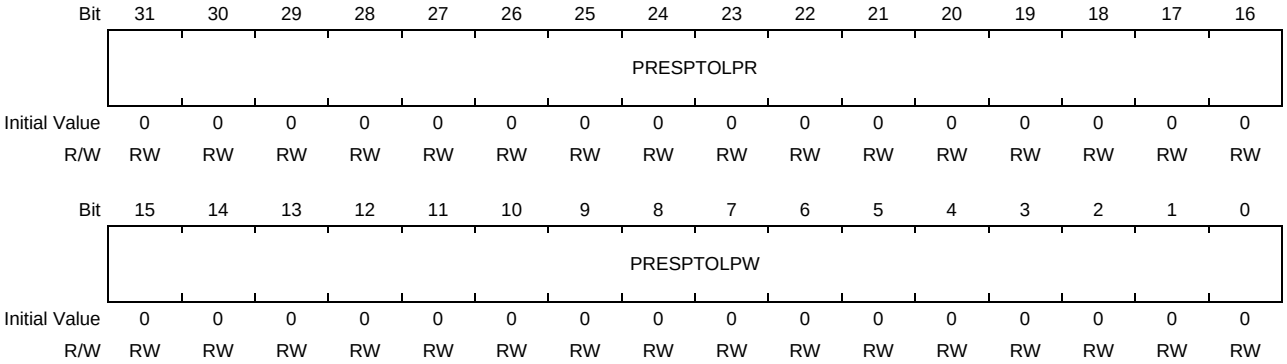
| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 18 | UEXPKTERR | 0 | RW | Interrupt enable for RXSR.UEXPKTERR<br>0: Disable interrupt of dsi_int_rcv when RXSR.UEXPKTERR =1<br>1: Enable interrupt of dsi_int_rcv when RXSR.UEXPKTERR =1 |
| 17 | ECCERR | 0 | RW | Interrupt enable for RXSR.ECCERR<br>0: Disable interrupt of dsi_int_rcv when RXSR.ECCERR=1<br>1: Enable interrupt of dsi_int_rcv when RXSR.ECCERR=1 |
| 16 | MLFERR | 0 | RW | Interrupt enable for RXSR.MLFERR<br>0: Disable interrupt of dsi_int_rcv when RXSR.MLFERR =1<br>1: Enable interrupt of dsi_int_rcv when RXSR.MLFERR =1 |
| 15 | EXTTEDET | 0 | RW | Interrupt enable for RXSR.EXTTEDET<br>0: Disable interrupt of dsi_int_rcv when RXSR.EXTTEDET =1<br>1: Enable interrupt of dsi_int_rcv when RXSR.EXTTEDET =1 |
| 14 | RXACK | 0 | RW | Interrupt enable for RXSR.RXACK<br>0: Disable interrupt of dsi_int_rcv when RXSR.RXACK =1<br>1: Enable interrupt of dsi_int_rcv when RXSR.RXACK =1 |
| 13 | RXTE | 0 | RW | Interrupt enable for RXSR.RXTE<br>0: Disable interrupt of dsi_int_rcv when RXSR.RXTE =1<br>1: Enable interrupt of dsi_int_rcv when RXSR.RXTE =1 |
| 12 | RXRTRG | 0 | RW | Interrupt enable for RXSR.RXRTRG<br>0: Disable interrupt of dsi_int_rcv when RXSR.RXRTRG =1<br>1: Enable interrupt of dsi_int_rcv when RXSR.RXRTRG =1 |
| 11 | RXUK5TRG | 0 | RW | Interrupt enable for RXSR.RXUK5TRG<br>0: Disable interrupt of dsi_int_rcv when RXSR.RXUK5TRG =1<br>1: Enable interrupt of dsi_int_rcv when RXSR.RXUK5TRG =1 |
| 10 | RXEOTP | 0 | RW | Interrupt enable for RXSR.RXEOTP<br>0: Disable interrupt of dsi_int_rcv when RXSR.RXEOTP =1<br>1: Enable interrupt of dsi_int_rcv when RXSR.RXEOTP =1 |
| 9 | — | 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 8 | RXRESP | 0 | RW | Interrupt enable for RXSR.RXRESP<br>0: Disable interrupt of dsi_int_rcv when RXSR.RXRESP =1<br>1: Enable interrupt of dsi_int_rcv when RXSR.RXRESP =1 |
| 7 to 3 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 2 | TATO | 0 | RW | Interrupt enable for RXSR.TATO<br>0: Disable interrupt of dsi_int_rcv when RXSR.TATO =1<br>1: Enable interrupt of dsi_int_rcv when RXSR.TATO =1 |
| 1 | LRXHTO | 0 | RW | Interrupt enable for RXSR.LRXHTO<br>0: Disable interrupt of dsi_int_rcv when RXSR.LRXHTO =1<br>1: Enable interrupt of dsi_int_rcv when RXSR.LRXHTO =1 |
| 0 | BTAREQEND | 0 | RW | Interrupt enable for RXSR.BTAREQEND<br>0: Disable interrupt of dsi_int_rcv when RXSR.BTAREQEND =1<br>1: Enable interrupt of dsi_int_rcv when RXSR.BTAREQEND =1 |

### 34.3.1.18 Peripheral Response Timeout BTA Set Register (PRESPTOBTASETR)



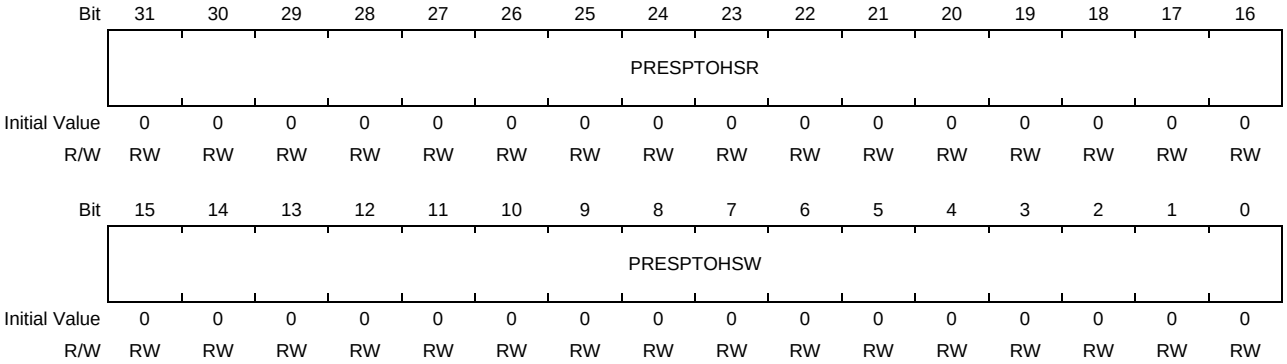

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 0 | PRESPTOBTA | All 0 | RW | Peripheral Response Timeout (Bus Turn Around) Count<br>Set Timeout value to measure the interval between Start of Rx mode and Start of Packet Receive along with Bus Turn-Around with SQCHxDSCyAR.BTA= H'3.<br>The limit Time for Timeout is calculated by formula of PRESPTOBTA * (period of MIPI_DSI_LPCLK).<br>When RXSR.PRESPTOERR is "1", the Timeout is detected.<br>Notice, the Timeout is not detected when PRESPTOBTA=0.<br>Since counting clock is asynchronous with Packet Receive, there is some jitter in the Timeout. |

### 34.3.1.19 Peripheral Response Timeout LP Set Register (PRESPTOLPSETR)




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 16 | PRESPTOLPR | All 0 | RW | Peripheral Response Timeout (LPDT READ Request) Count<br>Set Timeout value to measure the interval between Start of Rx mode and Start of Packet Receive along with Bus Turn-Around with SQCHxDSCyAR.BTA=H'2 and SQCHxDSCyAR.SPD=1b.<br>The limit Time for Timeout is calculated by formula of PRESPTOLPR * (period of MIPI_DSI_LPCLK).<br>When RXSR.PRESPTOERR is "1", the Timeout is detected.<br>Notice, the Timeout is not detected when PRESPTOLPR=0.<br>Since counting clock is asynchronous with Packet Receive, there is some jitter in the Timeout. |
| 15 to 0 | PRESPTOLPW | All 0 | RW | Peripheral Response Timeout (LPDT WRITE Request) Count<br>Set Timeout value to measure the interval between Start of Rx mode and Start of Packet Receive along with Bus Turn-Around with SQCHxDSCyAR.BTA=01b and SQCHxDSCyAR.SPD=1b.<br>The limit Time for Timeout is calculated by formula of PRESPTOLPW * (period of MIPI_DSI_LPCLK).<br>When RXSR.PRESPTOERR is "1", the Timeout is detected.<br>Notice, the Timeout is not detected when PRESPTOLPW=0.<br>Since counting clock is asynchronous with Packet Receive, there is some jitter in the Timeout. |

### 34.3.1.20 Peripheral Response Timeout HS Set Register (PRESPTOHSSETR)




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 16 | PRESPTOHSR | All 0 | RW | Peripheral Response Timeout (HS READ Request) Count<br>Set Timeout value to measure the interval between Start of Rx mode and Start of Packet Receive along with Bus Turn-Around with SQCHxDSCyAR.BTA=H'2 and SQCHxDSCyAR.SPD=0b.<br>The limit Time for Timeout is calculated by formula of PRESPTOHSR * (period of MIPI_DSI_LPCLK).<br>When RXSR.PRESPTOERR is "1", the Timeout is detected.<br>Notice, the Timeout is not detected when PRESPTOHSR=0.<br>Since counting clock is asynchronous with Packet Receive, there is some jitter in the Timeout. |
| 15 to 0 | PRESPTOHSW | All 0 | RW | Peripheral Response Timeout (HS WRITE Request) Count<br>Set Timeout value to measure the interval between Start of Rx mode and Start of Packet Receive along with Bus Turn-Around with SQCHxDSCyAR.BTA=H'1 and SQCHxDSCyAR.SPD=0b.<br>The limit Time for Timeout is calculated by formula of PRESPTOHSW * (period of MIPI_DSI_LPCLK).<br>When RXSR.PRESPTOERR is "1", the Timeout is detected.<br>Notice, the Timeout is not detected when PRESPTOHSW=0.<br>Since counting clock is asynchronous with Packet Receive, there is some jitter in the Timeout. |

### 34.3.1.21 Acknowledge and Error Report Packet Parameter Latest Info Register (AKEPLATIR)

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | VC3 | VC2 | VC1 | VC0 |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ERRRPTLAT | | | | | | | | | | | | | | | |
| 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 20 | − | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 19 | VC3 | 0 | R | Virtual Channel Identifier 3 Latest<br>This field shows Virtual channel identifier of received Acknowledge and Error Report Packet.<br>0: Virtual channel identifier is not 3<br>1: Virtual channel identifier is 3<br>This field shows latest information. Whenever an Acknowledge and Error Report Packet is received, this field is updated. |
| 18 | VC2 | 0 | R | Virtual Channel Identifier 2 Latest<br>This field shows Virtual channel identifier of received Acknowledge and Error Report Packet.<br>0: Virtual channel identifier is not 2<br>1: Virtual channel identifier is 2<br>This field shows latest information. Whenever an Acknowledge and Error Report Packet is received, this field is updated. |
| 17 | VC1 | 0 | R | Virtual Channel Identifier 1 Latest<br>This field shows Virtual channel identifier of received Acknowledge and Error Report Packet.<br>0: Virtual channel identifier is not 1<br>1: Virtual channel identifier is 1<br>This field shows latest information. Whenever an Acknowledge and Error Report Packet is received, this field is updated. |
| 16 | VC0 | 0 | R | Virtual Channel Identifier 0 Latest<br>This field shows Virtual channel identifier of received Acknowledge and Error Report Packet.<br>0: Virtual channel identifier is not 0<br>1: Virtual channel identifier is 0<br>This field shows latest information. Whenever an Acknowledge and Error Report Packet is received, this field is updated. |
| 15 to 0 | ERRRPTLAT | All 0 | R | Error Report Latest<br>This field shows Error Report of received Acknowledge and Error Report Packet.<br>This field shows latest information. Whenever an Acknowledge and Error Report Packet is received, this field is updated. |

### 34.3.1.22 Acknowledge and Error Report Packet Parameter Accumulate Status Register (AKEPACMSR)

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | VC3 | VC2 | VC1 | VC0 |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ERRRPTACM | | | | | | | | | | | | | | | |
| 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 20 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 19 | VC3 | 0 | R | Virtual Channel Identifier 3 Accumulate<br>This field is set to “1”, when an Acknowledge and Error Report Packet whose Virtual channel identifier = 3 is received.<br>This field shows accumulated information. This bit is cleared by AKEPSCR.VC3. |
| 18 | VC2 | 0 | R | Virtual Channel Identifier 2 Accumulate<br>This field is set to “1”, when an Acknowledge and Error Report Packet whose Virtual channel identifier = 2 is received.<br>This field shows accumulated information. This bit is cleared by AKEPSCR.VC2. |
| 17 | VC1 | 0 | R | Virtual Channel Identifier 1 Accumulate<br>This field is set to “1”, when an Acknowledge and Error Report Packet whose Virtual channel identifier = 1 is received.<br>This field shows accumulated information. This bit is cleared by AKEPSCR.VC1. |
| 16 | VC0 | 0 | R | Virtual Channel Identifier 0 Accumulate<br>This field is set to “1”, when an Acknowledge and Error Report Packet whose Virtual channel identifier = 0 is received.<br>This field shows accumulated information. This bit is cleared by AKEPSCR.VC0. |
| 15 to 0 | ERRRPTACM | All 0 | R | Error Report Accumulate<br>Each bit of this field is set to “1”, when an Acknowledge and Error Report Packet whose each Error Report bit = 1 is received.<br>This field shows accumulated information. Each bit is cleared by corresponding AKESCR.ERRRPTACM. |

### 34.3.1.23 Acknowledge and Error Report Packet Parameter Status Clear Register (AKEPSCR)

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | VC3 | VC2 | VC1 | VC0 |
| 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 | R0W | R0W | R0W | R0W |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | ERRRPTACM | | | | | | | | | | | | | | | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R0W | R0W | R0W | R0W | R0W | R0W | R0W | R0W | R0W | R0W | R0W | R0W | R0W | R0W | R0W | R0W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 20 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 19 | VC3 | 0 | R0W | AKEPACMSR.VC3 Clear<br>Set to “1” to clear AKEPACMSR.VC3.<br>Writing “0” has no effect. |
| 18 | VC2 | 0 | R0W | AKEPACMSR.VC2 Clear<br>Set to “1” to clear AKEPACMSR.VC2.<br>Writing “0” has no effect.This field shows accumulated information.<br>This bit is cleared by AKEPSCR.VC2. |
| 17 | VC1 | 0 | R0W | AKEPACMSR.VC1 Clear<br>Set to “1” to clear AKEPACMSR.VC1.<br>Writing “0” has no effect. |
| 16 | VC0 | 0 | R0W | AKEPACMSR.VC0 Clear<br>Set to “1” to clear AKEPACMSR.VC0.<br>Writing “0” has no effect. |
| 15 to 0 | ERRRPTACM | All 0 | R0W | Error Report Accumulate Clear<br>Set to “1” to clear each AKEPACMSR.ERRRPTACM.<br>Writing “0” has no effect. |

### 34.3.1.24 Rx Result Saved Status Register (RXRSSR)

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | SLT3VLD | SLT2VLD | SLT1VLD | SLT0VLD |
| 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 25 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 24 | — | 0 | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 23 to 4 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 3 | SLT3VLD | 0 | R | Slot 3 Valid<br>Response packet is received, or information is saved to RXRSS3R.<br>Write 1 to RXRSSCR.SLT3C to clear this bit.<br>This bit and RXRSS3R are set if next receive action caused with SQCHxDSCyCR.ACTCODE = 03 again. |
| 2 | SLT2VLD | 0 | R | Slot 2 Valid<br>Response packet is received, or information is saved to RXRSS2R.<br>Write 1 to RXRSSCR.SLT2C to clear this bit.<br>This bit and RXRSS2R are set if next receive action caused with SQCHxDSCyCR.ACTCODE = 02 again. |
| 1 | SLT1VLD | 0 | R | Slot 1 Valid<br>Response packet is received, or information is saved to RXRSS1R.<br>Write 1 to RXRSSCR.SLT1C to clear this bit.<br>This bit and RXRSS1R are set if next receive action caused with SQCHxDSCyCR.ACTCODE = 01 again. |
| 0 | SLT0VLD | 0 | R | Slot 0 Valid<br>Response packet is received, or information is saved to RXRSS0R.<br>Write 1 to RXRSSCR.SLT0C to clear this bit.<br>This bit and RXRSS0R are set if next receive action caused with SQCHxDSCyCR.ACTCODE = 00 again. |

### 34.3.1.25 Rx Result Saved Status Clear Register (RXRSSCR)

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | SLT3C | SLT2C | SLT1C | SLT0C |
| 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 | R0W | R0W | R0W | R0W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 25 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 24 | — | 0 | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 23 to 4 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 3 | SLT3C | 0 | R0W | RXRSSR.SLT3VLD Clear<br>Write 1 to clear RXRSSR.SLT3VLD. Writing “0” has no effect. |
| 2 | SLT2C | 0 | R0W | RXRSSR.SLT2VLD Clear<br>Write 1 to clear RXRSSR.SLT2VLD. Writing “0” has no effect. |
| 1 | SLT1C | 0 | R0W | RXRSSR.SLT1VLD Clear<br>Write 1 to clear RXRSSR.SLT1VLD. Writing “0” has no effect. |
| 0 | SLT0C | 0 | R0W | RXRSSR.SLT0VLD Clear<br>Write 1 to clear RXRSSR.SLT0VLD. Writing “0” has no effect. |

### 34.3.1.26 Rx Result Info Overwrite Status Register (RXRINFOOWSR)

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | SLT3INFOOW | SLT2INFOOW | SLT1INFOOW | SLT0INFOOW |
| 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 25 | — | All 0 | R | Reserved<br>When read, the initial value is read. The written value will be ignored. |
| 24 | — | 0 | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 23 to 4 | — | All 0 | R | Reserved<br>When read, the initial value is read. The written value will be ignored. |
| 3 | SLT3INFOOW | 0 | R | Slot 3 Information Overwrite<br>This bit is copy of RXRSS3R.INFOOW. |
| 2 | SLT2INFOOW | 0 | R | Slot 2 Information Overwrite<br>This bit is copy of RXRSS2R.INFOOW. |
| 1 | SLT1INFOOW | 0 | R | Slot 1 Information Overwrite<br>This bit is copy of RXRSS1R.INFOOW. |
| 0 | SLT0INFOOW | 0 | R | Slot 0 Information Overwrite<br>This bit is copy of RXRSS0R.INFOOW. |

### 34.3.1.27 Rx Result Info Overwrite Status Clear Register (RXRINFOOWSCR)

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | SLT3INFOOWC | SLT2INFOOWC | SLT1INFOOWC | SLT0INFOOWC |
| 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 | R0W | R0W | R0W | R0W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 25 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 24 | — | 0 | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 23 to 4 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 3 | SLT3INFOOWC | 0 | R0W | RXRSS3R.INFOOW Clear<br>Write 1 to clear RXRSS3R.INFOOW. Writing “0” has no effect.<br>To prevent missing of RXRSS3R.INFOOW, clear RXRSS3R.INFOOW before clearing RXRSSR.SLT3VLD. |
| 2 | SLT2INFOOWC | 0 | R0W | RXRSS2R.INFOOW Clear<br>Write 1 to clear RXRSS2R.INFOOW. Writing “0” has no effect.<br>To prevent missing of RXRSS2R.INFOOW, clear RXRSS2R.INFOOW before clearing RXRSSR.SLT2VLD. |
| 1 | SLT1INFOOWC | 0 | R0W | RXRSS1R.INFOOW Clear<br>Write 1 to clear RXRSS1R.INFOOW. Writing “0” has no effect.<br>To prevent missing of RXRSS1R.INFOOW, clear RXRSS1R.INFOOW before clearing RXRSSR.SLT1VLD. |
| 0 | SLT0INFOOWC | 0 | R0W | RXRSS0R.INFOOW Clear<br>Write 1 to clear RXRSS0R.INFOOW. Writing “0” has no effect.<br>To prevent missing of RXRSS0R.INFOOW, clear RXRSS0R.INFOOW before clearing RXRSSR.SLT0VLD. |

### 34.3.1.28 Rx Result Save Slot x Register (RXRSSxR)

x = 0, 1, 2, 3.

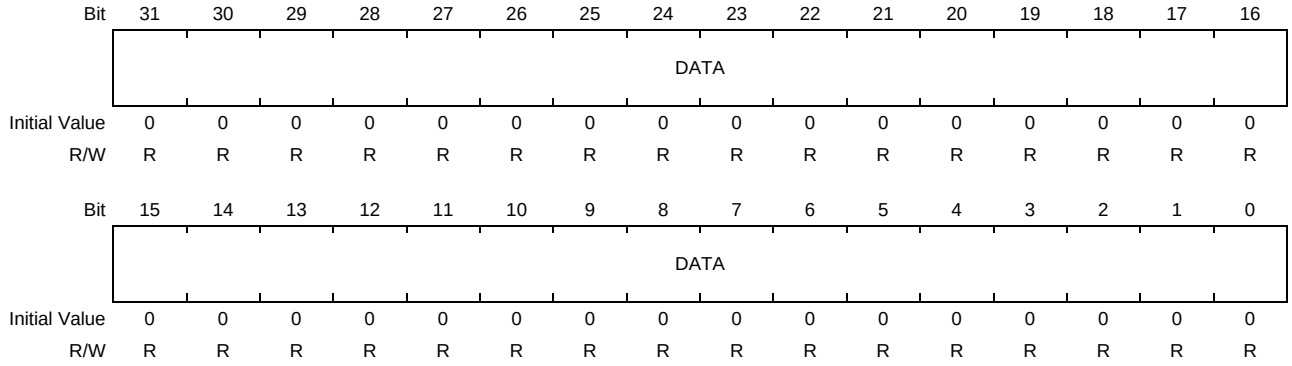
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | INFOOW | RXAKE | RXCORERR | RXPKTDFAIL | RXFAIL | RXFATALERR | RXSUC | FMT | VC | | DT | | | | | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | DATA1 | | | | | | | | DATA0 | | | | | | | |
| 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 | INFOOW | 0 | R | Information Overwrite<br>This bit is set if information of RXRSSxR are updated when RXRSSR.SLTxVLD is 1.<br>Write 1 to RXRINFOOWSCR.SLTxINFOOWC to clear this bit. |
| 30 | RXAKE | 0 | R | Receive Acknowledge and Error Report Packet<br>RXSR.RXAKE is also reported. |
| 29 | RXCORERR | 0 | R | Receive Correctable Error<br>Received packet has Correctable Error.<br>RXSR.ECCERR1B is also reported. |
| 28 | RXPKTDFAIL | 0 | R | Receive Packet Data Fail<br>Packet Header Saved Correctly but Data did not save correctly.<br>One or more of the following bits are also reported.<br>RXSR.CRCERR, RXSR.WCERR, RXSR.MAXRPSZERR, RXSR.UEXPKTERR, RXSR.RXOVFERR or RXSR.IBERR.<br>When both of this bit and RXRESP are asserted, it means redundant packet received in BTA. |
| 27 | RXFAIL | 0 | R | Receive Fail<br>Expected receive did not done.<br>One or more of the following bits are also reported.<br>RXSR.PRESPTOERR, RXSR.ECCERR, RXSR.MLFERR or RXSR.NORETERR. |
| 26 | RXFATALERR | 0 | R | Fatal Error<br>Fatal timeout occurred during BTA.<br>One or more of the following bits are also reported.<br>FERRSR.TATO or FERRSR.LRXHTO. |
| 25 | RXSUC | 0 | R | Receive Success<br>When this value is “1”, the Response Packet or ACK trigger is received.<br>One or more of the following bits are also reported.<br>RXSR.RXRESP or RXSR.RXACK. |
| 24 | FMT | 0 | R | Packet format of Rx Packet Header<br>0: Short packet<br>1: Long packet<br>This field is valid when RXPKTDFAIL=1 or (RXSUC=1 and DT!=0). |
| 23 to 22 | VC | All 0 | R | Identifier of virtual channel of Rx Packet Header<br>This field is valid when (RXSUC=1 and DT!=0). |
| 21 to 16 | DT | All 0 | R | Data type of Rx Packet Header<br>H’00: ACK trigger received<br>H’01 to H’3F: Data type<br>This field is valid when RXSUC=1. |

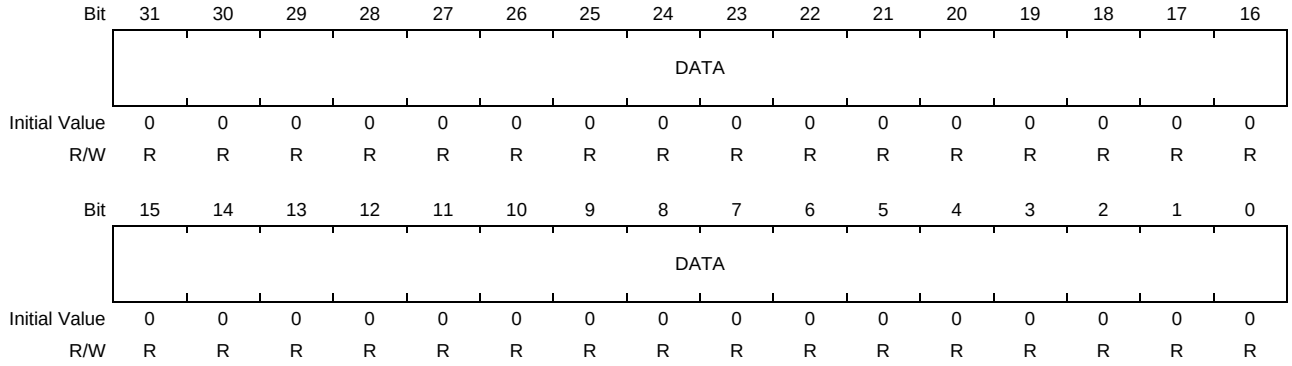
| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 to 8 | DATA1 | All 0 | R | Data1 of Rx Packet Header<br>When Rx Packet format is Long Packet, this value is upper 8 bits of the word count.<br>This field is valid when (RXSUC=1 and DT!=0). |
| 7 to 0 | DATA0 | All 0 | R | Data0 of Rx Packet Header<br>When Rx Packet format is Long Packet, this value is lower 8 bits of the word count.<br>This field is valid when (RXSUC=1 and DT!=0). |

### 34.3.1.29 Rx Packet Payload Data 0 Register (RXPPD0R)



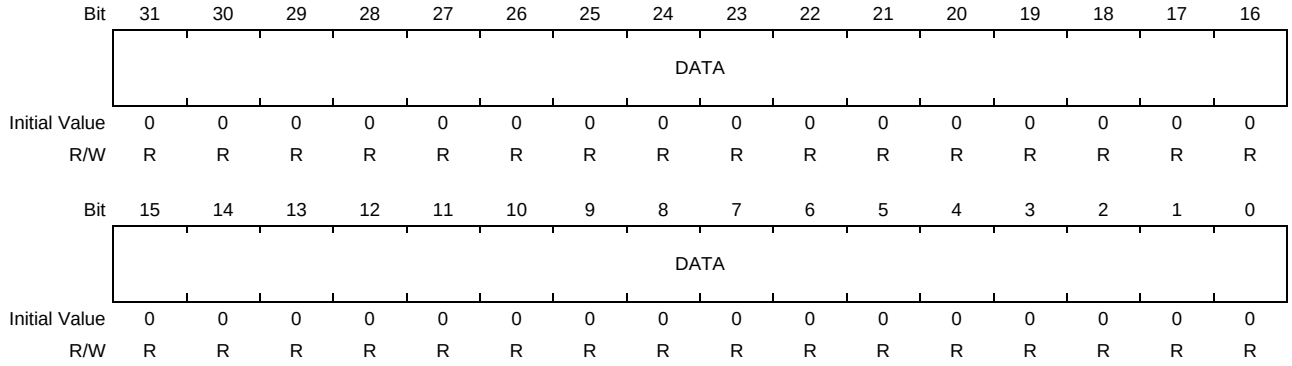

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 0 | DATA | All 0 | R | Rx Packet Payload Data 0<br>In the case of long packet receiving Operation with SQCHxDSCyBR.DTSEL = H'0, received packet payload are saved.<br>[7:0]: Data 0<br>[15:8]: Data 1<br>[23:16]: Data 2<br>[31:24]: Data 3 |

### 34.3.1.30 Rx Packet Payload Data 1 Register (RXPPD1R)



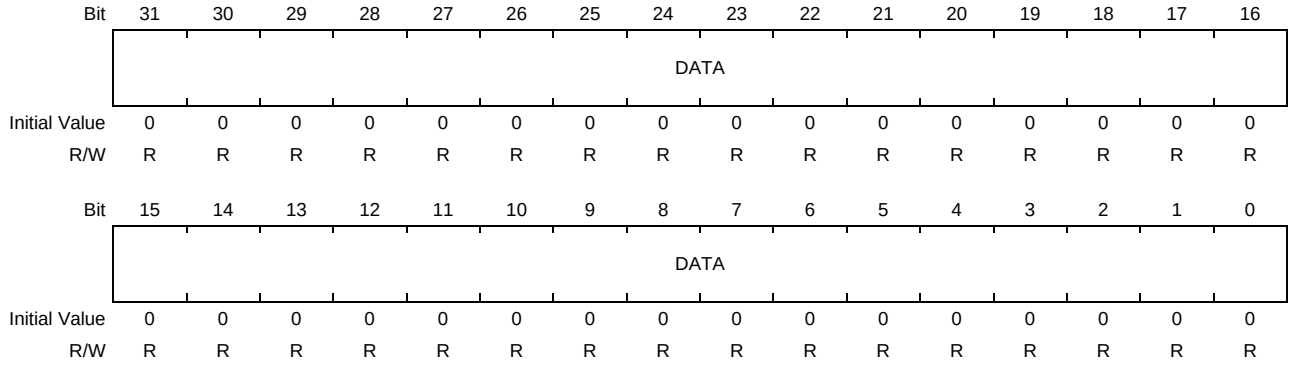

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 0 | DATA | All 0 | R | Rx Packet Payload Data 1<br>In the case of long packet receiving Operation with SQCHxDSCyBR.DTSEL = H'0.<br>Received packet payload are saved.<br>[7:0]: Data 4<br>[15:8]: Data 5<br>[23:16]: Data 6<br>[31:24]: Data 7 |

### 34.3.1.31 Rx Packet Payload Data 2 Register (RXPPD2R)



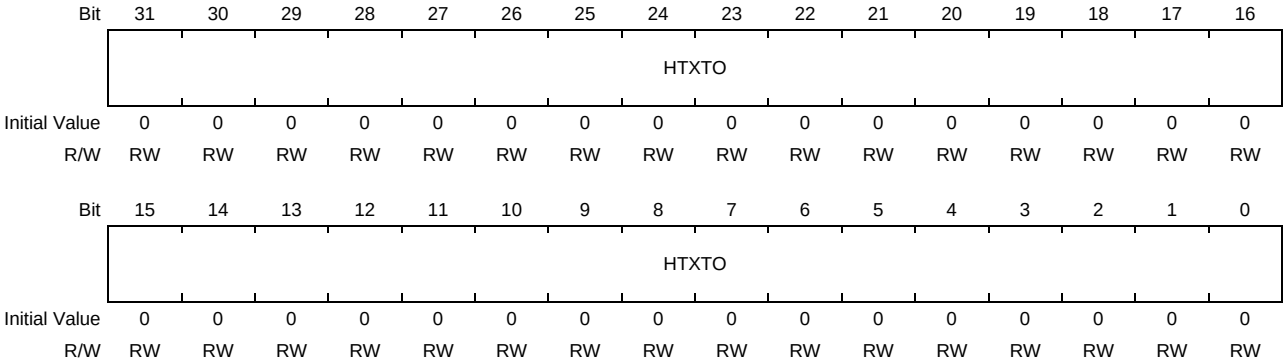

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 0 | DATA | All 0 | R | Rx Packet Payload Data 2<br>In the case of long packet receiving Operation with SQCHxDSCyBR.DTSEL = H'0.<br>Received packet payload are saved.<br>[7:0]: Data 8<br>[15:8]: Data 9<br>[23:16]: Data 10<br>[31:24]: Data 11 |

### 34.3.1.32 Rx Packet Payload Data 3 Register (RXPPD3R)



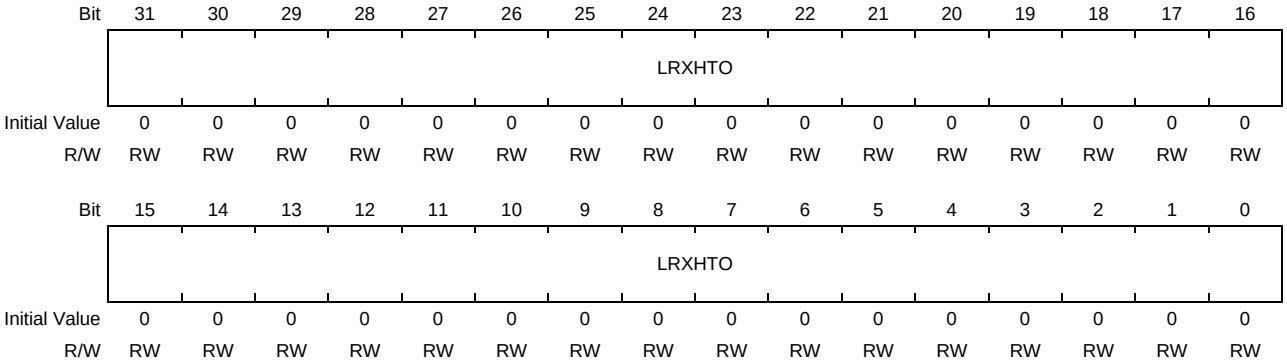

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 0 | DATA | All 0 | R | Rx Packet Payload Data 3<br>In the case of long packet receiving Operation with SQCHxDSCyBR.DTSEL = H'0.<br>Received packet payload are saved.<br>[7:0]: Data 12<br>[15:8]: Data 13<br>[23:16]: Data 14<br>[31:24]: Data 15 |

### 34.3.1.33 HSTX Timeout Set Register (HSTXTOSETR)



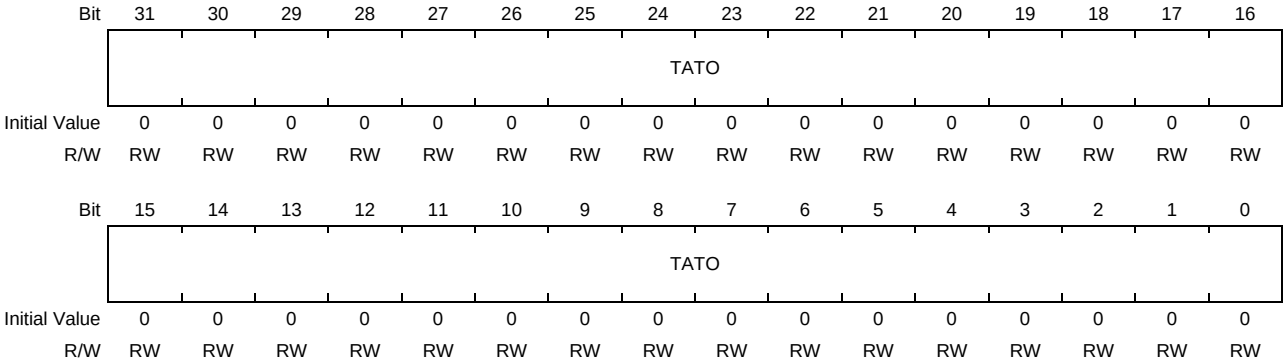

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 0 | HTXTO | All 0 | RW | HS TX Timeout Count (HTX_TO)<br>The limit time for timeout is calculated by formula of HTXTO * (period of highspeed serial UI * 32).<br>When FERRSR.HTXTO is “1”, the timeout is detected.<br>Notice, the timeout is not detected when HTXTO=0. |

### 34.3.1.34 LRX-H Timeout Set Register (LRXHTOSETR)




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 0 | LRXHTO | All 0 | RW | LP-RX Host Processor Timeout (LRX-H_TO)<br>The limit time for timeout is calculated by formula of LRXHTO * (period of MIPI_DSI_LPCLK).<br>When FERRSR.LRXHTO is “1”, the timeout is detected.<br>Notice, the timeout is not detected when LRXHTO=0. |

### 34.3.1.35 TA Timeout Set Register (TATOSETR)




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 0 | TATO | All 0 | RW | Turnaround Acknowledge Timeout (TA_TO)<br>The limit time for timeout is calculated by formula of TATO * (period of MIPI_DSI_LPCLK).<br>When FERRSR.TATO is “1”, the timeout is detected.<br>Notice, the timeout is not detected when TATO=0. |

### 34.3.1.36 Fatal Error Status Register (FERRSR)

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | ERRCLP1S | ERRCLP0S | — | — | — | — | — | — | ERRCLP1 | ERRCLP0 | ERRCTRL | ERRSYNESC | ERRESC |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | TATO | LRXHTO | HTXTO |
| 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 29 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 28 | ERRCLP1S | 0 | R | Status of Contention Error of LP1 (ErrContentionLP1) |
| 27 | ERRCLP0S | 0 | R | Status of Contention Error of LP0 (ErrContentionLP0) |
| 26 to 21 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 20 | ERRCLP1 | 0 | R | Contention Error of LP1 (ErrContentionLP1)<br>When this value is “1”, the Contention Error of LP1 (ErrContentionLP1) is detected on PPI Data Lane0.<br>Write “1” to FERRSCR.ERRCLP1 register to clear this field. |
| 19 | ERRCLP0 | 0 | R | Contention Error of LP0 (ErrContentionLP0)<br>When this value is “1”, the Contention Error of LP0 (ErrContentionLP0) is detected on PPI Data Lane0.<br>Write “1” to FERRSCR.ERRCLP0 register to clear this field. |
| 18 | ERRCTRL | 0 | R | Control Error (ErrControl)<br>When this value is “1”, the Control Error (ErrControl) is detected on PPI Data Lane0.<br>Write “1” to FERRSCR.ERRCTRL register to clear this field. |
| 17 | ERRSYNESC | 0 | R | Sync Error of LPDT (ErrSyncEsc)<br>When this value is “1”, the Sync Error of LPDT (ErrSyncEsc) is detected on PPI Data Lane0.<br>Write “1” to FERRSCR.ERRSYNESC register to clear this field. |
| 16 | ERRESC | 0 | R | Escape mode Entry Error (ErrEsc)<br>When this value is “1”, the Escape mode Entry Error (ErrEsc) is detected on PPI Data Lane0.<br>Write “1” to FERRSCR.ERRESC register to clear this field. |
| 15 to 3 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 2 | TATO | 0 | R | Turnaround Acknowledge Timeout (TA_TO)<br>When this value is “1”, the Turnaround Acknowledge Timeout (TA_TO) is detected.<br>Write “1” to FERRSCR.TATO register to clear this field. |
| 1 | LRXHTO | 0 | R | LP-RX Host Processor Timeout (LRX-H_TO)<br>When this value is “1”, the LP-RX Host Processor Timeout (LRX-H_TO) is detected.<br>Write “1” to FERRSCR.LRXHTO register to clear this field. |
| 0 | HTXTO | 0 | R | HS TX Timeout (HTX_TO)<br>When this value is “1”, the HS TX Timeout (HTX_TO) is detected.<br>Write “1” to FERRSCR.HTXTO register to clear this field. |

### 34.3.1.37 Fatal Error Status Clear Register (FERRSCR)

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | ERRCLP1 | ERRCLP0 | ERRCTRL | ERRSYNESC | ERRESC |
| 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 | R0W | R0W | R0W | R0W | R0W |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | TATO | LRXHTO | HTXTO |
| 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 | R0W | R0W | R0W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 21 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 20 | ERRCLP1 | 0 | R0W | FERRSR.ERRCLP1 Clear<br>Set to “1” to clear the FERRSR.ERRCLP1.<br>Writing “0” has no effect. |
| 19 | ERRCLP0 | 0 | R0W | FERRSR.ERRCLP0 Clear<br>Set to “1” to clear the FERRSR.ERRCLP0.<br>Writing “0” has no effect. |
| 18 | ERRCTRL | 0 | R0W | FERRSR.ERRCTRL Clear<br>Set to “1” to clear the FERRSR.ERRCTRL.<br>Writing “0” has no effect. |
| 17 | ERRSYNESC | 0 | R0W | FERRSR.ERRSYNESC Clear<br>Set to “1” to clear the FERRSR.ERRSYNESC.<br>Writing “0” has no effect. |
| 16 | ERRESC | 0 | R0W | FERRSR.ERRESC Clear<br>Set to “1” to clear the FERRSR.ERRESC.<br>Writing “0” has no effect. |
| 15 to 3 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 2 | TATO | 0 | R0W | FERRSR.TATO Clear<br>Set to “1” to clear the FERRSR.TATO.<br>Writing “0” has no effect. |
| 1 | LRXHTO | 0 | R0W | FERRSR.LRXHTO Clear<br>Set to “1” to clear the FERRSR.LRXHTO.<br>Writing “0” has no effect. |
| 0 | HTXTO | 0 | R0W | FERRSR.HTXTO Clear<br>Set to “1” to clear the FERRSR.HTXTO.<br>Writing “0” has no effect. |

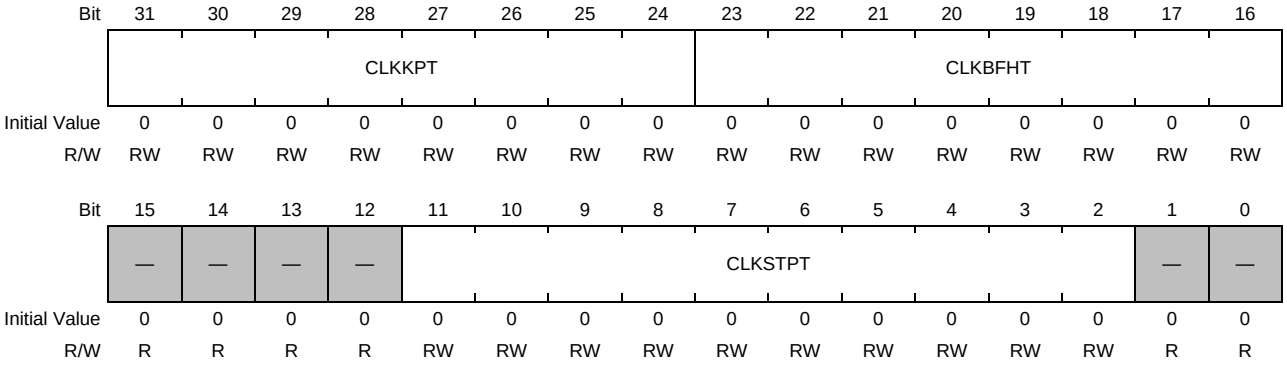
### 34.3.1.38 Fatal Error Interrupt Enable Register (FERRIER)

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | ERRCLP1 | ERRCLP0 | ERRCTRL | ERRSYNESC | ERRESC |
| 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 | RW | RW | RW | RW | RW |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | TATO | LRXHTO | HTXTO |
| 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 | RW | RW | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 21 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 20 | ERRCLP1 | 0 | RW | Interrupt enable for the Contention Error of LP1 on PPI Data Lane0 (ErrContentionLP1)<br>0: Disable interrupt of dsi_int_ferr when FERRSR.ERRCLP1=1<br>1: Enable interrupt of dsi_int_ferr when FERRSR.ERRCLP1=1 |
| 19 | ERRCLP0 | 0 | RW | Interrupt enable for the Contention Error of LP0 on PPI Data Lane0 (ErrContentionLP0)<br>0: Disable interrupt of dsi_int_ferr when FERRSR.ERRCLP0=1<br>1: Enable interrupt of dsi_int_ferr when FERRSR.ERRCLP0=1 |
| 18 | ERRCTRL | 0 | RW | Interrupt enable for the Control Error on PPI Data Lane0 (ErrControl)<br>0: Disable interrupt of dsi_int_ferr when FERRSR.ERRCTRL=1<br>1: Enable interrupt of dsi_int_ferr when FERRSR.ERRCTRL=1 |
| 17 | ERRSYNESC | 0 | RW | Interrupt enable for the LPDT Sync Error on PPI Data Lane0 (ErrSyncEsc)<br>0: Disable interrupt of dsi_int_ferr when FERRSR.ERRSYNESC=1<br>1: Enable interrupt of dsi_int_ferr when FERRSR.ERRSYNESC=1 |
| 16 | ERRESC | 0 | RW | Interrupt enable for the Escape mode Error on PPI Data Lane0 (ErrEsc)<br>0: Disable interrupt of dsi_int_ferr when FERRSR.ERRESC=1<br>1: Enable interrupt of dsi_int_ferr when FERRSR.ERRESC=1 |
| 15 to 3 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 2 | TATO | 0 | RW | Interrupt enable for Turnaround Acknowledge Timeout (TA_TO)<br>0: Disable interrupt of dsi_int_ferr when FERRSR.TATO=1<br>1: Enable interrupt of dsi_int_ferr when FERRSR.TATO=1 |
| 1 | LRXHTO | 0 | RW | Interrupt enable for LP-RX Host Processor Timeout (LRX_H_TO)<br>0: Disable interrupt of dsi_int_ferr when FERRSR.LRXHTO=1<br>1: Enable interrupt of dsi_int_ferr when FERRSR.LRXHTO=1 |
| 0 | HTXTO | 0 | RW | Interrupt enable for HS TX Timeout (HTX_TO)<br>0: Disable interrupt of dsi_int_ferr when FERRSR.HTXTO=1<br>1: Enable interrupt of dsi_int_ferr when FERRSR.HTXTO=1 |

### 34.3.1.39 Clock Lane Stop Time Set Register (CLSTPTSETR)




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 24 | CLKKPT | All 0 | RW | Clock Keep Time<br>Period time what HS Clock Lane keep active after HS packet transferred.<br>The time is calculated by formula of CLKKPT * (period of high-speed serial UI * 32).<br>This time corresponds to D-PHY specification's ($T_{HS-TRAIL}$ + $T_{CLK-POST}$). Set larger value than calculated.<br>Set appropriate value (other than 0) before HSCLKSETR.HSCLKRUN is set to 1. |
| 23 to 16 | CLKBFHT | All 0 | RW | Clock Beforehand Time<br>Period time what Clock Lane back to HS beforehand to HS packet transfer.<br>If HS transfers are scheduled in LP, clock lane will start beforehand to HS transfer by this setting.<br>The time is calculated by formula of CLKBFHT * (period of high-speed serial UI * 32).<br>This time corresponds to D-PHY specification's ($T_{LPX}$ + $T_{CLK-PREPARE}$ + $T_{CLK-ZERO}$ + $T_{CLK-PRE}$). Set larger value than calculated.<br>Set appropriate value (other than 0) before HSCLKSETR.HSCLKRUN is set to 1. |
| 15 to 12 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 11 to 2 | CLKSTPT | All 0 | RW | Clock Stop Time<br>Period time what HS Clock Lane is changed to LP and back to HS.<br>If data lane will end HS and time of CLKKPT is over, and scheduled time till next<br>HS transfers is more than this setting, clock lane will be changed to LP.<br>This setting is no meaning if HSCLKSETR.HSCLKMODE is 1b or HSCLKSETR.HSCLKRUN is 0b.<br>The time is calculated by formula of CLKSTPT * (period of high-speed serial UI * 32)<br>This time correspond to D-PHY specification's ($T_{HS-TRAIL}$ + $T_{CLK-POST}$ + $T_{CLK-TRAIL}$ + $T_{HS-EXIT}$ + $T_{LPX}$ *2 + $T_{LPX}$ + $T_{CLK-PREPARE}$ + $T_{CLK-ZERO}$ + $T_{CLK-PRE}$ + $T_{LPX}$ + $T_{HS-PREPARE}$ + $T_{HS-ZERO}$ + $T_{HS-SYNC}$). About each parameter, please refer D-PHY specification and **Figure 34.2**. Set larger value than calculated.<br>Setting 0 to this field is prohibited. |
| 1 to 0 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |

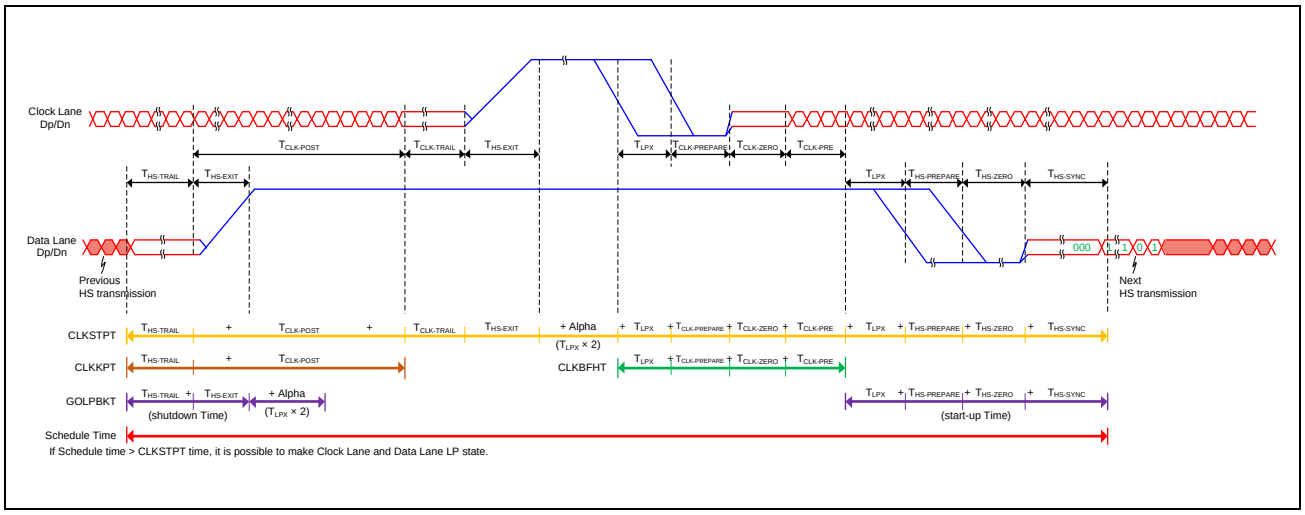



Figure 34.2 Switching the Clock Lane between Clock Transmission and Low-Power Mode

### 34.3.1.40 LP Transition Time Set Register (LPTRNSTSETR)

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | GOLPBKT | | | | | | | | | |
| 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 | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 10 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 9 to 0 | GOLPBKT | All 0 | RW | Go LP and Back Time<br>Period time what HS Data Lanes are changed to LP and back to HS.<br>If scheduled time between HS transfers is more than this setting, data lanes will be changed to LP.<br>The time is calculated by formula of GOLPBKT * (period of high-speed serial 8*UI).<br>This time correspond to D-PHY specification's ($T_{HS-TRAIL}$ + $T_{HS-EXIT}$ + $T_{LPX}$ *2 + $T_{LPX}$ + $T_{HS-PREPARE}$ + $T_{HS-ZERO}$ + $T_{HS-SYNC}$). About each parameter, please refer DPHY specification and **Figure 34.3**.<br>The unit of this setting is 8UI, so if the result of this formula contains half-finished than 8UI, round it up.<br>Set the rounded up value plus at least 3. |

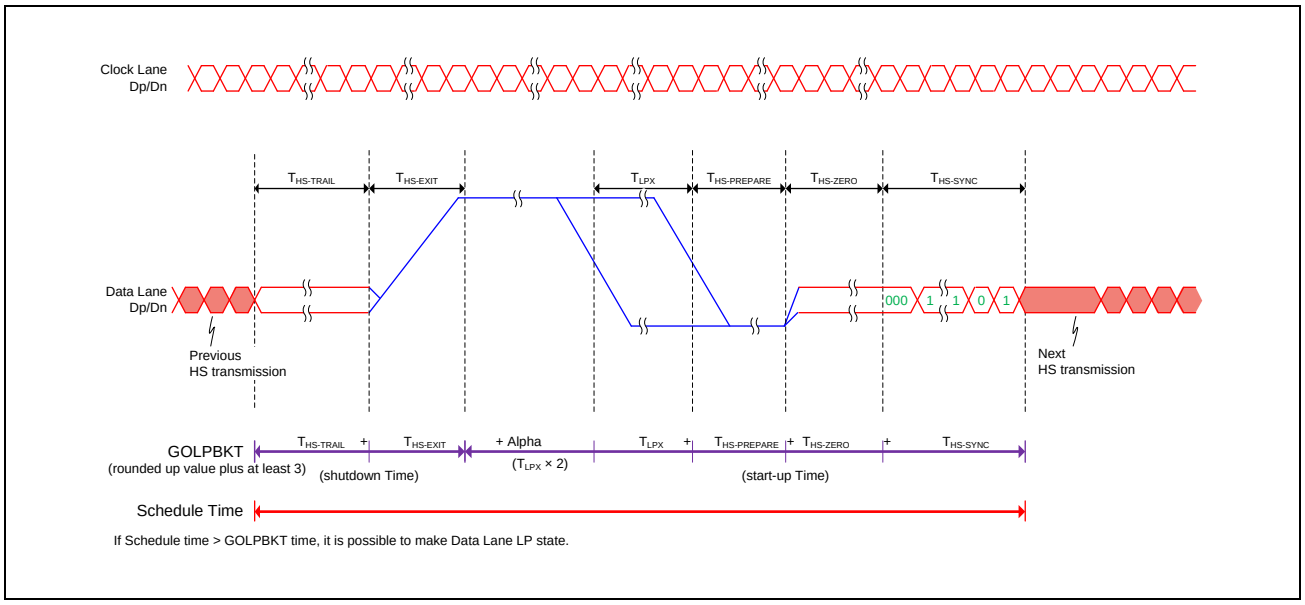



Figure 34.3 Switching the Data Lane between High-Speed Mode and Low-Power Mode

### 34.3.1.41 Physical Lane Status Register (PLSR)

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | DLFROMULPS | DLTOULPS | CLHS2LP | CLLP2HS | CLFROMULPS | CLTOULPS | — | — | — | — | — | — | — | — |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | DL0DIR | — | DL0TX2RX | DL0RX2TX | DLSTPST | | | | DLULPSACTN | | | | DL0RXULPSESC | DL0RXLPDTESC | CLSTPST | CLULPSACTN |
| Initial Value | 0 | 0 | 0 | 0 | *1 | *1 | *1 | *1 | 1 | 1 | 1 | 1 | 0 | 0 | *2 | 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, 30 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 29 | DLFROMULPS | 0 | R | From ULPS (Data Lane)<br>When this value is “1”, all PPI Data Lanes which enabled by TXSETR.NUMLANEUSE were changed to StopState from ULPS by ULPS Exit.<br>Write “1” to PLSCR.DLFROMULPS register to clear this field. |
| 28 | DLTOULPS | 0 | R | To ULPS (Data Lane)<br>When this value is “1”, all PPI Data Lanes which enabled by TXSETR.NUMLANEUSE were changed to ULPS from StopState by ULPS Enter.<br>Write “1” to PLSCR.DLTOULPS register to clear this field. |
| 27 | CLHS2LP | 0 | R | HS to LP (Clock Lane)<br>When this value is “1”, the clock lane transitions to LP from HS is detected.<br>Checking this bit during non-continuous clock mode operation is no meaning. So, ignore this bit when HSCLKSETR.HSCLKMODE=0b.<br>Write “1” to PLSCR.CLHS2LP register to clear this field. |
| 26 | CLLP2HS | 0 | R | LP to HS (Clock Lane)<br>When this value is “1”, the clock lane transitions to HS from LP is detected.<br>Checking this bit during non-continuous clock mode operation is no meaning. So, ignore this bit when HSCLKSETR.HSCLKMODE=0b.<br>Write “1” to PLSCR.CLLP2HS register to clear this field. |
| 25 | CLFROMULPS | 0 | R | From ULPS (Clock Lane)<br>When this value is “1”, the clock lane exits from ULPS is detected.<br>Write “1” to PLSCR.CLFROMULPS register to clear this field. |
| 24 | CLTOULPS | 0 | R | To ULPS (Clock Lane)<br>When this value is “1”, the clock lane enters to ULPS is detected.<br>Write “1” to PLSCR.CLTOULPS register to clear this field. |
| 23 to 16 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 15 | DL0DIR | 0 | R | Direction (Tx or Rx) of Data Lane0<br>0: PHY is in Tx Mode<br>1: PHY is in Rx Mode<br>This field shows a status of direction on PPI data lane0 after synchronized to MIPI_DSI_LPCLK.<br>So, this field does not show current status, during MIPI_DSI_LPCLK is not running. |
| 14 | — | 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 13 | DL0TX2RX | 0 | R | Tx to Rx on Data Lane0<br>When this value is “1”, the transition of Direction from 0 (Tx) to 1(Rx) is detected on Data Lane0.<br>Write “1” to PLSCR.DL0TX2RX register to clear this field. |
| 12 | DL0RX2TX | 0 | R | Rx to Tx on Data Lane0<br>When this value is “1”, the transition of Direction from 1 (Rx) to 0(Tx) is detected on Data Lane0.<br>Write “1” to PLSCR.DL0RX2TX register to clear this field. |
| 11 to 8 | DLSTPST | *1 | R | Status of Stopstate on Data Lane<br>bit 3: Stopstate of Data Lane3<br>bit 2: Stopstate of Data Lane2<br>bit 1: Stopstate of Data Lane1<br>bit 0: Stopstate of Data Lane0<br>This field shows a status of Stopstate on PPI data lane after synchronized to MIPI_DSI_LPCLK.<br>So, this field does not show current status, during MIPI_DSI_LPCLK is not running. |
| 7 to 4 | DLULPSACTN | H’F | R | Status of UlpsActiveNot on Data Lane<br>bit 3: UlpsActiveNot of Data Lane3<br>bit 2: UlpsActiveNot of Data Lane2<br>bit 1: UlpsActiveNot of Data Lane1<br>bit 0: UlpsActiveNot of Data Lane0<br>This field shows a status of UlpsActiveNot on PPI data lane after synchronized to MIPI_DSI_LPCLK.<br>So, this field does not show current status, during MIPI_DSI_LPCLK is not running. |
| 3 | DL0RXULPSESC | 0 | R | Status of RxUlpsEsc on Data Lane0<br>This field shows a status of RxUlpsEsc on PPI data lane0 after synchronized to MIPI_DSI_LPCLK.<br>So, this field does not show current status, during MIPI_DSI_LPCLK is not running. |
| 2 | DL0RXLPDTESC | 0 | R | Status of RxLpdtEsc on Data Lane0<br>This field shows a status of RxLpdtEsc on PPI data lane0 after synchronized to MIPI_DSI_LPCLK.<br>So, this field does not show current status, during MIPI_DSI_LPCLK is not running. |
| 1 | CLSTPST | *2 | R | Status of Stopstate on PPI Clock Lane<br>This field shows a status of Stopstate on PPI clock lane after synchronized to MIPI_DSI_LPCLK.<br>So, this field does not show current status, during MIPI_DSI_LPCLK is not running. |
| 0 | CLULPSACTN | 1 | R | Status of UlpsActiveNot on PPI Clock Lane<br>This field shows a status of UlpsActiveNot on PPI clock lane after synchronized to MIPI_DSI_LPCLK.<br>So, this field does not show current status, during MIPI_DSI_LPCLK is not running. |

Note 1. This Initial Value depends on ppi_dl0_stopstate, ppi_dl1_stopstate, ppi_dl2_stopstate and ppi_dl3_stopstate.

Note 2. This Initial Value depends on ppi_cl_stopstate.

### 34.3.1.42 Physical Lane Status Clear Register (PLSCR)

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | DLFROMULPS | DLTOULPS | CLHS2LP | CLLP2HS | CLFROMULPS | CLTOULPS | — | — | — | — | — | — | — | — |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R0W | R0W | R0W | R0W | R0W | R0W | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | DL0TX2RX | DL0RX2TX | — | — | — | — | — | — | — | — | — | — | — | — |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R0W | R0W | R | R | R | R | R | R | R | R | R | R | R | R |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31, 30 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 29 | DLFROMULPS | 0 | R0W | PLSR.DLFROMULPS Clear<br>Set to “1” to clear PLSR.DLFROMULPS.<br>Writing “0” has no effect. |
| 28 | DLTOULPS | 0 | R0W | PLSR.DLTOULPS Clear<br>Set to “1” to clear PLSR.DLTOULPS.<br>Writing “0” has no effect. |
| 27 | CLHS2LP | 0 | R0W | PLSR.CLHS2LP Clear<br>Set to “1” to clear PLSR.CLHS2LP.<br>Writing “0” has no effect. |
| 26 | CLLP2HS | 0 | R0W | PLSR.CLLP2HS Clear<br>Set to “1” to clear PLSR.CLLP2HS.<br>Writing “0” has no effect. |
| 25 | CLFROMULPS | 0 | R0W | PLSR.CLFROMULPS Clear<br>Set to “1” to clear PLSR.CLFROMULPS.<br>Writing “0” has no effect. |
| 24 | CLTOULPS | 0 | R0W | PLSR.CLTOULPS Clear<br>Set to “1” to clear PLSR.CLTOULPS.<br>Writing “0” has no effect. |
| 23 to 14 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 13 | DL0TX2RX | 0 | R0W | PLSR.DL0TX2RX Clear<br>Set to “1” to clear PLSR.DL0TX2RX.<br>Writing “0” has no effect. |
| 12 | DL0RX2TX | 0 | R0W | PLSR.DL0RX2TX Clear<br>Set to “1” to clear PLSR.DL0RX2TX.<br>Writing “0” has no effect. |
| 11 to 0 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |

### 34.3.1.43 Physical Lane Interrupt Enable Register (PLIER)

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | DLFROMULPS | DLTOULPS | CLHS2LP | CLLP2HS | CLFROMULPS | CLTOULPS | — | — | — | — | — | — | — | — |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | RW | RW | RW | RW | RW | RW | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | DL0TX2RX | DL0RX2TX | — | — | — | — | — | — | — | — | — | — | — | — |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | RW | RW | R | R | R | R | R | R | R | R | R | R | R | R |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31, 30 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 29 | DLFROMULPS | 0 | RW | Interrupt enable for the all data lane transition from ULPS<br>0: Disable interrupt of dsi_int_ppi when PLSR.DLFROMULPS =1<br>1: Enable interrupt of dsi_int_ppi when PLSR.DLFROMULPS =1 |
| 28 | DLTOULPS | 0 | RW | Interrupt enable for the all data lane transition to ULPS<br>0: Disable interrupt of dsi_int_ppi when PLSR.DLTOULPS =1<br>1: Enable interrupt of dsi_int_ppi when PLSR.DLTOULPS =1 |
| 27 | CLHS2LP | 0 | RW | Interrupt enable for the clock lane transition to LP from HS<br>0: Disable interrupt of dsi_int_ppi when PLSR.CLHS2LP=1<br>1: Enable interrupt of dsi_int_ppi when PLSR.CLHS2LP=1 |
| 26 | CLLP2HS | 0 | RW | Interrupt enable for the clock lane transition to HS from LP<br>0: Disable interrupt of dsi_int_ppi when PLSR.CLLP2HS=1<br>1: Enable interrupt of dsi_int_ppi when PLSR.CLLP2HS=1 |
| 25 | CLFROMULPS | 0 | RW | Interrupt enable for the clock lane transition from ULPS<br>0: Disable interrupt of dsi_int_ppi when PLSR.CLFROMULPS=1<br>1: Enable interrupt of dsi_int_ppi when PLSR.CLFROMULPS=1 |
| 24 | CLTOULPS | 0 | RW | Interrupt enable for the clock lane transition to ULPS<br>0: Disable interrupt of dsi_int_ppi when PLSR.CLTOULPS=1<br>1: Enable interrupt of dsi_int_ppi when PLSR.CLTOULPS=1 |
| 23 to 14 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 13 | DL0TX2RX | 0 | RW | Interrupt enable for the data lane 0 change direction from Tx to Rx<br>0: Disable interrupt of dsi_int_ppi when PLSR.DL0TX2RX =1<br>1: Enable interrupt of dsi_int_ppi when PLSR.DL0TX2RX =1 |
| 12 | DL0RX2TX | 0 | RW | Interrupt enable for the data lane 0 change direction from Rx to Tx<br>0: Disable interrupt of dsi_int_ppi when PLSR.DL0RX2TX =1<br>1: Enable interrupt of dsi_int_ppi when PLSR.DL0RX2TX =1 |
| 11 to 0 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |

### 34.3.1.44 Video-Input Channel 1 Set 0 Register (VICH1SET0R)

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | VSEN | — | HFPNOLP | HBPNOLP | HSANOLP | — | — | — | — | — | — | VSTPAFT | VSTART |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | RW | RW | RW | RW | RW | RW | R | R | R | R | R | R0W | R0W | R0W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 14 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 13 | — | 0 | RW | Reserved<br>When read, the initial value is read. The written value will be ignored. |
| 12 | VSEN | 0 | RW | Video Skew Calibration Enable<br>Skew Calibration is not supported. This bit should be fixed at 0. |
| 11 | — | 0 | RW | Reserved<br>When read, the initial value is read. The written value will be ignored. |
| 10 | HFPNOLP | 0 | RW | HFP period No LP<br>Set suppress not to transition LP during HFP period.<br>0: Does not suppress transition to LP during the period<br>1: Suppress transition to LP during the period and keep HS. HS is kept by sending blanking packet. |
| 9 | HBPNOLP | 0 | RW | HBP period No LP<br>Set suppress not to transition LP during HBP period.<br>0: Does not suppress transition to LP during the period<br>1: Suppress transition to LP during the period and keep HS. HS is kept by sending blanking packet. |
| 8 | HSANOLP | 0 | RW | HSA period No LP<br>Set suppress not to transition LP during HSA period. When TXESYNC=0, this field has no meaning.<br>0: Does not suppress transition to LP during the period<br>1: Suppress transition to LP during the period and keep HS. HS is kept by sending blanking packet. |
| 7 to 3 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 2 | — | 0 | R0W | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 1 | VSTPAFT | 0 | R0W | Video-Input signals stop after Video-Input Operation stop<br>Stop Video-Input Operation.<br>Video-Input Operation will end at the end of video-frame. After Video-Input Operation end, stop Video-Input signals.<br>If the Video-Input operation is stopped, VICH1SR.RUNNING bit will change to 0.<br>Set “1” to both VSTPAFT and VSTART is prohibited.<br>Set “1” during VICH1SR.RUNNING=0 is prohibited. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 0 | VSTART | 0 | R0W | Video-Input Operation Start<br>Write 1 to start Video-Input Operation.<br>Set “1” to both VSTART and VSTPAFT is prohibited.<br>Set “1” during VICH1SR.RUNNING=1 is prohibited. |

### 34.3.1.45 Video-Input Channel 1 Set 1 Register (VICH1SET1R)

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | CSPC | — | BPP | | | | | | CHBUFSZ | | | | — | — | — | — |
| Initial Value | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | DLY | | | | | | | | | | | | — | — |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | R | R |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 | CSPC | 0 | R | Color space of input video pixel<br>This field reflects color space of VICH1PPSETR.DT selected data type.<br>0: RGB<br>1: Reserved |
| 30 | — | 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 29 to 24 | BPP | H’10 | R | Bit Per Pixel of input video pixel<br>This field reflects bits per pixel of VICH1PPSETR.DT selected data type.<br>H’10: Reserved<br>H’12: 18bpp<br>H’18: 24bpp, 18bpp Loosely (RGB) |
| 23 to 20 | CHBUFSZ | H’6 | R | Channel Buffer Size<br>H’6: 8kB |
| 19 to 18 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 17 to 16 | — | H’3 | R | Reserved<br>When read, the initial value is read. The written value will be ignored. |
| 15 to 14 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 13 to 2 | DLY | All 0 | RW | Delay Value<br>The delay Time is calculated by formula of DLY * (period of high-speed serial UI * 32).<br>Setting 0 to this field is prohibited.<br>If VICH1SET0R.HFPNOLP=1, set this field more than HFP period.<br>If VICH1SET0R.HBPNOLP=1, set this field more than HBP period.<br>If VICH1SET0R.HSANOLP=1, set this field more than HSA period. |
| 1 to 0 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |

### 34.3.1.46 Video-Input Channel 1 Status Register (VICH1SR)

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | VBUFOVF | VBUFUDF | — | TIMERR | — | — | — | — |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | VIRDY | RUNNING | STOP | START |
| 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 26 | — | All 0 | R | Reserved<br>When read, the initial value is read. The written value will be ignored. |
| 25, 24 | — | 00 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 23 | VBUFOVF | 0 | R | Video-Input Buffer Overflow Error<br>Data overflow is occurred in Video-Input Buffer.<br>Write “1” to VICH1SCR.VBUFOVF register to clear this field. |
| 22 | VBUFUDF | 0 | R | Video-Input Buffer Underflow Error<br>Data underflow is occurred in Video-Input Buffer.<br>Write “1” to VICH1SCR.VBUFUDF register to clear this field. |
| 21 | — | 0 | R | Reserved<br>When read, the initial value is read. The written value will be ignored. |
| 20 | TIMERR | 0 | R | Timing Error<br>Timing error is occurred during video input operation.<br>Write “1” to VICH1SCR.TIMERR register to clear this field. |
| 19 to 15 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 14 | — | 0 | R | Reserved<br>When read, the initial value is read. The written value will be ignored. |
| 13 to 4 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 3 | VIRDY | 0 | R | Video-Input Signal Accept Ready<br>The value will be set to 1 when the input signal can accept by Video-Input Operation channel 1 ready. |
| 2 | RUNNING | 0 | R | Video-Input Operation channel 1 Transfer Running<br>The value will be set to 1 during the Video-Input Operation channel 1 Transfer running.<br>VICH1SET0R, VICH1SET1R, VICH1PPSETR, VICH1VSSETR, VICH1VPSETR, VICH1HSSETR and VICH1HPSETR are prohibited to change value during running (VICH1SR.RUNNING=1). |
| 1 | STOP | 0 | R | Video Input Operation channel 1 Transfer Stop<br>Write “1” to VICH1SCR.STOP register to clear this field. |
| 0 | START | 0 | R | Video Input Operation channel 1 Transfer Start.<br>Write “1” to VICH1SCR.START register to clear this field. |

### 34.3.1.47 Video-Input Channel 1 Status Clear Register (VICH1SCR)

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | VBUFOVF | VBUFUDF | — | TIMERR | — | — | — | — |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R0W | R0W | R0W | R0W | R0W | R | R | R0W | R0W | R0W | R0W | R | R | R | R |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | VIRDY | — | STOP | START |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R0W | R | R | R | R | R | R | R | R | R | R | R0W | R | R0W | R0W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 | — | 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 30 to 26 | — | All 0 | R0W | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value.<br>Operation is not guaranteed if a value other than the initial value is written. |
| 25 to 24 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 23 | VBUFOVF | 0 | R0W | VICH1SR.VBUFOVF Clear<br>Set to “1” to clear VICH1SR.VBUFOVF.<br>Writing “0” has no effect. |
| 22 | VBUFUDF | 0 | R0W | VICH1SR.VBUFUDF Clear<br>Set to “1” to clear VICH1SR.VBUFUDF.<br>Writing “0” has no effect. |
| 21 | — | 0 | R0W | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value.<br>Operation is not guaranteed if a value other than the initial value is written. |
| 20 | TIMERR | 0 | R0W | VICH1SR.TIMERR Clear<br>Set to “1” to clear VICH1SR.TIMERR.<br>Writing “0” has no effect. |
| 19 to 15 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 14 | — | 0 | R0W | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value.<br>Operation is not guaranteed if a value other than the initial value is written. |
| 13 to 4 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 3 | VIRDY | 0 | R0W | VICH1SR.VIRDY Clear<br>Set to “1” to clear VICH1SR.VIRDY.<br>Writing “0” has no effect. |
| 2 | — | 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 1 | STOP | 0 | R0W | VICH1SR.STOP Clear<br>Set to “1” to clear VICH1SR.STOP.<br>Writing “0” has no effect. |
| 0 | START | 0 | R0W | VICH1SR.START Clear<br>Set to “1” to clear VICH1SR.START.<br>Writing “0” has no effect. |

### 34.3.1.48 Video-Input Channel 1 Interrupt Enable Register (VICH1IER)

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | VBUFOVF | VBUFUDF | — | TIMERR | — | — | — | — |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | RW | RW | RW | RW | RW | R | R | RW | RW | RW | RW | R | R | R | R |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | VIRDY | — | STOP | START |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | RW | R | R | R | R | R | R | R | R | R | R | RW | R | RW | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 | — | 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 30 to 26 | — | All 0 | RW | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 25, 24 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 23 | VBUFOVF | 0 | RW | Interrupt enable for the Video-Input Buffer Overflow Error<br>0: Disable interrupt of dsi_int_vin1 when VICH1SR.VBUFOVF=1<br>1: Enable interrupt of dsi_int_vin1 when VICH1SR.VBUFOVF=1 |
| 22 | VBUFUDF | 0 | RW | Interrupt enable for the Video-Input Buffer Underflow Error<br>0: Disable interrupt of dsi_int_vin1 when VICH1SR.VBUFUDF=1<br>1: Enable interrupt of dsi_int_vin1 when VICH1SR.VBUFUDF=1 |
| 21 | — | 0 | RW | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 20 | TIMERR | 0 | RW | Interrupt enable for the Timing Error<br>0: Disable interrupt of dsi_int_vin1 when VICH1SR.TIMERR=1<br>1: Enable interrupt of dsi_int_vin1 when VICH1SR.TIMERR=1 |
| 19 to 15 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 14 | — | 0 | RW | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 13 to 4 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 3 | VIRDY | 0 | RW | Interrupt enable for the Video Input Accept Ready<br>0: Disable interrupt of dsi_int_vin1 when VICH1SR.VIRDY=1<br>1: Enable interrupt of dsi_int_vin1 when VICH1SR.VIRDY=1 |
| 2 | — | 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 1 | STOP | 0 | RW | Interrupt enable for the Video Input Operation channel 1Transfer Stop<br>0: Disable interrupt of dsi_int_vin1 when VICH1SR.STOP=1<br>1: Enable interrupt of dsi_int_vin1 when VICH1SR.STOP=1 |
| 0 | START | 0 | RW | Interrupt enable for the Video Input Operation channel 1 Transfer Start<br>0: Disable interrupt of dsi_int_vin1 when VICH1SR.START=1<br>1: Enable interrupt of dsi_int_vin1 when VICH1SR.START=1 |

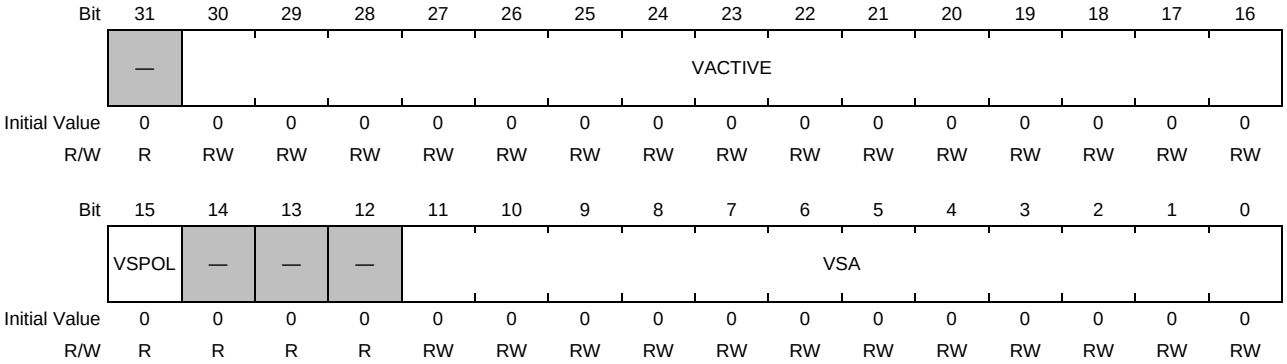
### 34.3.1.49 Video-Input Channel 1 Pixel Packet Set Register (VICH1PPSETR)

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | VC | | DT | | | | | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 |
| R/W | R | R | R | R | R | R | R | R | RW | RW | RW | RW | RW | RW | RW | RW |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TXESYNC | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | RW | R | R | R | R | R | R | R | R | R | R | R | RW | RW | RW | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 24 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 23, 22 | VC | All 0 | RW | Video-Input Operation Channel 1 Pixel Stream Packet Header Virtual Channel Identifier |
| 21 to 16 | DT | H'0E | RW | Video-Input Operation Channel 1 Pixel Stream Packet Header Data Type<br>Following values are allowed to set.<br>H'0E: Reserved<br>H'1E: Packed Pixel Stream, 18bit RGB<br>H'2E: Loosely Packed Pixel Stream, 18bit RGB<br>H'3E: Packed Pixel Stream, 24bit RGB<br>Other Setting values are prohibited |
| 15 | TXESYNC | 0 | RW | Transmit End of Sync Pulse<br>0: HSE and VSE are NOT transmitted<br>1: HSE and VSE are transmitted<br>"0" setting is used for "Burst Mode" sequence or "Non-Burst Mode with Sync Events" sequence.<br>"1" setting is used for "Non-Burst Mode with Sync Pulse" sequence. |
| 14 to 4 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 3 to 0 | — | All 0 | RW | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |

### 34.3.1.50 Video-Input Channel 1 Vertical Size Set Register (VICH1VSSETR)




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 | — | 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 30 to 16 | VACTIVE | All 0 | RW | Set Line count of VACTIVE (Vertical Active lines) |
| 15 | VSPOL | 0 | RW | Polarity of VSYNC<br>Set polarity of VSYNC signal.<br>0: High Active<br>1: Low Active |
| 14 to 12 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 11 to 0 | VSA | All 0 | RW | VSA Count<br>Set Line count of VSA (Vertical Sync Active) |

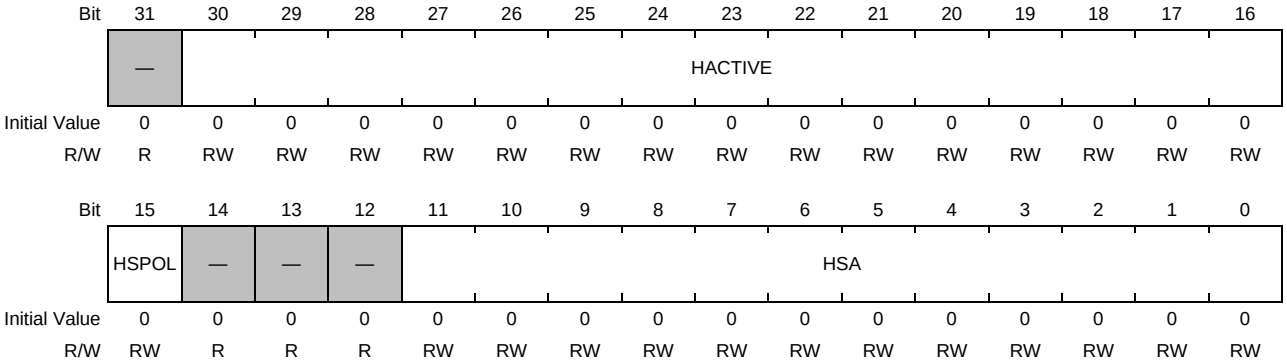
### 34.3.1.51 Video-Input Channel 1 Vertical Porch Set Register (VICH1VPSETR)

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | VFP | | | | | | | | | | | | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | VBP | | | | | | | | | | | | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 29 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 28 to 16 | VFP | All 0 | RW | VFP Count<br>Set Line count of VFP (Vertical Front Porch) |
| 15 to 13 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 12 to 0 | VBP | All 0 | RW | VBP Count<br>Set Line count of VBP (Vertical Back Porch) |

### 34.3.1.52 Video-Input Channel 1 Horizontal Size Set Register (VICH1HSSETR)




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 | — | 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 30 to 16 | HACTIVE | All 0 | RW | HACTIVE Count<br>Set pixel count of HACTIVE (Horizontal Active pixels) |
| 15 | HSPOL | 0 | RW | Polarity of HSYNC<br>Set polarity of HSYNC signal.<br>0: High Active<br>1: Low Active. |
| 14 to 12 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 11 to 0 | HSA | All 0 | RW | HSA Count<br>Set pixel count of HSA (Horizontal Sync Active) |

### 34.3.1.53 Video-Input Channel 1 Horizontal Porch Set Register (VICH1HPSETR)

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | HFP | | | | | | | | | | | | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | HBP | | | | | | | | | | | | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R | R | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 29 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 28 to 16 | HFP | All 0 | RW | HFP Count<br>Set pixel count of HFP (Horizontal Front Porch) |
| 15 to 13 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 12 to 0 | HBP | All 0 | RW | HBP Count<br>Set pixel count of HBP (Horizontal Back Porch) |

### 34.3.1.54 Sequence Channel x Set 0 Register (SQCHxSET0R)

"x" indicates channel number.

"x" is 0 or 1.

| 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 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R0W | R0W | R0W | R0W | R0W | R0W | R0W | R0W | RW | R | R | R | R | R | R | R |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | START |
| 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 | RW | RW | RW | RW | RW | R | R | R0W | R0W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 24 | — | All 0 | R0W | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 23 | — | 1 | RW | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 22 to 9 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 8 to 4 | — | All 0 | RW | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 3, 2 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 1 | — | 0 | R0W | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 0 | START | 0 | R0W | Sequence Operation Start<br>Write 1 to start running from Descriptor #0.<br>Set "1" during RUNNING=1 is prohibited.<br>Writing "0" has no effect. |

### 34.3.1.55 Sequence Channel x Set 1 Register (SQCHxSET1R)

“x” indicates channel number.

“x” is 0 or 1.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MAXDESNUM | | | | | | | | CHBUFSZ | | | | — | — | — | — |
| Initial Value | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | — | — | — | — | 0 | 0 | 0 | 0 |
| R/W | RW | RW | RW | RW | RW | RW | RW | RW | 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 24 | MAXDESNUM | H’08 | RW | Max Descriptor Number<br>This field indicate maximum number of descriptors. Writing value except the initial value is prohibited. (e.g. Setting 8 means descriptor can use #0 to #7) |
| 23 to 20 | CHBUFSZ | H’0 or H’3 | R | Sequence Channel x Buffer Size<br>H’0: 128B (x = 0 case)<br>H’3: 1kB (x = 1 case)<br>Setting transfer packet size larger than this field is prohibited. |
| 19 to 0 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |

### 34.3.1.56 Sequence Channel x Status Register (SQCHxSR)

"x" indicates channel number.

"x" is 0 or 1.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | RXAKE | RXCORERR | RXPKTDFAIL | RXFAIL | RXFATALERR | — | TXIBERR | — | — | — | — | PKTBIGERR | — | — | — |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | ADESFIN | — | — | — | AACTFIN | — | RUNNING | — | — |
| 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 | — | 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 30 | RXAKE | 0 | R | Receive Acknowledge and Error Report Packet<br>RXSR.RXAKE is also reported.<br>Write "1" to SQCHxSCR.RXAKE register to clear this field. |
| 29 | RXCORERR | 0 | R | Receive Correctable Error<br>Received packet has Correctable Error.<br>RXSR.ECCERR1B is also reported.<br>Write "1" to SQCHxSCR.RXCORERR register to clear this field. |
| 28 | RXPKTDFAIL | 0 | R | Receive Packet Data Fail<br>Packet Header Saved Correctly but Data did not save correctly.<br>One or more of the following bits are also reported.<br>RXSR.CRCERR, RXSR.WCERR, RXSR.UEXPKTERR, RXSR.MAXRPSZERR, RXSR.RXOVFERR or RXSR.IBERR.<br>Write "1" to SQCHxSCR.RXPKTDFAIL register to clear this field. |
| 27 | RXFAIL | 0 | R | Receive Fail<br>Expected receive did not done.<br>One or more of the following bits are also reported.<br>RXSR.PRESPTOERR, RXSR.ECCERR, RXSR.MLFERR or RXSR.NORETERR.<br>Write "1" to SQCHxSCR.RXFAIL register to clear this field. |
| 26 | RXFATALERR | 0 | R | Receive Fatal Error<br>Fatal timeout occurred during BTA.<br>One or more of the following bits are also reported.<br>FERRSR.TATO or FERRSR.LRXHTO.<br>Write "1" to SQCHxSCR.RXFATALERR register to clear this field. |
| 25 | — | 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 24 | TXIBERR | 0 | R | Tx Internal Bus Error<br>When this value is "1", the Internal Bus Read had failed response.<br>Write "1" to SQCHxSCR.TXIBERR register to clear this field. |
| 23 | — | 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 22 | — | 0 | R | Reserved<br>When read, the initial value is read. The written value will be ignored. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 21, 20 | — | 00 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 19 | PKTBIGERR | 0 | R | Packet is too big Error<br>Sequence operation channel 0:<br>Hard wired to 0.<br>Sequence operation channel 1:<br>This bit is asserted when length of sequence packet is bigger than Video-Input channel's BLLP.<br>Write "1" to SQCHxSCR.PKTBIGERR register to clear this field. |
| 18, 17 | — | 00 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 16, 15 | — | 00 | R | Reserved<br>When read, the initial value is read. The written value will be ignored. |
| 14 to 9 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 8 | ADESFIN | 0 | R | All Descriptor Finished by Setting<br>Descriptor finished by descriptor SQCHxDSCyAR.NXACT=H'0 setting.<br>Write "1" to SQCHxSCR.ADESFIN register to clear this field. |
| 7 to 5 | — | 000 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 4 | AACTFIN | 0 | R | Descriptor's All Action Finished<br>This bit is set if this descriptor's all action finished with SQCHxDSCyCR.FINACT[0] =1b.<br>Write "1" to SQCHxSCR.AACTFIN register to clear this field. |
| 3 | — | 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 2 | RUNNING | 0 | R | Sequence Operation Running<br>The value will be set to 1 during the sequence operation running.<br>Do not change descriptor values while this bit is 1.<br>Do not change the values in SQCHxSET0R and SQCHxSET1R. |
| 1, 0 | — | 00 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |

### 34.3.1.57 Sequence Channel x Status Clear Register (SQCHxSCR)

“x” indicates channel number.

“x” is 0 or 1.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | RXAKE | RXCORERR | RXPKTDFAIL | RXFAIL | RXFATALERR | — | TXIBERR | — | — | — | — | PKTBIGERR | — | — | — |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | R0W | R0W | R0W | R0W | R0W | R | R0W | R | R0W | R | R | R0W | R | R | R0W |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | ADESFIN | — | — | — | AACTFIN | — | — | — | — |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R0W | R | R | R | R | R | R | R0W | R | R | R | R0W | R | R | R | R |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 | — | 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 30 | RXAKE | 0 | R0W | SQCHxSR.RXAKE Clear<br>Set to “1” to clear SQCHxSR.RXAKE.<br>Writing “0” has no effect. |
| 29 | RXCORERR | 0 | R0W | SQCHxSR.RXCORERR Clear<br>Set to “1” to clear SQCHxSR.RXCORERR.<br>Writing “0” has no effect. |
| 28 | RXPKTDFAIL | 0 | R0W | SQCHxSR.RXPKTDFAIL Clear<br>Set to “1” to clear SQCHxSR.RXPKTDFAIL.<br>Writing “0” has no effect. |
| 27 | RXFAIL | 0 | R0W | SQCHxSR.RXFAIL Clear<br>Set to “1” to clear SQCHxSR.RXFAIL.<br>Writing “0” has no effect. |
| 26 | RXFATALERR | 0 | R0W | SQCHxSR.RXFATALERR Clear<br>Set to “1” to clear SQCHxSR.RXFATALERR.<br>Writing “0” has no effect. |
| 25 | — | 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 24 | TXIBERR | 0 | R0W | SQCHxSR.TXIBERR Clear<br>Set to “1” to clear SQCHxSR.TXIBERR.<br>Writing “0” has no effect. |
| 23 | — | 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 22 | — | 0 | R0W | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 21, 20 | — | 00 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 19 | PKTBIGERR | 0 | R0W | SQCHxSR.PKTBIGERR Clear<br>Set to “1” to clear SQCHxSR.PKTBIGERR.<br>Writing “0” has no effect. |
| 18, 17 | — | 00 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 16, 15 | — | 00 | R0W | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 14 to 9 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 8 | ADESFIN | 0 | R0W | SQCHxSR.ADESFIN Clear<br>Set to “1” to clear SQCHxSR.ADESFIN.<br>Writing “0” has no effect. |
| 7 to 5 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 4 | AACTFIN | 0 | R0W | SQCHxSR.AACTFIN Clear<br>Set to “1” to clear SQCHxSR.AACTFIN.<br>Writing “0” has no effect. |
| 3 to 0 | START | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |

### 34.3.1.58 Sequence Channel x Interrupt Enable Register (SQCHxIER)

"x" indicates channel number.

"x" is 0 or 1.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | RXAKE | RXCORERR | RXPKTDFAIL | RXFAIL | RXFATALERR | — | TXIBERR | — | — | — | — | PKTBIGERR | — | — | — |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | R | RW | RW | RW | RW | RW | R | RW | R | RW | R | R | RW | R | R | RW |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | ADESFIN | — | — | — | AACTFIN | — | — | — | — |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | RW | R | R | R | R | R | R | RW | R | R | R | RW | R | R | R | R |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 | — | 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 30 | RXAKE | 0 | RW | SInterrupt enable for the Receive Acknowledge and Error Report Packet<br>0: Disable interrupt of dsi_int_sqx when SQCHxSR.RXAKE=1<br>1: Enable interrupt of dsi_int_sqx when SQCHxSR.RXAKE=1 |
| 29 | RXCORERR | 0 | RW | Interrupt enable for the Receive Correctable Error<br>0: Disable interrupt of dsi_int_sqx when SQCHxSR.RXCORERR=1<br>1: Enable interrupt of dsi_int_sqx when SQCHxSR.RXCORERR=1 |
| 28 | RXPKTDFAIL | 0 | RW | Interrupt enable for the Receive Packet Data Fail<br>0: Disable interrupt of dsi_int_sqx when SQCHxSR.RXPKTDFAIL=1<br>1: Enable interrupt of dsi_int_sqx when SQCHxSR.RXPKTDFAIL=1 |
| 27 | RXFAIL | 0 | RW | Interrupt enable for the Receive Fail<br>0: Disable interrupt of dsi_int_sqx when SQCHxSR.RXFAIL=1<br>1: Enable interrupt of dsi_int_sqx when SQCHxSR.RXFAIL=1 |
| 26 | RXFATALERR | 0 | RW | Interrupt enable for the Receive Fatal Error<br>0: Disable interrupt of dsi_int_sqx when SQCHxSR.RXFATALERR=1<br>1: Enable interrupt of dsi_int_sqx when SQCHxSR.RXFATALERR=1 |
| 25 | — | 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 24 | TXIBERR | 0 | RW | Interrupt enable for the Tx Internal Bus Error<br>0: Disable interrupt of dsi_int_sqx when SQCHxSR.TXIBERR=1<br>1: Enable interrupt of dsi_int_sqx when SQCHxSR.TXIBERR=1 |
| 23 | — | 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 22 | — | 0 | RW | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 21, 20 | — | 00 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 19 | PKTBIGERR | 0 | RW | Interrupt enable for the Packet is too big Error<br>Sequence operation channel 0:<br>This field has no meaning.<br>Sequence operation channel 1:<br>0: Disable interrupt of dsi_int_sqx when SQCHxSR.PKTBIGERR=1<br>1: Enable interrupt of dsi_int_sqx when SQCHxSR.PKTBIGERR=1 |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 18, 17 | — | 00 | RW | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 16, 15 | — | 00 | R | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value. Operation is not guaranteed if a value other than the initial value is written. |
| 14 to 9 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 8 | ADESFIN | 0 | RW | Interrupt enable for the Descriptor Finished by Setting<br>0: Disable interrupt of dsi_int_sqx when SQCHxSR.ADESFIN=1<br>1: Enable interrupt of dsi_int_sqx when SQCHxSR.ADESFIN=1 |
| 7 to 5 | — | 000 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 4 | AACTFIN | 0 | RW | Interrupt enable for the Descriptor's All Action Finished<br>0: Disable interrupt of dsi_int_sqx when SQCHxSR.AACTFIN=1<br>1: Enable interrupt of dsi_int_sqx when SQCHxSR.AACTFIN=1 |
| 3 to 0 | START | 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |

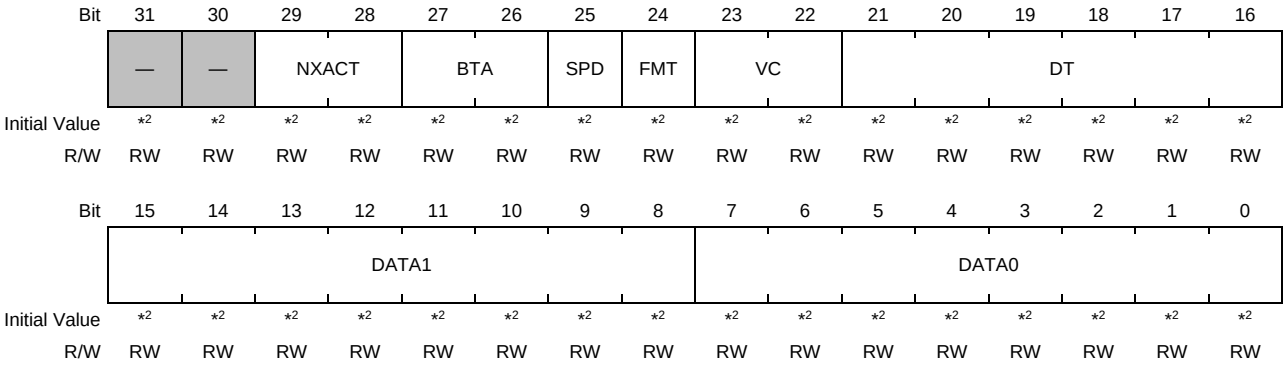
### 34.3.1.59 Sequence Channel x Descriptor y-A Register (SQCHxDSCyAR)

“x” indicates channel number.

“x” is 0 or 1.

“y” indicates descriptor number.

“y” is 0 to 7.




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 30 | — | *2 | RW | Reserved<br>When read, the last written value is read. When writing, be sure to write 0.<br>Operation is not guaranteed if a value other than 0 is written. |
| 29 to 28 | NXACT | *2 | RW | Next Action<br>H’0: Terminate Operation after this descriptor finished.<br>H’1: Operate next descriptor after this descriptor finished<br>H’2: Reserved<br>H’3: Reserved |
| 27 to 26 | BTA | *2 | RW | Bus Turn Around<br>Set Tx Request or no-operation with Bus Turn-Around (BTA) or not.<br>H’0: Tx Request without BTA or no-operation.<br>H’1: Tx non-Read Request with BTA.<br>H’2: Tx Read Request with BTA.<br>H’3: BTA only |
| 25 | SPD | *2 | RW | Speed<br>Set the speed type in Tx Request.<br>0: High Speed<br>1: Low Power<br>Prohibited to set this bit 1 during Video-Input Operation running. |
| 24 | FMT | *2 | RW | Format<br>Packet format of Tx Packet Header.<br>This field should be set by the data type.<br>0: Short Packet<br>1: Long Packet |
| 23 to 22 | VC | *2 | RW | Virtual Channel<br>Identifier of virtual channel of Tx Packet Header. |
| 21 to 16 | DT | *2 | RW | Data Type<br>Data type of Tx Packet Header. |
| 15 to 8 | DATA1 | *2 | RW | Data 1<br>Data1 of Tx Packet Header.*1<br>When Tx Packet format is Long Packet, this value is upper 8 bits of the word count. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 to 0 | DATA0 | *2 | RW | Data 0<br>Data0 of Tx Packet Header.*1<br>When Tx Packet format is Long Packet, this value is lower 8 bits of the word count. |

Note 1. • The maximum size for Command Transfer mode Long Packet by using Register is 16 Byte.
(SQCHxDSCyBR.DTSEL = H'0 and SQCHxDSCyAR.FMT=1)
• The maximum size for Command Transfer mode Long Packet by using internal bus memory is SQCHxSET1R.CHBUFSZ Byte.
(SQCHxDSCyBR.DTSEL = H'1 and SQCHxDSCyAR.FMT=1)

Note 2. Initial Values are determined by initial value of Descriptor RAM.

### 34.3.1.60 Sequence Channel x Descriptor y-B Register (SQCHxDSCyBR)

“x” indicates channel number.

“x” is 0 or 1.

“y” indicates descriptor number.

“y” is 0 to 7.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | DTSEL | | — | — | — | — | — | — | — | — |
| Initial Value | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 | *1 |
| R/W | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW |

| 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 | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW | RW |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 26 | — | *1 | RW | Reserved<br>When read, the last written value is read. When writing, be sure to write 0. Operation is not guaranteed if a value other than 0 is written. |
| 25 to 24 | DTSEL | *1 | RW | Data Select<br>H’0: Packet Payload Data register<br>H’1: Long Packet data use memory space.<br>In the case of write. Internal Read will be cause to SQCHxDSCyDR address.<br>In the case of Read. Internal Write will be cause to SQCHxDSCyDR address.<br>*Note*: Read Header always saved to RXRSSxR register.<br>H’2: Reserved<br>H’3: Reserved<br>If write request with short Packet, then set H’0.<br>When DTSEL=H’0, Long Packet write request data size shall be same or lower than 16Byte.When read request data to save has over 16 Byte word count, it reports as error. |
| 23, 22 | — | *1 | RW | Reserved<br>When read, the last written value is read. When writing, be sure to write 0. Operation is not guaranteed if a value other than 0 is written. |
| 21, 20 | — | *1 | RW | Reserved<br>When read, the last written value is read. When writing, be sure to write 0. Operation is not guaranteed if a value other than 0 is written. |
| 19 to 16 | — | *1 | RW | Reserved<br>When read, the last written value is read. When writing, be sure to write 0. Operation is not guaranteed if a value other than 0 is written.. |
| 15 to 12 | — | *1 | RW | Reserved<br>When read, the last written value is read. When writing, be sure to write 0. Operation is not guaranteed if a value other than 0 is written. |
| 11 to 0 | — | *1 | RW | Reserved<br>When read, the last written value is read. When writing, be sure to write 0. Operation is not guaranteed if a value other than 0 is written. |

Note 1. Initial Values are determined by initial value of Descriptor RAM.

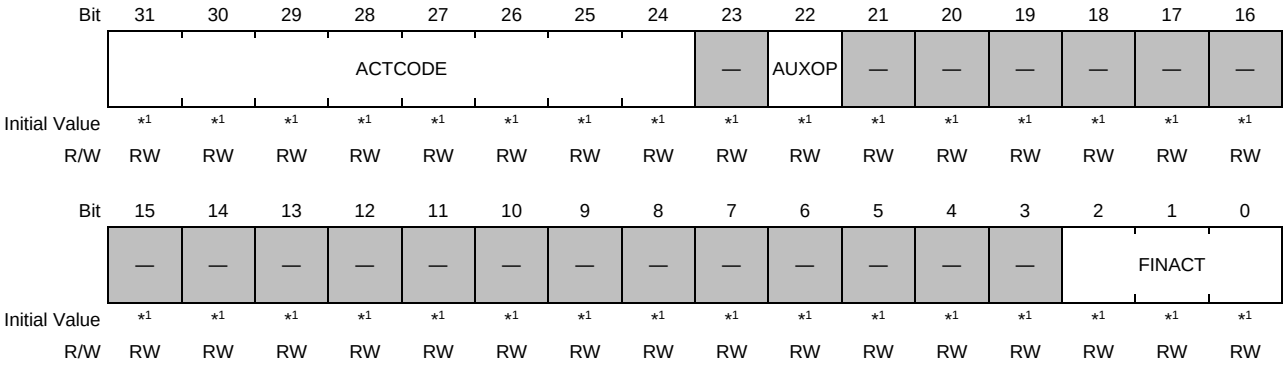
### 34.3.1.61 Sequence Channel x Descriptor y-C Register (SQCHxDSCyCR)

“x” indicates channel number.

“x” is 0 or 1.

“y” indicates descriptor number.

“y” is 0 to 7.




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 24 | ACTCODE | *1 | RW | Action Code<br>When BTA action, indicate rx result save slot number.<br>Setting value equal or more than 4 is prohibited.<br>Setting same rx result save slot number value to different descriptor is prohibited.<br>When AUXOP is 1.<br>H’00: Send Reset-Trigger.<br>H’01: Reserved<br>H’02: Reserved<br>H’03: Reserved<br>H’08: No-operation<br>Other:<br>Set H’00 in other case. |
| 23 | — | *1 | RW | Reserved<br>When read, the last written value is read. When writing, be sure to write 0. Operation is not guaranteed if a value other than 0 is written. |
| 22 | AUXOP | *1 | RW | Auxiliary Operation<br>Use for auxiliary operation execution.<br>When this bit is set to 1, SQCHxDSCyBR.BTA should be set to H’0.<br>Set this bit 1 with BTA=H’0.<br>Prohibited to set this bit 1 during Video-Input Operation running. |
| 21 to 3 | — | *1 | RW | Reserved<br>When read, the last written value is read. When writing, be sure to write 0. Operation is not guaranteed if a value other than 0 is written. |
| 2 to 0 | FINACT | *1 | RW | Finish Action<br>bit 0: set SQCHxSR.AACTFIN if this descriptor’s all action finished. (0b: Disable, 1b: Enable).<br>Since each descriptor’s operation is not same, descriptors may finish out-of-order.<br>For example, Descriptor#1 may be finished before Descriptor#0. |

Note 1. Initial Values are determined by initial value of Descriptor RAM.

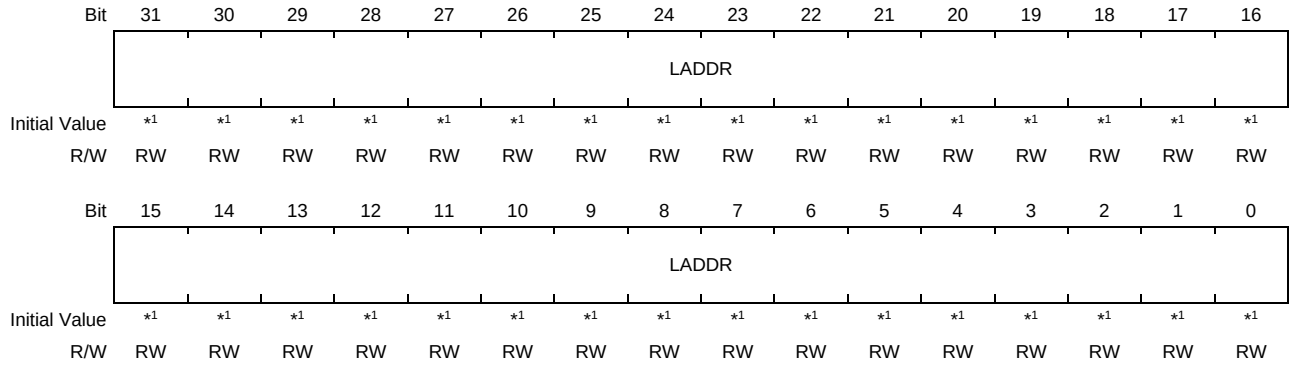
### 34.3.1.62 Sequence Channel x Descriptor y-D Register (SQCHxDSCyDR)

“x” indicates channel number.

“x” is 0 or 1.

“y” indicates descriptor number.

“y” is 0 to 7.




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 0 | LADDR | *1 | RW | Lower Address<br>Set Lower 32 bits address for long packet Payload Data in Sequence Operation channel x Descriptor y.<br>When Sequence Operation Data Select is “use memory space”<br>(SQCHxDSCyBR.DTSEL =H’1) and Sequence Operation Format is Long packet (SQCHxDSCyAR.FMT=1b), the DSI-Tx Module accesses the address by internal bus Master Read in Write action and internal bus Write in Read action.<br>Set lower three bits to “000b”. |

Note 1. Initial Values are determined by initial value of Descriptor RAM.

## 34.3.2 Resisters in D-PHY

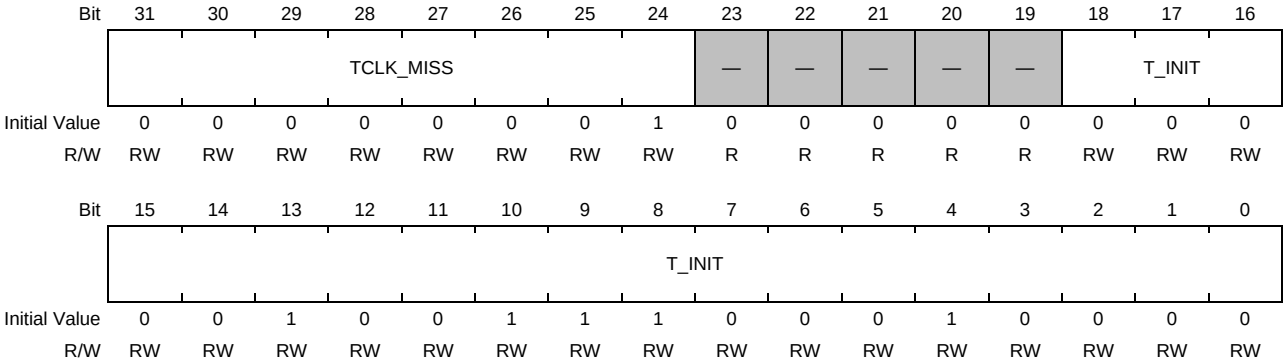
### 34.3.2.1 DSI D-PHY Control Register 0 (DSIDPHYCTRL0)

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CAL_EN_HSRX_OFS |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | CMN_MASTER_EN | — | — | — | — | — | RE_VDD_DETVCCQLV18 | EN_LDO1200 | EN_BGR |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| R/W | R | R | R | R | R | R | R | R/W | R | R | R | R | R | R/W | R/W | R/W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 17 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 16 | CAL_EN_HSRX_OFS | 1 | RW | HS-RX Offset Calibration enable<br>0b: HS-RX Start with Offset Calibration.<br>1b: HS-RX Start without Offset Calibration. |
| 15 to 9 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 8 | CMN_MASTER_EN | 1 | RW | Select D-PHY mode master or slave of All-lane.<br>1b: Master side DSI-v1.3.1 (D-PHY v2.1 CIL_MCNN,CIL_MFAA)<br>0b: Slave side CSI2-v1.3 (D-PHY v2.0 CIL_SCNN,CIL_SFEN) |
| 7 to 3 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 2 | RE_VDD_DETVCCQLV18 | 1 | RW | When VCCQLV18 is OFF, fix the terminal on the interface of the different power supply to Low.<br>1b: VCCQLV18 ON<br>0b: VCCQLV18 OFF |
| 1 | EN_LDO1200 | 0 | RW | LDO1200 Enable<br>1b: Enable<br>0b: Disable |
| 0 | EN_BGR | 0 | RW | BGR Enable<br>1b: Enable<br>0b: Disable |

### 34.3.2.2 DSI D-PHY Timing Register 0 (DSIDPHYTIM0)




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 24 | TCLK_MISS | 1 | RW | Timing parameter<br>Set 0. |
| 23 to 19 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 18 to 0 | T_INIT | 10000 | RW | Timing parameter*1 |

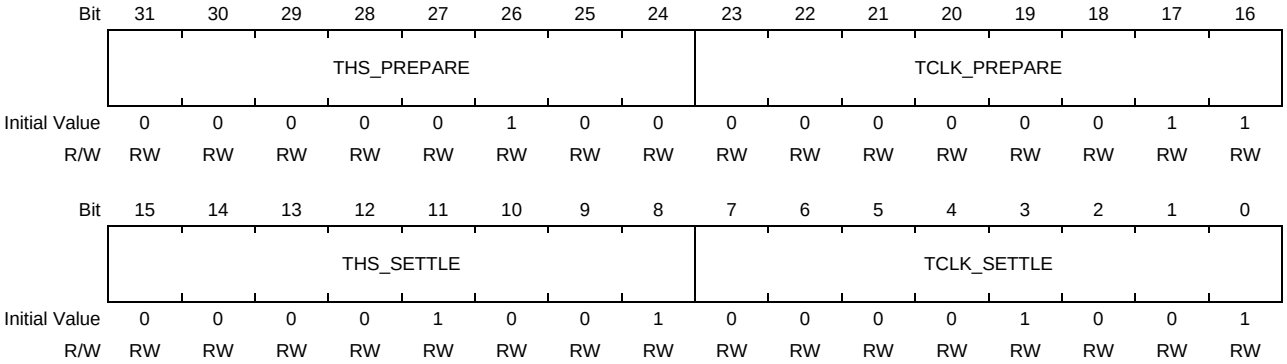
Note 1. **Table 34.6** lists the recommended setting values.

Table 34.6 Recommended Setting Values of the DSI D-PHY Timing Register

| No. | Pin Name | Transmission Rate | | | | | | Note |
|---|---|---|---|---|---|---|---|---|
| | | 80 Mbps | 125 Mbps | 250 Mbps | 360 Mbps | 720 Mbps | Over 720 Mbps | |
| 1 | T_INIT[18:0] | 19'd79801 (H'137B9) | 19'd79801 (H'137B9) | 19'd79801 (H'137B9) | 19'd79801 (H'137B9) | 19'd79801 (H'137B9) | 19'd79801 (H'137B9) | Min: 100 µs<br>Recommend: 600 µs |
| 2 | TCLK_PREPARE[7:0] | 8'd8 (H'8) | 8'd8 (H'8) | 8'd8 (H'8) | 8'd8 (H'8) | 8'd8 (H'8) | 8'd8 (H'8) | |
| 3 | THS_PREPARE[7:0] | 8'd13 (H'D) | 8'd12 (H'C) | 8'd12 (H'C) | 8'd10 (H'A) | 8'd9 (H'9) | 8'd9 (H'9) | |
| 4 | TCLK_ZERO[7:0] | 8'd33 (H'21) | 8'd33 (H'21) | 8'd33 (H'21) | 8'd33 (H'21) | 8'd33 (H'21) | 8'd33 (H'21) | |
| 5 | TCLK_PRE[7:0] | 8'd24 (H'18) | 8'd15 (H'F) | 8'd13 (H'D) | 8'd4 (H'4) | 8'd4 (H'4) | 8'd4 (H'4) | |
| 6 | TCLK_POST[7:0] | 8'd94 (H'5E) | 8'd94 (H'5E) | 8'd94 (H'5E) | 8'd35 (H'23) | 8'd35 (H'23) | 8'd35 (H'23) | |
| 7 | TCLK_TRAIL[7:0] | 8'd10 (H'A) | 8'd10 (H'A) | 8'd10 (H'A) | 8'd7 (H'7) | 8'd7 (H'7) | 8'd7 (H'7) | |
| 8 | THS_ZERO[7:0] | 8'd23 (H'17) | 8'd23 (H'17) | 8'd23 (H'17) | 8'd16 (H'10) | 8'd16 (H'10) | 8'd16 (H'10) | |
| 9 | THS_TRAIL[7:0] | 8'd17 (H'11) | 8'd17 (H'11) | 8'd16 (H'10) | 8'd9 (H'9) | 8'd9 (H'9) | 8'd9 (H'9) | |
| 10 | THS_EXIT[7:0] | 8'd13 (H'D) | 8'd13 (H'D) | 8'd13 (H'D) | 8'd13 (H'D) | 8'd13 (H'D) | 8'd13 (H'D) | |
| 11 | TLPX[7:0] | 8'd6 (H'6) | 8'd6 (H'6) | 8'd6 (H'6) | 8'd6 (H'6) | 8'd6 (H'6) | 8'd6 (H'6) | |

**Note:** The above table lists the recommended setting values. Please calculate and use linearly interpolated value for the setting of non-described in the table.

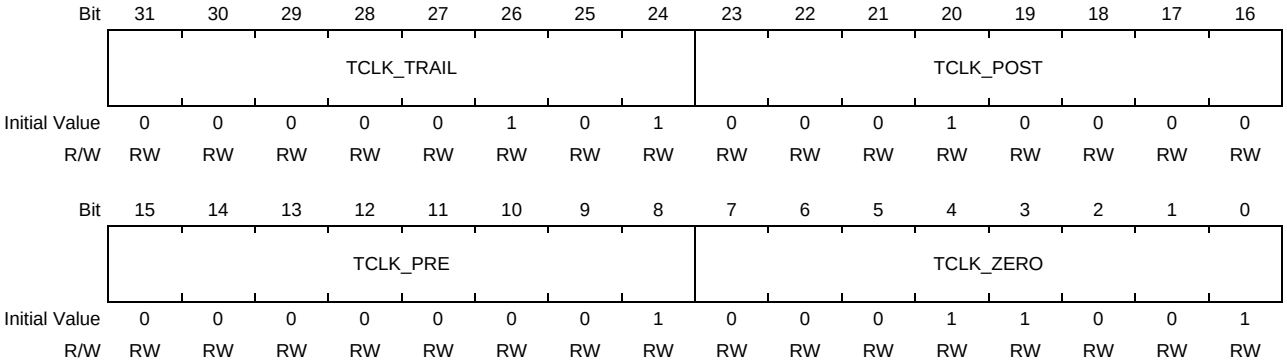
### 34.3.2.3 DSI D-PHY Timing Register 1 (DSIDPHYTIM1)




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 24 | THS_PREPARE | 4 | RW | Timing parameter*1 |
| 23 to 16 | TCLK_PREPARE | 3 | RW | Timing parameter*1 |
| 16 to 8 | THS_SETTLE | 9 | RW | Timing parameter<br>Set 0. |
| 7 to 0 | TCLK_SETTLE | 9 | RW | Timing parameter<br>Set 0. |

Note 1. **Table 34.6** lists the recommended setting values.

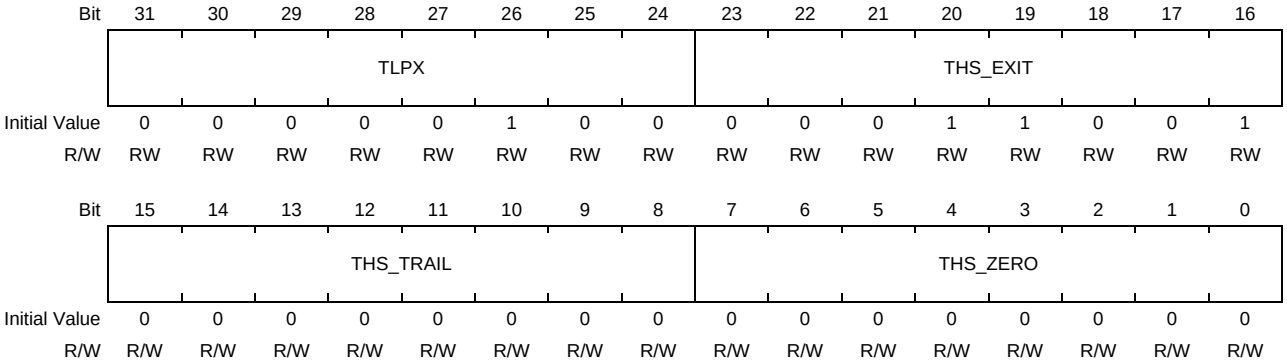
### 34.3.2.4 DSI D-PHY Timing Register 2 (DSIDPHYTIM2)




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 24 | TCLK_TRAIL | 5 | RW | Timing parameter*1 |
| 23 to 16 | TCLK_POST | 16 | RW | Timing parameter*1 |
| 15 to 8 | TCLK_PRE | 1 | RW | Timing parameter*1 |
| 7 to 0 | TCLK_ZERO | 25 | RW | Timing parameter*1 |

Note 1. **Table 34.6** lists the recommended setting values.

### 34.3.2.5 DSI D-PHY Timing Register 3 (DSIDPHYTIM3)




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 24 | TLPX | 4 | RW | Timing parameter*1 |
| 23 to 16 | THS_EXIT | 9 | RW | Timing parameter*1 |
| 15 to 8 | THS_TRAIL | 6 | RW | Timing parameter*1 |
| 7 to 0 | THS_ZERO | 11 | RW | Timing parameter*1 |

Note 1. **Table 34.6** lists the recommended setting values.

# 34.4 Operation

## 34.4.1 Interrupts

Interrupts can be divided into the following categories:

Table 34.7 Interrupt Signals (1/2)

| Interrupt Signal | Category | Interrupt Condition | Interrupt Source Register |
|---|---|---|---|
| dsi_int_sq0 | Sequence operation channel 0 interrupt | Receive Acknowledge and Error Report Packet in Sequence Operation BTA | SQCH0SR.RXAKE |
| | | Receive Correctable Error in Sequence Operation BTA | SQCH0SR.RXCORERR |
| | | Receive Packet Data Error in Sequence Operation BTA | SQCH0SR.RXPKTDFAIL |
| | | Receive Fail Error in Sequence Operation BTA | SQCH0SR.RXFAIL |
| | | Receive Fatal Error in Sequence Operation BTA | SQCH0SR.RXFATALERR |
| | | Transmit Internal Bus Error in Sequence Operation | SQCH0SR.TXIBERR |
| | | Descriptor Finished by Setting in Sequence Operation | SQCH0SR.ADESFIN |
| | | Descriptor's All Action Finished in Sequence Operation | SQCH0SR.AACTFIN |
| dsi_int_sq1 | Sequence operation channel 1 interrupt | Receive Acknowledge and Error Report Packet in Sequence Operation BTA | SQCH1SR.RXAKE |
| | | Receive Correctable Error in Sequence Operation BTA | SQCH1SR.RXCORERR |
| | | Receive Packet Data Error in Sequence Operation BTA | SQCH1SR.RXPKTDFAIL |
| | | Receive Fail Error in Sequence Operation BTA | SQCH1SR.RXFAIL |
| | | Receive Fatal Error in Sequence Operation BTA | SQCH1SR.RXFATALERR |
| | | Transmit Internal Bus Error in Sequence Operation | SQCH1SR.TXIBERR |
| | | Transmit Packet is too big Error in Sequence Operation | SQCH1SR.PKTBIGERR |
| | | Descriptor Finished by Setting in Sequence Operation | SQCH1SR.ADESFIN |
| | | Descriptor's All Action Finished in Sequence Operation | SQCH1SR.AACTFIN |
| dsi_int_vin1 | Video-Input operation channel 1 interrupt | Transmit Buffer Overflow in Video-Input Operation | VICH1SR.VBUFOVF |
| | | Transmit Buffer Underflow in Video-Input Operation | VICH1SR.VBUFUDF |
| | | Transmit Timing Error in Video-Input Operation | VICH1SR.TIMERR |
| | | Video-Input Signal Accept Ready | VICH1SR.VIRDY |
| | | Video-Input Operation Stopped | VICH1SR.STOP |
| | | Video-Input Operation Started | VICH1SR.START |

Table 34.7 Interrupt Signals (2/2)

| Interrupt Signal | Category | Interrupt Condition | Interrupt Source Register |
|---|---|---|---|
| dsi_int_rcv | DSI Packet Receive interrupt | Receive Acknowledge and Error Report Packet | RXSR.RXAKE |
| | | ECC 1bit Error | RXSR.ECCERR1B |
| | | Maximum Return Packet Size Error | RXSR.MAXRPSZERR |
| | | No Return Error | RXSR.NORETERR |
| | | Peripheral Response timeout | RXSR.PRESPTOERR |
| | | Receive Buffer overflow error | RXSR.RXOVFERR |
| | | Internal Bus Error detect on data save | RXSR.IBERR |
| | | CRC Error | RXSR.CRCERR |
| | | Word Count Error | RXSR.WCERR |
| | | Unexpected Packet Error | RXSR.UEXPKTERR |
| | | ECC Error | RXSR.ECCERR |
| | | Malformed Error | RXSR.MLFERR |
| | | External Tearing Effect | RXSR.EXTTEDET |
| | | Receive ACK trigger | RXSR.RXACK |
| | | Receive Tearing Effect trigger | RXSR.RXTE |
| | | Receive Reset trigger | RXSR.RXRTRG |
| | | Receive Unknown-5 trigger | RXSR.RXUK5TRG |
| | | Receive EoTp | RXSR.RXEOTP |
| | | Receive Response packet | RXSR.RXRESP |
| | | Turnaround Acknowledge Timeout | RXSR.TATO |
| | | LP-RX Host Processor Timeout | RXSR.LRXHTO |
| | | BTA request End | RXSR.BTAREQEND |
| dsi_int_ferr | DSI Fatal Error interrupt | Contention Error LP1 | FERRSR.ERRCLP1 |
| | | Contention Error LP0 | FERRSR.ERRCLP0 |
| | | Control Error | FERRSR.ERRCTRL |
| | | Sync Escape Error | FERRSR.ERRSYNESC |
| | | Error Escape | FERRSR.ERRESC |
| | | Turnaround Acknowledge Timeout | FERRSR.TATO |
| | | LP-RX Host Processor Timeout | FERRSR.LRXHTO |
| | | HS TX Timeout | FERRSR.HTXTO |
| dsi_int_ppi | DSI D-PHY PPI interrupt | Data Lanes return from ULPS | PLSR.DLFROMULPS |
| | | Data Lanes go to ULPS | PLSR.DLTOULPS |
| | | Clock Lane change HS to LP | PLSR.CLHS2LP |
| | | Clock Lane change LP to HS | PLSR.CLLP2HS |
| | | Clock Lanes return from ULPS | PLSR.CLFROMULPS |
| | | Clock Lanes go to ULPS | PLSR.CLTOULPS |
| | | Data Lane 0 change TX to RX | PLSR.DL0TX2RX |
| | | Data Lane 0 change RX to TX | PLSR.DL0RX2TX |

The interrupt signal consists of combination logic. The interrupt signal from a Status Register is enabled by the signal related Interrupt Enable Register. These interrupt signals are bound by OR logic.

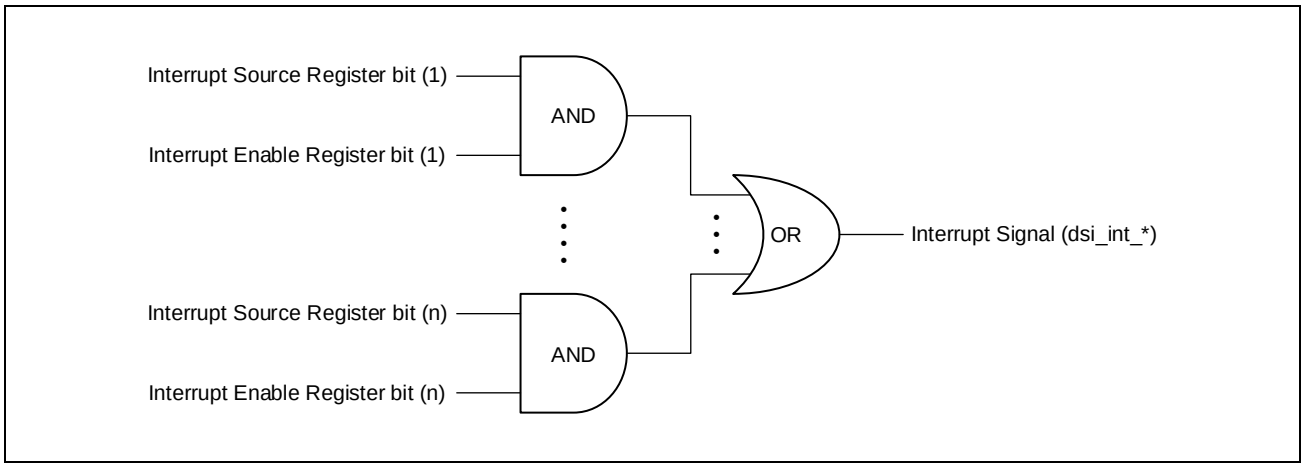



Figure 34.4 Interrupt signal output

## 34.4.2 Operation Flow

This chapter shows various operation procedures for using the DSI-Tx Module. As for the interrupt operation, Interrupt Enable bit of each interrupt source is enabled (set to 1) before use, and after confirming the interrupt, the interrupt source is cleared. And more, previous action's interrupt source must be cleared before setting the interrupt enable. Flows in this chapter indicate only interrupt bit. The DSI-Tx Module does not care whether firmware use the bits by interrupt action or flag polling action. These descriptions are omitted in each flow in this chapter unless otherwise specified. Error checks are also omitted in normal flow.

### 34.4.2.1 Reset

#### (1) Power on Reset and Initial Settings for All Operations

The power on sequence of the DSI-Tx Module is shown **Figure 34.5**.

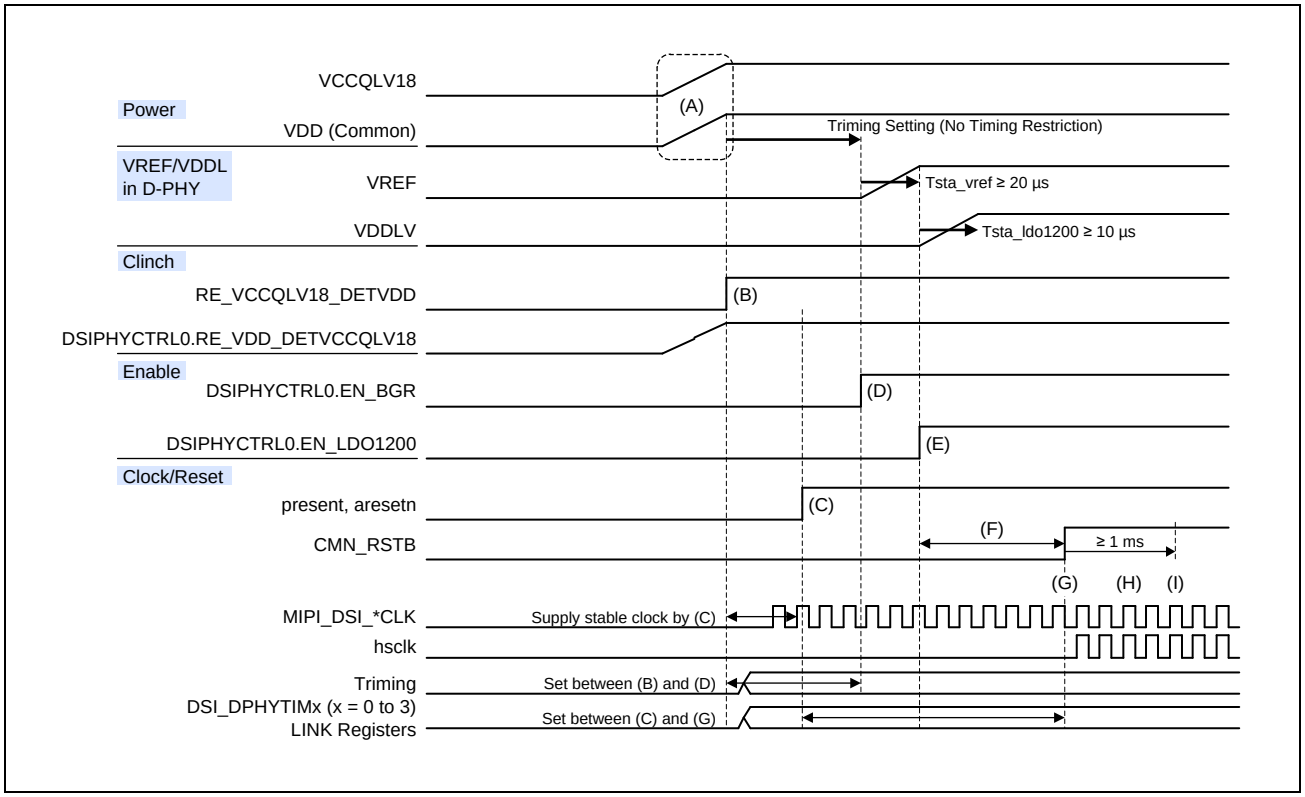



Figure 34.5 Power on sequence

(A) Bring up VDD and VCCQLV18. There are no restrictions on the order to bring up VDD and VCCQLV18.

(B) Set 1 to RE_VCCQLV18_DETVDD.

(C) De-assert preestn and aresetn after after MIPI_DSI_*CLK is stable.

(D) Write 1 to DSIDPHYCTRL0.EN_BGR.

(E) Write 1 to DSIDPHYCTRL0.EN_LDO1200 after waiting for more than 20 usec.

(F) Write DSIDPHYTIMx (x=0 to 3) registers and LINK registers after waiting for more than 10 usec.

(G) De-assert CMN_RSTB.

(H) Wait for more than 1 msec.

(I) The DSI-Tx Module is ready.

At power-on reset, all registers of the DSI-Tx Module are initialized.

Perform the initial settings of the register after canceling the power-on reset. It is not necessary to set the register that is used with the default value.

The registers that need to be initialized are as follows.

- Common Settings
  - TXSETR
  - ULPSSETR
  - DSISETR
  - CLSTPTSETR
  - LPTRNSTSETR
- Timeout Settings
  - PRESPTOBTASETR
  - PRESPTOLPSETR
  - PRESPTOHSSETR
  - HSTXTOSETR
  - LRXHTOSETR
  - TATOSETR

These registers can only be changed during the initial setup after a power-on reset and during a software reset in **Section 34.4.2.1(2)**. It is prohibited to change at any other time

### (2) Software Reset

**Figure 34.6** shows the operation during software reset. Make sure that APB clock (MIPI_DSI_PCLK), Internal Bus clock (MIPI_DSI_ACLK), Video clock (MIPI_DSI_VCLK), PLL Multiplied clock (MIPI_DSI_PLLCLK), Escape mode Transmit clock (MIPI_DSI_LPCLK) are running and CMN_RSTB is High before starting software reset.

Writing 1 to RSTCR.SWRST will issue a software reset. After issuing the software reset, check the start of the software reset process with RSTSR.SWRSTxx = 1. Then write 1 to RSTCR.FCETXSTP to transition the valid data lane of the D-PHY to the Tx Stop State. By checking RSTSR.DL0DIR = 0 and RSTSR.DLSTPST [x] = 1 of the valid data lane, it is possible to confirm that the valid data lane of D-PHY has transitioned to Tx Stop State. After confirming that the valid data lane of the D-PHY has transitioned to the Tx Stop State, write 0 to RSTCR.FCETXSTP. Then write 0 to RSTCR.SWRST and check the completion of the software reset process with RSTSR.SWRSTx = 0. Since all registers are synchronously reset except for some registers*1, the software reset process is not completed even if a software reset is issued while the clock is stopped.

**Note 1.** Registers that require initial settings as described in **Section 34.4.2.1(1)**, GPO0R register, GPO1R register, RSTCR register, RSTSR register, and SQCHxDSCyAR – SQCHxDSCyDR register are not initialized by software reset.

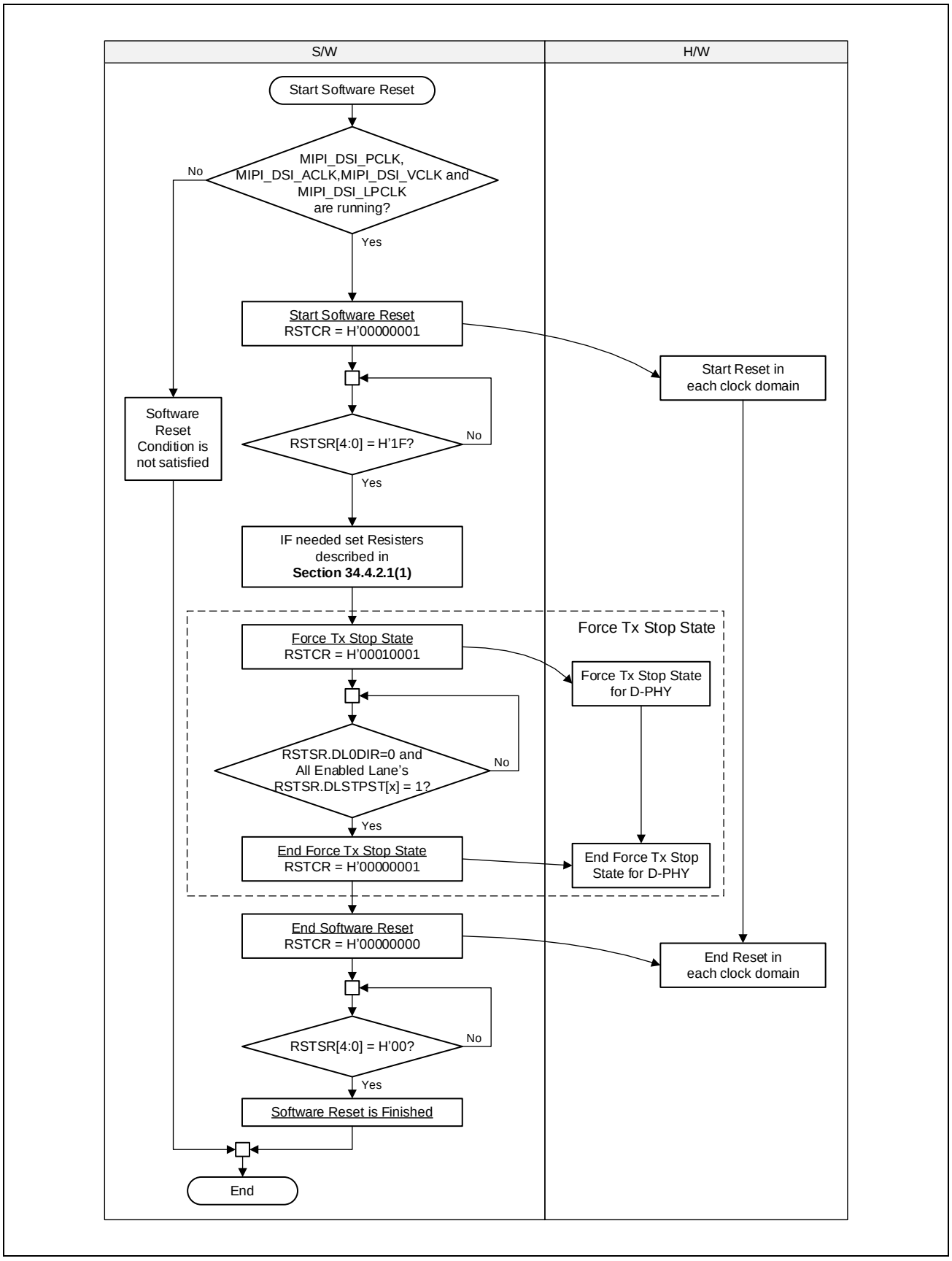



Figure 34.6 Software Reset

#### 34.4.2.2 Start/Stop of HS clock

The following describes the HS Clock Start / Stop sequence. HS Clock Start / Stop is controlled by Software.

**Figure 34.7** shows the operation at HS Clock Start.

Write 1 to TXSETR.CLEN to enable DPHY clock lanes before starting Video-Input Operation or Sequence Operation channel 1 operation.

After the clock lane is enabled and the source clock of the clock lane is stable, write 1 to HSCLKSETR.HSCLKRUN to set the clock lane to HS. It is forbidden to change HSCLKRUN while Sequence Channel is in operation.

At the same time as writing 1 to HSCLKSETR.HSCLKRUN, set continuous clock mode and non-continuous clock mode with HSCLKSETR.HSCLKMODE.

In continuous clock mode, always set the clock lane to HS. PLSR.CLLP2HS is set to 1 when the clock lane transitions from LP to HS. After checking, perform the following operations.

In non-continuous clock mode, the DSI-Tx Module automatically transitions the clock lane from LP to HS depending on the presence or absence of HS Transmission. PLSR.CLLP2HS and PLSR.CLHS2LP are set for each transition, but confirmation is not required.

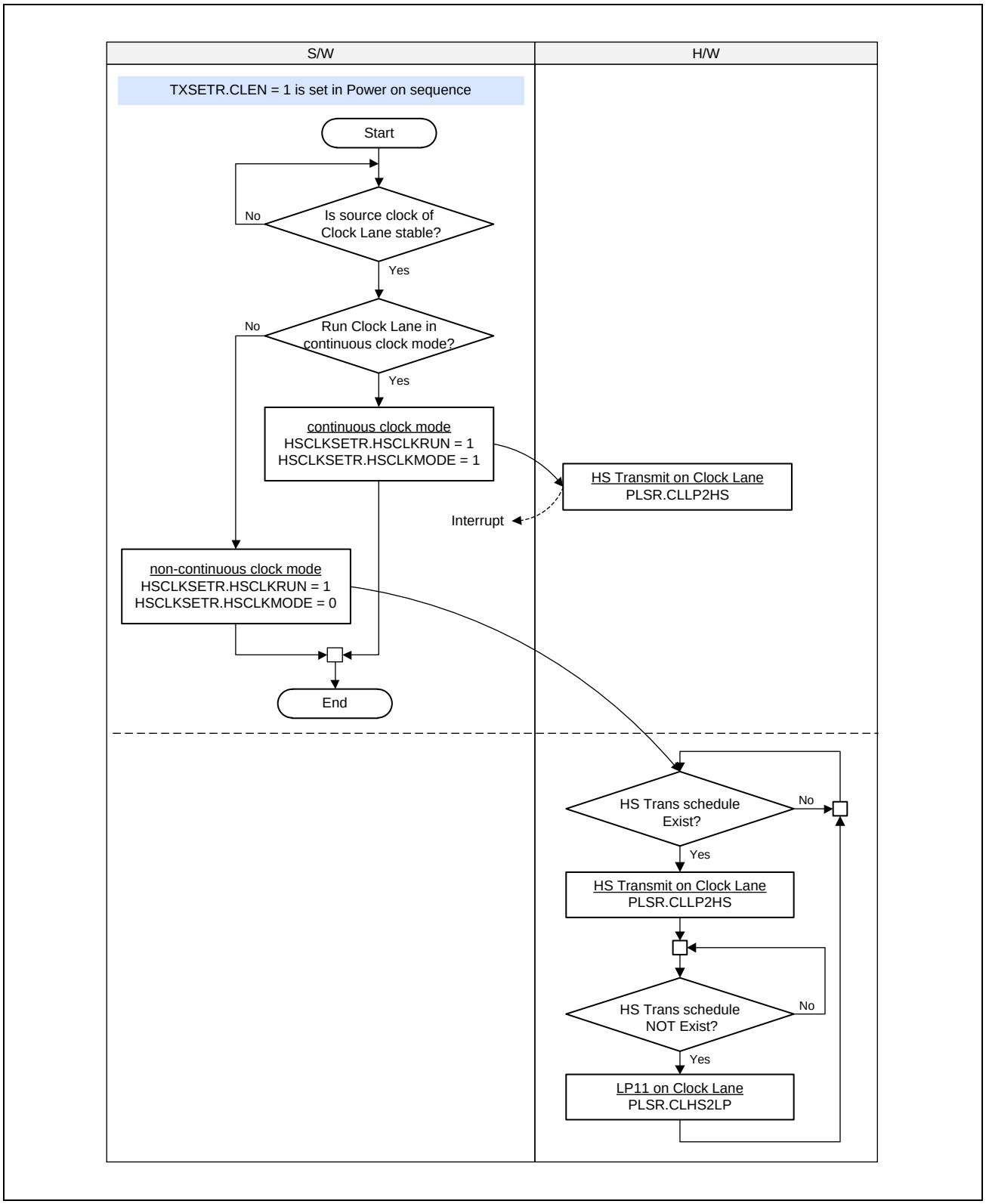



Figure 34.7 Start of HS clock

**Figure 34.8** shows the operation when HS Clock Stop. After confirming that Video-Input Operation and Sequence Operation channel 1 are not running, write 0 to HSCLKSETR.HSCLKRUN to make the clock lane LP. PLSR.CLHS2LP is set to 1 when the clock lane transitions from HS to LP. In non-continuous clock mode, you do not need to check PLSR.CLHS2LP because the clock lane is likely to be LP already when you write 0 to HSCLKSETR.HSCLKRUN. It is possible to check the clock lane status by reading PLSR.CLSTPST.

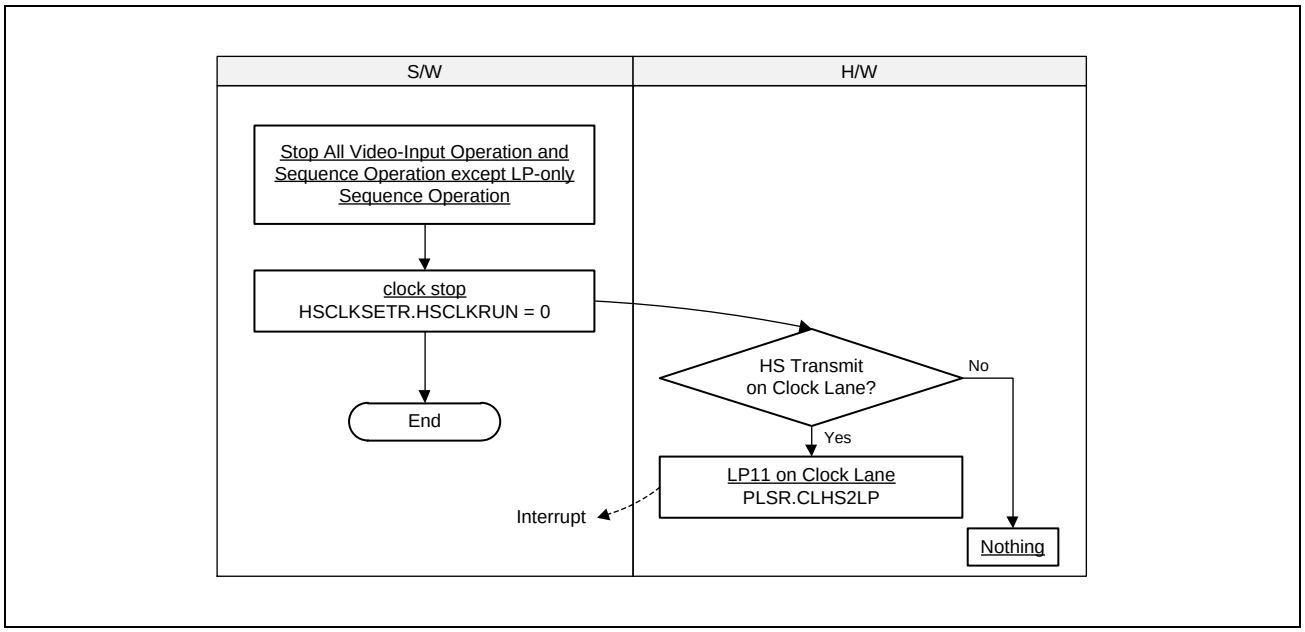



Figure 34.8 Stop of HS clock

#### 34.4.2.3 Sequence Operation

##### (1) Basic Running Sequence

Sequence Operation requires initial settings before the operation starts.

- Channel Register Setting
  - Set SQCHxSET1R, SQCHxIER
- Prepare Write Data
  - Prepare Write Data to Memory Area
  - Prepare Write Data to TXPPDxR Registers
- Allocate Read Data Area
  - Clear RXRSSR Register if previous valid flag remains.
  - Allocate each read action's Memory area.
- Write all Descriptors
- Confirm running HSCLK if Sequence channel 1 is targeted to run.

When "1" is written to SQCHxSET0R.START, Sequence Operation channel x starts the processing of Descriptor, and when the processing of Descriptor of SQCHxDSCyAR.NXACT = "00b" is completed, the processing of Descriptor ends.

Since Descriptor consists of SRAM, the initial value is indefinite. When the user sets the Descriptor, set all the bits.

After writing 1 to SQCHxSET0R.START on Sequence Operation channel x and SQCHxSR.RUNNING becomes 1, It is prohibited to change the settings of SQCHxSET1R, SQCHxDSCyAR, SQCHxDSCyBR, SQCHxDSCyCR, and SQCHxDSCyDR until SQCHxSR.RUNNING becomes 0.

HS-related operations are prohibited on Sequence Operation channel 0.

##### (2) Single Packet Transmission

The following describes sending non-read packets using the Sequence Operation channel.

For non-read packet transmission, set the following registers before packet transmission.

SQCHxDSCyAR.NXACT: "00b"

SQCHxDSCyAR.FMT: "0b" Short Packet or "1b" Long Packet

SQCHxDSCyAR.SPD: "0b" High Speed or "1b" Low Speed

SQCHxDSCyAR.BTA: "00b" without BTA or "01b" with BTA

SQCHxDSCyAR.VC, DT, DATA0, DATA1

SQCHxDSCyBR.DTSEL: Specify payload data storage location for non-read packets

"00b" TXPPD0R – TXPPD3R. Payloads size is limited up to 16 Bytes.

"01b" Memory space. Payloads size is limited up to SQCHxSET1R.CHBUFSZ Bytes.

SQCHxDSCyCR.AUXOP "0b"

When setting a Long Packet with SQCHxDSCyAR.FMT, prepare the payload data in the location specified by SQCHxDSCyBR.DTSEL in the form of little endian.

No payload data is required when configuring Short Packets with SQCHxDSCyAR.FMT.

After the above settings, write “1” to SQCHxSET0R.START to send non-read packets.

The user can know the completion of Single Packet Transmission by asserting SQCHxSR.ADESFIN.

If “01b” is set in SQCHxDSCyAR.BTA, BTA is performed after sending a non-read packet.

An ACK trigger or Acknowledge and Error Report Packet may be received when the bus rights are transferred to the peripheral.

The user can get an overview of the reception result by reading the RXRSSxR register with the number specified by SQCHxDSCyCR.ACTCODE. Also, by reading the RXSR register, the user can know the details of the reception result.

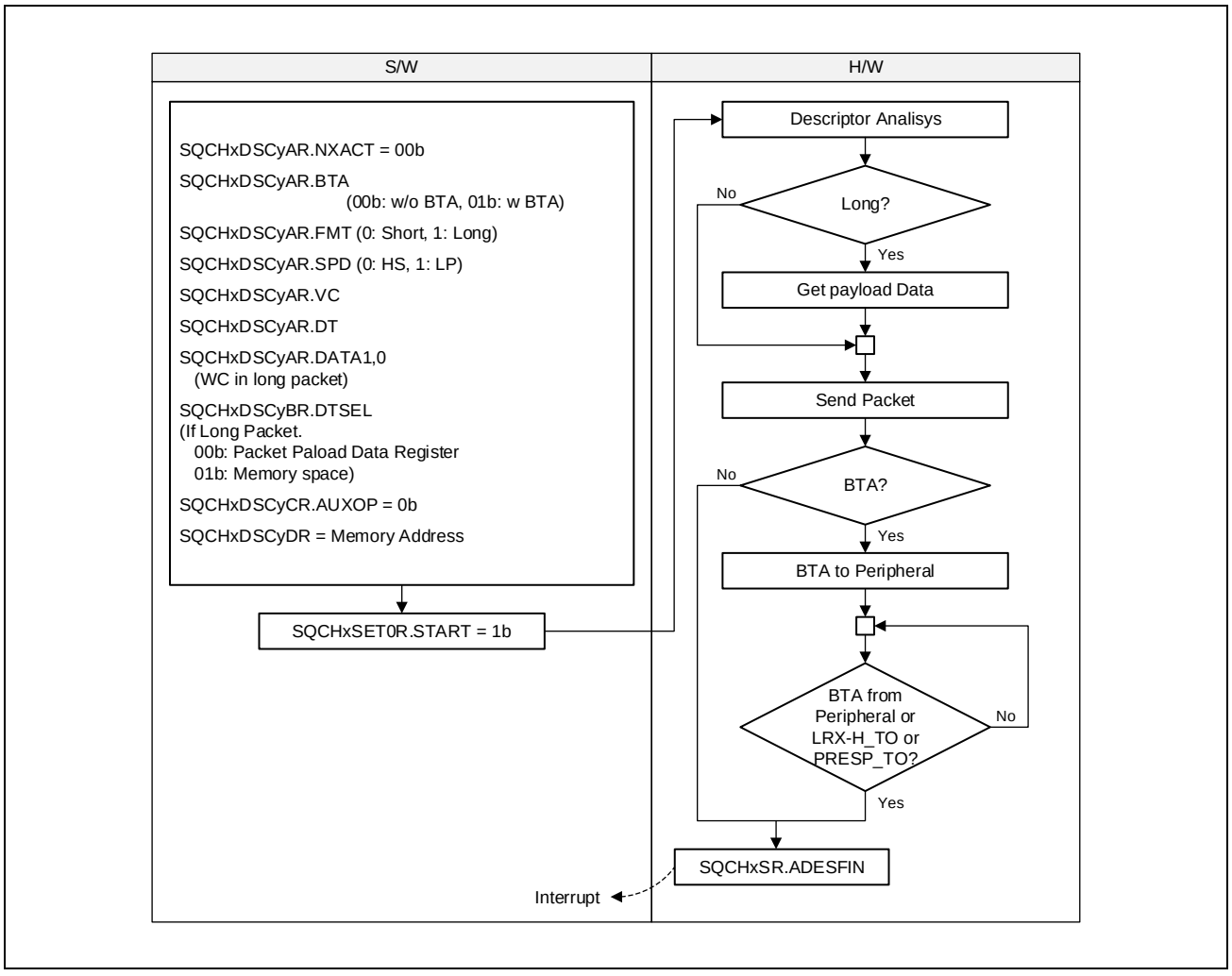



Figure 34.9 Single Packet Transfer

### (3) Single Packet Reception

This section describes sending read packets and receiving response packets using the Sequence Operation channel.

Since the read packet is a short packet, set the following registers before sending the packet.

SQCHxDSCyAR.NXACT: “00b”

SQCHxDSCyAR.FMT: “0b” Short Packet

SQCHxDSCyAR.SPD: “0b” High Speed or “1b” Low Speed

SQCHxDSCyAR.BTA: “10b” Read Request with BTA

SQCHxDSCyAR.VC, DT, DATA0, DATA1

SQCHxDSCyBR.DTSEL: Specify the payload data storage location of the response packet

“00b” RXPPD0R – RXPPD3R. Payloads size is limited up to 16 Bytes.

“01b” Memory space. Payloads size is limited up to RXBUFSZR.RXBUFSZ Bytes.

SQCHxDSCyCR.AUXOP: “0b”

SQCHxDSCyCR.ACTCODE: Rx Result Save Slot number

When SQCHxDSC00BR.DTSEL = “00b” is set, receiving payload data of 17 bytes or more is not supported. Therefore, if there is a possibility of receiving payload data of 17 bytes or more, please set SQCHxDSC00BR.DTSEL = “01b”.

After the above settings, write “1” to SQCHxSET0R.START to send the read packet.

BTA is performed after sending the read packet. When the bus right is transferred to the peripheral, the response packet is received. At this time, the Acknowledge and Error Report Packet may also be received at the same time.

The user can know the completion of Single Packet Reception by asserting SQCHxSR.ADESFIN.

The user can get an overview of the reception result by reading the RXRSSxR register with the number specified by SQCHxDSCyCR.ACTCODE, and can know the details of the previous reception result by reading the RXSR register.

If the received response packet is a Short Packet or a Long Packet with WC = 0, the setting value of SQCHxDSC00BR.DTSEL is meaningless.

**Figure 34.10** shows the flow of Single Packet Reception.

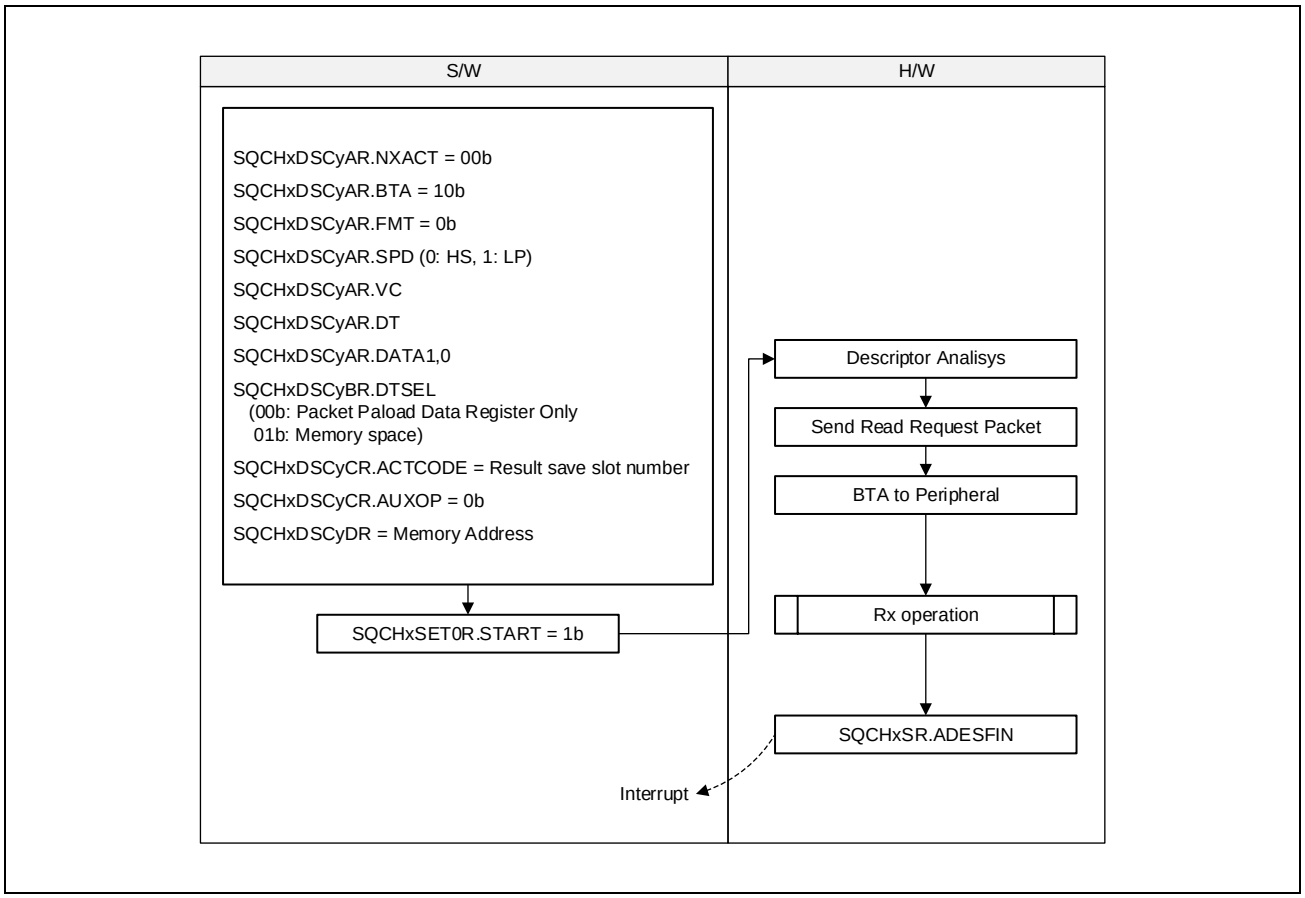



Figure 34.10 Single Packet Receive

### (4) Reset Trigger Transmission

The DSI-Tx Module can send triggers using the Sequence Operation channel.

Since the DSI-Tx Module is a host, the only trigger that can be sent is a reset trigger.

After setting SQCHxDSCyAR.NXACT = “00b”, SQCHxDSCyCR.AUXOP = “1b”, and SQCHxDSCyCR.ACTCODE = “H’00”, write “1” to SQCHxSET0R.START to send a reset trigger.

The user can know the completion of reset trigger transmission by asserting SQCHxSR.ADESFIN.

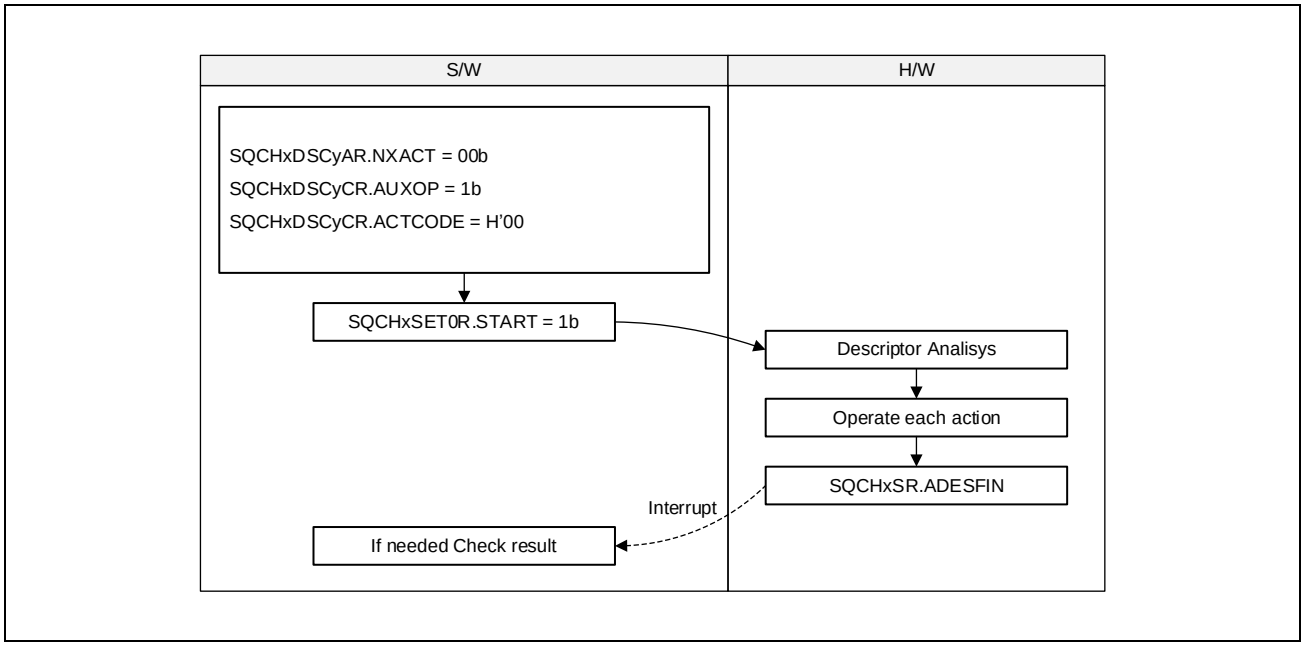



Figure 34.11 Trigger transmission

### (5) Execute Operation in Sequence

The following describes the descriptor in Sequence Operation.

By writing “1” to SQCHxSET0R.START, Sequence Operation channel x will be enabled and processing will be performed from descriptor # 0.

When the processing of the corresponding descriptor is completed, the next processing follows SQCHxDSCyAR.NXACT.

SQCHxDSCyAR.NXACT= “00b”: Sequence Operation channel x stop after the processing of the corresponding descriptor is completed

SQCHxDSCyAR.NXACT= “01b”: After the processing of the corresponding descriptor is completed, the next descriptor is executed. The next descriptor after descriptor # N is descriptor # (N + 1).

#### (6) Receiving EoTp

The DSI-Tx Module can receive End of Transmission Packets (EoTp).

The DSI-Tx Module notifies the reception of EoTp by setting RXSR.RXEOTP to “1”.

#### (7) Acknowledge and Error Reporting Mechanism

The user can know the reliability of the serial bus by using “Acknowledge and Error Reporting Mechanism”.

If Peripheral detects an error on the serial bus during the period from the previous Peripheral to Host communication to this Peripheral to Host communication, Peripheral will send an “Acknowledge and Error Report” packet.

When the DSI-Tx Module receives the "Acknowledge and Error Report” packet, it sets “1” to RXRSSxR.RXAKE, RXSR.RXAKE, and SQCHxSR.RXAKE. The Virtual Channel Identifier and Error Report of the latest “Acknowledge and Error Report” packet received are stored in the AKEPLATIR register, and the integrated value is displayed in the AKEPACMSR register.

### 34.4.2.4 Video-Input Operation

#### (1) Start of Video-Input Operation

The procedure for Start of Video-Input Operation is shown in **Figure 34.12**.

1. Initialize common settings. Refer **Section 34.4.2.1(1)**.
2. Some peripheral setting may be done by Sequence operation. Since it is differed from peripheral devices, it is out of this document.
3. Confirm Video-Input clock and hsclk stable and set HS clock. Refer **Section 34.4.2.2**.
4. Set Video-Input Operation channel parameters (VICH1PPSETR, VICH1VSSETR, VICH1VPSETR,VICH1HSSETR, VICH1HPSETR)
5. Calculate VICH1SET1R.DLY value and set. Refer **Section 34.4.2.4(7)**.
6. Set “1” to VICH1SET0R.VSTART with other VICH1SET0R setting values.
7. Wait until VICH1SR.VIRDY=1.
8. Start Video signal input

VICH1SET0R, VICH1SET1R, VICH1PPSETR, VICH1VSSETR, VICH1VPSETR, VICH1HSSETR and VICH1HPSETR are prohibited to change value between VICH1SET0R.VSTART=1 and VICH1SR.RUNNING=0 after VICH1SET0R.STOP=1.

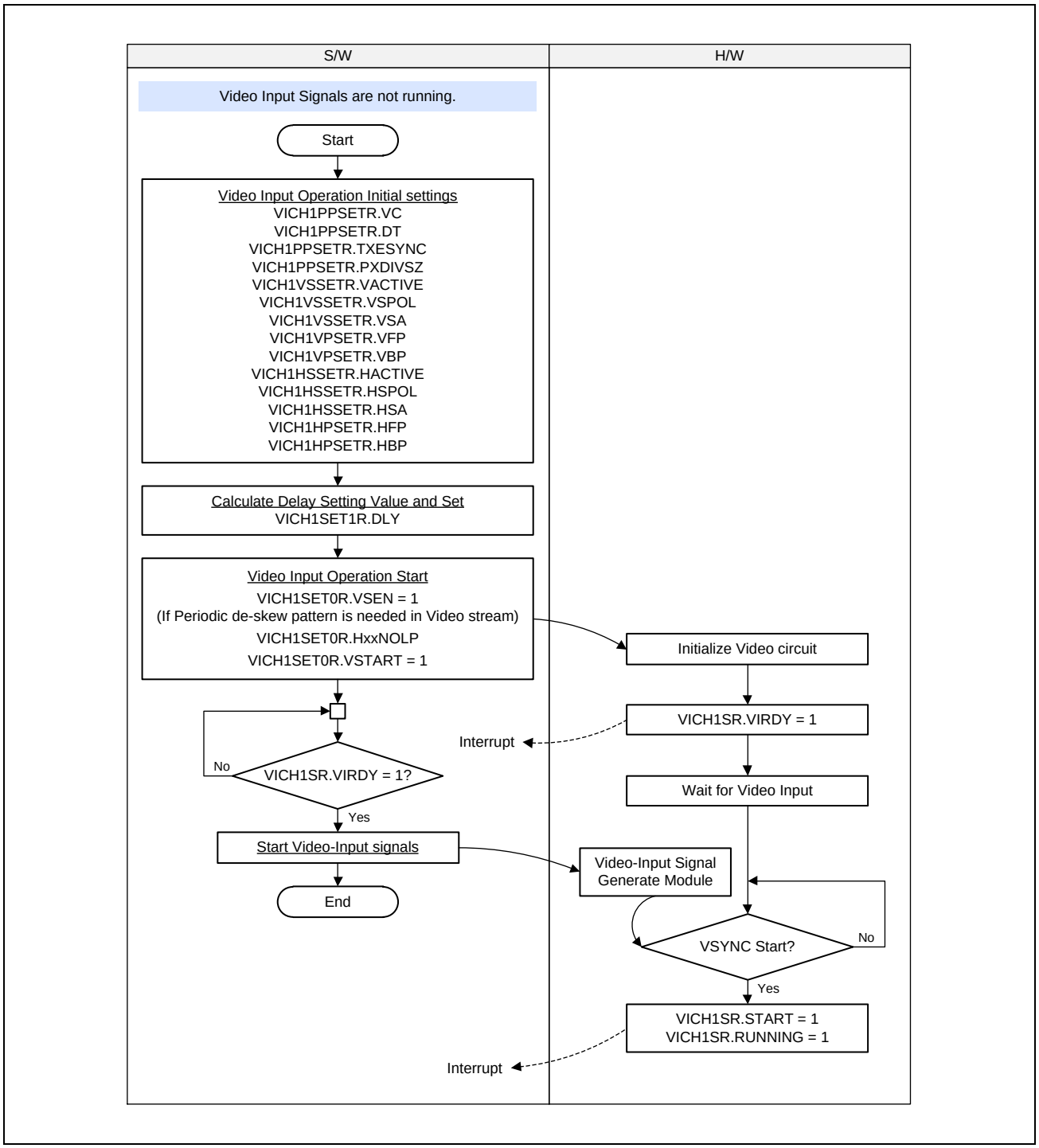



Figure 34.12 Video-Input channel Start

### (2) End of Video-Input Operation

The video-input operation stop flow is shown in **Figure 34.13**. Execute the flow when VICH1SR.RUNNING = 1.

(1) VICH1SET0R.VSTPAFT (**Figure 34.13**)

Write 1 to VICH1SET0R.VSTPAFT to request the Video-Input Operation to stop. When the DSI-Tx Module detects the start of one frame (assertion of VSYNC), it stops sending video mode packets and sets VICH1SR.STOP to 1.

After checking VICH1SR.STOP, stop the video input. The video input can be stopped at any time after VICH1SR.STOP becomes 1.

After stopping the video input, wait for LINKSR.HSBUSY to become 0.

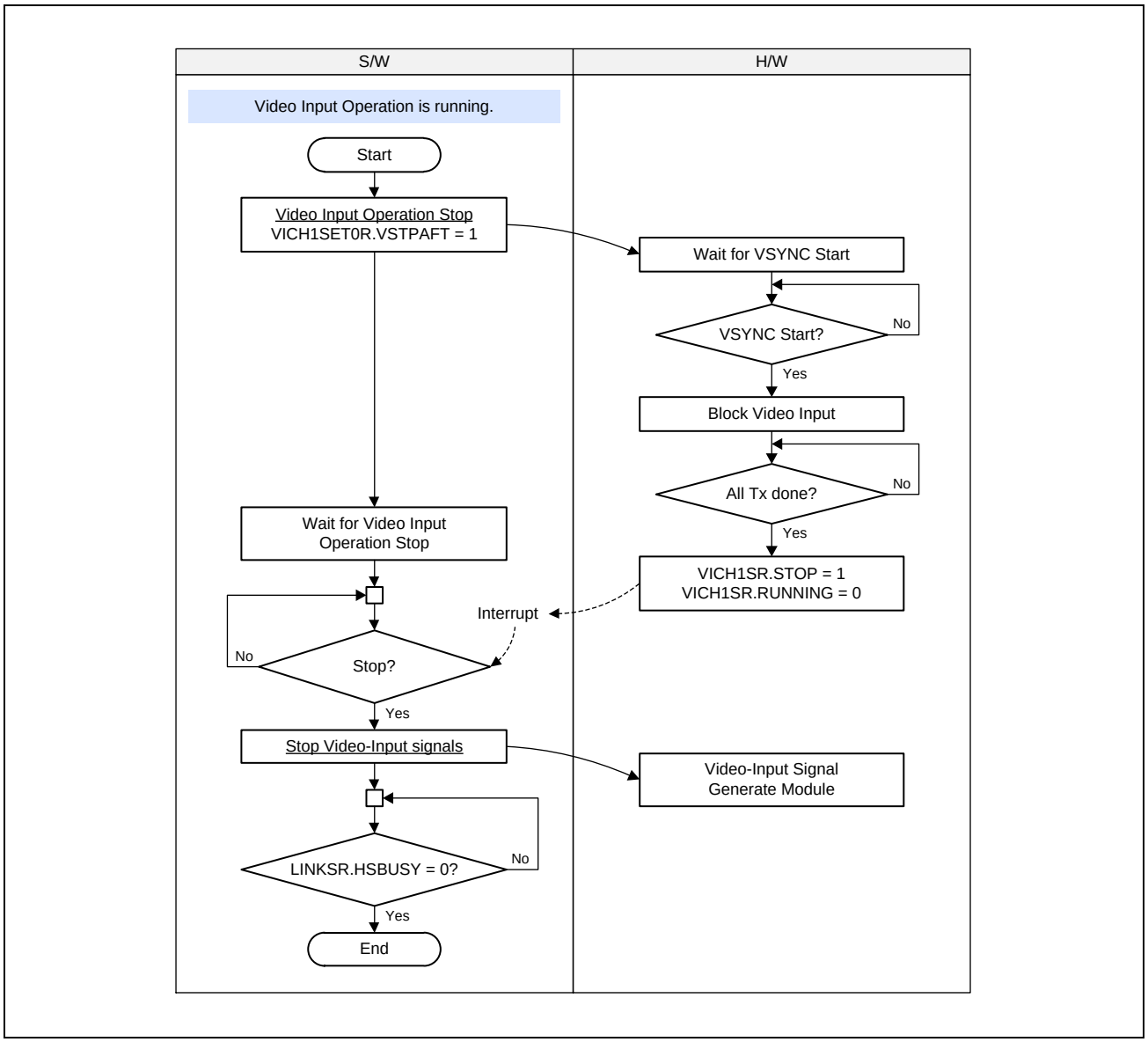



Figure 34.13 End of Video-Input Operation (VICH1SET0R.VSTPAFT)

### (3) Burst Mode

The DSI-Tx Module supports Burst Mode.

Burst Mode is a mode that time-compresses pixel data.

Make the D-PHY bandwidth wider than the Video-Input bandwidth for time compression of pixel data. Video transmissions with a 1-line pixel data size (VICH1HSSETR.HACTIVE * VICH1SET1R.BPP / 8) greater than 65,535 Bytes are not supported.

### (4) Non-Burst Mode

The DSI-Tx Module supports Non-Burst Mode.

Make the D-PHY bandwidth the same as the Video-Input bandwidth. Video transmissions with a 1-line pixel data size (VICH1HSSETR.HACTIVE * VICH1SET1R.BPP / 8) greater than 65,535 Bytes are not supported.

### (5) LP Transition and Blanking

The DSI-Tx Module automatically transitions the Data Lane from HS to LP if both of the following two conditions are met:

If it does not transition from HS to LP, it sends a Blanking Packet with the same VC as the VC in the previous packet to maintain the HS.

- There is time to make an HS => LP => HS transition before the next HS transfer starts.
- The period is not prohibited by VICH1SET0R.xxxNOLP to transition to LP

### (6) Non video packet action during Video-Input Operation Running

#### (a) Packet Transmission/Reception by Sequence operation

The DSI-Tx Module can send and receive packets (see **Section 34.4.2.3(2)** and **Section 34.4.2.3(3)**) using Sequence Operation while Video-Input Operation is operating*[1].

Make sure that all Sequence Operations are finished before starting the Video-Input Operation operation.

While Video-Input Operation is operating*[1], packet transmission / reception using Sequence Operation is performed only immediately after the HSE packet (VICH1PPSETR.TXESYNC = 1b) or VSS packet (VICH1PPSETR.TXESYNC = 0b) on the first horizontal line. Also, sending packets using Sequence Operation is possible only with HS (SQCHxDSCyAR.SPD = 0b). Sending with LP (SQCHxDSCyAR.SPD = 1b) is not supported. Send and receive packets that are completed within the BLLP period of the first horizontal line.

Execution of Sequence Operation processing is prohibited except for sending and receiving HS packets.

**Note 1.** “while Video-Input Operation is operating” here is different from VICH1SR.RUNNING. It means the period from setting VICH1SET0R.VSTART to 0 in the flow of **Figure 34.12** until VICH1SR.RUNNING becomes 0 in the flow of **Figure 34.13**. Also, please note that there is some latency from executing VICH1SET0R.VSTART to being transmitted to the inside of the module for clock transfer.

### (7) Delay about Video-Input Operation

Video-Input Operation run in some latency which call in the DSI-Tx Module “Delay”. “Delay” value must set to VICH1SET1R.DLY before operation.

“Delay” value is defined by calculation using below parameters.

Refer to attached excel file for the "RZA3M_MIPI-DSI_Calculated_register_table" about some calculation results of DLY.

(Please double-click the icon on the right side) 

- Clock frequencies
  - Video channel 1 clock (MIPI_DSI_VCLK) frequency
  - DSI High-Speed Transmit Word Clock (TxWordClkHS, hsclk) frequency
- Video parameters
  - HSA period, set by VICH1HSSETR.HSA
  - HACTIVE period, set by VICH1HSSETR.HACTIVE
  - HBP period, set by VICH1HPSETR.HBP
  - HFP period, set by VICH1HPSETR.HFP
  - BPP value, shown in VICH1SET1R.BPP and derived from VICH1PPSETR.DT
- Video transmit settings
  - The number of active lanes, set by TXSETR.NUMLANEUSE
  - Transmit mode, set by VICH1PPSETR.TXESYNC
  - No LP, set by VICH1SET0R.HSANOLP, HBPNOLP, HFPNOLP
  - Clock mode, set by HSCLKSETR.HSCLKMODE
  - Clock stop time, set by CLSTPTSETR.CLKSTPT
  - Go LP and back time, set by LPTRNSTSETR.GOLPBKT
- Other parameters
  - - Video-Input channel buffer size, shown in VICH1SET1R.CHBUFSZ

#### 34.4.2.5 Enter/Exit of ULPS

The following describes the ULPS transition / return sequence.

The clock lane and data lane (lanes 0 to 3) can be independently transitioned / restored to ULPS.

The clock lane can make ULPS transitions when TXSETR.CLEN = 1. It is forbidden to change TXSETR.CLEN to 0 while the clock lane is ULPS.

The data lane can make ULPS transitions when TXSETR.DLEN = 1. It is forbidden to change TXSETR.DLEN to 0 while the data lane is ULPS.

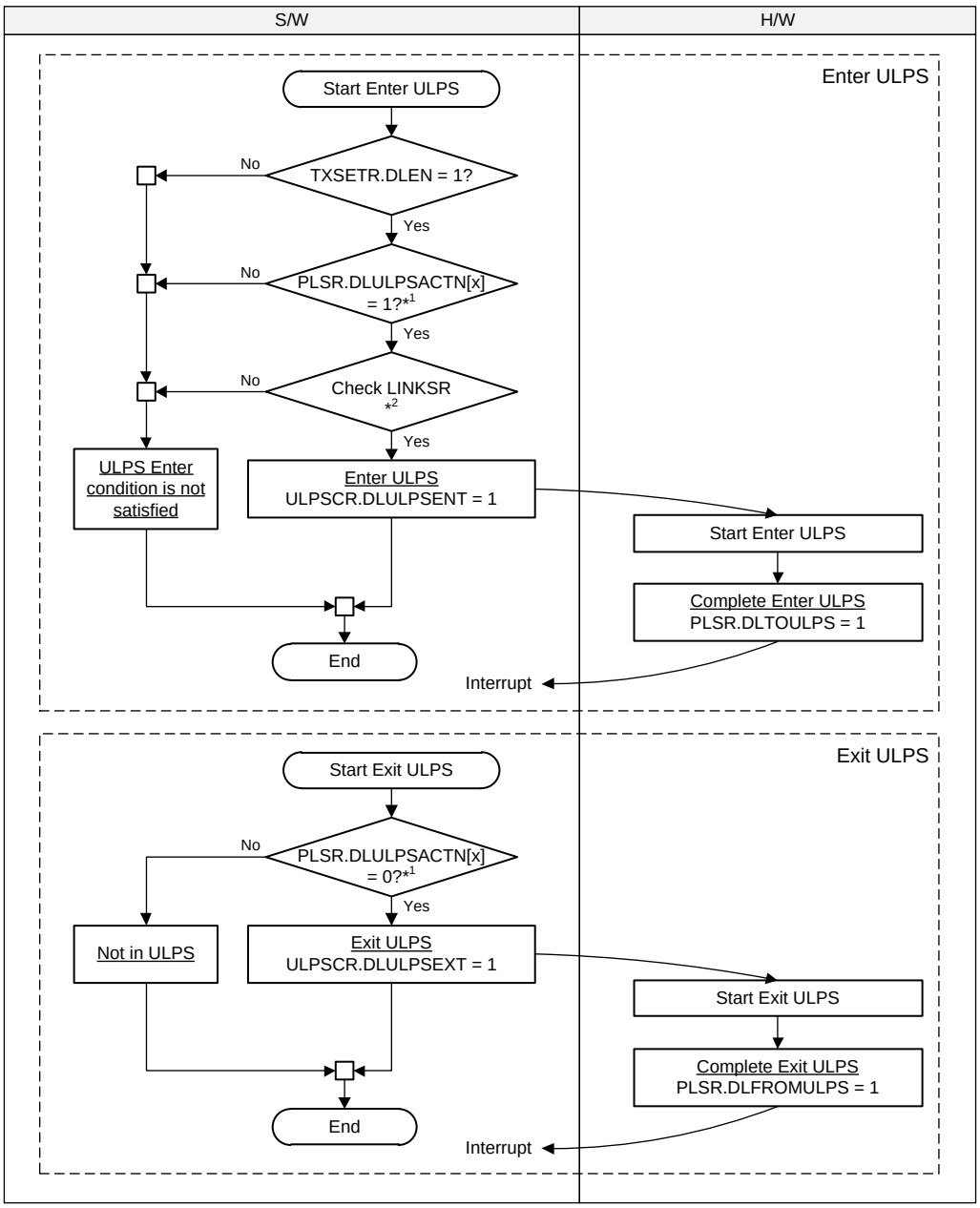
Data lanes enabled in TXSETR.NUMLANEUSE can be ULPS transitioned / restored. ULPS transition / return of data lanes is done simultaneously for all valid lanes. Only certain data lanes cannot be ULPS transitioned / restored.

When the data lane transitions to ULPS, another operation*[1] using the data lane cannot be performed until ULPS is restored.

When the clock lane transitions to ULPS, another operation*[2] using the clock lane cannot be performed until ULPS is restored.

**Note 1.** Sequence Operation, Video-Input Operation.

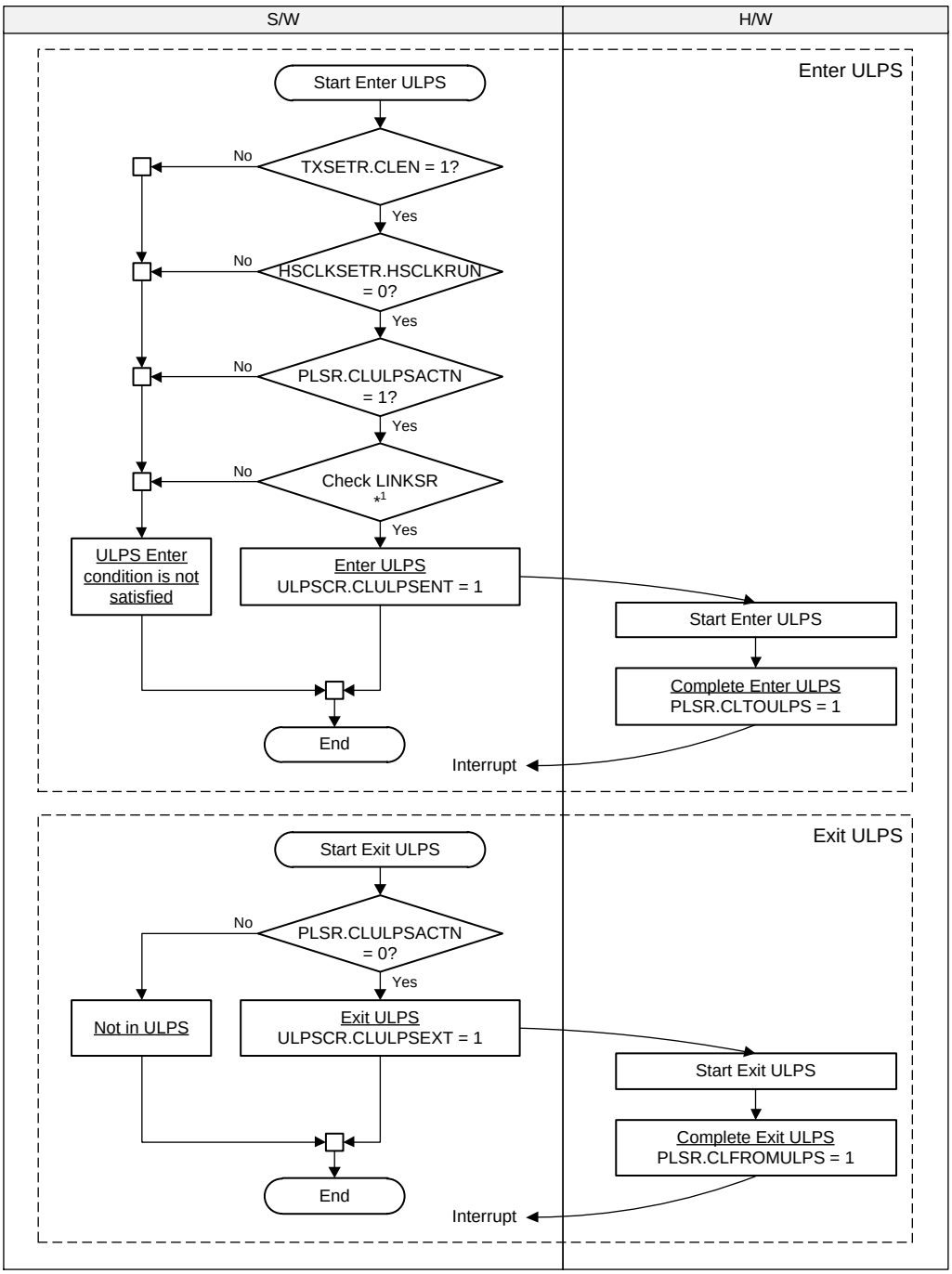
**Note 2.** HS Sequence Operation, Video-Input Operation.




Note 1. Check all enabled data lane which selected by TXSETR.NUMLANEUSE

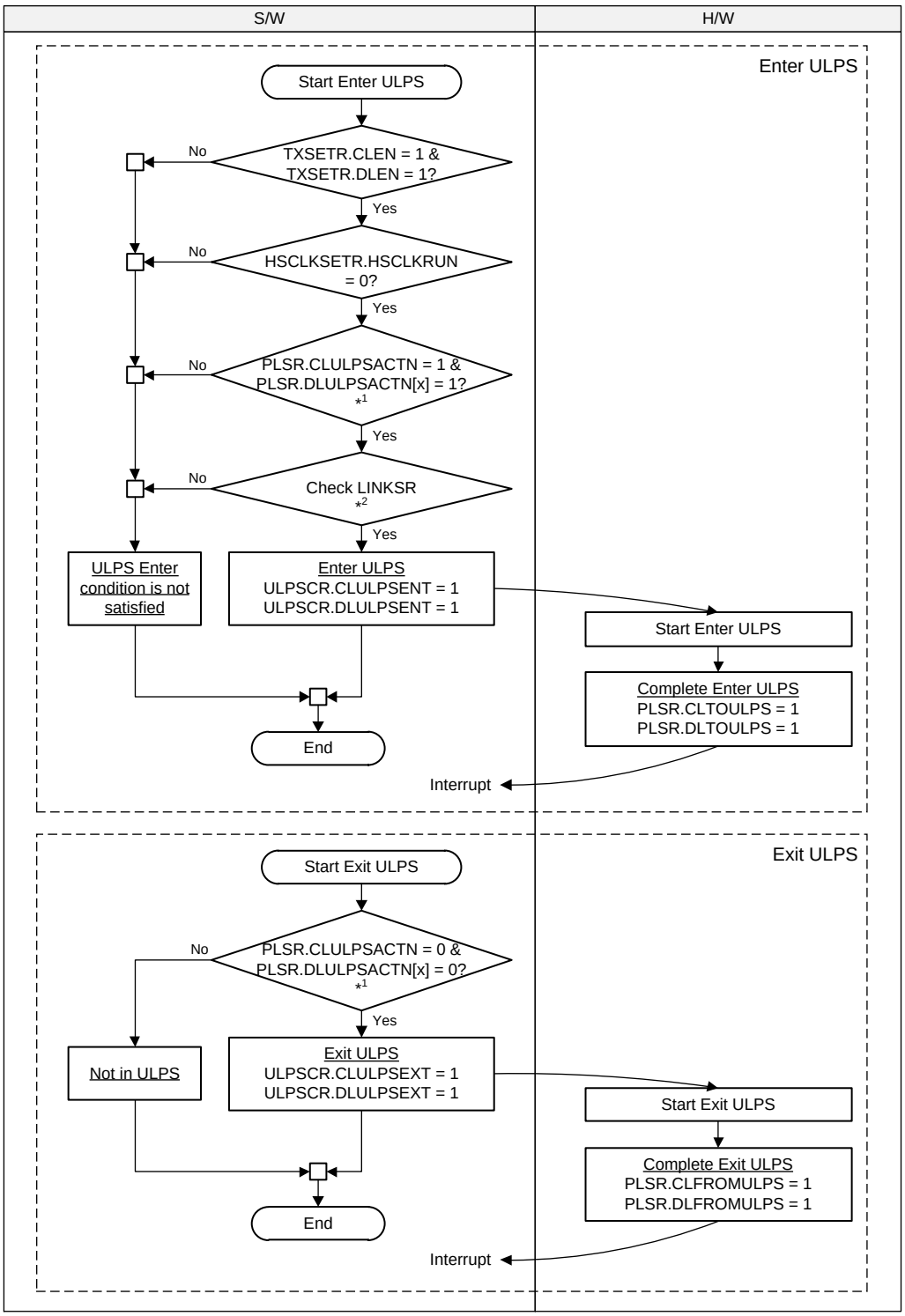
Note 2. (SQCHRUN0=0 and SQCHRUN1=0 and VICHRUN1=0 and HSBUSY=0 and LPBUSY=0)?

Figure 34.14 Enter/Exit of ULPS (Data Lane)




Note 1. (SQCHRUN1=0 and VICHRUN1=0 and HSBUSY=0)?

Figure 34.15 Enter/Exit of ULPS (Clock Lane)




Note 1. Check all enabled data lane which selected by TXSETR.NUMLANEUSE

Note 2. (SQCHRUN0=0 and SQCHRUN1=0 and VICHRUN1=0 and HSBUSY=0 and LPBUSY=0)?

Figure 34.16 Enter/Exit of ULPS (Clock and Data Lane)

#### 34.4.2.6 Error Handling

##### (1) Error Detection by Peripheral

Peripheral will notify you of the detected error by “Acknowledge and Error Report” packet.

When the DSI-Tx Module receives the “Acknowledge and Error Report” packet, it displays the latest result in the AKEPLATIR register and the accumulated result in the AKEPACMSR register.

When the user receives the “Acknowledge and Error Report” packet, review and change various settings as necessary.

##### (2) Error Detection in D-PHY

If the D-PHY detects an error, the DSI-Tx Module sets the corresponding register flag.

The errors supported by the DSI-Tx Module are as follows. For details on the error, refer to Annex A Logical PHY-Protocol Interface Description of MIPI Alliance Specification for D-PHY Version 2.1.

- ErrEsc
- ErrSyncEsc
- ErrControl
- ErrContentionLP0
- ErrContentionLP1

If the user detects this error, perform Software Reset (**Section 34.4.2.1(2)**) and Reset Trigger Transmission (**Section 34.4.2.3(4)**) if necessary.

When an ErrContentionLP0 / ErrContentionLP1 error occurs, be sure to execute Software Reset (**Section 34.4.2.1(2)**). At that time, write 1 to RSTCR.FCETXSTP, wait for the time when Contention is resolved (equivalent to LRX-H_TO), and then write 0 to RSTCR.FCETXSTP. Also, if necessary, perform Reset Trigger Transmission (**Section 34.4.2.3(4)**).

For more information on returning from Contention, see *Section 7.2, Contention Detection and Recovery* in *the MIPI Alliance Specification* for Display Serial Interface 1.3.1.

### (3) Timeout Error

The DSI-Tx Module has the timers defined in *Section 7.2.2, Contention Recovery Using Timers of the DSI standard* and the DSI-Tx Module original timers.

- Timers defined in DSI Specification
  - HS TX Timeout Error (HTX_TO)
  - LP-RX Host Processor Timeout Error (LRX-H_TO)
  - Turnaround Acknowledge Timeout Error (TA_TO)
- Original Timers
  - Rx Response Timeout Error

#### (a) HS TX Timeout (HTX_TO) Error

If this error occurs, perform Software Reset (**Section 34.4.2.1(2)**). In addition, perform Reset Trigger Transmission (**Section 34.4.2.3(4)**) if necessary.

If this error is detected, the setting value of HSTXTOSETR.HTXTO may be too small. When multiple channels are operating at the same time, multiple HS packets may be concatenated and transmitted continuously by HS. Set HSTXTOSETR.HTXTO to a value larger than the HS transmission period.

#### (b) LP-RX Host Processor Timeout (LRX-H_TO) Error

If this error occurs, perform Software Reset (**Section 34.4.2.1(2)**). In addition, perform Reset Trigger Transmission (**Section 34.4.2.3(4)**) if necessary.

If this error is detected, the setting value of LRXHTOSETR. LRXHTO may be too small.

There is a possibility of receiving multiple packets such as response packet and "Acknowledge and Error Report" packet in one reception.

Measure the time from the completion of the Bus Turnaround sequence from Host to Peripheral (LP-11 drive detection by Peripheral) to the completion of Bus Turnaround sequence from Peripheral to Host (LP-11 drive start by Host).

Check the Peripheral specifications, and set the value of DSI_LPCLK cycle count (rounded up) + X or more corresponding to the above time in the LRXHTOSETR.LRXHTO register.

#### (c) Turnaround Acknowledge Timeout (TA_TO) Error

If this error occurs, perform Software Reset (**Section 34.4.2.1(2)**) and Reset Trigger Transmission (**Section 34.4.2.3(4)**) if necessary.

If this error is detected, the setting value of TATOSETR.TATO may be too small.

Measure the time from the start of the Peripheral Bus Turnaround sequence (LP-10 drive by the Host) to the completion (LP-11 drive detection by the Peripheral) from the Host.

Check the Peripheral specifications and set the value of DSI_LPCLK cycle count (rounded up) + X or more corresponding to the above time in the TATOSETR.TATO register. This error also occurs if the Peripheral does not support bidirectional communication.

#### (d) Rx Response Timeout Error

Refer to the Peripheral Response Timeout Error (**Section 34.4.2.6.4.1.1**) for more information.

### (4) Rx Related Error in Sequence Operation

#### (a) Rx Packet Related Error in Sequence Operation

The following describes the received packet error detected by the DSI-Tx Module.

Detects up to one error per packet. (However, ECC 1bit Error may occur at the same time as other errors.)

Errors are determined in the following order:

Peripheral Response Timeout Error, Malformed Packet Error, ECC Error, Unexpected Packet Error, WC Error, CRC Error, Nothing Return Error

##### 34.4.2.6.4.1.1. Peripheral Response Timeout Error

RXSR.PRESPTOERR is set to 1 if the BTA does not start receiving packets or triggers within the time set in the register after the bus right transitions from Host to Peripheral.

If this error is detected, packets and triggers will not be received until the bus right transitions from Peripheral to Host.

[Timeout time setting register]

Peripheral Response Timeout value for BTA-only is set by PRESPTOBTASETR.

Peripheral Response Timeout value for LP Read with BTA is set by PRESPTOLPSETR.PRESPTOLPR.

Peripheral Response Timeout value for LP Write with BTA is set by PRESPTOLPSETR.PRESPTOLPW.

Peripheral Response Timeout value for HS Read with BTA is set by PRESPTOHSSETR.PRESPTOHSR.

Peripheral Response Timeout value for HS Write with BTA is set by PRESPTOHSSETR.PRESPTOHSW.

If this error is detected, the register setting value may be too small, Measure the time from the completion of the Bus Turnaround sequence of Peripheral from the Host (LP-11 drive detection by Peripheral) to the completion of reception of the first byte of the packet or the completion of trigger reception. After checking the Peripheral specifications, set the value of DSI_LPCLK cycle number (rounded up) + X or more corresponding to the above time.

If this error occurs, perform the necessary processing such as resending the previous command.

##### 34.4.2.6.4.1.2. Malformed Packet Error

If a packet less than 4 bytes is received, RXSR.MLFERR is set to 1. Since there is an error in the received data, perform necessary processing such as resending the previous command.

#### 34.4.2.6.4.1.3. ECC Error

ECC generation is required for DSI v1.3.1 standard compliant Peripherals. However, since ECC generation is an option for “earlier revision of DSI Peripherals”, the DSI-Tx Module can select whether to enable or disable ECC checks using DSISETR.ECCEN.

When connecting to a Peripheral that does not support ECC generation, disable ECC checking by setting DSISETR.ECCEN to 0.

##### 34.4.2.6.4.1.3.1. ECC 1bit Error

If an ECC 1bit error is detected, RXSR.ECCERR1B is set to 1 and 1bit error correction is performed.

The DSI-Tx Module automatically corrects the error and restores the normal packet header, so the subsequent packet reception processing continues.

##### 34.4.2.6.4.1.3.2. ECC Multi-bit Error

If an ECC multi-bit error is detected, RXSR.ECCERR is set to 1 and the received packet is discarded. If this error occurs, perform the necessary processing such as resending the previous command.

#### 34.4.2.6.4.1.4. Unexpected Packet Error

If an Unexpected Packet is received, RXSR.UEXPKTERR is set to 1.

When this error occurs, the packet header is stored in the RXRSSxR register, and the subsequent reception processing is stopped until the bus right transitions from Peripheral to Host.

The DSI-Tx Module determines the following packets as Unexpected Packets.

- Received packets in which Data Type field of the packet header is “Reserved” as described in Table23 Data Types for Peripheral-Sourced Packets of the DSI standard
- Received packets other than “Acknowledge and Error Report” packet and “EoTp” in BTA after sending non-read packet
- Received packets other than “Acknowledge and Error Report” packet and “EoTp” when the BTA is BTA only
- BTA after sending read packet receives response packet and then receives response packet again

#### 34.4.2.6.4.1.5. WC Error

RXSR.WCERR is set to 1 if a long packet with payload data shorter than the WC value is received.

When this error occurs, there is an error in the payload data length, but the payload data is not discarded due to this error, and the packet header is stored in the RXRSSxR register.

Since there is an error in the received data, perform necessary processing such as resending the previous command.

#### 34.4.2.6.4.1.6. CRC Error

To connect to Peripherals that do not support CRC generation, the DSI-Tx Module can enable or disable CRC checking for each Virtual Channel using DSISETR.CRCEN[3:0]. When connecting to a Peripheral that does not support CRC generation, disable the CRC check by setting the corresponding bit in DSISETR.CRCEN to 0.

When this error occurs, there is an error in the payload data, but the payload data is not discarded due to this error, and the packet header is stored in the RXRSSxR register.

If this error occurs, perform the necessary processing such as resending the previous command.

#### 34.4.2.6.4.1.7. Nothing Return Error

RXSR.NORETERR is set to 1 if the BTA transitions the bus right from Host to Peripheral and then does not receive packets or triggers and the bus right transitions from Peripheral to Host.

If this error occurs, perform the necessary processing such as resending the command. If the error recurs, perform Software Reset (**Section 34.4.2.1(2)**) and Reset Trigger Transmission (**Section 34.4.2.3(4)**) as necessary.

### (b) Rx Long Packet Data Payload Related Error in Sequence Operation

#### 34.4.2.6.4.2.1. Maximum Return Packet Size Error

RXSR.MAXRPSZERR is set to 1 if the WC value of the received long packet is greater than the DSISETR.MRPSZ setting.

When this error occurs, the packet header is stored in the RXRSSxR register and the payload data is discarded.

Although the payload data is discarded, errors related to payload data reception (WC Error, CRC Error) are detected.

If this error occurs, set DSISETR.MRPSZ to a value greater than or equal to the value set by the Set Max Return Packet Size command.

If you still get this error despite the correct settings, consider setting DSISETR.MRPSZ to a larger value because Peripheral is returning a large packet in violation of the DSI standard.

#### 34.4.2.6.4.2.2. Internal Bus Error

If Internal Bus Write detects an error, RXSR.IBERR is set to 1.

Review the settings of SQCHxDSCyDR.LADDR and DSISETR.MRPSZ.

#### 34.4.2.6.4.2.3. Rx Buffer Overflow Error

If the following Rx Buffer Overflow Error is detected, RXSR.RXOVFERR is set to 1.

In the case of SQCHxDSCyBR.DTSEL=00b Packet Payload Data register Mode:

- The Rx Payload Data is over 16 Byte length

In the case of SQCHxDSCyBR.DTSEL=01b Long Packet data use memory space Mode:

- RAM Overflow for the LP Rx is detected

If this error occurs, please execute the following.

In the case of Data register Mode:

- Change to use memory space Mode
- Change peripheral device’s Maximum Return Packet Size as not over 16 Byte

In the case of memory space Mode:

- Change peripheral device’s Maximum Return Packet Size smaller
- Make Internal Bus faster

### (5) Tx Packet Related Error in Sequence Operation

#### (a) Internal Bus Error

If Internal Bus Read detects an error, SQCHxSR.TXIBERR is set to 1. Review the settings of SQCHxDSCyAR.DATA1 / 0 and SQCHxDSCyDR.LADDR.

#### (b) Packet too big Error

SQCH1SR.PKTBIGERR is set to 1 if the Sequence Operation packet is of a size that cannot be sent within the BLLP period of the first horizontal line of the Video-Input Operation.

If this error occurs, the Sequence Operation packet will not be sent until the Video-Input Operation is completed.

### (6) Error in Video-Input operation

#### (a) Video-Input Buffer Overflow Error

VICH1SR.VBUFOVF is set to 1 if Overflow of Video-Input Buffer is detected.

If this error occurs, perform Software Reset (**Section 34.4.2.1(2)**). Also, take necessary measures such as increasing the frequency of TxWordClkHS (hsclk) and decreasing the VICH1SET1R.DLY value. In addition, perform Reset Trigger Transmission (**Section 34.4.2.3(4)**) if necessary.

#### (b) Video-Input Buffer Underflow Error

If the Video-Input Buffer Underflow is detected, VICH1SR.VBUFUDF is set to 1.

If this error occurs, perform Software Reset (**Section 34.4.2.1(2)**). Also, take necessary measures such as lowering the frequency of TxWordClkHS (hsclk) and increasing the VICH1SET1R.DLY value. In addition, perform Reset Trigger Transmission (**Section 34.4.2.3(4)**) if necessary.

#### (c) Timing Error

VICH1SR.TIMERR is set to 1 if no video packets (including Sync Event packets) were sent during the Video-Input Operation at the intended timing.

If this error occurs, perform Software Reset (**Section 34.4.2.1(2)**). Also, take necessary measures such as reviewing the setting value of LPTRNSTSETR.GOLPBKT. In addition, perform Reset Trigger Transmission (**Section 34.4.2.3(4)**) if necessary.

## 34.4.3 Interface

### 34.4.3.1 Video-Input Operation

#### (1) Video Input Timing

Video-Input Timing and Video-Input Parameters are described in **Figure 34.17** and **Table 34.8**.

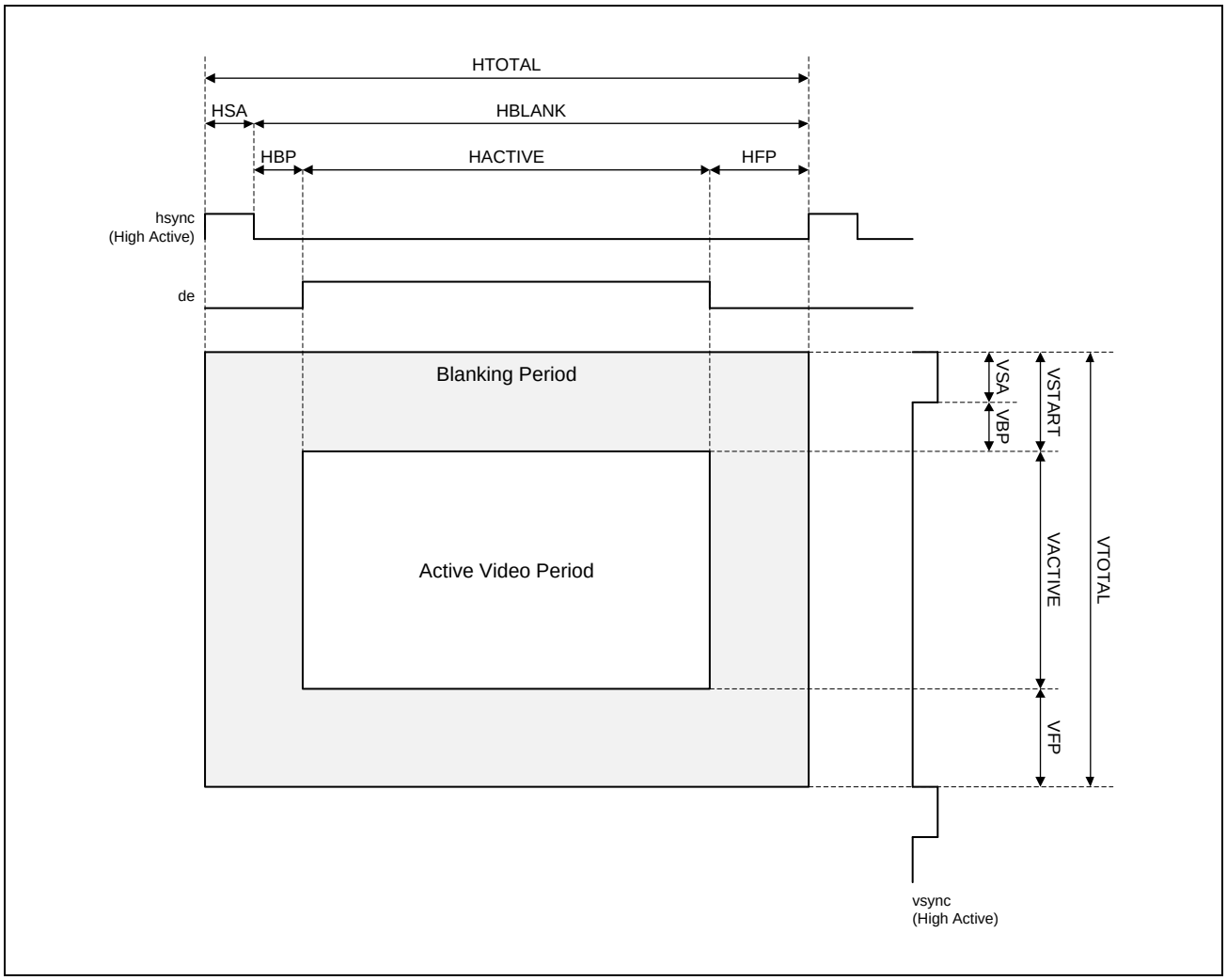



Figure 34.17 Video input Timing

Table 34.8 Configurations for Video Input interface (1/2)

- Common Rules for all mode

| Parameter | Description |
|---|---|
| All | All parameters related to Input Video should be set to the value of not less than 1.<br>Example for “All parameters”: HSA, HBP, HACTIVE, HFP, VSA, VBP, VACTIVE, VFP |
| HACTIVE | |

| VICH1PPSETR.DT | VICH1HSSETR.HACTIVE |
|---|---|
| H’1E: Packed Pixel Stream, 18bit RGB | This value should be the multiple of 4. |
| H’2E: Loosely Packed Pixel Stream, 18bit RGB<br>H’3E: Packed Pixel Stream, 24bit RGB | No Limitation |

| Parameter | Description |
|---|---|
| HFP | Rounddown (HFP*BPP/ 8, 0) >= 12, (12 = means Pixel Packet Header & footer 6 + Blanking Packet Header & footer 6) |
| HTOTAL | |

| VICH1PPSETR.DT | VICH1HSSETR.HSA +<br>VICH1HPSETR.HBP +<br>VICH1HSSETR.HACTIVE +<br>VICH1HPSETR.HFP |
|---|---|
| H’1E: Packed Pixel Stream, 18bit RGB | This value should be the multiple of 4. |
| H’2E: Loosely Packed Pixel Stream, 18bit RGB<br>H’3E: Packed Pixel Stream, 24bit RGB | No Limitation |

- Additional Rules for non-burst Sync Pulse mode

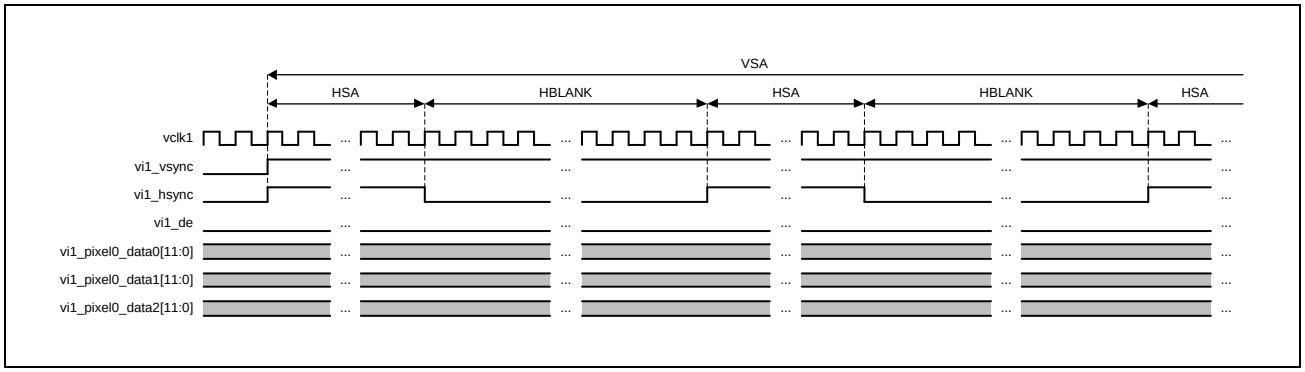
| Parameter | Description |
|---|---|
| Sequence Operation Packet | When a packet by sequence operation is transmitted during Video Opearation (VICH1SR.RUNNING=1), the period of (HBP+HACTIVE+HFP) should be set the value longer than sum of below periods.<br>• Period for the Tx Sync Event packet (4 Byte)<br>• Period for the packet by sequence operation<br>• Period for the Header and CRC of a Blanking Packet (6 Byte) |
| HBP+HACTIVE+HFP | Transition of HS-LP-HS is caused within the period between current Sync Event Packet and next Sync Event Packet, the period of (HBP+HACTIVE+HFP) should be set to the value within the range described below.<br>(HBP+HACTIVE+HFP) > (period for the Tx Sync Event Packet) + (period for transition time as HS-LP-HS) |
| HSA | Transmission of Blanking Packet is caused within the period between current Sync Event Packet and next Sync Event Packet, the period of HSA should be set to the value within the range described below.<br>(HSA*BPP)/8 >= 10<br>Notice, HSA should be set to the value not less than minimum value of (period for the Tx Sync Event Packet) + (period for the Tx Blanking Packet). |
| HBP | Transmission of Blanking Packet is caused within the period between current Sync Event Packet and next Sync Event Packet, the period of HPB should be set to the value within the range described below.<br>(HBP*BPP)/8 >= 10<br>Notice, HPB should be set to the value not less than minimum value of (period for the Tx Sync Event Packet) + (period for the Tx Blanking Packet). |

Table 34.8 Configurations for Video Input interface (2/2)

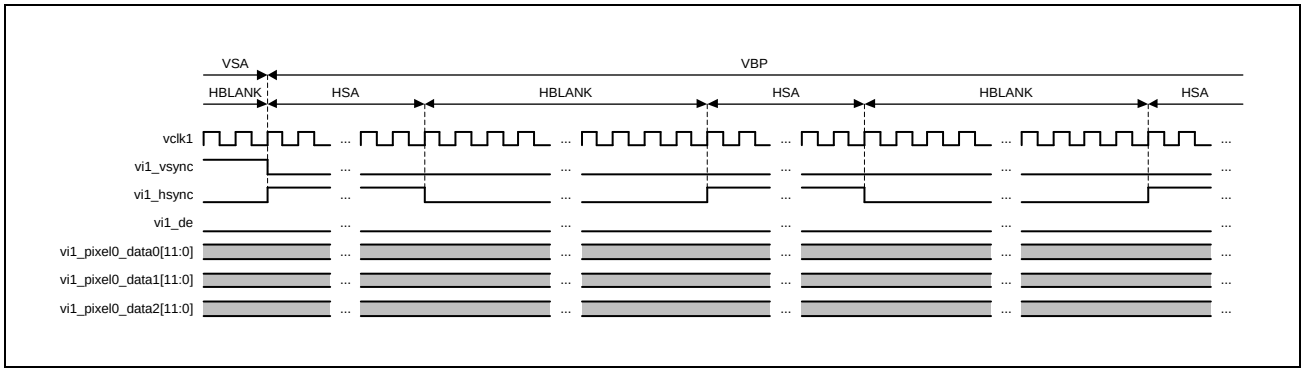
- Additional Rules for non-burst Sync-Event mode and Burst mode

| Parameter | Description |
|---|---|
| Sequence Operation Packet | When a packet by sequence operation is transmitted during Video Operation (VICH1SR.RUNNING=1), the period of HTOTAL should be set the value longer than sum of below periods.<br>• Period for the Tx Sync Event packet (4 Byte)<br>• Period for the packet by sequence operation<br>• Period for the Header and CRC of a Blanking Packet (6 Byte) |
| HTOTAL | Transition of HS-LP-HS is caused within the period between current Sync Event Packet and next Sync Event Packet, the period of HTOTAL should be set to the value within the range described below.<br>HTOTAL > (period for the Tx Sync Event Packet) +(period for transition time as HS-LPHS) |
| HSA+HBP | Transmission of Blanking Packet is caused within the period between current Sync Event Packet and next Sync Event Packet, the period of (HSA+HBP) should be set to the value within the range described below.<br>((HSA+HBP)*BPP)/8 >= 10<br>Notice, HSA+HBP should be set to the value not less than minimum value of (period for the Tx Sync Event Packet) + (period for the Tx Blanking Packet). |

(1) VSA Start (and VFP End)

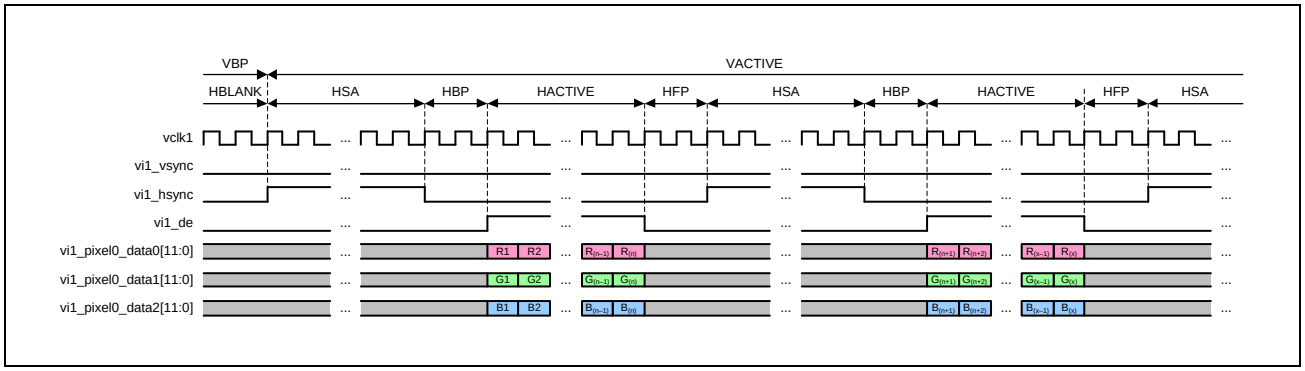



(2) VSA End and VBP Start

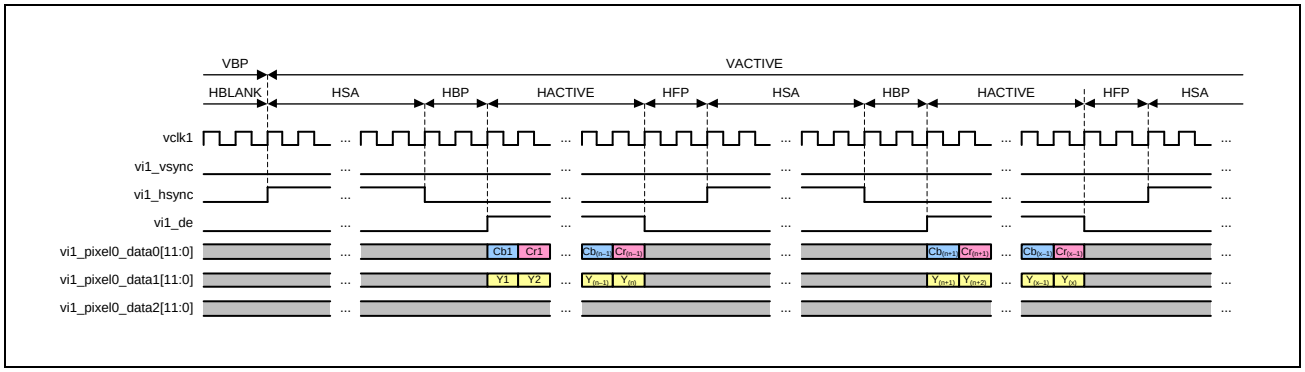


(3) VBP End and VACTIVE Start

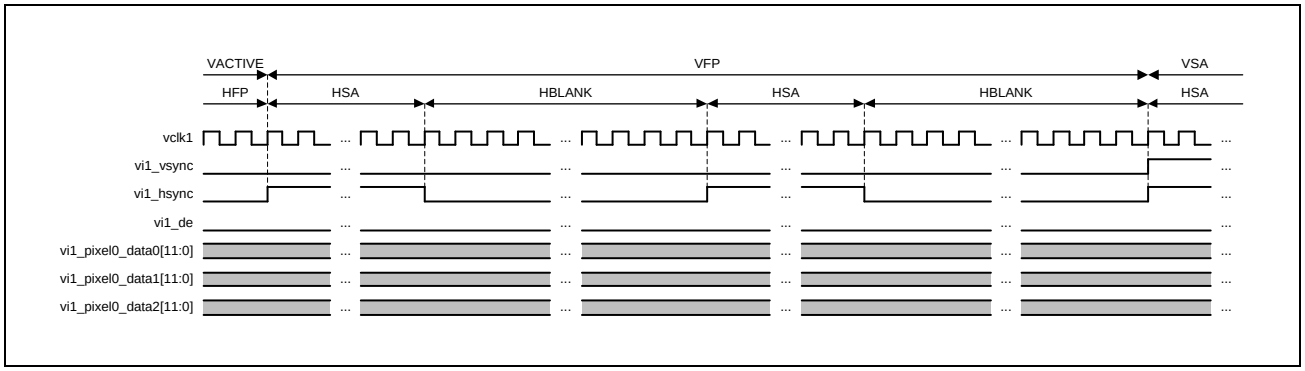
(a) RGB




(b) YCbCr 4: 2: 2




(4) VACTIVE End and VFP Start



#### 34.4.3.2 Video Input Start, End, Change Setting

(1) Video Input Start
Video input must be started after Video-Input Operation Start sequence. It must be start from V-Sync start edge.

(2) Video Input End
End of Video Input must follow sequence of **Section 34.4.2.4(2), End of Video-Input Operation**.

(3) Change of Video Input Setting
Changing of Video-Input setting must be done during Video-Input operation is not running.

#### 34.4.3.3 Clock Frequencies

Video clock and DSI HS Byte clock must follow the relationship.

**(1) Burst Mode**

Video clock Frequency [Hz] * Video Pixel Bit Depth (BPP) [bit]

<= DSI HS Byte clock Frequency[Hz] * 8[bit] * Number of DSI HS Data Lane

In the case of Video-Input bandwidth equals DSI output bandwidth, it will be indistinguishable from non-burst sync-event mode.

**(2) Non-Burst Mode**

Video clock Frequency [Hz] * Video Pixel Bit Depth (BPP) [bit]

= DSI HS Byte clock Frequency[Hz] * 8[bit] * Number of DSI HS Data Lane

## 34.5 Usage Note

### 34.5.1 Restriction and Notes

In this chapter, restriction and note of the DSI-Tx Module are described.

Table 34.9 Attention and Restriction List

| Items | Restriction and Notes |
|---|---|
| TxWordClkHS | Each Lanes TxWordClkHS must be synchronized. The DSI-Tx Module uses Data Lane0 TxWordClkHS as hsclk. |
| Unused input signals | All unused input signals must be input 0, except following signals.<br>Following signals must be input 1, when they are not used.<br>ppi_dl1_ulpsactivenot<br>ppi_dl2_ulpsactivenot<br>ppi_dl3_ulpsactivenot |
| TxReadyHS assert | D-PHY must be asserted TxReadyHS within ($2^{RELDSI2TX_LNK_RAM_WORD_DEPTH}$ – 2) hsclk cycles after TxRequestHS asserted. |

# 39. 2D Drawing Engine (DRW)

## 39.1 Overview

The 2D Drawing Engine (DRW) provides flexible functions that can support almost any object geometry, rather than being bound to only a few specific geometries such as lines, triangles, or circles. The edges of every object can be independently blurred or anti-aliased.

Rasterization is executed on the bounding box of an object from left to right and top to bottom at a rate of one pixel per clock cycle. The 2D Drawing Engine can also raster from bottom to top in certain cases, to optimize the performance. In addition, optimization methods are provided to avoid rasterization of many empty pixels of the bounding box.

The distances to the edges of the object are calculated by a set of edge equations for every pixel of the bounding box. These edge equations can be combined to describe the entire object. If a pixel is inside the object, it is selected for rendering, and if it is outside it is discarded. If a pixel is on an edge, an alpha value for anti-aliasing is chosen proportional to the distance between the pixel (the edge) and another pixel nearest to the edge. Every pixel that is selected for rendering can be textured. The resulting aRGB quadruple can be modified by a general raster operation approach independently for each of the four channels. The aRGB quadruple can then be blended with one of the multiple blend modes of the module.

The 2D Drawing Engine provides two inputs (texture read and framebuffer read) and one o

utput (framebuffer write). The internal color format is always aRGB (8888). The color formats from the inputs are converted to the internal format on read and converted back on write.

**Figure 39.1** shows examples of objects that can be drawn in hardware with the 2D Drawing Engine, **Figure 39.2** shows a simplified rendering pipeline setup, and **Figure 39.3** shows the block diagram of the module.

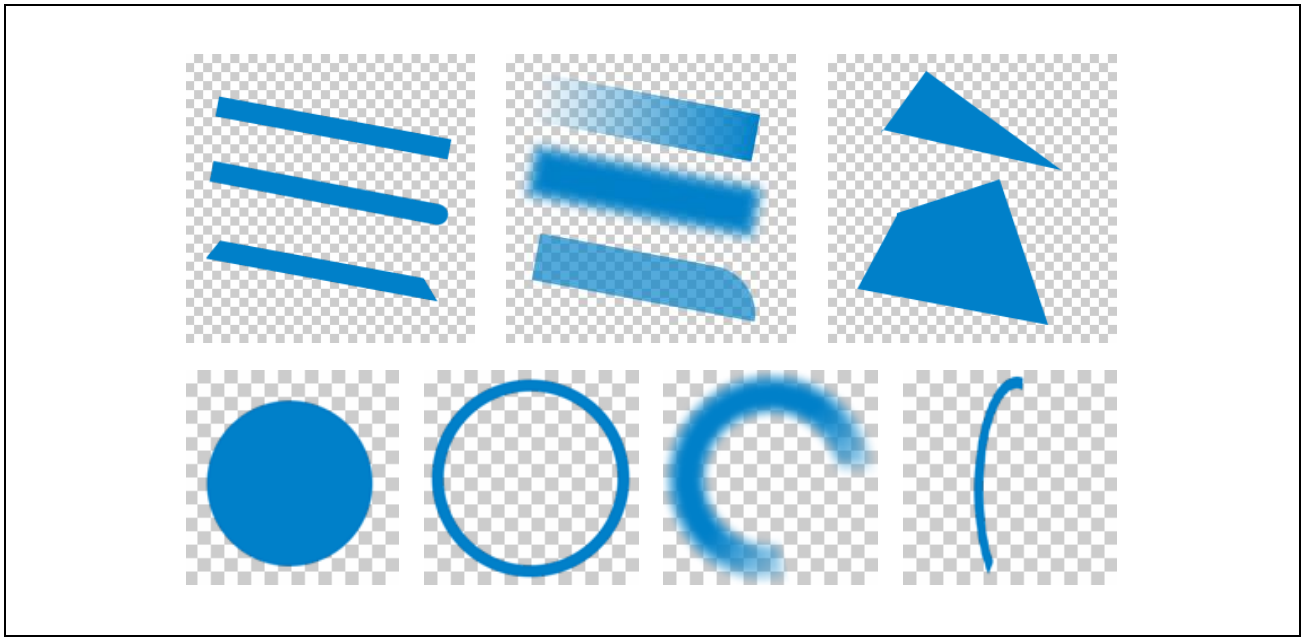


Figure 39.1 Examples of Drawing Objects

The 2D Drawing Engine also supports a display list mode, which efficiently decouples the CPU and graphics controller to enable rendering in parallel with other CPU activities.

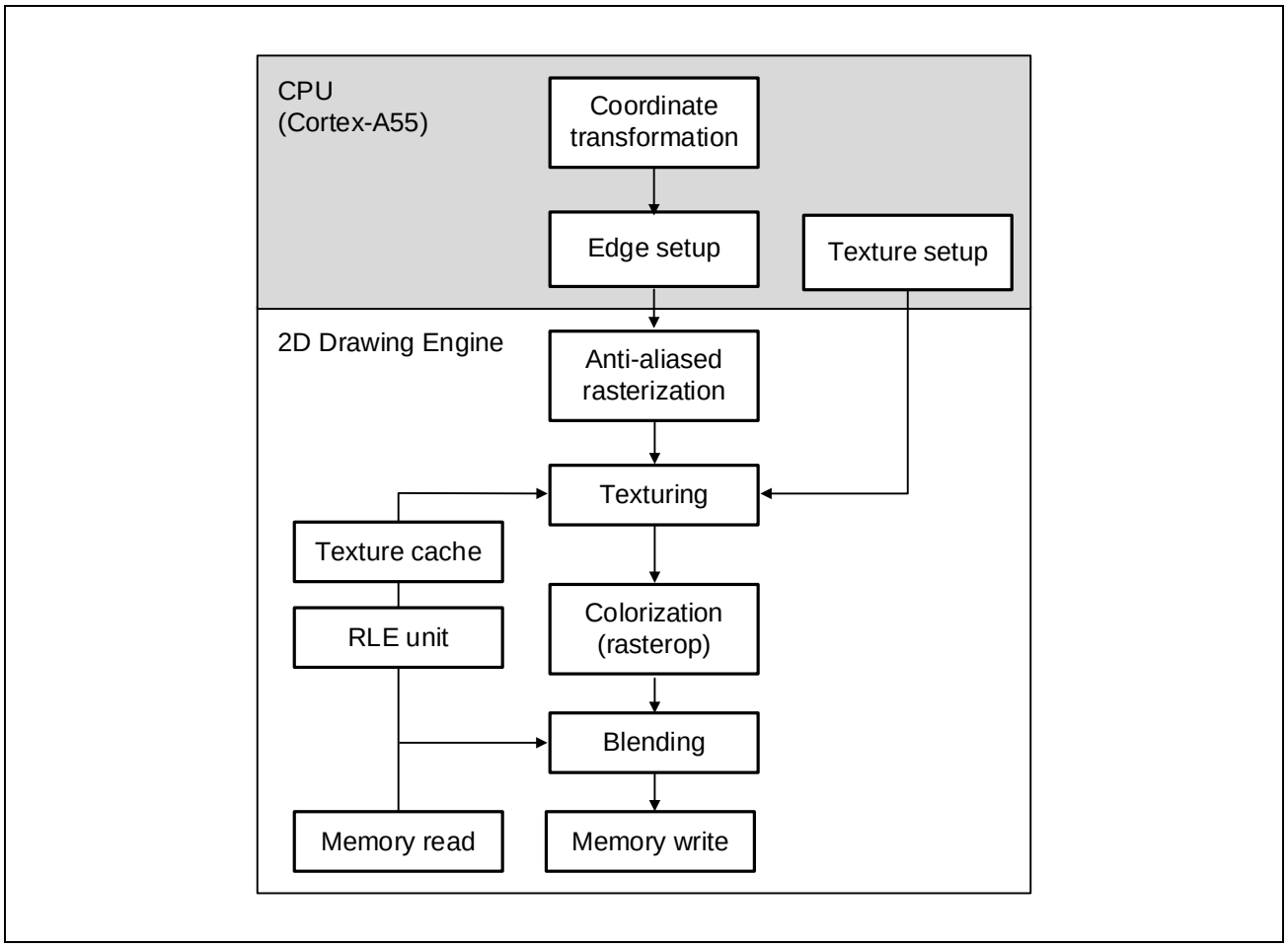



Figure 39.2 Simplified Rendering Pipeline Setup

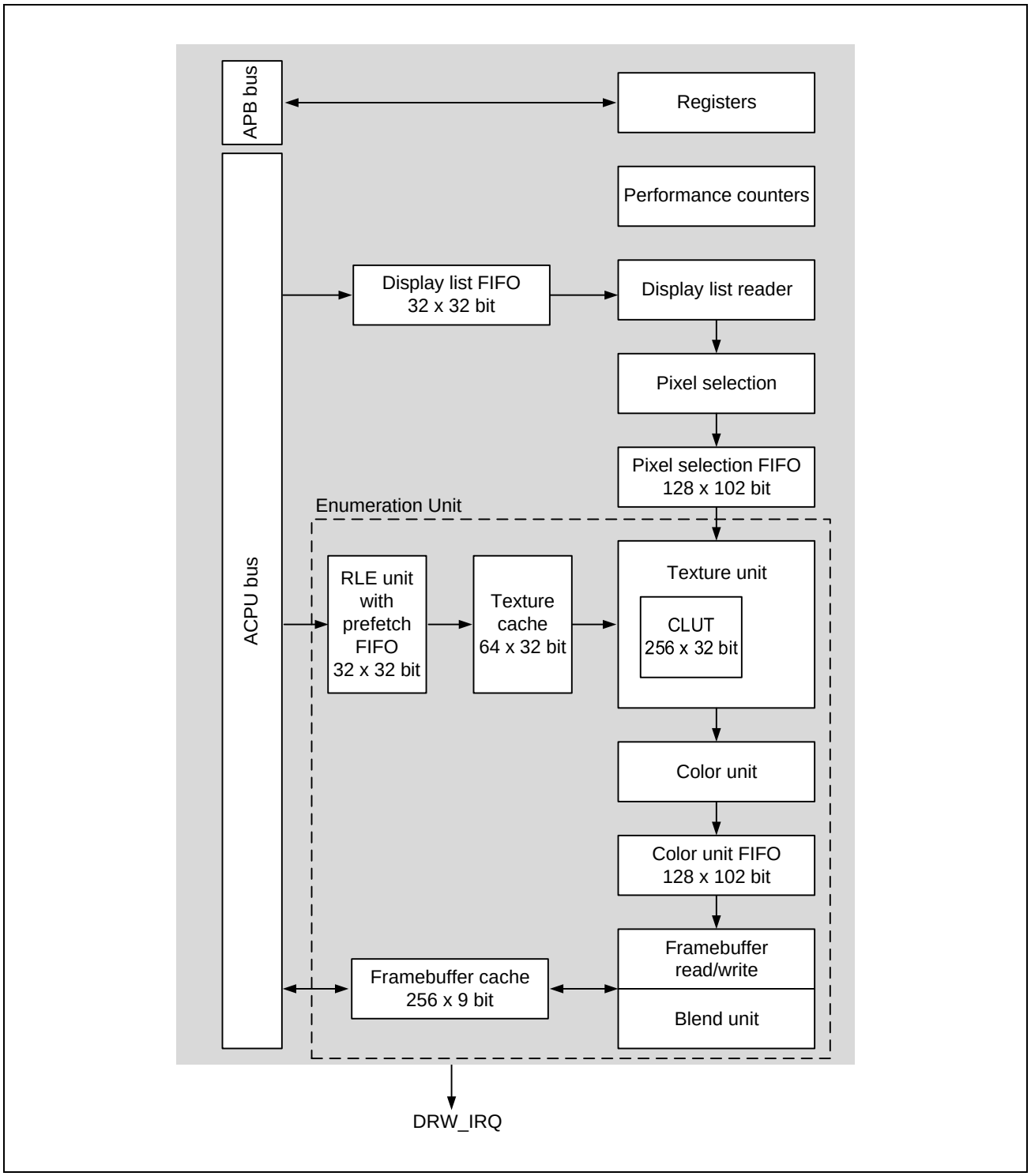



Figure 39.3 Block Diagram of the 2D Drawing Engine

The 2D Drawing Engine accesses the ACPU bus as bus master through separate caches for:

- Reading and writing pixel data from and to the framebuffer
- Reading textures
- Reading display lists

The control registers are accessed through the internal peripheral bus interface.

## 39.2 Register Descriptions

Symbols used in Register Descriptions:

Initial value: Register value after a power-on reset.
R/W: Readable/writable. The written value can be read.
R: Read-only.
W: Write-only.

2D-G (DRW) base address: H'0_11C8_0000 (Overall Address Space)

Table 39.1 Register Configuration (1/2)

| Register Name | Abbreviation | R/W | Offset Address | Initial value | Access Size |
|---|---|---|---|---|---|
| Geometry control register*1 | CONTROL | W | H'A000 | H'0000 0000 | 32 |
| Status control register*1 | STATUS | R | H'A000 | H'0000 0000 | 32 |
| Surface control register*1 | CONTROL2 | W | H'A004 | H'0000 0000 | 32 |
| Hardware version and feature set ID register*1 | HWREVISION | R | H'A004 | H'0FBE 0107 | 32 |
| Limiter 1 start value register | L1START | W | H'A010 | H'0000 0000 | 32 |
| Limiter 2 start value register | L2START | W | H'A014 | H'0000 0000 | 32 |
| Limiter 3 start value register | L3START | W | H'A018 | H'0000 0000 | 32 |
| Limiter 4 start value register | L4START | W | H'A01C | H'0000 0000 | 32 |
| Limiter 5 start value register | L5START | W | H'A020 | H'0000 0000 | 32 |
| Limiter 6 start value register | L6START | W | H'A024 | H'0000 0000 | 32 |
| Limiter 1 x-axis increment register | L1XADD | W | H'A028 | H'0000 0000 | 32 |
| Limiter 2 x-axis increment register | L2XADD | W | H'A02C | H'0000 0000 | 32 |
| Limiter 3 x-axis increment register | L3XADD | W | H'A030 | H'0000 0000 | 32 |
| Limiter 4 x-axis increment register | L4XADD | W | H'A034 | H'0000 0000 | 32 |
| Limiter 5 x-axis increment register | L5XADD | W | H'A038 | H'0000 0000 | 32 |
| Limiter 6 x-axis increment register | L6XADD | W | H'A03C | H'0000 0000 | 32 |
| Limiter 1 y-axis increment register | L1YADD | W | H'A040 | H'0000 0000 | 32 |
| Limiter 2 y-axis increment register | L2YADD | W | H'A044 | H'0000 0000 | 32 |
| Limiter 3 y-axis increment register | L3YADD | W | H'A048 | H'0000 0000 | 32 |
| Limiter 4 y-axis increment register | L4YADD | W | H'A04C | H'0000 0000 | 32 |
| Limiter 5 y-axis increment register | L5YADD | W | H'A050 | H'0000 0000 | 32 |
| Limiter 6 y-axis increment register | L6YADD | W | H'A054 | H'0000 0000 | 32 |
| Limiter 1 bandwidth parameter register | L1BAND | W | H'A058 | H'0000 0000 | 32 |
| Limiter 2 bandwidth parameter register | L2BAND | W | H'A05C | H'0000 0000 | 32 |
| Base color register | COLOR1 | W | H'A064 | H'0000 0000 | 32 |
| Secondary color register | COLOR2 | W | H'A068 | H'0000 0000 | 32 |
| Pattern register | PATTERN | W | H'A074 | H'0000 0000 | 32 |
| Bounding box dimension register | SIZE | W | H'A078 | H'0000 0000 | 32 |
| Framebuffer pitch and spanstore delay register | PITCH | W | H'A07C | H'0000 0000 | 32 |
| Framebuffer base address register | ORIGIN | W | H'A080 | H'0000 0000 | 32 |
| U limiter start value register | LUSTART | W | H'A090 | H'0000 0000 | 32 |
| U limiter x-axis increment register | LUXADD | W | H'A094 | H'0000 0000 | 32 |
| U limiter y-axis increment register | LUYADD | W | H'A098 | H'0000 0000 | 32 |
| V limiter start value integer part register | LVSTARTI | W | H'A09C | H'0000 0000 | 32 |
| V limiter start value fractional part register | LVSTARTF | W | H'A0A0 | H'0000 0000 | 32 |
| V limiter x-axis increment integer part register | LVXADDI | W | H'A0A4 | H'0000 0000 | 32 |

Table 39.1 Register Configuration (2/2)

| Register Name | Abbreviation | R/W | Offset Address | Initial value | Access Size |
|---|---|---|---|---|---|
| V limiter y-axis increment integer part register | LVYADDI | W | H'A0A8 | H'0000 0000 | 32 |
| V limiter increment fractional parts register | LVYXADDF | W | H'A0AC | H'0000 0000 | 32 |
| Texels per texture line register | TEXPITCH | W | H'A0B4 | H'0000 0000 | 32 |
| Texture size or texture address mask register | TEXMASK | W | H'A0B8 | H'0000 0000 | 32 |
| Texture base address register | TEXORIGIN | W | H'A0BC | H'0000 0000 | 32 |
| Interrupt control register | IRQCTL | W | H'A0C0 | H'0000 0000 | 32 |
| Cache control register | CACHECTL | W | H'A0C4 | H'0000 0000 | 32 |
| Display list start address register | DLISTSTART | W | H'A0C8 | H'0000 0000 | 32 |
| Performance counter 1 | PERFCOUNT1 | R/W | H'A0CC | H'0000 0000 | 32 |
| Performance counter 2 | PERFCOUNT2 | R/W | H'A0D0 | H'0000 0000 | 32 |
| Performance counters control register | PERFTRIGGER | W | H'A0D4 | H'0000 0000 | 32 |
| CLUT start address register | TEXCLADDR | W | H'A0DC | H'0000 0000 | 32 |
| CLUT data register | TEXCLDATA | W | H'A0E0 | H'0000 0000 | 32 |
| CLUT offset register | TEXCLOFFSET | W | H'A0E4 | H'0000 0000 | 32 |
| Color key register | COLKEY | W | H'A0E8 | H'0000 0000 | 32 |

Note 1. A write-only register and a read-only register are allocated to a single address.

## 39.2.1 Geometry Control Register (CONTROL)

**Offset Address :** DRW.CONTROL2 H'A004

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | SPANSTORE | SPANABORT | UNIONCD | UNIONAB | UNION56 | UNION34 | UNION12 | BAND2ENABLE |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | BAND1ENABLE | LIM6THRESHOLD | LIM5THRESHOLD | LIM4THRESHOLD | LIM3THRESHOLD | LIM2THRESHOLD | LIM1THRESHOLD | QUAD3ENABLE | QUAD2ENABLE | QUAD1ENABLE | LIM6ENABLE | LIM5ENABLE | LIM4ENABLE | LIM3ENABLE | LIM2ENABLE | LIM1ENABLE |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 24 | — | 0 | W | Reserved.<br>The write value should be 0. |
| 23 | SPANSTORE | 0 | W | Spanstore<br>0: Disable<br>1: Enable<br>Next line span start is always equal to or right of current-line span start. See **Section 39.6.2.6(1), Spanstore**. |
| 22 | SPANABORT | 0 | W | Spanabort<br>0: Disable<br>1: Enable<br>Shape is horizontally convex, only a single span per scan line. See **Section 39.6.2.6(2), Spanabort**. |
| 21 | UNIONCD | 0 | W | Combine Outputs C & D as Union<br>0: Select minimum/intersect between limiters C and D<br>1: Select maximum/union between limiters C and D<br>(Output is final.) |
| 20 | UNIONAB | 0 | W | Combine Outputs A & B as union<br>0: Select minimum/intersect between limiters A and B<br>1: Select maximum/union between limiters A and B<br>(Output is called C.) |
| 19 | UNION56 | 0 | W | Combine Limiter 5 & 6 as Union<br>0: Select minimum/intersect between limiters 1 and 2<br>1: Select maximum/union between limiters 1 and 2<br>(Output is called A.) |
| 18 | UNION34 | 0 | W | Combine Limiter 3 & 4 as Union<br>0: Select minimum/intersect between limiters 3 and 4<br>1: Select maximum/union between limiters 3 and 4<br>(Output is called B.) |
| 17 | UNION12 | 0 | W | Combine Limiter 1 and 2 as Union<br>0: Select minimum/intersect between limiters 1 and 2<br>1: Select maximum/union between limiters 1 and 2<br>(Output is called A.) |
| 16 | BAND2ENABLE | 0 | W | Enable Band Post Process for Limiter 2<br>0: Disable<br>1: Enable<br>(See LnBAND.) |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15 | BAND1ENABLE | 0 | W | Enable Band Post Process for Limiter 1<br>0: Disable<br>1: Enable<br>(See LnBAND.) |
| 14 | LIM6THRESHOLD | 0 | W | Enable Limiter 6 Threshold Mode<br>0: Disable<br>1: Enable |
| 13 | LIM5THRESHOLD | 0 | W | Enable Limiter 5 Threshold Mode<br>0: Disable<br>1: Enable |
| 12 | LIM4THRESHOLD | 0 | W | Enable Limiter 4 Threshold Mode<br>0: Disable<br>1: Enable |
| 11 | LIM3THRESHOLD | 0 | W | Enable Limiter 3 Threshold Mode<br>0: Disable<br>1: Enable |
| 10 | LIM2THRESHOLD | 0 | W | Enable Limiter 2 Threshold Mode<br>0: Disable<br>1: Enable |
| 9 | LIM1THRESHOLD | 0 | W | Enable Limiter 1 Threshold Mode<br>0: Disable<br>1: Enable |
| 8 | QUAD3ENABLE | 0 | W | Enable Quadratic Coupling of Limiters 5 and 6<br>0: Disable<br>1: Enable |
| 7 | QUAD2ENABLE | 0 | W | Enable Quadratic Coupling of Limiters 3 and 4<br>0: Disable<br>1: Enable |
| 6 | QUAD1ENABLE | 0 | W | Enable Quadratic Coupling of Limiters 1 and 2<br>0: Disable<br>1: Enable |
| 5 | LIM6ENABLE | 0 | W | Enable Limiter 6<br>0: Disable<br>1: Enable |
| 4 | LIM5ENABLE | 0 | W | Enable Limiter 5<br>0: Disable<br>1: Enable |
| 3 | LIM4ENABLE | 0 | W | Enable Limiter 4<br>0: Disable<br>1: Enable |
| 2 | LIM3ENABLE | 0 | W | Enable Limiter 3<br>0: Disable<br>1: Enable |
| 1 | LIM2ENABLE | 0 | W | Enable Limiter 2<br>0: Disable<br>1: Enable |
| 0 | LIM1ENABLE | 0 | W | Enable Limiter 1<br>0: Disable<br>1: Enable |

## 39.2.2 Surface Control Register (CONTROL2)

**Offset Address :** DRW.CONTROL2 H'A004

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | RLEPIXELWIDTH[1:0] | | BDIA | BSIA | CLUTFORMAT | COLKEYENABLE | CLUTENABLE | RLEENABLE | WRITEALPHA[1:0] | | WRITEFORMAT[1:0] | | READFORMAT[1:0] | | TEXTUREFILTERY | TEXTUREFILTERX |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TEXTURECLAMPY | TEXTURECLAMPX | BC2 | BDI | BSI | BDF | BSF | WRITEFORMAT[2] | BDFA | BSFA | READFORMAT[3:2] | | USEACB | PATTERNSOURCEL5 | TEXTUREENABLE | PATTERNENABLE |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31, 30 | RLEPIXELWIDTH[1:0] | 0 | W | Texel Width for RLE Unit<br>b31 b30<br>0 0: 1 byte per texel<br>0 1: 2 bytes per texel<br>1 0: 3 bytes per texel<br>1 1: 4 bytes per texel. |
| 29 | BDIA | 0 | W | Blend Destination Factor Inverted in Alpha Channel<br>• In alpha channel blending mode (USEACB = 1):<br>0: Use blend factor as specified through BDFA<br>1: Invert destination blend factor (1-x) |
| 28 | BSIA | 0 | W | Blend Source Factor Inverted in Alpha Channel<br>• In alpha channel blending mode (USEACB = 1):<br>0: Use blend factor as specified through BSFA<br>1: Invert blend source factor (1-x) |
| 27 | CLUTFORMAT | 0 | W | CLUT Format<br>0: Format CLUT as aRGB (8888)<br>1: Format CLUT as RGB (565) |
| 26 | COLKEYENABLE | 0 | W | Color Keying Enable<br>0: Disable color keying<br>1: Enable color keying |
| 25 | CLUTENABLE | 0 | W | CLUT Enable<br>0: Disable CLUT<br>1: Enable CLUT<br>If CLUTENABLE = 0 (CLUT disabled), the index is directly put on the RGB channels. |
| 24 | RLEENABLE | 0 | W | RLE Enable<br>0: Disable RLE<br>1: Enable RLE |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 23, 22 | WRITEALPHA[1:0] | 0 | W | Writeback Alpha Source for Framebuffer<br>• In non-alpha channel blending mode (USEACB = 0):<br>Sets the alpha source for the framebuffer.<br>b23 b22<br>0 0: Use alpha from color 2<br>0 1: Use source alpha (pixel coverage)<br>1 0: Use 0.0 as alpha<br>1 1: Use alpha from framebuffer<br>• In alpha channel blending mode (USEACB = 1):<br>Blends alpha in color 2 instead of framebuffer alpha.<br>b23 b22<br>0 0: BC2A = 1: Use alpha in color 2 as destination (DST_A)<br>Else: BC2A = 0: Use alpha from framebuffer as destination (DST_A) |
| 21, 20 | WRITEFORMAT[1:0] | 0 | W | Writeback Framebuffer Format<br>Pixel format of the framebuffer.<br>b8 b21 b20<br>0 0 0: 8 bpp a (8)<br>0 0 1: 16 bpp RGB (565)<br>0 1 0: 32 bpp aRGB (8888)<br>0 1 1: 16 bpp aRGB (4444)<br>Other settings are prohibited. |
| 19, 18 | READFORMAT[1:0] | 0 | W | Texture Format Descriptor<br>Pixel format of the texture buffer.<br>b5 b4 b19 b18<br>0 0 0 0: 8 bpp a (8)<br>0 0 0 1: 16 bpp RGB (565)<br>0 0 1 0: 32 bpp aRGB (8888)<br>0 0 1 1: 16 bpp aRGB (4444)<br>0 1 0 0: 16 bpp aRGB (1555)<br>0 1 0 1: 8 bpp aCLUT (44), 4 bit alpha and 4 bit indexed color<br>1 0 0 1: 8 bpp CLUT (8)/I (8), 8 bit indexed color/luminance<br>1 0 1 0: 4 bpp CLUT (4)/I (4), 4 bit indexed color/luminance<br>1 0 1 1: 2 bpp CLUT (2)/I (2), 2 bit indexed color/luminance<br>1 1 0 0: 1 bpp CLUT (1)/I (1), 1 bit indexed color/luminance.<br>Other settings are prohibited. |
| 17 | TEXTUREFILTERY | 0 | W | Linear Filtering on Texture V Axis<br>0: No filtering on texture V axis<br>1: Linear filtering on texture V axis |
| 16 | TEXTUREFILTERX | 0 | W | Linear Filtering on Texture U Axis<br>0: No filtering on texture u axis<br>1: Linear filtering on texture u axis |
| 15 | TEXTURECLAMPY | 0 | W | Calculating V Limiter Outside Used Texture<br>This bit describes what happens when the V limiter (y direction in texture space) calculates a v value outside of the used texture.<br>0: Texture wrap mode: Integer part of the calculated value from the V limiter is AND gated with TEXVMASK, resulting in a repetition of texture in the y/v direction<br>1: Texture clamp mode: Texture color at the border of the texture is taken, resulting in a repetition of texture border color in the y/v direction |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 14 | TEXTURECLAMPX | 0 | W | Calculating U Limiter Outside Used Texture<br>This bit describes what happens when the U limiter (x direction in texture space) calculates a u value outside of the used texture.<br>0: Texture wrap mode: Integer part of the calculated value from the U limiter is AND gated with TEXUMASK, resulting in a repetition of texture in the x/u direction<br>1: Texture clamp mode: Texture color at the border of the texture is taken, resulting in a repetition of texture border color in the x/u direction |
| 13 | BC2 | 0 | W | Blend color 2<br>Select of blend color 2 instead of framebuffer pixel.<br>0: Use pixel from framebuffer as destination (DST)<br>1: Use color 2 as destination (DST) |
| 12 | BDI | 0 | W | Blend Destination Factor Inverted<br>Destination factor is inverted (meaning 1-a or 1-1 depending on BDF).<br>0: Use blend factor as specified through BDF<br>1: Invert blend destination factor (1-x) |
| 11 | BSI | 0 | W | Blend Source Factor Inverted<br>Source factor is inverted (meaning 1-a or 1-1 depending on BSF).<br>0: Use blend factor as specified through BSF<br>1: Invert blend source factor (1-x) |
| 10 | BDF | 0 | W | Blend Destination Factor<br>Destination factor is alpha (factor is 1 per default).<br>0: Use 1.0 as blend destination factor<br>1: Use alpha as blend destination factor |
| 9 | BSF | 0 | W | Blend Source Factor<br>Source factor is alpha (factor is 1 per default).<br>0: Use 1.0 as blend source factor<br>1: Use alpha as blend source factor |
| 8 | WRITEFORMAT[2] | 0 | W | Writeback Framebuffer Format<br>Bit [2] of framebuffer pixel format. See the description of WRITEFORMAT[1:0] in this section. |
| 7 | BDFA | 0 | W | Blend Destination Factor for Alpha Channel<br>Valid in alpha channel blending mode (USEACB = 1).<br>0: Use 1.0 as blend destination factor for alpha channel<br>1: Use alpha as blend destination factor for alpha channel |
| 6 | BSFA | 0 | W | Blend Source Factor for Alpha Channel<br>Valid in alpha channel blending mode (USEACB = 1).<br>0: Use 1.0 as blend source factor for alpha channel<br>1: Use alpha as blend source factor for alpha channel |
| 5, 4 | READFORMAT[3:2] | 0 | W | Texture Format Descriptor<br>Bits [3] and [2] of the texture buffer format. See the detailed description of the READFORMAT[1:0] bit in this section. |
| 3 | USEACB | 0 | W | Alpha Blend Mode<br>0: Use WRITEALPHA[1:0] mode<br>1: Use full alpha channel blending mode |
| 2 | PATTERNSOURCEL5 | 0 | W | Limiter 5 Enable for Pattern Index<br>Limiter 5 is used as pattern index instead of the default U limiter. Limiter 5 can be combined with limiter 6 to form a quadratic limiter that can be used to make quadratic pattern functions to draw radial patterns. |
| 1 | TEXTUREENABLE | 0 | W | Texture Enable for Pixel Source<br>Pixel source is read from texture and used as an alpha to blend between COLOR1 and COLOR2.<br>0: Disable texture<br>1: Enable texture |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 0 | PATTERNENABLE | 0 | W | Pattern Color Enable for Pixel Source<br>Pixel source is a pattern color (blend of COLOR1 and COLOR2 depending on PATTERN and pattern index).<br>0: Disable pattern<br>1: Enable pattern<br>When patterns are used to fill a primitive an index into the pattern bit mask is generated for each pixel with the U limiter. Depending on the pattern bits the color is selected from COLOR1 and COLOR2 registers. Fractional indices can be interpolated between those two values by using TEXTUREFILTERX = 1. The pattern can be wrapped by using TEXTURECLAMPX = 0, and the mask must be set in the TEXMASK register using the mask for u. |

### 39.2.3 Interrupt Control Register (IRQCTL)

**Offset Address :** DRW.IRQCTL H'A0C0

| 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 | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | BUSIRQCLR | BUSIRQEN | DLISTIRQCLR | ENUMIRQCLR | DLISTIRQEN | ENUMIRQEN |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 6 | — | 0 | W | Reserved.<br>The write value should be 0. |
| 5 | BUSIRQCLR | 0 | W | Clear BUSIRQ<br>0: Do not clear BUSIRQ bus error interrupt<br>1: Clear BUSIRQ bus error interrupt |
| 4 | BUSIRQEN | 0 | W | BUSIRQ Interrupt Mask Enable<br>0: Disable (mask) BUSIRQ bus error interrupt<br>1: Enable (unmask) BUSIRQ bus error interrupt |
| 3 | DLISTIRQCLR | 0 | W | Clear DLISTIRQ<br>0: Do not clear DLISTIRQ display list interrupt<br>1: Clear DLISTIRQ display list interrupt |
| 2 | ENUMIRQCLR | 0 | W | Clear ENUMIRQ<br>0: Do not clear ENUMIRQ enumeration interrupt<br>1: Clear ENUMIRQ enumeration interrupt |
| 1 | DLISTIRQEN | 0 | W | DLISTIRQ Interrupt Mask Enable<br>0: Disable (mask) DLISTIRQ display list interrupt<br>1: Enable (unmask) DLISTIRQ display list interrupt |
| 0 | ENUMIRQEN | 0 | W | ENUMIRQ Interrupt Mask Enable<br>0: Disable (mask) ENUMIRQ enumeration interrupt<br>1: Enable (unmask) ENUMIRQ enumeration interrupt |

## 39.2.4 Cache Control Register (CACHECTL)

**Offset Address :** DRW.CACHECTL H'A0C4

| 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 | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | CFLUSHTX | CENABLETX | CFLUSHFX | CENABLEFX |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 4 | — | 0 | W | Reserved.<br>The write value should be 0. |
| 3 | CFLUSHTX | 0 | W | Flush Texture Cache<br>0: Do not flush the texture cache<br>1: Flush the texture cache |
| 2 | CENABLETX | 0 | W | Texture Cache Enable<br>0: Disable the texture cache<br>1: Enable the texture cache |
| 1 | CFLUSHFX | 0 | W | Flush Framebuffer Cache<br>0: Do not flush the framebuffer cache<br>1: Flush the framebuffer cache |
| 0 | CENABLEFX | 0 | W | Framebuffer Cache Enable<br>0: Disable the framebuffer cache<br>1: Enable the framebuffer cache |

## 39.2.5 Status Control Register (STATUS)

**Offset Address :** DRW.STATUS H'A000

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | BUSERRMDL | BUSERRMTXMRL | BUSERRMFB | — | BUSIRQ | DLISTIRQ | ENUMIRQ | DLISTACTIVE | CACHEDIRTY | BUSYWRITE | BUSYENUM |
| 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 11 | — | 0 | R | Reserved.<br>These bits are read as 0. |
| 10 | BUSERRMDL | 0 | R | Display List Bus Error Interrupt Triggered<br>0: No display list bus error occurred or interrupt disabled<br>1: Display list bus error interrupt triggered |
| 9 | BUSERRMTXMRL | 0 | R | Texture Bus Error Interrupt Triggered<br>0: No texture bus error occurred or interrupt disabled<br>1: Texture bus error interrupt triggered*1 |
| 8 | BUSERRMFB | 0 | R | Framebuffer Bus Error Interrupt Triggered<br>0: No framebuffer bus error occurred or interrupt disabled<br>1: Framebuffer bus error interrupt triggered |
| 7 | — | 0 | R | Reserved.<br>This bit is read as 0. |
| 6 | BUSIRQ | 0 | R | Bus Error Interrupt Triggered<br>0: No bus error occurred or interrupt disabled<br>1: Bus error interrupt triggered |
| 5 | DLISTIRQ | 0 | R | Display List Interrupt Triggered<br>0: Display list not finished or interrupt disabled<br>1: Display list finished interrupt triggered |
| 4 | ENUMIRQ | 0 | R | Enumeration Interrupt Triggered<br>0: Enumeration not finished or interrupt disabled<br>1: Enumeration finished interrupt triggered |
| 3 | DLISTACTIVE | 0 | R | Display List Reader Status<br>0: Display list reader is idle<br>1: Display list reader busy, and no direct write access to registers allowed |
| 2 | CACHEDIRTY | 0 | R | Framebuffer Cache Status<br>0: Framebuffer cache is not dirty<br>1: Framebuffer cache is dirty, and frame should not be flipped |
| 1 | BUSYWRITE | 0 | R | Framebuffer Writeback Status<br>0: Framebuffer writeback finished<br>1: Framebuffer writeback busy, framebuffer type cannot be changed |
| 0 | BUSYENUM | 0 | R | Enumeration Unit Status<br>0: Enumeration unit idle<br>1: Enumeration unit busy, new primitive cannot be started |

Note 1. Because the RLE unit is also reading data through the texture bus, an error during RLE data access is also reflected in this bit.

## 39.2.6 Hardware Version and Feature Set ID Register (HWREVISION)

**Offset Address :** DRW.HWREVISION H'A004

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | ACBLEND | — | COLORKEY | TEXCLUT256 | RLEUNIT | — | TEXCLU | PERFCOUNT | TXCACHE | FBCACHE | DLR | — |
| Initial Value | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | REV[11:0] | | | | | | | | | | | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 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 28 | — | 0 | R | Reserved.<br>These bits are read as 0. |
| 27 | ACBLEND | 1 | R | Alpha Channel Blending Available<br>Alpha channel blending is available. |
| 26 | — | 1 | R | Reserved.<br>This bit is read as 1. |
| 25 | COLORKEY | 1 | R | Color Key Available<br>Color key is available. |
| 24 | TEXCLUT256 | 1 | R | Texture CLUT Available<br>Texture CLUT is available. |
| 23 | RLEUNIT | 1 | R | RLE Unit Available<br>RLE unit is available. |
| 22 | — | 0 | R | Reserved.<br>This bit is read as 0. |
| 21 | TECLUT | 1 | R | Texture CLUT with 16 or 256 Entries Available<br>Texture CLUT with 16 or 256 entries is available. |
| 20 | PERFCOUNT | 1 | R | Two Performance Counter Available<br>Two performance counter is available. |
| 19 | TXCACHE | 1 | R | Texture Cache Available<br>Texture cache is available. |
| 18 | FBCACHE | 1 | R | Framebuffer Cache Available<br>Framebuffer cache is available. |
| 17 | DLR | 1 | R | Display List Reader Available<br>Display list reader is available. |
| 16 to 12 | — | 0 | R | Reserved.<br>These bits are read as 0. |
| 11 to 9 | REV[11:0] | 0 | R | Revision Number<br>Revision number. |
| 8 | REV[11:0] | 1 | R | Revision Number<br>Revision number. |
| 7 to 3 | REV[11:0] | 0 | R | Revision Number<br>Revision number. |
| 2 to 0 | REV[11:0] | 1 | R | Revision Number<br>Revision number. |

## 39.2.7 Base Color Register (COLOR1)

**Offset Address :** DRW.COLOR1 H'A064

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | COLOR1A[7:0] | | | | | | | | COLOR1R[7:0] | | | | | | | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | COLOR1G[7:0] | | | | | | | | COLOR1B[7:0] | | | | | | | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 24 | COLOR1A [7:0] | 0 | W | Alpha Channel of Color 1<br>Specifies alpha channel of color 1.<br>H'00: Transparent<br>…<br>H'FF: Opaque |
| 23 to 16 | COLOR1R [7:0] | 0 | W | Red Channel of Color 1<br>Specifies red channel of color 1. |
| 15 to 8 | COLOR1G [7:0] | 0 | W | Green Channel of Color 1<br>Specifies green channel of color 1. |
| 7 to 0 | COLOR1B [7:0] | 0 | W | Blue Channel of Color 1<br>Specifies blue channel of color 1. |

## 39.2.8 Secondary Color Register (COLOR2)

**Offset Address :** DRW.COLOR2 H'A068

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | COLOR2A[7:0] | | | | | | | | COLOR2R[7:0] | | | | | | | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | COLOR2G[7:0] | | | | | | | | COLOR2B[7:0] | | | | | | | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 24 | COLOR2A [7:0] | 0 | W | Alpha Channel of Color 2<br>Specifies alpha channel of color 2.<br>H'00: Transparent<br>…<br>H'FF: Opaque. |
| 23 to 16 | COLOR2R [7:0] | 0 | W | Red Channel of Color 2<br>Specifies red channel of color 2. |
| 15 to 8 | COLOR2G [7:0] | 0 | W | Green Channel of Color 2<br>Specifies green channel of color 2. |
| 7 to 0 | COLOR2B [7:0] | 0 | W | Blue Channel of Color 2<br>Specifies blue channel of color 2. |

## 39.2.9 Pattern Register (PATTERN)

**Offset Address :** DRW.PATTERN H'A074

| 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 | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | PATTERN[7:0] | | | | | | | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 8 | — | 0 | W | Reserved.<br>The write value should be 0. |
| 7 to 0 | PATTERN [7:0] | 0 | W | Bitmap of the Pattern<br>Specifies bitmap of the pattern. |

## 39.2.10 Limiter N Start Value Register (LnSTART)

**Offset Address :** DRW.L1START H'A010, DRW.L2START H'A014, DRW.L3START H'A018, DRW.L4START H'A01C, DRW.L5START H'A020, DRW.L6START H'A024

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

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

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 0 | LSTART[31:0] | 0 | W | Start Value of the nth Limiter<br>Specifies start value of the nth limiter. |

## 39.2.11 Limiter N X-Axis Increment Register (LnXADD)

**Offset Address :** DRW.L1XADD H'A028, DRW.L2XADD H'A02C, DRW.L3XADD H'A030, DRW.L4XADD H'A034, DRW.L5XADD H'A038, DRW.L6XADD H'A03C

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

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

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 0 | LXADD[31:0] | 0 | W | X-Axis Increment<br>Specifies x-axis increment. |

## 39.2.12 Limiter N Y-Axis Increment Register (LnYADD)

**Offset Address :** DRW.L1YADD H'A040, DRW.L2YADD H'A044, DRW.L3YADD H'A048, DRW.L4YADD H'A04C, DRW.L5YADD H'A050, DRW.L6YADD H'A054

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

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

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 0 | LYADD[31:0] | 0 | W | Y-Axis Increment<br>Specifies y-axis increment. |

## 39.2.13 Limiter M Bandwidth Parameter Register (LmBAND)

**Offset Address :** DRW.L1BAND H'A058, DRW.L2BAND H'A05C

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

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

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 0 | LBAND[31:0] | 0 | W | Limiter m Bandwidth Parameter<br>Specifies limiter m bandwidth parameter. |

## 39.2.14 Texture Base Address Register (TEXORIGIN)

**Offset Address :** DRW.TEXORIGIN H'A0BC

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

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

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 0 | TEXORIGIN [31:0] | 0 | W | Texture Base Address<br>Specifies texture base address. |

## 39.2.15 Texels Per Texture Line Register (TEXPITCH)

**Offset Address :** DRW.TEXPITCH H'A0B4

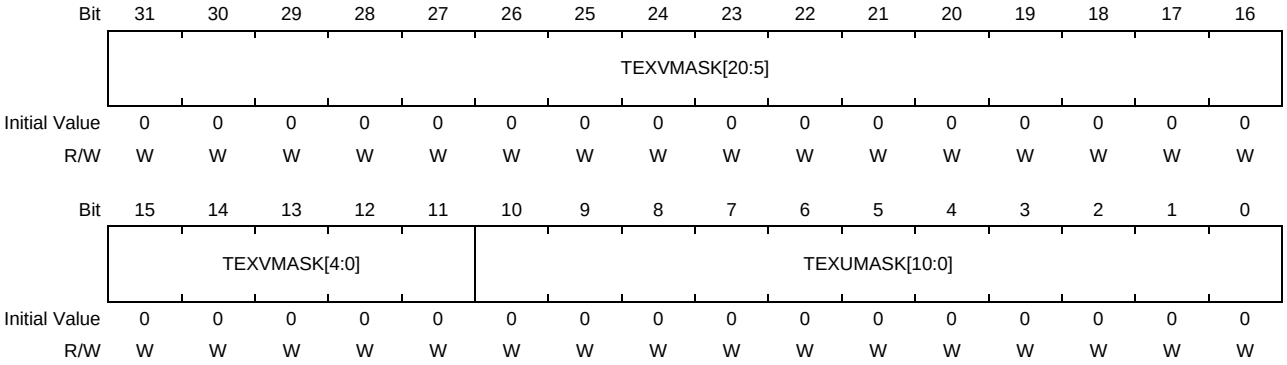
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0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

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | TEXPITCH[10:0] | | | | | | | | | | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 11 | — | 0 | W | Reserved.<br>The write value should be 0. |
| 10 to 0 | TEXPITCH [10:0] | 0 | W | Texels Per Texture Line<br>Specifies texels per texture line. |

## 39.2.16 Texture Size or Texture Address Mask Register (TEXMASK)

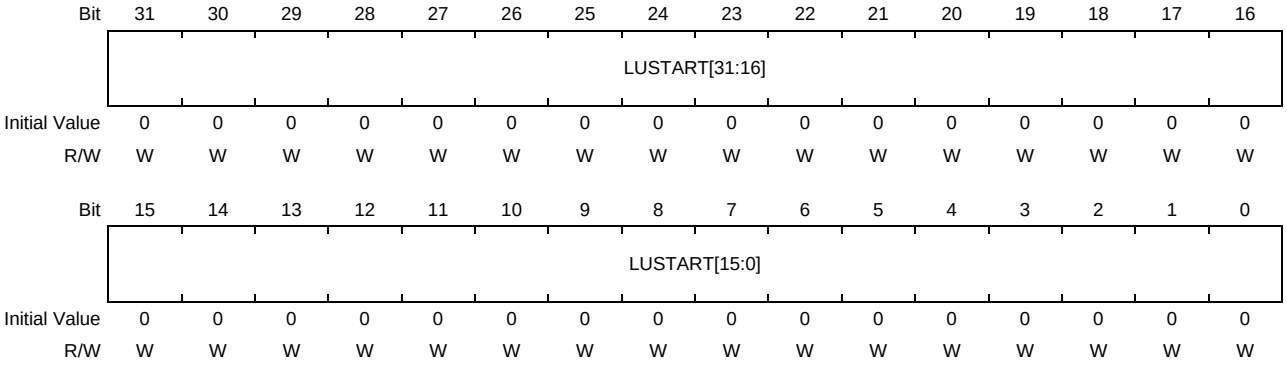
**Offset Address :** DRW.TEXMASK H'A0B8




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 11 | TEXVMASK [20:0] | 0 | W | V Mask in Texture Mode The write value should be 0.<br>Specifies the V mask. Set to TEXPITCH × (texture_height 1)<br>• In texture wrapping mode (CONTROL2.TEXTURECLAMPY = 0): Texture_height must be a power of 2<br>• In texture clamping mode (CONTROL2.TEXTURECLAMPY = 1): All heights up to 1024 are allowed. |
| 10 to 0 | TEXUMASK [10:0] | 0 | W | U Mask in Texture Mode<br>Specifies the U mask. Set to texture_width 1.<br>• In texture wrapping mode (CONTROL2.TEXTURECLAMPX = 0): Texture_width must be a power of 2<br>• In texture clamping mode (CONTROL2.TEXTURECLAMPX = 1): All widths up to 2048 are allowed. |

## 39.2.17 U Limiter Start Value Register (LUSTART)

**Offset Address :** DRW.LUSTART H'A090




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 0 | LUSTART [31:0] | 0 | W | U Limiter Start Value<br>Specifies U limiter start value. |

## 39.2.18 U Limiter X-Axis Increment Register (LUXADD)

**Offset Address :** DRW.LUXADD H'A094

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

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

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 0 | LUXADD [31:0] | 0 | W | U Limiter X-Axis Increment<br>Specifies U limiter x-axis increment. |

## 39.2.19 U Limiter Y-Axis Increment Register (LUYADD)

**Offset Address :** DRW.LUYADD H'A098

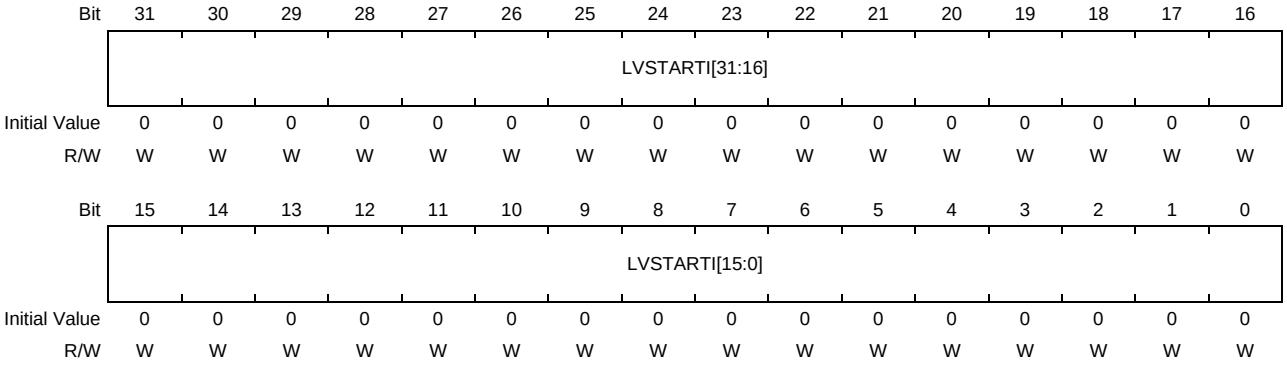
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | LUYADD[31:16] | | | | | | | | | | | | | | | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | W |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

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

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 0 | LUYADD [31:0] | 0 | W | U Limiter Y-Axis Increment<br>Specifies U limiter y-axis increment. |

## 39.2.20 V Limiter Start Value Integer Part Register (LVSTARTI)

**Offset Address :** DRW.LVSTARTI H'A09C




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 0 | LVSTARTI [31:0] | 0 | W | V Limiter Start Value Integer Part<br>Specifies integer part of V limiter start value. |

## 39.2.21 V Limiter Start Value Fractional Part Register (LVSTARTF)

**Offset Address :** DRW.LVSTARTF H'A0A0

| 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 | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W |

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

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 16 | — | 0 | W | Reserved.<br>The write value should be 0. |
| 15 to 0 | LVSTARTF [15:0] | 0 | W | V Limiter Start Value Fractional Part<br>Specifies fractional part of V limiter start value. |

## 39.2.22 V Limiter X-Axis Increment Integer Part Register (LVXADDI)

**Offset Address :** DRW.LVXADDI H'A0A4

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

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

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 0 | LVXADDI [31:0] | 0 | W | V Limiter X-Axis Increment Integer Part<br>Specifies integer part of V limiter x-axis increment. |

## 39.2.23 V Limiter Y-Axis Increment Integer Part Register (LVYADDI)

**Offset Address :** DRW.LVYADDI H'A0A8

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

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

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 0 | LVYADDI [31:0] | 0 | W | V Limiter Y-Axis Increment Integer Part<br>Specifies integer part of V limiter y-axis increment. |

## 39.2.24 V Limiter Increment Fractional Parts Register (LVYXADDF)

**Offset Address :** DRW.LVYXADDF H'A0AC

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | LVYADDF[15:0] | | | | | | | | | | | | | | | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W |

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

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 16 | LVYADDF [15:0] | 0 | W | V Limiter X-Axis Increment Fractional Part<br>Specifies fractional part of V limiter x-axis increment. |
| 15 to 0 | LVXADDF [15:0] | 0 | W | V Limiter Y-Axis Increment Fractional Part<br>Specifies fractional part of V limiter y-axis increment. |

## 39.2.25 CLUT Start Address Register (TEXCLADDR)

**Offset Address :** DRW.TEXCLADDR H'A0DC

| 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 | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | CLADDR[7:0] | | | | | | | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 8 | — | 0 | W | Reserved.<br>The write value should be 0. |
| 7 to 0 | CLADDR[7:0] | 0 | W | Texture CLUT Start Address<br>Specifies texture CLUT start address. |

## 39.2.26 CLUT Data Register (TEXCLDATA)

**Offset Address :** DRW.TEXCLDATA H'A0E0

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

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

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 0 | CLDATA [31:0] | 0 | W | Texture CLUT Data<br>Specifies texture CLUT data. |

## 39.2.27 CLUT Offset Register (TEXCLOFFSET)

**Offset Address :** DRW.TEXCLOFFSET H'A0E4

| 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 | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | CLOFFSET[7:0] | | | | | | | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 8 | — | 0 | W | Reserved.<br>The write value should be 0. |
| 7 to 0 | CLOFFSET [7:0] | 0 | W | Texture CLUT Offset<br>Specifies texture CLUT offset. CLOFFSET[7:0] is OR gated with the original index. |

## 39.2.28 Color Key Register (COLKEY)

**Offset Address :** DRW.COLKEY H'A0E8

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | COLKEYR[7:0] | | | | | | | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | COLKEYG[7:0] | | | | | | | | COLKEYB[7:0] | | | | | | | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 24 | — | 0 | W | Reserved.<br>The write value should be 0. |
| 23 to 16 | COLKEYR [7:0] | 0 | W | Red Channel of Color Key<br>Specifies red channel of color key. |
| 15 to 8 | COLKEYG [7:0] | 0 | W | Green Channel of Color Key<br>Specifies green channel of color key. |
| 7 to 0 | COLKEYB [7:0] | 0 | W | Blue Channel of Color Key<br>Specifies blue channel of color key. |

## 39.2.29 Bounding Box Dimension Register (SIZE)

**Offset Address :** DRW.SIZE H'A078

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | SIZEY[15:0] | | | | | | | | | | | | | | | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W |

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

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 16 | SIZEY[15:0] | 0 | W | Bounding Box Height<br>Specifies height of bounding box in pixels. Valid range: 0 to 1024. |
| 15 to 0 | SIZEX[15:0] | 0 | W | Bounding Box Width<br>Specifies width of bounding box in pixels. Valid range: 0 to 1024. |

## 39.2.30 Framebuffer Pitch And Spanstore Delay Register (PITCH)

**Offset Address :** DRW.PITCH H'A07C

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | SSD[15:0] | | | | | | | | | | | | | | | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W |

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

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 16 | SSD[15:0] | 0 | W | Spanstore Delay<br>Specifies number of scan lines to delay spanstore operations. |
| 15 to 0 | PITCH[15:0] | 0 | W | Pitch of the Framebuffer<br>A negative width can be used to render bottom-up instead of top-down. |

## 39.2.31 Framebuffer Base Address Register (ORIGIN)

**Offset Address :** DRW.ORIGIN H'A080

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

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

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 0 | ORIGIN[31:0] | 0 | W | Address of the First Pixel in Framebuffer<br>Writing to ORIGIN triggers the start of rendering. |

## 39.2.32 Display List Start Address Register (DLISTSTART)

**Offset Address :** DRW.DLISTSTART H'A0C8

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

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

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 0 | DLISTSTART [31:0] | 0 | W | Display List Start Address<br>Setting a new display list base address triggers execution of the new display list. Execution stops only when a new list is set or the current list terminates. |

## 39.2.33 Performance Counters Control Register (PERFTRIGGER)

**Offset Address :** DRW.PERFTRIGGER H'A0D4

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | PERFTRIGGER2[4:0] | | | | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W |

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | PERFTRIGGER1[4:0] | | | | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| R/W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W | W |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 21 | — | 0 | W | Reserved.<br>The write value should be 0. |
| 20 to 16 | PERFTRIGGER2[4:0] | 0 | W | Trigger of Performance Counter 2<br>Same as for PERFTRIGGER1, but for performance counter 2. |
| 15 to 5 | — | 0 | W | Reserved.<br>The write value should be 0. |
| 4 to 0 | PERFTRIGGER1[4:0] | 0 | W | Trigger of Performance Counter 1<br>Selects the internal event that increments the PERFCOUNT1 register.<br>0: Disable performance counter<br>1: Select 2D Drawing Engine active cycles<br>2: Select framebuffer read access<br>3: Select framebuffer write access<br>4: Select texture read access<br>5: Select invisible pixels (enumerated but selected with alpha 0%)<br>6: Select invisible pixels while internal FIFO is empty (lost cycles)<br>7: Select display list reader active cycles<br>8: Select framebuffer read hits<br>9: Select framebuffer read misses<br>10: Select framebuffer write hits<br>11: Select framebuffer write misses<br>12: Select texture read hits<br>13: Select texture read misses<br>31: Select every clock cycle (for use as timer). |

## 39.2.34 Performance Counter k (PERFCOUNTk) (k = 1, 2)

**Offset Address :** DRW.PERFCOUNT1 H'A0CC, DRW.PERFCOUNT2 H'A0D0

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PERFCOUNT[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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PERFCOUNT[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 | PERFCOUNT [31:0] | 0 | R/W | Performance Counter k Value<br>Specifies counter k value.<br>The counter is reset by writing PERFCOUNTk = H'0000 0000. |

# 39.3 Drawing Features

## 39.3.1 Drawing Features Summary

### 39.3.1.1 Color Formats

Supported color formats are:

**Framebuffer formats**

- 8-bit: a (8)
- 16-bit: RGB (565), aRGB (4444)
- 32-bit: aRGB (8888).

**Texture formats**

- 1-bit: CLUT (1)/I (1)
- 2-bit: CLUT (2)/I (2)
- 4-bit: CLUT (4)/I (4)
- 8-bit: a (8), CLUT (8)/I (8), aCLUT (44)
- 16-bit: aRGB (4444), aRGB (1555), RGB (565)
- 24-bit: RGB (888) (run length encoded (RLE) unit)
- 32-bit: aRGB (8888).

CLUT formats use a 256-entry color lookup table.

### 39.3.1.2 BitBLT Features

The 2D Drawing Engine supports the BitBLT features using its vector drawing function to draw a rectangle and texture it based on the selected BitBLT function. This approach results in the following BitBLT features:

- Fill
- Copy
- Stretch BitBLT
- Rotate and scale
- Alpha blending
- Bilinear filtering
- Color conversion
- Subpixel exact placement.

#### 39.3.1.3 Vector Drawing Features

The 2D Drawing Engine for vectors uses a half-plane rendering approach, which simplifies the implementation of the edge antialiasing and blurring features without much overhead.

When combining some of its functional units, this module can draw not only linear primitives such as lines or polygons but also quadratic equation-based primitives such as circles and ellipsoids.

The following primitives are supported:

- Lines
- Polygons
- Circles and ellipses
- Quadratic curves (software driver support)
- 2D texture mapping
- Bilinear filtering of the textures.

### 39.3.2 Vector Drawing

For a detailed explanation of the algorithms, see **Section 39.6, Rendering Pipeline**. Supported vector drawing includes:

#### Lines

- Arbitrary width
- Round endpoints
- Truncated endpoints
- Alpha gradients
- Soft edges (blurring)
- Render attribute: color, pattern, or texture.

#### Polygons

- Triangles and quadrangles (complex polygons are tessellated by software)
- Alpha gradients
- Soft edges (blurring)
- Per edge controls for anti-aliasing
- Render attribute: color, pattern, or texture.

#### Circles and ellipses

- All conic sections
- Filled or with arbitrary width
- Arcs of 0° to 360°
- Soft edges
- Alpha gradients
- Render attribute: color, pattern, or texture.

#### Quadratic Bézier

- Approximated by circle arcs
- Arbitrary width
- Round or truncated endpoints
- Outlines, blurring
- Alpha gradients
- Render attribute: color.

#### Texture mapping

- 2D array of pixels that can be mapped implicitly or explicitly on all primitives provided by the 2D Drawing Engine
- Translation, rotation, and scaling/shearing
- Bilinear filtering of the textures
- 3D-like texturing accomplished with line-by-line mapping if either the x or y coordinate value is a constant.

### 39.3.3 BitBLT

A dedicated BitBLT unit is not required in the 2D Drawing Engine. The rendering pipeline described for vector drawing is used as the BitBLT unit and already provides a 1 pixel/cycle throughput. For details, see **Section 39.6, Rendering Pipeline**.

#### 39.3.3.1 Fill

Any rectangle in the framebuffer can be filled with any value. Possible color formats are any 8-, 16-, or 32-bpp format.

The driver optimizes the fill to gain the full benefit of 32-bit parallel rasterization. If the color depth of the selected color format is less than 32 bpp, the driver corrects the alignment and fills 32 bits per clock cycle, resulting in two to four times faster fill performance in 8- and 16-bpp formats.

#### 39.3.3.2 Copy

Any rectangle in the framebuffer can be filled with any rectangular data from the texture input. When the texture input points to the framebuffer, copying from framebuffer to framebuffer is possible. To avoid copy problems because of overlapping source and destination areas, the copy start point can be selected from top left to bottom right. Possible color formats are any 8-, 16-, or 32-bpp format.

The driver optimizes the copy to gain the full benefit of 32-bit parallel rasterization. If the color depth of the selected color format is less than 32 bpp, the driver corrects the alignment and copies 32 bits per clock cycle, resulting in two to four times faster copy performance in 8-and 16-bpp formats.

#### 39.3.3.3 Stretch BitBLT

This is similar to the normal copy operation. Because the copy is done as a type of texture mapping, the full texture mapping feature set can be used. Any scaling ratios in the x and y directions is selectable, and filtering can be enabled independently for each axis.

#### 39.3.3.4 Rotate and Scale

This is similar to the normal copy operation. Because the copy is done as a type of texture mapping, the full set of texture mapping features can be used. Any scaling ratio in the x and y directions and any rotation angle are selectable. The user can enable or disable filtering for the scalers independently for each of the x and y directions.

#### 39.3.3.5 Alpha Blending

Alpha blending is a fundamental block in the rendering pipeline, so the full set of alpha blend features is available for any BitBLT operation. It is possible to copy an area and blend it over the destination by using any constant global alpha value (register value) or by using an alpha mask. The alpha mask is part of the texture data and can be either a per-pixel value together with a pixel color (aRGB formats) or an alpha-only format using a register color.

In addition to the color channels, the alpha channel can be blended. The formula for the alpha channel can be set independently from the formula for the color channels.

#### 39.3.3.6 Bilinear Filtering

The texture unit can be used to scale, rotate, or shear images. The texturing result can be filtered in the x and y directions independently. When selecting both filters, the result is a bilinear filtered texture. Using the unit twice with two independent textures would generate trilinear filtered bitmaps, improving the visual impression for high dynamic scale ratios.

#### 39.3.3.7 Color Conversion

Color conversion is required when using different texture formats than the framebuffer format. To save texture memory, formats having less bpp than that of the framebuffer are supported. The 2D Drawing Engine always operates internally with the 32-bpp aRGB (8888) format. All input data is converted into 32 bpp and is finally converted back into the framebuffer format.

# 39.4 Input and Output Data Formats

## 39.4.1 Source and Destination Data

There are two possible inputs, the framebuffer and the texture or pattern input. The output is always the framebuffer.

Every drawing operation is internally rendered in 32 bpp aRGB (8888). If the input color does not provide an alpha channel, the blue channel is taken as the alpha channel. This alpha can be substituted with any alpha (for example, by an external constant) during the colorization step in the 2D Drawing Engine.

## 39.4.2 Framebuffer Color Formats

**Table 39.2** shows the supported framebuffer color formats.

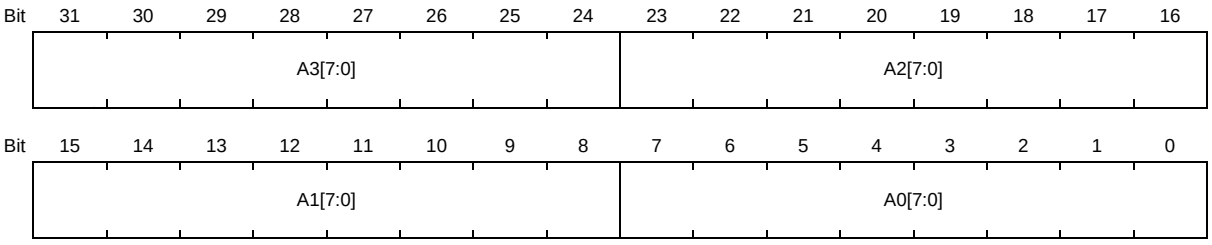
Table 39.2 Framebuffer Color Formats

| Framebuffer Memory Occupation | Format | Remarks |
|---|---|---|
| 8 bpp | a (8) | This color format uses 1 byte per pixel. The alpha channel is internally replicated on the red, blue, and green channels and can be substituted with any color during the color step in the 2D Drawing Engine. |
| 16 bpp | RGB (565) | This color format uses 2 bytes per pixel with 5 bits for red and blue and 6 bits for green. The blue color is taken as the alpha channel during color conversion. The alpha can be substituted with any alpha during the colorization step in the 2D Drawing Engine. |
| | aRGB (4444) | This color format uses 2 bytes per pixel with 4 bits for each color and alpha channel. |
| 32 bpp | aRGB (8888) | This color format uses 4 bytes per pixel with 8 bits for each color and alpha channel. |

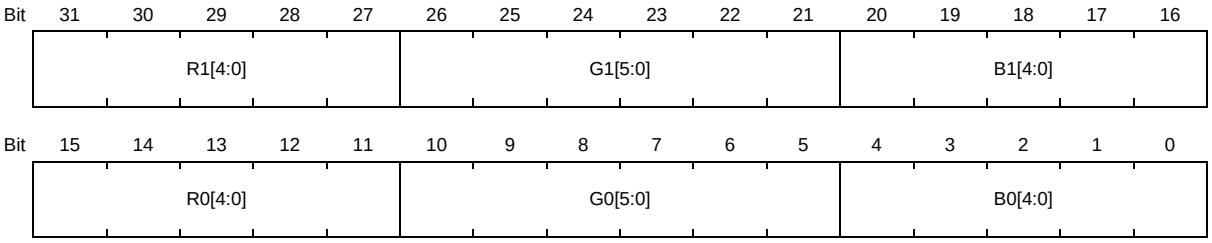
The framebuffer color format is selected in the Surface Control Register bits, CONTROL2.WRITEFORMAT[2:0].

The data allocation of the frame buffer format is as follows.

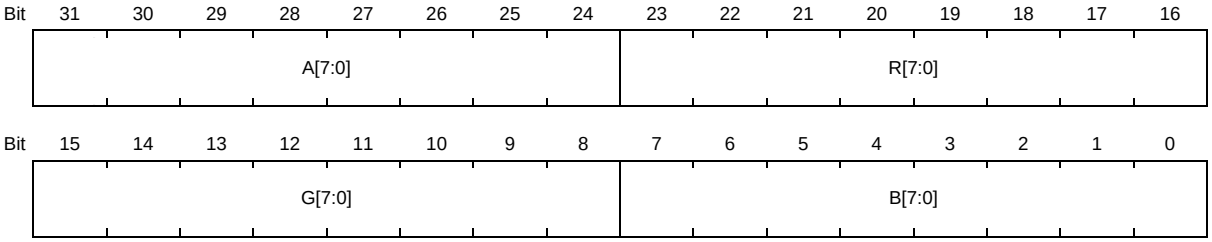
- a(8) (WRITEFORMAT[2:0] = 0)



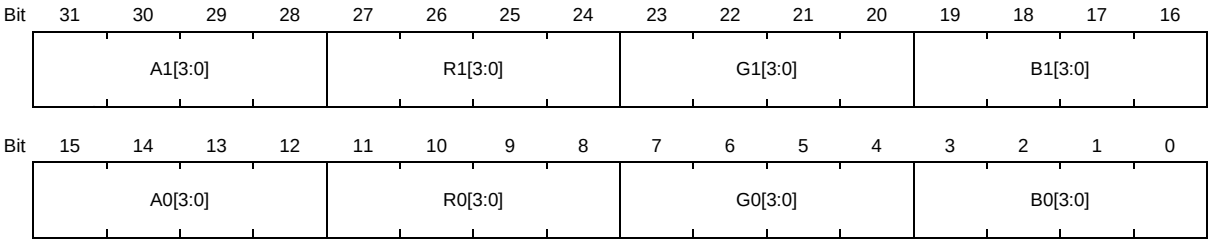

- RGB(565) (WRITEFORMAT[2:0] = 1)



- aRGB(8888) (WRITEFORMAT[2:0] = 2)

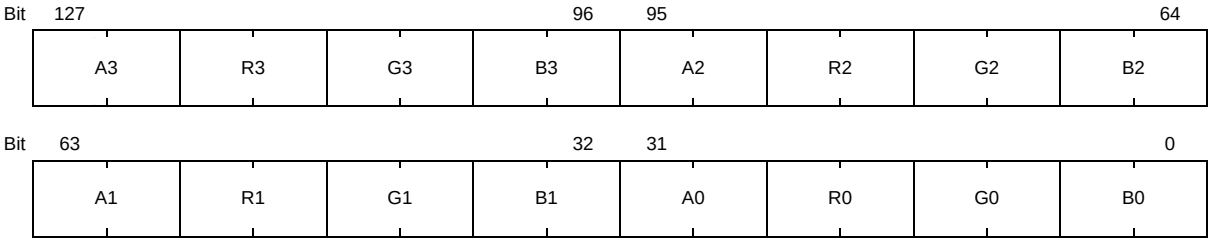



- aRGB(4444) (WRITEFORMAT[2:0] = 3)




The data allocation on the 128-bit ACPU bus is little-endian as shown below.

- aRGB(8888)




Other formats are also little-endian.

### 39.4.3 Texture Color Formats

**Table 39.3** shows the supported texture color formats.

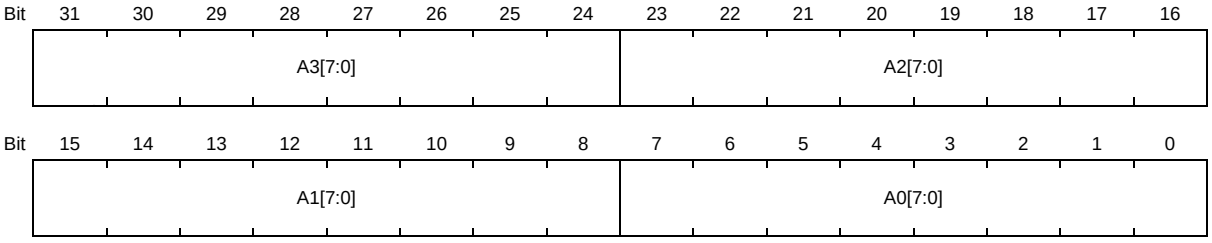
Table 39.3 Texture Color Formats

| Texture Memory Occupation | Format | Remarks |
|---|---|---|
| 1 bpp | CLUT (1)/I (1) | In this mode, a 1-bit index is used to address one of 256 predefined colors in the color lookup table. If the CLUT is not used, the index is taken as a luminance value. |
| 2 bpp | CLUT (2)/I (2) | In this mode, a 2-bit index is used to address one of 256 predefined colors in the color lookup table. If the CLUT is not used, the index is taken as a luminance value. |
| 4 bpp | CLUT (4)/I (4) | In this mode, a 4-bit index is used to address one of 256 predefined colors in the color lookup table. If the CLUT is not used, the index is taken as a luminance value. |
| 8 bpp | CLUT (8)/I (8) | In this mode, an 8-bit index is used to address one of 256 predefined colors in the color lookup table. If the CLUT is not used, the index is taken as a luminance value. |
| | a (8) | This color format uses 1 byte per pixel. The alpha channel is internally replicated on the red, blue, and green channels and can be substituted with any color during the colorization step in the 2D Drawing Engine. |
| | aCLUT (44) | This color format uses 1 byte per pixel. 4 bits are used as an alpha value and 4 bits are used as an index to a color palette. This approach saves space if 16 colors are sufficient to describe the image, because the next bigger alpha format would be 2-byte aRGB (4444). |
| 16 bpp | RGB (565) | This color format uses 2 bytes per pixel with 5 bits for red and blue and 6 bits for green. The blue color is taken as the alpha channel during color conversion. The alpha can be substituted with any alpha during the colorization step in the 2D Drawing Engine. |
| | aRGB (4444) | This color format uses 2 bytes per pixel with 4 bits for each color and alpha channel. |
| | aRGB (1555) | This color format uses 2 bytes per pixel. Every color channel has 5 bits and the topmost single bit is taken as an alpha value. This can be used to hold an image with a transparency mask. |
| 24 bpp | RGB (888) | This color format uses 3 bytes per pixel with 8 bits for each color channel. This format is only available as run length encoded data (RLE compression). |
| 32 bpp | aRGB (8888) | This color format uses 4 bytes per pixel with 8 bits for each color and alpha channel. |

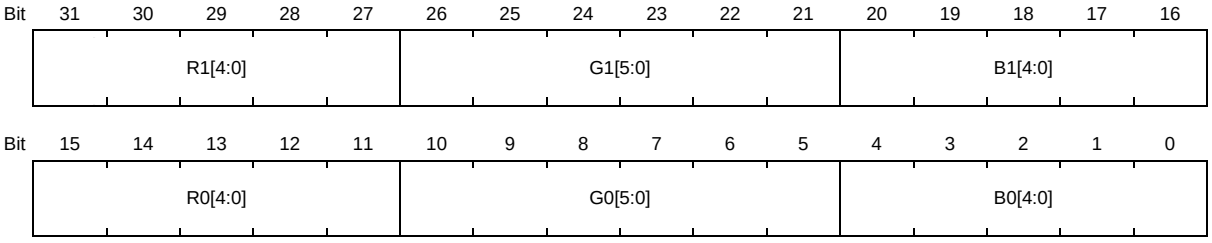
The texture color format is selected in the Surface Control Register bits, CONTROL2.READFORMAT[3:0].

The data allocation of the texture format is as follows.

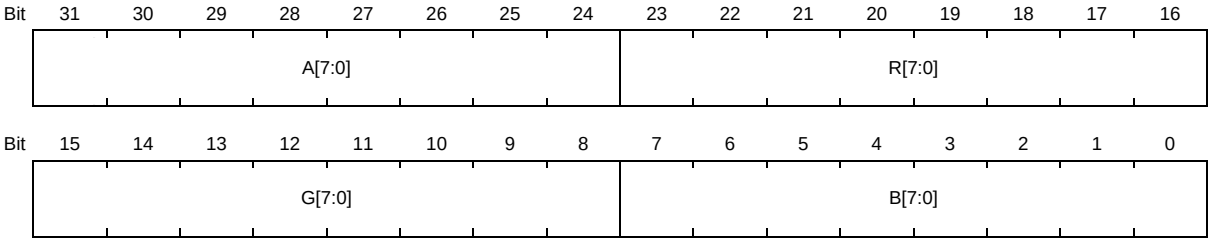
- a(8) (READFORMAT[3:0] = H’0)



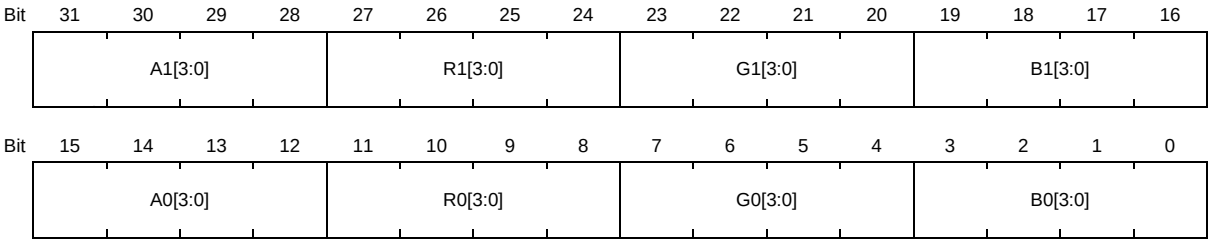

- RGB(565) (READFORMAT[3:0] = H’1)



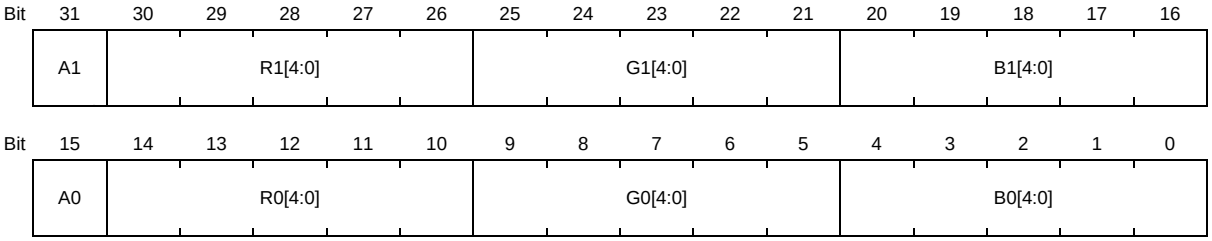
- aRGB(8888) (READFORMAT[3:0] = H’2)



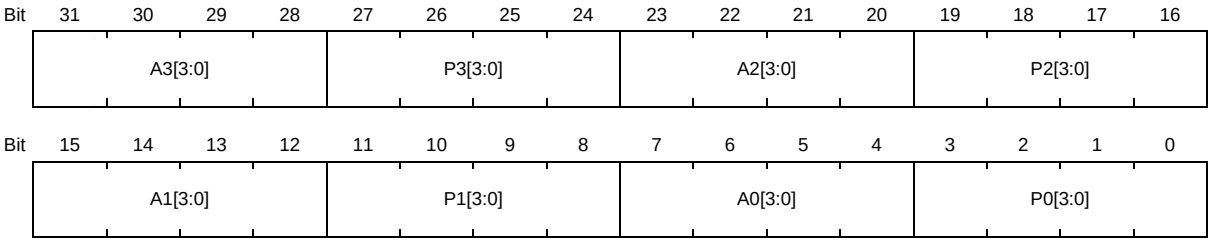

- aRGB(4444) (READFORMAT[3:0] = H’3)



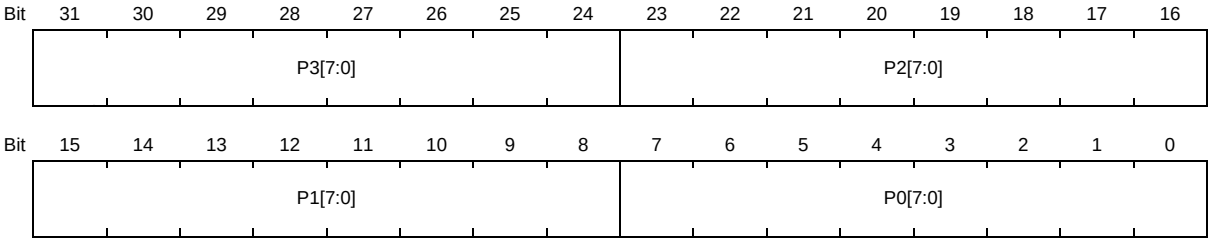

- aRGB(1555) (READFORMAT[3:0] = H’4)



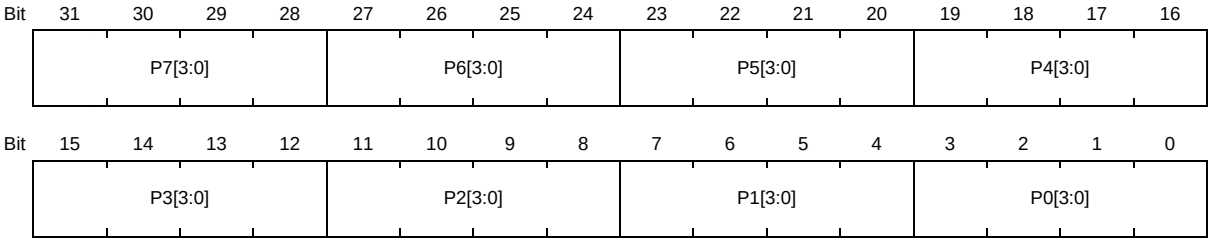

- aCLUT(44) (READFORMAT[3:0] = H’5)



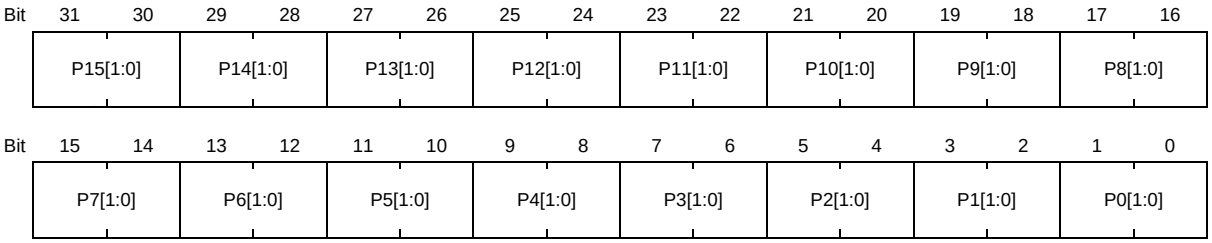

- CLUT(8) (READFORMAT[3:0] = H’9)



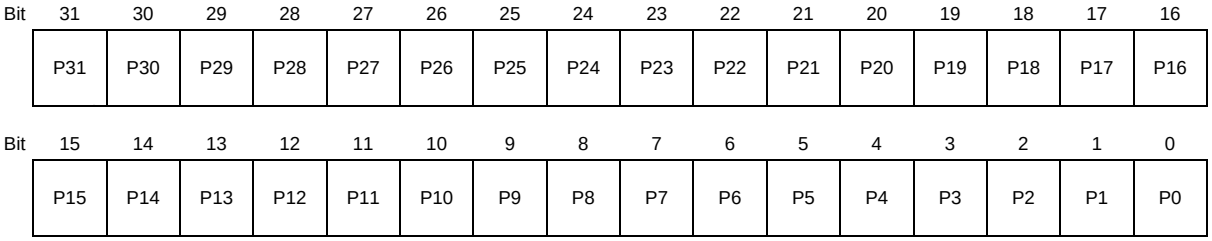
- CLUT(4) (READFORMAT[3:0] = H’A)




- CLUT(2) (READFORMAT[3:0] = H’B)




- CLUT(1) (READFORMAT[3:0] = H’C)




The data allocation on the 128-bit ACPU bus is the same little-endian as the frame buffer.

## 39.5 Texture Data Processing

**Figure 39.4** shows the processing of texture data.

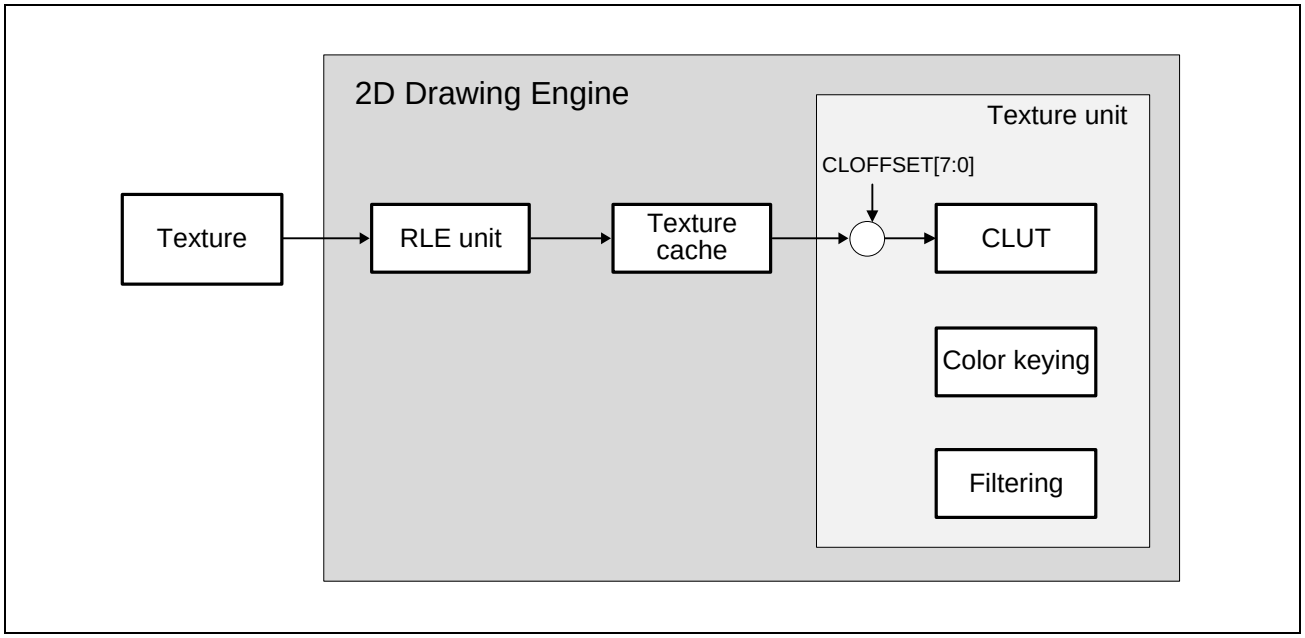



Figure 39.4 Texture Data Processing

### 39.5.1 Texture Color Format

**Table 39.4** shows the supported texture data formats.

Table 39.4 Texture Color Formats

| Texel Bit Width | Texel Format |
|---|---|
| 32 bits | aRGB (8888) |
| 24 bits | RGB (888) |
| 16 bits | aRGB (4444), aRGB (1555), RGB (565) |
| 8 bits | CLUT (8)/I (8), a (8), aCLUT (44) |
| 4 bits | CLUT (4)/I (4) |
| 2 bits | CLUT (2)/I (2) |
| 1 bit | CLUT (1)/I (1) |

### 39.5.2 Run Length Encoded (RLE) Unit

The RLE unit decompresses Targa-like compressed textures and hands the decompressed texel data over to the texture unit. The key features are:

- Support for Targa format
- Avoids the additional Targa limitation to scan lines
- Support for clipping of compressed images, in which the 2D Drawing Engine is allowed to copy only a portion of a larger original texture
- Control of the RLE unit in display list operation
- Bypassing of the RLE unit logic if uncompressed textures are fetched.

#### Texture cache

- The RLE unit feeds the texture cache. The texture cache can be disabled by setting CACHECTL.CENABLETX = 0.

NOTE

A texture cache flush operation (CACHECTL.CFLUSHTX = 1) is necessary at the beginning and end of every new RLE texture.

The texture cache and the RLE unit can be bypassed by setting CONTROL2.RLEENABLE = 0.

#### 39.5.2.1 RLE Texel Formats

**Table 39.5** lists the data formats supported by the RLE unit.

Table 39.5 Texel Formats Supported by the RLE Unit

| Memory Texel Format | RLE Parameters | RLE Coded Unit Format (CONTROL2.RLEPIXELWIDTH[1:0]) | Delivered Format |
|---|---|---|---|
| 32-bit<br>aRGB (8888) | Included in Targa and RLE formats | 32 bits<br>(11b) | 32 bits |
| 24-bit<br>RGB (888) | | 24 bits<br>(10b) | 32 bits*1 |
| 16-bit<br>aRGB (4444),<br>aRGB (1555),<br>RGB (565) | | 16 bits<br>(01b) | 16 bits |
| 8-bit<br>CLUT (8)/I (8), a (8),<br>aCLUT (44) | | 8 bits<br>(00b) | 8 bits |
| 4-bit<br>CLUT (4)/I (4) | Optional for RLE*2 | 8 = 2 × 4 bits | 4 + 4 bits |
| 2-bit<br>CLUT (2)/I (2) | No RLE | | |
| 1-bit<br>CLUT (1)/I (1) | | | |

Note 1. 24-bit RGB (888) encoded texels are delivered as aRGB (8888) with Alpha set to 1.

Note 2. Encoding of textures with 4 bits per texel is not defined by the Targa specification but can be done by:
- — Combining two 4-bit texels to one byte
- — Padding with 4 zero bits at the end of the file, if the number of texels is odd
- — Encoding as with 8-bit texels

**Texel addressing for RLE textures**

The address of a texel is the byte address of the first byte of the texel. The origin of the texture is given by the register TEXORIGIN.

NOTE

The RLE code must begin at a word boundary of the memory.

When the FIFO is filled, there is no provision to inhibit read access beyond the end of the RLE code. To avoid memory access violations, the RLE code must be padded by 32 memory words, where every bit of each word is set to 1.

#### 39.5.2.2 Targa RLE Format

Run-length encoded (RLE) images include two types of data elements:

- Run-length packets
- Raw packets.

The first field (1 byte) of each packet is called the “repetition count field”. The second field is called the “pixel value field” (1, 2, 3, or 4 bytes per pixel).

For run-length packets, the pixel value field contains a single pixel value.

For raw packets, this field holds a variable number of pixel values.

The highest order bit of the repetition count indicates whether the packet is a raw packet or a run-length packet, as follows:

- If bit [7] of the repetition count is set to 1, the packet is a run-length packet
- If bit [7] of the repetition count is set to zero, the packet is a raw packet.

The lower 7 bits of the repetition count specify how many pixel values are represented by the packet. For a run-length packet, this count indicates how many successive pixels have the pixel value specified in the pixel value field. For raw packets, this count specifies how many pixel values are actually contained in the next field. This 7-bit value is actually encoded as 1 less than the number of pixels in the packet (a value of 0 implies 1 pixel while a value of H’7F implies 128 pixels).

##### Run-length packet

Run-length packets are composed of two parts. The first is a repetition count and the second is the pixel value to repeat.

Table 39.6 Run-length Packet

| Field Name | Repetition Count Field | | Pixel Value Field |
|---|---|---|---|
| Contents | Packet Type<br>(must be 1 for run-length) | Pixel Count<br>(number of pixels encoded in this packet 1) | Pixel Data<br>(the shared pixel value to be used) |
| Field size | 1 bit | 7 bits | Pixel depth (**Table 39.5**) |

##### Raw packet

The raw packet always includes two fields. The first field is the repetition count and the second field is the pixel value field.

Table 39.7 Raw Packet

| Field Name | Repetition Count Field | | Pixel Value Field |
|---|---|---|---|
| Contents | Packet Type<br>(must be 0 for raw packet) | Pixel Count<br>(number of pixels encoded by this packet 1) | Pixel Data |
| Field size | 1 bit | 7 bits | Pixel depth × pixel count |

### 39.5.3 Color Lookup Table (CLUT)

The color lookup table receives an index that addresses one out of the 256 predefined colors.

The predefined color format can be selected as:

- CONTROL2.CLUTFORMAT = 0: aRGB (8888)
- CONTROL2.CLUTFORMAT = 1: RGB (565)

The CLUT is filled by the use of two registers:

- TEXCLDATA
  The aRGB (8888) color definition is written to this register, while the CLUT address is taken from TEXCLADDR.
- TEXCLADDR
  This is set to the first address of the CLUT to write to and is automatically incremented after each write to TEXCLDATA.

An offset for indexed formats (CLUT (1), CLUT (2), CLUT (4), and CLUT (8)) can be set up in the TEXCLOFFSET register to allow selecting an offset part of the CLUT. The CLUT index is calculated by CLUT (x) or TEXCLOFFSET.CLOFFSET[7:0].

#### 39.5.3.1 CLUT/I Pixel Data Formats

The following tables explain in which order the pixels are stored within the byte. The left-most pixel is stored at the lowest bit of the memory byte.

##### CLUT (1)/I (1) format

The CLUT (1)/I (1) format expresses 1 pixel by using a total of 1 bit.

| Memory byte | 7 (MSB) | 6 | 5 | 4 | 3 | 2 | 1 | 0 (LSB) |
|---|---|---|---|---|---|---|---|---|
| Pixel | P7 | P6 | P5 | P4 | P3 | P2 | P1 | P0 |

The left-most pixel is stored at lowest bit of the memory byte.

##### CLUT (2)/I (2) format

The CLUT (2)/I (2) format expresses 1 pixel by using a total of 2 bits.

| Memory byte | 7 (MSB) | 6 | 5 | 4 | 3 | 2 | 1 | 0 (LSB) |
|---|---|---|---|---|---|---|---|---|
| Pixel | P3 | | P2 | | P1 | | P0 | |

The left-most pixel is stored at lowest 2 bits of the memory byte.

##### CLUT (4)/I (4) format

The CLUT (4)/I (4) format expresses 1 pixel by using a total of 4 bits.

| Memory byte | 7 (MSB) | 6 | 5 | 4 | 3 | 2 | 1 | 0 (LSB) |
|---|---|---|---|---|---|---|---|---|
| Pixel | P1 | | | | P0 | | | |

The left-most pixel is stored at lowest 4 bits of the memory byte.

### 39.5.4 Color Keying

The 2D Drawing Engine provides a color keying unit in front of the texture unit. It operates as follows:

1. When color keying is enabled (CONTROL2.COLKEYENABLE = 1), the incoming color is compared with the transparent color defined by COLKEY.
2. If the colors match, the alpha and color values are set to 0 to mark the incoming color as transparent, and the color is handled as if the alpha value were pre-multiplied.
3. If the value does not match, alpha is set to 1.
4. Additional operations such as alpha_in × alpha_const are still possible.

With this approach, an object such as a round icon can be cut out from a rectangular texture and can be faded by a constant alpha over the background.

## 39.6 Rendering Pipeline

### 39.6.1 Coordinate Transformation

Coordinate transformation such as rotation, translation, projection, and scaling must be done on the application side. This is not part of the 2D Drawing Engine hardware or driver. Because all coordinates fed into the 2D Drawing Engine are in fixed point format, these calculations can be made in fixed point format and do not require a floating point unit.

### 39.6.2 Rasterization

During rasterization, the vector data of the object must be converted to pixel data. To convert the data, the program sets up the edge interpolation hardware, called a limiter, for each edge of the object that calculates a decision value. The limiter determines which side of the edge the pixel is positioned on. The 2D Drawing Engine includes six internal hardware limiters. In principle, the limiter registers contain the distance between the pixel being processed and the edge.

In the linear setup, a limiter describes a half plane. The intersection of all half planes is the object. If three half planes intersect, a triangle is created as shown in **Figure 39.5**.

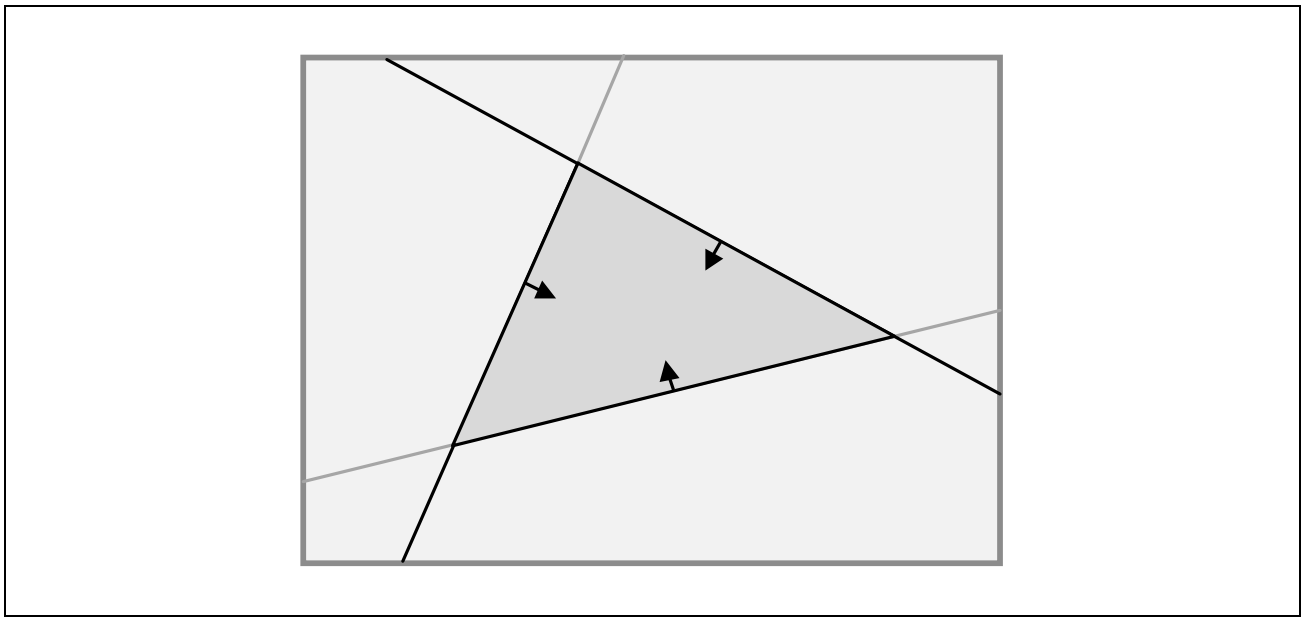


Figure 39.5 Intersection of Half Planes

The limiter output is clamped to an interval of [0:1]. In limiters 1 and 2 it is possible to apply a band filter before the clamping operation. In this case, the limiter is not describing a half plane but a small band. With this approach, a single limiter can describe a thick line of infinite length.

The output of the different limiters can be combined by the combiner units with a maximum or minimum operation. Maximum operation describes the union of both half planes, and minimum operation describes the intersection of both half planes. The final output is then used as an alpha value. Edge anti-aliasing can be done with no additional effort with this hardware.

To calculate the decision value for each possible pixel with a limiter, the bounding box of the object must be calculated. Then, the decision value for the top left corner of the bounding box must be calculated for each edge. Finally, the increments for a step in the x direction and a step in the y direction must be calculated. This is done by the CPU in the driver.

With this information, the 2D Drawing Engine scans the whole bounding box and calculates the decision value for every pixel incrementally. For a block diagram of the entire rasterization unit, see **Figure 39.6**.

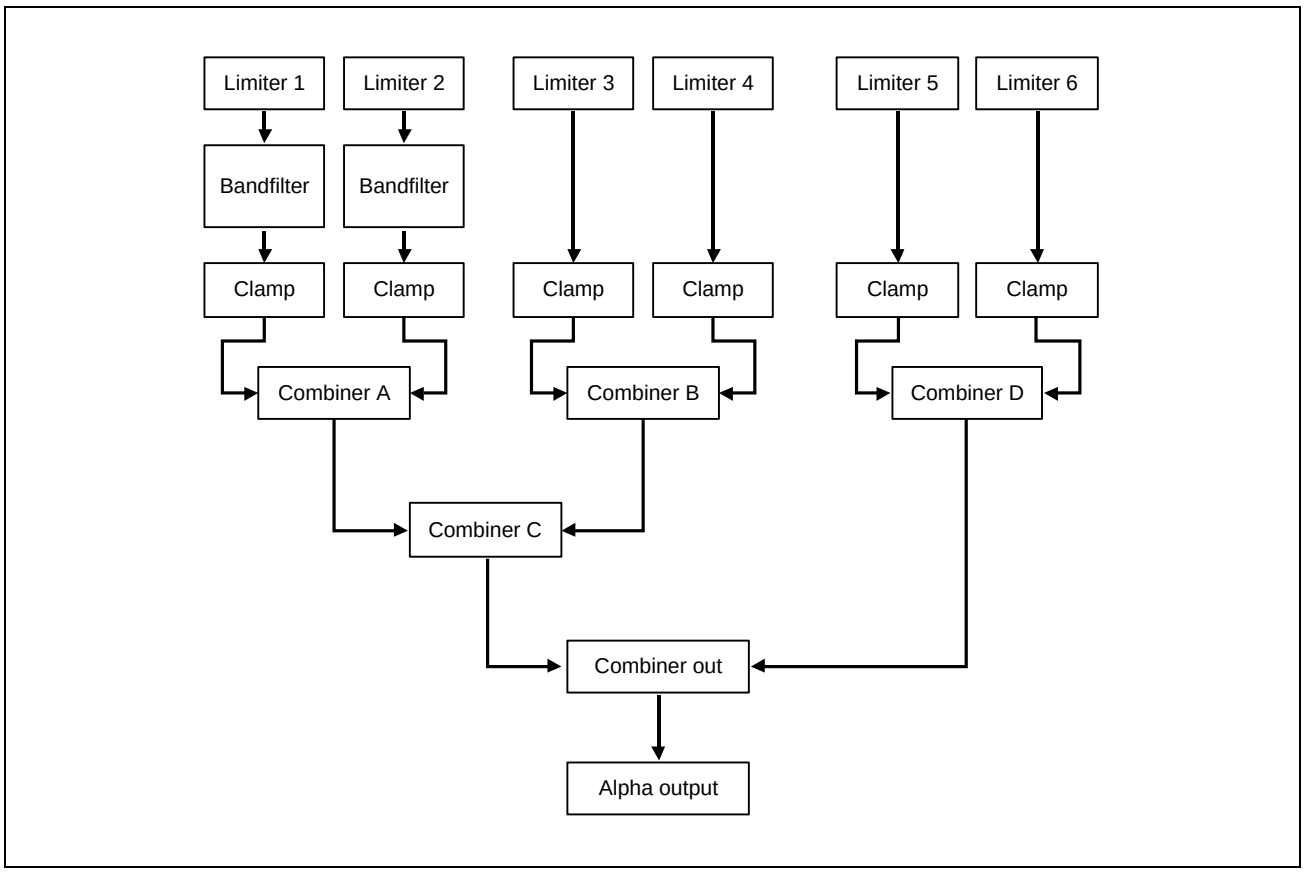



Figure 39.6 Block Diagram of Rasterization Unit

The limiters calculate distances and the combiner units combine them to an alpha value. The combiner units define the conditions for whether or not a pixel is in the bounding box. The alpha value must be greater than 0.

NOTE

It is possible to have all limiters switched off.

### 39.6.2.1 Edge Setup Linear Case

#### (1) Mathematical Background

To setup a linear edge, consider the line equation in the classical form:

This can be written as:

$$y = \tilde{a}x + \tilde{b}$$

This can be rewritten as:

$$0 = f(x,y) = \tilde{a}x - y + \tilde{b} = ax + by + c$$

$$\text{with } a = \tilde{a}, b = -1, c = \tilde{b}$$

This is a more general form. Consider a vector form of this equation:

$$\vec{p} = \begin{pmatrix} x \\ y \end{pmatrix}, \vec{n} = \begin{pmatrix} a \\ b \end{pmatrix} \Rightarrow f(x,y) = ax + by + c = \vec{p} \cdot \vec{n} + c$$

If a point $\vec{p}_0$ is on the line:

$$0 = f(x,y) = \vec{p}_0 \cdot \vec{n} + c \Rightarrow c = -\vec{p}_0 \cdot \vec{n}$$

Rewriting the constant, the equation becomes:

$$c = -\vec{p}_0 \cdot \vec{n} \Rightarrow f(x,y) = (\vec{p} - \vec{p}_0) \cdot \vec{n}$$

The vector $\vec{n}$ is called the normal vector and is perpendicular to the line. The setup can be seen in **Figure 39.7**.

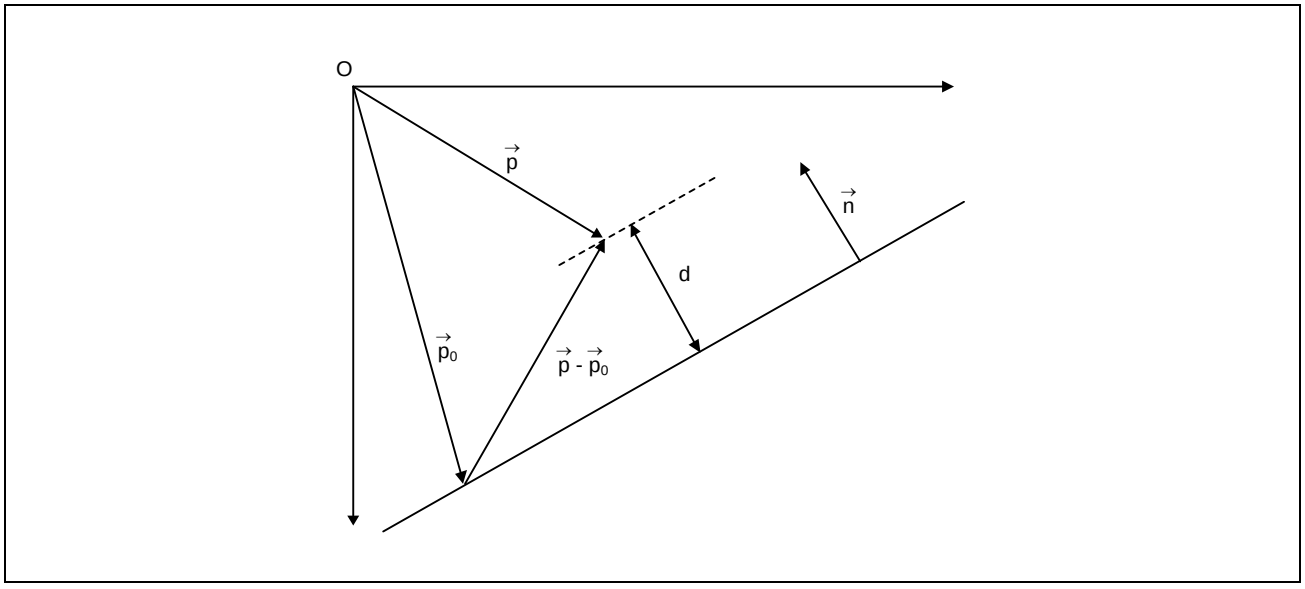



Figure 39.7 Distance of a Point to a Line

The projection of $\vec{p} - \vec{p}_0$ to $\vec{n}$ is the distance d of the point P to the line. In this case, f (x, y) describes the distance to the line of the pixel with coordinates (x, y).

To calculate the distance of every pixel of the bounding box incrementally, first the distance to the line at origin must be calculated:

$$f(0,0) = -\vec{p}_0 \cdot \vec{n} = c$$

Next the increments for a step in the x direction and a step in the y direction must be calculated:

$$f(\vec{p} + \vec{e}_x) = (\vec{p} + \vec{e}_x - \vec{p}_0) \cdot \vec{n} = f(\vec{p}) + \vec{e}_x \cdot \vec{n} = f(\vec{p}) + a$$

$$(\vec{p} + \vec{e}_y) = (\vec{p} + \vec{e}_y - \vec{p}_0) \cdot \vec{n} = f(\vec{p}) + \vec{e}_y \cdot \vec{n} = f(\vec{p}) + b$$

with $\vec{e}_x = \begin{pmatrix}1\\0\end{pmatrix}$, $\vec{e}_y = \begin{pmatrix}0\\1\end{pmatrix}$

Consequently, the new distance can be calculated from the old distance with the increments a and b. A step in the x direction changes the distance by a, and a step in the y direction changes the distance by b. The distance of the origin to the line is c.

With this information, the entire bounding box can be scanned. If the bounding box top left corner is not at the origin, an offset must be added.

### (2) Limiter Operation

The 2D Drawing Engine contains six limiters. Each limiter contains three registers:

- LnSTART
- LnXADD
- LnYADD.

See **Figure 39.8** for details. The limiters can be used in threshold mode, in which all values above 0.5 are set to 1 and all values below or equal to 0.5 are set to 0.

This feature is used when anti-aliasing is not wanted.

NOTE

In **Figure 39.8**, the following abbreviations are used:
start = LnSTART
xadd = LnXADD
yadd = LnYADD

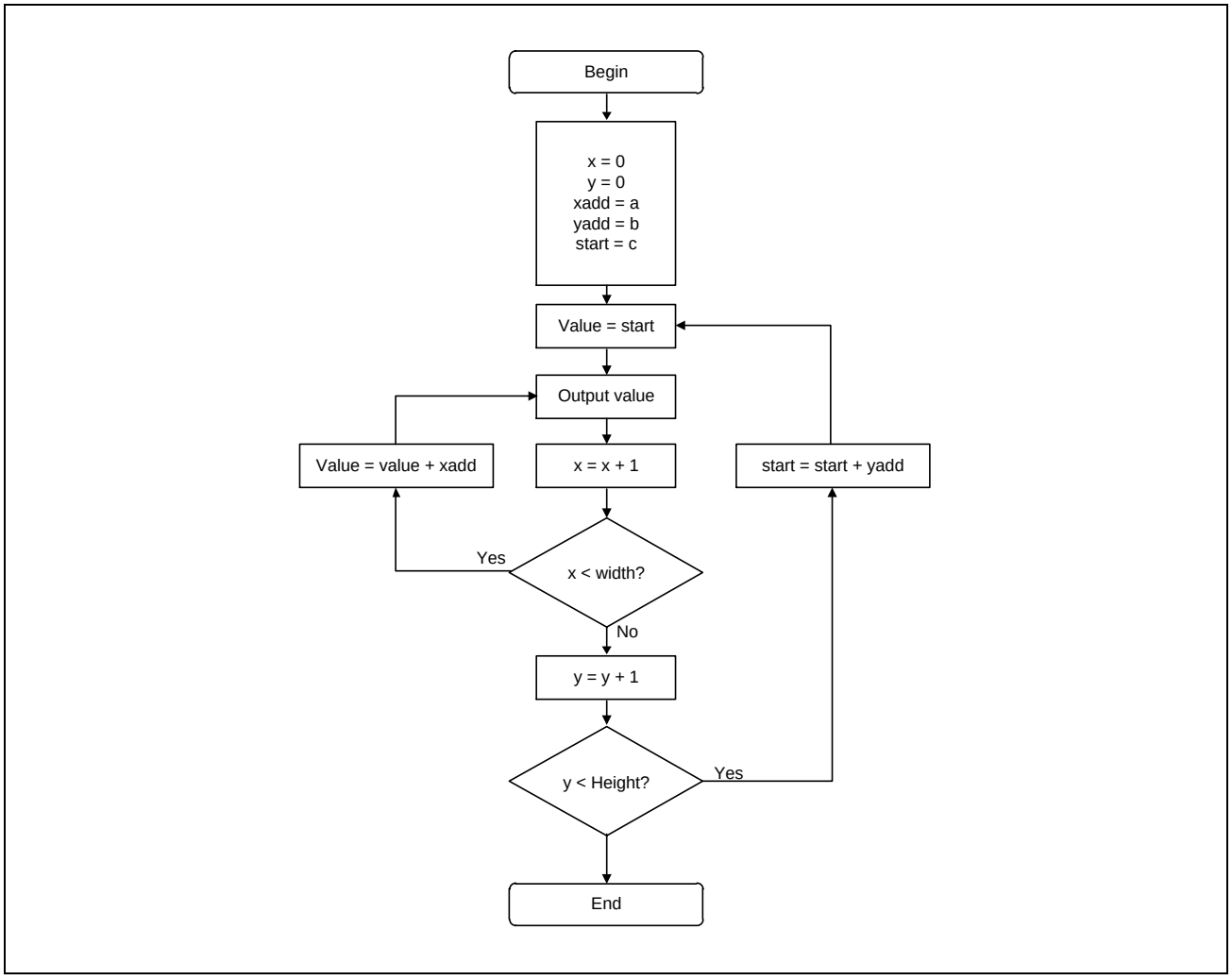



Figure 39.8 Operation Flow of the Linear Limiter

### (3) Example

If a straight line is given by the points P0 and P1, then the values are calculated as follows:

$$\Delta\vec{p} = \vec{p}_1 - \vec{p}_0 = \begin{pmatrix} x_1 - x_0 \\ y_1 - y_0 \end{pmatrix} = \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}$$

The normal vector (perpendicular but not unit size) is then:

$$\vec{n} = \begin{pmatrix} -\Delta y \\ \Delta x \end{pmatrix}$$

The not normalized distance between edge and origin is then:

$$\vec{p}_0 \cdot \vec{n} = -x_0 \Delta y = y_0 \Delta x$$

The limiter parameters would be:

$$start = -x_0 \Delta y + y_0 \Delta x$$

$$xadd = -\Delta y$$

$$yadd = \Delta x$$

In the normalized case, the normal vector is:

$$\vec{n} = \begin{pmatrix} -\Delta y \\ \Delta x \end{pmatrix} / \left(\sqrt{\Delta x^2 + \Delta y^2}\right)$$

The distance between edge and origin is:

$$\vec{p}_0 \cdot \vec{n} = (-x_0 \cdot \Delta y + y_0 \cdot \Delta x) / \left(\sqrt{\Delta x^2 + \Delta y^2}\right)$$

The limiter parameters would be:

$$start = (-x_0 \Delta y + y_0 \Delta x) / \left(\sqrt{\Delta x^2 + \Delta y^2}\right)$$

$$xadd = -\Delta y / \left(\sqrt{\Delta x^2 + \Delta y^2}\right)$$

$$yadd = \Delta x / \left(\sqrt{\Delta x^2 + \Delta y^2}\right)$$

Normalization is only required if anti-aliasing is used. The driver contains an optimized inverse square root function to speed up the normalization process.

### 39.6.2.2 Edge Setup Quadratic Case

#### (1) Mathematical Background

It is also possible to set up the limiters to incrementally calculate the following equation:

$$f(x,y) = ax^2 + by^2 + cx + dy + f$$

At the origin, the value is:

$$f(0,0) = f$$

The step in the x direction is:

$$f(x+1,y)$$

$$= a(x+1)^2 + by^2 + c(x+1) + dy + f$$

$$= ax^2 + 2ax + a + by^2 + cx + c + dy + f$$

$$= f(x,y) + 2ax + c + a$$

$$dx(x) = f(x+1,y) - f(x,y) = 2ax + c + a$$

The step in the y direction is:

$$f(x,y+1)$$

$$= ax^2 + b(y+1)^2 + cx + d(y+1) + f$$

$$= ax^2 + by^2 + 2by + b + cx + d(y+1) + f$$

$$= f(x,y) + 2by + d + b$$

$$dy(y) = f(x,y+1) - f(x,y) = 2by + d + b$$

In the quadratic case, the increments dx and dy depend on x and y and are not constant. They can be calculated incrementally:

$$d^2x = dx(x+1) - dx(x) = 2a(x+1) + c + 1 - (2ax + c + 1) = 2a$$

$$d^2y = dy(y+1) - dy(y) = 2b(y+1) + d + 1 - (2by + d + 1) = 2b$$

At the origin, the increments are:

$$dx(0) = c + 1 \text{ and } dy(0) = d + 1$$

By incrementing the value by dx and dy for every step in the x and y direction and incrementing dx, dy by $d^2x$, and $d^2y$ for every step in the x and y direction, the quadratic equation can be easily calculated for the whole bounding box.

### (2) Limiter Operation

In the quadratic case, two linear limiters are combined to operate as one quadratic limiter, called limiter 1 and limiter 2. The registers are:

- L1START, L1XADD, L1YADD
- L2START, L2XADD, L2YADD.

See **Figure 39.9** for details. The gray box is an addition that performs a different operation, as in the linear setup.

The limiters can be used in threshold mode, in which all values above 0.5 are set to 1 and all values below or equal to 0.5 are set to 0. This feature is used when anti-aliasing is not wanted.

NOTE

In **Figure 39.9**, the following abbreviations are used:
start1 = L1START, start2 = L2START
xadd1 = L1XADD, xadd2 = L2XADD
yadd1 = L1YADD, yadd2 = L2YADD.

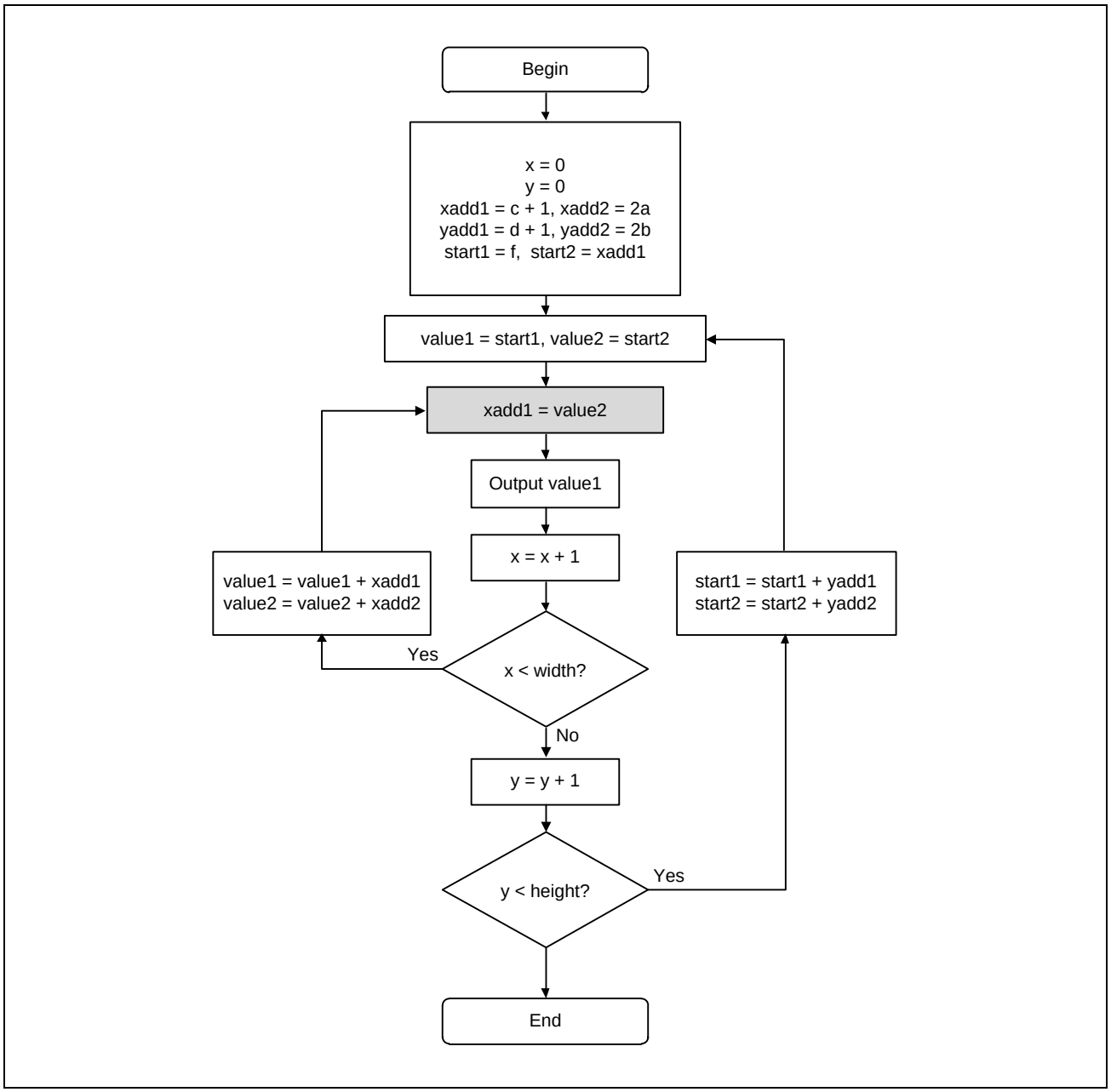



Figure 39.9 Operation Flow of the Quadratic Limiter

### (3) Example

Consider the equation for a circle with the center at $\vec{c} = \binom{s}{t}$ and radius r:

$$0 = f(x,y) = (x-s)^2 + (y-t)^2 - r^2$$

This equation can be rewritten as:

$$f(x,y) = x^2 - 2xs + s^2 + y^2 - 2yt + t^2 - r^2$$

This can be sorted to fit to the original equation:

$$f(x,y) = x^2 + y^2 - 2sx - 2yt + (s^2 + t^2 - r^2)$$

With the following assignments, the circle equation can be calculated incrementally:

$$a = 1$$

$$b = 1$$

$$c = -2s$$

$$d = -2t$$

$$f = s^2 + t^2 - r^2$$

For the limiters with the results calculated in **Section 39.6.2.1(1), Mathematical Background**, this would result in:

$$start1 = f = s^2 + t^2 - r^2$$

$$xadd1 = c + 1 = -2s + 1$$

$$yadd1 = d + 1 = -2t + 1$$

$$dstart2 = xadd1$$

$$xadd2 = 2a = 2$$

$$yadd2 = 2b = 2$$

### 39.6.2.3 Band Filter

The output of limiters 1 and 2 can be modified through a band filter. The band filter has a single filter parameter w.

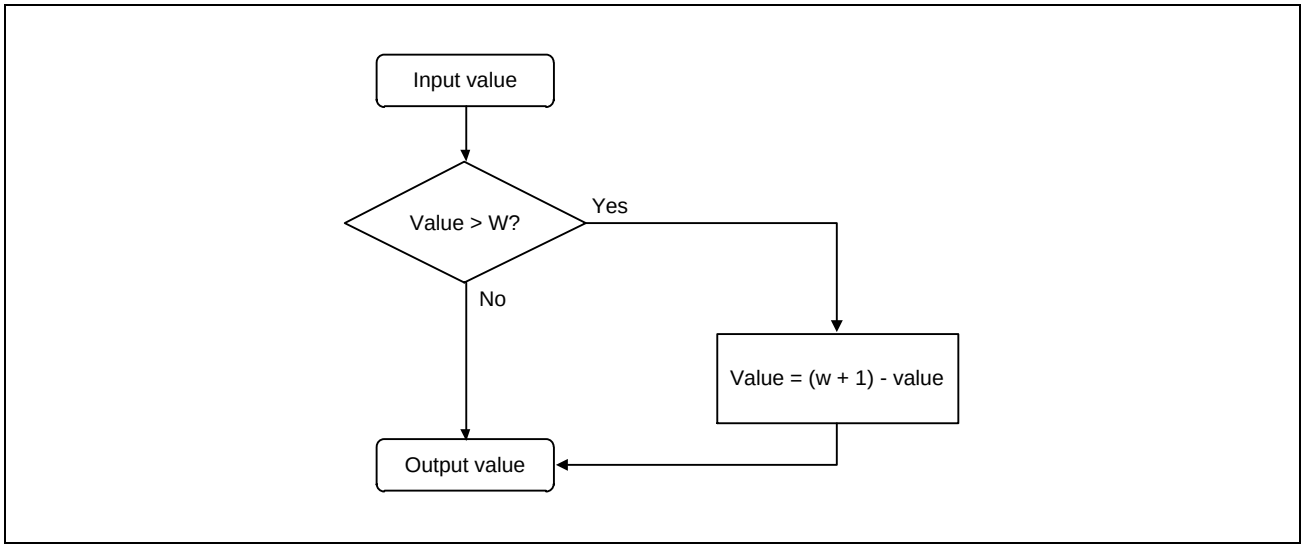



Figure 39.10 Band Filter

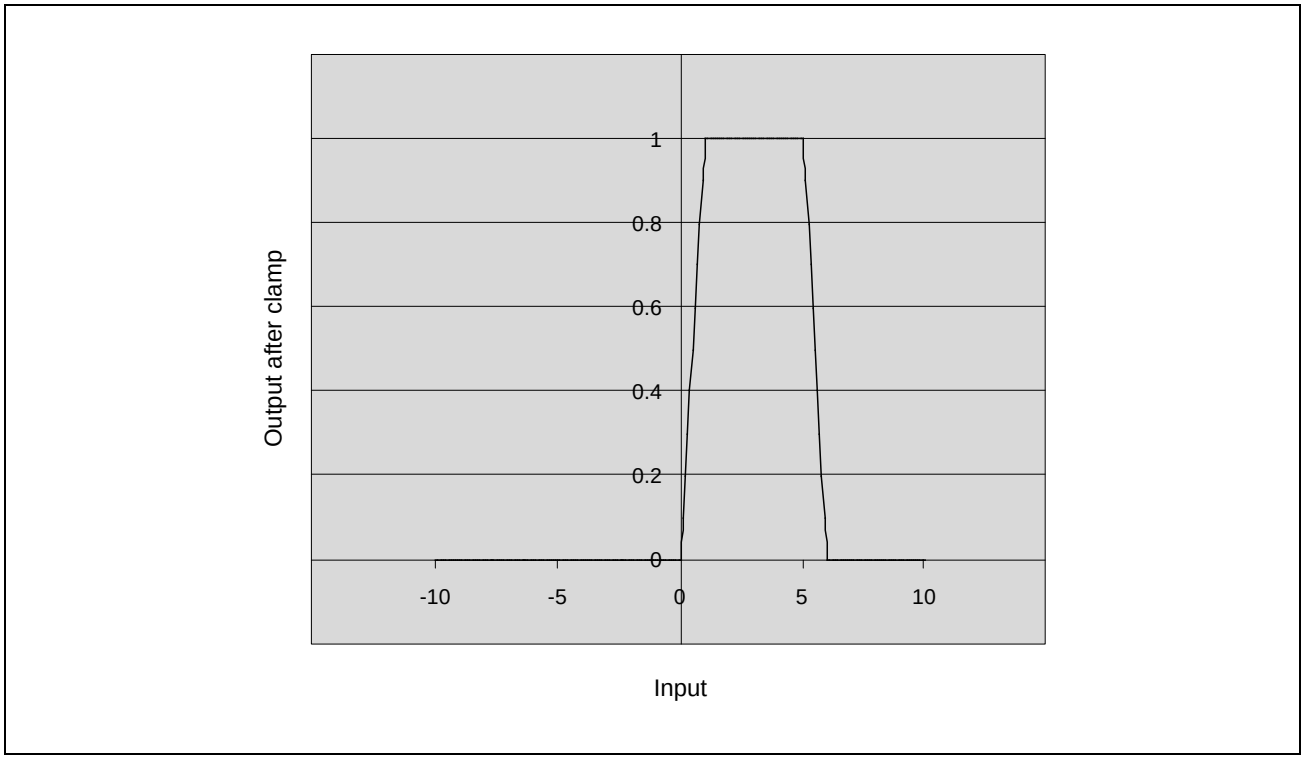



Figure 39.11 Band Filter Output After Clamp with w = 7

#### 39.6.2.4 Clamping Unit

The clamping unit cuts the limiter output to the interval [0:1].

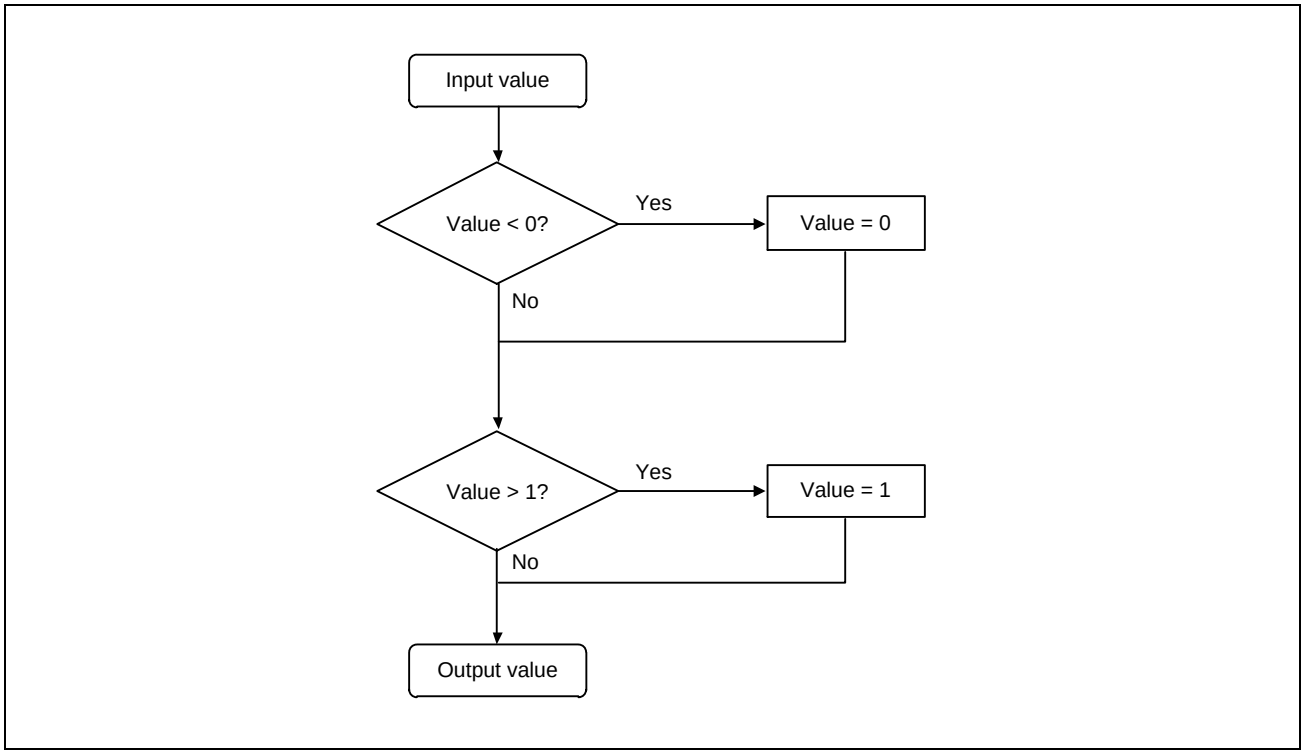



Figure 39.12 Clamping Unit

The clamping unit can be put into threshold mode, in which all values greater than 0.5 are set to 1, and all values below or equal to 0.5 are set to 0. This feature is used when anti-aliasing is not wanted, such as for shared edges.

### 39.6.2.5 Combiner Unit

The combiner unit can be operated in minimum mode and in maximum mode. In minimum mode the smaller value is output, and in maximum mode the larger value is output. The minimum mode represents the intersection, and the maximum mode represents the union of the two regions.

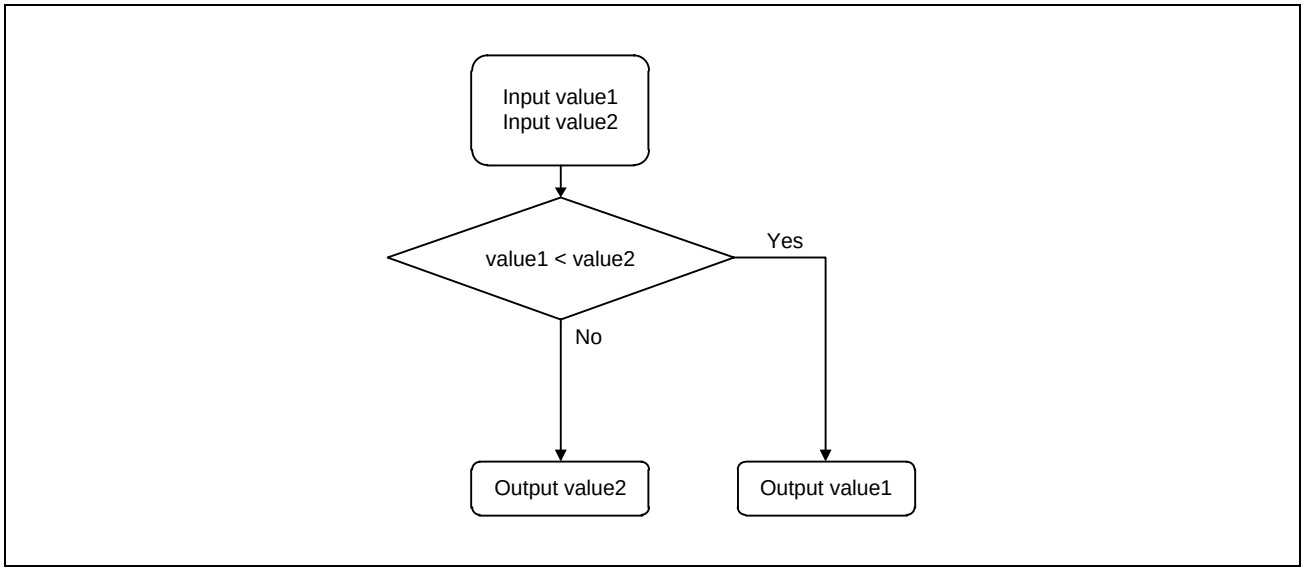



Figure 39.13 Combiner Operated in Minimum Mode with Intersection

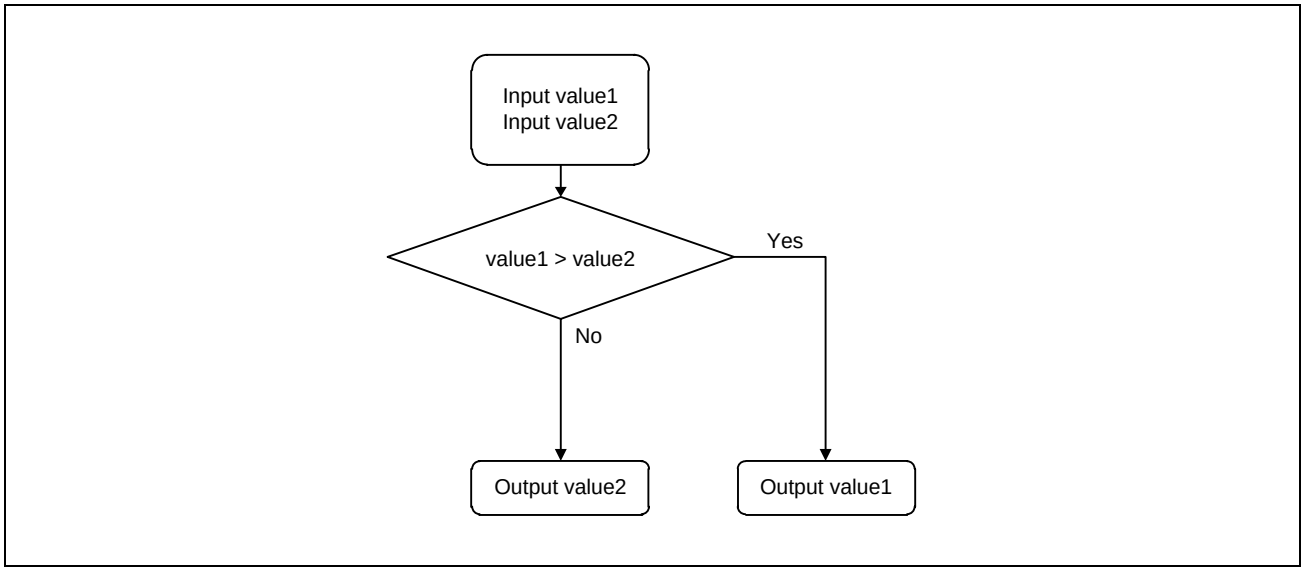



Figure 39.14 Combiner Operated in Maximum Mode with Union

### 39.6.2.6 Rasterization Optimization

During rasterization, it is necessary to step through the whole bounding box one pixel at a time. This requirement can lead to an unnecessary number of steps for pixels that are not drawn. The 2D Drawing Engine provides optimization methods designed to reduce the number of steps required during rasterization. One optimization relies on the fact that any convex primitive can have only one span per line (a span is a contiguous horizontal line). This form of optimization detects a span start and saves the information for the next line. Another optimization is to detect a span end and stop rasterization for the current line.

#### (1) Spanstore

Consider the case in **Figure 39.15**.

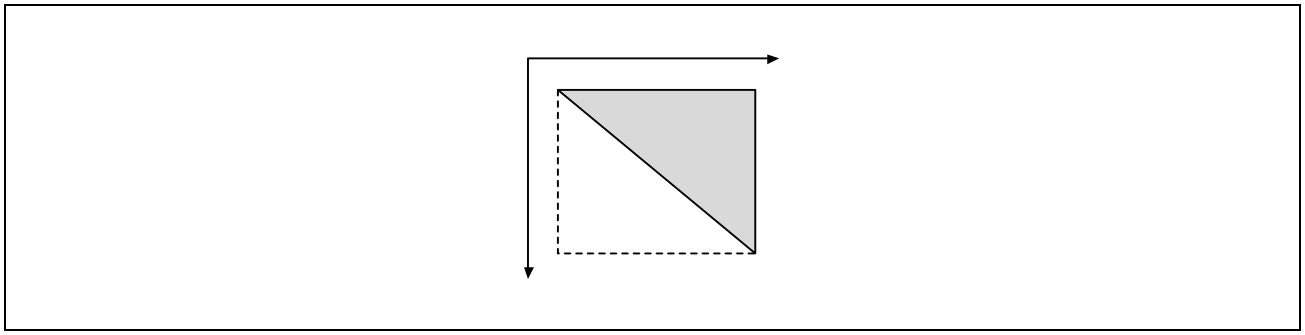


Figure 39.15 Triangle Where the First Edge is Monotone Growing

If the gray triangle must be rendered, half of the pixels processed by the rendering engine in the dotted bounding box would not be drawn. This situation can be optimized with the spanstore operation. When this operation is activated and a span start is detected, the x position of the start is detected.

In the next line the rendering starts with the stored x position. This only works if the edge is monotonically increasing ($y1 > y2 \quad x1 \geq x2$).

Consider the case in **Figure 39.16**.

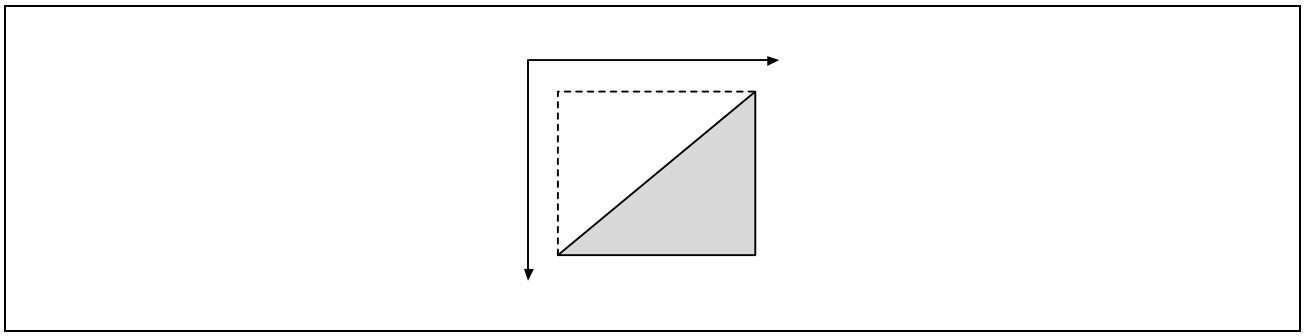


Figure 39.16 Triangle Where the First Edge is Monotone Falling

In this case, the normal spanstore operation cannot be performed. For this, the y direction of the rendering is reversed, and spanstore can operate again.

Consider the last case in **Figure 39.17**.

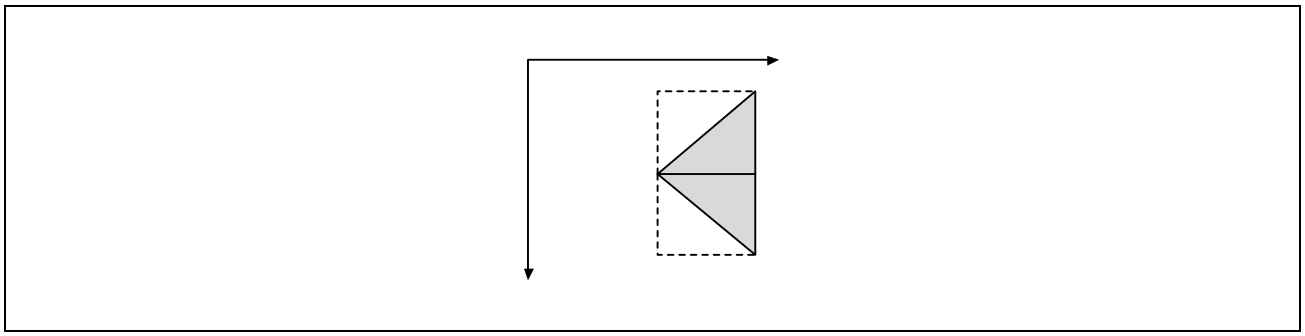


Figure 39.17 Triangle Where the First Edge is First Monotone Falling and then Monotone Growing

In this case, the triangle must be split and rendered as two parts for the spanstore optimization to work.

It is also possible to delay spanstore activation for a number of lines. This approach is used for rasterizing circles, as shown in **Figure 39.18**.

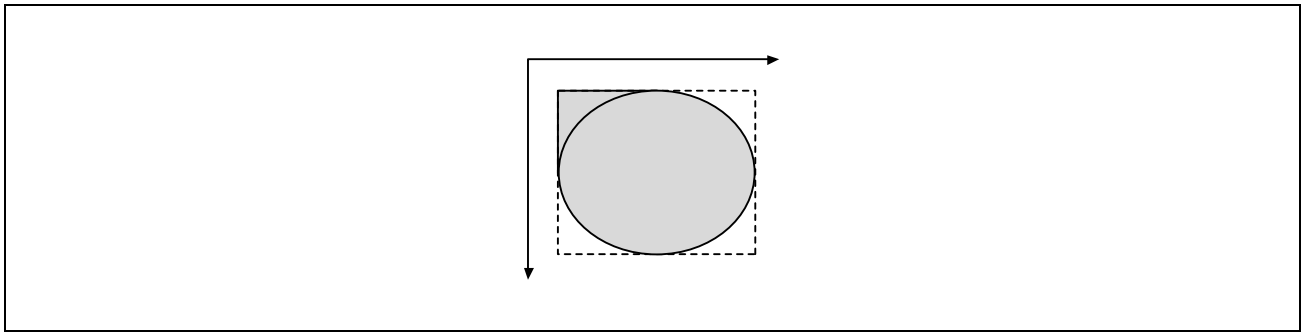


Figure 39.18 Full Circle Where the First Edge is Monotone Falling for the First Half and Monotone Growing for the Second Half

In this case, spanstore cannot be activated in the top left corner but can be activated in the bottom left corner. The empty corners in the top right and the bottom right cannot be rasterized because of the spanabort optimization.

### (2) Spanabort

The second optimization assumes that the object that must be drawn is convex, which means there is only one span per scan line to be drawn. A non-convex object includes an object such as a triangle that is not filled and only consists of a thick border.

For a convex object, the rasterization can be stopped when the end of the first span is detected. No other constraints apply to this optimization for convex objects.

### (3) Optimization Efficiency

The efficiency of the optimizations can be seen for a typical case in **Figure 39.19**. In this case, a triangle is always rendered as a single piece and is not separated into multiple triangles for higher optimizations. For this, the spanstore delay is used.

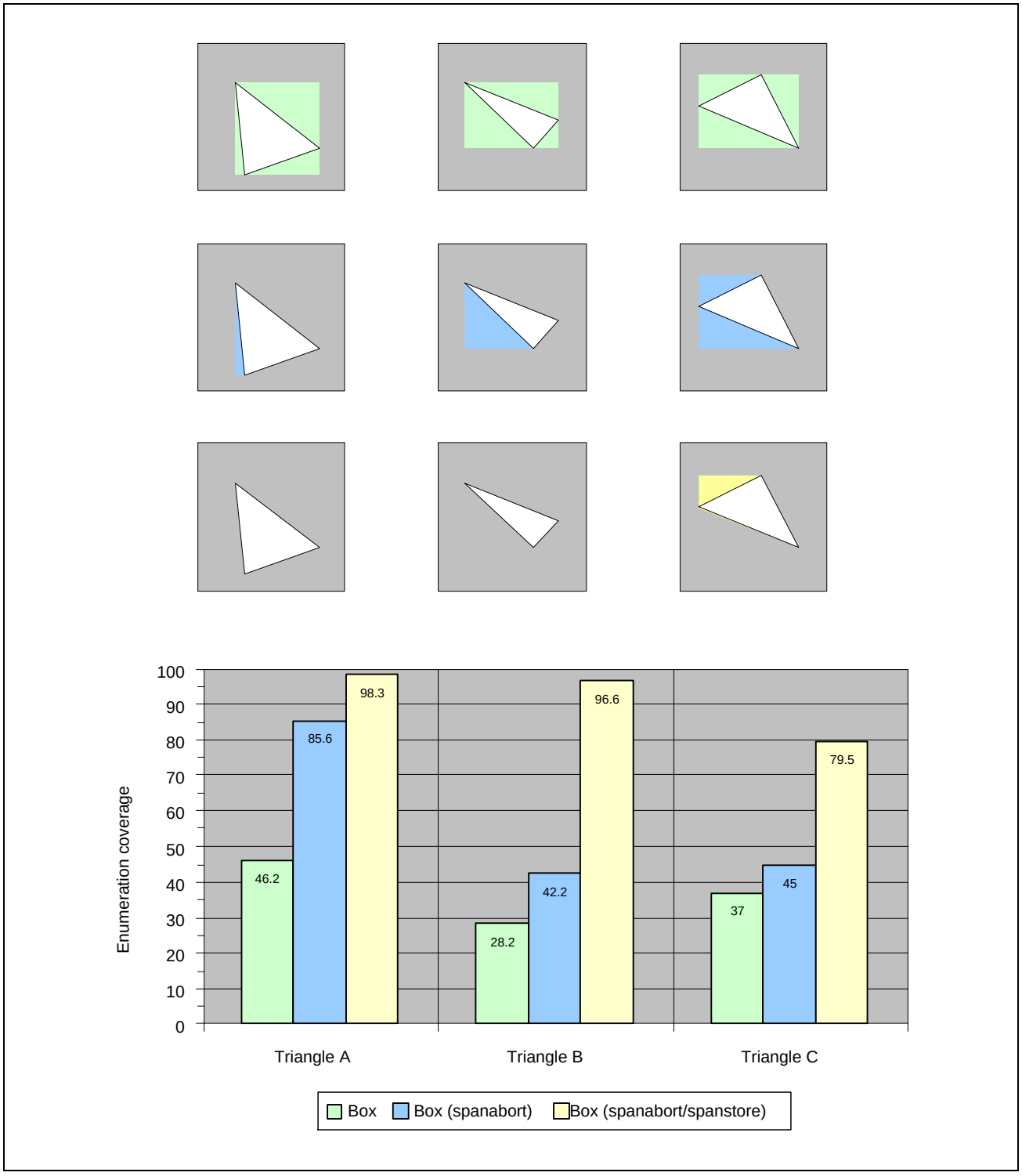



Figure 39.19 Efficiency of Spanstore and Spanabort Optimizations with Enumeration Coverage Equal to {pixels of primitive/pixels of bounding box}

### 39.6.3 Texturing

The texture unit can cover any primitive with a picture. The picture can be stretched, sheared, rotated, and translated in one step. To avoid aliasing, the result can be filtered bilinear in the u and v directions.

#### 39.6.3.1 Mathematical Background

The arbitrary mapping problem is completely determined by a mapping from 3 points in the object space (x, y) to 3 points in the texture space (u, v).

Consider the following mapping:

$$\vec{p}_0 = \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} \Rightarrow (\tilde{\vec{p}}_0) = \begin{pmatrix} u_0 \\ v_0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

$$\vec{p}_1 = \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} \Rightarrow (\tilde{\vec{p}}_1) = \begin{pmatrix} u_1 \\ v_1 \end{pmatrix} = \begin{pmatrix} w \\ 0 \end{pmatrix}$$

$$\vec{p}_2 = \begin{pmatrix} x_2 \\ y_2 \end{pmatrix} \Rightarrow (\tilde{\vec{p}}_2) = \begin{pmatrix} u_2 \\ v_2 \end{pmatrix} = \begin{pmatrix} 0 \\ h \end{pmatrix}$$

where w is the width of the texture and h is the height of the texture.

Examine **Figure 39.20** in the object space. To simplify calculations the difference vectors are taken as calculations:

$$\vec{d}_1 = \vec{p}_1 - \vec{p}_0$$

$$\vec{d}_2 = \vec{p}_2 - \vec{p}_0$$

This is equivalent to transforming from coordinate system O to coordinate system O’.

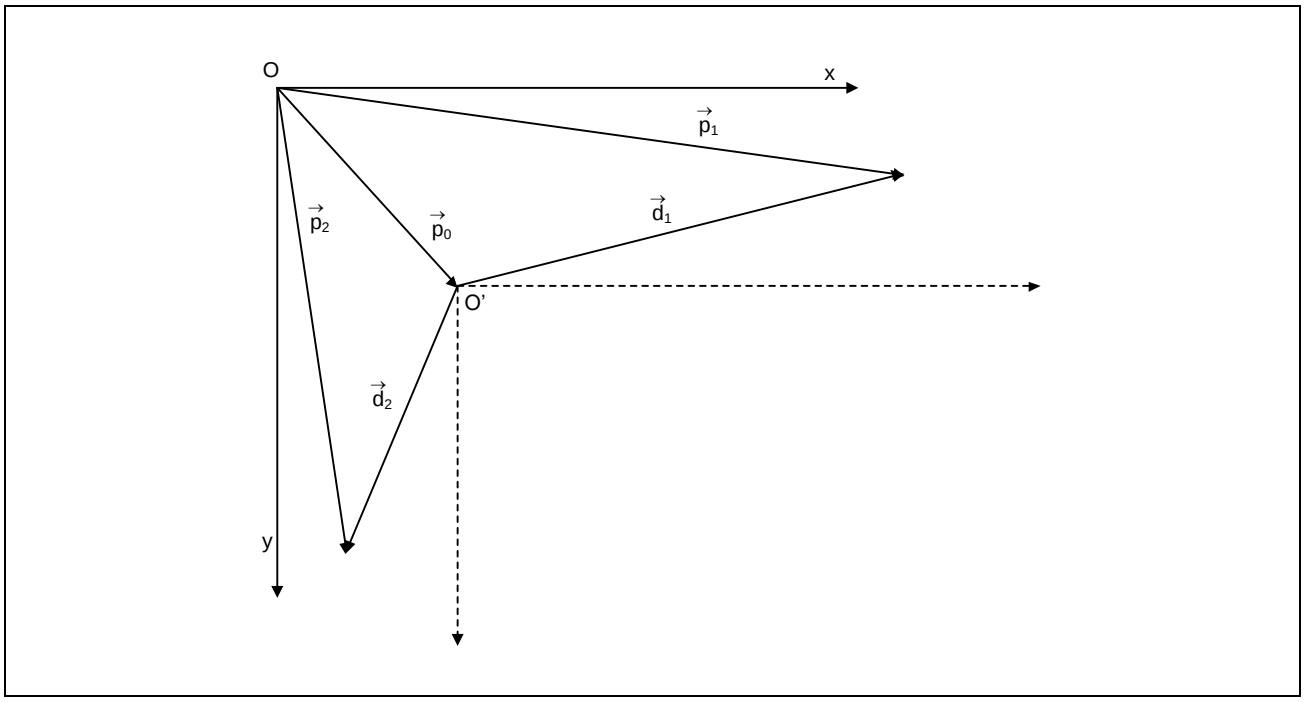



Figure 39.20 Texture Mapping, Object Space, and Transformation from Coordinate System O to O’ to Simplify Calculations

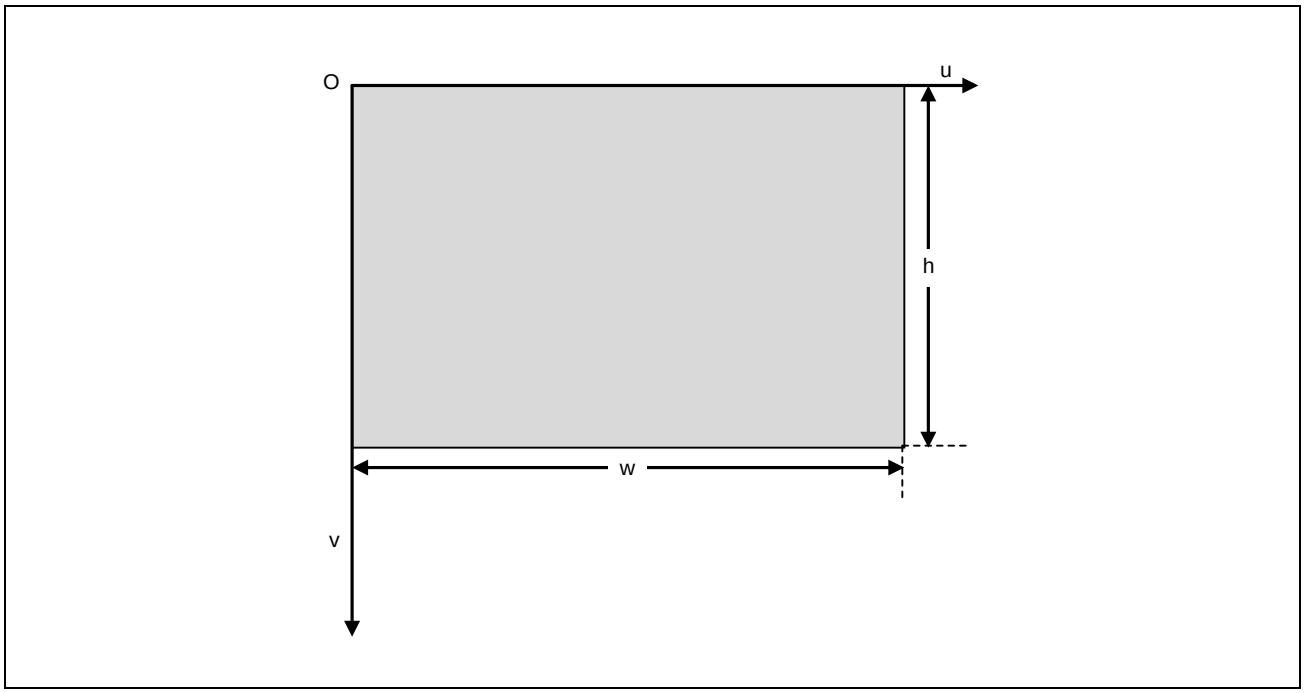



Figure 39.21 Texture Mapping, Texture Space, and Texture with Width w and Height h

In O', the mapping is:

$$\overrightarrow{p'}_0 = \begin{pmatrix}0\\0\end{pmatrix} \Rightarrow (\tilde{\vec{p}}_0) = \begin{pmatrix}u_0\\v_0\end{pmatrix} = \begin{pmatrix}0\\0\end{pmatrix}$$

$$\vec{d}_1 = \begin{pmatrix}dx_1\\dy_1\end{pmatrix} \Rightarrow (\tilde{\vec{p}}_1) = \begin{pmatrix}u_1\\v_1\end{pmatrix} = \begin{pmatrix}w\\0\end{pmatrix}$$

$$\vec{d}_2 = \begin{pmatrix}dx_2\\dy_2\end{pmatrix} \Rightarrow (\tilde{\vec{p}}_2) = \begin{pmatrix}u_2\\v_2\end{pmatrix} = \begin{pmatrix}0\\h\end{pmatrix}$$

This is a linear mapping that can be described by a 2 x 2 matrix.

$$(\tilde{\vec{p}}) = M \cdot \overrightarrow{p'} = \begin{bmatrix}m_{11} & m_{12}\\m_{21} & m_{21}\end{bmatrix} \cdot \overrightarrow{p'}$$

$$\Rightarrow \begin{pmatrix}w\\0\end{pmatrix} = M \cdot \vec{d}_1 \wedge \begin{pmatrix}0\\h\end{pmatrix} = M \cdot \vec{d}_2$$

If the equations are expanded and sorted, the result is two equation systems each with two unknowns. These can be described more easily with a new matrix.

Let $A = \begin{bmatrix}dx_1 & dy_1\\dx_2 & dy_2\end{bmatrix}$ then the equation system can be rewritten as:

$$\begin{pmatrix}w\\0\end{pmatrix} = A \cdot \begin{pmatrix}m_{11}\\m_{12}\end{pmatrix} \wedge \begin{pmatrix}0\\h\end{pmatrix} = A \cdot \begin{pmatrix}m_{21}\\m_{22}\end{pmatrix}$$

This can be easily solved with determinants.

$$\text{Let } C = \frac{1}{\det A} = \frac{1}{dx_1 \cdot dy_{2-dx_2 \cdot dy_1}}$$

The resulting constants are:

$$m_{11} = c \cdot w \cdot dy_2 = \frac{du}{dx}$$

$$m_{12} = -c \cdot w \cdot dx_2 = \frac{du}{dy}$$

$$m_{21} = c \cdot h \cdot dx_1 = \frac{dv}{dx}$$

$$m_{22} = -c \cdot h \cdot dx_2 = \frac{dv}{dy}$$

To calculate the start values for u and v at the top of the bounding box, the transformation from O' to O must be reversed. Let us and vs be the start values, then:

$$\begin{pmatrix} u_s \\ v_s \end{pmatrix} = M \cdot (-\vec{p}_0) = c \cdot \begin{pmatrix} -w \cdot (x_0 \cdot dy_2 - y_0 \cdot dx_2) \\ h \cdot (x_0 \cdot dy_1 - y_0 \cdot dx_1) \end{pmatrix}$$

**Examples)**

U and v are in the texture space. Enter the following into any case:

- Copy case:
  dx1 = 1, dx2 = 0, dy1 = 0, dy2 = 1
- Scaling case x scaling copy case:
  dx1 = f, dx2 = 0, dy1 = 0, dy2 = 1
  with f being the scaling factor in the x direction similar for the y direction
- Rotation case:
  dx1 = cosa, dx2 = -sina, dy1 = sina, dy2 = cosa
  with the angle between d1 and the x axis in the clockwise direction

#### 39.6.3.2 Limiter Operation

The texture limiters operate exactly in the same way as the linear limiters shown in **Figure 39.8**.

The register layout for the u limiter is the same:

- LUSTART = us
- LUXADD = du/dx
- LUYADD = du/dy.

The register layout for the v limiter is slightly different. TEXPITCH is multiplied to save one hardware multiplier.

- LVSTARTI = floor (vs) TEXPITCH
  Contains the integer part of the start value.

- LVSTARTF = (vs-floor (vs)) TEXPITCH
  Contains the fractional part of the start value.

- LVXADDI = floor (dv/dx) TEXPITCH
  Contains the integer part of dv/dx.

- LVYADD = floor (dv/dy) TEXPITCH
  Contains the integer part of dv/dy.

- LVYXADDF.LVXADDF = (dv/dx floor (dv/dx)) TEXPITCH
  Contains the fractional part of dv/dx

- LVYXADDF.LVYADDF = (dv/dy floor (dv/dy)) TEXPITCH
  Contains the fractional part of dv/dy

- TEXMASK.TEXUMASK, TEXMASK.TEXVMASK
  Contains a mask for u and v separately to wraparound values of u and v. This is useful for staying inside the limits of the texture or repeat a texture. Wrap around textures have to have a size multiple of 2.

- TEXPITCH
  Contains the width of the texture in pixels in framebuffer memory. This information is required to calculate the new address if stepping to a new line.

- TEXORIGIN
  Contains the base address of the texture.

### 39.6.4 Colorization

After a pixel is found to be part of the geometry, its color is calculated. The 2D Drawing Engine supports a very general color calculation scheme, allowing support for several color modes. This color scheme uses an interpolation between two color registers, COLOR1 and COLOR2. See **Figure 39.22** for details. COLOR1 and COLOR2 are marked as A, B in the figure for clarity.

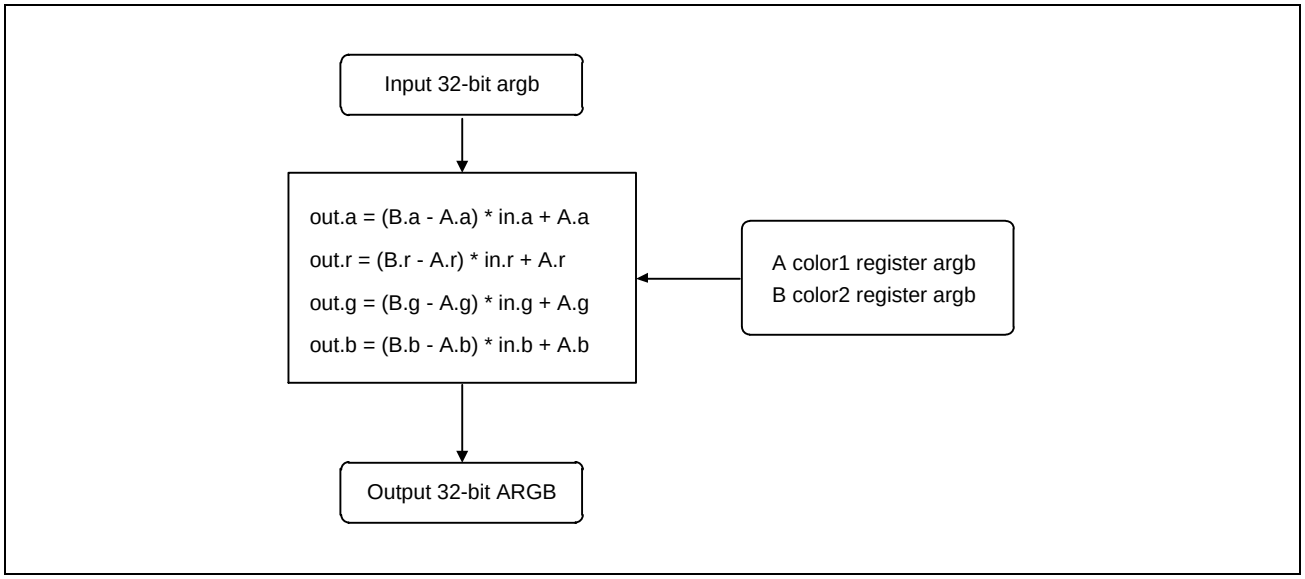



Figure 39.22 Colorization Step and Interpolation between the Two Color Registers, A (COLOR1) and B (COLOR2)

This general approach can be used to support several different color modes that can be individually applied to any color or alpha channel of the input.

Table 39.8 Colorization Operations

| Operation | Settings for A and B*1 |
|---|---|
| Copy | A = 0, B = ffh |
| Replace with a constant value v | A = v, B = v |
| Multiply by a constant value v | A = 0, B = v |
| Colorize an alpha texture with the RGB value v | A.a = 0, A.r = B.r, A.g = B.g, A.b = B.b,<br>B.a = ffh, B.r = v.r, B.g = v.g, B.b = v.b |
| Invert a channel | A = ffh, B = 0 |
| Invert multiply with v | A = v, B = 0 |
| Interpolate between color v and color u | A = v, B = u |

Note 1. A = COLOR1, B = COLOR2

## 39.6.5 Blending

### 39.6.5.1 Color Channel Blending

The last step before the pixel is written to the framebuffer is to blend the pixel with the data that is already written to the framebuffer. If blending is activated, the framebuffer, referred to as DST, must be read. SRC is the output from the colorization unit.

The following color blend modes are supported:

- SRC_ZERO
- SRC_ONE
- SRC_ALPHA
- SRC_ONE_MINUS_ALPHA
- DST_ZERO
- DST_ONE
- DST_ALPHA
- DST_ONE_MINUS_ALPHA

The selection of the color channel blend modes is performed with the following flags:

- BSF: blend source factor is alpha
- BSI: blend source factor invert
- BDF: blend destination factor is alpha
- BDI: blend destination factor invert

The formula for the blending is:

$dst = src \cdot f_S + dst \cdot f_D$

where:

BSF = 0, BSI = 0 → $f_S = 1$

BSF = 1, BSI = 0 → $f_S = \alpha$

BSF = 0, BSI = 1 → $f_S = 0$

BSF = 1, BSI = 1 → $f_S = 1 \quad \alpha$

BDF = 0, BDI = 0 → $f_D = 1$

BDF = 1, BDI = 0 → $f_D = \alpha$

BDF = 0, BDI = 1 → $f_D = 0$

BDF = 1, BDI = 1 → $f_D = 1 \quad \alpha$

**Table 39.9** lists all possible color channel blend modes.

Table 39.9 Color Channel Blend Modes

| Mode | BSF | BSI | BDF | BDI | Blend Equation |
|---|---|---|---|---|---|
| SRC_ONE DST_ONE | 0 | 0 | 0 | 0 | SRC + DST |
| SRC_ONE | 0 | 0 | 0 | 1 | SRC |
| SRC_ONE DST_ALPHA | 0 | 0 | 1 | 0 | SRC + DST × ALPHA |
| SRC_ONE DST_ONE_MINUS_ALPHA | 0 | 0 | 1 | 1 | SRC + DST × (1 ALPHA) |
| SRC_ZERO DST_ONE | 0 | 1 | 0 | 0 | DST |
| SRC_ZERO DST_ZERO | 0 | 1 | 0 | 1 | 0 |
| SRC_ZERO DST_ALPHA | 0 | 1 | 1 | 0 | DST × ALPHA |
| SRC_ZERO DST_ONE_MINUS_ALPHA | 0 | 1 | 1 | 1 | DST × (1 ALPHA) |
| SRC_ALPHA DST_ONE | 1 | 0 | 0 | 0 | SRC × ALPHA + DST |
| SRC_ALPHA | 1 | 0 | 0 | 1 | SRC × ALPHA |
| SRC_ALPHA DST_ALPHA | 1 | 0 | 1 | 0 | SRC × ALPHA + DST × ALPHA |
| SRC_ALPHA DST_ONE_MINUS_ALPHA | 1 | 0 | 1 | 1 | SRC × ALPHA + DST × (1 ALPHA) |
| SRC_ONE_MINUS_ALPHA DST_ONE | 1 | 1 | 0 | 0 | SRC × (1 ALPHA) + DST |
| SRC_ONE_MINUS_ALPHA | 1 | 1 | 0 | 1 | SRC × (1 ALPHA) |
| SRC_ONE_MINUS_ALPHA DST_ALPHA | 1 | 1 | 1 | 0 | SRC × (1 ALPHA) + DST × ALPHA |
| SRC_ONE_MINUS_ALPHA DST_ONE_MINUS_ALPHA | 1 | 1 | 1 | 1 | SRC × (1 ALPHA) + DST × (1 ALPHA) |

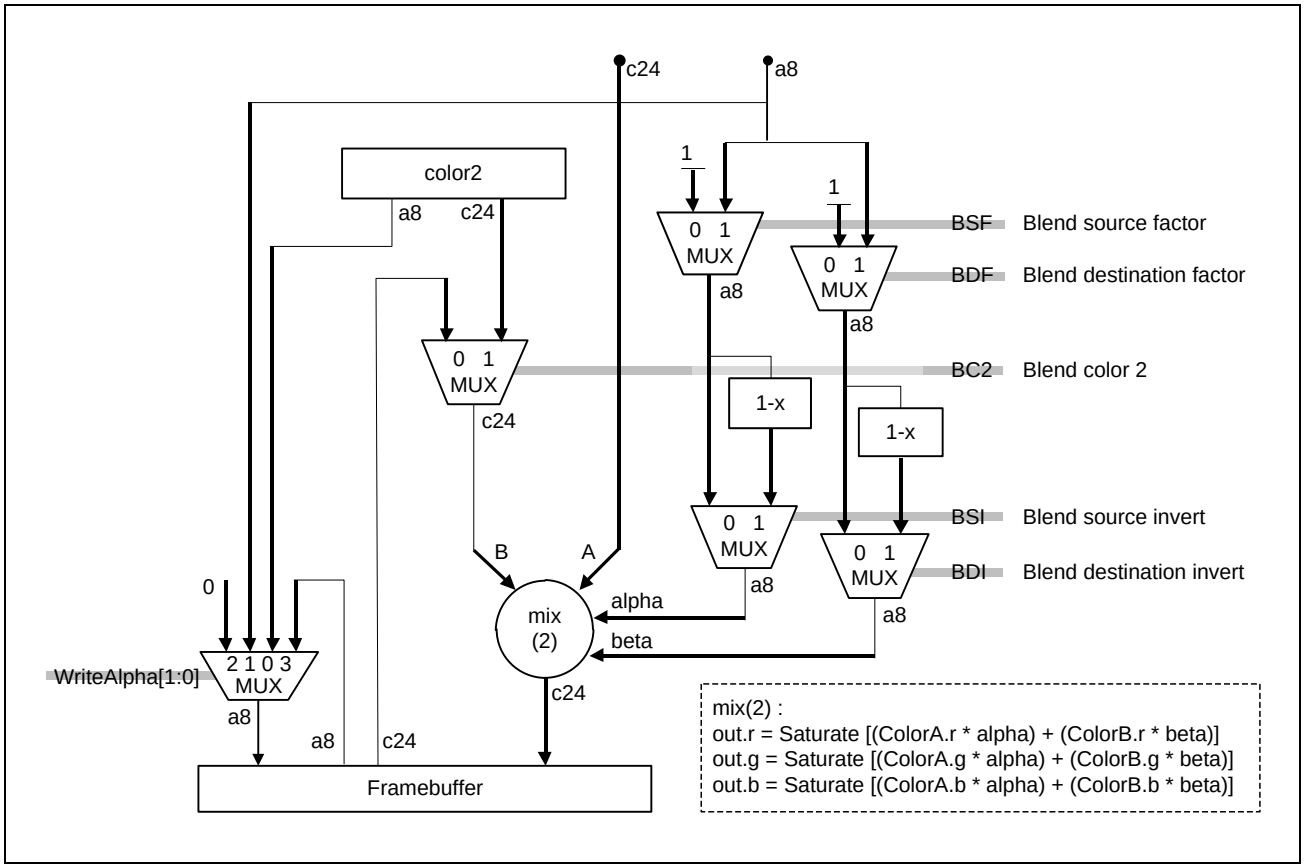



Figure 39.23 Color Channel Blend Unit when CONTROL2.USEACB = 0

#### 39.6.5.2 Alpha Channel Blending

The alpha channel can be blended in addition to the color channels. Alpha channel blending is enabled by setting CONTROL2.USEACB = 1. Alpha channel blending uses the same formulas and same blend modes as for the color channels. The alpha channel formulas can be set independently from the color channels.

The following alpha channel blend modes are supported:

- SRC_A_ZERO
- SRC_A_ONE
- SRC_A_SRC_A
- SRC_A_ONE_MINUS_SRC_A
- DST_A_ZERO
- DST_A_ONE
- DST_A_SRC_A
- DST_A_ONE_MINUS_SRC_A

The alpha channel blend modes selected with the following flags:

- BSFA: blend source factor is SRC_A
- BSIA: blend source factor invert
- BDFA: blend destination factor is SRC_A
- BDIA: blend destination factor invert

The formula for the blending is:

$dst_alpha = src_a \cdot f_{S}_a + dst_a \cdot f_{D}_a$

where:

BSFA = 0, BSIA = 0 → $f_{S}_a = 1$

BSFA = 1, BSIA = 0 → $f_{S}_a = src_a$

BSFA = 0, BSIA = 1 → $f_{S}_a = 0$

BSFA = 1, BSIA = 1 → $f_{S}_a = 1 \quad src_a$

BDFA = 0, BDIA = 0 → $f_{D}_a = 1$

BDFA = 1, BDIA = 0 → $f_{D}_a = src_a$

BDFA = 0, BDIA = 1 → $f_{D}_a = 0$

BDFA = 1, BDIA = 1 → $f_{D}_a = 1 \quad src_a$

**Table 39.9** lists all possible alpha channel blend modes.

Table 39.10 Alpha Channel Blend Modes

| Mode | BSF | BSI | BDF | BDI | Blend Equation |
|---|---|---|---|---|---|
| SRC_A_ONE DST_A_ONE | 0 | 0 | 0 | 0 | SRC_A + DST_A |
| SRC_A_ONE | 0 | 0 | 0 | 1 | SRC_A |
| SRC_A_ONE DST_A_SRC_A | 0 | 0 | 1 | 0 | SRC_A + DST_A × SRC_A |
| SRC_A_ONE DST_A_ONE_MINUS_SRC_A | 0 | 0 | 1 | 1 | SRC_A + DST_A × (1 SRC_A) |
| SRC_A_ZERO DST_A_ONE | 0 | 1 | 0 | 0 | DST_A |
| SRC_A_ZERO DST_A_ZERO | 0 | 1 | 0 | 1 | 0 |
| SRC_A_ZERO DST_A_SRC_A | 0 | 1 | 1 | 0 | DST_A × SRC_A |
| SRC_A_ZERO DST_A_ONE_MINUS_SRC_A | 0 | 1 | 1 | 1 | DST_A × (1 SRC_A) |
| SRC_A_SRC_A DST_A_ONE | 1 | 0 | 0 | 0 | SRC_A × SRC_A + DST_A |
| SRC_A_SRC_A | 1 | 0 | 0 | 1 | SRC_A × SRC_A |
| SRC_A_SRC_A DST_A_SRC_A | 1 | 0 | 1 | 0 | SRC_A × SRC_A + DST_A × SRC_A |
| SRC_A_SRC_A DST_A_ONE_MINUS_SRC_A | 1 | 0 | 1 | 1 | SRC_A × SRC_A + DST_A × (1 SRC_A) |
| SRC_A_ONE_MINUS_SRC_A DST_A_ONE | 1 | 1 | 0 | 0 | SRC_A × (1 SRC_A) + DST_A |
| SRC_A_ONE_MINUS_SRC_A | 1 | 1 | 0 | 1 | SRC_A × (1 SRC_A) |
| SRC_A_ONE_MINUS_SRC_A DST_A_SRC_A | 1 | 1 | 1 | 0 | SRC_A × (1 SRC_A) + DST_A × SRC_A |
| SRC_A_ONE_MINUS_SRC_A DST_A_ONE_MINUS_SRC_A | 1 | 1 | 1 | 1 | SRC_A × (1 SRC_A) + DST_A × (1 SRC_A) |

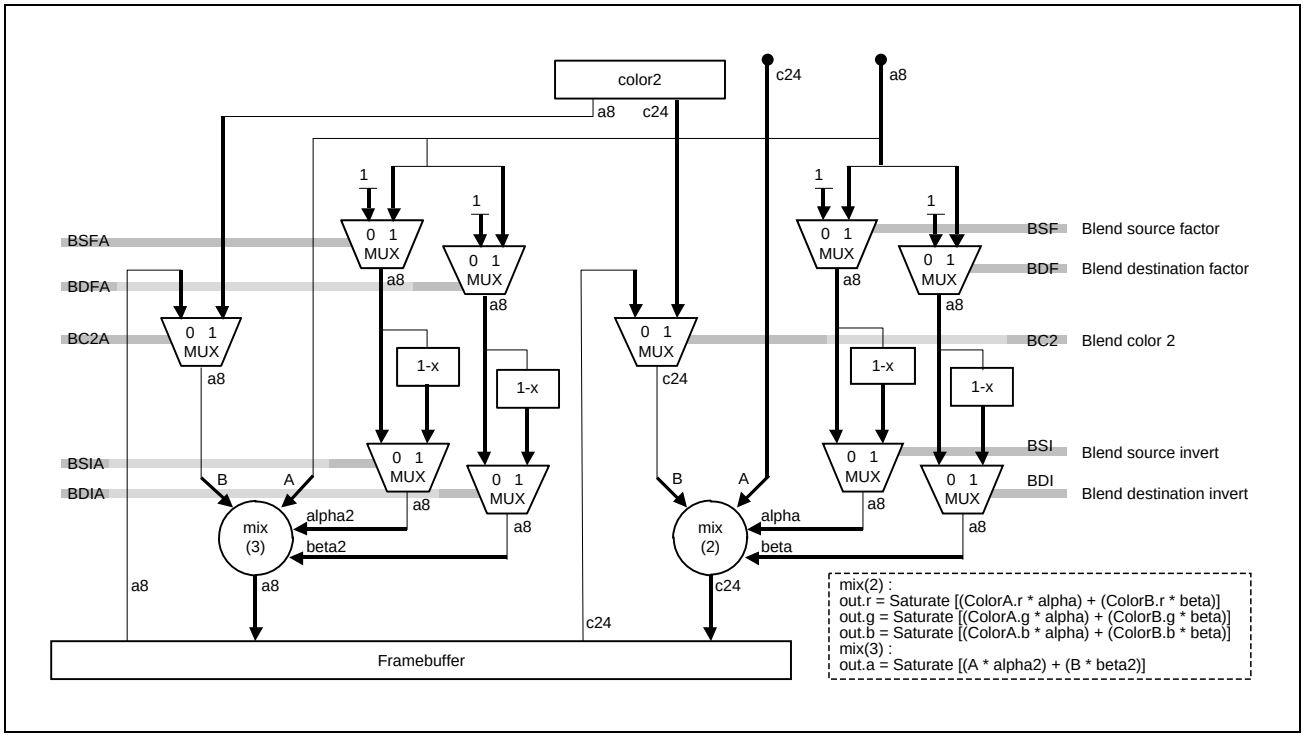



Figure 39.24 Alpha- and Color-Channel Blend Unit when CONTROL2.USEACB = 1

## 39.7 Rendering Modes

The rendering process can be performed in two different modes, register mode and display list mode.

### 39.7.1 Register Mode

In register mode, when operation is based on register settings, the host CPU configures and initiates each render process separately. To start a new render process, the host CPU must wait until the previous one is completed. In this mode, the host CPU is heavily engaged throughout the entire drawing procedure and is consequently to a large extent unavailable for other tasks.

The host CPU must set up all registers for performing a certain drawing operation before it can start the rendering process. A new register setup can only be started when the previous render process has completed. Before starting a new register setup, make sure that:

- STATUS.DLISTACTIVE = 0: Display list reader is idle
- STATUS.BUSYENUM = 0: Pixel selection unit is idle

Lastly, write the framebuffer start address to the ORIGIN register. This write triggers the 2D Drawing Engine to start rendering.

### 39.7.2 Display List Mode

In display list mode, the host CPU creates a display list in memory prior to starting the 2D Drawing Engine. A display list contains a bundle of render operations. When started, the 2D Drawing Engine executes the display list autonomously, and the host CPU is not involved with drawing operations most of the time. Use of a display list allows for fully asynchronous operation of the host CPU and the 2D Drawing Engine and offers the best possible system performance.

In this mode, the display list reader reads a memory block containing instructions on how to set the 2D Drawing Engine control registers and executes these control register writes accordingly.

#### Display list start

To start execution of a display list, which already resides in memory, the start address of the display list is written to the display list start address register DLISTSTART. Because rewriting DLISTSTART also stops any ongoing display list execution, make sure that the previous display list process is completed by either of the following two methods:

- Check that STATUS.DLISTACTIVE = 0, which indicates idle status of the display list reader
- Wait for the display list interrupt DRWDLISTIRQ, which indicates completion of the previous display list process

NOTE

Direct writing to 2D Drawing Engine registers while the display list mode is active (STATUS.DLISTACTIVE = 1) might lead to a 2D Drawing Engine hang up. To prevent this, always check that the display list reader is idle (STATUS.DLISTACTIVE = 0) before writing to any 2D Drawing Engine register.

### Display list format

Display lists are stored using direct register-to-value mappings, which means that a display list contains a one-byte index that addresses a certain register and the value to be written to the register. The register index is derived from the address offset of the register address and can be calculated by dividing the address offset by 4. For the index of each register, see **Table 39.11**.

As the 2D Drawing Engine registers are always 32 bits wide, each data unit (called a data word) to be written to a register is of the same size. An address word that contains the indices of the register to be written is stored in a packed notation with up to four indices stored in one 32-bit address word.

A display list command always starts with an address word, followed by up to four data words, one for each register. The indices are read and interpreted from LSB to MSB, so the register of the low byte index is written first.

### Example)

In the following example:

```
DWORD H’201A 1930   //  start of list address word
DWORD H’0000 0013   //  data word 1 (for register H’30)
DWORD H’FFFF FFAA //  data word 2 (for register H’19)
DWORD H’4033 6480   //  data word 3 (for register H’1A)
DWORD H’0001 0000   //  data word 4 (for register H’20)
DWORD…              //  next address word
```

This stream of dwords updates the DRW registers as follows:

- Write H’0000 0013 to register H’30 = 48, which is IRQCTL
- Write H’FFFF FFAA to register H’19 = 25, which is COLOR1
- Write H’4033 6480 to register H’1A = 26, which is COLOR2
- Write H’0001 0000 to register H’20 = 32, which is ORIGIN.

### Address word indices

Besides referencing a register, the indices of an address word can also have other meanings, depending on their value.

Table 39.11 Indices Function

| Index | Function |
|---|---|
| H'00 to H'7F | Register indices<br>Two register indices trigger additional actions:<br>H'20 = 32: A write to ORIGIN to set a new frame buffer address is delayed until the ongoing frame buffer write-back is complete, when STATUS.BUSYWRITE = 0.<br>H'32 = 50: A write to DLISTSTART sets a new display list start address stops the current display list and starts the new one. |
| H'80 | Gap index, which is used to fill unused bytes of an address word. For example, if fewer than four indices are required, the remaining bytes are filled with H'80. In this case, the number of the subsequent data words must be reduced accordingly. |
| H'FF | If the first index of an address word contains the special index H'FF, the subsequent (second) index is interpreted as follows:<br>Bit [0] set: Display list end.<br>Bit [1] set: Issue a full pipeline flush and wait (necessary before flip).<br>Bit [2] set: Wait for writeback complete (necessary before framebuffer format change).<br>Bits [7:3] Set all to 0s.<br>Bit [1] and [2] settings are mutually exclusive. All indices after the special index H'FF are ignored, and the next address word is read, if no display list end (bit [0] = 1) was set.<br>The remaining two indices must be set to H'00. |

**Note:** Gap indices H'80 must not be placed between other indices, for example as "index1 – H'80 - index3 - index4". Always fill all indices after H'80 with the gap index.
If any of the special indices H'80 and H'FF are used, no register index can follow after them in the same address word.

Table 39.12 2D Drawing Engine Registers Overview (1/2)

| Register Function | Symbol | Index |
|---|---|---|
| Control registers: | | |
| Geometry control | CONTROL | 0 |
| Surface control | CONTROL2 | 1 |
| Interrupt control | IRQCTL | 48 |
| Cache control | CACHECTL | 49 |
| Status control | STATUS | n.a. |
| Hardware version and feature set ID | HWREVISION | n.a. |
| Color registers: | | |
| Base color | COLOR1 | 25 |
| Secondary color | COLOR2 | 26 |
| Pattern | PATTERN | 29 |
| Limiter registers: | | |
| Limiter 1 start value | L1START | 4 |
| Limiter 2 start value | L2START | 5 |
| Limiter 3 start value | L3START | 6 |
| Limiter 4 start value | L4START | 7 |
| Limiter 5 start value | L5START | 8 |
| Limiter 6 start value | L6START | 9 |
| Limiter 1 x-axis increment | L1XADD | 10 |
| Limiter 2 x-axis increment | L2XADD | 11 |
| Limiter 3 x-axis increment | L3XADD | 12 |
| Limiter 4 x-axis increment | L4XADD | 13 |
| Limiter 5 x-axis increment | L5XADD | 14 |
| Limiter 6 x-axis increment | L6XADD | 15 |
| Limiter 1 y-axis increment | L1YADD | 16 |
| Limiter 2 y-axis increment | L2YADD | 17 |
| Limiter 3 y-axis increment | L3YADD | 18 |
| Limiter 4 y-axis increment | L4YADD | 19 |
| Limiter 5 y-axis increment | L5YADD | 20 |
| Limiter 6 y-axis increment | L6YADD | 21 |
| Limiter 1 bandwidth parameter | L1BAND | 22 |
| Limiter 2 bandwidth parameter | L2BAND | 23 |
| Texture registers: | | |
| Texture base address | TEXORIGIN | 47 |
| Texels per texture line | TEXPITCH | 45 |
| Texture size or texture address mask | TEXMASK | 46 |
| U limiter start value | LUSTART | 36 |
| U limiter x-axis increment | LUXADD | 37 |
| U limiter y-axis increment | LUYADD | 38 |
| V limiter start value integer part | LVSTARTI | 39 |
| V limiter start value fractional part | LVSTARTF | 40 |
| V limiter x-axis increment integer part | LVXADDI | 41 |
| V limiter y-axis increment integer part | LVYADDI | 42 |
| V limiter increment fractional parts | LVYXADDF | 43 |
| Color lookup table start address | TEXCLADDR | 55 |

Table 39.12 2D Drawing Engine Registers Overview (2/2)

| Register Function | Symbol | Index |
|---|---|---|
| Write Data to TEXCLADDR; after each data write, TEXCLADDR is incremented by 1. | TEXCLDATA | 56 |
| Offset to the index for the indexed texture formats i8, i4, i2, and i1 | TEXCLOFFSET | 57 |
| Compare value for R,G, B components of internal texel color representation. | COLKEY | 58 |
| Miscellaneous registers: | | |
| Bounding box dimension | SIZE | 30 |
| Framebuffer pitch and spanstore delay | PITCH | 31 |
| Address of the first pixel in framebuffer | ORIGIN | 32 |
| Display list start address | DLISTSTART | 50 |
| Performance counters control | PERFTRIGGER | 53 |
| Performance counter 1 | PERFCOUNT1 | 51 |
| Performance counter 2 | PERFCOUNT2 | 52 |

Note 1. These registers are read-only and cannot be accessed in display list mode, and they therefore have no index.

### 39.7.3 Stopping the Render Process

Stopping an ongoing render process requires a specific procedure, which is described in **Section 39.10, Stopping the 2D Drawing Engine Render Process**.

## 39.8 Interrupts

The 2D Drawing Engine generates three interrupts:

- DRWBUSIRQ
- DRWENUMIRQ
- DRWDLISTIRQ.

### 39.8.1 Interrupt Sources

#### DRWBUSIRQ

This is the 2D Drawing Engine bus error interrupt. It occurs when the 2D Drawing Engine attempts to access an undefined address range through the following:

- Framebuffer Base Address Register (ORIGIN)
- Texture Base Address Register (TEXORIGIN)
- Display List Start Address Register (DLISTSTART)

The interrupt source can be determined by the BUSERRMFB, BUSERRMTXMRL, and BUSERRMDL bits of the STATUS register.

NOTE

After a DRWBUSIRQ occurrence, you must apply a software reset.
For a software reset, see **Section 42, Low Power Mode**.

#### DRWENUMIRQ

This is the current render process finished interrupt.

#### DRWDLISTIRQ

This is the display list interrupt. It is asserted on completion of a display list process. DRWDLISTIRQ is activated if one of the following is true:

- The entire display list is complete
- The display list processing stops because a new display list start is triggered by a write to the Display List Start Address Register (DLISTSTART).

### 39.8.2 Interrupt Control

The three 2D Drawing Engine interrupts are combined into a single shared interrupt, DRW_IRQ, to the CPU. Each interrupt can be masked (disabled), or unmasked (enabled) by setting its associated enable bit in the Interrupt Control Register (IRQCTL).

The occurrence of an enabled interrupt (one with its mask bit set to 1 in IRQCTL) is monitored in the Status Control Register (STATUS) with its associated interrupt status bit is set to 1. The shared 2D Drawing Engine interrupt DRW_IRQ is then generated.

To clear the interrupt, the host CPU must write 1 to the interrupt clear bit in IRQCTL. The interrupt clear bit returns to 0 automatically.

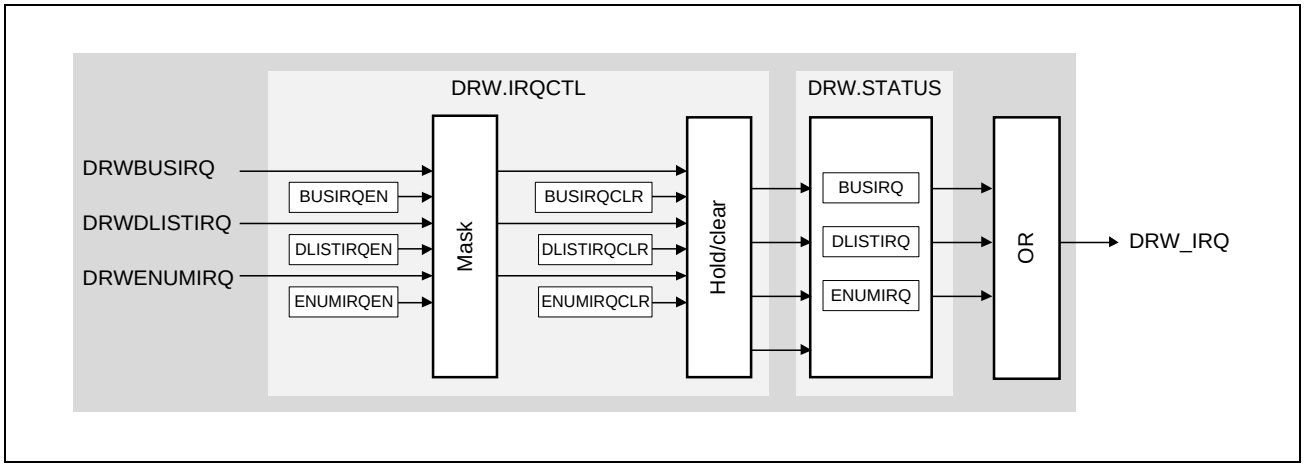



Figure 39.25 Interrupt Controller Unit (ICU)

## 39.9 Performance Counters

The 2D Drawing Engine features two independent 32-bit performance counter registers (PERFCOUNTk (k = 1, 2)) to count the number of occurrences of a certain event. The events to be counted can be set up independently for each performance counter register with the Performance Counter Control Registers, PERFTRIGGER.PERFTRIGGER2 for PERFCOUNT2 and PERFTRIGGER.PERFTRIGGER1 for PERFCOUNT1.

**Table 39.13** lists the performance counter trigger events that can be selected.

Table 39.13 Performance Counter Trigger Events

| PERFTRIGGER.PERFTRIGGERk | Event |
|---|---|
| 0 | Disable performance counter |
| 1 | 2D Drawing Engine active cycles |
| 2 | Framebuffer read access |
| 3 | Framebuffer write access |
| 4 | Texture read access |
| 5 | Invisible pixels (enumerated but selected with alpha 0%) |
| 6 | Invisible pixels while internal FIFO is empty (lost cycles) |
| 7 | Display list reader active cycles |
| 8 | Framebuffer read hits |
| 9 | Framebuffer read misses |
| 10 | Framebuffer write hits |
| 11 | Framebuffer write misses |
| 12 | Texture read hits |
| 13 | Texture read misses |
| 31 | Every clock cycle (for use as timer) |

## 39.10 Stopping the 2D Drawing Engine Render Process

If a render process has started either in register or display list mode, the 2D Drawing Engine processes the data autonomously until the render process is finished. Depending on the rendering, this process might take several milliseconds.

If the 2D Drawing Engine is to be disabled because, for example, this LSI enters a low-power mode, proceed as follows to stop the ongoing rendering:

1. Set the following registers as follows:
   SIZE = H’0001 0001
   Set bounding box dimensions to 1 pixel 1 line.
   CONTROL2 = H’0000 0000
   Color format a (8), no texture, CLUT.
   ORIGIN = UnmappedAddress
   The UnmappedAddress is an address that is not available for 2D Drawing Engine access.
   The recommended UnmappedAddress is given here under the key word “UnmappedAddress”.
   Alternatively do the same in display list mode:
   DWORD H’8020 011E // start of list address word
   DWORD H’0001 0001 // SIZE = H’0001 0001
   DWORD H’0000 0000 // CONTROL2 = H’0000 0000
   DWORD UnmappedAddress // ORIGIN = UnmappedAddress

2. Wait for the Bus Error corresponding to the unmapped address violation, which indicates access to an unmapped address and the stop of the render process.
   UnmappedAddress = H’D000 0000

3. Disable the 2D Drawing Engine, if wanted.

# 40. SD Host Interface

NOTE

Development of the SD host--related products needs the conclusion of the following agreement. SD Host/Ancillary Product License Agreement (SD HALA)

## 40.1 Overview

### 40.1.1 Features

- 1 channels
- Channel 0 supports SDHI
- SD memory/IO card interface (1-bit/4-bit SD bus)
- SD, SDHC, and SDXC SD memory card access supported
- Default, high-speed, UHS-I/SDR50, and SDR104 transfer modes supported
- SD clock (SD_CLK) frequency = SD0ϕ/4 frequency/$2^n$ (n = 0 to 9)
- Error check function: CRC7 (for command/response), CRC16 (for data)
- Interrupt request: 2
- Card detect function
- Write protect supported
- Backward-compatible, high-speed, transfer modes supported
- High-priority interrupt (HPI) supported

### Block Diagram

**Figure 40.1** shows a block diagram of the SD host interfaces.

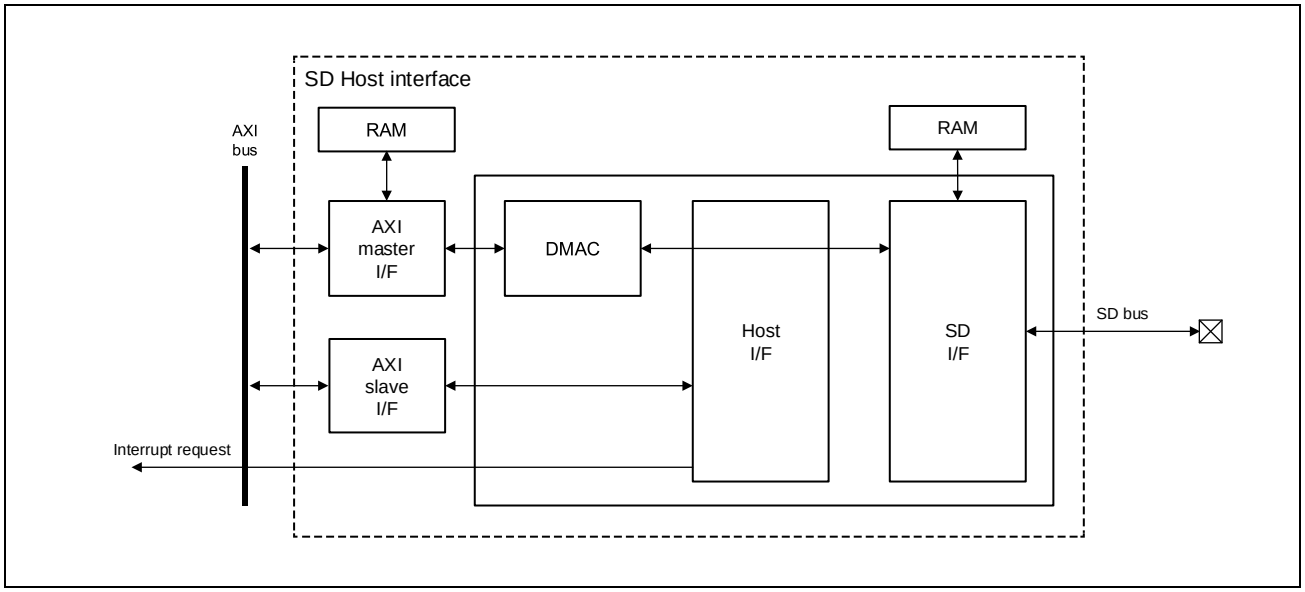



Figure 40.1 Block Diagram of SD Host Interface

### 40.1.2 External Pins

**Table 40.1** lists the input and output pins used by the interface. The operating voltage on the pins of the SD host interface is 3.3 V or 1.8 V. Before using the pins of the SD host interface, be sure to set the SD ch0 IO Voltage Mode Control Register (SD_ch0) and Driving Ability Control Register (IOLH).

For details, see the following sections:

**Section 41.3.7, Driving Ability Control Register**,
**Section 41.3.14, SD ch0 IO Voltage Mode Control Register**,

Table 40.1 Pin Configuration

| PIN Name | I/O | Function |
|---|---|---|
| SD0_CLK | O | SD clock output |
| SD0_CMD | I/O | SD command output, response input |
| SD0_DATA0 | I/O | SD Data 0 [bit 0] |
| SD0_DATA1 | I/O | SD Data 1 [bit 1], SDIO interrupt |
| SD0_DATA2 | I/O | SD Data 2 [bit 2], read wait |
| SD0_DATA3 | I/O | SD Data 3 [bit 3], Card detection |
| SD0_CD | I | SD card detection*1 |
| SD0_WP | I | SD write protection*1 |

Note 1. Fix to 1 when not in use.

### 40.1.3 Register Configuration

The base addresses is as follows.

Channel 0:
H'0_11C0_0000 (Cortex-A55 Address Space)

Table 40.2 Register Configurations (1/2)

| Name | Abbreviation | Address | Access Width | Mirror |
|---|---|---|---|---|
| Command type register | SD_CMD | H'0000 | 16/32/64 | |
| Command argument registers | SD_ARG | H'0010 | 16/32/64 | |
| | SD_ARG1 | H'0018 | 16/32/64 | SD_ARG[31:16] |
| Data stop register | SD_STOP | H'0020 | 16/32/64 | |
| Block count register | SD_SECCNT | H'0028 | 16/32/64 | |
| Card response registers | SD_RSP10 | H'0030 | 16/32/64 | |
| | SD_RSP1 | H'0038 | 16/32/64 | SD_RSP10[31:16] |
| | SD_RSP32 | H'0040 | 16/32/64 | SD_RSP10[63:32] |
| | SD_RSP3 | H'0048 | 16/32/64 | SD_RSP32[31:16] |
| | SD_RSP54 | H'0050 | 16/32/64 | |
| | SD_RSP5 | H'0058 | 16/32/64 | SD_RSP54[31:16] |
| | SD_RSP76 | H'0060 | 16/32/64 | SD_RSP54[63:32] |
| | SD_RSP7 | H'0068 | 16/32/64 | SD_RSP76[31:16] |
| SD card interrupt flag register 1 | SD_INFO1 | H'0070 | 16/32/64 | |
| SD card interrupt flag register 2 | SD_INFO2 | H'0078 | 16/32/64 | |
| SD_INFO1 interrupt mask register | SD_INFO1_MASK | H'0080 | 16/32/64 | |
| SD_INFO2 interrupt mask register | SD_INFO2_MASK | H'0088 | 16/32/64 | |
| SD clock control register | SD_CLK_CTRL | H'0090 | 16/32/64 | |
| Transfer data length register | SD_SIZE | H'0098 | 16/32/64 | |
| Card access control option register | SD_OPTION | H'00A0 | 16/32/64 | |
| SD error status register 1 | SD_ERR_STS1 | H'00B0 | 16/32/64 | |
| SD error status register 2 | SD_ERR_STS2 | H'00B8 | 16/32/64 | |
| SD buffer read/write register | SD_BUF0 | H'00C0 | 16/32/64 | |
| SDIO mode control register | SDIO_MODE | H'00D0 | 16/32/64 | |
| SDIO interrupt flag register | SDIO_INFO1 | H'00D8 | 16/32/64 | |
| SDIO_INFO1 interrupt mask register | SDIO_INFO1_MASK | H'00E0 | 16/32/64 | |
| DMA mode enable register | CC_EXT_MODE | H'0360 | 16/32/64 | |
| Software reset register | SOFT_RST | H'0380 | 16/32/64 | |
| Version register | VERSION | H'0388 | 16/32/64 | |
| Host interface mode setting register | HOST_MODE | H'0390 | 16/32/64 | |
| SD status register*1 | SD_STATUS | H'03C8 | 16/32/64 | |
| DMAC mode register | DM_CM_DTRAN_MODE | H'0820 | 16/32/64 | |
| DMAC control register | DM_CM_DTRAN_CTRL | H'0828 | 16/32/64 | |
| DMAC reset register | DM_CM_RST | H'0830 | 16/32/64 | |
| DMAC interrupt register 1 | DM_CM_INFO1 | H'0840 | 16/32/64 | |
| DM_CM_INFO1 interrupt mask register | DM_CM_INFO1_MASK | H'0848 | 16/32/64 | |
| DMAC interrupt register 2 | DM_CM_INFO2 | H'0850 | 16/32/64 | |

Table 40.2 Register Configurations (2/2)

| Name | Abbreviation | Address | Access Width | Mirror |
|---|---|---|---|---|
| DM_CM_INFO2 interrupt mask register | DM_CM_INFO2_MASK | H'0858 | 16/32/64 | |
| DMAC address register | DM_DTRAN_ADDR | H'0880 | 16/32/64 | |
| SCC register area*2 | — | — | — | — |

Note 1. Only for channel 0.

Note 2. Refer to **Section 40.7, SCC Register Descriptions**.

## 40.2 Register Description

### 40.2.1 Command Type Register (SD_CMD)

The command type register (SD_CMD) is used to select the command type and response type. The command sequence is started by writing to SD_CMD.

For details on the SD_CMD setting, refer to **Section 40.4.14, Example of SD_CMD Register Setting**.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 63 to 16 | — | All 0 | R | These bits are always read as 0. The write value should always be 0.<br>Operation is not guaranteed if a value other than 0 is written to these bits. |
| 15 | MD7 | 00 | R/W | Multiple Block Transfer Mode (enabled at multiple block transfer) |
| 14 | MD6 | | | 00: CMD12 is automatically issued at multiple block transfer.<br>01: CMD12 is not automatically issued at multiple block transfer.<br>10: Setting prohibited<br>11: Setting prohibited |
| 13 | MD5 | 0 | R/W | Single/Multiple Block Transfer (enabled when the command with data is handled)<br>0: Single block transfer<br>1: Multi block transfer |
| 12 | MD4 | 0 | R/W | Write/Read Mode (enabled when the command with data is handled)<br>0: Write (SD host interfaces -> SD card)<br>1: Read (SD host interfaces <- SD card) |
| 11 | MD3 | 0 | R/W | Data Mode (Command Type)<br>0: Command without data transfer (bc, bcr, ac)<br>1: Command with data transfer (adtc) |
| 10 | MD2 | 000 | R/W | Mode/Response Type |
| 9 | MD1 | | | 000: Normal mode |
| 8 | MD0 | | | The response type and the transfer mode are selected by SD_CMD[7:0], and the SD_CMD[15:11] setting is disabled.<br>001: Setting prohibited<br>010: Setting prohibited<br>011: Extended mode/No response<br>100: Extended mode/R1, R5, R6, or R7 response from the SD card<br>101: Extended mode/R1b response from the SD card<br>110: Extended mode/R2 response from the SD card<br>111: Extended mode/R3 or R4 response from the SD card<br>Some commands cannot be used in normal mode. For details, see **Section 40.4.14, Example of SD_CMD Register Setting** to select mode/response type. |
| 7 | C1 | 00 | R/W | 00: CMD |
| 6 | C0 | | | 01: ACMD<br>10: Setting prohibited<br>11: Setting prohibited |
| 5 | CF45 | 000000 | R/W | Command Index<br>These bits specify Command Format[45:40] (command index).<br>[Examples]<br>CMD6: SD_CMD[7:0] = 0000 0110b<br>CMD18: SD_CMD[7:0] = 0001 0010b<br>ACMD13: SD_CMD[7:0] = 0100 1101b |
| 4 | CF44 | | | |
| 3 | CF43 | | | |
| 2 | CF42 | | | |
| 1 | CF41 | | | |
| 0 | CF40 | | | |

**Note:** SD_CMD cannot be written to when the CBSY bit in SD_INFO2 is 1.

### 40.2.2 Command Argument Register (SD_ARG)

Command arguments for SD cards are set in the SD command argument registers (SD_ARG). Set the command arguments before writing to SD_CMD.

Note that the argument of CMD12 within command sequences is H'0000 0000 regardless of the setting of SD_ARG.

SD_ARG

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 62 to 32 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0.<br>Operation is not guaranteed if a value other than 0 is written to these bits. |
| 31 to 0 | CF39 to CF8 | All 0 | R/W | Set command format[39:8] (argument). |

SD_ARG1 (Mirror of SD_ARG[31:16])

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 63 to 16 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0.<br>Operation is not guaranteed if a value other than 0 is written to these bits. |
| 15 to 0 | CF39 to CF24 | All 0 | R/W | Set command format[39:24] (argument). |

### 40.2.3 Data Stop Register (SD_STOP)

The data stop register (SD_STOP) is used to enable or disable block counting at multiple block transfer, and to control the issuing of CMD12 within command sequences.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 63 to 19 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. Operation is not guaranteed if a value other than 0 is written to these bits. |
| 18 | — | 0*1 | R/W | Reserved<br>The write value should always be 0. |
| 17 | HPIMODE | 0*1 | R/W | HPI Mode Enable<br>0: Disables HPI mode.<br>1: Enables HPI mode. |
| 16 | HPICMD | 0*1 | R/W*3 | HPI Command Issue<br>When HPICMD is set to 1 while HPIMODE is 1, the HPI command (CMD12) is issued. This bit is cleared to 0 when reception of the response to CMD12 is completed.<br>The timing with which this bit is set to 1 is as follows.<br>• After reception of the response to CMD12 that was issued by setting the STP bit to 1 has been completed during the CMD6/CMD38 or CMD25 sequence.<br>• After reception of the response to CMD24/CMD25 has been completed<br>After HPICMD is set to 1, do not write 0 to this bit while the CBSY bit in SD_INFO2 is 1. Do not set this bit to 1 when the CBSY bit in SD_INFO2 is 0. |
| 15 to 9 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. Operation is not guaranteed if a value other than 0 is written to these bits. |
| 8 | SEC | 0*1 | R/W | Block Count Enable*2<br>0: Disables SD_SECCNT setting value.<br>1: Enables SD_SECCNT setting value.<br>Set SEC to 1 at multiple block transfer.<br>When SD_CMD is set as follows to start the command sequence while SEC is set to 1, CMD12 is automatically issued to stop multi-block transfer with the number of blocks which is set to SD_SECCNT.<br>1. CMD18 or CMD25 in normal mode (SD_CMD[10:8] = 000)<br>2. SD_CMD[15:13] = 001 in extended mode (CMD12 is automatically issued, multiple block transfer)<br>When the command sequence is halted because of a communications error or timeout, CMD12 is not automatically issued. |
| 7 to 1 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. Operation is not guaranteed if a value other than 0 is written to these bits. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 0 | STP | 0*1 | R/W | Stop<br>● When STP is set to 1 during multiple block transfer, CMD12 is issued to halt the transfer through the SD host interface.<br>However, if a command sequence is halted because of a communications error or timeout, CMD12 is not issued. Although continued buffer access is possible even after STP has been set to 1, the buffer access error bit (ERR5 or ERR4) in SD_INFO2 will be set accordingly.<br>● When STP has been set to 1 during transfer for single block write, the access end flag is set when SD_BUF becomes empty, and CMD12 is not issued. If SD_BUF does contain data, the access end flag is set on completion of reception of the busy state without CMD12 having been issued.<br>● When STP has been set to 1 during transfer for single block read, the access end flag is set immediately after setting of the STP bit and CMD12 is not issued.<br>● When STP is set to 1 during reception of the busy state after an R1b response, the access end flag is set on completion of reception of the busy state without CMD12 having been issued.<br>● When STP is set to 1 after a command sequence has been completed, CMD12 is not issued and the access end flag is not set.<br>● Set STP to 1 after the response end flag has been set.<br>● Set STP to 0 after the response end flag has been set. |

Note 1. The initial value is applied at a reset and when the SDRST bit in SOFT_RST is 0.

Note 2. Do not change the value of this bit when the CBSY bit in SD_INFO2 is set to 1.

Note 3. It is effective only if 1 is written.

### 40.2.4 Block Count Register (SD_SECCNT)

The block count register (SD_SECCNT) is used to specify the number of transfer blocks at multiple block transfer.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 63 to 32 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. Operation is not guaranteed if a value other than 0 is written to these bits. |
| 31 to 0 | CNT31 to CNT0 | All 0 | R/W | Number of Transfer Blocks*1<br>When H'0000 0001 is set, the number of transfer blocks is 1.<br>:<br>When H'0000 FFFF is set, the number of transfer blocks is 65535.<br>:<br>When H'FFFF FFFF is set, the number of transfer blocks is 4294967295.<br>Do not set this register to H'0000 0000 if multiple blocks are to be transferred. |

Note 1. Do not change the value of these bits when the CBSY bit in SD_INFO2 is set to 1.

### 40.2.5 SD Card Response Registers (SD_RSP)

The SD card response registers (SD_RSP) hold the response from the SD card.

SD_RSP10

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 63 to 0 | R71 to R8 | All 0 | R | Hold the response from the SD card |

SD_RSP1 (Mirror of SD_RSP10[31:16])

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 63 to 16 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0.<br>Operation is not guaranteed if a value other than 0 is written to these bits. |
| 15 to 0 | R39 to R24 | All 0 | R | Hold the response from the SD card |

SD_RSP32 (Mirror of SD_RSP10[63:32])

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 63 to 32 | — | All0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0.<br>Operation is not guaranteed if a value other than 0 is written to these bits. |
| 31 to 0 | R71 to R40 | All 0 | R | Hold the response from the SD card |

SD_RSP3 (Mirror of SD_RSP32[31:16])

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 63 to 16 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0.<br>Operation is not guaranteed if a value other than 0 is written to these bits. |
| 15 to 0 | R71 to R56 | All 0 | R | Hold the response from the SD card |

SD_RSP54

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 63 to 56 | — | All 0 | R | These bits are always read as 0. The write value should always be 0.<br>Operation is not guaranteed if a value other than 0 is written to these bits. |
| 55 to 0 | R127 to R72 | All 0 | R | Hold the response from the SD card |

SD_RSP5 (Mirror of SD_RSP54[31:16])

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 63 to 16 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0.<br>Operation is not guaranteed if a value other than 0 is written to these bits. |
| 15 to 0 | R103 to R88 | All 0 | R | Hold the response from the SD card |

SD_RSP76 (Mirror of SD_RSP54[63:32])

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 63 to 24 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0.<br>Operation is not guaranteed if a value other than 0 is written to these bits. |
| 23 to 0 | R127 to R104 | All 0 | R | Hold the response from the SD card |

SD_RSP7 (Mirror of SD_RSP76[31:16])

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 63 to 8 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0.<br>Operation is not guaranteed if a value other than 0 is written to these bits. |
| 7 to 0 | R127 to R120 | All 0 | R | Hold the response from the SD card |

**Table 40.3** lists the response types and corresponding SD_RSP registers.

Table 40.3 Response Types and Corresponding SD_RSP Registers

| Response Types | SD_RSP Registers |
|---|---|
| R1, R1b[39:8] | SD_RSP10<br>SD_RSP54*1 |
| R2[127:8] | SD_RSP54 and SD_RSP10 |
| R3[39:8] | SD_RSP10 |
| R4[39:8] | SD_RSP10 |
| R5[39:8] | SD_RSP10 |
| R6[39:8] | SD_RSP10 |
| R7[39:8] | SD_RSP10 |

Note 1. The response to CMD18 and to CMD25 is stored in both R[39:8] and R[103:72].
This makes it possible to confirm the response to CMD18 and CMD25 by reading R[103:72] even if the response to automatic CMD12 is stored in R[39:8].

### 40.2.6 SD Card Interrupt Flag Register (SD_INFO1)

The SD card interrupt flag register 1 (SD_INFO1) indicates the response end and access end in the command sequence. This register also indicates the card detect/write protect state.

For CMD12 and CMD52 (SDIO abort) at multiple block transfer, INFO0 is not set but only INFO2 is set.

Even if the command sequence is halted because of a communications error or timeout, INFO0 or INFO2 is set.

INFO10, INFO9, and INFO8 change depending on the SDDAT3 state after a reset is released and continue to change in 4-bit transfer mode.

To clear a flag, write 0 to the bit to be cleared and 1 to the other bits.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 63 to 17 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. Operation is not guaranteed if a value other than 0 is written to these bits. |
| 16 | HPIRES | 0*2 | R/W*1 | Response Reception Completion<br>[Setting condition]<br>When reception of the response to CMD12 that was issued by setting the STP bit to 1 is completed during the CMD6/CMD38 or CMD25 sequence in HPI mode.<br>[Clearing condition]<br>When 0 is written to HPIRES |
| 15 to 11 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. Operation is not guaranteed if a value other than 0 is written to these bits. |
| 10 | INFO10 | Unknown | R | Indicates the SDDAT3 state.<br>0: SDDAT3 is set to 0.<br>1: SDDAT3 is set to 1. |
| 9 | INFO9 | 0 | R/W*1 | SDDAT3 Card Insertion<br>[Setting condition]<br>After change in SDDAT3 from 0 to 1, two cycles of 2 P1φ have elapsed with SDDAT3 held 1.<br>[Clearing condition]<br>When 0 is written to INFO9 |
| 8 | INFO8 | 0 | R/W*1 | SDDAT3 Card Removal<br>[Setting condition]<br>After change in SDDAT3 from 1 to 0, two cycles of 2 P1φ have elapsed with SDDAT3 held 0.<br>[Clearing condition]<br>When 0 is written to INFO8 |
| 7 | INFO7 | Unknown | R | Write Protect<br>Indicates the ISDWP state.<br>0: ISDWP is set to 1.<br>1: ISDWP is set to 0. |
| 6 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. Operation is not guaranteed if a value other than 0 is written to this bit. |
| 5 | INFO5 | Unknown | R | Indicates the ISDCD state.<br>0: Indicates that Mcycle has elapsed with ISDCD held 1.<br>1: Indicates that Mcycle has elapsed with ISDCD held 0.<br>Mcycle is set by bits 3 to 0 in SD_OPTION. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 4 | INFO4 | 0 | R/W*1 | ISDCD Card Insertion<br>[Setting condition]<br>After change in ISDCD from 1 to 0, Mcycle has elapsed with ISDCD held 0.<br>[Clearing condition]<br>When 0 is written to INFO4<br>Mcycle is set by bits 3 to 0 in SD_OPTION. |
| 3 | INFO3 | 0 | R/W*1 | ISDCD Card Removal<br>[Setting condition]<br>After change in ISDCD from 0 to 1, Mcycle has elapsed with ISDCD held 1.<br>[Clearing condition]<br>When 0 is written to INFO3<br>Mcycle is set by bits 3 to 0 in SD_OPTION. |
| 2 | INFO2 | 0*2 | R/W*1 | Access End<br>[Setting conditions]<br>1. When read access to the buffer is completed in the case of transfer for single block read<br>2. When read access to the buffer for the last block of data is completed in the case of transfer for multiple block read<br>3. When read access to the buffer and reception of the response to CMD12 are completed in the case of transfer for multiple block read with automatic issuing of CMD12<br>4. When reception of the busy state after reception of the CRC status is completed in the case of transfer for single block write<br>5. When reception of the busy state after reception of the CRC status of the last block of data is completed in the case of transfer for multiple block write<br>6. When reception of the response busy state for CMD12 is completed in the case of transfer for multiple block write with automatic issuing of CMD12<br>7. When reception of the response to CMD12 that was issued by setting the STP bit to1 is completed in the case of transfer for multiple block read<br>8. When reception of the response busy state for CMD12 that was issued by setting the STP bit to1 is completed in the case of transfer for multiple block write<br>9. When reception of the response to CMD52 that was issued by setting the IOABT bit to1 is completed in the case of transfer for multiple block read<br>10. When reception of the response to CMD52 that was issued by setting the IOABT bit to1 is completed in the case of transfer for multiple block write<br>In addition to the above conditions, this bit is set when a command sequence is halted because of a communications error or timeout.<br>[Clearing condition]<br>When 0 is written to INFO2<br>When the access end bit is set to 1, the command sequence is terminated. |
| 1 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. Operation is not guaranteed if a value other than 0 is written to this bit. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 0 | INFO0 | 0*2 | R/W*1 | Response End<br>[Setting conditions]<br>1. When the reception of the response is completed<br>2. When the transmission of the command not requiring a response is completed,<br>3. When receiving busy reception after R1b response<br>4. When reception of the response to CMD52 that was issued by setting the C52PUB bit to1 is completed in the case of transfer for multiple block read<br>5. When reception of the response to CMD52 that was issued by setting the C52PUB bit to1 is completed in the case of transfer for multiple block write,<br>In addition to the above conditions, this bit is set when a command sequence is halted because of a communications error or timeout.<br>[Clearing condition]<br>When 0 is written to INFO0<br>When issuing a command without data, the command sequence ends when the response end is set to 1. |

Note 1. It is effective only if 0 is written.

Note 2. The value is initialized by a reset and also in the case of a reset by the SDRST bit in SOFT_RST.

### 40.2.7 SD Card Interrupt Flag Register (SD_INFO2)

The SD card interrupt flag register 2 (SD_INFO2) indicates the access status of the SD buffer (SD_BUF) and SD card. To clear a flag, write 0 to the bit to be cleared and 1 to the other bits.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 63 to 16 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. Operation is not guaranteed if a value other than 0 is written to these bits. |
| 15 | ILA | 0*2 | R/W*1 | Illegal Access Error<br>[Setting conditions]<br>1. Writing of data to SD_CMD within a command sequence (CBSY=1)<br>2. When SD_CMD[11] = 1 (command with data transfer) and SD_CMD[7:0] = 0000 1100b (CMD12) are set in SD_CMD<br>[Clearing condition]<br>When 0 is written to ILA |
| 14 | CBSY | 0*2 | R | Command Type Register Busy<br>0: A command sequence has been completed.<br>1: A command sequence is being executed. |
| 13 | SCLKDIVEN | 1*2 | R | 0: The SD bus (CMD, DAT) is busy. Do not attempt to write to the SD_CLK_CTRL register.<br>1: The SD bus (CMD, DAT) is not busy.<br>When a command sequence is started by writing to SD_CMD, the CBSY bit is set to 1 and, at the same time, the SCLKDIVEN bit is set to 0. The SCLKDIVEN bit is set to 1 after 8 cycles of SDCLK have elapsed after setting of the CBSY bit to 0 due to completion of the command sequence. |
| 12 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. Operation is not guaranteed if a value other than 0 is written to this bit. |
| 11 | — | 0 | R/W | Reserved<br>The write value should always be 1. |
| 10 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. Operation is not guaranteed if a value other than 0 is written to this bit. |
| 9 | BWE | 0*2 | R/W*1 | SD_BUF Write Enable<br>0: Data cannot be written in SD_BUF0.<br>1: Data can be written in SD_BUF0.<br>[Setting conditions]<br>1. When SD_BUF is empty at single block transfer<br>2. When either bank 1 or bank 2 of SD_BUF is empty at multiple block transfer<br>[Clearing conditions]<br>1. When 0 is written to BWE<br>2. Writing of a block of data to SD_BUF by DMA transfer<br>When data is written to SD_BUF0 by the CPU, clear BWE and then write amount of data specified by SD_SIZE*3 |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 8 | BRE | 0*2 | R/W*1 | SD_BUF Read Enable<br>0: Data cannot be read from SD_BUF0.<br>1: Data can be read from SD_BUF0.<br>[Setting conditions]<br>1. When data set in SD_SIZE is stored in SD_BUF at single block transfer<br>2. When data set in SD_SIZE is stored in either bank 1 or bank 2 of SD_BUF at multiple block transfer<br>[Clearing conditions]<br>1. When 0 is written to BRE<br>2. Reading of a block of data from SD_BUF by DMA transfer<br>When data is read from SD_BUF0 by the CPU, clear BRE and then read amount of data specified by SD_SIZE*3.<br>Even if a CRC error or an END error occurs while block data is read, data is stored in SD_BUF and BRE is set. |
| 7 | DAT0 | Unknown | R | SDDAT0<br>Indicates the SDDAT0 state.<br>0: SDDAT0 is set to 0.<br>1: SDDAT0 is set to 1.<br>If the data timeout (ERR3) is set but the response timeout (ERR6) is not set after the Erase command has been issued, the end of the Erase sequence (DAT0 = 1) is confirmed by polling DAT0.<br>If a communications error or timeout occurs during a write sequence, the DAT0 bit may retain the value 0.<br>While the SD clock (SDCLK) is stopped, the DAT0 bit retains the value before the clock is stopped. |
| 6 | ERR6 | 0*2 | R/W*1 | Response Timeout<br>[Setting condition]<br>When a response is not received even after 640 cycles of SDCLK have elapsed (including a response to a command issued within a command sequence*5)<br>[Clearing condition]<br>When 0 is written to ERR6<br>The command sequence is halted by a response timeout.*4 |
| 5 | ERR5 | 0*2 | R/W*1 | SD_BUF Illegal Read Access<br>[Setting conditions]<br>1. When SD_BUF is empty while SD_BUF0 is read<br>2. When data with a CRC error or END error is read from SD_BUF0<br>[Clearing condition]<br>When 0 is written to ERR5 |
| 4 | ERR4 | 0*2 | R/W*1 | SD_BUF Illegal Write Access<br>[Setting conditions]<br>1. When data is written to SD_BUF0 while it is not in the data read/write command state<br>2. When data is written to SD_BUF0 while SD_BUF is full<br>3. When data is written to SD_BUF0 while an error occurs in the CRC status or CRC status length<br>4. When data is written to SD_BUF0 while the interface remains in a busy state for at least Ncycle after the CRC status<br>[Clearing condition]<br>When 0 is written to ERR4<br>Ncycle is set by bits 7 to 4 in SD_OPTION. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 3 | ERR3 | 0*2 | R/W*1 | Data Timeout (except response timeout)<br>[Setting conditions]<br>1. After R1b response, the interface remains in a busy state (SDDAT0 = 0) for at least Ncycle.<br>2. After CRC status, the interface remains in a busy state (SDDAT0 = 0) for at least Ncycle.<br>3. After write data, the CRC status is not received even after Ncycle has elapsed.<br>4. After read command, read data is not received even after Ncycle has elapsed.<br>5. After CMD12 has been issued within a command sequence, the interface remains in a busy state (SDDAT0 = 0) for at least Ncycle.<br>6. After the reception of read data, read data for the next block are not received even after Ncycle has elapsed.<br>7. After release of the read wait state, read data for the next block are not received even after Ncycle has elapsed.<br>[Clearing condition]<br>When 0 is written to ERR3<br>Ncycle is set by bits 7 to 4 in SD_OPTION.<br>The command sequence is halted by the data timeout. |
| 2 | ERR2 | 0*2 | R/W*1 | END Error<br>[Setting conditions]<br>1. When an error occurs in the response length (and the end bit has not been detected)<br>2. When an error occurs in the read data length (and the end bit has not been detected among the valid bits)<br>3. When an error occurs in the CRC status length (and the end bit has not been detected)<br>4. An error in the length of a response to a command issued within a command sequence*5 (i.e. the end bit has not been detected)<br>[Clearing condition]<br>When 0 is written to ERR2<br>The command sequence is halted by the End error.*4 |
| 1 | ERR1 | 0*2 | R/W*1 | CRC Error<br>[Setting conditions]<br>1. When an error occurs in the CRC status (i.e. the received CRC status was not ’010’)<br>2. When a CRC error occurs in the read data<br>3. When a CRC error occurs in the response<br>4. A CRC error in the response to a command issued within a command sequence*5<br>[Clearing condition]<br>When 0 is written to ERR1<br>The command sequence is halted by the CRC error.*4 |
| 0 | ERR0 | 0*2 | R/W*1 | CMD Error<br>[Setting conditions]<br>1. The command index of the transmitted command differing from the command index of the received response<br>2. The command index of a command issued within a command sequence*5 differing from the command index of the received response<br>[Clearing condition]<br>When 0 is written to ERR0<br>The command sequence is halted by the CMD error.*4 |

Note 1. It is effective only if 0 is written.

Note 2. The initial value is applied at a reset and when the SDRST bit in SOFT_RST is 0.

Note 3. When the WMODE bit in HOST_MODE is 0, the single byte from the fraction of a full 16-bit unit is regarded as excess data due to an odd value for the number of bytes setting in SD_SIZE. When the WMODE bit in HOST_MODE is 1, the single byte or three bytes from the fraction of a full 64-bit unit are regarded as excess data due to an odd value for the number of bytes

setting in SD_SIZE, or the two bytes from the fraction of a full 64-bit unit are regarded as excess data due if the value for the number of bytes setting in SD_SIZE is even but is not on a four-byte boundary.

Note 4. After the C52PUB bit in SDIO_MODE has been set to 1, if a communications error or timeout for response occurs in response to the CMD52 that is issued, since the command sequence has not been completed, complete the sequence with error processing as in usage examples in **Figure 40.15** under **Section 40.4.8, IO_RW_EXTENDED (CMD53/Multiple Block Read)** or in **Figure 40.18** under **Section 40.4.9, IO_RW_EXTENDED (CMD53/Multiple Block Write)**.

Note 5. “Command issued within a command sequence” refers to CMD12 when automatic issuing is enabled for multiple block transfer by the setting in SD_CMD, CMD12 when the STP bit in SD_STOP is set to 1, or CMD52 when the C52PUB or IOABT bit in SDIO_MODE is set to 1.

### 40.2.8 SD_INFO1 Interrupt Mask Register (SD_INFO1_MASK)

The SD_INFO1 interrupt mask register (SD_INFO1_MASK) is used to enable or disable the SD_INFO1 interrupt.

When 0 is set in SD_INFO1_MASK while the corresponding flag in SD_INFO1 is set, an interrupt occurs.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 63 to 17 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. Operation is not guaranteed if a value other than 0 is written to these bits. |
| 16 | IMASK16 | 1 | R/W | HPIRES interrupt masked |
| 15 to 10 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. Operation is not guaranteed if a value other than 0 is written to these bits. |
| 9 | IMASK9 | 1 | R/W | INFO9 interrupt masked |
| 8 | IMASK8 | 1 | R/W | INFO8 interrupt masked |
| 7 to 5 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. Operation is not guaranteed if a value other than 0 is written to these bits. |
| 4 | IMASK4 | 1 | R/W | INFO4 interrupt masked |
| 3 | IMASK3 | 1 | R/W | INFO3 interrupt masked |
| 2 | IMASK2 | 1 | R/W | INFO2 interrupt masked |
| 1 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. Operation is not guaranteed if a value other than 0 is written to this bit. |
| 0 | IMASK0 | 1 | R/W | INFO0 interrupt masked |

### 40.2.9 SD_INFO2 Interrupt Mask Register (SD_INFO2_MASK)

The SD_INFO2 interrupt mask register (SD_INFO2_MASK) is used to enable or disable the SD_INFO2 interrupt.

When 0 is set in SD_INFO2_MASK while the corresponding flag in SD_INFO2 is set, an interrupt occurs.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 63 to 16 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. Operation is not guaranteed if a value other than 0 is written to these bits. |
| 15 | IMASK | 1 | R/W | ILA interrupt masked |
| 14 to 12 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. Operation is not guaranteed if a value other than 0 is written to these bits. |
| 11 | — | 1 | R/W | Reserved<br>The write value should always be 1. |
| 10 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. Operation is not guaranteed if a value other than 0 is written to this bit. |
| 9 | BMASK1 | 1 | R/W | BWE interrupt masked |
| 8 | BMASK0 | 1 | R/W | BRE interrupt masked |
| 7 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. Operation is not guaranteed if a value other than 0 is written to this bit. |
| 6 | EMASK6 | 1 | R/W | ERR6 interrupt masked |
| 5 | EMASK5 | 1 | R/W | ERR5 interrupt masked |
| 4 | EMASK4 | 1 | R/W | ERR4 interrupt masked |
| 3 | EMASK3 | 1 | R/W | ERR3 interrupt masked |
| 2 | EMASK2 | 1 | R/W | ERR2 interrupt masked |
| 1 | EMASK1 | 1 | R/W | ERR1 interrupt masked |
| 0 | EMASK0 | 1 | R/W | ERR0 interrupt masked |

### 40.2.10 SD Clock Control Register (SD_CLK_CTRL)

The SD clock control register (SD_CLK_CTRL) is used to control the SD clock (SDCLK) output and to set the frequency. Set SCLKEN to 1 before writing to SD_CMD to issue a command. Do not write to SD_CLK_CTRL while the SCLKDIVEN bit in SD_INFO2 is set to 0.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 63 to 17 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. Operation is not guaranteed if a value other than 0 is written to these bits. |
| 16 | — | 0 | R/W | Reserved<br>The write value should always be 0. |
| 15 to 11 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. Operation is not guaranteed if a value other than 0 is written to these bits. |
| 10 | — | 0*1 | R/W | Reserved<br>The write value should always be 0. |
| 9 | SDCLKOFFEN | 0 | R/W | SD Clock (SDCLK) Output Automatic Control Enable<br>0: Automatic control for SD clock (SDCLK) output is disabled.<br>1: Automatic control for SD clock (SDCLK) output is enabled.<br>This function of automatic control for SD clock (SDCLK) output causes SDCLK output only within a command sequence.<br>The timing with which SDCLK output starts and stops is as follows.<br>SDCLK output starts after writing to SD_CMD.<br>SDCLK output stops when 8 cycles of SDCLK have elapsed after the end of the command sequence.<br>In addition, SDCLK is fixed to 0 while SCLKEN of SD_CLK_CTRL is 0, regardless of the value of this bit. |
| 8 | SCLKEN | 0*1 | R/W*2 | SD Clock (SDCLK) Output Control Enable<br>0: SD clock (SDCLK) output is disabled. The SDCLK signal is fixed 0.<br>1: SD clock (SDCLK) output is enabled. |
| 7 | DIV7 | 0 | R/W*2 | SD Clock (SDCLK)<br>1000 0000b: (SD0ϕ/4)/512<br>0100 0000b: (SD0ϕ/4)/256<br>0010 0000b: (SD0ϕ/4)/128<br>0001 0000b: (SD0ϕ/4)/64<br>0000 1000b: (SD0ϕ/4)/32<br>0000 0100b: (SD0ϕ/4)/16<br>0000 0010b: (SD0ϕ/4)/8<br>0000 0001b: (SD0ϕ/4)/4<br>0000 0000b: (SD0ϕ/4)/2<br>1111 1111b: SD0ϕ/4<br>Other settings are prohibited. |
| 6 | DIV6 | 0 | R/W*2 | |
| 5 | DIV5 | 1 | R/W*2 | |
| 4 | DIV4 | 0 | R/W*2 | |
| 3 | DIV3 | 0 | R/W*2 | |
| 2 | DIV2 | 0 | R/W*2 | |
| 1 | DIV1 | 0 | R/W*2 | |
| 0 | DIV0 | 0 | R/W*2 | |

Note 1. This initial value is applied at a reset and when the SDRST bit in SOFT_RST is 0.

Note 2. Writing to SD_CLK_CTRL is impossible when the CBSY bit in SD_INFO2 is 1.

**Notes on when setting the SD clock (SDCLK) to P1ϕ (DIV[7:0] = 1111 1111b)**

When changing the setting of bits DIV[7:0] to 1111 1111b, or from 1111 1111b to other setting, perform the following processing before writing to SD_CMD.

(1) Set the SCLKEN bit to 0 by writing to SD_CLK_CTRL. (Do not change the setting of bits other than SCLKEN at this time.)

(2) Change the setting of bits DIV[7:0] by writing to SD_CLK_CTRL. (Do not change the setting of bits other than DIV[7:0] at this time. The SCLKEN bit should retain the value 0.)

(3) Set the SCLKEN bit to 1 by writing to SD_CLK_CTRL. (Do not change the setting of bits other than SCLKEN at this time.)

Also when changing the setting of bits DIV[7:0] to 1111 1111b after having set the SDRST bit in SOFT_RST to 0 and then changed it to 1, perform this processing before writing to SD_CMD.

### 40.2.11 Transfer Data Length Register (SD_SIZE)

The transfer data length register (SD_SIZE) is used to specify the transfer data size.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 63 to 12 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. Operation is not guaranteed if a value other than 0 is written to these bits. |
| 11, 10 | — | All 0 | R | Reserved |
| 9 to 0 | LEN9 to LEN0 | 1000000000 | R/W | Transfer Data Size*1<br>These bits specify a size between 1 and 512 bytes for single block transfer.<br>In cases of multiple block transfer with automatic issuing of CMD12 (CMD18 and CMD25), the only specifiable transfer data size is 512 bytes.<br>Furthermore, in cases of multiple block transfer without automatic issuing of CMD12, as well as 512 bytes, 32, 64, 128, and 256 bytes are specifiable. However, in the reading of 32, 64, 128, and 256 bytes for the transfer of multiple blocks, this is restricted to multiple block transfer by CMD53.<br>Additionally, if a command accompanies data transfer, do not set these bits to 0. Do not specify a data size larger than 512 bytes. |

Note 1. Do not change the values of these bits when the CBSY bit in SD_INFO2 is 1.

### 40.2.12 SD Card Access Control Option Register (SD_OPTION)

The SD card access control option register (SD_OPTION) is used to set the bus width and timeout counter.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 63 to 16 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. Operation is not guaranteed if a value other than 0 is written to these bits. |
| 15 | WIDTH | 0*1 | R/W | Bus width*2<br>WIDTH prohibited<br>WIDTH = 0: 4 bits width<br>WIDTH = 1: 1 bit width |
| 14 | — | 1 | R | Reserved<br>This bit is always read as 1. The write value should always be 1. Operation is not guaranteed if a value other than 1 is written to this bit. |
| 13 | — | 0*1 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. Operation is not guaranteed if a value other than 0 is written to this bit. |
| 12 to 10 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. Operation is not guaranteed if a value other than 0 is written to these bits. |
| 9 | EXTOP | 0*1 | R/W | Timeout Mode Select<br>0: Bits TOP27 to TOP24 specify the timeout count from SDCLK*$2^{13}$ to SDCLK*$2^{27}$.<br>1: Bits TOP27 to TOP24 specify the timeout count from SDCLK*$2^{14}$ to SDCLK*$2^{28}$. |
| 8 | TOUTMASK | 0*1 | R/W | Timeout Mask<br>0: Enables timeout.<br>1: Disables timeout. (The ERR6 and ERR3 bits in SD_INFO2 and the E6 to E0 bits in SD_ERR_STS2 are not set.)<br>If a timeout occurs while it is disabled, perform a software reset to terminate a command sequence. |
| 7 | TOP27 | 1*1 | R/W | Timeout Counter*2<br>0000: SDCLK*$2^{13}$<br>0001: SDCLK*$2^{14}$<br>:<br>1101: SDCLK*$2^{26}$<br>1110: SDCLK*$2^{27}$<br>1111: Setting prohibited |
| 6 | TOP26 | 1*1 | R/W | |
| 5 | TOP25 | 1*1 | R/W | |
| 4 | TOP24 | 0*1 | R/W | |
| 3 | CTOP24 | 1*1 | R/W | Card Detect Time Counter<br>0000: SDCLK*$2^{10}$<br>0001: SDCLK*$2^{11}$<br>:<br>1101: SDCLK*$2^{23}$<br>1110: SDCLK*$2^{24}$<br>1111: Setting prohibited |
| 2 | CTOP23 | 1*1 | R/W | |
| 1 | CTOP22 | 1*1 | R/W | |
| 0 | CTOP21 | 0*1 | R/W | |

Note 1. The initial value is applied at a reset and when the SDRST bit in SOFT_RST is 0.

Note 2. Do not change the values of these bits when the CBSY bit in SD_INFO2 is 1.

### 40.2.13 SD Error Status Register 1 (SD_ERR_STS1)

The SD error status register 1 (SD_ERR_STS1) indicates the CRC status, CRC error, End error, and CMD error.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 63 to 32 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0.<br>Operation is not guaranteed if a value other than 0 is written to these bits. |
| 31 to 15 | — | Unknown | R | Reserved<br>These bits are always read as unknown. The write value should always be 0. |
| 14 | E14 | 0*1 | R | These bits hold the CRC status. (normal: 010) |
| 13 | E13 | 1*1 | | |
| 12 | E12 | 0*1 | | |
| 11 | E11 | 0*1 | R | Set to 1 when an error occurs in the CRC status. |
| 10 | E10 | 0*1 | R | Set to 1 when a CRC error occurs in the read data. |
| 9 | E9 | 0*1 | R | Set to 1 when a CRC error occurs in the response to a command issued within a command sequence*2. In cases where CMD12 is issued by setting a command index in SD_CMD, this is indicated in E8. |
| 8 | E8 | 0*1 | R | Set to 1 when a CRC error occurs in a response (other than a response to a command issued within a command sequence*2). |
| 7, 6 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. Operation is not guaranteed if a value other than 0 is written to these bits. |
| 5 | E5 | 0*1 | R | Set to 1 when an error occurs in the CRC status length (and the end bit has not been detected). |
| 4 | E4 | 0*1 | R | Set to 1 when an error occurs in the read data length (and the end bit has not been detected among the valid bits). |
| 3 | E3 | 0*1 | R | Set to 1 when an error occurs in the response length to a command issued within a command sequence*2. In cases where CMD12 is issued by setting a command index in SD_CMD, this is indicated in E2. |
| 2 | E2 | 0*1 | R | Set to 1 when an error occurs in the response length (other than a response to a command issued within a command sequence*2). |
| 1 | E1 | 0*1 | R | Set to 1 when an error occurs in the command index of the response to a command issued within a command sequence*2. In cases where CMD12 is issued by setting a command index in SD_CMD, this is Indicated in E0. |
| 0 | E0 | 0*1 | R | Set to 1 when an error occurs in the command index of a response (other than a response to a command issued within a command sequence*2). |

Note 1. The initial value is applied at a reset and when the SDRST bit in SOFT_RST is 0.

Note 2. “Command issued within a command sequence” refers to CMD12 when automatic issuing is enabled for multiple block transfer by the setting in SD_CMD, CMD12 when the STP bit in SD_STOP is set to 1, or CMD52 when the C52PUB or IOABT bit in SDIO_MODE is set to 1.

### 40.2.14 SD Error Status Register 2 (SD_ERR_STS2)

The SD error status register 2 (SD_ERR_STS2) indicates the timeout state. Ncycle is set by bits 7 to 4 in SD_OPTION.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 63 to 7 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. Operation is not guaranteed if a value other than 0 is written to these bits. |
| 6 | E6 | 0*1 | R | Set to 1 when the interface remains in a busy state for at least Ncycle after the CRC status. |
| 5 | E5 | 0*1 | R | Set to 1 when the CRC status is not received even after Ncycle has elapsed after data writing. |
| 4 | E4 | 0*1 | R | Set to 1 when read data is not received even after Ncycle has elapsed after the command has been read.<br>Set to 1 when read data for the next block are not received even after Ncycle has elapsed after the reception of read data.<br>Set to 1 when read data for the next block are not received even after Ncycle has elapsed after release of the read wait state. |
| 3 | E3 | 0*1 | R | Set to 1 when the interface remains in a busy state for at least Ncycle after CMD12 has been issued within a command sequence. In cases where CMD12 is issued by setting a command index in SD_CMD, this is indicated in E2. |
| 2 | E2 | 0*1 | R | Set to 1 when the interface remains in a busy state for at least Ncycle after R1b response. |
| 1 | E1 | 0*1 | R | Set to 1 when the response to a command issued within a command sequence*2 is not received even after 640 cycles of SDCLK have elapsed. In cases where CMD12 is issued by setting a command index in SD_CMD, this is indicated in E0. |
| 0 | E0 | 0*1 | R | Set to 1 when the response (other than a response to a command issued within a command sequence*2) is not received even after 640 cycles of SDCLK have elapsed. |

Note 1. The initial value is applied at a reset and when the SDRST bit in SOFT_RST is 0.

Note 2. “Command issued within a command sequence” refers to CMD12 when automatic issuing is enabled for multiple block transfer by the setting in SD_CMD, CMD12 when the STP bit in SD_STOP is set to 1, or CMD52 when the C52PUB or IOABT bit in SDIO_MODE is set to 1.

### 40.2.15 SD Buffer Read/Write Register (SD_BUF0)

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 63 to 0 | BUF63 to BUF0 | Unknown | R/W | When writing to the SD card, the write data is written to this register.<br>When reading from the SD card, the read data is read from this register.<br>This register is internally connected to two 512-byte buffers (SD_BUF).<br>When both buffers are not empty at multiple block read, suspend data reception by stopping the SD clock. When either buffer becomes empty, restart data reception by starting supply of the SD clock. |

**Note:** When using the DMAC, the bus width should be fixed at 64 bits.

### 40.2.16 SDIO Mode Control Register (SDIO_MODE)

The SDIO mode control register (SDIO_MODE) controls the CMD52 issuance and the read wait state at multiple block transfer, and the reception of SDIO interrupt. C52PUB and IOABT should not be set to 1 simultaneously.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 63 to 10 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. Operation is not guaranteed if a value other than 0 is written to these bits. |
| 9 | C52PUB | 0 | R/W | SDIO None Abort<br>• When C52PUB is set to 1 in the CMD53 (multiple block) write sequence, CMD52 is automatically issued between blocks if SD_BUF becomes empty.C52PUB is automatically cleared to 0 after reception of the response to CMD52 is completed. Additionally, if C52PUB is set to 1 while the last block is being transferred, CMD52 is not issued. In this case, C52PUB is automatically cleared to 0 after the access end flag has been set to 1.<br>• When C52PUB and RWREQ are set to 1 in the CMD53 (multiple block) read sequence, the block transfer enters the read wait state between blocks and CMD52 is automatically issued. C52PUB is automatically cleared to 0 after reception of the response to CMD52 is completed. Additionally, if C52PUB is set to 1 while the last block is being transferred, CMD52 is not issued. In this case, C52PUB is automatically cleared to 0 after the access end flag has been set to 1.<br>• If C52PUB is set to 1 in the CMD53 (multiple block) read sequence, be sure to set RWREQ to 1 as well as C52PUB.<br>• Set SD_ARG before setting C52PUB to 1.<br>• Set C52PUB to 1 after the response end flag has been set. |
| 8 | IOABT | 0 | R/W | SDIO Abort<br>• When IOABT is set to 1 in the CMD53 (multiple block) sequence, the CMD53 sequence is halted and CMD52 is issued.<br>However, if a command sequence is halted because of a communications error or timeout, CMD52 is not issued.<br>Although continued buffer access is possible even after IOABT has been set to 1, the buffer access error bit (ERR5 or ERR4) in SD_INFO2 will be set accordingly. Set SD_ARG before setting IOABT to 1.<br>• When IOABT has been set to 1 during transfer for single block write, the access end flag is set when SD_BUF becomes empty, and CMD52 is not issued. If SD_BUF does contain data, the access end flag is set on completion of reception of the busy state without CMD52 having been issued.<br>• When IOABT has been set to 1 during transfer for single block read, the access end flag is set immediately after setting of IOABT and CMD52 is not issued.<br>• When IOABT is set to 1 during reception of the busy state after an R1b response, the access end flag is set on completion of reception of the busy state without CMD52 having been issued.<br>• When IOABT is set to 1 after a command sequence has been completed, CMD52 is not issued and the access end flag is not set.<br>• Set IOABT to 1 after the response end flag has been set. Set IOABT to 0 after the access end flag has been set. |
| 7 to 3 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. Operation is not guaranteed if a value other than 0 is written to these bits. |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 2 | RWREQ | 0 | R/W | Read Wait Request<br>When RWREQ is set to 1 in the CMD53 (multiple block) read sequence, the block transfer enters the read wait state between blocks.<br>[Read wait state releasing]<br>(1) The read wait state is released, when RWREQ is cleared to 0 in the read wait state.<br>(2) When IOABT is set to 1 in the read wait state, RWREQ is automatically cleared to 0 after CMD52 has been issued, and then the read wait state is released.<br>(3) When C52PUB and RWREQ are set to 1 simultaneously in the CMD53 (multiple block) read sequence, the read wait state is not automatically released. Therefore, after the CMD52 response is received, clear RWREQ. (Be sure to set RWREQ and C52PUB simultaneously.)<br>When RWREQ is set to 1 while the last block in the CMD53 (multiple block) read sequence is transferred, the read wait state is not entered and RWREQ is automatically cleared to 0 by setting access end.<br>Set RWREQ to 1 after the response end flag has been set. |
| 1 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. Operation is not guaranteed if a value other than 0 is written to this bit. |
| 0 | IOMOD | 0 | R/W | SDIO Mode*1<br>0: Disables the SD host interfaces to receive SDIO interrupt from the SDIO card<br>1: Enables the SD host interfaces to receive SDIO interrupt from the SDIO card |

Note 1. Do not change the value of this bit when the CBSY bit in SD_INFO2 is set to 1.

### 40.2.17 SDIO Interrupt Flag Register (SDIO_INFO1)

The SDIO interrupt flag register (SDIO_INFO1) indicates the status regarding to the SDIO card access. To clear a flag, write 0 to the bit to be cleared and 1 to the other bits.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 63 to 16 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. Operation is not guaranteed if a value other than 0 is written to these bits. |
| 15 | EXWT | 0*2 | R/W*1 | [Setting condition]<br>While the last block in the CMD53 (multiple block) read sequence is transferred, RWREQ in SDIO_MODE is set to 1.<br>[Clearing condition]<br>When 0 is written to EXWT |
| 14 | EXPUB52 | 0*2 | R/W*1 | [Setting conditions]<br>1. While the last block in the CMD53 (multiple block) sequence is transferred, C52PUB in SDIO_MODE is set to 1.<br>2. While C52PUB is set to 1 in the CMD53 (multiple block) write sequence, the last block is transferred.<br>[Clearing condition]<br>When 0 is written to EXPUB52 |
| 13 to 3 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. Operation is not guaranteed if a value other than 0 is written to these bits. |
| 2, 1 | — | All 0*2 | R/W | Reserved<br>The write value should always be 1. The value may change during operation. |
| 0 | IOIRQ | 0*2 | R/W*1 | [Setting condition]<br>When SDIO interrupt from an SDIO card is received while IOMOD in SDIO_MODE is set to 1.<br>[Clearing condition]<br>When 0 is written to IOIRQ*3 |

Note 1. It is effective only if 0 is written.

Note 2. The initial value is applied at a reset and when the SDRST bit in SOFT_RST is 0.

Note 3. Before clearing this bit, access the SDIO card to negate the SDIO interrupt signal from the SDIO card. If the interrupt signal is not negated, this bit may be set again.

### 40.2.18 SDIO_INFO1 Interrupt Mask Register (SDIO_INFO1_MASK)

The SDIO_INFO1 interrupt mask register (SDIO_INFO1_MASK) enables or disables the SD_INFO1 interrupt. When 0 is set in SDIO_INFO1_MASK while the corresponding flag in SD_INFO1 is set, an interrupt occurs.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 63 to 16 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. Operation is not guaranteed if a value other than 0 is written to these bits. |
| 15 | MEXWT | 1 | R/W | EXWT interrupt masked |
| 14 | MEXPUB52 | 1 | R/W | EXPUB52 interrupt masked |
| 13 to 3 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. Operation is not guaranteed if a value other than 0 is written to these bits. |
| 2 | — | 1 | R/W | Reserved<br>The write value should always be 1. |
| 1 | — | 1 | R/W | Reserved<br>The write value should always be 1. |
| 0 | IOMSK | 1 | R/W | IOIRQ interrupt masked |

### 40.2.19 DMA Mode Enable Register (CC_EXT_MODE)

The DMA mode enable register (CC_EXT_MODE) enables the DMA transfer.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 63 to 13 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. Operation is not guaranteed if a value other than 0 is written to these bits. |
| 12 | — | 1 | R | Reserved<br>This bit is always read as 1. The write value should always be 1. Operation is not guaranteed if a value other than 1 is written to this bit. |
| 11, 10 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. Operation is not guaranteed if a value other than 0 is written to these bits. |
| 9, 8 | — | All 0 | R/W | Reserved<br>The write value should always be 0. |
| 7, 6 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. Operation is not guaranteed if a value other than 0 is written to these bits. |
| 5 | — | 0 | R/W | Reserved<br>The write value should always be 0. |
| 4 | — | 1 | R | Reserved<br>This bit is always read as 1. The write value should always be 1. Operation is not guaranteed if a value other than 1 is written to this bit. |
| 3, 2 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. Operation is not guaranteed if a value other than 0 is written to these bits. |
| 1 | DMASDRW | 0 | R/W | SD_BUF Read/Write DMA Transfer*1<br>0: The SD_BUF read/write DMA transfer is disabled.<br>1: The SD_BUF read/write DMA transfer is enabled. |
| 0 | — | 0 | R/W | Reserved<br>The write value should always be 0. |

Note 1. Do not change the value of this bit when the CBSY bit in SD_INFO2 is set to 1.

### 40.2.20 Software Reset Register (SOFT_RST)

The software reset register (SOFT_RST) sets a software reset. Also use this register to check that release from the reset state has been completed before attempting to use the SD host interfaces and before attempting access to the other registers.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 63 to 3 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. Operation is not guaranteed if a value other than 0 is written to these bits. |
| 2, 1 | — | 11 | R | Reserved<br>These bits are always read as 1. The write value should always be 1. Operation is not guaranteed if a value other than 1 is written to these bits. |
| 0 | SDRST | 1 | R/W | Software Reset of SD Interface Unit<br>0: Reset<br>1: Reset released |

### 40.2.21 Version Register (VERSION)

The version register (VERSION) indicates the version of the SD host interfaces.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 63 to 16 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. Operation is not guaranteed if a value other than 0 is written to these bits. |
| 15, 14 | UR7, UR6 | 11 | R | Reserved<br>These bits are always read as 1. The write value should always be 1. Operation is not guaranteed if a value other than 1 is written to these bits. |
| 13, 12 | UR5, UR4 | 00 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. Operation is not guaranteed if a value other than 0 is written to these bits. |
| 11 to 8 | UR3 to UR0 | H'C | R | Version of Renesas' IP |
| 7 to 0 | IP7 to IP0 | H'10 | R | Version of introductory IP |

### 40.2.22 Host Interface Mode Setting Register (HOST_MODE)

The host interface mode setting register (HOST_MODE) selects the width for access to the data bus.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 63 to 9 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. Operation is not guaranteed if a value other than 0 is written to these bits. |
| 8 | BUSWIDTH | 0 | R/W | Width for Access to SD_BUF*1*2<br>Read or write access to SD_BUF0 can be performed with the specified width for access.<br>0: 16-bit access<br>1: 32-bit access<br>This bit is enabled while the WMODE bit is set to 1. |
| 7 to 2 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. Operation is not guaranteed if a value other than 0 is written to these bits. |
| 1 | ENDIAN | 0 | R/W | SD_BUF0 data swap |
| 0 | WMODE | 0 | R/W | Width for Access to SD_BUF*1*2<br>Read or write access to SD_BUF0 can be performed with the specified width for access.<br>0: 64-bit access<br>1: 16-bit or 32-bit access |

Note 1. Do not change the value of this bit when the CBSY bit in SD_INFO2 is set to 1.

Note 2. When using the built-in DMAC of this module, fix the bus width to 64 bits.

### 40.2.23 SD Status Register (SD_STATUS)

The effective bit of the SD status register controls the output value on the SD0_RST# pin.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 63 to 2 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. Operation is not guaranteed if a value other than 0 is written to these bits. |
| 1 | SD_RST | 0 | R/W | Controls the output value on the SD0_RST# pin.<br>0: The output value on the SD0_RST# pin is 0.<br>1: The output value on the SD0_RST# pin is 1. |
| 0 | — | 0 | R | Reserved<br>This bit is always read as 0. The write value should always be 0. Operation is not guaranteed if a value other than 0 is written to this bit. |

### 40.2.24 DMAC Transfer Mode Register (DM_CM_DTRAN_MODE)

The DMAC Transfer Mode Register (DM_CM_DTRAN_MODE) sets the operation mode of the module built-in DMAC.

High 32 bits (bit 63 to 32) are read only 0.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 63 to 18 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. Operation is not guaranteed if a value other than 0 is written to these bits. |
| 17, 16 | CH_NUM | 00 | R/W | DMAC channel selector<br>00 : SD down stream<br>01 : SD up stream<br>Other settings are prohibited. |
| 15 to 6 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. Operation is not guaranteed if a value other than 0 is written to these bits. |
| 5, 4 | BUS_WIDTH [1:0] | 11 | R/W | Bus width selector<br>11: 64-bit<br>Other settings are prohibited. |
| 3 to 0 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. Operation is not guaranteed if a value other than 0 is written to these bits. |

### 40.2.25 DMAC Transfer Control Register (DM_CM_DTRAN_CTRL)

The DMAC Transfer Control Register (DM_CM_DTRAN_CTRL) controls the module built-in DMAC operation. High 32 bits (bit 63-32) are read only 0.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 63 to 9 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. Operation is not guaranteed if a value other than 0 is written to these bits. |
| 8 | — | 0 | R/W | Reserved<br>The write value should always be 0. |
| 7 to 1 | — | All 0 | R | Reserved |
| 0 | DM_START | 0 | R/W | DMAC Start<br>Writing 1 to this bit starts DMAC operation. This bit is automatically cleared when DMA transfer is started. |

### 40.2.26 DMAC Reset Register (DM_CM_RST)

High 32 bits (bit 63 to 32) are read only 0.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 63 to 32 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. Operation is not guaranteed if a value other than 0 is written to these bits. |
| 31 to 10 | — | All 1 | R | Reserved<br>These bits are always read as 1. The write value should always be 1. |
| 9 | DTRANRST1 | 1 | R/W | Soft resets of the module built-in DMAC channel 1<br>0: Reset<br>1: Reset released |
| 8 | DTRANRST0 | 1 | R/W | Soft resets of the module built-in DMAC channel 0<br>0: Reset<br>1: Reset released |
| 7 to 1 | — | All 1 | R | Reserved<br>These bits are always read as 1. The write value should always be 1. |
| 0 | SEQRST | 1 | R/W | Soft resets of the sequencer<br>0: Reset<br>1: Reset released |

NOTE

Make sure there is no communication to the SD device before applying a software reset using this register.

### 40.2.27 DMAC Interrupt Register 1 (DM_CM_INFO1)

The DMAC interrupt register 1 (DM_CM_INFO1) indicates the status of the module built-in DMAC and a sequencer. To clear a flag, write 0 to the bit to be cleared and 1 to the other bits.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 63 to 21 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. Operation is not guaranteed if a value other than 0 is written to these bits. |
| 20 | DTRANEND1 | 0*1 | R/W | Module built-in DMAC Channel 1 Transfer End<br>[Setting conditions]<br>1. When transfer of DMAC channel 1 is completed<br>2. When an error occurs on DMAC channel 1<br>[Clearing condition]<br>When 0 is written to DTRANEND1 |
| 19 to 17 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. Operation is not guaranteed if a value other than 0 is written to these bits. |
| 16 | DTRANEND0 | 0*2 | R/W | Module built-in DMAC Channel 0 Transfer End<br>[Setting conditions]<br>1. When transfer of DMAC channel 0 is completed<br>2. When an error occurs on DMAC channel 0<br>[Clearing condition]<br>When 0 is written to DTRANEND0 |
| 15 to 1 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. Operation is not guaranteed if a value other than 0 is written to these bits. |
| 0 | SEQEND | 0*3 | R/W | Sequencer Operation End<br>[Setting conditions]<br>1. When operation of a sequencer is completed<br>2. When a sequencer error occurs<br>[Clearing condition]<br>When 0 is written to SEQEND |

Note 1. The initial value is applied at a reset and when the DTRANRST1 bit in DM_CM_RST is 0.

Note 2. The initial value is applied at a reset and when the DTRANRST0 bit in DM_CM_RST is 0.

Note 3. The initial value is applied at a reset and when the SEQRST bit in DM_CM_RST is 0.

### 40.2.28 DM_CM_INFO1 Interrupt Mask Register (DM_CM_INFO1_MASK)

The DM_CM_INFO1 interrupt mask register (DM_CM_INFO1_MASK) enables or disables the DM_CM_INFO1 interrupt. When 0 is set in DM_CM_INFO1_MASK while the corresponding flag in DM_CM_INFO1 is set, an interrupt occurs.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 63 to 32 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. Operation is not guaranteed if a value other than 0 is written to these bits. |
| 31 to 21 | — | All 1 | R | Reserved<br>These bits are always read as 1. The write value should always be 1. Operation is not guaranteed if a value other than 1 is written to these bits. |
| 20 | DTRANEND1_MASK | 1 | R/W | DTRANEND1 interrupt masked |
| 19 to 17 | — | All 1 | R/W | Reserved<br>The write value should always be 1. |
| 16 | DTRANEND0_MASK | 1 | R/W | DTRANEND0 interrupt masked |
| 15 to 1 | — | All 1 | R | Reserved<br>These bits are always read as 1. The write value should always be 1. Operation is not guaranteed if a value other than 1 is written to these bits. |
| 0 | SEQEND_MASK | 1 | R/W | SEQEND interrupt masked |

### 40.2.29 DMAC Interrupt Register 2 (DM_CM_INFO2)

The DMAC interrupt register 2 (DM_CM_INFO2) indicates the status of the module built-in DMAC and a sequencer. To clear a flag, write 0 to the bit to be cleared and 1 to the other bits.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 63 to 20 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. Operation is not guaranteed if a value other than 0 is written to these bits. |
| 19, 18 | — | 00 | R | Reserved<br>The write value should always be 0. |
| 17 | DTRANERR1 | 0*1 | R/W | Module built-in DMAC Channel 1 Error<br>[Setting condition]<br>When an error occurs on DMAC channel 1<br>[Clearing condition]<br>When 0 is written to DTRANERR1 |
| 16 | DTRANERR0 | 0*2 | R/W | Module built-in DMAC Channel 0 Error<br>[Setting condition]<br>When an error occurs on the DMAC channel 0<br>[Clearing condition]<br>When 0 is written to DTRANERR0 |
| 15 to 1 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. Operation is not guaranteed if a value other than 0 is written to these bits. |
| 0 | SEQERR | 0*3 | R/W | Sequencer error<br>[Setting condition]<br>When a sequencer error occurs<br>[Clearing condition]<br>When 0 is written to SEQERR |

Note 1. The initial value is applied at a reset and when the DTRANRST1 bit in DM_CM_RST is 0.

Note 2. The initial value is applied at a reset and when the DTRANRST0 bit in DM_CM_RST is 0.

Note 3. The initial value is applied at a reset and when the SEQRST bit in DM_CM_RST is 0.

### 40.2.30 DM_CM_INFO2 Interrupt Mask Register (DM_CM_INFO2_MASK)

The DM_CM_INFO2 interrupt mask register (DM_CM_INFO2_MASK) enables or disables the DM_CM_INFO2 interrupt. When 0 is set in DM_CM_INFO2_MASK while the corresponding flag in DM_CM_INFO2 is set, an interrupt occurs.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 63 to 32 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. Operation is not guaranteed if a value other than 0 is written to these bits. |
| 31 to 20 | — | All 1 | R | Reserved<br>These bits are always read as 1. The write value should always be 1. Operation is not guaranteed if a value other than 1 is written to these bits. |
| 19, 18 | — | 11 | R | Reserved<br>The write value should always be 1. |
| 17 | DTRANERR1_MASK | 1 | R/W | DTRANERR1 interrupt masked |
| 16 | DTRANERR0_MASK | 1 | R/W | DTRANERR0 interrupt masked |
| 15 to 1 | — | All 1 | R | Reserved<br>These bits are always read as 1. The write value should always be 1. Operation is not guaranteed if a value other than 1 is written to these bits. |
| 0 | SEQERR_MASK | 1 | R/W | SEQERR interrupt masked |

### 40.2.31 DMAC Transfer Address Register (DM_DTRAN_ADDR)

The DMAC Transfer Address Register (DM_DTRAN_ADDR) sets the transfer destination and source address of the module built-in DMAC.

Higher-order 32 bits (bit 63 to 32) are read only 0.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 63 to 32 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. Operation is not guaranteed if a value other than 0 is written to these bits. |
| 31 to 3 | DADDR | All 0 | R/W | Destination address / Source address (8 byte unit)<br>Note that the value of DM_DTRAN_ADDR + transfer data length is less than or equal to $2^{32}$. |
| 2 to 0 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0. Operation is not guaranteed if a value other than 0 is written to these bits. |

# 40.3 Operation

## 40.3.1 SD Interface

### (1) SD Data Format

When data is read from the SD card, the procedure is as follows.

1. The SD host interface receives data from the SD card via the SDDAT signal. (SDDAT signal: see **Figure 40.2**, and **Figure 40.3**.)
2. The receive data is stored in SD_BUF of the SD host interfaces. (SD_BUF store data: see **Figure 40.4**)
3. The data stored in SD_BUF is read from SD_BUF0. (Reading from SD_BUF0: see **Table 40.4**)

When data is written to the SD card, the above procedure will be reversed.

When accessing SD_BUF0, caution should be taken for the transfer order in SDDAT and the store order in SD_BUF. In addition, data stored in SD_BUF0 can be replaced in bytes with the EXT_SWAP. (See **Figure 40.4**)

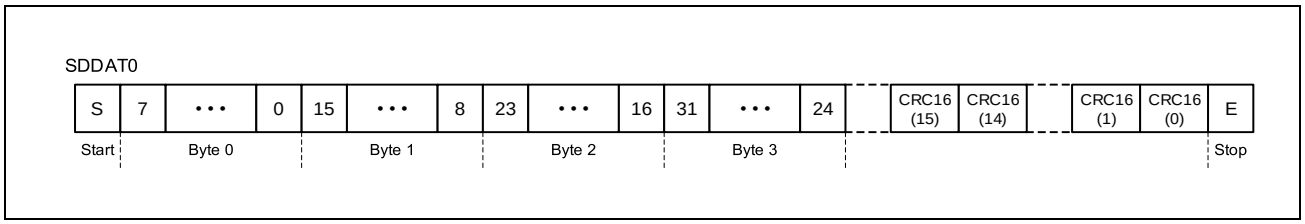



Figure 40.2 SDDAT in 1-Bit Width Mode

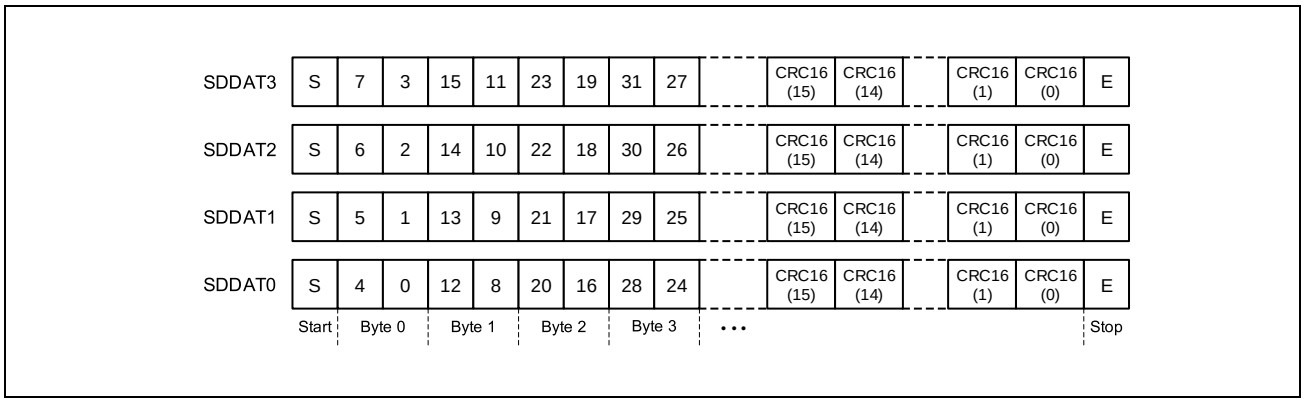



Figure 40.3 SDDAT in 4-Bit Width Mode

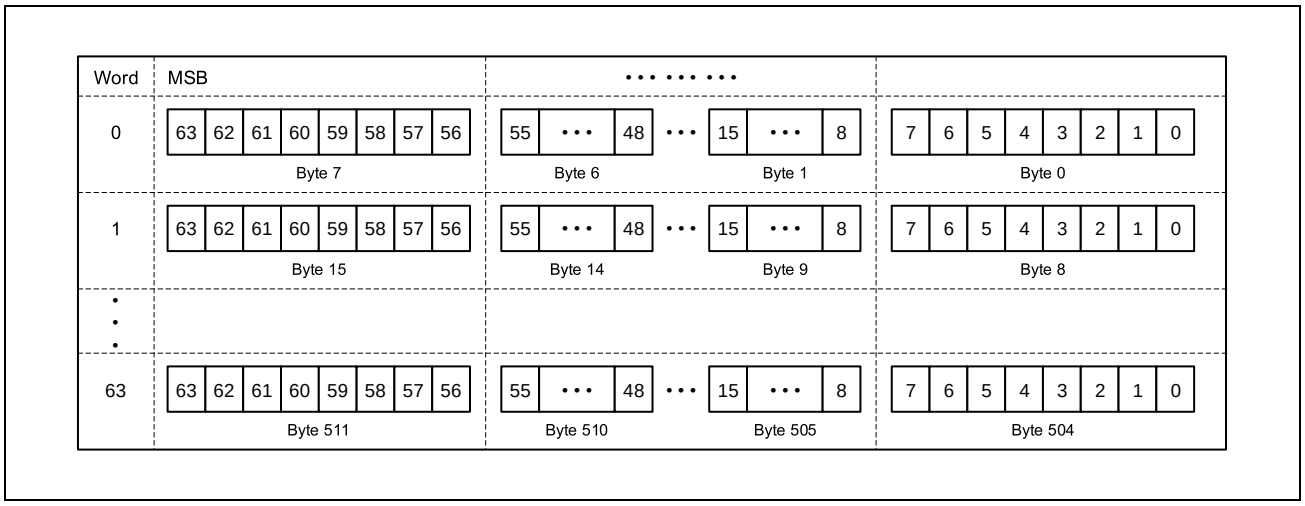



Figure 40.4 SD_BUF Store Data

Table 40.4 Reading from SD_BUF0

| WMODE*1 | BUSWIDTH*1 | ENDIAN*1 | Read Data*2 |
|---|---|---|---|
| 0 | 0 | 0 | H'0123456789ABCDEF |
| 0 | 0 | 1 | H'EFCDAB8967452301 |
| 1 | 1 | 0 | H'89ABCDEF (1st)<br>H'01234567 (2nd) |
| 1 | 1 | 1 | H'EFCDAB89 (1st)<br>H'67452301 (2nd) |
| 1 | 0 | 0 | H'CDEF (1st)<br>H'89AB (2nd)<br>H'4567 (3rd)<br>H'0123 (4th) |
| 1 | 0 | 1 | H'EFCD (1st)<br>H'AB89 (2nd)<br>H'6745 (3rd)<br>H'2301 (4th) |

Note 1. The name of a bit in HOST_MODE.

Note 2. When the data stored in SD_BUF is H'0123456789ABCDEF

### (2) Bus Signal Voltage Switch

Change the electric potential of the bus signal in the following procedure after checking that the SD card supports 1.8 V.

(1) Issuing CMD11
Perform command sequence processing of CMD11.

(2) Stopping the SD clock (a)
Set the SCLKEN bit in the SD_CLK_CTRL to 0 to stop*[1] the output of the SD clock. When the SDCLKOFFEN bit in the SD_CLK_CTRL register is 1, the SDCLKOFFEN bit is also set to 0.

**Note 1.** When the SDCLKOFFEN bit in the SD_CLK_CTRL register is 1, the SD clock has automatically been stopped.

(3) Checking the value of SDDAT
Check that the DAT0 bit in the SD_INFO2 register is 0.

(4) Changing the supply voltage of the host device
Change the voltage which is supplied through the power supply pin of the given channel (SD0_PVDD for channel 0) from 3.3 V to 1.8 V.

For details, see the following sections:

**Section 41.3.4, Port Function Control Register**,
**Section 41.3.7, Driving Ability Control Register**,
**Section 41.3.14, SD ch0 IO Voltage Mode Control Register**,

(5) Starting supply of the SD clock (b)
After the SD clock has been stopped ((a) above) and 5 ms or more has elapsed, set the SCLKEN bit in the SD_CLK_CTRL register to 1 and allow the output of the SD clock. The SDCLKOFFEN bit must be 0.

(6) Checking the value of SDDAT
After supplying the SD clock has been started ((b) above) and 1 ms or more has elapsed, check that the DAT0 bit in the SD_INFO2 register is 1. It is possible to set the SDCLKOFFEN bit in the SD_CLK_CTRL register to 1 and allow SD Clock (SDCLK) Output Automatic Control Enable.

## 40.3.2 Card Detect/Write Protect

### (1) Card Detect

The SD host interface has two types of card detect functions as described in the following.

- Card detect with ISDCD

  **Figure 40.5** shows the timing chart of card detect using ISDCD. ISDCD is connected to the card socket and pulled up on the host device. The resistance of the pull-up resistor is decided by the specification of the SD host device.

  [Card insertion]

  ISDCD is pulled down when a card is inserted. At this time, if ISDCD has been pulled down for the Mcycle period (set in SD_OPTION), INFO4 in SD_INFO1 is set to 1. (It is cleared to by writing 0.)

  [Card removal]

  ISDCD is pulled up when a card is removed. At this time, if ISDCD has been pulled up for the Mcycle period (set in SD_OPTION), INFO3 in SD_INFO1 is set to 1. (It is cleared to by writing 0.)

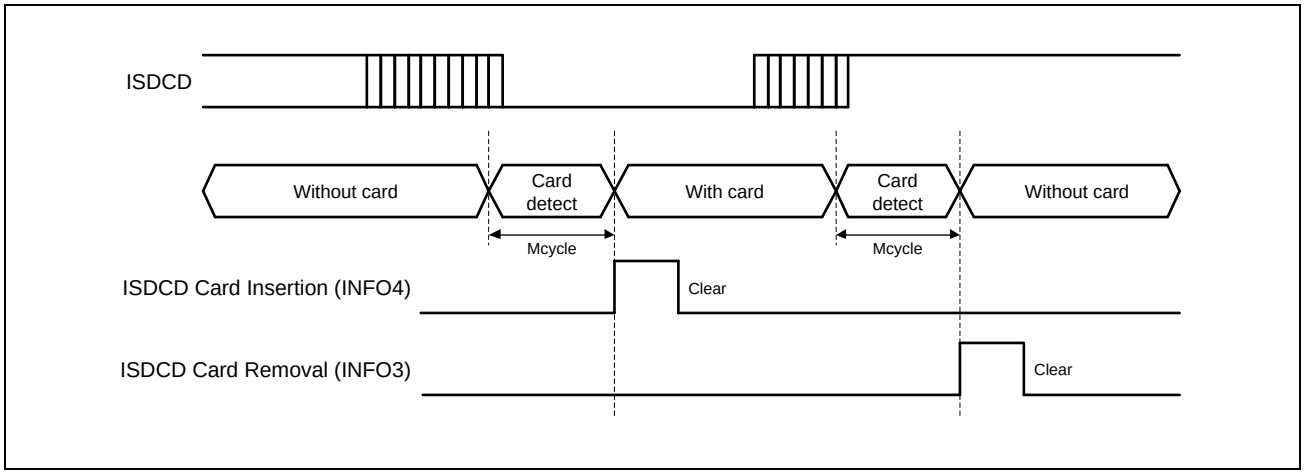



Figure 40.5 Example of Card Detect with ISDCD

- SD card detect with SDDAT3

  **Figure 40.6** shows the timing chart when the SD card is detected using SDDAT3. In addition, SDDAT3 is pulled down on the host device. The resistance of the pull-down resistor is decided by the specification of the SD host device.

  [Card insertion]

  When an SD card is inserted, SDDAT3 is pulled up. Accordingly, INFO9 in SD_INFO1 is set to 1. (It is cleared to by writing 0.)

  [Card removal]

  When an SD card is removed, SDDAT3 is pulled down. Accordingly, INFO8 in SD_INFO1 is set to 1. (It is cleared to by writing 0.)

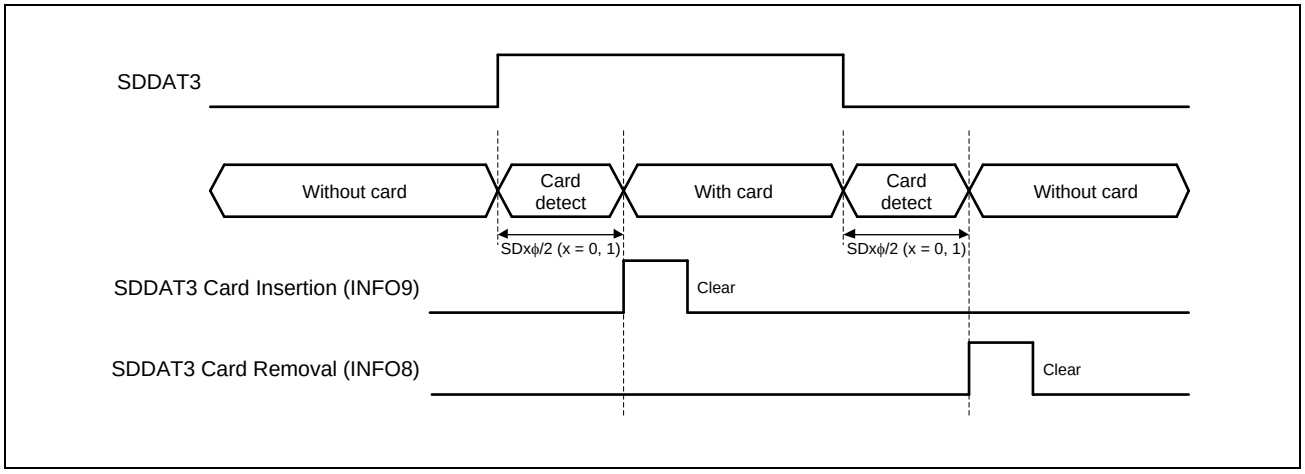



Figure 40.6 SD Card Detect with SDDAT3

### (2) Write Protect

The SD host interface has two types of write protect functions.

- Write protect with ISDWP

  ISDWP is connected to the card socket, and pulled up or pulled down by the card insertion. The selection of pulling up or pulling down and the resistance value is decided by the specifications of the SD host device. As the ISDWP state is reflected to INFO7 in SD_INFO1, the write protect is decided after the SD card is inserted.

- Write protect with command

  The card's internal write protection and the card lock/unlock operation are realized by the command.

## 40.3.3 Interrupt Request

### (1) Interrupt Request

The SD host interface has the interrupt requests shown in **Table 40.5** shows the relationship between the interrupt flag registers and the interrupt mask registers. When a bit in an interrupt mask register is set to 0, an interrupt occurs by setting the corresponding bit in the interrupt flag register to 1.

To clear a flag, write 0 to the bit to be cleared and 1 to the other bits.

Table 40.5 Interrupt Request

| | Interrupt Flag Register | | Interrupt Mask Register | |
|---|---|---|---|---|
| Interrupt Request | Register Name | Bit Name | Register Name | Bit Name |
| Card access interrupt*1 | SD_INFO1 | INFO2 | SD_INFO1_MASK | IMASK2 |
| | | INFO0 | | IMASK0 |
| | SD_INFO2 | ILA | SD_INFO2_MASK | IMASK |
| | | BWE | | BMASK1 |
| | | BRE | | BMASK0 |
| | | ERR6 | | EMASK6 |
| | | ERR5 | | EMASK5 |
| | | ERR4 | | EMASK4 |
| | | ERR3 | | EMASK3 |
| | | ERR2 | | EMASK2 |
| | | ERR1 | | EMASK1 |
| | | ERR0 | | EMASK0 |
| SDIO access interrupt*2 | SDIO_INFO1 | EXWT | SDIO_INFO1_MASK | MEXWT |
| | | EXPUB52 | | MEXPUB52 |
| | | IOIRQ | | IOMSK |
| Card detect interrupt*1 | SD_INFO1 | INFO9 | SD_INFO1_MASK | IMASK9 |
| | | INFO8 | | IMASK8 |
| | | INFO4 | | IMASK4 |
| | | INFO3 | | IMASK3 |
| DMAC interrupt*1 | DM_CM_INFO1 | DTRANEND1 | DM_CM_INFO1_MASK | DTRANEND1_MASK |
| | | DTRANEND0 | | DTRANEND0_MASK |
| | | SEQEND | | SEQEND_MASK |
| | DM_CM_INFO2 | DTRANERR1 | DM_CM_INFO2_MASK | DTARERR1_MASK |
| | | DTRANERR0 | | DTARERR0_MASK |
| | | SEQERR | | SEQERR_MASK |

Note 1. Interrupt signal “OXMNIRQ” is asserted.

Note 2. Interrupt signal both “OXMNIRQ” and “OXASIOIRQ” are asserted.

### 40.3.4 Communications Errors and Timeouts

- Communications Errors and Timeouts

**Table 40.6** and **Table 40.7** show the relationships between the SD card interrupt flag register and SD error status register for communications errors and timeouts, respectively. When a bit in the SD card interrupt flag register is set to 1, the corresponding bit in the SD error status register is set to 1. The values of the SD error status register are cleared by writing to SD_CMD or writing 0 to the SDRST bit in SOFT_RST.

Table 40.6 Communications Errors

| Communication Error | SD Interrupt Flag Register | | SD Error Status Register | | Description |
|---|---|---|---|---|---|
| | Register Name | Bit Name | Register Name | Bit Name | |
| END error | SD_INFO2 | ERR2 | SD_ERR_STS1 | E5 | When an error occurs in the CRC status length |
| | | | | E4 | When an error occurs in read data length |
| | | | | E3 | When an error occurs in the response length to a command issued within a command sequence |
| | | | | E2 | When an error occurs in the response length (other than a response to a command issued within a command sequence) |
| CRC error | | ERR1 | | E11 | When an error occurs in the CRC status |
| | | | | E10 | When a CRC error occurs in the read data |
| | | | | E9 | When a CRC error occurs in the response to a command issued within a command sequence |
| | | | | E8 | When a CRC error occurs in the response (other than a response to a command issued within a command sequence) |
| CMD error | | ERR0 | | E1 | The command index of the transmitted command differed from the command index of the received response (for a command issued within a command sequence) |
| | | | | E0 | The command index of the transmitted command differed from the command index of the received response (for a command issued other than within a command sequence) |

Table 40.7 Timeouts

| Timeout | SD Interrupt Flag Register | | SD Error Status Register | | Description |
|---|---|---|---|---|---|
| | Register Name | Bit Name | Register Name | Bit Name | |
| Response timeout | SD_INFO2 | ERR6 | SD_ERR_STS2 | E1 | When the response to a command issued within a command sequence is not received even after 640 cycles of SDCLK have elapsed |
| | | | | E0 | When the response (other than a response to a command issued within a command sequence) is not received even after 640 cycles of SDCLK have elapsed |
| Data timeout (other than response timeout) | | ERR3 | | E6 | When the interface remains in a busy state (SDDAT0 = 0) for at least Ncycle*1 after the CRC status |
| | | | | E5 | When the CRC status is not received t even after Ncycle*1 has elapsed after data writing |
| | | | | E4 | When read data is not received even after Ncycle*1 has elapsed after read command |
| | | | | | When read data for the next block are not received even after Ncycle*1 has elapsed after the reception of read data |
| | | | | | When read data for the next block are not received even after Ncycle*1 has elapsed after release of the read wait state |
| | | | | E3 | When the interface remains in a busy state (SDDAT0 = 0) for at least Ncycle*1 after CMD12 has been issued within a command sequence |
| | | | | E2 | When the interface remains in a busy state (SDDAT0 = 0) for at least Ncycle*1 after R1b response |

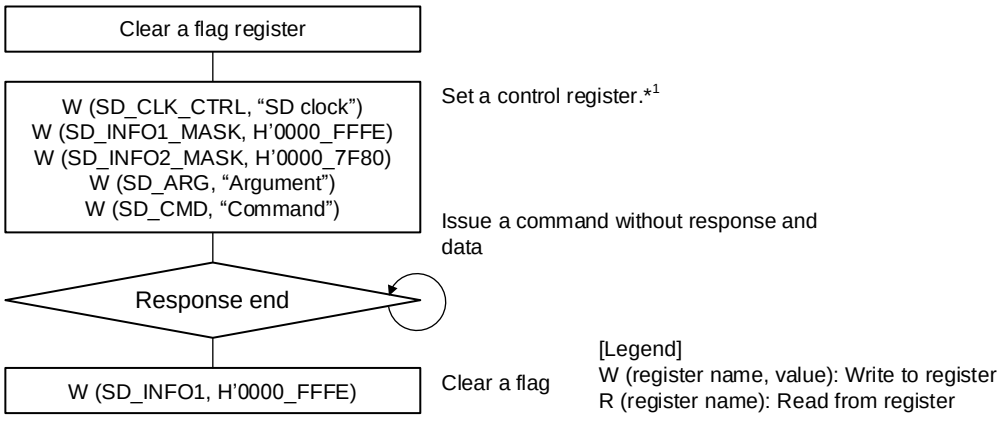
Note 1. Ncycle is set by bit 7 to bit 4 in SD_OPTION.

# 40.4 Usage Example

## 40.4.1 Command without Data Transfer

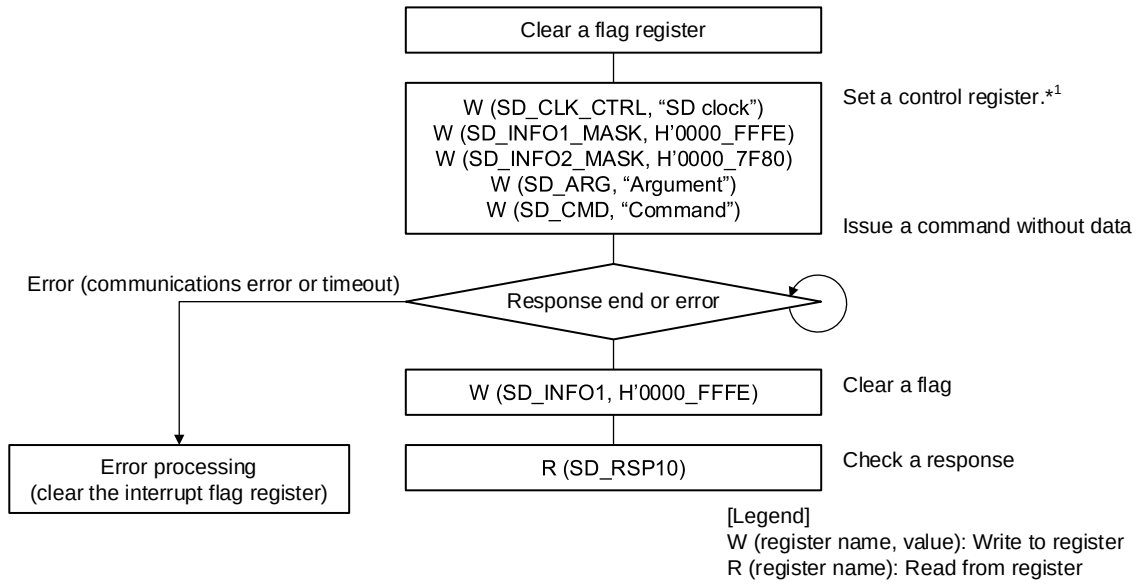
### (1) Flowchart

**Figure 40.7** and **Figure 40.8** show flowchart examples.




Note 1. Refer to **(5)** in **Section 40.5, Usage Notes**.

Figure 40.7 Flow Example of Command without Response and Data




Note 1. Refer to **(5)** in **Section 40.5, Usage Notes**.

Figure 40.8 Flow Example of Command without Data

### (2) Operation for Command without Data Transfer

The legend below is used for description of register read/write.

W (register name, value):Write to register

R (register name):Read from register

The operation is described as below.

#### (a) Command without response and data

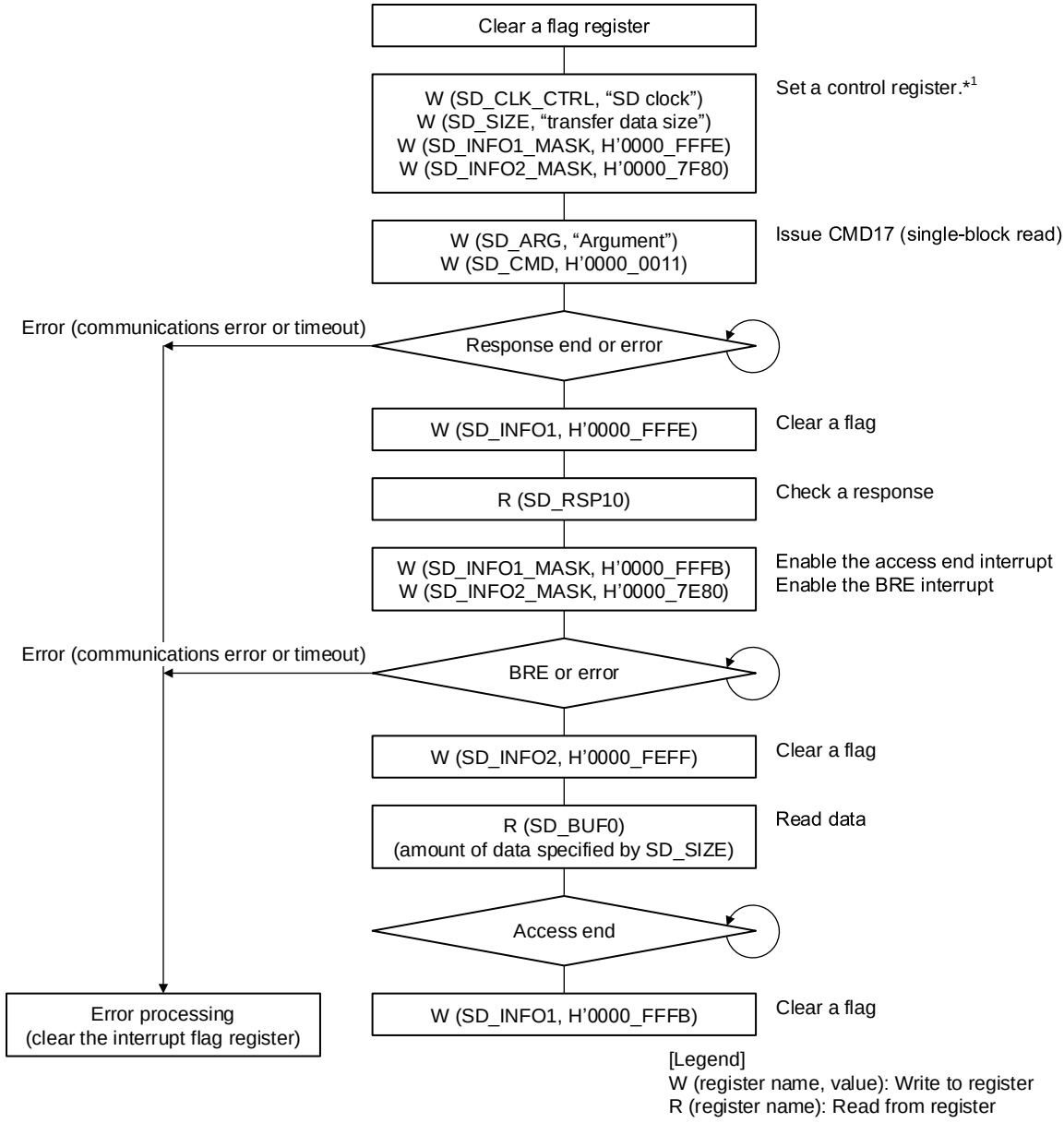
1. Flag register clear
   First, clear the bits in the flag register. (SD_INFO1 and SD_INFO2)
2. Control register set
   Set the SD clock (SDCLK), interrupt mask, and so on. (SD_CLK_CTRL, SD_INFO1_MASK, and SD_INFO2_MASK)
3. Command issue
   Set CMD Argument in SD_ARG and write to SD_CMD.
   Accordingly, CMD is issued, and the operation is started.
4. Flag clear
   When transmission of a command is completed, INFO0 (response end) in SD_INFO1 is set to 1 to generate an interrupt. Clear INFO0 to 0.

#### (b) Command without data

1. Flag register clear
   First, clear the bits in the flag register. (SD_INFO1 and SD_INFO2)
2. Control register set
   Set the SD clock (SDCLK), interrupt mask, and so on. (SD_CLK_CTRL, SD_INFO1_MASK, and SD_INFO2_MASK)
3. Command issue
   Set CMD Argument in SD_ARG and write to SD_CMD.
   Accordingly, CMD is issued, and the operation is started.
4. Flag clear
   When a response is received, INFO0 (response end) in SD_INFO1 is set to 1 to generate an interrupt. Clear INFO0 to 0.
5. Read a response from SD_RSP10.
   Furthermore, perform error processing (clear the interrupt flag register) if an error occurs (a communications error or timeout).

## 40.4.2 Single Block Read

### (1) Flowchart




**Note:** The error processing of SDR104 requires a re-tuning.

Note 1. Refer to **(5)** in **Section 40.5, Usage Notes**.

Figure 40.9 Single Block Read Flowchart Example

### (2) Operation for Single Block Read

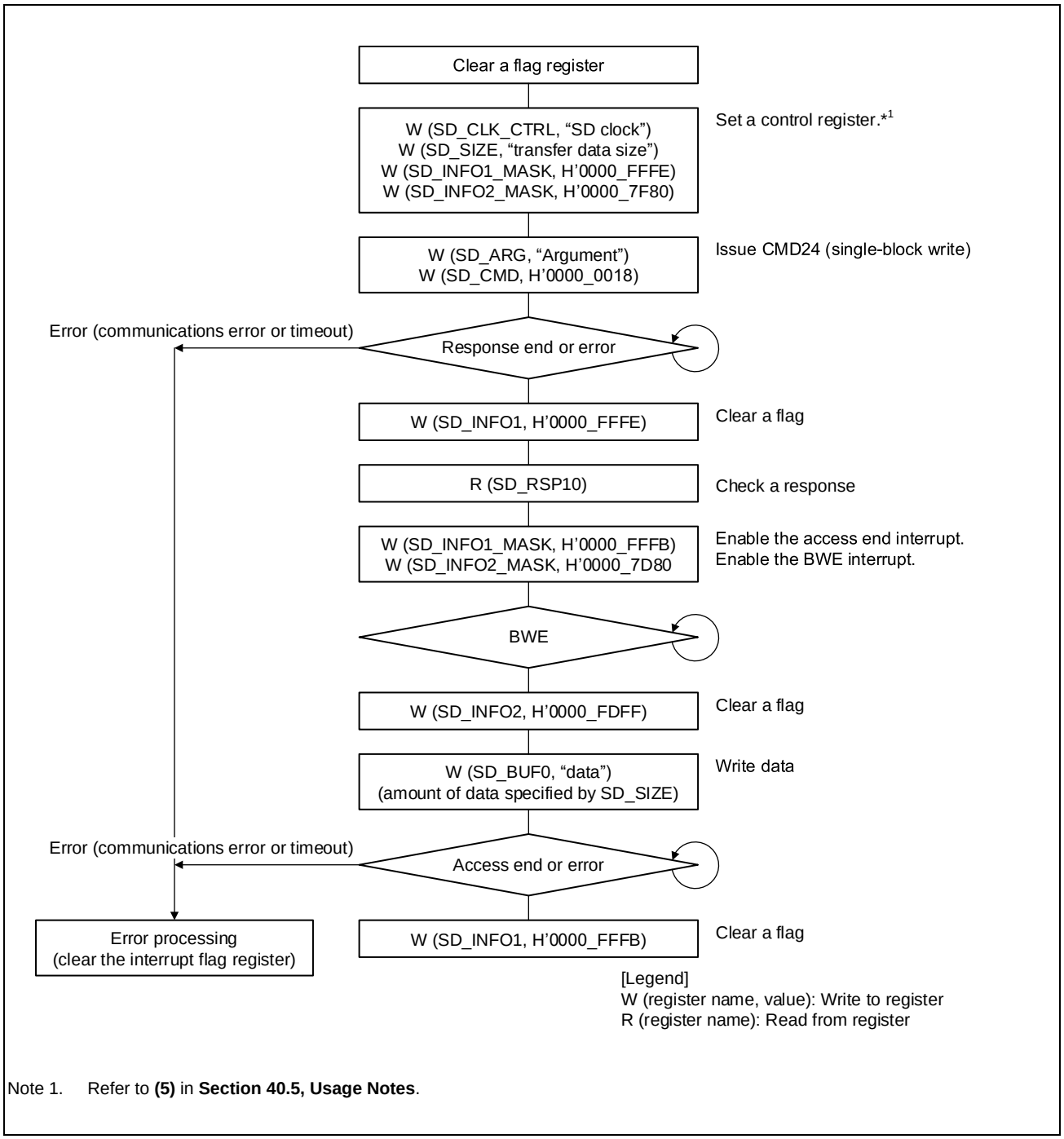
The operation of the single block read is described below.

1. Flag register clear
   First, clear the bits in the flag register. (SD_INFO1 and SD_INFO2)
2. Control register set
   Set the SD clock (SDCLK), transfer data size, interrupt mask, and so on. (SD_CLK_CTRL, SD_SIZE, SD_INFO1_MASK, and SD_INFO2_MASK)
3. Command issue (CMD17)
   Set CMD17 Argument in SD_ARG and write H'0000 0011 to SD_CMD.
   Accordingly, CMD17 is issued, and the single block read operation is started.
4. Response check
   On receiving the response, INFO0 (response end) in SD_INFO1 is set to 1 to generate an interrupt. Clear INFO0 to 0 and read the response from SD_RSP10.
   If the result of response decoding is an error, the command sequence can be halted by setting the STP bit in SD_STP or the IOABT bit in SDIO_MODE to 1. Furthermore, this causes CMD12 and CMD52 to not be issued.
   If the command sequence is halted, the INFO2 bit (access end) in SD_INFO1 is set to 1 to generate an interrupt.
5. Data receive from SD card and data read
   Write H'0000 FFFB to SD_INFO1_MASK to enable the access end interrupt. In addition, write H'0000 7E80 to SD_INFO2_MASK to enable the BRE interrupt. When the data receive from the SD card is completed, the BRE bit in SD_INFO2 is set to 1 to generate an interrupt. Clear the BRE bit to 0 and read the amount of data specified by SD_SIZE from SD_BUF0.
   However, a communications error or timeout may be generated if data are being received while reading of SD_BUF0 is in progress.
6. Operation complete
   When the data read from SD_BUF0 is completed, INFO2 (access end) in SD_INFO1 is set to 1 to generate an interrupt. Clear INFO2 to 0 to end the single block read operation.

Furthermore, perform error processing (clear the interrupt flag register) if an error occurs (a communications error or timeout).

### 40.4.3 Single Block Write

#### (1) Flowchart




Note 1. Refer to **(5)** in **Section 40.5, Usage Notes**.

Figure 40.10 Single Block Write Flowchart Example

### (2) Operation for Single Block Write

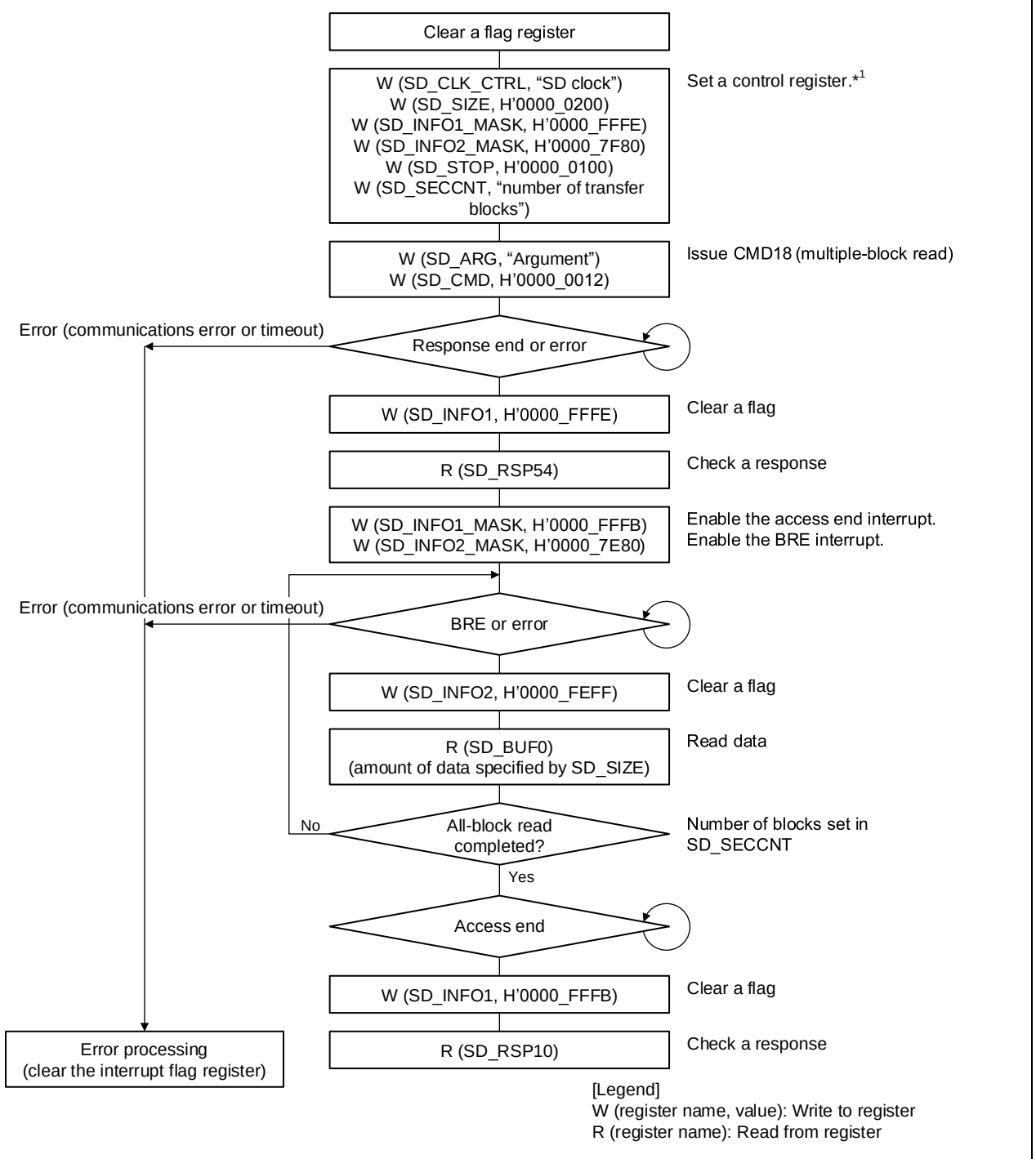
The operation of the single block write is described below.

1. Flag register clear
   First, clear the bits in the flag register. (SD_INFO1 and SD_INFO2)
2. Control register set
   Set the SD clock (SDCLK), transfer data size, interrupt mask, and so on. (SD_CLK_CTRL, SD_SIZE, SD_INFO1_MASK, and SD_INFO2_MASK)
3. Command issue (CMD24)
   Set CMD24 Argument in SD_ARG and write H'0000 0018 to SD_CMD.
   Accordingly, CMD24 is issued, and the single block write operation is started.
4. Response check
   On receiving the response, INFO0 (response end) in SD_INFO1 is set to 1 to generate an interrupt. Clear INFO0 to 0 and read the response from SD_RSP10.
   If the result of response decoding is an error, the command sequence can be halted by setting the STP bit in SD_STP or the IOABT bit in SDIO_MODE to 1. Furthermore, this causes CMD12 and CMD52 to not be issued.
   If the command sequence is halted, the INFO2 bit (access end) in SD_INFO1 is set to 1 to generate an interrupt.
5. Data write and data transmit to SD card
   Write H'0000 FFFB to SD_INFO1_MASK to enable the access end interrupt. In addition, write H'0000 7D80 to SD_INFO2_MASK to enable the BWE interrupt. When SD_BUF0 is ready for the data to be written, the BWE bit in SD_INFO2 is set to 1 to generate an interrupt. Clear the BWE bit to 0 and write the amount of data specified by SD_SIZE to SD_BUF0. When the data write to SD_BUF0 is completed, data is transmitted to the SD card. Then, the CRC status and busy state are received from the SD card.
   However, a communications error or timeout may be generated if data are being transmitted after writing to SD_BUF0.
6. Operation complete
   When the CRC status and busy state are received from the SD card, INFO2 (access end) in SD_INFO1 is set to 1 to generate an interrupt. Clear the INFO2 bit to 0 to end the single block write operation.

Furthermore, perform error processing (clear the interrupt flag register) if an error occurs (a communications error or timeout).

### 40.4.4 Multiple Block Read

#### (1) Flowchart




**Note:** The error processing of SDR104 requires a re-tuning.

Note 1. Refer to **(5)** in **Section 40.5, Usage Notes**.

Figure 40.11 Multiple Block Read Flowchart Example

### (2) Operation for Multiple Block Read

The operation of the multiple block read is described below.

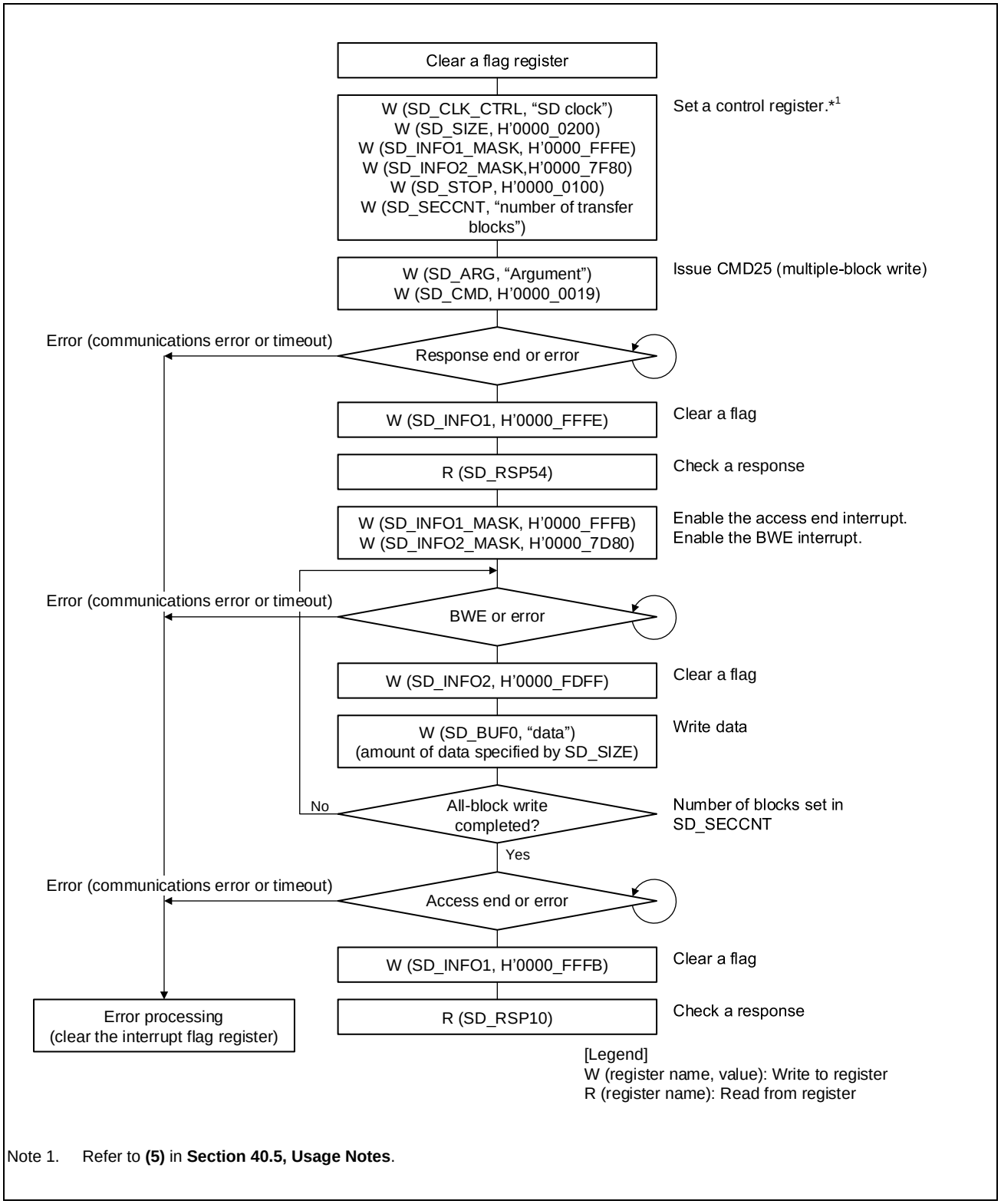
1. Flag register clear
   First, clear the bits in the flag register. (SD_INFO1 and SD_INFO2)
2. Control register set
   Set the SD clock (SDCLK), transfer data size, interrupt mask, and so on. (SD_CLK_CTRL, SD_SIZE, SD_INFO1_MASK, and SD_INFO2_MASK)
   Set the SEC bit in SD_STOP to 1, and set the number of transfer blocks in SD_SECCNT.
3. Command issue (CMD18)
   Set CMD18 Argument in SD_ARG and write H'0000 0012 to SD_CMD.
   Accordingly, CMD18 is issued, and the multiple block read operation is started.
4. Response check
   On receiving the response, INFO0 (response end) in SD_INFO1 is set to 1 to generate an interrupt. Clear INFO0 to 0 and read the response from SD_RSP54.
   If the result of response decoding is an error, the command sequence can be halted by setting the STP bit in SD_STP to 1. Setting the STP bit to 1 also causes CMD12 to be issued and the response received. If the command sequence is halted because the access end interrupt has been enabled, an interrupt will be generated by setting of the INFO2 bit (access end) in SD_INFO1 to 1 when reception of the response has been completed. Clear the INFO2 bit to 0 and read the response.
5. Data receive from SD card and data read
   Write H'0000 FFFB to SD_INFO1_MASK to enable the access end interrupt. In addition, write H'0000 7E80 to SD_INFO2_MASK to enable the BRE interrupt. When one-block data receive from the SD card is completed, the BRE bit in SD_INFO2 is set to 1 to generate an interrupt. Clear the BRE bit to 0 and read the amount of data specified by SD_SIZE from SD_BUF0. Doing this repeats transfer of the number of blocks set in SD_SECCNT. However, a communications error or timeout may be generated if data are being received while reading of SD_BUF0 is in progress. CMD12 is automatically issued to stop multi-block transfer with the number of blocks which is set to SD_SECCNT and the response is received. At this point, CMD12 Argument is automatically set to H'0000_0000.
6. Operation complete
   When all-block data read and the CMD12 response receive are completed, INFO2 (access end) in SD_INFO1 is set to 1 to generate an interrupt. Clear INFO2 to 0 to read the response. This is the end of multiple block read operation.

Furthermore, perform error processing (clear the interrupt flag register) if an error occurs (a communications error or timeout).

### 40.4.5 Multiple Block Write (when Using Internal Timer)

#### (1) Flowchart

**Figure 40.12** shows the flowchart when using an internal timer.




Note 1. Refer to **(5)** in **Section 40.5, Usage Notes**.

Figure 40.12 Multiple Block Write Flowchart Example (when Using Internal Timer)

### (2) Operation for Multiple Block Write

The operation of the multiple block write is described below.

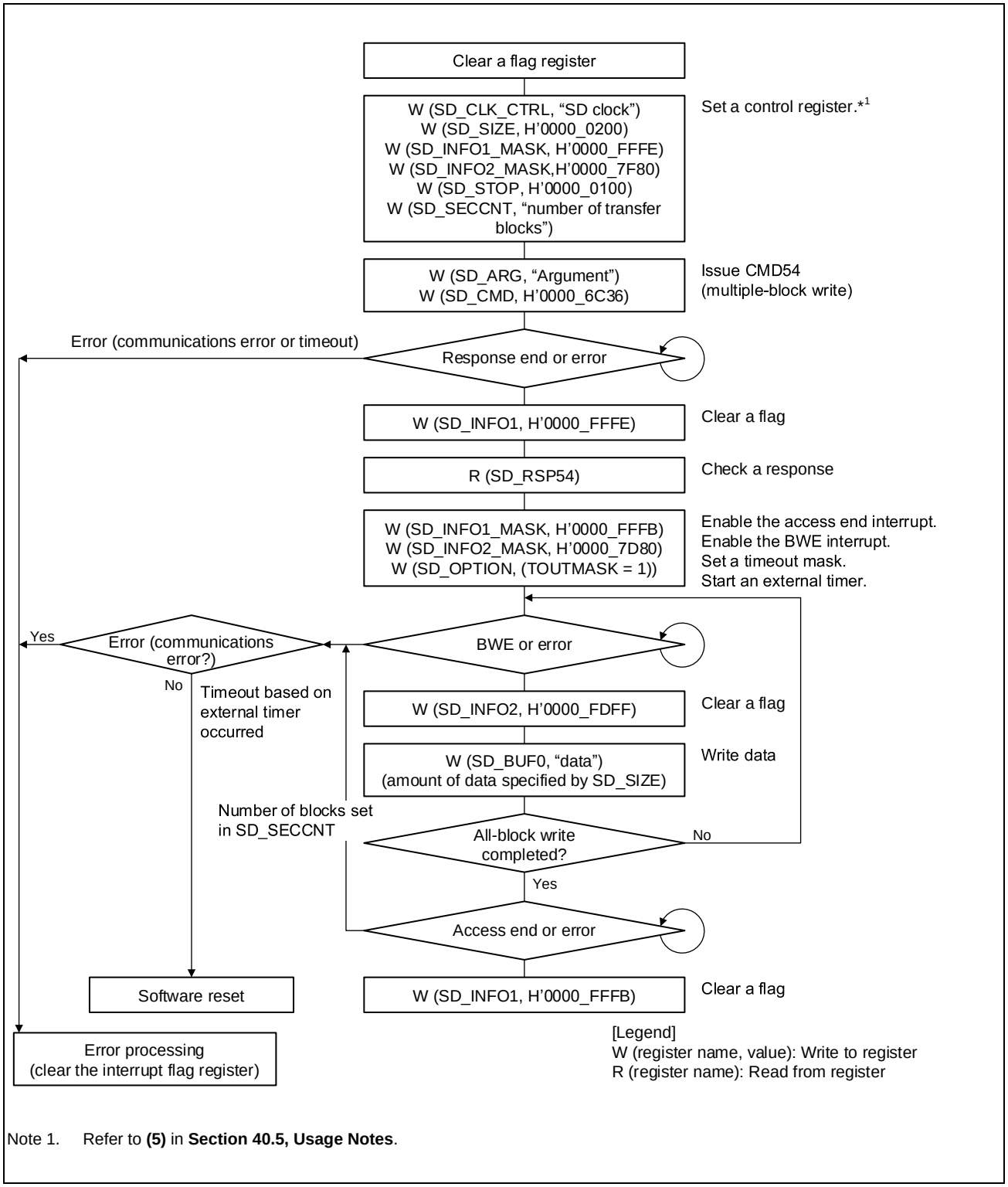
1. Flag register clear
   First, clear the bits in the flag register. (SD_INFO1 and SD_INFO2)
2. Control register set
   Set the SD clock (SDCLK), transfer data size, interrupt mask, and so on. (SD_CLK_CTRL, SD_SIZE, SD_INFO1_MASK, and SD_INFO2_MASK)
   Set the SEC bit in SD_STOP to 1, and set the number of transfer blocks in SD_SECCNT.
3. Command issue (CMD25)
   Set CMD25 Argument in SD_ARG and write H'0000 0019 to SD_CMD.
   Accordingly, CMD25 is issued, and the multiple block write operation is started.
4. Response check
   On receiving the response, INFO0 (response end) in SD_INFO1 is set to 1 to generate an interrupt. Clear INFO0 to 0 and read the response from SD_RSP54.
   If the result of response decoding is an error, the command sequence can be halted by setting the STP bit in SD_STP to 1. Setting the STP bit to 1 also causes CMD12 to be issued and the response received. If the command sequence is halted because the access end interrupt has been enabled, an interrupt will be generated by setting of the INFO2 bit (access end) bit in SD_INFO1 to 1 when reception of the response has been completed. Clear the INFO2 bit to 0 and read the response.
5. Data write and data transmit to SD card
   Write H'0000 FFFB to SD_INFO1_MASK to enable the access end interrupt. In addition, write H'0000 7D80 to SD_INFO2_MASK to enable the BWE interrupt. When SD_BUF0 is ready for the data to be written, the BWE bit in SD_INFO2 is set to 1 to generate an interrupt. Clear the BWE bit to 0 and write the amount of data specified by SD_SIZE to SD_BUF0. When the data write to SD_BUF0 is completed, data is transmitted to the SD card. Then, the CRC status and busy state are received from the SD card. Doing this repeats transfer of the number of blocks set in SD_SECCNT.
   However, a communications error or timeout may be generated if data are being transmitted while writing to SD_BUF0 is in progress. CMD12 is automatically issued to stop multi-block transfer with the number of blocks which is set to SD_SECCNT and the response is received. At this point, CMD12 Argument is automatically set to H'0000 0000.
6. Operation complete
   When all-block data transmit and the CRC status receive are completed, the INFO2 bit (access end) in SD_INFO1 is set to 1 to generate an interrupt. Clear the INFO2 bit to 0 to read the response. This is the end of multiple block write operation.

Furthermore, perform error processing (clear the interrupt flag register) if an error occurs (a communications error or timeout).

## 40.4.6 Multiple Block Write (when Using External Timer)

### (1) Flowchart

The flowchart when using an external timer instead of an internal timer of this module is shown below.




Note 1. Refer to **(5)** in **Section 40.5, Usage Notes**.

Figure 40.13 Multiple Block Write Flowchart Example (when Using External Timer)

### (2) Operation for Multiple Block Write

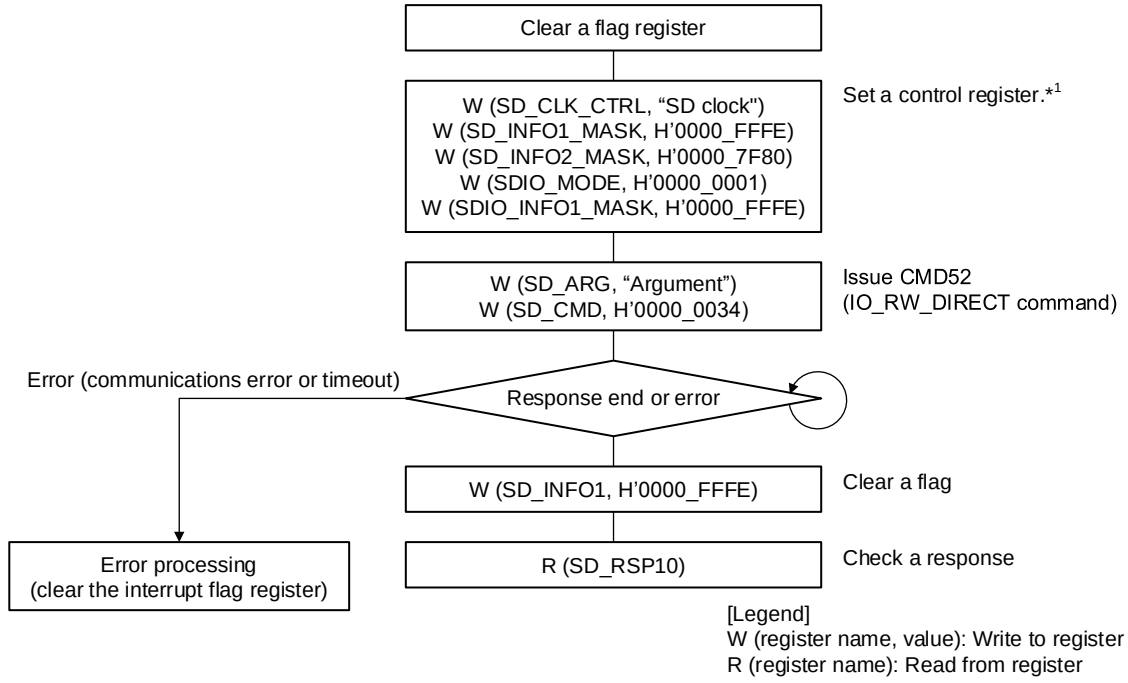
The operation of the multiple block write is described below.

1. Flag register clear
   First, clear the bits in the flag register. (SD_INFO1 and SD_INFO2)
2. Control register set
   Set the SD clock (SDCLK), transfer data size, interrupt mask, and so on. (SD_CLK_CTRL, SD_SIZE, SD_INFO1_MASK, and SD_INFO2_MASK)
   Set the SEC bit in SD_STOP to 1, and set the number of transfer blocks in SD_SECCNT.
3. Command issue (CMD54)
   Set CMD54 Argument in SD_ARG and write H'0000 6C36 to SD_CMD.
   Accordingly, CMD54 is issued, and the multiple block write operation is started.
4. Response check
   On receiving the response, INFO0 (response end) in SD_INFO1 is set to 1 to generate an interrupt. Clear INFO0 to 0 and read the response from SD_RSP54.
   If the result of response decoding is an error, the command sequence can be halted by setting the STP bit in SD_STP to 1. Setting the STP bit to 1 also causes CMD12 to be issued and the response received. If the command sequence is halted because the access end interrupt has been enabled, an interrupt will be generated by setting of the INFO2 bit (access end) bit in SD_INFO1 to 1 when reception of the response has been completed. Clear the INFO2 bit to 0 and read the response.
5. Data write and data transmit to SD card
   Write H'0000 FFFB to SD_INFO1_MASK to enable the access end interrupt. In addition, write H'0000 7D80 to SD_INFO2_MASK to enable the BWE interrupt. Set the TOUTMASK bit in SD_OPTION to disable timeout and start an external timer.
   When SD_BUF0 is ready for the data to be written, the BWE bit in the SD_INFO2 resister is set to 1 to generate an interrupt. Clear the BWE bit to 0 and write the amount of data specified by SD_SIZE to SD_BUF0. When the data write to SD_BUF0 is completed, data is transmitted to the SD card. Then, the CRC status and busy state are received from the SD card. Doing this repeats transfer of the number of blocks set in SD_SECCNT. However, a communications error may be generated if data are being transmitted while writing to SD_BUF0 is in progress.
6. Operation complete
   When all-block data transmit and the CRC status receive are completed, the INFO2 bit (access end) in SD_INFO1 is set to 1 to generate an interrupt. Clear the INFO2 bit to 0 to read the response. This is the end of multiple block write operation.

Furthermore, perform error processing (clear the interrupt flag register) if an error occurs at response reception (a communications error or timeout) or at data transmission. Perform a software reset if a timeout occurs at data transmission based on an external timer.

### 40.4.7 IO_RW_DIRECT Commend (CMD52)

#### (1) Flowchart



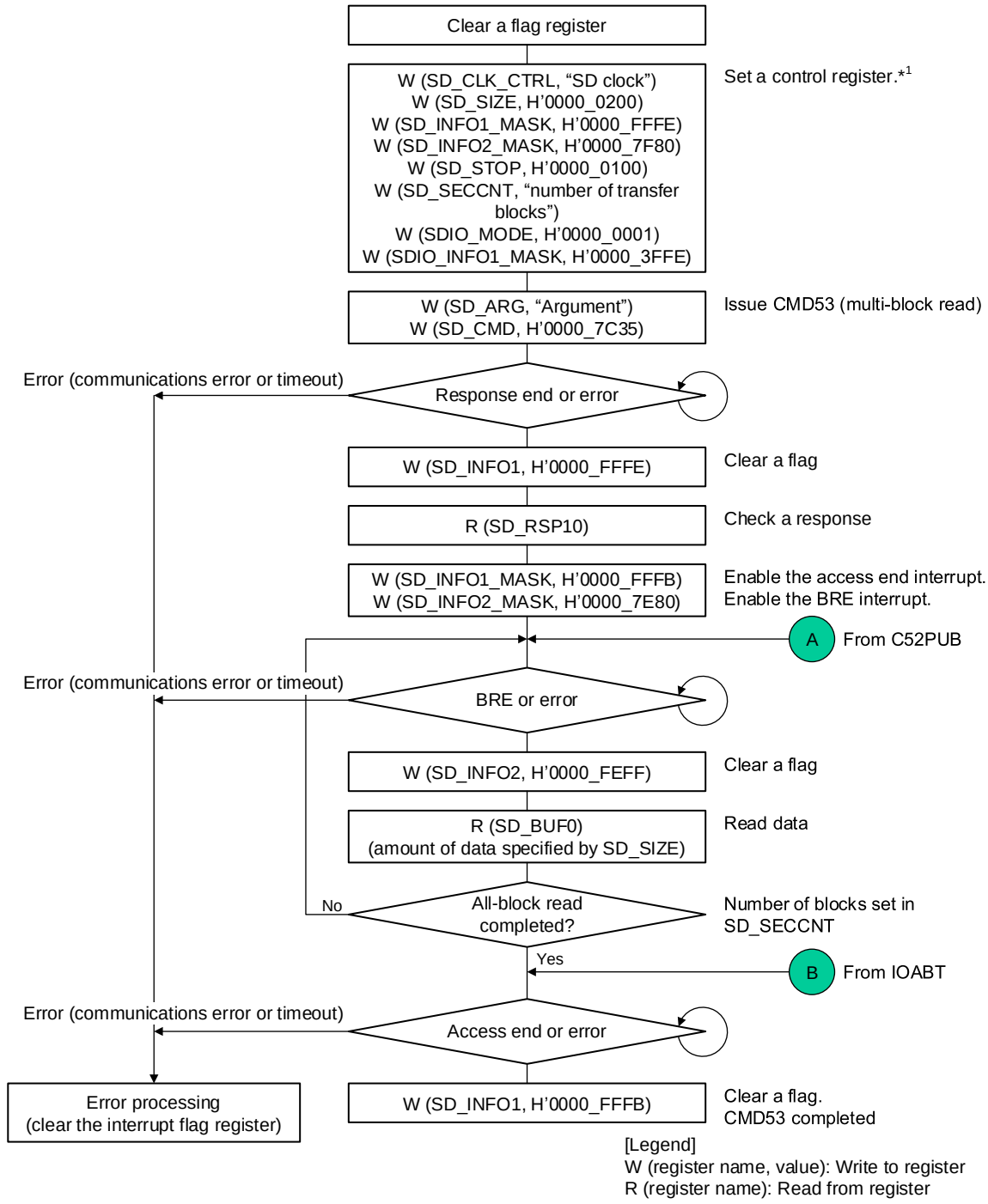

Note 1. Refer to **(5)** in **Section 40.5, Usage Notes**.

Figure 40.14 IO_RW_DIRECT Command (CMD52) Flowchart Example

### 40.4.8 IO_RW_EXTENDED (CMD53/Multiple Block Read)

#### (1) Flowchart

**Figure 40.15** shows a flowchart example for CMD53 (multiple block read).




Note 1. Refer to **(5)** in **Section 40.5, Usage Notes**.

Figure 40.15 CMD53 (Multiple Block Read) Flowchart Example

**Figure 40.16** shows a flowchart example when CMD52 (SDIO abort) is issued at CMD53 (multiple block read).

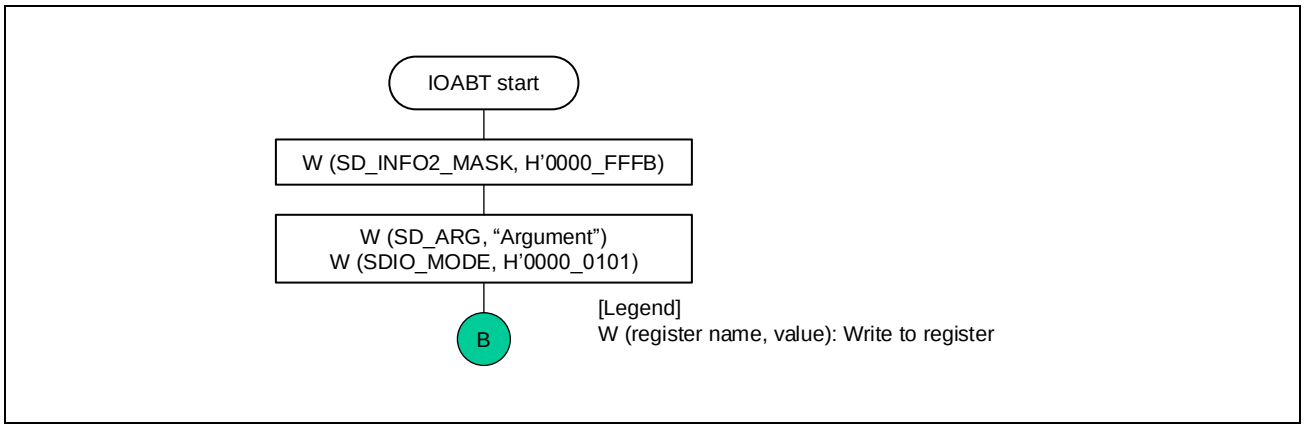



Figure 40.16 Flowchart Example when CMD52 (SDIO Abort) is Issued at CMD53 (Multiple Block Read)

**Figure 40.17** shows a flowchart example when CMD52 (SDIO none abort) is issued at CMD53 (multiple block read) while the SD host interface is in the read wait state.

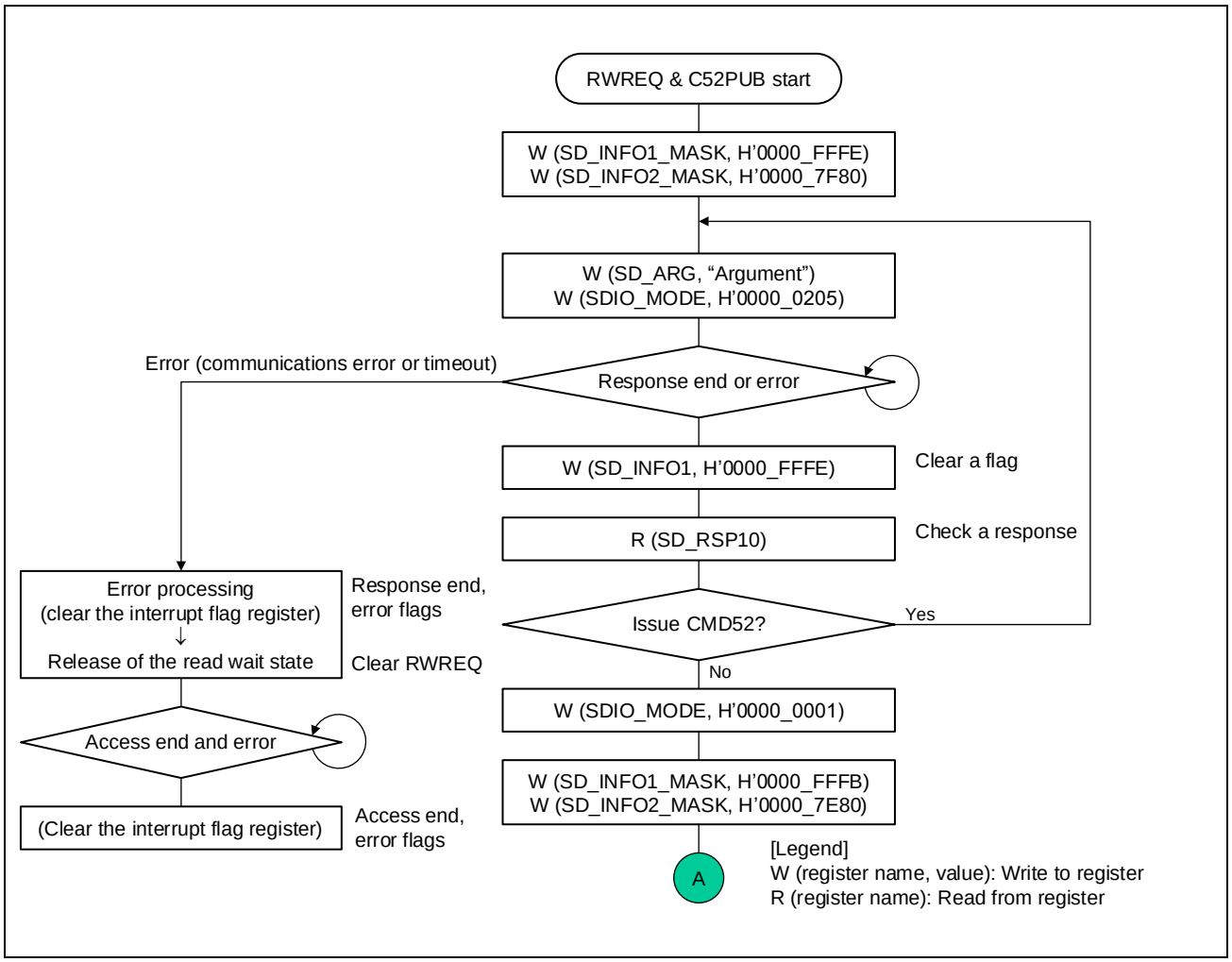
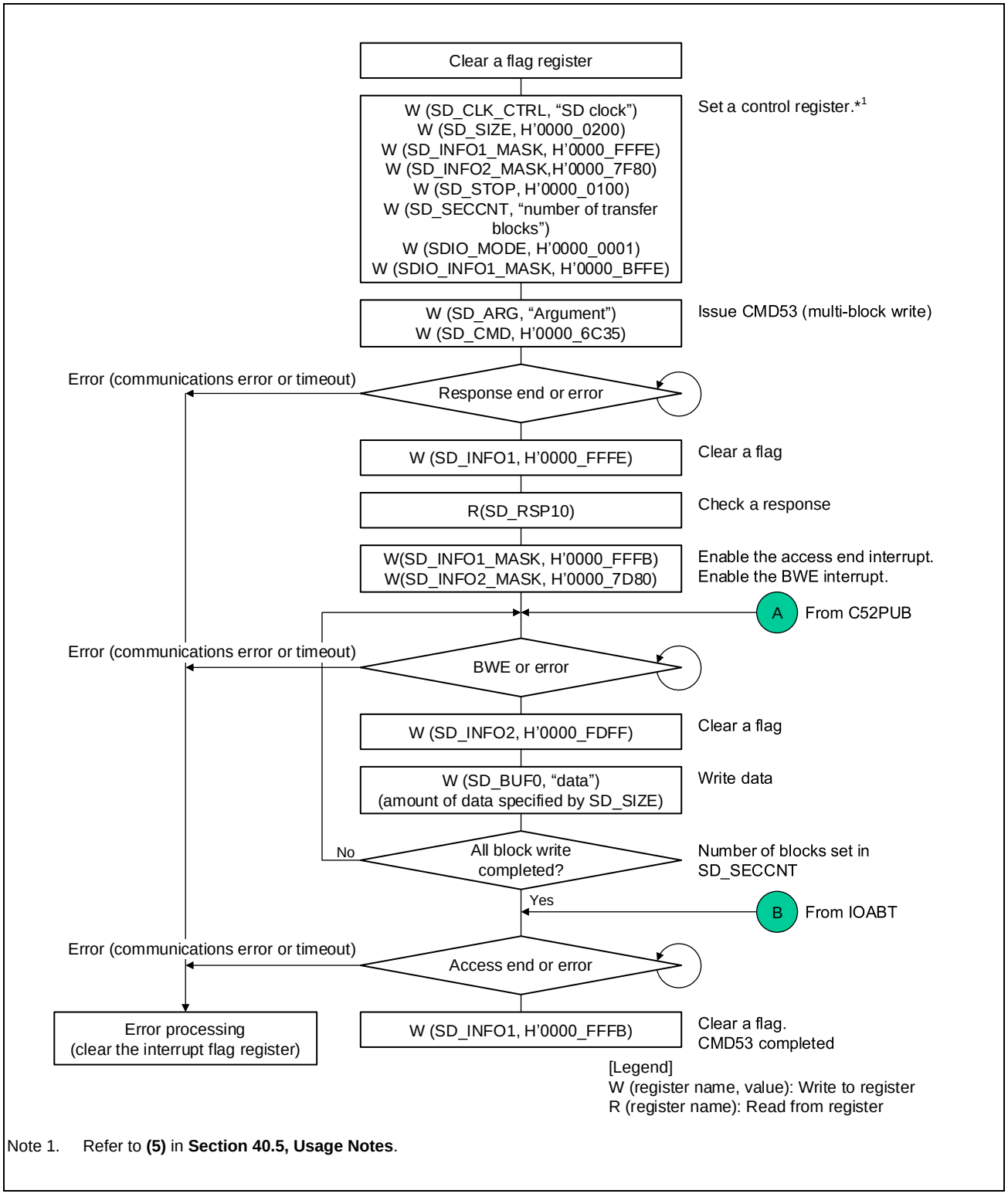



Figure 40.17 Flowchart Example when CMD52 (SDIO None Abort) is Issued after Read Wait State is Entered at CMD53 (Multi Block Read)

## 40.4.9 IO_RW_EXTENDED (CMD53/Multiple Block Write)

### (1) Flowchart

**Figure 40.18** shows a flowchart example for CMD53 (multiple block write).




Note 1. Refer to **(5)** in **Section 40.5, Usage Notes**.

Figure 40.18 CMD53 (Multiple Block Write) Flowchart Example

**Figure 40.19** shows a flowchart example when CMD52 (SDIO abort) is issued at CMD53 (multiple block write).

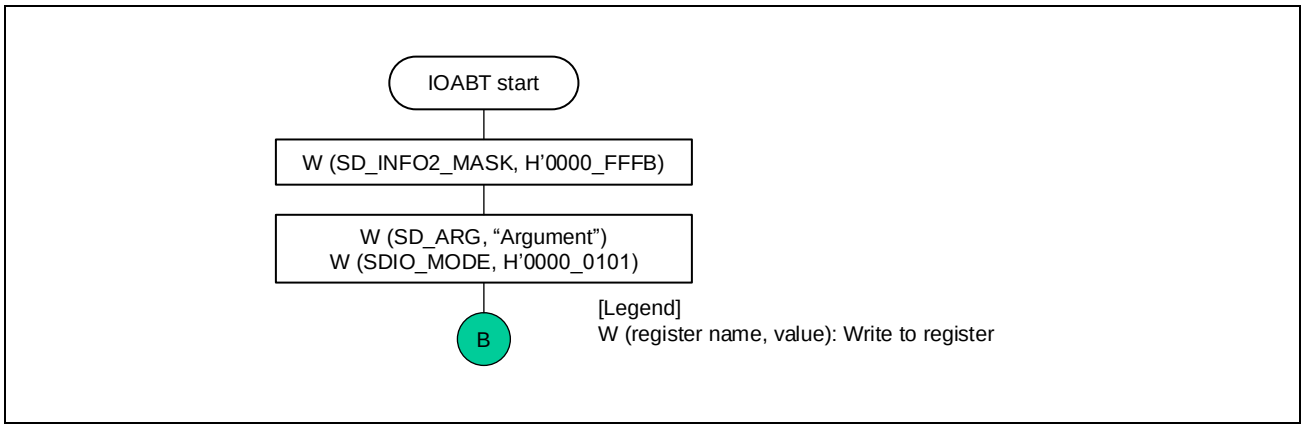



Figure 40.19 Flowchart Example when CMD52 (SDIO Abort) is Issued at CMD53 (Multiple Block Write)

**Figure 40.20** shows a flowchart example when CMD52 (SDIO none abort) is issued at CMD53 (multiple block write).

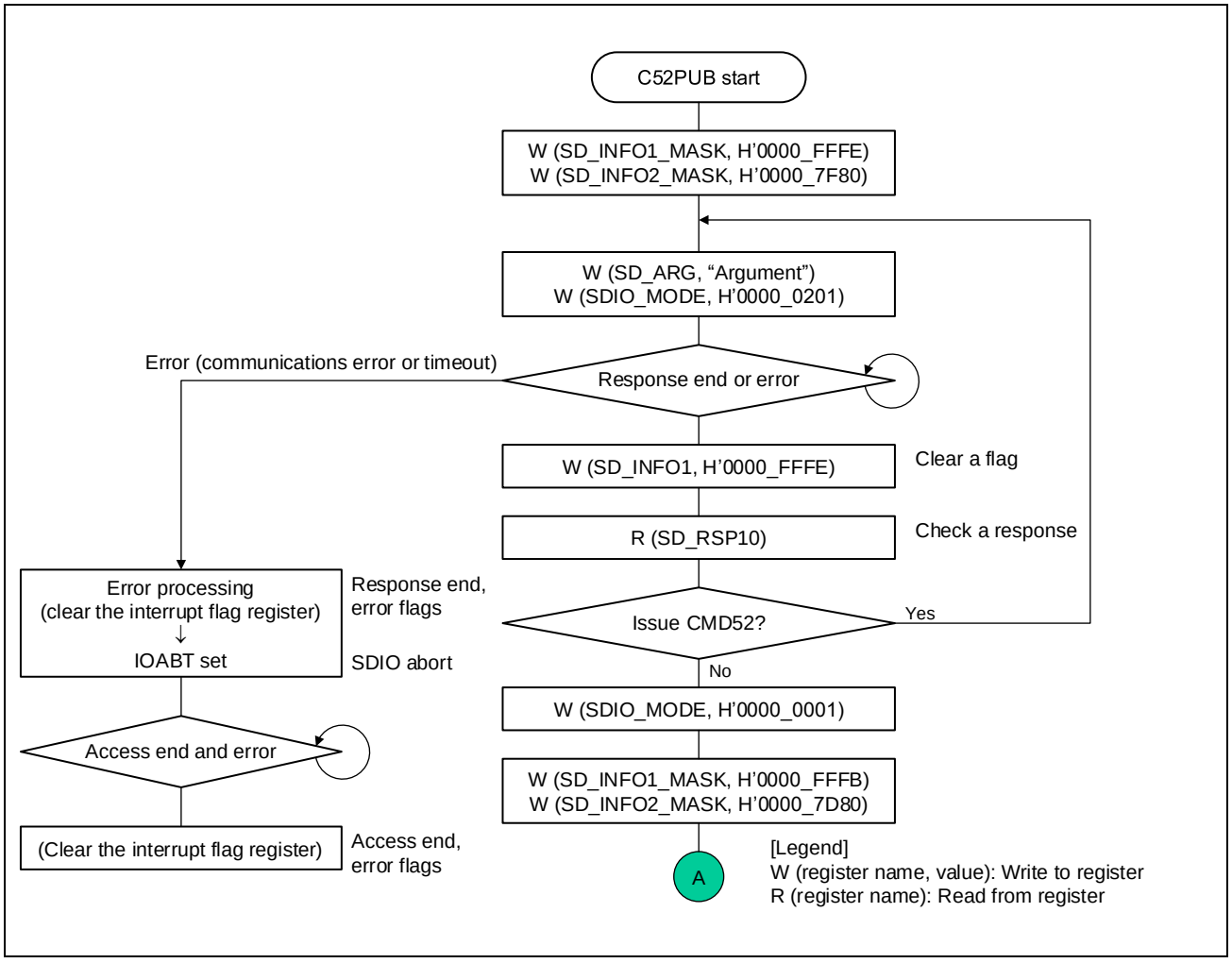
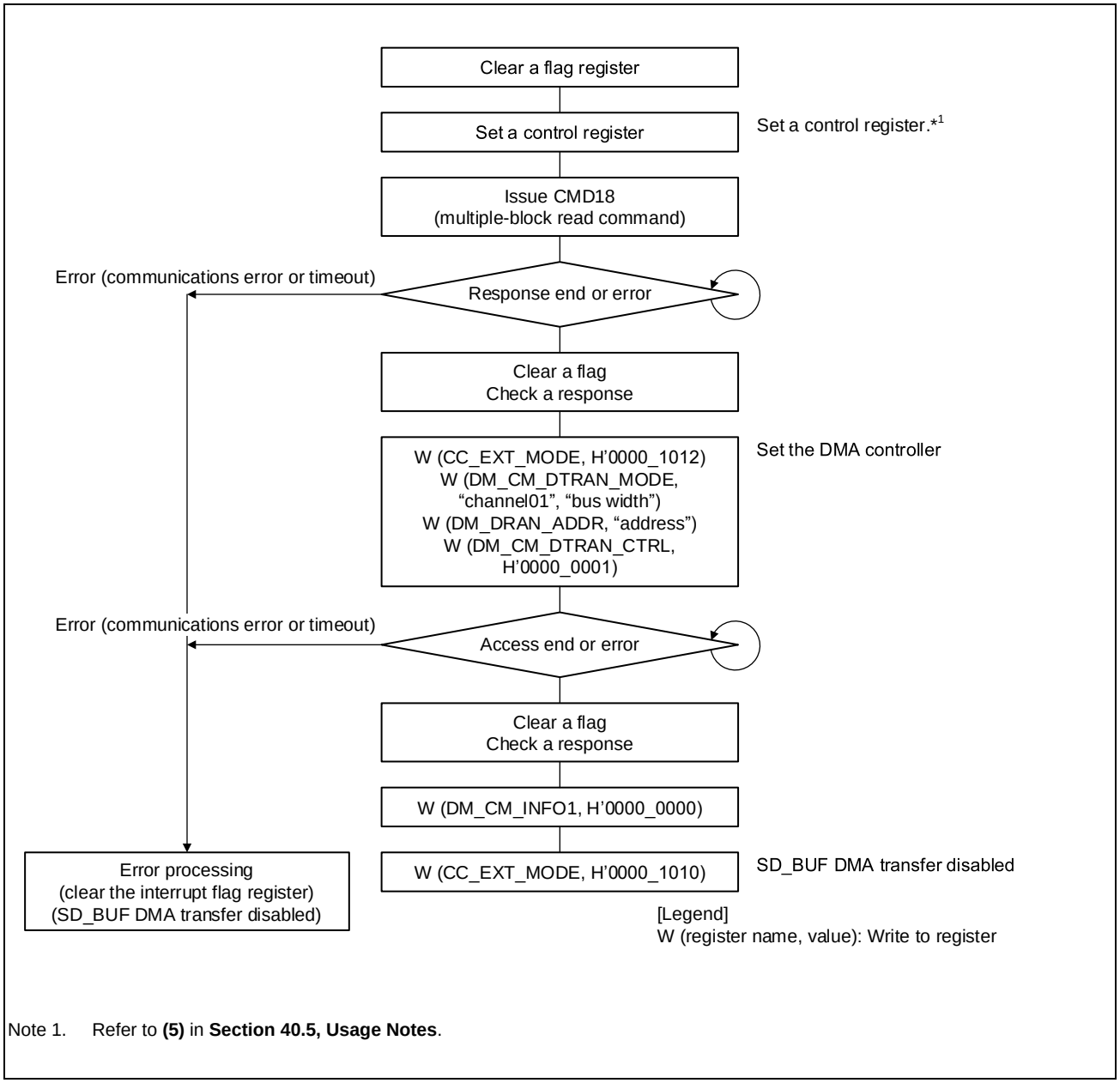



Figure 40.20 Flowchart Example when CMD52 (SDIO None Abort) is Issued at CMD53 (Multiple Block Write)

## 40.4.10 DMA Transfer

### (1) SD_BUF DMA Transfer

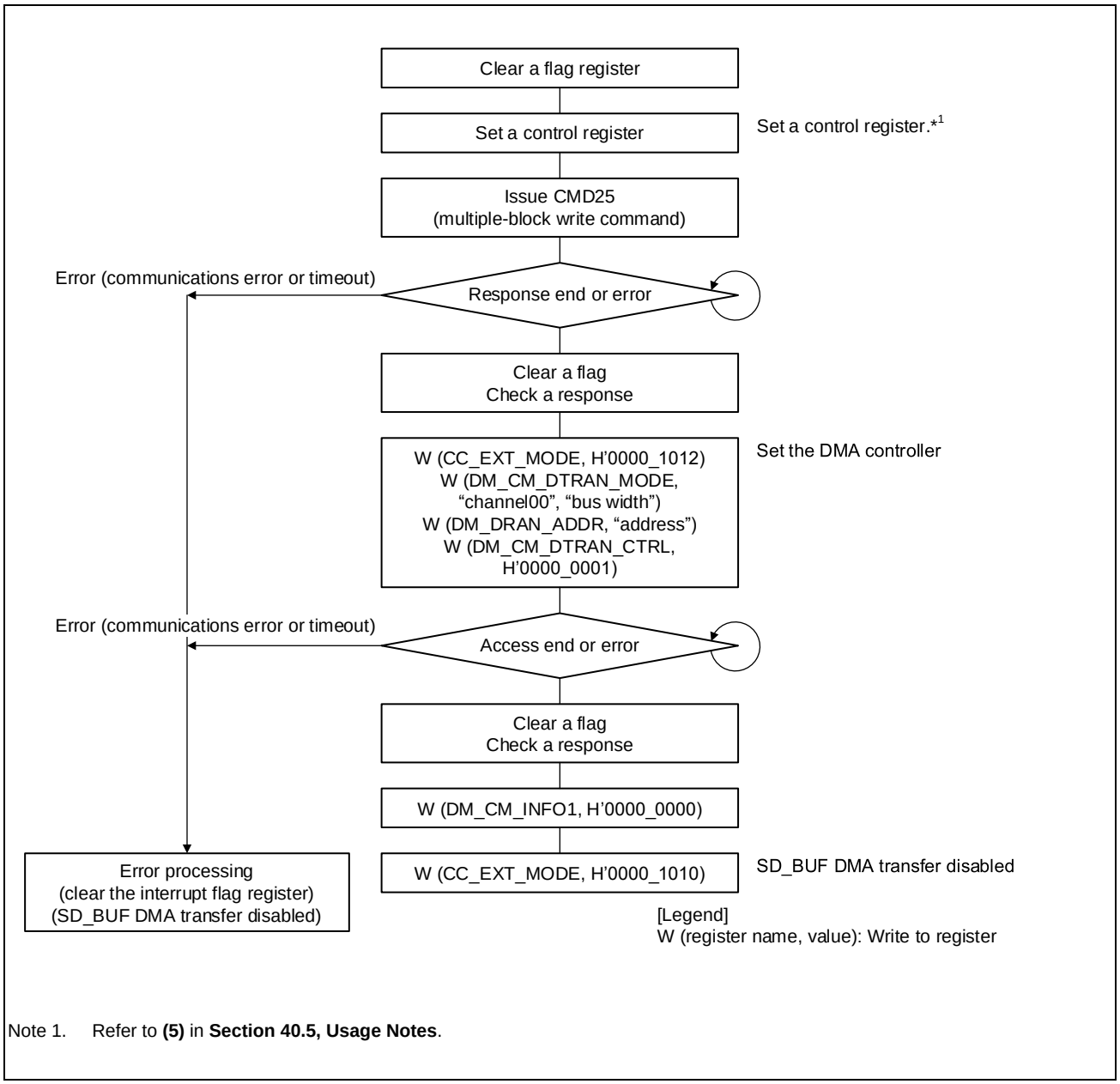
**Figure 40.21** shows a flowchart example for SD_BUF DMA read when CMD18 (multiple block read) is issued.




Note 1. Refer to **(5)** in **Section 40.5, Usage Notes**.

Figure 40.21 SD_BUF DMA Read Flowchart Example

**Figure 40.22** shows a flowchart example for SD_BUF DMA write when CMD25 (multiple block write) is issued.

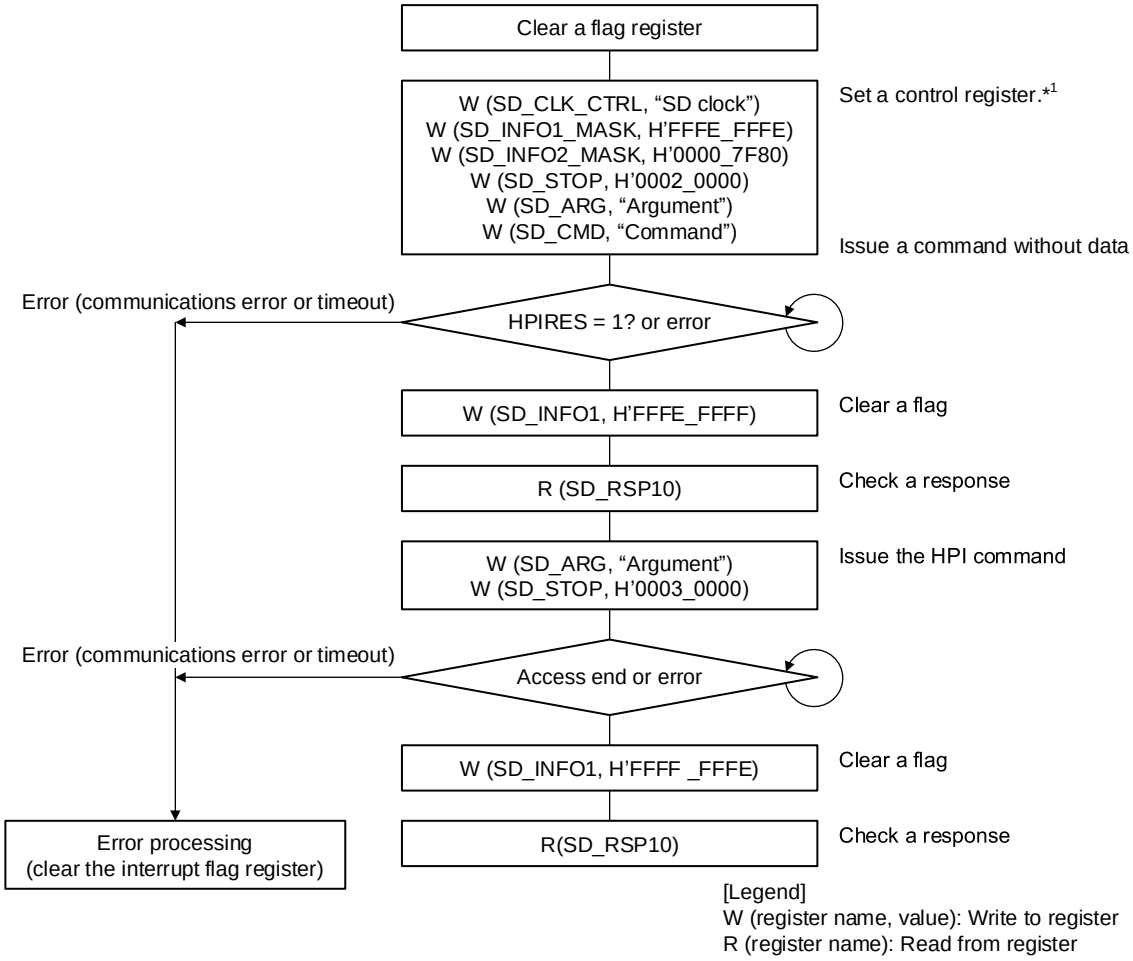



Note 1. Refer to **(5)** in **Section 40.5, Usage Notes**.

Figure 40.22 SD_BUF DMA Write Flowchart Example

## 40.4.11 High-Priority Interrupt (without Data Transfer)

### (1) Flowchart




Note 1. Refer to **(5)** in **Section 40.5, Usage Notes**.

Figure 40.23 Example of the High-Priority Interrupt (without Data Transfer) Flowchart

### (2) Operation for High-Priority Interrupt without Data Transfer

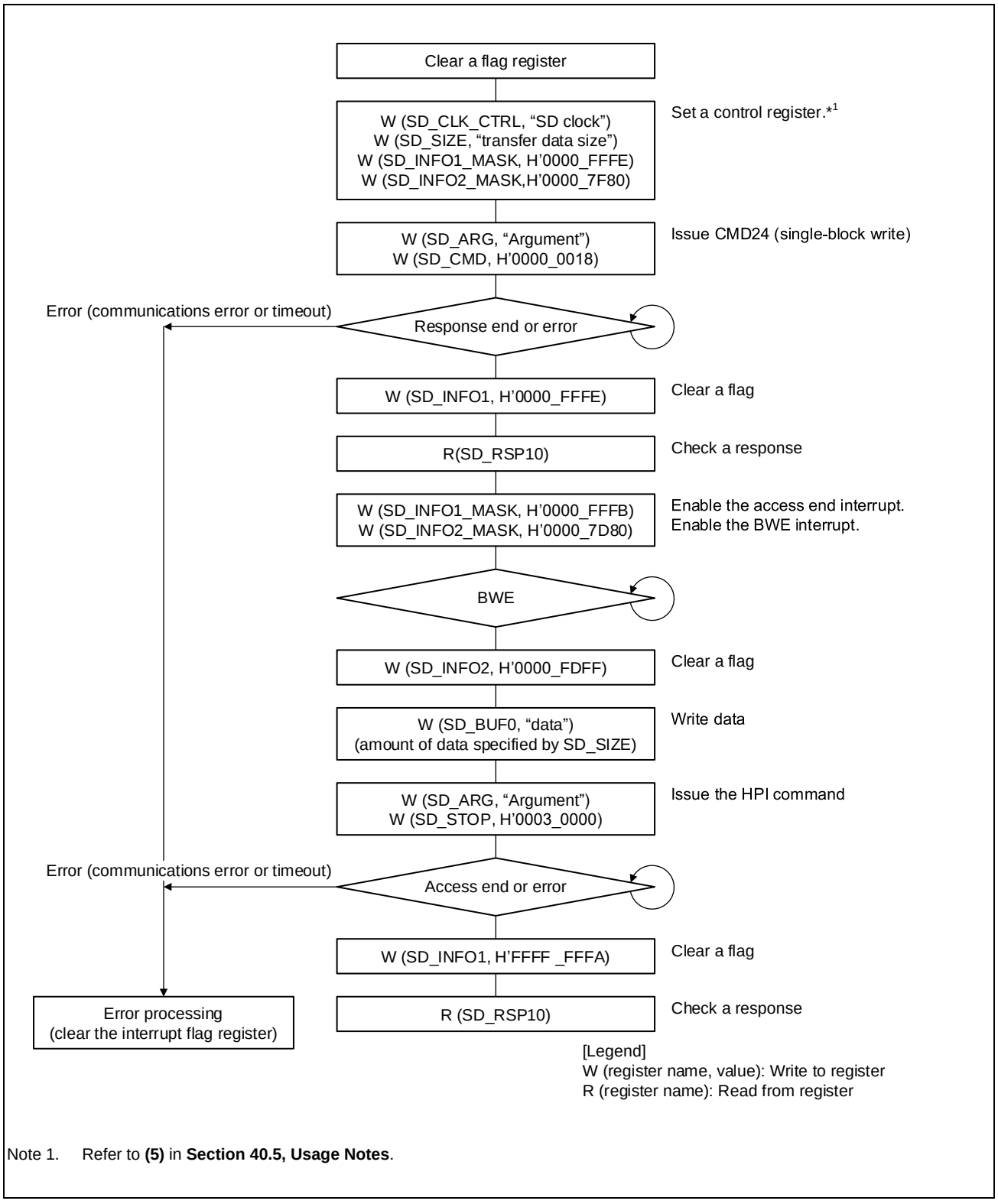
The operation of the high-priority interrupt (HPI) without data transfer is described below.

1. Flag register clear
   First, clear the bits in the flag register. (SD_INFO1 and SD_INFO2)
2. Control register set
   Set the SD clock (SDCLK), HPI enable, interrupt mask, and so on. (SD_CLK_CTRL, SD_INFO1_MASK, and SD_INFO2_MASK)
3. Command issue
   Set the CMD Argument in SD_ARG and write to SD_CMD.
   Accordingly, CMD is issued, and the operation is started.
4. Flag clear
   On receiving the response, the HPIRES bit in SD_INFO1 is set to 1 to generate an interrupt. Clear the HPIRES bit to 0.
5. Read the response from SD_RSP10.
6. HPI command issue
   Set the HPI command argument in SD_ARG and set the HPIMODE and HPICMD bits in SD_STOP to 1.
7. Operation complete
   When reception of the response to the HPI command is completed and the busy state is released, the INFO0 bit in SD_INFO1 is set to 1 to generate an interrupt. Clear the INFO0 bit to 0 to read the response from SD_RSP10.
   This is the end of HPI operation.

Furthermore, perform error processing (clear the interrupt flag register) if an error occurs (a communications error or timeout).

## 40.4.12 High-Priority Interrupt (at Single Block Write)

### (1) Flowchart




Note 1. Refer to **(5)** in **Section 40.5, Usage Notes**.

Figure 40.24 Example of the High-Priority Interrupt (at Single Block Write) Flowchart

### (2) Operation for HPI at Single Block Write

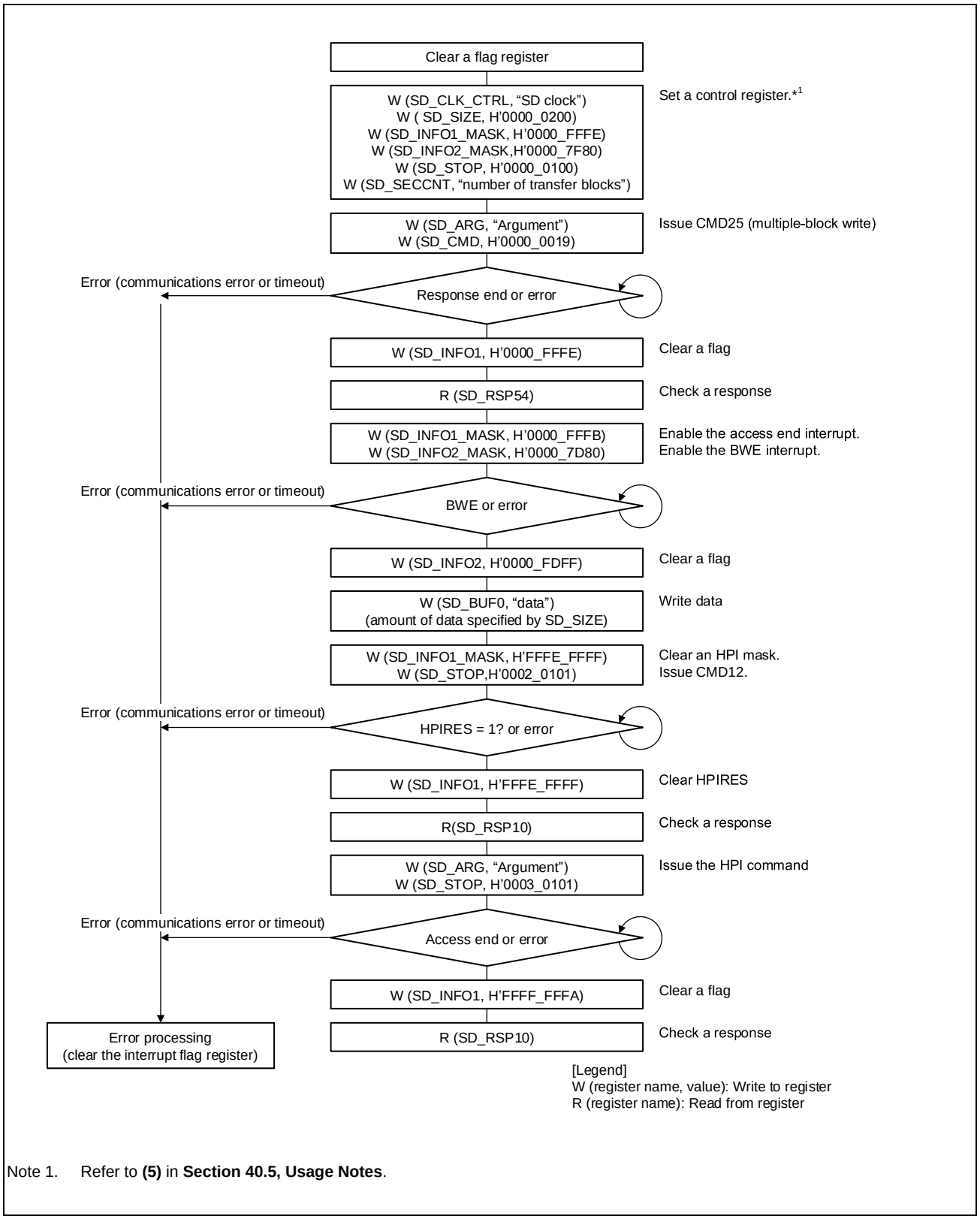
The operation of HPI at single block write is described below.

1. Flag register clear
   First, clear the bits in the flag register. (SD_INFO1 and SD_INFO2)
2. Control register set
   Set the SD clock (SDCLK), transfer data size, interrupt mask, and so on. (SD_CLK_CTRL, SD_SIZE, SD_INFO1_MASK, and SD_INFO2_MASK)
3. Command issue (CMD24)
   Set CMD24 Argument in SD_ARG and write H'0000 0018 to SD_CMD.
   Accordingly, CMD24 is issued, and the single block write operation is started.
4. Response check
   On receiving the response, INFO0 (response end) in SD_INFO1 is set to 1 to generate an interrupt. Clear INFO0 to 0 and read the response from SD_RSP10.
   If the result of response decoding is an error, the command sequence can be halted by setting the STP bit in SD_STP or the IOABT bit in SDIO_MODE to 1. Furthermore, this causes CMD12 and CMD52 to not be issued.
   If the command sequence is halted, the INFO2 bit (access end) in SD_INFO1 is set to 1 to generate an interrupt.
5. Data write and data transmit to SD card
   Write H'0000 FFFB to SD_INFO1_MASK to enable the access end interrupt. In addition, write H'0000 7D80 to SD_INFO2_MASK to enable the BWE interrupt. When SD_BUF0 is ready for the data to be written, the BWE bit in SD_INFO2 is set to 1 to generate an interrupt. Clear the BWE bit to 0 and write the amount of data specified by SD_SIZE to SD_BUF0. When the data write to SD_BUF0 is completed, data is transmitted to the SD card.
6. HPI command issue
   Set the HPI command argument in SD_ARG and set the HPIMODE and HPICMD bits in SD_STOP to 1.
7. Operation complete
   When reception of the response to the HPI command is completed and the busy state is released, the INFO2 and INFO0 bits in SD_INFO1 are set to 1 to generate an interrupt. Clear the INFO2 and INFO0 bits to 0 to read the response from SD_RSP10. This is the end of HPI operation.

Furthermore, perform error processing (clear the interrupt flag register) if an error occurs (a communications error or timeout).

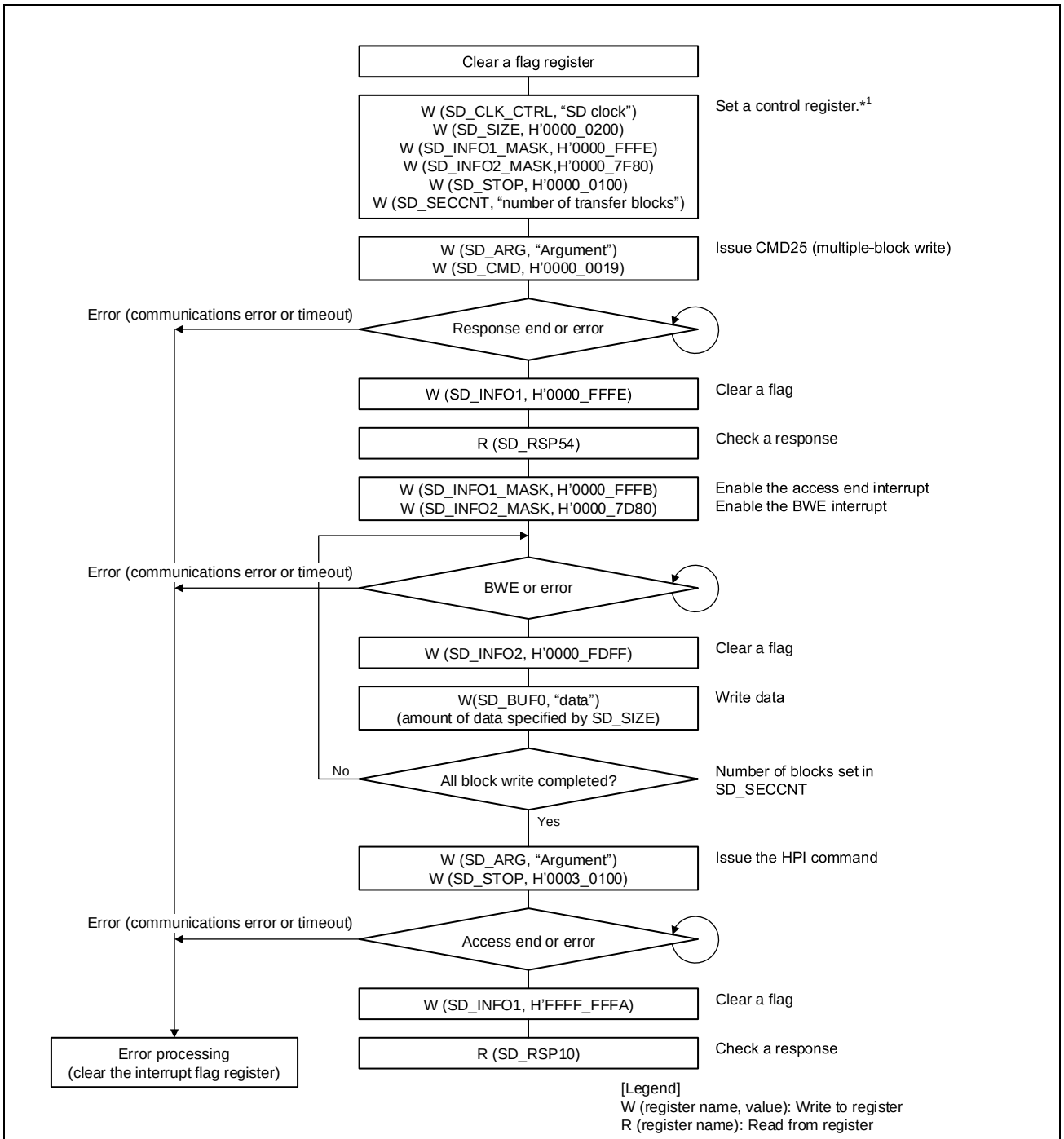
## 40.4.13 High-Priority Interrupt (at Multiple Block Write)

### (1) Flowchart




Note 1. Refer to **(5)** in **Section 40.5, Usage Notes**.

Figure 40.25 Example of the High-Priority Interrupt (at Multiple Block Write) Flowchart (a)




Note 1. Refer to **(5)** in **Section 40.5, Usage Notes**.

Figure 40.26 Example of the High-Priority Interrupt (at Multiple Block Write) Flowchart (b)

### (2) Operation for HPI at Multiple Block Transfer

The operation of HPI at the multiple block write is described below.

#### (a) When not all the data has been written to SD_BUF

1. Flag register clear
   First, clear the bits in the flag register. (SD_INFO1 and SD_INFO2)
2. Control register set
   Set the SD clock (SDCLK), transfer data size, interrupt mask, and so on. (SD_CLK_CTRL, SD_SIZE, SD_INFO1_MASK, and SD_INFO2_MASK)
   Set the SEC bit in SD_STOP to 1, and set the number of transfer blocks in SD_SECCNT.
3. Command issue (CMD25)
   Set CMD25 Argument in SD_ARG and write H'0000 0019 to SD_CMD.
   Accordingly, CMD25 is issued, and the multiple block write operation is started.
4. Response check
   On receiving the response, INFO0 (response end) in SD_INFO1 is set to 1 to generate an interrupt. Clear INFO0 to 0 and read the response from SD_RSP54.
   If the result of response decoding is an error, the command sequence can be halted by setting the STP bit in SD_STP to 1. Setting the STP bit to 1 also causes CMD12 to be issued and the response received. If the command sequence is halted because the access end interrupt has been enabled, an interrupt will be generated by setting of the INFO2 bit (access end) bit in SD_INFO1 to 1 when reception of the response has been completed. Clear the INFO2 bit to 0 and read the response.
5. Data write and data transmit to SD card
   Write H'0000 FFFB to SD_INFO1_MASK to enable the access end interrupt. In addition, write H'0000 7D80 to SD_INFO2_MASK to enable the BWE interrupt. When SD_BUF0 is ready for the data to be written, the BWE bit in SD_INFO2 is set to 1 to generate an interrupt. Clear the BWE bit to 0 and write the amount of data specified by SD_SIZE to SD_BUF0. When the data write to SD_BUF0 is completed, data is transmitted to the SD card.
   Then, the CRC status and busy state are received from the SD card.
   However, a communications error or timeout may be generated if data are being transmitted while writing to SD_BUF0 is in progress.
6. Command issue (CMD12)
   Write H'FFFE FFFF to SD_INFO1_MASK to enable the HPIRES interrupt. Write H'0002 0101 to SD_STOP, which causes CMD12 to be issued.
7. Response check
   When the response is received, the HPIRES bit in SD_INFO1 is set to 1 to generate an interrupt. Clear the HPIRES bit to 0 to read the response from SD_RSP10.
8. HPI command issue (CMD25)
   Set the HPI command argument in SD_ARG and set the HPIMODE and HPICMD bits in SD_STOP to 1.
9. Operation complete
   When reception of the response to the HPI command is completed and the busy state is released, the INFO2 and INFO0 bits in SD_INFO1 are set to 1 to generate an interrupt. Clear the INFO2 and INFO0 bits to 0 to read the response from SD_RSP10. This is the end of HPI operation.

Furthermore, perform error processing (clear the interrupt flag register) if an error occurs (a communications error or timeout).

**(b) When all the data has been written to SD_BUF**

1. Flag register clear
   First, clear the bits in the flag register. (SD_INFO1 and SD_INFO2)
2. Control register set
   Set the SD clock (SDCLK), transfer data size, interrupt mask, and so on. (SD_CLK_CTRL, SD_SIZE, SD_INFO1_MASK, and SD_INFO2_MASK)
   Set the SEC bit in SD_STOP to 1, and set the number of transfer blocks in SD_SECCNT.
3. Command issue (CMD25)
   Set CMD25 Argument in SD_ARG and write H'0000 0019 to SD_CMD.
   Accordingly, CMD25 is issued, and the multiple block write operation is started.
4. Response check
   On receiving the response, INFO0 (response end) in SD_INFO1 is set to 1 to generate an interrupt. Clear INFO0 to 0 and read the response from SD_RSP54.
   If the result of response decoding is an error, the command sequence can be halted by setting the STP bit in SD_STP to 1. Setting the STP bit to 1 also causes CMD12 to be issued and the response received. If the command sequence is halted because the access end interrupt has been enabled, an interrupt will be generated by setting of the INFO2 bit (access end) bit in SD_INFO1 to 1 when reception of the response has been completed. Clear the INFO2 bit to 0 and read the response.
5. Data write and data transmit to SD card
   Write H'0000 FFFB to SD_INFO1_MASK to enable the access end interrupt. In addition, write H'0000 7D80 to SD_INFO2_MASK to enable the BWE interrupt. When SD_BUF0 is ready for the data to be written, the BWE bit in SD_INFO2 is set to 1 to generate an interrupt. Clear the BWE bit to 0 and write the amount of data specified by SD_SIZE to SD_BUF0. When the data write to SD_BUF0 is completed, data is transmitted to the SD card.
   Then, the CRC status and busy state are received from the SD card. Doing this repeats transfer of the number of blocks set in SD_SECCNT.
   However, a communications error or timeout may be generated if data are being transmitted while writing to SD_BUF0 is in progress.
6. HPI command issue
   Set the HPI command argument in SD_ARG and set the HPIMODE and HPICMD bits in SD_STOP to 1.
7. Operation complete
   When reception of the response to the HPI command is completed and the busy state is released, the INFO2 and INFO0 bits in SD_INFO1 are set to 1 to generate an interrupt. Clear the INFO2 and INFO0 bits to 0 to read the response from SD_RSP10. This is the end of HPI operation.

Furthermore, perform error processing (clear the interrupt flag register) if an error occurs (a communications error or timeout).

### 40.4.14 Example of SD_CMD Register Setting

**Table 40.8** lists the example of SD_CMD (SD interface) register setting.

Table 40.8 Example of SD_CMD Register Setting (SD) (1/2)

| Type | Command | Example of SD_CMD Register Setting | Remark |
|---|---|---|---|
| CMD | CMD0 | H'0000 0000 | |
| | CMD2 | H'0000 0002 | |
| | CMD3 | H'0000 0003 | |
| | CMD4 | H'0000 0004 | |
| | CMD5 | H'0000 0705 or H'0000 0005 | |
| | CMD6 | H'0000 1C06 or H'0000 0006 | |
| | CMD7 | H'0000 0007 | When the card is placed in the deselected state, the response timeout flag will be set since there is no response. |
| | CMD8 | H'0000 0408 or H'0000 0008 | |
| | CMD9 | H'0000 0009 | |
| | CMD10 | H'0000 000A | |
| | CMD11 | H'0000 040B or H'0000 000B | |
| | CMD12 | H'0000 000C | |
| | CMD13 | H'0000 000D | |
| | CMD15 | H'0000 000F | |
| | CMD16 | H'0000 0010 | |
| | CMD17 | H'0000 0011 | |
| | CMD18 | H'0000 0012 | With auto CMD12 enabled (other than in SDR104 mode) |
| | | H'0000 7C12 | With auto CMD12 disabled (only in SDR104 mode) |
| | CMD19 | H'0000 1C13 or H'0000 0013 | |
| | CMD20 | H'0000 0514 or H'0000 0014 | |
| | CMD23 | H'0000 0417 or H'0000 0017 | |
| | CMD24 | H'0000 0018 | |
| | CMD25 | H'0000 0019 | With auto CMD12 enabled (other than in SDR104 mode) |
| | | H'0000 6C19 | With auto CMD12 disabled (only in SDR104 mode) |
| | CMD27 | H'0000 001B | |
| | CMD28 | H'0000 001C | |
| | CMD29 | H'0000 001D | |
| | CMD30 | H'0000 001E | |
| | CMD32 | H'0000 0020 | |
| | CMD33 | H'0000 0021 | |
| | CMD38 | H'0000 0026 | |
| | CMD42 | H'0000 002A | |
| | CMD48 | H'0000 1C30 | |
| | CMD49 | H'0000 0C31 | |
| | CMD52 | H'0000 0434 or H'0000 0034 | |

Table 40.8 Example of SD_CMD Register Setting (SD) (2/2)

| Type | Command | Example of SD_CMD Register Setting | Remark |
|---|---|---|---|
| CMD | CMD53 | H'0000 1C35 | Single read |
| | | H'0000 0C35 | Single write |
| | | H'0000 7C35 | Multiple read |
| | | H'0000 6C35 | Multiple write |
| | | H'0000 0035 | The value on the left can be set irrespective of whether single or multi.<br>However, the CF39 bit in SD_ARG must be set as follows.<br>Read: 0<br>Write: 1 |
| | CMD55 | H'0000 0037 | |
| | CMD56 | H'0000 0038 | |
| | CMD58 | H'0000 7C3A | |
| | CMD59 | H'0000 6C3B | |
| ACMD | ACMD6 | H'0000 0046 | |
| | ACMD13 | H'0000 004D | |
| | ACMD22 | H'0000 0056 | |
| | ACMD23 | H'0000 0057 | |
| | ACMD41 | H'0000 0069 | |
| | ACMD42 | H'0000 006A | |
| | ACMD51 | H'0000 0073 | |

## 40.5 Usage Notes

### (1) SD_BUF Illegal Write Access

When writing data to SD_BUF0 after the single block write or multi block write command is issued, the data of the size specified by SD_SIZE must be written to.

If the data of the size which exceeds the size specified by SD_SIZE is written to, the ERR4 bit in SD_INFO2 is set to 1. In addition, the data written to SD_BUF0 may not be transmitted and it causes the SCLKDIVEN bit in SD_INFO2 to hold the value of 0. In such cases, clearing the SDRST bit in SOFT_RST to 0 and then restoring its value to 1 clears the SCLKDIVEN bit to 1.

However, for the single byte (in the case of 16- or 32-bit access) or three bytes (in the case of 32-bit access) when the number of bytes setting in SD_SIZE is odd, or the fraction of bytes when the number of bytes setting in SD_SIZE is even (in the case of 32-bit access), since the portion of dummy data writing is regarded as excess data and ignored, it is not within the scope of the above description (the fraction of bytes: the two bytes that are not in a four-byte unit).

### (2) Block Number Limitation for Multiple Block Read

When performing a multiple block read of one or two blocks, depending on the timing with which the response register is read, the response value may not be read properly. This must be avoided by either of the following countermeasures.

1) When receiving one or two blocks of data, use single block reading.
2) Read the response to CMD18 from SD_RSP54.

#### <Mechanism of incorrect reading>

**Figure 40.27** shows the processing flows of SD host interface (hardware) operation and software operation when a multiple block read is performed on two blocks. As shown in the incorrect operation of **Figure 40.27**, when an interrupt is generated on reception of the CMD18 response and the timing with which the SD card response register (SD_RSP10) is read by the interrupt is delayed, the data during the CMD12 response reception or the CMD12 response may be read. In the case of a multiple block read of three or more blocks, CMD12 is not issued until the block of data has been read, so this problem does not arise. Furthermore, in the case of a multiple block write, since the CMD25 response is read before the block of data is sent, the problem does not arise.

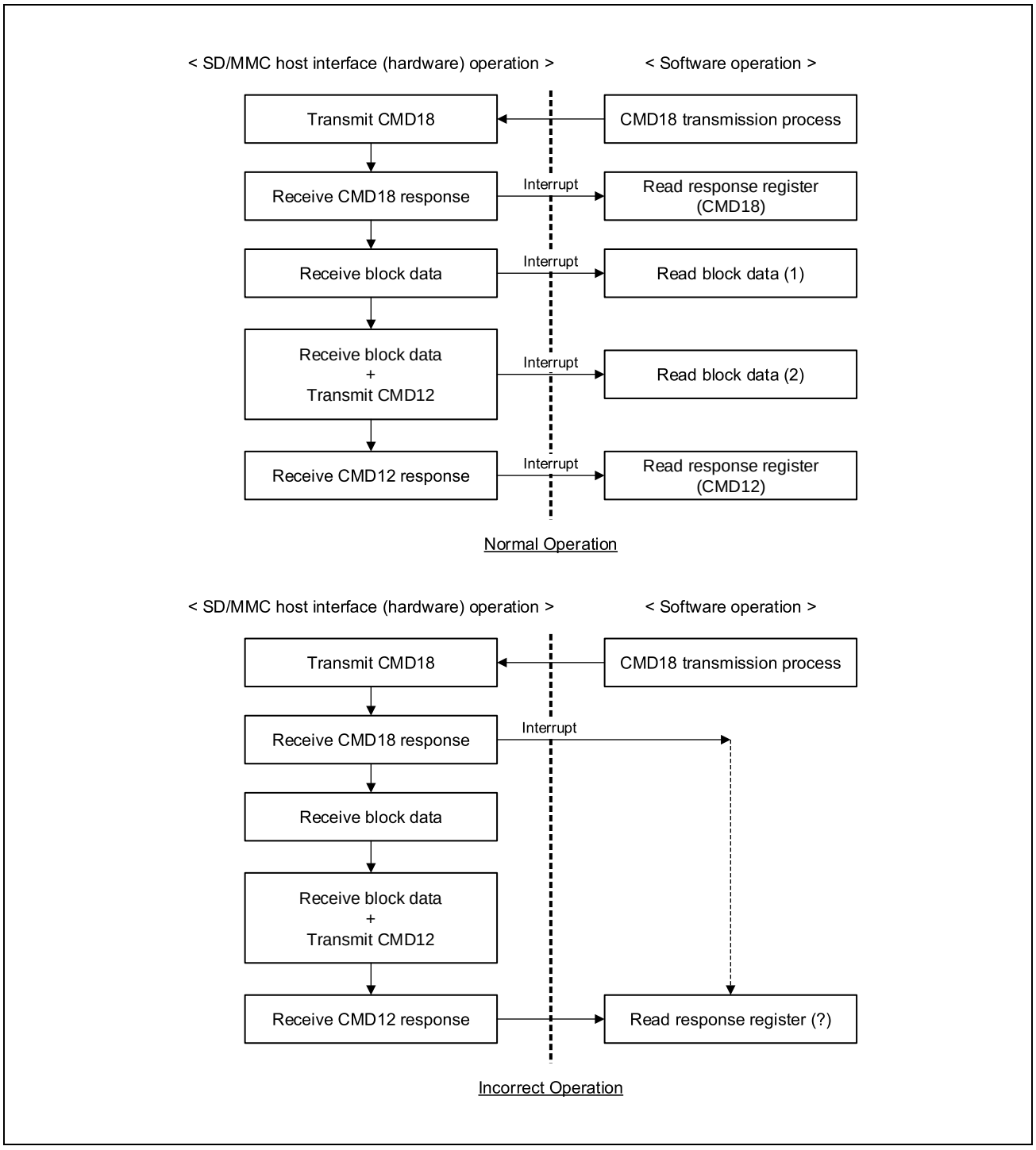



Figure 40.27 Flowcharts for Multiple Block Read Operation (Two Blocks)

### (3) Automatic Control of SDCLK Output

In the SD Card standard, 74 cycles of SDCLK must be output before initialization of the card. For this reason, use automatic control of SDCLK output after 74 cycles of SDCLK have been output. Furthermore, if automatic control of SDCLK output was in use, SDCLK output is stopped on completion of the sequence for a communications error or timeout. Thus, in cases where state transitions within the SD card are necessary and so on after completion of the sequence, release automatic control of SDCLK output and restart supply of SDCLK to the SD card.

### (4) Control of the C52PUB Setting for Multiple Block Write

If the C52PUB bit in SDIO_MODE is set to 1 during a sequence of multiple block write due to CMD53, CMD52 is not issued until SD_BUF becomes empty. For this reason, set the C52PUB bit after suspending writing to SD_BUF by following the appropriate procedure below.

- When DMA transfer is not in use

1. Before setting the C52PUB bit, suspend writing to SD_BUF by making the setting in SD_INFO2 to disable BWE interrupts.
2. Set the C52PUB bit in SDIO_MODE to 1 (so that CMD52 is issued when SD_BUF becomes empty).
3. After the INFO0 interrupt processing in SD_INFO1 due to the issuing of CMD52 has been completed, restart writing to SD_BUF by making the setting in SD_INFO2 to enable BWE interrupts.

- When DMA transfer is in use

1. Every time DMA transfer of the value set in SD_SIZE × n blocks (where n = 1, 2, ...) proceeds, suspend writing to SD_BUF by DMA transfer before the C52PUB bit is set.
2. Set the C52PUB bit in SDIO_MODE to 1 (so that CMD52 is issued when SD_BUF becomes empty).
3. After the INFO0 interrupt processing in SD_INFO1 due to the issuing of CMD52 has been completed, restart writing to SD_BUF by DMA transfer.

### (5) Notes on SD_CLK_CTRL Register Settings

When the SCLKDIVEN bit in SD_INFO2 is 0, SD_CLK_CTRL cannot be written to. Before writing to SD_CLK_CTRL, be sure to check that the SCLKDIVEN bit in SD_INFO2 is 1.

### (6) Restrictions on specifications

1. The SDIO suspend/resume is not supported.
2. The SPI bus is not supported.

### (7) STP bit setting during multiple block read

During execution of multiple block read with automatic CMD12 execution by setting the SEC bit in SD_STOP to 1, even if the STP bit in SD_STOP is set to 1 to forcibly stop the execution, the command sequence may not stop depending on the timing of setting the STP bit.

To avoid this, when setting the STP bit in SD_STOP to 1 during multiple block transfer, clear the SEC bit in SD_STOP to 0 at the same time. (Even when the SCLKDIVEN bit in SD_INFO2 is 0, change the SEC bit from 1 to 0.)

When the command sequence has not stopped because the SEC bit was not cleared to 0, the command sequence can be stopped by clearing the SDRST bit in SOFT_RST to 0.

When forcibly terminating the CMD53 multiple block transfer through the IOABT bit in SDIO_MODE, be sure to leave the SEC bit in SD_STOP as 1.

# 40.6 Sampling Clock Controller (SCC)

## 40.6.1 Features

This module controls a sampling clock (hereafter referred to as the SCC sampling clock) that is used for SD UHS-I/SDR104. When this module is used with the LSI, SD UHS-I/SDR104 can be supported.

## 40.6.2 SCC Block Diagram

**Figure 40.28** shows a block diagram of the sampling clock controller.

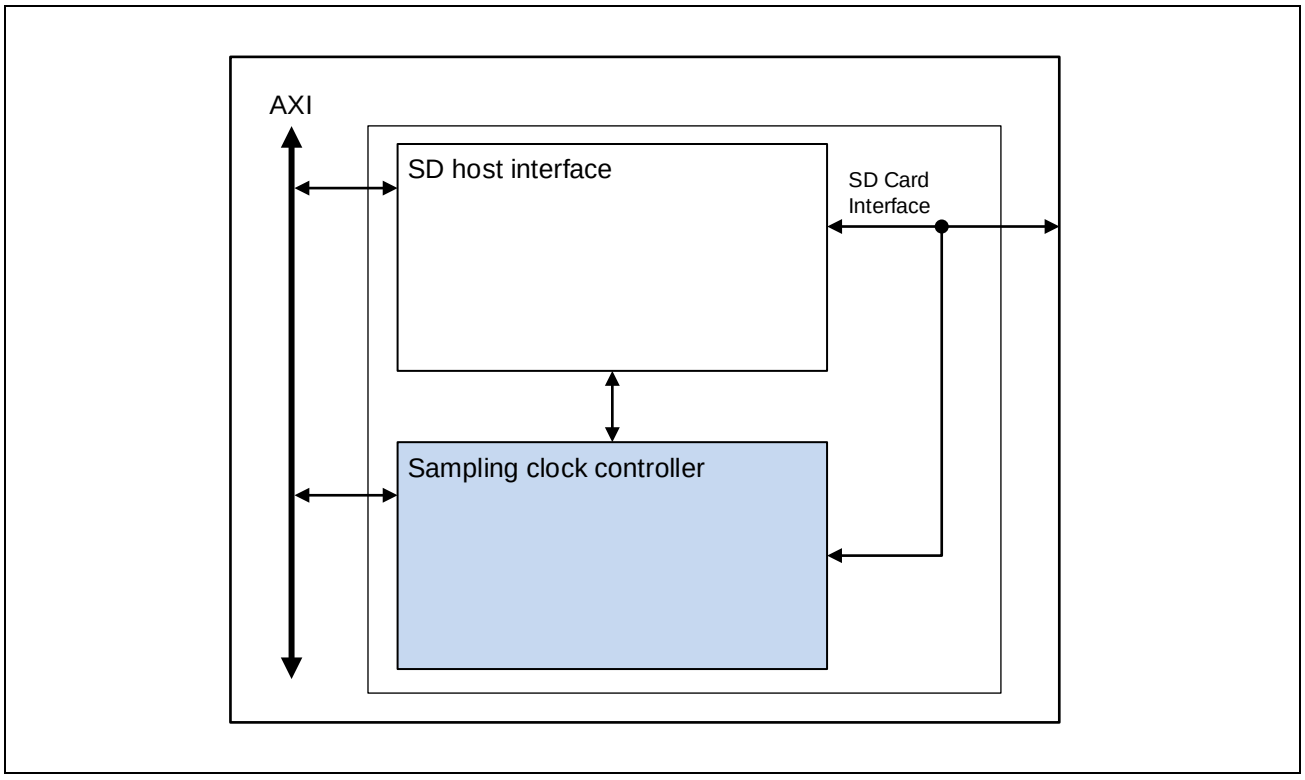



Figure 40.28 Block Diagram of the Sampling Clock Controller

### 40.6.3 SCC Register Configuration

**Table 40.9** shows the SCC registers. Regarding the base address of registers as follows, refter to **Section 40.1.3**.

Channel 0: H'0_11C01000 (Cortex-A55)

Table 40.9 Register Configuration

| Name | Abbreviation | Address [9:0] | Access Size |
|---|---|---|---|
| Initial setting register | SCC_DTCNTL | H'1000 | 32/64 |
| Sampling clock position setting register | SCC_TAPSET | H'1008 | 32/64 |
| Hardware adjustment register 1 | SCC_DT2FF | H'1010 | 32/64 |
| Sampling clock selection register | SCC_CKSEL | H'1018 | 32/64 |
| Sampling clock position correction register | SCC_RVSCNTL | H'1020 | 32/64 |
| Sampling clock position correction request register | SCC_RVSREQ | H'1028 | 32/64 |
| Sampling data comparison register | SCC_SMPCMP | H'1030 | 32/64 |
| Hardware adjustment register 2 | SCC_TMPPORT | H'1038 | 32/64 |

# 40.7 SCC Register Descriptions

## 40.7.1 Initial Setting Register (SCC_DTCNTL)

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 63 to 24 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0.<br>Operation is not guaranteed if a value other than 0 is written to these bits. |
| 23 to 16 | TAPNUM7 to TAPNUM0 | H'08 | R/W | When bits DIV7 to DIV0 in the SD_CLK_CTRL register are H'FF (1: 1 mode), set these bits to H'08. |
| 15 to 1 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0.<br>Operation is not guaranteed if a value other than 0 is written to these bits. |
| 0 | TAPEN | 0 | R/W | SCC Sampling Clock Operation Enable<br>0: SCC sampling clock operation is disabled.<br>1: SCC sampling clock operation is enabled. |

## 40.7.2 Sampling Clock Position Setting Register (SCC_TAPSET)

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 63 to 8 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0.<br>Operation is not guaranteed if a value other than 0 is written to these bits. |
| 7 to 0 | TAPSET7 to TAPSET0 | H'00 | R/W | SCC Sampling Clock Position<br>• Set the tuning result in the range from 0 to TAPNUM-1. |

## 40.7.3 Sampling Clock Selection Register (SCC_CKSEL)

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 63 to 1 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0.<br>Operation is not guaranteed if a value other than 0 is written to these bits. |
| 0 | DTSEL | 0 | R/W | Sampling Clock Selection<br>0: An SCC sampling clock is not used (for other than SDR104).<br>1: An SCC sampling clock is used (for SDR104).<br>• For SDR104, set DTSEL to 1.<br>DIV[7:0] in the SD_CLK_CTRL register to H'FF (1:1 mode).<br>• When this bit is switched, stop the SD clock output from the SD host interface (set SCLKEN in SD_CLK_CTRL to 0). |

### 40.7.4 Sampling Clock Position Correction Register (SCC_RVSCNTL)

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 63 to 16 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0.<br>Operation is not guaranteed if a value other than 0 is written to these bits. |
| 15 to 8 | TAPSEL7 to TAPSEL0 | H'00 | R | SCC Sampling Clock Position Display<br>• Displays the SCC sampling clock position selected by hardware.<br>• After RVSEN has been set to 1, the value may differ from that of TAPSET. |
| 7 to 2 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0.<br>Operation is not guaranteed if a value other than 0 is written to these bits. |
| 1 | RVSW | 0 | R/W | This bit is read as 0. The write value should be 1. |
| 0 | RVSEN | 0 | R/W | SCC Sampling Clock Position Correction Enable<br>0: SCC sampling clock position correction is disabled.<br>1: SCC sampling clock position correction is enabled.<br>• When RVSEN is set to 1 after tuning has been performed, this module corrects the SCC sampling clock position each time of a command sequence of the SD host interface.<br>• However, when RVSERR is 1, this module does not correct the SCC sampling clock position.<br>• While tuning is being performed, set RVSEN to 0. |

### 40.7.5 Sampling Clock Position Correction Request Register (SCC_RVSREQ)

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 63 to 3 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0.<br>Operation is not guaranteed if a value other than 0 is written to these bits. |
| 2 | RVSERR | 0 | R/W | SCC Sampling Clock Position Correction Error<br>0: There is no correction error.<br>1: There is a correction error.<br>• If this bit is set to 1 after a command sequence, write 0 to this bit and perform tuning again.<br>• Ignore this bit while tuning is being performed.<br>• Writing 1 to this bit is disabled and writing 0 to this bit is only enabled. |
| 1 | REQTAPUP | 0 | R/W | SCC Sampling Clock Position Positive Direction Correction Request<br>0: There is no correction request.<br>1: There is a correction request.<br>• If this bit is set to 1 after a command sequence, write 0 to this bit and rewrite TAPSET in the positive direction (when TAPSEL = TAPNUM-1, set 0 to TAPSET).<br>• When RVSEN is 1, this bit is disabled (this bit is not set to 1).<br>• Writing 1 to this bit is disabled and writing 0 to this bit is only enabled. |
| 0 | REQTAPDWN | 0 | R/W | SCC Sampling Clock Position Negative Direction Correction Request<br>0: There is no correction request.<br>1: There is a correction request.<br>• If this bit is set to 1 after a command sequence, write 0 to this bit and rewrite TAPSET in the negative direction (when TAPSEL = 0, set TAPNUM-1 to TAPSET).<br>• When RVSEN is 1, this bit is disabled (this bit is not set to 1).<br>• Writing 1 to this bit is disabled and writing 0 to this bit is only enabled. |

### 40.7.6 Hardware Adjustment Register 1 (SCC_DT2FF)

This register makes a setting that SD_DATA, which has been fetched by the sampling clock at each TAP position, is used in the appropriate timing.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 63 to 16 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0.<br>Operation is not guaranteed if a value other than 0 is written to these bits. |
| 15 to 8 | DT2NE | H'03 | R/W | Hardware Adjustment 2<br>This is a setting register for adjusting the timing inside the IP when using Tuning.<br>When using Tuning, set H'02 to this bit, make the setting before setting 1 in SCC_CKSEL [0] .DTSEL. |
| 7 to 0 | DT2NS | H'00 | R/W | Hardware Adjustment 1<br>This is a setting register for adjusting the timing inside the IP when using Tuning.<br>When using Tuning, set H'07 to this bit., make the setting before setting 1 in SCC_CKSEL [0] .DTSEL. |

### 40.7.7 Sampling data comparison register (SCC_SMPCMP)

Data comparison register indicates the result of the comparison of the sampling data. The subject of the comparison is the before and the behind TAP.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 63 to 25 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0.<br>Operation is not guaranteed if a value other than 0 is written to these bits. |
| 24 to 16 | CMPNGU | All 0 | R/W | Comparison of sampling data with the previous TAP Clock. Bit 16-23 is the comparison result of data 0-7. Bit 24 is the comparison result of CMD.<br>0: Match<br>1: Mismatch<br>< Clear conditions.><br>The start of the command sequence<br>Write to SCC_TAPSET register |
| 15 to 9 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0.<br>Operation is not guaranteed if a value other than 0 is written to these bits. |
| 8 to 0 | CMPNGD | All 0 | R/W | Comparison of sampling data with the after TAP Clock. Bit 0-7 is the comparison result of data 0-7. Bit 8 is the comparison result of CMD.<br>0: Match<br>1: Mismatch<br>< Clear conditions.><br>The start of the command sequence<br>Write to SCC_TAPSET register |

### 40.7.8 Hardware adjustment register 2 (SCC_TMPPORT)

This register makes a setting that SD_DATA, which has been fetched by the sampling clock at each TAP position, is used in the appropriate timing.

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 63 to16 | — | All 0 | R | Reserved<br>These bits are always read as 0. The write value should always be 0.<br>Operation is not guaranteed if a value other than 0 is written to these bits. |
| 15 to 0 | TMPOUT | H’0000 | R/W | Hardware adjustment 3<br>When using the delay tuning mechanism, set the following values.<br>When TAPNUM = 8, set H’0000 to these bits.<br>When operating with other than SDR104, this register adjusts the clock delay of the Flip Flop that latches the received data from the outside of the chip.<br>For this product, set H’0000 when transferring at 3.3V, and set H’0001 when transferring at 1.8V. |

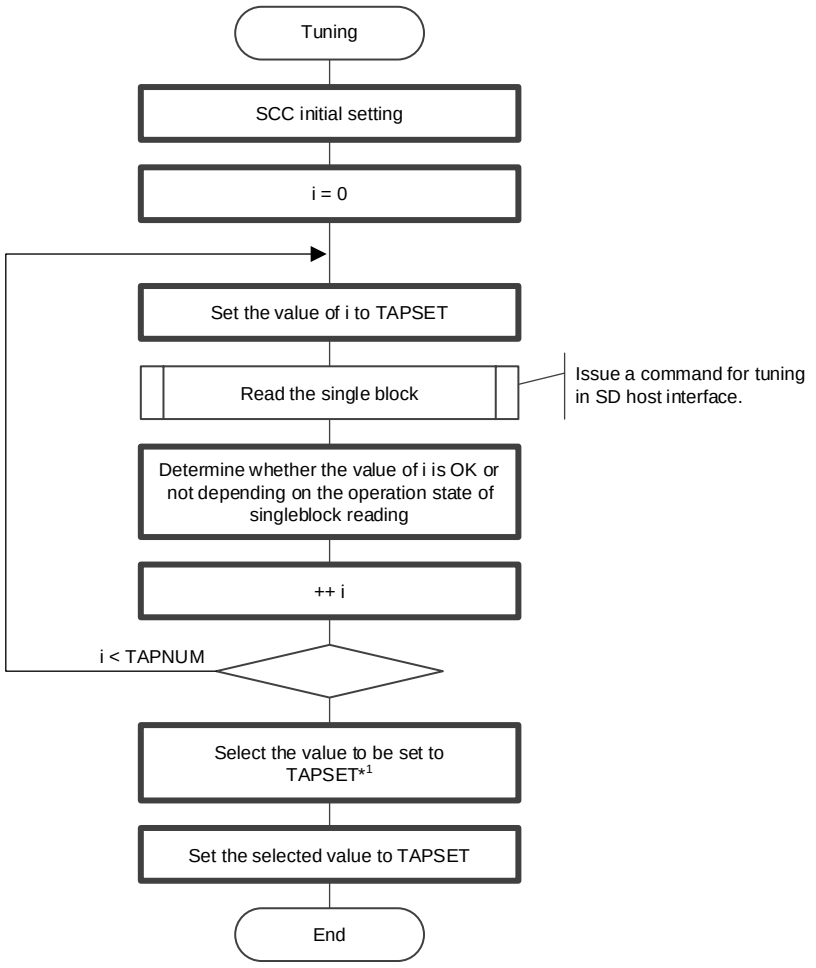
# 40.8 Usage Example of SCC

## 40.8.1 Tuning

SCC is tuned by using operation of single-block reading.

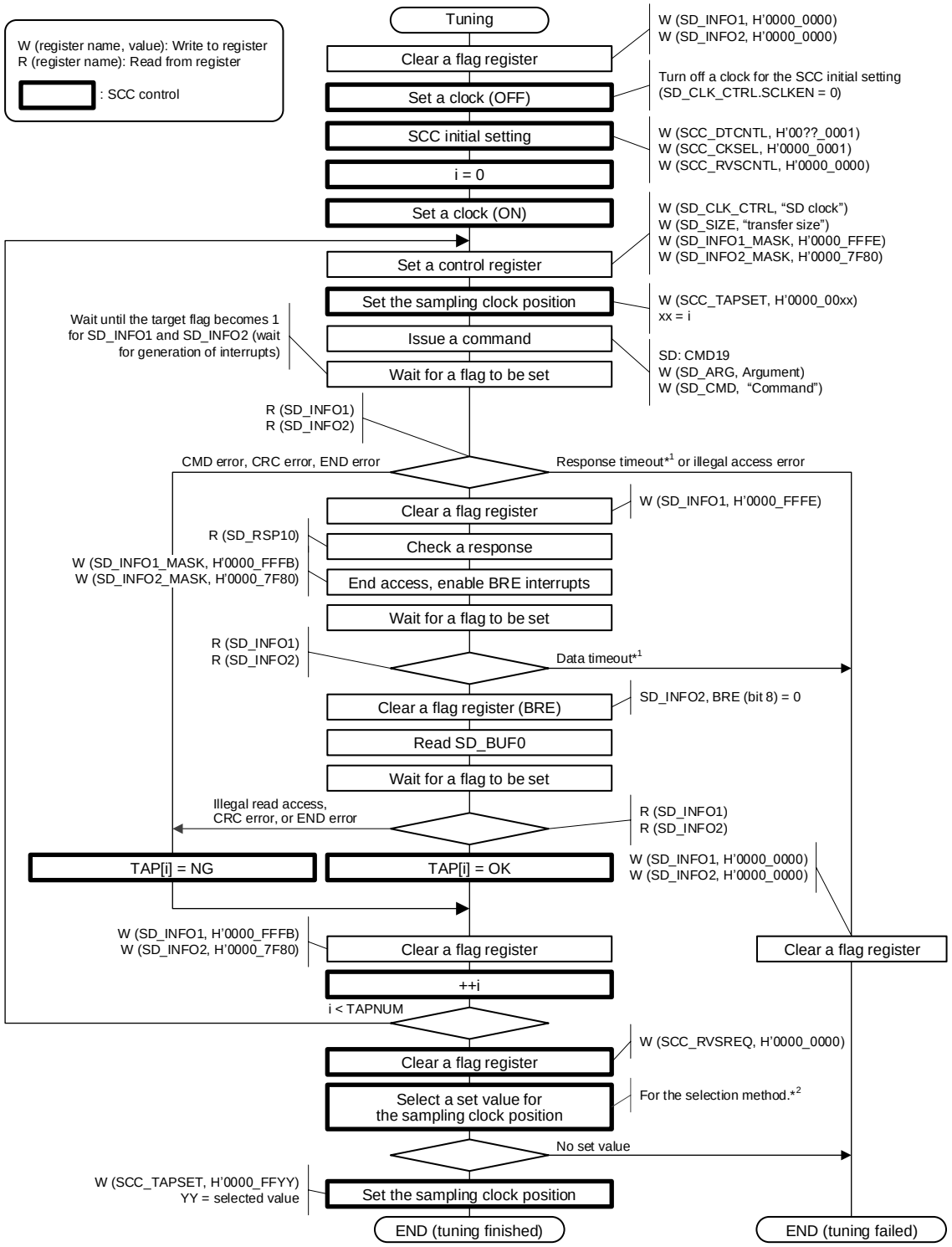
As shown in **Figure 40.29**, check whether the single-block read command normally ends when the sampling clock position is changed from 0 to TAPNUM-1 and save the result. After checking, confirm that there exists the range which has three or more continuous normal ends (OK). Then, the median value within the continuous range is determined as the final adjustment value.

**Figure 40.30** and **Table 40.10** show the detailed tuning flow and the method how to select the sampling clock position (example when TAPNUM = 8).




Note 1. See the example shown in **Table 40.10**.

Figure 40.29 Example of Tuning Flow (Outline)




Note 1. If all of the TAP[i] has become "NG".

Note 2. See **Table 40.10**.

Figure 40.30 Example of Tuning Flow (Detailed)

Table 40.10 Example of the Method How to Select the Sampling Clock Position (when TAPNUM = 8)

| Item | i | Case 1 | Case 2 | Case 3 | Case 4 | Case 5*1 |
|---|---|---|---|---|---|---|
| TAP[i] | 0 | NG | OK | NG | OK | OK |
| | 1 | OK | OK | NG | NG | OK |
| | 2 | OK | NG | OK | NG | OK |
| | 3 | OK(←) | NG | OK | NG | OK |
| | 4 | OK | NG | NG | NG | OK |
| | 5 | OK | OK | NG | OK | OK |
| | 6 | NG | OK | NG | OK(←) | OK |
| Max. value→ | 7 | NG | OK(←) | NG | OK | OK |
| | (0) | NG | OK | NG | OK | OK |
| | (1) | OK | OK | NG | NG | OK |
| | (2) | OK | NG | OK | NG | OK |
| | (3) | OK | NG | OK | NG | OK |
| | (4) | OK | NG | NG | NG | OK |
| | (5) | OK | OK | NG | OK | OK |
| | (6) | NG | OK | NG | OK | OK |
| | (7) | NG | OK | NG | OK | OK |
| Selected value | | i = 3 | i = 7 | Fail | i = 6 or 7 | i = 0 to 7 |

**Remarks:** (←): Example of selection, (x): repeated display of index x of TAP[x]

(a) The sampling clock position is selected by considering a margin in the range which has three or more continuous 'TAP[i] = OK'.

(b) The sampling clock position is repeated from 0 after the maximum value (TAPNUM-1). In case 2 above, that position is continued in the order of 5→6→7→0→1.

Note 1. If all of the TAP [i] is OK, the sampling clock position is selected by identifying the change point of data. Change point of the data can be found in the value of SCC_SMPCMP register. Usage example is **Section 40.8.3, Change point of the input data**.

## 40.8.2 Sampling Clock Position Correction after Tuning

After tuning, correction of the sampling clock position may be required when a command is issued.

There are manual and automatic correction methods. After a command sequence, if the CMD, CRC, END error or time out occurs or the correction error occurs, tuning will be performed again. The following shows examples of manual and automatic correction methods.

### (1) Manual correction of the sampling clock position

**Figure 40.31** shows the flow of manual correction of the sampling clock position. **Table 40.11** shows set values determined when correction is required (when TAPNUM = 8).

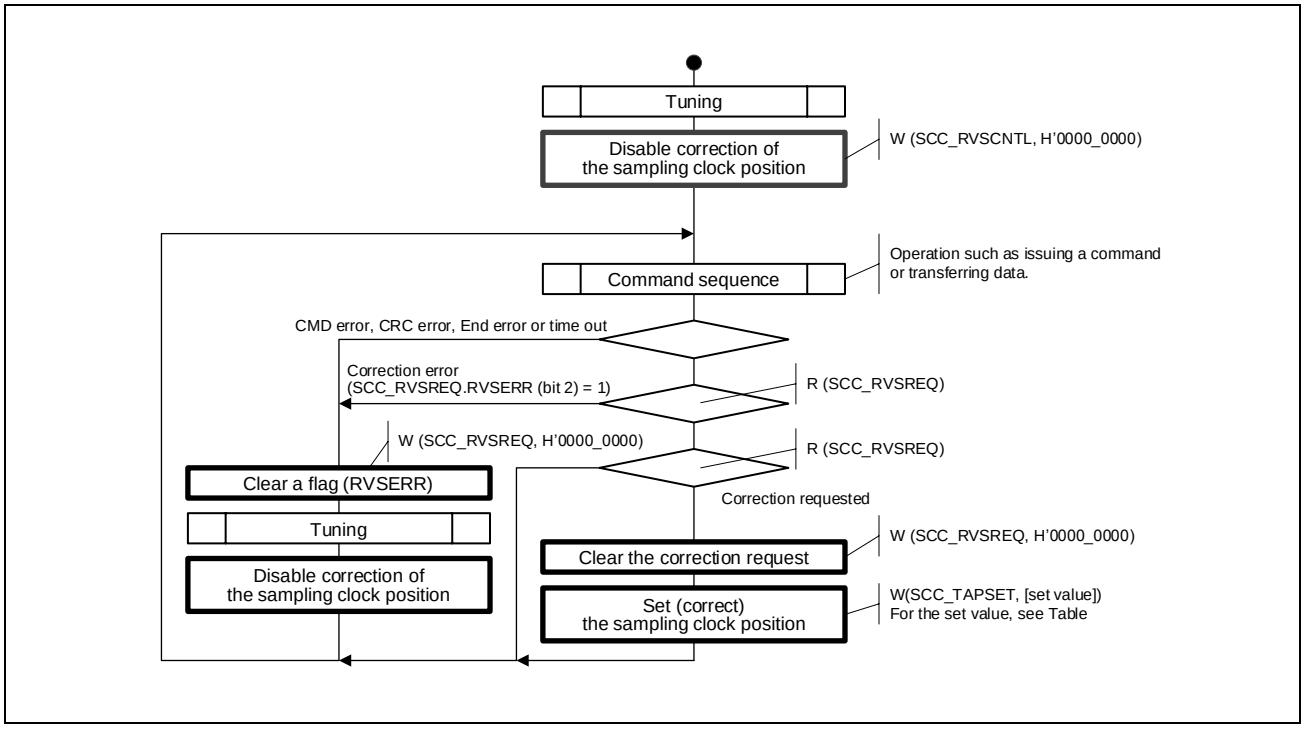



Figure 40.31 Flow of Manual Correction of the Sampling Clock Position (Example)

Table 40.11 Set Values for TAPSET when Correction is Required (when TAPNUM = 8)

| No. | Current value of TAPSET | Value set to TAPSET when REQTAPUP = 1 | Value set to TAPSET when REQTAPDWN = 1 |
|---|---|---|---|
| 1 | 0 | 1 | 7 |
| 2 | 1 | 2 | 0 |
| 3 | 2 | 3 | 1 |
| 4 | 3 | 4 | 2 |
| 5 | 4 | 5 | 3 |
| 6 | 5 | 6 | 4 |
| 7 | 6 | 7 | 5 |
| 8 | 7 | 0 | 6 |

**Note:** As is the case in the tuning selection method, the sampling clock position is 0 after the maximum value (TAPNUM-1).

### (2) Automatic correction of the sampling clock position

**Figure 40.32** shows the flow of automatic correction of the sampling clock position.

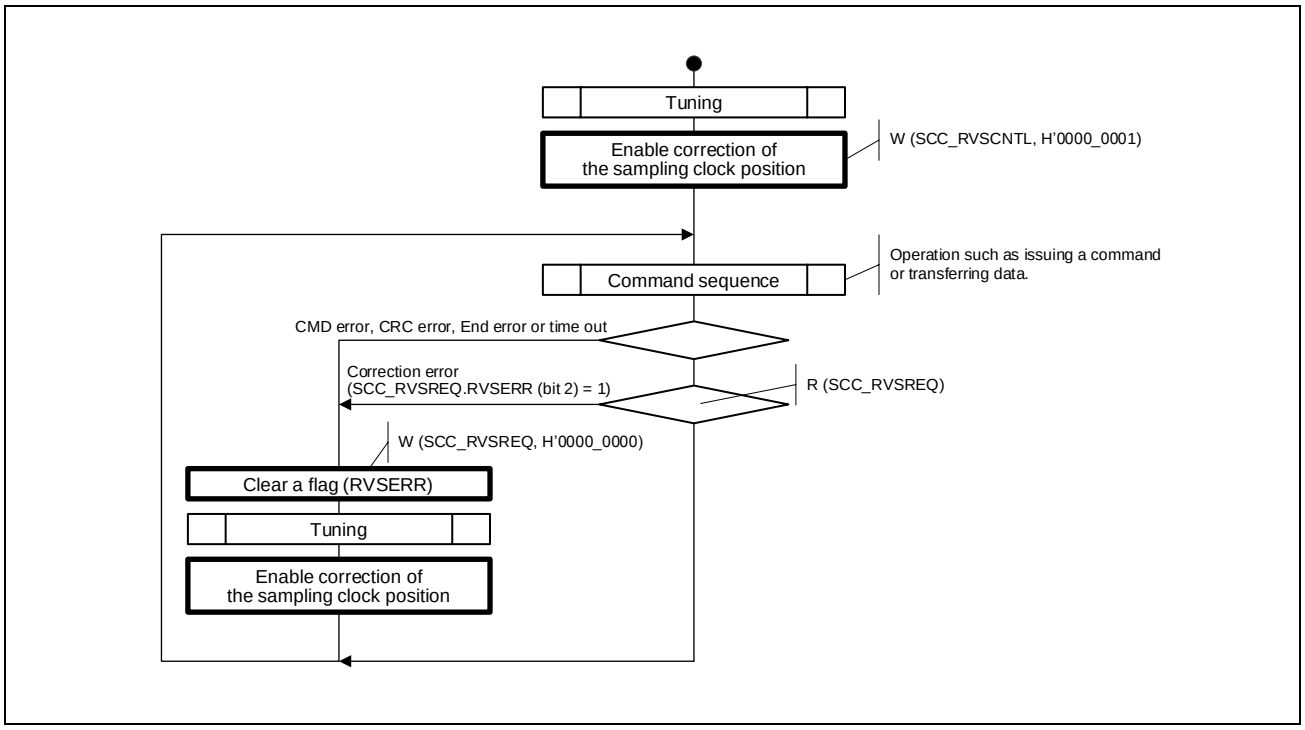
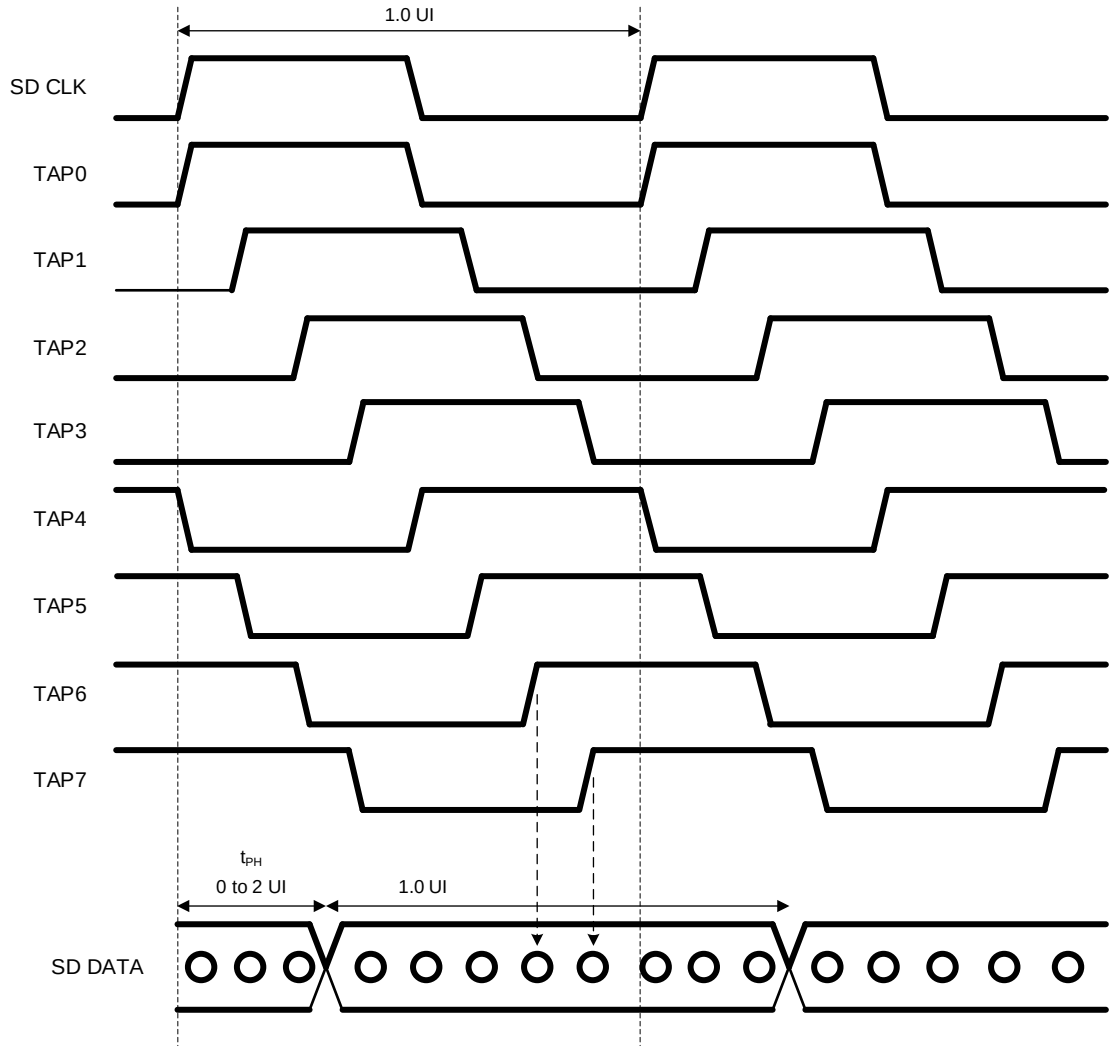



Figure 40.32 Flow of Automatic Correction of the Sampling Clock Position (Example)

### 40.8.3 Change point of the input data

Tuning captures the data by the TAP clock selected. However, also captures the data by the previous TAP clock and the behind the TAP clock at the same time. This result is reflected in the sampling data comparison register (SCC_SMPCMP). Point of mismatch before and after the selected TAP clock is the changing point of the data. In this example, it is desirable to set as TAP6 or TAP7.




**Note:** After recognizing the changing point of the input data, need to set the TAP clock.

Figure 40.33 Example, All taps is OK.

**Figure 40.33** shows change point of data between TAP2 and TAP3. Change point of the data can be confirmed by sampling data comparison register (SCC_SMPCMP). When the tuning of TAP2 or TAP3, CMPNGU bit of sampling data comparison register indicates a mismatch. As the width of the input data is 1 (UI), select TAP6 or TAP7 which is the median of next TAP3 from TAP3.

# 41. General Purpose Input Output Port (GPIO)

This LSI can use up to 70 general-purpose I/O ports.

Each port of the general-purpose I/O port is multiplexed with the terminal of the peripheral function, and either the general-purpose I/O port function or the peripheral function can be selected by setting the register.

## 41.1 Features

### 41.1.1 General Purpose Input Output Port (GPIO)

General-purpose input/output ports (GPIOs) provide general-purpose ports that can be configured as inputs or outputs.

When set as an output, it can be written to an internal register to control the state driven by the output pin. When set as an input, you can read the state of the input by reading the state of the internal registers. You can also generate interrupts using the GPIO input pin.

- GPIO Output Control (Port Register [Pn])
- GPIO Input, Output Enable (Port Mode Register [PMn])
- GPIO Input Monitor (Port Input Register [PINn])
- Interrupt used the GPIO input (Interrupt Enable Control Register[ISEL])

Also, it able to control the following on some ports

- IO voltage mode control for each port
- Buffer drive ability control for each port
- Slew Rate Switching for each port
- Pull Up / Pull down Switching for each port
- Digital noise filter control for each port

### 41.1.2 Port Function Control

Switching of each port function can be controlled by the following register settings.

- GPIO function and peripheral function selection (Port Mode Control Register[PMCn])
- Selection of each peripheral function (Port Function Control Register[PFCm]) [Function1 to 8]

NOTE

Please refer to another file for the pin function for the peripheral functions that can be selected.
Please refer to above excel file for the “Pin function list” described in each chapter.

Do not select the same peripheral function for multiple different external terminals.

### 41.1.3 Special Purpose Port Function Control

The settings of the following specific ports can be controlled by registers.

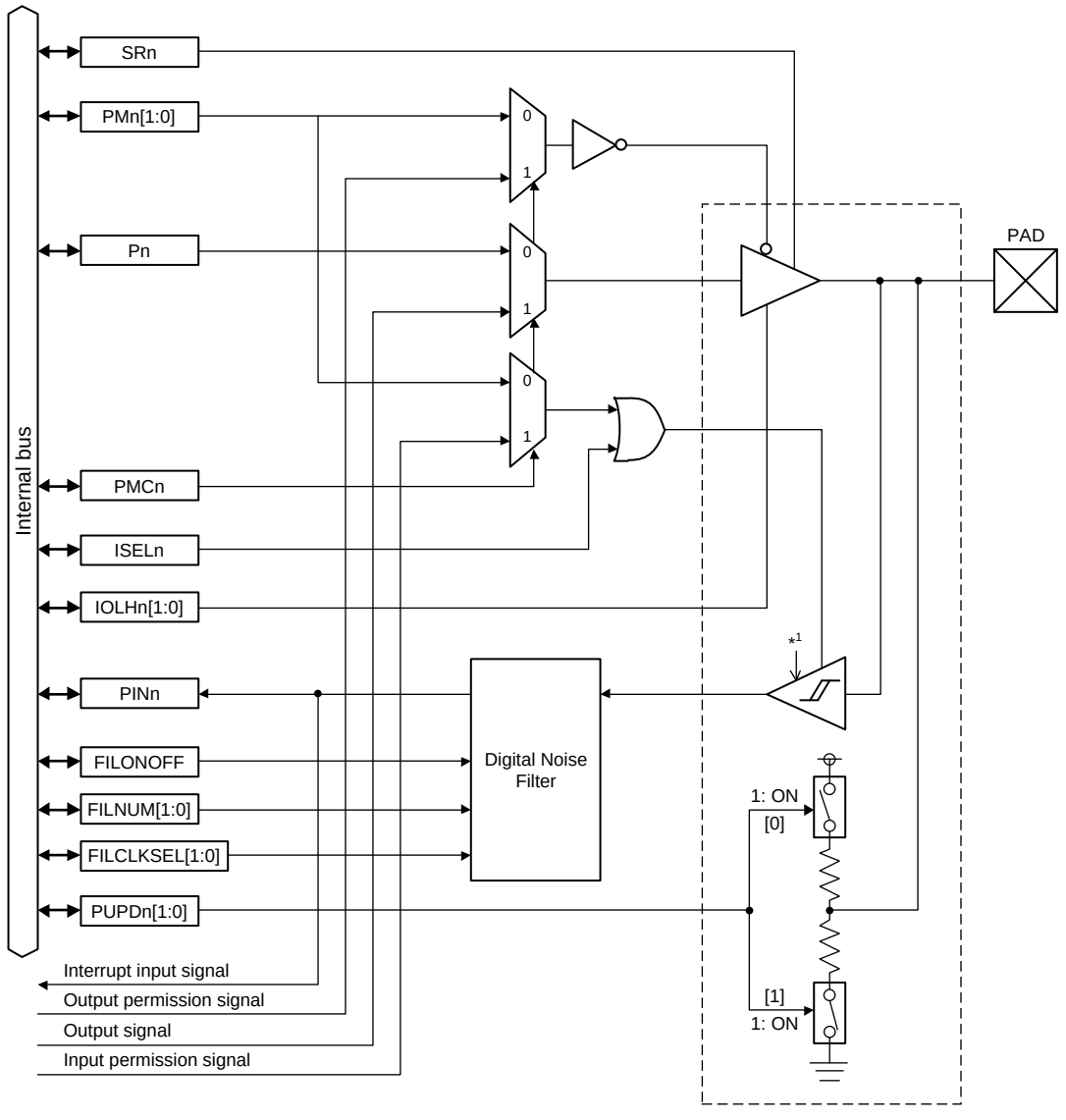
- IO voltage mode control for each port
- Buffer drive ability control for each port
- Slew Rate Switching for each port
- Digital noise filter control for each port

The specific ports shown below are targeted.

- Target Functions: SDHI ch0, QSPI、RIIC
- Target Signals: NMI, TMS/SWDIO, WDTOVF_PERROUT#

### 41.1.4 General Purpose Input Output Port Configuration

The configuration of the general-purpose I/O port is shown in **Figure 41.1**.




Note 1. When RIIC * is selected as the peripheral module, the Schmitt trigger circuit is enabled.

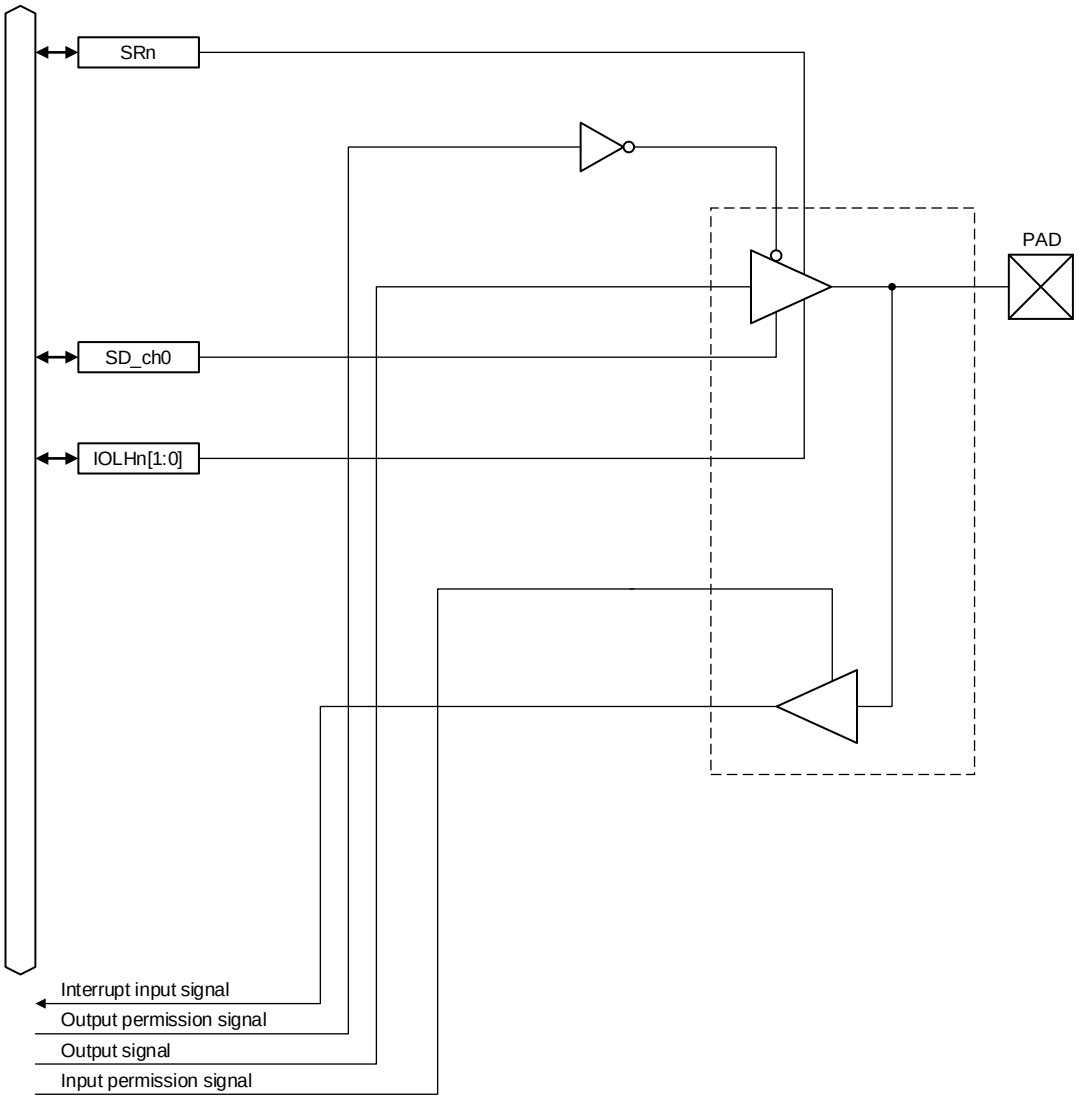
Note. P22_0 to P22_4 pins have function of "IEN" register and "SD_ch0" register in addition to the above. P23_0 to P23_1 pins have function of "QSPI" register in addition to the above.

Figure 41.1 P0_0 to P12_1, P20_0 to P23_1 (Multiplexed peripheral functions)

### 41.1.5 Special purpose port Configuration

The functions and the signal of configurations shown below are shown in **Figure 41.2** to **Figure 41.7**.

- Target Functions: SDHI ch0, QSPI, RIIC
- Target Signals: NMI, TMS/SWDIO, WDTOVF_PERROUT#




**Note:** The IENn registers do not exist at the SD0_CLK and SD0_RST# pins.

Figure 41.2 SDHI ch0 (SD0_CLK)

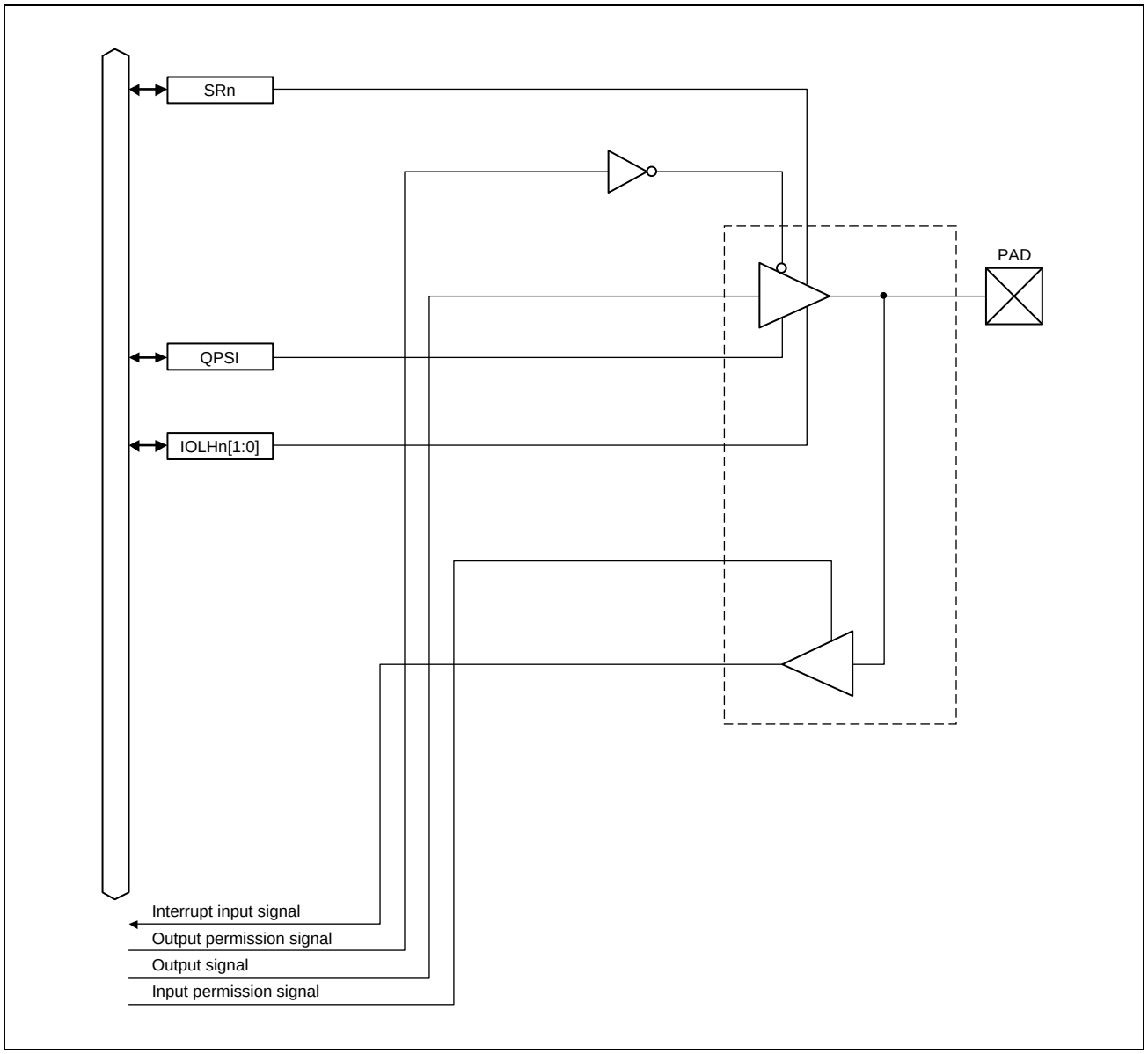



Figure 41.3 QSPI (QSPI0_SPCLK, QSPI0_IO[1:0], QSPI0_SSL)

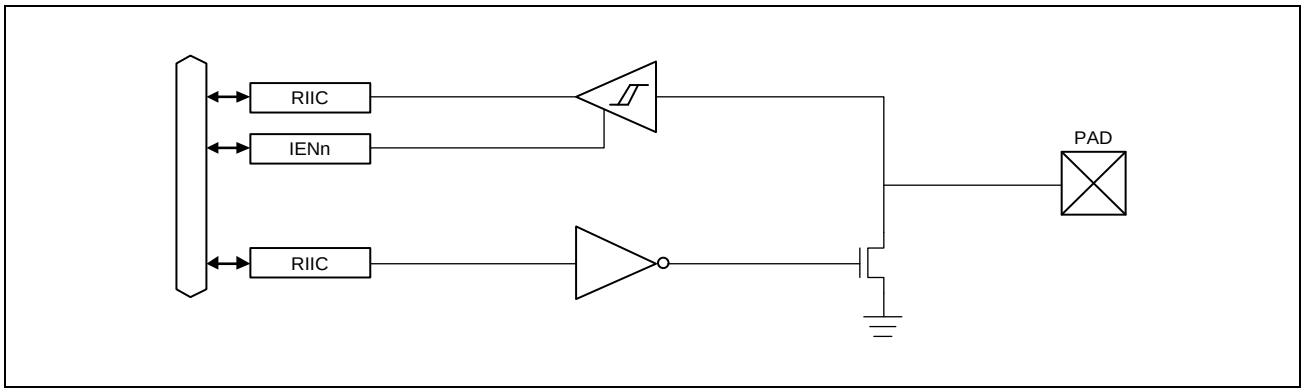



Figure 41.4 RIIC (RIIC0_SCL, RIIC0_SDA)

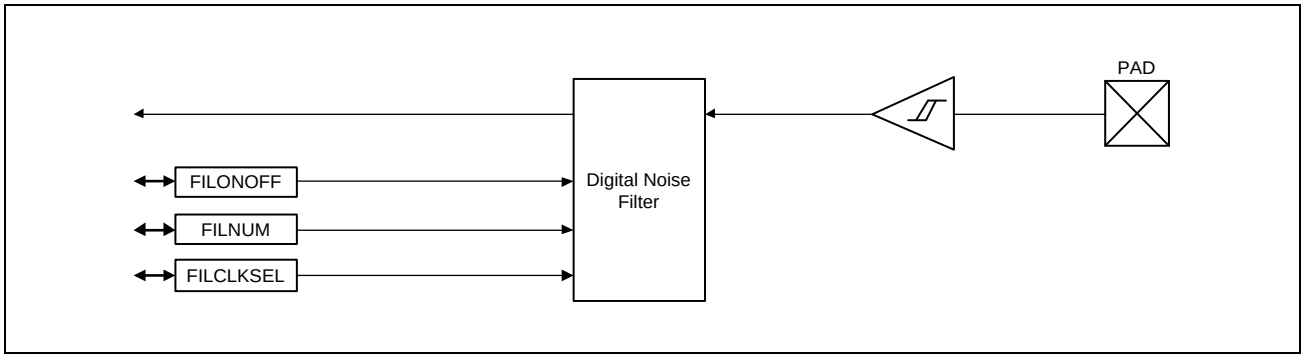



Figure 41.5 NMI

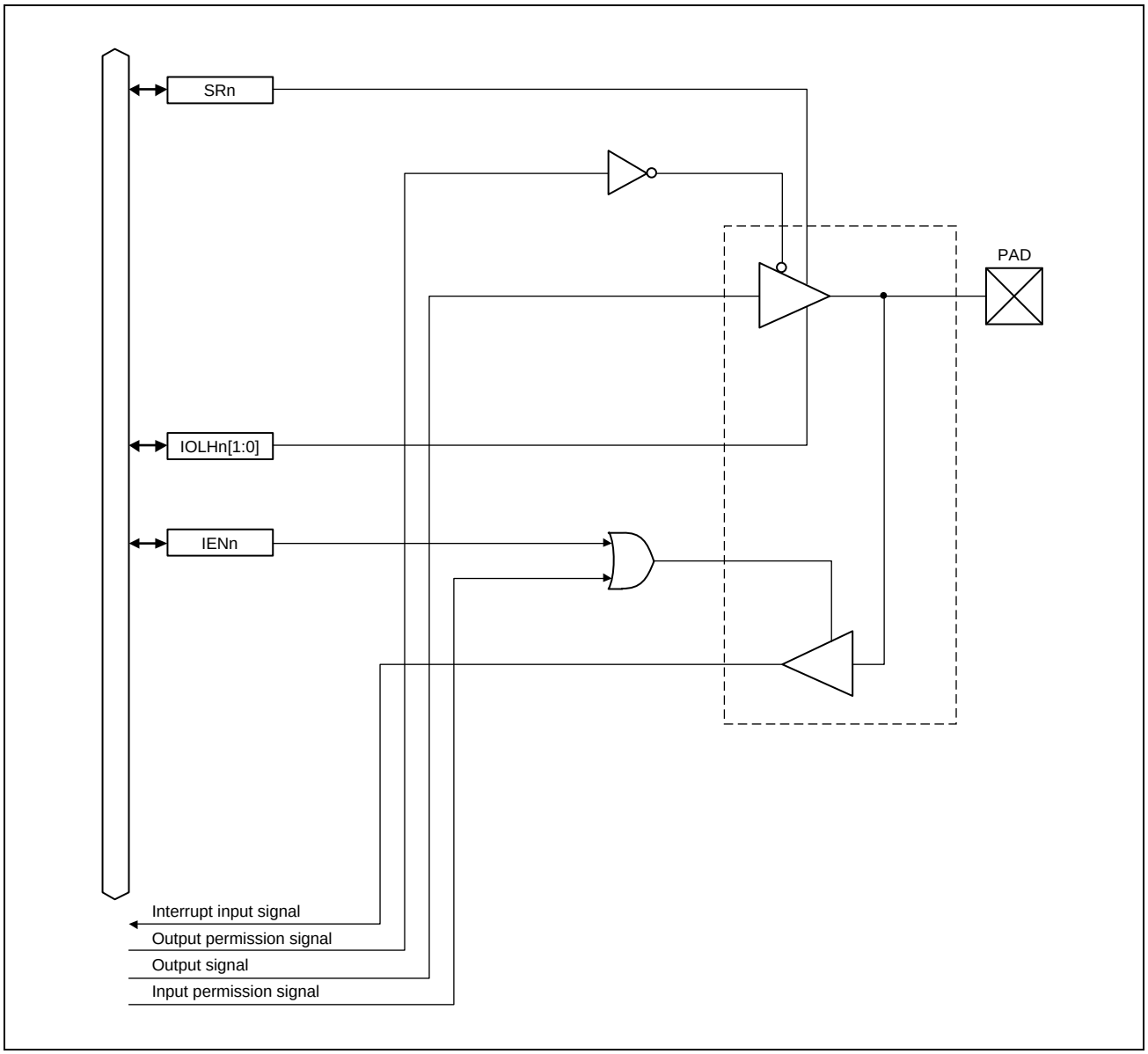



Figure 41.6 TMS/SWDIO

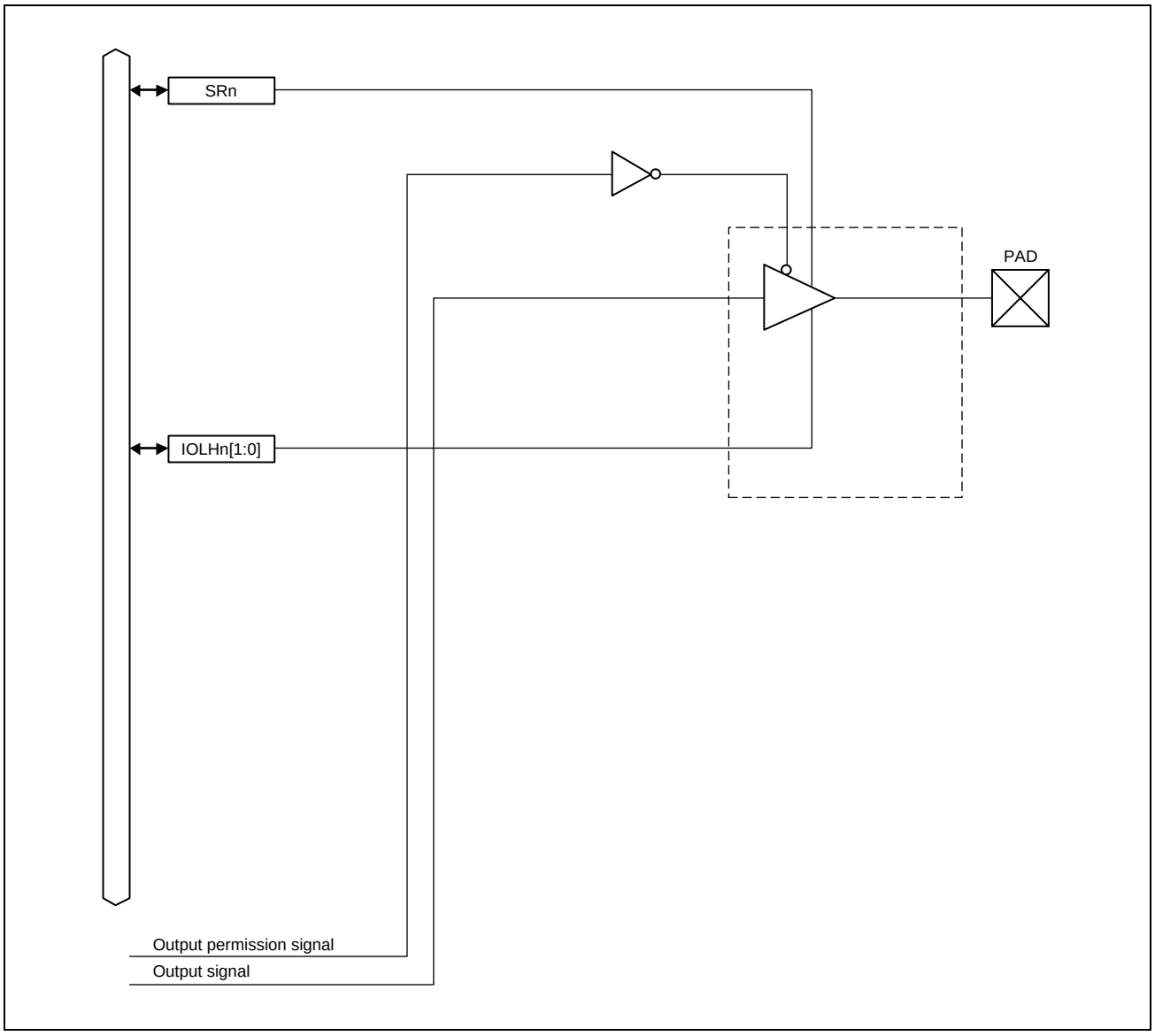



Figure 41.7 WDTOVF_PERROUT#

## 41.2 Register Configuration

The base address is shown below.

Base address: H'0_1103_0000 (Cortex-A55 Address Space)

The register addresses shown in the following chapters will be offset addresses from the above base address.

NOTE

Access to areas for which Offset address is not defined in the register map is prohibited.

Please align the addresses according to the access size.

### 41.2.1 Port Register (Pn)

| Register Name | Abbreviation | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|
| PORT REGISTER01 | P01 | R/W | H'00 | H'0001 | 8 |
| PORT REGISTER02 | P02 | R/W | H'00 | H'0002 | 8 |
| PORT REGISTER04 | P04 | R/W | H'00 | H'0004 | 8 |
| PORT REGISTER05 | P05 | R/W | H'00 | H'0005 | 8 |
| PORT REGISTER10 | P10 | R/W | H'00 | H'0010 | 8 |
| PORT REGISTER11 | P11 | R/W | H'00 | H'0011 | 8 |
| PORT REGISTER12 | P12 | R/W | H'00 | H'0012 | 8 |
| PORT REGISTER13 | P13 | R/W | H'00 | H'0013 | 8 |
| PORT REGISTER14 | P14 | R/W | H'00 | H'0014 | 8 |
| PORT REGISTER15 | P15 | R/W | H'00 | H'0015 | 8 |
| PORT REGISTER16 | P16 | R/W | H'00 | H'0016 | 8 |
| PORT REGISTER17 | P17 | R/W | H'00 | H'0017 | 8 |
| PORT REGISTER18 | P18 | R/W | H'00 | H'0018 | 8 |
| PORT REGISTER19 | P19 | R/W | H'00 | H'0019 | 8 |
| PORT REGISTER1A | P1A | R/W | H'00 | H'001A | 8 |
| PORT REGISTER1B | P1B | R/W | H'00 | H'001B | 8 |
| PORT REGISTER1C | P1C | R/W | H'00 | H'001C | 8 |

### 41.2.2 Port Mode Register (PMn)

| Register Name | Abbreviation | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|
| PORT MODE REGISTER01 | PM01 | R/W | H'0000 | H'0102 | 8/16 |
| PORT MODE REGISTER02 | PM02 | R/W | H'0000 | H'0104 | 8/16 |
| PORT MODE REGISTER04 | PM04 | R/W | H'0000 | H'0108 | 8/16 |
| PORT MODE REGISTER05 | PM05 | R/W | H'0000 | H'010A | 8/16 |
| PORT MODE REGISTER10 | PM10 | R/W | H'0000 | H'0120 | 8/16 |
| PORT MODE REGISTER11 | PM11 | R/W | H'0000 | H'0122 | 8/16 |
| PORT MODE REGISTER12 | PM12 | R/W | H'0000 | H'0124 | 8/16 |
| PORT MODE REGISTER13 | PM13 | R/W | H'0000 | H'0126 | 8/16 |
| PORT MODE REGISTER14 | PM14 | R/W | H'0000 | H'0128 | 8/16 |

| Register Name | Abbreviation | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|
| PORT MODE REGISTER15 | PM15 | R/W | H'0000 | H'012A | 8/16 |
| PORT MODE REGISTER16 | PM16 | R/W | H'0000 | H'012C | 8/16 |
| PORT MODE REGISTER17 | PM17 | R/W | H'0000 | H'012E | 8/16 |
| PORT MODE REGISTER18 | PM18 | R/W | H'0000 | H'0130 | 8/16 |
| PORT MODE REGISTER19 | PM19 | R/W | H'0000 | H'0132 | 8/16 |
| PORT MODE REGISTER1A | PM1A | R/W | H'0000 | H'0134 | 8/16 |
| PORT MODE REGISTER1B | PM1B | R/W | H'0000 | H'0136 | 8/16 |
| PORT MODE REGISTER1C | PM1C | R/W | H'0000 | H'0138 | 8/16 |

### 41.2.3 Port Mode Control Register (PMCn)

| Register Name | Abbreviation | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|
| PORT MODE CONTROL REGISTER01 | PMC01 | R/W | H'00 | H'0201 | 8 |
| PORT MODE CONTROL REGISTER02 | PMC02 | R/W | H'00 | H'0202 | 8 |
| PORT MODE CONTROL REGISTER04 | PMC04 | R/W | H'00 | H'0204 | 8 |
| PORT MODE CONTROL REGISTER05 | PMC05 | R/W | H'00 | H'0205 | 8 |
| PORT MODE CONTROL REGISTER10 | PMC10 | R/W | H'00 | H'0210 | 8 |
| PORT MODE CONTROL REGISTER11 | PMC11 | R/W | H'00 | H'0211 | 8 |
| PORT MODE CONTROL REGISTER12 | PMC12 | R/W | H'00 | H'0212 | 8 |
| PORT MODE CONTROL REGISTER13 | PMC13 | R/W | H'00 | H'0213 | 8 |
| PORT MODE CONTROL REGISTER14 | PMC14 | R/W | H'00 | H'0214 | 8 |
| PORT MODE CONTROL REGISTER15 | PMC15 | R/W | H'00 | H'0215 | 8 |
| PORT MODE CONTROL REGISTER16 | PMC16 | R/W | H'00 | H'0216 | 8 |
| PORT MODE CONTROL REGISTER17 | PMC17 | R/W | H'00 | H'0217 | 8 |
| PORT MODE CONTROL REGISTER18 | PMC18 | R/W | H'00 | H'0218 | 8 |
| PORT MODE CONTROL REGISTER19 | PMC19 | R/W | H'00 | H'0219 | 8 |
| PORT MODE CONTROL REGISTER1A | PMC1A | R/W | H'00 | H'021A | 8 |
| PORT MODE CONTROL REGISTER1B | PMC1B | R/W | H'00 | H'021B | 8 |
| PORT MODE CONTROL REGISTER1C | PMC1C | R/W | H'00 | H'021C | 8 |

### 41.2.4 Port Function Control Register (PFCm)

| Register Name | Abbreviation | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|
| PORT FUNCTION CONTROL REGISTER01 | PFC01 | R/W | H'0000 0000 | H'0404 | 8/16/32 |
| PORT FUNCTION CONTROL REGISTER02 | PFC02 | R/W | H'0000 0000 | H'0408 | 8/16/32 |
| PORT FUNCTION CONTROL REGISTER04 | PFC04 | R/W | H'0000 0000 | H'0410 | 8/16/32 |
| PORT FUNCTION CONTROL REGISTER05 | PFC05 | R/W | H'0000 0000 | H'0414 | 8/16/32 |
| PORT FUNCTION CONTROL REGISTER10 | PFC10 | R/W | H'0000 0000 | H'0440 | 8/16/32 |
| PORT FUNCTION CONTROL REGISTER11 | PFC11 | R/W | H'0000 0000 | H'0444 | 8/16/32 |
| PORT FUNCTION CONTROL REGISTER12 | PFC12 | R/W | H'0000 0000 | H'0448 | 8/16/32 |
| PORT FUNCTION CONTROL REGISTER13 | PFC13 | R/W | H'0000 0000 | H'044C | 8/16/32 |
| PORT FUNCTION CONTROL REGISTER14 | PFC14 | R/W | H'0000 0000 | H'0450 | 8/16/32 |
| PORT FUNCTION CONTROL REGISTER15 | PFC15 | R/W | H'0000 0000 | H'0454 | 8/16/32 |
| PORT FUNCTION CONTROL REGISTER16 | PFC16 | R/W | H'0000 0000 | H'0458 | 8/16/32 |
| PORT FUNCTION CONTROL REGISTER17 | PFC17 | R/W | H'0000 0000 | H'045C | 8/16/32 |

| Register Name | Abbreviation | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|
| PORT FUNCTION CONTROL REGISTER18 | PFC18 | R/W | H'0000 0000 | H'0460 | 8/16/32 |
| PORT FUNCTION CONTROL REGISTER19 | PFC19 | R/W | H'0000 0000 | H'0464 | 8/16/32 |
| PORT FUNCTION CONTROL REGISTER1A | PFC1A | R/W | H'0000 0000 | H'0468 | 8/16/32 |
| PORT FUNCTION CONTROL REGISTER1B | PFC1B | R/W | H'0000 0000 | H'046C | 8/16/32 |
| PORT FUNCTION CONTROL REGISTER1C | PFC1C | R/W | H'0000 0000 | H'0470 | 8/16/32 |

### 41.2.5 Port Input Register (PINn)

| Register Name | Abbreviation | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|
| PORT INPUT REGISTER01 | PIN01 | R | H'00 | H'0801 | 8 |
| PORT INPUT REGISTER02 | PIN02 | R | H'00 | H'0802 | 8 |
| PORT INPUT REGISTER04 | PIN04 | R | H'00 | H'0804 | 8 |
| PORT INPUT REGISTER05 | PIN05 | R | H'00 | H'0805 | 8 |
| PORT INPUT REGISTER10 | PIN10 | R | H'00 | H'0810 | 8 |
| PORT INPUT REGISTER11 | PIN11 | R | H'00 | H'0811 | 8 |
| PORT INPUT REGISTER12 | PIN12 | R | H'00 | H'0812 | 8 |
| PORT INPUT REGISTER13 | PIN13 | R | H'00 | H'0813 | 8 |
| PORT INPUT REGISTER14 | PIN14 | R | H'00 | H'0814 | 8 |
| PORT INPUT REGISTER15 | PIN15 | R | H'00 | H'0815 | 8 |
| PORT INPUT REGISTER16 | PIN16 | R | H'00 | H'0816 | 8 |
| PORT INPUT REGISTER17 | PIN17 | R | H'00 | H'0817 | 8 |
| PORT INPUT REGISTER18 | PIN18 | R | H'00 | H'0818 | 8 |
| PORT INPUT REGISTER19 | PIN19 | R | H'00 | H'0819 | 8 |
| PORT INPUT REGISTER1A | PIN1A | R | H'00 | H'081A | 8 |
| PORT INPUT REGISTER1B | PIN1B | R | H'00 | H'081B | 8 |
| PORT INPUT REGISTER1C | PIN1C | R | H'00 | H'081C | 8 |

### 41.2.6 Interrupt Enable Control Register (ISEL)

| Register Name | Abbreviation | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|
| INTERRUPT ENABLE CONTROL REGISTER01_L | ISEL01_L | R/W | H'0000 0000 | H'2C08 | 8/16/32 |
| INTERRUPT ENABLE CONTROL REGISTER01_H | ISEL01_H | R/W | H'0000 0000 | H'2C0C | 8/16/32 |
| INTERRUPT ENABLE CONTROL REGISTER02 | ISEL02 | R/W | H'0000 0000 | H'2C10 | 8/16/32 |
| INTERRUPT ENABLE CONTROL REGISTER04_L | ISEL04_L | R/W | H'0000 0000 | H'2C20 | 8/16/32 |
| INTERRUPT ENABLE CONTROL REGISTER04_H | ISEL04_H | R/W | H'0000 0000 | H'2C24 | 8/16/32 |
| INTERRUPT ENABLE CONTROL REGISTER05_L | ISEL05_L | R/W | H'0000 0000 | H'2C28 | 8/16/32 |
| INTERRUPT ENABLE CONTROL REGISTER05_H | ISEL05_H | R/W | H'0000 0000 | H'2C2C | 8/16/32 |
| INTERRUPT ENABLE CONTROL REGISTER10 | ISEL10 | R/W | H'0000 0000 | H'2C80 | 8/16/32 |
| INTERRUPT ENABLE CONTROL REGISTER11_L | ISEL11_L | R/W | H'0000 0000 | H'2C88 | 8/16/32 |
| INTERRUPT ENABLE CONTROL REGISTER11_H | ISEL11_H | R/W | H'0000 0000 | H'2C8C | 8/16/32 |
| INTERRUPT ENABLE CONTROL REGISTER12 | ISEL12 | R/W | H'0000 0000 | H'2C90 | 8/16/32 |
| INTERRUPT ENABLE CONTROL REGISTER13 | ISEL13 | R/W | H'0000 0000 | H'2C98 | 8/16/32 |
| INTERRUPT ENABLE CONTROL REGISTER14_L | ISEL14_L | R/W | H'0000 0000 | H'2CA0 | 8/16/32 |
| INTERRUPT ENABLE CONTROL REGISTER14_H | ISEL14_H | R/W | H'0000 0000 | H'2CA4 | 8/16/32 |
| INTERRUPT ENABLE CONTROL REGISTER15_L | ISEL15_L | R/W | H'0000 0000 | H'2CA8 | 8/16/32 |

| Register Name | Abbreviation | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|
| INTERRUPT ENABLE CONTROL REGISTER15_H | ISEL15_H | R/W | H'0000 0000 | H'2CAC | 8/16/32 |
| INTERRUPT ENABLE CONTROL REGISTER16_L | ISEL16_L | R/W | H'0000 0000 | H'2CB0 | 8/16/32 |
| INTERRUPT ENABLE CONTROL REGISTER16_H | ISEL16_H | R/W | H'0000 0000 | H'2CB4 | 8/16/32 |
| INTERRUPT ENABLE CONTROL REGISTER17 | ISEL17 | R/W | H'0000 0000 | H'2CB8 | 8/16/32 |
| INTERRUPT ENABLE CONTROL REGISTER18 | ISEL18 | R/W | H'0000 0000 | H'2CC0 | 8/16/32 |
| INTERRUPT ENABLE CONTROL REGISTER19_L | ISEL19_L | R/W | H'0000 0000 | H'2CC8 | 8/16/32 |
| INTERRUPT ENABLE CONTROL REGISTER19_H | ISEL19_H | R/W | H'0000 0000 | H'2CCC | 8/16/32 |
| INTERRUPT ENABLE CONTROL REGISTER1A | ISEL1A | R/W | H'0000 0000 | H'2CD0 | 8/16/32 |
| INTERRUPT ENABLE CONTROL REGISTER1B | ISEL1B | R/W | H'0000 0000 | H'2CD8 | 8/16/32 |
| INTERRUPT ENABLE CONTROL REGISTER1C | ISEL1C | R/W | H'0000 0000 | H'2CE0 | 8/16/32 |

### 41.2.7 Driving Ability Control Register (IOLH)

| Register Name | Abbreviation | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|
| INTERRUPT ENABLE CONTROL REGISTER01_L | IOLH01_L | R/W | *1 | H'1008 | 8/16/32 |
| INTERRUPT ENABLE CONTROL REGISTER01_H | IOLH01_H | R/W | *1 | H'100C | 8/16/32 |
| INTERRUPT ENABLE CONTROL REGISTER02 | IOLH02 | R/W | *1 | H'1010 | 8/16/32 |
| INTERRUPT ENABLE CONTROL REGISTER04_L | IOLH04_L | R/W | *1 | H'1020 | 8/16/32 |
| INTERRUPT ENABLE CONTROL REGISTER04_H | IOLH04_H | R/W | *1 | H'1024 | 8/16/32 |
| INTERRUPT ENABLE CONTROL REGISTER05_L | IOLH05_L | R/W | *1 | H'1028 | 8/16/32 |
| INTERRUPT ENABLE CONTROL REGISTER05_H | IOLH05_H | R/W | *1 | H'102C | 8/16/32 |
| INTERRUPT ENABLE CONTROL REGISTER06 | IOLH06 | R/W | *1 | H'1030 | 8/16/32 |
| INTERRUPT ENABLE CONTROL REGISTER10 | IOLH10 | R/W | *1 | H'1080 | 8/16/32 |
| INTERRUPT ENABLE CONTROL REGISTER11_L | IOLH11_L | R/W | *1 | H'1088 | 8/16/32 |
| INTERRUPT ENABLE CONTROL REGISTER11_H | IOLH11_H | R/W | *1 | H'108C | 8/16/32 |
| INTERRUPT ENABLE CONTROL REGISTER12 | IOLH12 | R/W | *1 | H'1090 | 8/16/32 |
| INTERRUPT ENABLE CONTROL REGISTER13 | IOLH13 | R/W | *1 | H'1098 | 8/16/32 |
| INTERRUPT ENABLE CONTROL REGISTER14_L | IOLH14_L | R/W | *1 | H'10A0 | 8/16/32 |
| INTERRUPT ENABLE CONTROL REGISTER14_H | IOLH14_H | R/W | *1 | H'10A4 | 8/16/32 |
| INTERRUPT ENABLE CONTROL REGISTER15_L | IOLH15_L | R/W | *1 | H'10A8 | 8/16/32 |
| INTERRUPT ENABLE CONTROL REGISTER15_H | IOLH15_H | R/W | *1 | H'10AC | 8/16/32 |
| INTERRUPT ENABLE CONTROL REGISTER16_L | IOLH16_L | R/W | *1 | H'10B0 | 8/16/32 |
| INTERRUPT ENABLE CONTROL REGISTER16_H | IOLH16_H | R/W | *1 | H'10B4 | 8/16/32 |
| INTERRUPT ENABLE CONTROL REGISTER17 | IOLH17 | R/W | *1 | H'10B8 | 8/16/32 |
| INTERRUPT ENABLE CONTROL REGISTER18 | IOLH18 | R/W | *1 | H'10C0 | 8/16/32 |
| INTERRUPT ENABLE CONTROL REGISTER19_L | IOLH19_L | R/W | *1 | H'10CC | 8/16/32 |
| INTERRUPT ENABLE CONTROL REGISTER19_H | IOLH19_H | R/W | *1 | H'10C8 | 8/16/32 |
| INTERRUPT ENABLE CONTROL REGISTER1A | IOLH1A | R/W | *1 | H'10D0 | 8/16/32 |
| INTERRUPT ENABLE CONTROL REGISTER1B | IOLH1B | R/W | *1 | H'10D8 | 8/16/32 |
| INTERRUPT ENABLE CONTROL REGISTER1C | IOLH1C | R/W | *1 | H'10E0 | 8/16/32 |

Note 1. For the initial value, refer to **Section 41.3.7, Driving Ability Control Register (IOLH)**.

### 41.2.8 Slew Rate Switching Register (SR)

| Register Name | Abbreviation | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|
| SLEW RATE SWITCHING REGISTER01_L | SR01_L | R/W | *1 | H'1408 | 8/16/32 |
| SLEW RATE SWITCHING REGISTER01_H | SR01_H | R/W | *1 | H'140C | 8/16/32 |
| SLEW RATE SWITCHING REGISTER02 | SR02 | R/W | *1 | H'1410 | 8/16/32 |
| SLEW RATE SWITCHING REGISTER04_L | SR04_L | R/W | *1 | H'1420 | 8/16/32 |
| SLEW RATE SWITCHING REGISTER04_H | SR04_H | R/W | *1 | H'1424 | 8/16/32 |
| SLEW RATE SWITCHING REGISTER05_L | SR05_L | R/W | *1 | H'1428 | 8/16/32 |
| SLEW RATE SWITCHING REGISTER05_H | SR05_H | R/W | *1 | H'142C | 8/16/32 |
| SLEW RATE SWITCHING REGISTER06 | SR06 | R/W | *1 | H'1430 | 8/16/32 |
| SLEW RATE SWITCHING REGISTER10 | SR10 | R/W | *1 | H'1480 | 8/16/32 |
| SLEW RATE SWITCHING REGISTER11_L | SR11_L | R/W | *1 | H'1488 | 8/16/32 |
| SLEW RATE SWITCHING REGISTER11_H | SR11_H | R/W | *1 | H'148C | 8/16/32 |
| SLEW RATE SWITCHING REGISTER12 | SR12 | R/W | *1 | H'1490 | 8/16/32 |
| SLEW RATE SWITCHING REGISTER13 | SR13 | R/W | *1 | H'1498 | 8/16/32 |
| SLEW RATE SWITCHING REGISTER14_L | SR14_L | R/W | *1 | H'14A0 | 8/16/32 |
| SLEW RATE SWITCHING REGISTER14_H | SR14_H | R/W | *1 | H'14A4 | 8/16/32 |
| SLEW RATE SWITCHING REGISTER15_L | SR15_L | R/W | *1 | H'14A8 | 8/16/32 |
| SLEW RATE SWITCHING REGISTER15_H | SR15_H | R/W | *1 | H'14AC | 8/16/32 |
| SLEW RATE SWITCHING REGISTER16_L | SR16_L | R/W | *1 | H'14B0 | 8/16/32 |
| SLEW RATE SWITCHING REGISTER16_H | SR16_H | R/W | *1 | H'14B4 | 8/16/32 |
| SLEW RATE SWITCHING REGISTER17 | SR17 | R/W | *1 | H'14B8 | 8/16/32 |
| SLEW RATE SWITCHING REGISTER18 | SR18 | R/W | *1 | H'14C0 | 8/16/32 |
| SLEW RATE SWITCHING REGISTER19_L | SR19_L | R/W | *1 | H'14C8 | 8/16/32 |
| SLEW RATE SWITCHING REGISTER19_H | SR19_H | R/W | *1 | H'14CC | 8/16/32 |
| SLEW RATE SWITCHING REGISTER1A | SR1A | R/W | *1 | H'14D0 | 8/16/32 |
| SLEW RATE SWITCHING REGISTER1B | SR1B | R/W | *1 | H'14D8 | 8/16/32 |
| SLEW RATE SWITCHING REGISTER1C | SR1C | R/W | *1 | H'14E0 | 8/16/32 |

Note 1. For the initial value, refer to **Section 41.3.8, Slew Rate Switching Register (SR)**.

### 41.2.9 Pull Up/Pull down Switching Register (PUPD)

| Register Name | Abbreviation | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|
| PULL UP/PULL DOWN SWITCHING REGISTER01 | PUPD01_L | R/W | H'0000 0000 | H'1C08 | 8/16/32 |
| PULL UP/PULL DOWN SWITCHING REGISTER01 | PUPD01_H | R/W | H'0000 0000 | H'1C0C | 8/16/32 |
| PULL UP/PULL DOWN SWITCHING REGISTER02 | PUPD02 | R/W | H'0000 0000 | H'1C10 | 8/16/32 |
| PULL UP/PULL DOWN SWITCHING REGISTER04 | PUPD04_L | R/W | H'0000 0000 | H'1C20 | 8/16/32 |
| PULL UP/PULL DOWN SWITCHING REGISTER04 | PUPD04_H | R/W | H'0000 0000 | H'1C24 | 8/16/32 |
| PULL UP/PULL DOWN SWITCHING REGISTER05 | PUPD05_L | R/W | H'0000 0000 | H'1C28 | 8/16/32 |
| PULL UP/PULL DOWN SWITCHING REGISTER05 | PUPD05_H | R/W | H'0000 0000 | H'1C2C | 8/16/32 |
| PULL UP/PULL DOWN SWITCHING REGISTER10 | PUPD10 | R/W | H'0000 0000 | H'1C80 | 8/16/32 |
| PULL UP/PULL DOWN SWITCHING REGISTER11 | PUPD11_L | R/W | H'0000 0000 | H'1C88 | 8/16/32 |
| PULL UP/PULL DOWN SWITCHING REGISTER11 | PUPD11_H | R/W | H'0000 0000 | H'1C8C | 8/16/32 |
| PULL UP/PULL DOWN SWITCHING REGISTER12 | PUPD12 | R/W | H'0000 0000 | H'1C90 | 8/16/32 |
| PULL UP/PULL DOWN SWITCHING REGISTER13 | PUPD13 | R/W | H'0000 0000 | H'1C98 | 8/16/32 |
| PULL UP/PULL DOWN SWITCHING REGISTER14 | PUPD14_L | R/W | H'0000 0000 | H'1CA0 | 8/16/32 |
| PULL UP/PULL DOWN SWITCHING REGISTER14 | PUPD14_H | R/W | H'0000 0000 | H'1CA4 | 8/16/32 |
| PULL UP/PULL DOWN SWITCHING REGISTER15 | PUPD15_L | R/W | H'0000 0000 | H'1CA8 | 8/16/32 |
| PULL UP/PULL DOWN SWITCHING REGISTER15 | PUPD15_H | R/W | H'0000 0000 | H'1CAC | 8/16/32 |
| PULL UP/PULL DOWN SWITCHING REGISTER16 | PUPD16_L | R/W | H'0000 0000 | H'1CB0 | 8/16/32 |
| PULL UP/PULL DOWN SWITCHING REGISTER16 | PUPD16_H | R/W | H'0000 0000 | H'1CB4 | 8/16/32 |
| PULL UP/PULL DOWN SWITCHING REGISTER17 | PUPD17 | R/W | H'0000 0000 | H'1CB8 | 8/16/32 |
| PULL UP/PULL DOWN SWITCHING REGISTER18 | PUPD18 | R/W | H'0000 0000 | H'1CC0 | 8/16/32 |
| PULL UP/PULL DOWN SWITCHING REGISTER19 | PUPD19_L | R/W | H'0000 0000 | H'1CC8 | 8/16/32 |
| PULL UP/PULL DOWN SWITCHING REGISTER19 | PUPD19_H | R/W | H'0000 0000 | H'1CCC | 8/16/32 |
| PULL UP/PULL DOWN SWITCHING REGISTER1A | PUPD1A | R/W | H'0000 0000 | H'1CD0 | 8/16/32 |
| PULL UP/PULL DOWN SWITCHING REGISTER1B | PUPD1B | R/W | H'0000 0000 | H'1CD8 | 8/16/32 |
| PULL UP/PULL DOWN SWITCHING REGISTER1C | PUPD1C | R/W | H'0000 0000 | H'1CE0 | 8/16/32 |

### 41.2.10 Write protected register (PWPR)

| Register Name | Abbreviation | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|
| WRITE PROTECTED REGISTER | PWPR | R/W | H'80 | H'3014 | 8 |

### 41.2.11 Digital Noise Filter Switching Register (FILONOFF)

| Register Name | Abbreviation | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|
| DIGITAL NOISE FILTER SWITCHING REGISTER00 | FILONOFF00 | R/W | H'0000 0000 | H'2000 | 8/16/32 |
| DIGITAL NOISE FILTER SWITCHING REGISTER01_L | FILONOFF01_L | R/W | H'0000 0000 | H'2008 | 8/16/32 |
| DIGITAL NOISE FILTER SWITCHING REGISTER01_H | FILONOFF01_H | R/W | H'0000 0000 | H'200C | 8/16/32 |
| DIGITAL NOISE FILTER SWITCHING REGISTER02 | FILONOFF02 | R/W | H'0000 0000 | H'2010 | 8/16/32 |
| DIGITAL NOISE FILTER SWITCHING REGISTER04_L | FILONOFF04_L | R/W | H'0000 0000 | H'2020 | 8/16/32 |
| DIGITAL NOISE FILTER SWITCHING REGISTER04_H | FILONOFF04_H | R/W | H'0000 0000 | H'2024 | 8/16/32 |
| DIGITAL NOISE FILTER SWITCHING REGISTER05_L | FILONOFF05_L | R/W | H'0000 0000 | H'2028 | 8/16/32 |
| DIGITAL NOISE FILTER SWITCHING REGISTER05_H | FILONOFF05_H | R/W | H'0000 0000 | H'202C | 8/16/32 |
| DIGITAL NOISE FILTER SWITCHING REGISTER10 | FILONOFF10 | R/W | H'0000 0000 | H'2080 | 8/16/32 |

| Register Name | Abbreviation | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|
| DIGITAL NOISE FILTER SWITCHING REGISTER11_L | FILONOFF11_L | R/W | H'0000 0000 | H'2088 | 8/16/32 |
| DIGITAL NOISE FILTER SWITCHING REGISTER11_H | FILONOFF11_H | R/W | H'0000 0000 | H'208C | 8/16/32 |
| DIGITAL NOISE FILTER SWITCHING REGISTER12 | FILONOFF12 | R/W | H'0000 0000 | H'2090 | 8/16/32 |
| DIGITAL NOISE FILTER SWITCHING REGISTER13 | FILONOFF13 | R/W | H'0000 0000 | H'2098 | 8/16/32 |
| DIGITAL NOISE FILTER SWITCHING REGISTER14_L | FILONOFF14_L | R/W | H'0000 0000 | H'20A0 | 8/16/32 |
| DIGITAL NOISE FILTER SWITCHING REGISTER14_H | FILONOFF14_H | R/W | H'0000 0000 | H'20A4 | 8/16/32 |
| DIGITAL NOISE FILTER SWITCHING REGISTER15_L | FILONOFF15_L | R/W | H'0000 0000 | H'20A8 | 8/16/32 |
| DIGITAL NOISE FILTER SWITCHING REGISTER15_H | FILONOFF15_H | R/W | H'0000 0000 | H'20AC | 8/16/32 |
| DIGITAL NOISE FILTER SWITCHING REGISTER16_L | FILONOFF16_L | R/W | H'0000 0000 | H'20B0 | 8/16/32 |
| DIGITAL NOISE FILTER SWITCHING REGISTER16_H | FILONOFF16_H | R/W | H'0000 0000 | H'20B4 | 8/16/32 |
| DIGITAL NOISE FILTER SWITCHING REGISTER17 | FILONOFF17 | R/W | H'0000 0000 | H'20B8 | 8/16/32 |
| DIGITAL NOISE FILTER SWITCHING REGISTER18 | FILONOFF18 | R/W | H'0000 0000 | H'20C0 | 8/16/32 |
| DIGITAL NOISE FILTER SWITCHING REGISTER19_L | FILONOFF19_L | R/W | H'0000 0000 | H'20C8 | 8/16/32 |
| DIGITAL NOISE FILTER SWITCHING REGISTER19_H | FILONOFF19_H | R/W | H'0000 0000 | H'20CC | 8/16/32 |
| DIGITAL NOISE FILTER SWITCHING REGISTER1A | FILONOFF1A | R/W | H'0000 0000 | H'20D0 | 8/16/32 |
| DIGITAL NOISE FILTER SWITCHING REGISTER1B | FILONOFF1B | R/W | H'0000 0000 | H'20D8 | 8/16/32 |
| DIGITAL NOISE FILTER SWITCHING REGISTER1C | FILONOFF1C | R/W | H'0000 0000 | H'20E0 | 8/16/32 |

### 41.2.12 Digital Noise Filter Number Register (FILNUM)

| Register Name | Abbreviation | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|
| DIGITAL NOISE FILTER NUMBER REGISTER00 | FILNUM00 | R/W | H'0000 0000 | H'2400 | 8/16/32 |
| DIGITAL NOISE FILTER NUMBER REGISTER01_L | FILNUM01_L | R/W | H'0000 0000 | H'2408 | 8/16/32 |
| DIGITAL NOISE FILTER NUMBER REGISTER01_H | FILNUM01_H | R/W | H'0000 0000 | H'240C | 8/16/32 |
| DIGITAL NOISE FILTER NUMBER REGISTER02 | FILNUM02 | R/W | H'0000 0000 | H'2410 | 8/16/32 |
| DIGITAL NOISE FILTER NUMBER REGISTER04_L | FILNUM04_L | R/W | H'0000 0000 | H'2420 | 8/16/32 |
| DIGITAL NOISE FILTER NUMBER REGISTER04_H | FILNUM04_H | R/W | H'0000 0000 | H'2424 | 8/16/32 |
| DIGITAL NOISE FILTER NUMBER REGISTER05_L | FILNUM05_L | R/W | H'0000 0000 | H'2428 | 8/16/32 |
| DIGITAL NOISE FILTER NUMBER REGISTER05_H | FILNUM05_H | R/W | H'0000 0000 | H'242C | 8/16/32 |
| DIGITAL NOISE FILTER NUMBER REGISTER10 | FILNUM10 | R/W | H'0000 0000 | H'2480 | 8/16/32 |
| DIGITAL NOISE FILTER NUMBER REGISTER11_L | FILNUM11_L | R/W | H'0000 0000 | H'2488 | 8/16/32 |
| DIGITAL NOISE FILTER NUMBER REGISTER11_H | FILNUM11_H | R/W | H'0000 0000 | H'248C | 8/16/32 |
| DIGITAL NOISE FILTER NUMBER REGISTER12 | FILNUM12 | R/W | H'0000 0000 | H'2490 | 8/16/32 |
| DIGITAL NOISE FILTER NUMBER REGISTER13 | FILNUM13 | R/W | H'0000 0000 | H'2498 | 8/16/32 |
| DIGITAL NOISE FILTER NUMBER REGISTER14_L | FILNUM14_L | R/W | H'0000 0000 | H'24A0 | 8/16/32 |
| DIGITAL NOISE FILTER NUMBER REGISTER14_H | FILNUM14_H | R/W | H'0000 0000 | H'24A4 | 8/16/32 |
| DIGITAL NOISE FILTER NUMBER REGISTER15_L | FILNUM15_L | R/W | H'0000 0000 | H'24A8 | 8/16/32 |
| DIGITAL NOISE FILTER NUMBER REGISTER15_H | FILNUM15_H | R/W | H'0000 0000 | H'24AC | 8/16/32 |
| DIGITAL NOISE FILTER NUMBER REGISTER16_L | FILNUM16_L | R/W | H'0000 0000 | H'24B0 | 8/16/32 |
| DIGITAL NOISE FILTER NUMBER REGISTER16_H | FILNUM16_H | R/W | H'0000 0000 | H'24B4 | 8/16/32 |
| DIGITAL NOISE FILTER NUMBER REGISTER17 | FILNUM17 | R/W | H'0000 0000 | H'24B8 | 8/16/32 |
| DIGITAL NOISE FILTER NUMBER REGISTER18 | FILNUM18 | R/W | H'0000 0000 | H'24C0 | 8/16/32 |
| DIGITAL NOISE FILTER NUMBER REGISTER19_L | FILNUM19_L | R/W | H'0000 0000 | H'24C8 | 8/16/32 |
| DIGITAL NOISE FILTER NUMBER REGISTER19_H | FILNUM19_H | R/W | H'0000 0000 | H'24CC | 8/16/32 |
| DIGITAL NOISE FILTER NUMBER REGISTER1A | FILNUM1A | R/W | H'0000 0000 | H'24D0 | 8/16/32 |
| DIGITAL NOISE FILTER NUMBER REGISTER1B | FILNUM1B | R/W | H'0000 0000 | H'24D8 | 8/16/32 |

| Register Name | Abbreviation | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|
| DIGITAL NOISE FILTER NUMBER REGISTER1C | FILNUM1C | R/W | H'0000 0000 | H'24E0 | 8/16/32 |

## 41.2.13 Digital Noise Filter Clock Selection Register (FILCLKSEL)

| Register Name | Abbreviation | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|
| DIGITAL NOISE FILTER CLOCK SELECTION REGISTER00 | FILCLKSEL00 | R/W | H'0000 0000 | H'2800 | 8/16/32 |
| DIGITAL NOISE FILTER CLOCK SELECTION REGISTER01_L | FILCLKSEL01_L | R/W | H'0000 0000 | H'2808 | 8/16/32 |
| DIGITAL NOISE FILTER CLOCK SELECTION REGISTER01_H | FILCLKSEL01_H | R/W | H'0000 0000 | H'280C | 8/16/32 |
| DIGITAL NOISE FILTER CLOCK SELECTION REGISTER02 | FILCLKSEL02 | R/W | H'0000 0000 | H'2810 | 8/16/32 |
| DIGITAL NOISE FILTER CLOCK SELECTION REGISTER04_L | FILCLKSEL04_L | R/W | H'0000 0000 | H'2820 | 8/16/32 |
| DIGITAL NOISE FILTER CLOCK SELECTION REGISTER04_H | FILCLKSEL04_H | R/W | H'0000 0000 | H'2824 | 8/16/32 |
| DIGITAL NOISE FILTER CLOCK SELECTION REGISTER05_L | FILCLKSEL05_L | R/W | H'0000 0000 | H'2828 | 8/16/32 |
| DIGITAL NOISE FILTER CLOCK SELECTION REGISTER05_H | FILCLKSEL05_H | R/W | H'0000 0000 | H'282C | 8/16/32 |
| DIGITAL NOISE FILTER CLOCK SELECTION REGISTER10 | FILCLKSEL10 | R/W | H'0000 0000 | H'2880 | 8/16/32 |
| DIGITAL NOISE FILTER CLOCK SELECTION REGISTER11_L | FILCLKSEL11_L | R/W | H'0000 0000 | H'2888 | 8/16/32 |
| DIGITAL NOISE FILTER CLOCK SELECTION REGISTER11_H | FILCLKSEL11_H | R/W | H'0000 0000 | H'288C | 8/16/32 |
| DIGITAL NOISE FILTER CLOCK SELECTION REGISTER12 | FILCLKSEL12 | R/W | H'0000 0000 | H'2890 | 8/16/32 |
| DIGITAL NOISE FILTER CLOCK SELECTION REGISTER13 | FILCLKSEL13 | R/W | H'0000 0000 | H'2898 | 8/16/32 |
| DIGITAL NOISE FILTER CLOCK SELECTION REGISTER14_L | FILCLKSEL14_L | R/W | H'0000 0000 | H'28A0 | 8/16/32 |
| DIGITAL NOISE FILTER CLOCK SELECTION REGISTER14_H | FILCLKSEL14_H | R/W | H'0000 0000 | H'28A4 | 8/16/32 |
| DIGITAL NOISE FILTER CLOCK SELECTION REGISTER15_L | FILCLKSEL15_L | R/W | H'0000 0000 | H'28A8 | 8/16/32 |
| DIGITAL NOISE FILTER CLOCK SELECTION REGISTER15_H | FILCLKSEL15_H | R/W | H'0000 0000 | H'28AC | 8/16/32 |
| DIGITAL NOISE FILTER CLOCK SELECTION REGISTER16_L | FILCLKSEL16_L | R/W | H'0000 0000 | H'28B0 | 8/16/32 |
| DIGITAL NOISE FILTER CLOCK SELECTION REGISTER16_H | FILCLKSEL16_H | R/W | H'0000 0000 | H'28B4 | 8/16/32 |
| DIGITAL NOISE FILTER CLOCK SELECTION REGISTER17 | FILCLKSEL17 | R/W | H'0000 0000 | H'28B8 | 8/16/32 |
| DIGITAL NOISE FILTER CLOCK SELECTION REGISTER18 | FILCLKSEL18 | R/W | H'0000 0000 | H'28C0 | 8/16/32 |
| DIGITAL NOISE FILTER CLOCK SELECTION REGISTER19_L | FILCLKSEL19_L | R/W | H'0000 0000 | H'28C8 | 8/16/32 |
| DIGITAL NOISE FILTER CLOCK SELECTION REGISTER19_H | FILCLKSEL19_H | R/W | H'0000 0000 | H'28CC | 8/16/32 |
| DIGITAL NOISE FILTER CLOCK SELECTION REGISTER1A | FILCLKSEL1A | R/W | H'0000 0000 | H'28D0 | 8/16/32 |
| DIGITAL NOISE FILTER CLOCK SELECTION REGISTER1B | FILCLKSEL1B | R/W | H'0000 0000 | H'28D8 | 8/16/32 |
| DIGITAL NOISE FILTER CLOCK SELECTION REGISTER1C | FILCLKSEL1C | R/W | H'0000 0000 | H'28E0 | 8/16/32 |

## 41.2.14 SD ch0 IO Voltage Mode Control Register (SD_ch0)

| Register Name | Abbreviation | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|
| SD CH0 IO VOLTAGE MODE CONTROL REGISTER | SD_ch0 | R/W | *1 | H3000 | 8/16/32 |

Note 1. For the initial value, refer to **Section 41.3.14, SD ch0 IO Voltage Mode Control Register (SD_ch0)**.

### 41.2.15 QSPI IO Voltage Mode Control Register (QSPI)

| Register Name | Abbreviation | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|
| QSPI IO VOLTAGE MODE CONTROL REGISTER | QSPI | R/W | *1 | H3008 | 8/16/32 |

Note 1. For the initial value, refer to **Section 41.3.15, QSPI IO Voltage Mode Control Register (QSPI)**.

### 41.2.16 Input Enable Control Register (IEN)

| Register Name | Abbreviation | R/W | Initial Value | Address | Access Size |
|---|---|---|---|---|---|
| INPUT ENABLE CONTROL REGISTER02 | IEN02 | R/W | H'0000 0000 | H'1810 | 8/16/32 |
| INPUT ENABLE CONTROL REGISTER04_L | IEN04_L | R/W | H'0000 0000 | H'1820 | 8/16/32 |
| INPUT ENABLE CONTROL REGISTER04_H | IEN04_H | R/W | H'0000 0000 | H'1824 | 8/16/32 |
| INPUT ENABLE CONTROL REGISTER07 | IEN07 | R/W | H'0000 0000 | H'1838 | 8/16/32 |

# 41.3 Register Descriptions

## 41.3.1 Port Register (Pn)

The Pn register sets the GPIO output value.

For the offset address, refer to **Section 41.2.1, Port Register (Pn)**.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | P7 | P6 | P5 | P4 | P3 | P2 | P1 | P0 |
| 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 | P7 | 0b | R/W | These bits set the value of the GPIO pin. (P7 to P0)<br>0b: Low is output.<br>1b: High is output.<br>Please refer to the table below for the correspondence of each terminal. |
| 6 | P6 | 0b | R/W | |
| 5 | P5 | 0b | R/W | |
| 4 | P4 | 0b | R/W | |
| 3 | P3 | 0b | R/W | |
| 2 | P2 | 0b | R/W | |
| 1 | P1 | 0b | R/W | |
| 0 | P0 | 0b | R/W | |

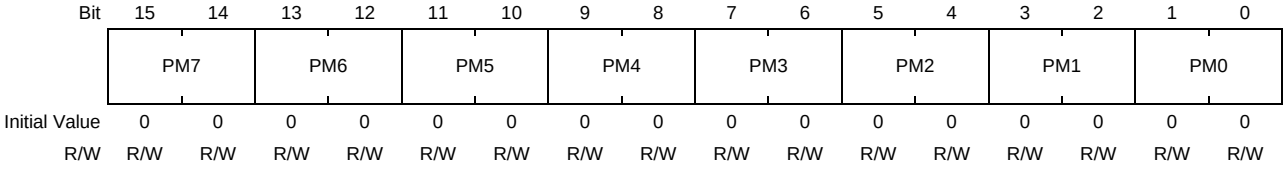
Table 41.1 Correspondence between register and each terminal

| Bit Name | bit5 | bit4 | bit3 | bit2 | bit1 | bit0 |
|---|---|---|---|---|---|---|
| P01 | P20_5 | P20_4 | P20_3 | P20_2 | P20_1 | P20_0 |
| P02 | - | - | P21_2 | P21_1 | P21_0 | - |
| P04 | P22_4 | P22_3 | P22_2 | P22_1 | P22_0 | - |
| P05 | - | P23_1 | P23_0 | - | - | - |
| P10 | - | - | P0_3 | P0_2 | P0_1 | P0_0 |
| P11 | - | P1_4 | P1_3 | P1_2 | P1_1 | P1_0 |
| P12 | - | - | P2_3 | P2_2 | P2_1 | P2_0 |
| P13 | - | - | P3_3 | P3_2 | P3_1 | P3_0 |
| P14 | P4_5 | P4_4 | P4_3 | P4_2 | P4_1 | P4_0 |
| P15 | - | P5_4 | P5_3 | P5_2 | P5_1 | P5_0 |
| P16 | - | P6_4 | P6_3 | P6_2 | P6_1 | P6_0 |
| P17 | - | - | - | - | P7_1 | P7_0 |
| P18 | - | - | P8_3 | P8_2 | P8_1 | P8_0 |
| P19 | - | P9_4 | P9_3 | P9_2 | P9_1 | P9_0 |
| P1A | - | - | P10_3 | P10_2 | P10_1 | P10_0 |
| P1B | - | - | P11_3 | P11_2 | P11_1 | P11_0 |
| P1C | - | - | - | - | P12_1 | P12_0 |

### 41.3.2 Port Mode Register (PMn)

The PMn register sets input/output of GPIO.

For the offset address, refer to **Section 41.2.2, Port Mode Register (PMn)**.




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 15, 14 | PM7 | 00b | R/W | These bits set input/output of the GPIO pin. (PM7 to PM0)<br>00b: Hi-Z (Initial Value)<br>01b: Input Mode<br>10b: Output Mode (Input Disable)<br>11b: Output Mode (Input Enable)<br>Please refer to the table below for the correspondence of each terminal. |
| 13, 12 | PM6 | 00b | R/W | |
| 11, 10 | PM5 | 00b | R/W | |
| 9, 8 | PM4 | 00b | R/W | |
| 7, 6 | PM3 | 00b | R/W | |
| 5, 4 | PM2 | 00b | R/W | |
| 3, 2 | PM1 | 00b | R/W | |
| 1, 0 | PM0 | 00b | R/W | |

Table 41.2 Correspondence between register and each terminal

| Bit Name | bit11-10 | bit9-8 | bit7-6 | bit5-4 | bit3-2 | bit1-0 |
|---|---|---|---|---|---|---|
| PM01 | P20_5 | P20_4 | P20_3 | P20_2 | P20_1 | P20_0 |
| PM02 | - | - | P21_2 | P21_1 | P21_0 | - |
| PM04 | P22_4 | P22_3 | P22_2 | P22_1 | P22_0 | - |
| PM05 | - | P23_1 | P23_0 | - | - | - |
| PM10 | - | - | P0_3 | P0_2 | P0_1 | P0_0 |
| PM11 | - | P1_4 | P1_3 | P1_2 | P1_1 | P1_0 |
| PM12 | - | - | P2_3 | P2_2 | P2_1 | P2_0 |
| PM13 | - | - | P3_3 | P3_2 | P3_1 | P3_0 |
| PM14 | P4_5 | P4_4 | P4_3 | P4_2 | P4_1 | P4_0 |
| PM15 | - | P5_4 | P5_3 | P5_2 | P5_1 | P5_0 |
| PM16 | - | P6_4 | P6_3 | P6_2 | P6_1 | P6_0 |
| PM17 | - | - | - | - | P7_1 | P7_0 |
| PM18 | - | - | P8_3 | P8_2 | P8_1 | P8_0 |
| PM19 | - | P9_4 | P9_3 | P9_2 | P9_1 | P9_0 |
| PM1A | - | - | P10_3 | P10_2 | P10_1 | P10_0 |
| PM1B | - | - | P11_3 | P11_2 | P11_1 | P11_0 |
| PM1C | - | - | - | - | P12_1 | P12_0 |

### 41.3.3 Port Mode Control Register (PMCn)

The PMCn register switches the mode for the multiplexed pins of GPIO.

For the offset address, refer to **Section 41.2.3, Port Mode Control Register (PMCn)**.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | PMC7 | PMC6 | PMC5 | PMC4 | PMC3 | PMC2 | PMC1 | PMC0 |
| Initial Value | 0 *1 | 0 *1 | 0 *1 | 0 *1 | 0 *1 | 0 *1 | 0 *1 | 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 | PMC7 | 0b *1 | R/W | These bits switch the mode for the multiplexed pin of the GPIO pin. (PMC7 to PMC0)<br>0b: Port Mode (GPIO)<br>1b: Peripheral Function Mode (Peripheral Function)<br>Please refer to the table below for the correspondence of each terminal. |
| 6 | PMC6 | 0b *1 | R/W | |
| 5 | PMC5 | 0b *1 | R/W | |
| 4 | PMC4 | 0b *1 | R/W | |
| 3 | PMC3 | 0b *1 | R/W | |
| 2 | PMC2 | 0b *1 | R/W | |
| 1 | PMC1 | 0b *1 | R/W | |
| 0 | PMC0 | 0b *1 | R/W | |

Note.1: P20_0 to P20_5, P21_0 to P21_2, P22_0 to P22_4 and P23_0 to P23_1 have an initial value of 1 and all other ports have an initial value of 0.

Table 41.3 Correspondence between register and each terminal

| Bit Name | bit5 | bit4 | bit3 | bit2 | bit1 | bit0 |
|---|---|---|---|---|---|---|
| PMC01 | P20_5 | P20_4 | P20_3 | P20_2 | P20_1 | P20_0 |
| PMC02 | - | - | P21_2 | P21_1 | P21_0 | - |
| PMC04 | P22_4 | P22_3 | P22_2 | P22_1 | P22_0 | - |
| PMC05 | - | P23_1 | P23_0 | - | - | - |
| PMC10 | - | - | P0_3 | P0_2 | P0_1 | P0_0 |
| PMC11 | - | P1_4 | P1_3 | P1_2 | P1_1 | P1_0 |
| PMC12 | - | - | P2_3 | P2_2 | P2_1 | P2_0 |
| PMC13 | - | - | P3_3 | P3_2 | P3_1 | P3_0 |
| PMC14 | P4_5 | P4_4 | P4_3 | P4_2 | P4_1 | P4_0 |
| PMC15 | - | P5_4 | P5_3 | P5_2 | P5_1 | P5_0 |
| PMC16 | - | P6_4 | P6_3 | P6_2 | P6_1 | P6_0 |
| PMC17 | - | - | - | - | P7_1 | P7_0 |
| PMC18 | - | - | P8_3 | P8_2 | P8_1 | P8_0 |
| PMC19 | - | P9_4 | P9_3 | P9_2 | P9_1 | P9_0 |
| PMC1A | - | - | P10_3 | P10_2 | P10_1 | P10_0 |
| PMC1B | - | - | P11_3 | P11_2 | P11_1 | P11_0 |
| PMC1C | - | - | - | - | P12_1 | P12_0 |

- PMC register of P21_0, P21_1 and P21_2 are switched as follows, depending on the BSCANP and SWD_JTAG settings.

| BSCANP | SWD_JTAG | PMC register |
|---|---|---|
| 0 | 1 | Switchable |
| Other than the above | | 1'b1 (fixed) |

### 41.3.4 Port Function Control Register (PFCm)

The PFCm register sets multiplexed functions.

This register can be write-protected by the PWPR register.

For the offset address, refer to **Section 41.2.4, Port Function Control Register (PFCm)**.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | PFC7 | | | — | PFC6 | | | — | PFC5 | | | — | PFC4 | | |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | PFC3 | | | — | PFC2 | | | — | PFC1 | | | — | PFC0 | | |
| 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, 27, 23, 19, 15, 11, 7, 3 | — | All 0 | R/W | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value.<br>Operation is not guaranteed if a value other than the initial value is written. |
| 30 to 28 | PFC7 | 000b | R/W | These bits set the multiplexed function of the GPIO pin. (PFC7 to PFC0)<br>000b: Function1 (Initial value)<br>001b: Function2<br>010b: Function3<br>011b: Function4<br>100b: Function5<br>101b: Function6<br>110b: Function7<br>111b: Function8<br>Please refer to the table below for the correspondence of each terminal. |
| 26 to 24 | PFC6 | 000b | R/W | |
| 22 to 20 | PFC5 | 000b | R/W | |
| 18 to 16 | PFC4 | 000b | R/W | |
| 14 to 12 | PFC3 | 000b | R/W | |
| 10 to 8 | PFC2 | 000b | R/W | |
| 6 to 4 | PFC1 | 000b | R/W | |
| 2 to 0 | PFC0 | 000b | R/W | |

**Note:** Please refer to “Pin function list” for the function of Function (1 to 8).

Table 41.4 Correspondence between register and each terminal (1/2)

| Bit Name | bit22-20 | bit18-16 | bit14-12 | bit10-8 | bit6-4 | bit2-0 |
|---|---|---|---|---|---|---|
| PFC01 | P20_5 | P20_4 | P20_3 | P20_2 | P20_1 | P20_0 |
| PFC02 | - | - | P21_2 | P21_1 | P21_0 | - |
| PFC04 | P22_4 | P22_3 | P22_2 | P22_1 | P22_0 | - |
| PFC05 | - | P23_1 | P23_0 | - | - | - |
| PFC10 | - | - | P0_3 | P0_2 | P0_1 | P0_0 |
| PFC11 | - | P1_4 | P1_3 | P1_2 | P1_1 | P1_0 |
| PFC12 | - | - | P2_3 | P2_2 | P2_1 | P2_0 |
| PFC13 | - | - | P3_3 | P3_2 | P3_1 | P3_0 |
| PFC14 | P4_5 | P4_4 | P4_3 | P4_2 | P4_1 | P4_0 |
| PFC15 | - | P5_4 | P5_3 | P5_2 | P5_1 | P5_0 |
| PFC16 | - | P6_4 | P6_3 | P6_2 | P6_1 | P6_0 |
| PFC17 | - | - | - | - | P7_1 | P7_0 |
| PFC18 | - | - | P8_3 | P8_2 | P8_1 | P8_0 |

Table 41.4 Correspondence between register and each terminal (2/2)

| Bit Name | bit22-20 | bit18-16 | bit14-12 | bit10-8 | bit6-4 | bit2-0 |
|---|---|---|---|---|---|---|
| PFC19 | - | P9_4 | P9_3 | P9_2 | P9_1 | P9_0 |
| PFC1A | - | - | P10_3 | P10_2 | P10_1 | P10_0 |
| PFC1B | - | - | P11_3 | P11_2 | P11_1 | P11_0 |
| PFC1C | - | - | - | - | P12_1 | P12_0 |

- PFC register of P21_0, P21_1 and P21_2 are switched as follows, depending on the BSCANP and SWD_JTAG settings.

| BSCANP | SWD_JTAG | PFC register |
|---|---|---|
| 0 | 1 | Switchable |
| Other than the above | | 3'b000 (fixed) |

### 41.3.5 Port Input Register (PINn)

The PINn register is a read-only register to monitor input values of input pins.

For the offset address, refer to **Section 41.2.5, Port Input Register (PINn)**.

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | PIN7 | PIN6 | PIN5 | PIN4 | PIN3 | PIN2 | PIN1 | PIN0 |
| Initial Value | x | x | x | x | x | x | x | x |
| R/W | R | R | R | R | R | R | R | R |

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 7 | PIN7 | x | R | Input value of the input pin (PIN7 to PIN0)<br>Please refer to the table below for the correspondence of each terminal. |
| 6 | PIN6 | x | R | |
| 5 | PIN5 | x | R | |
| 4 | PIN4 | x | R | |
| 3 | PIN3 | x | R | |
| 2 | PIN2 | x | R | |
| 1 | PIN1 | x | R | |
| 0 | PIN0 | x | R | |

Table 41.5 Correspondence between register and each terminal

| Bit Name | bit5 | bit4 | bit3 | bit2 | bit1 | bit0 |
|---|---|---|---|---|---|---|
| PIN01 | P20_5 | P20_4 | P20_3 | P20_2 | P20_1 | P20_0 |
| PIN02 | - | - | P21_2 | P21_1 | P21_0 | - |
| PIN04 | P22_4 | P22_3 | P22_2 | P22_1 | P22_0 | - |
| PIN05 | - | P23_1 | P23_0 | - | - | - |
| PIN10 | - | - | P0_3 | P0_2 | P0_1 | P0_0 |
| PIN11 | - | P1_4 | P1_3 | P1_2 | P1_1 | P1_0 |
| PIN12 | - | - | P2_3 | P2_2 | P2_1 | P2_0 |
| PIN13 | - | - | P3_3 | P3_2 | P3_1 | P3_0 |
| PIN14 | P4_5 | P4_4 | P4_3 | P4_2 | P4_1 | P4_0 |
| PIN15 | - | P5_4 | P5_3 | P5_2 | P5_1 | P5_0 |
| PIN16 | - | P6_4 | P6_3 | P6_2 | P6_1 | P6_0 |
| PIN17 | - | - | - | - | P7_1 | P7_0 |
| PIN18 | - | - | P8_3 | P8_2 | P8_1 | P8_0 |
| PIN19 | - | P9_4 | P9_3 | P9_2 | P9_1 | P9_0 |
| PIN1A | - | - | P10_3 | P10_2 | P10_1 | P10_0 |
| PIN1B | - | - | P11_3 | P11_2 | P11_1 | P11_0 |
| PIN1C | - | - | - | - | P12_1 | P12_0 |

### 41.3.6 Interrupt Enable Control Register (ISEL)

The ISEL register controls whether to enable or disable interrupts.

The pin set as the GPIO input port can be used as an external interrupt input. Controls whether this feature is enabled or disabled. When enabled, this pin can be used as an interrupt by setting the interrupt controller.

For the offset address, refer to **Section 41.2.6, Interrupt Enable Control Register (ISEL)**.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | ISEL3 | — | — | — | — | — | — | — | ISEL2 |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | ISEL1 | — | — | — | — | — | — | — | ISEL0 |
| 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 25,<br>23 to 17,<br>15 to 9,<br>7 to 1 | — | All 0 | R/W | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value.<br>Operation is not guaranteed if a value other than the initial value is written. |
| 24 | ISEL3 | 0b | R/W | The ISEL register controls whether to enable or disable interrupts.<br>0b: Disabled (Initial value)<br>1b: Enabled<br>Please refer to the table below for the correspondence of each terminal. |
| 16 | ISEL2 | 0b | R/W | |
| 8 | ISEL1 | 0b | R/W | |
| 0 | ISEL0 | 0b | R/W | |

**Note:** Before setting this register, disable interrupt detection in the Interrupt Controller.
If PMCn bit = 1, ISEL bit = 1 is prohibited.

Table 41.6 Correspondence between register and each terminal (1/2)

| Bit Name | bit24 | bit16 | bit8 | bit0 |
|---|---|---|---|---|
| ISEL01_L | P20_3 | P20_2 | P20_1 | P20_0 |
| ISEL01_H | - | - | P20_5 | P20_4 |
| ISEL02 | P21_2 | P21_1 | P21_0 | - |
| ISEL04_L | P22_2 | P22_1 | P22_0 | - |
| ISEL04_H | - | - | P22_4 | P22_3 |
| ISEL05_L | P23_0 | - | - | - |
| ISEL05_H | - | - | - | P23_1 |
| ISEL10 | P0_3 | P0_2 | P0_1 | P0_0 |
| ISEL11_L | P1_3 | P1_2 | P1_1 | P1_0 |
| ISEL11_H | - | - | - | P1_4 |
| ISEL12 | P2_3 | P2_2 | P2_1 | P2_0 |
| ISEL13 | P3_3 | P3_2 | P3_1 | P3_0 |
| ISEL14_L | P4_3 | P4_2 | P4_1 | P4_0 |
| ISEL14_H | - | - | P4_5 | P4_4 |
| ISEL15_L | P5_3 | P5_2 | P5_1 | P5_0 |

Table 41.6 Correspondence between register and each terminal (2/2)

| Bit Name | bit24 | bit16 | bit8 | bit0 |
|---|---|---|---|---|
| ISEL15_H | - | - | - | P5_4 |
| ISEL16_L | P6_3 | P6_2 | P6_1 | P6_0 |
| ISEL16_H | - | - | - | P6_4 |
| ISEL17 | - | - | P7_1 | P7_0 |
| ISEL18 | P8_3 | P8_2 | P8_1 | P8_0 |
| ISEL19_L | P9_3 | P9_2 | P9_1 | P9_0 |
| ISEL19_H | - | - | - | P9_4 |
| ISEL1A | P10_3 | P10_2 | P10_1 | P10_0 |
| ISEL1B | P11_3 | P11_2 | P11_1 | P11_0 |
| ISEL1C | - | - | P12_1 | P12_0 |

## 41.3.7 Driving Ability Control Register (IOLH)

This register sets the buffer drive ability of GPIO and Special Purpose Port.

For the offset address, refer to **Section 41.2.7, Driving Ability Control Register (IOLH)**.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | IOLH3 | | — | — | — | — | — | — | IOLH2 | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | *1 | *1 | 0 | 0 | 0 | 0 | 0 | 0 | *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 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | IOLH1 | | — | — | — | — | — | — | IOLH0 | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | *1 | *1 | 0 | 0 | 0 | 0 | 0 | 0 | *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 | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 26, 23 to 18, 15 to 10, 7 to 2 | — | All 0 | R/W | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value.<br>Operation is not guaranteed if a value other than the initial value is written. |
| 25, 24 | IOLH3 | *1 | R/W | These bits set the drive Ability of each pin. (IOLH3 to IOLH0) |
| 17, 16 | IOLH2 | *1 | R/W | |
| 9, 8 | IOLH1 | *1 | R/W | |
| 1, 0 | IOLH0 | *1 | R/W | |

| Register Value | Pin Group-A |
|---|---|
| 00b | 2 mA |
| 01b (Initial Value) | 4 mA |
| 10b | 8 mA |
| 11b | 12 mA |

| | Pin Group-B | | |
|---|---|---|---|
| | Cload | | |
| Register Value | 3.3 V | 1.8 V | Output Impedance |
| 00b | 5 pF | 5 pF | × 0.5 = 100Ω |
| 01b | 12 pF | 11 pF | × 0.75 = 66Ω |
| 10b (Initial Value) | 26 pF | 15 pF | × 1.0 = 50Ω |
| 11b | 40 pF | 21 pF | × 1.5 = 33Ω |

Note 1. Refer to the above value for Initial Value.
Please refer to the table below for the correspondence of each pin (including Pin Group).

Table 41.7 Correspondence between register and each terminal (1/2)

| Bit Name | bit25-24 | bit17-16 | bit9-8 | bit1-0 | Pin Group |
|---|---|---|---|---|---|
| IOLH01_L | P20_3 | P20_2 | P20_1 | P20_0 | A |
| IOLH01_H | - | - | P20_5 | P20_4 | A |
| IOLH02 | P21_2 | P21_1 | P21_0 | TMS/SWDIO | A |
| IOLH04_L | P22_2 | P22_1 | P22_0 | SD0_CLK | B |
| IOLH04_H | - | - | P22_4 | P22_3 | B |
| IOLH05_L | P23_0 | QSPI0_IO1 | QSPI0_IO0 | QSPI0_SPCLK | B |
| IOLH05_H | - | - | QSPI0_SSL | P23_1 | B |
| IOLH06 | - | - | - | WDTOVF_PERROUT# | A |
| IOLH10 | P0_3 | P0_2 | P0_1 | P0_0 | A |
| IOLH11_L | P1_3 | P1_2 | P1_1 | P1_0 | A |
| IOLH11_H | - | - | - | P1_4 | A |
| IOLH12 | P2_3 | P2_2 | P2_1 | P2_0 | A |
| IOLH13 | P3_3 | P3_2 | P3_1 | P3_0 | A |
| IOLH14_L | P4_3 | P4_2 | P4_1 | P4_0 | A |

Table 41.7 Correspondence between register and each terminal (2/2)

| Bit Name | bit25-24 | bit17-16 | bit9-8 | bit1-0 | Pin Group |
|---|---|---|---|---|---|
| IOLH14_H | - | - | P4_5 | P4_4 | A |
| IOLH15_L | P5_3 | P5_2 | P5_1 | P5_0 | A |
| IOLH15_H | - | - | - | P5_4 | A |
| IOLH16_L | P6_3 | P6_2 | P6_1 | P6_0 | A |
| IOLH16_H | - | - | - | P6_4 | A |
| IOLH17 | - | - | P7_1 | P7_0 | A |
| IOLH18 | P8_3 | P8_2 | P8_1 | P8_0 | A |
| IOLH19_L | P9_3 | P9_2 | P9_1 | P9_0 | A |
| IOLH19_H | - | - | - | P9_4 | A |
| IOLH1A | P10_3 | P10_2 | P10_1 | P10_0 | A |
| IOLH1B | P11_3 | P11_2 | P11_1 | P11_0 | A |
| IOLH1C | - | - | P12_1 | P12_0 | A |

### 41.3.8 Slew Rate Switching Register (SR)

The SR register sets the slew rate of the GPIO pins and Special purpose pins.

For the offset address, refer to **Section 41.2.8, Slew Rate Switching Register (SR)**.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | SR3 | — | — | — | — | — | — | — | SR2 |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 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 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | SR1 | — | — | — | — | — | — | — | SR0 |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 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 25,<br>23 to 17,<br>15 to 9,<br>7 to 1 | — | All 0 | R/W | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value.<br>Operation is not guaranteed if a value other than the initial value is written. |
| 24 | SR3 | 1b | R/W | These bits control the slew rate of each pin (SR7 to SR0)<br>0b: Slow<br>1b: Fast (Initial value)<br>Please refer to the table below for the correspondence of each terminal. |
| 16 | SR2 | 1b | R/W | |
| 8 | SR1 | 1b | R/W | |
| 0 | SR0 | 1b | R/W | |

**Note:** The WDTOVF_PERROUT# bit (H'1468 bit 0) in the SR register is reset by the assertion of the RST_WDTOVFN signal from the CPG.

Table 41.8 Correspondence between register and each terminal (1/2)

| Bit Name | bit24 | bit16 | bit8 | bit0 |
|---|---|---|---|---|
| SR01_L | P20_3 | P20_2 | P20_1 | P20_0 |
| SR01_H | - | - | P20_5 | P20_4 |
| SR02 | P21_2 | P21_1 | P21_0 | TMS/SWDIO |
| SR04_L | P22_2 | P22_1 | P22_0 | SD0_CLK |
| SR04_H | - | - | P22_4 | P22_3 |
| SR05_L | P23_0 | QSPI0_IO1 | QSPI0_IO0 | QSPI0_SPCLK |
| SR05_H | - | - | QSPI0_SSL | P23_1 |
| SR06 | - | - | - | WDTOVF_PERROUT# |
| SR10 | P0_3 | P0_2 | P0_1 | P0_0 |
| SR11_L | P1_3 | P1_2 | P1_1 | P1_0 |
| SR11_H | - | - | - | P1_4 |
| SR12 | P2_3 | P2_2 | P2_1 | P2_0 |
| SR13 | P3_3 | P3_2 | P3_1 | P3_0 |
| SR14_L | P4_3 | P4_2 | P4_1 | P4_0 |
| SR14_H | - | - | P4_5 | P4_4 |
| SR15_L | P5_3 | P5_2 | P5_1 | P5_0 |
| SR15_H | - | - | - | P5_4 |
| SR16_L | P6_3 | P6_2 | P6_1 | P6_0 |

Table 41.8 Correspondence between register and each terminal (2/2)

| Bit Name | bit24 | bit16 | bit8 | bit0 |
|---|---|---|---|---|
| SR16_H | - | - | - | P6_4 |
| SR17 | - | - | P7_1 | P7_0 |
| SR18 | P8_3 | P8_2 | P8_1 | P8_0 |
| SR19_L | P9_3 | P9_2 | P9_1 | P9_0 |
| SR19_H | - | - | - | P9_4 |
| SR1A | P10_3 | P10_2 | P10_1 | P10_0 |
| SR1B | P11_3 | P11_2 | P11_1 | P11_0 |
| SR1C | - | - | P12_1 | P12_0 |

### 41.3.9 Pull Up/Pull down Switching Register (PUPD)

The PUPD register sets pull-up and pull-down of the GPIO pins and the special purpose pins.

For the offset address, refer to **Section 41.2.9, Pull Up/Pull down Switching Register (PUPD)**.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | PUPD3 | | — | — | — | — | — | — | PUPD2 | |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | PUPD1 | | — | — | — | — | — | — | PUPD0 | |
| 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, 23 to 18, 15 to 10, 7 to 2 | — | All 0 | R/W | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value.<br>Operation is not guaranteed if a value other than the initial value is written. |
| 25, 24 | PUPD3 | 00b | R/W | These bits set pull-up and pull-down of each pin (PUPD7 to PUPD0)<br>00b: Neither pull-up nor pull-down is set.<br>01b : Pull-up is selected.<br>10b: Pull-down is selected.<br>11b: Setting prohibited<br>Please refer to the table below for the correspondence of each terminal. |
| 17, 16 | PUPD2 | 00b | R/W | |
| 9, 8 | PUPD1 | 00b | R/W | |
| 1, 0 | PUPD0 | 00b | R/W | |

Table 41.9 Correspondence between register and each terminal (1/2)

| Bit Name | bit25-24 | bit17-16 | bit9-8 | bit1-0 |
|---|---|---|---|---|
| PUPD01_L | P20_3 | P20_2 | P20_1 | P20_0 |
| PUPD01_H | - | - | P20_5 | P20_4 |
| PUPD02 | P21_2 | P21_1 | P21_0 | - |
| PUPD04_L | P22_2 | P22_1 | P22_0 | - |
| PUPD04_H | - | - | P22_4 | P22_3 |
| PUPD05_L | P23_0 | QSPI0_IO1 | QSPI0_IO0 | - |
| PUPD05_H | - | - | - | P23_1 |
| PUPD10 | P0_3 | P0_2 | P0_1 | P0_0 |
| PUPD11_L | P1_3 | P1_2 | P1_1 | P1_0 |
| PUPD11_H | - | - | - | P1_4 |
| PUPD12 | P2_3 | P2_2 | P2_1 | P2_0 |
| PUPD13 | P3_3 | P3_2 | P3_1 | P3_0 |
| PUPD14_L | P4_3 | P4_2 | P4_1 | P4_0 |
| PUPD14_H | - | - | P4_5 | P4_4 |
| PUPD15_L | P5_3 | P5_2 | P5_1 | P5_0 |
| PUPD15_H | - | - | - | P5_4 |
| PUPD16_L | P6_3 | P6_2 | P6_1 | P6_0 |
| PUPD16_H | - | - | - | P6_4 |

Table 41.9 Correspondence between register and each terminal (2/2)

| Bit Name | bit25-24 | bit17-16 | bit9-8 | bit1-0 |
|---|---|---|---|---|
| PUPD17 | - | - | P7_1 | P7_0 |
| PUPD18 | P8_3 | P8_2 | P8_1 | P8_0 |
| PUPD19_L | P9_3 | P9_2 | P9_1 | P9_0 |
| PUPD19_H | - | - | - | P9_4 |
| PUPD1A | P10_3 | P10_2 | P10_1 | P10_0 |
| PUPD1B | P11_3 | P11_2 | P11_1 | P11_0 |
| PUPD1C | - | - | P12_1 | P12_0 |

### 41.3.10 Write protected register (PWPR)

The PWPR register sets write permission / prohibition for the PFC register and the PFCWE bit of this register.

For the offset address, refer to **Section 41.2.10, Write protected register (PWPR)**.

| 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/W | R/W | R/W | R/W | 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 |
| | — | — | — | — | — | — | — | — | BOWI | PFCWE | — | — | — | — | — | — |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 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 8 | — | All 0 | R/W | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value.<br>Operation is not guaranteed if a value other than the initial value is written. |
| 7 | BOWI | 1b | R/W | 0b: Writing a value to the PFCWE bit enabled<br>1b: Writing a value to the PFCWE bit disabled |
| 6 | PFCWE | 0b | R/W | 0b: Writing a value to the PFC register disabled<br>1b: Writing a value to the PFC register enabled |
| 5 to 0 | — | All 0 | R/W | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value.<br>Operation is not guaranteed if a value other than the initial value is written. |

### 41.3.11 Digital Noise Filter Switching Register (FILONOFF)

This register controls whether the digital noise filter are enabled / disabled for GPIO input pin and NMI pin.

For the offset address, refer to **Section 41.2.11, Digital Noise Filter Switching Register (FILONOFF)**.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | FILON3 | — | — | — | — | — | — | — | FILON2 |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | FILON1 | — | — | — | — | — | — | — | FILON0 |
| 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 25,<br>23 to 17,<br>15 to 9,<br>7 to 1 | — | All 0 | R/W | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value.<br>Operation is not guaranteed if a value other than the initial value is written. |
| 24 | FILON3 | 0b | R/W | These bits set the enable or the disable of Digital noise filter of each pin. (FILON7 to FILON0)<br>0b: No filter is used.<br>1b: A filter is used.<br>Please refer to the table below for the correspondence of each terminal. |
| 16 | FILON2 | 0b | R/W | |
| 8 | FILON1 | 0b | R/W | |
| 0 | FILON0 | 0b | R/W | |

**Note:** Before setting this register, disable interrupt detection of the Interrupt controller.

Table 41.10 Correspondence between register and each terminal (1/2)

| Bit Name | bit24 | bit16 | bit8 | bit0 |
|---|---|---|---|---|
| FILONOFF00 | - | - | NMI | - |
| FILONOFF01_L | P20_3 | P20_2 | P20_1 | P20_0 |
| FILONOFF01_H | - | - | P20_5 | P20_4 |
| FILONOFF02 | P21_2 | P21_1 | P21_0 | - |
| FILONOFF04_L | P22_2 | P22_1 | P22_0 | - |
| FILONOFF04_H | - | - | P22_4 | P22_3 |
| FILONOFF05_L | P23_0 | - | - | - |
| FILONOFF05_H | - | - | - | P23_1 |
| FILONOFF10 | P0_3 | P0_2 | P0_1 | P0_0 |
| FILONOFF11_L | P1_3 | P1_2 | P1_1 | P1_0 |
| FILONOFF11_H | - | - | - | P1_4 |
| FILONOFF12 | P2_3 | P2_2 | P2_1 | P2_0 |
| FILONOFF13 | P3_3 | P3_2 | P3_1 | P3_0 |
| FILONOFF14_L | P4_3 | P4_2 | P4_1 | P4_0 |
| FILONOFF14_H | - | - | P4_5 | P4_4 |
| FILONOFF15_L | P5_3 | P5_2 | P5_1 | P5_0 |
| FILONOFF15_H | - | - | - | P5_4 |
| FILONOFF16_L | P6_3 | P6_2 | P6_1 | P6_0 |

Table 41.10 Correspondence between register and each terminal (2/2)

| Bit Name | bit24 | bit16 | bit8 | bit0 |
|---|---|---|---|---|
| FILONOFF16_H | - | - | - | P6_4 |
| FILONOFF17 | - | - | P7_1 | P7_0 |
| FILONOFF18 | P8_3 | P8_2 | P8_1 | P8_0 |
| FILONOFF19_L | P9_3 | P9_2 | P9_1 | P9_0 |
| FILONOFF19_H | - | - | - | P9_4 |
| FILONOFF1A | P10_3 | P10_2 | P10_1 | P10_0 |
| FILONOFF1B | P11_3 | P11_2 | P11_1 | P11_0 |
| FILONOFF1C | - | - | P12_1 | P12_0 |

## 41.3.12 Digital Noise Filter Number Register (FILNUM)

This register sets the number of FF stages of digital noise filter for GPIO pin and NMI pin.

For the offset address, refer to **Section 41.2.12, Digital Noise Filter Number Register (FILNUM)**.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | FILNUM3 | | — | — | — | — | — | — | FILNUM2 | |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | FILNUM1 | | — | — | — | — | — | — | FILNUM0 | |
| 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, 23 to 18, 15 to 10, 7 to 2 | — | All 0 | R/W | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value.<br>Operation is not guaranteed if a value other than the initial value is written. |
| 25, 24 | FILNUM3 | 00b | R/W | 00b: 4-stage filter (41.666 ns x 4 = 166.666 ns) (initial value)<br>01b: 8-stage filter (41.666 ns x 8 = 333.333 ns)<br>10b: 12-stage filter (41.666 ns x 12 = 500 ns)<br>11b: 16-stage filter (41.666 ns x 16 = 666.666 ns)<br>*Note*: This value is the value when the external clock is 24MHz.<br><br>Please refer to the table below for the correspondence of each terminal. |
| 17, 16 | FILNUM2 | 00b | R/W | |
| 9, 8 | FILNUM1 | 00b | R/W | |
| 1, 0 | FILNUM0 | 00b | R/W | |

**Note:** Before setting this register, disable interrupt detection of the Interrupt controller.

Table 41.11 Correspondence between register and each terminal (1/2)

| Bit Name | bit25-24 | bit17-16 | bit9-8 | bit1-0 |
|---|---|---|---|---|
| FILNUM00 | - | - | NMI | - |
| FILNUM01_L | P20_3 | P20_2 | P20_1 | P20_0 |
| FILNUM01_H | - | - | P20_5 | P20_4 |
| FILNUM02 | P21_2 | P21_1 | P21_0 | - |
| FILNUM04_L | P22_2 | P22_1 | P22_0 | - |
| FILNUM04_H | - | - | P22_4 | P22_3 |
| FILNUM05_L | P23_0 | - | - | - |
| FILNUM05_H | - | - | - | P23_1 |
| FILNUM10 | P0_3 | P0_2 | P0_1 | P0_0 |
| FILNUM11_L | P1_3 | P1_2 | P1_1 | P1_0 |
| FILNUM11_H | - | - | - | P1_4 |
| FILNUM12 | P2_3 | P2_2 | P2_1 | P2_0 |
| FILNUM13 | P3_3 | P3_2 | P3_1 | P3_0 |
| FILNUM14_L | P4_3 | P4_2 | P4_1 | P4_0 |
| FILNUM14_H | - | - | P4_5 | P4_4 |
| FILNUM15_L | P5_3 | P5_2 | P5_1 | P5_0 |

Table 41.11 Correspondence between register and each terminal (2/2)

| Bit Name | bit25-24 | bit17-16 | bit9-8 | bit1-0 |
|---|---|---|---|---|
| FILNUM15_H | - | - | - | P5_4 |
| FILNUM16_L | P6_3 | P6_2 | P6_1 | P6_0 |
| FILNUM16_H | - | - | - | P6_4 |
| FILNUM17 | - | - | P7_1 | P7_0 |
| FILNUM18 | P8_3 | P8_2 | P8_1 | P8_0 |
| FILNUM19_L | P9_3 | P9_2 | P9_1 | P9_0 |
| FILNUM19_H | - | - | - | P9_4 |
| FILNUM1A | P10_3 | P10_2 | P10_1 | P10_0 |
| FILNUM1B | P11_3 | P11_2 | P11_1 | P11_0 |
| FILNUM1C | - | - | P12_1 | P12_0 |

### 41.3.13 Digital Noise Filter Clock Selection Register (FILCLKSEL)

The FILCLKSEL register selects the divided clock to be input to digital noise filters.

For the offset address, refer to **Section 41.2.13, Digital Noise Filter Clock Selection Register (FILCLKSEL)**.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | FILCLK3 | | — | — | — | — | — | — | FILCLK2 | |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | FILCLK1 | | — | — | — | — | — | — | FILCLK0 | |
| 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, 23 to 18, 15 to 10, 7 to 2 | — | All 0 | R/W | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value.<br>Operation is not guaranteed if a value other than the initial value is written. |
| 25, 24 | FILCLK3 | 00b | R/W | 00b: Not divided (initial value)<br>01b: Divided by 9000 (41.666 ns x 9000 = 375,000 ns)<br>10b: Divided by 18000 (41.666 ns x 18000 = 750,000 ns)<br>11b: Divided by 36000 (41.666 ns x 36000 = 1,500,000 ns)<br>*Note*: This value is the value when the external clock is 24MHz.<br><br>Please refer to the table below for the correspondence of each terminal. |
| 17, 16 | FILCLK2 | 00b | R/W | |
| 9, 8 | FILCLK1 | 00b | R/W | |
| 1, 0 | FILCLK0 | 00b | R/W | |

**Note:** Before setting this register, disable interrupt detection of the Interrupt controller.

Table 41.12 Correspondence between register and each terminal (1/2)

| Bit Name | bit25-24 | bit17-16 | bit9-8 | bit1-0 |
|---|---|---|---|---|
| FILCLKSEL00 | - | - | NMI | - |
| FILCLKSEL01_L | P20_3 | P20_2 | P20_1 | P20_0 |
| FILCLKSEL01_H | - | - | P20_5 | P20_4 |
| FILCLKSEL02 | P21_2 | P21_1 | P21_0 | - |
| FILCLKSEL04_L | P22_2 | P22_1 | P22_0 | - |
| FILCLKSEL04_H | - | - | P22_4 | P22_3 |
| FILCLKSEL05_L | P23_0 | - | - | - |
| FILCLKSEL05_H | - | - | - | P23_1 |
| FILCLKSEL10 | P0_3 | P0_2 | P0_1 | P0_0 |
| FILCLKSEL11_L | P1_3 | P1_2 | P1_1 | P1_0 |
| FILCLKSEL11_H | - | - | - | P1_4 |
| FILCLKSEL12 | P2_3 | P2_2 | P2_1 | P2_0 |
| FILCLKSEL13 | P3_3 | P3_2 | P3_1 | P3_0 |
| FILCLKSEL14_L | P4_3 | P4_2 | P4_1 | P4_0 |
| FILCLKSEL14_H | - | - | P4_5 | P4_4 |
| FILCLKSEL15_L | P5_3 | P5_2 | P5_1 | P5_0 |

Table 41.12 Correspondence between register and each terminal (2/2)

| Bit Name | bit25-24 | bit17-16 | bit9-8 | bit1-0 |
|---|---|---|---|---|
| FILCLKSEL15_H | - | - | - | P5_4 |
| FILCLKSEL16_L | P6_3 | P6_2 | P6_1 | P6_0 |
| FILCLKSEL16_H | - | - | - | P6_4 |
| FILCLKSEL17 | - | - | P7_1 | P7_0 |
| FILCLKSEL18 | P8_3 | P8_2 | P8_1 | P8_0 |
| FILCLKSEL19_L | P9_3 | P9_2 | P9_1 | P9_0 |
| FILCLKSEL19_H | - | - | - | P9_4 |
| FILCLKSEL1A | P10_3 | P10_2 | P10_1 | P10_0 |
| FILCLKSEL1B | P11_3 | P11_2 | P11_1 | P11_0 |
| FILCLKSEL1C | - | - | P12_1 | P12_0 |

The operation of the digital noise filter will be explained using a timing chart using the external terminal P0_0 as an example.

### (1) When the digital noise filter is disabled (FILONOFF = 0)

The timing chart of FILONOFF = 0 (OFF), FILNUM [1: 0] = “00” (FF4 stage), FILCLKSEL [1: 0] = “00” (without frequency division) is shown below.

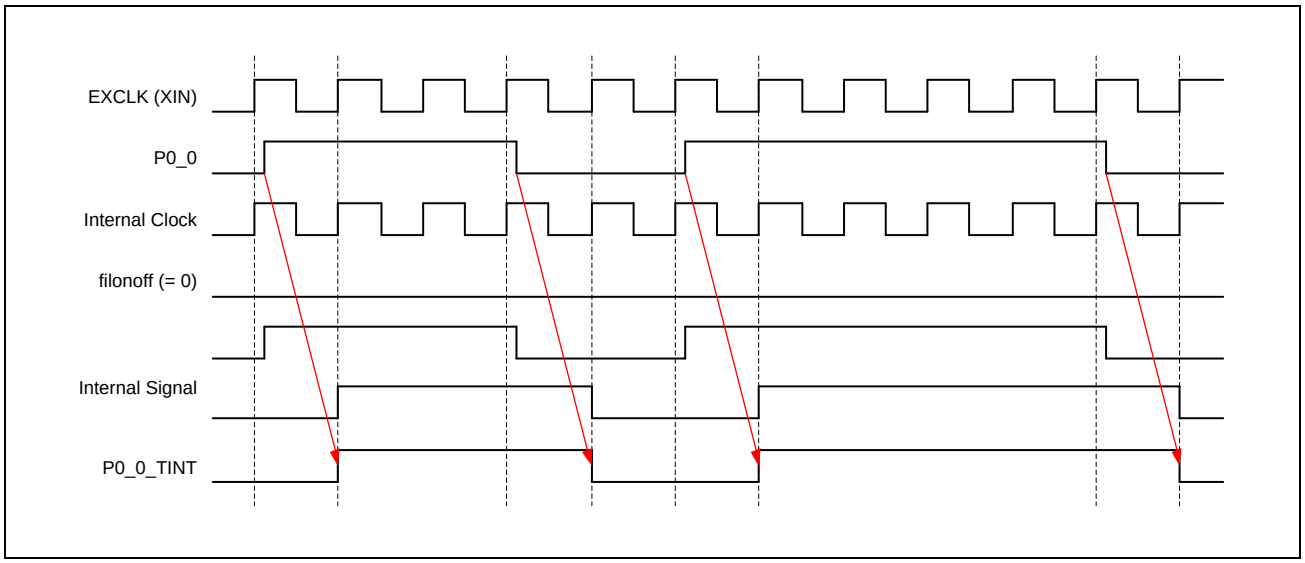



Figure 41.8 Digital noise filter (FILONOFF = 0) timing chart

The signal at the external terminal (P0_0) is not filtered regardless of the number of stages set in the FILNUM register.

The digital noise filter circuit inserts one stage of FF in the output of a digital noise filter.

Therefore, the external terminal (P0_0) signal is propagated internally with a delay of one cycle. (EXCLK (XIN)).

### (2) When the digital noise filter is enabled (FILONOFF = 1)

The timing chart of FILONOFF = 1 (ON), FILNUM [1: 0] = “00” (FF4 stage), FILCLKSEL [1: 0] = “00” (without frequency division) is shown below.

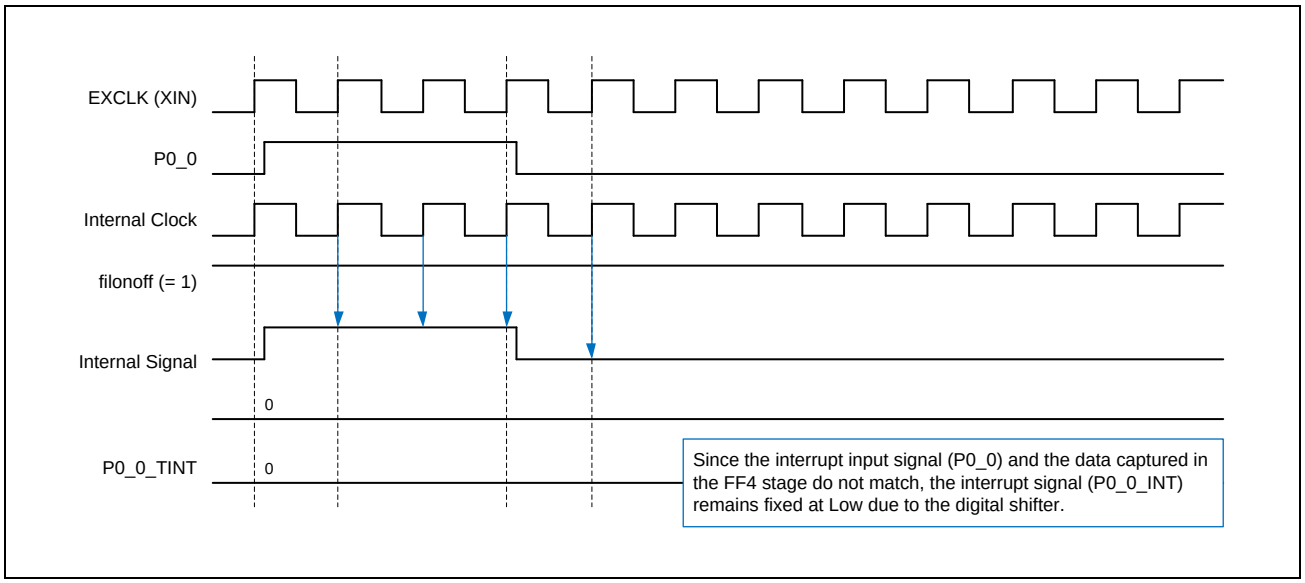



Figure 41.9 Digital noise filter (FILONOFF = 1) Timing chart (1)

The signal of the external terminal (P0_0) is filtered, if even one signal captured by each FF for the number of FF stages set in the FILNUM register does not match.

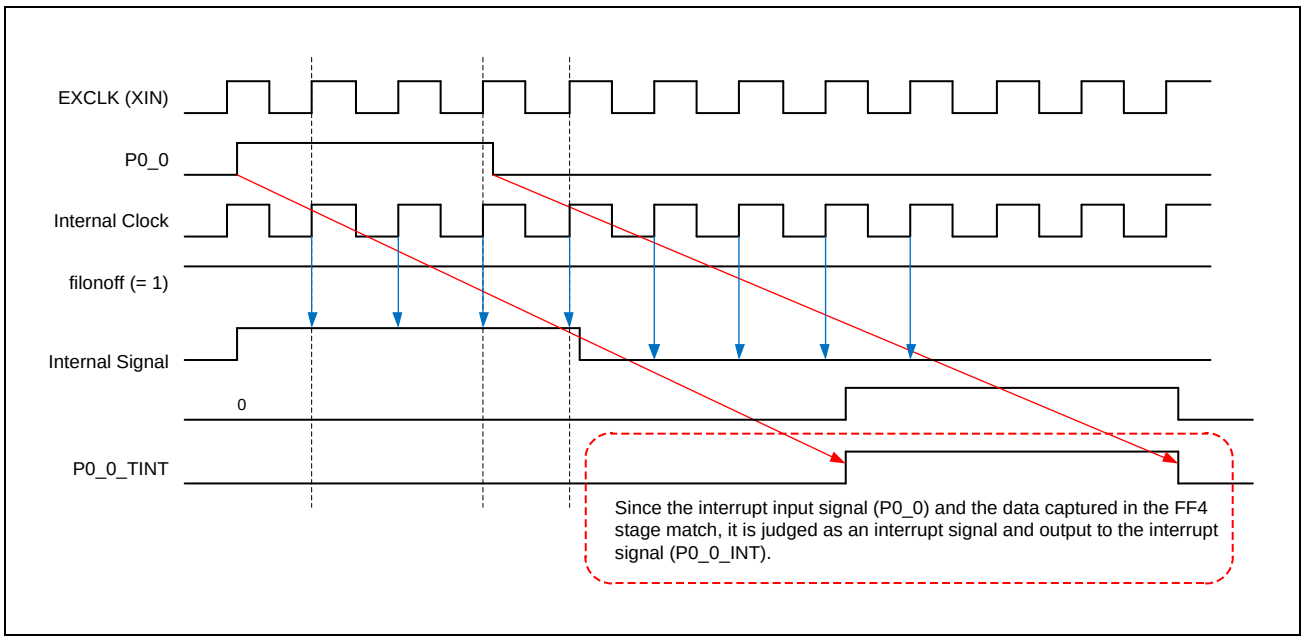



Figure 41.10 Digital noise filter (FILONOFF = 1) Timing chart (2)

The external terminal (P0_0) signal is the signal captured by each FF for the number of FF stages set in the FILNUM register. If they all match, no filtering is done.

The digital noise filter circuit outputs to the interrupt signal with a delay of FILTER_CYCLE.

[FILTER_CYCLE Calculation]

FILTER_CYCLE = FILNUM_FF + $FILTER_FF

FILNUM_FF: Number of FF stages set in the FILNUM register @EXCLK(XIN)
FILTER_FF: FF 2 stage in digital noise filter @EXCLK(XIN) + Digital noise filter output FF 1 stage @EXCLK(XIN)

**Note:** One stage of FF is inserted in the final stage of the digital noise filter circuit. Therefore, interrupt signals less than 1CLK @ EXCLK (XIN) may be removed regardless of whether the digital noise filter is enabled or disabled.

### 41.3.14 SD ch0 IO Voltage Mode Control Register (SD_ch0)

This register sets the IO voltage mode control of SD ch0.

For the offset address, refer to **Section 41.2.14, SD ch0 IO Voltage Mode Control Register (SD_ch0)**.

| 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/W | R/W | R/W | R/W | 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 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | SD0_PVDD |
| Initial Value | 0 | 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 1 | — | All 0 | R/W | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value.<br>Operation is not guaranteed if a value other than the initial value is written. |
| 0 | SD0_PVDD | 0b*1 | R/W | This bit sets the IO voltage mode.<br>0b: I/O domain voltage ≥ 3.3V (Initial value)*1<br>1b: I/O domain voltage ≤ 1.8V |

Note 1. The initial value of this register

Note: When setting this register, do not set the IO voltage mode lower than the applied voltage. For example, if the applied voltage is set to 3.3 V and the register is set to 1.8 V, the reliability of this LSI will be affected.

## 41.3.15 QSPI IO Voltage Mode Control Register (QSPI)

This register sets the IO voltage mode control of QSPI and Octa Memory.

For the offset address, refer to **Section 41.2.15, QSPI IO Voltage Mode Control Register (QSPI)**.

| 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/W | R/W | R/W | R/W | 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 |
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | QSPI_PVDD |
| Initial Value | 0 | 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*1 |

Note 1. In Boot Mode 3, 4, 6 and 7, R / W is R (Read Only).

| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 to 1 | — | All 0 | R/W | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value.<br>Operation is not guaranteed if a value other than the initial value is written. |
| 0 | QSPI_PVDD | 0b*1 | R/W | This bit sets the IO voltage mode.<br>0b: I/O domain voltage ≥ 3.3V (Initial value)*1<br>1b: I/O domain voltage ≤ 1.8V |

Note 1. The initial value of this register

- The IO voltage of QSPI0_SPCLK, QSPI0_IO[3-0], QSPI0_SSL are set by each mode of this register and MD_BOOT[2: 0].
  - When Boot Modes 3, 4, 6 and 7 are not selected: This register can be Read/Write (R/W). The initial value is the above value.
  - When Boot Modes 3, 4, 6 and 7 are selected: This register is Read Only (R), and the voltage mode of the selected device is as follows.

| MD_BOOT[2:0] | | | Mode | Selection Function | Selection Device | Control method |
|---|---|---|---|---|---|---|
| 0 | 1 | 1 | Boot Mode 3 | SPI Multi I/O Bus Controller | Single/Quad Serial NOR flash memory (1.8 V) | By Hardware |
| 1 | 0 | 0 | Boot Mode 4 | SPI Multi I/O Bus Controller | Single/Quad Serial NOR flash memory (3.3 V) | By Hardware |
| 1 | 1 | 0 | Boot Mode 6 | SPI Multi I/O Bus Controller | Single/Quad Serial NAND flash memory (1.8 V) | By Hardware |
| 1 | 1 | 1 | Boot Mode 7 | SPI Multi I/O Bus Controller | Single/Quad Serial NAND flash memory (3.3 V) | By Hardware |
| Other | | | QSPI Control by register | | | |

**Note:** When setting this register, do not set the IO voltage mode lower than the applied voltage. For example, if the applied voltage is set to 1.8V and the register is set to 3.3V, the reliability of this LSI will be affected.

## 41.3.16 Input Enable Control Register (IEN)

This register is a register that controls the input of the special purpose port.

For the offset address, refer to **Section 41.2.16, Input Enable Control Register (IEN)**.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | IEN3 | — | — | — | — | — | — | — | IEN2 |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | IEN1 | — | — | — | — | — | — | — | IEN0 |
| 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 25,<br>23 to 17,<br>15 to 9,<br>7 to 1 | — | All 0 | R/W | Reserved<br>When read, the initial value is read. When writing, be sure to write the initial value.<br>Operation is not guaranteed if a value other than the initial value is written. |
| 24 | IEN3 | 0b | R/W | These bits control inputs of p bit of a pin with a multiplexed function (except for GPIO) and a dedicated pin (p = 0 to 7)<br>0b: Input disabled (Initial value)<br>1b: Input enabled<br>Please refer to the table below for the correspondence of each terminal. |
| 16 | IEN2 | 0b | R/W | |
| 8 | IEN1 | 0b | R/W | |
| 0 | IEN0 | 0b | R/W | |

Table 41.13 Correspondence between register and each terminal

| Bit Name | bit24 | bit16 | bit8 | bit0 |
|---|---|---|---|---|
| IEN02 | - | - | - | TMS/SWDIO |
| IEN04_L | P22_2 | P22_1 | P22_0 | - |
| IEN04_H | - | - | P22_4 | P22_3 |
| IEN07 | - | - | RIIC0_SCL | RIIC0_SDA |

## 41.4 Operation

The operating procedure for the general-purpose I/O port function and the operating procedure for peripheral functions are shown below.

### 41.4.1 Operation for GPIO Function

(1) Set the PMCn register to “0” (initial value = 0).

(2) Set the GPIO input / output with the PMn register.

(3) Set the GPIO output value in the Pn register.

(4) The input value of the input terminal can be detected by the PINn register.

(5) When setting the interrupt input using the GPIO input pin, set the ISEL register to “1”.

- Set the FILONOFF, IOLH, SR, and PUPD registers between (1) and (2) as necessary.
- Set the FILONOFF register when using GPIO to set a digital noise filter for interrupt input and GPIO input.
- Set the FILNUM and FILCLKSEL registers according to the purpose of use.

### 41.4.2 Operation for Peripheral Function

(1) Set the PMCn register to “1” (initial value = 0).

(2) Set the PWPR register to allow writing to the PFCm register.

After setting the PWPR.B0WI bit to “0” (initial value = 1), set the PWPR.PFCWE bit to “1” (initial value = 0).

Select the required function from Functions 1-8. (Hereafter, Function1 setting example)

(3) Set PFCm = 001b and switch to Function1.

(4) Set the PFCm register to write-protected. After setting the PWPR.PFCWE bit to “0”, set the PWPR.B0WI bit to “1”.

- Set the FILONOFF, IOLH, SR, and PUPD registers between (1) and (2) as necessary.
- Set the FILONOFF register when using GPIO to set a digital noise filter for the IRQ[0-7].
- Set the FILNUM and FILCLKSEL registers according to the purpose of use. And also, the NMI input can set the noise filter by the same registers.

# 42. Low Power Mode

This chapter describes the lower mode of this LSI.

## 42.1 Overview

This LSI has two modes of low power consumption shown in the following table.

These two modes work independently.

Table 42.1 Low Power Mode

| Low Power Mode | Description | Return factor |
|---|---|---|
| Module Standby Mode | Stop the clock of specific peripheral module by setting the CPG register.<br>Refer to **Section 42.2, Module Standby Mode** for detail. | CPG register setting |
| Cortex-A55 Sleep Mode | Put Cortex-A55 into a low power state by issuing a WFI instruction.<br>Refer to **Section 2.3.1, Cortex-A55 Sleep Mode** and **Section 42.3, Cortex-A55 Sleep Mode**for detail. | Interrupt |

**Note:** When a system reset is applied, the LSI is released from the low power mode and initialized.

## 42.2 Module Standby Mode

### 42.2.1 Overview

The Module Standby Mode is a mode that requests the clock stop of the module specified by the master.

The purpose of this mode is to reduce power consumption by stopping unnecessary functions.

The master which can request the clock stop is Cortex-A55. So, Cortex-A55 cannot be placed in the Module Standby Mode. To reduce the power consumption in the CPUs, use the CPU sleep modes to be described later.

This mode is implemented by stopping the clock supply by the CPG register and switching the MSTOP signal of the bus through CPG register settings.

If the master accesses a module that has the clock stopped and the MSTOP bit set, a bus error will occur.

The following registers are used in the Module Standby Mode.

Table 42.2 Module Standby Mode related registers

| Register name | Abbreviation | Description |
|---|---|---|
| Clock Control Register | CPG_CLKON_*** | The detail information of each register is described in **Section 6, System Controller (SYSC)** and **Section 7, Clock Pulse Generator (CPG)**.<br>The usage of these registers is described in **Section 42.2.2, Operation**. |
| MSTOP Register | CPG_BUS_***_MSTOP | |

The following figure shows the block connection overview related Module Standby Mode.

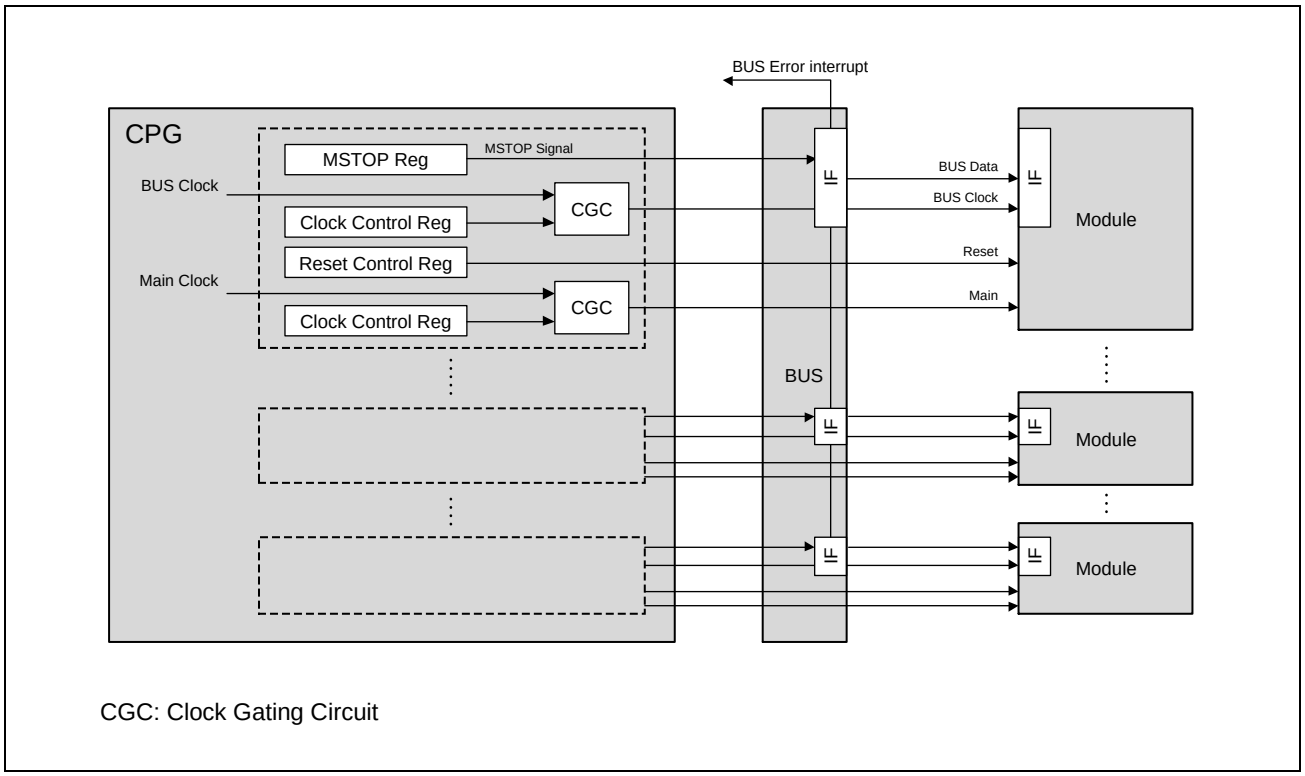



Figure 42.1 Block connection overview for Module Standby Mode

### 42.2.2 Operation

The following chart shows the procedure to enter and return from the Module Standby Mode.

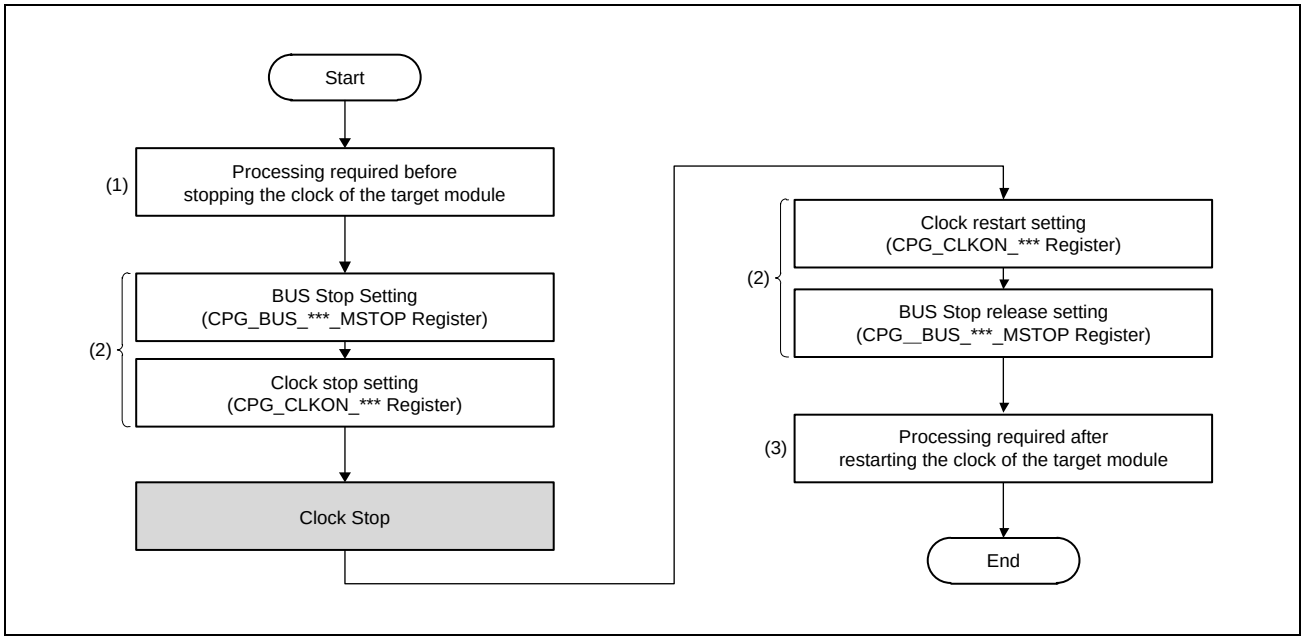



Figure 42.2 Module Standby Mode procedure

(1) Depend on the module, there may be a request to stop during communication, image processing, and various processing. In this case, if the managed CPU can be stopped unilaterally for each module, a stop request is immediately issued to the CPG. For those that cannot be stopped until a certain process is completed, after confirming that the managed CPU has stopped for each module, a stop request is made to the CPG.

(2) Refer to **Section 42.2.2.1, Setting of Clock control and MSTOP register** regarding to the setting of Clock Control Register and MSTOP Register.

(3) Whether the reset control is required depends on the module; check the specifications of the target module. Set the reset control register to 0 to enter the reset state. Set 1 for the release from the reset state.

### 42.2.2.1 Setting of Clock control and MSTOP register

The following table shows the combination of the Clock Control register and the MSTOP register to stop the clock of the target module.

The Module Standby Mode can be applied only to the modules shown in the table below.

- When entering the Module Standby Mode:

  Set the MSTOP Register bit of the target module to 1.
  Set the Clock Control Register bit of the target module to 0.

- When exiting the Module Standby Mode:

  Set the Clock Control Register bit of the target module to 1.
  Set the MSTOP Register bit of the target module to 0.

Table 42.3 Registers for Module Standby Mode (1/2)

| Target module for Module Standby Mode | Clock Control Register | | MSTOP Register | |
|---|---|---|---|---|
| | Register Name (CPG_CLKON_***) | bit | Register Name (CPG_BUS_***_MSTOP) | bit |
| GIC | GIC600 | [0] | REG1 | [7] |
| IA55 | IA55 | [1:0] | PERI_CPU | [13] |
| CoreSight | CST | [10:0] | PERI_CPU | [2] |
| SYC | SYC | [1] | PERI_CPU | [3] |
| DMAC | DMAC_REG | [1:0] | REG1 | [3:0] |
| GTM ch0 | GTM | [0] | REG0 | [4] |
| GTM ch1 | | [1] | | [5] |
| GTM ch2 | | [2] | | [6] |
| MTU | MTU | [0] | MCPU1 | [2] |
| POE3 | POE3 | [0] | MCPU1 | [9] |
| WDT ch0 | WDT | [1:0] | REG0 | [2] |
| SPI Multi | SPI_MULTI | [1:0] | MCPU1 | [1] |
| SDHI ch0 | SDHI | [3:0] | PERI_COM | [0] |
| MIPI DSI | MIPI_DSI | [5:0] | PERI_VIDEO | [6:5] |
| LCDC | LCDC | [1:0] | PERI_VIDEO | [9:7] |
| SSI ch0 | SSI | [1:0] | MCPU1 | [10] |
| USB2.0 ch0 | USB | [0], [2] | PERI_COM | [6:5] |
| USB2.0 PHY *1 | | [3] | | [4] |
| I2C ch0 | I2C | [0] | MCPU2 | [10] |
| I2C ch1 | | [1] | | [11] |
| SCIF ch0 | SCIF | [0] | MCPU2 | [1] |
| SCIF ch1 | | [1] | | [2] |
| SCIF ch2 | | [2] | | [3] |
| SCIF ch3 | | [3] | | [4] |
| SCIF ch4 | | [4] | | [5] |
| SCI ch0 | SCI | [0] | MCPU2 | [7] |
| SCI ch1 | | [1] | | [8] |

Table 42.3 Registers for Module Standby Mode (2/2)

| Target module for Module Standby Mode | Clock Control Register | | MSTOP Register | |
|---|---|---|---|---|
| | Register Name (CPG_CLKON_***) | bit | Register Name (CPG_BUS_***_MSTOP) | bit |
| IRDA | IRDA | [0] | MCPU2 | [6] |
| RSPI ch0 | RSPI | [0] | MCPU1 | [14] |
| RSPI ch1 | | [1] | | [15] |
| TSU | TSU | [0] | MCPU2 | [15] |
| DRW | DRW | [0] | PERI_COM | [9] |

Note 1. The clock for USB2.0 PHY can only be stopped when the clock for USB2.0 ch0 is stopped.

## 42.3 Cortex-A55 Sleep Mode

### 42.3.1 Overview

Refer to **Section 2.3.1, Cotrex-A55 Sleep Mode** for detail.

The following table shows the registers related to the Cortex-A55 Sleep Mode.

Refer to **Section 6, System Controller (SYSC)** for detail.

Table 42.4 Cotrex-A55 Sleep Mode related registers

| Register name | Abbreviation | Bit name |
|---|---|---|
| Lowpower Sequence Control Register2 | SYS_LP_CTL2 | bit[0]: CA55_STBYCTL |

# 43. Debug Interface

## 43.1 Overview

This LSI has a debug interface for boundary scan function and debug support for Cortex-A55.

### 43.1.1 Features

The functions of the debug interface of this LSI are shown below.

- Debug Interface
  - Support JTAG and SWD
- Debugger control function (DAP function)
  - Direct control of IP without going through CPU by switching Access Port (AP)
- Trace data support
  - Cortex-A55 trace output
  - Buffer function of trace data by ETF
  - Trace data output to system bus by using ETR
- Boundary Scan
  - Connection test between this LSI and other LSIs on the user board
- Other functions
  - Interlocking operation of Cortex-A55, system counter (SYC), watchdog timer, general timer (GTM), and debug component by cross trigger
  - Support WDT counter stop control function
  - Add time information to trace data by Time Stamp (24MHz operation)

### 43.1.2 Block Diagram

**Figure 43.1** shows the block diagram of the debug interface, and **Figure 43.2** shows the block diagram of the Core Sight System (CST).

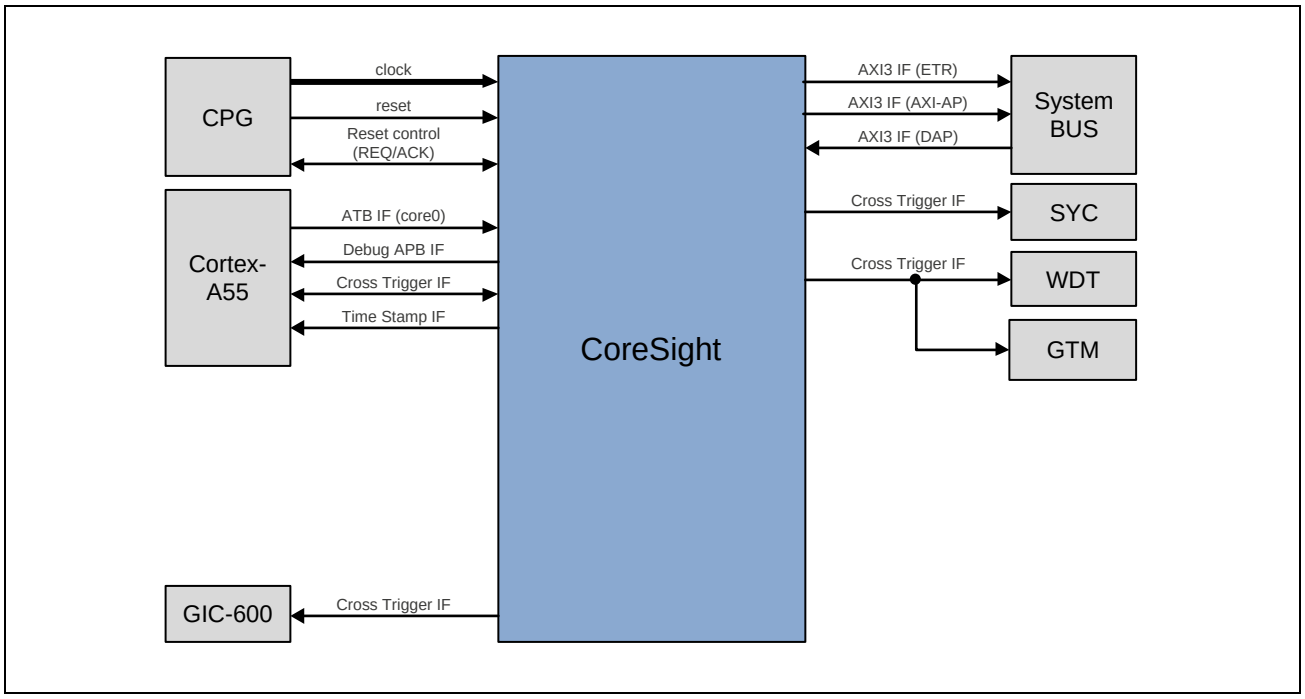



Figure 43.1 Debug System Block Diagram

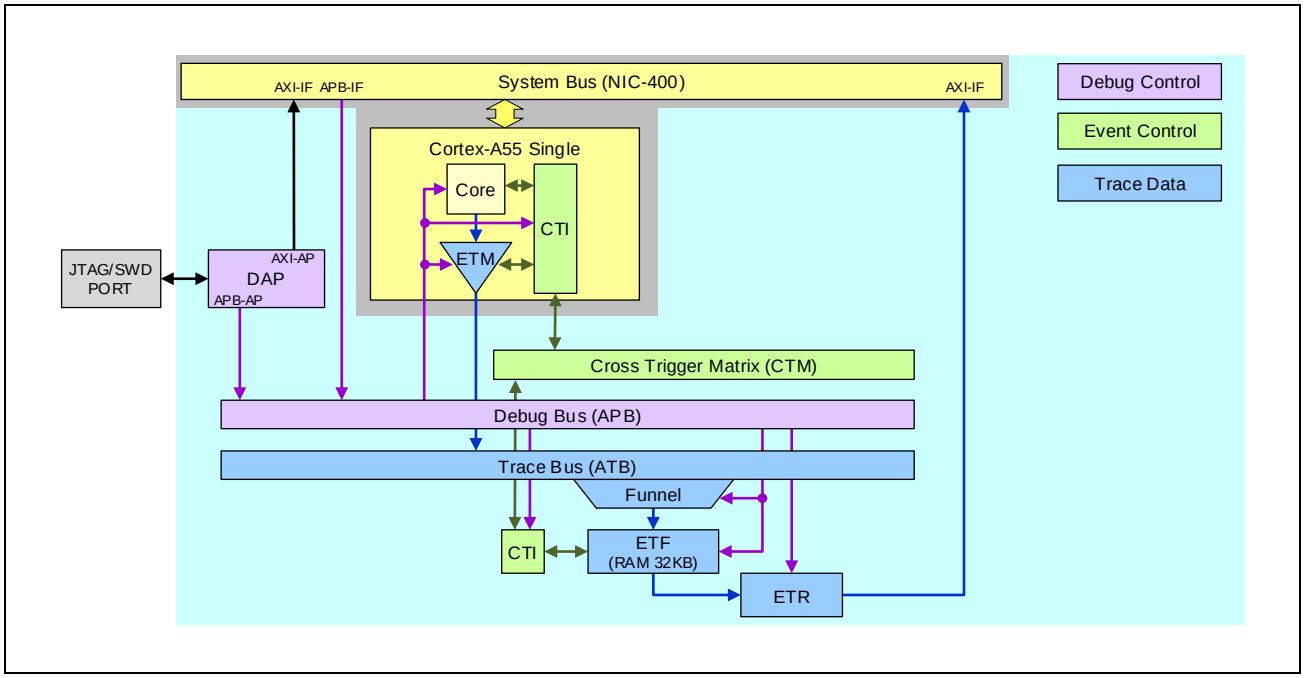



Figure 43.2 Core Sight System Block Diagram

- Debug Bus (APB)

  Bus used for controlling debug functions and acquiring debug data (register values and trace data in On-Chip Buffer)

  - DAP: Debug Access Port

- Cross Trigger Matrix (CTM)

  Functions required for co-debugging triggered by mutual events between CPU and Component

  - CTI: Cross Trigger Interface

- Trace Bus (ATB)

  A data bus which the trace data output by ETM flows

  - ETM: Embedder Trace Macrocell
  - ETR: Embedder Trace Router
  - ETF: Embedder Trace FIFO
  - Funnel: Trace data merge function

For details on ETM, ETR, and ETF, refer to “CoreSight Trace Memory Controller Technical Reference Manual”.

### 43.1.3 External Pins

- JTAG pins

The debug interface supports JTAG IF and SWD IF. **Table 43.1** shows the JTAG pins and their functions.

Table 43.1 JTAG Pins

| Pin Name | I/O | Function | | Name |
|---|---|---|---|---|
| | | JTAG IF*1 | SWD IF*1 | |
| TCK/SWDCLK | Input | TCK | SWCLK | Test clock,<br>Serial wire clock |
| TMS/SWDIO | Input,<br>Input/Output | TMS | SWDIO | Test mode select,<br>Serial wire data input/output |
| TDI | Input | TDI | — | Test data input |
| TDO | Output | TDO | — | Test data output |
| TRST# | Input | TRST | — | Test reset |

Note 1. JTAG IF is available when SWD_JTAG = 0. SWD IF is available when SWD_JTAG = 1.

- SWD_JTAG pin

**Table 43.2** shows the functions of the SWD_JTAG pin.

Table 43.2 SWD_JTAG Pins

| Pin Name | I/O | Function |
|---|---|---|
| SWD_JTAG | Input | Switches between JTAG mode and SWD mode in debug operation.<br>0: Connect the debugger to the JTAG IF pins and debug in JTAG mode. The JTAG IF pin operates as a dedicated pin.<br>1: Connect the debugger to the SWD IF pins and debug in SWD mode. The TRST#, TDI, TDO pins can be used as GPIO pins. *1 |

Note 1. Boundary scan cannot be performed in SWD mode.

The SWD_JTAG value must be fixed before releasing the power-on reset (PRST#).

- DEBUGEN pins

**Table 43.3** shows the functions of the DEBUGEN pin.

Table 43.3 DEBUGEN Pins

| Pin Name | I/O | Function |
|---|---|---|
| DEBUGEN | Input | Switches between normal operation and debug operation.<br>0: Normal operation<br>1: Debug operation |

The DEBUGEN value must be fixed before releasing the power-on reset (PRST#). **Table 43.4** shows the difference in operation between normal operation and debug operation.

Table 43.4 Difference in Operation by DEBUGEN

| | Normal Operation (DEBUGEN＝0) | Debug Operation (DEBUGEN＝1) | Related Module |
|---|---|---|---|
| Debug function module (CoreSight (other than DAP)) | Boot impossible (Reset state) | Boot (Refer **Section 43.6.1** for reset condition and range) | CPG (Reset control) |

- BSCANP pins

**Table 43.5** shows the functions of the BSCANP pin.

Table 43.5 BSCANP Pins

| Pin Name | I/O | Function |
|---|---|---|
| BSCANP | Input | Switches between normal operation and boundary scan test mode.<br>0: Normal operation<br>1: Boundary scan test mode |

Boundary scan test mode is a mode in which a connection test is performed between the terminals of this LSI and other ICs connected on the customer's board and is performed when BSCANP = 1.

The actual access is controlled by the 5 terminals of JTAG when BSCANP = 1.

**Table 43.6** shows the operation modes of this LSI according to the values of the DEBUGEN pin and BSCANP pin.

Table 43.6 Operating Mode

| Pin Name | | |
|---|---|---|
| BSCANP | DEBUGEN | Operating Mode |
| 0 | 0 | Normal operation |
| 0 | 1 | Debug operation |
| 1 | * | Boundary scan test mode |

### 43.1.4 Treatment of Unused Pins

When the external terminal of the debug interface is not used, handle it as shown in **Table 43.7**.

Table 43.7 Handling of Unused Pins

| Pin Name | Handle*1 |
|---|---|
| TCK/SWDCLK | Fix the level on the pins (pull them up or down, or connect them to the power supply or ground level) |
| TMS/SWDIO | Fix the level on the pins (pull them up or down, or connect them to the power supply or ground level) |
| TDI | Fix the level on the pins (pull them up or down, or connect them to the power supply or ground level) |
| TDO | Open-circuit |
| TRST#*2 | Fix this pin at a low level (pull down or connect to the ground level) |
| DEBUGEN | Fix this pin at a low level (pull down or connect to the ground level) |
| BSCANP | Fix this pin at a low level (pull down or connect to the ground level) |
| SWD_JTAG*3 | Fix the level on the pins (pull them up or down, or connect them to the power supply or ground level) |

Note 1. Mount the pull-up/pull-down resistors on the board side

Note 2. TRST# should be fixed at low level in non-debug mode, and TRST# should be changed from low level to high level in debug mode.

Note 3. When TRST#, TDI, and TDO pins are used as a GPIO pins, fix this pin at a high level (pull up or connect to the power supply)

## 43.2 Address Space

The CoreSight of this LSI has a 4 MB debug system address space. The address space of the debug system is shown in **Table 43.8**.

Table 43.8 Address Space of the Debug System

| Address | Component*1 |
|---|---|
| CoreSight Debug Components (Overall Address Space of this LSI) | |
| H'0_10C0_0000 ~ H'0_10C0_FFFF | CoreSight: ROM table |
| H'0_10C1_0000 ~ H'0_10C1_FFFF | CoreSight: Timestamp generator |
| H'0_10C2_0000 ~ H'0_10C2_FFFF | CoreSight: ETF |
| H'0_10C3_0000 ~ H'0_10C3_FFFF | CoreSight: ETR |
| H'0_10C4_0000 ~ H'0_10C4_FFFF | CoreSight: Trace Funnel |
| H'0_10C5_0000 ~ H'0_10C5_FFFF | CoreSight: CTI0 |
| H'0_10C6_0000 ~ H'0_10C6_FFFF | CoreSight: CTI1 |
| H'0_10C7_0000 ~ H'0_10CE_FFFF | CoreSight: Reserved |
| H'0_10CF_0000 ~ H'0_10CF_FFFF | CoreSight: Reserved |
| Reserved | |
| H'0_10D0_0000 ~ H'0_10DF_FFFF | Reserved |
| Cortex-A55 Debug Components (Overall Address Space of this LSI) | |
| H'0_10E0_0000 ~ H'0_10E0_FFFF | Cortex-A55: ROM table |
| H'0_10E1_0000 ~ H'0_10E1_FFFF | Cortex-A55: Core0 Debug |
| H'0_10E2_0000 ~ H'0_10E2_FFFF | Cortex-A55: Core0 CTI |
| H'0_10E3_0000 ~ H'0_10E3_FFFF | Cortex-A55: Core0 PMU |
| H'0_10E4_0000 ~ H'0_10E4_FFFF | Cortex-A55: Core0 ETM |
| H'0_10E5_0000 ~ H'0_10ED_FFFF | Reserved |
| H'0_10EE_0000 ~ H'0_10EE_FFFF | Reserved |
| H'0_10EF_0000 ~ H'0_10F0_FFFF | Reserved |
| H'0_10F1_0000 ~ H'0_10F1_FFFF | Reserved |
| H'0_10F2_0000 ~ H'0_10F2_FFFF | Reserved |
| H'0_10F3_0000 ~ H'0_10F3_FFFF | Reserved |
| H'0_10F4_0000 ~ H'0_10F4_FFFF | Reserved |
| H'0_10F5_0000 ~ H'0_10FF_FFFF | Cortex-A55: Reserved |
| Reserved Address Space | |
| H'E000_0000 ~ H'E000_0FFF | Reserved |
| H'E000_1000 ~ H'E000_1FFF | Reserved |
| H'E000_2000 ~ H'E000_2FFF | Reserved |
| H'E000_3000 ~ H'E000_DFFF | Reserved |
| H'E000_E000 ~ H'E000_EFFF | Reserved |
| H'E000_F000 ~ H'E002_DFFF | Reserved |
| H'E002_E000 ~ H'E002_EFFF | Reserved |
| H'E002_F000 ~ H'E004_0FFF | Reserved |
| H'E004_1000 ~ H'E004_1FFF | Reserved |
| H'E004_2000 ~ H'E004_2FFF | Reserved |
| H'E004_3000 ~ H'E00F_EFFF | Reserved |
| H'E00F_F000 ~ H'E00F_FFFF | Reserved |

Note 1. Access to the Reserved area is prohibited. Operation is not guaranteed when accessed.

NOTE

- The CoreSight Debug Components area and Cortex-A55 Debug Components area can be accessed via the system bus and APB-AP.
  APB-AP is connected to port-0 of DAP. For the port configuration of DAP, refer to **Section 43.6.2, DAP**.
- Each component is placed on a 64KB address boundary according to the ARMv8 Debug memory map. As an entity, the first 4 KB is valid.

## 43.3 Register Descriptions

### 43.3.1 ID Register (BSID)

The BSID register is a 32-bit register that cannot be accessed from the CPU. When the BSCANP pin is 1, it can be read from the TDO pin by setting the IDCODE command on the TAP controller. It cannot be written.

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

| Bit | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | BSID[15:0] | | | | | | | | | | | | | | | |
| Initial Value | 1 | 0 | 1 | 0 | 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 | BSID[31:0] | H’0867_A447 | — | This is the ID register specified in JTAG.<br>The upper 4 bits (bit31-28) may change depending on the chip version. |

### 43.3.2 Other Registers

For details on each register other than the BSID register, refer to “ARM CoreSight SoC-400 Technical Reference Manual” and “CoreSight Trace Memory Controller Technical Reference Manual”.

## 43.4 Debug Connection

### 43.4.1 Debug Connection Mode

There is the following only one mode for the debug connection method of this LSI.

| Authentication Mode | Method |
|---|---|
| No authentication | Allow access from the debugger without authentication (initial setting) |

## 43.5 Debug Access Range

The debug access range by CoreSight is for all slaves as well as Cortex-A55. For details, refer to **Section 5.3, Accessible Areas**.

## 43.6 Operation

### 43.6.1 Debug Related Reset

Please refer to **Section 7.3.2.1, Range of Reset Application** for debug-related reset and reset range.

### 43.6.2 DAP

DAP is equipped with SWJ-DP as Debug Port (DP) and APB-AP, AXI-AP, AHB-AP as Access Port (AP).

The DAP port numbers are shown in **Table 43.9**. Indicates the correspondence between the port number specified in the DP SELECT register and the AP, and its usage.

Table 43.9 DAP Port Number

| Port number | AP | Usage |
|---|---|---|
| Port-0 | APB-AP | Debug Component recognition and settings |
| Port-1 | AXI-AP | System IP control (access to system bus) |

### 43.6.3 Trace

#### 43.6.3.1 Data Flow

In CoreSight, all trace data is aggregated once by Trace Funnel, and stored in ETF, or output to the system bus via ETR in the subsequent stage. Setting this port number to the Trace Funnel according to the route of the trace data is needed. The Trace Funnel port numbers are shown in **Table 43.10**.

Table 43.10 Trace Funnel Port Number

| Port Number | Source Components |
|---|---|
| Port-0 | CA55: Core0 ETM |
| Port-1 | Reserved |

#### 43.6.3.2 Timestamp

If time information want to be included in the trace data, the Timestamp Generator can be used. The timestamp replicator port numbers are shown in **Table 43.11**. The Timestamp Generator runs on a 24-MHz clock.

Table 43.11 Timestamp Replicator Port Number

| Port Number | Target Components |
|---|---|
| Port-0 | CA55: Core0 ETM |

### 43.6.4 Cross Trigger

The CTI (Cross Trigger Interface) implemented in CoreSight, Cortex-A55 is used to communicate debug events.

#### 43.6.4.1 CoreSight Cross Trigger Connectiion

CoreSight has two CTIs. The connection specifications are shown in **Table 43.12** to **Table 43.15**.

Table 43.12 CoreSight CTI0 Trigger Input

| Trigger Input Bit | Source Components | Description |
|---|---|---|
| [7] | (Unused) | — |
| [6] | (Unused) | — |
| [5] | (Unused) | — |
| [4] | (Unused) | — |
| [3] | ETR | ETR Event Output 1 (FULL) |
| [2] | ETR | ETR Event Output 0 (ACQCOMP) |
| [1] | ETF | ETF Event Output 1 (FULL) |
| [0] | ETF | ETF Event Output 0 (ACQCOMP) |

Table 43.13 CoreSight CTI0 Trigger Output

| Trigger Output Bit | Destination Components | Description |
|---|---|---|
| [7] | SYC | SYC Event input1 (RESTART REQ/ACK) |
| [6] | SYC | SYC Event input0 (HALT REQ) |
| [5] | WDT/OSTM | WDT/OSTM Event input1 (RESTART REQ/ACK) |
| [4] | WDT/OSTM | WDT/OSTM Event input0 (HALT REQ) |
| [3] | ETR | ETR Event Input 1 (TRIGIN) |
| [2] | ETR | ETR Event Input 0 (FLUSHIN) |
| [1] | ETF | ETF Event Input 1 (TRIGIN) |
| [0] | ETF | ETF Event Input 0 (FLUSHIN) |

Table 43.14 CoreSight CTI1 Trigger Input

| Trigger Input Bit | Source Components | Description |
|---|---|---|
| [7] | (Unused) | — |
| [6] | (Unused) | — |
| [5] | (Unused) | — |
| [4] | (Unused) | — |
| [3] | (Unused) | — |
| [2] | (Unused) | — |
| [1] | (Unused) | — |
| [0] | (Unused) | — |

Table 43.15 CoreSight CTI1 trigger output

| Trigger Output Bit | Destination Components | Description |
|---|---|---|
| [7] | (Unused) | — |
| [6] | (Unused) | — |
| [5] | (Unused) | — |
| [4] | (Unused) | — |
| [3] | (Unused) | — |
| [2] | (Unused) | — |
| [1] | Cortex-A55 | Cortex-A55 Event Input (PMUSNAPSHOTREQ/PMUSNAPSHOTACK) |
| [0] | GIC-600 | GIC-600 Event Input (sample_req/sample_ack) |

### 43.6.4.2 Cortex-A55 Cross Trigger Connectiion

The Cortex-A55 has one CTI (Cortex-A55 Core0 CTI). The connection specifications for each are shown in **Table 43.16** and **Table 43.17**.

Table 43.16 Cortex-A55 CTI0 trigger input

| Trigger Input Bit | Source Components | Description |
|---|---|---|
| [7] | Cortex-A55: Core0 ETM | ETM Trace Output 3 trigger event |
| [6] | Cortex-A55: Core0 ETM | ETM Trace Output 2 trigger event |
| [5] | Cortex-A55: Core0 ETM | ETM Trace Output 1 trigger event |
| [4] | Cortex-A55: Core0 ETM | ETM Trace Output 0 trigger event |
| [3] | (Unused) | — |
| [2] | Cortex-A55: Core0 PE | Profiling sample trigger event |
| [1] | Cortex-A55: Core0 PE | Performance Monitors Overflow trigger event |
| [0] | Cortex-A55: Core0 PE | Cross-halt trigger event |

Table 43.17 Cortex-A55 CTI0 Trigger Output

| Trigger Output Bit | Destination Components | Description |
|---|---|---|
| [7] | Cortex-A55: Core0 ETM | Generic Trace External Input 3 trigger event |
| [6] | Cortex-A55: Core0 ETM | Generic Trace External Input 2 trigger event |
| [5] | Cortex-A55: Core0 ETM | Generic Trace External Input 1 trigger event |
| [4] | Cortex-A55: Core0 ETM | Generic Trace External Input 0 trigger event |
| [3] | (Unused) | — |
| [2] | External (GIC-600) | Generic CTI Interrupt trigger event |
| [1] | Cortex-A55: Core0 PE | Restart Request trigger event |
| [0] | Cortex-A55: Core0 PE | Debug Request trigger event |

### 43.6.5 WDT Counter Stop Control

This LSI has a WDT counter stop control function linked with a debug event to prevent an unintended reset by WDT during debug operation. When a debug event occurs, CoreSight asserts the WDT counter stop request signal, controls the WDT counter stop signal (CNTSTOP), and stops counting. The target WDT is all WDT. For details on the WDT count stop function, refer to **Section 6.4.2, WDT Stop control function**.

### 43.6.6 GTM Counter Stop Control

This LSI has a GTM counter stop control function linked with debug events to prevent unintended GTM counts during debug operation. When a debug event occurs, CoreSight asserts a GTM counter stop request signal to stop the GTM count.

### 43.6.7 SYC Counter Stop Control

This LSI has a SYC counter stop control function linked with debug events to prevent unintended SYC counts during debug operation. When a debug event occurs, CoreSight asserts a SYC counter stop request signal to stop the SYC count.

# 44. Thermal Sensor Unit (TSU)

This LSI chip incorporates a thermal sensor unit (TSU) that measures the temperature inside the LSI.

## 44.1 Overview

### 44.1.1 Features

- This LSI has a single TSU.
- The thermal sensor in this unit measures temperatures in the range from −40°C to 110°C with an accuracy of ±3°C.
- The TSU repeats measurement at 20-µs intervals, and automatically updates the results of measurement.

### 44.1.2 Block Diagram

**Figure 44.1** is a block diagram of the TSU.

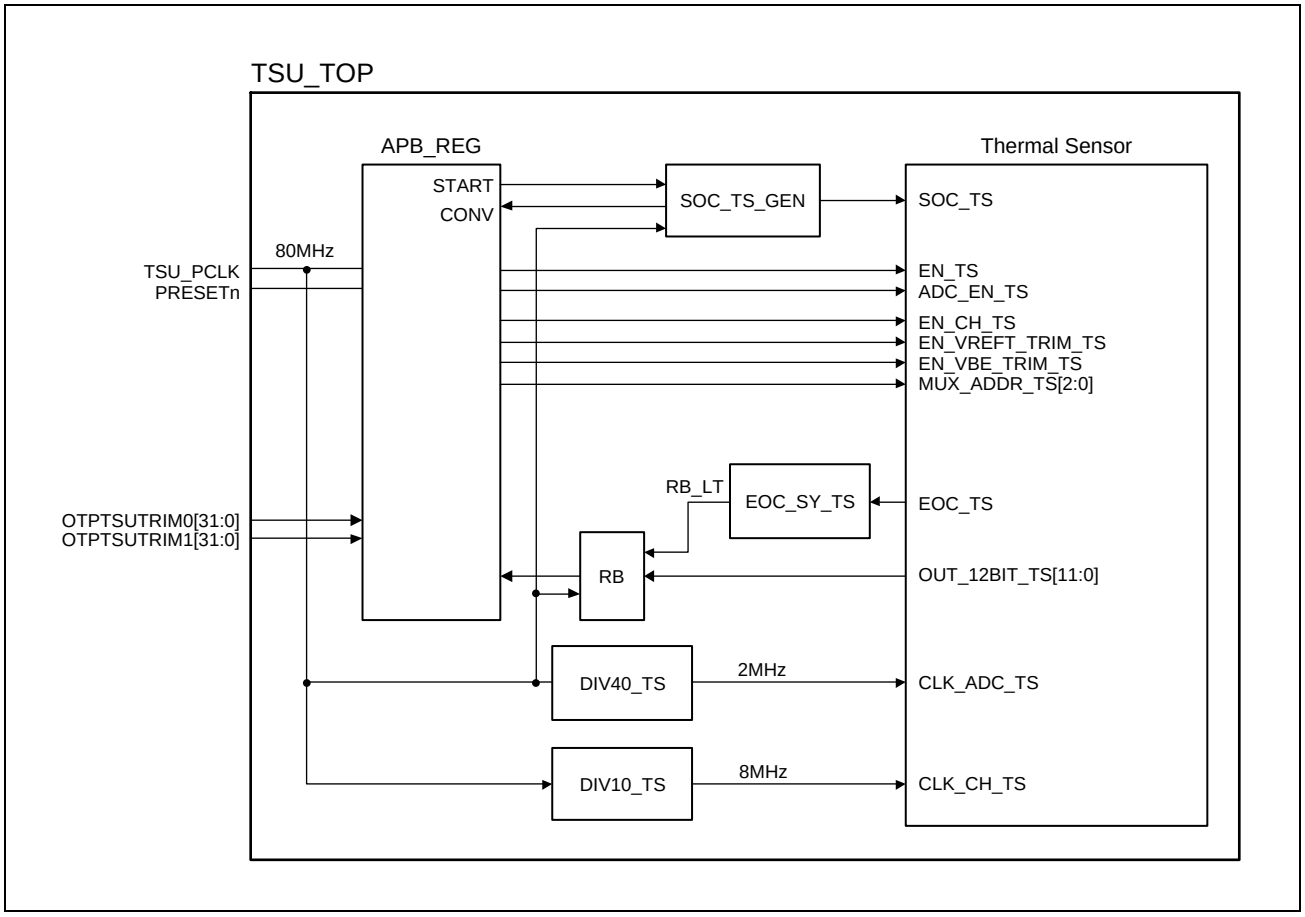



Figure 44.1 Block Diagram of the TSU

The TSU consists of the following blocks.

- APB_REG
- Thermal Sensor
- DIV40_TS and DIV10_TS
- SOC_TS_GEN
- EOC_SY_TS
- Read Buffer (RB)

#### 44.1.2.1 APB_REG

- The APB_REG block provides an external interface for the TSU.
- This block contains control and status registers for the thermal sensor.

#### 44.1.2.2 Thermal Sensor

- The thermal sensor block includes the internal probe, A/D converter and other items.
- The thermal sensor generates 12-bit digital values corresponding to the temperature and outputs them through the OUT_12BIT_TS[11:0] port.

#### 44.1.2.3 DIV40_TS and DIV10_TS

- The TSU operates with the TSU_PCLK clock (80 MHz).
- The DIV40_TS block is a divide-by-40 divider which divides TSU_PCLK to generate a 2-MHz clock frequency to be supplied to the A/D converter in the thermal sensor as CLK_ADC_TS.
- The DIV10_TS block is a divide-by-10 divider which divides TSU_PCLK to generate an 8-MHz clock frequency to be supplied to the thermal sensor core as CLK_CH_TS.

#### 44.1.2.4 SOC_TS_GEN

- The SOC_TS_GEN block generates the SOC_TS signal to be supplied to the thermal sensor for control over the timing of its operations.
- The SOC_TS_GEN block generates the SOC_TS signal every 20 µs. This is triggered by the START signal from the APB_REG block.
- The SOC_TS_GEN block generates the CONV signal that indicates the operating state of the thermal sensor. The value of the CONV signal can be read through the APB_REG block.

#### 44.1.2.5 EOC_SY_TS

- The EOC_SY_TS block detects the EOC_TS signal from the thermal sensor and generates the RB_LT signal in synchronization with TSU_PCLK.
- The EOC_TS signal indicates the end of A/D conversion of the voltage for the temperature measured by the thermal sensor.

#### 44.1.2.6 Read Buffer (RB)

- The RB block captures the 12-bit digital value from the OUT_12BIT_TS[11:0] port terminals of the thermal sensor in a register. This is triggered by the RB_LT signal from the EOC_SY_TS block. This value corresponds to the measured temperature.
- The value captured in the RB block can be read from the TSU through the APB_REG block.

### 44.1.3 External Pins

The TSU has no external pins.

### 44.1.4 Interrupts

The TSU has no interrupts.

## 44.2 Register Configuration

This section describes the registers for use in controlling the TSU.

Base Address: H'0_1005_9400 (Cortex-A55 Address Space)

### 44.2.1 List of Register

**Table 44.1** lists the TSU registers. All registers are accessible in 32-bit units.

Table 44.1 List of the TSU Registers

| Register Name | Abbreviation | R/W | Initial Value | Offset Address | Access Size |
|---|---|---|---|---|---|
| Sensor Mode Register | TSU_SM | R/W | H'0000_0000 | H'00 | 32 |
| Sensor Trigger Register | TSU_ST | R/W | H'0000_0000 | H'04 | 32 |
| Sensor ADC Data Register | TSU_SAD | R | H'0000_0000 | H'0C | 32 |
| Sensor Status Register | TSU_SS | R | H'0000_0000 | H'10 | 32 |
| TSU OTP Calibration Register 0 | OTPTSUTRIM0_REG | R | — | H'18 | 32 |
| TSU OTP Calibration Register 1 | OTPTSUTRIM1_REG | R | — | H'1C | 32 |

# 44.3 Register Descriptions

## 44.3.1 Sensor Mode Register (TSU_SM)

This register is used to specify the operating mode of the thermal sensor.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | ADC_EN_TS | ADC_EN_TS |
| 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 |
|---|---|---|---|---|
| 31 to 2 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 1 | ADC_EN_TS | 0b | R/W | These bits specify the operating mode of the thermal sensor. |
| 0 | EN_TS | 0b | R/W | |

| ADC_EN_TS | EN_TS | Operating mode of the thermal sensor |
|---|---|---|
| 0b | 0b | Power-saving mode |
| 0b | 1b | ADC power-saving mode |
| 1b | 0b | Power-saving mode |
| 1b | 1b | Normal operating mode |

### 44.3.2 Sensor Trigger Register (TSU_ST)

This register is used to start and stop the cycles of A/D conversion by the thermal sensor. The A/D conversion is repeated at 20-µs intervals once the conversion is started.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | RSV | START |
| 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 |
|---|---|---|---|---|
| 31 to 2 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 1 | RSV | 0b | R/W | The software may use this bit as a note.<br>The written value is read. |
| 0 | START | 0b | R/W | This bit starts and stops the cycles of A/D conversion by the thermal sensor.<br>0b: Stops the A/D conversion*2<br>1b: Starts the A/D conversion*1 |

Note 1. Before setting the START bit to 1b, be sure to set the TSU_SM register. Specifically, set the TSU_SM register, wait for 60 µs, and then set the START bit to 1b.

Note 2. When the START bit is to be set to 0b, be sure to read the A/D-converted value before making the setting.

### 44.3.3 Sensor ADC Data Register (TSU_SAD)

This register holds the digital value after A/D conversion of the voltage from the thermal sensor.

| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | OUT_12BIT_TS | | | | | | | | | | | |
| 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 12 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 11 to 0 | OUT_12BIT_TS | H'000 | R | These bits hold the digital value after A/D conversion of the voltage from the thermal sensor. The value is 12-bit unsigned. |

### 44.3.4 Sensor Status Register (TSU_SS)

This register indicates the state of A/D conversion in the thermal sensor.

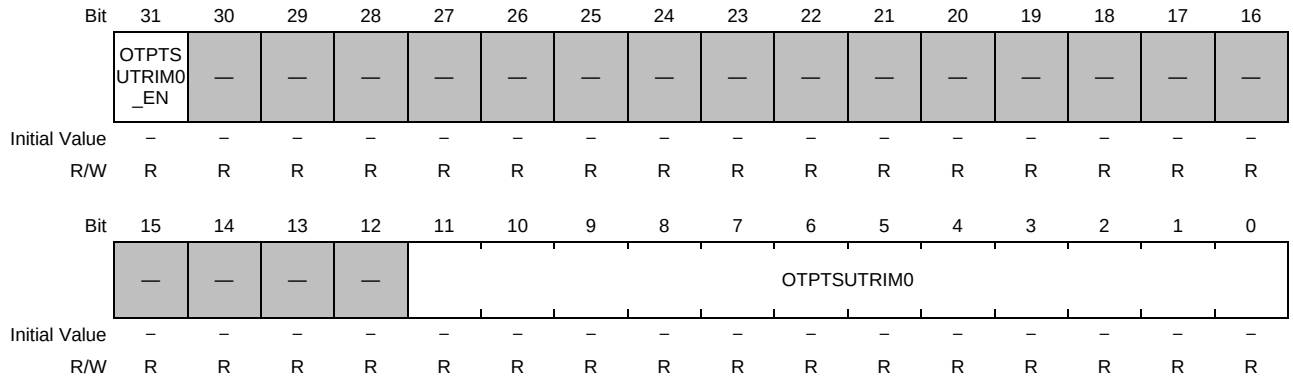
| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — | CONV |
| 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 1 | — | All 0 | R | Reserved<br>Whenever it is read, 0 is read. The written value will be ignored. |
| 0 | CONV | 0b | R | These bits indicate the state of A/D conversion in the thermal sensor.<br>0b: A/D conversion is stopped.<br>1b: A/D conversion is ongoing. |

### 44.3.5 TSU OTP Calibration Register 0 (OTPTSUTRIM0_REG)

This register holds the correction value for calibration to be used in calculating the temperature




| Bit | Bit Name | Initial Value | R/W | Description |
|---|---|---|---|---|
| 31 | OTPTSUTRIM0_EN | — | R | This bit indicates whether [11:0] OTPTSUTRIM0 is valid or invalid.<br>0b: [11:0] OTPTSUTRIM0 is invalid.<br>1b: [11:0] OTPTSUTRIM0 is valid. |
| 30 to 12 | — | — | R | Reserved<br>When read, the value is indefinite value. The written value will be ignored. |
| 11 to 0 | OTPTSUTRIM0 | — | R | These bits hold the correction value for calibration to be used in calculating the temperature. |

### 44.3.6 TSU OTP Calibration Register 1 (OTPTSUTRIM1_REG)

This register holds the correction value for calibration to be used in calculating the temperature.

| Bit | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | OTPTSUTRIM1_EN | — | — | — | — | — | — | — | — | — | — | — | — | — | — | — |
| 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 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | — | — | — | — | OTPTSUTRIM1 | | | | | | | | | | | |
| 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 | OTPTSUTRIM1_EN | — | R | This bit indicates whether [11:0] OTPTSUTRIM1 is valid or invalid.<br>0b: [11:0] OTPTSUTRIM1 is invalid.<br>1b: [11:0] OTPTSUTRIM1 is valid. |
| 30 to 12 | — | — | R | Reserved<br>When read, the value is indefinite value. The written value will be ignored. |
| 11 to 0 | OTPTSUTRIM1 | — | R | These bits hold the correction value for calibration to be used in calculating the temperature. |

## 44.4 Operation

### 44.4.1 Procedure for Starting the A/D Conversion

This section describes the procedure for starting the A/D conversion. Follow this procedure to acquire the A/D converted values for the measured temperatures.

1. Start supplying the clock to the TSU.
2. Release the TSU from the reset state.
3. Set the EN_TS and ADC_EN_TS bits in the TSU_SM register to 1b to place the thermal sensor in the normal operating mode.
4. Wait for 60 µs, leaving the value of the START bit as 0b.
5. Set the START bit in the TSU_ST register to 1b. The thermal sensor then starts the cycles of A/D conversion and the CONV bit in the TSU_SS register indicates 1b.
6. Once 20 µs have passed, the first A/D converted value from the thermal sensor is automatically stored in the TSU_SAD register. The value in that register is then updated every 20 µs. Reading the TSU_SAD register returns the latest A/D converted value.

**CAUTION**

Ensure that the setting of START is 1b when reading an A/D converted value from the TSU_SAD register.

### 44.4.2 Procedure for Stopping the A/D Conversion

This section describes the procedure for stopping the A/D conversion.

1. Set the START bit in the TSU_ST register to 0.
2. Poll the CONV bit in the TSU_SS register until it indicates 0b (A/D conversion has stopped).
3. Once you have confirmed that conversion has stopped, set the EN_TS and ADC_EN_TS bits in the TSU_SM register to 0b to place the thermal sensor in the power-saving mode.

**CAUTION**

If you intend to restart the A/D conversion after having stopped it, start from step 3 in **Section 44.4.1, Procedure for Starting the A/D Conversion**.

### 44.4.3 Confirming the State of the A/D Conversion by Monitoring the CONV Bit

The A/D conversion by the TSU can be started and stopped by the START bit in the TSU_ST register. However, the timing of stopping the A/D conversion by the thermal sensor is not synchronous with the START bit being set to 0 (stopping the A/D conversion). Monitor the CONV bit in the TSU_SS register to confirm that the A/D conversion by the thermal sensor has stopped.

### 44.4.4 Procedure for Measuring the Temperature

To calculate the temperature Tj, read registers TSU_SAD, OTPTSUTRIM0_REG, and OTPTSUTRIM1_REG, and follow the procedure below.

1. Read the value of the TSU_SAD register eight times every at least 20 µs. The read values are defined as TSCode[0] to TSCode[7].

2. Calculate the average of the values read, TSCodeAVE, by using the following formula.

$$\mathrm{TSCodeAVE} = \frac{(\mathrm{TSCode}[0] + \cdots + \mathrm{TSCode}[7])}{8}$$

3. Apply curvature correction and calculate DSENSOR by using the following formula.

$$\mathrm{DSENSOR} = \frac{\mathrm{TSCodeAVE}}{(1 + \mathrm{TSCodeAVE} \times 0.000013)}$$

4. Read the correction values for use in calibration from registers OTPTSUTRIM0_REG and OTPTSUTRIM1_REG and use the following formula to calculate Tj.

$$\mathrm{Tj} = (\mathrm{DSENSOR} - \mathrm{OTPTSUTRIM1}) \times \left(\frac{165}{(\mathrm{OTPTSUTRIM0} - \mathrm{OTPTSUTRIM1})}\right) - 40$$

## 44.5 Usage Note

When the START bit in the TSU_ST register is to be set to 0b, be sure to read the A/D-converted value in the TSU_SAD register before making the setting.

# 48. Electrical Characteristics

## 48.1 Absolute Maximum Ratings

Table 48.1 Absolute Maximum Ratings

| Item | | Symbol | Value | Unit |
|---|---|---|---|---|
| Power supply voltage (3.3 V) | | $PV_{DD}$<br>$USB_V_{DD33}$ | −0.5 to +3.8 | V |
| Power supply voltage (1.8-V/3.3-V switchable) | | $SD0_PV_{DD}$ | −0.5 to +3.8 | V |
| | | $SPI_PV_{DD}$ | | |
| Power supply voltage (1.8 V) | | $V_{DD18}$ | −0.5 to +2.5 | V |
| | | $PLL1_AV_{DD18}$ | | |
| | | $PLL23_AV_{DD18}$ | | |
| | | $PLL4_AV_{DD18}$ | | |
| | | $PLL5_AV_{DD18}$ | | |
| | | $DSI_V_{DD18}$ | | |
| | | $USB_V_{DD18}$ | | |
| | | $JTAG_PV_{DD}$ | | |
| | | $OTP_V_{DD18}$ | | |
| DDR power supply voltage (DDR3L) | | DDR_VDDQ1,<br>DDR_VDDQ2 | −0.5 to +2.5 | V |
| Power supply voltage (1.1-V) | | $V_{DD}$ | −0.5 to +1.5 | V |
| | | $PLL23_DV_{DD11}$ | | |
| | | $PLL5_DV_{DD11}$ | | |
| Input voltage | 3.3-V I/O input pins | — | −0.3 to 3.3-V power supply ($PV_{DD}$, $USB_V_{DD33}$) + 0.3 | V |
| | 1.8-V/3.3-V switchable I/O input pins | — | −0.3 to 1.8-V/3.3-V switchable power supply ($SD0_PV_{DD}$, $SPI_PV_{DD}$) + 0.3 | V |
| | 1.8-V I/O input pins | — | −0.3 to 1.8-V power supply ($V_{DD18}$, $DSI_V_{DD18}$, $USB_V_{DD18}$) + 0.3 | V |
| Operating temperature | Ambient temperature | $T_a$ | −40 to +85 *1 | °C |
| | Junction temperature | $T_j$ | −40 to +110 | °C |
| Storage temperature | Ambient temperature | $T_{stg}$ | −40 to +150 | °C |

Note 1. If wider temp is required than this range, use case has to be investigated.

## 48.2 Power Supply

Table 48.2 Power Supply

| Item | Symbol | Min. | Typ. | Max. | Unit | Remarks |
|---|---|---|---|---|---|---|
| Power supply voltage (3.3 V) | $PV_{DD}$ | 2.97 | 3.30 | 3.63 | V | — |
| Power supply voltage (1.8 V) | $V_{DD18}$ | 1.62 | 1.80 | 1.98 | V | — |
| | JTAG_PVDD | 1.62 | 1.80 | 1.98 | V | — |
| | OTP_$V_{DD18}$ | 1.65 | 1.80 | 1.95 | V | — |
| Power supply voltage (1.1 V) | $V_{DD}$ | 1.05 | 1.10 | 1.15 | V | — |
| Power supply voltage (USB) | USB_$V_{DD33}$ | 3.00 | 3.30 | 3.60 | V | — |
| | USB_$V_{DD18}$ | 1.65 | 1.80 | 1.95 | V | — |
| Power supply voltage (SD) | SD0_$PV_{DD}$ | 2.97 | 3.30 | 3.63 | V | When 3.3 V is supplied |
| | | 1.70 | 1.80 | 1.95 | V | When 1.8 V is supplied |
| Power supply voltage (SPI) | SPI_$PV_{DD}$ | 2.97 | 3.30 | 3.63 | V | When 3.3 V is supplied |
| | | 1.70 | 1.80 | 1.95 | V | When 1.8 V is supplied |
| Power supply voltage (DSI) | DSI_$V_{DD18}$ | 1.65 | 1.80 | 1.95 | V | — |
| Power supply voltage (DDR3L) | DDR_VDDQ1, DDR_VDDQ2 | 1.283 | 1.35 | 1.45 | V | |
| Power supply voltage (PLL) | PLL1_$AV_{DD18}$, PLL23_$AV_{DD18}$, PLL4_$AV_{DD18}$, PLL5_$AV_{DD18}$ | 1.62 | 1.80 | 1.98 | V | — |
| | PLL23_$DV_{DD11}$, PLL5_$DV_{DD11}$ | 1.05 | 1.10 | 1.15 | V | — |

## 48.3 Power-On/Power-Off Sequence

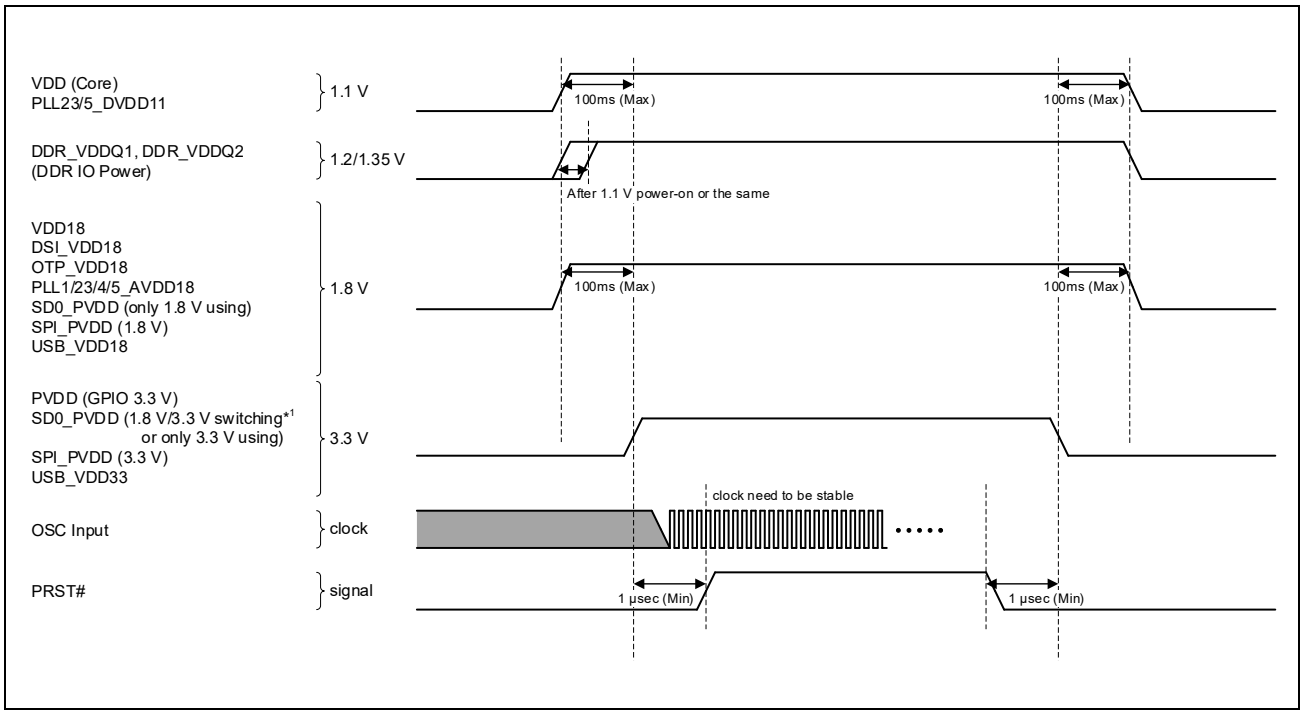


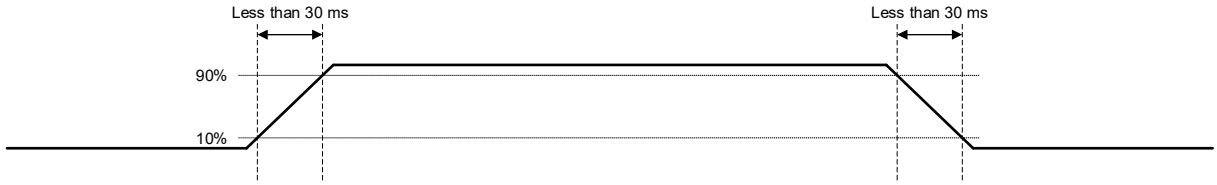

Figure 48.1 Power-On/Power-Off Sequence

NOTES

1. Turn on 3.3 V after 1.1 V/1.8 V.

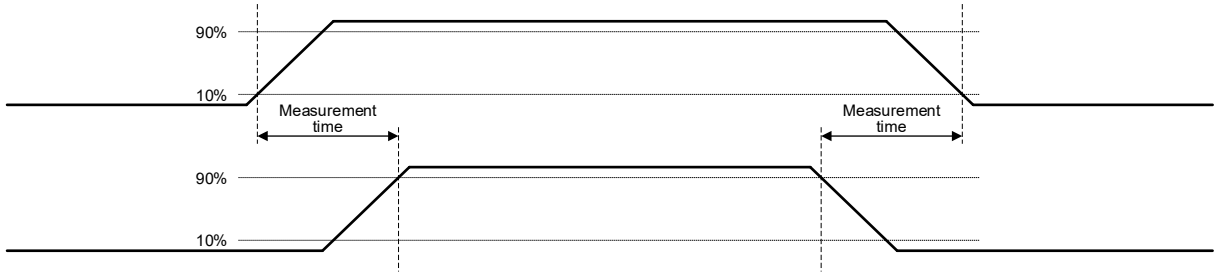
   *Note 1.* About SD0_$PV_{DD}$, especially in the case of switching between 1.8 V and 3.3 V at same power rail, 1.8-V power-on/off timing can also follow 3.3-V power rail sequence as shown in the Figure 48.1.

2. Turn on the DDR IO power supply at the same time as or after the 1.1-V power supply.
3. From first power rising start to last power rising end must be within 100 ms.
4. The power-off sequence is the reverse of the power-on sequence
   (PRST# → Low ⇨ 3.3 V = OFF ⇨ Other = OFF)
5. PRST# should be changed from Low to High after the 3.3-V power supply is turned on, after 1 µsec, and after the input clock from the oscillator stabilizes.




Note: The rise/fall time of each power supply should be within 30 msec.
Start point / end point of rising / falling time of each power supply should be 10% or 90% level of each power supply.

Figure 48.2 Power Up Time/Power Down Time (1)




Note: In case of there is restriction of sequence between different power supply, measurement time is starting point that is 10% of earlier rising power voltage level to end point that is 90% of later rising power voltage level, and starting point that is 90% of earlier falling power voltage level to end point that is 10% of later falling power voltage level.

Figure 48.3 Power Up Time/Power Down Time (2)

## 48.4 DC Characteristics

Table 48.3 DC Characteristics (1) [3.3-V I/O]

| Item | | Symbol | Min. | Typ. | Max. | Unit | Remarks |
|---|---|---|---|---|---|---|---|
| High-level input voltage | | $V_{IH}$ | 2 | — | $PV_{DD}$ + 0.3 | V | |
| Low-level input voltage | | $V_{IL}$ | −0.3 | — | 0.8 | V | |
| Hysteresis threshold ↑ | | $V_T$+ | 0.9 | — | 2.1 | V | |
| Hysteresis threshold ↓ | | $V_T$− | 0.7 | — | 1.9 | V | |
| Input hysteresis voltage | | $V_{HYS}$ | 0.306 | — | 0.420 | V | |
| Output logic high voltage | ($I_{OH}$ = –2mA) | $V_{OH}$ | $PV_{DD}$ – 0.4 | — | $PV_{DD}$ | V | |
| | ($I_{OH}$ = –4mA) | | | | | | |
| | ($I_{OH}$ = –8mA) | | | | | | |
| | ($I_{OH}$ = –12mA) | | | | | | |
| Output logic low voltage | ($I_{OL}$ = 2mA) | $V_{OL}$ | 0 | — | 0.4 | V | |
| | ($I_{OL}$ = 4mA) | | | | | | |
| | ($I_{OL}$ = 8mA) | | | | | | |
| | ($I_{OL}$ = 12mA) | | | | | | |
| Weak pull-up resistor (input mode) | | $R_{UP}$ | 7K | — | 100K | Ω | |
| Weak pull-down resistor (input mode) | | $R_{DN}$ | 7K | — | 100K | Ω | |

Note 1. Schmitt can be used at only RIIC mode.

Table 48.4 DC Characteristics (2) [1.8-V I/O]

| Item | | Symbol | Min. | Typ. | Max. | Unit | Remarks |
|---|---|---|---|---|---|---|---|
| High-level input voltage | | $V_{IH}$ | 0.65 × $V_{DD18}$ | — | $V_{DD18}$ + 0.3 | V | |
| Low-level input voltage | | $V_{IL}$ | −0.3 | — | 0.35 × $V_{DD18}$ | V | |
| Output logic high voltage | ($I_{OH}$ = –2mA) | $V_{OH}$ | $V_{DD18}$ – 0.4 | — | $V_{DD18}$ | V | |
| | ($I_{OH}$= –4mA) | | | | | | |
| | ($I_{OH}$ = –8mA) | | | | | | |
| | ($I_{OH}$ = –12mA) | | | | | | |
| Output logic low voltage | ($I_{OL}$ = 2mA) | $V_{OL}$ | 0 | — | 0.4 | V | |
| | ($I_{OL}$ = 4mA) | | | | | | |
| | ($I_{OL}$ = 8mA) | | | | | | |
| | ($I_{OL}$ = 12mA) | | | | | | |

Table 48.5 DC Characteristics (3) [3.3-V Input]

| Item | Symbol | Min. | Typ. | Max. | Unit | Remarks |
|---|---|---|---|---|---|---|
| High-level input voltage | $V_{IH}$ | 2.3 | — | $PV_{DD}$ + 0.3 | V | |
| Low-level input voltage | $V_{IL}$ | −0.3 | — | 0.8 | V | |
| Hysteresis threshold ↑ | $V_T$+ | 0.9 | — | 2.1 | V | |
| Hysteresis threshold ↓ | $V_T$− | 0.7 | — | 1.9 | V | |
| Input hysteresis voltage | $V_{HYS}$ | 0.306 | — | 0.420 | V | |

Table 48.6 DC Characteristics (4) [1.8-V Input]

| Item | Symbol | Min. | Typ. | Max. | Unit | Remarks |
|---|---|---|---|---|---|---|
| High-level input voltage | $V_{IH}$ | 0.65 × $V_{DD18}$ | — | $V_{DD18}$ + 0.3 | V | |
| Low-level input voltage | $V_{IL}$ | −0.3 | — | 0.35 × $V_{DD18}$ | V | |

Table 48.7 DC Characteristics (5) [3.3 V I/O (SD, QSPI)]

| Item | | Symbol | Min. | Typ. | Max. | Unit | Remarks |
|---|---|---|---|---|---|---|---|
| High-level input voltage | | $V_{IH}$ | 0.625 × SDn_$PV_{DD}$ | — | SDn_$PV_{DD}$ + 0.3 | V | |
| Low-level input voltage | | $V_{IL}$ | SDn_$PV_{DD}$ − 0.3 | — | 0.25 × SDn_$PV_{DD}$ | V | |
| Output logic high voltage | (× 0.5) | $V_{OH}$ | 0.75 × SDn_$PV_{DD}$ | — | SDn_$PV_{DD}$ | V | |
| | (× 0.75) | | | | | | |
| | (× 1.0) | | | | | | |
| | (× 1.5) | | | | | | |
| Output logic low voltage | (× 0.5) | $V_{OL}$ | 0 | — | 0.125 × SDn_$PV_{DD}$ | V | |
| | (× 0.75) | | | | | | |
| | (× 1.0) | | | | | | |
| | (× 1.5) | | | | | | |

Note: n = 0

Table 48.8 DC Characteristics (6) [1.8 V I/O (SD, QSPI)]

| Item | | Symbol | Min. | Typ. | Max. | Unit | Remarks |
|---|---|---|---|---|---|---|---|
| High-level input voltage | | $V_{IH}$ | 1.27 | — | 2.00 | V | |
| Low-level input voltage | | $V_{IL}$ | $SDn_PV_{DD}$ − 0.3 | — | 0.58 | V | |
| Output logic high voltage | (× 0.5) | $V_{OH}$ | 0.8 × $SDn_PV_{DD}$ | — | $SDn_PV_{DD}$ | V | |
| | (× 0.75) | | | | | | |
| | (× 1.0) | | | | | | |
| | (× 1.5) | | | | | | |
| Output logic low voltage | (× 0.5) | $V_{OL}$ | 0 | — | 0.2 × $SDn_PV_{DD}$ | V | |
| | (× 0.75) | | | | | | |
| | (× 1.0) | | | | | | |
| | (× 1.5) | | | | | | |

Note: n = 0

Table 48.9 DC Characteristics(7) [$I^2C$ Open Drain 3.3 V I/O]

| Item | Symbol | Min. | Typ. | Max. | Unit | Remarks |
|---|---|---|---|---|---|---|
| External pull-up supply | $V_{VDDP}$ | 2.7 | — | $PV_{DD}$ | V | |
| Input high voltage | $V_{IH}$ | 0.7 × $V_{VDDP}$ | — | $V_{VDDP}$ + 0.5 | V | |
| Input Low Voltage | $V_{IL}$ | −0.5 | — | 0.3 × $V_{VDDP}$ | V | |
| Low level output current | $I_{OL}$ | 20 | — | — | mA | $V_{OL}$ = 0.4 V |
| Low level output voltage | $V_{OL}$ | — | — | 0.4 | V | $V_{VDDP}$ > 2 V |
| Input hysteresis | $V_{HYST}$ | 0.1 × $V_{VDDP}$ | — | — | mV | |

Table 48.10 DC Characteristics (8) [1.35-V Input (DDR3L)]

| Item | Symbol | Min. | Typ. | Max. | Unit | Remarks |
|---|---|---|---|---|---|---|
| DC input high level | $V_{IH}$(DC) | $V_{REF}$ + 0.09 | — | — | V | |
| DC input low level | $V_{IL}$(DC) | — | — | $V_{REF}$ − 0.09 | | $V_{REF}$ =0.5 x $DDR_V_{DDQ}$ |
| AC input high level | $V_{IH}$(AC) | $V_{REF}$ + 0.135 | — | — | V | |
| AC input low level | $V_{IL}$(AC) | — | — | $V_{REF}$ − 0.135 | V | $V_{REF}$ =0.5 x $DDR_V_{DDQ}$ |
| DC differential input high | $V_{IHdiff}$ | 0.18 | — | — | V | |
| DC differential input low | $V_{ILdiff}$ | — | — | −0.18 | V | |
| AC differential input high | $V_{IHdiff}$(AC) | 2 × ($V_{IH}$(AC) − $V_{REF}$) | — | — | V | |
| AC differential input low | $V_{ILdiff}$(AC) | — | — | 2 × ($V_{REF}$ − $V_{IL}$(AC)) | V | $V_{REF}$ =0.5 x $DDR_V_{DDQ}$ |
| Differential input cross point voltage relative to 0.5 × DDR_VDDQ for DQS | $V_{IX}$(DQS) | −0.075 | — | 0.075 | V | |

Table 48.11 DC Characteristics (9) [1.35-V Output (DDR3L)]

| Item | Symbol | Min. | Typ. | Max. | Unit | Remarks |
|---|---|---|---|---|---|---|
| DC output high measurement level (for IV curve linearity) | $V_{OH}$(DC) | — | 0.8 × DDR_$V_{DDQ}$ | — | V | The swing of ± 0.1 × DDR_$V_{DDQ}$ is based on approximately 50% of the static single-ended output high or low swing with a driver impedance of 40Ω and an effective test load of 25Ω to $V_{TT}$ = DDR_$V_{DDQ}$/2. |
| DC output mid measurement level (for IV curve linearity) | $V_{OM}$(DC) | — | 0.5 × DDR_$V_{DDQ}$ | — | V | |
| DC output low measurement level (for IV curve linearity) | $V_{OL}$(DC) | — | 0.2 × DDR_$V_{DDQ}$ | — | V | |
| AC output high measurement level (for output slew rate) | $V_{OH}$(AC) | — | VTT + 0.1 × DDR_$V_{DDQ}$ | — | V | |
| AC output low measurement level (for output slew rate) | $V_{OL}$(AC) | — | VTT − 0.1 × DDR_$V_{DDQ}$ | — | V | |

Table 48.12 DC Characteristics (10) [DDR3L Cross Point Voltage For Differential Output Signals (CK/DQS)]

| Item | Symbol | Min. | Typ. | Max. | Unit | Remarks |
|---|---|---|---|---|---|---|
| Differential output cross point voltage relative to 0.5 × DDR_VDDQ | $V_{OX}$(CK/DQS) | −0.1 | — | 0.1 | V | |

Table 48.13 DC Characteristics (11) [USB 2.0]

| Item | Symbol | Min. | Typ. | Max. | Unit | Remarks |
|---|---|---|---|---|---|---|
| Input levels for low/full speed | | | | | | |
| High (driven) | $V_{IH}$ | 2.0 | — | — | V | |
| Low | $V_{IL}$ | — | — | 0.8 | V | |
| Differential input sensitivity | $V_{DI}$ | 0.2 | — | — | V | |
| Differential common mode range | $V_{CM}$ | 0.8 | — | 2.5 | V | |
| Input levels for high speed | | | | | | |
| High-speed squelch detection threshold (differential signal amplitude) | $V_{HSSQ}$ | 100 | — | 150 | mV | |
| High-speed data signaling common mode voltage range (guideline for receiver) | $V_{HSCM}$ | −50 | — | 500 | mV | |
| Output levels for low/full speed | | | | | | |
| Low | $V_{OL}$ | 0.0 | — | 0.3 | V | |
| High (driven) | $V_{OH}$ | 2.8 | — | 3.6 | V | |
| Output signal crossover voltage | $V_{CRS}$ | 1.3 | — | 2.0 | V | |
| Output levels for high-speed | | | | | | |
| High-speed idle level | $V_{HSOI}$ | −10.0 | — | 10.0 | mV | |
| High-speed data signaling high | $V_{HSOH}$ | 360 | — | 440 | mV | |
| High-speed data signaling low | $V_{HSOL}$ | −10.0 | — | 10.0 | mV | |
| Chirp J level (differential voltage) | $V_{CHIRPJ}$ | 700 | — | 1100 | mV | |
| Chirp K level (differential voltage) | $V_{CHIRPK}$ | −900 | — | −500 | mV | |

Table 48.14 DC Characteristics (12) [Current Consumption]

| Item | Symbol | Max Current | Unit | Remarks |
|---|---|---|---|---|
| Power Supply voltage(3.3V) | PVDD | 690.00 | mA | PVDD = 3.63V |
| Power Supply voltage(1.8V) | JTAG_PVDD | 10.00 | mA | JTAG_PVDD = 1.98V |
| | OTP_VDD18 | 20.00 | mA | OTP_VDD18 = 1.95V |
| | VDD18 | 10.00 | mA | VDD18 = 1.98V |
| Power Supply voltage(1.1V) | VDD | 1870.00 | mA | VDD = 1.15V |
| Power supply voltage (USB/3.3V) | USB_VDD33 | 20.00 | mA | USB_VDD33 = 3.60V |
| Power supply voltage (USB/1.8V) | USB_VDD18 | 140.00 | mA | USB_VDD18 = 1.95V |
| Power supply voltage (SD/3.3V) | SD0_PVDD | 220.00 | mA | SD0_PVDD = 3.63V |
| Power supply voltage (SD/1.8V) | SD0_PVDD | 70.00 | mA | SD0_PVDD = 1.95V |
| Power supply voltage (SPI/3.3V) | SPI_PVDD | 50.00 | mA | SPI_PVDD = 3.63V |
| Power supply voltage (SPI/1.8V) | SPI_PVDD | 90.00 | mA | SPI_PVDD = 1.95V |
| Power supply voltage (DSI/1.8V) | DSI_VDD18 | 100.00 | mA | DSI_VDD18 = 1.95V |
| Power supply voltage (DDR/1.35V) | DDR_VDDQ1 | 410.00 | mA | DDR_VDDQ1 = 1.45V |
| | DDR_VDDQ2 | | | DDR_VDDQ2 = 1.45V |
| Power supply voltage (PLL/1.8V) | PLL1_AVDD18 | 10.00 | mA | PLL1_AVDD18 = 1.98V |
| | PLL23_AVDD18 | 20.00 | mA | PLL23_AVDD18 = 1.98V |
| | PLL4_AVDD18 | 10.00 | mA | PLL4_AVDD18 = 1.98V |
| | PLL5_AVDD18 | 10.00 | mA | PLL5_AVDD18 = 1.98V |
| Power supply voltage (PLL/1.1V) | PLL23_DVDD11 | 20.00 | mA | PLL23_DVDD11 =1.15V |
| | PLL5_DVDD11 | 10.00 | mA | PLL5_DVDD11 =1.15V |

Note: Tj : Max Condition

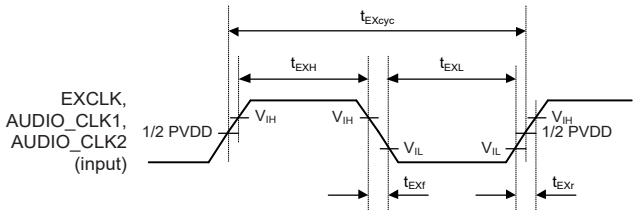
## 48.5 AC Characteristics

Conditions: $V_{DD}$ = PLL23_$DV_{DD11}$ = PLL5_$DV_{DD11}$ = 1.1 ± 0.05 V,
PLL1_$AV_{DD18}$ = PLL23_$AV_{DD18}$ = PLL4_$AV_{DD18}$ = 1.8 ± 0.18 V,
$V_{DD18}$ = JTAG_$PV_{DD}$ = 1.8 ± 0.18 V, OTP_$V_{DD18}$ = 1.8 V ± 0.15 V,
USB_$V_{DD18}$ = DSI_$V_{DD18}$ = 1.8 ± 0.15 V,
$PV_{DD}$ = 3.3 ± 0.33 V, USB_$V_{DD33}$ = 3.3 ± 0.3 V, DDR_$V_{DDQ}$ = 1.2 ± 0.06 V/1.283 to 1.45 V,
SPI_$PV_{DD}$ = SDn_$PV_{DD}$ (n = 0) = 3.3 ± 0.33 V/1.70 to 1.95 V, $V_{SS}$ = 0 V,
$T_a$ = −40 to +85°C, $T_j$ = −40 to +110°C

### 48.5.1 Clock Timing

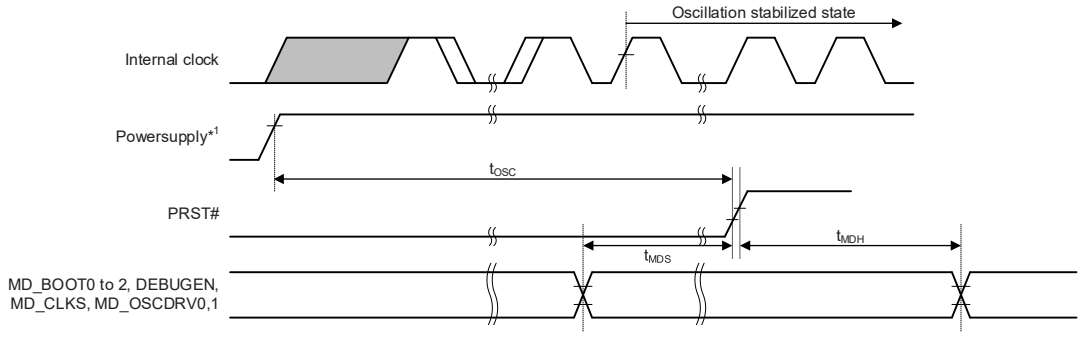
Table 48.15 Clock Timing Table

| Item | Symbol | Min. | Max. | Unit | Figures |
|---|---|---|---|---|---|
| EXCLK clock input frequency | $f_{EX}$ | 24 – 100ppm | 24 + 100ppm | MHz | Figure 48.4 |
| EXCLK clock input cycle time | $t_{EXcyc}$ | 41.67 | 41.67 | ns | |
| AUDIO_CLK1, AUDIO_CLK2 clock input frequency (external clock is input) | $f_{EX}$ | 10 | 50 | MHz | |
| AUDIO_CLK1, AUDIO_CLK2 clock input cycle time (external clock is input) | $t_{EXcyc}$ | 20 | 100 | ns | |
| EXCLK, AUDIO_CLK1, AUDIO_CLK2 clock input low level pulse width | $t_{EXL}$ | 0.4 | 0.6 | $t_{EXcyc}$ | |
| EXCLK, AUDIO_CLK1, AUDIO_CLK2 clock input high level pulse width | $t_{EXH}$ | 0.4 | 0.6 | $t_{EXcyc}$ | |
| EXCLK, AUDIO_CLK1, AUDIO_CLK2 clock input rise time | $t_{EXr}$ | — | 4 | ns | |
| EXCLK, AUDIO_CLK1, AUDIO_CLK2 clock input fall time | $t_{EXf}$ | — | 4 | ns | |
| Oscillator stabilization time | $t_{OSC}$ | — | 1 | ms | Figure 48.5, Figure 48.6 |
| Mode hold time | $t_{MDH}$ | — | 100 | ns | |
| Mode setup time | $t_{MDS}$ | — | 100 | ns | |



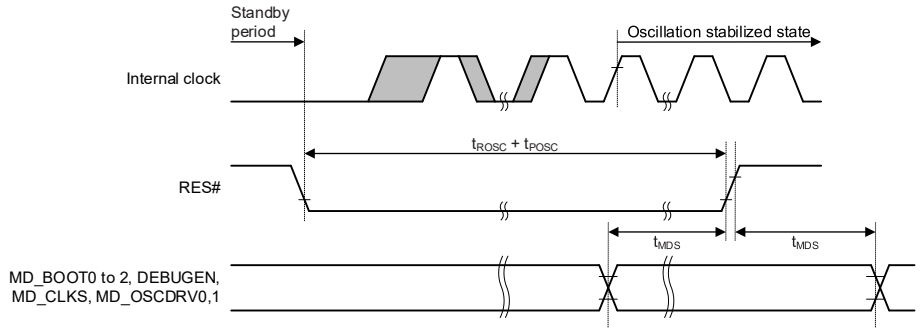

Note: When the clock is input on the EXCLK, AUDIO_CLK1 or AUDIO_CLK2

Figure 48.4 EXCLK, AUDIO_CLK1 and AUDIO_CLK2 Clock Input Timing




Note 1. $V_{DD}$, PLLn_$DV_{DD11}$, PLLn_$AV_{DD18}$, $V_{DD18}$, JTAG_$PV_{DD}$, OTP_$V_{DD18}$, USB_$V_{DD18}$, DSI_$V_{DD18}$, $PV_{DD}$,USB_$V_{DD33}$, DDR_$V_{DDQ}$, SPI_$PV_{DD}$, SDn_$PV_{DD}$ (n = 0)

Figure 48.5 Power-On Oscillation Settling Time




Note: Oscillation setting time when the internal oscillator is used.

Figure 48.6 Oscillation Settling Time on Return from Standby (Return by Reset)

## 48.5.2 SDHI Access Timing

### 48.5.2.1 SDHI Access Timing (SDR 3.3-V)

Table 48.16 SDHC AC Access Timing (SDR at 3.3-V Operation)

| Item | Symbol | Default Speed Mode (16.67 MHz) | | High Speed Mode (33.33 MHz) | | Unit | Figures |
|---|---|---|---|---|---|---|---|
| | | Min. | Max. | Min. | Max. | | |
| SD_CLK clock cycle | $t_{SDCYC}$ | 60.00 | — | 30.0 | — | ns | Figure 48.7 |
| SD_CLK clock high level width | $t_{SDWH}$ | 23.50 | — | 13.50 | — | ns | |
| SD_CLK clock low level width | $t_{SDWL}$ | 23.50 | — | 13.50 | — | ns | |
| SD_CLK clock rise time | $t_{SDLH}$ | — | 10 | — | 3 | ns | |
| SD_CLK clock fall time | $t_{SDHL}$ | — | 10 | — | 3 | ns | |
| SD_CMD,SD_DATA output delay | $t_{SDODLY}$ | −4.50 | 4.0 | −4.50 | 4.0 | ns | |
| SD_CMD,SD_DATA input set up time | $t_{SDIS}$ | 5.5 | — | 5.5 | — | ns | |
| SD_CMD,SD_DATA input hold time | $t_{SDIH}$ | 2.0 | — | 2.0 | — | ns | |
| SD_CMD,SD_DATA input data width | $t_{SDIDW}$ | — | — | — | — | ns | |

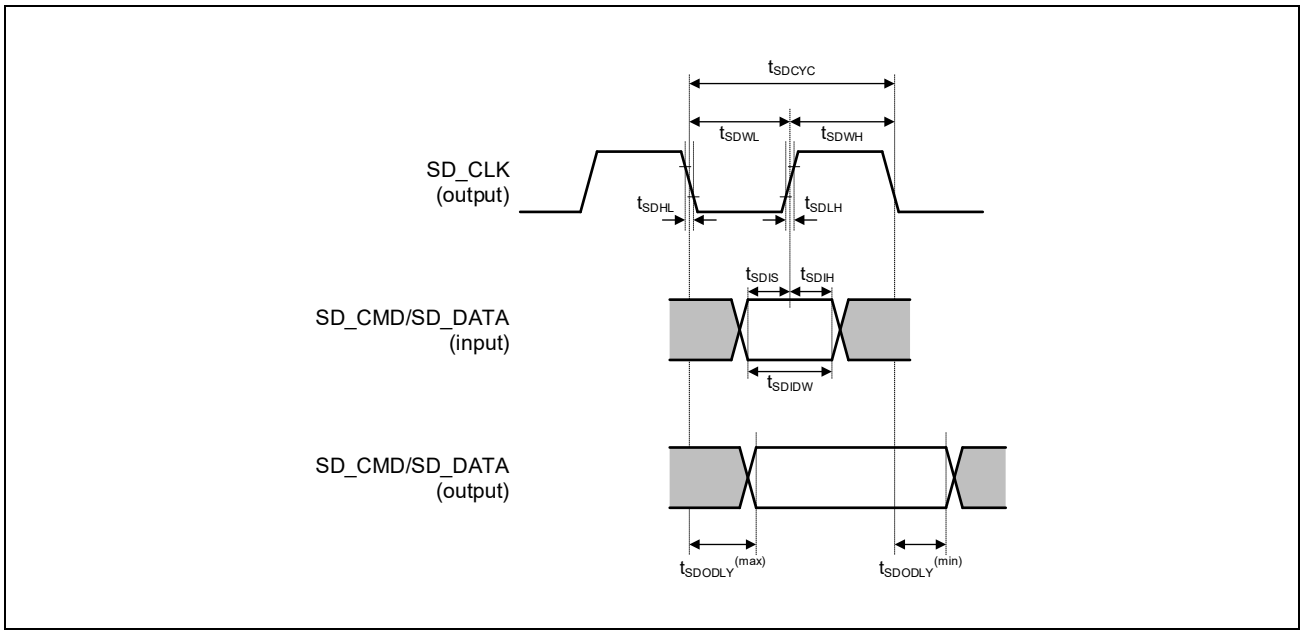



Figure 48.7 SDHC Interface Timing (SDR 3.3-V Power Supply)

NOTE

The disclosure of other characteristics of the SD interface needs the conclusion of the following agreement.

- SD Host/Ancillary Product License Agreement (SD HALA)

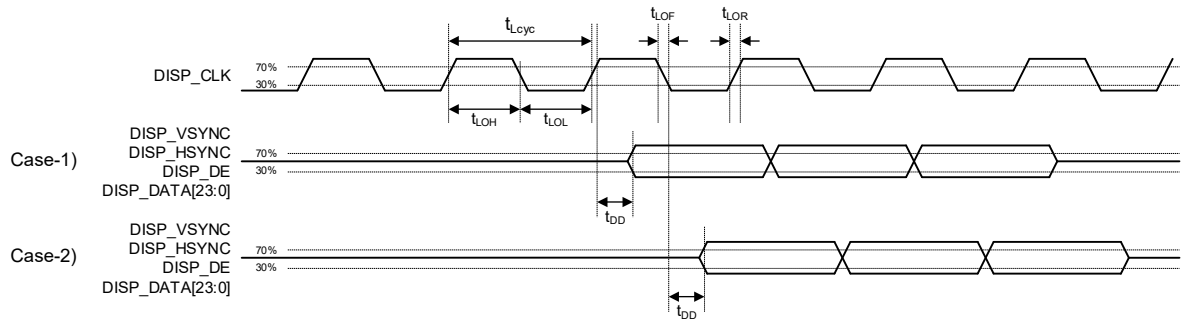
For details, contact your local sales representatives.

### 48.5.3 LCDC Access Timing

Table 48.17 LCDC AC Access Timing

| Item | Symbol | Min. | Max. | Unit | Figures |
|---|---|---|---|---|---|
| DCLK period | $t_{Lcyc}$ | 11.5 | 172.33 | ns | **Figure 48.8** |
| DCLK low pulse width | $t_{LOL}$ | $t_{Lcyc}/2 - 1.06$ | $t_{Lcyc}/2 + 1.06$ | ns | |
| DCLK high pulse width | $t_{LOH}$ | $t_{Lcyc}/2 - 1.06$ | $t_{Lcyc}/2 + 1.06$ | ns | |
| DCLK rise time | $t_{LOR}$ | — | 3 | ns | |
| DCLK fall time | $t_{LOF}$ | — | 3 | ns | |
| Data output dealy | $t_{DD}$ | −1.5 | 1.5 | ns | |

**Note:** AC access timing condition: SR = 1 (fast) / Drive capacity setting 12 mA / Load capacity 30 pF.




**Note:** *Case-1*) DISP_CLK forward rotation output mode
When DPI_CLKMD bit in DU_DITR0 register is 0.

*Case-2*) DISP_CLK inverted output mode
When DPI_CLKMD bit in DU_DITR0 register is 1.

Figure 48.8 LCDC AC Access Timing

## 48.5.4 USB 2.0 Host/Function Module Access Timing

### 48.5.4.1 USB 2.0 Low-Speed Access Timing

Table 48.18 USB Transceiver Timing (Low-Speed)

| Item | Symbol | Min. | Max. | Unit | Figures |
|---|---|---|---|---|---|
| Rise time | $t_{LR}$ | 75 | 300 | ns | Figure 48.9 |
| Fall time | $t_{LF}$ | 75 | 300 | ns | |
| Rise/fall time lag | $t_{LR}/t_{LF}$ | 80 | 125 | % | |

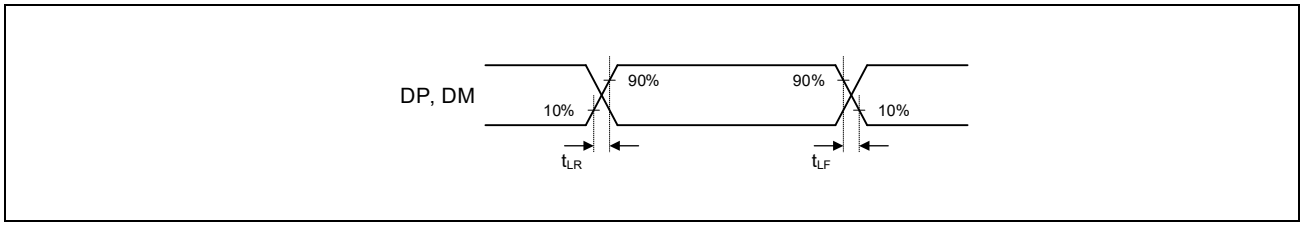



Figure 48.9 USB0_DP, USB1_DP, USB0_DM, and USB1_DM Output Timing (Low-Speed)

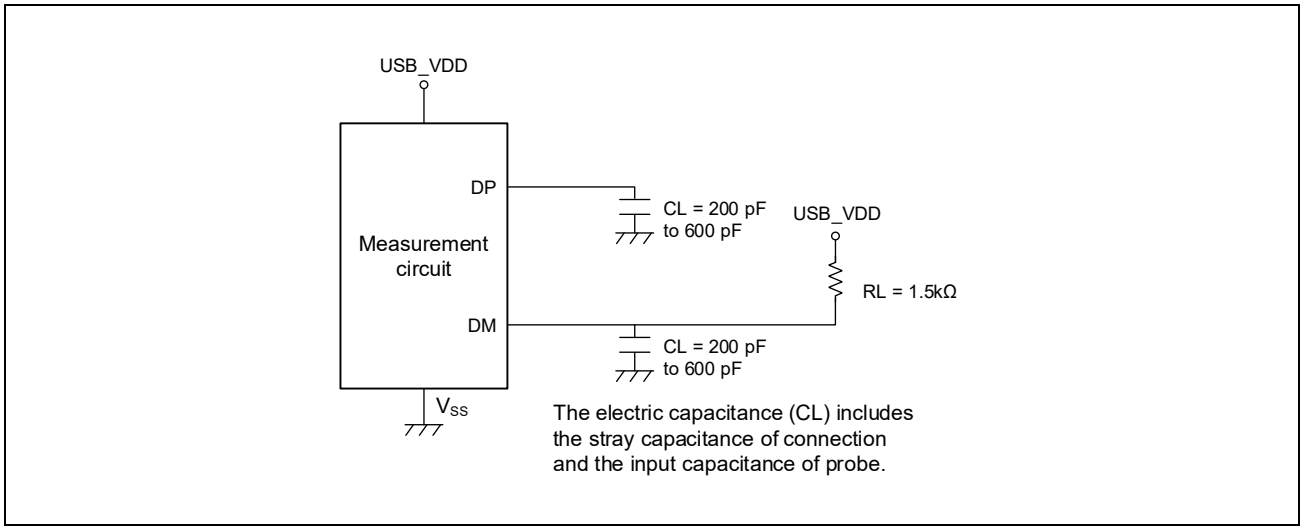



Figure 48.10 Measurement Circuit (Low-Speed)

### 48.5.4.2 USB 2.0Full-Speed Access Timing

Table 48.19 USB Transceiver Timing (Full-Speed)

| Item | Symbol | Min. | Max. | Unit | Figures |
|---|---|---|---|---|---|
| Rise time | $t_{FR}$ | 4 | 20 | ns | Figure 48.11 |
| Fall time | $t_{FF}$ | 4 | 20 | ns | |
| Rise/fall time lag | $t_{FR}/t_{FF}$ | 90 | 111.11 | % | |

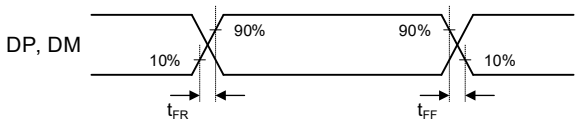



Figure 48.11 USB0_DP, USB1_DP, USB0_DM, and USB1_DM Output Timing (Full-Speed)

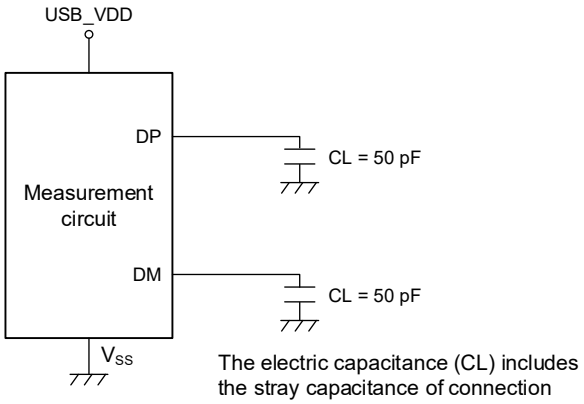



Figure 48.12 Measurement Circuit (Full-Speed)

### 48.5.4.3 USB 2.0 Hi-Speed Access Timing

Table 48.20 USB Transceiver Timing (Hi-Speed)

| Item | Symbol | Min. | Max. | Unit | Figures |
|---|---|---|---|---|---|
| Rise edge rate | $t_{HSR}$ | — | 2133 | V/µs | Figure 48.13 |
| Fall edge rate | $t_{HSF}$ | — | 2133 | V/µs | |
| Output driver resistance | $Z_{HSDRV}$ | 40.5 | 49.5 | Ω | |

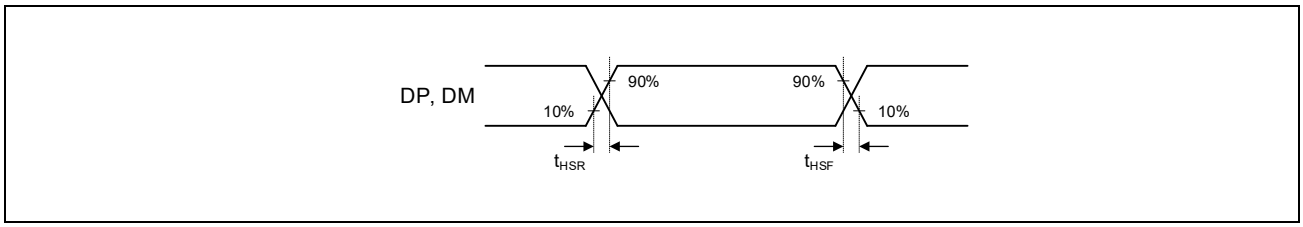



Figure 48.13 USB0_DP, USB1_DP, USB0_DM, and USB1_DM Output Timing (Hi-Speed)

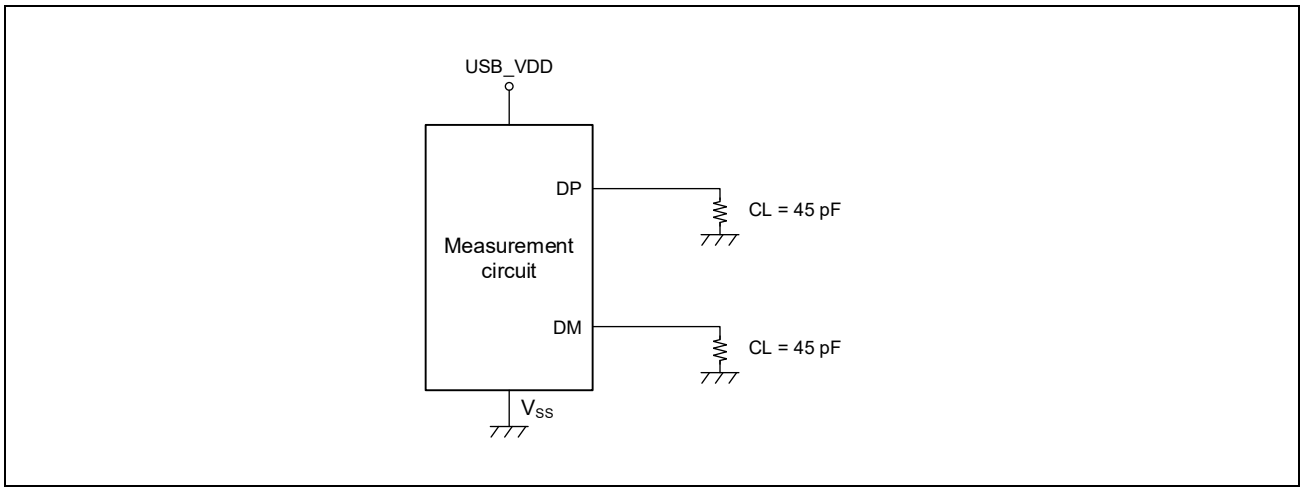



Figure 48.14 Measurement Circuit (Hi-Speed)

### 48.5.5 JTAG Debugger Interface Access Timing

Table 48.21 Debugger IF Timing

| Item | Symbol | Min. | Max. | Unit | Figures |
|---|---|---|---|---|---|
| TCK_SWCLK cycle time | $t_{TCKcyc}$ | 50 | — | ns | Figure 48.15 |
| TCK_SWCLK high pulse width | $t_{TCKH}$ | 20 | — | ns | Figure 48.16 |
| TCK_SWCLK low pulse width | $t_{TCKL}$ | 20 | — | ns | |
| TDI setup time | $t_{TDIS}$ | 15 | — | ns | |
| TDI hold time | $t_{TDIH}$ | 15 | — | ns | |
| TMS_SWDIO setup time | $t_{TMSS}$ | 15 | — | ns | |
| TMS_SWDIO hold time | $t_{TMSH}$ | 15 | — | ns | |
| SWDIO delay time | $t_{SWDO}$ | — | 14 | ns | |
| TDO delay time | $t_{TDOD}$ | — | 14 | ns | |
| Capture register setup time | $t_{CAPTS}$ | 10 | — | ns | Figure 48.17 |
| Capture register hold time | $t_{CAPTH}$ | 10 | — | ns | |
| Update register delay time | $t_{UPDATED}$ | — | 20 | ns | |

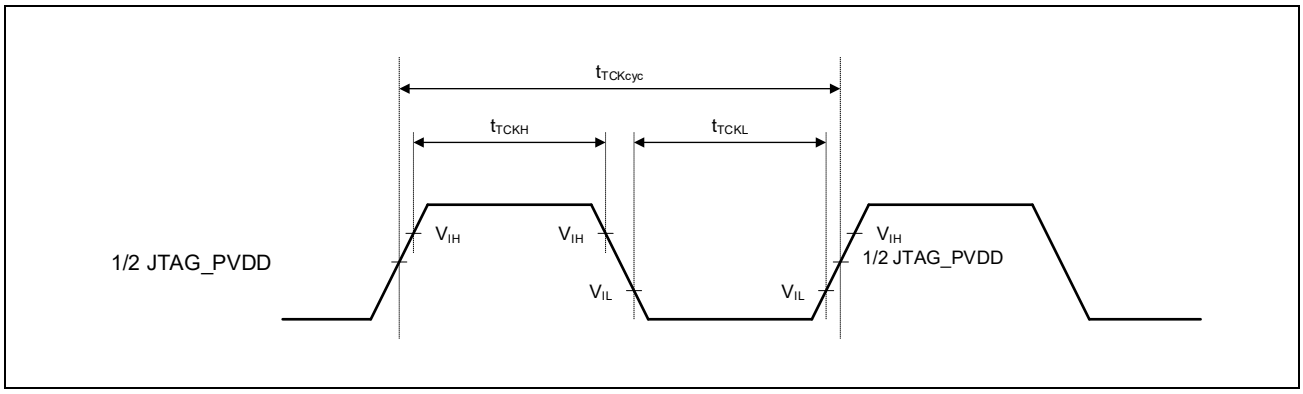



Figure 48.15 TCK_SWDCLK Input Timing

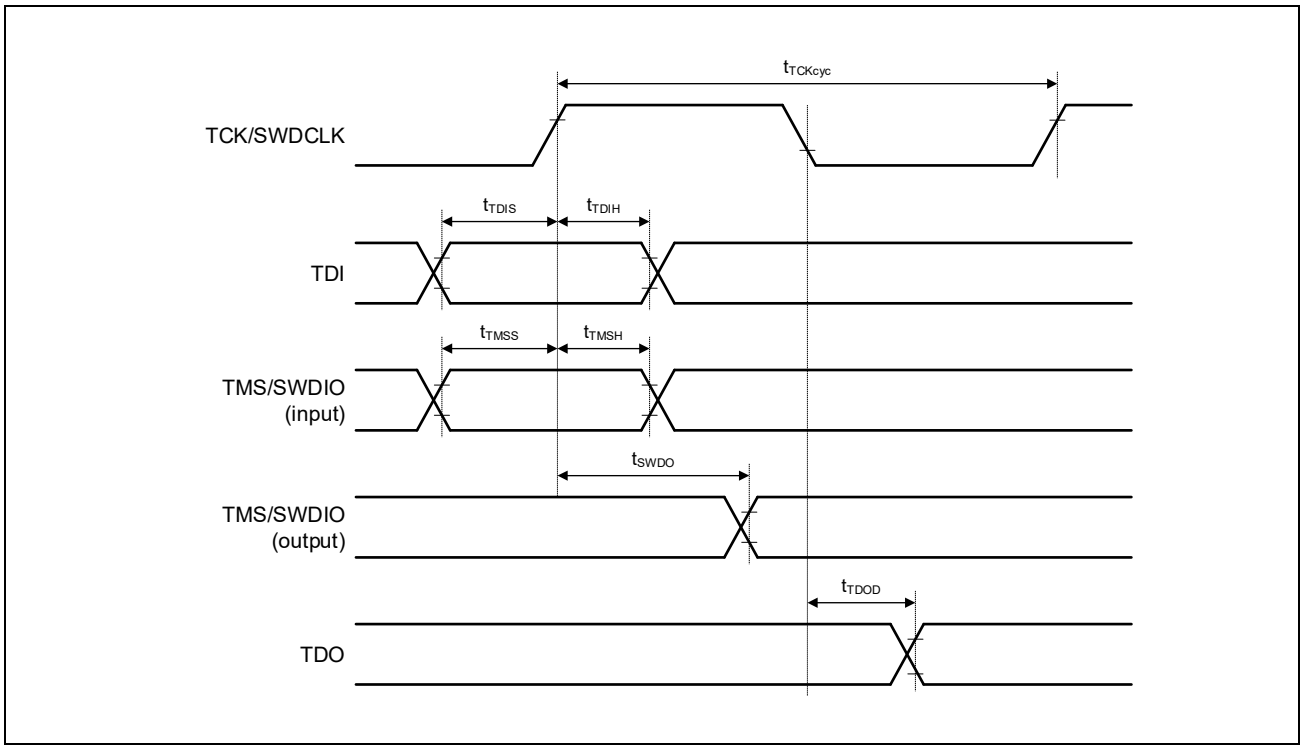



Figure 48.16 Data Transfer Timing

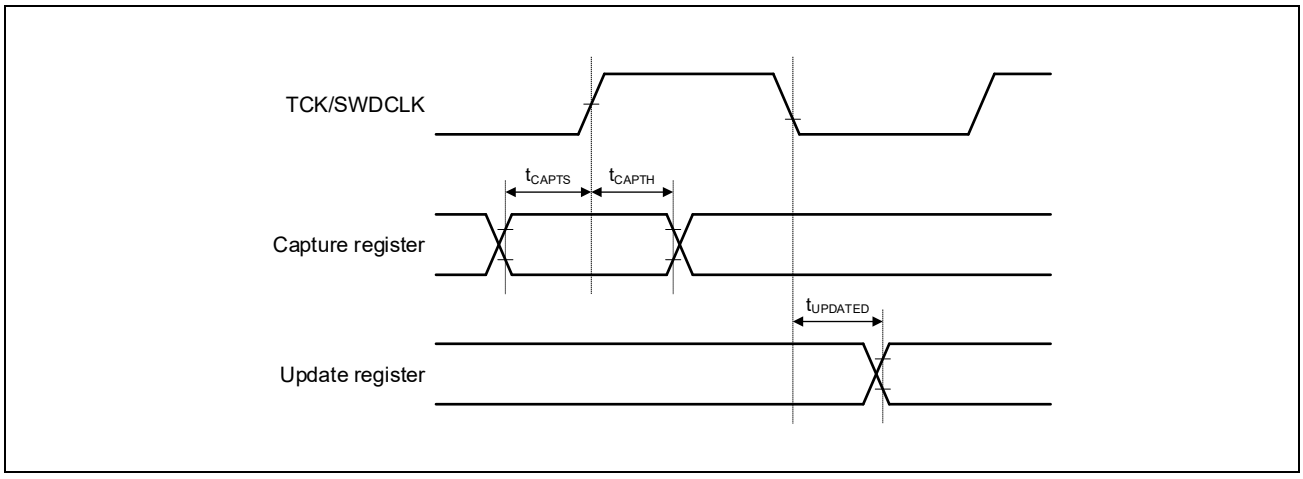



Figure 48.17 Boundary Scan Input/Output I/O Timing

### 48.5.6 SPI Multi I/O Bus Controller Access Timing

Table 48.22 SPI Multi I/O Bus Contoller Access Timing

| Item | | Symbol | 1.8 V (Serial flash/octal-SPI flash connected) Min. | 1.8 V (Serial flash/octal-SPI flash connected) Max. | 3.3 V (Serial flash connected) Min. | 3.3 V (Serial flash connected) Max. | Unit | Figures |
|---|---|---|---|---|---|---|---|---|
| Clock cycle | | $t_{SPBcyc}$ | 15.0 | — | 15.0 | — | ns | Figure 48.18 |
| CLK high pulse width | | $t_{SPBWH}$ | 0.45 | 0.55 | 0.45 | 0.55 | $t_{SPBcyc}$ | Figure 48.18 |
| CLK low pulse width | | $t_{SPBWL}$ | 0.45 | 0.55 | 0.45 | 0.55 | $t_{SPBcyc}$ | Figure 48.18 |
| CLK rise time | | $t_{SPBR}$ | — | 1.0 | — | 3.2 | ns | Figure 48.18 |
| CLK fall time | | $t_{SPBF}$ | — | 1.0 | — | 3.2 | ns | Figure 48.18 |
| Data input setup time | QSPI0_SPCLK base point (SDR mode timing adjusted) | $t_{SU}$ | 6.7 | — | 7.5 | — | ns | Figure 48.19 |
| | QSPI0_SPCLK base point (DDR mode timing adjusted) | | 4.5 | — | 4.5 | — | ns | Figure 48.20 |
| Data input hold time | QSPI0_SPCLK base point (SDR mode timing adjusted) | $t_{H}$ | 0.5 | — | 0.5 | — | ns | Figure 48.19 |
| | QSPI0_SPCLK base point (DDR mode timing adjusted) | | 1.0 | — | 1.0 | — | ns | Figure 48.20 |
| SSL setup time | | $t_{LEAD}$ | 1.5 × $t_{SPBcyc}$ − 3 | 8.5 × $t_{SPBcyc}$ + 3 | 1.5 × $t_{SPBcyc}$ − 3 | 8.5 × $t_{SPBcyc}$ + 3 | ns | Figure 48.19, Figure 48.20 |
| SSL hold time | | $t_{LAG}$ | 1 × $t_{SPBcyc}$ − 3 | 8 × $t_{SPBcyc}$ + 3 | 1 × $t_{SPBcyc}$ − 3 | 8 × $t_{SPBcyc}$ + 3 | ns | Figure 48.19, Figure 48.20 |
| Continuous transfer delay time | | $t_{TD}$ | 1 × $t_{SPBcyc}$ − 3 | 8 × $t_{SPBcyc}$ + 3 | 1 × $t_{SPBcyc}$ − 3 | 8 × $t_{SPBcyc}$ + 3 | ns | Figure 48.19, Figure 48.20 |
| Data output delay time | SDR | $t_{OD}$ | — | 2.0 | — | 5.0 | ns | Figure 48.19 |
| | DDR | | — | 6.5*2 | — | 7.5*3 | ns | Figure 48.20 |
| Data output hold time | SDR | $t_{OH}$ | −2.0 | — | -5.0 | — | ns | Figure 48.19 |
| | DDR | | 1.0*2 | — | 2.1*3 | — | ns | Figure 48.20 |
| Data output buffer off time | SDR | $t_{BOFF}$ | — | 2.0 | — | 3.0 | ns | **Figure 48.21** |
| | DDR | | — | 2.0 | — | 3.0 | ns | **Figure 48.21** |

Note 1. Output load: 15 pF/driving ability: 12 mA

Note 2. QSPI0_SPCLK frequency: 100 MHz.

Note 3. QSPI0_SPCLK frequency: 66 MHz

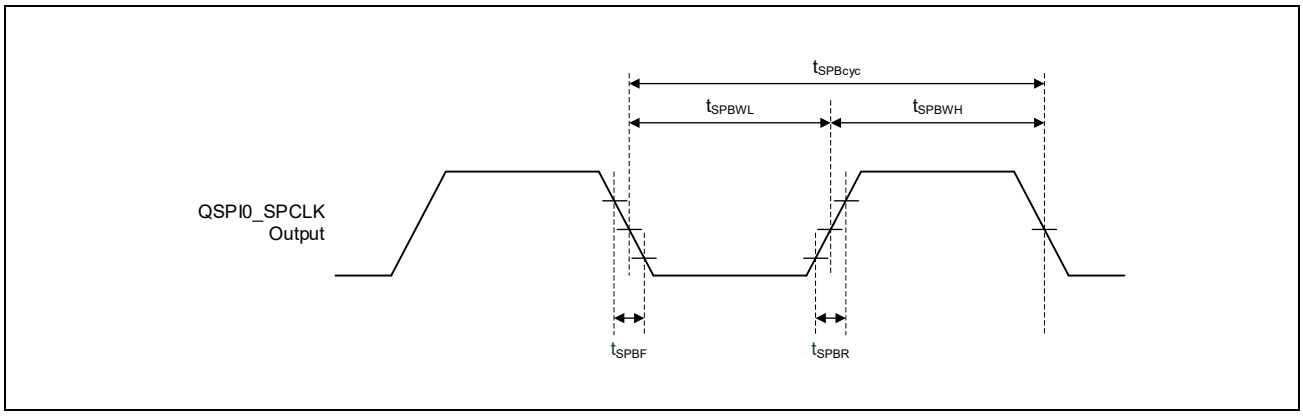



Figure 48.18 Clock Timing

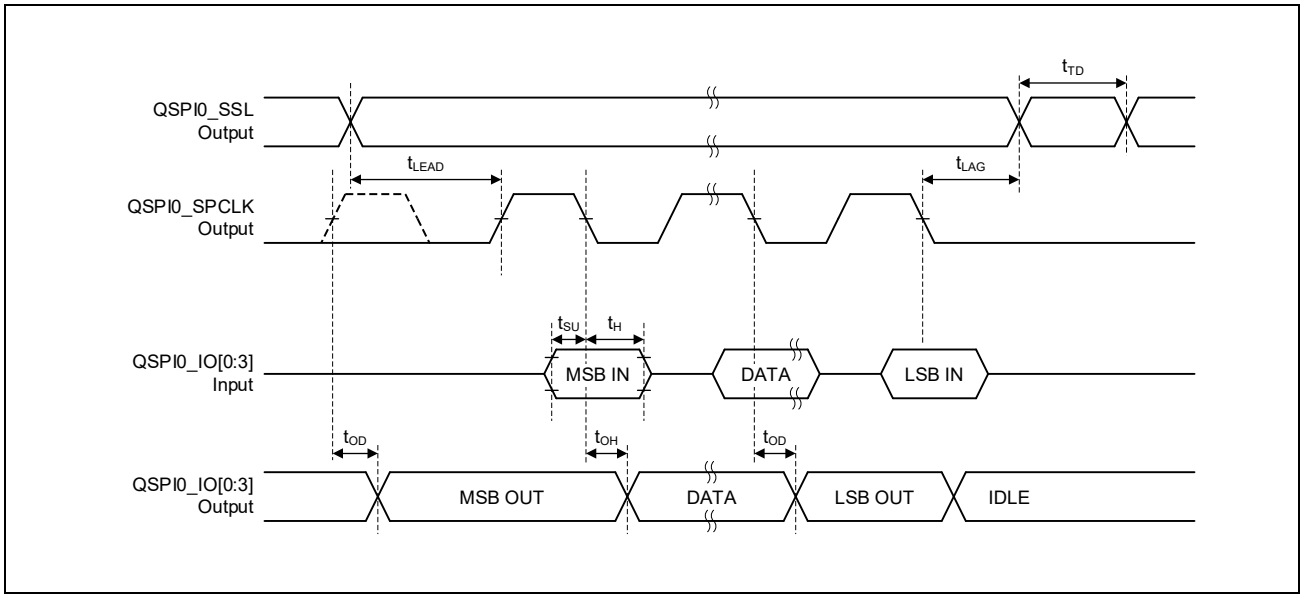



Figure 48.19 SDR Transfer Format Transmission and Reception Timing

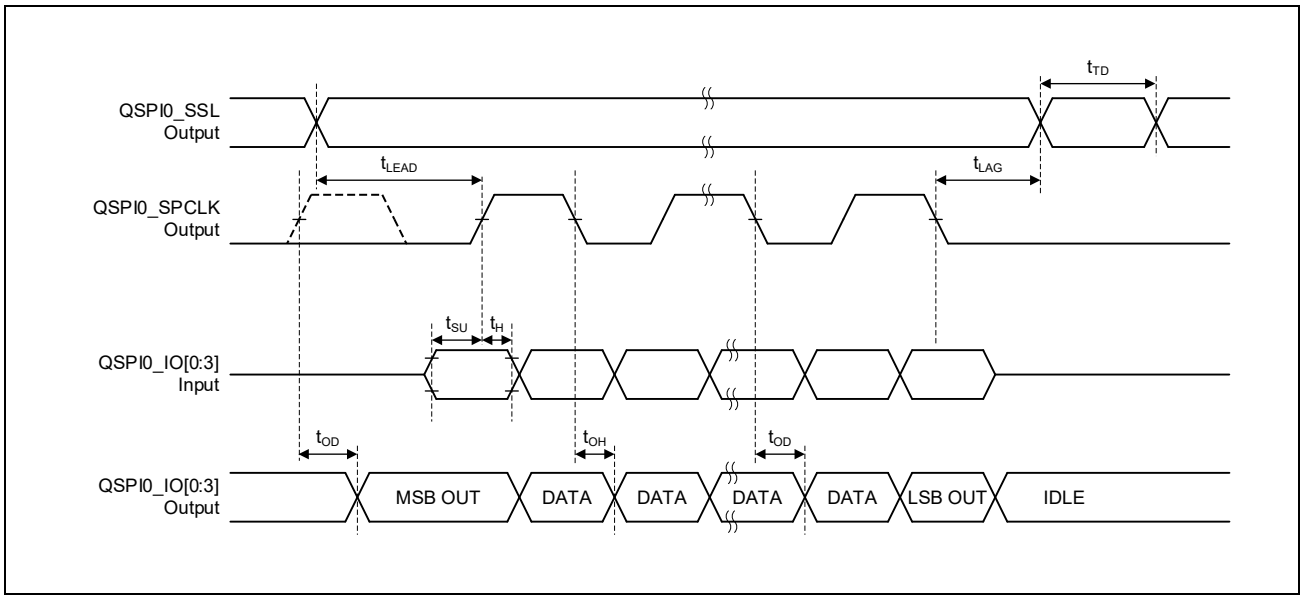



Figure 48.20 DDR Transfer Format Transmission and Reception Timing

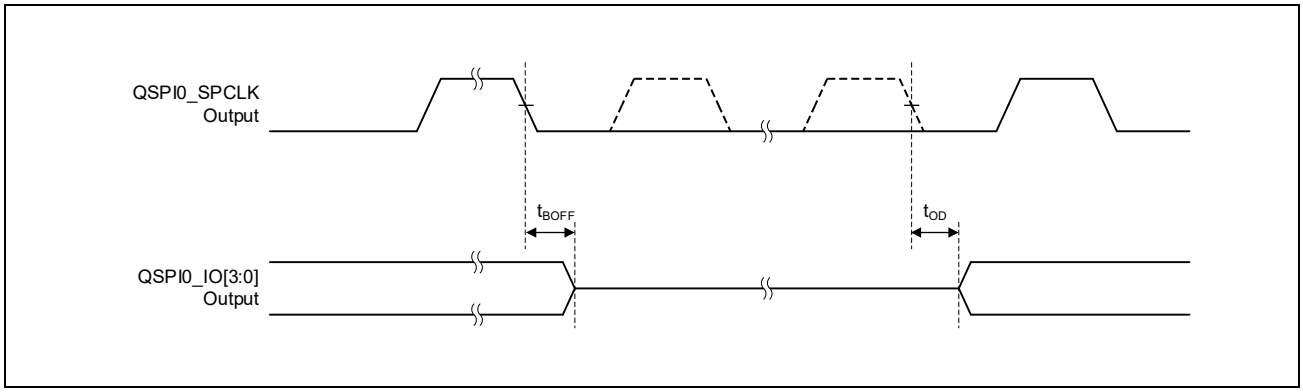



Figure 48.21 Timing for Switching the Buffers On and Off

### 48.5.7 Control Signal Access Timing

Table 48.23 Control Signal Timing

| Item | Symbol | Min. | Max. | Unit | Figures |
|---|---|---|---|---|---|
| PRST# pulse width | $t_{RESW}$ | 20 | — | $t_{cyc}$*1 | Figure 48.22 |
| TRST# pulse width | $t_{TRSW}$ | 20 | — | $t_{cyc}$*1 | |
| NMI pulse width | $t_{NMIW}$ | 20 | — | $t_{cyc}$*1 | Figure 48.24 |
| IRQ pulse width | $t_{IRQW}$ | 20 | — | $t_{cyc}$*1 | |
| TINT pulse width | $t_{TINTW}$ | 20 | — | $t_{cyc}$*1 | |
| PRST# input rise time | $t_{RSr}$ | — | 500 | µs | Figure 48.23 |

Note 1. $t_{cyc}$ = 41.666 ns (24 MHz)

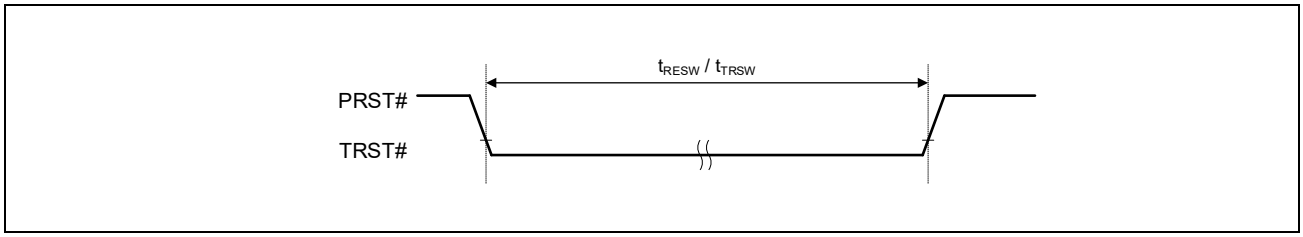



Figure 48.22 Reset Input Timing 1

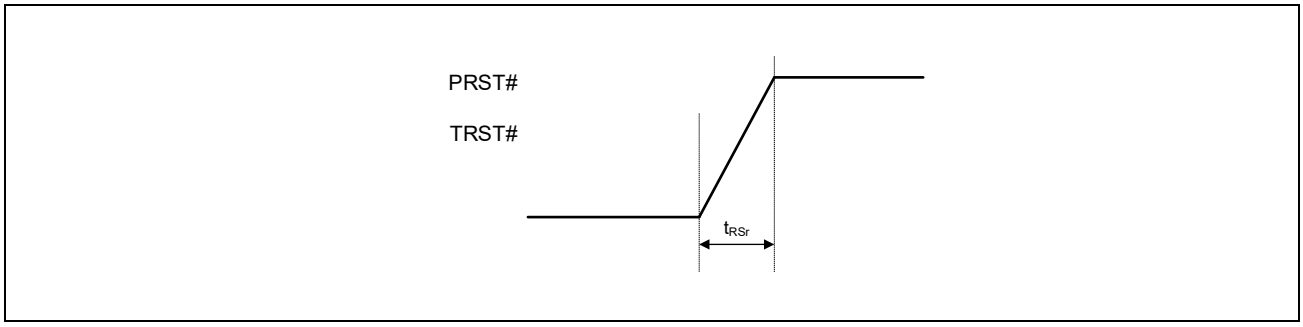
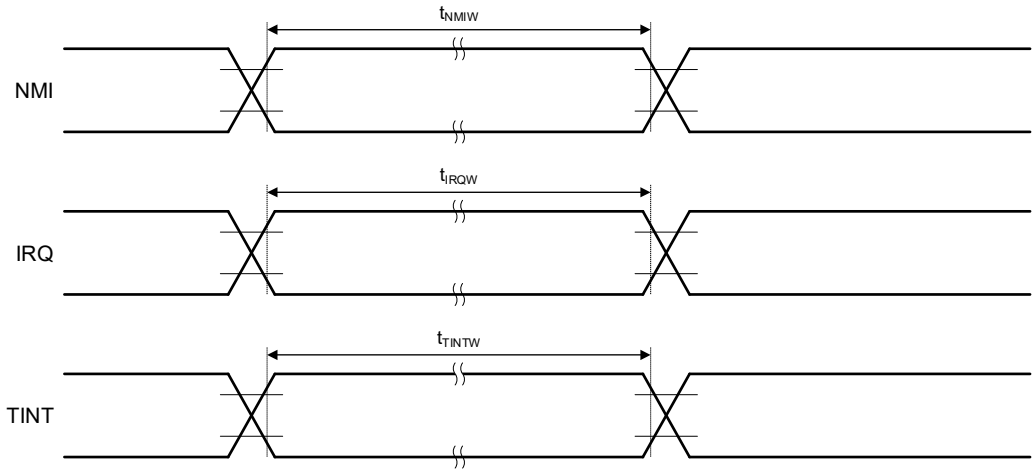



Figure 48.23 Reset Input Timing 2




Note: This specification “(Min. 20 $t_{cycle}$)” is the min. pulse width the case that the digital noise filter is off. Refer to Section 41, General Purpose Input Output Port (GPIO) for Digital Noise Filter Settings in detail.

Figure 48.24 Interrupt Signal Input Timing

### 48.5.8 Serial Sound Interface (SSIF-2) Access Timing

Table 48.24 SSIF-2 Timing

| Item | I/O | Symbol | Min. | Max. | Unit | Figures |
|---|---|---|---|---|---|---|
| Output clock cycle | Output | $t_O$ | 80 | 64000 | ns | Figure 48.25 |
| Input clock cycle | Input | $t_I$ | 80 | — | ns | |
| Clock high | Bidirectional | $t_{HC}$ | 32 | — | ns | |
| Clock low | | $t_{LC}$ | 32 | — | ns | |
| Clock rise time/clock fall time | Output | $t_{RC}/t_{FC}$ | — | 25 | ns | |
| Setup time | Input | $t_{SR}$ | 25 | — | ns | Figure 48.26, Figure 48.27, Figure 48.28 |
| Hold time | | $t_{HR}$ | 5 | — | ns | |
| SILRCK output delay time | Output | $t_{DTR}$ | −5 | 25 | ns | |
| Data output delay time (Noise canceler not in use) | | $t_{DTR}$ | −5 | 25 | ns | |
| Data output delay time (Noise canceler in use) | | $t_{DTR}$ | 10 | 50 | ns | |

Note: AC access timing condition: drive ability 12mA, output load 30pF, slew rate = fast

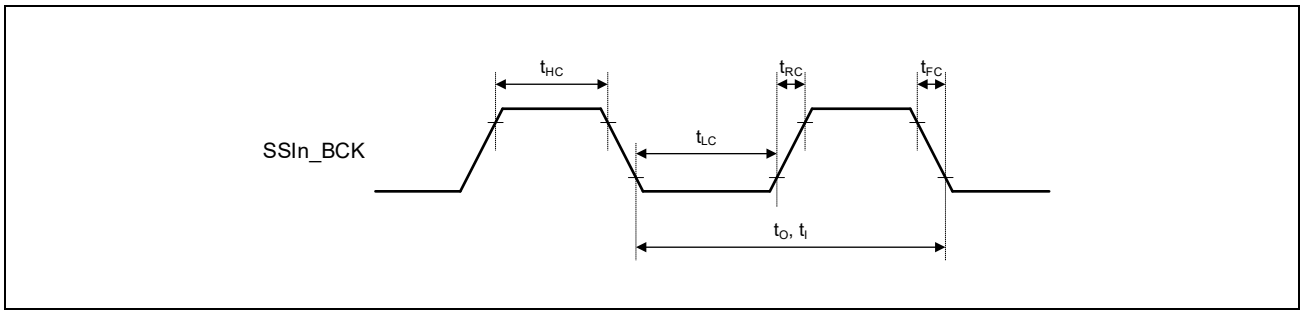



Figure 48.25 Bit Clock Input/Output Timing

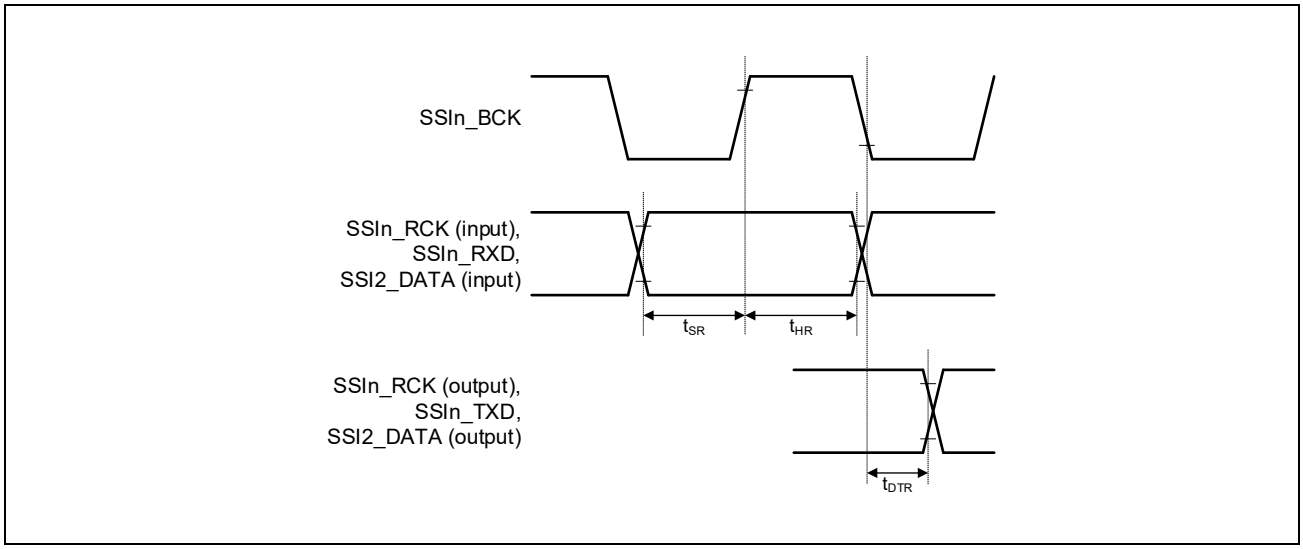



Figure 48.26 Transmission and Reception Timing (SSIBCK Falling Output)

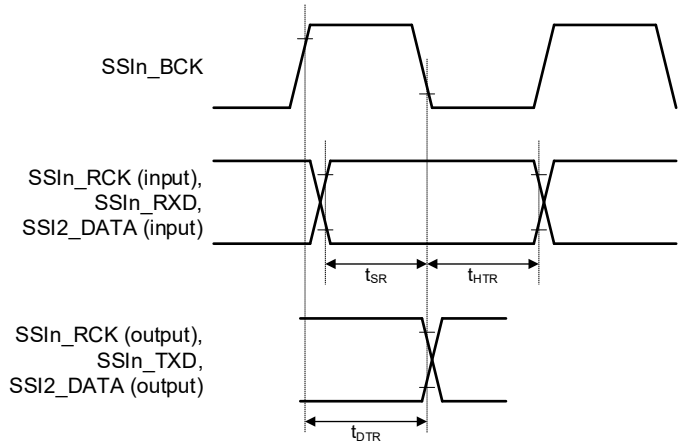
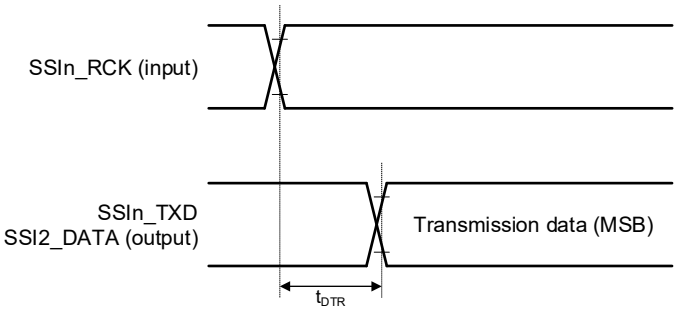



Figure 48.27 Transmission and Reception Timing (SSIBCK Rising Output)




Note: Output timing of the MSB when slave transmission satisfies the following condition a) or b)
a) SSICR.DEL = 1, SSICR.SDTA = 0
b) SSICR.DEL = 1, SSICR.SDTA = 1, without padding

Figure 48.28 Transmission Timing (Slave, in Synchronization with SSILRCK)

### 48.5.9 Multi-Function Timer Pulse Unit 3 (MTU3a) Access Timing

Table 48.25 MTU3a Timing

| Item | | | Symbol | Min. | Max. | Unit*1 | Figures |
|---|---|---|---|---|---|---|---|
| MTU3a | Input capture input pulse width | Single-edge setting | $t_{MTICW}$ | 1.5 | — | $t_{p1cyc}$*1 | Figure 48.29 |
| | | Both-edge setting | | 2.5 | — | | |
| | Timer clock pulse width | Single-edge setting | $t_{MTCKWH}$, $t_{MTCKWL}$ | 1.5 | — | $t_{p1cyc}$*1 | Figure 48.30 |
| | | Both-edge setting | | 2.5 | — | | |
| | | Phase counting mode | | 2.5 | — | | |

Note: AC access timing condition: drive ability 4mA, output load 30pF, slew rate = fast

Note 1. $t_{p1cyc}$ indicates peripheral clock means MTU_X_MCLK_MTU3 (P0ϕ).

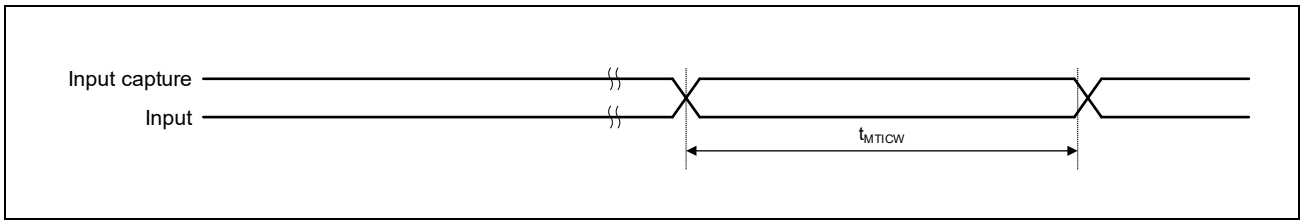



Figure 48.29 MTU3a Input Capture Input Timing

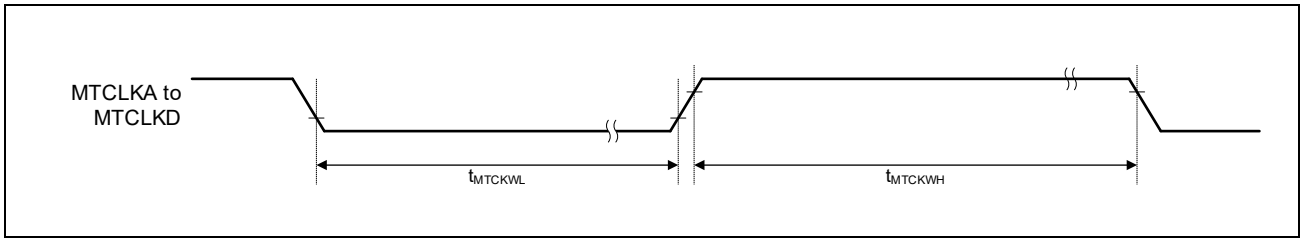



Figure 48.30 MTU3a Clock Input Timing

### 48.5.10 Port Output Enable 3 (POE3) Access Timing

Table 48.26 POE3 Timing

| Item | | Symbol | Min. | Max. | Unit | Figures |
|---|---|---|---|---|---|---|
| POE3 | POEn# input pulse width | $t_{POE3W}$ | 1.5 | — | $t_{p1cy}$*1 | Figure 48.31 |

Note 1. $t_{p1cyc}$ indicates peripheral clock means POE3_CLKM_POE (P0ϕ).

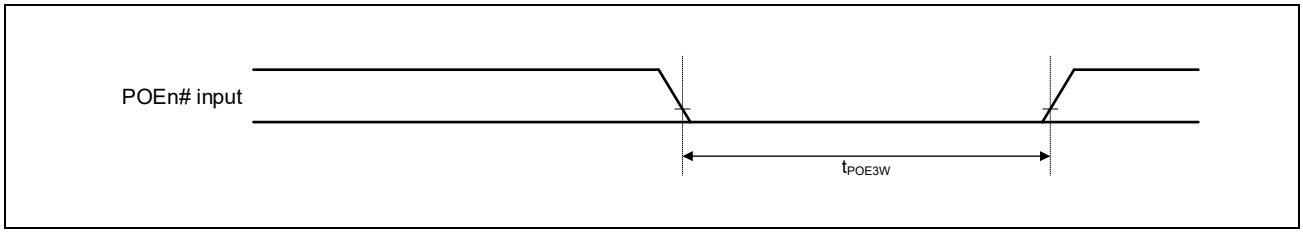



Figure 48.31 POEn# Input Pulse Timing

## 48.5.11 I²C Bus Interface Access Timing

Table 48.27 I²C Bus Interface Timing

| Item | Symbol | I/O | Standard Mode (Sm) | | Fast Mode (Fm) | | Fast Mode Plus (Fm+) | | Unit |
|---|---|---|---|---|---|---|---|---|---|
| | | | Min. | Max. | Min. | Max. | Min. | Max. | |
| SCL clock frequency | $f_{CLK}$ | I/O | 0 | 100 | 0 | 400 | 0 | 1000 | kHz |
| Bus free time (between stop and start condition) | $t_{BUF}$ | I/O | 4.7 | — | 1.3 | — | 0.5 | — | µs |
| Hold time*1 | $t_{HD:STA}$ | I/O | 4.0 | — | 0.6 | — | 0.26 | — | µs |
| Low period of SCL clock | $t_{LOW}$ | I/O | 4.7 | — | 1.3 | — | 0.5 | — | µs |
| High period of SCL clock | $t_{HIGH}$ | I/O | 4.0 | — | 0.6 | — | 0.26 | — | µs |
| Setup time for start / restart condition | $t_{SU:STA}$ | I/O | 4.7 | — | 0.6 | — | 0.26 | — | µs |
| Data hold time (I²C bus device) | $t_{HD:DAT}$ | I/O | 0*2 | — | 0*2 | — | 0 | — | µs |
| Data setup time | $t_{SU:DAT}$ | I/O | 250 | — | 100*3 | — | 50 | — | ns |
| SDA and SCL signal rise time | $t_R$ | Input | — | 1000 | 20 | 300 | — | 120 | ns |
| SDA and SCL signal fall time*3 | $t_F$ | Input | — | 300 | 20 × ($PV_{DD}$/ 5.5 V) | 300 | 20 × ($PV_{DD}$/ 5.5 V) | 120 | ns |
| | | Output | — | 300 | 20 × ($PV_{DD}$/ 5.5 V)*6 | 300*6 | 20 × ($PV_{DD}$/ 5.5 V)*7 | 120*7 | ns |
| Setup time for STOP condition | $t_{SU:STO}$ | I/O | 4.0 | — | 0.6 | — | 0.26 | — | µs |
| Capacitive load for each bus line | $C_b$ | — | — | 400*4 | — | 400*4 | — | 550*4 | pF |
| Pulse width of spikes that must be suppressed by the input filter | $t_{SP}$ | Input | — | — | 0 | 50*5 | 0 | 50*5 | ns |

Note: In the above table and subsequently, SCL and SDA refer to the RIICnSCL and RIICnSDA signals, respectively.

Note: AC access timing condition: drive ability 4mA, output load 400pF, slew rate = slow

Note 1. The first clock pulse is generated on the SCL line after the start condition has been issued and the hold time has elapsed.

Note 2. This module requires a minimum of 300 ns hold time internally for the SDA signal to handle the period over which the falling edge of SCL has not reached a defined level (time until the CnSCL signal reaches $V_{IL}$ (max.) from $V_{IH}$ (min.)).

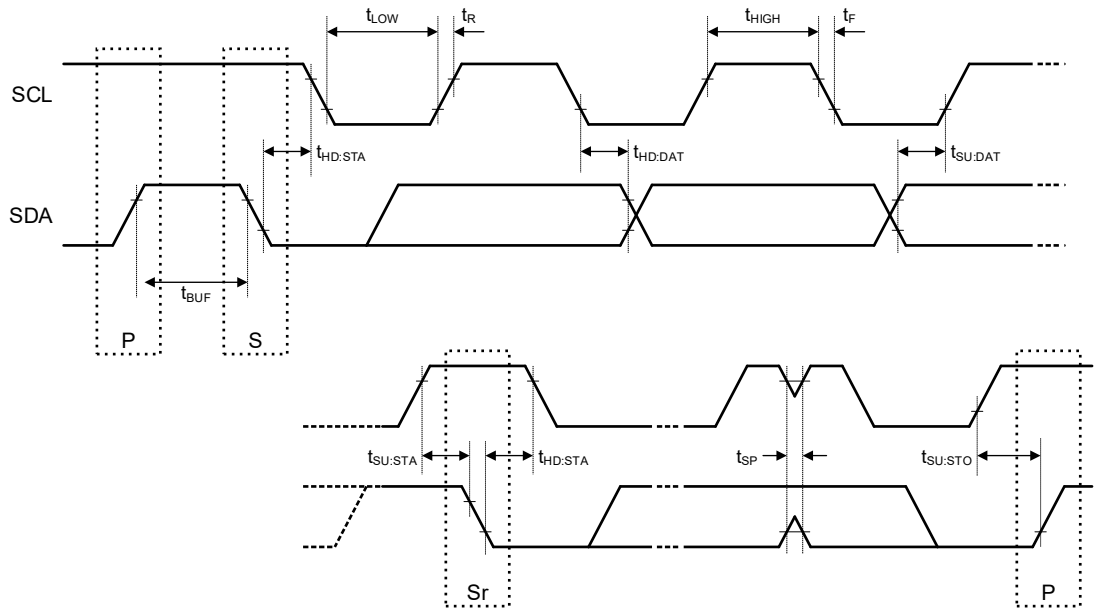
Note 3. The fast-mode I²C bus device can be used in the standard mode I²C bus system. In this case, the minimum value of the data setup time ($t_{SU:\,DAT}$ (min.) 250 [ns]) must be satisfied.
If the system does not extend the low period of SCL clock ($t_{LOW}$), this condition is automatically satisfied. If the system extends the low period of SCL clock ($t_{LOW}$), transmit the subsequent data bit to the SDA line before the SCL line is released ($t_R$ (max.) + $t_{SU:\,DAT}$ (min.) = 1000 + 250 = 1250 [ns]: (standard mode I²C bus specification)).

Note 4. Total capacitance of one bus line. The allowable maximum bus capacitance may differ from this specification, depending on the actual operating voltage and frequency of an application. For techniques to cope with a large bus capacitance, see the I²C bus specification provided by NXP Semiconductors.

Note 5. Noise is removed by the analog and digital input filters. The level of noise reduction of the digital input filter is determined by the period of internal reference clock (IICϕ) and the NF[1:0] bits in RIICnMR3. For details, refer to Section 26, I²C Bus Interface.

Note 6. External pull-up resistor is required 1077Ω to 1770Ω when using RIIC ch2 or RIIC ch3.

Note 7. External pull-up resistor is required 240Ω to 257Ω when using RIIC ch2 or RIIC ch3.




Note: P: Stop condition
S: Start condition
Sr: Start condition for retransmission

Figure 48.32 Input/Output Timing

## 48.5.12 Serial Communications Interface with FIFO (SCIFA) Access Timing

Table 48.28 SCIFA Timing

| Item | | | Symbol | Min. | Max. | Unit | Figures |
|---|---|---|---|---|---|---|---|
| SCIFA | Input clock cycle | Asynchronous | $t_{Scyc}$ | 4 | — | $t_{p1cyc}$*1 | Figure 48.33 |
| | | Clocked synchronous | | 12 | — | | |
| | Input clock pulse width | | $t_{SCKW}$ | 0.4 | 0.6 | $t_{p1cyc}$*1 | |
| | Input clock rise time | | $t_{SCKr}$ | — | 5 | ns | |
| | Input clock fall time | | $t_{SCKf}$ | — | 5 | ns | |
| | Output clock cycle | Asynchronous*2 | $t_{Scyc}$ | 8 | — | $t_{p1cyc}$*1 | |
| | | Clocked synchronous | | 4 | — | | |
| | Output clock pulse width | | $t_{SCKW}$ | 0.4 | 0.6 | $t_{p1cyc}$*1 | |
| | Output clock rise time | | $t_{SCKr}$ | — | 9 | ns | |
| | Output clock fall time | | $t_{SCKf}$ | — | 9 | ns | |
| | Transmit data delay time | Internal clock | $t_{TXD}$ | −10 | 10 | ns | Figure 48.34 |
| | | External clock | | 3 × $t_{p1cyc}$*1 | 4 × $t_{p1cyc}$*1 + 20 | | |
| | Receive data setup time | Internal clock | $t_{RXS}$ | 3 × $t_{p1cyc}$*1 + 20 | — | ns | |
| | | External clock | | $t_{p1cyc}$*1 + 10 | — | | |
| | Receive data hold time | Internal clock | $t_{RXH}$ | −3 × $t_{p1cyc}$*1 | — | ns | |
| | | External clock | | 2 × $t_{p1cyc}$*1 + 10 | — | | |

Note: AC access timing condition: drive ability 12mA, output load 30pF, slew rate = fast

Note 1. $t_{p1cyc}$ indicates peripheral clock means SCIFn_CLK_PCK (P0ϕ) (n = 0 to 4).

Note 2. When the SEMR.ABCS0 and SEMR.BGDM bits are set to 1.

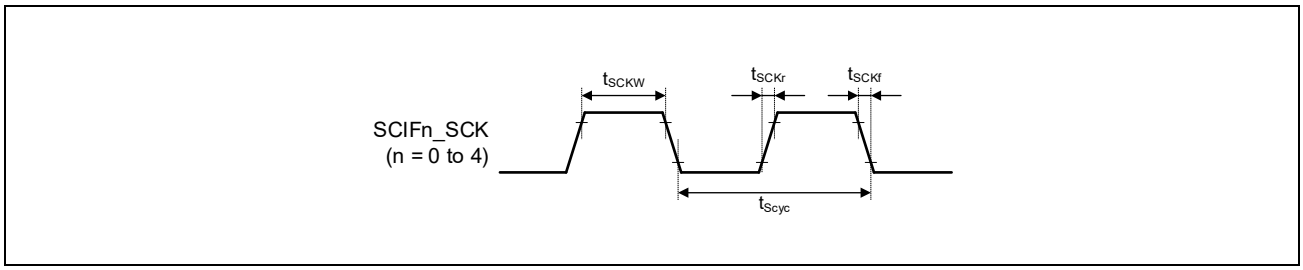
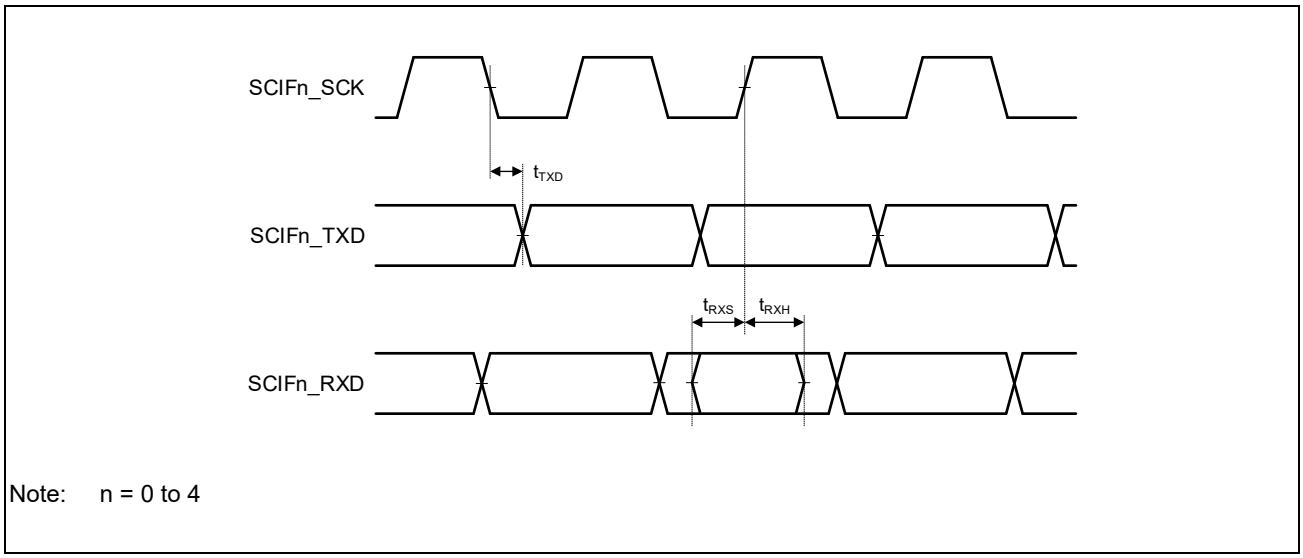



Figure 48.33 SCK Input Clock Timing




Note: n = 0 to 4

Figure 48.34 SCIFA Input/Output Timing in Clocked Synchronous Mode

## 48.5.13 Serial Communications Interface (SCIg) Access Timing

Table 48.29 SCIg Timing

| Item | | | Symbol | Min. | Max. | Unit | Figures |
|---|---|---|---|---|---|---|---|
| SCIg | Input clock cycle | Asynchronous | $t_{Scyc}$ | 4 | — | $t_{p1cyc}$*1 | Figure 48.35 |
| | | Clocked synchronous | | 6 | — | | |
| | Input clock pulse width | | $t_{SCKW}$ | 0.4 | 0.6 | $t_{p1cyc}$*1 | |
| | Input clock rise time | | $t_{SCKr}$ | — | 5 | ns | |
| | Input clock fall time | | $t_{SCKf}$ | — | 5 | ns | |
| | Output clock cycle | Asynchronous*2 | $t_{Scyc}$ | 8 | — | $t_{p1cyc}$*1 | |
| | | Clocked synchronous | | 4 | — | | |
| | Output clock pulse width | | $t_{SCKW}$ | 0.4 | 0.6 | $t_{p1cyc}$*1 | |
| | Output clock rise time | | $t_{SCKr}$ | — | 5 | ns | |
| | Output clock fall time | | $t_{SCKf}$ | — | 5 | ns | |
| | Transmit data delay time | Clocked synchronous | $t_{TXD}$ | — | 28 | ns | Figure 48.36 |
| | Receive data setup time | Clocked synchronous | $t_{RXS}$ | 15 | — | ns | |
| | Receive data hold time | Clocked synchronous | $t_{RXH}$ | 5 | — | ns | |

Note: AC access timing condition: drive ability 12mA, output load 30pF, slew rate = fast

Note 1. $t_{p1cyc}$ indicates peripheral clock means SCIn_SCK (P0ϕ) (n = 0 to 1).

Note 2. When the SEMR.ABCS0 and SEMR.BGDM bits are set to 1.

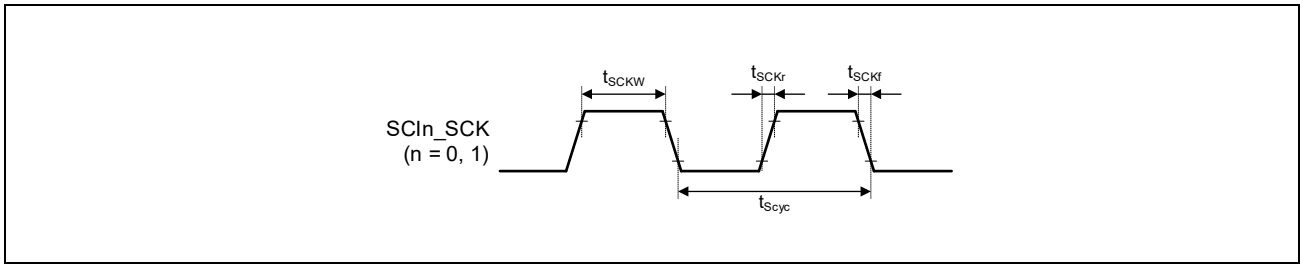
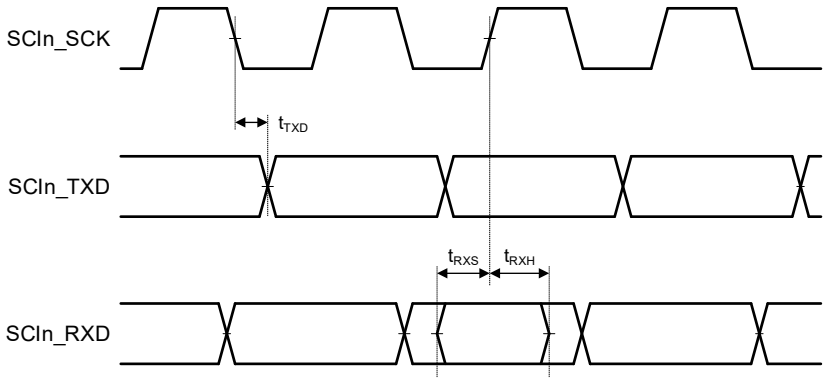



Figure 48.35 SCK Input Clock Timing




Note: n = 0, 1

Figure 48.36 SCIFA Input/Output Timing in Clocked Synchronous Mode

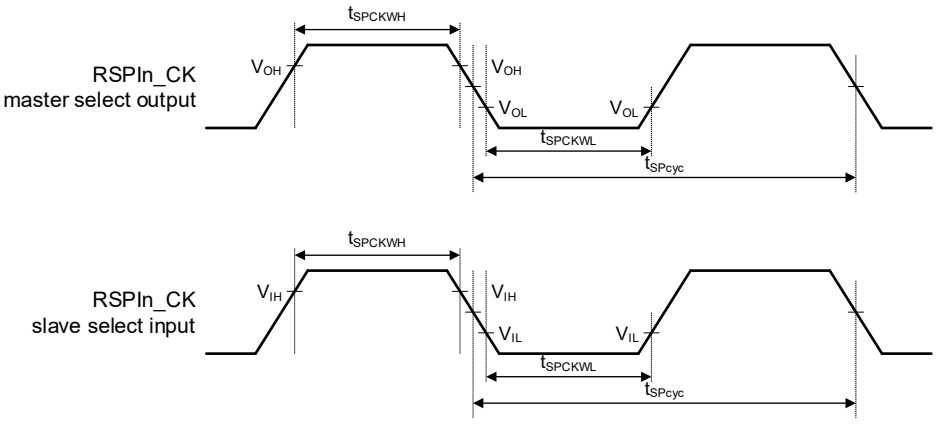
## 48.5.14 Renesas Serial Peripheral Interface (RSPI) Access Timing

Table 48.30 Renesas Serial Peripheral Interface Timing

| Item | | Symbol | Min. | Max. | Unit | Figure |
|---|---|---|---|---|---|---|
| RSPCK clock cycle | Master | $t_{SPcyc}$ | 2 | 4096 | $t_{p1cyc}$*1 | Figure 48.37 |
| | Slave | | 8 | 4096 | | |
| RSPCK clock high pulse width | Master | $t_{SPCKWH}$ | 0.4 | — | $t_{SPcyc}$*1 | |
| | Slave | | 0.4 | — | | |
| RSPCK clock low pulse width | Master | $t_{SPCKWL}$ | 0.4 | — | $t_{SPcyc}$*1 | |
| | Slave | | 0.4 | — | | |
| Data input setup time | Master | $t_{SU}$ | 9 | — | ns | Figure 48.38 to Figure 48.41 |
| | Slave | | 0 | — | $t_{p1cyc}$*1 | |
| Data input hold time | Master | $t_H$ | 0 | — | ns | |
| | Slave | | 4 | — | $t_{p1cyc}$*1 | |
| SSL setup time | Master | $t_{LEAD}$ | 1 × $t_{SPcyc}$ − 20 | 8 × $t_{SPcyc}$ | ns | |
| | Slave | | 4 | — | $t_{p1cyc}$*1 | |
| SSL hold time | Master | $t_{LAG}$ | 1 × $t_{SPcyc}$ | 8 × $t_{SPcyc}$ + 20 | ns | |
| | Slave | | 4 | — | $t_{p1cyc}$*1 | |
| Data output delay time | Master | $t_{OD}$ | — | 16 | ns | |
| | Slave | | — | 4 | $t_{p1cyc}$*1 | |
| Data output hold time | Master | $t_{OH}$ | 0 | — | ns | |
| | Slave | | 2 | — | $t_{p1cyc}$*1 | |
| Continuous transmission delay time | Master | $t_{TD}$ | 1 × $t_{SPcyc}$ + 2 × $t_{cyc}$ | 8 × $t_{SPcyc}$ + 2 × $t_{cyc}$ | ns | |
| | Slave | | 4 × $t_{cyc}$ | — | | |
| Slave access time | | $t_{SA}$ | — | 4 | $t_{p1cyc}$*1 | Figure 48.40, Figure 48.41 |
| Slave out release time | | $t_{REL}$ | — | 3 | $t_{p1cyc}$*1 | |

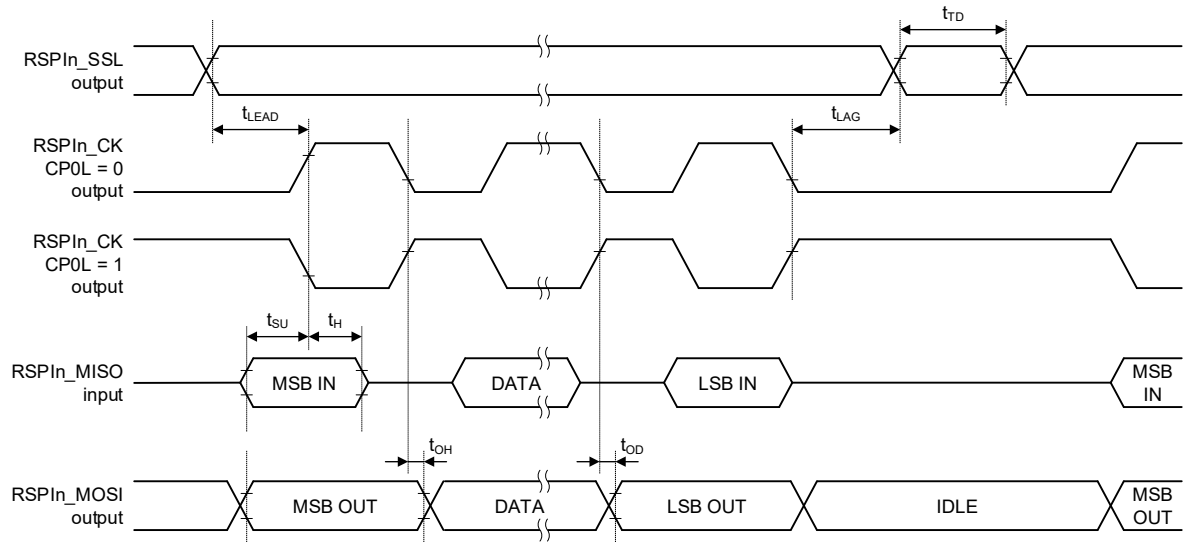
Note: AC access timing condition: drive ability 12mA, output load 30pF, slew rate = fast

Note 1. $t_{p1cyc}$ indicates peripheral clock means RSPIn_CLKB (P0ϕ) (0 to 2).



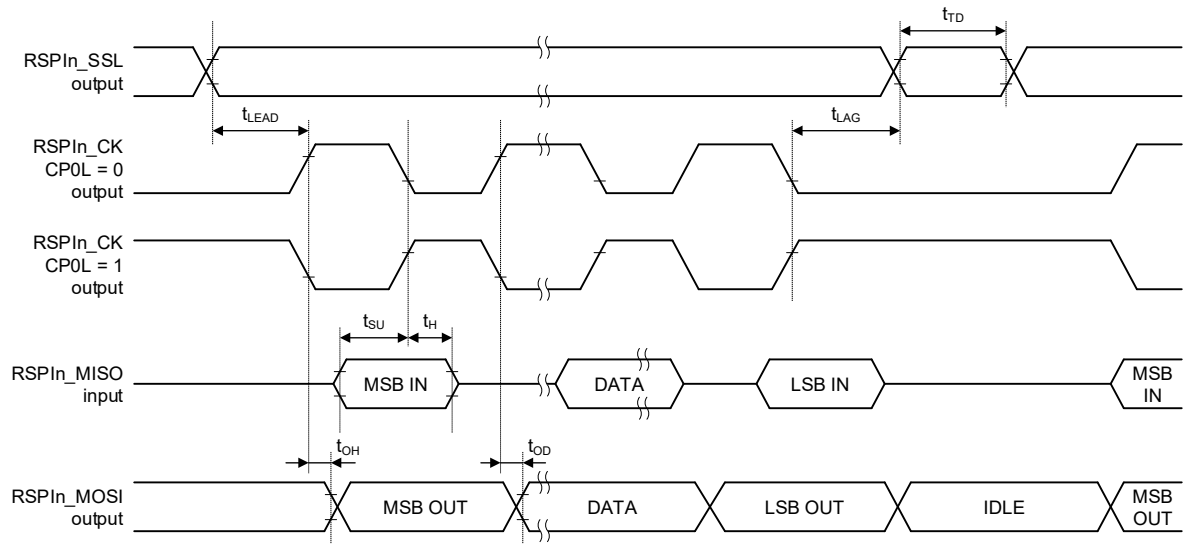

Note: n = 0 to 2

Figure 48.37 Clock Timing



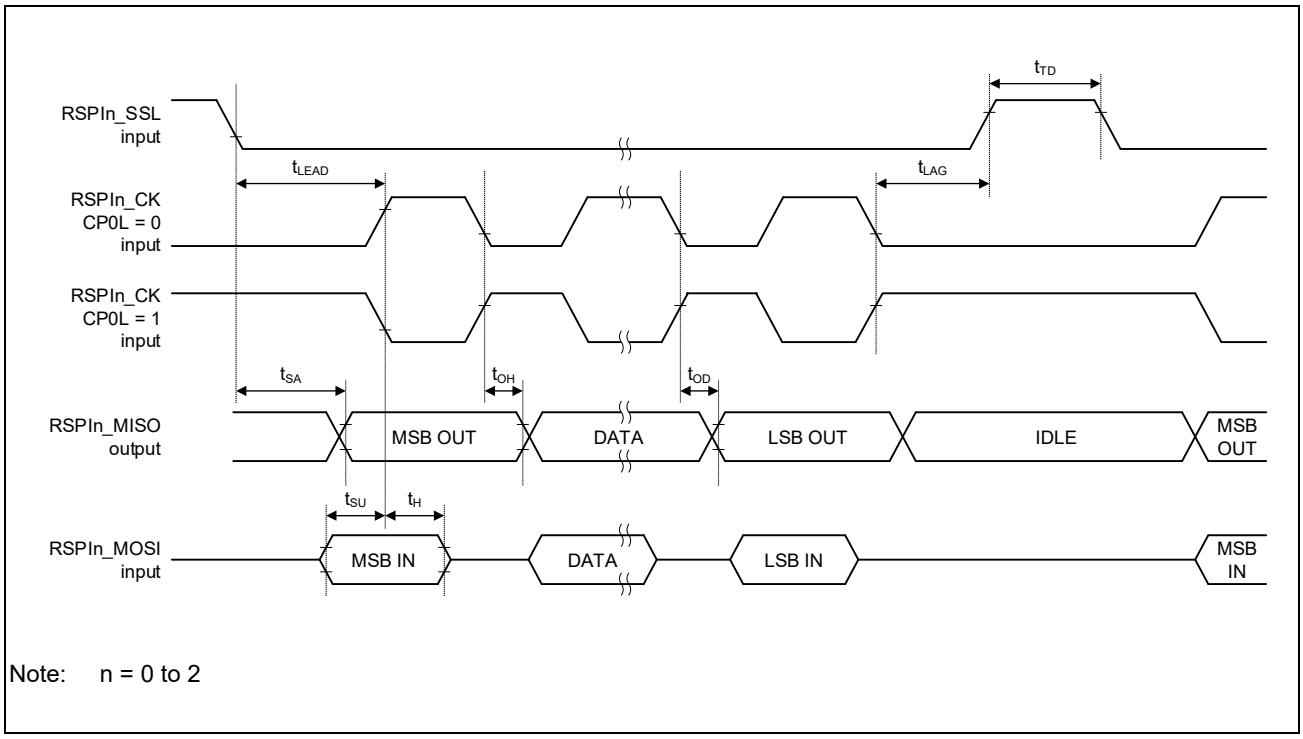

Note: n = 0 to 2

Figure 48.38 Transmission and Reception Timing (Master, CPHA = 0)



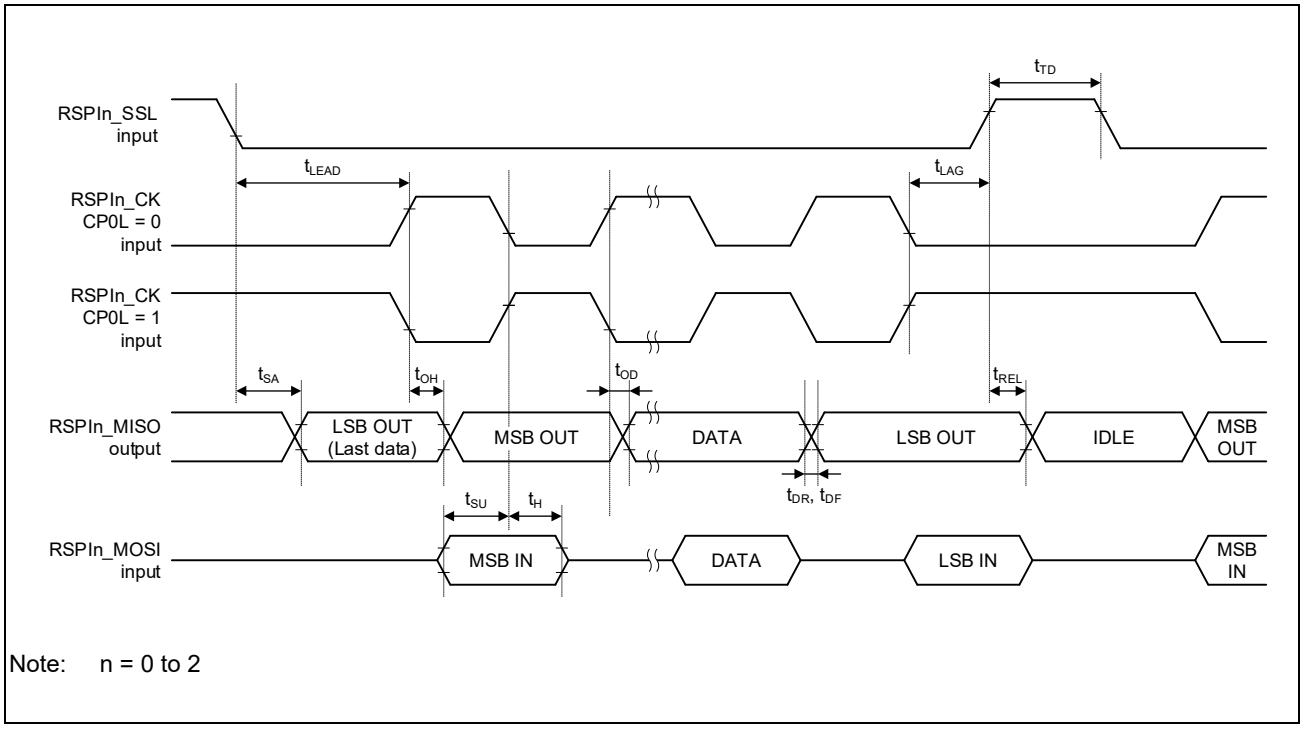

Note: n = 0 to 2

Figure 48.39 Transmission and Reception Timing (Master, CPHA = 1)




Note: n = 0 to 2

Figure 48.40 Transmission and Reception Timing (Slave, CPHA = 0)




Note: n = 0 to 2

Figure 48.41 Transmission and Reception Timing (Slave, CPHA = 1)

### 48.5.15 Watchdog Timer Access Timing

Table 48.31 Watchdog Timer Timing

| Item | Symbol | Min. | Max. | Unit | Figures |
|---|---|---|---|---|---|
| WDTOVF_PERROUT# Output Time | $t_L$ | 64 | 64 | $t_{P1cyc}$*1 | Figure 48.42 |

Note 1. $t_{P1cyc}$ indicates peripheral clock means WDTn_CLK (OSCCLK) (n = 0).

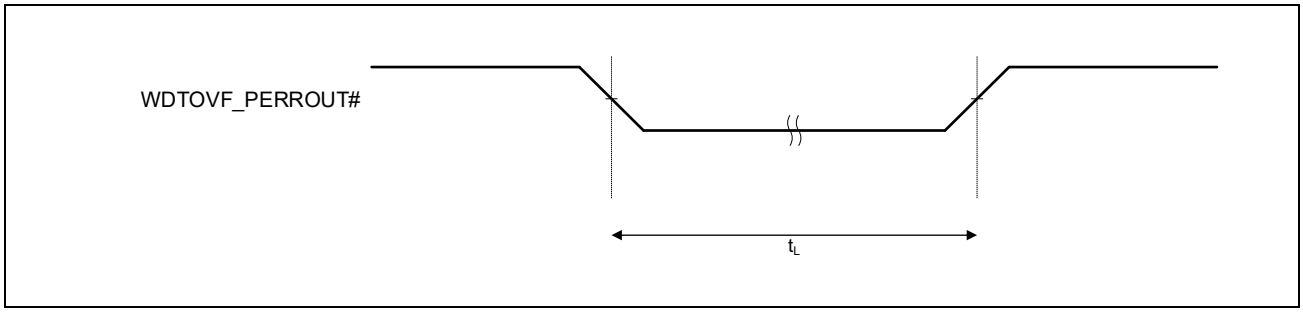
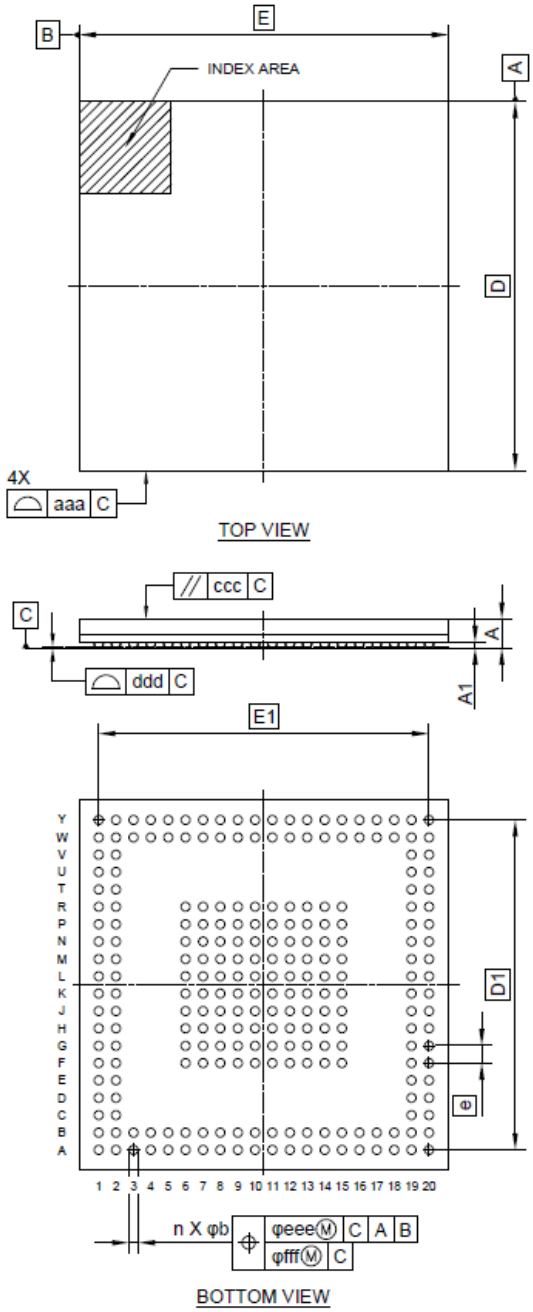



Figure 48.42 Watchdog Timer Output Timing

# APPENDIX A PACKAGE DIMENSIONS

## A.1 Package Dimensions

| JEITA Package code | RENESAS code | MASS(TYP.)[g] |
|---|---|---|
| P-LFBGA244-17x17-0.80 | PLBG0244GB-A | 0.69 |




| Reference Symbol | Dimension in Millimeters | | |
|---|---|---|---|
| | Min. | Nom. | Max. |
| D | – | 17.00 | – |
| E | – | 17.00 | – |
| D1 | – | 15.20 | – |
| E1 | – | 15.20 | – |
| A | – | – | 1.45 |
| A1 | 0.23 | – | – |
| b | 0.38 | 0.43 | 0.48 |
| e | – | 0.80 | – |
| aaa | – | – | 0.15 |
| ccc | – | – | 0.20 |
| ddd | – | – | 0.13 |
| eee | – | – | 0.15 |
| fff | – | – | 0.08 |
| n | – | 244 | – |

Figure A.1-1 Package Dimensions

| REVISION HISTORY | RZ/A3M Group User's Manual: Hardware |
|---|---|

| Rev. | Date | Description | |
|---|---|---|---|
| | | Page | Summary |
| 1.00 | Nov 30, 2023 | — | First edition issued |
| 1.05 | Mar 22, 2024 | 34. MIPI DSI | |
| | | 34-8, 34-9 | Table 34.3 Register list (2/3), (3/3)<br>Modified the access size of the SQCHxDSCyAR, SQCHxDSCyBR, SQCHxDSCyCR and SQCHxDSCyDR registers (8/16/32bit -> 32bit ) |
| | | 34-80 | 34.3.1.59 Sequence Channel x Descriptor y-A Register (SQCHxDSCyAR):<br>Corrected the description of the reserved bit (Bit 31 to 30) to "When writing, be sure to write 0". |
| | | 34-82 | 34.3.1.60 Sequence Channel x Descriptor y-B Register (SQCHxDSCyBR):<br>Corrected the description of the reserved bit (Bit 31 to 26, 23, 22, 21, 20, 19 to 16, 15 to 12, 11 to 0) to "When writing, be sure to write 0". |
| | | 34-83 | 34.3.1.61 Sequence Channel x Descriptor y-C Register (SQCHxDSCyCR):<br>Corrected the description of the reserved bit (Bit 23, 21 to 3) to "When writing, be sure to write 0". |
| | | 34-101 | Figure 34.9 Single Packet Transfer:<br>Removed the description of SQCHxDSCyBR.STT bit and SQCHxDSCyBR.REPEAT bit. |
| | | 34-103 | Figure 34.10 Single Packet Receive:<br>Removed the description of SQCHxDSCyBR.STT bit and SQCHxDSCyBR.REPEAT bit. |
| | | 34-104 | Figure 34.11 Trigger transmission:<br>Removed the description of SQCHxDSCyBR.STT bit and SQCHxDSCyBR.REPEAT bit. |
| 1.10 | Mar 31, 2025 | 1. Overview | |
| | | 1-5 | 1.2.7 Sound Interface, Sampling Rate Converter (SRC), changed to unsupported. |
| | | 1-9 | 1.4 Pin, 1.4.1 Pin Assignment, 1.4.2 External Pins, sections are added, and pin function list file is attached to this manual. |
| | | 5. LSI Internal Bus | |
| | | 5-2 | 5.1.2 Block Diagram of LSI Internal Bus, Figure 5.1 Configuration of LSI Internal Bus, Bus name corrected from MPU bus to MCPU bus, and SRC(1ch) changed to unsupported. |
| | | 5-6 | 5.1.2 Table 5.1 Detailed Address Space , Table 5.1 Detailed Address Space (3/3), SRC(Reg), and SRC (Memory) changed to unsupported. |
| | | 5-7 | 5.3 Accessible Areas, Table 5.2 Accesible Areas (1/2), SRC (Memory/Reg) line, changed to unsupported. |
| | | 7. CPG | |
| | | 7-4 | 7.1.4.1 Clock System Diagram, Figure 7.2 Clock System Diagram (1), M4$\phi$ clock and path is added. |
| | | 7-5 | 7.1.4.1 Clock System Diagram, Figure 7.2 Clock System Diagram (2), M1$\Phi$ clock and path is added. Signale names for around PLL5, and the note added. |
| | | 7-6 | 7.1.4.2 Clock List, changed to referring to clock list file attached to this manual. |
| | | 7-9 to 7-11 | 7.2.3 Register Configuration, Table 7.5 Register list, Related registers with SRC changed to unsupported. |
| | | 7-69 | 7.2.4.43 Clock Control Register SRC (CPG_CLKON_SRC), changed to unsupported. |
| | | 7-97 | 7.2.4.71 Clock Monitor Register SRC (CPG_CLKMON_SRC), changed to unsupported. |
| | | 7-126 | 7.2.4.98 Reset Control Register SRC (CPG_RST_SRC), changed to unsupported. |
| | | 7-154 | 7.2.4.125 Reset Monitor Register SRC (CPG_RSTMON_SRC), changed to unsupported. |
| | | 7-168 | 7.2.4.134 MSTOP Register MCPU1 (CPG_BUS_MCPU1_MSTOP), Bit-19, Bit-3 changed to reseeved. |
| | | 7-196 | 7.3.2.3 WDT System Reset, (4) Timing of Cortex-A55 Cold Reset due to the WDT Source, Figure 7.4 Note2. Description changed from CA55_RST0[0] to CA55_RST1[0]. |
| | | 7-199 | 7.4.2 Procedures for Supplying and Stopping Reset Signals, Setting for stopping/supplying the reset signal description with setting values are corrected |

| Rev. | Date | Description | |
|---|---|---|---|
| | | Page | Summary |
| 1.10 | Mar 31, 2025 | 7-199 | 7.4.2 Procedures for Supplying and Stopping Reset Signals, Supplying a Reset Singnal No.2 descrptions are added. |
| | | 8. Interrupt Controller | |
| | | 8-8 | 8.2 Interrupt Mapping, Table 8.2 Interrupt mapping (5/13),MIP-DSI, SPI 142, 143, Cause of interrupt name corrected, SPI 148 changed to reserved. |
| | | 8-10 | 8.2 Interrupt Mapping, Table 8.2 Interrupt mapping (7/13), TCIU8 of MTU3a (ch8) changed to unsupported. |
| | | 8-13 | 8.2 Interrupt Mapping,Table 8.2 Interrupt mapping (10/13), Interrupt Source of SRC changed to unsupported. |
| | | 14. Direct Memory Access Controller | |
| | | 14-25 | 14.4.9 Channel Configuration Register n (CHCFG_n), AM[2:0] bits descriptions area added for Auto-Request, and On-Chip Peripheral Module Request Mode. Bit setting for “000” is changed from “(initial value)” to “Not used”. |
| | | 14-47 | 14.5.2 DMA Transfer Requests, (2) On-Chip Peripheral Module Request Mode, Table 14.4 On-Chip Module Requests (3/4). Request Source of SRC is removed. |
| | | 14-88, 14-89 | 14.9 Points for Caution, Workaround description for “Problem of the Interrupt Signal Overtaking DMA Transfer” is added. |
| | | 16. Multi-Function Timer Pulse Unit 3 (MTU3a) | |
| | | 16-1 | Above of 16.1 Overview, A note for unsupporting A/D converter in this section is added. |
| | | 17. Port Output Enable 3 (POE3) | |
| | | 17-24, 17-25 | 17.3.1 MTU Pin Selection. Table 17.6 Correspondence between MTU Pins (1/2)(2/2). Corresponding Ports are corrected matching for Pin Function List. |
| | | 25. SPI Multi I/O Bus Controller | |
| | | 25-61 | 25.5.14 Timing Adjustment, Above of Figure 25.21 descriptions are corrected. |
| | | 27. Serial Sound Interface (SSIF-2) | |
| | | 27-1 | 27.1.1 Features, Table 27.1 Features of SSIF-2, Communication mode, removed about description of half-duplex function. |
| | | 27-3 | 27.1.2 Block Diagram, Figure 27.2 SSIF-2 Block Diagram, Peripheral bus name is corrected. SSIn_TXD and SSIn_RXD are separated, and Note for the part of full dupliex was removed. |
| | | 27-42 | 27.4.1.4 FIFO Status Register (SSIFSR), Bit 13 to 8, Bit Name is corrected from RDC[4:0] to RDC[5:0]. |
| | | 27-43 | 27.4.1.4 FIFO Status Register (SSIFSR), TDC[5:0] Bit, description of condition value with no space to write data is corrected from H’10 to H’20. |
| | | 27-43 | 27.4.1.4 FIFO Status Register (SSIFSR), TDE Bit, int the detail description, register name and bit name are corrected from “SSIFCR.TTRG” to “SSISCR.TDES”. |
| | | 27-44 | 27.4.1.4 FIFO Status Register (SSIFSR), TDE Bit, description in [Setting condition]. register name and bit name are corrected from “SSIFCR.TTRG” to “SSISCR.TDES”. |
| | | 27-58 | 27.4.2.2 Monaural Format, Description for the format, bit name is corrected from “SSICR.FLM[1:0]” to “SSICR.FRM[1:0]”. |
| | | 27-80 | 27.5.3 Interrupt Sources, Tabl 27.18 Interrupt Sources, INT_ssif_dma_rt’s line removed. |
| | | 27-83 | 27.5.3.4 INT_ssif_dma_rt Interrupt [half-duplex communication] removed section. |
| | | 27-85 | 27.6.1.1 Attention by communication using DMA, Table 27.19 Interrupt to expect DMA access,INT_ssif_dma_rt lines removed, INT_ssif_dma_rx line changed Half duplex to Full duplex. |
| | | 29. Sampling Rate Converter (SRC) | |
| | | ALL | Section of 29. SRC removed not to supported. |
| | | 34. MIPI DSI | |
| | | 34-5 | 34.1.3 Clock, Table 34.1 Clocks and the frequency range, Clock name MIPI_DSI_PLLCLK description corrected from “Transfer rate x 4” to “Transfer rate x 2”. |
| | | 34-109 | 34.4.2.4 Video-Input Operation (7) Delay aout video-input Operation, added description about referring attached file. |

| Rev. | Date | Description | |
|---|---|---|---|
| | | Page | Summary |
| 1.10 | Mar 31, 2025 | 34-115 | 34.4.2.6 Error Handling, (3) Timeout Error, (b) LP-RX Host Processor Timeout (LRX-H_TO) Error, The last paragraph corrected from “register.S” to “register.”. |
| | | 41. General Purpose Input Output Port (GPIO) | |
| | | 41-3 | 41.1.4 General Purpose Input Output Port Configuration, Note 1 description on Fiugre 41.1 P0_0 to P12_1, P20_0 to P23_1 (Multiplexed peripheral functions). Unavailable condition “SMT = 1 and” is removed. |
| | | 42. Low Power Mode | |
| | | 42-5 | 42.2.2.1 Setting of Clock control and MSTOP register, When exiting the Module Standby Mode, order of the condition is exchanged. |
| | | 48. Electrical Characteristics | |
| | | 48-7 | 48.4.8 DC Characteristics, Table 48.8 DC Characteristics (6) [1.8 V I/O (SD, QSPI)], VOH min’s multiplied value corrected from 0.7 to 0.8, and VOL max’s multipled value corrected from 0.3 to 0.2. |
| | | 48-9 | 48.5.8 DC Characteristics, Table 14 DC Characteristics (12) [Current Consumption], Max condition value of item (Power supply voltage (SPI/3.3V)) corrected from 160.00 mA to 50.00 mA. |
| | | 48-12 | 48.5.2.1 SDHI Access Timing (SDR 3.3-V), Table 48.16 SDHC AC Access Timing (SDR at 3.3-V Operation), Minimum condition value of Items (tSDWH, tSDWL, tSDODLY) are corrected. |
| | | 48-19 | 48.5.6 SPI Multi I/O Bus Controller Access Timing, Table 48.22 SPI Multi I/O Bus Contoller Access Timing, Minimum condition value of Item, “Clock cycle” is corrected. |

RZ/A3M Group User’s Manual: Hardware

Publication Date: Rev.1.00 Nov 30, 2023
Rev.1.10 Mar 31, 2025

Published by: Renesas Electronics Corporation

# RZ/A3M Group

Renesas Electronics Corporation

R01UH1038EJ0110