User Manual

32

V850E2/MN4

User's Manual: Hardware

RENESAS MCU
V850E2/Mx4 microcontrollers

μPD70F3510
μPD70F3512
μPD70F3514
μPD70F3515

Renesas Electronics
www.renesas.com

R01UH0011EJ0400, Rev.4.00
Sep 15, 2014

Notice

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

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

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

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

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

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

(2012.4)

General Precautions in the Handling of MPU/MCU Products

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

1. Handling of Unused Pins

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

2. Processing at Power-on

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

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

3. Prohibition of Access to Reserved Addresses

   Access to reserved addresses is prohibited.

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

4. Clock Signals

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

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

5. Differences between Products

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

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

How to Use This Manual

Readers

This manual is intended for users who wish to understand the functions of the V850E2/MN4 and design application systems using the V850E2/MN4 (μPD70F3510, 70F3512, 70F3514, and 70F3515).

Purpose

This manual is intended to give users an understanding of the hardware functions of the V850E2/MN4 shown in the Organization below.

Organization

This manual is divided into two parts: Hardware (this manual) and Architecture (V850E2M Architecture User's Manual).

Hardware	Architecture
Pin functions	Data types
CPU function	Register set
On-chip peripheral functions	Instruction format and instruction set
Flash memory programming	Interrupts and exceptions
Electrical specifications	Pipeline operation

How to read this manual

It is assumed that the readers of this manual have general knowledge in the fields of electrical engineering, logic circuits, and microcontrollers.

To understand the overall functions of the V850E2/MN4.

Read this manual according to the Contents.

To understand the details of an instruction function

Refer to the V850E2M Architecture User's Manual available separately.

The mark <R> shows major revised points.

The revised points can be easily searched by copying an "<R>" in the PDF file and specifying it in the "Find what:" field.

Chapter 1 Introduction

The V850E2/MN4 is a Renesas Electronics V850 single-chip microcontroller.

1.1 Overview

The V850E2/MN4 is a 32-bit single-chip microcontroller that integrates up to two V850E2M 32-bit RISC CPU cores, which were developed by enhancing the performance of the V850E1 CPU and V850E2 CPU. The V850E2/MN4 includes flash memory, RAM, and various peripheral functions, and is used to perform large-capacity data processing and advanced real-time control.

(1) V850E2M CPU

The V850E2M CPU further enhances the performance of the V850E2 CPU.

The V850E2M CPU improves the CPU processing performance through optimization of the two-way superscalar seven-stage pipeline control. A floating point unit (FPU) that conforms to IEEE754-1985 is also included. The V850E2/MN4 can include up to two V850E2M CPUs.

Note that, because the instruction codes are compatible with the V850, V850E1, and V850E2 CPUs at the object code level, conventional system software assets can be used as is.

(2) Multi-layer internal system bus

The internal system bus that connects each CPU and DMA controller to the peripheral functions has multiple layers. The bus has a total of three layers: one layer for each CPU, and one layer for the DMA controller. The multilayer internal system bus reduces the overhead due to bus arbitration, achieving high-speed real-time operation.

(3) On-chip flash memory

The V850E2/MN4 includes large-capacity flash memory for which high-speed access is possible, and, because programs can be rewritten with the V850E2/MN4 mounted in an application system, the system development time can be reduced. The maintainability can also be dramatically improved after shipment.

(4) External memory interface

Two memory controllers that have a separate-bus configuration are included as an external memory interface. Because a primary memory controller (suitable for high-speed access) that can be connected to SRAM/SDRAM and a secondary memory controller (suitable for large-capacity data processing) that can be connected to SRAM/SDRAM are included, the system performance can be improved.

In addition, the DMA controller can be used to perform CPU-internal calculations and data transfers while data is transferred with the external memory. Therefore, while large-capacity data such as video and audio data is being processed, it is possible to simultaneously perform real-time control such as motor and communication control, by using the internal flash memory and RAM to execute high-speed instructions.

(5) Various peripheral functions

The V850E2/MN4 includes peripheral functions such as a DMA controller, timer array, UART, CSI, CAN, A/D converter, USB function controller, USB host controller, Ethernet controller, on-chip debugger, and two memory controllers. Because a system can be set up without externally attaching the above functions, the cost, number of components, and mounting area can be reduced.

1.2 Features

Minimum instruction execution time	5.0 ns (during 200-MHz operation using internal system clock)
Instruction set	V850E2M
	Two-way superscalar seven-stage pipeline control
	General-purpose registers: 32 bits x 32 registers
	FPU that conforms to the ANSI/IEEE 754-1985 standards
	Up to two on-chip CPUs
Flash memory	Single-core product: 1 MB
	Dual-core product: 1 MB, 2 MB
Flash cache	Single-core product: 16 KB (four-way set associative)
	Dual-core product: 16 KB (four-way set associative) x 2
Internal RAM	Single-core product: 64 KB
	Dual-core product: 64 KB x 2
H-bus-shared memory	64 KB
Clock generator	Multiplied by 20 using a PLL

External bus interface	Two memory controllers
	Primary memory controller (can connect to SRAM/SDRAM)
	Secondary memory controller (can connect to SRAM/SDRAM)
Interrupts and exceptions	Non-maskable interrupt: 1 external source, internal T.B.D.
	Maskable interrupts: 28 external sources, internal T.B.D.
	16-level programmable priority control
DMA	DMA controller: 16 channels
	DTS: Up to 128 channels
I/O	Input: Seven pins, I/O: 181 pins
Timers	16-bit timer array: 16 channels x four units
	32-bit timer array: Four channels x one unit
	16-bit encoder timer: Two channels
	OS timer: One channel (single-core product) or two channels (dual-core product)
	Watchdog timer:
	One channel (single-core product) or two channels (dual-core product)
Serial interface	Asynchronous serial interface (UART): Six channels
	Clock synchronous serial interface (CSI): Six channels
	Asynchronous serial interface (UART) (FIFO): Four channels
	Clock synchronous serial interface (CSI) (FIFO): Four channels
	I^2C: Six channels
	CAN: Two channels
	USB function controller: One channel
	USB host controller: One channel
	Ethernet controller: One channel
A/D converter	10-bit resolution A/D converter: 12 channels
	(The resolution is 12 bits when using a 5.0 V A/D converter power supply.)
Package	304-pin plastic FBGA (19 x 19)
CMOS structure	Completely static circuit

Note The numbers of channels indicate how many are included in the product. The actual number of usable channels depends on the multiple-function pins.

1.3 Application Fields

The V850E2/MN4 can be used in servos, inverters, robots, NC machine tools, various printers, information appliances, home appliances, manufacturing devices, and other devices.

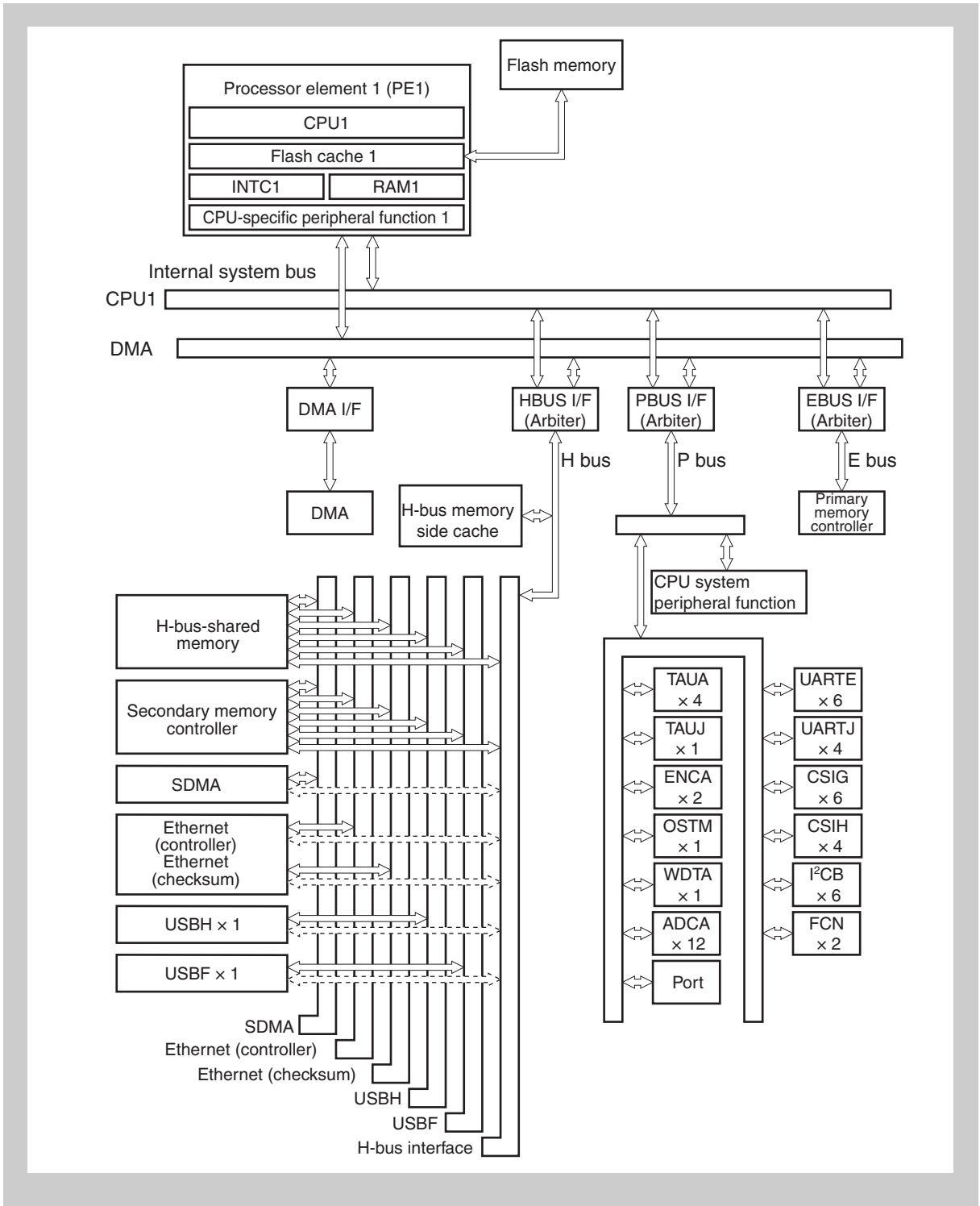
1.4 Ordering Information

Product Name	Package	CPU Core	Internal Flash Memory	Internal RAM	CAN/ Ethernet
μPD70F3510F1-HN6-A	304-pin plastic FBGA	Single	1 MB	64 KB	Not available
μPD70F3512F1-HN6-A	304-pin plastic FBGA	Single	1 MB	64 KB	Available
μPD70F3514F1-HN6-A	304-pin plastic FBGA	Dual	1 MB	64 KB x 2	Available
μPD70F3515F1-HN6-A	304-pin plastic FBGA	Dual	2 MB	64 KB x 2	Available

1.5 Function Block Diagrams

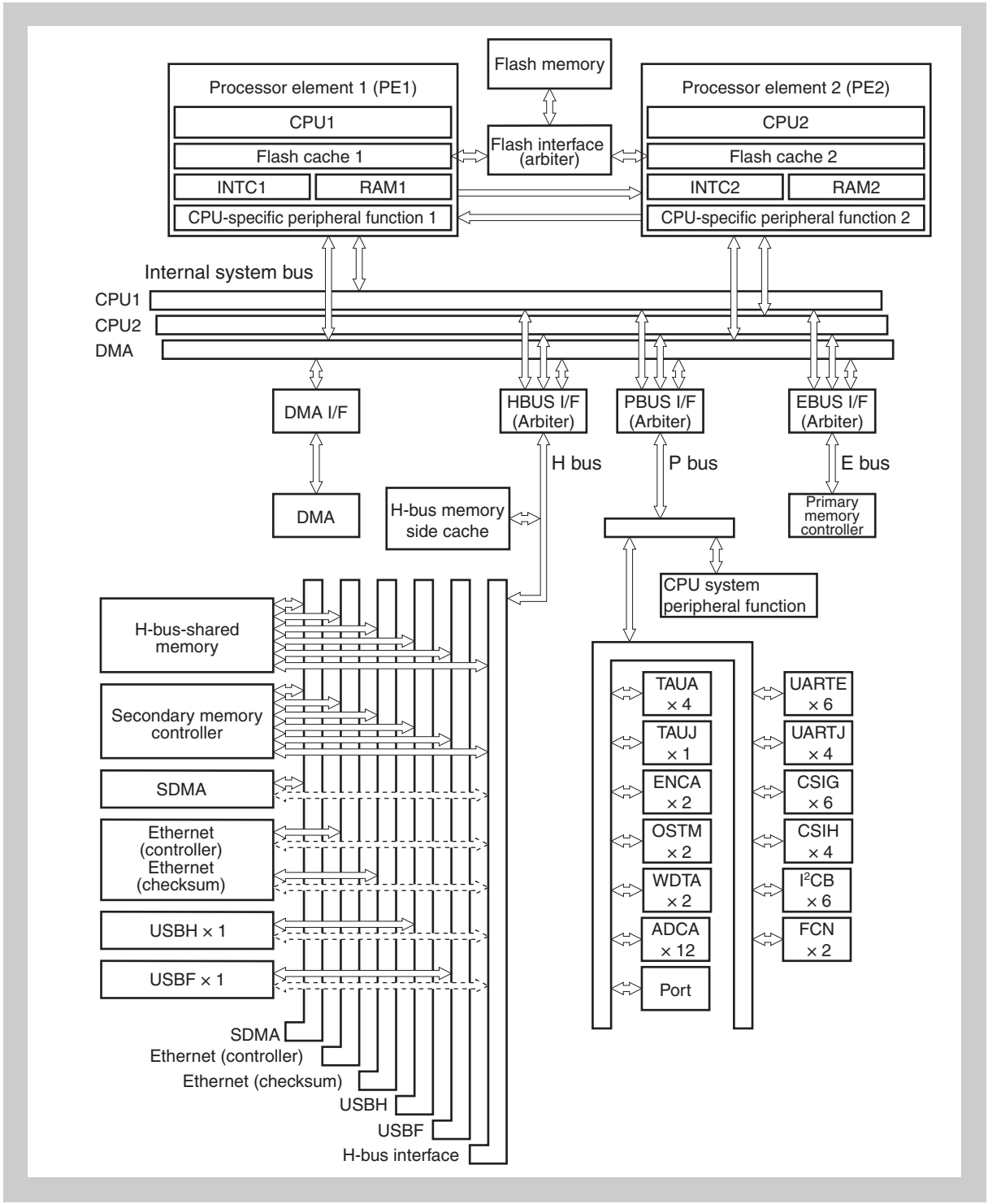
1.5.1 Single-core product

- μPD70F3510F1-HN6-A
- μPD70F3512F1-HN6-A



1.5.2 Dual-core product

- μPD70F3514F1-HN6-A
- μPD70F3515F1-HN6-A



Chapter 2 Pin Function

2.1 Pin Assignment

- 304-pin plastic FBGA (19 × 19)

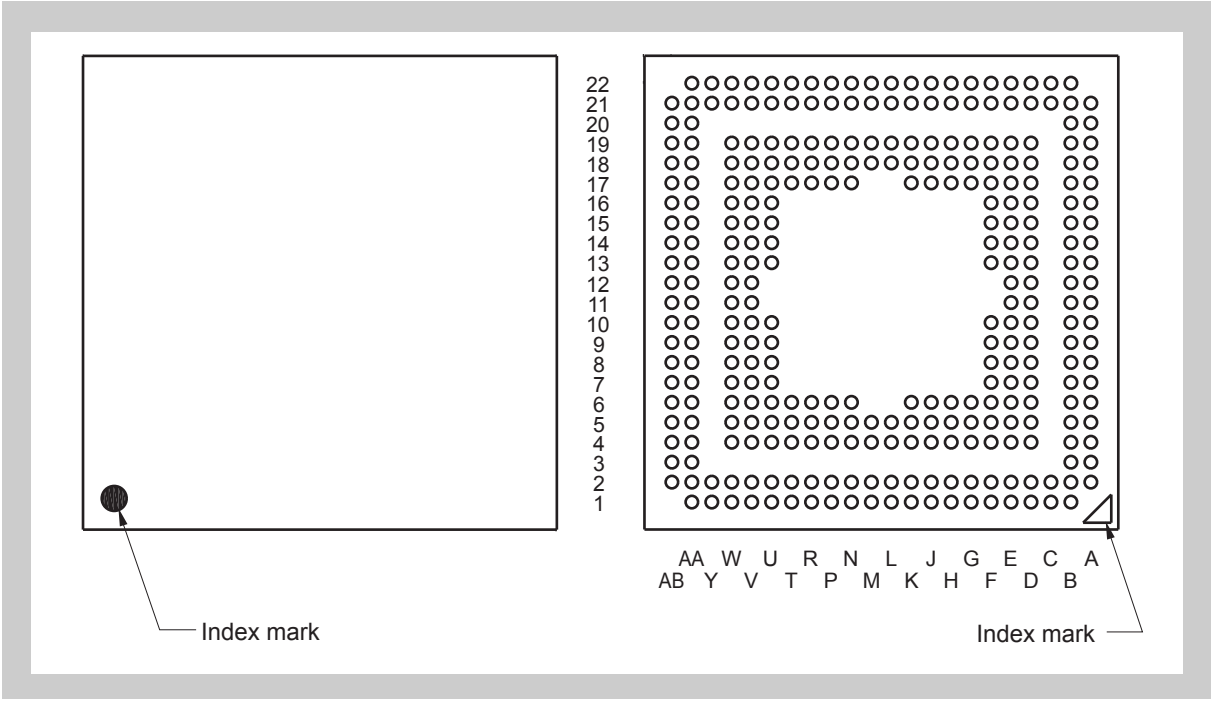


Table 2-1 Pin Assignment (1/8)

Pin No.	Name
A2	P7_1/S_D1/TA2_I1/TA2_O1
A3	P7_3/S_D3/TA2_I3/TA2_O3
A4	P7_5/S_D5/TA2_I5/TA2_O5
A5	P7_7/S_D7/TA2_I7/TA2_O7
A6	P7_9/S_D9/TA2_I9/TA2_O9
A7	P7_12/S_D12/TA2_I12/TA2_O12
A8	P7_14/S_D14/TA2_I14/TA2_O14
A9	P8_2/S_D18/TA0_I2/TA0_O2/TE0_TI1
A10	P8_6/S_D22/TA0_I6/TA0_O6/TE0_BI
A11	UDPH
A12	UDMH
A13	UDPF
A14	UDMF
A15	P6_1/ETH_MDIO/$\overline{\text{DMATC1}}$/TJ_I3/TJ_O3/P_A25
A16	P5_10/ETH_RXCLK/TA3_I10/TA3_O10/P_BUSRQ
A17	P5_7/ETH_TXCLK/TA3_I7/TA3_O7/INTP27/CSI0F_CS5
A18	P5_6/ETH_TXEN/TA3_I6/TA3_O6/INTP26/CSI0F_CS4/$\overline{\text{P\_SDCAS}}$
A19	P5_4/ETH_TXD1/TA3_I4/TA3_O4/INTP24/CSI0F_CS2/$\overline{\text{P\_REFRQ}}$

Table 2-1 Pin Assignment (2/8)

Pin No.	Name
A20	P5_2/ETH_TXD3/TA3_I2/TA3_O2/INTP22/CSI0F_CS0/P_ULDQM
A21	P5_0/ETH_CRS/TA3_I0/TA3_O0/INTP20/P_LLDQM/CSI0F_RYI/CSI0F_RYO
B1	V_{SS}
B2	P7_2/S_D2/TA2_I2/TA2_O2
B3	P7_4/S_D4/TA2_I4/TA2_O4
B4	P7_6/S_D6/TA2_I6/TA2_O6
B5	P7_8/S_D8/TA2_I8/TA2_O8
B6	P7_10/S_D10/TA2_I10/TA2_O10
B7	EV_{DD}
B8	V_{SS}
B9	P8_3/S_D19/TA0_I3/TA0_O3
B10	P8_7/S_D23/TA0_I7/TA0_O7
B11	P8_10/S_D26/TA0_I10/TA0_O10
B12	V_{SS}
B13	UV_{DD}
B14	P6_0/ETH_MDC/$\overline{\text{DMATC0}}$/TJ_I2/TJ_O2/P_A24
B15	P5_12/ETH_RXD0/TA3_I12/TA3_O12/$\overline{\text{P\_BCYST}}$
B16	EV_{DD}
B17	V_{SS}
B18	P5_5/ETH_TXD0/TA3_I5/TA3_O5/INTP25/CSI0F_CS3/$\overline{\text{P\_SDRAS}}$
B19	P5_3/ETH_TXD2/TA3_I3/TA3_O3/INTP23/CSI0F_CS1/P_UUDQM
B20	P5_1/ETH_COL/TA3_I1/TA3_O1/INTP21/CSI0F_SSI/P_LUDQM
B21	P4_12/$\overline{\text{P\_HLDAK}}$/INTP23/$\overline{\text{DMAAK2}}$/SO0F
B22	P4_13/$\overline{\text{P\_HLDRQ}}$/$\overline{\text{DMATC2}}$/SI0F/RXD0F/SDA0
C1	P9_1/S_BUSCLK/CSI2F_CS1
C2	P7_0/S_D0/TA2_I0/TA2_O0
C21	P4_10/$\overline{\text{P\_CS3}}$/INTP22/$\overline{\text{DMATC5}}$
C22	P4_11/$\overline{\text{P\_WAIT}}$/SCK0F/TXD0F/SCL0
D1	P9_4/S_LLDQM/TA3_I3/TA3_O3/CSI2F_CS4
D2	P9_3/$\overline{\text{S\_SDRAS}}$/TA3_I2/TA3_O2/CSI2F_CS3
D4	P9_0/S_SDCKE/TA3_I0/TA3_O0/INTP10/CSI2F_CS0
D5	P7_11/S_D11/TA2_I11/TA2_O11
D6	P7_13/S_D13/TA2_I13/TA2_O13
D7	P8_0/S_D16/TA0_I0/TA0_O0/TE0_TI0
D8	P8_1/S_D17/TA0_I1/TA0_O1
D9	P8_4/S_D20/TA0_I4/TA0_O4/TE0_AI
D10	P8_8/S_D24/TA0_I8/TA0_O8/TE0_ZI
D11	P8_11/S_D27/TA0_I11/TA0_O11
D12	P8_13/S_D29/TA0_I13/TA0_O13
D13	FLMD0
D14	P5_15/ETH_RXD3/TA3_I15/TA3_O15/$\overline{\text{DMAAK1}}$/TJ_I1/TJ_O1
D15	P5_13/ETH_RXD1/TA3_I13/TA3_O13
D16	P5_11/ETH_RXDV/TA3_I11/TA3_O11/$\overline{\text{P\_SDWE}}$
D17	P5_8/ETH_TXER/TA3_I8/TA3_O8/CSI0F_CS6/P_SDCKE

Table 2-1 Pin Assignment (3/8)

Pin No.	Name
D18	P5_9/ETH_RXER/TA3_I9/TA3_O9/CSI0F_CS7/$\overline{\text{P\_CS4}}$
D19	MODE2
D21	P4_8/$\overline{\text{P\_CS1}}$/$\overline{\text{P\_BCYST}}$/INTP20
D22	P4_9/$\overline{\text{P\_CS2}}$/INTP21/$\overline{\text{DMAAK5}}$
E1	P9_7/S_UUDQM/$\overline{\text{S\_DMATC3}}$/ESO1/CSI2F_CS7
E2	P9_5/S_LUDQM/TA3_I4/TA3_O4/CSI2F_CS5
E4	P9_2/$\overline{\text{S\_SDCAS}}$/TA3_I1/TA3_O1/CSI2F_CS2
E5	P7_15/S_D15/TA2_I15/TA2_O15
E6	MODE3
E7	V_{SS}
E8	EV_{DD}
E9	P8_5/S_D21/TA0_I5/TA0_O5
E10	P8_9/S_D25/TA0_I9/TA0_O9
E11	P8_12/S_D28/TA0_I12/TA0_O12
E12	P8_14/S_D30/TA0_I14/TA0_O14
E13	P8_15/S_D31/TA0_I15/TA0_O15
E14	P5_14/ETH_RXD2/TA3_I14/TA3_O14/$\overline{\text{DMAAK0}}$/TJ_I0/TJ_O0
E15	IV_{DD}
E16	V_{SS}
E17	EV_{DD}
E18	V_{SS}
E19	V_{SS}
E21	P4_6/P_A22/INTP11/$\overline{\text{DMAAK3}}$/SI3F/RXD3F/SDA3
E22	P4_7/P_A23/INTP12/SCK3F/TXD3F/SCL3/ADTRG20
F1	P9_9/$\overline{\text{S\_LUWR}}$/$\overline{\text{S\_DMARQ2}}$/$\overline{\text{S\_DMATC0}}$/SO4
F2	P9_8/$\overline{\text{S\_LLWR}}$/$\overline{\text{S\_DMAAK2}}$/CSI4_RYI/CSI4_RYO
F4	P9_6/S_ULDQM/$\overline{\text{S\_DMATC2}}$/ESO0/CSI2F_CS6
F5	EV_{DD}
F6	EV_{DD}
F7	V_{SS}
F8	EV_{DD}
F9	EV_{DD}
F10	V_{SS}
F13	V_{SS}
F14	EV_{DD}
F15	IV_{DD}
F16	V_{SS}
F17	EV_{DD}
F18	EV_{DD}
F19	P4_3/P_A19/INTP8/$\overline{\text{DMAAK4}}$/SO3F
F21	P4_4/P_A20/INTP9/$\overline{\text{DMATC4}}$/SI3/RXD3
F22	P4_5/P_A21/INTP10/$\overline{\text{DMATC3}}$/SCK3/TXD3/ADTRG10
G1	P9_12/$\overline{\text{S\_RD}}$/TA3_I5/TA3_O5/INTP11
G2	P9_11/$\overline{\text{S\_UUWR}}$/$\overline{\text{S\_DMARQ3}}$/SCK4/TXD4/SCL4/ESO3

Table 2-1 Pin Assignment (4/8)

Pin No.	Name
G4	P9_10/S_ULWR/S_DMAAK3/SI4/RXD4/SDA4/ESO2
G5	IV_{DD}
G6	IV_{DD}
G17	V_{SS}
G18	V_{SS}
G19	P4_0/P_A16/INTP5/CSI3F_RYI/CSI3F_RYO
G21	P4_1/P_A17/INTP6/CSI3F_SSI
G22	P4_2/P_A18/INTP7/SO3
H1	P9_14/S_LLBE/TA3_I7/TA3_O7/INTP13/CSI1_RYI/CSI1_RYO
H2	P9_13/S_WR/TA3_I6/TA3_O6/INTP12/CSI1_SSI
H4	P9_15/S_LUBE/INTP14/SO1
H5	V_{SS}
H6	V_{SS}
H17	IV_{DD}
H18	IV_{DD}
H19	P3_13/P_A13/TA1_I13/TA1_O13/CSI3F_CS7
H21	P3_14/P_A14/TA1_I14/TA1_O14/CSI3_RYI/CSI3_RYO
H22	P3_15/P_A15/TA1_I15/TA1_O15/CSI3_SSI
J1	P10_2/S_SDWE/TA3_I8/TA3_O8/INTP17/CSI2F_SSI
J2	P10_0/S_ULBE/INTP15/SI1/RXD1/S_DMATC0
J4	P10_1/S_UUBE/INTP16/SCK1/TXD1/S_DMATC1
J5	EV_{DD}
J6	EV_{DD}
J17	EV_{DD}
J18	EV_{DD}
J19	P3_10/P_A10/TA1_I10/TA1_O10/CSI3F_CS4
J21	P3_11/P_A11/TA1_I11/TA1_O11/CSI3F_CS5
J22	P3_12/P_A12/TA1_I12/TA1_O12/CSI3F_CS6
K1	P11_0/S_A1/TA1_I0/TA1_O0/TE1_TI0
K2	P10_3/S_BCYST/TA3_I9/TA3_O9/INTP18/S_DMATC1/CSI4_SSI
K4	P11_1/S_A2/TA1_I1/TA1_O1
K5	V_{SS}
K6	V_{SS}
K17	V_{SS}
K18	P3_4/P_A4/TA1_I4/TA1_O4/TE1_AI
K19	P3_7/P_A7/TA1_I7/TA1_O7/CSI3F_CS1
K21	P3_8/P_A8/TA1_I8/TA1_O8/CSI3F_CS2/TE1_ZI
K22	P3_9/P_A9/TA1_I9/TA1_O9/CSI3F_CS3
L1	P11_3/S_A4/TA1_I3/TA1_O3
L2	P11_2/S_A3/TA1_I2/TA1_O2/TE1_TI1
L4	P11_5/S_A6/TA1_I5/TA1_O5
L5	P11_4/S_A5/TA1_I4/TA1_O4/TE1_AI
L18	P3_3/P_A3/TA1_I3/TA1_O3
L19	P3_2/P_A2/TA1_I2/TA1_O2/TE1_TI1

Table 2-1 Pin Assignment (5/8)

Pin No.	Name
L21	P3_5/P_A5/TA1_I5/TA1_O5
L22	P3_6/P_A6/TA1_I6/TA1_O6/CSI3F_CS0/TE1_BI
M1	P11_7/S_A8/TA1_I7/TA1_O7
M2	P11_6/S_A7/TA1_I6/TA1_O6/TE1_BI
M4	P11_12/S_A13/TA1_I12/TA1_O12/CSI1F_CS2
M5	P11_13/S_A14/TA1_I13/TA1_O13/CSI1F_CS3
M18	P2_7/$\overline{\text{P\_WR}}$/$\overline{\text{P\_RW}}$/NTP19
M19	P3_0/P_A0/TA1_I0/TA1_O0/TE1_TI0/INTP18
M21	P3_1/P_A1/TA1_I1/TA1_O1
M22	V_{SS}
N1	P11_9/S_A10/TA1_I9/TA1_O9
N2	P11_8/S_A9/TA1_I8/TA1_O8/TE1_ZI
N4	P12_1/S_A18/INTP1/CSI1F_CS7/ADCNV1
N5	P12_2/S_A19/INTP2/ADCNV2/CSI0_SSI
N6	IV_{DD}
N17	EV_{DD}
N18	P2_3/$\overline{\text{P\_ULBE}}$/$\overline{\text{P\_ULWR}}$/INTP15/TJ_I2/TJ_O2
N19	P2_4/$\overline{\text{P\_UUBE}}$/$\overline{\text{P\_UUWR}}$/INTP16/TJ_I3/TJ_O3
N21	P2_5/$\overline{\text{P\_RD}}$/INTP17
N22	P2_6/P_BUSCLK
P1	P11_11/S_A12/TA1_I11/TA1_O11/CSI1F_CS1
P2	P11_10/S_A11/TA1_I10/TA1_O10/CSI1F_CS0
P4	P12_7/S_A24/INTP7/SO2F
P5	P12_8/S_A25/INTP8/SI2F/RXD2F/SDA2
P6	V_{SS}
P17	V_{SS}
P18	P0_1/P_D1/TA0_I1/TA0_O1
P19	P0_0/P_D0/TA0_I0/TA0_O0/TE0_TI0
P21	P2_1/$\overline{\text{P\_LLBE}}$/$\overline{\text{P\_LLWR}}$/INTP13/TJ_I0/TJ_O0
P22	P2_2/$\overline{\text{P\_LUBE}}$/$\overline{\text{P\_LUWR}}$/INTP14/TJ_I1/TJ_O1
R1	P11_15/S_A16/TA1_I15/TA1_O15/CSI1F_CS5
R2	P11_14/S_A15/TA1_I14/TA1_O14/CSI1F_CS4
R4	P10_7/$\overline{\text{S\_CS3}}$/$\overline{\text{S\_SDCS}}$/INTP26/CSI1F_RYI/CSI1F_RYO
R5	V_{SS}
R6	V_{SS}
R17	V_{SS}
R18	P0_5/P_D5/TA0_I5/TA0_O5
R19	P0_4/P_D4/TA0_I4/TA0_O4/TE0_AI
R21	P0_3/P_D3/TA0_I3/TA0_O3
R22	P0_2/P_D2/TA0_I2/TA0_O2/TE0_TI1
T1	P12_3/S_A20/INTP3/CSI0_RYI/CSI0_RYO
T2	P12_0/S_A17/INTP0/CSI1F_CS6/ADCNV0
T4	P10_11/$\overline{\text{S\_REFRQ}}$/TA3_I15/TA3_O15/CSI2F_RYI/CSI2F_RYO
T5	EV_{DD}

Table 2-1 Pin Assignment (6/8)

Pin No.	Name
T6	EV_{DD}
T17	IV_{DD}
T18	P0_9/P_D9/TA0_I9/TA0_O9
T19	P0_8/P_D8/TA0_I8/TA0_O8/TE0_ZI
T21	P0_7/P_D7/TA0_I7/TA0_O7
T22	P0_6/P_D6/TA0_I6/TA0_O6/TE0_BI
U1	P12_5/S_A22/INTP5/SI0/RXD0
U2	P12_4/S_A21/INTP4/SO0
U4	P10_10/$\overline{\text{S\_HLDRQ}}$/TA3_I14/TA3_O14/SCK2/TXD2
U5	P10_9/$\overline{\text{S\_HLDAK}}$/TA3_I13/TA3_O13/SO2/INTP27
U6	IV_{DD}
U7	V_{SS}
U8	EV_{DD}
U9	V_{SS}
U10	DV_{DD}
U13	V_{SS}
U14	$PLLV_{DD}$
U15	$PLLV_{SS}$
U16	IV_{DD}
U17	EV_{DD}
U18	V_{SS}
U19	P0_12/P_D12/TA0_I12/TA0_O12
U21	P0_11/P_D11/TA0_I11/TA0_O11
U22	P0_10/P_D10/TA0_I10/TA0_O10
V1	P12_6/S_A23/INTP6/SCK0/TXD0/ADTRG11
V2	P12_9/S_A26/INTP9/SCK2F/TXD2F/SCL2/ADTRG21
V4	P10_8/$\overline{\text{S\_WAIT}}$/SI2/RXD2
V5	P14_0/ANI06
V6	P14_4/ANI10
V7	ANI04
V8	ANI00
V9	MDO6
V10	MDO2
V11	$\overline{\text{EVTO}}$
V12	DV_{DD}
V13	V_{SS}
V14	$\overline{\text{TRST}}$
V15	V_{SS}
V16	IV_{DD}
V17	EV_{DD}
V18	V_{SS}
V19	V_{SS}
V21	P0_14/P_D14/TA0_I14/TA0_O14
V22	P0_13/P_D13/TA0_I13/TA0_O13

Table 2-1 Pin Assignment (7/8)

Pin No.	Name
W1	P10_4/$\overline{S\_CS0}$/TA3_I10/TA3_O10/INTP19/CSI2_SSI
W2	P10_5/$\overline{S\_CS1}$/TA3_I11/TA3_O11/INTP24/CSI2_RYI/CSI2_RYO
W4	P13_1/$\overline{S\_CS3}$/$\overline{S\_DMAAK0}$/ADTRG00/INTP0/CSI5_RYI/CSI5_RYO
W5	P14_1/ANI07
W6	P14_5/ANI11
W7	ANI05
W8	ANI01
W9	MDO7
W10	MDO3
W11	$\overline{EVTI}$
W12	MSEO0
W13	TDO/FLSO
W14	TDI/FLRXD/FLSI
W15	TMS
W16	P2_0/NMI
W17	P1_13/P_D29/TA2_I13/TA2_O13
W18	EV_{DD}
W19	P1_6/P_D22/TA2_I6/TA2_O6/ESO3
W21	P1_0/P_D16/TA2_I0/TA2_O0/ADCNV0
W22	P0_15/P_D15/TA0_I15/TA0_O15
Y1	P10_6/$\overline{S\_CS2}$/TA3_I12/TA3_O12/INTP25/CSI1F_SSI
Y2	P13_7/CAN1RXD/SI5/RXD5/SDA5
Y21	P1_2/P_D18/TA2_I2/TA2_O2/ADCNV2
Y22	P1_1/P_D17/TA2_I1/TA2_O1/ADCNV1
AA1	P13_6/CAN1TXD/SCK5/TXD5/SCL5/INTP4
AA2	P13_5/CAN0RXD/SI1F/RXD1F/SDA1
AA3	P13_3/$\overline{S\_DMAAK1}$/SO5/INTP2/$\overline{OCI}$
AA4	V_{SS}
AA5	P14_2/ANI08
AA6	AV_{REFM}
AA7	AV_{REFP}
AA8	ANI02
AA9	IV_{DD}
AA10	MDO4
AA11	MDO0
AA12	MSEO1
AA13	$\overline{TRDY}$
AA14	$OSCV_{DD}$
AA15	X2
AA16	FLMD1
AA17	P1_14/P_D30/TA2_I14/TA2_O14
AA18	P1_11/P_D27/TA2_I11/TA2_O11
AA19	P1_9/P_D25/TA2_I9/TA2_O9
AA20	P1_7/P_D23/TA2_I7/TA2_O7

Table 2-1 Pin Assignment (8/8)

Pin No.	Name
AA21	P1_4/P_D20/TA2_I4/TA2_O4/ESO1
AA22	P1_3/P_D19/TA2_I3/TA2_O3/ESO0
AB2	P13_4/CAN0TXD/SCK1F/TXD1F/SCL1/INTP3
AB3	P13_0/$\overline{\text{S\_DMARQ0}}$/ADTRG01/UCLK/CSI5_SSI
AB4	P13_2/$\overline{\text{S\_DMARQ1}}$/SO1F/INTP1/PPON
AB5	P14_3/ANI09
AB6	AV_{SS}
AB7	AV_{DD}
AB8	ANI03
AB9	V_{SS}
AB10	MDO5
AB11	MDO1
AB12	MCKO
AB13	TCK/FLSCK
AB14	$OSCV_{SS}$
AB15	X1
AB16	$\overline{\text{RESET}}$
AB17	P1_15/P_D31/TA2_I15/TA2_O15
AB18	P1_12/P_D28/TA2_I12/TA2_O12
AB19	P1_10/P_D26/TA2_I10/TA2_O10
AB20	P1_8/P_D24/TA2_I8/TA2_O8
AB21	P1_5/P_D21/TA2_I5/TA2_O5/ESO2

2.2 Pin Functions

Table 2-2 Port Functions (1/6)

Pin Name	Pin No.	I/O	Description	Alternate-Function Pin
P0_0	P19	I/O	Port 0 16-bit I/O port	P_D0/TA0_I0/TA0_O0/TE0_TI0
P0_1	P18			P_D1/TA0_I1/TA0_O1
P0_2	R22			P_D2/TA0_I2/TA0_O2/TE0_TI1
P0_3	R21			P_D3/TA0_I3/TA0_O3
P0_4	R19			P_D4/TA0_I4/TA0_O4/TE0_AI
P0_5	R18			P_D5/TA0_I5/TA0_O5
P0_6	T22			P_D6/TA0_I6/TA0_O6/TE0_BI
P0_7	T21			P_D7/TA0_I7/TA0_O7
P0_8	T19			P_D8/TA0_I8/TA0_O8/TE0_ZI
P0_9	T18			P_D9/TA0_I9/TA0_O9
P0_10	U22			P_D10/TA0_I10/TA0_O10
P0_11	U21			P_D11/TA0_I11/TA0_O11
P0_12	U19			P_D12/TA0_I12/TA0_O12
P0_13	V22			P_D13/TA0_I13/TA0_O13
P0_14	V21			P_D14/TA0_I14/TA0_O14
P0_15	W22			P_D15/TA0_I15/TA0_O15
P1_0	W21	I/O	Port 1 16-bit I/O port	P_D16/TA2_I0/TA2_O0/ADCNV0
P1_1	Y22			P_D17/TA2_I1/TA2_O1/ADCNV1
P1_2	Y21			P_D18/TA2_I2/TA2_O2/ADCNV2
P1_3	AA22			P_D19/TA2_I3/TA2_O3/ESO0
P1_4	AA21			P_D20/TA2_I4/TA2_O4/ESO1
P1_5	AB21			P_D21/TA2_I5/TA2_O5/ESO2
P1_6	W19			P_D22/TA2_I6/TA2_O6/ESO3
P1_7	AA20			P_D23/TA2_I7/TA2_O7
P1_8	AB20			P_D24/TA2_I8/TA2_O8
P1_9	AA19			P_D25/TA2_I9/TA2_O9
P1_10	AB19			P_D26/TA2_I10/TA2_O10
P1_11	AA18			P_D27/TA2_I11/TA2_O11
P1_12	AB18			P_D28/TA2_I12/TA2_O12
P1_13	W17			P_D29/TA2_I13/TA2_O13
P1_14	AA17			P_D30/TA2_I14/TA2_O14
P1_15	AB17			P_D31/TA2_I15/TA2_O15
P2_0	W16	Input	Port 2 1-bit input port	NMI

Table 2-2 Port Functions (2/6)

Pin Name	Pin No.	I/O	Description	Alternate-Function Pin
P2_1	P21	I/O	Port 2 7-bit I/O port	$\overline{\text{P\_LLBE}}$/$\overline{\text{P\_LLWR}}$/INTP13/TJ_I0/TJ_O0
P2_2	P22			$\overline{\text{P\_LUBE}}$/$\overline{\text{P\_LUWR}}$/INTP14/TJ_I1/TJ_O1
P2_3	N18			$\overline{\text{P\_ULBE}}$/$\overline{\text{P\_ULWR}}$/INTP15/TJ_I2/TJ_O2
P2_4	N19			$\overline{\text{P\_UUBE}}$/$\overline{\text{P\_UUWR}}$/INTP16/TJ_I3/ TJ_O3
P2_5	N21			$\overline{\text{P\_RD}}$/INTP17
P2_6	N22			P_BUSCLK
P2_7	M18			$\overline{\text{P\_WR}}$/$\overline{\text{P\_RW}}$/INTP19
P3_0	M19	I/O	Port 3 16-bit I/O port	P_A0/TA1_I0/TA1_O0/TE1_TI0/INTP18
P3_1	M21			P_A1/TA1_I1/TA1_O1
P3_2	L19			P_A2/TA1_I2/TA1_O2/TE1_TI1
P3_3	L18			P_A3/TA1_I3/TA1_O3
P3_4	K18			P_A4/TA1_I4/TA1_O4/TE1_AI
P3_5	L21			P_A5/TA1_I5/TA1_O5
P3_6	L22			P_A6/TA1_I6/TA1_O6/CSI3F_CS0/ TE1_BI
P3_7	K19			P_A7/TA1_I7/TA1_O7/CSI3F_CS1
P3_8	K21			P_A8/TA1_I8/TA1_O8/CSI3F_CS2/ TE1_ZI
P3_9	K22			P_A9/TA1_I9/TA1_O9/CSI3F_CS3
P3_10	J19			P_A10/TA1_I10/TA1_O10/CSI3F_CS4
P3_11	J21			P_A11/TA1_I11/TA1_O11/CSI3F_CS5
P3_12	J22			P_A12/TA1_I12/TA1_O12/CSI3F_CS6
P3_13	H19			P_A13/TA1_I13/TA1_O13/CSI3F_CS7
P3_14	H21			P_A14/TA1_I14/TA1_O14/CSI3_RYI/ CSI3_RYO
P3_15	H22			P_A15/TA1_I15/TA1_O15/CSI3_SSI
P4_0	G19	I/O	Port 4 14-bit I/O port	P_A16/INTP5/CSI3F_RYI/CSI3F_RYO
P4_1	G21			P_A17/INTP6/CSI3F_SSI
P4_2	G22			P_A18/INTP7/SO3
P4_3	F19			P_A19/INTP8/$\overline{\text{DMAAK4}}$/SO3F
P4_4	F21			P_A20/INTP9/$\overline{\text{DMATC4}}$/SI3/RXD3
P4_5	F22			P_A21/INTP10/$\overline{\text{DMATC3}}$/SCK3/TXD3/ ADTRG10
P4_6	E21			P_A22/INTP11/$\overline{\text{DMAAK3}}$/SI3F/RXD3F/ SDA3
P4_7	E22			P_A23/INTP12/SCK3F/TXD3F/SCL3/ ADTRG20
P4_8	D21			$\overline{\text{P\_CS1}}$/$\overline{\text{P\_BCYST}}$/INTP20
P4_9	D22			$\overline{\text{P\_CS2}}$/INTP21/$\overline{\text{DMAAK5}}$
P4_10	C21			$\overline{\text{P\_CS3}}$/INTP22/$\overline{\text{DMATC5}}$
P4_11	C22			$\overline{\text{P\_WAIT}}$/SCK0F/TXD0F/SCL0
P4_12	B21			$\overline{\text{P\_HLDAK}}$/INTP23/$\overline{\text{DMAAK2}}$/SO0F
P4_13	B22			$\overline{\text{P\_HLDRQ}}$/$\overline{\text{DMATC2}}$/SI0F/RXD0F/SDA0

Table 2-2 Port Functions (3/6)

Pin Name	Pin No.	I/O	Description	Alternate-Function Pin
P5_0	A21	I/O	Port 5 16-bit I/O port	ETH_CRS/TA3_I0/TA3_O0/INTP20/ P_LLDQM/CSI0F_RYI/CSI0F_RYO
P5_1	B20			ETH_COL/TA3_I1/TA3_O1/INTP21/ CSI0F_SSI/P_LUDQM
P5_2	A20			ETH_TXD3/TA3_I2/TA3_O2/INTP22/ CSI0F_CS0/P_ULDQM
P5_3	B19			ETH_TXD2/TA3_I3/TA3_O3/INTP23/ CSI0F_CS1/P_UUDQM
P5_4	A19			ETH_TXD1/TA3_I4/TA3_O4/INTP24/ CSI0F_CS2/$\overline{\text{P\_REFRQ}}$
P5_5	B18			ETH_TXD0/TA3_I5/TA3_O5/INTP25/ CSI0F_CS3/$\overline{\text{P\_SDRAS}}$
P5_6	A18			ETH_TXEN/TA3_I6/TA3_O6/INTP26/ CSI0F_CS4/$\overline{\text{P\_SDCAS}}$
P5_7	A17			ETH_TXCLK/TA3_I7/TA3_O7/INTP27/ CSI0F_CS5
P5_8	D17			ETH_TXER/TA3_I8/TA3_O8/ CSI0F_CS6/P_SDCKE
P5_9	D18			ETH_RXER/TA3_I9/TA3_O9/ CSI0F_CS7/$\overline{\text{P\_CS4}}$
P5_10	A16			ETH_RXCLK/TA3_I10/TA3_O10/ P_BUSRQ
P5_11	D16			$\overline{\text{ETH\_RXDV}}$/TA3_I11/TA3_O11/ $\overline{\text{P\_SDWE}}$
P5_12	B15			$\overline{\text{ETH\_RXD0}}$/TA3_I12/TA3_O12/ $\overline{\text{P\_BCYST}}$
P5_13	D15			ETH_RXD1/TA3_I13/TA3_O13
P5_14	E14			$\overline{\text{ETH\_RXD2}}$/TA3_I14/TA3_O14/ $\overline{\text{DMAAK0}}$/TJ_I0/TJ_O0
P5_15	D14			$\overline{\text{ETH\_RXD3}}$/TA3_I15/TA3_O15/ $\overline{\text{DMAAK1}}$/TJ_I1/TJ_O1
P6_0	B14	I/O	Port 6 2-bit I/O port	ETH_MDC/$\overline{\text{DMATC0}}$/TJ_I2/TJ_O2/ P_A24
P6_1	A15			ETH_MDIO/$\overline{\text{DMATC1}}$/TJ_I3/TJ_O3/ P_A25

Table 2-2 Port Functions (4/6)

Pin Name	Pin No.	I/O	Description	Alternate-Function Pin
P7_0	C2	I/O	Port 7 16-bit I/O port	S_D0/TA2_I0/TA2_O0
P7_1	A2			S_D1/TA2_I1/TA2_O1
P7_2	B2			S_D2/TA2_I2/TA2_O2
P7_3	A3			S_D3/TA2_I3/TA2_O3
P7_4	B3			S_D4/TA2_I4/TA2_O4
P7_5	A4			S_D5/TA2_I5/TA2_O5
P7_6	B4			S_D6/TA2_I6/TA2_O6
P7_7	A5			S_D7/TA2_I7/TA2_O7
P7_8	B5			S_D8/TA2_I8/TA2_O8
P7_9	A6			S_D9/TA2_I9/TA2_O9
P7_10	B6			S_D10/TA2_I10/TA2_O10
P7_11	D5			S_D11/TA2_I11/TA2_O11
P7_12	A7			S_D12/TA2_I12/TA2_O12
P7_13	D6			S_D13/TA2_I13/TA2_O13
P7_14	A8			S_D14/TA2_I14/TA2_O14
P7_15	E5			S_D15/TA2_I15/TA2_O15
P8_0	D7	I/O	Port 8 16-bit I/O port	S_D16/TA0_I0/TA0_O0/TE_TI0
P8_1	D8			S_D17/TA0_I1/TA0_O1
P8_2	A9			S_D18/TA0_I2/TA0_O2/TE0_TI1
P8_3	B9			S_D19/TA0_I3/TA0_O3
P8_4	D9			S_D20/TA0_I4/TA0_O4/TE0_AI
P8_5	E9			S_D21/TA0_I5/TA0_O5
P8_6	A10			S_D22/TA0_I6/TA0_O6/TE0_BI
P8_7	B10			S_D23/TA0_I7/TA0_O7
P8_8	D10			S_D24/TA0_I8/TA0_O8/TE0_ZI
P8_9	E10			S_D25/TA0_I9/TA0_O9
P8_10	B11			S_D26/TA0_I10/TA0_O10
P8_11	D11			S_D27/TA0_I11/TA0_O11
P8_12	E11			S_D28/TA0_I12/TA0_O12
P8_13	D12			S_D29/TA0_I13/TA0_O13
P8_14	E12			S_D30/TA0_I14/TA0_O14
P8_15	E13			S_D31/TA0_I15/TA0_O15

Table 2-2 Port Functions (5/6)

Pin Name	Pin No.	I/O	Description	Alternate-Function Pin
P9_0	D4	I/O	Port 9 16-bit I/O port	S_SDCKE/TA3_I0/TA3_O0/INTP10/ CSI2F_CS0
P9_1	C1			S_BUSCLK/CSI2F_CS1
P9_2	E4			$\overline{\text{S\_SDCAS}}$/TA3_I1/TA3_O1/CSI2F_CS2
P9_3	D2			$\overline{\text{S\_SDRAS}}$/TA3_I2/TA3_O2/CSI2F_CS3
P9_4	D1			S_LLDQM/TA3_I3/TA3_O3/CSI2F_CS4
P9_5	E2			S_LUDQM/TA3_I4/TA3_O4/CSI2F_CS5
P9_6	F4			S_ULDQM/$\overline{\text{S\_DMATC2}}$/ESO0/ CSI2F_CS6
P9_7	E1			S_UUDQM/$\overline{\text{S\_DMATC3}}$/ESO1/ CSI2F_CS7
P9_8	F2			$\overline{\text{S\_LLWR}}$/$\overline{\text{S\_DMAAK2}}$/CSI4_RYI/ CSI4_RYO
P9_9	F1			$\overline{\text{S\_LUWR}}$/$\overline{\text{S\_DMARQ2}}$/$\overline{\text{S\_DMATC0}}$/SO4
P9_10	G4			$\overline{\text{S\_ULWR}}$/$\overline{\text{S\_DMAAK3}}$/SI4/RXD4/SDA4/ ESO2
P9_11	G2			$\overline{\text{S\_UUWR}}$/$\overline{\text{S\_DMARQ3}}$/SCK4/TXD4/ SCL4/ESO3
P9_12	G1			$\overline{\text{S\_RD}}$/TA3_I5/TA3_O5/INTP11
P9_13	H2			$\overline{\text{S\_WR}}$/TA3_I6/TA3_O6/INTP12/ CSI1_SSI
P9_14	H1			$\overline{\text{S\_LLBE}}$/TA3_I7/TA3_O7/INTP13/ CSI1_RYI/CSI1_RYO
P9_15	H4			$\overline{\text{S\_LUBE}}$/INTP14/SO1
P10_0	J2	I/O	Port 10 12-bit I/O port	$\overline{\text{S\_ULBE}}$/INTP15/SI1/RXD1/$\overline{\text{S\_DMATC0}}$
P10_1	J4			$\overline{\text{S\_UUBE}}$/INTP16/SCK1/TXD1/ $\overline{\text{S\_DMATC1}}$
P10_2	J1			$\overline{\text{S\_SDWE}}$/TA3_I8/TA3_O8/INTP17/ CSI2F_SSI
P10_3	K2			$\overline{\text{S\_BCYST}}$/TA3_I9/TA3_O9/INTP18/ $\overline{\text{S\_DMATC1}}$/CSI4_SSI
P10_4	W1			$\overline{\text{S\_CS0}}$/TA3_I10/TA3_O10/INTP19/ CSI2_SSI
P10_5	W2			$\overline{\text{S\_CS1}}$/TA3_I11/TA3_O11/INTP24/ CSI2_RYI/CSI2_RYO
P10_6	Y1			$\overline{\text{S\_CS2}}$/TA3_I12/TA3_O12/INTP25/ CSI1F_SSI
P10_7	R4			$\overline{\text{S\_CS3}}$/$\overline{\text{S\_SDCS}}$/INTP26/CSI1F_RYI/ CSI1F_RYO
P10_8	V4			$\overline{\text{S\_WAIT}}$/SI2/RXD2
P10_9	U5			$\overline{\text{S\_HLDAK}}$/TA3_I13/TA3_O13/SO2/ INTP27
P10_10	U4			$\overline{\text{S\_HLDRQ}}$/TA3_I14/TA3_O14/SCK2/ TXD2
P10_11	T4			$\overline{\text{S\_REFRQ}}$/TA3_I15/TA3_O15/ CSI2F_RYI/CSI2F_RYO

Table 2-2 Port Functions (6/6)

Pin Name	Pin No.	I/O	Description	Alternate-Function Pin
P11_0	K1	I/O	Port 11 16-bit I/O port	S_A1/TA1_I0/TA1_O0/TE1_TI0
P11_1	K4			S_A2/TA1_I1/TA1_O1
P11_2	L2			S_A3/TA1_I2/TA1_O2/TE1_TI1
P11_3	L1			S_A4/TA1_I3/TA1_O3
P11_4	L5			S_A5/TA1_I4/TA1_O4/TE1_AI
P11_5	L4			S_A6/TA1_I5/TA1_O5
P11_6	M2			S_A7/TA1_I6/TA1_O6/TE1_BI
P11_7	M1			S_A8/TA1_I7/TA1_O7
P11_8	N2			S_A9/TA1_I8/TA1_O8/TE1_ZI
P11_9	N1			S_A10/TA1_I9/TA1_O9
P11_10	P2			S_A11/TA1_I10/TA1_O10/CSI1F_CS0
P11_11	P1			S_A12/TA1_I11/TA1_O11/CSI1F_CS1
P11_12	M4			S_A13/TA1_I12/TA1_O12/CSI1F_CS2
P11_13	M5			S_A14/TA1_I13/TA1_O13/CSI1F_CS3
P11_14	R2			S_A15/TA1_I14/TA1_O14/CSI1F_CS4
P11_15	R1			S_A16/TA1_I15/TA1_O15/CSI1F_CS5
P12_0	T2	I/O	Port 12 10-bit I/O port	S_A17/INTP0/CSI1F_CS6/ADCNV0
P12_1	N4			S_A18/INTP1/CSI1F_CS7/ADCNV1
P12_2	N5			S_A19/INTP2/ADCNV2/CSI0_SSI
P12_3	T1			S_A20/INTP3/CSI0_RYI/CSI0_RYO
P12_4	U2			S_A21/INTP4/SO0
P12_5	U1			S_A22/INTP5/SI0/RXD0
P12_6	V1			S_A23/INTP6/SCK0/TXD0/ADTRG11
P12_7	P4			S_A24/INTP7/SO2F
P12_8	P5			S_A25/INTP8/SI2F/RXD2F/SDA2
P12_9	V2			S_A26/INTP9/SCK2F/TXD2F/SCL2/ADTRG21
P13_0	AB3	I/O	Port 13 8-bit I/O port	$\overline{\text{S\_DMARQ0}}$/ADTRG01/UCLK/CSI5_SSI
P13_1	W4			$\overline{\text{S\_CS3}}$/$\overline{\text{S\_DMAAK0}}$/ADTRG00/INTP0/CSI5_RYI/CSI5_RYO
P13_2	AB4			$\overline{\text{S\_DMARQ1}}$/SO1F/INTP1/PPON
P13_3	AA3			$\overline{\text{S\_DMAAK1}}$/SO5/INTP2/$\overline{\text{OCI}}$
P13_4	AB2			CAN0TXD/SCK1F/TXD1F/SCL1/INTP3
P13_5	AA2			CAN0RXD/SI1F/RXD1F/SDA1
P13_6	AA1			CAN1TXD/SCK5/TXD5/SCL5/INTP4
P13_7	Y2			CAN1RXD/SI5/RXD5/SDA5
P14_0	V5	Input	Port 14 6-bit I/O port	ANI06
P14_1	W5			ANI07
P14_2	AA5			ANI08
P14_3	AB5			ANI09
P14_4	V6			ANI10
P14_5	W6			ANI11

Table 2-3 External Bus Interface Pins (1/6)

Pin Name	Pin No.	I/O	Description	Alternate-Function Pin
P_D0	P19	I/O	Primary memory controller (PMEM) 32-bit data bus for external memory	P0_0/TA0_I0/TA0_O0/TE0_TI0
P_D1	P18			P0_1/TA0_I1/TA0_O1
P_D2	R22			P0_2/TA0_I2/TA0_O2/TE0_TI1
P_D3	R21			P0_3/TA0_I3/TA0_O3
P_D4	R19			P0_4/TA0_I4/TA0_O4/TE0_AI
P_D5	R18			P0_5/TA0_I5/TA0_O5
P_D6	T22			P0_6/TA0_I6/TA0_O6/TE0_BI
P_D7	T21			P0_7/TA0_I7/TA0_O7
P_D8	T19			P0_8/TA0_I8/TA0_O8/TE0_ZI
P_D9	T18			P0_9/TA0_I9/TA0_O9
P_D10	U22			P0_10/TA0_I10/TA0_O10
P_D11	U21			P0_11/TA0_I11/TA0_O11
P_D12	U19			P0_12/TA0_I12/TA0_O12
P_D13	V22			P0_13/TA0_I13/TA0_O13
P_D14	V21			P0_14/TA0_I14/TA0_O14
P_D15	W22			P0_15/TA0_I15/TA0_O15
P_D16	W21			P1_0/TA2_I0/TA2_O0/ADCNV0
P_D17	Y22			P1_1/TA2_I1/TA2_O1/ADCNV1
P_D18	Y21			P1_2/TA2_I2/TA2_O2/ADCNV2
P_D19	AA22			P1_3/TA2_I3/TA2_O3/ES00
P_D20	AA21			P1_4/TA2_I4/TA2_O4/ESO1
P_D21	AB21			P1_5/TA2_I5/TA2_O5/ESO2
P_D22	W19			P1_6/TA2_I6/TA2_O6/ESO3
P_D23	AA20			P1_7/TA2_I7/TA2_O7
P_D24	AB20			P1_8/TA2_I8/TA2_O8
P_D25	AA19			P1_9/TA2_I9/TA2_O9
P_D26	AB19			P1_10/TA2_I10/TA2_O10
P_D27	AA18			P1_11/TA2_I11/TA2_O11
P_D28	AB18			P1_12/TA2_I12/TA2_O12
P_D29	W17			P1_13/TA2_I13/TA2_O13
P_D30	AA17			P1_14/TA2_I14/TA2_O14
P_D31	AB17			P1_15/TA2_I15/TA2_O15
$\overline{\text{P\_LLBE}}$	P21	Output	Byte enable signal output for external data bus (D0 to D7) of PMEMC	P2_1/$\overline{\text{P\_LLWR}}$/INTP13/TJ_I0/TJ_O0
$\overline{\text{P\_LUBE}}$	P22	Output	Byte enable signal output of external data bus (D8 to D15) of PMEMC	P2_2/$\overline{\text{P\_LUWR}}$/INTP14/TJ_I1/TJ_O1
$\overline{\text{P\_ULBE}}$	N18	Output	Byte enable signal output of external data bus (D16 to D23) of PMEMC	P2_3/$\overline{\text{P\_ULWR}}$/INTP15/TJ_I2/TJ_O2

Table 2-3 External Bus Interface Pins (2/6)

Pin Name	Pin No.	I/O	Description	Alternate-Function Pin
$\overline{\text{P\_UUBE}}$	N19	Output	Byte enable signal output for external data bus (D24 to D31) of PMEMC	P2_4/$\overline{\text{P\_UUWR}}$/INTP16/TJ_I3/TJ_O3
$\overline{\text{P\_RD}}$	N21	Output	Read strobe signal output of external data bus of PMEMC	P2_5/INTP17
P_BUSCLK	N22	Output	Bus clock output of PMEMC	P2_6
$\overline{\text{P\_WR}}$	M18	Output	Write strobe signal output for external data bus of PMEMC	P2_7/$\overline{\text{P\_RW}}$/INTP19
$\overline{\text{P\_RW}}$	M18	Output	Read/write status signal output for external data bus of PMEMC	P2_7/$\overline{\text{P\_WR}}$/INTP19
$\overline{\text{P\_LLWR}}$	P21	Output	Write strobe signal output of external data bus (D0 to D7) of PMEMC	P2_1/$\overline{\text{P\_LLBE}}$/INTP13/TJ_I0/TJ_O0
$\overline{\text{P\_LUWR}}$	P22	Output	Write strobe signal output of external data bus (D8 to 15) of PMEMC	P2_2/$\overline{\text{P\_LUBE}}$/INTP14/TJ_I1/TJ_O1
$\overline{\text{P\_ULWR}}$	N18	Output	Write strobe signal output for external data bus (D16 to D23) of PMEMC	P2_3/$\overline{\text{P\_ULBE}}$/INTP15/TJ_I2/TJ_O2
$\overline{\text{P\_UUWR}}$	N19	Output	Write strobe signal output for external data bus (D24 to D31) of PMEMC	P2_4/$\overline{\text{P\_UUBE}}$/INTP16/TJ_I3/TJ_O3
P_LLDQM	A21	Output	I/O mask signal output for SDRAM (D0 to D7) of PMEMC	P5_0/ETH_CRS/TA3_I0/TA3_O0/ INTP20/CSI0F_RYI/CSI0F_RYO
P_LUDQM	B20	Output	I/O mask signal output for SDRAM (D8 to D15) of PMEMC	P5_1/ETH_COL/TA3_I1/TA3_O1/ INTP21/CSI0F_SSI
P_ULDQM	A20	Output	I/O mask signal output for SDRAM (D16 to D23) of PMEMC	P5_2/ETH_TXD3/TA3_I2/TA3_O2/ INTP22/CSI0F_CS0
P_UUDQM	B19	Output	I/O mask signal output for SDRAM (D24 to D31) of PMEMC	P5_3/ETH_TXD2/TA3_I3/TA3_O3/ INTP23/CSI0F_CS1
$\overline{\text{P\_REFRQ}}$	A19	Output	Refresh request of PMEMC	P5_4/ETH_TXD1/TA3_I4/TA3_O4/ INTP24/CSI0F_CS2
$\overline{\text{P\_SDRAS}}$	B18	Output	Row address strobe signal output for SDRAM of PMEMC	P5_5/ETH_TXD0/TA3_I5/TA3_O5/ INTP25/CSI0F_CS3
$\overline{\text{P\_SDCAS}}$	A18	Output	Column address strobe signal output for SDRAM of PMEMC	P5_6/ETH_TXEN/TA3_I6/TA3_O6/ INTP26/CSI0F_CS4
P_SDCKE	D17	Output	SDRAM clock enable output signal of PMEMC	P5_8/ETH_TXER/TA3_I8/TA3_O8/ CSI0F_CS6
P_BUSRQ	A16	Output	CPU bus request report output of PMEMC	P5_10/ETH_RXCLK/TA3_I10/TA3_O10
$\overline{\text{P\_SDWE}}$	D16	Output	Data write enable output for SDRAM of PMEMC	P5_11/ETH_RXDV/TA3_I11/TA3_O11
$\overline{\text{P\_BCYST}}$	B15	Output	Bus cycle start for SDRAM of PMEMC	P5_12/ETH_RXD0/TA3_I12/TA3_O12
	D21			P4_8/$\overline{\text{P\_CS1}}$/INTP20

Table 2-3 External Bus Interface Pins (3/6)

Pin Name	Pin No.	I/O	Description	Alternate-Function Pin
P_A0	M19	Output	Primary memory controller (PMEMC) 26-bit address bus for external memory	P3_0/TA1_I0/TA1_O0/TE1_TI0/INTP18
P_A1	M21			P3_1/TA1_I1/TA1_O1
P_A2	L19			P3_2/TA1_I2/TA1_O2/TE1_TI1
P_A3	L18			P3_3/TA1_I3/TA1_O3
P_A4	K18			P3_4/TA1_I4/TA1_O4/TE1_AI
P_A5	L21			P3_5/TA1_I5/TA1_O5
P_A6	L22			P3_6/TA1_I6/TA1_O6/CSI3F_CS0/ TE1_BI
P_A7	K19			P3_7/TA1_I7/TA1_O7/CSI3F_CS1
P_A8	K21			P3_8/TA1_I8/TA1_O8/CSI3F_CS2/ TE1_ZI
P_A9	K22			P3_9/TA1_I9/TA1_O9/CSI3F_CS3
P_A10	J19			P3_10/TA1_I10/TA1_O10/CSI3F_CS4
P_A11	J21			P3_11/TA1_I11/TA1_O11/CSI3F_CS5
P_A12	J22			P3_12/TA1_I12/TA1_O12/CSI3F_CS6
P_A13	H19			P3_13/TA1_I13/TA1_O13/CSI3F_CS7
P_A14	H21			P3_14/TA1_I14/TA1_O14/CSI3_RYI/ CSI3_RYO
P_A15	H22			P3_15/TA1_I15/TA1_O15/CSI3_SSI
P_A16	G19			P4_0/INTP5/CSI3F_RYI/CSI3F_RYO
P_A17	G21			P4_1/INTP6/CSI3F_SSI
P_A18	G22			P4_2/INTP7/SO3
P_A19	F19			P4_3/INTP8/$\overline{\text{DMAAK4}}$/SO3F
P_A20	F21			P4_4/INTP9/$\overline{\text{DMATC4}}$/SI3/RXD3
P_A21	F22			P4_5/INTP10/$\overline{\text{DMATC3}}$/SCK3/TXD3/ ADTRG10
P_A22	E21			P4_6/INTP11/$\overline{\text{DMAAK3}}$/SI3F/RXD3F/ SDA3
P_A23	E22			P4_7/INTP12/SCK3F/TXD3F/SCL3/ ADTRG20
P_A24	B14			P6_0/ETH_MDC/$\overline{\text{DMATC0}}$/TJ_I2/TJ_O2
P_A25	A15			P6_1/ETH_MDIO/$\overline{\text{DMATC1}}$/TJ_I3/ TJ_O3
$\overline{\text{P\_CS1}}$	D21	Output	Chip select signal output for external memory of PMEMC	P4_8/$\overline{\text{P\_BCYST}}$/INTP20
$\overline{\text{P\_CS2}}$	D22			P4_9/INTP21/$\overline{\text{DMAAK5}}$
$\overline{\text{P\_CS3}}$	C21			P4_10/INTP22/$\overline{\text{DMATC5}}$
$\overline{\text{P\_CS4}}$	D18			P5_9/ETH_RXER/TA3_I9/TA3_O9/ CSI0F_CS7
$\overline{\text{P\_WAIT}}$	C22	Input	External wait request input of PMEMC	P4_11/SCK0F/TXD0F/SCL0
$\overline{\text{P\_HLDAK}}$	B21	Output	Bus hold acknowledge output of PMEMC	P4_12/INTP23/$\overline{\text{DMAAK2}}$/SO0F
$\overline{\text{P\_HLDRQ}}$	B22	Input	Bus hold request input of PMEMC	P4_13/$\overline{\text{DMATC2}}$/SI0F/RXD0F/SDA0

Table 2-3 External Bus Interface Pins (4/6)

Pin Name	Pin No.	I/O	Description	Alternate-Function Pin
S_D0	C2	I/O	Secondary memory controller (SMEMC) 32-bit data bus for external memory	P7_0/TA2_I0/TA2_O0
S_D1	A2			P7_1/TA2_I1/TA2_O1
S_D2	B2			P7_2/TA2_I2/TA2_O2
S_D3	A3			P7_3/TA2_I3/TA2_O3
S_D4	B3			P7_4/TA2_I4/TA2_O4
S_D5	A4			P7_5/TA2_I5/TA2_O5
S_D6	B4			P7_6/TA2_I6/TA2_O6
S_D7	A5			P7_7/TA2_I7/TA2_O7
S_D8	B5			P7_8/TA2_I8/TA2_O8
S_D9	A6			P7_9/TA2_I9/TA2_O9
S_D10	B6			P7_10/TA2_I10/TA2_O10
S_D11	D5			P7_11/TA2_I11/TA2_O11
S_D12	A7			P7_12/TA2_I12/TA2_O12
S_D13	D6			P7_13/TA2_I13/TA2_O13
S_D14	A8			P7_14/TA2_I14/TA2_O14
S_D15	E5			P7_15/TA2_I15/TA2_O15
S_D16	D7			P8_0/TA0_I0/TA0_O0/TE_TI0
S_D17	D8			P8_1/TA0_I1/TA0_O1
S_D18	A9			P8_2/TA0_I2/TA0_O2/TE0_TI1
S_D19	B9			P8_3/TA0_I3/TA0_O3
S_D20	D9			P8_4/TA0_I4/TA0_O4/TE0_AI
S_D21	E9			P8_5/TA0_I5/TA0_O5
S_D22	A10			P8_6/TA0_I6/TA0_O6/TE0_BI
S_D23	B10			P8_7/TA0_I7/TA0_O7
S_D24	D10			P8_8/TA0_I8/TA0_O8/TE0_ZI
S_D25	E10			P8_9/TA0_I9/TA0_O9
S_D26	B11			P8_10/TA0_I10/TA0_O10
S_D27	D11			P8_11/TA0_I11/TA0_O11
S_D28	E11			P8_12/TA0_I12/TA0_O12
S_D29	D12			P8_13/TA0_I13/TA0_O13
S_D30	E12			P8_14/TA0_I14/TA0_O14
S_D31	E13			P8_15/TA0_I15/TA0_O15
S_SDCKE	D4	Output	SDRAM clock enable output signal of SMEMC	P9_0/TA3_I0/TA3_O0/INTP10/ CSI2F_CS0
S_BUSCLK	C1	Output	Bus clock output of SMEMC	P9_1/CSI2F_CS1
$\overline{\text{S\_SDCAS}}$	E4	Output	Column address strobe signal output for SDRAM of SMEMC	P9_2/TA3_I1/TA3_O1/CSI2F_CS2
$\overline{\text{S\_SDRAS}}$	D2	Output	Row address strobe signal output for SDRAM	P9_3/TA3_I2/TA3_O2/CSI2F_CS3
S_LLDQM	D1	Output	I/O mask signal output for SDRAM (D0 to D7) of SMEMC	P9_4/TA3_I3/TA3_O3/CSI2F_CS4
S_LUDQM	E2	Output	I/O mask signal output for SDRAM (D8 to D15) of SMEMC	P9_5/TA3_I4/TA3_O4/CSI2F_CS5

Table 2-3 External Bus Interface Pins (5/6)

Pin Name	Pin No.	I/O	Description	Alternate-Function Pin
S_ULDQM	F4	Output	I/O mask signal output for SDRAM (D16 to D23) of SMEMC	P9_6/S_DMATC2/ESO0/CSI2F_CS6
S_UUDQM	E1	Output	I/O mask signal output for SDRAM (D0 to D7) of SMEMC	P9_7/S_DMATC3/ESO1/CSI2F_CS7
S_LLWR	F2	Output	Write strobe signal output for external data bus (D0 to D7) of SMEMC	P9_8/S_DMAAK2/CSI4_RYI/CSI4_RYO
S_LUWR	F1	Output	Write strobe signal output for external data bus (D8 to 15) of SMEMC	P9_9/S_DMARQ2/S_DMATC0/SO4
S_ULWR	G4	Output	Write strobe signal output for external data bus (D16 to D23) of SMEMC	P9_10/S_DMAAK3/SI4/RXD4/SDA4/ ESO2
S_UUWR	G2	Output	Write strobe signal output for external data bus (D24 to D31) of SMEMC	P9_11/S_DMARQ3/SCK4/TXD4/SCL4/ ESO3
S_RD	G1	Output	Read strobe signal output for external data bus of SMEMC	P9_12/TA3_I5/TA3_O5/INTP11
S_WR	H2	Output	Write strobe signal output for external data bus of SMEMC	P9_13/TA3_I6/TA3_O6/INTP12/ CSI1_SSI
S_LLBE	H1	Output	Byte enable signal output for external data bus (D0 to D7) of SMEMC	P9_14/TA3_I7/TA3_O7/INTP13/ CSI1_RYI/CSI1_RYO
S_LUBE	H4	Output	Byte enable signal output for external data bus (D8 to D15) of SMEMC	P9_15/INTP14/SO1
S_ULBE	J2	Output	Byte enable signal output for external data bus (D16 to D23) of SMEMC	P10_0/INTP15/SI1/RXD1/S_DMATC0
S_UUBE	J4	Output	Byte enable signal output for external data bus (D24 to D31) of SMEMC	P10_1/INTP16/SCK1/TXD1/S_DMATC1
S_SDWE	J1	Output	Write enable signal output for SDRAM of SMEMC	P10_2/TA3_I8/TA3_O8/INTP17/ CSI2F_SSI
S_BCYST	K2	Output	Strobe signal output indicating bus cycle start of SMEMC	P10_3/TA3_I9/TA3_O9/INTP18/ S_DMATC1/CSI4_SSI
S_CS0	W1	Output	Chip select signal output for external memory of SMEMC	P10_4/TA3_I10/TA3_O10/INTP19/ CSI2_SSI
S_CS1	W2			P10_5/TA3_I11/TA3_O11/INTP24/ CSI2_RYI/CSI2_RYO
S_CS2	Y1			P10_6/TA3_I12/TA3_O12/INTP25/ CSI1F_SSI
S_CS3	R4			P10_7/S_SDCS/INTP26/CSI1F_RYI/ CSI1F_RYO
	W4			P13_1/S_DMAAK0/ADTRG00/INTP0/ CSI5_RYI/CSI5_RYO
S_WAIT	V4	Input	External wait request input of SMEMC	P10_8/SI2/RXD2
S_HLDAK	U5	Output	Bus hold acknowledge output of SMEMC	P10_9/TA3_I13/TA3_O13/SO2/INTP27

Table 2-3 External Bus Interface Pins (6/6)

Pin Name	Pin No.	I/O	Description	Alternate-Function Pin
$\overline{\text{S\_HLDRQ}}$	U4	Input	Bus hold request input of SMEMC	P10_10/TA3_I14/TA3_O14/SCK2/TXD2
$\overline{\text{S\_REFRQ}}$	T4	Output	Refresh request signal output for SDRAM of SMEMC	P10_11/TA3_I15/TA3_O15/CSI2F_RYI/ CSI2F_RYO
$\overline{\text{S\_SDCS}}$	R4	Output	Chip select signal output for external SDRAM of SMEMC	P10_7/$\overline{\text{S\_CS3}}$/INTP26/CSI1F_RYI/ CSI1F_RYO
S_A1	K1	Output	26-bit address bus for external memory of SMEMC	P11_0/TA1_I0/TA1_O0/TE1_TI0
S_A2	K4			P11_1/TA1_I1/TA1_O1
S_A3	L2			P11_2/TA1_I2/TA1_O2/TE1_TI1
S_A4	L1			P11_3/TA1_I3/TA1_O3
S_A5	L5			P11_4/TA1_I4/TA1_O4/TE1_AI
S_A6	L4			P11_5/TA1_I5/TA1_O5
S_A7	M2			P11_6/TA1_I6/TA1_O6/TE1_BI
S_A8	M1			P11_7/TA1_I7/TA1_O7
S_A9	N2			P11_8/TA1_I8/TA1_O8/TE1_ZI
S_A10	N1			P11_9/TA1_I9/TA1_O9
S_A11	P2			P11_10/TA1_I10/TA1_O10/CSI1F_CS0
S_A12	P1			P11_11/TA1_I11/TA1_O11/CSI1F_CS1
S_A13	M4			P11_12/TA1_I12/TA1_O12/CSI1F_CS2
S_A14	M5			P11_13/TA1_I13/TA1_O13/CSI1F_CS3
S_A15	R2			P11_14/TA1_I14/TA1_O14/CSI1F_CS4
S_A16	R1			P11_15/TA1_I15/TA1_O15/CSI1F_CS5
S_A17	T2			P12_0/INTP0/CSI1F_CS6/ADCNV0
S_A18	N4			P12_1/INTP1/CSI1F_CS7/ADCNV1
S_A19	N5			P12_2/INTP2/ADCNV2/CSI0_SSI
S_A20	T1			P12_3/INTP3/CSI0_RYI/CSI0_RYO
S_A21	U2			P12_4/INTP4/SO0
S_A22	U1			P12_5/INTP5/SI0/RXD0
S_A23	V1			P12_6/INTP6/SCK0/TXD0/ADTRG11
S_A24	P4			P12_7/INTP7/SO2F
S_A25	P5			P12_8/INTP8/SI2F/RXD2F/SDA2
S_A26	V2			P12_9/INTP9/SCK2F/TXD2F/SCL2/ ADTRG21

Table 2-4 Peripheral Function Pins (1/19)

Pin Name	Pin No.	I/O	Description	Alternate-Function Pin
ADCNV0	W21	Output	Status output for A/D converter	P1_0/P_D16/TA2_I0/TA2_O0
	T2	Output		P12_0/S_A17/INTP0/CSI1F_CS6
ADCNV1	Y22	Output		P1_1/P_D17/TA2_I1/TA2_O1
	N4	Output		P12_1/S_A18/INTP1/CSI1F_CS7/ ADCNV1
ADCNV2	Y21	Output		P1_2/P_D18/TA2_I2/TA2_O2
	N5	Output		P12_2/S_A19/INTP2/CSI0_SSI
ADTRG00	W4	Input	A/D converter external trigger input	P13_1/$\overline{S\_CS3}$/$\overline{S\_DMAAK0}$/INTP0/ CSI5_RYI/CSI5_RYO
ADTRG01	AB3	Input		P13_0/$\overline{S\_DMARQ0}$/UCLK/CSI5_SSI
ADTRG10	F22	Input		P4_5/P_A21/INTP10/$\overline{DMATC3}$/SCK3/ TXD3/
ADTRG11	V1	Input		P12_6/S_A24/INTP7/SO2F
ADTRG20	E22	Input		P4_7/P_A23/INTP12/SCK3F/TXD3F/ SCL3
ADTRG21	V2	Input		P12_9/S_A26/INTP9/SCK2F/TXD2F/ SCL2
ANI00	V8	Input	Analog input to A/D converter	–
ANI01	W8	Input		–
ANI02	AA8	Input		–
ANI03	AB8	Input		–
ANI04	V7	Input		–
ANI05	W7	Input		–
ANI06	V5	Input		P14_0
ANI07	W5	Input		P14_1
ANI08	AA5	Input		P14_2
ANI09	AB5	Input		P14_3
ANI10	V6	Input		P14_4
ANI11	W6	Input		P14_5
AV_{DD}	AB7	–	Positive power supply for A/D converter (3.3 or 5.0 V)	–
AV_{REFM}	AA6	Input	Reference voltage input to A/D converter	–
AV_{REFP}	AA7	Input		–
AV_{SS}	AB6	–	Ground potential for A/D converter	–
CAN0RXD	AA2	Input	CAN0 receive data input	P13_5/SI1F/RXD1F/SDA1
CAN1RXD	Y2	Input	CAN1 receive data input	P13_7/SI5/RXD5/SDA5
CAN0TXD	AB2	Output	CAN0 transmit data output	P13_4/SCK1F/TXD1F/SCL1/INTP3
CAN1TXD	AA1	Output	CAN1 transmit data output	P13_6/SCK5/TXD5/SCL5/INTP4

Table 2-4 Peripheral Function Pins (2/19)

Pin Name	Pin No.	I/O	Description	Alternate-Function Pin
CSI0F_CS0	A20	Output	CSI0 (FIFO) chip select signal output	P5_2/ETH_TXD3/TA3_I2/TA3_O2/ INTP22/P_ULDQM
CSI0F_CS1	B19			P5_3/ETH_TXD2/TA3_I3/TA3_O3/ INTP23/P_UUDQM
CSI0F_CS2	A19			P5_4/ETH_TXD1/TA3_I4/TA3_O4/ INTP24/P_REFRQ
CSI0F_CS3	B18			P5_5/ETH_TXD0/TA3_I5/TA3_O5/ INTP25/P_SDRAS
CSI0F_CS4	A18			P5_6/ETH_TXEN/TA3_I6/TA3_O6/ INTP26/P_SDCAS
CSI0F_CS5	A17			P5_7/ETH_TXCLK/TA3_I7/TA3_O7/ INTP27
CSI0F_CS6	D17			P5_8/ETH_TXER/TA3_I8/TA3_O8/ P_SDCKE
CSI0F_CS7	D18			P5_9/ETH_RXER/TA3_I9/TA3_O9/ P_CS4
CSI1F_CS0	A20	Output	CSI1 (FIFO) chip select signal output	P11_10/S_A11/TA1_I10/TA1_O10
CSI1F_CS1	B19			P11_11/S_A12/TA1_I11/TA1_O11
CSI1F_CS2	M4			P11_12/S_A13/TA1_I12/TA1_O12
CSI1F_CS3	M5			P11_13/S_A14/TA1_I13/TA1_O13
CSI1F_CS4	A18			P11_14/S_A15/TA1_I14/TA1_O14
CSI1F_CS5	A17			P11_15/S_A16/TA1_I15/TA1_O15
CSI1F_CS6	D17			P12_0/S_A17/INTP0/ADCNV0
CSI1F_CS7	D18			P12_1/S_A18/INTP1/ADCNV1
CSI2F_CS0	D4	Output	CSI2 (FIFO) chip select signal output	P9_0/S_SDCKE/TA3_I0/TA3_O0/INTP10
CSI2F_CS1	C1			P9_1/S_BUSCLK
CSI2F_CS2	E4			P9_2/S_SDCAS/TA3_I1/TA3_O1
CSI2F_CS3	D2			P9_3/S_SDRAS/TA3_I2/TA3_O2
CSI2F_CS4	D1			P9_4/S_LLDQM/TA3_I3/TA3_O3
CSI2F_CS5	E2			P9_5/S_LUDQM/TA3_I4/TA3_O4
CSI2F_CS6	F4			P9_6/S_ULDQM/S_DMATC2/ESO0
CSI2F_CS7	E1			P9_7/S_UUDQM/S_DMATC3/ESO1
CSI3F_CS0	L22	Output	CSI3 (FIFO) chip select signal output	P3_6/P_A6/TA1_I6/TA1_O6/TE1_BI
CSI3F_CS1	K19			P3_7/P_A7/TA1_I7/TA1_O7
CSI3F_CS2	K21			P3_8/P_A8/TA1_I8/TA1_O8/TE1_ZI
CSI3F_CS3	K22			P3_9/P_A9/TA1_I9/TA1_O9
CSI3F_CS4	J19			P3_10/P_A10/TA1_I10/TA1_O10
CSI3F_CS5	J21			P3_11/P_A11/TA1_I11/TA1_O11
CSI3F_CS6	J22			P3_12/P_A12/TA1_I12/TA1_O12
CSI3F_CS7	H19			P3_13/P_A13/TA1_I13/TA1_O13
CSI0F_RYI	A21	Input	CSIH0 ready/busy input	P5_0/ETH_CRS/TA3_I0/TA3_O0/INTP20/ P_LLDQM/CSI0F_RYO
CSI1F_RYI	R4	Input	CSIH1 ready/busy input	P10_7/S_CS3/S_SDCS/INTP26/ CSI1F_RYO
CSI2F_RYI	T4	Input	CSIH2 ready/busy input	P10_11/S_REFRQ/TA3_I15/TA3_O15/ RYOH2
CSI3F_RYI	G19	Input	CSIH3 ready/busy input	P4_0/P_A16/INTP5/CSI3F_RYO

Table 2-4 Peripheral Function Pins (3/19)

Pin Name	Pin No.	I/O	Description	Alternate-Function Pin
CSI0_RYI	T1	Input	CSIG0 ready/busy input	P12_3/S_A20/INTP3/CSI0_RYO
CSI1_RYI	H1	Input	CSIG1 ready/busy input	P9_14/S_LLBE/TA3_I7/TA3_O7/INTP13/ CSI1_RYO
CSI2_RYI	W2	Input	CSIG2 ready/busy input	P10_5/S_CS1/TA3_I11/TA3_O11/ INTP24/CSI2_RYO
CSI3_RYI	H21	Input	CSIG3 ready/busy input	P3_14/P_A14/TA1_I14/TA1_O14/ CSI3_RYO
CSI4_RYI	F2	Input	CSIG4 ready/busy input	P9_8/S_LLWR/S_DMAAK2/CSI4_RYO
CSI5_RYI	W4	Input	CSIG5 ready/busy input	P13_1/S_CS3/S_DMAAK0/ADTRG00/ INTP0/CSI5_RYO
CSI0F_RYO	A21	Output	CSIH0 ready/busy output	P5_0/ETH_CRS/TA3_I0/TA3_O0/INTP20/ P_LLDQM/CSI0F_RYI
CSI1F_RYO	R4	Output	CSIH1 ready/busy output	P10_7/S_CS3/S_SDCS/INTP26/ CSI1F_RYI
CSI2F_RYO	T4	Output	CSIH2 ready/busy output	P10_11/S_REFRQ/TA3_I15/TA3_O15/ RYIH2
CSI3F_RYO	G19	Output	CSIH3 ready/busy output	P4_0/P_A16/INTP5/CSI3F_RYI
CSI0_RYO	T1	Output	CSIG0 ready/busy output	P12_3/S_A20/INTP3/CSI0_RYI
CSI1_RYO	H1	Output	CSIG1 ready/busy output	P9_14/S_LLBE/TA3_I7/TA3_O7/INTP13/ CSI1_RYI
CSI2_RYO	W2	Output	CSIG2 ready/busy output	P10_5/S_CS1/TA3_I11/TA3_O11/ INTP24/CSI2_RYI
CSI3_RYO	H21	Output	CSIG3 ready/busy output	P3_14/P_A14/TA1_I14/TA1_O14/ CSI3_RYI
CSI4_RYO	F2	Output	CSIG4 ready/busy output	P9_8/S_LLWR/S_DMAAK2/CSI4_RYI
CSI5_RYO	W4	Output	CSIG5 ready/busy output	P13_1/S_CS3/S_DMAAK0/ADTRG00/ INTP0/CSI5_RYI
CSI0F_SSI	A21	Input	CSIH0 slave select input	P5_1/ETH_COL/TA3_I1/TA3_O1/INTP21/ P_LUDQM/CSI0F_RYI/CSI0F_RYO
CSI1F_SSI	Y1		CSIH1 slave select input	P10_6/S_CS2/TA3_I12/TA3_O12/INTP25
CSI2F_SSI	J1		CSIH2 slave select input	P10_2/S_SDWE/TA3_I8/TA3_O8/INTP17
CSI3F_SSI	G22		CSIH3 slave select input	P4_1/P_A17/INTP6
CSI0_SSI	N5		CSIG0 slave select input	P12_2/S_A19/INTP2/ADCNV2
CSI1_SSI	H2		CSIG1 slave select input	P9_13/S_WR/TA3_I6/TA3_O6/INTP12/
CSI2_SSI	W1		CSIG2 slave select input	P10_4/S_CS0/TA3_I10/TA3_O10/INTP19
CSI3_SSI	H22		CSIG3 slave select input	P3_15/P_A15/TA1_I15/TA1_O15
CSI4_SSI	K2		CSIG4 slave select input	P10_3/S_BCYST/TA3_I9/TA3_O9/ INTP18/S_DMATC1/
CSI5_SSI	AB3		CSIG5 slave select input	P13_0/S_DMARQ0/ADTRG01/UCLK
DMAAK0	E14	Output	DMA0 acknowledge signal output	P5_14/ETH_RXD2/TA3_I14/TA3_O14/ TJ_I0/TJ_O0
DMAAK1	D14	Output	DMA1 acknowledge signal output	P5_15/ETH_RXD3/TA3_I15/TA3_O15/ TJ_I1/TJ_O1
DMAAK2	B21	Output	DMA2 acknowledge signal output	P4_12/P_HLDAK/INTP23/SO0F
DMAAK3	E21	Output	DMA3 acknowledge signal output	P4_6/P_A22/INTP11/SI3F/RXD3F/SDA3
DMAAK4	F19	Output	DMA4 acknowledge signal output	P4_3/P_A19/INTP8/SO3F
DMAAK5	D22	Output	DMA5 acknowledge signal output	P4_9/P_CS2/INTP21
DMATC0	B14	Output	DMA0 transfer completion signal output	P6_0/ETH_MDC/TJ_I2/TJ_O2/P_A24

Table 2-4 Peripheral Function Pins (4/19)

Pin Name	Pin No.	I/O	Description	Alternate-Function Pin
$\overline{\text{DMATC1}}$	A15	Output	DMA1 transfer completion signal output	P6_1/ETH_MDIO/TJ_I3/TJ_O3/P_A25
DMATC2	B22	Output	DMA2 transfer completion signal output	P4_13/$\overline{\text{P\_HLDRQ}}$/SI0F/RXD0F/SDA0
$\overline{\text{DMATC3}}$	F22	Output	DMA3 transfer completion signal output	P4_5/P_A21/INTP10/SCK3/TXD3/ ADTRG10
$\overline{\text{DMATC4}}$	F21	Output	DMA4 transfer completion signal output	P4_4/P_A20/INTP9/SI3/RXD3
$\overline{\text{DMATC5}}$	C21	Output	DMA5 transfer completion signal output	P4_10/$\overline{\text{P\_CS3}}$/INTP22
ESO0	AA22	Input	Hi-Z control request for motor control function	P1_3/P_D19/TA2_I3/TA2_O3
	F4			P9_6/S_ULDQM/$\overline{\text{S\_DMATC2}}$/ CSI2F_CS6
ESO1	AA21			P1_4/P_D20/TA2_I4/TA2_O4
	E1			P9_7/S_UUDQM/$\overline{\text{S\_DMATC3}}$/ CSI0F_CS7
ESO2	AB21			P1_5/P_D21/TA2_I5/TA2_O5
	G4			P9_10/$\overline{\text{S\_ULWR}}$/$\overline{\text{S\_DMAAK3}}$/SI4/RXD4/ SDA4
ESO3	W19			P1_6/P_D22/TA2_I6/TA2_O6
	G2			P9_11/$\overline{\text{S\_UUWR}}$/$\overline{\text{S\_DMARQ3}}$/SCK4/ TXD4/SCL4

Table 2-4 Peripheral Function Pins (5/19)

Pin Name	Pin No.	I/O	Description	Alternate-Function Pin
ETH_COL	B20	Input	Ethernet connection pins	P5_1/TA3_I1/TA3_O1/INTP21/ CSI0F_SSI
ETH_MDC	B14	Output		P6_0/$\overline{\text{DMATC0}}$/TJ_I2/TJ_O2/P_A24
ETH_MDIO	A15	I/O		P6_1/$\overline{\text{DMATC1}}$/TJ_I3/TJ_O3/P_A25
ETH_RXCLK	A16	Input		P5_10/TA3_I10/TA3_O10/P_BUSRQ
ETH_RXD0	B15	Input		P5_12/TA3_I12/TA3_O12/$\overline{\text{P\_BCYST}}$
ETH_RXD1	D15	Input		P5_13/TA3_I13/TA3_O13
ETH_RXD2	E14	Input		P5_14/TA3_I14/TA3_O14/$\overline{\text{DMAAK0}}$/ TJ_I0/TJ_O0
ETH_RXD3	D14	Input		P5_15/TA3_I15/TA3_O15/$\overline{\text{DMAAK1}}$/ TJ_I1/TJ_O1
ETH_RXDV	D16	Input		P5_11/TA3_I11/TA3_O11/$\overline{\text{P\_SDWE}}$
ETH_TXCLK	A17	Input		P5_7/TA3_I7/TA3_O7/INTP27/ CSI0F_CS5
ETH_TXD0	B18	Output		P5_5/TA3_I5/TA3_O5/INTP25/ CSI0F_CS3/$\overline{\text{P\_SDRAS}}$
ETH_TXD1	A19	Output		P5_4/TA3_I4/TA3_O4/INTP24/ CSI0F_CS2/$\overline{\text{P\_REFRQ}}$
ETH_TXD2	B19	Output		P5_3/TA3_I3/TA3_O3/INTP23/ CSI0F_CS1/P_UUDQM
ETH_TXD3	A20	Output		P5_2/TA3_I2/TA3_O2/INTP22/ CSI0F_CS0/P_ULDQM
ETH_TXEN	A18	Output		P5_6/TA3_I6/TA3_O6/INTP26/ CSI0F_CS4/$\overline{\text{P\_SDCAS}}$
ETH_CRS	A21	Input		P5_0/TA3_I0/TA3_O0/INTP20/P_LLDQM/ CSI0F_RYI/CSI0F_RYO
ETH_RXER	D18	Input		P5_9/TA3_I9/TA3_O9/CSI0F_CS7/ $\overline{\text{P\_CS4}}$
ETH_TXER	D17	Output		P5_8/TA3_I8/TA3_O8/CSI0F_CS6/ P_SDCKE

Table 2-4 Peripheral Function Pins (6/19)

Pin Name	Pin No.	I/O	Description	Alternate-Function Pin
INTP0	T2	Input	External maskable interrupt request input	P12_0/S_A17/CSI1F_CS6/ADCNV0
	W4			P13_1/$\overline{\text{S\_CS3}}$/S_DMAAK0/ADTRG00/ CSI5_RYI/CSI5_RYO
INTP1	N4			P12_1/S_A18/CSI1F_CS7/ADCNV1
	AB4			P13_2/$\overline{\text{S\_DMARQ1}}$/SO1F/PPON
INTP2	N5			P12_2/S_A19/ADCNV2/CSI0_SSI
	AA3			P13_3/$\overline{\text{S\_DMAAK1}}$/SO5/$\overline{\text{OCI}}$
INTP3	T1			P12_3/S_A20/CSI0_RYI/CSI0_RYO
	AB2			P13_4/CAN0TXD/SCK1F/TXD1F/SCL1
INTP4	U2			P12_4/S_A21/SO0
	AA1			P13_6/CAN1TXD/SCK5/TXD5/SCL5
INTP5	G19			P4_0/P_A16/CSI3F_RYI/CSI3F_RYO
	U1			P12_5/S_A22/SI0/RXD0
INTP6	G21			P4_1/P_A17/CSI3F_SSI
	V1			P12_6/S_A23/SCK0/TXD0/ADTRG11
INTP7	G22			P4_2/P_A18/SO3
	P4			P12_7/S_A24/SO2F
INTP8	F19			P4_3/P_A19/$\overline{\text{DMAAK4}}$/SO3F
	P5			P12_8/S_A25/SI2F/RXD2F/SDA2
INTP9	F21			P4_4/P_A20/$\overline{\text{DMATC4}}$/SI3/RXD3
	V2			P12_9/S_A26/SCK2F/TXD2F/SCL2/ ADTRG21
INTP10	F22			P4_5/P_A21/$\overline{\text{DMATC3}}$/SCK3/TXD3/ ADTRG10
	D4			P9_0/S_SDCKE/TA3_I0/TA3_O0/ CSI2F_CS0
INTP11	E21			P4_6/P_A22/$\overline{\text{DMAAK3}}$/SI3F/RXD3F/ SDA3
	G1			P9_12/$\overline{\text{S\_RD}}$/TA3_I5/TA3_O5
INTP12	E22			P4_7/P_A23/SCK3F/TXD3F/SCL3/ ADTRG20
	H2			P9_13/$\overline{\text{S\_WR}}$/TA3_I6/TA3_O6/CSI1_SSI
INTP13	P21			P2_1/$\overline{\text{P\_LLBE}}$/$\overline{\text{P\_LLWR}}$/TJ_I0/TJ_O0
	H1			P9_14/$\overline{\text{S\_LLBE}}$/TA3_I7/TA3_O7/ CSI1_RYI/CSI1_RYO
INTP14	P22			P2_2/$\overline{\text{P\_LUBE}}$/$\overline{\text{P\_LUWR}}$/TJ_I1/TJ_O1
	H4			P9_15/$\overline{\text{S\_LUBE}}$/SO1
INTP15	N18			P2_3/$\overline{\text{P\_ULBE}}$/$\overline{\text{P\_ULWR}}$/TJ_I2/TJ_O2
	J2			P10_0/$\overline{\text{S\_ULBE}}$/SI1/RXD1/$\overline{\text{S\_DMATC0}}$
INTP16	N19			P2_4/$\overline{\text{P\_UUBE}}$/$\overline{\text{P\_UUWR}}$/TJ_I3/TJ_O3
	J4			P10_1/$\overline{\text{S\_UUBE}}$/SCK1/TXD1/$\overline{\text{S\_DMATC1}}$
INTP17	N21			P2_5/$\overline{\text{P\_RD}}$
	J1			P10_2/$\overline{\text{S\_SDWE}}$/TA3_I8/TA3_O8/ CSI2F_SSI

Table 2-4 Peripheral Function Pins (7/19)

Pin Name	Pin No.	I/O	Description	Alternate-Function Pin
INTP18	M19	Input	External maskable interrupt request input	P3_0/P_A0/TA1_I0/TA1_O0/TE1_TI0
	K2			P10_3/S_BCYST/TA3_I9/TA3_O9/ S_DMATC1/CSI4_SSI
INTP19	M18			P2_7/P_WR/P_RW
	W1			P10_4/S_CS0/TA3_I10/TA3_O10/ CSI2_SSI
INTP20	D21			P4_8/P_CS1/P_BCYST
	A21			P5_0/ETH_CRS/TA3_I0/TA3_O0/ P_LLDQM/CSI0F_RYI/CSI0F_RYO
INTP21	D22			P4_9/P_CS2/DMAAK5
	B20			P5_1/ETH_COL/TA3_I1/TA3_O1/ CSI0F_SSI/P_LUDQM
INTP22	C21			P4_10/P_CS3/DMATC5
	A20			P5_2/ETH_TXD3/TA3_I2/TA3_O2/ CSI0F_CS0/P_ULDQM
INTP23	B21			P4_12/P_HLDAK/DMAAK2/SO0F
	B19			P5_3/ETH_TXD2/TA3_I3/TA3_O3/ CSI0F_CS1/P_UUDQM
INTP24	B19			P5_4/ETH_TXD1/TA3_I4/TA3_O4/ CSI0F_CS2/P_REFRQ
	W2			P10_5/S_CS1/TA3_I11/TA3_O11/ CSI2_RYI/CSI2_RYO
INTP25	B18			P5_5/ETH_TXD0/TA3_I5/TA3_O5/ CSI0F_CS3/P_SDRAS
	Y1			P10_6/S_CS2/TA3_I12/TA3_O12/ CSI1F_SSI
INTP26	A18			P5_6/ETH_TXEN/TA3_I6/TA3_O6/ CSI0F_CS4/P_SDCAS
	R4			P10_7/S_CS3/S_SDCS/CSI1F_RYI/ CSI1F_RYO
INTP27	A17			P5_7/ETH_TXCLK/TA3_I7/TA3_O7/ CSI0F_CS5
	U5			P10_9/S_HLDAK/TA3_I13/TA3_O13/SO2
NMI	W16	Input	Non-maskable interrupt request input	P2_0
OCI	AA3	Input	Overcurrent detection input	P13_3/S_DMAAK1/SO5/INTP2
PPON	AB4	Output	USB power supply output	P13_2/S_DMARQ1/SOF1/INTP1

Table 2-4 Peripheral Function Pins (8/19)

Pin Name	Pin No.	I/O	Description	Alternate-Function Pin
RXD0	U1	Input	UART0 serial receive data input	P12_5/S_A22/INTP5/SI0
RXD1	J2	Input	UART1 serial receive data input	P10_0/S_ULBE/INTP15/SI1/S_DMATC0
RXD2	V4	Input	UART2 serial receive data input	P10_8/S_WAIT/SI2
RXD3	F21	Input	UART3 serial receive data input	P4_4/P_A20/INTP9/DMATC4/SI3
RXD4	G4	Input	UART4 serial receive data input	P9_10/S_ULWR/S_DMAAK3/SI4/SDA4/ESO2
RXD5	Y2	Input	UART5 serial receive data input	P13_7/CAN1RXD/SI5/SDA5
RXD0F	B22	Input	UART0 (FIFO) serial receive data input	P4_13/P_HLDRQ/DMATC2/SI0F/SDA0
RXD1F	AA2	Input	UART1 (FIFO) serial receive data input	P13_5/CAN0RXD/SI1F/SDA1
RXD2F	P5	Input	UART2 (FIFO) serial receive data input	P12_8/S_A25/INTP8/SI2F/SDA2
RXD3F	E21	Input	UART3 (FIFO) serial receive data input	P4_6/P_A22/INTP11/DMAAK3/SI3F/SDA3
S_DMAAK0	W4	Output	DMA0 acknowledge signal output of SMEMC	P13_1/S_CS3/ADTRG00/INTP0/ CSI5_RYI/CSI5_RYO
S_DMAAK1	AA3	Output	DMA1 acknowledge signal output of SMEMC	P13_3/SO5/INTP2/OCI
S_DMAAK2	F2	Output	DMA2 acknowledge signal output of SMEMC	P9_8/S_LLWR/CSI4_RYI/CSI4_RYO
S_DMAAK3	G4	Output	DMA3 acknowledge signal output of SMEMC	P9_10/S_ULWR/SI4/RXD4/SDA4/ESO2
S_DMARQ0	AB3	Input	DMA0 request signal input of SMEMC	P13_0/ADTRG01/UCLK/CSI5_SSI
S_DMARQ1	AB4	Input	DMA1 request signal input of SMEMC	P13_2/SO1F/INTP1/PPON
S_DMARQ2	F1	Input	DMA2 request signal input of SMEMC	P9_9/S_LUWR/S_DMATC0/SO4
S_DMARQ3	G2	Input	DMA3 request signal input of SMEMC	P9_11/S_UUWR/SCK4/TXD4/SCL4/ ESO3
S_DMATC0	F1	Output	DMA0 transfer completion signal output of SMEMC	P9_9/S_LUWR/S_DMARQ2/SO4
	J2			P10_0/S_ULBE/INTP15/SI1/RXD1
S_DMATC1	J4	Output	DMA1 transfer completion signal output of SMEMC	P10_1/S_UUBE/INTP16/SCK1/TXD1
	K2			P10_3/S_BCYST/TA3_I9/TA3_O9/ INTP18/CSI4_SSI
S_DMATC2	F4	Output	DMA2 transfer completion signal output of SMEMC	P9_6/S_ULDQM/ESO0/CSI2F_CS6
S_DMATC3	E1	Output	DMA3 transfer completion signal output of SMEMC	P9_7/S_UUDQM/ESO1/CSI2F_CS7
SCK0	V1	I/O	CSI0 serial clock I/O	P12_6/S_A23/INTP6/TXD0/ADTRG11
SCK1	J4	I/O	CSI1 serial clock I/O	P10_1/S_UUBE/INTP16/TXD1/S_DMATC1
SCK2	U4	I/O	CSI2 serial clock I/O	P10_10/S_HLDRQ/TA3_I14/TA3_O14/TXD2
SCK3	F22	I/O	CSI3 serial clock I/O	P4_5/P_A21/INTP10/DMATC3/TXD3/ADTRG10
SCK4	G2	I/O	CSI4 serial clock I/O	P9_11/S_UUWR/S_DMARQ3/TXD4/SCL4/ESO3
SCK5	AA1	I/O	CSI5 serial clock I/O	P13_6/CAN1TXD/TXD5/SCL5/INTP4
SCK0F	C22	I/O	CSI0 (FIFO) serial clock I/O	P4_11/P_WAIT/TXD0F/SCL0
SCK1F	AB2	I/O	CSI1 (FIFO) serial clock I/O	P13_4/CAN0TXD/TXD1F/SCL1/INTP3
SCK2F	V2	I/O	CSI2 (FIFO) serial clock I/O	P12_9/S_A26/INTP9/TXD2F/SCL2/ADTRG21
SCK3F	E22	I/O	CSI3 (FIFO) serial clock I/O	P4_7/P_A23/INTP12/TXD3F/SCL3/ADTRG20

Table 2-4 Peripheral Function Pins (9/19)

Pin Name	Pin No.	I/O	Description	Alternate-Function Pin
SCL0	C22	I/O	I^2C0 serial clock I/O	P4_11/$\overline{P\_WAIT}$/SCK0F/TXD0F
SCL1	AB2	I/O	I^2C1 serial clock I/O	P13_4/CAN0TXD/SCK1F/TXD1F/INTP3
SCL2	V2	I/O	I^2C2 serial clock I/O	P12_9/S_A26/INTP9/SCK2F/TXD2F/ ADTRG21
SCL3	E22	I/O	I^2C3 serial clock I/O	P4_7/P_A23/INTP12/SCK3F/TXD3F/ ADTRG20
SCL4	G2	I/O	I^2C4 serial clock I/O	P9_11/$\overline{S\_UUWR}$/$\overline{S\_DMARQ3}$/SCK4/ TXD4/ESO3
SCL5	AA1	I/O	I^2C5 serial clock I/O	P13_6/CAN1TXD/SCK5/TXD5/INTP4
SDA0	B22	I/O	I^2C0 data I/O	P4_13/$\overline{P\_HLDRQ}$/$\overline{DMATC2}$/SI0F/RXD0F
SDA1	AA2	I/O	I^2C1 data I/O	P13_5/CAN0RXD/SI1F/RXD1F
SDA2	P5	I/O	I^2C2 data I/O	P12_8/S_A25/INTP8/SI2F/RXD2F
SDA3	E21	I/O	I^2C3 data I/O	P4_6/P_A22/INTP11/DMAAK3/SI3F/RXD3F
SDA4	G4	I/O	I^2C4 data I/O	P9_10/$\overline{S\_ULWR}$/$\overline{S\_DMAAK3}$/SI4/RXD4/ESO2
SDA5	Y2	I/O	I^2C5 data I/O	P13_7/CAN1RXD/SI5/RXD5
SI0	U1	Input	SCI0 serial receive data input	P12_5/S_A22/INTP5/RXD0
SI1	J2	Input	SCI1 serial receive data input	P10_0/$\overline{S\_ULBE}$/INTP15/RXD1/$\overline{S\_DMATC0}$
SI2	V4	Input	SCI2 serial receive data input	P10_8/$\overline{S\_WAIT}$/RXD2
SI3	F21	Input	SCI3 serial receive data input	P4_4/P_A20/INTP9/$\overline{DMATC4}$/RXD3
SI4	G4	Input	SCI4 serial receive data input	P9_10/$\overline{S\_ULWR}$/$\overline{S\_DMAAK3}$/RXD4/SDA4
SI5	Y2	Input	SCI5 serial receive data input	P13_7/CAN1RXD/RXD5/SDA5
SI0F	B22	Input	SCI0 (FIFO) serial receive data input	P4_13/$\overline{P\_HLDRQ}$/$\overline{DMATC2}$/RXD0F/SDA0
SI1F	AA2	Input	SCI1 (FIFO) serial receive data input	P13_5/CAN0RXD/RXD1F/SDA1
SI2F	P5	Input	SCI2 (FIFO) serial receive data input	P12_8/S_A25/INTP8/RXD2F/SDA2
SI3F	E21	Input	SCI3 (FIFO) serial receive data input	P4_6/P_A22/INTP11/$\overline{DMAAK3}$/RXD3F/SDA3
SO0	U2	Output	SCI0 serial transmit data output	P12_4/S_A21/INTP4
SO1	H4	Output	SCI1 serial transmit data output	P9_15/$\overline{S\_LUBE}$/INTP14
SO2	U5	Output	SCI2 serial transmit data output	P10_9/$\overline{S\_HLDAK}$/TA3_I13/TA3_O13/INTP27
SO3	G22	Output	SCI3 serial transmit data output	P4_2/P_A18/INTP7
SO4	F1	Output	SCI4 serial transmit data output	P9_9/$\overline{S\_LUWR}$/$\overline{S\_DMARQ2}$/$\overline{S\_DMATC0}$
SO5	AA3	Output	SCI5 serial transmit data output	P13_3/$\overline{S\_DMAAK1}$/INTP2/$\overline{OCI}$
SO0F	B21	Output	SCI0 (FIFO) serial transmit data output	P4_12/$\overline{P\_HLDAK}$/INTP23/$\overline{DMAAK2}$
SO1F	AB4	Output	SCI1 (FIFO) serial transmit data output	P13_2/$\overline{S\_DMARQ1}$/INTP1/PPON
SO2F	P4	Output	SCI2 (FIFO) serial transmit data output	P12_7/S_A24/INTP7
SO3F	F19	Output	SCI3 (FIFO) serial transmit data output	P4_3/P_A19/INTP8/$\overline{DMAAK4}$

Table 2-4 Peripheral Function Pins (10/19)

Pin Name	Pin No.	I/O	Description	Alternate-Function Pin
TA0_I0	P19	Input	16-bit timer array 0 external input	P0_0/P_D0/TA0_O0/TE0_TI0
	D7			P8_0/S_D16/TA0_O0/TE_TI0
TA0_I1	P18			P0_1/P_D1/TA0_O1
	D8			P8_1/S_D17/TA0_O1
TA0_I2	R22			P0_2/P_D2/TA0_O2/TE0_TI1
	A9			P8_2/S_D18/TA0_O2/TE0_TI1
TA0_I3	R21			P0_3/P_D3/TA0_O3
	B9			P8_3/S_D19/TA0_O3
TA0_I4	R19			P0_4/P_D4/TA0_O4/TE0_AI
	D9			P8_4/S_D20/TA0_O4/TE0_AI
TA0_I5	R18			P0_5/P_D5/TA0_O5
	E9			P8_5/S_D21/TA0_O5
TA0_I6	T22			P0_6/P_D6/TA0_O6/TE0_BI
	A10			P8_6/S_D22/TA0_O6/TE0_BI
TA0_I7	T21			P0_7/P_D7/TA0_O7
	B10			P8_7/S_D23/TA0_O7
TA0_I8	T19			P0_8/P_D8/TA0_O8/TE0_ZI
	D10			P8_8/S_D24/TA0_O8/TE0_ZI
TA0_I9	T18			P0_9/P_D9/TA0_O9
	E10			P8_9/S_D25/TA0_O9
TA0_I10	U22			P0_10/P_D10/TA0_O10
	B11			P8_10/S_D26/TA0_O10
TA0_I11	U21			P0_11/P_D11/TA0_O11
	D11			P8_11/S_D27/TA0_O11
TA0_I12	U19			P0_12/P_D12/TA0_O12
	E11			P8_12/S_D28/TA0_O12
TA0_I13	V22			P0_13/P_D13/TA0_O13
	D12			P8_13/S_D29/TA0_O13
TA0_I14	V21			P0_14/P_D14/TA0_O14
	E12			P8_14/S_D30/TA0_O14
TA0_I15	W22			P0_15/P_D15/TA0_O15
	E13			P8_15/S_D31/TA0_O15

Table 2-4 Peripheral Function Pins (11/19)

Pin Name	Pin No.	I/O	Description	Alternate-Function Pin
TA1_I0	M19	Input	16-bit timer array 1 external input	P3_0/P_A0/TA1_O0/TE1_TI0/INTP18
	K1			P11_0/S_A1/TA1_O0/TE1_TI0
TA1_I1	M21			P3_1/P_A1/TA1_O1
	K4			P11_1/S_A2/TA1_O1
TA1_I2	L19			P3_2/P_A2/TA1_O2/TE1_TI1
	L2			P11_2/S_A3/TA1_O2/TE1_TI1
TA1_I3	L18			P3_3/P_A3/TA1_O3
	L1			P11_3/S_A4/TA1_O3
TA1_I4	K18			P3_4/P_A4/TA1_O4/TE1_AI
	L5			P11_4/S_A5/TA1_O4/TE1_AI
TA1_I5	L21			P3_5/P_A5/TA1_O5
	L4			P11_5/S_A6/TA1_O5
TA1_I6	L22			P3_6/P_A6/TA1_O6/CSI3F_CS0/TE1_BI
	M2			P11_6/S_A7/TA1_O6/TE1_BI
TA1_I7	K19			P3_7/P_A7/TA1_O7/CSI3F_CS1
	M1			P11_7/S_A8/TA1_O7
TA1_I8	K21			P3_8/P_A8/TA1_O8/CSI3F_CS2/TE1_ZI
	N2			P11_8/S_A9/TA1_O8/TE1_ZI
TA1_I9	K22			P3_9/P_A9/TA1_O9/CSI3F_CS3
	N1			P11_9/S_A10/TA1_O9
TA1_I10	J19			P3_10/P_A10/TA1_O10/CSI3F_CS4
	P2			P11_10/S_A11/TA1_O10/CSI1F_CS0
TA1_I11	J21			P3_11/P_A11/TA1_O11/CSI3F_CS5
	P1			P11_11/S_A12/TA1_O11/CSI1F_CS1
TA1_I12	J22			P3_12/P_A12/TA1_O12/CSI3F_CS6
	M4			P11_12/S_A13/TA1_O12/CSI1F_CS2
TA1_I13	H19			P3_13/P_A13/TA1_O13/CSI3F_CS7
	M5			P11_13/S_A14/TA1_O13/CSI1F_CS3
TA1_I14	H21			P3_14/P_A14/TA1_O14/CSI3_RYI/ CSI3_RYO
	R2			P11_14/S_A15/TA1_O14/CSI1F_CS4
TA1_I15	H22			P3_15/P_A15/TA1_O15/CSI3_SSI
	R1			P11_15/S_A16/TA1_O15/CSI1F_CS5

Table 2-4 Peripheral Function Pins (12/19)

Pin Name	Pin No.	I/O	Description	Alternate-Function Pin
TA2_I0	W21	Input	16-bit timer array 2 external input	P1_0/P_D16/TA2_O0/ADCNV0
	C2			P7_0/S_D0/TA2_O0
TA2_I1	Y22			P1_1/P_D17/TA2_O1/ADCNV1
	A2			P7_1/S_D1/TA2_O1
TA2_I2	Y21			P1_2/P_D18/TA2_O2/ADCNV2
	B2			P7_2/S_D2/TA2_O2
TA2_I3	AA22			P1_3/P_D19/TA2_O3/ESO0
	A3			P7_3/S_D3/TA2_O3
TA2_I4	AA21			P1_4/P_D20/TA2_O4/ESO1
	B3			P7_4/S_D4/TA2_O4
TA2_I5	AB21			P1_5/P_D21/TA2_O5/ESO2
	A4			P7_5/S_D5/TA2_O5
TA2_I6	W19			P1_6/P_D22/TA2_O6/ESO3
	B4			P7_6/S_D6/TA2_O6
TA2_I7	AA20			P1_7/P_D23/TA2_O7
	A5			P7_7/S_D7/TA2_O7
TA2_I8	AB20			P1_8/P_D24/TA2_O8
	B5			P7_8/S_D8/TA2_O8
TA2_I9	AA19			P1_9/P_D25/TA2_O9
	A6			P7_9/S_D9/TA2_O9
TA2_I10	AB19			P1_10/P_D26/TA2_O10
	B6			P7_10/S_D10/TA2_O10
TA2_I11	AA18			P1_11/P_D27/TA2_O11
	D5			P7_11/S_D11/TA2_O11
TA2_I12	AB18			P1_12/P_D28/TA2_O12
	A7			P7_12/S_D12/TA2_O12
TA2_I13	W17			P1_13/P_D29/TA2_O13
	D6			P7_13/S_D13/TA2_O13
TA2_I14	AA17			P1_14/P_D30/TA2_O14
	A8			P7_14/S_D14/TA2_O14
TA2_I15	AB17			P1_15/P_D31/TA2_O15
	E5			P7_15/S_D15/TA2_O15

Table 2-4 Peripheral Function Pins (13/19)

Pin Name	Pin No.	I/O	Description	Alternate-Function Pin
TA3_I0	A21	Input	16-bit timer array 3 external input	P5_0/ETH_CRS/TA3_O0/INTP20/ P_LLDQM/CSI0F_RYI/CSI0F_RYO
	D4			P9_0/S_SDCKE/TA3_O0/INTP10/ CSI2F_CS0
TA3_I1	B20			P5_1/ETH_COL/TA3_O1/INTP21/ CSI0F_SSI/P_LUDQM
	E4			P9_2/$\overline{\text{S\_SDCAS}}$/TA3_O1/CSI2F_CS2
TA3_I2	A20			P5_2/ETH_TXD3/TA3_O2/INTP22/ CSI0F_CS0/P_ULDQM
	D2			P9_3/$\overline{\text{S\_SDRAS}}$/TA3_O2/CSI2F_CS3
TA3_I3	B19			P5_3/ETH_TXD2/TA3_O3/INTP23/ CSI0F_CS1/P_UUDQM
	D1			P9_4/S_LLDQM/TA3_O3/CSI2F_CS4
TA3_I4	A19			P5_4/ETH_TXD1/TA3_O4/INTP24/ CSI0F_CS2/$\overline{\text{P\_REFRQ}}$
	E2			P9_5/S_LUDQM/TA3_O4/CSI2F_CS5
TA3_I5	B18			P5_5/ETH_TXD0/TA3_O5/INTP25/ CSI0F_CS3/$\overline{\text{P\_SDRAS}}$
	G1			P9_12/$\overline{\text{S\_RD}}$/TA3_O5/INTP11
TA3_I6	A18			P5_6/ETH_TXEN/TA3_O6/INTP26/ CSI0F_CS4/$\overline{\text{P\_SDCAS}}$
	H2			P9_13/$\overline{\text{S\_WR}}$/TA3_O6/INTP12/CSI1_SSI
TA3_I7	A17			P5_7/ETH_TXCLK/TA3_O7/INTP27/ CSI0F_CS5
	H1			P9_14/$\overline{\text{S\_LLBE}}$/TA3_O7/INTP13/ CSI1_RYI/CSI1_RYO
TA3_I8	D17			P5_8/ETH_TXER/TA3_O8/CSI0F_CS6/ P_SDCKE
	J1			P10_2/$\overline{\text{S\_SDWE}}$/TA3_O8/INTP17/ CSI2F_SSI
TA3_I9	D18			P5_9/ETH_RXER/TA3_O9/CSI0F_CS7/$\overline{\text{P\_CS4}}$
	K2			P10_3/$\overline{\text{S\_BCYST}}$/TA3_O9/INTP18/ $\overline{\text{S\_DMATC1}}$/CSI4_SSI
TA3_I10	A16			P5_10/ETH_RXCLK/TA3_O10/P_BUSRQ
	W1			P10_4/$\overline{\text{S\_CS0}}$/TA3_O10/INTP19/CSI2_SSI
TA3_I11	D16			P5_11/ETH_RXDV/TA3_O11/$\overline{\text{P\_SDWE}}$
	W2			P10_5/$\overline{\text{S\_CS1}}$/TA3_O11/INTP24/ CSI2_RYI/CSI2_RYO
TA3_I12	B15			P5_12/ETH_RXD0/TA3_O12/$\overline{\text{P\_BCYST}}$
	Y1			P10_6/$\overline{\text{S\_CS2}}$/TA3_O12/INTP25/CSI1F_SSI
TA3_I13	D15			P5_13/ETH_RXD1/TA3_O13
	U5			P10_9/$\overline{\text{S\_HLDAK}}$/TA3_O13/SO2/INTP27
TA3_I14	E14			P5_14/ETH_RXD2/TA3_O14/$\overline{\text{DMAAK0}}$/ TJ_I0/TJ_O0
	U4			P10_10/$\overline{\text{S\_HLDRQ}}$/TA3_O14/SCK2/TXD2
TA3_I15	D14			P5_15/ETH_RXD3/TA3_O15/$\overline{\text{DMAAK1}}$/ TJ_I1/TJ_O1
	T4			P10_11/$\overline{\text{S\_REFRQ}}$/TA3_O15/CSI2F_RYI/ CSI2F_RYO

Table 2-4 Peripheral Function Pins (14/19)

Pin Name	Pin No.	I/O	Description	Alternate-Function Pin
TA0_O0	P19	Output	16-bit timer array 0 external output	P0_0/P_D0/TA0_I0/TE0_TI0
	D7			P8_0/S_D16/TA0_I0/TE_TI0
TA0_O1	P18			P0_1/P_D1/TA0_I1
	D8			P8_1/S_D17/TA0_I1
TA0_O2	R22			P0_2/P_D2/TA0_I2/TE0_TI1
	A9			P8_2/S_D18/TA0_I2/TE0_TI1
TA0_O3	R21			P0_3/P_D3/TA0_I3
	B9			P8_3/S_D19/TA0_I3
TA0_O4	R19			P0_4/P_D4/TA0_I4/TE0_AI
	D9			P8_4/S_D20/TA0_I4/TE0_AI
TA0_O5	R18			P0_5/P_D5/TA0_I5
	E9			P8_5/S_D21/TA0_I5
TA0_O6	T22			P0_6/P_D6/TA0_I6/TE0_BI
	A10			P8_6/S_D22/TA0_I6/TE0_BI
TA0_O7	T21			P0_7/P_D7/TA0_I7
	B10			P8_7/S_D23/TA0_I7
TA0_O8	T19			P0_8/P_D8/TA0_I8/TE0_ZI
	D10			P8_8/S_D24/TA0_I8/TE0_ZI
TA0_O9	T18			P0_9/P_D9/TA0_I9
	E10			P8_9/S_D25/TA0_I9
TA0_O10	U22			P0_10/P_D10/TA0_I10
	B11			P8_10/S_D26/TA0_I10
TA0_O11	U21			P0_11/P_D11/TA0_I11
	D11			P8_11/S_D27/TA0_I11
TA0_O12	U19			P0_12/P_D12/TA0_I12
	E11			P8_12/S_D28/TA0_I12
TA0_O13	V22			P0_13/P_D13/TA0_I13
	D12			P8_13/S_D29/TA0_I13
TA0_O14	V21			P0_14/P_D14/TA0_I14
	E12			P8_14/S_D30/TA0_I14
TA0_O15	W22			P0_15/P_D15/TA0_I15
	E13			P8_15/S_D31/TA0_I15

Table 2-4 Peripheral Function Pins (15/19)

Pin Name	Pin No.	I/O	Description	Alternate-Function Pin
TA1_O0	M19	Output	16-bit timer array 1 external output	P3_0/P_A0/TA1_I0/TE1_TI0/INTP18
	K1			P11_0/S_A1/TA1_I0/TE1_TI0
TA1_O1	M21			P3_1/P_A1/TA1_I1
	K4			P11_1/S_A2/TA1_I1
TA1_O2	L19			P3_2/P_A2/TA1_I2/TE1_TI1
	L2			P11_2/S_A3/TA1_I2/TE1_TI1
TA1_O3	L18			P3_3/P_A3/TA1_I3
	L1			P11_3/S_A4/TA1_I3
TA1_O4	K18			P3_4/P_A4/TA1_I4/TE1_AI
	L5			P11_4/S_A5/TA1_I4/TE1_AI
TA1_O5	L21			P3_5/P_A5/TA1_I5
	L4			P11_5/S_A6/TA1_I5
TA1_O6	L22			P3_6/P_A6/TA1_I6/CSI3F_CS0/TE1_BI
	M2			P11_6/S_A7/TA1_I6/TE1_BI
TA1_O7	K19			P3_7/P_A7/TA1_I7/CSI3F_CS1
	M1			P11_7/S_A8/TA1_I7
TA1_O8	K21			P3_8/P_A8/TA1_I8/CSI3F_CS2/TE1_ZI
	N2			P11_8/S_A9/TA1_I8/TE1_ZI
TA1_O9	K22			P3_9/P_A9/TA1_I9/CSI3F_CS3
	N1			P11_9/S_A10/TA1_I9
TA1_O10	J19			P3_10/P_A10/TA1_I10/CSI3F_CS4
	P2			P11_10/S_A11/TA1_I10/CSI1F_CS0
TA1_O11	J21			P3_11/P_A11/TA1_I11/CSI3F_CS5
	P1			P11_11/S_A12/TA1_I11/CSI1F_CS1
TA1_O12	J22			P3_12/P_A12/TA1_I12/CSI3F_CS6
	M4			P11_12/S_A13/TA1_I12/CSI1F_CS2
TA1_O13	H19			P3_13/P_A13/TA1_I13/CSI3F_CS7
	M5			P11_13/S_A14/TA1_I13/CSI1F_CS3
TA1_O14	H21			P3_14/P_A14/TA1_I14/CSI3_RYI/ CSI3_RYO
	R2			P11_14/S_A15/TA1_I14/CSI1F_CS4
TA1_O15	H22			P3_15/P_A15/TA1_I15/SSHI7
	R1			P12_15/S_A16/TA1_I15/CSI1F_CS5

Table 2-4 Peripheral Function Pins (16/19)

Pin Name	Pin No.	I/O	Description	Alternate-Function Pin
TA2_O0	W21	Output	16-bit timer array 2 external output	P1_0/P_D16/TA2_I0/ADCNV0
	C2			P7_0/S_D0/TA2_I0
TA2_O1	Y22			P1_1/P_D17/TA2_I1/ADCNV1
	A2			P7_1/S_D1/TA2_I1
TA2_O2	Y21			P1_2/P_D18/TA2_I2/ADCNV2
	B2			P7_2/S_D2/TA2_I2
TA2_O3	AA22			P1_3/P_D19/TA2_I3/ESO0
	A3			P7_3/S_D3/TA2_I3
TA2_O4	AA21			P1_4/P_D20/TA2_I4/ESO1
	B3			P7_4/S_D4/TA2_I4
TA2_O5	AB21			P1_5/P_D21/TA2_I5/ESO2
	A4			P7_5/S_D5/TA2_I5
TA2_O6	W19			P1_6/P_D22/TA2_I6/ESO3
	B4			P7_6/S_D6/TA2_I6
TA2_O7	AA20			P1_7/P_D23/TA2_I7
	A5			P7_7/S_D7/TA2_I7
TA2_O8	AB20			P1_8/P_D24/TA2_I8
	B5			P7_8/S_D8/TA2_I8
TA2_O9	AA19			P1_9/P_D25/TA2_I9
	A6			P7_9/S_D9/TA2_I9
TA2_O10	AB19			P1_10/P_D26/TA2_I10
	B6			P7_10/S_D10/TA2_I10
TA2_O11	AA18			P1_11/P_D27/TA2_I11
	D5			P7_11/S_D11/TA2_I11
TA2_O12	AB18			P1_12/P_D28/TA2_I12
	A7			P7_12/S_D12/TA2_I12
TA2_O13	W17			P1_13/P_D29/TA2_I13
	D6			P7_13/S_D13/TA2_I13
TA2_O14	AA17			P1_14/P_D30/TA2_I14
	A8			P7_14/S_D14/TA2_I14
TA2_O15	AB17			P1_15/P_D31/TA2_I15
	E5			P7_15/S_D15/TA2_I15

Table 2-4 Peripheral Function Pins (17/19)

Pin Name	Pin No.	I/O	Description	Alternate-Function Pin
TA3_O0	A21	Output	16-bit timer array 3 external output	P5_0/ETH_CRS/TA3_I0/INTP20/ P_LLDQM/CSI0F_RYI/CSI0F_RYO
	D4			P9_0/S_SDCKE/TA3_I0/INTP10/CSI2F_CS0
TA3_O1	B20			P5_1/ETH_COL/TA3_I1/INTP21/ CSI0F_SSI/P_LUDQM
	E4			P9_2/S_SDCAS/TA3_I1/CSI2F_CS2
TA3_O2	A20			P5_2/ETH_TXD3/TA3_I2/INTP22/ CSI0F_CS0/P_ULDQM
	D2			P9_3/S_SDRAS/TA3_I2/CSI2F_CS3
TA3_O3	B19			P5_3/ETH_TXD2/TA3_I3/INTP23/ CSI0F_CS1/P_UUDQM
	D1			P9_4/S_LLDQM/TA3_I3/CSI2F_CS4
TA3_O4	A19			P5_4/ETH_TXD1/TA3_I4/INTP24/ CSI0F_CS2/P_REFRQ
	E2			P9_5/S_LUDQM/TA3_I4/CSI2F_CS5
TA3_O5	B18			P5_5/ETH_TXD0/TA3_I5/INTP25/ CSI0F_CS3/P_SDRAS
	G1			P9_12/S_RD/TA3_I5/INTP11
TA3_O6	A18			P5_6/ETH_TXEN/TA3_I6/INTP26/ CSI0F_CS4/P_SDCAS
	H2			P9_13/S_WR/TA3_I6/INTP12/CSI1_SSI
TA3_O7	A17			P5_7/ETH_TXCLK/TA3_I7/INTP27/CSI0F_CS5
	H1			P9_14/S_LLBE/TA3_I7/INTP13/ CSI1_RYI/CSI1_RYO
TA3_O8	D17			P5_8/ETH_TXER/TA3_I8/CSI0F_CS6/ P_SDCKE
	J1			P10_2/S_SDWE/TA3_I8/INTP17/CSI2F_SSI
TA3_O9	D18			P5_9/ETH_RXER/TA3_I9/CSI0F_CS7/P_CS4
	K2			P10_3/S_BCYST/TA3_I9/INTP18/ S_DMATC1/CSI4_SSI
TA3_O10	A16			P5_10/ETH_RXCLK/TA3_I10/P_BUSRQ
	W1			P10_4/S_CS0/TA3_I10/INTP19/CSI2_SSI
TA3_O11	D16			P5_11/ETH_RXDV/TA3_I11/P_SDWE
	W2			P10_5/S_CS1/TA3_I11/INTP24/ CSI2_RYI/CSI2_RYO
TA3_O12	B15			P5_12/ETH_RXD0/TA3_I12/P_BCYST
	Y1			P10_6/S_CS2/TA3_I12/INTP25/CSI1F_SSI
TA3_O13	D15			P5_13/ETH_RXD1/TA3_I13
	U5			P10_9/S_HLDAK/TA3_I13/SO2/INTP27
TA3_O14	E14			P5_14/ETH_RXD2/TA3_I14/DMAAK0/ TJ_I0/TJ_O0
	U4			P10_10/S_HLDRQ/TA3_I14/SCK2/TXD2
TA3_O15	D14			P5_15/ETH_RXD3/TA3_I15/DMAAK1/ TJ_I1/TJ_O1
	T4			P10_11/S_REFRQ/TA3_I15/CSI2F_RYI/ CSI2F_RYO

Table 2-4 Peripheral Function Pins (18/19)

Pin Name	Pin No.	I/O	Description	Alternate-Function Pin
TE0_AI	R19	Input	16-bit encoder timer 0 external input	P0_4/P_D4/TA0_I4/TA0_O4
	D9			P8_4/S_D20/TA0_I4/TA0_O4
TE0_BI	T22			P0_6/P_D6/TA0_I6/TA0_O6
	A10			P8_6/S_D22/TA0_I6/TA0_O6
TE0_TI0	P19			P0_0/P_D0/TA0_I0/TA0_O0
	D7			P8_0/S_D16/TA0_I0/TA0_O0
TE0_TI1	R22			P0_2/P_D2/TA0_I2/TA0_O2
	A9			P8_2/S_D18/TA0_I2/TA0_O2
TE0_ZI	T19			P0_8/P_D8/TA0_I8/TA0_O8
	D10			P8_8/S_D24/TA0_I8/TA0_O8
TE1_AI	K18	Input	16-bit encoder timer 1 external input	P3_4/P_A4/TA1_I4/TA1_O4
	L5			P11_4/S_A5/TA1_I4/TA1_O4
TE1_BI	L22			P3_6/P_A6/TA1_I6/TA1_O6/CSI3F_CS0
	M2			P11_6/S_A7/TA1_I6/TA1_O6
TE1_TI0	M19			P3_0/P_A0/TA1_I0/TA1_O0/INTP18
	K1			P11_0/S_A1/TA1_I0/TA1_O0
TE1_TI1	L19			P3_2/P_A2/TA1_I2/TA1_O2
	L2			P11_2/S_A3/TA1_I2/TA1_O2
TE1_ZI	K21			P3_8/P_A8/TA1_I8/TA1_O8/CSI3F_CS2
	N2			P11_8/S_A9/TA1_I8/TA1_O8
TJ_I0	P21	Input	32-bit timer array external input	P2_1/$\overline{\text{P\_LLBE}}$/$\overline{\text{P\_LLWR}}$/INTP13/TJ_O0
	E14			P5_14/ETH_RXD2/TA3_I14/TA3_O14/ $\overline{\text{DMAAK0}}$/TJ_O0
TJ_I1	P22			P2_2/$\overline{\text{P\_LUBE}}$/$\overline{\text{P\_LUWR}}$/INTP14/TJ_O1
	D14			P5_15/ETH_RXD3/TA3_I15/TA3_O15/ $\overline{\text{DMAAK1}}$/TJ_O1
TJ_I2	N18			P2_3/$\overline{\text{P\_ULBE}}$/$\overline{\text{P\_ULWR}}$/INTP15/TJ_O2
	B14			P6_0/ETH_MDC/$\overline{\text{DMATC0}}$/TJ_O2/P_A24
TJ_I3	N19			P2_4/$\overline{\text{P\_UUBE}}$/$\overline{\text{P\_UUWR}}$/INTP16/TJ_O3
	A15			P6_1/ETH_MDIO/$\overline{\text{DMATC1}}$/TJ_O3/ P_A25
TJ_O0	P21	Output	32-bit timer array external output	P2_1/$\overline{\text{P\_LLBE}}$/$\overline{\text{P\_LLWR}}$/INTP13/TJ_I0
	E14			P5_14/ETH_RXD2/TA3_I14/TA3_O14/ $\overline{\text{DMAAK0}}$/TJ_I0
TJ_O1	P22			P2_2/$\overline{\text{P\_LUBE}}$/$\overline{\text{P\_LUWR}}$/INTP14/TJ_I1
	D14			P5_15/ETH_RXD3/TA3_I15/TA3_O15/ $\overline{\text{DMAAK1}}$/TJ_I1
TJ_O2	N18			P2_3/$\overline{\text{P\_ULBE}}$/$\overline{\text{P\_ULWR}}$/INTP15/TJ_I2
	B14			P6_0/ETH_MDC/$\overline{\text{DMATC0}}$/TJ_I2/P_A24
TJ_O3	N19			P2_4/$\overline{\text{P\_UUBE}}$/$\overline{\text{P\_UUWR}}$/INTP16/TJ_I3
	A15			P6_1/ETH_MDIO/$\overline{\text{DMATC1}}$/TJ_I3/P_A25

Table 2-4 Peripheral Function Pins (19/19)

Pin Name	Pin No.	I/O	Description	Alternate-Function Pin
TXD0	V1	Output	UART0 serial transmit data output	P12_6/S_A23/INTP6/SCK0/ADTRG11
TXD1	J4	Output	UART1 serial transmit data output	P10_1/$\overline{\text{S\_UUBE}}$/INTP16/SCK1/ $\overline{\text{S\_DMATC1}}$
TXD2	U4	Output	UART2 serial transmit data output	P10_10/$\overline{\text{S\_HLDRQ}}$/TA3_I14/TA3_O14/ SCK2
TXD3	F22	Output	UART3 serial transmit data output	P4_5/P_A21/INTP10/DMATC3/SCK3/ ADTRG10
TXD4	G2	Output	UART4 serial transmit data output	P9_11/$\overline{\text{S\_UUWR}}$/$\overline{\text{S\_DMARQ3}}$/SCK4/ SCL4/ESO3
TXD5	AA1	Output	UART5 serial transmit data output	P13_6/CAN1TXD/SCK5/SCL5/INTP4
TXD0F	C22	Output	UART0 (FIFO) serial transmit data output	P4_11/$\overline{\text{P\_WAIT}}$/SCK0F/SCL0
TXD1F	AB2	Output	UART1 (FIFO) serial transmit data output	P13_4/CAN0TXD/SCK1F/SCL1/INTP3
TXD2F	V2	Output	UART2 (FIFO) serial transmit data output	P12_9/S_A26/INTP9/SCK2F/SCL2/ ADTRG21
TXD3F	E22	Output	UART3 (FIFO) serial transmit data output	P4_7/P_A23/INTP12/SCK3F/SCL3/ ADTRG20
UCLK	AB3	Input	USB clock signal input	P13_0/$\overline{\text{S\_DMARQ0}}$/ADTRG01/CSI5_SSI
UDMF	A14	I/O	USB data I/O (–) function	–
UDMH	A12	I/O	USB data I/O (–) host	–
UDPF	A13	I/O	USB data I/O (+) function	–
UDPH	A11	I/O	USB data I/O (+) host	–
UV_{DD}	B13	–	Positive power supply for USB (3.3 V)	–

Table 2-5 Other Pins (1/2)

Pin Name	Pin No.	I/O	Description	Alternate-Function Pin
DV_{DD}	U10,V12	–	Positive power supply for debugging (3.3 V)	–
EV_{DD}	[a]	–	Positive power supply for external pins (3.3 V) (including power supply for flash)	–
$\overline{EVTI}$	W11	Input	Pin for debugging	–
$\overline{EVTO}$	V11	Output	Pin for debugging	–
FLMD0	D13	Input	Flash memory programming mode setting pins	–
FLMD1	AA16			–
FLRXD	W14	Input	Flash memory programmer pin (for 1-wired UART)	TDI/FLSI
FLSCK	AB13	I/O	Flash memory programmer pins (for 3-wired HS CSI)	TCK
FLSI	W14	Input		TDI/FLRXD
FLSO	W13	Output		TDO
IV_{DD}	[b]	–	Positive power supply for internal units (1.2 V)	–
MCKO	AB12	Output	Pin for debugging	–
MDO0	AA11	Output	Trace data output	–
MDO1	AB11			–
MDO2	V10			–
MDO3	W10			–
MDO4	AA10			–
MDO5	AB10			–
MDO6	V9			–
MDO7	W9			–
MODE2	D19	Input	Operation mode setting	–
MODE3	E6			–
MSEO0	W12	Output	Pins for debugging	–
MSEO1	AA12			–
$OSCV_{DD}$	AA14	–	Positive power supply for resonators (3.3 V)	–
$OSCV_{SS}$	AB14	–	Ground potential for resonators	–
$PLLV_{DD}$	U14	–	Positive power supply for PLL synthesizers (1.2 V)	–
$PLLV_{SS}$	U15	–	Ground potential for PLL synthesizers	–
$\overline{RESET}$	AB16	Input	System reset input	–
TCK	AB13	Input	JTAG interface (clock input)	FLSCK
TDI	W14	Input	JTAG interface (data input)	FLRXD/FLSI
TDO	W13	Output	JTAG interface (data output)	FLSO
TMS	W15	Input	JTAG Interface (mode select signal input)	–
$\overline{TRDY}$	AA13	Output	Pin for debugging	–

Table 2-5 Other Pins (2/2)

Pin Name	Pin No.	I/O	Description	Alternate-Function Pin
$\overline{TRST}$	V14	Input	JTAG Interface (reset signal input)	–
V_{SS}	[c]	–	Ground potential for internal units and external pins	–
X1	AB15	Input	Crystal connection pins for system clock oscillation	–
X2	AA15	–		–

a) B7, B16, E8, E17, F5, F6, F8, F9, F14, F17, F18, J5, J6, J17, J18, N17, T5, T6, U8, U17, V17, W18
b) E15, F15, G5, G6, H17, H18, N6, T17, U6, U16, V16, AA9
c) B1, B8, B12, B17, E7, E16, E18, E19, F7, F10, F13, F16, G17, G18, H5, H6, K5, K6, K17, M22, P6, P17, R5, R6, R17, U7, U9, U13, U18, V13, V15, V18, V19, AA4, AB9

2.3 Duplication of Pin Functions

For some functions of this product, the input and/or output is assigned to two different pins. Note, however, that the function can be enabled for only one of the corresponding pins at a time. In other words, it is prohibited to enable the function for both the corresponding pins at the same time.

For example, while the a/b/c pin functions as b, the b/d/e pin must not function as b. Specify the d or e function for the b/d/e pin at this time.

Caution When using the Pn_m port as the alternative output function (PMCn.PMCnm = 1, PMn.PMnm = 0), the level of the Pn_m pin can be read by PPRn.PPRnm by enabling the bidirectional mode (PBDCn.PBDCnm = 1).

However, note that in this case, the level of the Pn_m pin is input to the same alternative input function.

Table 2-6 Duplication of Pin Functions (1/9)

I/O	Assigned Function	Alternate Functions	Pin No.
Output	ADCNV0	P1_0/P_D16/TA2_I0/TA2_O0	W21
		P12_0/S_A17/INTP0/CSI1F_CS6	T2
	ADCNV1	P1_1/P_D17/TA2_I1/TA2_O1	Y22
		P12_1/S_A18/INTP1/CSI1F_CS7/ADCNV1	N4
	ADCNV2	P1_2/P_D18/TA2_I2/TA2_O2	Y21
		P12_2/S_A19/INTP2/CSI0_SSI	N5
Input	ESO0	P1_4/P_D19/TA2_I3/TA2_O3	AA22
		P9_6/S_ULDQM/$\overline{\text{S\_DMATC2}}$/CSI2F_CS6	F4
	ESO1	P1_4/P_D20/TA2_I4/TA2_O4	AA21
		P9_7/S_UUDQM/$\overline{\text{S\_DMATC3}}$/CSI2F_CS7	E1
	ESO2	P1_5/P_D21/TA2_I5/TA2_O5	AB21
		P9_10/$\overline{\text{S\_ULWR}}$/$\overline{\text{S\_DMAAK3}}$/SI4/RXD4/SDA4	G4
	ESO3	P1_6/P_D22/TA2_I6/TA2_O6	W19
		P9_11/$\overline{\text{S\_UUWR}}$/$\overline{\text{S\_DMARQ3}}$/SCK4/TXD4/SCL4	G2
Input	$\overline{\text{P\_BCYST}}$	P5_12/ETH_RXD0/TA3_I12/TA3_O12	B15
		P4_8/$\overline{\text{P\_CS1}}$/INTP20	D21
Input	INTP0	P12_0/S_A17/CSI1F_CS6/ADCNV0	T2
		P13_1/$\overline{\text{S\_CS3}}$/$\overline{\text{S\_DMAAK0}}$/ADTRG00/CSI5_RYI/CSI5_RYO	W4
	INTP1	P12_1/S_A18/CSI1F_CS7/ADCNV1	N4
		P13_2/$\overline{\text{S\_DMARQ1}}$/SO1F/PPON	AB4
	INTP2	P12_2/S_A19/ADCNV2/CSI0_SSI	N5
		P13_3/$\overline{\text{S\_DMAAK1}}$/SO5/$\overline{\text{OCI}}$	AA3
	INTP3	P12_3/S_A20/CSI0_RYI/CSI0_RYO	T1
		P13_4/CAN0TXD/SCK1F/TXD1F/SCL1	AB2
	INTP4	P12_4/S_A21/SO0	U2
		P13_6/CAN1TXD/SCK5/TXD5/SCL5	AA1
	INTP5	P4_0/P_A16/CSI3F_RYI/CSI3F_RYO	G19
		P12_5/S_A22/SI0/RXD0	U1

Table 2-6 Duplication of Pin Functions (2/9)

I/O	Assigned Function	Alternate Functions	Pin No.
Input	INTP6	P4_1/P_A17/CSI3F_SSI	G21
		P12_6/S_A23/SCK0/TXD0/ADTRG11	V1
	INTP7	P4_2/P_A18/SO3	G22
		P12_7/S_A24/SO2F	P4
	INTP8	P4_3/P_A19/$\overline{\text{DMAAK4}}$/SO3F	F19
		P12_8/S_A25/SI2F/RXD2F/SDA2	P5
	INTP9	P4_4/P_A20/$\overline{\text{DMATC4}}$/SI3/RXD3	F21
		P12_9/S_A26/SCK2F/TXD2F/SCL2/ADTRG21	V2
	INTP10	P4_5/P_A21/$\overline{\text{DMATC3}}$/SCK3/TXD3/ADTRG10	F22
		P9_0/S_SDCKE/TA3_I0/TA3_O0/CSI2F_CS0	D4
	INTP11	P4_6/P_A22/$\overline{\text{DMAAK3}}$/SI3F/RXD3F/SDA3	E21
		P9_12/$\overline{\text{S\_RD}}$/TA3_I5/TA3_O5	G1
	INTP12	P4_7/P_A23/SCK3F/TXD3F/SCL3/ADTRG20	E22
		P9_13/$\overline{\text{S\_WR}}$/TA3_I6/TA3_O6/CSI1_SSI	H2
	INTP13	P2_1/$\overline{\text{P\_LLBE}}$/$\overline{\text{P\_LLWR}}$/TJ_I0/TJ_O0	P21
		P9_14/$\overline{\text{S\_LLBE}}$/TA3_I7/TA3_O7/CSI1_RYI/CSI1_RYO	H1
	INTP14	P2_2/$\overline{\text{P\_LUBE}}$/$\overline{\text{P\_LUWR}}$/TJ_I1/TJ_O1	P22
		P9_15/$\overline{\text{S\_LUBE}}$/SO1	H4
	INTP15	P2_3/$\overline{\text{P\_ULBE}}$/$\overline{\text{P\_ULWR}}$/TJ_I2/TJ_O2	N18
		P10_0/$\overline{\text{S\_ULBE}}$/SI1/RXD1/$\overline{\text{S\_DMATC0}}$	J2
	INTP16	P2_4/$\overline{\text{P\_UUBE}}$/$\overline{\text{P\_UUWR}}$/TJ_I3/TJ_O3	N19
		P10_1/$\overline{\text{S\_UUBE}}$/SCK1/TXD1/$\overline{\text{S\_DMATC1}}$	J4
	INTP17	P2_5/$\overline{\text{P\_RD}}$	N21
		P10_2/$\overline{\text{S\_SDWE}}$/TA3_I8/TA3_O8/CSI2F_SSI	J1
	INTP18	P3_0/P_A0/TA1_I0/TA1_O0/TE1_TI0	M19
		P10_3/$\overline{\text{S\_BCYST}}$/TA3_I9/TA3_O9/$\overline{\text{S\_DMATC1}}$/CSI4_SSI	K2
	INTP19	P2_7/$\overline{\text{P\_WR}}$/$\overline{\text{P\_RW}}$	M18
		P10_4/$\overline{\text{S\_CS0}}$/TA3_I10/TA3_O10/CSI2_SSI	W1
	INTP20	P4_8/$\overline{\text{P\_CS1}}$/$\overline{\text{P\_BCYST}}$	D21
		P5_0/ETH_CRS/TA3_I0/TA3_O0/P_LLDQM/CSI0F_RYI/CSI0F_RYO	A21
	INTP21	P4_9/$\overline{\text{P\_CS2}}$/$\overline{\text{DMAAK5}}$	D22
		P5_1/ETH_COL/TA3_I1/TA3_O1/CSI0F_SSI/P_LUDQM	B20
	INTP22	P4_10/$\overline{\text{P\_CS3}}$/$\overline{\text{DMATC5}}$	C21
		P5_2/ETH_TXD3/TA3_I2/TA3_O2/CSI0F_CS0/P_ULDQM	A20
	INTP23	P4_12/$\overline{\text{P\_HLDAK}}$/$\overline{\text{DMAAK2}}$/SO0F	B21
		P5_3/ETH_TXD2/TA3_I3/TA3_O3/CSI0F_CS1/P_UUDQM	B19
	INTP24	P5_4/ETH_TXD1/TA3_I4/TA3_O4/CSI0F_CS2/$\overline{\text{P\_REFRQ}}$	B19
		P10_5/$\overline{\text{S\_CS1}}$/TA3_I11/TA3_O11/CSI2_RYI/CSI2_RYO	W2
	INTP25	P5_5/ETH_TXD0/TA3_I5/TA3_O5/CSI0F_CS3/$\overline{\text{P\_SDRAS}}$	B18
		P10_6/$\overline{\text{S\_CS2}}$/TA3_I12/TA3_O12/CSI1F_SSI	Y1
	INTP26	P5_6/ETH_TXEN/TA3_I6/TA3_O6/CSI0F_CS4/$\overline{\text{P\_SDCAS}}$	A18
		P10_7/$\overline{\text{S\_CS3}}$/$\overline{\text{S\_SDCS}}$/CSI1F_RYI/CSI1F_RYO	R4
	INTP27	P5_7/ETH_TXCLK/TA3_I7/TA3_O7/CSI0F_CS5	A17
		P10_9/$\overline{\text{S\_HLDAK}}$/TA3_I13/TA3_O13/SO2	U5

Table 2-6 Duplication of Pin Functions (3/9)

I/O	Assigned Function	Alternate Functions	Pin No.
Output	S_DMATC0	P9_9/S_LUWR/S_DMARQ2/SO4	F1
		P10_0/S_ULBE/INTP15/SI1/RXD1	J2
	S_DMATC1	P10_1/S_UUBE/INTP16/SCK1/TXD1	J4
		P10_3/S_BCYST/TA3_I9/TA3_O9/INTP18/CSI4_SSI	K2
	S_CS3	P10_7/S_SDCS/INTP26/CSI1F_RYI/CSI1F_RYO	R4
		P13_1/S_DMAAK0/ADTRG00/INTP0/CSI5_RYI/CSI5_RYO	W4
Input	TA0_I0	P0_0/P_D0/TA0_O0/TE0_TI0	P19
		P8_0/S_D16/TA0_O0/TE_TI0	D7
	TA0_I1	P0_1/P_D1/TA0_O1	P18
		P8_1/S_D17/TA0_O1	D8
	TA0_I2	P0_2/P_D2/TA0_O2/TE0_TI1	R22
		P8_2/S_D18/TA0_O2/TE0_TI1	A9
	TA0_I3	P0_3/P_D3/TA0_O3	R21
		P8_3/S_D19/TA0_O3	B9
	TA0_I4	P0_4/P_D4/TA0_O4/TE0_AI	R19
		P8_4/S_D20/TA0_O4/TE0_AI	D9
	TA0_I5	P0_5/P_D5/TA0_O5	R18
		P8_5/S_D21/TA0_O5	E9
	TA0_I6	P0_6/P_D6/TA0_O6/TE0_BI	T22
		P8_6/S_D22/TA0_O6/TE0_BI	A10
	TA0_I7	P0_7/P_D7/TA0_O7	T21
		P8_7/S_D23/TA0_O7	B10
	TA0_I8	P0_8/P_D8/TA0_O8/TE0_ZI	T19
		P8_8/S_D24/TA0_O8/TE0_ZI	D10
	TA0_I9	P0_9/P_D9/TA0_O9	T18
		P8_9/S_D25/TA0_O9	E10
	TA0_I10	P0_10/P_D10/TA0_O10	U22
		P8_10/S_D26/TA0_O10	B11
	TA0_I11	P0_11/P_D11/TA0_O11	U21
		P8_11/S_D27/TA0_O11	D11
	TA0_I12	P0_12/P_D12/TA0_O12	U19
		P8_12/S_D28/TA0_O12	E11
	TA0_I13	P0_13/P_D13/TA0_O13	V22
		P8_13/S_D29/TA0_O13	D12
	TA0_I14	P0_14/P_D14/TA0_O14	V21
		P8_14/S_D30/TA0_O14	E12
	TA0_I15	P0_15/P_D15/TA0_O15	W22
		P8_15/S_D31/TA0_O15	E13
	TA1_I0	P3_0/P_A0/TA1_O0/TE1_TI0/INTP18	M19
		P11_0/S_A1/TA1_O0/TE1_TI0	K1
	TA1_I1	P3_1/P_A1/TA1_O1	M21
		P11_1/S_A2/TA1_O1	K4
	TA1_I2	P3_2/P_A2/TA1_O2/TE1_TI1	L19
		P11_2/S_A3/TA1_O2/TE1_TI1	L2

Table 2-6 Duplication of Pin Functions (4/9)

I/O	Assigned Function	Alternate Functions	Pin No.
Input	TA1_I3	P3_3/P_A3/TA1_O3	L18
		P11_3/S_A4/TA1_O3	L1
	TA1_I4	P3_4/P_A4/TA1_O4/TE1_AI	K18
		P11_4/S_A5/TA1_O4/TE1_AI	L5
	TA1_I5	P3_5/P_A5/TA1_O5	L21
		P11_5/S_A6/TA1_O5	L4
	TA1_I6	P3_6/P_A6/TA1_O6/CSI3F_CS0/TE1_BI	L22
		P11_6/S_A7/TA1_O6/TE1_BI	M2
	TA1_I7	P3_7/P_A7/TA1_O7/CSI3F_CS1	K19
		P11_7/S_A8/TA1_O7	M1
	TA1_I8	P3_8/P_A8/TA1_O8/CSI3F_CS2/TE1_ZI	K21
		P11_8/S_A9/TA1_O8/TE1_ZI	N2
	TA1_I9	P3_9/P_A9/TA1_O9/CSI3F_CS3	K22
		P11_9/S_A10/TA1_O9	N1
	TA1_I10	P3_10/P_A10/TA1_O10/CSI3F_CS4	J19
		P11_10/S_A11/TA1_O10/CSI1F_CS0	P2
	TA1_I11	P3_11/P_A11/TA1_O11/CSI3F_CS5	J21
		P11_11/S_A12/TA1_O11/CSI1F_CS1	P1
	TA1_I12	P3_12/P_A12/TA1_O12/CSI3F_CS6	J22
		P11_12/S_A13/TA1_O12/CSI1F_CS2	M4
	TA1_I13	P3_13/P_A13/TA1_O13/CSI3F_CS7	H19
		P11_13/S_A14/TA1_O13/CSI1F_CS3	M5
	TA1_I14	P3_14/P_A14/TA1_O14/CSI3_RYI/CSI3_RYO	H21
		P11_14/S_A15/TA1_O14/CSI1F_CS4	R2
	TA1_I15	P3_15/P_A15/TA1_O15/CSI3_SSI	H22
		P11_15/S_A16/TA1_O15/CSI1F_CS5	R1
	TA2_I0	P1_0/P_D16/TA2_O0/ADCNV0	W21
		P7_0/S_D0/TA2_O0	C2
	TA2_I1	P1_1/P_D17/TA2_O1/ADCNV1	Y22
		P7_1/S_D1/TA2_O1	A2
	TA2_I2	P1_2/P_D18/TA2_O2/ADCNV2	Y21
		P7_2/S_D2/TA2_O2	B2
	TA2_I3	P1_3/P_D19/TA2_O3/ESO0	AA22
		P7_3/S_D3/TA2_O3	A3
	TA2_I4	P1_4/P_D20/TA2_O4/ESO1	AA21
		P7_4/S_D4/TA2_O4	B3
	TA2_I5	P1_5/P_D21/TA2_O5/ESO2	AB21
		P7_5/S_D5/TA2_O5	A4
	TA2_I6	P1_6/P_D22/TA2_O6/ESO3	W19
		P7_6/S_D6/TA2_O6	B4
	TA2_I7	P1_7/P_D23/TA2_O7	AA20
		P7_7/S_D7/TA2_O7	A5
	TA2_I8	P1_8/P_D24/TA2_O8	AB20
		P7_8/S_D8/TA2_O8	B5

Table 2-6 Duplication of Pin Functions (5/9)

I/O	Assigned Function	Alternate Functions	Pin No.
Input	TA2_I9	P1_9/P_D25/TA2_O9	AA19
		P7_9/S_D9/TA2_O9	A6
	TA2_I10	P1_10/P_D26/TA2_O10	AB19
		P7_10/S_D10/TA2_O10	B6
	TA2_I11	P1_11/P_D27/TA2_O11	AA18
		P7_11/S_D11/TA2_O11	D5
	TA2_I12	P1_12/P_D28/TA2_O12	AB18
		P7_12/S_D12/TA2_O12	A7
	TA2_I13	P1_13/P_D29/TA2_O13	W17
		P7_13/S_D13/TA2_O13	D6
	TA2_I14	P1_14/P_D30/TA2_O14	AA17
		P7_14/S_D14/TA2_O14	A8
	TA2_I15	P1_15/P_D31/TA2_O15	AB17
		P7_15/S_D15/TA2_O15	E5
	TA3_I0	P5_0/ETH_CRS/TA3_O0/INTP20/P_LLDQM/CSI0F_RYI/ CSI0F_RYO	A21
		P9_0/S_SDCKE/TA3_O0/INTP10/CSI2F_CS0	D4
	TA3_I1	P5_1/ETH_COL/TA3_O1/INTP21/CSI0F_SSI/P_LUDQM	B20
		P9_2/$\overline{\text{S\_SDCAS}}$/TA3_O1/CSI2F_CS2	E4
	TA3_I2	P5_2/ETH_TXD3/TA3_O2/INTP22/CSI0F_CS0/P_ULDQM	A20
		P9_3/$\overline{\text{S\_SDRAS}}$/TA3_O2/CSI2F_CS3	D2
	TA3_I3	P5_3/ETH_TXD2/TA3_O3/INTP23/CSI0F_CS1/P_UUDQM	B19
		P9_4/S_LLDQM/TA3_O3/CSI2F_CS4	D1
	TA3_I4	P5_4/ETH_TXD1/TA3_O4/INTP24/CSI0F_CS2/$\overline{\text{P\_REFRQ}}$	A19
		P9_5/S_LUDQM/TA3_O4/CSI2F_CS5	E2
	TA3_I5	P5_5/ETH_TXD0/TA3_O5/INTP25/CSI0F_CS3/$\overline{\text{P\_SDRAS}}$	B18
		P9_12/$\overline{\text{S\_RD}}$/TA3_O5/INTP11	G1
	TA3_I6	P5_6/ETH_TXEN/TA3_O6/INTP26/CSI0F_CS4/$\overline{\text{P\_SDCAS}}$	A18
		P9_13/$\overline{\text{S\_WR}}$/TA3_O6/INTP12/CSI1_SSI	H2
	TA3_I7	P5_7/ETH_TXCLK/TA3_O7/INTP27/CSI0F_CS5	A17
		P9_14/$\overline{\text{S\_LLBE}}$/TA3_O7/INTP13/CSI1_RYI/CSI1_RYO	H1
	TA3_I8	P5_8/ETH_TXER/TA3_O8/CSI0F_CS6/P_SDCKE	D17
		P10_2/$\overline{\text{S\_SDWE}}$/TA3_O8/INTP17/CSI2F_SSI	J1
	TA3_I9	P5_9/ETH_RXER/TA3_O9/CSI0F_CS7/$\overline{\text{P\_CS4}}$	D18
		P10_3/$\overline{\text{S\_BCYST}}$/TA3_O9/INTP18/$\overline{\text{S\_DMATC1}}$/CSI4_SSI	K2
	TA3_I10	P5_10/ETH_RXCLK/TA3_O10/P_BUSRQ	A16
		P10_4/$\overline{\text{S\_CS0}}$/TA3_O10/INTP19/CSI2_SSI	W1
	TA3_I11	P5_11/ETH_RXDV/TA3_O11/$\overline{\text{P\_SDWE}}$	D16
		P10_5/$\overline{\text{S\_CS1}}$/TA3_O11/INTP24/CSI2_RYI/CSI2_RYO	W2
	TA3_I12	P5_12/ETH_RXD0/TA3_O12/$\overline{\text{P\_BCYST}}$	B15
		P10_6/$\overline{\text{S\_CS2}}$/TA3_O12/INTP25/CSI1F_SSI	Y1
	TA3_I13	P5_13/ETH_RXD1/TA3_O13	D15
		P10_9/$\overline{\text{S\_HLDAK}}$/TA3_O13/SO2/INTP27	U5

Table 2-6 Duplication of Pin Functions (6/9)

I/O	Assigned Function	Alternate Functions	Pin No.
Input	TA3_I14	P5_14/ETH_RXD2/TA3_O14/$\overline{\text{DMAAK0}}$/TJ_I0/TJ_O0	E14
		P10_10/$\overline{\text{S\_HLDRQ}}$/TA3_O14/SCK2/TXD2	U4
	TA3_I15	P5_15/ETH_RXD3/TA3_O15/$\overline{\text{DMAAK1}}$/TJ_I1/TJ_O1	D14
		P10_11/$\overline{\text{S\_REFRQ}}$/TA3_O15/CSI2F_RYI/CSI2F_RYO	T4
	TE0_AI	P0_4/P_D4/TA0_I4/TA0_O4	R19
		P8_4/S_D20/TA0_I4/TA0_O4	D9
	TE0_BI	P0_6/P_D6/TA0_I6/TA0_O6	T22
		P8_6/S_D22/TA0_I6/TA0_O6	A10
	TE0_TI0	P0_0/P_D0/TA0_I0/TA0_O0	P19
		P8_0/S_D16/TA0_I0/TA0_O0	D7
	TE0_TI1	P0_2/P_D2/TA0_I2/TA0_O2	R22
		P8_2/S_D18/TA0_I2/TA0_O2	A9
	TE0_ZI	P0_8/P_D8/TA0_I8/TA0_O8	T19
		P8_8/S_D24/TA0_I8/TA0_O8	D10
	TE1_AI	P3_4/P_A4/TA1_I4/TA1_O4	K18
		P11_4/S_A5/TA1_I4/TA1_O4	L5
	TE1_BI	P3_6/P_A6/TA1_I6/TA1_O6/CSI3F_CS0	L22
		P11_6/S_A7/TA1_I6/TA1_O6	M2
	TE1_TI0	P3_0/P_A0/TA1_I0/TA1_O0/INTP18	M19
		P11_0/S_A1/TA1_I0/TA1_O0	K1
	TE1_TI1	P3_2/P_A2/TA1_I2/TA1_O2	L19
		P11_2/S_A3/TA1_I2/TA1_O2	L2
	TE1_ZI	P3_8/P_A8/TA1_I8/TA1_O8/CSI3F_CS2	K21
		P11_8/S_A9/TA1_I8/TA1_O8	N2
	TJ_I0	P2_1/$\overline{\text{P\_LLBE}}$/$\overline{\text{P\_LLWR}}$/INTP13/TJ_O0	P21
		P5_14/ETH_RXD2/TA3_I14/TA3_O14/$\overline{\text{DMAAK0}}$/TJ_O0	E14
	TJ_I1	P2_2/$\overline{\text{P\_LUBE}}$/$\overline{\text{P\_LUWR}}$/INTP14/TJ_O1	P22
		P5_15/ETH_RXD3/TA3_I15/TA3_O15/$\overline{\text{DMAAK1}}$/TJ_O1	D14
	TJ_I2	P2_3/$\overline{\text{P\_ULBE}}$/$\overline{\text{P\_ULWR}}$/INTP15/TJ_O2	N18
		P6_0/ETH_MDC/$\overline{\text{DMATC0}}$/TJ_O2/P_A24	B14
	TJ_I3	P2_4/$\overline{\text{P\_UUBE}}$/$\overline{\text{P\_UUWR}}$/INTP16/TJ_O3	N19
		P6_1/ETH_MDIO/$\overline{\text{DMATC1}}$/TJ_O3/P_A25	A15
Output	TA0_O0	P0_0/P_D0/TA0_I0/TE0_TI0	P19
		P8_0/S_D16/TA0_I0/TE_TI0	D7
	TA0_O1	P0_1/P_D1/TA0_I1	P18
		P8_1/S_D17/TA0_I1	D8
	TA0_O2	P0_2/P_D2/TA0_I2/TE0_TI1	R22
		P8_2/S_D18/TA0_I2/TE0_TI1	A9
	TA0_O3	P0_3/P_D3/TA0_I3	R21
		P8_3/S_D19/TA0_I3	B9
	TA0_O4	P0_4/P_D4/TA0_I4/TE0_AI	R19
		P8_4/S_D20/TA0_I4/TE0_AI	D9
	TA0_O5	P0_5/P_D5/TA0_I5	R18
		P8_5/S_D21/TA0_I5	E9

Table 2-6 Duplication of Pin Functions (7/9)

I/O	Assigned Function	Alternate Functions	Pin No.
Output	TA0_O6	P0_6/P_D6/TA0_I6/TE0_BI	T22
		P8_6/S_D22/TA0_I6/TE0_BI	A10
	TA0_O7	P0_7/P_D7/TA0_I7	T21
		P8_7/S_D23/TA0_I7	B10
	TA0_O8	P0_8/P_D8/TA0_I8/TE0_ZI	T19
		P8_8/S_D24/TA0_I8/TE0_ZI	D10
	TA0_O9	P0_9/P_D9/TA0_I9	T18
		P8_9/S_D25/TA0_I9	E10
	TA0_O10	P0_10/P_D10/TA0_I10	U22
		P8_10/S_D26/TA0_I10	B11
	TA0_O11	P0_11/P_D11/TA0_I11	U21
		P8_11/S_D27/TA0_I11	D11
	TA0_O12	P0_12/P_D12/TA0_I12	U19
		P8_12/S_D28/TA0_I12	E11
	TA0_O13	P0_13/P_D13/TA0_I13	V22
		P8_13/S_D29/TA0_I13	D12
	TA0_O14	P0_14/P_D14/TA0_I14	V21
		P8_14/S_D30/TA0_I14	E12
	TA0_O15	P0_15/P_D15/TA0_I15	W22
		P8_15/S_D31/TA0_I15	E13
	TA1_O0	P3_0/P_A0/TA1_I0/TE1_TI0/INTP18	M19
		P11_0/S_A1/TA1_I0/TE1_TI0	K1
	TA1_O1	P3_1/P_A1/TA1_I1	M21
		P11_1/S_A2/TA1_I1	K4
	TA1_O2	P3_2/P_A2/TA1_I2/TE1_TI1	L19
		P11_2/S_A3/TA1_I2/TE1_TI1	L2
	TA1_O3	P3_3/P_A3/TA1_I3	L18
		P11_3/S_A4/TA1_I3	L1
	TA1_O4	P3_4/P_A4/TA1_I4/TE1_AI	K18
		P11_4/S_A5/TA1_I4/TE1_AI	L5
	TA1_O5	P3_5/P_A5/TA1_I5	L21
		P11_5/S_A6/TA1_I5	L4
	TA1_O6	P3_6/P_A6/TA1_I6/CSI3F_CS0/TE1_BI	L22
		P11_6/S_A7/TA1_I6/TE1_BI	M2
	TA1_O7	P3_7/P_A7/TA1_I7/CSI3F_CS1	K19
		P11_7/S_A8/TA1_I7	M1
	TA1_O8	P3_8/P_A8/TA1_I8/CSI3F_CS2/TE1_ZI	K21
		P11_8/S_A9/TA1_I8/TE1_ZI	N2
	TA1_O9	P3_9/P_A9/TA1_I9/CSI3F_CS3	K22
		P11_9/S_A10/TA1_I9	N1
	TA1_O10	P3_10/P_A10/TA1_I10/CSI3F_CS4	J19
		P11_10/S_A11/TA1_I10/CSI1F_CS0	P2
	TA1_O11	P3_11/P_A11/TA1_I11/CSI3F_CS5	J21
		P11_11/S_A12/TA1_I11/CSI1F_CS1	P1

Table 2-6 Duplication of Pin Functions (8/9)

I/O	Assigned Function	Alternate Functions	Pin No.
Output	TA1_O12	P3_12/P_A12/TA1_I12/CSI3F_CS6	J22
		P11_12/S_A13/TA1_I12/CSI1F_CS2	M4
	TA1_O13	P3_13/P_A13/TA1_I13/CSI3F_CS7	H19
		P11_13/S_A14/TA1_I13/CSI1F_CS3	M5
	TA1_O14	P3_14/P_A14/TA1_I14/CSI3_RYI/CSI3_RYO	H21
		P11_14/S_A15/TA1_I14/CSI1F_CS4	R2
	TA1_O15	P3_15/P_A15/TA1_I15/CSI3_SSI	H22
		P12_15/S_A16/TA1_I15/CSI1F_CS5	R1
	TA2_O0	P1_0/P_D16/TA2_I0/ADCNV0	W21
		P7_0/S_D0/TA2_I0	C2
	TA2_O1	P1_1/P_D17/TA2_I1/ADCNV1	Y22
		P7_1/S_D1/TA2_I1	A2
	TA2_O2	P1_2/P_D18/TA2_I2/ADCNV2	Y21
		P7_2/S_D2/TA2_I2	B2
	TA2_O3	P1_3/P_D19/TA2_I3/ESO0	AA22
		P7_3/S_D3/TA2_I3	A3
	TA2_O4	P1_4/P_D20/TA2_I4/ESO1	AA21
		P7_4/S_D4/TA2_I4	B3
	TA2_O5	P1_5/P_D21/TA2_I5/ESO2	AB21
		P7_5/S_D5/TA2_I5	A4
	TA2_O6	P1_6/P_D22/TA2_I6/ESO3	W19
		P7_6/S_D6/TA2_I6	B4
	TA2_O7	P1_7/P_D23/TA2_I7	AA20
		P7_7/S_D7/TA2_I7	A5
	TA2_O8	P1_8/P_D24/TA2_I8	AB20
		P7_8/S_D8/TA2_I8	B5
	TA2_O9	P1_9/P_D25/TA2_I9	AA19
		P7_9/S_D9/TA2_I9	A6
	TA2_O10	P1_10/P_D26/TA2_I10	AB19
		P7_10/S_D10/TA2_I10	B6
	TA2_O11	P1_11/P_D27/TA2_I11	AA18
		P7_11/S_D11/TA2_I11	D5
	TA2_O12	P1_12/P_D28/TA2_I12	AB18
		P7_12/S_D12/TA2_I12	A7
	TA2_O13	P1_13/P_D29/TA2_I13	W17
		P7_13/S_D13/TA2_I13	D6
	TA2_O14	P1_14/P_D30/TA2_I14	AA17
		P7_14/S_D14/TA2_I14	A8
	TA2_O15	P1_15/P_D31/TA2_I15	AB17
		P7_15/S_D15/TA2_I15	E5
	TA3_O0	P5_0/ETH_CRS/TA3_I0/INTP20/P_LLDQM/CSI0F_RYI/CSI0F_RYO	A21
		P9_0/S_SDCKE/TA3_I0/INTP10/CSI2F_CS0	D4
	TA3_O1	P5_1/ETH_COL/TA3_I1/INTP21/CSI0F_SSI/P_LUDQM	B20
		P9_2/$\overline{\text{S\_SDCAS}}$/TA3_I1/CSI2F_CS2	E4

Table 2-6 Duplication of Pin Functions (9/9)

I/O	Assigned Function	Alternate Functions	Pin No.
Output	TA3_O2	P5_2/ETH_TXD3/TA3_I2/INTP22/CSI0F_CS0/P_ULDQM	A20
		P9_3/S_SDRAS/TA3_I2/CSI2F_CS3	D2
	TA3_O3	P5_3/ETH_TXD2/TA3_I3/INTP23/CSI0F_CS1/P_UUDQM	B19
		P9_4/S_LLDQM/TA3_I3/CSI2F_CS4	D1
	TA3_O4	P5_4/ETH_TXD1/TA3_I4/INTP24/CSI0F_CS2/P_REFRQ	A19
		P9_5/S_LUDQM/TA3_I4/CSI2F_CS5	E2
	TA3_O5	P5_5/ETH_TXD0/TA3_I5/INTP25/CSI0F_CS3/P_SDRAS	B18
		P9_12/S_RD/TA3_I5/INTP11	G1
	TA3_O6	P5_6/ETH_TXEN/TA3_I6/INTP26/CSI0F_CS4/P_SDCAS	A18
		P9_13/S_WR/TA3_I6/INTP12/CSI1_SSI	H2
	TA3_O7	P5_7/ETH_TXCLK/TA3_I7/INTP27/CSI0F_CS5	A17
		P9_14/S_LLBE/TA3_I7/INTP13/CSI1_RYI/CSI1_RYO	H1
	TA3_O8	P5_8/ETH_TXER/TA3_I8/CSI0F_CS6/P_SDCKE	D17
		P10_2/S_SDWE/TA3_I8/INTP17/CSI2F_SSI	J1
	TA3_O9	P5_9/ETH_RXER/TA3_I9/CSI0F_CS7/P_CS4	D18
		P10_3/S_BCYST/TA3_I9/INTP18/S_DMATC1/CSI4_SSI	K2
	TA3_O10	P5_10/ETH_RXCLK/TA3_I10/P_BUSRQ	A16
		P10_4/S_CS0/TA3_I10/INTP19/CSI2_SSI	W1
	TA3_O11	P5_11/ETH_RXDV/TA3_I11/P_SDWE	D16
		P10_5/S_CS1/TA3_I11/INTP24/CSI2_RYI/CSI2_RYO	W2
	TA3_O12	P5_12/ETH_RXD0/TA3_I12/P_BCYST	B15
		P10_6/S_CS2/TA3_I12/INTP25/CSI1F_SSI	Y1
	TA3_O13	P5_13/ETH_RXD1/TA3_I13	D15
		P10_9/S_HLDAK/TA3_I13/SO2/INTP27	U5
	TA3_O14	P5_14/ETH_RXD2/TA3_I14/DMAAK0/TJ_I0/TJ_O0	E14
		P10_10/S_HLDRQ/TA3_I14/SCK2/TXD2	U4
	TA3_O15	P5_15/ETH_RXD3/TA3_I15/DMAAK1/TJ_I1/TJ_O1	D14
		P10_11/S_REFRQ/TA3_I15/CSI2F_RYI/CSI2F_RYO	T4
	TJ_O0	P2_1/P_LLBE/P_LLWR/INTP13/TJ_I0	P21
		P5_14/ETH_RXD2/TA3_I14/TA3_O14/DMAAK0/TJ_I0	E14
	TJ_O1	P2_2/P_LUBE/P_LUWR/INTP14/TJ_I1	P22
		P5_15/ETH_RXD3/TA3_I15/TA3_O15/DMAAK1/TJ_I1	D14
	TJ_O2	P2_3/P_ULBE/P_ULWR/INTP15/TJ_I2	N18
		P6_0/ETH_MDC/DMATC0/TJ_I2/P_A24	B14
	TJ_O3	P2_4/P_UUBE/P_UUWR/INTP16/TJ_I3	N19
		P6_1/ETH_MDIO/DMATC1/TJ_I3/P_A25	A15

2.4 Pin State After Reset

The table below lists the state of each pin after reset.

Table 2-7 Pin State After Reset (1/8)

Pin No.	Pin Name	After Reset
A2	P7_1/S_D1/TA2_I1/TA2_O1	Hi-z
A3	P7_3/S_D3/TA2_I3/TA2_O3	Hi-z
A4	P7_5/S_D5/TA2_I5/TA2_O5	Hi-z
A5	P7_7/S_D7/TA2_I7/TA2_O7	Hi-z
A6	P7_9/S_D9/TA2_I9/TA2_O9	Hi-z
A7	P7_12/S_D12/TA2_I12/TA2_O12	Hi-z
A8	P7_14/S_D14/TA2_I14/TA2_O14	Hi-z
A9	P8_2/S_D18/TA0_I2/TA0_O2/TE0_TI1	Hi-z
A10	P8_6/S_D22/TA0_I6/TA0_O6/TE0_BI	Hi-z
A11	UDPH	Hi-z
A12	UDMH	Hi-z
A13	UDPF	Hi-z
A14	UDMF	Hi-z
A15	P6_1/ETH_MDIO/$\overline{\text{DMATC1}}$/TJ_I3/TJ_O3/P_A25	Hi-z
A16	P5_10/ETH_RXCLK/TA3_I10/TA3_O10/P_BUSRQ	Hi-z
A17	P5_7/ETH_TXCLK/TA3_I7/TA3_O7/INTP27/CSI0F_CS5	Hi-z
A18	P5_6/ETH_TXEN/TA3_I6/TA3_O6/INTP26/CSI0F_CS4/$\overline{\text{P\_SDCAS}}$	Hi-z
A19	P5_4/ETH_TXD1/TA3_I4/TA3_O4/INTP24/CSI0F_CS2/$\overline{\text{P\_REFRQ}}$	Hi-z
A20	P5_2/ETH_TXD3/TA3_I2/TA3_O2/INTP22/CSI0F_CS0/P_ULDQM	Hi-z
A21	P5_0/ETH_CRS/TA3_I0/TA3_O0/INTP20/P_LLDQM/CSI0F_RYI/CSI0F_RYO	Hi-z
B1	V_{SS}	–
B2	P7_2/S_D2/TA2_I2/TA2_O2	Hi-z
B3	P7_4/S_D4/TA2_I4/TA2_O4	Hi-z
B4	P7_6/S_D6/TA2_I6/TA2_O6	Hi-z
B5	P7_8/S_D8/TA2_I8/TA2_O8	Hi-z
B6	P7_10/S_D10/TA2_I10/TA2_O10	Hi-z
B7	EV_{DD}	–
B8	V_{SS}	–
B9	P8_3/S_D19/TA0_I3/TA0_O3	Hi-z
B10	P8_7/S_D23/TA0_I7/TA0_O7	Hi-z
B11	P8_10/S_D26/TA0_I10/TA0_O10	Hi-z
B12	V_{SS}	–
B13	UV_{DD}	–
B14	P6_0/ETH_MDC/$\overline{\text{DMATC0}}$/TJ_I2/TJ_O2/P_A24	Hi-z
B15	P5_12/ETH_RXD0/TA3_I12/TA3_O12/$\overline{\text{P\_BCYST}}$	Hi-z
B16	EV_{DD}	–
B17	V_{SS}	–
B18	P5_5/ETH_TXD0/TA3_I5/TA3_O5/INTP25/CSI0F_CS3/$\overline{\text{P\_SDRAS}}$	Hi-z
B19	P5_3/ETH_TXD2/TA3_I3/TA3_O3/INTP23/CSI0F_CS1/P_UUDQM	Hi-z
B20	P5_1/ETH_COL/TA3_I1/TA3_O1/INTP21/CSI0F_SSI/P_LUDQM	Hi-z

Table 2-7 Pin State After Reset (2/8)

Pin No.	Pin Name	After Reset
B21	P4_12/P_HLDAK/INTP23/DMAAK2/SO0F	Hi-z
B22	P4_13/P_HLDRQ/DMATC2/SI0F/RXD0F/SDA0	Hi-z
C1	P9_1/S_BUSCLK/CSI2F_CS1	Output (f_X)
C2	P7_0/S_D0/TA2_I0/TA2_O0	Hi-z
C21	P4_10/P_CS3/INTP22/DMATC5	Hi-z
C22	P4_11/P_WAIT/SCK0F/TXD0F/SCL0	Hi-z
D1	P9_4/S_LLDQM/TA3_I3/TA3_O3/CSI2F_CS4	Hi-z
D2	P9_3/S_SDRAS/TA3_I2/TA3_O2/CSI2F_CS3	Hi-z
D4	P9_0/S_SDCKE/TA3_I0/TA3_O0/INTP10/CSI2F_CS0	Hi-z
D5	P7_11/S_D11/TA2_I11/TA2_O11	Hi-z
D6	P7_13/S_D13/TA2_I13/TA2_O13	Hi-z
D7	P8_0/S_D16/TA0_I0/TA0_O0/TE_TI0	Hi-z
D8	P8_1/S_D17/TA0_I1/TA0_O1	Hi-z
D9	P8_4/S_D20/TA0_I4/TA0_O4/TE0_AI	Hi-z
D10	P8_8/S_D24/TA0_I8/TA0_O8/TE0_ZI	Hi-z
D11	P8_11/S_D27/TA0_I11/TA0_O11	Hi-z
D12	P8_13/S_D29/TA0_I13/TA0_O13	Hi-z
D13	FLMD0	Input
D14	P5_15/ETH_RXD3/TA3_I15/TA3_O15/DMAAK1/TJ_I1/TJ_O1	Hi-z
D15	P5_13/ETH_RXD1/TA3_I13/TA3_O13	Hi-z
D16	P5_11/ETH_RXDV/TA3_I11/TA3_O11/P_SDWE	Hi-z
D17	P5_8/ETH_TXER/TA3_I8/TA3_O8/CSI0F_CS6/P_SDCKE	Hi-z
D18	P5_9/ETH_RXER/TA3_I9/TA3_O9/CSI0F_CS7/P_CS4	Hi-z
D19	MODE2	Input
D21	P4_8/P_CS1/P_BCYST/INTP20	Hi-z
D22	P4_9/P_CS2/INTP21/DMAAK5	Hi-z
E1	P9_7/S_UUDQM/S_DMATC3/ESO1/CSI2F_CS7	Hi-z
E2	P9_5/S_LUDQM/TA3_I4/TA3_O4/CSI2F_CS5	Hi-z
E4	P9_2/S_SDCAS/TA3_I1/TA3_O1/CSI2F_CS2	Hi-z
E5	P7_15/S_D15/TA2_I15/TA2_O15	Hi-z
E6	MODE3	Input
E7	V_{SS}	–
E8	EV_{DD}	–
E9	P8_5/S_D21/TA0_I5/TA0_O5	Hi-z
E10	P8_9/S_D25/TA0_I9/TA0_O9	Hi-z
E11	P8_12/S_D28/TA0_I12/TA0_O12	Hi-z
E12	P8_14/S_D30/TA0_I14/TA0_O14	Hi-z
E13	P8_15/S_D31/TA0_I15/TA0_O15	Hi-z
E14	P5_14/ETH_RXD2/TA3_I14/TA3_O14/DMAAK0/TJ_I0/TJ_O0	Hi-z
E15	IV_{DD}	–
E16	V_{SS}	–
E17	EV_{DD}	–
E18	V_{SS}	–
E19	V_{SS}	–

Table 2-7 Pin State After Reset (3/8)

Pin No.	Pin Name	After Reset
E21	P4_6/P_A22/INTP11/DMAAK3/SI3F/RXD3F/SDA3	Hi-z
E22	P4_7/P_A23/INTP12/SCK3F/TXD3F/SCL3/ADTRG20	Hi-z
F1	P9_9/S_LUWR/S_DMARQ2/S_DMATC0/SO4	Hi-z
F2	P9_8/S_LLWR/S_DMAAK2/CSI4_RYI/CSI4_RYO	Hi-z
F4	P9_6/S_ULDQM/S_DMATC2/ESO0/CSI2F_CS6	Hi-z
F5	EV_{DD}	–
F6	EV_{DD}	–
F7	V_{SS}	–
F8	EV_{DD}	–
F9	EV_{DD}	–
F10	V_{SS}	–
F13	V_{SS}	–
F14	EV_{DD}	–
F15	IV_{DD}	–
F16	V_{SS}	–
F17	EV_{DD}	–
F18	EV_{DD}	–
F19	P4_3/P_A19/INTP8/DMAAK4/SO3F	Hi-z
F21	P4_4/P_A20/INTP9/DMATC4/SI3/RXD3	Hi-z
F22	P4_5/P_A21/INTP10/DMATC3/SCK3/TXD3/ADTRG10	Hi-z
G1	P9_12/S_RD/TA3_I5/TA3_O5/INTP11	Hi-z
G2	P9_11/S_UUWR/S_DMARQ3/SCK4/TXD4/SCL4/ESO3	Hi-z
G4	P9_10/S_ULWR/S_DMAAK3/SI4/RXD4/SDA4/ESO2	Hi-z
G5	IV_{DD}	–
G6	IV_{DD}	–
G17	V_{SS}	–
G18	V_{SS}	–
G19	P4_0/P_A16/INTP5/CSI3F_RYI/CSI3F_RYO	Hi-z
G21	P4_1/P_A17/INTP6/CSI3F_SSI	Hi-z
G22	P4_2/P_A18/INTP7/SO3	Hi-z
H1	P9_14/S_LLBE/TA3_I7/TA3_O7/INTP13/CSI1_RYI/CSI1_RYO	Hi-z
H2	P9_13/S_WR/TA3_I6/TA3_O6/INTP12/CSI1_SSI	Hi-z
H4	P9_15/S_LUBE/INTP14/SO1	Hi-z
H5	V_{SS}	–
H6	V_{SS}	–
H17	IV_{DD}	–
H18	IV_{DD}	–
H19	P3_13/P_A13/TA1_I13/TA1_O13/CSI3F_CS7	Hi-z
H21	P3_14/P_A14/TA1_I14/TA1_O14/CSI3_RYI/CSI3_RYO	Hi-z
H22	P3_15/P_A15/TA1_I15/TA1_O15/CSI3_SSI	Hi-z
J1	P10_2/S_SDWE/TA3_I8/TA3_O8/INTP17/CSI2F_SSI	Hi-z
J2	P10_0/S_ULBE/INTP15/SI1/RXD1/S_DMATC0	Hi-z
J4	P10_1/S_UUBE/INTP16/SCK1/TXD1/S_DMATC1	Hi-z
J5	EV_{DD}	–

Table 2-7 Pin State After Reset (4/8)

Pin No.	Pin Name	After Reset
J6	EV_{DD}	–
J17	EV_{DD}	–
J18	EV_{DD}	–
J19	P3_10/P_A10/TA1_I10/TA1_O10/CSI3F_CS4	Hi-z
J21	P3_11/P_A11/TA1_I11/TA1_O11/CSI3F_CS5	Hi-z
J22	P3_12/P_A12/TA1_I12/TA1_O12/CSI3F_CS6	Hi-z
K1	P11_0/S_A1/TA1_I0/TA1_O0/TE1_TI0	Hi-z
K2	P10_3/$\overline{S\_BCYST}$/TA3_I9/TA3_O9/INTP18/$\overline{S\_DMATC1}$/CSI4_SSI	Hi-z
K4	P11_1/S_A2/TA1_I1/TA1_O1	Hi-z
K5	V_{SS}	–
K6	V_{SS}	–
K17	V_{SS}	–
K18	P3_4/P_A4/TA1_I4/TA1_O4/TE1_AI	Hi-z
K19	P3_7/P_A7/TA1_I7/TA1_O7/CSI3F_CS1	Hi-z
K21	P3_8/P_A8/TA1_I8/TA1_O8/CSI3F_CS2/TE1_ZI	Hi-z
K22	P3_9/P_A9/TA1_I9/TA1_O9/CSI3F_CS3	Hi-z
L1	P11_3/S_A4/TA1_I3/TA1_O3	Hi-z
L2	P11_2/S_A3/TA1_I2/TA1_O2/TE1_TI1	Hi-z
L4	P11_5/S_A6/TA1_I5/TA1_O5	Hi-z
L5	P11_4/S_A5/TA1_I4/TA1_O4/TE1_AI	Hi-z
L18	P3_3/P_A3/TA1_I3/TA1_O3	Hi-z
L19	P3_2/P_A2/TA1_I2/TA1_O2/TE1_TI1	Hi-z
L21	P3_5/P_A5/TA1_I5/TA1_O5	Hi-z
L22	P3_6/P_A6/TA1_I6/TA1_O6/CSI3F_CS0/TE1_BI	Hi-z
M1	P11_7/S_A8/TA1_I7/TA1_O7	Hi-z
M2	P11_6/S_A7/TA1_I6/TA1_O6/TE1_BI	Hi-z
M4	P11_12/S_A13/TA1_I12/TA1_O12/CSI1F_CS2	Hi-z
M5	P11_13/S_A14/TA1_I13/TA1_O13/CSI1F_CS3	Hi-z
M18	P2_7/$\overline{P\_WR}$/$\overline{P\_RW}$/NTP19	Hi-z
M19	P3_0/P_A0/TA1_I0/TA1_O0/TE1_TI0/INTP18	Hi-z
M21	P3_1/P_A1/TA1_I1/TA1_O1	Hi-z
M22	V_{SS}	–
N1	P11_9/S_A10/TA1_I9/TA1_O9	Hi-z
N2	P11_8/S_A9/TA1_I8/TA1_O8/TE1_ZI	Hi-z
N4	P12_1/S_A18/INTP1/CSI1F_CS7/ADCNV1	Hi-z
N5	P12_2/S_A19/INTP2/ADCNV2/CSI0_SSI	Hi-z
N6	IV_{DD}	–
N17	EV_{DD}	–
N18	P2_3/$\overline{P\_ULBE}$/$\overline{P\_ULWR}$/INTP15/TJ_I2/TJ_O2	Hi-z
N19	P2_4/$\overline{P\_UUBE}$/$\overline{P\_UUWR}$/INTP16/TJ_I3/TJ_O3	Hi-z
N21	P2_5/$\overline{P\_RD}$/INTP17	Hi-z
N22	P2_6/P_BUSCLK	Outpu (f_X)
P1	P11_11/S_A12/TA1_I11/TA1_O11/CSI1F_CS1	Hi-z
P2	P11_10/S_A11/TA1_I10/TA1_O10/CSI1F_CS0	Hi-z

Table 2-7 Pin State After Reset (5/8)

Pin No.	Pin Name	After Reset
P4	P12_7/S_A24/INTP7/SO2F	Hi-z
P5	P12_8/S_A25/INTP8/SI2F/RXD2F/SDA2	Hi-z
P6	V_{SS}	–
P17	V_{SS}	–
P18	P0_1/P_D1/TA0_I1/TA0_O1	Hi-z
P19	P0_0/P_D0/TA0_I0/TA0_O0/TE0_TI0	Hi-z
P21	P2_1/$\overline{\text{P\_LLBE}}$/$\overline{\text{P\_LLWR}}$/INTP13/TJ_I0/TJ_O0	Hi-z
P22	P2_2/$\overline{\text{P\_LUBE}}$/$\overline{\text{P\_LUWR}}$/INTP14/TJ_I1/TJ_O1	Hi-z
R1	P11_15/S_A16/TA1_I15/TA1_O15/CSI1F_CS5	Hi-z
R2	P11_14/S_A15/TA1_I14/TA1_O14/CSI1F_CS4	Hi-z
R4	P10_7/$\overline{\text{S\_CS3}}$/$\overline{\text{S\_SDCS}}$/INTP26/CSI1F_RYI/CSI1F_RYO	Hi-z
R5	V_{SS}	–
R6	V_{SS}	–
R17	V_{SS}	–
R18	P0_5/P_D5/TA0_I5/TA0_O5	Hi-z
R19	P0_4/P_D4/TA0_I4/TA0_O4/TE0_AI	Hi-z
R21	P0_3/P_D3/TA0_I3/TA0_O3	Hi-z
R22	P0_2/P_D2/TA0_I2/TA0_O2/TE0_TI1	Hi-z
T1	P12_3/S_A20/INTP3/CSI0_RYI/CSI0_RYO	Hi-z
T2	P12_0/S_A17/INTP0/CSI1F_CS6/ADCNV0	Hi-z
T4	P10_11/$\overline{\text{S\_REFRQ}}$/TA3_I15/TA3_O15/CSI2F_RYI/CSI2F_RYO	Hi-z
T5	EV_{DD}	–
T6	EV_{DD}	–
T17	IV_{DD}	–
T18	P0_9/P_D9/TA0_I9/TA0_O9	Hi-z
T19	P0_8/P_D8/TA0_I8/TA0_O8/TE0_ZI	Hi-z
T21	P0_7/P_D7/TA0_I7/TA0_O7	Hi-z
T22	P0_6/P_D6/TA0_I6/TA0_O6/TE0_BI	Hi-z
U1	P12_5/S_A22/INTP5/SI0/RXD0	Hi-z
U2	P12_4/S_A21/INTP4/SO0	Hi-z
U4	P10_10/$\overline{\text{S\_HLDRQ}}$/TA3_I14/TA3_O14/SCK2/TXD2	Hi-z
U5	P10_9/$\overline{\text{S\_HLDAK}}$/TA3_I13/TA3_O13/SO2/INTP27	Hi-z
U6	IV_{DD}	–
U7	V_{SS}	–
U8	EV_{DD}	–
U9	V_{SS}	–
U10	DV_{DD}	–
U13	V_{SS}	–
U14	$PLLV_{DD}$	–
U15	$PLLV_{SS}$	–
U16	IV_{DD}	–
U17	EV_{DD}	–
U18	V_{SS}	–
U19	P0_12/P_D12/TA0_I12/TA0_O12	Hi-z

Table 2-7 Pin State After Reset (6/8)

Pin No.	Pin Name	After Reset
U21	P0_11/P_D11/TA0_I11/TA0_O11	Hi-z
U22	P0_10/P_D10/TA0_I10/TA0_O10	Hi-z
V1	P12_6/S_A23/INTP6/SCK0/TXD0/ADTRG11	Hi-z
V2	P12_9/S_A26/INTP9/SCK2F/TXD2F/SCL2/ADTRG21	Hi-z
V4	P10_8/$\overline{\text{S\_WAIT}}$/SI2/RXD2	Hi-z
V5	P14_0/ANI06	Hi-z
V6	P14_4/ANI10	Hi-z
V7	ANI04	Hi-z
V8	ANI00	Hi-z
V9	MDO6	Output (undefined)
V10	MDO2	Output (undefined)
V11	EVTO	Output (undefined)
V12	DV_{DD}	–
V13	V_{SS}	–
V14	$\overline{\text{TRST}}$	Input
V15	V_{SS}	–
V16	IV_{DD}	–
V17	EV_{DD}	–
V18	V_{SS}	–
V19	V_{SS}	–
V21	P0_14/P_D14/TA0_I14/TA0_O14	Hi-z
V22	P0_13/P_D13/TA0_I13/TA0_O13	Hi-z
W1	P10_4/$\overline{\text{S\_CS0}}$/TA3_I10/TA3_O10/INTP19/CSI2_SSI	Hi-z
W2	P10_5/$\overline{\text{S\_CS1}}$/TA3_I11/TA3_O11/INTP24/CSI2_RYI/CSI2_RYO	Hi-z
W4	P13_1/$\overline{\text{S\_CS3}}$/$\overline{\text{S\_DMAAK0}}$/ADTRG00/INTP0/CSI5_RYI/CSI5_RYO	Hi-z
W5	P14_1/ANI07	Hi-z
W6	P14_5/ANI11	Hi-z
W7	ANI05	Hi-z
W8	ANI01	Hi-z
W9	MDO7	Output (undefined)
W10	MDO3	Output (undefined)
W11	EVTI	Hi-z
W12	MSEO0	Output (undefined)
W13	TDO/FLSO	Hi-z
W14	TDI/FLRXD/FLSI	Hi-z
W15	TMS	Hi-z
W16	P2_0/NMI	Hi-z
W17	P1_13/P_D29/TA2_I13/TA2_O13	Hi-z
W18	EV_{DD}	–
W19	P1_6/P_D22/TA2_I6/TA2_O6/ESO3	Hi-z

Table 2-7 Pin State After Reset (7/8)

Pin No.	Pin Name	After Reset
W21	P1_0/P_D16/TA2_I0/TA2_O0/ADCNV0	Hi-z
W22	P0_15/P_D15/TA0_I15/TA0_O15	Hi-z
Y1	P10_6/$\overline{\text{S\_CS2}}$/TA3_I12/TA3_O12/INTP25/CSI1F_SSI	Hi-z
Y2	P13_7/CAN1RXD/SI5/RXD5/SDA5	Hi-z
Y21	P1_2/P_D18/TA2_I2/TA2_O2/ADCNV2	Hi-z
Y22	P1_1/P_D17/TA2_I1/TA2_O1/ADCNV1	Hi-z
AA1	P13_6/CAN1TXD/SCK5/TXD5/SCL5/INTP4	Hi-z
AA2	P13_5/CAN0RXD/SI1F/RXD1F/SDA1	Hi-z
AA3	P13_3/$\overline{\text{S\_DMAAK1}}$/SO5/INTP2/$\overline{\text{OCI}}$	Hi-z
AA4	V_{SS}	–
AA5	P14_2/ANI08	Hi-z
AA6	AV_{REFM}	–
AA7	AV_{REFP}	–
AA8	ANI02	Hi-z
AA9	IV_{DD}	–
AA10	MDO4	Output (undefined)
AA11	MDO0	Output (undefined)
AA12	MSEO1	Output (undefined)
AA13	TRDY	Hi-z
AA14	$OSCV_{DD}$	–
AA15	X2	–
AA16	FLMD1	Input
AA17	P1_14/P_D30/TA2_I14/TA2_O14	Hi-z
AA18	P1_11/P_D27/TA2_I11/TA2_O11	Hi-z
AA19	P1_9/P_D25/TA2_I9/TA2_O9	Hi-z
AA20	P1_7/P_D23/TA2_I7/TA2_O7	Hi-z
AA21	P1_4/P_D20/TA2_I4/TA2_O4/ESO1	Hi-z
AA22	P1_3/P_D19/TA2_I3/TA2_O3/ESO0	Hi-z
AB2	P13_4/CAN0TXD/SCK1F/TXD1F/SCL1/INTP3	Hi-z
AB3	P13_0/$\overline{\text{S\_DMARQ0}}$/ADTRG01/UCLK/CSI5_SSI	Hi-z
AB4	P13_2/$\overline{\text{S\_DMARQ1}}$/SO1F/INTP1/PPON	Hi-z
AB5	P14_3/ANI09	Hi-z
AB6	AV_{SS}	–
AB7	AV_{DD}	–
AB8	ANI03	Hi-z
AB9	V_{SS}	–
AB10	MDO5	Output (undefined)
AB11	MDO1	Output (undefined)
AB12	MCKO	Output (undefined)
AB13	TCK/FLSCK	Hi-z

Table 2-7 Pin State After Reset (8/8)

Pin No.	Pin Name	After Reset
AB14	$OSCV_{SS}$	–
AB15	X1	Input
AB16	$\overline{RESET}$	Input
AB17	P1_15/P_D31/TA2_I15/TA2_O15	Hi-z
AB18	P1_12/P_D28/TA2_I12/TA2_O12	Hi-z
AB19	P1_10/P_D26/TA2_I10/TA2_O10	Hi-z
AB20	P1_8/P_D24/TA2_I8/TA2_O8	Hi-z
AB21	P1_5/P_D21/TA2_I5/TA2_O5/ESO2	Hi-z

2.5 Default States and Recommended Connection of Unused Pins

Table 2-8 Default States and Recommended Connection of Unused Pins (1/9)

Pin No.	Pin Name	Default State		Recommended Connection of Unused Pins
		Function	I/O	
A2	P7_1/S_D1/TA2_I1/TA2_O1	P7_1	Input	Independently connect to EV_{DD} or V_{SS} via a resistor.
A3	P7_3/S_D3/TA2_I3/TA2_O3	P7_3	Input	
A4	P7_5/S_D5/TA2_I5/TA2_O5	P7_5	Input	
A5	P7_7/S_D7/TA2_I7/TA2_O7	P7_7	Input	
A6	P7_9/S_D9/TA2_I9/TA2_O9	P7_9	Input	
A7	P7_12/S_D12/TA2_I12/TA2_O12	P7_12	Input	
A8	P7_14/S_D14/TA2_I14/TA2_O14	P7_14	Input	
A9	P8_2/S_D18/TA0_I2/TA0_O2/TE0_TI1	P8_2	Input	
A10	P8_6/S_D22/TA0_I6/TA0_O6/TE0_BI	P8_6	Input	
A11	UDPH	UDPH	Output	Leave open.
A12	UDMH	UDMH	Output	Leave open.
A13	UDPF	UDPF	Input	Independently connect to V_{SS} via a resistor.
A14	UDMF	UDMF	Input	Independently connect to V_{SS} via a resistor.
A15	P6_1/ETH_MDIO/$\overline{\text{DMATC1}}$/TJ_I3/TJ_O3/P_A25	P6_1	Input	Independently connect to EV_{DD} or V_{SS} via a resistor.
A16	P5_10/ETH_RXCLK/TA3_I10/TA3_O10/P_BUSRQ	P5_10	Input	
A17	P5_7/ETH_TXCLK/TA3_I7/TA3_O7/INTP27/ CSI0F_CS5	P5_7	Input	
A18	P5_6/ETH_TXEN/TA3_I6/TA3_O6/INTP26/ CSI0F_CS4/$\overline{\text{P\_SDCAS}}$	P5_6	Input	
A19	P5_4/ETH_TXD1/TA3_I4/TA3_O4/INTP24/ CSI0F_CS2/$\overline{\text{P\_REFRQ}}$	P5_4	Input	
A20	P5_2/ETH_TXD3/TA3_I2/TA3_O2/INTP22/ CSI0F_CS0/P_ULDQM	P5_2	Input	
A21	P5_0/ETH_CRS/TA3_I0/TA3_O0/INTP20/ P_LLDQM/CSI0F_RYI/CSI0F_RYO	P5_0	Input	
B1	V_{SS}	V_{SS}	–	Always connect to the ground directly.
B2	P7_2/S_D2/TA2_I2/TA2_O2	P7_2	Input	Independently connect to EV_{DD} or V_{SS} via a resistor.
B3	P7_4/S_D4/TA2_I4/TA2_O4	P7_4	Input	
B4	P7_6/S_D6/TA2_I6/TA2_O6	P7_6	Input	
B5	P7_8/S_D8/TA2_I8/TA2_O8	P7_8	Input	
B6	P7_10/S_D10/TA2_I10/TA2_O10	P7_10	Input	
B7	EV_{DD}	EV_{DD}	–	Always connect to the power supply.
B8	V_{SS}	V_{SS}	–	Always connect to the ground directly.
B9	P8_3/S_D19/TA0_I3/TA0_O3	P8_3	Input	Independently connect to EV_{DD} or V_{SS} via a resistor.
B10	P8_7/S_D23/TA0_I7/TA0_O7	P8_7	Input	
B11	P8_10/S_D26/TA0_I10/TA0_O10	P8_10	Input	

Table 2-8 Default States and Recommended Connection of Unused Pins (2/9)

Pin No.	Pin Name	Default State		Recommended Connection of Unused Pins
		Function	I/O	
B12	V_{SS}	V_{SS}	–	Always connect to the ground directly.
B13	UV_{DD}	UV_{DD}	–	Always connect to the power supply.
B14	P6_0/ETH_MDC/$\overline{\text{DMATC0}}$/TJ_I2/TJ_O2/P_A24	P6_0	Input	Independently connect to EV_{DD} or V_{SS} via a resistor.
B15	P5_12/ETH_RXD0/TA3_I12/TA3_O12/$\overline{\text{P\_BCYST}}$	P5_12	Input	
B16	EV_{DD}	EV_{DD}	–	Always connect to the power supply.
B17	V_{SS}	V_{SS}	–	Always connect to the ground directly.
B18	P5_5/ETH_TXD0/TA3_I5/TA3_O5/INTP25/ CSI0F_CS3/$\overline{\text{P\_SDRAS}}$	P5_5	Input	Independently connect to EV_{DD} or V_{SS} via a resistor.
B19	P5_3/ETH_TXD2/TA3_I3/TA3_O3/INTP23/ CSI0F_CS1/P_UUDQM	P5_3	Input	
B20	P5_1/ETH_COL/TA3_I1/TA3_O1/INTP21/ CSI0F_SSI/P_LUDQM	P5_1	Input	
B21	P4_12/$\overline{\text{P\_HLDAK}}$/INTP23/$\overline{\text{DMAAK2}}$/SO0F	P4_12	Input	
B22	P4_13/$\overline{\text{P\_HLDRQ}}$/$\overline{\text{DMATC2}}$/SI0F/RXD0F/SDA0	P4_13	Input	
C1	P9_1/S_BUSCLK/CSI2F_CS1	S_BUSCLK	Output	Leave open.
C2	P7_0/S_D0/TA2_I0/TA2_O0	P7_0	Input	Independently connect to EV_{DD} or V_{SS} via a resistor.
C21	P4_10/$\overline{\text{P\_CS3}}$/INTP22/$\overline{\text{DMATC5}}$	P4_10	Input	
C22	P4_11/$\overline{\text{P\_WAIT}}$/SCK0F/TXD0F/SCL0	P4_11	Input	
D1	P9_4/S_LLDQM/TA3_I3/TA3_O3/CSI2F_CS4	P9_4	Input	
D2	P9_3/$\overline{\text{S\_SDRAS}}$/TA3_I2/TA3_O2/CSI2F_CS3	P9_3	Input	
D4	P9_0/S_SDCKE/TA3_I0/TA3_O0/INTP10/ CSI2F_CS0	P9_0	Input	
D5	P7_11/S_D11/TA2_I11/TA2_O11	P7_11	Input	
D6	P7_13/S_D13/TA2_I13/TA2_O13	P7_13	Input	
D7	P8_0/S_D16/TA0_I0/TA0_O0/TE_TI0	P8_0	Input	
D8	P8_1/S_D17/TA0_I1/TA0_O1	P8_1	Input	
D9	P8_4/S_D20/TA0_I4/TA0_O4/TE0_AI	P8_4	Input	
D10	P8_8/S_D24/TA0_I8/TA0_O8/TE0_ZI	P8_8	Input	
D11	P8_11/S_D27/TA0_I11/TA0_O11	P8_11	Input	
D12	P8_13/S_D29/TA0_I13/TA0_O13	P8_13	Input	
D13	FLMD0	FLMD0	Input	Connect to V_{SS} directly if the FLMD0 signal is always low. Independently connect to V_{SS} via a resistor (100 kΩ is recommended) when in flash memory programming mode, boundary scan mode, or flash memory self-programming mode.
D14	P5_15/ETH_RXD3/TA3_I15/TA3_O15/$\overline{\text{DMAAK1}}$/ TJ_I1/TJ_O1	P5_15	Input	Independently connect to EV_{DD} or V_{SS} via a resistor.
D15	P5_13/ETH_RXD1/TA3_I13/TA3_O13	P5_13	Input	
D16	P5_11/ETH_RXDV/TA3_I11/TA3_O11/$\overline{\text{P\_SDWE}}$	P5_11	Input	

Table 2-8 Default States and Recommended Connection of Unused Pins (3/9)

Pin No.	Pin Name	Default State		Recommended Connection of Unused Pins
		Function	I/O	
D17	P5_8/ETH_TXER/TA3_I8/TA3_O8/CSI0F_CS6/ P_SDCKE	P5_8	Input	Independently connect to EV_{DD} or V_{SS} via a resistor.
D18	P5_9/ETH_RXER/TA3_I9/TA3_O9/CSI0F_CS7/ $\overline{P\_CS4}$	P5_9	Input	
D19	MODE2	MODE2	Input	Independently connect to V_{SS} via a resistor.
D21	P4_8/$\overline{P\_CS1}$/$\overline{P\_BCYST}$/INTP20	P4_8	Input	Independently connect to EV_{DD} or V_{SS} via a resistor.
D22	P4_9/$\overline{P\_CS2}$/INTP21/$\overline{DMAAK5}$	P4_9	Input	
E1	P9_7/S_UUDQM/$\overline{S\_DMATC3}$/ESO1/CSI2F_CS7	P9_7	Input	
E2	P9_5/S_LUDQM/TA3_I4/TA3_O4/CSI2F_CS5	P9_5	Input	
E4	P9_2/$\overline{S\_SDCAS}$/TA3_I1/TA3_O1/CSI2F_CS2	P9_2	Input	
E5	P7_15/S_D15/TA2_I15/TA2_O15	P7_15	Input	
E6	MODE3	MODE3	Input	Independently connect to V_{SS} via a resistor.
E7	V_{SS}	V_{SS}	–	Always connect to the ground directly.
E8	EV_{DD}	EV_{DD}	–	Always connect to the power supply.
E9	P8_5/S_D21/TA0_I5/TA0_O5	P8_5	Input	Independently connect to EV_{DD} or V_{SS} via a resistor.
E10	P8_9/S_D25/TA0_I9/TA0_O9	P8_9	Input	
E11	P8_12/S_D28/TA0_I12/TA0_O12	P8_12	Input	
E12	P8_14/S_D30/TA0_I14/TA0_O14	P8_14	Input	
E13	P8_15/S_D31/TA0_I15/TA0_O15	P8_15	Input	
E14	P5_14/ETH_RXD2/TA3_I14/TA3_O14/$\overline{DMAAK0}$/ TJ_I0/TJ_O0	P5_14	Input	
E15	IV_{DD}	IV_{DD}	–	Always connect to the power supply.
E16	V_{SS}	V_{SS}	–	Always connect to the ground directly.
E17	EV_{DD}	EV_{DD}	–	Always connect to the power supply.
E18	V_{SS}	V_{SS}	–	Always connect to the ground directly.
E19	V_{SS}	V_{SS}	–	
E21	P4_6/P_A22/INTP11/$\overline{DMAAK3}$/SI3F/RXD3F/SDA3	P4_6	Input	Independently connect to EV_{DD} or V_{SS} via a resistor.
E22	P4_7/P_A23/INTP12/SCK3F/TXD3F/SCL3/ ADTRG20	P4_7	Input	
F1	P9_9/$\overline{S\_LUWR}$/$\overline{S\_DMARQ2}$/$\overline{S\_DMATC0}$/SO4	P9_9	Input	
F2	P9_8/$\overline{S\_LLWR}$/$\overline{S\_DMAAK2}$/CSI4_RYI/CSI4_RYO	P9_8	Input	
F4	P9_6/S_ULDQM/$\overline{S\_DMATC2}$/ESO0/CSI2F_CS6	P9_6	Input	
F5	EV_{DD}	EV_{DD}	–	Always connect to the power supply.
F6	EV_{DD}	EV_{DD}	–	
F7	V_{SS}	V_{SS}	–	Always connect to the ground directly.
F8	EV_{DD}	EV_{DD}	–	Always connect to the power supply.
F9	EV_{DD}	EV_{DD}	–	

Table 2-8 Default States and Recommended Connection of Unused Pins (4/9)

Pin No.	Pin Name	Default State		Recommended Connection of Unused Pins
		Function	I/O	
F10	V_{SS}	V_{SS}	–	Always connect to the ground directly.
F13	V_{SS}	V_{SS}	–	
F14	EV_{DD}	EV_{DD}	–	Always connect to the power supply.
F15	IV_{DD}	IV_{DD}	–	
F16	V_{SS}	V_{SS}	–	Always connect to the ground directly.
F17	EV_{DD}	EV_{DD}	–	Always connect to the power supply.
F18	EV_{DD}	EV_{DD}	–	
F19	P4_3/P_A19/INTP8/$\overline{\text{DMAAK4}}$/SO3F	P4_3	Input	Independently connect to EV_{DD} or V_{SS} via a resistor.
F21	P4_4/P_A20/INTP9/$\overline{\text{DMATC4}}$/SI3/RXD3	P4_4	Input	
F22	P4_5/P_A21/INTP10/$\overline{\text{DMATC3}}$/SCK3/TXD3/ ADTRG10	P4_5	Input	
G1	P9_12/$\overline{\text{S\_RD}}$/TA3_I5/TA3_O5/INTP11	P9_12	Input	
G2	P9_11/$\overline{\text{S\_UUWR}}$/$\overline{\text{S\_DMARQ3}}$/SCK4/TXD4/SCL4/ ESO3	P9_11	Input	
G4	P9_10/$\overline{\text{S\_ULWR}}$/$\overline{\text{S\_DMAAK3}}$/SI4/RXD4/SDA4/ ESO2	P9_10	Input	
G5	IV_{DD}	IV_{DD}	–	Always connect to the power supply.
G6	IV_{DD}	IV_{DD}	–	
G17	V_{SS}	V_{SS}	–	Always connect to the ground directly.
G18	V_{SS}	V_{SS}	–	
G19	P4_0/P_A16/INTP5/CSI3F_RYI/CSI3F_RYO	P4_0	Input	Independently connect to EV_{DD} or V_{SS} via a resistor.
G21	P4_1/P_A17/INTP6/CSI3F_SSI	P4_1	Input	
G22	P4_2/P_A18/INTP7/SO3	P4_2	Input	
H1	P9_14/$\overline{\text{S\_LLBE}}$/TA3_I7/TA3_O7/INTP13/ CSI1_RYI/CSI1_RYO	P9_14	Input	
H2	P9_13/$\overline{\text{S\_WR}}$/TA3_I6/TA3_O6/INTP12/CSI1_SSI	P9_13	Input	
H4	P9_15/$\overline{\text{S\_LUBE}}$/INTP14/SO1	P9_15	Input	
H5	V_{SS}	V_{SS}	–	Always connect to the ground directly.
H6	V_{SS}	V_{SS}	–	
H17	IV_{DD}	IV_{DD}	–	Always connect to the power supply.
H18	IV_{DD}	IV_{DD}	–	
H19	P3_13/P_A13/TA1_I13/TA1_O13/CSI3F_CS7	P3_13	Input	Independently connect to EV_{DD} or V_{SS} via a resistor.
H21	P3_14/P_A14/TA1_I14/TA1_O14/CSI3_RYI/ CSI3_RYO	P3_14	Input	
H22	P3_15/P_A15/TA1_I15/TA1_O15/CSI3_SSI	P3_15	Input	
J1	P10_2/$\overline{\text{S\_SDWE}}$/TA3_I8/TA3_O8/INTP17/ CSI2F_SSI	P10_2	Input	
J2	P10_0/$\overline{\text{S\_ULBE}}$/INTP15/SI1/RXD1/$\overline{\text{S\_DMATC0}}$	P10_0	Input	
J4	P10_1/$\overline{\text{S\_UUBE}}$/INTP16/SCK1/TXD1/$\overline{\text{S\_DMATC1}}$	P10_1	Input	
J5	EV_{DD}	EV_{DD}	–	Always connect to the power supply.
J6	EV_{DD}	EV_{DD}	–	
J17	EV_{DD}	EV_{DD}	–	
J18	EV_{DD}	EV_{DD}	–	

Table 2-8 Default States and Recommended Connection of Unused Pins (5/9)

Pin No.	Pin Name	Default State		Recommended Connection of Unused Pins
		Function	I/O	
J19	P3_10/P_A10/TA1_I10/TA1_O10/CSI3F_CS4	P3_10	Input	Independently connect to EV_{DD} or V_{SS} via a resistor.
J21	P3_11/P_A11/TA1_I11/TA1_O11/CSI3F_CS5	P3_11	Input	
J22	P3_12/P_A12/TA1_I12/TA1_O12/CSI3F_CS6	P3_12	Input	
K1	P11_0/S_A1/TA1_I0/TA1_O0/TE1_TI0	P11_0	Input	
K2	P10_3/$\overline{\text{S\_BCYST}}$/TA3_I9/TA3_O9/INTP18/ $\overline{\text{S\_DMATC1}}$/CSI4_SSI	P10_3	Input	
K4	P11_1/S_A2/TA1_I1/TA1_O1	P11_1	Input	
K5	V_{SS}	V_{SS}	–	Always connect to the ground directly.
K6	V_{SS}	V_{SS}	–	
K17	V_{SS}	V_{SS}	–	
K18	P3_4/P_A4/TA1_I4/TA1_O4/TE1_AI	P3_4	Input	Independently connect to EV_{DD} or V_{SS} via a resistor.
K19	P3_7/P_A7/TA1_I7/TA1_O7/CSI3F_CS1	P3_7	Input	
K21	P3_8/P_A8/TA1_I8/TA1_O8/CSI3F_CS2/TE1_ZI	P3_8	Input	
K22	P3_9/P_A9/TA1_I9/TA1_O9/CSI3F_CS3	P3_9	Input	
L1	P11_3/S_A4/TA1_I3/TA1_O3	P11_3	Input	
L2	P11_2/S_A3/TA1_I2/TA1_O2/TE1_TI1	P11_2	Input	
L4	P11_5/S_A6/TA1_I5/TA1_O5	P11_5	Input	
L5	P11_4/S_A5/TA1_I4/TA1_O4/TE1_AI	P11_4	Input	
L18	P3_3/P_A3/TA1_I3/TA1_O3	P3_3	Input	
L19	P3_2/P_A2/TA1_I2/TA1_O2/TE1_TI1	P3_2	Input	
L21	P3_5/P_A5/TA1_I5/TA1_O5	P3_5	Input	
L22	P3_6/P_A6/TA1_I6/TA1_O6/CSI3F_CS0/TE1_BI	P3_6	Input	
M1	P11_7/S_A8/TA1_I7/TA1_O7	P11_7	Input	
M2	P11_6/S_A7/TA1_I6/TA1_O6/TE1_BI	P11_6	Input	
M4	P11_12/S_A13/TA1_I12/TA1_O12/CSI1F_CS2	P11_12	Input	
M5	P11_13/S_A14/TA1_I13/TA1_O13/CSI1F_CS3	P11_13	Input	
M18	P2_7/$\overline{\text{P\_WR}}$/$\overline{\text{P\_RW}}$/NTP19	P2_7	Input	
M19	P3_0/P_A0/TA1_I0/TA1_O0/TE1_TI0/INTP18	P3_0	Input	
M21	P3_1/P_A1/TA1_I1/TA1_O1	P3_1	Input	
M22	V_{SS}	V_{SS}	–	Always connect to the ground directly.
N1	P11_9/S_A10/TA1_I9/TA1_O9	P11_9	Input	Independently connect to EV_{DD} or V_{SS} via a resistor.
N2	P11_8/S_A9/TA1_I8/TA1_O8/TE1_ZI	P11_8	Input	
N4	P12_1/S_A18/INTP1/CSI1F_CS7/ADCNV1	P12_1	Input	
N5	P12_2/S_A19/INTP2/ADCNV2/CSI0_SSI	P12_2	Input	
N6	IV_{DD}	IV_{DD}	–	Always connect to the power supply.
N17	EV_{DD}	EV_{DD}	–	
N18	P2_3/$\overline{\text{P\_ULBE}}$/$\overline{\text{P\_ULWR}}$/INTP15/TJ_I2/TJ_O2	P2_3	Input	Independently connect to EV_{DD} or V_{SS} via a resistor.
N19	P2_4/$\overline{\text{P\_UUBE}}$/$\overline{\text{P\_UUWR}}$/INTP16/TJ_I3/TJ_O3	P2_4	Input	
N21	P2_5/$\overline{\text{P\_RD}}$/INTP17	P2_5	Input	
N22	P2_6/P_BUSCLK	P_BUSCLK	Output	Leave open.

Table 2-8 Default States and Recommended Connection of Unused Pins (6/9)

Pin No.	Pin Name	Default State		Recommended Connection of Unused Pins
		Function	I/O	
P1	P11_11/S_A12/TA1_I11/TA1_O11/CSI1F_CS1	P11_11	Input	Independently connect to EV_{DD} or V_{SS} via a resistor.
P2	P11_10/S_A11/TA1_I10/TA1_O10/CSI1F_CS0	P11_10	Input	
P4	P12_7/S_A24/INTP7/SO2F	P12_7	Input	
P5	P12_8/S_A25/INTP8/SI2F/RXD2F/SDA2	P12_8	Input	
P6	V_{SS}	V_{SS}	–	Always connect to the ground directly.
P17	V_{SS}	V_{SS}	–	
P18	P0_1/P_D1/TA0_I1/TA0_O1	P0_1	Input	Independently connect to EV_{DD} or V_{SS} via a resistor.
P19	P0_0/P_D0/TA0_I0/TA0_O0/TE0_TI0	P0_0	Input	
P21	P2_1/$\overline{\text{P\_LLBE}}$/$\overline{\text{P\_LLWR}}$/INTP13/TJ_I0/TJ_O0	P2_1	Input	
P22	P2_2/$\overline{\text{P\_LUBE}}$/$\overline{\text{P\_LUWR}}$/INTP14/TJ_I1/TJ_O1	P2_2	Input	
R1	P11_15/S_A16/TA1_I15/TA1_O15/CSI1F_CS5	P11_15	Input	
R2	P11_14/S_A15/TA1_I14/TA1_O14/CSI1F_CS4	P11_14	Input	
R4	P10_7/$\overline{\text{S\_CS3}}$/$\overline{\text{S\_SDCS}}$/INTP26/CSI1F_RYI/ CSI1F_RYO	P10_7	Input	
R5	V_{SS}	V_{SS}	–	Always connect to the ground directly.
R6	V_{SS}	V_{SS}	–	
R17	V_{SS}	V_{SS}	–	
R18	P0_5/P_D5/TA0_I5/TA0_O5	P0_5	Input	Independently connect to EV_{DD} or V_{SS} via a resistor.
R19	P0_4/P_D4/TA0_I4/TA0_O4/TE0_AI	P0_4	Input	
R21	P0_3/P_D3/TA0_I3/TA0_O3	P0_3	Input	
R22	P0_2/P_D2/TA0_I2/TA0_O2/TE0_TI1	P0_2	Input	
T1	P12_3/S_A20/INTP3/CSI0_RYI/CSI0_RYO	P12_3	Input	
T2	P12_0/S_A17/INTP0/CSI1F_CS6/ADCNV0	P12_0	Input	
T4	P10_11/$\overline{\text{S\_REFRQ}}$/TA3_I15/TA3_O15/CSI2F_RYI/ CSI2F_RYO	P10_11	Input	
T5	EV_{DD}	EV_{DD}	–	Always connect to the power supply.
T6	EV_{DD}	EV_{DD}	–	
T17	IV_{DD}	IV_{DD}	–	
T18	P0_9/P_D9/TA0_I9/TA0_O9	P0_9	Input	Independently connect to EV_{DD} or V_{SS} via a resistor.
T19	P0_8/P_D8/TA0_I8/TA0_O8/TE0_ZI	P0_8	Input	
T21	P0_7/P_D7/TA0_I7/TA0_O7	P0_7	Input	
T22	P0_6/P_D6/TA0_I6/TA0_O6/TE0_BI	P0_6	Input	
U1	P12_5/S_A22/INTP5/SI0/RXD0	P12_5	Input	
U2	P12_4/S_A21/INTP4/SO0	P12_4	Input	
U4	P10_10/$\overline{\text{S\_HLDRQ}}$/TA3_I14/TA3_O14/SCK2/TXD2	P10_10	Input	
U5	P10_9/$\overline{\text{S\_HLDAK}}$/TA3_I13/TA3_O13/SO2/INTP27	P10_9	Input	
U6	IV_{DD}	IV_{DD}	–	Always connect to the power supply.
U7	V_{SS}	V_{SS}	–	
U8	EV_{DD}	EV_{DD}	–	
U9	V_{SS}	V_{SS}	–	Always connect to the ground directly.
U10	DV_{DD}	DV_{DD}	–	Always connect to the power supply.
U13	V_{SS}	V_{SS}	–	Always connect to the ground directly.

Table 2-8 Default States and Recommended Connection of Unused Pins (7/9)

Pin No.	Pin Name	Default State		Recommended Connection of Unused Pins
		Function	I/O	
U14	$PLLV_{DD}$	$PLLV_{DD}$	–	Always connect to the power supply.
U15	$PLLV_{SS}$	$PLLV_{SS}$	–	Always connect to the ground directly.
U16	IV_{DD}	IV_{DD}	–	Always connect to the power supply.
U17	EV_{DD}	EV_{DD}	–	
U18	V_{SS}	V_{SS}	–	Always connect to the ground directly.
U19	P0_12/P_D12/TA0_I12/TA0_O12	P0_12	Input	Independently connect to EV_{DD} or V_{SS} via a resistor.
U21	P0_11/P_D11/TA0_I11/TA0_O11	P0_11	Input	
U22	P0_10/P_D10/TA0_I10/TA0_O10	P0_10	Input	
V1	P12_6/S_A23/INTP6/SCK0/TXD0/ADTRG11	P12_6	Input	
V2	P12_9/S_A26/INTP9/SCK2F/TXD2F/SCL2/ ADTRG21	P12_9	Input	
V4	P10_8/$\overline{\text{S\_WAIT}}$/SI2/RXD2	P10_8	Input	
V5	P14_0/ANI06	P14_0	Input	Independently connect to AV_{DD} or AV_{SS} via a resistor.
V6	P14_4/ANI10	P14_4	Input	
V7	ANI04	ANI04	Input	
V8	ANI00	ANI00	Input	
V9	MDO6	MDO6	Output	Leave open.
V10	MDO2	MDO2	Output	
V11	EVTO	$\overline{\text{EVTO}}$	Output	
V12	DV_{DD}	DV_{DD}	–	Always connect to the power supply.
V13	V_{SS}	V_{SS}	–	Always connect to the ground directly.
V14	$\overline{\text{TRST}}$	$\overline{\text{TRST}}$	Input	Independently connect to V_{SS} via a resistor.
V15	V_{SS}	V_{SS}	–	Always connect to the ground directly.
V16	IV_{DD}	IV_{DD}	–	Always connect to the power supply.
V17	EV_{DD}	EV_{DD}	–	
V18	V_{SS}	V_{SS}	–	Always connect to the ground directly.
V19	V_{SS}	V_{SS}	–	
V21	P0_14/P_D14/TA0_I14/TA0_O14	P0_14	Input	Independently connect to EV_{DD} or V_{SS} via a resistor.
V22	P0_13/P_D13/TA0_I13/TA0_O13	P0_13	Input	
W1	P10_4/$\overline{\text{S\_CS0}}$/TA3_I10/TA3_O10/INTP19/ CSI2_SSI	P10_4	Input	
W2	P10_5/$\overline{\text{S\_CS1}}$/TA3_I11/TA3_O11/INTP24/ CSI2_RYI/CSI2_RYO	P10_5	Input	
W4	P13_1/$\overline{\text{S\_CS3}}$/$\overline{\text{S\_DMAAK0}}$/ADTRG00/INTP0/ CSI5_RYI/CSI5_RYO	P13_1	Input	
W5	P14_1/ANI07	P14_1	Input	Independently connect to AV_{DD} or AV_{SS} via a resistor.
W6	P14_5/ANI11	P14_5	Input	
W7	ANI05	ANI05	Input	
W8	ANI01	ANI01	Input	

Table 2-8 Default States and Recommended Connection of Unused Pins (8/9)

Pin No.	Pin Name	Default State		Recommended Connection of Unused Pins
		Function	I/O	
W9	MDO7	MDO7	Output	Leave open.
W10	MDO3	MDO3	Output	
W11	EVTI	$\overline{\text{EVTI}}$	Input	Independently connect to DV_{DD} via a resistor.
W12	MSEO0	MSEO0	Output	Leave open.
W13	TDO/FLSO	TDO	Output	
W14	TDI/FLRXD/FLSI	TDI	Input	Independently connect to DV_{DD} via a resistor.
W15	TMS	TMS	Input	
W16	P2_0/NMI	P2_0	Input	Independently connect to EV_{DD} or V_{SS} via a resistor.
W17	P1_13/P_D29/TA2_I13/TA2_O13	P1_13	Input	
W18	EV_{DD}	EV_{DD}	–	Always connect to the power supply.
W19	P1_6/P_D22/TA2_I6/TA2_O6/ESO3	P1_6	Input	Independently connect to EV_{DD} or V_{SS} via a resistor.
W21	P1_0/P_D16/TA2_I0/TA2_O0/ADCNV0	P1_0	Input	
W22	P0_15/P_D15/TA0_I15/TA0_O15	P0_15	Input	
Y1	P10_6/$\overline{\text{S\_CS2}}$/TA3_I12/TA3_O12/INTP25/CSI1F_SSI	P10_6	Input	
Y2	P13_7/CAN1RXD/SI5/RXD5/SDA5	P13_7	Input	
Y21	P1_2/P_D18/TA2_I2/TA2_O2/ADCNV2	P1_2	Input	
Y22	P1_1/P_D17/TA2_I1/TA2_O1/ADCNV1	P1_1	Input	
AA1	P13_6/CAN1TXD/SCK5/TXD5/SCL5/INTP4	P13_6	Input	
AA2	P13_5/CAN0RXD/SI1F/RXD1F/SDA1	P13_5	Input	
AA3	P13_3/$\overline{\text{S\_DMAAK1}}$/SO5/INTP2/$\overline{\text{OCI}}$	P13_3	Input	
AA4	V_{SS}	V_{SS}	–	Always connect to the ground directly.
AA5	P14_2/ANI08	P14_2	Input	Independently connect to AV_{DD} or AV_{SS} via a resistor.
AA6	AV_{REFM}	AV_{REFM}	Input	Directly connect to AV_{SS}.
AA7	AV_{REFP}	AV_{REFP}	Input	Directly connect to AV_{DD}.
AA8	ANI02	ANI02	Input	Independently connect to AV_{DD} or AV_{SS} via a resistor.
AA9	IV_{DD}	IV_{DD}	–	Always connect to the power supply.
AA10	MDO4	MDO4	Output	Leave open.
AA11	MDO0	MDO0	Output	
AA12	MSEO1	MSEO1	Output	
AA13	TRDY	$\overline{\text{TRDY}}$	Output	Independently connect to DV_{DD} via a resistor.
AA14	$OSCV_{DD}$	$OSCV_{DD}$	–	Always connect to the power supply.
AA15	X2	X2	–	–
AA16	FLMD1	FLMD1	Input	If not in the boundary scan mode, independently connect to V_{SS} via a resistor.

Table 2-8 Default States and Recommended Connection of Unused Pins (9/9)

Pin No.	Pin Name	Default State		Recommended Connection of Unused Pins
		Function	I/O	
AA17	P1_14/P_D30/TA2_I14/TA2_O14	P1_14	Input	Independently connect to EV_{DD} or V_{SS} via a resistor.
AA18	P1_11/P_D27/TA2_I11/TA2_O11	P1_11	Input	
AA19	P1_9/P_D25/TA2_I9/TA2_O9	P1_9	Input	
AA20	P1_7/P_D23/TA2_I7/TA2_O7	P1_7	Input	
AA21	P1_4/P_D20/TA2_I4/TA2_O4/ESO1	P1_4	Input	
AA22	P1_3/P_D19/TA2_I3/TA2_O3/ESO0	P1_3	Input	
AB2	P13_4/CAN0TXD/SCK1F/TXD1F/SCL1/INTP3	P13_4	Input	
AB3	P13_0/$\overline{S\_DMARQ0}$/ADTRG01/UCLK/CSI5_SSI	P13_0	Input	
AB4	P13_2/$\overline{S\_DMARQ1}$/SO1F/INTP1/PPON	P13_2	Input	
AB5	P14_3/ANI09	P14_3	Input	Independently connect to AV_{DD} or AV_{SS} via a resistor.
AB6	AV_{SS}	AV_{SS}	–	Always connect to the ground directly.
AB7	AV_{DD}	AV_{DD}	–	Always connect to the power supply.
AB8	ANI03	ANI03	Input	Independently connect to AV_{DD} or AV_{SS} via a resistor.
AB9	V_{SS}	V_{SS}	–	Always connect to the ground directly.
AB10	MDO5	MDO5	Output	Leave open.
AB11	MDO1	MDO1	Output	
AB12	MCKO	MCKO	Output	
AB13	TCK/FLSCK	TCK	Input	Independently connect to DV_{DD} via a resistor.
AB14	$OSCV_{SS}$	$OSCV_{SS}$	–	Always connect to the ground directly.
AB15	X1	X1	Input	–
AB16	$\overline{RESET}$	$\overline{RESET}$	Input	–
AB17	P1_15/P_D31/TA2_I15/TA2_O15	P1_15	Input	Independently connect to EV_{DD} or V_{SS} via a resistor.
AB18	P1_12/P_D28/TA2_I12/TA2_O12	P1_12	Input	
AB19	P1_10/P_D26/TA2_I10/TA2_O10	P1_10	Input	
AB20	P1_8/P_D24/TA2_I8/TA2_O8	P1_8	Input	
AB21	P1_5/P_D21/TA2_I5/TA2_O5/ESO2	P1_5	Input	

2.6 Pin Input Circuit Types

2.6.1 Input circuit types for port input pins and input pins that have port or other functions

The input circuit type is determined by the settings of the port function control registers PIS, PISE, and PISA, as shown in the table below.

Table 2-9 Input Circuit Types for Pins That Have Port Input Function

PISnm	PISEnm	PISAnm	Input Circuit Type
0	0	0	CMOS
1	0	0	Schmitt 2
0	1	0	Schmitt 1
×	×	1	LVTTL
Settings other than the above are prohibited.			

Note For details about the setting of each pin, see *8.4.4 “V850E2/MN4 port control register settings” on page 243*.

2.6.2 Input circuit types for other input pins

The input circuit types for input pins other than those shown in *2.6.1 "Input circuit types for port input pins and input pins that have port or other functions"* are listed in the table below.

Table 2-10 Input Circuit Types for Other Input Pins

Pin No.	Pin Name	Input Circuit Type
AB16	RESET	Schmitt 2

Chapter 3 Operation Modes

3.1 Features

- The operation mode is specified according to the levels of the FLMD0, FLMD1, MODE2, and MODE3 pins.
- There are three operation modes:
  - Normal operation mode (single-chip mode)
  - Flash memory programming mode
  - Boundary scan mode

3.2 Operation Modes

3.2.1 Normal operation mode

This mode makes it possible to start instruction processing by branching to the reset entry address in the internal flash memory after a system reset ends.

3.2.2 Flash memory programming mode

This mode makes it possible to use a flash memory programmer to program the internal flash memory.

3.2.3 Boundary scan mode

This mode makes it possible to use a test method specified by IEEE 1149.1 to check the connections between devices mounted on an application system board.

3.3 Specifying the Operation Mode

The operation mode is specified according to the status of the FLMD0, FLMD1, MODE2, and MODE3 pins. The specifications of these pins are fixed for the application system, and, if they are changed during operation, the operation is not guaranteed. (However, this does not include changing the FLMD0 pin during flash memory self programming.)

Table 3-1 Operation Modes

FLMD0	FLMD1	MODE2	MODE3	Operation Mode
L	L	L	L	Normal operation mode
H	L	L	L	Flash memory programming mode
H	H	L	L	Boundary scan mode
Other than the above				Setting prohibited

Note L: Low level

H: High level

Chapter 4 CPU

4.1 Features

- V850E2M CPU core
- Protection functions
  - System register protection
  - Memory protection
  - Peripheral-device protection
  - Timing monitoring
- Multi-layer system bus
- Multiprocessor (dual-core) configuration[a]
  - Up to two V850E2M CPUs
  - Multiprocessor support

a) Only μPD70F3514 and 70F3515

4.2 Configuration

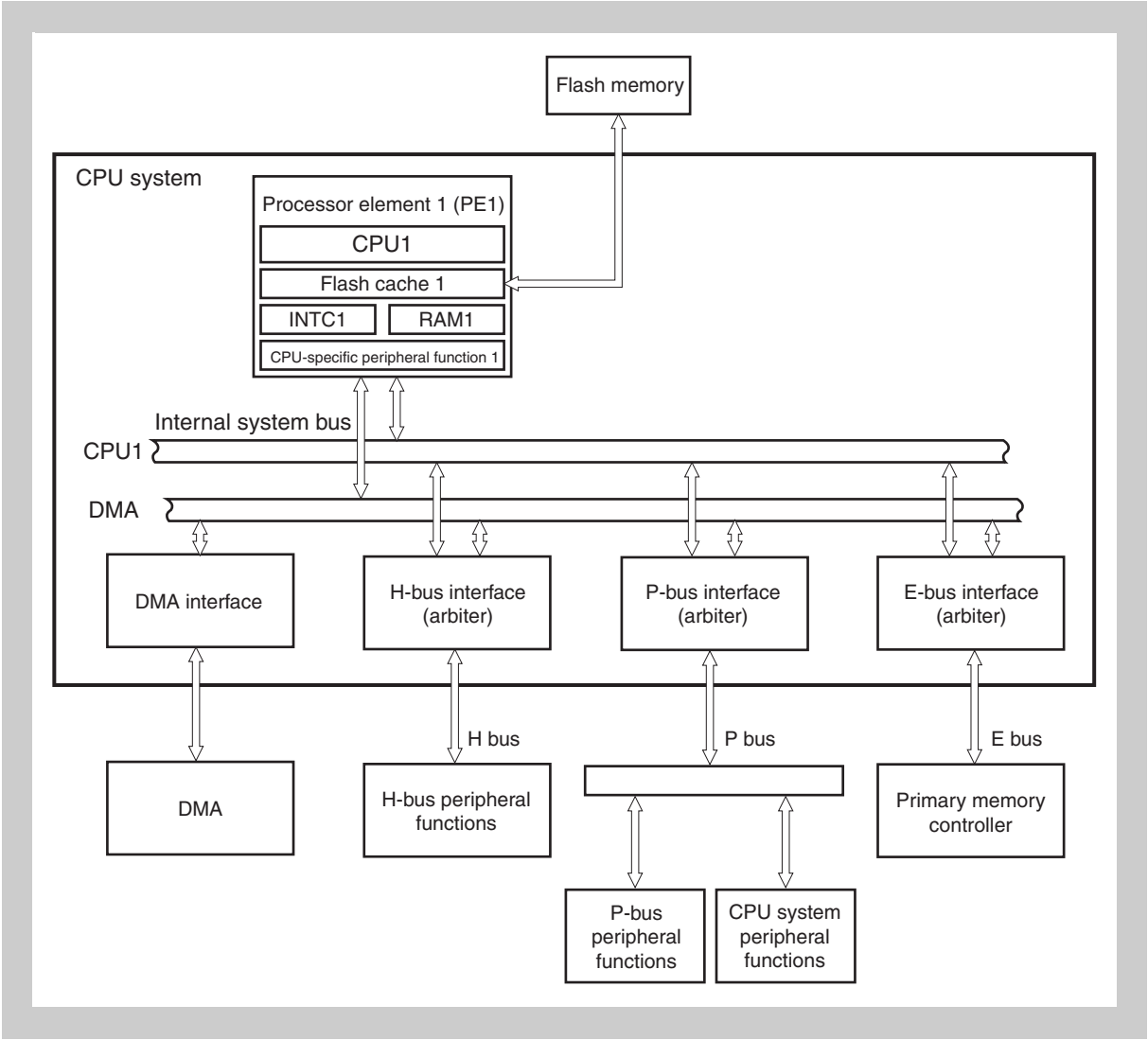


Figure 4-1 CPU system configuration (single core)

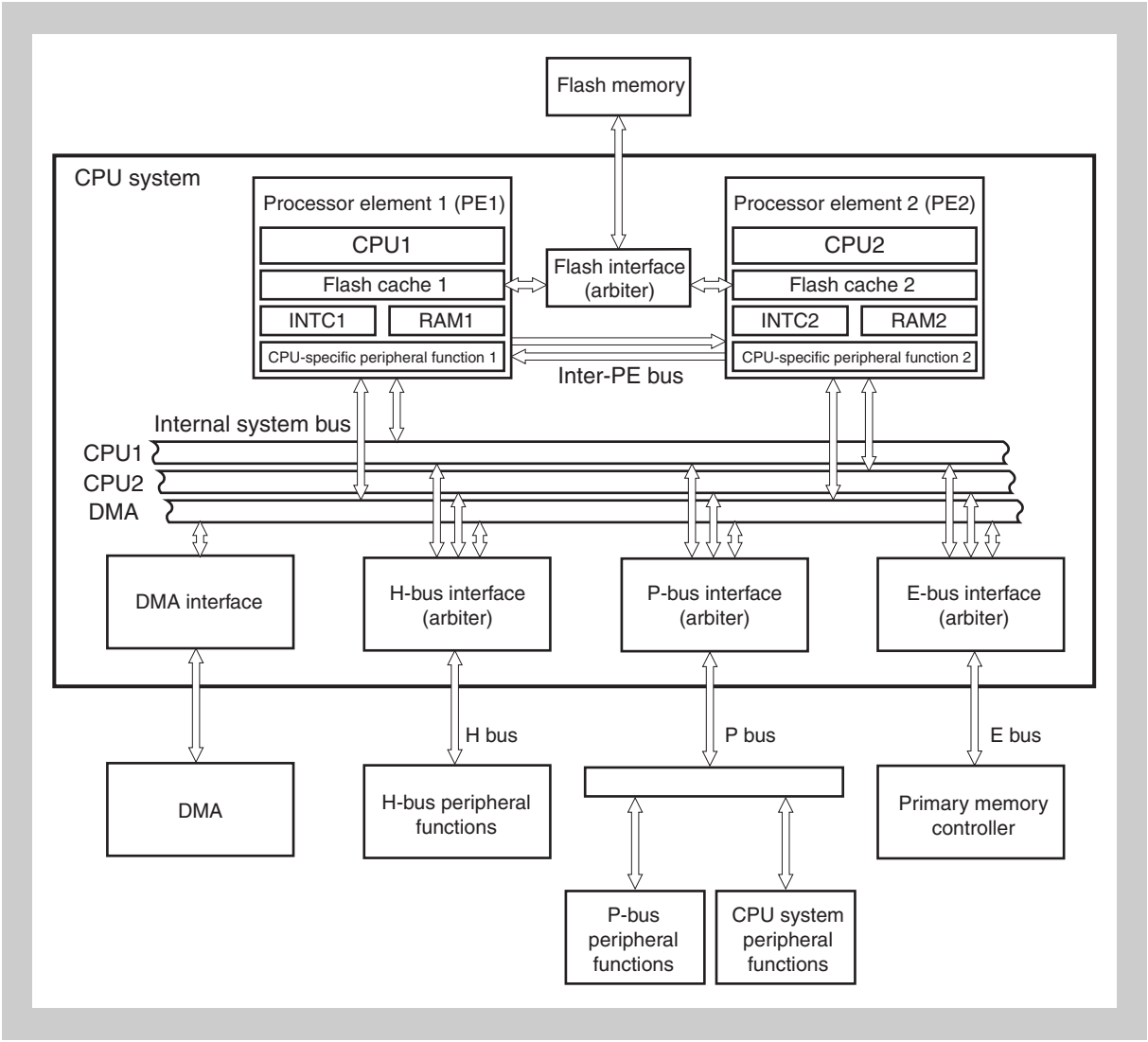


Figure 4-2 CPU system configuration (dual core)

4.3 CPU Core

The V850E2/MN4 uses a V850E2M CPU.

This CPU complies with the V850E2v3 architecture, is designed to provide high performance, advanced functionality, and high reliability, and is intended for device controlling microcontrollers used in embedded systems. The CPU uses seven-stage pipeline control to enable single-clock execution of address calculations, arithmetic logic operations, data transfers, and almost all other instruction processing.

Note that, because the V850E2M CPU is compatible with the V850, V850E1, and V850E2 CPUs at the object code level, conventional system software assets can be used as is.

For details about the V850E2M CPU, see the ***V850E2M Architecture User's Manual (R01US0001EJ)***.

4.4 Address Space

4.4.1 CPU address space

The V850E2M CPU uses a 32-bit architecture that supports a linear address space of up to 4 GB.

However, for this product, instruction fetch addresses and data access addresses are 29 bits, and the accessible address range, which includes both the program space and data space, is a maximum of 512 MB.

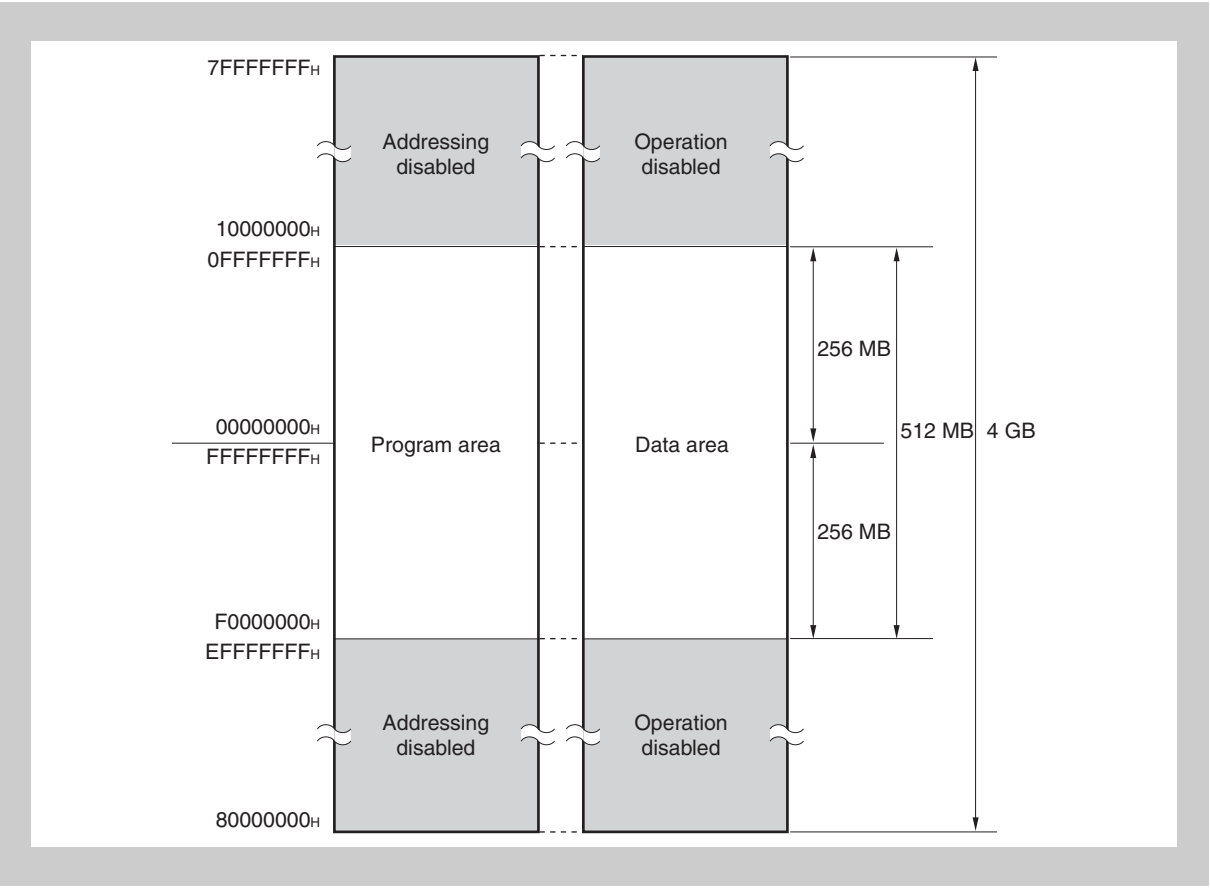


Figure 4-3 CPU address space

<R> For instruction address addressing, a value whose 29th bit is sign-extended is automatically specified for the higher 3 bits of the register holding the instruction address. Therefore, the addressable range is 00000000_H to $0FFFFFFE_H$ and $F0000000_H$ to $FFFFFFFE_H$ (and the least significant bit is always 0). Be sure to place the instructions and the tables referenced using the SWITCH, CALLT, and SYSCALL instructions in the instruction addressable address range.

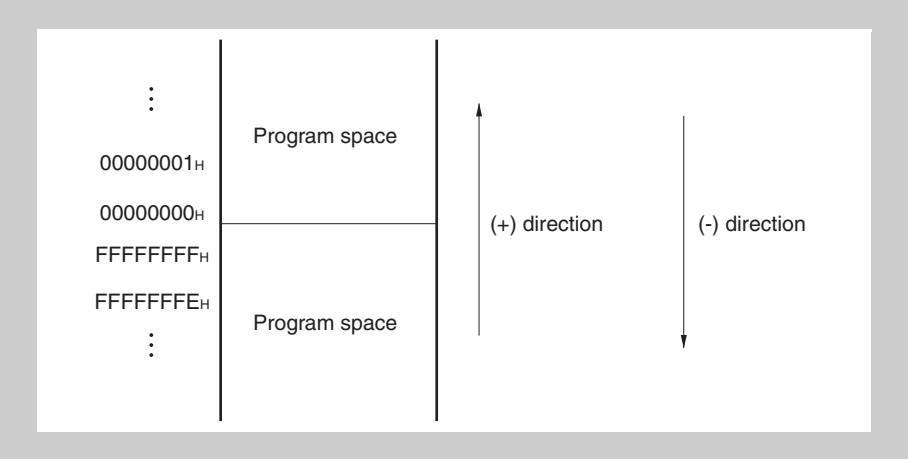
Because, when addressing a physically placed area exceeding 512 MB, mirroring is performed using addresses excluding the higher 3 bits, data access is possible regardless of the placement within the 4 GB space. However, to ensure future compatibility, it is prohibited to run data addressing assuming such mirroring.

4.4.2 CPU address space wraparound

<R> **(1) Program space**

For instruction address addressing, a value whose 29th bit is sign-extended is automatically specified for the higher 3 bits of the register holding the instruction address.

Therefore, the highest address of the program space, FFFFFFFFH, and the lowest address, 00000000H, as well as the addresses are contiguous, and wraparound occurs at the boundaries of these addresses.



<R> **Figure 4-4 Program space wraparound**

(2) Data space

The result of an operand address calculation operation that exceeds 29 bits is ignored.

Therefore, the highest address of the data space, FFFFFFFFH, and the lowest address, 00000000H, are contiguous, and wraparound occurs at the boundary of these addresses.

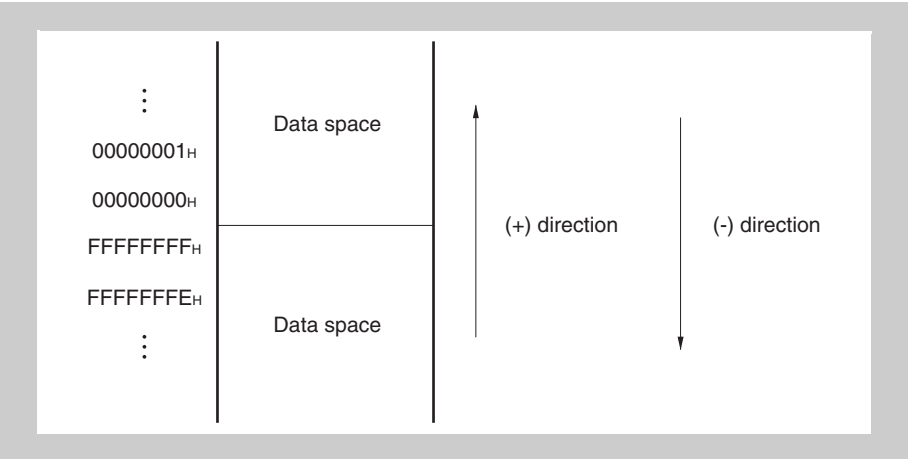


Figure 4-5 Data space wraparound

4.4.3 Memory map

The areas shown below are reserved for the V850E2/MN4.

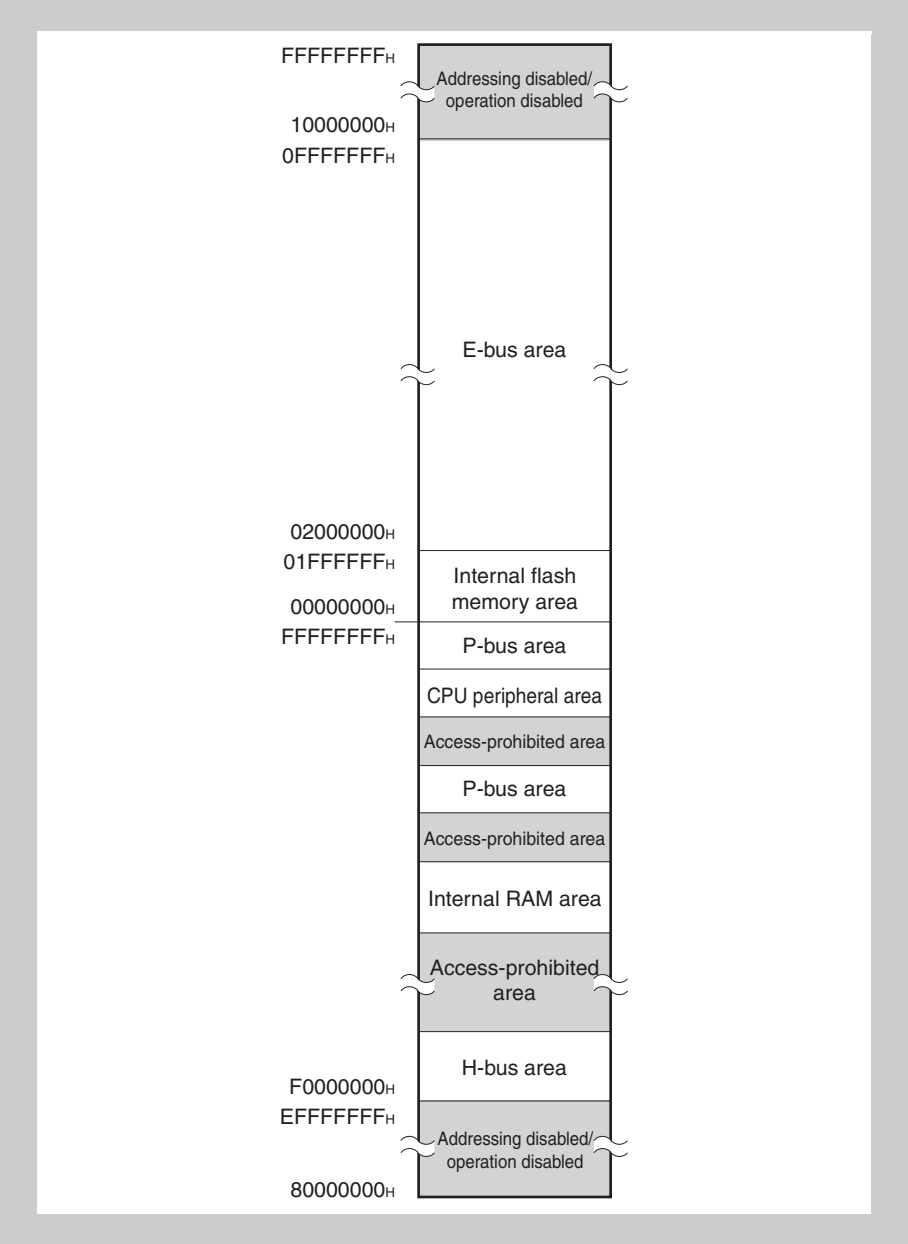


Figure 4-6 Memory map

4.4.4 Areas

(1) Internal flash memory area

The 32 MB from 00000000_H to $01FFFFFF_H$ are reserved as the internal flash memory area.

(a) Internal flash memory (1 MB)

For the products below, 1 MB is provided from 00000000_H to $000FFFFF_H$.

Accessing the addresses 00100000_H to $01FFFFFF_H$ is prohibited.

- μPD70F3510, 70F3512, and 70F3514

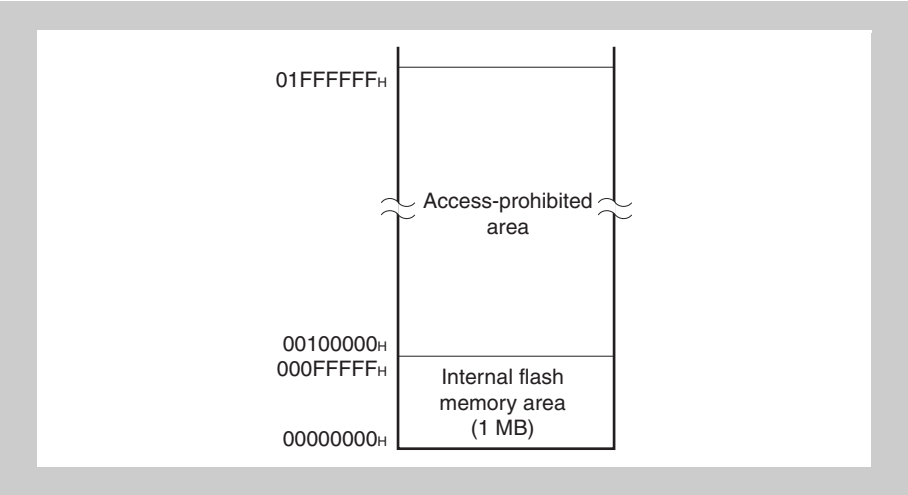


Figure 4-7 Internal flash memory area (1 MB)

(b) Internal flash memory (2 MB)

For the product below, 2 MB are provided from 00000000_H to $001FFFFF_H$.

Accessing the addresses 00200000_H to $01FFFFFF_H$ is prohibited.

- μPD70F3515

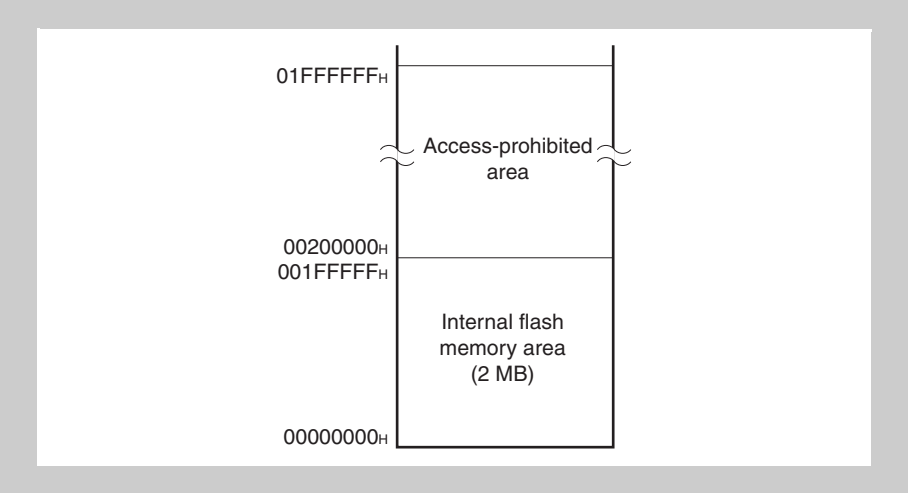


Figure 4-8 Internal flash memory area (2 MB)

(2) Internal RAM area

The 4 MB from $FEA00000_H$ to $FEDFFFFF_H$ are reserved as the internal RAM area.

(a) Internal RAM (64 KB)

For the products below, 64 KB are provided from $FEDF0000_H$ to $FEDFFFFF_H$.

Accessing the addresses $FEA00000_H$ to $FEDEFFFF_H$ is prohibited.

- μPD70F3510 and 70F3512

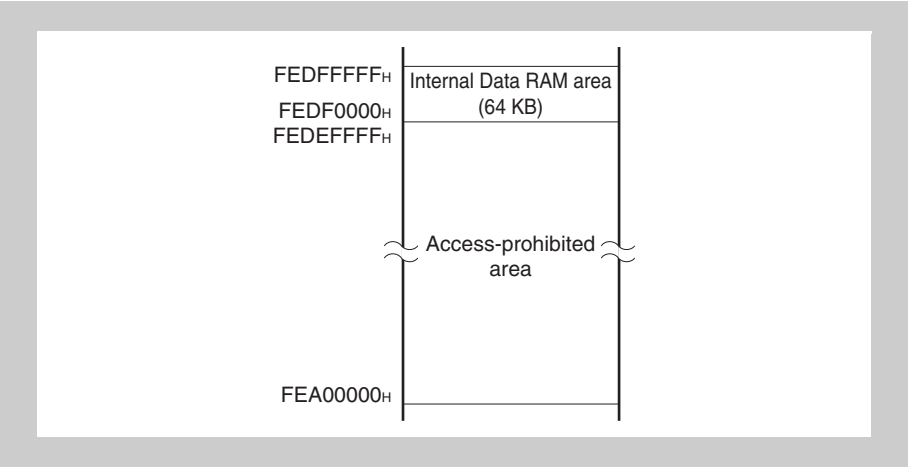


Figure 4-9 Internal RAM area (64 KB)

(b) Internal RAM (64 KB x 2)

For the products below, 64 KB are provided from $FEDF0000_H$ to $FEDFFFFF_H$, and another 64 KB are provided from $FEBF0000_H$ to $FEBFFFFF_H$.

Accessing the addresses $FEC00000_H$ to $FEDEFFFF_H$ and the addresses $FEA00000_H$ to $FEBEFFFF_H$ is prohibited.

- μPD70F3514 and 70F3515

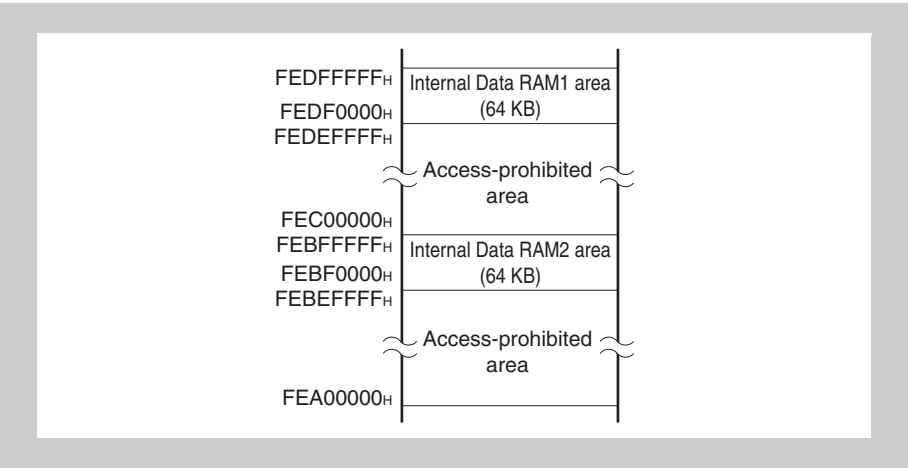


Figure 4-10 Internal RAM area (64 KB x 2)

(3) External memory area and on-chip peripheral I/O area

(a) CPU peripherals

12 KB are provided from $FFFF5000_H$ to $FFFF7FFF_H$ as the area for CPU-specific peripheral functions and CPU system peripheral functions.

Peripheral I/O registers that have functions to specify the operation mode for and monitor the status of the on-chip peripheral I/O are mapped to the on-chip peripheral I/O area.

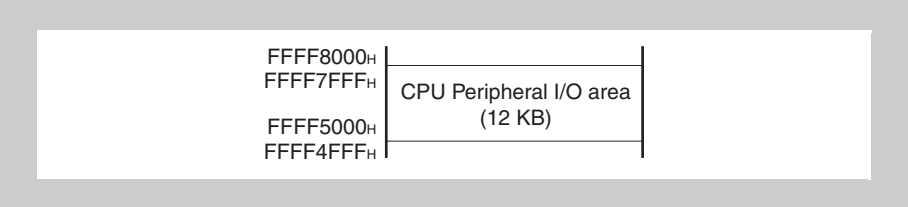


Figure 4-11 On-chip peripheral I/O area (12 KB)

(b) E bus

This is a 224 MB external memory area for the primary memory controller. The external memory area is the area from 02000000_H to $0FFFFFFF_H$.

Accessing the external memory area uses chip select signals assigned to each memory block.

Table 4-1 CS spaces (E bus)

CS	Address	Size
P_CS1	02000000_H-$03FFFFFF_H$	32 MB
P_CS2	04000000_H-$07FFFFFF_H$	64 MB
P_CS3	08000000_H-$0BFFFFFF_H$	64 MB
P_CS4	$0C000000_H$-$0FFFFFFF_H$	64 MB

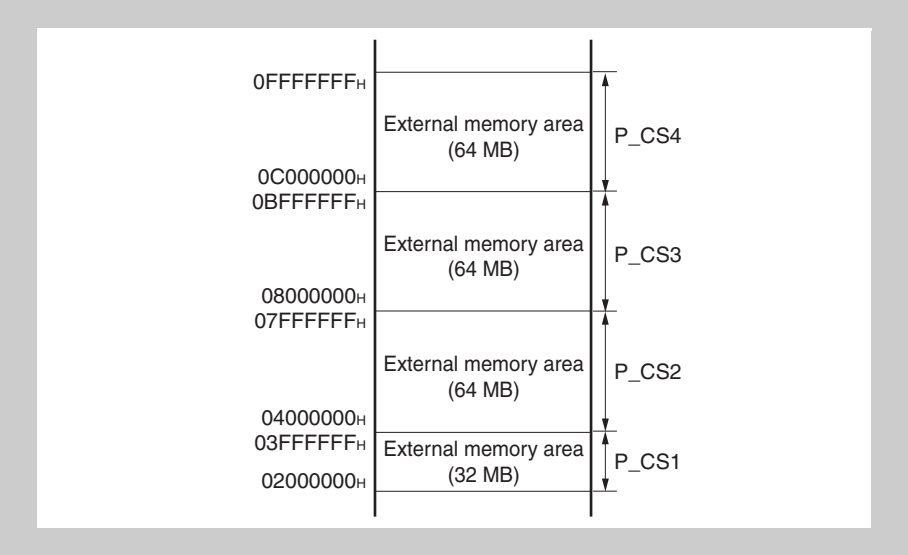


Figure 4-12 CS spaces (E bus)

(c) P bus

The 32 KB from $FFFF8000_H$ to $FFFFFFFF_H$, as well as the 4 MB + 256 KB from $FF400000_H$ to $FF83FFFF_H$, are provided as the area for the P bus.

Peripheral I/O registers that have functions to specify the operation mode for and monitor the status of the on-chip peripheral I/O are mapped to the on-chip peripheral I/O area.

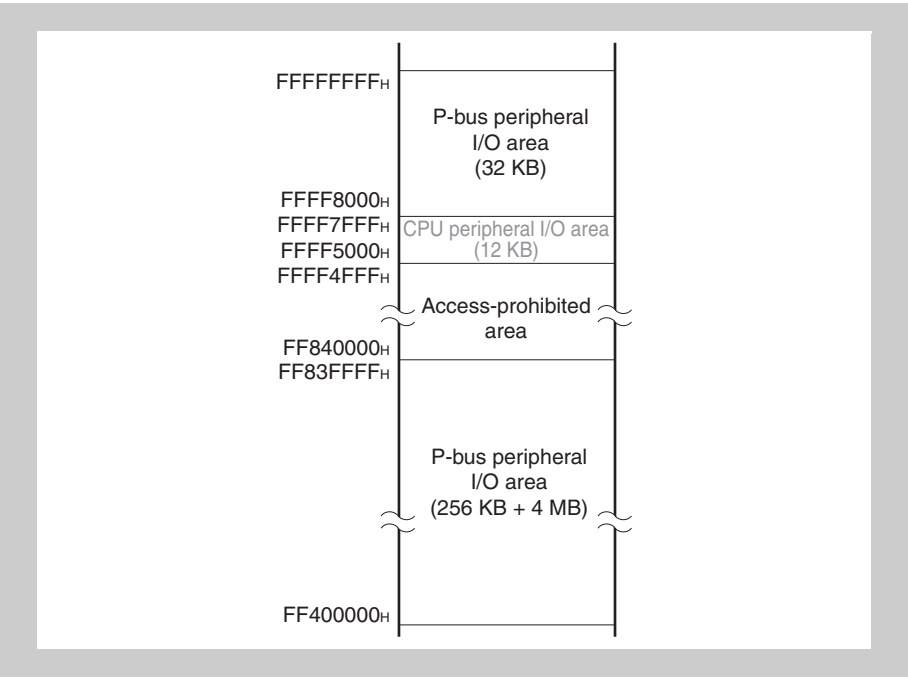


Figure 4-13 On-chip peripheral I/O space (P bus)

(d) H bus

The 160 MB from $F0000000_H$ to $F9FFFFFF_H$ are provided as the on-chip peripheral I/O area for the H bus.

Peripheral I/O registers that have functions to specify the operation mode for and monitor the status of the on-chip peripheral I/O are mapped to the on-chip peripheral I/O area.

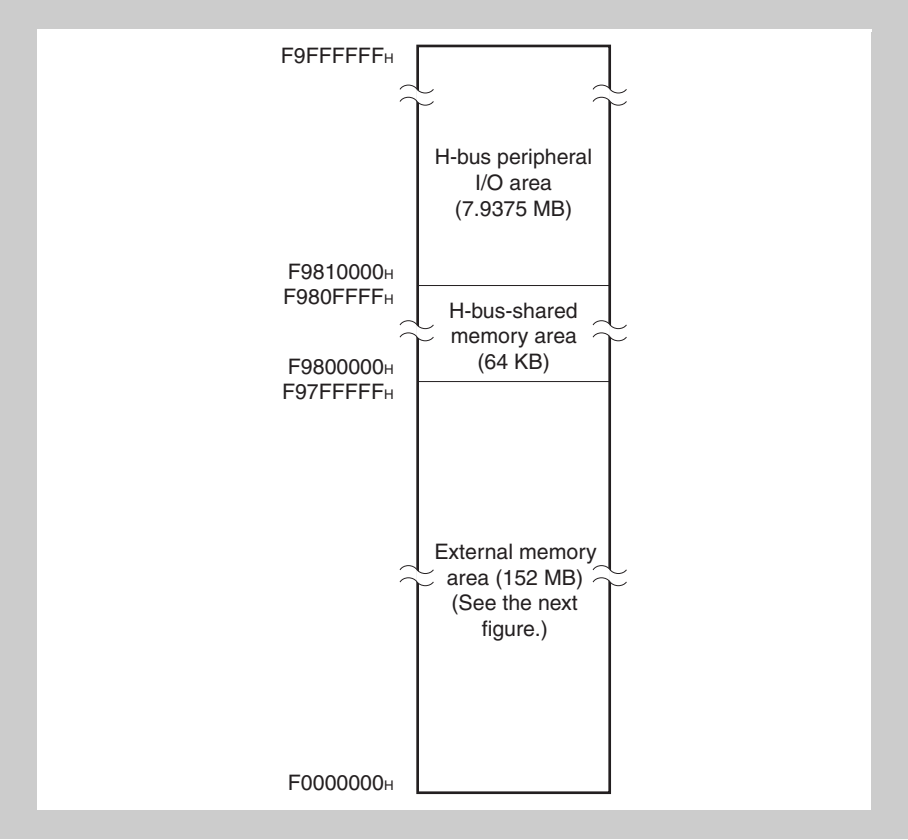


Figure 4-14 Memory map (H bus)

Note that this area includes a 152 MB external memory area for the secondary memory controller. The external memory area is the area from $F0000000_H$ to $F97FFFFF_H$.

Accessing the external memory area uses chip select signals assigned to each memory block.

Table 4-2 CS spaces (H bus)

CS	Address	Size
S_SDCS	$F0000000_H$-$F7FFFFFF_H$	128 MB
S_CS0	$F8000000_H$-$F8FFFFFF_H$	16 MB
S_CS1	$F9000000_H$-$F93FFFFF_H$	4 MB
S_CS2	$F9400000_H$-$F95FFFFF_H$	2 MB
S_CS3	$F9600000_H$-$F97FFFFF_H$	2 MB

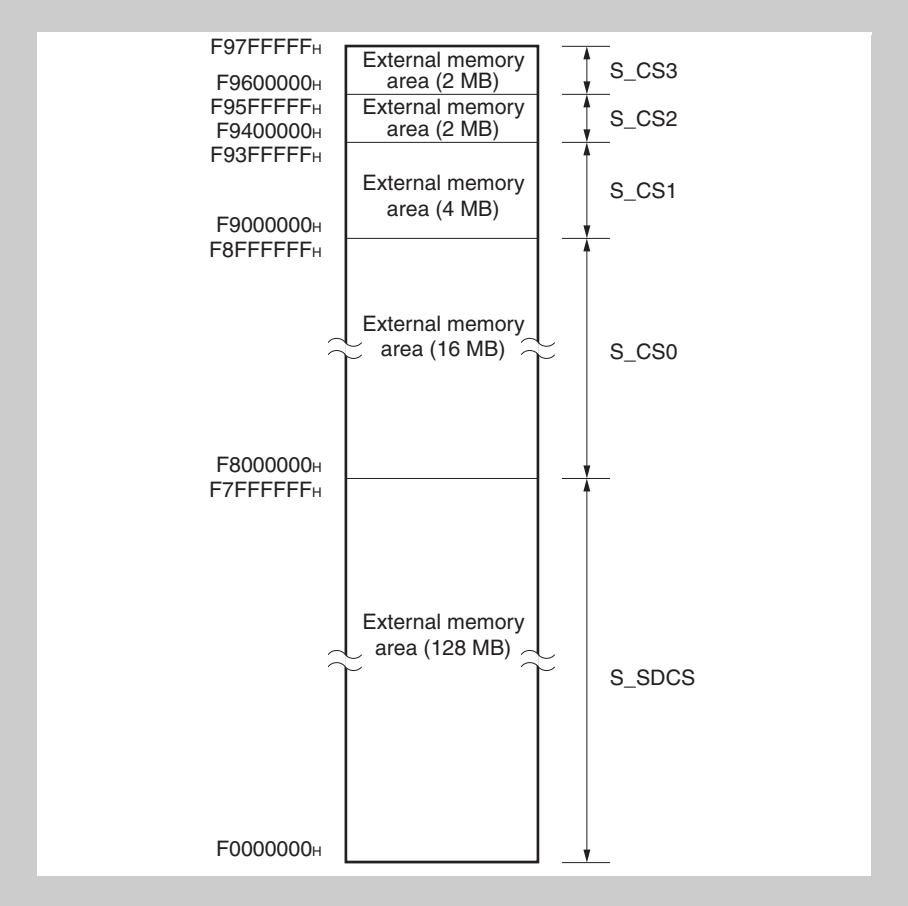


Figure 4-15 External memory area (H bus)

4.5 Processor Elements (PE)

The entity to which the CPU, flash cache, RAM, interrupt controller, and CPU-specific peripheral functions are connected is called a processor element (PE). V850E2/MN4 has up to two PEs, PE1 and PE2.

For the functions below, which are included in each PE, the I/O is assigned to the same address space, but the physical implementations differ.

- Flash cache
- Interrupt controller
- CPU-specific peripheral functions

  For details, see *4.9.1 (3) "CPU-specific peripheral functions and CPU system peripheral functions"*.

4.6 Bus Architecture

4.6.1 Internal system bus

This internal system bus connects each PE and DMA to the P bus, H bus, and E bus. This internal system bus is in a multi-layer configuration, and PE1, PE2, and DMA each has its own layer. This makes it possible to simultaneously access different buses (the P bus, H bus, and E bus) for each PE and DMA. Note that arbitration is performed using each bus interface if there is a conflict when trying to access the same bus.

4.6.2 P bus

This is the bus to which P bus peripherals (TAUA, TAUJ, ENCA, OSTM, WDTA, ADCA, ports, UARTE, UARTJ, CSIG, CSIH, I2CB, and FCN) are connected.

4.6.3 H bus

This is the bus to which the H-bus-shared memory, H-bus memory side cache, secondary memory controller, USB function controller, USB host controller, and Ethernet controller are connected.

(1) Initial settings

Initial settings are necessary to use the H bus. After a reset, perform the following procedure to specify these settings before accessing the H bus:

<1> Set the area to normal access by using the ETARCFG0 register.
Set the MODE bits (bits 7 to 4) to “0000”$_B$ (normal access).

<2> Specify the base address for the area by using the ETARADRS0 register.
Set the base address of the target area to $F0000000_H$ by using bits 28 to 12.

<3> Specify the area size by using the ETARMASK0 register.
Set the mask value for the base address of the target area to $1FFFFFFF_H$ by using bits 28 to 12.

<4> Enable the area by using the ETARCFG0 register.
Set the EN bit (bit 0) to “1” (enabled).

(a) ETA area 0 setting register (ETARCFG0)

This register is used to specify whether area 0 is enabled or disabled and its operation mode.

Access This register can be read or written in 16-bit units.

Address $FFFF7140_H$

Initial value 0000_H. This register is initialized by any reset.

Caution Be sure to clear bits 15 to 1 to “0”.

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
7	6	5	4	3	2	1	0
MODE				0	0	0	EN
R/W	R/W	R/W	R/W	R	R	R	R/W

Table 4-3 ETARCFG0 register contents

Bit Position	Bit Name	Description
7:4	MODE	These bits specify the operation mode for area 0. Set these bits to “0000”. All other values are prohibited.
0	EN	This bit specifies whether area 0 is enabled or disabled. 0: Disabled 1: Enabled.

(b) ETA area 0 address register (ETARADRS0)

This register is used to set the base address that specifies area 0.

Access This register can be read or written in 32-bit units. However, if the higher 16 bits of the ETARADRS0 register are used as the ETARADRS0H register, and the lower 16 bits are used as the ETARADRS0L register, the ETARADRS0 register can be read or written in 16-bit units.

Address ETARADRS0: $FFFF7150_H$, ETARADRS0L: $FFFF7150_H$, ETARADRS0H: $FFFF7152_H$

Initial value After power-on: Undefined

After a reset: The previous value is retained. This register is initialized by any reset.

31	30	29	28	27	26	25	24
ETARADRS[31:24]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
ETARADRS[23:16]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
ETARADRS[15:8]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
ETARADRS[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 4-4 ETARADRS0 register contents

Bit Position	Bit Name	Description
31:0	ETARADRS [31:0]	These bits specify the base address that specifies area 0. Bit 28 is regarded as the sign bit for bits 31 to 29, in which a sign extended value is stored. Be sure to set this to $F0000000_H$.

(c) ETA area 0 mask register (ETARMASK0)

This register is used to specify the mask value for the base address that specifies area 0.

When setting this register, be sure to specify a value that consists of consecutive 1s from the lower bits.

Access This register can be read or written in 32-bit units. However, if the higher 16 bits of the ETARMASK0 register are used as the ETARMASK0H register, and the lower 16 bits are used as the ETARMASK0L register, the ETARMASK0 register can be read or written in 16-bit units.

Address ETARMASK0: $FFFF7154_H$, ETARMASK0L: $FFFF7154_H$, ETARMASK0H: $FFFF7156_H$

Initial value After power-on: Undefined

After a reset: The previous value is retained. This register is initialized by any reset.

31	30	29	28	27	26	25	24
ETARMASK[31:24]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
ETARMASK[23:16]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
ETARMASK[15:8]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
ETARMASK[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 4-5 ETARMASK0 register contents

Bit Position	Bit Name	Description
31:0	ETARMASK [31:0]	These bits are used to specify a mask value for the base address that specifies area 0. Be sure to set this to $1FFFFFFF_H$.

(d) H-bus wait insertion limit register (ETAWRL)

This register is used to specify the maximum number of inserted wait cycles on the H bus.

When a peripheral I/O connected to the H bus is accessed, if the number of inserted wait cycles reaches the value specified for this register, a system error exception (SYSERR) is reported to the CPU.

Access This register can be read or written in 16-bit units.

Address $FFFF7106_H$

Initial value $00FF_H$. This register is initialized by any reset.

Cautions

1. For how to permit the system error exception, see the section about the SEG_CONT register.
2. When using the bus hold function of the secondary memory controller, set this register to 0000_H.

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
7	6	5	4	3	2	1	0
ETAWRL							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 4-6 ETARCFG0 register contents

Bit Position	Bit Name	Description
7:0	ETAWRL	These bits specify the maximum number of inserted wait cycles on the H bus.

ETAWRL	Maximum number of inserted wait cycles on H bus
0000 0000	No limited
0000 0001	One clock
0000 0010	Two clocks
0000 0010	Three clocks
:	:
11111101	253 clocks
11111110	254 clocks
11111111	255 clocks

(2) Multi-layer structure

The H bus has a multi-layer internal configuration, so, as long as no slaves are duplicated, each bus master can access each slave.

The transfer targets are shown below.

Table 4-7 H-bus transfer targets

Master (Transfer Source)	Slave (Transfer Destination)						
	Shared H-Bus Memory	Secondary Memory Controller	USB Function	USB Host	Ethernet Controller (Dedicated DMA)	Ethernet Controller (Tx Checksum)	Secondary Memory Controller (Dedicated DMA)
CPU1/CPU2	Available	Available	Available[a]	Available[a]	Available[a]	Available[a]	Available[a]
DMA	Available	Available	Not available	Not available	Not available	Not available	Not available
USB function	Available	Available	Not available	Not available	Not available	Not available	Not available
USB host	Available	Available	Not available	Not available	Not available	Not available	Not available
Ethernet controller (dedicated DMA)	Available	Available	Not available	Not available	Not available	Not available	Not available
Ethernet controller (Tx checksum dedicated DMA)	Available	Available	Not available	Not available	Not available	Not available	Not available
Secondary memory controller dedicated DMA	Available	Available	Not available	Not available	Not available	Not available	Not available

a) Only control register access is possible.

4.6.4 E bus

This is the bus to which the primary memory controller is connected.

For details about the primary memory controller, see *Chapter 11 "Primary Memory Controller (PMEMC)"* on page 487.

4.6.5 Inter-PE bus

The V850E2/MN4 incorporates up to two PEs, and each PE has internal RAM.

Each CPU can have each PE and RAM reference or update each other.

The inter-PE bus is used to reference or update RAM2 from CPU1 or RAM1 from CPU2.

4.6.6 Arbitration

(1) Flash cache

There are two types of access requests to the flash cache: instruction fetch requests and data access requests. For the flash cache, a data access request is always prioritized (according to a fixed priority).

(2) Flash memory

For a dual-core product, arbitration is performed for access requests to the flash memory from two CPUs.

Arbitration is performed in two stages, after the request from a CPU is divided into a fetch request and preload request.

- 1st stage

  When there is a fetch request from either CPU1 or CPU2, that request is accepted. When fetch requests are issued simultaneously from CPU1 and CPU2, the fetch request from the CPU from which the previously accepted fetch request was not issued is accepted.

- 2nd stage

  If no fetch request is issued from both CPU1 and CPU2 (when there was no request to be selected in the 1st stage), preload requests from CPU1 and CPU2 are accepted.
  When there is a preload request from either CPU1 or CPU2, that request is accepted.
  When preload requests are issued simultaneously from CPU1 and CPU2, the preload request from the CPU from which the previously accepted preload request was not issued is accepted.

(3) Internal RAM

Internal RAM 1 and 2 arbitrate the following three types of access requests: instruction fetch requests, data access requests (requests from within one PE), and requests from the other PE.

The arbitration policy of internal RAM 1 and 2 is shown below.

- Between a request from the other PE and a data access request:
  Round-robin arbitration is performed.
- Between a data access request and an instruction access request:
  The data access request is prioritized as the initial status, and round-robin arbitration is performed when there are successive requests.
- Between a request from the other PE and an instruction access request:
  The request from the other PE is always prioritized (fixed priority).

(4) E bus I/F, P bus I/F, H bus I/F

For the interfaces of the E bus, P bus, and H bus, arbitration is performed on access requests from the masters (CPU1, CPU2, and DMA) on each layer of a system bus in a multi-layer configuration.

Arbitration is performed in the following two stages.

- 1st stage
  Round-robin arbitration of CPU1 and CPU2 is performed.
- 2nd stage
  A fixed priority order is used. DMA access is prioritized.

4.7 Interrupt Function

Each processor element includes an interrupt controller.

Except for some interrupts, interrupts are generally distributed to the interrupt controller of each PE.

The I/O of the interrupt controller for each PE is assigned to the same address space, but the physical implementations differ.

4.7.1 Interrupt sources

(1) PE-shared interrupts

Interrupts defined as shared interrupts are simultaneously distributed to each PE and reported to the respective interrupt controllers. When the interrupt acknowledgment conditions are met by a given PE, the processing is transferred to the corresponding interrupt handler address.

Even when an interrupt is acknowledged by one of the PEs, the interrupt requests of the other PE are not cancelled.

Therefore, when it is necessary to prevent a given interrupt request from being serviced more than once at the same time, one of the following actions must be taken for shared interrupts:

<i> Mask that interrupt, depending on the interrupt function of each PE.

<ii> Perform exclusive startup control in the interrupt handler program of each PE. If it is detected by way of software, such as a semaphore, that execution of the interrupt handler program was started by the other PE first, do not execute the processing and perform the return operation to the first PE.

Because the interrupt handler addresses are all shared among the PEs, the processing is transferred to the same program. To distribute the processing contents among PEs, branch the processing appropriately after referencing the PEID register to check the PE number.

Figure 4-16 “Shared interrupt servicing (when masking the interrupt)” and *Figure 4-17 “Shared interrupt servicing (when using software to perform exclusive startup control)”* show an image of shared interrupt servicing.

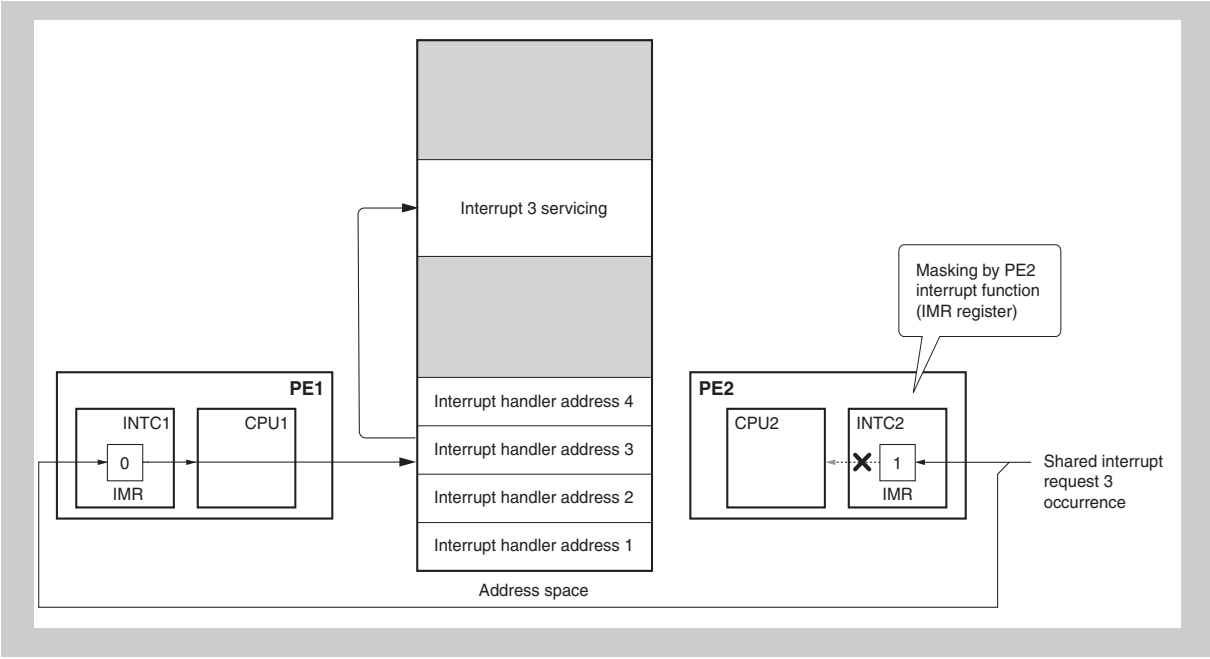


Figure 4-16 Shared interrupt servicing (when masking the interrupt)

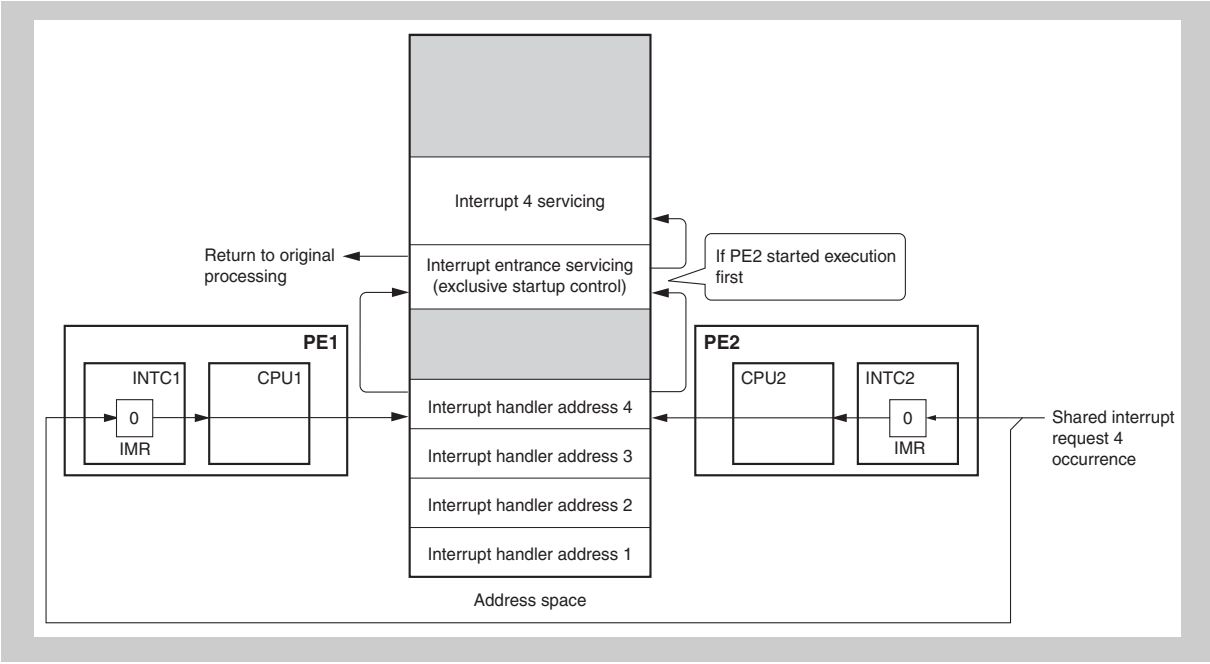


Figure 4-17 Shared interrupt servicing (when using software to perform exclusive startup control)

For V850E2/MN4, a software model that distributes a given EI level maskable interrupt (EIINT) request to only one of the CPUs by exclusively enabling interrupts for each PE is recommended. Therefore, for the IMR register of each PE, specify which EIINT interrupt to acknowledge. Clear the IMR.EIMK bit of the INTC setting register of the PE that is to acknowledge the interrupt to “0”.

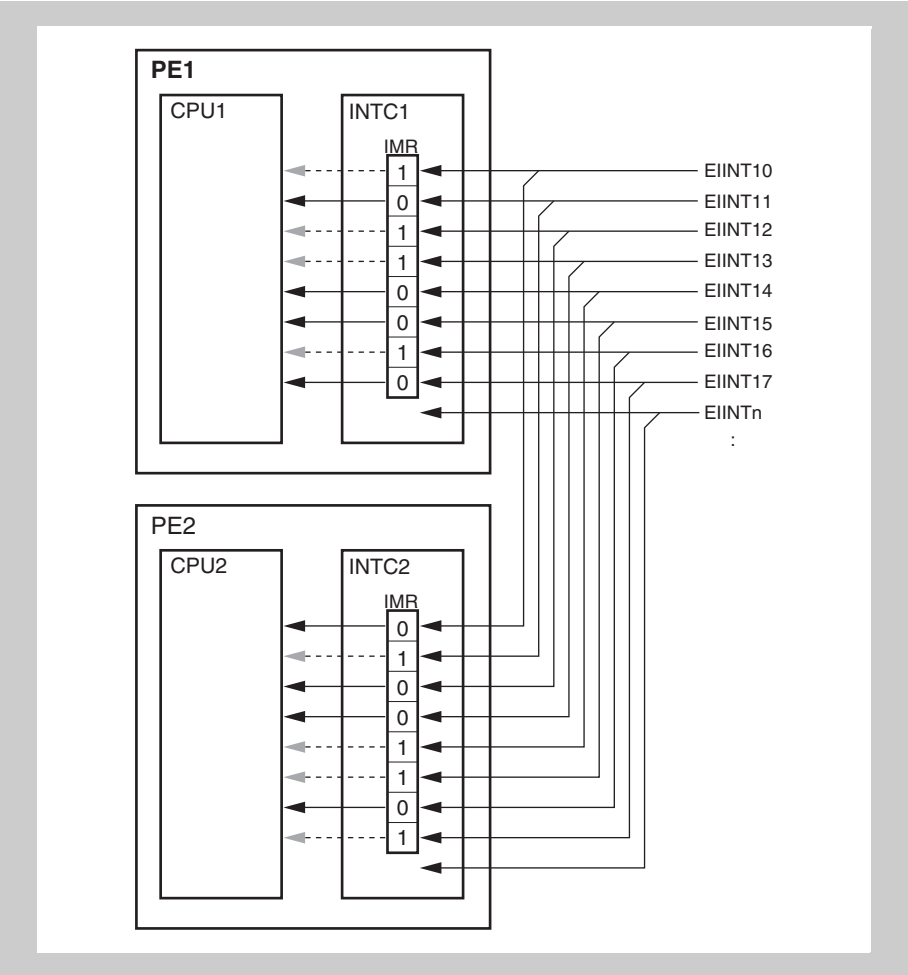


Figure 4-18 Exclusive setting of the EIINT interrupt mask

The following interrupts are shared for the V850E2/MN4:

- FE-level non-maskable interrupt (FENMI)
- FE-level maskable interrupt (FEINT)
- EI-level maskable interrupt channels 10 to 255 (EIINT10 to EIINT255)

(2) PE-dedicated interrupts

Interrupts defined as PE-dedicated interrupts are reported to the interrupt controller of one specified PE. If the notified PE meets the interrupt acknowledgment conditions, the processing is transferred to the corresponding interrupt handler address.

Each PE-dedicated interrupt is provided with a V850E2/MN4 interrupt function as a different interrupt request, but, in the case of a PE-dedicated interrupt allocated to the same channel, the processing is transferred to the program of the same interrupt handler address.

Therefore, to divide the processing contents per PE, branch the processing appropriately after referencing the PEID register to check the PE number.

Figure 4-19 "PE-dedicated interrupt servicing image" shows an image of PE-dedicated interrupt servicing.

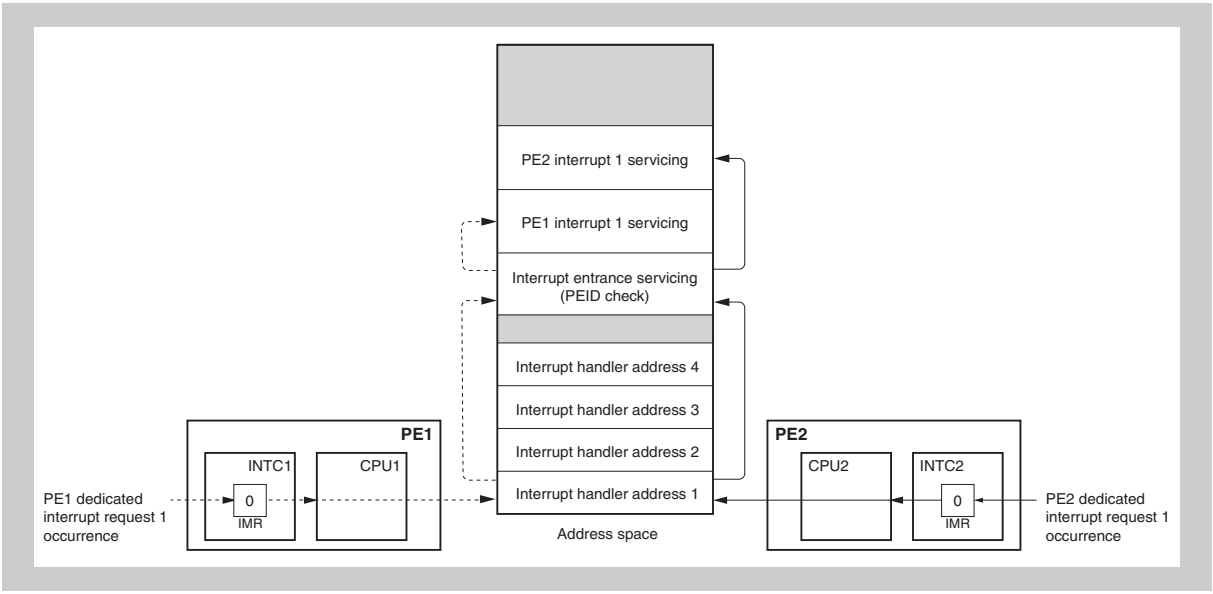


Figure 4-19 PE-dedicated interrupt servicing image

The following interrupts are PE-dedicated interrupts for the V850E2/MN4:

- EI-level maskable interrupt channels 0 to 9

4.7.2 Interrupt distribution

V850E2/MN4 distributes interrupts to each PE in the CPU subsystem. The following table lists the distribution destination of each interrupt function.

Table 4-8 Distribution destinations of each interrupt function

Interrupt Function and Request Source			μPD70F3514 and 70F3515 (Multiprocessor Configuration)			μPD70F3510 and 70F3512 (Single-Processor Configuration)	
			Usability	Interrupt Distribution Destination		Usability	Interrupt Distribution Destination
				PE1	PE2		PE1
Shared interrupts	Request to the interrupt controller	FENMI	Usable	FENMI	FENMI	Usable	FENMI
		FEINT	Usable	FEINT	FEINT	Usable	FEINT
		EIINT[0:9]	Prohibited	–	–	Prohibited	–
		EIINT[10:255]	Usable	EIINT[10:255][a]	EIINT[10:255][a]	Usable	EIINT[10:255][a]
PE-dedicated interrupts	Request through inter-CPU interrupt request register MIR	When MIR0.M1 is specified from PE1	Usable	–	EIINT0	Prohibited	–
		When MIR1.M1 is specified from PE1		–	EIINT1		–
		When MIR0.M1 is specified from PE2		EIINT0	–		–
		When MIR1.M1 is specified from PE2		EIINT1	–		–
		Other than the above		–	–		–
	Request to the interrupt controller	PE1 PE guard	Usable	EIINT2	–	Prohibited	–
		PE2 PE guard		–	EIINT2		–

a) The interrupt distribution destination EIINT[10:255] refers to EI-level maskable interrupts. For details, see *9.2.3 “EI level maskable interrupts” on page 325* .

4.8 Flash Cache

4.8.1 Configuration

For the V850E2/MN4, a 16 KB, four-way set associative cache dedicated to the flash memory (called the flash cache below) is mounted between the CPU and flash memory. The flash cache and flash memory are interconnected by a 128-bit dedicated bus, minimizing the penalty when a cache miss hit occurs.

The flash cache is made up of the following functions:

- Cache function
- Preload function
- Data buffer function
- Cache clear function

4.8.2 Control registers

(a) Flash cache operation setting register (FCCTL0)

This register is used to set up the functions of the flash cache.

When the setting of this register has been changed, clear the contents of the cache by manipulating the FCCTL1 register.

Access This register can be read or written in 16-bit units.

Address FCCTL0: $FFFF6480_H$

Initial value $40B7_H$. This register is initialized by any reset.

Caution Be sure to set this to $40B7_H$ or $41B7_H$. For details, see *Table 4-11 "FCCTL0 register settings"*.

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

Table 4-9 FCCTL0 register contents

Bit Position	Bit Name	Description
15:0	–	Be sure to set this to 40B7H or 41B7H. For details, see *Table 4-11 "FCCTL0 register settings"*.

(b) Flash cache trigger function register (FCCTL1)

This register is used to clear the flash cache.

Access This register can be written in 16-bit units. However, when using the lower 8 bits of the FCCTL1 register as the FCCTL1L register, FCCTL1 can be written in 8-bit or 1-bit units.

Address FCCTL1: $FFFF6482_H$, FCCTL1L: $FFFF6482_H$

Initial value 0000_H. This register is initialized by any reset.

Caution Be sure to clear bits 15 to 1 to “0”.

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	FCCTL1 CHCLR
R	R	R	R	R	R	R	W

Table 4-10 FCCTL1 register contents

Bit Position	Bit Name	Description
0	FCCTL1 CHCLR	Cache clear trigger This bit clears the cache. This bit is always 0 when it is read. 1: Clears the cache contents.

4.8.3 Operation

(1) Cache function

A 16 KB 4-way set associative cache has four ways consisting of 256 entry blocks each, where one line consists of 4 words, to make up 16 KB of capacity. If a cache miss occurs, refilling is performed in line units (= 16 bytes) by using a pseudo-LRU switching algorithm.

(a) FCCTL0 register settings

The settings below can be specified for flash cache operations by using the FCCTL0 register.

Be sure to specify one of the operation modes after a reset. All other values are prohibited.

Table 4-11 FCCTL0 register settings

Type	FCCTL0 Register Setting	Operation
Standard setting	$40B7_H$	This setting specifies the use of the flash cache as an instruction cache. This setting increases the performance. The access cycle fluctuates according to the cache hit/miss-hit rate.
Performance prioritizing setting	$41B7_H$	This setting specifies the use of the flash cache as a mixed instruction/ data cache. This increases the data access speed. If using a lot of data, instructions stored in the cache might be flushed out, decreasing performance.

(2) Preload function

In the case of instruction fetch address access from the CPU, the flash memory contents at addresses +16 bytes, +32 bytes, and +48 bytes are stored in the flash cache before access from the CPU. As a result, when the CPU fetches subsequent instructions, it can use the instructions stored in the cache without having to fetch the flash memory contents anew, thereby realizing efficient CPU instruction execution.

(3) Data buffer function

Two 128-bit data-access-dedicated buffers are provided in the flash cache. The contents of past data access are stored in this data buffer. During data access, reading from the flash memory is performed in 128-bit units, and 128 bits of data are stored in two data buffers. During the next data access, in the case of access to data already stored in these data buffers, that data is fetched from these data buffers, and the flash memory is not accessed. Therefore, constant value access to the flash memory is performed at high speed.

(4) Cache clear function

The flash cache contents and data buffer contents can be cleared by writing “1” to the FCCTL1CHCLR bit of the FCCTL1 register. The internal data of the cache is invalidated in batch. When the flash memory contents are rewritten through flash self programming, clear the cache contents by using this function.

4.9 Multiprocessor Support

4.9.1 Distributed shared type address map

The V850E2/MN4 uses a distributed shared type memory configuration for efficient multiprocessing by way of multiple PEs. Except for CPU-specific peripheral functions, all memory resources can be shared using the same address. In the case of CPU-specific peripheral functions, individual resources are allocated for the same address, but the function is equivalent for each PE.

Therefore, use is possible without changing the software in both the single-processor configuration and multiprocessor configuration.

(1) Code memory sharing

Each CPU performs instruction fetching and data access for the flash memory. Because the flash memory address is the same as seen from the respective CPUs, the same instruction is fetched when fetching an instruction by using the same address from the respective CPUs. When instruction fetching or data access occurs from both CPUs, access from one of the CPUs is enabled by way of the arbitration function, and the other CPU is forced to wait to access the flash memory. The CPU that was forced to wait fetches an instruction after the CPU that obtained access permission first finishes fetching an instruction.

Instruction fetches occur with extremely high frequency, and, to reduce the number of arbitrations, each PE provides a flash cache for caching the contents of the flash memory between the CPU and flash memory. When a flash cache is hit, an instruction is supplied from the flash cache, and the flash memory itself is not accessed.

For the V850E2/MN4, the same address is used for the reset handler address and interrupt handler address for both CPUs. When the same exception occurs, both CPUs read and execute the same contents from the same flash memory address. To execute different processing (during bootup, etc.) for each CPU with a program placed at the same handler address, execute processing according to the PE number after referencing the local PE number by using the PE number report function.

Note that, for the V850E2/MN4, programs can be executed in the RAM. Programs placed in RAM1 and RAM2 can be executed from CPU1 and CPU2, without distinction between the two.

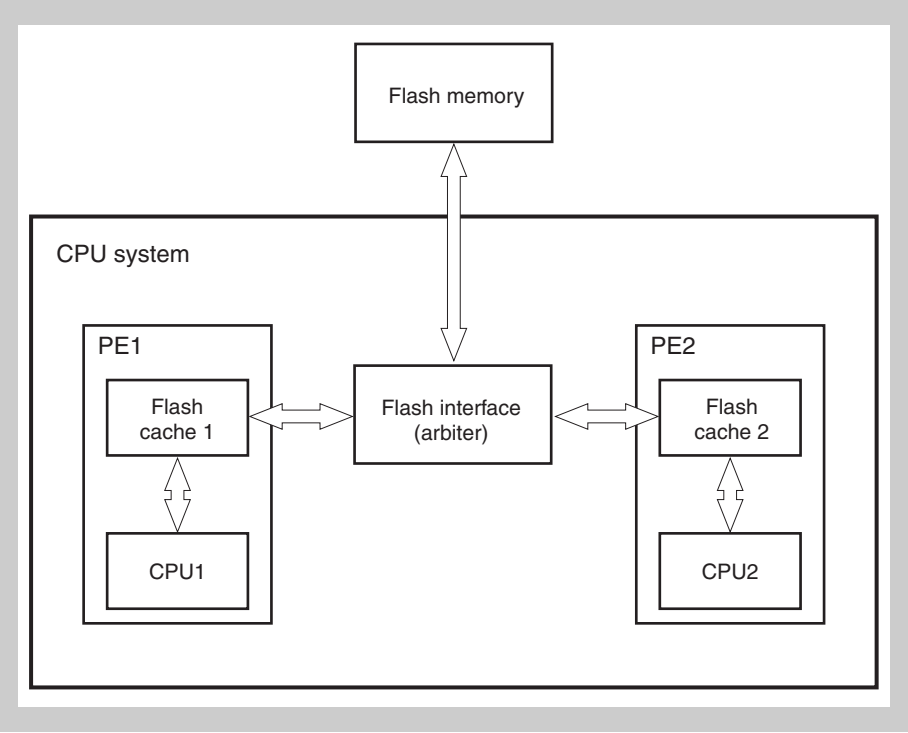


Figure 4-20 Code memory (flash memory) sharing

(2) Internal RAM sharing

The V850E2/MN4 uses a configuration in which the dedicated memory of each CPU can be referenced mutually, making it possible to directly reference and transfer the transfer data. This makes it possible to provide inter-CPU communication that has a high degree of freedom and no software overhead.

Non-overlapping addresses are assigned on the respective memory maps for these internal RAMs. As a result, an internal RAM indicated by a unique address is a single location throughout the entire V850E2/MN4. That internal RAM can be accessed using this address from either CPU. No special procedure is necessary to access the internal RAM of the other CPU.

(3) CPU-specific peripheral functions and CPU system peripheral functions

The following two types of peripheral functions are provided to control the basic functions provided with the V850E2/MN4:

- CPU-specific peripheral functions
- CPU system peripheral functions

(a) CPU-specific peripheral functions

These are peripheral functions provided separately for each PE. For the most part, function registers for specifying the operation settings of each PE and control registers for the function devices (the interrupt controller, cache, etc.) of each PE are connected.

For the V850E2/MN4, each PE separately provides the same CPU-specific peripheral functions.

For the V850E2/MN4, the following CPU-specific peripheral functions are provided:

- Interrupt controller (INTC)
- Flash cache
- PEID (processor element ID) register
- System error controller
- Inter-CPU interrupts
- Mutual exclusion variable register/mutual exclusion control register[b]
- Peripheral device protection function setting register
- Timing monitor function setting register

These CPU-specific peripheral functions are placed at the same addresses from each CPU, and, when accessed from a CPU, the CPU-specific peripheral function corresponding to that CPU responds. The CPU-specific peripheral functions of another CPU can only be accessed from that CPU.

These CPU-specific peripheral functions are accessed through blocking access. When reading a register, the CPU does not execute subsequent instructions until the read operation is complete. When writing to a register, the CPU does not execute subsequent instructions until the write operation is complete. These CPU-specific peripheral functions influence the operation of the CPUs themselves, so register write and read operations are done in synchronization with the operation of the CPUs.

(b) CPU system peripheral functions

The CPU system peripheral functions include control registers for the basic functions provided by the V850E2/MN4 without distinction among the PEs.

These peripheral functions can be accessed and manipulated from any CPU. Note that these peripheral functions are accessed through non-blocking access. When writing to these peripheral functions, the CPU executes subsequent processing without waiting for completion of the write operation. When reading from these peripheral functions, the CPU executes executable subsequent instructions without waiting for completion of the read operation.

For the V850E2/MN4, the following registers for controlling the CPU system peripheral functions are provided:

- H-bus interface setting register
- Primary memory controller setting register
- DMA setting register

b) This function is a resource shared by each CPU.

(4) E bus/P bus/H bus area sharing

For the V850E2/MN4, equal access to I/O placed in the E bus/P bus/H bus area is possible from all masters (each PE and DMA). Therefore, the programs for controlling I/O and peripheral functions can be allocated to any CPU. A high degree of freedom is given when allocating the programs controlling peripheral functions to the respective CPUs.

When peripheral functions belonging to the same bus are accessed from different masters, arbitration is done within the V850E2/MN4 and access is executed from either master. Access from the other master is forced to wait until the access from the first master is complete. When a peripheral function of a different bus is accessed, access without waiting is possible.

These peripheral functions are accessed through non-blocking access. When writing to these peripheral functions, the CPU executes subsequent processing without waiting for completion of the write operation. When reading from these peripheral functions, the CPU executes executable subsequent instructions without waiting for completion of the read operation.

However, the order of operations during access from the same master is guaranteed. If the master accesses peripheral function A, which belongs to the P bus, and then peripheral function B, which belongs to the H bus, a bus cycle is output for peripheral function B after completion of the bus cycle for peripheral function A.

(5) Reset/exception handler address area sharing

For the V850E2/MN4, branching to the same address (the same code) is executed during interrupt/exception acceptance for sharing of the interrupt/exception programming model. As a result, interrupt servicing can be transferred from one CPU to another.

The interrupt/exception processing is mainly supplied by the operating system, but the same functions can be provided for each PE without increasing the amount of code.

The start address on a reset is also the same.

4.9.2 Inter-PE RAM access

For the V850E2/MN4, internal RAM is mounted for each PE in the case of a multiprocessor configuration. Each internal RAM is prioritized for use by the corresponding PE, but it can also be read or written by the other PE.

Each internal RAM is placed at a separate address location in the memory space. By reading from or writing to the address of the internal RAM to be accessed, the internal RAM is accessed if the address is in the internal RAM. If the address is in the internal RAM of the other PE, the internal RAM of that PE is accessed. No special access procedure for the internal RAM connected to the other PE is required other than specifying the address and accessing that internal RAM.

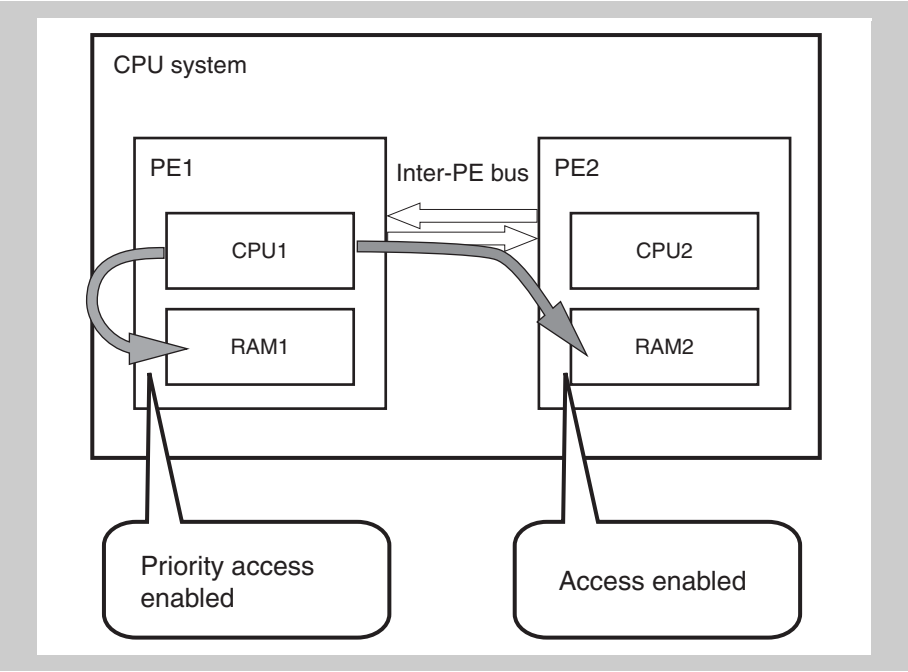


Figure 4-21 RAM access from PE1

4.9.3 Processor element IDs

The V850E2/MN4 includes up to two PEs. A function is available for reporting the processor number in cases when differentiating the processing according to the PE number is desired when executing programs by using the CPU of each PE. CPU processor numbers can be checked by using the processor element ID register (PEID).

The PEID register provided in each CPU can be read from either CPU through 16-bit read access to the address FFFF6490H. When the PEID register is read from the CPU of each PE, different values are read. The value that can be read indicates the respective PE number. Use this information to branch the processing of each CPU or perform similar processing.

By using the processor number report function, the processing contents of each CPU can be changed even though the program code is the same.

Caution The processor number report function can be used in both the multiprocessor configuration and single processor configuration. However, in the single processor configuration, "0001_H" is always read because PE2 does not exist.

(1) Processor element ID register (PEID)

The PEID register returns the processor element ID of the CPU that accesses this register.

Access This register is read-only, in 16-bit units.

Address $FFFF6490_H$

Initial value PEID of PE1: 0001_H, PEID of PE2: 0002_H

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PEID															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 4-12 PEID register contents

Bit Position	Bit Name	Description
15:0	PEID	These bits indicate a processor element ID. If PEID was read from PE1: 0001_H can be read. If PEID was read from PE2: 0002_H can be read.

4.9.4 Mutual exclusion support function

The V850E2/MN4 provides the following registers supporting mutual-exclusion processing by software:

- Mutual-exclusion variable registers (MEV)
- Mutual-exclusion control registers (MEC)

Variables read or written among multiple CPUs are called “shared resources". Coordination is done while exchanging data through these shared resources. When multiple CPUs simultaneously access shared resources, the program might perform unintended operations depending on the access sequence.

In such a case, it is necessary to specify a shared flag (access rights) for these shared resources to perform control so as to prevent simultaneous access from multiple CPUs. This processing is called “mutual-exclusion processing".

(1) Mutual-exclusion processing example

- Define shared flags (access rights) for shared resources. These access rights are defined by the contents of a given register.
- When a CPU accesses a shared resource, first it checks the shared flag to see whether the other CPU is currently accessing that resource.
- If the shared flag indicates that the resource is being used by the other CPU, the CPU waits until the shared flag indicates the unused status (or executes some other processing).
- If the shared flag indicates that the resource is not being used, a value indicating that the resource is in use is written to the flag. Next, read and write processing is performed for this shared resource.
- Upon completion of access to the shared resource, the flag is set to a value indicating that the resource is not in use.

The generally used method is that, when a given CPU references this shared flag (access rights) and determines that the other CPU is using the corresponding resource, the former CPU references this shared flag (access rights) again (polling processing) and waits until the latter CPU releases the access right.

Because such processing (polling processing) frequently uses the CPU bus, when this shared flag is placed in memory frequently accessed by the CPU (internal RAM, etc.), the general processing of the other CPU and this polling processing might conflict on the bus of that internal RAM and degrade the system performance.

For the V850E2/MN4, such degradation is prevented by specially providing registers for shared flags (access rights) (mutual-exclusion variable registers (MEV) and mutual-exclusion control registers (MEC)) that are connected to a different bus than the one used for internal RAM access. The execution of polling processing for these registers does not inhibit internal RAM access from the other CPU or degrade the system performance.

(2) Mutual-exclusion variable registers 0 to 7 (MEV0 to MEV7)

Mutual-exclusion processing is performed by placing shared-flag variables and semaphore variables in these registers and accessing these registers by using the CAXI and SET1 instructions.

Because these mutual-exclusion variable registers are placed on a bus different from that used by the CPU for internal RAM access, even when CPU1 executes polling processing required for mutual-exclusion processing for this MEV, CPU2 can simultaneously access the internal RAM of both CPU1 and CPU2. Therefore, degradation in system performance can be prevented, compared to when semaphore variables are placed in general memory resources.

Access These registers can be read or written in 32-bit units.

Note, however, that the MEVn (n = 0 to 7) register can be read and written in 16-bit units if its upper 16 bits and lower 16 bits are used as the MEVnH and MEVnL registers, respectively. In addition, 8-bit access is possible if the upper 8 bits and lower 8 bits of the MEVnH (n = 0 to 7) register are used as the MEVnHH and MEVnHL registers, respectively, and the upper 8 bits and lower 8 bits of the MEVnL register are used as the MEVnLH and MEVnLL registers, respectively.

Address MEV0: $FFFF6900_H$,
MEV0L: $FFFF6900_H$, MEV0H: $FFFF6902_H$,
MEV0LL: $FFFF6900_H$, MEV0LH: $FFFF6901_H$,
MEV0HL: $FFFF6902_H$, MEV0HH: $FFFF6903_H$,
MEV1: $FFFF6904_H$,
MEV1L: $FFFF6904_H$, MEV1H: $FFFF6906_H$,
MEV1LL: $FFFF6904_H$, MEV1LH: $FFFF6905_H$,
MEV1HL: $FFFF6906_H$, MEV1HH: $FFFF6907_H$,
MEV2: $FFFF6908_H$,
MEV2L: $FFFF6908_H$, MEV2H: $FFFF690A_H$,
MEV2LL: $FFFF6908_H$, MEV2LH: $FFFF6909_H$,
MEV2HL: $FFFF690A_H$, MEV2HH: $FFFF690B_H$,
MEV3: $FFFF690C_H$,
MEV3L: $FFFF690C_H$, MEV3H: $FFFF690E_H$,
MEV3LL: $FFFF690C_H$, MEV3LH: $FFFF690D_H$,
MEV3HL: $FFFF690E_H$, MEV3HH: $FFFF690F_H$,
MEV4: $FFFF6910_H$,
MEV4L: $FFFF6910_H$, MEV4H: $FFFF6912_H$,
MEV4LL: $FFFF6910_H$, MEV4LH: $FFFF6911_H$,
MEV4HL: $FFFF6912_H$, MEV4HH: $FFFF6913_H$,
MEV5: $FFFF6914_H$,
MEV5L: $FFFF6914_H$, MEV5H: $FFFF6916_H$,
MEV5LL: $FFFF6914_H$, MEV5LH: $FFFF6915_H$,
MEV5HL: $FFFF6916_H$, MEV5HH: $FFFF6917_H$,
MEV6: $FFFF6918_H$,
MEV6L: $FFFF6918_H$, MEV6H: $FFFF691A_H$,
MEV6LL: $FFFF6918_H$, MEV6LH: $FFFF6919_H$,
MEV6HL: $FFFF691A_H$, MEV6HH: $FFFF691B_H$,
MEV7: $FFFF691C_H$,
MEV7L: $FFFF691C_H$, MEV7H: $FFFF691E_H$,
MEV7LL: $FFFF691C_H$, MEV7LH: $FFFF691D_H$,
MEV7HL: $FFFF691E_H$, MEV7HH: $FFFF691F_H$

Initial value 00000000_H

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Operation example

(1) To obtain access rights, the mutual-exclusion variable registers are read, and whether the read value is "0" is checked. If the read value is "0", a value other than "0" is written to obtain access rights. (Use the CAXI or SET1 instruction for this processing.)

(2) If the access rights have been acquired by the other CPU (indicated by the read value being other than "0"), a register read check is repeated until the access rights are revoked (and the read value becomes "0").

(3) Reading and writing are performed for the mutual-exclusion resource.

(4) Write "0" to the registers to revoke the access rights. During mutual-exclusion processing using the MEV0 to MEV7 registers, the meaning of the register data can be determined by software, so that mutual-exclusion control with a high degree of freedom can be realized.

(3) Mutual-exclusion control registers 0 to 7 (MEC0 to MEC7)

Each register has its own state machine and manages the CPU that currently has access rights. When one of these registers is read by a CPU, it returns a value according to that CPU. If the semaphore controlled by software supports only three states (acquired by the local CPU, acquired by the other CPU, not acquired by either CPU), the number of processes and instructions required for mutual-exclusion can be reduced using these mutual-exclusion control registers.

These mutual-exclusion control registers are placed on a bus different from that used by the CPUs for internal RAM access, so that even if CPU1 executes polling of the mutual-exclusion control registers, CPU2 can access the internal RAM of CPU1 or CPU2 at the same time. Therefore, degradation in system performance can be prevented, compared to when semaphore variables are placed in general memory resources.

Access These registers can be read or written in 8-bit units.

Address MEC0: $FFFF6980_H$, MEC1: $FFFF6981_H$, MEC2: $FFFF6982_H$, MEC3: $FFFF6983_H$, MEC4: $FFFF6984_H$, MEC5: $FFFF6985_H$, MEC6: $FFFF6986_H$, MEC7: $FFFF6987_H$

Initial value 00_H

7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Read access Tries to obtain access rights. The result is indicated by the read value.

Table 4-13 MEC register contents (when reading)

Read Value	Description	Remark
00_H	Obtained	Indicates that new access rights have been obtained.
01_H	Obtained	Indicates that access rights have already been obtained.
02_H	Failed	The other CPU has obtained access rights.

Write access Revokes access rights. However, writing is ignored if the local CPU has not obtained access rights.

Table 4-14 MEC register contents (when writing)

Write Value	Description	Remark
00_H	Revoke	Revokes access rights.
01_H	Continue	Continues providing access rights.
$0F_H$[a]	Initialize	Forcibly revokes the write access right.
Others	Invalid	The write access is ignored.

a) This register is valid only when $0F_H$ is written to it even if the local CPU has not obtained access rights. (Even if the local CPU has not obtained access rights, forcibly revoking the rights is possible.)

Table 4-15 MEC register states and values read from each CPU

MEC State	State Description	MEMC Value Read from CPU1	MEMC Value Read from CPU2
IDLE	Neither CPU has access rights.	0 (access rights obtained along with the read operation)	0 (access rights obtained along with the read operation)
CPU1	CPU1 has the access rights.	1	2
CPU2	CPU2 has the access rights.	2	1

Operation example

(1) To acquire the access rights, read the mutual-exclusion control registers. “0” is read if the access rights are successfully obtained, “1” is read if the access rights have already been obtained, and “2” is read if the access rights have already been obtained by the other CPU. The access right state or access rights can be obtained just by accessing them by using the ld.b instruction.

(2) If the access rights could not be obtained, repeatedly read the MEC registers until they are obtained.

(3) Reading and writing are performed for the mutual-exclusion resource.

(4) Revoke the access rights by writing to the registers. The access rights can be revoked just by writing "0" by using the st.b instruction. (The rights cannot be revoked by writing "1" or "2".)

The meaning of the register data is defined by hardware.

When performing mutual-exclusion processing by using these data values, high-speed mutual-exclusion control with a small code size is possible by using these registers.

4.9.5 Atomic bus cycle

For the V850E2/MN4, during the execution of the CAXI, SET1, CLR1, or NOT1 instruction of the CPU, the bus path up to the access target of these instructions is locked and access from the other CPU is prohibited. As a result, non-divisible (atomic) execution of the CAXI, SET1, CLR1, and NOT1 instructions is provided.

This means that the CAXI, SET1, CLR1, and NOT1 instructions can be used for mutual-exclusion processing among multiple CPUs.

4.9.6 Inter-PE interrupt function

In the multiprocessor configuration of the V850E2/MN4, it is possible for the CPU of one PE to request that the CPU of the other PE generate an interrupt.

The V850E2/MN4 provides an inter-PE interrupt function through the following two registers:

- MIR0 register: Register requesting that the other PE generate an EIINT0 interrupt
- MIR1 register: Register requesting that the other PE generate an EIINT1 interrupt

The other CPU is requested to generate an EI level maskable interrupt by writing to the MIR0 and MIR1 registers.

The MIR0 and MIR1 registers are placed at the same address as seen from each CPU. By writing to these registers from one CPU, the other CPU is requested to generate an interrupt. A CPU cannot request an interrupt from itself.

If an inter-CPU interrupt is requested from the other CPU, an interrupt of EIINT channel 0 (for a request through MIR0) or EIINT channel 1 (for a request through MIR1) is generated for the CPU.

Caution The inter-PE interrupt request function is supported only for a multiprocessor product (μPD70F3514, 70F3515). It cannot be used for a single processor product (μPD70F3510, 70F3512).

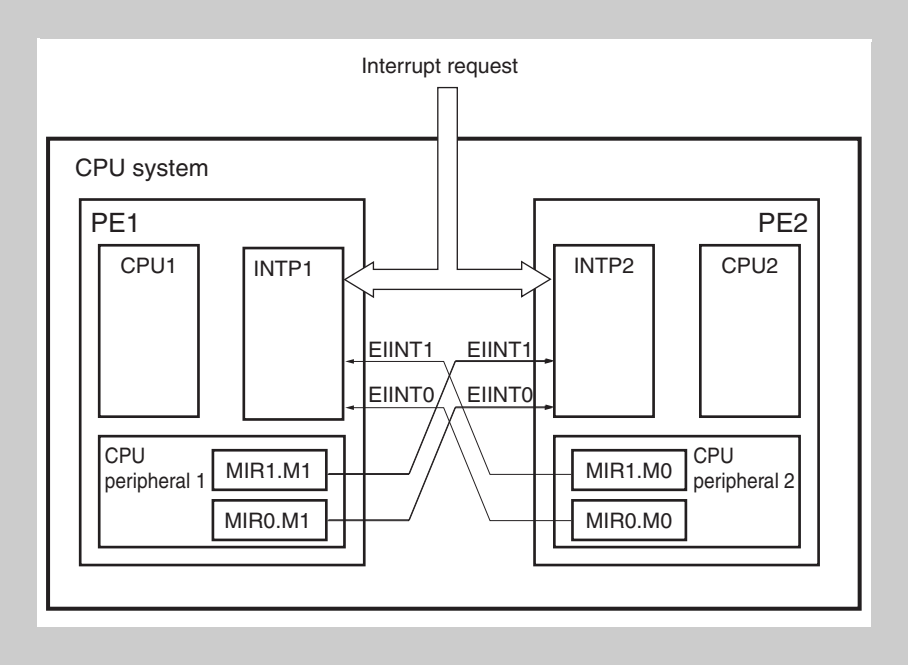


Figure 4-22 Inter-PE interrupt function

(1) Inter-PE interrupt request registers (MIR0, MIR1)

The inter-PE interrupt request registers issue an interrupt request from CPU1 to CPU2 or vice versa. The CPU of which an interrupt is requested generates the interrupt from interrupt channel 0 (for an interrupt request from MIR0) or 1 (for an interrupt request from MIR1).

Access These registers are write-only, in 32-bit units. Note, however, that the MIRn (n = 0, 1) register is written-only in 16-bit units if its lower 16 bits are used as the MIRnL register. When read, 0 is always returned.

Address MIR0: $FFFF6800_H$

MIR1: $FFFF6804_H$

Initial value 00000000_H. These registers are initialized by any reset.

(a) MIR0

31	30	29	28	27	26	25	24
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
7	6	5	4	3	2	1	0
0	0	0	0	0	0	M1[a]	M0[b]
R	R	R	R	R	R	R/W	R/W

a) This can only be set for the MIR0 register of PE1. For PE2, clear this to "0".
b) This can only be set for the MIR0 register of PE2. For PE1, clear this to "0".

(b) MIR1

31	30	29	28	27	26	25	24
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
7	6	5	4	3	2	1	0
0	0	0	0	0	0	M1[a]	M0[b]
R	R	R	R	R	R	R/W	R/W

a) This can only be set for the MIR1 register of PE1. For PE2, clear this to "0".
b) This can only be set for the MIR1 register of PE2. For PE1, clear this to "0".

Table 4-16 MIRn register contents

Bit Position	Bit Name	Description
31:0	MIR[31:0]	PE1 - When “1” is written to the M1 bit of the MIR0 register, PE2 is requested to generate an EIINT0 interrupt. - When “1” is written to the M1 bit of the MIR1 register, PE2 is requested to generate an EIINT1 interrupt. PE2 - When “1” is written to the M0 bit of the MIR0 register, PE1 is requested to generate an EIINT0 interrupt. - When “1” is written to the M0 bit of the MIR1 register, PE1 is requested to generate an EIINT1 interrupt.

4.9.7 Memory synchronization function (SYNCM)

The V850E2/MN4 supports the memory synchronization instruction (SYNCM) defined in the V850E2M architecture. Through this memory synchronization function, the SYNCM instruction can be used as a synchronization primitive, and a memory barrier can be realized through software control.

(1) Memory synchronization manipulation

The following memory barrier manipulations are executed for the V850E2/MN4 when the SYNCM instruction is executed:

- The completion of memory access through load/store instructions already output by the CPU that executed the SYNCM instruction is guaranteed.
- Here, completion of memory access through the load instruction refers to the guarantee that the loaded data has been retrieved by the CPU.
- Similarly, completion of memory access through the store instruction refers to the guarantee that, even if the store data has been accessed by another CPU or DMA subsystem in the V850E2/MN4, the changed data will be referenced as a result of the store instruction.

Notes

1. Load instructions include instructions that read from memory, in addition to the regular load instruction.
2. Store instructions include instructions that write to memory, in addition to the regular store instruction.
3. Referencing the results of store instructions through memory synchronization is guaranteed only for the CPU1, CPU2, and DMA masters. For example, if CPU1 uses a memory synchronization instruction to manipulate the memory barrier, only the CPU1, CPU2, and DMA masters can definitely use the written results of CPU1. Bus masters and other devices connected to other external bus interfaces cannot use these results.

Upon execution of the SYNCM instruction, the CPU stops instruction execution and executes memory barrier manipulations. After completion of the memory barrier manipulations, the CPU resumes its operation and executes subsequent instructions.

(2) Guarantee of access completion for interface function

Through memory synchronization, synchronization of inter-master memory referencing on the internal V850E2/MN4 bus is performed at all times, but whether completion of access to slaves through the interface function is guaranteed depends on each interface.

Actually, whether each interface function of the V850E2/MN4 guarantees the completion of bus access when the memory synchronization instruction (SYNCM) has been executed depends on the function of each interface. For some interfaces, completion of access to slave devices cannot be guaranteed even if the memory synchronization instruction (SYNCM) is executed.

Table 4-17 "Memory synchronization operation results for each interface" shows whether the completion of access through the instructions (load instructions, store instructions, and bit manipulation instructions) accompanying the previously executed memory manipulation can be guaranteed during memory synchronization instruction (SYNCM) execution.

Table 4-17 Memory synchronization operation results for each interface

Area	Access Completion Guarantee
Flash memory area	Yes
CPU system peripheral function area	Yes
CPU-specific peripheral function area	Yes
Internal RAM area	Yes
External memory (primary memory controller) area	Yes
P bus area	Yes
H bus area	No[a]

a) Access completion can be guaranteed through specific actions.

<1> Flushing the H-bus memory side cache

<2> Reading from an area other than the H-bus memory side cache area or H-bus-shared memory area

<3> Reading from H-bus-shared memory bank 0

<4> Reading from H-bus-shared memory bank 1

If access completion must be guaranteed for an interface that does not guarantee access completion with only the memory synchronization instruction, perform the access completion guarantee procedure defined for each interface along with execution of the memory synchronization instruction.

4.9.8 Resources that can also be used in the single-processor configuration

In the single-processor configuration of the V850E2/MN4, the operation of functions supported in the multiprocessor configuration is defined as follows.

Table 4-18 Single-processor configuration functions

Function	μPD70F3514, 70F3515 (Multiprocessor Configuration)	μPD70F3510, 70F3512 (Single-Processor Configuration)	Remark
CPU local RAM access function	Available	Not available	Because no other CPU exists
Processor number report function	Available	Available	
Atomic bus cycle guarantee (mutual-exclusion processing instruction)	Available	Available	
Mutual-exclusion variable registers (MEV registers)	Available	Not available	
Mutual-exclusion control registers (MEC registers)	Available	Not available	
Memory synchronization function (SYNCM instruction)	Available	Available	
Inter-PE interrupt function	Available	Not available	Because no other CPU exists
Inter-PE interrupt register (MIR0, MIR1)	Available	Not available	Because no other CPU exists

4.10 Peripheral Protection Function

4.10.1 Peripheral-device protection function (PPU) support

Because registers placed in a P-bus area perform PPU control, a one-PPU area is assigned for each function macro group.

Note that the provided registers are divided and placed in the OS and user areas according to the register specifications. (For details about the mapping of division placement, see the mapping of each P-bus area.)

The registers that can be accessed are fixed according to the placement of divided areas. In other words, a register in a user area cannot be accessed from within a P-bus area in which only an OS area is placed. Conversely, a register in an OS area cannot be accessed from within a P-bus area in which only a user area is placed.

Because PPU settings can be individually specified for each P-bus area, the PPU function can be efficiently set up by specifying "special access" only for P-bus areas in which OS areas are placed.

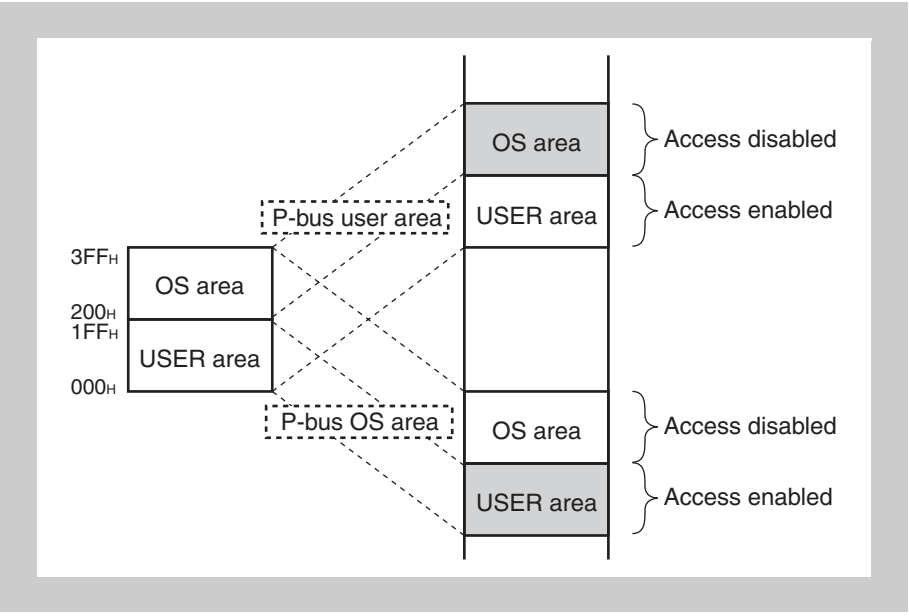


Figure 4-23 PPU control example (for TAUA)

PPU base address

The addresses of registers related to peripheral protection functions described in the "V850E2M Architecture User's Manual" are defined as offset addresses. The base address is as follows:

<PPU_base> = FFFF 5100_H

PPU areas and registers

The control registers for each protection area include the following four registers:

- PPVn: This is used to specify whether to enable general peripheral device protection.
- PPTn: This is used to specify the type of protection for general peripheral devices.
- PPPn: This is used to specify settings for OS peripheral devices.
- PPSn: This is used to specify settings for special peripheral devices.

For these registers, n = 0 to 8.

These 32-bit registers have the following bits (where m = 0 to 31):

- PPVn.PPVnm
- PPTn.PPTnm
- PPPn.PPPnm
- PPSn.PPSnm

Table 4-19 "PPU protection areas and protection targets" shows the protection address ranges, the control registers and bits, and the respective area names.

Table 4-19 PPU protection areas and protection targets (1/9)

Protection Control Registers		Address Range	Placement Function Name	Protection Range
PPVn, PPTn, PPPn, PPSn registers n =	PPVnm, PPTnm, PPPnm, PPSnm bits m =			
0	0	FFFF 6000$_H$ - FFFF 645F$_H$	INTC	1120 B
	1	FFFF 6480$_H$ - FFFF 6487$_H$	Flash cache control	8 B
	2	Reserved		
	3	FFFF 64B0$_H$ - FFFF 64B3$_H$	SysErrGen	3 B
	4	Reserved		
	5	Reserved		
	6	Reserved		
	7	FFFF 6700$_H$ - FFFF 67FF$_H$	Reserved (Debugging function suitable for a development tool)	3 B
	8	FFFF 6800$_H$ - FFFF 687F$_H$	Inter-PE interrupts	128 B
	9	FFFF 6900$_H$ - FFFF 697F$_H$	Mutual-exclusion variable registers	128 B
	10	FFFF 6980$_H$ - FFFF 699F$_H$	Mutual-exclusion control registers	160 B
	11	FFFF 69A0$_H$ - FFFF 69BF$_H$	PE guard	32 B
	12	Reserved		
	13	Reserved		
	14	Reserved		
	15	Reserved		
	16	FFFF 5000$_H$ - FFFF53FF$_H$	Processor protection (PPU/TSU)	1 K
	17	Reserved		
	18	Reserved		
	19	Reserved		
	20	Reserved		
	21	FFFF 7100$_H$ - FFFF71FF$_H$	H-bus interface control	8 B
	22	FFFF 7200$_H$ - FFFF 72FF$_H$	Primary memory controller	256 B
	23	FFFF 7300$_H$ - FFFF 73FF$_H$	DMAC	256 B
	24	FFFF 7400$_H$ - FFFF 74FF$_H$	DMAC	256 B
	25	FFFF 7500$_H$ - FFFF 76FF$_H$	DMAC	512 B
	26	FFFF 7700$_H$ - FFFF 78FF$_H$	DMAC	512 B
	27	FFFF 7900$_H$ - FFFF 7AFF$_H$	DMAC	512 B
	28	FFFF 7B00$_H$ - FFFF 7CFF$_H$	DMAC/DTS	512 B
	29	FFFF 7D00$_H$ - FFFF 7EFF$_H$	DTS	512 B
	30	Reserved		
	31	FFFF 7F80$_H$ - FFFF 7EFF$_H$	H-bus interface control	128 B

Table 4-19 PPU protection areas and protection targets (2/9)

Protection Control Registers		Address Range	Placement Function Name	Protection Range
PPVn, PPTn, PPPn, PPSn registers n =	PPVnm, PPTnm, PPPnm, PPSnm bits m =			
1	0	FF40 0000$_H$ - FF40 FFFF$_H$	Port Pnm control	64 KB
	1	FF41 0000$_H$ - FF41 FFFF$_H$	Port filter control	
	2	FF42 0000$_H$ - FF42 FFFF$_H$	Clock generator reset circuit	
	3	FF43 0000$_H$ - FF43 FFFF$_H$	Reserved	
	4	FF44 0000$_H$ - FF44 FFFF$_H$	Reserved	
	5	FF45 0000$_H$ - FF45 FFFF$_H$	Reserved	
	6	FF46 0000$_H$ - FF46 FFFF$_H$	Reserved	
	7	FF47 0000$_H$ - FF47 FFFF$_H$	Reserved	
	8	FF48 0000$_H$ - FF48 FFFF$_H$	FCN0	
	9	FF49 0000$_H$ - FF49 FFFF$_H$		
	10	FF4A 0000$_H$ - FF4A FFFF$_H$	FCN1	
	11	FF4B 0000$_H$ - FF4B FFFF$_H$		
	12	FF4C 0000$_H$ - FF4C FFFF$_H$	Reserved	
	13	FF4D 0000$_H$ - FF4D FFFF$_H$	Reserved	
	14	FF4E 0000$_H$ - FF4E FFFF$_H$	Reserved	
	15	FF4F 0000$_H$ - FF4F FFFF$_H$	Reserved	
	16	FF50 0000$_H$ - FF50 FFFF$_H$	Reserved	
	17	FF51 0000$_H$ - FF51 FFFF$_H$	Reserved	
	18	FF52 0000$_H$ - FF52 FFFF$_H$	Reserved	
	19	FF53 0000$_H$ - FF53 FFFF$_H$	Reserved	
	20	FF54 0000$_H$ - FF54 FFFF$_H$	Reserved	
	21	FF55 0000$_H$ - FF55 FFFF$_H$	Reserved	
	22	FF56 0000$_H$ - FF56 FFFF$_H$	Reserved	
	23	FF57 0000$_H$ - FF57 FFFF$_H$	Reserved	
	24	FF58 0000$_H$ - FF58 FFFF$_H$	Reserved	
	25	FF59 0000$_H$ - FF59 FFFF$_H$	Reserved	
	26	FF5A 0000$_H$ - FF5A FFFF$_H$	Reserved	
	27	FF5B 0000$_H$ - FF5B FFFF$_H$	Reserved	
	28	FF5C 0000$_H$ - FF5C FFFF$_H$	UARTJ0	
	29	FF5D 0000$_H$ - FF5D FFFF$_H$	UARTJ1	
	30	FF5E 0000$_H$ - FF5E FFFF$_H$	UARTJ2	
	31	FF5F 0000$_H$ - FF5F FFFF$_H$	UARTJ3	

Table 4-19 PPU protection areas and protection targets (3/9)

Protection Control Registers		Address Range	Placement Function Name	Protection Range
PPVn, PPTn, PPPn, PPSn registers n =	PPVnm, PPTnm, PPPnm, PPSnm bits m =			
2	0	FF60 0000$_H$ - FF60 FFFF$_H$	UARTE0	64 KB
	1	FF61 0000$_H$ - FF61 FFFF$_H$	UARTE1	
	2	FF62 0000$_H$ - FF62 FFFF$_H$	UARTE2	
	3	FF63 0000$_H$ - FF63 FFFF$_H$	UARTE3	
	4	FF64 0000$_H$ - FF64 FFFF$_H$	UARTE4	
	5	FF65 0000$_H$ - FF65 FFFF$_H$	UARTE5	
	6	FF66 0000$_H$ - FF66 FFFF$_H$	Reserved	
	7	FF67 0000$_H$ - FF67 FFFF$_H$	Reserved	
	8	FF68 0000$_H$ - FF68 FFFF$_H$	Reserved	
	9	FF69 0000$_H$ - FF69 FFFF$_H$	Reserved	
	10	FF6A0000$_H$ - FF6A FFFF$_H$	Reserved	
	11	FF6B 0000$_H$ - FF6B FFFF$_H$	Reserved	
	12	FF6C 0000$_H$ - FF6C FFFF$_H$	CSIH0	
	13	FF6D 0000$_H$ - FF6D FFFF$_H$	CSIH1	
	14	FF6E 0000$_H$ - FF6E FFFF$_H$	CSIH2	
	15	FF6F 0000$_H$ - FF6F FFFF$_H$	CSIH3	
	16	FF70 0000$_H$ - FF70 FFFF$_H$	CSIG0	
	17	FF71 0000$_H$ - FF71 FFFF$_H$	CSIG1	
	18	FF72 0000$_H$ - FF72 FFFF$_H$	CSIG2	
	19	FF73 0000$_H$ - FF73 FFFF$_H$	CSIG3	
	20	FF74 0000$_H$ - FF74 FFFF$_H$	CSIG4	
	21	FF75 0000$_H$ - FF75 FFFF$_H$	CSIG5	
	22	FF76 0000$_H$ - FF76 FFFF$_H$	Reserved	
	23	FF77 0000$_H$ - FF77 FFFF$_H$	Reserved	
	24	FF78 0000$_H$ - FF78 FFFF$_H$	Reserved	
	25	FF79 0000$_H$ - FF79 FFFF$_H$	Reserved	
	26	FF7A 0000$_H$ - FF7A FFFF$_H$	Reserved	
	27	FF7B 0000$_H$ - FF7B FFFF$_H$	Reserved	
	28	FF7C 0000$_H$ - FF7C FFFF$_H$	Reserved	
	29	FF7D 0000$_H$ - FF7D FFFF$_H$	Reserved	
	30	FF7E 0000$_H$ - FF7E FFFF$_H$	Reserved	
	31	FF7F 0000$_H$ - FF7F FFFF$_H$	Reserved	

Table 4-19 PPU protection areas and protection targets (4/9)

Protection Control Registers		Address Range	Placement Function Name	Protection Range
PPVn, PPTn, PPPn, PPSn registers n =	PPVnm, PPTnm, PPPnm, PPSnm bits m =			
3	0	FF80 0000$_H$ - FF80 0FFF$_H$	OSTM0	4 KB
	1	FF80 1000$_H$ - FF80 1FFF$_H$	OSTM1	
	2	FF80 2000$_H$ - FF80 2FFF$_H$	Reserved	
	3	FF80 3000$_H$ - FF80 3FFF$_H$	Reserved	
	4	FF80 4000$_H$ - FF80 4FFF$_H$	Reserved	
	5	FF80 5000$_H$ - FF80 5FFF$_H$	Reserved	
	6	FF80 6000$_H$ - FF80 6FFF$_H$	Reserved	
	7	FF80 7000$_H$ - FF80 7FFF$_H$	Reserved	
	8	FF80 8000$_H$ - FF80 8FFF$_H$	TAUA0	
	9	FF80 9000$_H$ - FF80 9FFF$_H$	TAUA1	
	10	FF80 A000$_H$ - FF80 AFFF$_H$	TAUA2	
	11	FF80 B000$_H$ - FF80 BFFF$_H$	TAUA3	
	12	FF80 C000$_H$ - FF80 CFFF$_H$	Reserved	
	13	FF80 D000$_H$ - FF80 DFFF$_H$	Reserved	
	14	FF80 E000$_H$ - FF80 EFFF$_H$	Reserved	
	15	FF80 F000$_H$ - FF80 FFFF$_H$	Reserved	
	16	FF81 0000$_H$ - FF81 0FFF$_H$	Reserved	
	17	FF81 1000$_H$ - FF81 1FFF$_H$	TAUJ0	
	18	FF81 2000$_H$ - FF81 2FFF$_H$	Reserved	
	19	FF81 3000$_H$ - FF81 3FFF$_H$	Reserved	
	20	FF81 4000$_H$ - FF81 4FFF$_H$	Reserved	
	21	FF81 5000$_H$ - FF81 5FFF$_H$	TAPA0	
	22	FF81 6000$_H$ - FF81 6FFF$_H$	TAPA1	
	23	FF81 7000$_H$ - FF81 7FFF$_H$	TAPA2	
	24	FF81 8000$_H$ - FF81 8FFF$_H$	TAPA3	
	25	FF81 9000$_H$ - FF81 9FFF$_H$	ENCA0	
	26	FF81 A000$_H$ - FF81 AFFF$_H$	ENCA1	
	27	FF81 B000$_H$ - FF81 BFFF$_H$	Reserved	
	28	FF81 C000$_H$ - FF81 CFFF$_H$	Reserved	
	29	FF81 D000$_H$ - FF81 DFFF$_H$	ADCA0	
	30	FF81 E000$_H$ - FF81 EFFF$_H$	Reserved	
	31	FF81 F000$_H$ - FF81 FFFF$_H$	Reserved	

Table 4-19 PPU protection areas and protection targets (5/9)

Protection Control Registers		Address Range	Placement Function Name	Protection Range
PPVn, PPTn, PPPn, PPSn registers n =	PPVnm, PPTnm, PPPnm, PPSnm bits m =			
4	0	FF82 0000$_H$ - FF82 0FFF$_H$	IICB0	4 KB
	1	FF82 1000$_H$ - FF82 1FFF$_H$	IICB1	
	2	FF82 2000$_H$ - FF82 2FFF$_H$	IICB2	
	3	FF82 3000$_H$ - FF82 3FFF$_H$	IICB3	
	4	FF82 4000$_H$ - FF82 4FFF$_H$	IICB4	
	5	FF82 5000$_H$ - FF82 5FFF$_H$	IICB5	
	6	FF82 6000$_H$ - FF82 6FFF$_H$	Reserved	
	7	FF82 7000$_H$ - FF82 7FFF$_H$	Reserved	
	8	FF82 8000$_H$ - FF82 8FFF$_H$	TAUA0	
	9	FF82 9000$_H$ - FF82 9FFF$_H$	TAUA1	
	10	FF82 A000$_H$ - FF82 AFFF$_H$	TAUA2	
	11	FF82 B000$_H$ - FF82 BFFF$_H$	TAUA3	
	12	FF82 C000$_H$ - FF82 CFFF$_H$	Reserved	
	13	FF82 D000$_H$ - FF82 DFFF$_H$	Reserved	
	14	FF82 E000$_H$ - FF82 EFFF$_H$	Reserved	
	15	FF82 F000$_H$ - FF82 FFFF$_H$	Reserved	
	16	FF83 0000$_H$ - FF83 0FFF$_H$	Reserved	
	17	FF83 1000$_H$ - FF83 1FFF$_H$	TAUJ0	
	18	FF83 2000$_H$ - FF83 2FFF$_H$	Reserved	
	19	FF83 3000$_H$ - FF83 3FFF$_H$	Reserved	
	20	FF83 4000$_H$ - FF83 4FFF$_H$	Reserved	
	21	FF83 5000$_H$ - FF83 5FFF$_H$	TAPA0	
	22	FF83 6000$_H$ - FF83 6FFF$_H$	TAPA1	
	23	FF83 7000$_H$ - FF83 7FFF$_H$	TAPA2	
	24	FF83 8000$_H$ - FF83 8FFF$_H$	TAPA3	
	25	FF83 9000$_H$ - FF83 9FFF$_H$	ENCA0	
	26	FF83 A000$_H$ - FF83 AFFF$_H$	ENCA1	
	27	FF83 B000$_H$ - FF83 BFFF$_H$	Reserved	
	28	FF83 C000$_H$ - FF83 CFFF$_H$	Reserved	
	29	FF83 D000$_H$ - FF83 DFFF$_H$	ADCA0	
	30	FF83 E000$_H$ - FF83 EFFF$_H$	Reserved	
	31	FF83 F000$_H$ - FF83 FFFF$_H$	Reserved	

Table 4-19 PPU protection areas and protection targets (6/9)

Protection Control Registers		Address Range	Placement Function Name	Protection Range
PPVn, PPTn, PPPn, PPSn registers n =	PPVnm, PPTnm, PPPnm, PPSnm bits m =			
5	0	FFFF 8000_H - FFFF $80FF_H$	Port	256 B
	1	FFFF 8100_H - FFFF $81FF_H$	Port	
	2	FFFF 8200_H - FFFF $82FF_H$	Port	
	3	FFFF 8300_H - FFFF $83FF_H$	Port	
	4	FFFF 8400_H - FFFF $84FF_H$	Port	
	5	FFFF 8500_H - FFFF $85FF_H$	Port	
	6	FFFF 8600_H - FFFF $86FF_H$	Port	
	7	FFFF 8700_H - FFFF $87FF_H$	Port	
	8	FFFF 8800_H - FFFF $88FF_H$	Port	
	9	FFFF 8900_H - FFFF $89FF_H$	Port	
	10	FFFF $8A00_H$ - FFFF $8AFF_H$	Port	
	11	FFFF $8B00_H$ - FFFF $8BFF_H$	Port	
	12	FFFF $8C00_H$ - FFFF $8CFF_H$	Port	
	13	FFFF $8D00_H$ - FFFF $8DFF_H$	Port	
	14	FFFF $8E00_H$ - FFFF $8EFF_H$	Port	
	15	FFFF $8F00_H$ - FFFF $8FFF_H$	Port	
	16	FFFF 9000_H - FFFF $90FF_H$	Port	
	17	FFFF 9100_H - FFFF $91FF_H$	Port	
	18	FFFF 9200_H - FFFF $92FF_H$	Port	
	19	FFFF 9300_H - FFFF $93FF_H$	Port	
	20	FFFF 9400_H - FFFF $94FF_H$	Port	
	21	FFFF 9500_H - FFFF $95FF_H$	Port	
	22	FFFF 9600_H - FFFF $96FF_H$	Port	
	23	FFFF 9700_H - FFFF $97FF_H$	Port	
	24	FFFF 9800_H - FFFF $98FF_H$	Port	
	25	FFFF 9900_H - FFFF $99FF_H$	Port	
	26	FFFF $9A00_H$ - FFFF $9AFF_H$	Port	
	27	FFFF $9B00_H$ - FFFF $9BFF_H$	Port	
	28	FFFF $9C00_H$ - FFFF $9CFF_H$	Port	
	29	FFFF $9D00_H$ - FFFF $9DFF_H$	Port	
	30	FFFF $9E00_H$ - FFFF $9EFF_H$	Port	
	31	FFFF $9F00_H$ - FFFF $9FFF_H$	Port	

Table 4-19 PPU protection areas and protection targets (7/9)

Protection Control Registers		Address Range	Placement Function Name	Protection Range
PPVn, PPTn, PPPn, PPSn registers n =	PPVnm, PPTnm, PPPnm, PPSnm bits m =			
6	0	FFFF A000$_H$ - FFFF A0FF$_H$	Port	256 B
	1	FFFF A100$_H$ - FFFF A1FF$_H$	Port	
	2	FFFF A200$_H$ - FFFF A2FF$_H$	Port	
	3	FFFF A300$_H$ - FFFF A3FF$_H$	Port	
	4	FFFF A400$_H$ - FFFF A4FF$_H$	Port	
	5	FFFF A500$_H$ - FFFF A5FF$_H$	Port	
	6	FFFF A600$_H$ - FFFF A6FF$_H$	Port	
	7	FFFF A700$_H$ - FFFF A7FF$_H$	Port	
	8	FFFF A800$_H$ - FFFF A8FF$_H$	Port	
	9	FFFF A900$_H$ - FFFF A9FF$_H$	Port	
	10	FFFF AA00$_H$ - FFFF AAFF$_H$	Port	
	11	FFFF AB00$_H$ - FFFF ABFF$_H$	Port	
	12	FFFF AC00$_H$ - FFFF ACFF$_H$	Port	
	13	FFFF AD00$_H$ - FFFF ADFF$_H$	Port	
	14	FFFF AE00$_H$ - FFFF AEFF$_H$	Port	
	15	FFFF AF00$_H$ - FFFF AFFF$_H$	Port	
	16	FFFF B000$_H$ - FFFF B0FF$_H$	Port	
	17	FFFF B100$_H$ - FFFF B1FF$_H$	Port	
	18	FFFF B200$_H$ - FFFF B2FF$_H$	Port	
	19	FFFF B300$_H$ - FFFF B3FF$_H$	Port	
	20	FFFF B400$_H$ - FFFF B4FF$_H$	Port	
	21	FFFF B500$_H$ - FFFF B5FF$_H$	Port	
	22	FFFF B600$_H$ - FFFF B6FF$_H$	Port	
	23	FFFF B700$_H$ - FFFF B7FF$_H$	Port	
	24	FFFF B800$_H$ - FFFF B8FF$_H$	Port	
	25	FFFF B900$_H$ - FFFF B9FF$_H$	Port	
	26	FFFF BA00$_H$ - FFFF BAFF$_H$	Port	
	27	FFFF BB00$_H$ - FFFF BBFF$_H$	Port	
	28	FFFF BC00$_H$ - FFFF BCFF$_H$	Port	
	29	FFFF BD00$_H$ - FFFF BDFF$_H$	Port	
	30	FFFF BE00$_H$ - FFFF BEFF$_H$	Port	
	31	FFFF BF00$_H$ - FFFF BFFF$_H$	Port	

Table 4-19 PPU protection areas and protection targets (8/9)

Protection Control Registers		Address Range	Placement Function Name	Protection Range
PPVn, PPTn, PPPn, PPSn registers n =	PPVnm, PPTnm, PPPnm, PPSnm bits m =			
7	0	FFFF C000$_H$ - FFFF C0FF$_H$	WDTA0	256 B
	1	FFFF C100$_H$ - FFFF C1FF$_H$	WDAT1	
	2	FFFF C200$_H$ - FFFF C2FF$_H$	TAUJ0	
	3	FFFF C300$_H$ - FFFF C3FF$_H$	Reserved	
	4	FFFF C400$_H$ - FFFF C4FF$_H$	TAUA0	
	5	FFFF C500$_H$ - FFFF C5FF$_H$	TAUA0	
	6	FFFF C600$_H$ - FFFF C6FF$_H$	TAUA0	
	7	FFFF C700$_H$ - FFFF C7FF$_H$	TAUA0	
	8	FFFF C800$_H$ - FFFF C8FF$_H$	TAUA1	
	9	FFFF C900$_H$ - FFFF C9FF$_H$	TAUA1	
	10	FFFF CA00$_H$ - FFFF CAFF$_H$	TAUA1	
	11	FFFF CB00$_H$ - FFFF CBFF$_H$	TAUA1	
	12	FFFF CC00$_H$ - FFFF CCFF$_H$	TAUA2	
	13	FFFF CD00$_H$ - FFFF CDFF$_H$	TAUA2	
	14	FFFF CE00$_H$ - FFFF CEFF$_H$	TAUA2	
	15	FFFF CF00$_H$ - FFFF CFFF$_H$	TAUA2	
	16	FFFF D000$_H$ - FFFF D0FF$_H$	TAUA3	
	17	FFFF D100$_H$ - FFFF D1FF$_H$	TAUA3	
	18	FFFF D200$_H$ - FFFF D2FF$_H$	TAUA3	
	19	FFFF D300$_H$ - FFFF D3FF$_H$	TAUA3	
	20	FFFF D400$_H$ - FFFF D4FF$_H$	TAPA0	
	21	FFFF D500$_H$ - FFFF D5FF$_H$	TAPA1	
	22	FFFF D600$_H$ - FFFF D6FF$_H$	TAPA2	
	23	FFFF D700$_H$ - FFFF D7FF$_H$	TAPA3	
	24	FFFF D800$_H$ - FFFF D8FF$_H$	ENCA0	
	25	FFFF D900$_H$ - FFFF D9FF$_H$	ENCA1	
	26	FFFF DA00$_H$ - FFFF DAFF$_H$	Reserved	
	27	FFFF DB00$_H$ - FFFF DBFF$_H$	Reserved	
	28	FFFF DC00$_H$ - FFFF DCFF$_H$	ADCA0	
	29	FFFF DD00$_H$ - FFFF DDFF$_H$	ADCA0	
	30	FFFF DE00$_H$ - FFFF DEFF$_H$	Reserved	
	31	FFFF DF00$_H$ - FFFF DFFF$_H$	Reserved	

Table 4-19 PPU protection areas and protection targets (9/9)

Protection Control Registers		Address Range	Placement Function Name	Protection Range
PPVn, PPTn, PPPn, PPSn registers n =	PPVnm, PPTnm, PPPnm, PPSnm bits m =			
8	0	FFFF E000$_H$ - FFFF E0FF$_H$	CSIH0	256 B
	1	FFFF E100$_H$ - FFFF E1FF$_H$	CSIH1	
	2	FFFF E200$_H$ - FFFF E2FF$_H$	CSIH2	
	3	FFFF E300$_H$ - FFFF E3FF$_H$	CSIH3	
	4	FFFF E400$_H$ - FFFF E4FF$_H$	CSIG0	
	5	FFFF E500$_H$ - FFFF E5FF$_H$	CSIG1	
	6	FFFF E600$_H$ - FFFF E6FF$_H$	CSIG2	
	7	FFFF E700$_H$ - FFFF E7FF$_H$	CSIG3	
	8	FFFF E800$_H$ - FFFF E8FF$_H$	CSIG4	
	9	FFFF E900$_H$ - FFFF E9FF$_H$	CSIG5	
	10	FFFF EA00$_H$ - FFFF EAFF$_H$	UARTJ0	
	11	FFFF EB00$_H$ - FFFF EBFF$_H$	UARTJ1	
	12	FFFF EC00$_H$ - FFFF ECFF$_H$	UARTJ2	
	13	FFFF ED00$_H$ - FFFF EDFF$_H$	UARTJ3	
	14	FFFF EE00$_H$ - FFFF EEFF$_H$	UARTE0	
	15	FFFF EF00$_H$ - FFFF EFFF$_H$	UARTE1	
	16	FFFF F000$_H$ - FFFF F0FF$_H$	UARTE2	
	17	FFFF F100$_H$ - FFFF F1FF$_H$	UARTE3	
	18	FFFF F200$_H$ - FFFF F2FF$_H$	UARTE4	
	19	FFFF F300$_H$ - FFFF F3FF$_H$	UARTE5	
	20	FFFF F400$_H$ - FFFF F4FF$_H$	IICB0	
	21	FFFF F500$_H$ - FFFF F5FF$_H$	IICB1	
	22	FFFF F600$_H$ - FFFF F6FF$_H$	IICB2	
	23	FFFF F700$_H$ - FFFF F7FF$_H$	IICB3	
	24	FFFF F800$_H$ - FFFF F8FF$_H$	IICB4	
	25	FFFF F900$_H$ - FFFF F9FF$_H$	IICB5	
	26	FFFF FA00$_H$ - FFFF FAFF$_H$	Reserved	
	27	FFFF FB00$_H$ - FFFF FBFF$_H$	Reserved	
	28	FFFF FC00$_H$ - FFFF FCFF$_H$	Reserved	
	29	FFFF FD00$_H$ - FFFF FDFF$_H$	Reserved	
	30	FFFF FE00$_H$ - FFFF FEFF$_H$	Reserved	
	31	FFFF FF00$_H$ - FFFF FFFF$_H$	Reserved	

4.11 Timing Monitoring Function

TSU base address

The addresses of registers related to the timing monitoring function described in the "V850E2M Architecture User's Manual" are defined as offset addresses. The base address is as follows:

The base address is as follows:

<TSU_base> = FFFF 5000_H

4.12 Other Protection Functions

4.12.1 PE guard function

The PE guard protection function is provided for when the multiprocessor configuration is used.

The V850E2/MN4 supports setting up two spaces for which writing to the internal RAM of one PE from the other can be enabled or disabled. These two spaces can be set as either access-prohibited spaces or access-enabled spaces.

For PE1, this can be set only for writing from PE2 to internal RAM 1.

For PE2, this can be set only for writing from PE1 to internal RAM 2.

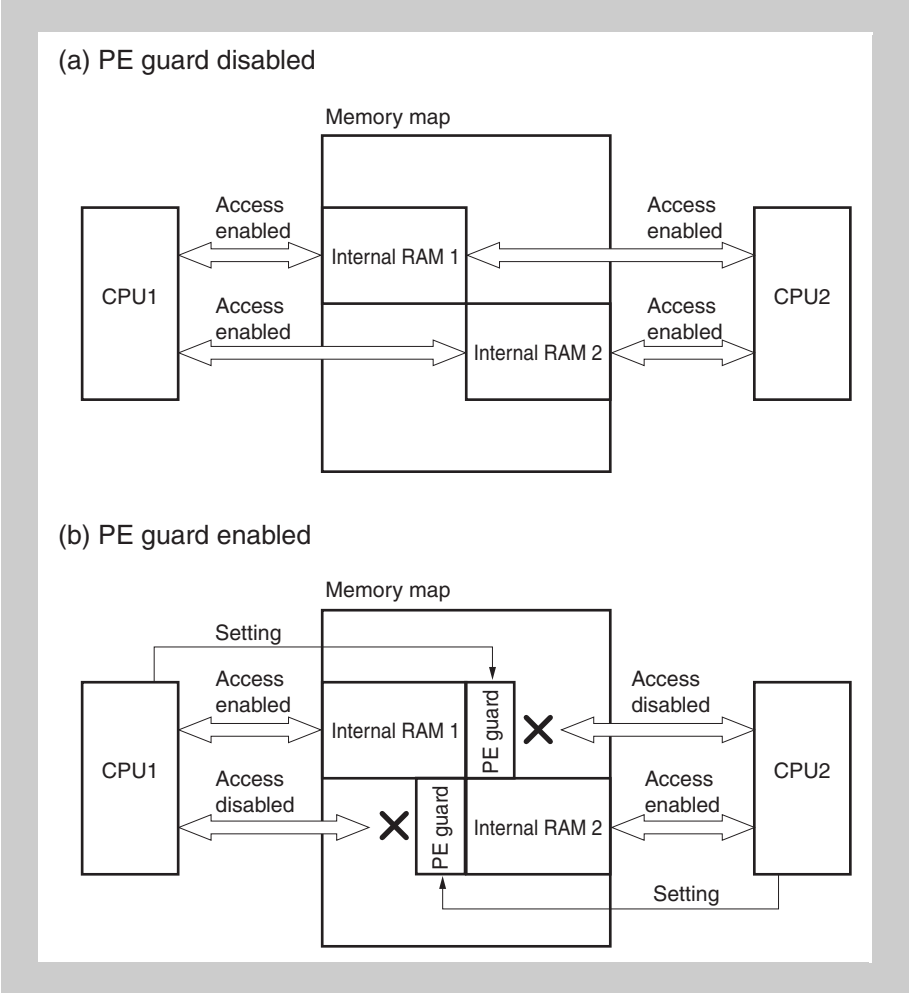


Figure 4-24 PE guard protection

If this setting is accessed and a violation is detected, the following occur:

- EIINT2 is issued to the CPU that specified the setting.
- The SYSERR exception is issued to the CPU that committed the violation.[c]

Note Remark The occurrence of violations can also be ascertained by referencing PEGEC.PEO0 and PEGEC.PEO1.

(1) Specification of guard space for PE guard

Two spaces to be guarded can be defined for each CPU (for each internal RAM). Two types, one that enables access and another that disables it, can be selected. Specification is done by using the PEGSR.GAASE bit.

c) SYSERR exception generation can be masked. For details, see *4.12.3 "System error report function"*.

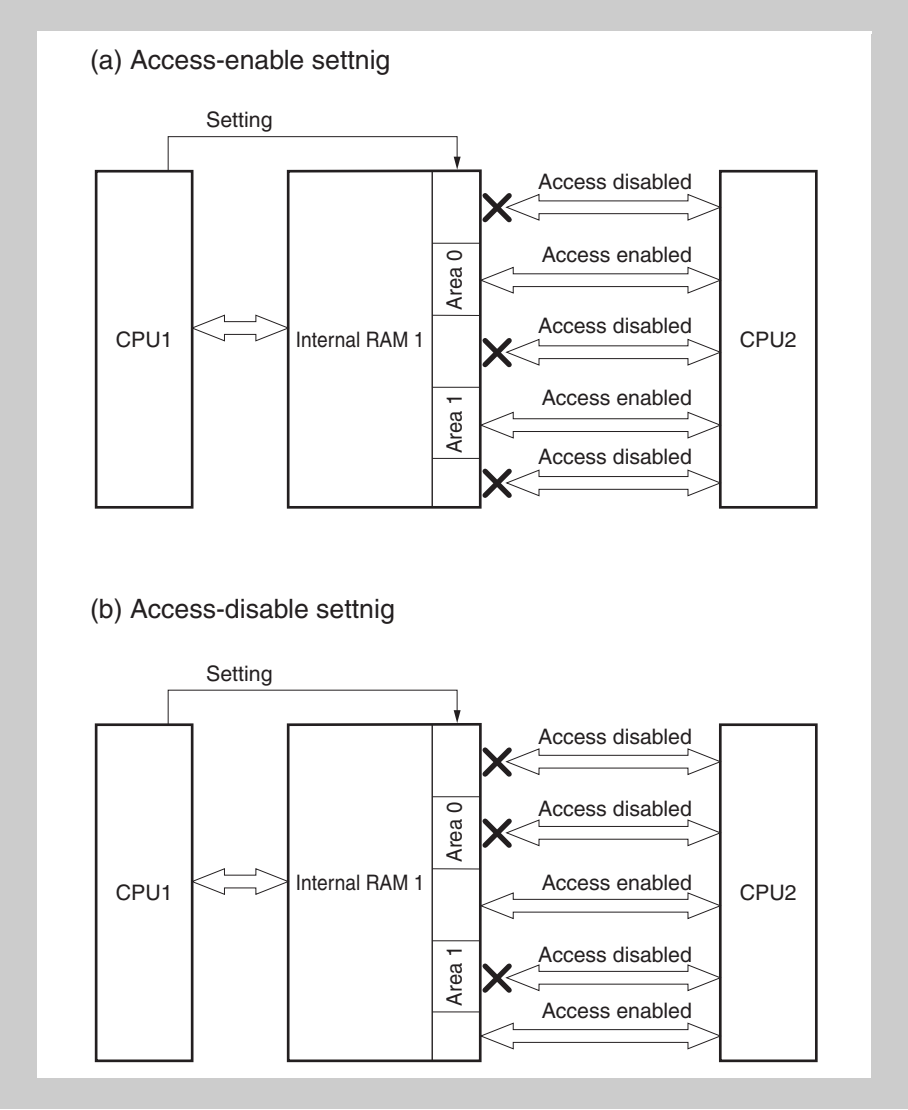


Figure 4-25 Space specification

The locations of areas 0 and 1 are specified by using the following base address specification register and mask address specification register.

For simultaneous execution of base address specification and mask address specification, these registers are set by way of the base address shadow register (PEGSA) and the mask address shadow register (PEGSM).

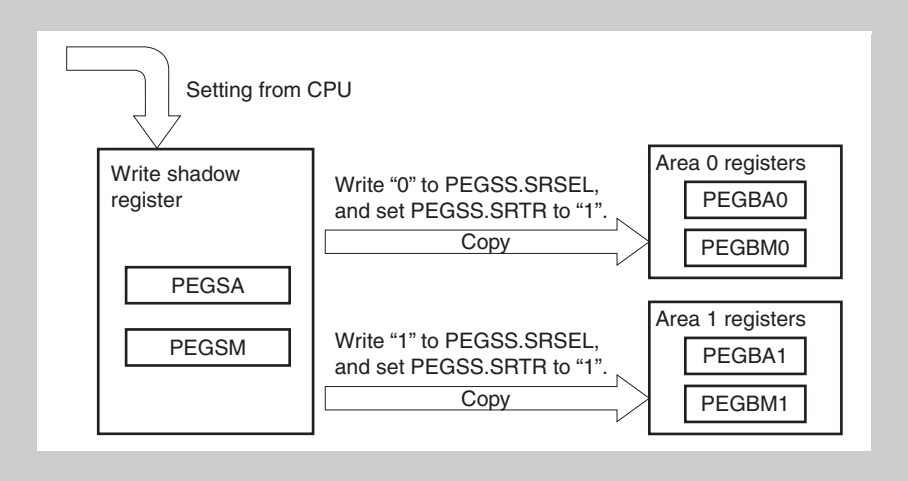


Figure 4-26 Transfer from shadow register

Caution To specify the disabled access type

When specifying the address, specify a bit pattern whose bit positions higher than the size of the internal RAM for each CPU have been set (1) (mask specification) for the mask address register.

(2) 9.3.2 Starting PE guard

To set up the spaces to be guarded and start guarding, perform the following procedure:

<1> Specify the base address for the shadow register PEGSB and the mask address for the shadow register PEGSM.

<2> Specify the number (0 or 1) of the space to be transferred for PEGSS.SRSEL, and set PEGSS.SRTR to 1. As a result, the value is transferred to PEGBA0 (or PEGBA1) and PEGMA0 (or PEGMA1).

<3> Repeat steps <1> and <2> above once for each channel to be set.

<4> Specify the protection policy setting for PEGSR.GAASE (1: Disable, 0: Enable), and set PEGSR.GAWEN0 and PEGSR.GAWEN1 to 1.

(3) Stopping PE guard

To stop PE guard protection, perform the following procedure:

- Clear PEGSR.GAWEN1 to 0.
- Clear PEGSR.GAWEN0 to 0.
- Clear PEGSR.GAASE to 0.

(4) PE guard setting registers

These registers are used to set up the PE guard function.

(a) PE guard function setting register (PEGSR)

This register is used to set up the PE guard function.

Access This register can be read or written in 16-bit units. Note, however, that the PEGSR register can be read and written in 8-bit units if its upper 8 bits and lower 8 bits are used as the PEGSRH and PEGSRL registers, respectively.

Address $FFFF69A0_H$

Initial value 0000_H. This register is initialized by any reset.

Caution Be sure to clear bits 14 to 2 to "0".

15	14	13	12	11	10	9	8
GAASE	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R
7	6	5	4	3	2	1	0
0	0	0	0	0	0	GAWEN1	GAWEN0
R	R	R	R	R	R	R/W	R/W

Table 4-20 PEGSR register contents

Bit Position	Bit Name	Description
15	GAASE	This bit specifies the PE guard area attribute. 0: Disables guard areas 0 and 1, and enables the other areas (initial value). 1: Enables guard areas 0 and 1, and disables the other areas.
1	GAWEN1	This bit enables or disables guarding of the area indicated by guard area 1. 0: Disabled (initial value) 1: Enabled
0	GAWEN0	This bit enables or disables guarding of the area indicated by guard area 0. 0: Disabled (initial value) 1: Enabled

Note The PE guard function is invalid when GAASE = 0, GAWEN1 = 0, and GAWEN0 = 0.

(b) PE guard area setting register (PEGSS)

This register is used to specify settings related to the guard area of the PE guard function.

Access This register can be read or written in 16-bit units. Note, however, that the PEGSS register can be read and written in 8-bit units if its upper 8 bits and lower 8 bits are used as the PEGSSH and PEGSSL registers, respectively.

Address $FFFF69A2_H$

Initial value 0000_H. This register is initialized by any reset.

Caution Be sure to clear bits 15 to 9 and 7 to 1 to “0”.

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	SRSEL
R	R	R	R	R	R	R	R/W
7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	GAWEN0
R	R	R	R	R	R	R	R/W

Table 4-21 PEGSS register contents

Bit Position	Bit Name	Description
8	SRSEL	This bit specifies the PEGBA and PEGMA channel to which to transfer the contents saved in the shadow register. 0: Channel 0 (initial value) 1: Channel 1
1	STSR	This bit is used when transferring the contents saved to the shadow register to PEGBAx and PEGMAx. This bit is automatically cleared to 0 after a transfer (and the value of x is determined by the contents of SRSEL). 0: Disabled (initial value) 1: Enabled

(c) Guard area base address registers 0 and 1 (PEGBA0, PEGBA1)

These registers indicate the base address of the area to be guarded.

Access These registers are read-only, in 32-bit units.

Address PEGBA0: $FFFF69B0_H$, PEGBA1: $FFFF69B8_H$

Initial value 00000000_H. These registers are initialized by any reset.

Caution Be sure to clear bits 31 to 23 to "0".

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	PEGBAn						
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PEGBAn															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 4-22 PEGBAn register contents

Bit Position	Bit Name	Description
[22:0]	PEGBAn	These bits indicate the base address of guard area n (n = 0, 1).

(d) Guard area mask address registers 0 and 1 (PEGMA0, PEGMA1)

These registers indicate the mask address of the area to be guarded.

Access These registers are read-only, in 32-bit units.

Address PEGMA0: $FFFF69B4_H$, PEGMA1: $FFFF69BC_H$

Initial value 00000000_H. These registers are initialized by any reset.

Caution Be sure to clear bits 31 to 23 to "0".

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	PEGMAn						
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PEGMAn															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 4-23 PEGMAn register contents

Bit Position	Bit Name	Description
[22:0]	PEGMAn	These bits indicate the mask address of guard area n (n = 0, 1).

(e) Base address shadow register (PEGSB)

This is a shadow register for the base address.

Access This register can be read or written in 32-bit units.

Address $FFFF69A8_H$

Initial value 00000000_H. This register is initialized by any reset.

Caution Be sure to clear bits 31 to 29 and 11 to 0 to “0”.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	PEGSB												
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
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PEGSB				0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R

Table 4-24 PEGSB register contents

Bit Position	Bit Name	Description
[28:12]	PEGSB	Shadow register for the base address

(f) Mask address shadow register (PEGSM)

This is a shadow register for the mask address.

Access This register can be read or written in 32-bit units.

Address $FFFF69AC_H$

Initial value 00000000_H. This register is initialized by any reset.

Caution Be sure to clear bits 31 to 29 and 11 to 0 to “0”.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	PEGSM												
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
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PEGSM				0	0	0	0	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R	R	R	R	R

Table 4-25 PEGSM register contents

Bit Position	Bit Name	Description
[28:12]	PEGSM	Shadow register for the mask address

(g) PE guard function error cause register (PEGEC)

This register holds the cause of an error if an error occurs.

Access This register can be read or written in 16-bit units. Note, however, that the PEGEC register can be read and written in 8-bit units if its upper 8 bits and lower 8 bits are used as the PEGECH and PEGECL registers, respectively.

Address $FFFF69A4_H$

Initial value 0000_H. This register is initialized by any reset.

Caution Be sure to clear bits 14 to 2 to "0".

15	14	13	12	11	10	9	8
PEOR	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R
7	6	5	4	3	2	1	0
0	0	0	0	0	0	PEO1	PEO0
R	R	R	R	R	R	R/W	R/W

Table 4-26 PEGEC register contents

Bit Position	Bit Name	Description
15	PEOR	If the attribute of the guard area is enabled, this bit indicates that an access error occurred in a prohibited area other than the guard area. 0: No error occurred (initial value). 1: An error occurred.
1	PEO1	If the attribute of the guard area is disabled, this bit indicates that an access error occurred in the area indicated by guard area 1. 0: No error occurred (initial value). 1: An error occurred.
0	PEO0	If the attribute of the guard area is disabled, this bit indicates that an access error occurred in the area indicated by guard area 0. 0: No error occurred (initial value). 1: An error occurred.

Note Once each of the above bits has been set (1), it is not cleared to 0, except by a reset, unless 0 is written to it.

4.12.2 Instruction access error report function

When an error occurs during instruction fetch access by the CPU, an instruction access error (MEP exception, exception cause code 00000330_H) is reported only when execution of the instructions included at the address where the error actually occurred is attempted. The cause of errors during instruction fetch access is as follows:

- Detection of an external wait error of the primary memory controller

MEP exceptions are exceptions for which restoration and recovery are not possible without acknowledgment conditions. When an MEP exception occurs, the CPU cannot continue processing normally. In an MEP exception handler program, after the execution of suitable processing, execute a system reset (or execute a substitute program in that CPU). When an MEP exception occurs where the MEP exception handler program is placed, the CPU again branches to the MEP exception handler and MEP exception processing is executed again. In addition, even during MEP exception processing, SYSERR exceptions and FE level non-maskable interrupt exceptions might occur. If this happens, SYSERR exception or FE level non-maskable interrupt exception processing starts when the error occurs.

4.12.3 System error report function

When a critical error occurs in the system (a system error), a system error exception (SYSERR exception) can be reported to the CPU.

(1) System error cause

(a) EXT area error

Detected if the CPU accesses the external bus area or H bus area for data.

- External wait error of primary memory controller
- Wait error on the H bus
- Error when a peripheral I/O register of the secondary DMA is accessed with a bus width other than 32 bits

(b) Reserved area access

Detected when the reserved area is accessed

- Error when the area from $FFFF0000_H$ to $FFFF4FFF_H$, which is reserved for future expansion of CPU-specific peripheral devices, is accessed

(c) DMA error

Detected error due to DMA

- Error due to (a) or (b) above during DMA

(2) System error report setting register

– SEG_CONT: SYSERR exception report setting register

– SEG_FLAG: System error cause saving register

(a) System error control register (SEG_CONT)

This register is used to specify whether SYSERR exceptions caused by system error sources are to occur.

For details about the system error source report funciton, see *4.12.3 “System error report function”*.

Access This register can be read or written in 16-bit units. Note, however, that the SEG_CONT register can be read and written in 8- or 1- bit units if its lower 8 bits are used as the SEG/CONTL register.

Address $FFFF64B0_H$

Initial value 0000_H. This register is initialized by any reset.

Caution Be sure to clear bits 15 to 8, 6, 4, 3, 1, and 0 to “0”.

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
7	6	5	4	3	2	1	0
SEG_CONT DMAE	0	SEG_CONT SEGE	0	0	SEG_CONT EXTE	0	0
R	R	R	R	R	R/W	R	R

Table 4-27 SEG_CONT register contents

Bit Position	Bit Name	Description
7	SEG_CONT DMAE	This bit enables DMA error reporting. This bit sets the operation to perform if a system error occurs due to DMA. 0: No SYSERR exception occurs (initial value). 1: A SYSERR exception occurs.
5	SEG_CONT SEGE	This bit enables reserved area access reporting. This bit sets the operation to perform if the CPU accesses the reserved area ($FFFF0000_H$ to $FFFF4FFF_H$). 0: No SYSERR exception occurs (initial value). 1: A SYSERR exception occurs.
2	SEG_CONT EXTE	This bit enables EXT area error reporting. This bit sets the operation to perform if an error occurs when the CPU accesses the external memory area or peripheral I/O area for data. 0: No SYSERR exception occurs (initial value). 1: A SYSERR exception occurs.

(b) System error control register (SEG_FLAG)

This flag register is used to save whether each system error has been detected.

When a system error is detected, the corresponding error cause flag of this register is set (1). Each flag can be directly set in order to request the SYSERR exception by writing data to this register.

For details about the system error source report funciton, see *4.12.3 "System error report function"*.

Each flag of this register is cleared by writing "0" to it after reading "1" from it. Even if "0" is written to a flag that is "0" when it is read, the flag is set if an error occurs after the flag has been read and before it is written.

Access This register can be read or written in 16-bit units. Note, however, that the SEG_FLAG register can be read and written in 8- or 1-bit units if its lower 8 bits are used as the SEG_FLAGL register.

Address $FFFF64B2_H$

Initial value 0000_H. This register is initialized by any reset.

Caution Be sure to clear bits 15 to 3, 1, and 0 to "0".

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
7	6	5	4	3	2	1	0
SEG_FLAG DMAF	0	SEG_FLAG SEGF	0	0	SEG_FLAG EXTF	0	0
R/W	R	R/W	R	R	R/W	R	R

Table 4-28 SEG_FLAG register contents

Bit Position	Bit Name	Description
7	SEG_FLAG DMAF	DMA error flag This flag is set if a system error is detected upon an error report from the DMA unit. 0: No DMA error occurs (initial value). 1: A DMA error occurrs.
5	SEG_FLAG SEGF	Reserved area error flag This flag is set if the CPU accesses reserved area ($FFFF0000_H$ to $FFFF4FFF_H$) for data. 0: No reserved area access error occurs (initial value). 1: A reserved area access error occurs.
2	SEG_FLAG EXTF	EXT area error flag This flag is set if an error occurs when the CPU accesses the external memory area or H bus area for data. 0: No EXT area error occurs (initial value). 1: An EXT area error occurs.

(3) Processing when a SYSERR exception occurs

When a SYSERR exception occurs, the CPU cannot continue processing normally. After suitable processing by the SYSERR exception handler, reset the system.

Once a SYSERR exception has been reported to the CPU, reporting additional SYSERR exceptions is not possible unless the error source bit is cleared explicitly by software.

MEP exceptions and FENMI exceptions might occur during SYSERR exception processing, but, in this case, the MEP exception/FENMI exception processing starts at that point in time. If the operation switches from the SYSERR exception processing to the MEP exception/FENMI exception processing in this way, no SYSERR exception occurs again during this exception processing. (This is because, once a SYSERR exception is reported to the CPU, reporting additional SYSERR exceptions is not possible unless the error source bit is cleared.)

However, because complete restoration and recovery from a SYSERR exception are not possible, when a SYSERR exception occurs, the assumption is that initialization will generally be performed through a reset.

(4) Function to block subsequent access

When a system error occurs, a SYSERR exception is reported to the CPU. Multiple instruction slips might occur from when the SYSERR exception is reported to the CPU until the CPU starts SYSERR exception processing.

Therefore, a function is provided to block access to CPU-external devices by subsequent instructions during the interval from when SYSERR occurs until the CPU acknowledges the SYSERR exception.

This prevents the illegal updating of external resources.

However, because complete restoration and recovery from a system error exception are not possible, when a SYSERR exception occurs, the assumption is that initialization will generally be performed through a reset.

Chapter 5 Reset Function

5.1 Features

Three types of reset sources

- System resets triggered by $\overline{\text{RESET}}$ pin input
- System resets triggered by a watchdog timer overflow
- System resets triggered by software

5.2 Configuration

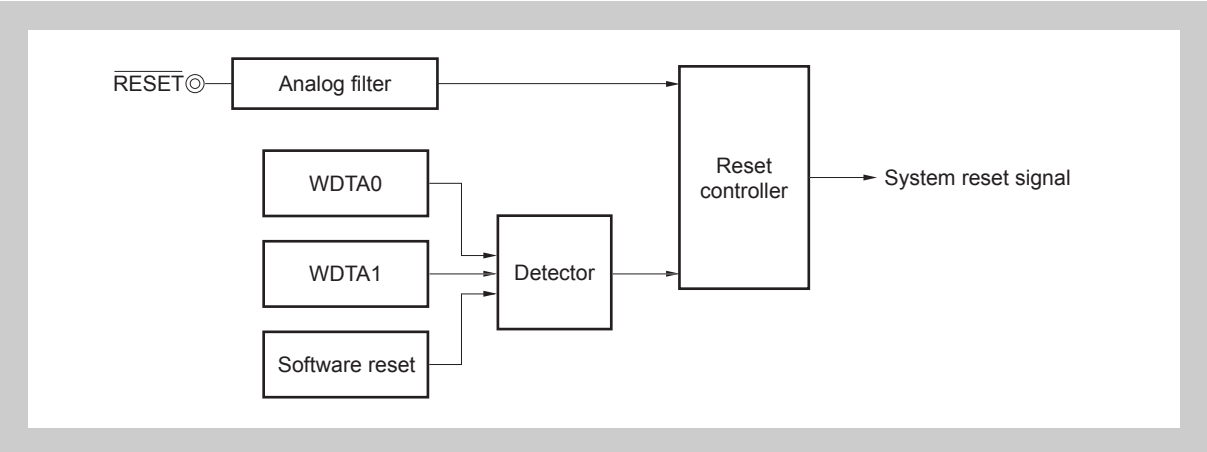


Figure 5-1 Block Diagram of the Reset Circuit

5.3 Control Registers

5.3.1 Reset source flag register (RESF)

This register indicates which reset source a received reset request is from. The bit that corresponds to this source is set.

Access This register is read-only, in 8-bit units.

Address FF42 0040_H

Initial value 00_H. $\overline{\text{RESET}}$ pin input clears this register to 00H. The initial value of this register differs for a reset due to a source other than $\overline{\text{RESET}}$ pin input. Note that bits of this register that have been set are cleared by setting the corresponding bits of the RESC register.

	7	6	5	4	3	2	1	0
RESF	0	0	0	0	0	RESSTG2IN2	RESSTG2IN1	RESSTG2IN0
	R	R	R	R	R	R	R	R

Table 5-1 RESF Register Contents

Bit Position	Bit Name	Description
2	RESSTG2IN2	This bit indicates whether the software reset signal has been generated. 0: Not generated 1: Generated
1	RESSTG2IN1	This bit indicates whether the WDTA1 overflow signal (WDTA1RES) has been generated. 0: Not generated 1: Generated
0	RESSTG2IN0	This bit indicates whether the WDTA0 overflow signal (WDTA0RES) has been generated. 0: Not generated 1: Generated

Note When an emulator is connected, the above bits are not set if the reset mask is enabled.

5.3.2 Reset source clear register (RESC)

This register clears the RESF register. Note that bits of the RESF register that have been set are cleared by setting the corresponding bits of the RESC register.

Access This register is write-only, in 8-bit units. When read, 00_H is always returned.

Address $FF42\ 0044_H$

Initial value 00_H

	7	6	5	4	3	2	1	0
RESC	0	0	0	0	0	STG2 RESCR2	STG2 RESCR1	STG2 RESCR0
	W	W	W	W	W	W	W	W

Table 5-2 RESC Register Contents

Bit Position	Bit Name	Description
2	STG2RESCR2	This bit clears the software reset flag. Setting this bit clears the RESSTG2IN2 bit.
1	STG2RESCR1	This bit clears the WDTA1 overflow signal (WDTA1RES) generation flag. Setting this bit clears the RESSTG2IN1 bit.
0	STG2RESCR0	This bit clears the WDTA0 overflow signal (WDTA0RES) generation flag. Setting this bit clears the RESSTG2IN0 bit.

Note When an emulator is connected, the above bits are not set if the reset mask is enabled.

5.3.3 Software reset enable register (SWRSTEN)

Access This register can be read or written in 32-bit units.

Address FFFF DA04$_H$

Initial value 0000 0000$_H$. This register is initialized by any reset.

	31	30	29	28	27	26	25	24
SWRSTEN	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R
	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R
	15	14	13	12	11	10	9	8
	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	SWRSTEN
	R	R	R	R	R	R	R	R/W

Table 5-3 SWRSTEN Register Contents

Bit Position	Bit Name	Description
0	SWRSTEN	This bit enables or disables the generation of software resets. 0: Disable the generation of software resets. 1: Enable the generation of software resets.

5.3.4 Software reset trigger register (SWREST)

This trigger register generates software resets. When the SWRSTEN.SWRSTEN bit is set, a software reset can be generated by writing 00$_H$ to SWREST.

Access This register can be read or written in 8-bit units.

Address FF42 0000$_H$

Initial value 2C$_H$

	7	6	5	4	3	2	1	0
SWREST	SWREST7	0	1	0	1	1	0	0
	R/W	R	R	R	R	R	R	R

5.4 Operation

5.4.1 Operation for system resets triggered by $\overline{\text{RESET}}$ pin input

When a low-level signal is input to the $\overline{\text{RESET}}$ pin, a system reset occurs, and the hardware is initialized as specified.

When the $\overline{\text{RESET}}$ pin input is changed from low to high, the reset ends and the CPU starts executing a program from address 0000 0000_H.

Note that no oscillation stabilization time is inserted when the input is changed in this way. Therefore, for $\overline{\text{RESET}}$ pin input while the clock generator is stopped (a reset while the power is on), the oscillation stabilization time must be secured by adjusting the low-level width of the $\overline{\text{RESET}}$ pin input.

Table 5-4 Hardware Statuses When a Reset Signal Is Input

Hardware Name	During Reset Period	After the Reset
Clock generator	For details, see *6.4.1 "Operating status of each clock" on page 185* .	
CPU	Operation stopped	Operation started
Internal RAM, H-bus shared memory	Undefined	
Debugging function	Operation stopped	Operation possible
On-chip peripheral registers	Initialized to the specified status	
On-chip peripheral functions other than the above	Operation stopped	Starting operation possible
Pin functions	For details, see *2.4 "Pin State After Reset"*.	For details, see *2.5 "Default States and Recommended Connection of Unused Pins"*.

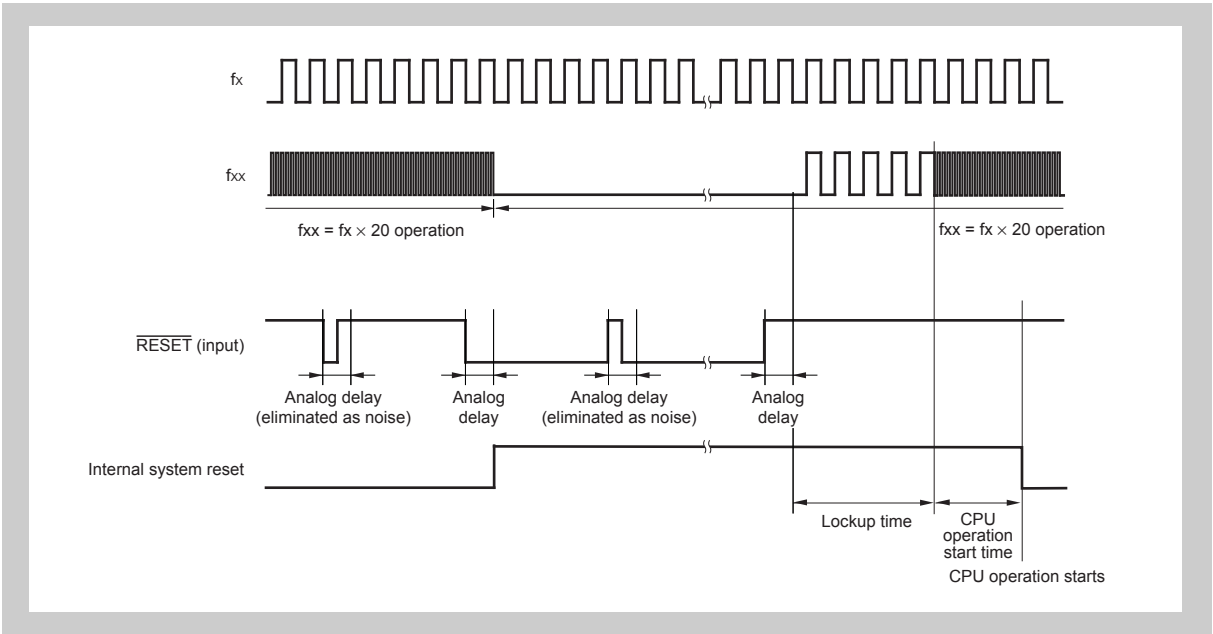


Figure 5-2 Operation for Resets Triggered by the $\overline{\text{RESET}}$ Pin

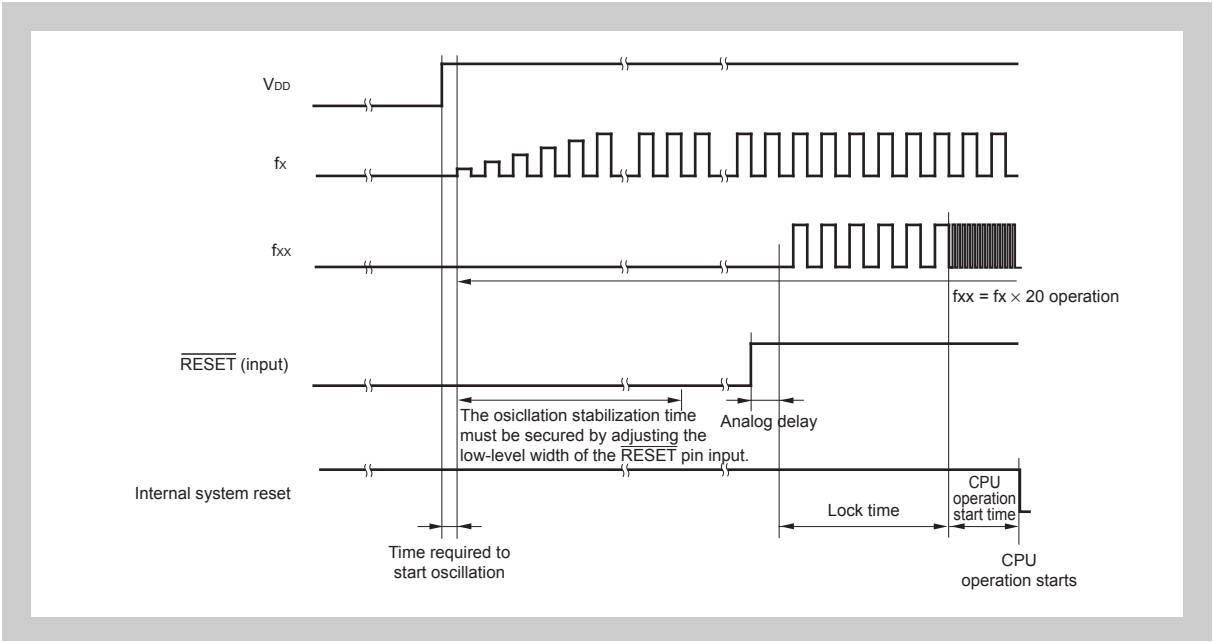


Figure 5-3 Operation for Resets While the Power Is On

5.4.2 System resets triggered by a watchdog timer overflow

When the watchdog timer is set up such that an overflow triggers a reset, a system reset occurs if the watchdog timer overflows, and the hardware is initialized as specified. After the watchdog timer overflows, the system is reset, and then this reset automatically ends.

5.4.3 System resets triggered by software

After the SWRSTEN bit of the software reset enable register SWRSTEN is set to enable the generation of software resets, if 00_H is written to the software reset trigger register, a system reset occurs, and the hardware is initialized as specified.

After writing to the software reset trigger register, the system is reset, and then this reset automatically ends.

To perform a software reset:

1. Interrupt or stop all peripheral functions being used and DMA operations, thereby stopping operation.
2. Disable interrupts.
3. Set the SWRSTEN.SWRSTEN bit.
4. Write 00_H to the SWREST register.
5. Set up an infinite loop so that subsequent instructions are not executed.

Chapter 6 Clock Generator

The clock generator is a circuit that generates the clocks supplied to the CPU and peripheral hardware. The generator includes a PLL circuit used to multiply the oscillated or supplied clock frequency by 20. The generator also makes it possible to divide the clock frequency supplied to the CPU and internal peripheral functions, or to stop supplying clocks to achieve low power consumption.

The features of the clock generator are described below.

6.1 Features

- Oscillation by connecting to an oscillator
- A PLL (Phase Locked Loop) is used to multiply the oscillator input frequency f_X by 20.
- Oscillator input frequency f_X = 7.2 MHz to 10 MHz
- PLL lockup time control
- USB-dedicated clock input
- MII-dedicated clock input
- Power saving control
- HALT mode

6.2 Configuration

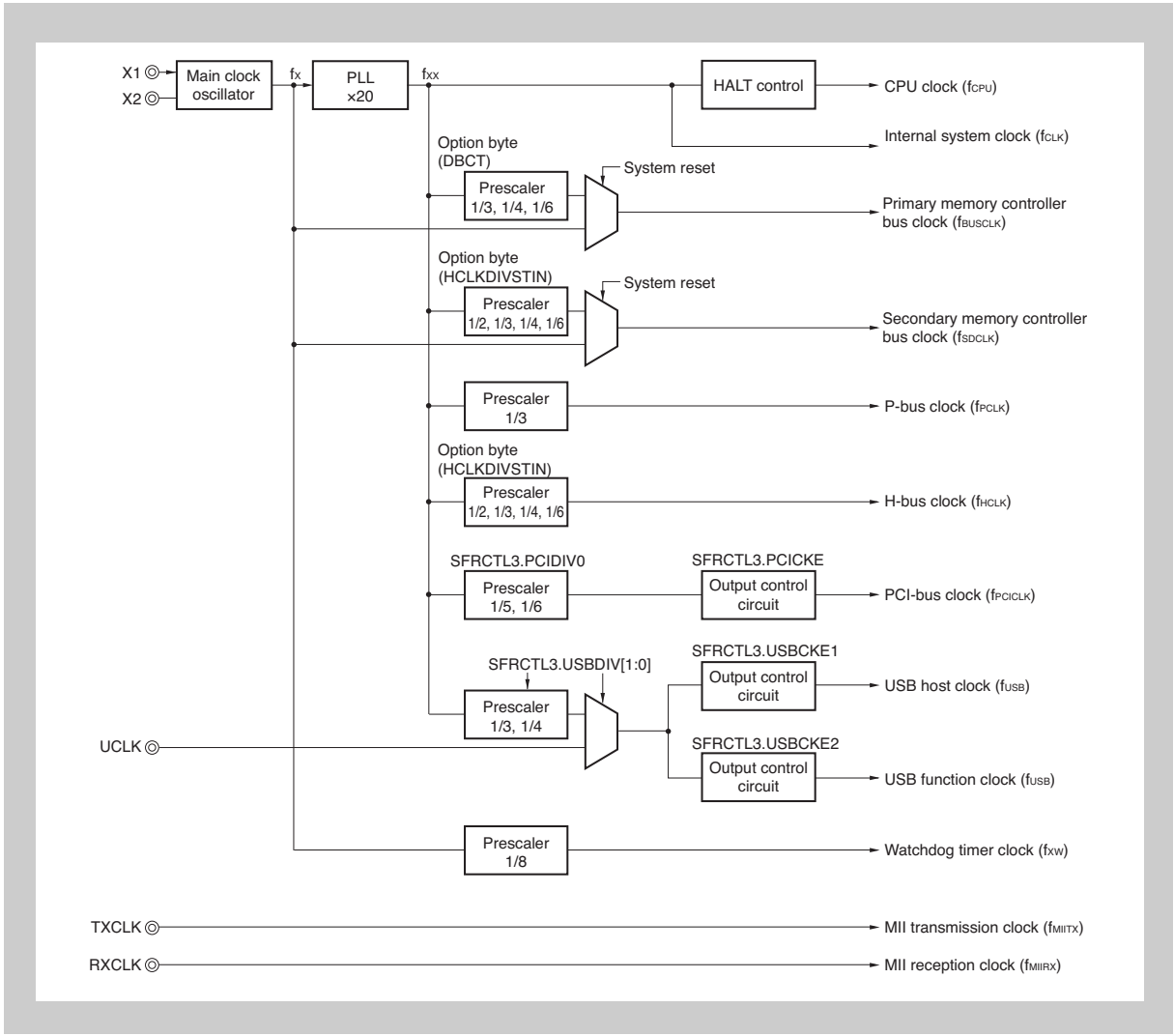


Figure 6-1 Clock Generator Block Diagram

6.3 Control Registers

6.3.1 Special clock frequency control register (SFRCTL3)

The SFRCTL3 register is used to control the clock supplied to the USB function and USB host controllers.

When the clock settings for the SFRCTL3 register are changed, check the new setting values by using the SFR3CKST register, and then access the register of the target peripheral function.

Access This register can be read or written in 8-bit units.

Address FF42 0038$_H$

Default value DC$_H$. This register is initialized by any reset.

Notes

1. Be sure to set bit 6 to 1 and clear bit 5 to 0.
2. Do not access the registers related to the USB function controller when clock supply to the USB function controller is prohibited (USBCKE2 = 0). Do not access the registers related to the USB host controller when clock supply to the USB host controller is prohibited (USBCKE1 = 0) or when PCI bus clock supply to the USB host controller is prohibited (PCICKE = 0).

	7	6	5	4	3	2	1	0
SFRCTL3	PCICKE	1	0	PCIDIV0	USBCLK2	USBCKE1	USBDIV1	USBDIV0
	R/W	R	R	R/W	R/W	R/W	R/W	R/W

Table 6-1 SFRCTL3 Register Contents

Bit Position	Bit Name	Meaning
7	PCICKE	Specify whether to enable supplying the PCI-bus clock to the USB host controller. 0: Disable supplying the clock. 1: Enable supplying the clock.
4	PCIDIV0	Select the division factor for the PCI-bus clock supplied to the USB host controller. 0: $f_{XX}/5$ (f_{XX}: 144 MHz to 166 MHz) 1: $f_{XX}/6$ (f_{XX}: 166 MHz to 200 MHz)
3	USBCKE2	Specify whether to enable supplying the clock to the USB function controller. 0: Disable supplying the clock. 1: Enable supplying the clock.
2	USBCKE1	Specify whether to enable supplying the clock to the USB host controller. 0: Disable supplying the clock. 1: Enable supplying the clock.
1, 0	USBDIV[1:0]	Select the clock supplied to the USB function or USB host controller. Be sure to specify settings so the supplied clock (f_{USB}) is 48 MHz.

USBDIV1	USBDIV0	Clock Selection
0	0	External clock (UCLK)
0	1	Setting prohibited
1	0	$f_{XX}/3$ (f_{XX} = 144 MHz operation)
1	1	$f_{XX}/4$ (f_{XX} = 192 MHz operation)

6.3.2 SFRCTL3 clock selection status register (SFR3CKST)

The SFR3CKST register is used to indicate the values specified for the SFRCTL3 register.

Access This register is read-only, in 16-bit units.

Address FF42 003C$_H$

Default value 40DC$_H$. This register is initialized by any reset.

	15	14	13	12	11	10	9	8
SFR3CKST	HCLKDIV2S	HCLKDIV1S	HCLKDIV0S	0	0	0	0	0
	R	R	R	R	R	R	R	R
	7	6	5	4	3	2	1	0
	PCICKEN	1	0	PCIDIV0S	USBCK2EN	USBCK1EN	USBDIV1S	USBDIV0S
	R	R	R	R	R	R	R	R

Caution If it is specified not to supply the USB host and function clocks, the statuses (initial value and write data) of SFRCTL3.USBCKE2 and SFRCTL3.USBCKE1 are not applied to SFR3CKST.USBCKE2EN and SFR3CKST.USBCKE2EN, respectively.
To read SFR3CKST to use the result with such a setting, mask the corresponding bit by using software.

Table 6-2 SFR3CKST Register Contents (1/2)

Bit Position	Bit Name	Meaning
15:13	HCLKDIV[2:0]S	These bits indicate the division factor for the H-bus clock (f_{HCLK}) and secondary memory controller bus clock (f_{SDCLK}). The values of the HCLKDIVSTIN[2:0] bits in the option byte OPBT0 register are applied to the HCLKDIV[2:0]S bits.
7	PCICKEN	This bit indicates whether supplying the PCI-bus clock to the USB host controller is enabled (the result of setting the SFRCTL3.PCICKE bit). 0: Supplying the clock is disabled. 1: Supplying the clock is enabled.
4	PCIDIV0S	This bit indicates the division factor of the PCI-bus clock supplied to the USB host controller (the result of setting the SFRCTL3.PCIDIV0 bit). 0: f_{XX}/5 (f_{XX}: 144 MHz to 166 MHz) 1: f_{XX}/6 (f_{XX}: 166 MHz to 200 MHz)
3	USBCKE2EN	This bit indicates whether supplying the clock to the USB function controller is enabled (the result of setting the SFRCTL3.USBCKE2 bit). 0: Supplying the clock is disabled. 1: Supplying the clock is enabled.

HCLKDIV2S	HCLKDIV1S	HCLKDIV0S	Division Factor
0	0	1	1/2
0	1	0	1/3
0	1	1	1/4
1	1	1	1/6
Other than the above			Setting prohibited

Table 6-2 SFR3CKST Register Contents (2/2)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Meaning</th></tr>
<tr><td>2</td><td>USBCKE1EN</td><td>This bit indicates whether supplying the clock to the USB host controller is enabled (the result of setting the SFRCTL3.USBCKE1 bit).
0: Supplying the clock is disabled.
1: Supplying the clock is enabled.</td></tr>
<tr><td>1, 0</td><td>USBDIV[1:0]S</td><td>These bits indicate which clock is supplied to the USB function and USB host controllers (the result of setting the SFRCTL3.USBDIV[1:0] bits).
<table>
<tr><th>USBDIV1S</th><th>USBDIV0S</th><th>Clock Selection</th></tr>
<tr><td>0</td><td>0</td><td>External clock (UCLK)</td></tr>
<tr><td>0</td><td>1</td><td>Setting prohibited</td></tr>
<tr><td>1</td><td>0</td><td>$f_{XX}/3$ (f_{XX} = 144 MHz operation)</td></tr>
<tr><td>1</td><td>1</td><td>$f_{XX}/4$ (f_{XX} = 192 MHz operation)</td></tr>
</table>
</td></tr>
</table>

6.4 Operation

6.4.1 Operating status of each clock

Table 6-3 Operating Status of Each Clock

Clock		Normal Operation	HALT Mode	Reset Period	PLL Lockup Time
Name	Symbol				
Main clock oscillator	f_X	✓	✓	✓	✓
Main clock frequency	f_{XX}	✓	✓	×	✓
CPU clock frequency	f_{CPU}	✓	×	×	×
Internal system clock frequency	f_{CLK}	✓	✓	×	×
Primary memory controller bus clock frequency	f_{BUSCLK}	✓	✓	✓[a]	✓[a]
Secondary memory controller bus clock frequency	f_{SDCLK}	✓	✓	✓[a]	✓[a]
P-bus clock frequency	f_{PCLK}	✓	✓	×	×
H-bus clock frequency	f_{HCLK}	✓[b]	✓[b]	×	×
PCI-bus clock frequency	f_{PCICLK}	✓[b]	✓[b]	×	×
USB host controller clock frequency	f_{USB}	✓[c]	✓[c]	×	×
USB function controller clock frequency	f_{USB}	✓[c]	✓[c]	×	×
WDT clock frequency	f_{XW}	✓	✓	×	×
MII transmission clock frequency	f_{MIITX}	✓[d]	✓[d]	×	×
MII reception clock frequency	f_{MIIRX}	✓[d]	✓[d]	×	×

a) The main clock oscillator frequency (f_X) is output.
b) This might be “×” depending on the output control settings.
c) This might be “×” depending on the output control settings and pin settings.
d) This might be “×” depending on the pin settings.

Note ✓: The clock oscillates or is supplied.

×: The clock does not oscillate or is not supplied.

Chapter 7 Option Bytes

7.1 Features

- The option bytes are stored as 32 bits of data in a dedicated internal flash memory area.
- This 32-bit data is used to set the clock generator division factor.
- Be sure to specify values for the option bytes by using a dedicated flash memory programmer or by performing flash memory self programming.

7.2 Data Structure

7.2.1 Option bytes

The option bytes are 32 bits of data stored in a dedicated internal flash memory area. When writing a program to the V850E2/MN4, be sure to specify values for the option bytes by using a dedicated flash memory programmer or by performing flash memory self programming.

The data in this area cannot be rewritten during program execution.

Option bytes

31	30	29	28	27	26	25	24
OPJTAG	ENBSEROCD	MNI21	MNI20	1	1	1	1

23	22	21	20	19	18	17	16
1	1	1	1	1	1	1	1

15	14	13	12	11	10	9	8
1	1	1	1	1	1	1	1

7	6	5	4	3	2	1	0
1	1	HCLKDIVSTIN2	HCLKDIVSTIN1	HCLKDIVSTIN0	DBCT2	DBCT1	DBCT0

Table 7-1 Option Bytes Contents (1/2)

Bit Position	Bit Name	Function
31	OPJTAG	This is a debugging function suitable for a development tool. Be sure to set this to 1.
30	ENBSEROCD	This is a debugging function suitable for a development tool. Be sure to set this to 1.
29	MNI21	This is a debugging function suitable for a development tool. Be sure to set this to 1.
28	MNI20	This is a debugging function suitable for a development tool. Be sure to set this to 1.
5:3	HCLKDIVSTIN[2:0]	Specify the division factors for the H-bus clock (f_{HCLK}) and secondary memory controller bus clock (f_{SDCLK}).

HCLKDIVSTIN[2:0]	Division Factor
001	1/2[a]
010	1/3[b]
011	1/4
111	1/6
Other than the above	Setting prohibited

Table 7-1 Option Bytes Contents (2/2)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>2-0</td><td>DBCT[2:0]</td><td>Specify the division factor for the primary memory controller bus clock (f_{BUSCLK}).
<table>
<tr><th>DBCT[2:0]</th><th>Division Factor</th></tr>
<tr><td>010</td><td>1/3[b]</td></tr>
<tr><td>011</td><td>1/4</td></tr>
<tr><td>111</td><td>1/6</td></tr>
<tr><td>Other than the above</td><td>Setting prohibited</td></tr>
</table>
</td></tr>
</table>

a) External buses cannot be used.
b) SDRAM cannot be used for a frequency higher than 50 MHz.

7.3 Control Register

7.3.1 Option byte storage register (OPBT0)

The OPBT0 register is used to indicate the values of the option bytes stored in a dedicated internal flash memory area.

The data in the OPBT0 register cannot be rewritten during program execution.

Access This register is read-only, in 32-bit units.

Address FF47 000C_H

Default value This depends on which values are written to the option bytes.

	31	30	29	28	27	26	25	24
OPBT0	OPJTAG	ENBSER OCD	MNI21	MNI20	1	1	1	1
	R	R	R	R	R	R	R	R
	23	22	21	20	19	18	17	16
	1	1	1	1	1	1	1	1
	R	R	R	R	R	R	R	R
	15	14	13	12	11	10	9	8
	1	1	1	1	1	1	1	1
	R	R	R	R	R	R	R	R
	7	6	5	4	3	2	1	0
	1	1	HCLKDIV STIN2	HCLKDIV STIN1	HCLKDIV STIN0	DBCT2	DBCT1	DBCT0
	R	R	R	R	R	R	R	R

Table 7-2 OPBT0 Contents (1/2)

Bit Position	Bit Name	Function
31	OPJTAG	This is a debugging function suitable for a development tool.
30	ENBSER OCD	This is a debugging function suitable for a development tool.
29	MNI21	This is a debugging function suitable for a development tool.
28	MNI20	This is a debugging function suitable for a development tool.
5-3	HCLKDIV STIN[2:0]	These bits indicate the division factor for the H-bus clock (f_{HCLK}) and secondary memory controller bus clock (f_{SDCLK}). HCLKDIVSTIN[2:0] / Division Factor 001 / 1/2[a] 010 / 1/3[b] 011 / 1/4 111 / 1/6 Other than the above / Setting prohibited

Table 7-2 OPBT0 Contents (2/2)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>2-0</td><td>DBCT[2:0]</td><td>These bits indicate the division factor for the primary memory controller bus clock (f_{BUSCLK}).
<table>
<tr><th>DBCT[2:0]</th><th>Division Factor</th></tr>
<tr><td>010</td><td>1/3[b]</td></tr>
<tr><td>011</td><td>1/4</td></tr>
<tr><td>111</td><td>1/6</td></tr>
<tr><td>Other than the above</td><td>Setting prohibited</td></tr>
</table>
</td></tr>
</table>

a) External buses cannot be used.
b) SDRAM cannot be used for a frequency higher than 50 MHz.

Chapter 8 Port Functions

This chapter provides a generic description of the port functions.

8.1 "Features" describes the specifications specific to each product, such as the pin groups and the base addresses of registers.

8.2 "Overview" describes the features of the port function provided for all the ports.

8.4 "V850E2/MN4 Port Functions" covers the individual functions of the pins of the V850E2/MN4.

8.1 Features

Port group

This microcontroller has port groups, numbered as shown in the following table.

Table 8-1 Port groups of V850E2/MN4

Port group	
Number of groups	15
Group name	P0 to P14

Cautions

1. The V850E2/MN4 does not have the port universal control register (PUCC) and port special buffer control register (PSBC). Do not access the areas for these registers (<PORTn_base_OS> + 4900_H + n x 4 and <PORTn_base_OS> + $4D00_H$ + n x 4).
2. To use the P14_[15:0] port that is shared with the A/D converter function as the port function, set the ADCAnCTL1.ADCAnGPS bit to turn on the A/D converter (ADCA0). For details, see *25.4.2 (2) "ADCAnCTL1 – A/D converter mode control register 1" on page 1733*.
3. Even if the open drain output is specified by using the port open drain control register (PODCn), it is prohibited to externally input the voltage equal to or greater than 3.6 V to pins other than the 5-V tolerant pins (Y2, AA1, AA2, AB2). In addition, it is prohibited to specify the push-pull output for a 5-V tolerant pin if the voltage equal to or greater than 3.6 V is externally input to that pin, regardless of whether a pull-up resistor is connected.

Port group index n

Throughout this chapter, the individual port groups are identified by the index "n" (n = 0 to 15), for example PMCn indicates the port mode control register for the Pn pin.

Register addresses

All port register addresses are given as addresses offset from the individual base address <PORTn_base_USER> or <PORTn_base_OS>.
The base addresses <PORTn_base_USER> and <PORTn_base_OS> are listed in the following table:

Table 8-2 Port base addresses

Base address	Address
<PORTn_base_OS>	FF40 0000_H
<PORTn_base_USER>	FFFF 8000_H

8.2 Overview

This microcontroller has various pins for input/output functions, known as ports. The ports are organized in port groups.

This microcontroller also has several control registers to allocate other than general purpose input/output functions to the pins.

For a description of the terms pin, port, or port group, see *8.2.1 “Terms” on page 194*.

Features summary

- Configuration possible for individual pins.
- The following functions can be selected for the main pins:
  - 5 types of input buffer characteristics
  - 2 types of output buffer characteristics
  - Open drain emulation
  - Pull-up or pull-down resistor connection
- The following registers are provided for the main ports:
  - Register for reading the pin values
  - Port register
  - Port set/reset register
  - Register for output inversion

8.2.1 Terms

In this section, the following terms are used:

- **Pin**

  Denotes the physical pin. Every pin is denoted by a unique pin number.
  A pin can be used in several modes. The pin name is determined by the selected mode, with a name indicating the pin function.

- **Port group**

  Denotes a group of pins. A port control register is provided for pins belonging to the same port group.

- **Port mode / Port**

  A pin in port mode works as a general purpose input/output pin. It is then called “port”.
  The corresponding name is Pn_m. For example, P0_7 denotes port 7 of port group 0. It is referenced as “port P0_7”.

- **Alternative mode**

  In alternative mode, a pin can be used for in various non-general purpose input/output functions, for example as the input/output pin of on-chip peripherals.
  The corresponding pin name depends on the selected function. For example, pin INTP0 denotes the pin for one of the external interrupt inputs.
  Note that two or more different names can refer to the same physical pin, for example P0_0 and INTP0. The different names indicate the function in which the pin is being operated.

- **Port type**

  The control circuit is determined through specification of the setting register. Control circuits of different types are referred to by their port type.

8.2.2 Overview of pin functions

Pins can function in three modes.

- Port mode (PMCn.PMCnm = 0)

  In the port mode, pins function as general purpose I/O ports.
  Input/output is selected with PMn.PMnm.

- S/W I/O control mode (PMCn.PMCnm = 1, PIPCn.PIPCnm = 0)

  In the S/W I/O control mode, pins are used for the alternative function. Input/output is selected by setting the Mn.PMnm control bit by software.

- Direct I/O control mode (PMCn.PMCnm = 1, PIPCn.PIPCnm = 1)

  In the direct I/O control mode, pins are used for the alternative function. Unlike the S/W I/O control mode, input/output is directly controlled by the alternative function.

The register settings are outlined in the table below.

Table 8-3 Pin function settings (overview)

Mode	Bit			I/O
	PMCnm	PMnm	PIPCnm	
Port mode	0	0	X	Output
		1[a]		Input
S/W I/O control mode	1	0	0	Output
		1	0	Input
Direct I/O control mode		X	1	Control by alternative function

a) Be sure to enable the input buffer (PIBCnm = 1).

When a pin is used in the alternative mode (PMCn.PMCnm = 1), one of up to four different alternative functions is selected with the PFCn and PFCEn registers.

- S/W I/O control function (PIPCn.PIPCnm = 0):
  - Output (PMnm = 0): ALT-OUT1 to ALT-OUT4
  - Input (PMnm = 1): ALT-IN1 to ALT-IN4
- Direct I/O control function (PIPCn.PIPCnm = 1):
  - I/Os of ALT-OUT1 to ALT-OUT4 and ALT-IN1 to ALT-IN4 are directly selected by the alternative function.

Table 8-4 Alternative mode selection (PMCn.PMCnm = 1)

Function	Register			I/O
	PM	PFCE	PFC	
Alternative output mode 1 (ALT-OUT1)	0	0	0	Output
Alternative input mode 1 (ALT-OUT1)	1			Input
Alternative output mode 2 (ALT-OUT2)	0	0	1	Output
Alternative input mode 2 (ALT-OUT2)	1			Input
Alternative output mode 3 (ALT-OUT3)	0	1	0	Output
Alternative input mode 3 (ALT-OUT3)	1			Input
Alternative output mode 4 (ALT-OUT4)	0	1	1	Output
Alternative input mode 4 (ALT-OUT4)	1			Input

When a pin is in the alternative mode (PMCn.PMCnm = 1), one of up to four different alternative functions is selected with the PFCn, PFCEn registers.

8.2.3 Pin data input/output

The registers used for data input/output are described below.

The location that is read via the PPRn register differs according to the pin mode.

Output data In the port mode (PMCn.PMCnm = 0), the value of Pn.Pnm is output from the Pn_m pin.

Input data In the read operation of the PPRn register, one of the value of the Pn_m pin, value of the port related bit Pn.Pnm, or output value from the alternative function is read.

The read source of PPRn depends on the pin mode and the setting of control bits.

The PPRn read mode differences are shown in the following table.

Table 8-5 PPRnm read values

PMCnm	PMnm	PIBCnm	PIPCnm	PODCnm	Mode	PPRnm read value
0	1	0	X	X	Port input, input buffer disabled	Pn.Pnm register
		1		X	Port input, input buffer enabled	Pn_m pin
	0	X		0	Port push-pull output	Pn.Pnm register[a]
				1	Port open drain output	
1	1	X	0	X	Input for S/W I/O control	Pn_m pin
	0			0	Push-pull output for S/W I/O control	Internal output signal for alternative function[a]
				1	Open drain output for S/W I/O control	
	X		1	0	Push-pull output for direct output control	I/O port in alternative function mode: • Input: Pn_m pin • Output: Internal output signal for alternative function[a]
				1	Open drain output for direct output control	

a) When PBDCnm = 1, the Pn_m pin level is read by the PPRnm register.

Effect of each control bit listed in the above table:

- PMCn.PMCnm

  This bit selects the port mode (PMCnm = 0) or the alternative function mode (PMCnm = 1).

- PMn.PMnm

  This bit selects input (PMnm = 1) or output (PMnm = 0) in the port mode (PMCnm = 0) and S/W I/O control mode (PMCnm = 1, PIPCnm = 0).

- PIBCn.PIBCnm

  This bit disables (PIBCnm = 0) or enables (PIBCnm = 1) the input buffer in the input port mode (PMCnm = 0, PMnm = 1). When the input buffer is disabled, PPRnm reads the Pn.Pnm bit, and when the input buffer is enabled, the level of the Pn_m pin is read.

- PIPCn.PIPCnm

  This bit selects the S/W I/O control mode or the direct I/O control mode.

- PODCn.PODCnm

  This bit selects push-pull (PODCnm = 0) or open drain (PODCnm = 1) output.

- PBDCn.PBDCnm

  When this bit is set to 1, the level of the Pn_m pin is forcibly read. In other words, when the port is in the output mode, the bidirectional mode is enabled, allowing the level of the Pn_m pin to be read.

Caution When using the Pn_m port for the alternative output function (PMCn.PMCnm = 1, PMn.PMnm = 0), do not read the level of the Pn_m pin by using the PPRn register.

Pn register write

In the port mode (PMCn.PMCnm = 0), the data output from the Pn_m port is stored in the Pn register.

The Pn data can be rewritten with two different methods.

- Direct write to Pn register

  New data can be written directly to the Pn register.

- Indirect bit manipulation of Pn register (set/reset/not)

  Bit manipulation of the Pn register (set/reset/not) can be performed indirectly using two registers.

  – Port set/reset register PSRn

    When PSRn.PSRn (m+16) = 1, the value of the PSRn.PSRnm bit determines the value of the Pn.Pnm bit.

    In other words, the Pnm bit can be set/reset without a direct write to the Pn register.

  – Port NOT register PNOTn

    When PNOTn.PNOTnm = 1 is set, the Pn.Pnm bit can be inverted without a direct write to the Pn register.

Indirect bit manipulation (set/reset/not) of the Pn register allows rewrite of the bit or bits that need to be updated, without affecting the other bits in the Pn register that do not need updating.

8.2.4 Port control logic diagram

The following figure shows the logic diagram of the port control functions.

Caution The logic shown in this figure is for reference purposes only and does not represent the actual circuit.

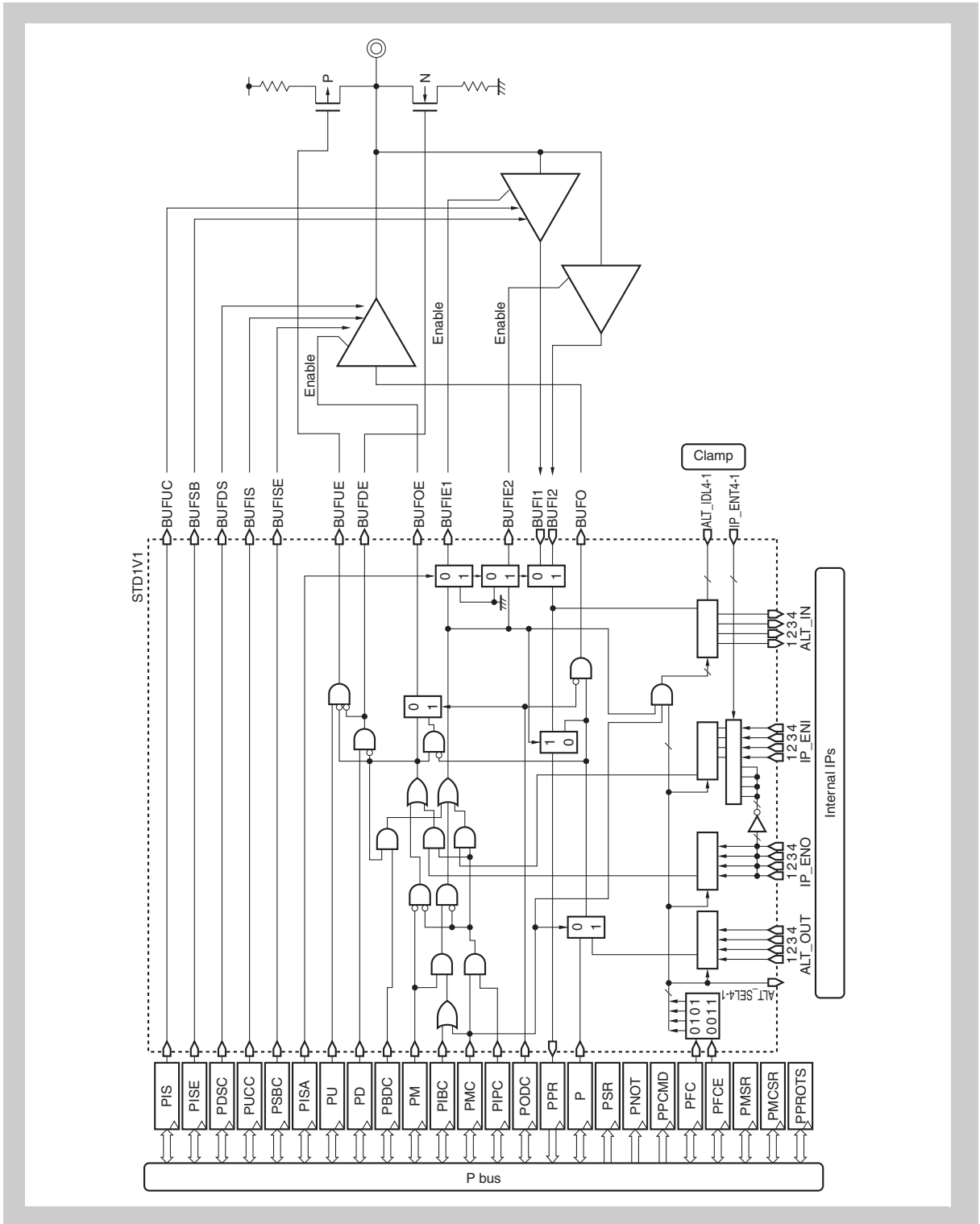


Figure 8-1 Logic diagram of port control

8.2.5 Write protected registers

The write protected registers are protected from accidental write access due to incorrect program execution, etc.

The following port registers have this special write protection function.

- Port drive strength control registers PDSCn
- Port open drain control registers PODCn
- Port universal control registers PUCCn
- Port special buffer control registers PSBCn

(1) Port protection release sequence

Write access to write protected registers is possible only with a special protection release sequence.

1. Write fixed value A5H to protection command register PPCMDn.
2. Write the required value to the protected register.
3. Write the required bit inverted value to the protected register.
4. Write the required value to the protected register.
5. Check that PPROTSn.PPROTSn_0 = 0, and that the required values have been correctly written to the protected register.

8.3 Port Group Configuration Registers

This section starts with an overview of all configuration registers and then presents all registers in detail. The configuration registers are grouped as follows:

- *8.3.2 “Pin function configuration” on page 203*
- *8.3.3 “Pin data input/output” on page 210*
- *8.3.4 “Configuration of electrical characteristics” on page 214*

8.3.1 Overview

The following registers are used for the configuration of the individual pins of the port groups:

Table 8-6 Registers for port group configuration

Register name	Shortcut	Address
Pin function configuration		
Port mode control register	PMCn	<PORTn_base_USER> + 0400_H + n x 4
Port mode control set/reset register	PMCSR	<PORTn_base_USER> + 0900_H + n x 4
Port IP control register	PIPCn	<PORTn_base_OS> + 4200_H + n x 4
Port mode register	PMn	<PORTn_base_USER> + 0300_H + n x 4
Port mode set/reset register	PMSR	<PORTn_base_USER> + 0800_H + n x 4
Port input buffer control register	PIBCn	<PORTn_base_OS> + 4000_H + n x 4
Port function control register	PFCn	<PORTn_base_USER> + 0500_H + n x 4
Port function control expansion register	PFCEn	<PORTn_base_USER> + 0600_H + n x 4
Pin data input/output		
Port bi-direction control register	PBDCn	<PORTn_base_OS> + 4100_H + n x 4
Port pin read register	PPRn	<PORTn_base_USER> + 0200_H + n x 4
Port register	Pn	<PORTn_base_USER> + 0000_H + n x 4
Port NOT register	PNOTn	<PORTn_base_USER> + 0700_H + n x 4
Port set/reset register	PSRn	<PORTn_base_USER> + 0100_H + n x 4
Configuration of electrical characteristics		
Pull-up option register	PUn	<PORTn_base_OS> + 4300_H + n x 4
Pull-down option register	PDn	<PORTn_base_OS> + 4400_H + n x 4
Port universal control register	PUCCn	<PORTn_base_OS> + 4900_H + n x 4
Port drive strength control register	PDSCn	<PORTn_base_OS> + 4600_H + n x 4
Port open drain control register	PODCn	<PORTn_base_OS> + 4500_H + n x 4
Port input buffer selection register	PISn	<PORTn_base_OS> + 4700_H + n x 4
Port input buffer selection expansion register	PISEn	<PORTn_base_OS> + 4800_H + n x 4
Port input buffer selection addition register	PISAn	<PORTn_base_OS> + $4A00_H$ + n x 4
Port special buffer control register	PSBCn	<PORTn_base_OS> + $4D00_H$ + n x 4
Port register protection		
Port register protection command register	PPCMDn	<PORTn_base_OS> + $4C00_H$ + n x 4
Port protection status register	PPROTSn	<PORTn_base_OS> + $4B00_H$ + n x 4

<PORTn_base> The base addresses <PORTn_base_OS> and <PORTn_base_USER> of PORTn are defined in the first section of this chapter under the key word “Register addresses”.

Initial value of registers The initial value after reset depends on the port. The initial values are provided in *8.4.3 “V850E2/MN4 port control registers” on page 235*, not in the following register descriptions.

8.3.2 Pin function configuration

(1) PMCn - Port mode control register

This register specifies whether the individual pins of port group n are in port mode or in alternative mode.

Access This register can be read/written in 16-bit units.

Address <PORTn_base_USER> + 0400_H + n x 4

Initial Value See *8.4.3 "V850E2/MN4 port control registers" on page 235*.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PMCn15	PMCn14	PMCn13	PMCn12	PMCn11	PMCn10	PMCn9	PMCn8	PMCn7	PMCn6	PMCn5	PMCn4	PMCn3	PMCn2	PMCn1	PMCn0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 8-7 PMCn register contents

Bit position	Bit name	Function
15 to 0	PMC[15:0]	Specifies the operation mode of the corresponding pin: 0: Port mode 1: Alternative mode

(2) PMCSRn - Port mode control set/reset register

This register provides an alternative method for writing data to any bit in the PMCn register.

The upper 16 bits of PMCSRn specify whether to write data to the PMCn.PMCnm bits specified by the lower 16 bits of PMCSRn.

Access This register can be read/written in 32-bit units.
Bits 31 to 16 are always read as 0000_H.
Reading bits 15 to 0 returns the value of PMCn.

Address <PORTn_base_USER> + 0900_H + n x 4

Initial Value See *8.4.3 "V850E2/MN4 port control registers" on page 235*.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PMCSRn31	PMCSRn30	PMCSRn29	PMCSRn28	PMCSRn27	PMCSRn26	PMCSRn25	PMCSRn24	PMCSRn23	PMCSRn22	PMCSRn21	PMCSRn20	PMCSRn19	PMCSRn18	PMCSRn17	PMCSRn16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PMCSRn15	PMCSRn14	PMCSRn13	PMCSRn12	PMCSRn11	PMCSRn10	PMCSRn9	PMCSRn8	PMCSRn7	PMCSRn6	PMCSRn5	PMCSRn4	PMCSRn3	PMCSRn2	PMCSRn1	PMCSRn0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 8-8 PMCSRn register contents

Bit position	Bit name	Function
31 to 16	PMCSRn[31:16]	Enable bits that specify whether to write the values of the lower bits of the corresponding PMCSRnm to PMCnm. 0: PMCnm is independent of PMCSRnm. 1: The value of PMCnm is that of PMCSRnm. Example: If PMCSRn.PMCSRn31 = 1, the value of the PMCSRn.PMCSRn15 bit is written to the PMCn.PMCn15 bit.
15 to 0	PMCSRn[15:0]	Data bits that specify the value of PMCnm when the value of PMCSRn (m+16) of the corresponding upper bits is 1. 0: PMCnm = 0 1: PMCnm = 1

(3) PIPCn - Port IP control register

This register specifies whether the I/O direction of the Pn_m pin is controlled by the port mode register PMn.PMnm or the alternative function.

If the Pn_m pin is used in the alternative mode (PMCn.PMCnm = 1), PIPCn.PIPCnm must be set to 1 for the alternative function to control the I/O direction of Pn_m directly.

As a result, the alternative function performs I/O control and the setting of PMn.PMnm is disabled.

Access This register can be read/written in 16-bit units.

Address <PORTn_base_OS> + 4200_H + n x 4

Initial Value See *8.4.3 "V850E2/MN4 port control registers" on page 235*.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PIPCn15	PIPCn14	PIPCn13	PIPCn12	PIPCn11	PIPCn10	PIPCn9	PIPCn8	PIPCn7	PIPCn6	PIPCn5	PIPCn4	PIPCn3	PIPCn2	PIPCn1	PIPCn0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 8-9 PIPCn register contents

Bit position	Bit name	Function
15 to 0	PIPC[15:0]	Specifies the I/O control mode. 0: The I/O mode is controlled by PMn.PMnm (S/W I/O control). 1: The I/O mode is controlled by the peripheral function (direct I/O control).

(4) PMn - Port mode register

The PMn register specifies whether the individual pins of the port group n are in input mode or in output mode.

Access This register can be read/written in 16-bit units.

Address <PORTn_base_USER> + 0300_H + n x 4

Initial Value See *8.4.3 "V850E2/MN4 port control registers" on page 235*.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PMn15	PMn14	PMn13	PMn12	PMn11	PMn10	PMn9	PMn8	PMn7	PMn6	PMn5	PMn4	PMn3	PMn2	PMn1	PMn0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 8-10 PMn register contents

Bit position	Bit name	Function
15 to 0	PMn[15:0]	Specifies input/output mode of the corresponding pin: 0: Output mode (output enabled) 1: Input mode (output disabled)

Notes

1. To use a port in input port mode (PMCn.PMCnm = 0 and PMn.PMnm = 1), the input buffer must be enabled (PIBCn.PIBCnm = 1).
2. By default, PMnm specifies the I/O direction in port mode (PMCn.PMCnm = 0) and alternative mode (PMCn.PMCnm=1), since PIPCn.PIPCnm = 0 after reset.

(5) PMSRn - Port mode set/reset register

This register provides an alternative method for writing data to any bit in the PMn register.

The upper 16 bits of PMSRn specify whether to write data to the PMn.PMnm bits specified by the lower 16 bits of PMSRn.

Access This register can be read/written in 32-bit units.
Bits 31 to 16 are always read as 0000_H.
Reading bits 15 to 0 returns the value of PMn.

Address <PORTn_base_USER> + 0800_H + n x 4

Initial Value See *8.4.3 "V850E2/MN4 port control registers" on page 235*.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PMSRn31	PMSRn30	PMSRn29	PMSRn28	PMSRn27	PMSRn26	PMSRn25	PMSRn24	PMSRn23	PMSRn22	PMSRn21	PMSRn20	PMSRn19	PMSRn18	PMSRn17	PMSRn16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PMSRn15	PMSRn14	PMSRn13	PMSRn12	PMSRn11	PMSRn10	PMSRn9	PMSRn8	PMSRn7	PMSRn6	PMSRn5	PMSRn4	PMSRn3	PMSRn2	PMSRn1	PMSRn0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 8-11 PMSRn register contents

Bit position	Bit name	Function
31 to 16	PMSRn[31:16]	Enable bits that specify whether to write the values of the lower bits of the corresponding PMSRnm to PMnm. 0: PMnm is independent of PMSRnm. 1: The value of PMnm is that of PMSRnm. Example: If PMSRn.PMSRn31 = 1, the value of the PMSRn.PMSRn15 bit is written to the PMn.PMn15 bit.
15 to 0	PMSRn[15:0]	Data bits that specify the value of PMnm when the value of PMSRn (m+16) of the corresponding upper bits is 1. 0: PMnm = 0 1: PMnm = 1

(6) PIBCn - Port input buffer control register

In input port mode (PMCn.PMCnm = 0 and PMn.PMnm = 1) this register enables/disables the port pin's input buffer.

Access This register can be read/written in 16-bit units.

Address <PORTn_base_OS> + 4000_H + n x 4

Initial Value See *8.4.3 "V850E2/MN4 port control registers" on page 235*.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PIBCn 15	PIBCn 14	PIBCn 13	PIBCn 12	PIBCn 11	PIBCn 10	PIBCn 9	PIBCn 8	PIBCn 7	PIBCn 6	PIBCn 5	PIBCn 4	PIBCn 3	PIBCn 2	PIBCn 1	PIBCn 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

Table 8-12 PIBCn register contents

Bit position	Bit name	Function
15 to 0	PIBCn[15:0]	Enables/disables the input buffer: 0: Input buffer disabled 1: Input buffer enabled

Note When the input buffer is disabled, it does not consume current even when the pin level is Hi-Z state. Thus the pin does not need to be fixed to a high or low level externally.

Caution Settings in this register are overruled in bi-directional mode (PBDCn.PBDCnm = 1).

(7) PFCn - Port function control register

This register specifies the alternative functions of pins, together with the PFCEn register.

Some alternative functions perform direct I/O control of the Pn_m pin. In the case of such alternative functions, PIPCn.PIPCnm must be set to 1.

For the other alternative functions, I/O is specified by PMn.PMnm.

Access This register can be read/written in 16-bit units.

Address <PORTn_base_USER> + 0500_H + n x 4

Initial Value See *8.4.3 "V850E2/MN4 port control registers" on page 235*.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PFCn15	PFCn14	PFCn13	PFCn12	PFCn11	PFCn10	PFCn9	PFCn8	PFCn7	PFCn6	PFCn5	PFCn4	PFCn3	PFCn2	PFCn1	PFCn0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 8-13 PFCn register contents

Bit position	Bit name	Function
15 to 0	PFCn[15:0]	Specifies the alternative function of a pin. See *Table 8-4 "Alternative mode selection (PMCn.PMCnm = 1)" on page 196*for details.

(8) PFCEn - Port function control expansion register

This register specifies the alternative functions of pins, together with the PFCn register.

Some alternative functions perform direct I/O control of the Pn_m pin. In the case of such alternative functions, PIPCn.PIPCnm must be set to 1.

For the other alternative functions, I/O is specified by PMn.PMnm.

Access This register can be read/written in 16-bit units.

Address <PORTn_base_USER> + 0600_H + n x 4

Initial Value See *8.4.3 "V850E2/MN4 port control registers" on page 235*.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PFCEn15	PFCEn14	PFCEn13	PFCEn12	PFCEn11	PFCEn10	PFCEn9	PFCEn8	PFCEn7	PFCEn6	PFCEn5	PFCEn4	PFCEn3	PFCEn2	PFCEn1	PFCEn0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 8-14 PFCEn register contents

Bit position	Bit name	Function
15 to 0	PFCEn[15:0]	Specifies the alternative function of a pin. See *Table 8-4 "Alternative mode selection (PMCn.PMCnm = 1)" on page 196*for details.

8.3.3 Pin data input/output

(1) PBDCn - Port bi-direction control register

This register enables the input buffer and allows the level of the Pn_m pin to be always read via PPRn.PPRnm.

Access This register can be read/written in 16-bit units.

Address <PORTn_base_OS> + 4100_H + n x 4

Initial Value See *8.4.3 "V850E2/MN4 port control registers" on page 235.*

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PBDCn 15	PBDCn 14	PBDCn 13	PBDCn 12	PBDCn 11	PBDCn 10	PBDCn 9	PBDCn 8	PBDCn 7	PBDCn 6	PBDCn 5	PBDCn 4	PBDCn 3	PBDCn 2	PBDCn 1	PBDCn 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

Table 8-15 PBDCn register contents

Bit position	Bit name	Function
15 to 0	PBDCn[15:0]	Enables/disables bi-directional mode of the corresponding pin: 0: Bi-directional mode disabled 1: Bi-directional mode enabled

Caution When using the Pn_m port for the alternative output function (PMCn.PMCnm = 1, PMn.PMnm = 0), do not read the level of the Pn_m pin by using the PPRn register.

Note When PBDCn = 1 is set, the port mode setting of PMn.PMnm is ignored.

(2) PPRn - Port pin read register

This register indicates the actual Pn_m pin level, Pn.Pnm bit value, or the output level of an alternative function. The value that is read differs according to the control setting, as shown in *Table 8-5 "PPRnm read values" on page 197*.

Access This register is read-only, in 16-bit units.

Address <PORTn_base_USER> + 0200_H + n x 4

Initial Value See *8.4.3 "V850E2/MN4 port control registers" on page 235*.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PPRn15	PPRn14	PPRn13	PPRn12	PPRn11	PPRn10	PPRn9	PPRn8	PPRn7	PPRn6	PPRn5	PPRn4	PPRn3	PPRn2	PPRn1	PPRn0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 8-16 PPRn register contents

Bit position	Bit name	Function
15 to 0	PPRn[15:0]	Pn_m pin, Pn.Pnm value, or alternative function output

(3) Pn - Port register

This register specifies or holds the data Pn.Pnm to be output via the related port Pn_m in output port mode (PMCn.PMCnm = 0 and PMn.PMnm = 0).

Access This register can be read/written in 16-bit units.

Address <PORTn_base_USER> + 0000_H + n x 4

Initial Value See *8.4.3 "V850E2/MN4 port control registers" on page 235*.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Pn15	Pn14	Pn13	Pn12	Pn11	Pn10	Pn9	Pn8	Pn7	Pn6	Pn5	Pn4	Pn3	Pn2	Pn1	Pn0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 8-17 Pn register contents

Bit position	Bit name	Function
15 to 0	Pn[15:0]	Sets the output level of pin m (m = 0 to 15): 0: Outputs low level 1: Outputs high level

Note The bits of this register can be manipulated by different means. Refer to *8.2.3 "Pin data input/output"* under the keyword "Pn register write".

(4) PNOTn - Port NOT register

This register allows to invert a bit Pnm of the port register Pn without directly writing to Pn.

Access This register can be read/written in 16-bit units. It is always read as 0000_H.

Address <PORTn_base_USER> + 0700_H + n x 4

Initial Value See *8.4.3 "V850E2/MN4 port control registers" on page 235*.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PNOTn15	PNOTn14	PNOTn13	PNOTn12	PNOTn11	PNOTn10	PNOTn9	PNOTn8	PNOTn7	PNOTn6	PNOTn5	PNOTn4	PNOTn3	PNOTn2	PNOTn1	PNOTn0
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 8-18 PNOTn register contents

Bit position	Bit name	Function
15 to 0	PNOTn[15:0]	Specifies if Pn.Pnm is inverted: 0: Pn.Pnm is not inverted (Pnm → Pnm) 1: Pn.Pnm is inverted ($\overline{Pnm}$ → Pnm)

(5) PSRn - Port set/reset register

This register provides an alternative method for writing data to any bit in the Pn register.

The upper 16 bits of PSRn specify whether to write data to the Pn.Pnm bits specified by the lower 16 bits of PSRn.

Access This register can be read/written in 32-bit units.
Bits 31 to 16 are always read as 0000_H.
Reading bits 15 to 0 returns the value of Pn.

Address <PORTn_base_USER> + 0100_H + n x 4

Initial Value See *8.4.3 "V850E2/MN4 port control registers" on page 235*.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PSRn31	PSRn30	PSRn29	PSRn28	PSRn27	PSRn26	PSRn25	PSRn24	PSRn23	PSRn22	PSRn21	PSRn20	PSRn19	PSRn18	PSRn17	PSRn16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSRn15	PSRn14	PSRn13	PSRn12	PSRn11	PSRn10	PSRn9	PSRn8	PSRn7	PSRn6	PSRn5	PSRn4	PSRn3	PSRn2	PSRn1	PSRn0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 8-19 PSRn register contents

Bit position	Bit name	Function
31 to 16	PSRn[31:16]	Enable bits that specify whether to write the values of the lower bits of the corresponding PSRnm to Pnm. 0: Pnm is independent of PSRnm. 1: The value of Pnm is that of PSRnm. Example: If PSRn.PSRn31 = 1, the value of bit PSRn.PSRn15 is written to bit Pn.Pn15 and output.
15 to 0	PSRn[15:0]	Data bits that specify the value of Pnm when the value of PSRn (m+16) of the corresponding upper bits is 1. 0: Pnm = 0 1: Pnm = 1

8.3.4 Configuration of electrical characteristics

(1) PUn - Pull-up option register

This register specifies whether to connect an internal pull-up resistor to an input pin.

Access This register can be read/written in 16-bit units.

Address <PORTn_base_OS> + 4300_H + n x 4

Initial Value See *8.4.3 "V850E2/MN4 port control registers" on page 235*.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PUn15	PUn14	PUn13	PUn12	PUn11	PUn10	PUn9	PUn8	PUn7	PUn6	PUn5	PUn4	PUn3	PUn2	PUn1	PUn0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 8-20 PUn register contents

Bit position	Bit name	Function
15 to 0	PUn[15:0]	Specifies whether to connect an internal pull-up resistor to the corresponding pin: 0: No internal pull-up resistor connected 1: An internal pull-up resistor connected

Notes
1. If a pin is configured that both an internal pull-up resistor (PUn.PUnm = 1) and pull-down resistor (PDn.PDnm = 1) are connected, the pull-down resistor is automatically selected and the pull-up resistor is not connected.
2. The internal pull-up resistor has no effect when the pin is operated in output mode.

(2) PDn - Pull-down option register

This register specifies whether to connect an internal pull-down resistor to an input pin.

Access This register can be read/written in 16-bit units.

Address <PORTn_base_OS> + 4400_H + n x 4

Initial Value See *8.4.3 "V850E2/MN4 port control registers" on page 235*.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PDn15	PDn14	PDn13	PDn12	PDn11	PDn10	PDn9	PDn8	PDn7	PDn6	PDn5	PDn4	PDn3	PDn2	PDn1	PDn0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 8-21 PDn register contents

Bit position	Bit name	Function
15 to 0	PDn[15:0]	Specifies whether to connect an internal pull-down resistor to the corresponding pin: 0: No internal pull-down resistor connected 1: An internal pull-down resistor connected

Notes
1. If a pin is configured that both an internal pull-up resistor (PUn.PUnm = 1) and pull-down resistor (PDn.PDnm = 1) are connected, the pull-down resistor is automatically selected and the pull-up resistor is not connected.
2. The internal pull-down resistor has no effect when the pin is operated in output mode.

(3) PDSCn - Port drive strength control register

This register specifies the output driver strength of the port pin. This function is also related to the fast mode (high drive strength) and slow mode (low drive strength) of the output buffer.

This register is one of the OS registers of SPF.

Access This register can be read/written in 32-bit units.

The correct write sequence using the PPCMD register is required in order to update this register.

Address <PORTn_base_OS> + 4600_H + n x 4

Initial Value See *8.4.3 "V850E2/MN4 port control registers" on page 235.*

Caution The upper 16 bits are also inverted through a sequence for writing to a protected port register. For details, see *8.3.5 "Port register protection"*.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PDSCn 15	PDSCn 14	PDSCn 13	PDSCn 12	PDSCn 11	PDSCn 10	PDSCn 9	PDSCn 8	PDSCn 7	PDSCn 6	PDSCn 5	PDSCn 4	PDSCn 3	PDSCn 2	PDSCn 1	PDSCn 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

Table 8-22 PDSCn register contents

Bit position	Bit name	Function
15 to 0	PDSCn[15:0]	Specifies the port drive strength of the output buffer of the port pin. 0: Low drive strength (slow mode) 1: High drive strength (fast mode) **Note** To use this register together with an I/O buffer that can control the output current limit, specify the current limiting function. 0: Current limiting function enabled 1: Current limiting function disabled

(4) PUCCn - Port universal control register

This register expands the function for specifying output buffer characteristics.

This register can specify up to 4 output buffer characteristics, together with port drive strength control register PDSCn.

This register is one of the OS registers of SPF.

Access This register can be read/written in 32-bit units.

The correct write sequence using the PPCMD register is required in order to update this register.

Address <PORTn_base_OS> + 4900_H + n x 4

Initial Value See *8.4.3 "V850E2/MN4 port control registers" on page 235.*

Caution The upper 16 bits are also inverted through a sequence for writing to a protected port register. For details, see *8.3.5 "Port register protection"*.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PUCCn 15	PUCCn 14	PUCCn 13	PUCCn 12	PUCCn 11	PUCCn 10	PUCCn 9	PUCCn 8	PUCCn 7	PUCCn 6	PUCCn 5	PUCCn 4	PUCCn 3	PUCCn 2	PUCCn 1	PUCCn 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

Table 8-23 PUCCn register contents

Bit position	Bit name	Function
15 to 0	PUCCn[15:0]	Specifies the output buffer characteristics of pin m (m = 0 to 15), together with bits PDSCn[15:0].

PUCCnm	PDSCnm	Output buffer characteristics
0	0	Selection 1
0	1	Selection 2
1	0	Selection 3
1	1	Selection 4

Note The type of output buffer characteristics is not specified. However, when an IO buffer that can control the output current limit is used, specify the current limiting function.
0: Lower current limit
1: Upper current limit

(5) PODCn - Port open drain control register

This register selects push-pull or open-drain as output buffer function.

Access This register can be read/written in 32-bit units.

Writing to this register is protected by a special sequence of instructions. See *Table 8.3.5 "Port register protection" on page 222* for details.

Address <PORTn_base_OS> + 4500_H + n x 4

Initial Value See *8.4.3 "V850E2/MN4 port control registers" on page 235*.

Caution The upper 16 bits are also inverted through a sequence for writing to a protected port register. For details, see *8.3.5 "Port register protection"*.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PODCn 15	PODCn 14	PODCn 13	PODCn 12	PODCn 11	PODCn 10	PODCn 9	PODCn 8	PODCn 7	PODCn 6	PODCn 5	PODCn 4	PODCn 3	PODCn 2	PODCn 1	PODCn 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

Table 8-24 PODCn register contents

Bit position	Bit name	Function
15 to 0	PODCn[15:0]	Specifies the output buffer function: 0: Push-pull 1: Open-drain

(6) PISn - Port input buffer selection register

This register specifies the input buffer characteristics.

If a port has up to four input buffer characteristics, the port input buffer selection expansion register PISEn is also valid.

If a port has up to five input buffer characteristics, the port input buffer selection expansion register PISEn and the port input buffer selection addition register PISAn are also valid.

Access This register can be read/written in 16-bit units.

Address <PORTn_base_OS> + 4700_H + n x 4

Initial Value See *8.4.3 "V850E2/MN4 port control registers" on page 235.*

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PISn15	PISn14	PISn13	PISn12	PISn11	PISn10	PISn9	PISn8	PISn7	PISn6	PISn5	PISn4	PISn3	PISn2	PISn1	PISn0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 8-25 PISn register contents

Bit position	Bit name	Function
15 to 0	PISn[15:0]	Specifies the input buffer characteristic: 0: Type 1 1: Type 2

Note The definition of type 1 and type 2 is given in *8.4.1 "Input circuit types of port input pins and alternate-function input pins" on page 228.* Refer also to the Electrical Target Specification for input buffer characteristics details.

(7) PISEn - Port input buffer selection expansion register

This register specifies the input buffer characteristics, together with the port input buffer selection register PISn.

If a port has up to five input buffer characteristics, the port input buffer selection addition register PISAn is also valid.

Access This register can be read/written in 16-bit units.

Address <PORTn_base_OS> + 4800_H + n x 4

Initial Value See *8.4.3 "V850E2/MN4 port control registers" on page 235.*

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PISEn15	PISEn14	PISEn13	PISEn12	PISEn11	PISEn10	PISEn9	PISEn8	PISEn7	PISEn6	PISEn5	PISEn4	PISEn3	PISEn2	PISEn1	PISEn0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 8-26 PISEn register contents

Bit position	Bit name	Function
15 to 0	PISEn[15:0]	Specifies the input buffer characteristic of pin m (m = 0 to 15), together with bits PISn[15:0]:

PISEnm	PISnm	Input buffer characteristic
0	0	Type 1
0	1	Type 2
1	0	Type 3
1	1	Type 4

Note The definition of type 1 to type 4 is given in *8.4.1 "Input circuit types of port input pins and alternate-function input pins" on page 228*. Refer also to the Electrical Target Specification for input buffer characteristics details.

(8) PISAn - Port input buffer selection addition register

This register specifies the input buffer characteristics, together with the port input buffer selection register PISn and the port input buffer selection expansion register PISEn.

Access This register can be read/written in 16-bit units.

Address <PORTn_base_OS> + $4A00_H$ + n x 4

Initial Value See *8.4.3 "V850E2/MN4 port control registers" on page 235.*

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PISAn15	PISAn14	PISAn13	PISAn12	PISAn11	PISAn10	PISAn9	PISAn8	PISAn7	PISAn6	PISAn5	PISAn4	PISAn3	PISAn2	PISAn1	PISAn0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 8-27 PISAn register contents

Bit position	Bit name	Function
15 to 0	PISAn[15:0]	Specifies the input buffer characteristic of pin m (m = 0 to 15), together with bits PISn[15:0] and PISEn[15:0]:

PISAnm	PISEnm	PISnm	Input buffer characteristic
0	0	0	Type 1
	0	1	Type 2
	1	0	Type 3
	1	1	Type 4
1	X	X	Type 5

Note The definition of type 1 to type 5 is given in *8.4.1 "Input circuit types of port input pins and alternate-function input pins" on page 228*. Refer also to the Electrical Target Specification for detailed input buffer characteristics.

(9) PSBCn - Port special buffer control register

This microcontroller may be equipped with I/O buffers with dedicated properties for special functions. This register specifies whether the individual pins of port group n are general purpose I/O pins or if they are special I/O pins.

Access This register can be read/written in 32-bit units.

Writing to this register is protected by a special sequence of instructions. See *Table 8.3.5 "Port register protection" on page 222* for details.

Address <PORTn_base_OS> + $4D00_H$ + n x 4

Initial Value See *8.4.3 "V850E2/MN4 port control registers" on page 235.*

Caution The upper 16 bits are also inverted through a sequence for writing to a protected port register. For details, see *8.3.5 "Port register protection"*.

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSBCn15	PSBCn14	PSBCn13	PSBCn12	PSBCn11	PSBCn10	PSBCn9	PSBCn8	PSBCn7	PSBCn6	PSBCn5	PSBCn4	PSBCn3	PSBCn2	PSBCn1	PSBCn0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 8-28 PSBCn register contents

Bit position	Bit name	Function
15 to 0	PSBC[15:0]	Specifies the I/O mode: 0: General purpose I/O 1: Special I/O

Note For details about the ports that have the special buffers of the V850E2/MN4, see *Port group* in *8.2 "Overview"*.

8.3.5 Port register protection

(1) PPCMDn - Port register protection command register

This register is the command register for port registers that are protected.

Access This register can be written in 8-bit units.

"0" is always read for bits 7 to 0.

Address <PORTn_base_OS> + $4C00_H$ + n x 4

Initial Value 00_H. This register is initialized by any reset.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0
W	W	W	W	W	W	W	W

Table 8-29 PPCMDn register contents

Bit position	Bit name	Function
7 to 0	–	Command allowing write to port register that are protected

(2) PPROTSn - Port protection status register

This register indicates the status of the write sequence for the port registers that are protected.

Access This register can be read in 8-bit units.

Write operation to this register is ignored.

Address <PORTn_base_OS> + $4B00_H$ + n x 4

Initial Value 00_H. This register is initialized by any reset.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	PPROTSn_0
R	R	R	R	R	R	R	R

Table 8-30 PPROTSn register contents

Bit position	Bit name	Function
0	PPROTSn_0	Checks for the occurrence of write sequence errors in the port registers that are protected. 0: No protection error 1: Protection error occurred

(3) Protected port registers

PDSCn - Port drive strength control register

PUCCn - Port universal control register

PODCn - Port open drain control register

PSBCn - Port special buffer control register

(4) Write sequence of protected port register

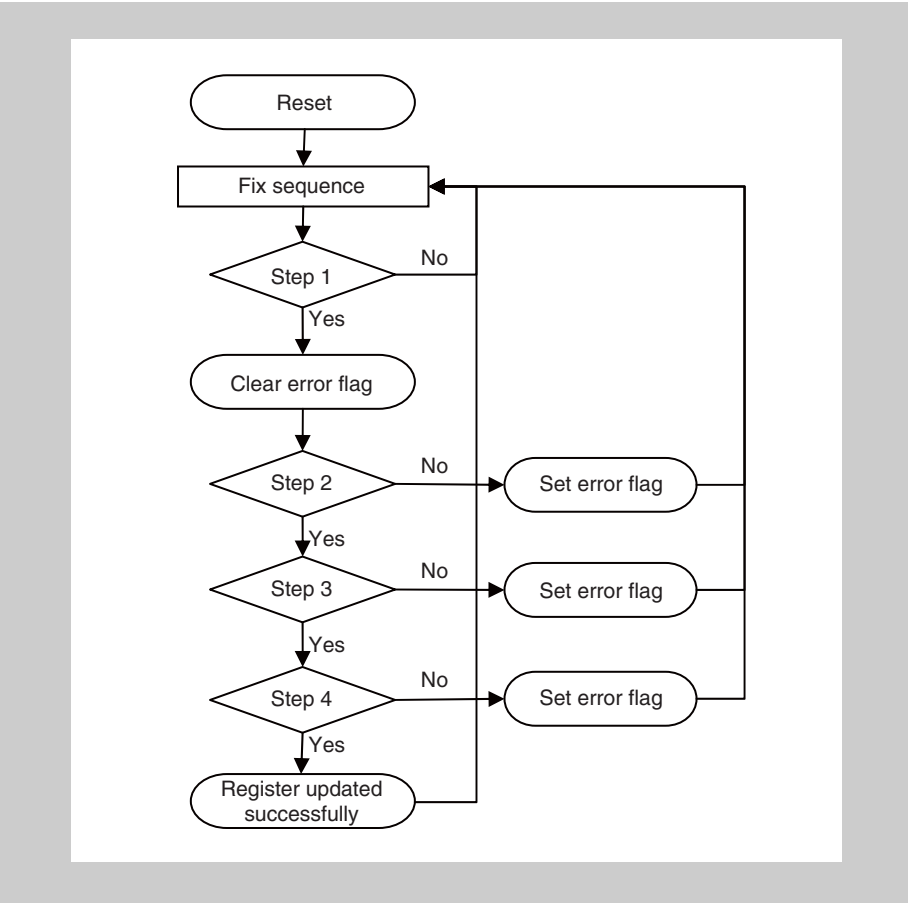


Figure 8-2 Write sequence of protected port register

Step 1 Write A5H to the PPCMD register to initialize the write sequence.

Step 2 Write data in 32-bit nits to the protected register (the register is not updated).

Step 3 Write the inverted data in 32-bit units to the same protected register (the register is not updated).

Step 4 Again write data in 32-bit units to the same protected register (update is successful).

8.3.6 Example port settings

Examples of the port settings are shown in the flowchart below.

Caution If the port has been set to alternative output mode by setting PIPCn.PIPCnm to 0, the port might briefly enter alternative input mode. This will occur between when the PMCn.PMCnm bit is set to 1 and when the PMn.PMnm bit is set to 0. If an interrupt-related signal is specified as an alternate function of a port, the mode becomes the alternative input mode temporarily, and the interrupt either does not occur or is ignored in port mode.

(1) Batch setting

An example of specifying batch port settings is shown in the flowchart below.

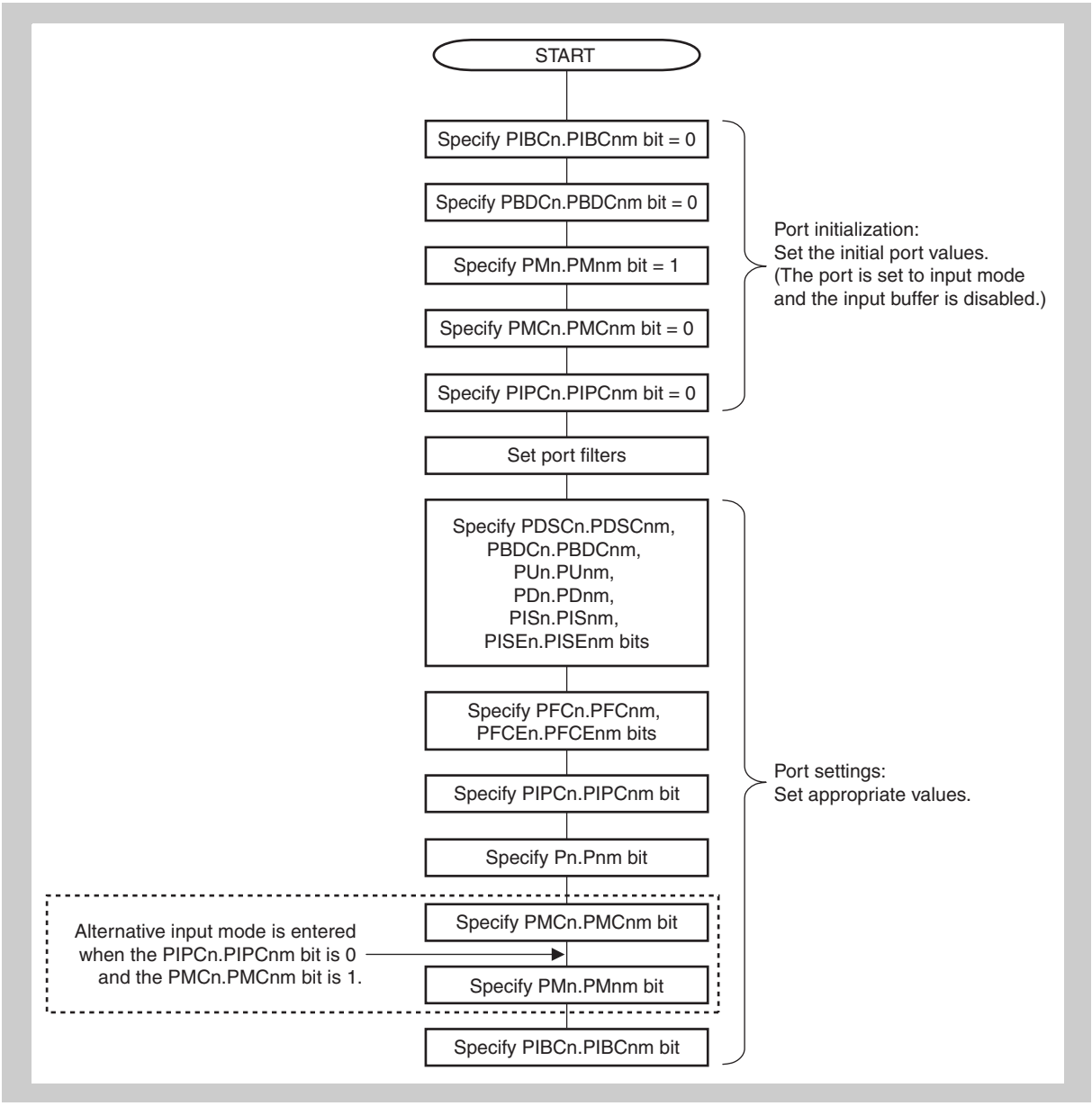


Figure 8-3 Example of port settings (when specified in batch)

(2) Individual settings

An example of specifying individual port settings is shown in the flowchart below

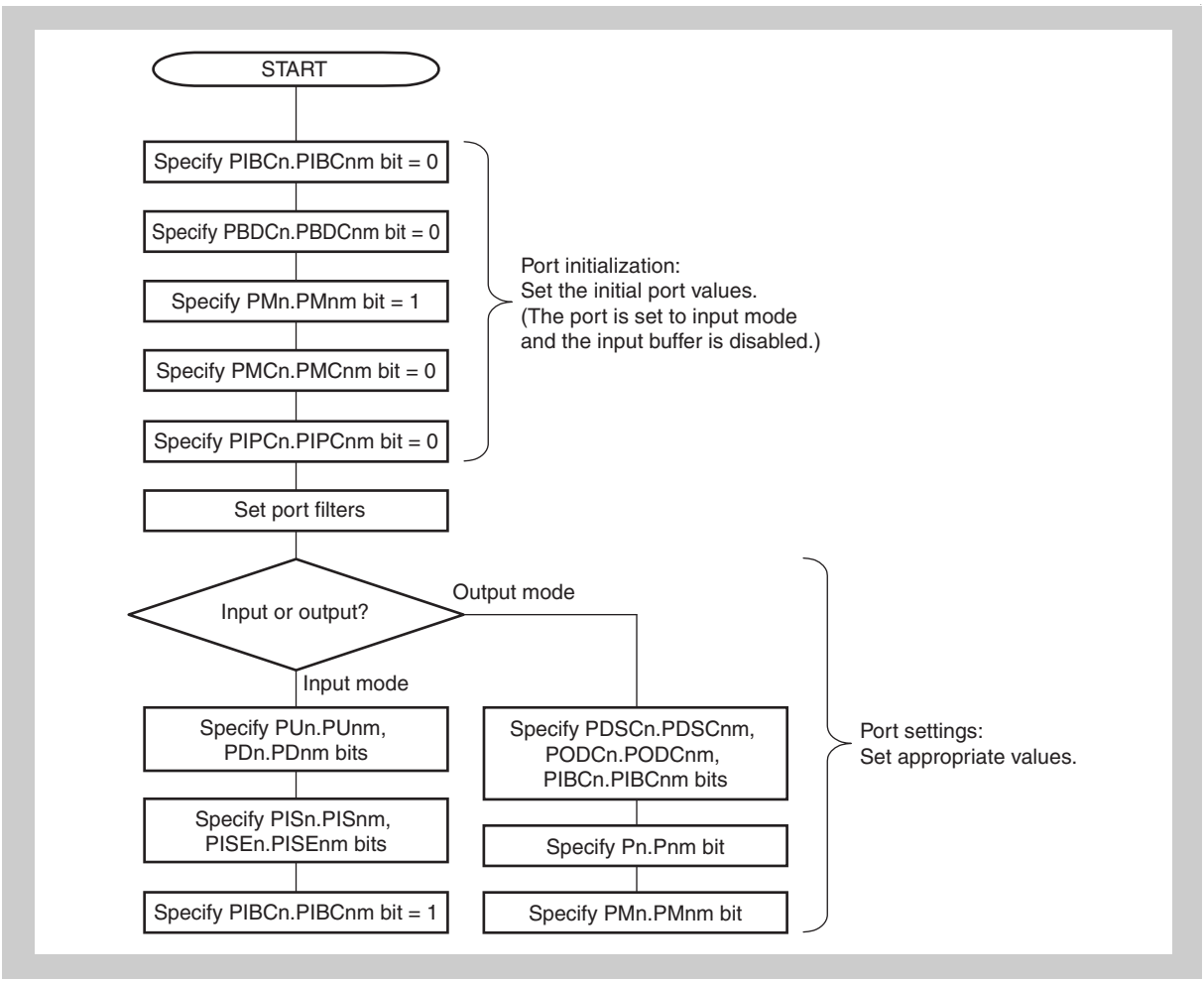


Figure 8-4 Example of port settings (in port mode)

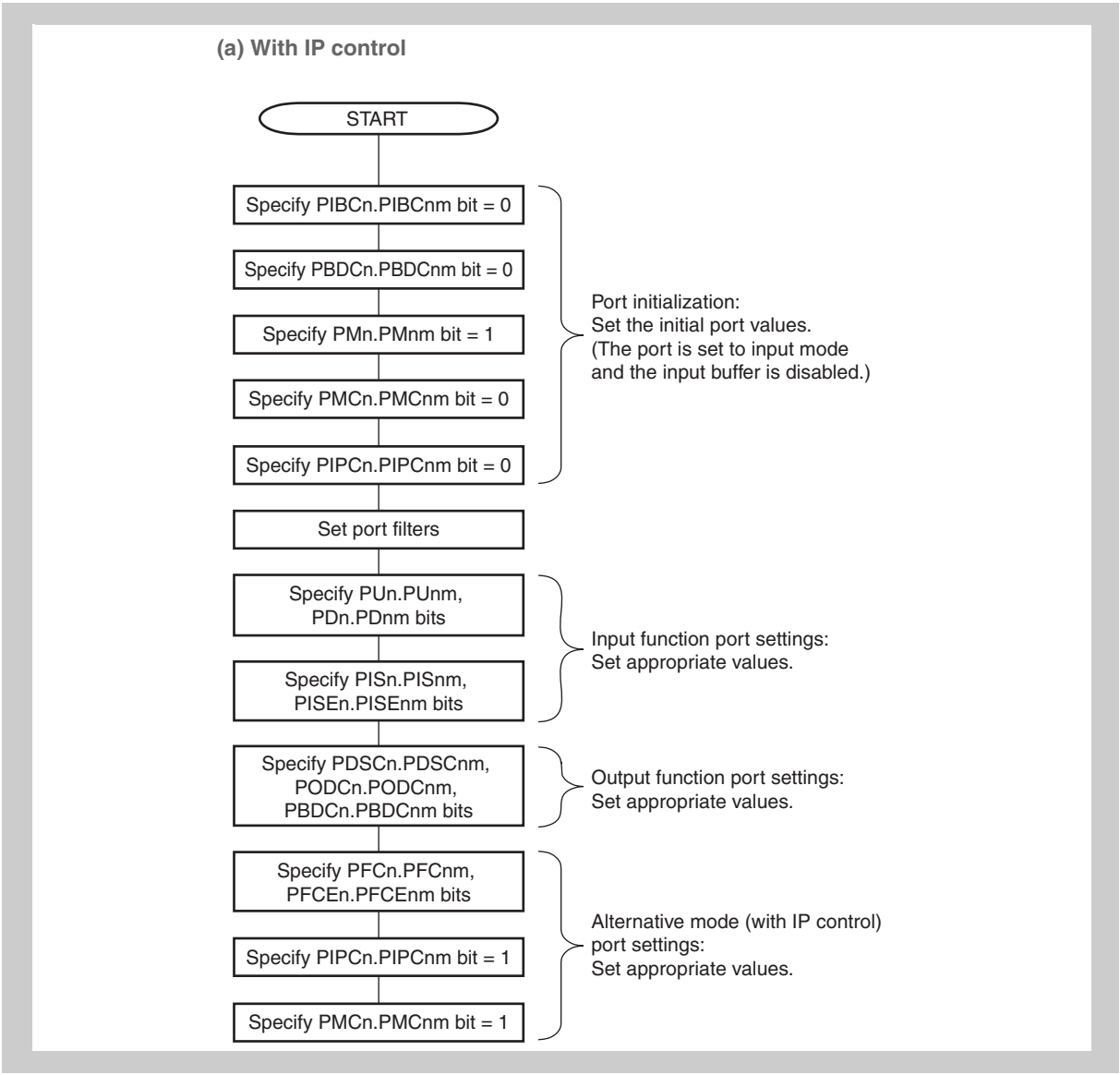


Figure 8-5 Example of port settings (in alternative mode) (1/2)

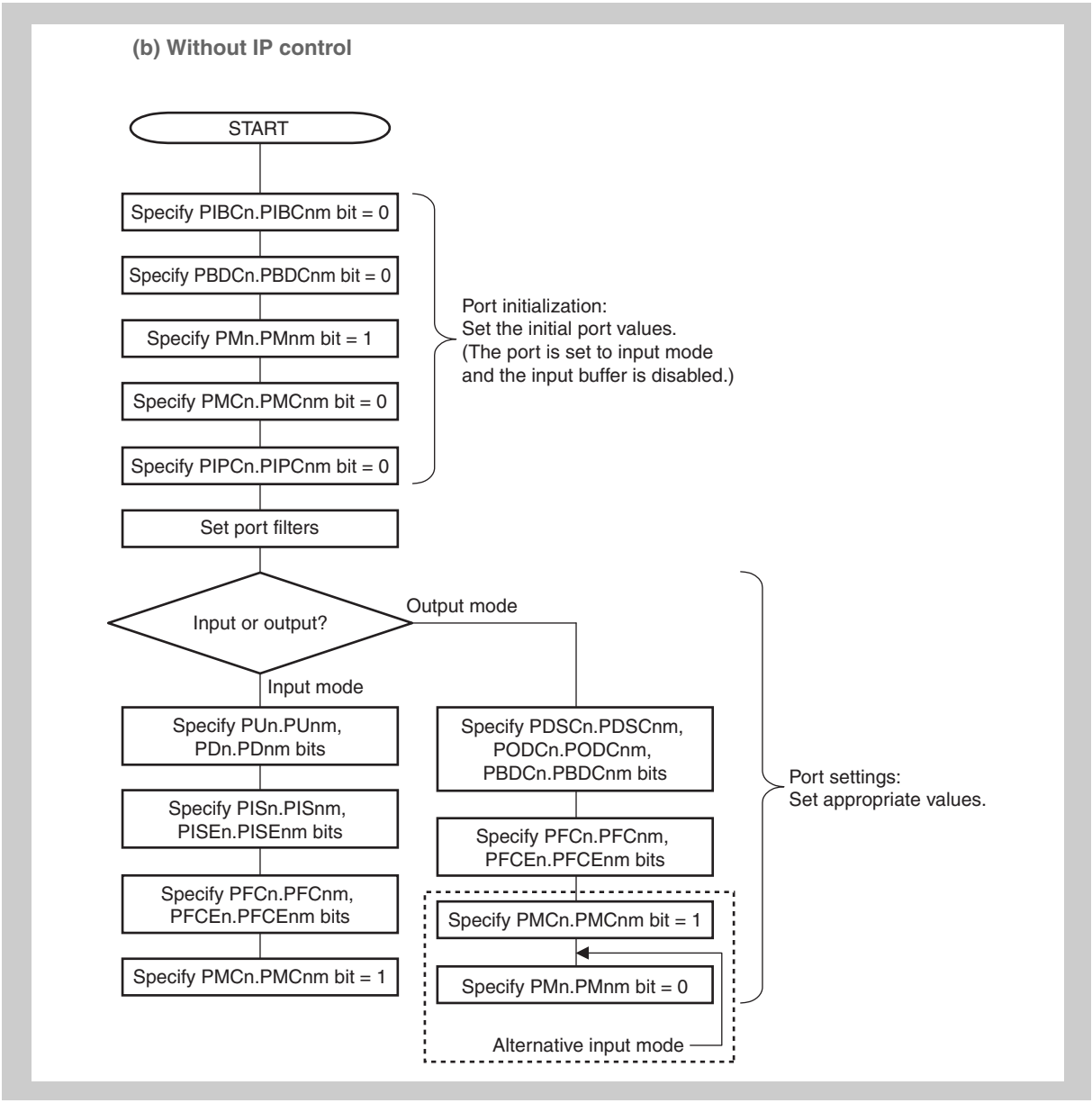


Figure 8-5 Example of port settings (in alternative mode) (2/2)

8.4 V850E2/MN4 Port Functions

This section describes port functions, alternative functions, and port control registers for the V850E2/MN4.

8.4.1 Input circuit types of port input pins and alternate-function input pins

The input circuit type is determined by the settings of the port function control registers PIS, PISE, and PISA, as follows:

Table 8-31 Input circuit types of port input pins and alternate-function input pins

PISnm	PISEnm	PISAnm	Input circuit type
0	0	0	CMOS
1	0	0	Schmitt 2
0	1	0	Schmitt 1
×	×	1	LVTTL
The settings other than the above are prohibited.			

For details about how to set each pin, see *8.4.3 "V850E2/MN4 port control registers"*.

8.4.2 General I/O function

*Table 8-32 "List of port pins and alternative functions for port groups 0 to 13"*shows the port functions and alternative functions of the V850E2/MN4.

Various modes can be specified by changing the values of the PMCn_m, PFCn_m, PFCEn_m, and PMn_m bits.

Table 8-32 List of port pins and alternative functions for port groups 0 to 13 (1/7)

Port mode	Alternative mode							
PMCn_m = 0	PMCn_m = 1							
	PFCEn_m = 0, PFCn_m = 0		PFCEn_m = 0, PFCn_m = 1		PFCEn_m = 1, PFCn_m = 0		PFCEn_m = 1, PFCn_m = 1	
	PMn_m = 1	PMn_m = 0	PMn_m = 1	PMn_m = 0	PMn_m = 1	PMn_m = 0	PMn_m = 1	PMn_m = 0
	ALT_IN1	ALT_OUT1	ALT_IN2	ALT_OUT2	ALT_IN3	ALT_OUT3	ALT_IN4	ALT_OUT4
Port group 0:								
P0_0	P_D0	P_D0	TA0_I0	TA0_O0	TE0_TI0			
P0_1	P_D1	P_D1	TA0_I1	TA0_O1				
P0_2	P_D2	P_D2	TA0_I2	TA0_O2	TE0_TI1			
P0_3	P_D3	P_D3	TA0_I3	TA0_O3				
P0_4	P_D4	P_D4	TA0_I4	TA0_O4	TE0_AI			
P0_5	P_D5	P_D5	TA0_I5	TA0_O5				
P0_6	P_D6	P_D6	TA0_I6	TA0_O6	TE0_BI			
P0_7	P_D7	P_D7	TA0_I7	TA0_O7				

Table 8-32 List of port pins and alternative functions for port groups 0 to 13 (2/7)

Port mode	Alternative mode							
PMCn_m = 0	PMCn_m = 1							
	PFCEn_m = 0, PFCn_m = 0		PFCEn_m = 0, PFCn_m = 1		PFCEn_m = 1, PFCn_m = 0		PFCEn_m = 1, PFCn_m = 1	
	PMn_m = 1	PMn_m = 0	PMn_m = 1	PMn_m = 0	PMn_m = 1	PMn_m = 0	PMn_m = 1	PMn_m = 0
	ALT_IN1	ALT_OUT1	ALT_IN2	ALT_OUT2	ALT_IN3	ALT_OUT3	ALT_IN4	ALT_OUT4
P0_8	P_D8	P_D8	TA0_I8	TA0_O8	TE0_ZI			
P0_9	P_D9	P_D9	TA0_I9	TA0_O9				
P0_10	P_D10	P_D10	TA0_I10	TA0_O10				
P0_11	P_D11	P_D11	TA0_I11	TA0_O11				
P0_12	P_D12	P_D12	TA0_I12	TA0_O12				
P0_13	P_D13	P_D13	TA0_I13	TA0_O13				
P0_14	P_D14	P_D14	TA0_I14	TA0_O14				
P0_15	P_D15	P_D15	TA0_I15	TA0_O15				
Port group 1:								
P1_0	P_D16	P_D16	TA2_I0	TA2_O0		ADCNV0		
P1_1	P_D17	P_D17	TA2_I1	TA2_O1		ADCNV1		
P1_2	P_D18	P_D18	TA2_I2	TA2_O2		ADCNV2		
P1_3	P_D19	P_D19	TA2_I3	TA2_O3	ESO0			
P1_4	P_D20	P_D20	TA2_I4	TA2_O4	ESO1			
P1_5	P_D21	P_D21	TA2_I5	TA2_O5	ESO2			
P1_6	P_D22	P_D22	TA2_I6	TA2_O6	ESO3			
P1_7	P_D23	P_D23	TA2_I7	TA2_O7				
P1_8	P_D24	P_D24	TA2_I8	TA2_O8				
P1_9	P_D25	P_D25	TA2_I9	TA2_O9				
P1_10	P_D26	P_D26	TA2_I10	TA2_O10				
P1_11	P_D27	P_D27	TA2_I11	TA2_O11				
P1_12	P_D28	P_D28	TA2_I12	TA2_O12				
P1_13	P_D29	P_D29	TA2_I13	TA2_O13				
P1_14	P_D30	P_D30	TA2_I14	TA2_O14				
P1_15	P_D31	P_D31	TA2_I15	TA2_O15				
Port group 2:								
P2_0	NMI							
P2_1	INTP13	$\overline{\text{P\_LLBE}}$		$\overline{\text{P\_LLWR}}$	TJ_I0	TJ_O0		
P2_2	INTP14	$\overline{\text{P\_LUBE}}$		$\overline{\text{P\_LUWR}}$	TJ_I1	TJ_O1		
P2_3	INTP15	$\overline{\text{P\_ULBE}}$		$\overline{\text{P\_ULWR}}$	TJ_I2	TJ_O2		
P2_4	INTP16	$\overline{\text{P\_UUBE}}$		$\overline{\text{P\_UUWR}}$	TJ_I3	TJ_O3		
P2_5	INTP17	$\overline{\text{P\_RD}}$						
P2_6		P_BUSCLK						
P2_7	INTP19	P_WR		$\overline{\text{P\_RW}}$				
Port group 3:								
P3_0	TE1_TI0	P_A0	TA1_I0	TA1_O0	INTP18			

Table 8-32 List of port pins and alternative functions for port groups 0 to 13 (3/7)

Port mode	Alternative mode							
PMCn_m = 0	PMCn_m = 1							
	PFCEn_m = 0, PFCn_m = 0		PFCEn_m = 0, PFCn_m = 1		PFCEn_m = 1, PFCn_m = 0		PFCEn_m = 1, PFCn_m = 1	
	PMn_m = 1	PMn_m = 0	PMn_m = 1	PMn_m = 0	PMn_m = 1	PMn_m = 0	PMn_m = 1	PMn_m = 0
	ALT_IN1	ALT_OUT1	ALT_IN2	ALT_OUT2	ALT_IN3	ALT_OUT3	ALT_IN4	ALT_OUT4
P3_1		P_A1	TA1_I1	TA1_O1				
P3_2	TE1_TI1	P_A2	TA1_I2	TA1_O2				
P3_3		P_A3	TA1_I3	TA1_O3				
P3_4	TE1_AI	P_A4	TA1_I4	TA1_O4				
P3_5		P_A5	TA1_I5	TA1_O5				
P3_6	TE1_BI	P_A6	TA1_I6	TA1_O6				CSI3F_CS0
P3_7		P_A7	TA1_I7	TA1_O7				CSI3F_CS1
P3_8	TE1_ZI	P_A8	TA1_I8	TA1_O8				CSI3F_CS2
P3_9		P_A9	TA1_I9	TA1_O9				CSI3F_CS3
P3_10		P_A10	TA1_I10	TA1_O10				CSI3F_CS4
P3_11		P_A11	TA1_I11	TA1_O11				CSI3F_CS5
P3_12		P_A12	TA1_I12	TA1_O12				CSI3F_CS6
P3_13		P_A13	TA1_I13	TA1_O13				CSI3F_CS7
P3_14		P_A14	TA1_I14	TA1_O14			CSI3_RYI	CSI3_RYO
P3_15		P_A15	TA1_I15	TA1_O15			CSI3_SSI	
Port group 4:								
P4_0	INTP5	P_A16					CSI3F_RYI	CSI3F_RYO
P4_1	INTP6	P_A17					CSI3F_SSI	
P4_2	INTP7	P_A18						SO3
P4_3	INTP8	P_A19		$\overline{\text{DMAAK4}}$				SO3F
P4_4	INTP9	P_A20	RXD3	$\overline{\text{DMATC4}}$			SI3	
P4_5	INTP10	P_A21	ADTRG10	TXD3		$\overline{\text{DMATC3}}$	SCK3	SCK3
P4_6	INTP11	P_A22	RXD3F	$\overline{\text{DMAAK3}}$	SDA3	SDA3	SI3F	
P4_7	INTP12	P_A23	ADTRG20	TXD3F	SCL3	SCL3	SCK3F	SCK3F
P4_8	INTP20	$\overline{\text{P_CS1}}$		$\overline{\text{P_BCYST}}$				
P4_9	INTP21	$\overline{\text{P_CS2}}$		$\overline{\text{DMAAK5}}$				
P4_10	INTP22	$\overline{\text{P_CS3}}$		$\overline{\text{DMATC5}}$				
P4_11	$\overline{\text{P_WAIT}}$			TXD0F	SCL0	SCL0	SCK0F	SCK0F
P4_12	INTP23	$\overline{\text{P_HLDAK}}$		$\overline{\text{DMAAK2}}$				SO0F
P4_13	$\overline{\text{P_HLDRQ}}$		RXD0F	$\overline{\text{DMATC2}}$	SDA0	SDA0	SI0F	
Port group 5:								
P5_0	INTP20	P_LLDQM	ETH_CRS		TA3_I0	TA3_O0	CSI0F_RYI	CSI0F_RYO
P5_1	INTP21	P_LUDQM	ETH_COL		TA3_I1	TA3_O1	CSI0F_SSI	
P5_2	INTP22	P_ULDQM		ETH_TXD3	TA3_I2	TA3_O2		CSI0F_CS0
P5_3	INTP23	P_UUDQM		ETH_TXD2	TA3_I3	TA3_O3		CSI0F_CS1
P5_4	INTP24	$\overline{\text{P_REFRQ}}$		ETH_TXD1	TA3_I4	TA3_O4		CSI0F_CS2

Table 8-32 List of port pins and alternative functions for port groups 0 to 13 (4/7)

Port mode	Alternative mode							
PMCn_m = 0	PMCn_m = 1							
	PFCEn_m = 0, PFCn_m = 0		PFCEn_m = 0, PFCn_m = 1		PFCEn_m = 1, PFCn_m = 0		PFCEn_m = 1, PFCn_m = 1	
	PMn_m = 1	PMn_m = 0	PMn_m = 1	PMn_m = 0	PMn_m = 1	PMn_m = 0	PMn_m = 1	PMn_m = 0
	ALT_IN1	ALT_OUT1	ALT_IN2	ALT_OUT2	ALT_IN3	ALT_OUT3	ALT_IN4	ALT_OUT4
P5_5	INTP25	$\overline{\text{P\_SDRAS}}$		ETH_TXD0	TA3_I5	TA3_O5		CSI0F_CS3
P5_6	INTP26	$\overline{\text{P\_SDCAS}}$		ETH_TXEN	TA3_I6	TA3_O6		CSI0F_CS4
P5_7	INTP27		ETH_TXCLK		TA3_I7	TA3_O7		CSI0F_CS5
P5_8		P_SDCKE		ETH_TXER	TA3_I8	TA3_O8		CSI0F_CS6
P5_9		$\overline{\text{P\_CS4}}$	ETH_RXER		TA3_I9	TA3_O9		CSI0F_CS7
P5_10		$\overline{\text{P\_BUSRQ}}$	ETH_RXCLK		TA3_I10	TA3_O10		
P5_11		$\overline{\text{P\_SDWE}}$	ETH_RXDV		TA3_I11	TA3_O11		
P5_12		$\overline{\text{P\_BCYST}}$	ETH_RXD0		TA3_I12	TA3_O12		
P5_13			ETH_RXD1		TA3_I13	TA3_O13		
P5_14		$\overline{\text{DMAAK0}}$	ETH_RXD2		TA3_I14	TA3_O14	TJ_I0	TJ_O0
P5_15		$\overline{\text{DMAAK1}}$	ETH_RXD3		TA3_I15	TA3_O15	TJ_I1	TJ_O1
Port group 6:								
P6_0		$\overline{\text{DMATC0}}$		ETH_MDC		P_A24	TJ_I2	TJ_O2
P6_1		$\overline{\text{DMATC1}}$	ETH_MDIO	ETH_MDIO		P_A25	TJ_I3	TJ_O3
Port group 7:								
P7_0	S_D0	S_D0	TA2_I0	TA2_O0				
P7_1	S_D1	S_D1	TA2_I1	TA2_O1				
P7_2	S_D2	S_D2	TA2_I2	TA2_O2				
P7_3	S_D3	S_D3	TA2_I3	TA2_O3				
P7_4	S_D4	S_D4	TA2_I4	TA2_O4				
P7_5	S_D5	S_D5	TA2_I5	TA2_O5				
P7_6	S_D6	S_D6	TA2_I6	TA2_O6				
P7_7	S_D7	S_D7	TA2_I7	TA2_O7				
P7_8	S_D8	S_D8	TA2_I8	TA2_O8				
P7_9	S_D9	S_D9	TA2_I9	TA2_O9				
P7_10	S_D10	S_D10	TA2_I10	TA2_O10				
P7_11	S_D11	S_D11	TA2_I11	TA2_O11				
P7_12	S_D12	S_D12	TA2_I12	TA2_O12				
P7_13	S_D13	S_D13	TA2_I13	TA2_O13				
P7_14	S_D14	S_D14	TA2_I14	TA2_O14				
P7_15	S_D15	S_D15	TA2_I15	TA2_O15				
Port group 8:								
P8_0	S_D16	S_D16	TA0_I0	TA0_O0	TE_TI0			
P8_1	S_D17	S_D17	TA0_I1	TA0_O1				
P8_2	S_D18	S_D18	TA0_I2	TA0_O2	TE0_TI1			
P8_3	S_D19	S_D19	TA0_I3	TA0_O3				

Table 8-32 List of port pins and alternative functions for port groups 0 to 13 (5/7)

Port mode	Alternative mode							
PMCn_m = 0	PMCn_m = 1							
	PFCEn_m = 0, PFCn_m = 0		PFCEn_m = 0, PFCn_m = 1		PFCEn_m = 1, PFCn_m = 0		PFCEn_m = 1, PFCn_m = 1	
	PMn_m = 1	PMn_m = 0	PMn_m = 1	PMn_m = 0	PMn_m = 1	PMn_m = 0	PMn_m = 1	PMn_m = 0
	ALT_IN1	ALT_OUT1	ALT_IN2	ALT_OUT2	ALT_IN3	ALT_OUT3	ALT_IN4	ALT_OUT4
P8_4	S_D20	S_D20	TA0_I4	TA0_O4	TE0_TA			
P8_5	S_D21	S_D21	TA0_I5	TA0_O5				
P8_6	S_D22	S_D22	TA0_I6	TA0_O6	TE0_TB			
P8_7	S_D23	S_D23	TA0_I7	TA0_O7				
P8_8	S_D24	S_D24	TA0_I8	TA0_O8	TE0_TZ			
P8_9	S_D25	S_D25	TA0_I9	TA0_O9				
P8_10	S_D26	S_D26	TA0_I10	TA0_O10				
P8_11	S_D27	S_D27	TA0_I11	TA0_O11				
P8_12	S_D28	S_D28	TA0_I12	TA0_O12				
P8_13	S_D29	S_D29	TA0_I13	TA0_O13				
P8_14	S_D30	S_D30	TA0_I14	TA0_O14				
P8_15	S_D31	S_D31	TA0_I15	TA0_O15				
Port group 9:								
P9_0	INTP10	S_SDCKE			TA3_I0	TA3_O0		CSI2F_CS0
P9_1		S_BUSCLK						CSI2F_CS1
P9_2		$\overline{\text{S\_SDCAS}}$			TA3_I1	TA3_O1		CSI2F_CS2
P9_3		$\overline{\text{S\_SDRAS}}$			TA3_I2	TA3_O2		CSI2F_CS3
P9_4		S_LLDQM			TA3_I3	TA3_O3		CSI2F_CS4
P9_5		S_LUDQM			TA3_I4	TA3_O4		CSI2F_CS5
P9_6		S_ULDQM	ESO0	$\overline{\text{S\_DMATC2}}$				CSI2F_CS6
P9_7		S_UUDQM	ESO1	$\overline{\text{S\_DMATC3}}$				CSI2F_CS7
P9_8		$\overline{\text{S\_LLWR}}$		$\overline{\text{S\_DMAAK2}}$			CSI4_RYI	CSI4_RYO
P9_9	$\overline{\text{S\_DMARQ2}}$	$\overline{\text{S\_LUWR}}$		$\overline{\text{S\_DMATC0}}$				SO4
P9_10	ESO2	$\overline{\text{S\_ULWR}}$	RXD4	$\overline{\text{S\_DMAAK3}}$	SDA4	SDA4	SI4	
P9_11	$\overline{\text{S\_DMARQ3}}$	$\overline{\text{S\_UUWR}}$	ESO3	TXD4	SCL4	SCL4	SCK4	SCK4
P9_12	INTP11	$\overline{\text{S\_RD}}$			TA3_I5	TA3_O5		
P9_13	INTP12	$\overline{\text{S\_WR}}$			TA3_I6	TA3_O6	CSI1_SSI	
P9_14	INTP13	$\overline{\text{S\_LLBE}}$			TA3_I7	TA3_O7	CSI1_RYI	CSI1_RYO
P9_15	INTP14	$\overline{\text{S\_LUBE}}$						SO1
Port group 10:								
P10_0	INTP15	$\overline{\text{S\_ULBE}}$	RXD1	$\overline{\text{S\_DMATC0}}$			SI1	
P10_1	INTP16	$\overline{\text{S\_UUBE}}$		TXD1		$\overline{\text{S\_DMATC1}}$	SCK1	SCK1
P10_2	INTP17	$\overline{\text{S\_WE}}$			TA3_I8	TA3_O8	CSI2F_SSI	
P10_3	INTP18	$\overline{\text{S\_BCYST}}$		$\overline{\text{S\_DMATC1}}$	TA3_I9	TA3_O9	CSI4_SSI	
P10_4	INTP19	$\overline{\text{S\_CS0}}$			TA3_I10	TA3_O10	CSI2_SSI	
P10_5	INTP24	$\overline{\text{S\_CS1}}$			TA3_I11	TA3_O11	CSI2_RYI	CSI2_RYO

Table 8-32 List of port pins and alternative functions for port groups 0 to 13 (6/7)

Port mode	Alternative mode							
PMCn_m = 0	PMCn_m = 1							
	PFCEn_m = 0, PFCn_m = 0		PFCEn_m = 0, PFCn_m = 1		PFCEn_m = 1, PFCn_m = 0		PFCEn_m = 1, PFCn_m = 1	
	PMn_m = 1	PMn_m = 0	PMn_m = 1	PMn_m = 0	PMn_m = 1	PMn_m = 0	PMn_m = 1	PMn_m = 0
	ALT_IN1	ALT_OUT1	ALT_IN2	ALT_OUT2	ALT_IN3	ALT_OUT3	ALT_IN4	ALT_OUT4
P10_6	INTP25	$\overline{S\_CS2}$			TA3_I12	TA3_O12	CSI1F_SSI	
P10_7	INTP26	$\overline{S\_CS3}$				$\overline{S\_SDCS}$	CSI1F_RYI	CSI1F_RYO
P10_8	$\overline{S\_WAIT}$		RXD2				SI2	
P10_9	INTP27	$\overline{S\_HLDAK}$			TA3_I13	TA3_O13		SO2
P10_10	$\overline{S\_HLDRQ}$			TXD2	TA3_I14	TA3_O14	SCK2	SCK2
P10_11		$\overline{S\_REFRQ}$			TA3_I15	TA3_O15	CSI2F_RYI	CSI2F_RYO
Port group 11:								
P11_0	TE1_TI0	S_A1	TA1_I0	TA1_O0				
P11_1		S_A2	TA1_I1	TA1_O1				
P11_2	TE1_TI1	S_A3	TA1_I2	TA1_O2				
P11_3		S_A4	TA1_I3	TA1_O3				
P11_4	TE1_AI	S_A5	TA1_I4	TA1_O4				
P11_5		S_A6	TA1_I5	TA1_O5				
P11_6	TE1_BI	S_A7	TA1_I6	TA1_O6				
P11_7		S_A8	TA1_I7	TA1_O7				
P11_8	TE1_ZI	S_A9	TA1_I8	TA1_O8				
P11_9		S_A10	TA1_I9	TA1_O9				
P11_10		S_A11	TA1_I10	TA1_O10				CSI1F_CS0
P11_11		S_A12	TA1_I11	TA1_O11				CSI1F_CS1
P11_12		S_A13	TA1_I12	TA1_O12				CSI1F_CS2
P11_13		S_A14	TA1_I13	TA1_O13				CSI1F_CS3
P11_14		S_A15	TA1_I14	TA1_O14				CSI1F_CS4
P11_15		S_A16	TA1_I15	TA1_O15				CSI1F_CS5
Port group 12:								
P12_0	INTP0	S_A17		ADCNV0				CSI1F_CS6
P12_1	INTP1	S_A18		ADCNV1				CSI1F_CS7
P12_2	INTP2	S_A19		ADCNV2			CSI0_SSI	
P12_3	INTP3	S_A20					CSI0_RYI	CSI0_RYO
P12_4	INTP4	S_A21						SO0
P12_5	INTP5	S_A22	RXD0				SI0	
P12_6	INTP6	S_A23	ADTRG11	TXD0			SCK0	SCK0
P12_7	INTP7	S_A24						SO2F
P12_8	INTP8	S_A25	RXD2F		SDA2	SDA2	SI2F	
P12_9	INTP9	S_A26	ADTRG21	TXD2F	SCL2	SCL2	SCK2F	SCK2F

Table 8-32 List of port pins and alternative functions for port groups 0 to 13 (7/7)

Port mode	Alternative mode							
PMCn_m = 0	PMCn_m = 1							
	PFCEn_m = 0, PFCn_m = 0		PFCEn_m = 0, PFCn_m = 1		PFCEn_m = 1, PFCn_m = 0		PFCEn_m = 1, PFCn_m = 1	
	PMn_m = 1	PMn_m = 0	PMn_m = 1	PMn_m = 0	PMn_m = 1	PMn_m = 0	PMn_m = 1	PMn_m = 0
	ALT_IN1	ALT_OUT1	ALT_IN2	ALT_OUT2	ALT_IN3	ALT_OUT3	ALT_IN4	ALT_OUT4
Port group 13:								
P13_0	UCLK		ADTRG01		$\overline{\text{S\_DMARQ0}}$		CSI5_SSI	
P13_1	INTP0	$\overline{\text{S\_CS3}}$	ADTRG00			$\overline{\text{S\_DMAAK0}}$	CSI5_RYI	CSI5_RYO
P13_2	INTP1				$\overline{\text{S\_DMARQ1}}$	PPON		SO1F
P13_3	INTP2				$\overline{\text{OCI}}$	$\overline{\text{S\_DMAAK1}}$		SO5
P13_4	INTP3	CAN0TXD		TXD1F	SCL1	SCL1	SCK1F	SCK1F
P13_5	CAN0RXD		RXD1F		SDA1	SDA1	SI1F	
P13_6	INTP4	CAN1TXD		TXD5	SCL5	SCL5	SCK5	SCK5
P13_7	CAN1RXD		RXD5		SDA5	SDA5	SI5	
Port group 14:								
P14_0	ANI06							
P14_1	ANI07							
P14_2	ANI08							
P14_3	ANI09							
P14_4	ANI010							
P14_5	ANI011							

Table 8-33 List of port pins and alternative functions for port group 14

PIBC14 = 1 (Input mode)	PIBC14 = 0 (Alternative mode)
Port group 14:	
P14_0	ANI06
P14_1	ANI07
P14_2	ANI08
P14_3	ANI09
P14_4	ANI010
P14_5	ANI011

8.4.3 V850E2/MN4 port control registers

Table 8-34 "V850E2/MN4 port control registers for groups 0 to 3" through *Table 8-37 "V850E2/MN4 port control registers for groups 12 to 14"* list the registers for controlling the V850E2/MN4 ports, as well as their address and initial value. When writing to an invalid bit (indicated as "X"), always write the initial value.

Conventions

A: Register address

I: Initial value

B: Valid or invalid

– 1: Valid, X: Invalid, –: Not supported

Right: bit 0 Left: bit 15

Table 8-34 V850E2/MN4 port control registers for groups 0 to 3 (1/2)

Register		Port group n =			
		0	1	2	3
Pn	A:	FFFF 8000$_H$	FFFF 8004$_H$	FFFF 8008$_H$	FFFF 800C$_H$
	I:	0000$_H$	0000$_H$	0000$_H$	0000$_H$
	B:	1111 1111 1111 1111	1111 1111 1111 1111	xxxx xxxx 1111 1111	1111 1111 1111 1111
PSRn	A:	FFFF 8100$_H$	FFFF 8104$_H$	FFFF 8108$_H$	FFFF 810C$_H$
	I:	0000 0000$_H$	0000 0000$_H$	0000 0000$_H$	0000 0000$_H$
	B:	1111 1111 1111 1111	1111 1111 1111 1111	xxxx xxxx 1111 1111	1111 1111 1111 1111
PNOTn	A:	FFFF 8700$_H$	FFFF 8704$_H$	FFFF 8708$_H$	FFFF 870C$_H$
	I:	0000$_H$	0000$_H$	0000$_H$	0000$_H$
	B:	1111 1111 1111 1111	1111 1111 1111 1111	xxxx xxxx 1111 1111	1111 1111 1111 1111
PPRn	A:	FFFF 8200$_H$	FFFF 8204$_H$	FFFF 8208$_H$	FFFF 820C$_H$
	I:	0000$_H$	0000$_H$	0000$_H$[a]	0000$_H$
	B:	1111 1111 1111 1111	1111 1111 1111 1111	xxxx xxxx 1111 1111	1111 1111 1111 1111
PMn	A:	FFFF 8300$_H$	FFFF 8304$_H$	FFFF 8308$_H$	FFFF 830C$_H$
	I:	FFFF$_H$	FFFF$_H$	FFBF$_H$	FFFF$_H$
	B:	1111 1111 1111 1111	1111 1111 1111 1111	xxxx xxxx 1111 1111	1111 1111 1111 1111
PMCn	A:	FFFF 8400$_H$	FFFF 8404$_H$	FFFF 8408$_H$	FFFF 840C$_H$
	I:	0000$_H$	0000$_H$	0040$_H$	0000$_H$
	B:	1111 1111 1111 1111	1111 1111 1111 1111	xxxx xxxx 1111 1111	1111 1111 1111 1111
PFCn	A:	FFFF 8500$_H$	FFFF 8504$_H$	FFFF 8508$_H$	FFFF 850C$_H$
	I:	0000$_H$	0000$_H$	0000$_H$	0000$_H$
	B:	1111 1111 1111 1111	1111 1111 1111 1111	xxxx xxxx 1111 1111	1111 1111 1111 1111
PFCEn	A:	FFFF 8600$_H$	FFFF 8604$_H$	FFFF 8608$_H$	FFFF 860C$_H$
	I:	0000$_H$	0000$_H$	0000$_H$	0000$_H$
	B:	1111 1111 1111 1111	1111 1111 1111 1111	xxxx xxxx 1111 1111	1111 1111 1111 1111
PMSRn	A:	FFFF 8800$_H$	FFFF 8804$_H$	FFFF 8808$_H$	FFFF 880C$_H$
	I:	0000 FFFF$_H$	0000 FFFF$_H$	0000 FFBF$_H$	0000 FFFF$_H$
	B:	1111 1111 1111 1111	1111 1111 1111 1111	xxxx xxxx 1111 1111	1111 1111 1111 1111

Table 8-34 V850E2/MN4 port control registers for groups 0 to 3 (2/2)

Register		Port group n =			
		0	1	2	3
PMCSRn	A:	FFFF 8900_H	FFFF 8904_H	FFFF 8908_H	FFFF $890C_H$
	I:	0000 0000_H	0000 0000_H	0000 0040_H	0000 0000_H
	B:	1111 1111 1111 1111	1111 1111 1111 1111	xxxx xxxx 1111 1111	1111 1111 1111 1111
PIBCn	A:	FF40 4000_H	FF40 4004_H	FF40 4008_H	FF40 $400C_H$
	I:	0000_H	0000_H	0000_H	0000_H
	B:	1111 1111 1111 1111	1111 1111 1111 1111	xxxx xxxx 1111 1111	1111 1111 1111 1111
PBDCn	A:	FF40 4100_H	FF40 4104_H	FF40 4108_H	FF40 $410C_H$
	I:	0000_H	0000_H	0000_H	0000_H
	B:	1111 1111 1111 1111	1111 1111 1111 1111	xxxx xxxx 1111 1111	1111 1111 1111 1111
PIPCn	A:	FF40 4200_H	FF40 4204_H	FF40 4208_H	FF40 $420C_H$
	I:	0000_H	0000_H	0000_H	0000_H
	B:	1111 1111 1111 1111	1111 1111 1111 1111	xxxx xxxx 1111 1111	1111 1111 1111 1111
PUn	A:	–	–	FF40 4308_H	FF40 $430C_H$
	I:	–	–	0000_H	0000_H
	B:	–	–	xxxx xxxx 1111 1111	1111 1111 1111 1111
PDn	A:	–	–	FF40 4408_H	FF40 $440C_H$
	I:	–	–	0000_H	0000_H
	B:	–	–	xxxx xxxx 1111 1111	1111 1111 1111 1111
PODCn	A:	FF40 4500_H	FF40 4504_H	FF40 4508_H	FF40 $450C_H$
	I:	0000 0000_H	0000 0000_H	0000 0000_H	0000 0000_H
	B:	1111 1111 1111 1111	1111 1111 1111 1111	xxxx xxxx 1111 1111	1111 1111 1111 1111
PDSCn	A:	FF40 4600_H	FF40 4604_H	FF40 4608_H	FF40 $460C_H$
	I:	0000 0000_H	0000 0000_H	0000 0040_H	0000 0000_H
	B:	1111 1111 1111 1111	1111 1111 1111 1111	xxxx xxxx 1111 1111	1111 1111 1111 1111
PISn	A:	–	–	FF40 4708_H	–
	I:	–	–	0000_H	–
	B:	–	–	xxxx xxxx 1111 1111	–
PISEn	A:	–	–	FF40 4808_H	–
	I:	–	–	0000_H	–
	B:	–	–	xxxx xxxx 1111 1111	–
PISAn	A:	FF40 $4A00_H$	FF40 $4A04_H$	FF40 $4A08_H$	–
	I:	0000_H	0000_H	0000_H	–
	B:	1111 1111 1111 1111	1111 1111 1111 1111	1111 1111 1111 1111	–
PPCMDn	A:	FF40 $4C00_H$	FF40 $4C04_H$	FF40 $4C08_H$	FF40 $4C0C_H$
	I:	00_H	00_H	00_H	00_H
	B:	1111 1111	1111 1111	1111 1111	1111 1111
PPROTSn	A:	FF40 $4B00_H$	FF40 $4B04_H$	FF40 $4B08_H$	FF40 $4B0C_H$
	I:	00_H	00_H	00_H	00_H
	B:	xxxx xxx1	xxxx xxx1	xxxx xxx1	xxxx xxx1

a) Bit 6 depends on the state of P_BUSCLK.

Table 8-35 V850E2/MN4 port control registers for groups 4 to 7 (1/2)

Register		Port group n =			
		4	5	6	7
Pn	A:	FFFF 8010$_H$	FFFF 8014$_H$	FFFF 8018$_H$	FFFF 801C$_H$
	I:	0000$_H$	0000$_H$	0000$_H$	0000$_H$
	B:	xx11 1111 1111 1111	1111 1111 1111 1111	xxxx xxxx xxxx xx11	1111 1111 1111 1111
PSRn	A:	FFFF 8110$_H$	FFFF 8114$_H$	FFFF 8118$_H$	FFFF 811C$_H$
	I:	0000 0000$_H$	0000 0000$_H$	0000 0000$_H$	0000 0000$_H$
	B:	xx11 1111 1111 1111	1111 1111 1111 1111	xxxx xxxx xxxx xx11	1111 1111 1111 1111
PNOTn	A:	FFFF 8710$_H$	FFFF 8714$_H$	FFFF 8718$_H$	FFFF 871C$_H$
	I:	0000$_H$	0000$_H$	0000$_H$	0000$_H$
	B:	xx11 1111 1111 1111	1111 1111 1111 1111	xxxx xxxx xxxx xx11	1111 1111 1111 1111
PPRn	A:	FFFF 8210$_H$	FFFF 8214$_H$	FFFF 8218$_H$	FFFF 821C$_H$
	I:	0000$_H$	0000$_H$	0000$_H$	0000$_H$
	B:	xx11 1111 1111 1111	1111 1111 1111 1111	xxxx xxxx xxxx xx11	1111 1111 1111 1111
PMn	A:	FFFF 8310$_H$	FFFF 8314$_H$	FFFF 8318$_H$	FFFF 831C$_H$
	I:	FFFF$_H$	FFFF$_H$	FFFF$_H$	FFFF$_H$
	B:	xx11 1111 1111 1111	1111 1111 1111 1111	xxxx xxxx xxxx xx11	1111 1111 1111 1111
PMCn	A:	FFFF 8410$_H$	FFFF 8414$_H$	FFFF 8418$_H$	FFFF 841C$_H$
	I:	0000$_H$	0000$_H$	0000$_H$	0000$_H$
	B:	xx11 1111 1111 1111	1111 1111 1111 1111	xxxx xxxx xxxx xx11	1111 1111 1111 1111
PFCn	A:	FFFF 8510$_H$	FFFF 8514$_H$	FFFF 8518$_H$	FFFF 851C$_H$
	I:	0000$_H$	0000$_H$	0000$_H$	0000$_H$
	B:	xx11 1111 1111 1111	1111 1111 1111 1111	xxxx xxxx xxxx xx11	1111 1111 1111 1111
PFCEn	A:	FFFF 8610$_H$	FFFF 8614$_H$	FFFF 8618$_H$	FFFF 861C$_H$
	I:	0000$_H$	0000$_H$	0000$_H$	0000$_H$
	B:	xx11 1111 1111 1111	1111 1111 1111 1111	xxxx xxxx xxxx xx11	1111 1111 1111 1111
PMSRn	A:	FFFF 8810$_H$	FFFF 8814$_H$	FFFF 8818$_H$	FFFF 881C$_H$
	I:	0000 FFFF$_H$	0000 FFFF$_H$	0000 FFFF$_H$	0000 FFFF$_H$
	B:	xx11 1111 1111 1111	1111 1111 1111 1111	xxxx xxxx xxxx xx11	1111 1111 1111 1111
PMCSRn	A:	FFFF 8910$_H$	FFFF 8914$_H$	FFFF 8918$_H$	FFFF 891C$_H$
	I:	0000 0000$_H$	0000 0000$_H$	0000 0000$_H$	0000 0000$_H$
	B:	xx11 1111 1111 1111	1111 1111 1111 1111	xxxx xxxx xxxx xx11	1111 1111 1111 1111
PIBCn	A:	FF40 4010$_H$	FF40 4014$_H$	FF40 4018$_H$	FF40 401C$_H$
	I:	0000$_H$	0000$_H$	0000$_H$	0000$_H$
	B:	xx11 1111 1111 1111	1111 1111 1111 1111	xxxx xxxx xxxx xx11	1111 1111 1111 1111
PBDCn	A:	FF40 4110$_H$	FF40 4114$_H$	FF40 4118$_H$	FF40 411C$_H$
	I:	0000$_H$	0000$_H$	0000$_H$	0000$_H$
	B:	xx11 1111 1111 1111	1111 1111 1111 1111	xxxx xxxx xxxx xx11	1111 1111 1111 1111
PIPCn	A:	FF40 4210$_H$	FF40 4214$_H$	FF40 4218$_H$	FF40 421C$_H$
	I:	0000$_H$	0000$_H$	0000$_H$	0000$_H$
	B:	xx11 1111 1111 1111	1111 1111 1111 1111	xxxx xxxx xxxx xx11	1111 1111 1111 1111

Table 8-35 V850E2/MN4 port control registers for groups 4 to 7 (2/2)

Register		Port group n =			
		4	5	6	7
PUn	A:	FF40 4310_H	FF40 4314_H	FF40 4318_H	FF40 $431C_H$
	I:	0000_H	0000_H	0000_H	0000_H
	B:	xx11 1111 1111 1111	1111 1111 1111 1111	xxxx xxxx xxxx xx11	1111 1111 1111 1111
PDn	A:	FF40 4410_H	FF40 4414_H	FF40 4418_H	FF40 $441C_H$
	I:	0000_H	0000_H	0000_H	0000_H
	B:	xx11 1111 1111 1111	1111 1111 1111 1111	xxxx xxxx xxxx xx11	1111 1111 1111 1111
PODCn	A:	FF40 4510_H	FF40 4514_H	FF40 4518_H	FF40 $451C_H$
	I:	0000 0000_H	0000 0000_H	0000 0000_H	0000 0000_H
	B:	xx11 1111 1111 1111	1111 1111 1111 1111	xxxx xxxx xxxx xx11	1111 1111 1111 1111
PDSCn	A:	FF40 4610_H	FF40 4614_H	FF40 4618_H	FF40 $461C_H$
	I:	0000 0000_H	0000 0000_H	0000 0000_H	0000 0000_H
	B:	xx11 1111 1111 1111	1111 1111 1111 1111	xxxx xxxx xxxx xx11	1111 1111 1111 1111
PISn	A:	FF40 4710_H	–	–	–
	I:	0000_H	–	–	–
	B:	xx11 1111 1111 1111	–	–	–
PISEn	A:	FF40 4810_H	–	–	–
	I:	0000_H	–	–	–
	B:	xx11 1111 1111 1111	–	–	–
PISAn	A:	FF40 $4A10_H$	–	–	FF40 $4A1C_H$
	I:	0000_H	–	–	0000_H
	B:	xx11 1111 1111 1111	–	–	1111 1111 1111 1111
PPCMDn	A:	FF40 $4C10_H$	FF40 $4C14_H$	FF40 $4C18_H$	FF40 $4C1C_H$
	I:	00_H	00_H	00_H	00_H
	B:	1111 1111	1111 1111	1111 1111	1111 1111
PPROTSn	A:	FF40 $4B10_H$	FF40 $4B14_H$	FF40 $4B18_H$	FF40 $4B1C_H$
	I:	00_H	00_H	00_H	00_H
	B:	xxxx xxx1	xxxx xxx1	xxxx xxx1	xxxx xxx1

Table 8-36 V850E2/MN4 port control registers for groups 8 to 11 (1/2)

Register		Port group n =			
		8	9	10	11
Pn	A:	FFFF 8020$_H$	FFFF 8024$_H$	FFFF 8028$_H$	FFFF 802C$_H$
	I:	0000$_H$	0000$_H$	0000$_H$	0000$_H$
	B:	1111 1111 1111 1111	1111 1111 1111 1111	xxxx 1111 1111 1111	1111 1111 1111 1111
PSRn	A:	FFFF 8120$_H$	FFFF 8124$_H$	FFFF 8128$_H$	FFFF 812C$_H$
	I:	0000 0000$_H$	0000 0000$_H$	0000 0000$_H$	0000 0000$_H$
	B:	1111 1111 1111 1111	1111 1111 1111 1111	xxxx 1111 1111 1111	1111 1111 1111 1111
PNOTn	A:	FFFF 8720$_H$	FFFF 8724$_H$	FFFF 8728$_H$	FFFF 872C$_H$
	I:	0000$_H$	0000$_H$	0000$_H$	0000$_H$
	B:	1111 1111 1111 1111	1111 1111 1111 1111	xxxx 1111 1111 1111	1111 1111 1111 1111
PPRn	A:	FFFF 8220$_H$	FFFF 8224$_H$	FFFF 8228$_H$	FFFF 822C$_H$
	I:	0000$_H$	0000$_H$[a]	0000$_H$	0000$_H$
	B:	1111 1111 1111 1111	1111 1111 1111 1111	xxxx 1111 1111 1111	1111 1111 1111 1111
PMn	A:	FFFF 8320$_H$	FFFF 8324$_H$	FFFF 8328$_H$	FFFF 832C$_H$
	I:	FFFF$_H$	FFFD$_H$	FFFF$_H$	FFFF$_H$
	B:	1111 1111 1111 1111	1111 1111 1111 1111	xxxx 1111 1111 1111	1111 1111 1111 1111
PMCn	A:	FFFF 8420$_H$	FFFF 8424$_H$	FFFF 8428$_H$	FFFF 842C$_H$
	I:	0000$_H$	0002$_H$	0000$_H$	0000$_H$
	B:	1111 1111 1111 1111	1111 1111 1111 1111	xxxx 1111 1111 1111	1111 1111 1111 1111
PFCn	A:	FFFF 8520$_H$	FFFF 8524$_H$	FFFF 8528$_H$	FFFF 852C$_H$
	I:	0000$_H$	0000$_H$	0000$_H$	0000$_H$
	B:	1111 1111 1111 1111	1111 1111 1111 1111	xxxx 1111 1111 1111	1111 1111 1111 1111
PFCEn	A:	FFFF 8620H	FFFF 8624H	FFFF 8628H	FFFF 862CH
	I:	0000$_H$	0000$_H$	0000$_H$	0000$_H$
	B:	1111 1111 1111 1111	1111 1111 1111 1111	xxxx 1111 1111 1111	1111 1111 1111 1111
PMSRn	A:	FFFF 8820H	FFFF 8824H	FFFF 8828H	FFFF 882CH
	I:	0000 FFFF$_H$	0000 FFFD$_H$	0000 FFFF$_H$	0000 FFFF$_H$
	B:	1111 1111 1111 1111	1111 1111 1111 1111	xxxx 1111 1111 1111	1111 1111 1111 1111
PMCSRn	A:	FFFF 8920H	FFFF 8924H	FFFF 8928H	FFFF 892CH
	I:	0000 0000$_H$	0000 0002$_H$	0000 0000$_H$	0000 0000$_H$
	B:	1111 1111 1111 1111	1111 1111 1111 1111	xxxx 1111 1111 1111	1111 1111 1111 1111
PIBCn	A:	FF40 4020H	FF40 4024H	FF40 4028H	FF40 402CH
	I:	0000$_H$	0000$_H$	0000$_H$	0000$_H$
	B:	1111 1111 1111 1111	1111 1111 1111 1111	xxxx 1111 1111 1111	1111 1111 1111 1111
PBDCn	A:	FF40 4120H	FF40 4124H	FF40 4128H	FF40 412CH
	I:	0000$_H$	0000$_H$	0000$_H$	0000$_H$
	B:	1111 1111 1111 1111	1111 1111 1111 1111	xxxx 1111 1111 1111	1111 1111 1111 1111
PIPCn	A:	FF40 4220H	FF40 4224H	FF40 4228H	FF40 422CH
	I:	0000$_H$	0000$_H$	0000$_H$	0000$_H$
	B:	1111 1111 1111 1111	1111 1111 1111 1111	xxxx 1111 1111 1111	1111 1111 1111 1111

Table 8-36 V850E2/MN4 port control registers for groups 8 to 11 (2/2)

Register		Port group n =			
		8	9	10	11
PUn	A:	FF40 4320H	FF40 4324H	FF40 4328H	FF40 432CH
	I:	0000_H	0000_H	0000_H	0000_H
	B:	1111 1111 1111 1111	1111 1111 1111 1111	xxxx 1111 1111 1111	1111 1111 1111 1111
PDn	A:	FF40 4420H	FF40 4424H	FF40 4428H	FF40 442CH
	I:	0000_H	0000_H	0000_H	0000_H
	B:	1111 1111 1111 1111	1111 1111 1111 1111	xxxx 1111 1111 1111	1111 1111 1111 1111
PODCn	A:	FF40 4520H	FF40 4524H	FF40 4528H	FF40 452CH
	I:	0000 0000_H	0000 0000_H	0000 0000_H	0000 0000_H
	B:	1111 1111 1111 1111	1111 1111 1111 1111	xxxx 1111 1111 1111	1111 1111 1111 1111
PDSCn	A:	FF40 4620H	FF40 4624H	FF40 4628H	FF40 462CH
	I:	0000 0000_H	0000 0002_H	0000 0000_H	0000 0000_H
	B:	1111 1111 1111 1111	1111 1111 1111 1111	xxxx 1111 1111 1111	1111 1111 1111 1111
PISn	A:	–	FF40 4724H	FF40 4728H	–
	I:	–	0000_H	0000_H	–
	B:	–	1111 1111 1111 1111	xxxx 1111 1111 1111	–
PISEn	A:	–	FF40 4824H	FF40 4828H	–
	I:	–	0000_H	0000_H	–
	B:	–	1111 1111 1111 1111	xxxx 1111 1111 1111	–
PISAn	A:	FF40 4A20H	FF40 4A24H	FF40 4A28H	–
	I:	0000_H	0000_H	0000_H	–
	B:	1111 1111 1111 1111	1111 1111 1111 1111	xxxx 1111 1111 1111	–
PPCMDn	A:	FF40 4C20H	FF40 4C24H	FF40 4C28H	FF40 4C2CH
	I:	00_H	00_H	00_H	00_H
	B:	1111 1111	1111 1111	1111 1111	1111 1111
PPROTSn	A:	FF40 4B20H	FF40 4B24H	FF40 4B28H	FF40 4B2CH
	I:	00_H	00_H	00_H	00_H
	B:	xxxx xxx1	xxxx xxx1	xxxx xxx1	xxxx xxx1

a) Bit 1 depends on the S_BUSCLK state.

Table 8-37 V850E2/MN4 port control registers for groups 12 to 14 (1/2)

Register		Port group n =		
		12	13	14
Pn	A:	FFFF 8030H	FFFF 8034H	–
	I:	0000_H	0000_H	–
	B:	xxxx xx11 1111 1111	xxxx xxxx 1111 1111	–
PSRn	A:	FFFF 8130H	FFFF 8134H	–
	I:	0000 0000_H	0000 0000_H	–
	B:	xxxx xx11 1111 1111	xxxx xxxx 1111 1111	–
PNOTn	A:	FFFF 8730H	FFFF 8734H	–
	I:	0000_H	0000_H	–
	B:	xxxx xx11 1111 1111	xxxx xxxx 1111 1111	–
PPRn	A:	FFFF 8230H	FFFF 8234H	FFFF 8238H
	I:	0000_H	0000_H	0000_H
	B:	xxxx xx11 1111 1111	xxxx xxxx 1111 1111	xxxx xxxx xx11 1111
PMn	A:	FFFF 8330H	FFFF 8334H	–
	I:	$FFFF_H$	$FFFF_H$	–
	B:	xxxx xx11 1111 1111	xxxx xxxx 1111 1111	–
PMCn	A:	FFFF 8430H	FFFF 8434H	–
	I:	0000_H	0000_H	–
	B:	xxxx xx11 1111 1111	xxxx xxxx 1111 1111	–
PFCn	A:	FFFF 8530H	FFFF 8534H	–
	I:	0000_H	0000_H	–
	B:	xxxx xx11 1111 1111	xxxx xxxx 1111 1111	–
PFCEn	A:	FFFF 8630H	FFFF 8634H	–
	I:	0000_H	0000_H	–
	B:	xxxx xx11 1111 1111	xxxx xxxx 1111 1111	–
PMSRn	A:	FFFF 8830H	FFFF 8834H	–
	I:	0000 $FFFF_H$	0000 $FFFF_H$	–
	B:	xxxx xx11 1111 1111	xxxx xxxx 1111 1111	–
PMCSRn	A:	FFFF 8930H	FFFF 8934H	–
	I:	0000 0000_H	0000 0000_H	–
	B:	xxxx xx11 1111 1111	xxxx xxxx 1111 1111	–
PIBCn	A:	FF40 4030H	FF40 4034H	FF40 4038H
	I:	0000_H	0000_H	0000_H
	B:	xxxx xx11 1111 1111	xxxx xxxx 1111 1111	xxxx xxxx xx11 1111
PBDCn	A:	FF40 4130H	FF40 4134H	–
	I:	0000_H	0000_H	–
	B:	xxxx xx11 1111 1111	xxxx xxxx 1111 1111	–
PIPCn	A:	FF40 4230H	FF40 4234H	–
	I:	0000_H	0000_H	–
	B:	xxxx xx11 1111 1111	xxxx xxxx 1111 1111	–

Table 8-37 V850E2/MN4 port control registers for groups 12 to 14 (2/2)

Register		Port group n =		
		12	13	14
PUn	A:	FF40 4330H	FF40 4334H	–
	I:	0000_H	0000_H	–
	B:	xxxx xx11 1111 1111	xxxx xxxx 1111 1111	–
PDn	A:	FF40 4430H	FF40 4434H	–
	I:	0000_H	0000_H	–
	B:	xxxx xx11 1111 1111	xxxx xxxx 1111 1111	–
PODCn	A:	FF40 4530H	FF40 4534H	–
	I:	0000 0000_H	0000 0000_H	–
	B:	xxxx xx11 1111 1111	xxxx xxxx 1111 1111	–
PDSCn	A:	FF40 4630H	FF40 4634H	–
	I:	0000 0000_H	0000 0000_H	–
	B:	xxxx xx11 1111 1111	xxxx xxxx 1111 1111	–
PISn	A:	FF40 4730H	FF40 4734H	–
	I:	0000_H	0000_H	–
	B:	xxxx xx11 1111 1111	xxxx xxxx 1111 1111	–
PISEn	A:	FF40 4830H	FF40 4834H	–
	I:	0000_H	0000_H	–
	B:	xxxx xx11 1111 1111	xxxx xxxx 1111 1111	–
PISAn	A:	–	FF40 4A34H	–
	I:	–	0000_H	–
	B:	–	xxxx xxxx 1111 1111	–
PPCMDn	A:	FF40 4C30H	FF40 4C34H	–
	I:	00_H	00_H	–
	B:	1111 1111	1111 1111	–
PPROTSn	A:	FF40 4B30H	FF40 4B34H	–
	I:	00_H	00_H	–
	B:	xxxx xxx1	xxxx xxx1	–

8.4.4 V850E2/MN4 port control register settings

Table 8-38 "Register setting for each pin function" shows the setting of each register for alternative functions of the pins.

When using each function, be sure to specify the corresponding bits of the control register by referring to this table.

Be sure to specify "0" for bits indicated by "*0*", and "1" for bits indicated by "*1*".

Specify "0" or "1" as necessary for bits indicated by *"1/0"*.

When it is necessary to write a value to a bit that is indicated by "*N/A*", be sure to specify the initial value. Do not access bits indicated by "–" because no function is assigned for these bits in the corresponding register.

Pins that have an I/O function are divided into pins that require the settings for input and output and pins that require the setting for either input or output.

For pins that require the settings for input and output, the registers that have the specified setting for input and output must be set up with the appropriate value.

For pins that require the settings for input or output, specify the register setting according to the function used.

Table 8-38 Register setting for each pin function (1/63)

n	m	Function	Pin name	Feature							Characteristics						
Group	Bit			PMCnm	PMnm	PFCnm	PFCEnm	PIBCnm	PIPCnm	PBDCnm	PUnm	PDnm	PDSCnm	PODCnm	PISnm	PISEnm	PISAnm
0	0	Port mode (input)	P0_0	0	1	N/A	N/A	1	N/A	N/A	–	–	N/A	N/A	–	–	1
		Port mode (output)	P0_0	0	0	N/A	N/A	N/A	N/A	1/0	–	–	0	1/0	–	–	N/A
		ALT1-IN	P_D0[a]	1	N/A	0	0	N/A	1	N/A	–	–	N/A	N/A	–	–	1
		ALT2-IN	TA0_I0	1	1	1	0	N/A	N/A	N/A	–	–	N/A	N/A	–	–	1
		ALT3-IN	TE0_TI0	1	1	0	1	N/A	N/A	N/A	–	–	N/A	N/A	–	–	1
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT1-OUT	P_D0[a]	1	N/A	0	0	N/A	1	1/0	–	–	1	1/0	–	–	N/A
		ALT2-OUT	TA0_O0	1	0	1	0	N/A	N/A	1/0	–	–	0	1/0	–	–	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
0	1	Port mode (input)	P0_1	0	1	N/A	N/A	1	N/A	N/A	–	–	N/A	N/A	–	–	1
		Port mode (output)	P0_1	0	0	N/A	N/A	N/A	N/A	1/0	–	–	0	1/0	–	–	N/A
		ALT1-IN	P_D1[a]	1	N/A	0	0	N/A	1	N/A	–	–	N/A	N/A	–	–	1
		ALT2-IN	TA0_I1	1	1	1	0	N/A	N/A	N/A	–	–	N/A	N/A	–	–	1
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT1-OUT	P_D1[a]	1	N/A	0	0	N/A	1	1/0	–	–	1	1/0	–	–	N/A
		ALT2-OUT	TA0_O1	1	0	1	0	N/A	N/A	1/0	–	–	0	1/0	–	–	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A

Table 8-38 Register setting for each pin function (2/63)

n	m	Function	Pin name	Feature							Characteristics						
Group	Bit			PMCnm	PMnm	PFCnm	PFCEnm	PIBCnm	PIPCnm	PBDCnm	PUnm	PDnm	PDSCnm	PODCnm	PISnm	PISEnm	PISAnm
0	2	Port mode (input)	P0_2	0	1	N/A	N/A	1	N/A	N/A	–	–	N/A	N/A	–	–	1
		Port mode (output)	P0_2	0	0	N/A	N/A	N/A	N/A	1/0	–	–	0	1/0	–	–	N/A
		ALT1-IN	P_D2[a]	1	N/A	0	0	N/A	1	N/A	–	–	N/A	N/A	–	–	1
		ALT2-IN	TA0_I2	1	1	1	0	N/A	N/A	N/A	–	–	N/A	N/A	–	–	1
		ALT3-IN	TE0_TI1	1	1	0	1	N/A	N/A	N/A	–	–	N/A	N/A	–	–	1
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT1-OUT	P_D2[a]	1	N/A	0	0	N/A	1	1/0	–	–	1	1/0	–	–	N/A
		ALT2-OUT	TA0_O2	1	0	1	0	N/A	N/A	1/0	–	–	0	1/0	–	–	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
0	3	Port mode (input)	P0_3	0	1	N/A	N/A	1	N/A	N/A	–	–	N/A	N/A	–	–	1
		Port mode (output)	P0_3	0	0	N/A	N/A	N/A	N/A	1/0	–	–	0	1/0	–	–	N/A
		ALT1-IN	P_D3[a]	1	N/A	0	0	N/A	1	N/A	–	–	N/A	N/A	–	–	1
		ALT2-IN	TA0_I3	1	1	1	0	N/A	N/A	N/A	–	–	N/A	N/A	–	–	1
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT1-OUT	P_D3[a]	1	N/A	0	0	N/A	1	1/0	–	–	1	1/0	–	–	N/A
		ALT2-OUT	TA0_O3	1	0	1	0	N/A	N/A	1/0	–	–	0	1/0	–	–	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
0	4	Port mode (input)	P0_4	0	1	N/A	N/A	1	N/A	N/A	–	–	N/A	N/A	–	–	1
		Port mode (output)	P0_4	0	0	N/A	N/A	N/A	N/A	1/0	–	–	0	1/0	–	–	N/A
		ALT1-IN	P_D4[a]	1	N/A	0	0	N/A	1	N/A	–	–	N/A	N/A	–	–	1
		ALT2-IN	TA0_I4	1	1	1	0	N/A	N/A	N/A	–	–	N/A	N/A	–	–	1
		ALT3-IN	TE0_AI	1	1	0	1	N/A	N/A	N/A	–	–	N/A	N/A	–	–	1
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT1-OUT	P_D4[a]	1	N/A	0	0	N/A	1	1/0	–	–	1	1/0	–	–	N/A
		ALT2-OUT	TA0_O4	1	0	1	0	N/A	N/A	1/0	–	–	0	1/0	–	–	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A

Table 8-38 Register setting for each pin function (3/63)

n	m	Function	Pin name	Feature							Characteristics						
Group	Bit			PMCnm	PMnm	PFCnm	PFCEnm	PIBCnm	PIPCnm	PBDCnm	PUnm	PDnm	PDSCnm	PODCnm	PISnm	PISEnm	PISAnm
0	5	Port mode (input)	P0_5	0	1	N/A	N/A	1	N/A	N/A	–	–	N/A	N/A	–	–	1
		Port mode (output)	P0_5	0	0	N/A	N/A	N/A	N/A	1/0	–	–	0	1/0	–	–	N/A
		ALT1-IN	P_D5[a]	1	N/A	0	0	N/A	1	N/A	–	–	N/A	N/A	–	–	1
		ALT2-IN	TA0_I5	1	1	1	0	N/A	N/A	N/A	–	–	N/A	N/A	–	–	1
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT1-OUT	P_D5[a]	1	N/A	0	0	N/A	1	1/0	–	–	1	1/0	–	–	N/A
		ALT2-OUT	TA0_O5	1	0	1	0	N/A	N/A	1/0	–	–	0	1/0	–	–	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
0	6	Port mode (input)	P0_6	0	1	N/A	N/A	1	N/A	N/A	–	–	N/A	N/A	–	–	1
		Port mode (output)	P0_6	0	0	N/A	N/A	N/A	N/A	1/0	–	–	0	1/0	–	–	N/A
		ALT1-IN	P_D6[a]	1	N/A	0	0	N/A	1	N/A	–	–	N/A	N/A	–	–	1
		ALT2-IN	TA0_I6	1	1	1	0	N/A	N/A	N/A	–	–	N/A	N/A	–	–	1
		ALT3-IN	TE0_BI	1	1	0	1	N/A	N/A	N/A	–	–	N/A	N/A	–	–	1
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT1-OUT	P_D6[a]	1	N/A	0	0	N/A	1	1/0	–	–	1	1/0	–	–	N/A
		ALT2-OUT	TA0_O6	1	0	1	0	N/A	N/A	1/0	–	–	0	1/0	–	–	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
0	7	Port mode (input)	P0_7	0	1	N/A	N/A	1	N/A	N/A	–	–	N/A	N/A	–	–	1
		Port mode (output)	P0_7	0	0	N/A	N/A	N/A	N/A	1/0	–	–	0	1/0	–	–	N/A
		ALT1-IN	P_D7[a]	1	N/A	0	0	N/A	1	N/A	–	–	N/A	N/A	–	–	1
		ALT2-IN	TA0_I7	1	1	1	0	N/A	N/A	N/A	–	–	N/A	N/A	–	–	1
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT1-OUT	P_D7[a]	1	N/A	0	0	N/A	1	1/0	–	–	1	1/0	–	–	N/A
		ALT2-OUT	TA0_O7	1	0	1	0	N/A	N/A	1/0	–	–	0	1/0	–	–	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A

Table 8-38 Register setting for each pin function (4/63)

n	m	Function	Pin name	Feature							Characteristics						
Group	Bit			PMCnm	PMnm	PFCnm	PFCEnm	PIBCnm	PIPCnm	PBDCnm	PUnm	PDnm	PDSCnm	PODCnm	PISnm	PISEnm	PISAnm
0	8	Port mode (input)	P0_8	0	1	N/A	N/A	1	N/A	N/A	–	–	N/A	N/A	–	–	1
		Port mode (output)	P0_8	0	0	N/A	N/A	N/A	N/A	1/0	–	–	0	1/0	–	–	N/A
		ALT1-IN	P_D8[a]	1	N/A	0	0	N/A	1	N/A	–	–	N/A	N/A	–	–	1
		ALT2-IN	TA0_I8	1	1	1	0	N/A	N/A	N/A	–	–	N/A	N/A	–	–	1
		ALT3-IN	TE0_ZI	1	1	0	1	N/A	N/A	N/A	–	–	N/A	N/A	–	–	1
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT1-OUT	P_D8[a]	1	N/A	0	0	N/A	1	1/0	–	–	1	1/0	–	–	N/A
		ALT2-OUT	TA0_O8	1	0	1	0	N/A	N/A	1/0	–	–	0	1/0	–	–	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
0	9	Port mode (input)	P0_9	0	1	N/A	N/A	1	N/A	N/A	–	–	N/A	N/A	–	–	1
		Port mode (output)	P0_9	0	0	N/A	N/A	N/A	N/A	1/0	–	–	0	1/0	–	–	N/A
		ALT1-IN	P_D9[a]	1	N/A	0	0	N/A	1	N/A	–	–	N/A	N/A	–	–	1
		ALT2-IN	TA0_I9	1	1	1	0	N/A	N/A	N/A	–	–	N/A	N/A	–	–	1
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT1-OUT	P_D9[a]	1	N/A	0	0	N/A	1	1/0	–	–	1	1/0	–	–	N/A
		ALT2-OUT	TA0_O9	1	0	1	0	N/A	N/A	1/0	–	–	0	1/0	–	–	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
0	10	Port mode (input)	P0_10	0	1	N/A	N/A	1	N/A	N/A	–	–	N/A	N/A	–	–	1
		Port mode (output)	P0_10	0	0	N/A	N/A	N/A	N/A	1/0	–	–	0	1/0	–	–	N/A
		ALT1-IN	P_D10[a]	1	N/A	0	0	N/A	1	N/A	–	–	N/A	N/A	–	–	1
		ALT2-IN	TA0_I10	1	1	1	0	N/A	N/A	N/A	–	–	N/A	N/A	–	–	1
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT1-OUT	P_D10[a]	1	N/A	0	0	N/A	1	1/0	–	–	1	1/0	–	–	N/A
		ALT2-OUT	TA0_O10	1	N/A	1	0	N/A	1	1/0	–	–	0	1/0	–	–	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A

Table 8-38 Register setting for each pin function (5/63)

n	m	Function	Pin name	Feature							Characteristics						
Group	Bit			PMCnm	PMnm	PFCnm	PFCEnm	PIBCnm	PIPCnm	PBDCnm	PUnm	PDnm	PDSCnm	PODCnm	PISnm	PISEnm	PISAnm
0	11	Port mode (input)	P0_11	0	1	N/A	N/A	1	N/A	N/A	–	–	N/A	N/A	–	–	1
		Port mode (output)	P0_11	0	0	N/A	N/A	N/A	N/A	1/0	–	–	0	1/0	–	–	N/A
		ALT1-IN	P_D11[a]	1	N/A	0	0	N/A	1	N/A	–	–	N/A	N/A	–	–	1
		ALT2-IN	TA0_I11	1	1	1	0	N/A	N/A	N/A	–	–	N/A	N/A	–	–	1
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT1-OUT	P_D11[a]	1	N/A	0	0	N/A	1	1/0	–	–	1	1/0	–	–	N/A
		ALT2-OUT	TA0_O11	1	N/A	1	0	N/A	1	1/0	–	–	0	1/0	–	–	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
0	12	Port mode (input)	P0_12	0	1	N/A	N/A	1	N/A	N/A	–	–	N/A	N/A	–	–	1
		Port mode (output)	P0_12	0	0	N/A	N/A	N/A	N/A	1/0	–	–	0	1/0	–	–	N/A
		ALT1-IN	P_D12[a]	1	N/A	0	0	N/A	1	N/A	–	–	N/A	N/A	–	–	1
		ALT2-IN	TA0_I12	1	1	1	0	N/A	N/A	N/A	–	–	N/A	N/A	–	–	1
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT1-OUT	P_D12[a]	1	N/A	0	0	N/A	1	1/0	–	–	1	1/0	–	–	N/A
		ALT2-OUT	TA0_O12	1	N/A	1	0	N/A	1	1/0	–	–	0	1/0	–	–	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
0	13	Port mode (input)	P0_13	0	1	N/A	N/A	1	N/A	N/A	–	–	N/A	N/A	–	–	1
		Port mode (output)	P0_13	0	0	N/A	N/A	N/A	N/A	1/0	–	–	0	1/0	–	–	N/A
		ALT1-IN	P_D13[a]	1	N/A	0	0	N/A	1	N/A	–	–	N/A	N/A	–	–	1
		ALT2-IN	TA0_I13	1	1	1	0	N/A	N/A	N/A	–	–	N/A	N/A	–	–	1
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT1-OUT	P_D13[a]	1	N/A	0	0	N/A	1	1/0	–	–	1	1/0	–	–	N/A
		ALT2-OUT	TA0_O13	1	N/A	1	0	N/A	1	1/0	–	–	0	1/0	–	–	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A

Table 8-38 Register setting for each pin function (6/63)

n	m	Function	Pin name	Feature							Characteristics						
Group	Bit			PMCnm	PMnm	PFCnm	PFCEnm	PIBCnm	PIPCnm	PBDCnm	PUnm	PDnm	PDSCnm	PODCnm	PISnm	PISEnm	PISAnm
0	14	Port mode (input)	P0_14	0	1	N/A	N/A	1	N/A	N/A	–	–	N/A	N/A	–	–	1
		Port mode (output)	P0_14	0	0	N/A	N/A	N/A	N/A	1/0	–	–	0	1/0	–	–	N/A
		ALT1-IN	P_D14[a]	1	N/A	0	0	N/A	1	N/A	–	–	N/A	N/A	–	–	1
		ALT2-IN	TA0_I14	1	1	1	0	N/A	N/A	N/A	–	–	N/A	N/A	–	–	1
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT1-OUT	P_D14[a]	1	N/A	0	0	N/A	1	1/0	–	–	1	1/0	–	–	N/A
		ALT2-OUT	TA0_O14	1	N/A	1	0	N/A	1	1/0	–	–	0	1/0	–	–	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
0	15	Port mode (input)	P0_15	0	1	N/A	N/A	1	N/A	N/A	–	–	N/A	N/A	–	–	1
		Port mode (output)	P0_15	0	0	N/A	N/A	N/A	N/A	1/0	–	–	0	1/0	–	–	N/A
		ALT1-IN	P_D15[a]	1	N/A	0	0	N/A	1	N/A	–	–	N/A	N/A	–	–	1
		ALT2-IN	TA0_I15	1	1	1	0	N/A	N/A	N/A	–	–	N/A	N/A	–	–	1
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT1-OUT	P_D15[a]	1	N/A	0	0	N/A	1	1/0	–	–	1	1/0	–	–	N/A
		ALT2-OUT	TA0_O15	1	N/A	1	0	N/A	1	1/0	–	–	0	1/0	–	–	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
1	0	Port mode (input)	P1_0	0	1	N/A	N/A	1	N/A	N/A	–	–	N/A	N/A	–	–	1
		Port mode (output)	P1_0	0	0	N/A	N/A	N/A	N/A	1/0	–	–	0	1/0	–	–	N/A
		ALT1-IN	P_D16[a]	1	N/A	0	0	N/A	1	N/A	–	–	N/A	N/A	–	–	1
		ALT2-IN	TA2_I0	1	1	1	0	N/A	N/A	N/A	–	–	N/A	N/A	–	–	1
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT1-OUT	P_D16[a]	1	N/A	0	0	N/A	1	1/0	–	–	1	1/0	–	–	N/A
		ALT2-OUT	TA2_O0	1	0	1	0	N/A	N/A	1/0	–	–	0	1/0	–	–	N/A
		ALT3-OUT	ADCNV0	1	0	0	1	N/A	N/A	1/0	–	–	0	1/0	–	–	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A

Table 8-38 Register setting for each pin function (7/63)

n	m	Function	Pin name	Feature							Characteristics						
Group	Bit			PMCnm	PMnm	PFCnm	PFCEnm	PIBCnm	PIPCnm	PBDCnm	PUnm	PDnm	PDSCnm	PODCnm	PISnm	PISEnm	PISAnm
1	1	Port mode (input)	P1_1	0	1	N/A	N/A	1	N/A	N/A	–	–	N/A	N/A	–	–	1
		Port mode (output)	P1_1	0	0	N/A	N/A	N/A	N/A	1/0	–	–	0	1/0	–	–	N/A
		ALT1-IN	P_D17[a]	1	N/A	0	0	N/A	1	N/A	–	–	N/A	N/A	–	–	1
		ALT2-IN	TA2_I1	1	1	1	0	N/A	N/A	N/A	–	–	N/A	N/A	–	–	1
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT1-OUT	P_D17[a]	1	N/A	0	0	N/A	1	1/0	–	–	1	1/0	–	–	N/A
		ALT2-OUT	TA2_O1	1	0	1	0	N/A	N/A	1/0	–	–	0	1/0	–	–	N/A
		ALT3-OUT	ADCNV1	1	0	0	1	N/A	N/A	1/0	–	–	0	1/0	–	–	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
1	2	Port mode (input)	P1_2	0	1	N/A	N/A	1	N/A	N/A	–	–	N/A	N/A	–	–	1
		Port mode (output)	P1_2	0	0	N/A	N/A	N/A	N/A	1/0	–	–	0	1/0	–	–	N/A
		ALT1-IN	P_D18[a]	1	N/A	0	0	N/A	1	N/A	–	–	N/A	N/A	–	–	1
		ALT2-IN	TA2_I2	1	1	1	0	N/A	N/A	N/A	–	–	N/A	N/A	–	–	1
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT1-OUT	P_D18[a]	1	N/A	0	0	N/A	1	1/0	–	–	1	1/0	–	–	N/A
		ALT2-OUT	TA2_O2	1	0	1	0	N/A	N/A	1/0	–	–	0	1/0	–	–	N/A
		ALT3-OUT	ADCNV2	1	0	0	1	N/A	N/A	1/0	–	–	0	1/0	–	–	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
1	3	Port mode (input)	P1_3	0	1	N/A	N/A	1	N/A	N/A	–	–	N/A	N/A	–	–	1
		Port mode (output)	P1_3	0	0	N/A	N/A	N/A	N/A	1/0	–	–	0	1/0	–	–	N/A
		ALT1-IN	P_D19[a]	1	N/A	0	0	N/A	1	N/A	–	–	N/A	N/A	–	–	1
		ALT2-IN	TA2_I3	1	1	1	0	N/A	N/A	N/A	–	–	N/A	N/A	–	–	1
		ALT3-IN	ESO0	1	1	0	1	N/A	N/A	N/A	–	–	N/A	N/A	–	–	1
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT1-OUT	P_D19[a]	1	N/A	0	0	N/A	1	1/0	–	–	1	1/0	–	–	N/A
		ALT2-OUT	TA2_O3	1	0	1	0	N/A	N/A	1/0	–	–	0	1/0	–	–	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A

Table 8-38 Register setting for each pin function (8/63)

n	m	Function	Pin name	Feature							Characteristics						
Group	Bit			PMCnm	PMnm	PFCnm	PFCEnm	PIBCnm	PIPCnm	PBDCnm	PUnm	PDnm	PDSCnm	PODCnm	PISnm	PISEnm	PISAnm
1	4	Port mode (input)	P1_4	0	1	N/A	N/A	1	N/A	N/A	–	–	N/A	N/A	–	–	1
		Port mode (output)	P1_4	0	0	N/A	N/A	N/A	N/A	1/0	–	–	0	1/0	–	–	N/A
		ALT1-IN	P_D20[a]	1	N/A	0	0	N/A	1	N/A	–	–	N/A	N/A	–	–	1
		ALT2-IN	TA2_I4	1	1	1	0	N/A	N/A	N/A	–	–	N/A	N/A	–	–	1
		ALT3-IN	ESO1	1	1	0	1	N/A	N/A	N/A	–	–	N/A	N/A	–	–	1
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT1-OUT	P_D20[a]	1	N/A	0	0	N/A	1	1/0	–	–	1	1/0	–	–	N/A
		ALT2-OUT	TA2_O4	1	0	1	0	N/A	N/A	1/0	–	–	0	1/0	–	–	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
1	5	Port mode (input)	P1_5	0	1	N/A	N/A	1	N/A	N/A	–	–	N/A	N/A	–	–	1
		Port mode (output)	P1_5	0	0	N/A	N/A	N/A	N/A	1/0	–	–	0	1/0	–	–	N/A
		ALT1-IN	P_D21[a]	1	N/A	0	0	N/A	1	N/A	–	–	N/A	N/A	–	–	1
		ALT2-IN	TA2_I5	1	1	1	0	N/A	N/A	N/A	–	–	N/A	N/A	–	–	1
		ALT3-IN	ESO2	1	1	0	1	N/A	N/A	N/A	–	–	N/A	N/A	–	–	1
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT1-OUT	P_D21[a]	1	N/A	0	0	N/A	1	1/0	–	–	1	1/0	–	–	N/A
		ALT2-OUT	TA2_O5	1	0	1	0	N/A	N/A	1/0	–	–	0	1/0	–	–	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
1	6	Port mode (input)	P1_6	0	1	N/A	N/A	1	N/A	N/A	–	–	N/A	N/A	–	–	1
		Port mode (output)	P1_6	0	0	N/A	N/A	N/A	N/A	1/0	–	–	0	1/0	–	–	N/A
		ALT1-IN	P_D22[a]	1	N/A	0	0	N/A	1	N/A	–	–	N/A	N/A	–	–	1
		ALT2-IN	TA2_I6	1	1	1	0	N/A	N/A	N/A	–	–	N/A	N/A	–	–	1
		ALT3-IN	ESO3	1	1	0	1	N/A	N/A	N/A	–	–	N/A	N/A	–	–	1
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT1-OUT	P_D22[a]	1	N/A	0	0	N/A	1	1/0	–	–	1	1/0	–	–	N/A
		ALT2-OUT	TA2_O6	1	0	1	0	N/A	N/A	1/0	–	–	0	1/0	–	–	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A

Table 8-38 Register setting for each pin function (9/63)

n	m	Function	Pin name	Feature							Characteristics						
Group	Bit			PMCnm	PMnm	PFCnm	PFCEnm	PIBCnm	PIPCnm	PBDCnm	PUnm	PDnm	PDSCnm	PODCnm	PISnm	PISEnm	PISAnm
1	7	Port mode (input)	P1_7	0	1	N/A	N/A	1	N/A	N/A	–	–	N/A	N/A	–	–	1
		Port mode (output)	P1_7	0	0	N/A	N/A	N/A	N/A	1/0	–	–	0	1/0	–	–	N/A
		ALT1-IN	P_D23[a]	1	N/A	0	0	N/A	1	N/A	–	–	N/A	N/A	–	–	1
		ALT2-IN	TA2_I7	1	1	1	0	N/A	N/A	N/A	–	–	N/A	N/A	–	–	1
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT1-OUT	P_D23[a]	1	N/A	0	0	N/A	1	1/0	–	–	1	1/0	–	–	N/A
		ALT2-OUT	TA2_O7	1	0	1	0	N/A	N/A	1/0	–	–	0	1/0	–	–	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
1	8	Port mode (input)	P1_8	0	1	N/A	N/A	1	N/A	N/A	–	–	N/A	N/A	–	–	1
		Port mode (output)	P1_8	0	0	N/A	N/A	N/A	N/A	1/0	–	–	0	1/0	–	–	N/A
		ALT1-IN	P_D24[a]	1	N/A	0	0	N/A	1	N/A	–	–	N/A	N/A	–	–	1
		ALT2-IN	TA2_I8	1	1	1	0	N/A	N/A	N/A	–	–	N/A	N/A	–	–	1
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT1-OUT	P_D24[a]	1	N/A	0	0	N/A	1	1/0	–	–	1	1/0	–	–	N/A
		ALT2-OUT	TA2_O8	1	0	1	0	N/A	N/A	1/0	–	–	0	1/0	–	–	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
1	9	Port mode (input)	P1_9	0	1	N/A	N/A	1	N/A	N/A	–	–	N/A	N/A	–	–	1
		Port mode (output)	P1_9	0	0	N/A	N/A	N/A	N/A	1/0	–	–	0	1/0	–	–	N/A
		ALT1-IN	P_D25[a]	1	N/A	0	0	N/A	1	N/A	–	–	N/A	N/A	–	–	1
		ALT2-IN	TA2_I9	1	1	1	0	N/A	N/A	N/A	–	–	N/A	N/A	–	–	1
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT1-OUT	P_D25[a]	1	N/A	0	0	N/A	1	1/0	–	–	1	1/0	–	–	N/A
		ALT2-OUT	TA2_O9	1	0	1	0	N/A	N/A	1/0	–	–	0	1/0	–	–	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A

Table 8-38 Register setting for each pin function (10/63)

n	m	Function	Pin name	Feature							Characteristics						
Group	Bit			PMCnm	PMnm	PFCnm	PFCEnm	PIBCnm	PIPCnm	PBDCnm	PUnm	PDnm	PDSCnm	PODCnm	PISnm	PISEnm	PISAnm
1	10	Port mode (input)	P1_10	0	1	N/A	N/A	1	N/A	N/A	–	–	N/A	N/A	–	–	1
		Port mode (output)	P1_10	0	0	N/A	N/A	N/A	N/A	1/0	–	–	0	1/0	–	–	N/A
		ALT1-IN	P_D26[a]	1	N/A	0	0	N/A	1	N/A	–	–	N/A	N/A	–	–	1
		ALT2-IN	TA2_I10	1	1	1	0	N/A	N/A	N/A	–	–	N/A	N/A	–	–	1
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT1-OUT	P_D26[a]	1	N/A	0	0	N/A	1	1/0	–	–	1	1/0	–	–	N/A
		ALT2-OUT	TA2_O10	1	0	1	0	N/A	N/A	1/0	–	–	0	1/0	–	–	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
1	11	Port mode (input)	P1_11	0	1	N/A	N/A	1	N/A	N/A	–	–	N/A	N/A	–	–	1
		Port mode (output)	P1_11	0	0	N/A	N/A	N/A	N/A	1/0	–	–	0	1/0	–	–	N/A
		ALT1-IN	P_D27[a]	1	N/A	0	0	N/A	1	N/A	–	–	N/A	N/A	–	–	1
		ALT2-IN	TA2_I11	1	1	1	0	N/A	N/A	N/A	–	–	N/A	N/A	–	–	1
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT1-OUT	P_D27[a]	1	N/A	0	0	N/A	1	1/0	–	–	1	1/0	–	–	N/A
		ALT2-OUT	TA2_O11	1	0	1	0	N/A	N/A	1/0	–	–	0	1/0	–	–	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
1	12	Port mode (input)	P1_12	0	1	N/A	N/A	1	N/A	N/A	–	–	N/A	N/A	–	–	1
		Port mode (output)	P1_12	0	0	N/A	N/A	N/A	N/A	1/0	–	–	0	1/0	–	–	N/A
		ALT1-IN	P_D28[a]	1	N/A	0	0	N/A	1	N/A	–	–	N/A	N/A	–	–	1
		ALT2-IN	TA2_I12	1	1	1	0	N/A	N/A	N/A	–	–	N/A	N/A	–	–	1
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT1-OUT	P_D28[a]	1	N/A	0	0	N/A	1	1/0	–	–	1	1/0	–	–	N/A
		ALT2-OUT	TA2_O12	1	0	1	0	N/A	N/A	1/0	–	–	0	1/0	–	–	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A

Table 8-38 Register setting for each pin function (11/63)

n	m	Function	Pin name	Feature							Characteristics						
Group	Bit			PMCnm	PMnm	PFCnm	PFCEnm	PIBCnm	PIPCnm	PBDCnm	PUnm	PDnm	PDSCnm	PODCnm	PISnm	PISEnm	PISAnm
1	13	Port mode (input)	P1_13	0	1	N/A	N/A	1	N/A	N/A	–	–	N/A	N/A	–	–	1
		Port mode (output)	P1_13	0	0	N/A	N/A	N/A	N/A	1/0	–	–	0	1/0	–	–	N/A
		ALT1-IN	P_D29[a]	1	N/A	0	0	N/A	1	N/A	–	–	N/A	N/A	–	–	1
		ALT2-IN	TA2_I13	1	1	1	0	N/A	N/A	N/A	–	–	N/A	N/A	–	–	1
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT1-OUT	P_D29[a]	1	N/A	0	0	N/A	1	1/0	–	–	1	1/0	–	–	N/A
		ALT2-OUT	TA2_O13	1	N/A	1	0	N/A	1	1/0	–	–	0	1/0	–	–	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
1	14	Port mode (input)	P1_14	0	1	N/A	N/A	1	N/A	N/A	–	–	N/A	N/A	–	–	1
		Port mode (output)	P1_14	0	0	N/A	N/A	N/A	N/A	1/0	–	–	0	1/0	–	–	N/A
		ALT1-IN	P_D30[a]	1	N/A	0	0	N/A	1	N/A	–	–	N/A	N/A	–	–	1
		ALT2-IN	TA2_I14	1	1	1	0	N/A	N/A	N/A	–	–	N/A	N/A	–	–	1
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT1-OUT	P_D30[a]	1	N/A	0	0	N/A	1	1/0	–	–	1	1/0	–	–	N/A
		ALT2-OUT	TA2_O14	1	0	1	0	N/A	N/A	1/0	–	–	0	1/0	–	–	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
1	15	Port mode (input)	P1_15	0	1	N/A	N/A	1	N/A	N/A	–	–	N/A	N/A	–	–	1
		Port mode (output)	P1_15	0	0	N/A	N/A	N/A	N/A	1/0	–	–	0	1/0	–	–	N/A
		ALT1-IN	P_D31[a]	1	N/A	0	0	N/A	1	N/A	–	–	N/A	N/A	–	–	1
		ALT2-IN	TA2_I15	1	1	1	0	N/A	N/A	N/A	–	–	N/A	N/A	–	–	1
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT1-OUT	P_D31[a]	1	N/A	0	0	N/A	1	1/0	–	–	1	1/0	–	–	N/A
		ALT2-OUT	TA2_O15	1	N/A	1	0	N/A	1	1/0	–	–	0	1/0	–	–	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A	N/A	–	–	N/A

Table 8-38 Register setting for each pin function (12/63)

n	m	Function	Pin name	Feature							Characteristics						
Group	Bit			PMCnm	PMnm	PFCnm	PFCEnm	PIBCnm	PIPCnm	PBDCnm	PUnm	PDnm	PDSCnm	PODCnm	PISnm	PISEnm	PISAnm
2	0	Port mode (input)	P2_0	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	0	0	0
		Port mode (output)	P2_0	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT1-IN	NMI	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	1	0	0
		ALT2-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT1-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT2-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
2	1	Port mode (input)	P2_1	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		Port mode (output)	P2_1	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT1-IN	INTP13	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		ALT2-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-IN	TJ_I0	1	1	0	1	N/A	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT1-OUT	$\overline{P\_LLBE}$	1	N/A	0	0	N/A	1	1/0	N/A	N/A	N/A	1/0	N/A	N/A	N/A
		ALT2-OUT	$\overline{P\_LLWR}$	1	N/A	1	0	N/A	1	1/0	N/A	N/A	N/A	1/0	N/A	N/A	N/A
		ALT3-OUT	TJ_O0	1	0	0	1	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
2	2	Port mode (input)	P2_2	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		Port mode (output)	P2_2	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT1-IN	INTP14	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		ALT2-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-IN	TJ_I1	1	1	0	1	N/A	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT1-OUT	$\overline{P\_LUBE}$	1	N/A	0	0	N/A	1	1/0	N/A	N/A	N/A	1/0	N/A	N/A	N/A
		ALT2-OUT	$\overline{P\_LUWR}$	1	N/A	1	0	N/A	1	1/0	N/A	N/A	N/A	1/0	N/A	N/A	N/A
		ALT3-OUT	TJ_O1	1	0	0	1	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

Table 8-38 Register setting for each pin function (13/63)

n	m	Function	Pin name	Feature							Characteristics						
Group	Bit			PMCnm	PMnm	PFCnm	PFCEnm	PIBCnm	PIPCnm	PBDCnm	PUnm	PDnm	PDSCnm	PODCnm	PISnm	PISEnm	PISAnm
2	3	Port mode (input)	P2_3	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		Port mode (output)	P2_3	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT1-IN	INTP15	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		ALT2-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-IN	TJ_I2	1	1	0	1	N/A	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT1-OUT	$\overline{P\_ULBE}$	1	N/A	0	0	N/A	1	1/0	N/A	N/A	N/A	1/0	N/A	N/A	N/A
		ALT2-OUT	$\overline{P\_ULWR}$	1	N/A	1	0	N/A	1	1/0	N/A	N/A	N/A	1/0	N/A	N/A	N/A
		ALT3-OUT	TJ_O2	1	0	0	1	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
2	4	Port mode (input)	P2_4	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		Port mode (output)	P2_4	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT1-IN	INTP16	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		ALT2-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-IN	TJ_I3	1	1	0	1	N/A	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT1-OUT	$\overline{P\_UUBE}$	1	N/A	0	0	N/A	1	1/0	N/A	N/A	N/A	1/0	N/A	N/A	N/A
		ALT2-OUT	$\overline{P\_UUWR}$	1	N/A	1	0	N/A	1	1/0	N/A	N/A	N/A	1/0	N/A	N/A	N/A
		ALT3-OUT	TJ_O3	1	0	0	1	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
2	5	Port mode (input)	P2_5	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		Port mode (output)	P2_5	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT1-IN	INTP17	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		ALT2-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT1-OUT	P_RD	1	N/A	0	0	N/A	1	1/0	N/A	N/A	N/A	1/0	N/A	N/A	N/A
		ALT2-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

Table 8-38 Register setting for each pin function (14/63)

n	m	Function	Pin name	Feature							Characteristics						
Group	Bit			PMCnm	PMnm	PFCnm	PFCEnm	PIBCnm	PIPCnm	PBDCnm	PUnm	PDnm	PDSCnm	PODCnm	PISnm	PISEnm	PISAnm
2	6	Port mode (input)	P2_6	0	1	N/A	N/A	1	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	1
		Port mode (output)	P2_6	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	N/A	N/A	N/A
		ALT1-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT2-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT1-OUT	P_BUSCLK	1	0	0	0	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT2-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
2	7	Port mode (input)	P2_7	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		Port mode (output)	P2_7	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT1-IN	INTP19	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		ALT2-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT1-OUT	$\overline{P\_WR}$	1	N/A	0	0	N/A	1	1/0	N/A	N/A	N/A	1/0	N/A	N/A	N/A
		ALT2-OUT	$\overline{P\_RW}$	1	N/A	1	0	N/A	1	1/0	N/A	N/A	N/A	1/0	N/A	N/A	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
3	0	Port mode (input)	P3_0	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		Port mode (output)	P3_0	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–
		ALT1-IN	TE1_TI0	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT2-IN	TA1_I0	1	1	1	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT3-IN	INTP18	1	1	0	1	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT1-OUT	P_A0	1	N/A	0	0	N/A	1	1/0	N/A	N/A	N/A	1/0	–	–	–
		ALT2-OUT	TA1_O0	1	0	1	0	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–

Table 8-38 Register setting for each pin function (15/63)

n	m	Function	Pin name	Feature							Characteristics						
Group	Bit			PMCnm	PMnm	PFCnm	PFCEnm	PIBCnm	PIPCnm	PBDCnm	PUnm	PDnm	PDSCnm	PODCnm	PISnm	PISEnm	PISAnm
3	1	Port mode (input)	P3_1	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		Port mode (output)	P3_1	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–
		ALT1-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT2-IN	TA1_I1	1	1	1	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT1-OUT	P_A1	1	N/A	0	0	N/A	1	1/0	N/A	N/A	N/A	1/0	–	–	–
		ALT2-OUT	TA1_O1	1	0	1	0	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
3	2	Port mode (input)	P3_2	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		Port mode (output)	P3_2	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–
		ALT1-IN	TE1_TI1	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT2-IN	TA1_I2	1	1	1	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT1-OUT	P_A2	1	N/A	0	0	N/A	1	1/0	N/A	N/A	N/A	1/0	–	–	–
		ALT2-OUT	TA1_O2	1	0	1	0	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
3	3	Port mode (input)	P3_3	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		Port mode (output)	P3_3	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–
		ALT1-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT2-IN	TA1_I3	1	1	1	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT1-OUT	P_A3	1	N/A	0	0	N/A	1	1/0	N/A	N/A	N/A	1/0	–	–	–
		ALT2-OUT	TA1_O3	1	0	1	0	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–

Table 8-38 Register setting for each pin function (16/63)

n	m	Function	Pin name	Feature							Characteristics						
Group	Bit			PMCnm	PMnm	PFCnm	PFCEnm	PIBCnm	PIPCnm	PBDCnm	PUnm	PDnm	PDSCnm	PODCnm	PISnm	PISEnm	PISAnm
3	4	Port mode (input)	P3_4	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		Port mode (output)	P3_4	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–
		ALT1-IN	TE1_AI	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT2-IN	TA1_I4	1	1	1	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT1-OUT	P_A4	1	N/A	0	0	N/A	1	1/0	N/A	N/A	N/A	1/0	–	–	–
		ALT2-OUT	TA1_O4	1	0	1	0	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
3	5	Port mode (input)	P3_5	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		Port mode (output)	P3_5	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–
		ALT1-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT2-IN	TA1_I5	1	1	1	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT1-OUT	P_A5	1	N/A	0	0	N/A	1	1/0	N/A	N/A	N/A	1/0	–	–	–
		ALT2-OUT	TA1_O5	1	0	1	0	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
3	6	Port mode (input)	P3_6	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		Port mode (output)	P3_6	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–
		ALT1-IN	TE1_BI	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT2-IN	TA1_I6	1	1	1	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT1-OUT	P_A6	1	N/A	0	0	N/A	1	1/0	N/A	N/A	N/A	1/0	–	–	–
		ALT2-OUT	TA1_O6	1	0	1	0	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT4-OUT	CSI3F_CS0	1	0	1	1	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–

Table 8-38 Register setting for each pin function (17/63)

n	m	Function	Pin name	Feature							Characteristics						
Group	Bit			PMCnm	PMnm	PFCnm	PFCEnm	PIBCnm	PIPCnm	PBDCnm	PUnm	PDnm	PDSCnm	PODCnm	PISnm	PISEnm	PISAnm
3	7	Port mode (input)	P3_7	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		Port mode (output)	P3_7	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–
		ALT1-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT2-IN	TA1_I7	1	1	1	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT1-OUT	P_A7	1	N/A	0	0	N/A	1	1/0	N/A	N/A	N/A	1/0	–	–	–
		ALT2-OUT	TA1_O7	1	0	1	0	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT4-OUT	CSI3F_CS1	1	0	1	1	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–
3	8	Port mode (input)	P3_8	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		Port mode (output)	P3_8	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–
		ALT1-IN	TE1_ZI	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT2-IN	TA1_I8	1	1	1	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT1-OUT	P_A8	1	N/A	0	0	N/A	1	1/0	N/A	N/A	N/A	1/0	–	–	–
		ALT2-OUT	TA1_O8	1	0	1	0	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT4-OUT	CSI3F_CS2	1	0	1	1	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–
3	9	Port mode (input)	P3_9	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		Port mode (output)	P3_9	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–
		ALT1-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT2-IN	TA1_I9	1	1	1	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT1-OUT	P_A9	1	N/A	0	0	N/A	1	1/0	N/A	N/A	N/A	1/0	–	–	–
		ALT2-OUT	TA1_O9	1	0	1	0	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT4-OUT	CSI3F_CS3	1	0	1	1	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–

Table 8-38 Register setting for each pin function (18/63)

n	m	Function	Pin name	Feature							Characteristics						
Group	Bit			PMCnm	PMnm	PFCnm	PFCEnm	PIBCnm	PIPCnm	PBDCnm	PUnm	PDnm	PDSCnm	PODCnm	PISnm	PISEnm	PISAnm
3	10	Port mode (input)	P3_10	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		Port mode (output)	P3_10	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–
		ALT1-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT2-IN	TA1_I10	1	1	1	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT1-OUT	P_A10	1	N/A	0	0	N/A	1	1/0	N/A	N/A	N/A	1/0	–	–	–
		ALT2-OUT	TA1_O10	1	N/A	1	0	N/A	1	1/0	N/A	N/A	1	1/0	–	–	–
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT4-OUT	CSI3F_CS4	1	0	1	1	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–
3	11	Port mode (input)	P3_11	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		Port mode (output)	P3_11	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–
		ALT1-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT2-IN	TA1_I11	1	1	1	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT1-OUT	P_A11	1	N/A	0	0	N/A	1	1/0	N/A	N/A	N/A	1/0	–	–	–
		ALT2-OUT	TA1_O11	1	N/A	1	0	N/A	1	1/0	N/A	N/A	1	1/0	–	–	–
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT4-OUT	CSI3F_CS5	1	0	1	1	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–
3	12	Port mode (input)	P3_12	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		Port mode (output)	P3_12	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–
		ALT1-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT2-IN	TA1_I12	1	1	1	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT1-OUT	P_A12	1	N/A	0	0	N/A	1	1/0	N/A	N/A	N/A	1/0	–	–	–
		ALT2-OUT	TA1_O12	1	N/A	1	0	N/A	1	1/0	N/A	N/A	1	1/0	–	–	–
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT4-OUT	CSI3F_CS6	1	0	1	1	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–

Table 8-38 Register setting for each pin function (19/63)

n	m	Function	Pin name	Feature							Characteristics						
Group	Bit			PMCnm	PMnm	PFCnm	PFCEnm	PIBCnm	PIPCnm	PBDCnm	PUnm	PDnm	PDSCnm	PODCnm	PISnm	PISEnm	PISAnm
3	13	Port mode (input)	P3_13	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		Port mode (output)	P3_13	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–
		ALT1-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT2-IN	TA1_I13	1	1	1	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT1-OUT	P_A13	1	N/A	0	0	N/A	1	1/0	N/A	N/A	N/A	1/0	–	–	–
		ALT2-OUT	TA1_O13	1	N/A	1	0	N/A	1	1/0	N/A	N/A	1	1/0	–	–	–
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT4-OUT	CSI3F_CS7	1	0	1	1	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–
3	14	Port mode (input)	P3_14	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		Port mode (output)	P3_14	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–
		ALT1-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT2-IN	TA1_I14	1	1	1	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT4-IN	CSI3_RYI	1	N/A	1	1	N/A	1	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT1-OUT	P_A14	1	N/A	0	0	N/A	1	1/0	N/A	N/A	N/A	1/0	–	–	–
		ALT2-OUT	TA1_O14	1	N/A	1	0	N/A	1	1/0	N/A	N/A	1	1/0	–	–	–
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT4-OUT	CSI3_RYO	1	N/A	1	1	N/A	1	1/0	N/A	N/A	1	1/0	–	–	–
3	15	Port mode (input)	P3_15	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		Port mode (output)	P3_15	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–
		ALT1-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT2-IN	TA1_I15	1	1	1	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT4-IN	CSI3_SSI	1	1	1	1	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT1-OUT	P_A15	1	N/A	0	0	N/A	1	1/0	N/A	N/A	N/A	1/0	–	–	–
		ALT2-OUT	TA1_O15	1	N/A	1	0	N/A	1	1/0	N/A	N/A	1	1/0	–	–	–
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–

Table 8-38 Register setting for each pin function (20/63)

n	m	Function	Pin name	Feature							Characteristics						
Group	Bit			PMCnm	PMnm	PFCnm	PFCEnm	PIBCnm	PIPCnm	PBDCnm	PUnm	PDnm	PDSCnm	PODCnm	PISnm	PISEnm	PISAnm
4	0	Port mode (input)	P4_0	0	1	N/A	N/A	1/0	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		Port mode (output)	P4_0	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT1-IN	INTP5	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		ALT2-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT4-IN	CSI3F_RYI	1	N/A	1	1	N/A	1	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		ALT1-OUT	P_A16	1	N/A	0	0	N/A	1	1/0	N/A	N/A	N/A	1/0	N/A	N/A	N/A
		ALT2-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT4-OUT	CSI3F_RYO	1	N/A	1	1	N/A	1	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
4	1	Port mode (input)	P4_1	0	1	N/A	N/A	1/0	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		Port mode (output)	P4_1	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT1-IN	INTP6	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		ALT2-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT4-IN	CSI3F_SSI	1	1	1	1	1/0	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		ALT1-OUT	P_A17	1	N/A	0	0	N/A	1	1/0	N/A	N/A	N/A	1/0	N/A	N/A	N/A
		ALT2-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
4	2	Port mode (input)	P4_2	0	1	N/A	N/A	1/0	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		Port mode (output)	P4_2	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT1-IN	INTP7	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		ALT2-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT1-OUT	P_A18	1	N/A	0	0	N/A	1	1/0	N/A	N/A	N/A	1/0	N/A	N/A	N/A
		ALT2-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT4-OUT	SO3	1	N/A	1	1	N/A	1	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A

Table 8-38 Register setting for each pin function (21/63)

n	m	Function	Pin name	Feature							Characteristics						
Group	Bit			PMCnm	PMnm	PFCnm	PFCEnm	PIBCnm	PIPCnm	PBDCnm	PUnm	PDnm	PDSCnm	PODCnm	PISnm	PISEnm	PISAnm
4	3	Port mode (input)	P4_3	0	1	N/A	N/A	1/0	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		Port mode (output)	P4_3	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT1-IN	INTP8	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		ALT2-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT1-OUT	P_A19	1	N/A	0	0	N/A	1	1/0	N/A	N/A	N/A	1/0	N/A	N/A	N/A
		ALT2-OUT	$\overline{\text{DMAAK4}}$	1	0	1	0	N/A	N/A	1/0	N/A	N/A	N/A	1/0	N/A	N/A	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT4-OUT	SO3F	1	N/A	1	1	N/A	1	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
4	4	Port mode (input)	P4_4	0	1	N/A	N/A	1/0	N/A	N/A	1/0	1/0	N/A	N/A	0	0	0
		Port mode (output)	P4_4	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	N/A	N/A	N/A
		ALT1-IN	INTP9	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	0	0	0
		ALT2-IN	RXD3	1	1	1	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	1	0	0
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT4-IN	SI3	1	1	1	1	N/A	N/A	N/A	1/0	1/0	N/A	N/A	1	0	0
		ALT1-OUT	P_A20	1	N/A	0	0	N/A	1	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT2-OUT	$\overline{\text{DMATC4}}$	1	0	1	0	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
4	5	Port mode (input)	P4_5	0	1	N/A	N/A	1/0	N/A	N/A	1/0	1/0	N/A	N/A	0	0	0
		Port mode (output)	P4_5	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	N/A	N/A	N/A
		ALT1-IN	INTP10	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	0	0	0
		ALT2-IN	ADTRG10	1	1	1	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	0	0	0
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT4-IN	SCK3[b]	1	N/A	1	1	N/A	1	N/A	1/0	1/0	N/A	N/A	1	0	0
		ALT1-OUT	P_A21	1	N/A	0	0	N/A	1	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT2-OUT	TXD3	1	0	1	0	N/A	N/A	1/0	N/A	N/A	0	1/0	N/A	N/A	N/A
		ALT3-OUT	$\overline{\text{DMATC3}}$	1	0	0	1	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT4-OUT	SCK3[b]	1	N/A	1	1	N/A	1	1/0	N/A	N/A	0	1/0	N/A	N/A	N/A

Table 8-38 Register setting for each pin function (22/63)

n	m	Function	Pin name	Feature							Characteristics						
Group	Bit			PMCnm	PMnm	PFCnm	PFCEnm	PIBCnm	PIPCnm	PBDCnm	PUnm	PDnm	PDSCnm	PODCnm	PISnm	PISEnm	PISAnm
4	6	Port mode (input)	P4_6	0	1	N/A	N/A	1/0	N/A	N/A	1/0	1/0	N/A	N/A	0	0	0
		Port mode (output)	P4_6	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	N/A	N/A	N/A
		ALT1-IN	INTP11	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	0	0	0
		ALT2-IN	RXD3F	1	1	1	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	1	0	0
		ALT3-IN	SDA3[a]	1	0	0	1	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0	1	0
		ALT4-IN	SI3F	1	1	1	1	N/A	N/A	N/A	1/0	1/0	N/A	N/A	1	0	0
		ALT1-OUT	P_A22	1	N/A	0	0	N/A	1	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT2-OUT	DMAAK3	1	0	1	0	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT3-OUT	SDA3[a]	1	0	0	1	N/A	N/A	1	N/A	N/A	0	1	N/A	N/A	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
4	7	Port mode (input)	P4_7	0	1	N/A	N/A	1/0	N/A	N/A	1/0	1/0	N/A	N/A	0	0	0
		Port mode (output)	P4_7	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	N/A	N/A	N/A
		ALT1-IN	INTP12	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	0	0	0
		ALT2-IN	ADTRG20	1	1	1	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	0	0	0
		ALT3-IN	SCL3[a]	1	0	0	1	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0	1	0
		ALT4-IN	SCK3F[b]	1	N/A	1	1	N/A	1	N/A	1/0	1/0	N/A	N/A	1	0	0
		ALT1-OUT	P_A23	1	N/A	0	0	N/A	1	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT2-OUT	TXD3F	1	0	1	0	N/A	N/A	1/0	N/A	N/A	0	1/0	N/A	N/A	N/A
		ALT3-OUT	SCL3[a]	1	0	0	1	N/A	N/A	1	N/A	N/A	0	1	N/A	N/A	N/A
		ALT4-OUT	SCK3F[b]	1	N/A	1	1	N/A	1	1/0	N/A	N/A	0	1/0	N/A	N/A	N/A
4	8	Port mode (input)	P4_8	0	1	N/A	N/A	1/0	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		Port mode (output)	P4_8	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT1-IN	INTP20	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		ALT2-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT1-OUT	P_CS1	1	N/A	0	0	N/A	1	1/0	N/A	N/A	N/A	1/0	N/A	N/A	N/A
		ALT2-OUT	P_BCYST	1	0	1	0	N/A	N/A	1/0	N/A	N/A	N/A	1/0	N/A	N/A	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

Table 8-38 Register setting for each pin function (23/63)

n	m	Function	Pin name	Feature							Characteristics						
Group	Bit			PMCnm	PMnm	PFCnm	PFCEnm	PIBCnm	PIPCnm	PBDCnm	PUnm	PDnm	PDSCnm	PODCnm	PISnm	PISEnm	PISAnm
4	9	Port mode (input)	P4_9	0	1	N/A	N/A	1/0	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		Port mode (output)	P4_9	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT1-IN	INTP21	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		ALT2-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT1-OUT	$\overline{P\_CS2}$	1	N/A	0	0	N/A	1	1/0	N/A	N/A	N/A	1/0	N/A	N/A	N/A
		ALT2-OUT	$\overline{DMAAK5}$	1	0	1	0	N/A	N/A	1/0	N/A	N/A	N/A	1/0	N/A	N/A	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
4	10	Port mode (input)	P4_10	0	1	N/A	N/A	1/0	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		Port mode (output)	P4_10	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT1-IN	INTP22	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		ALT2-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT1-OUT	$\overline{P\_CS3}$	1	N/A	0	0	N/A	1	1/0	N/A	N/A	N/A	1/0	N/A	N/A	N/A
		ALT2-OUT	$\overline{DMATC5}$	1	0	1	0	N/A	N/A	1/0	N/A	N/A	N/A	1/0	N/A	N/A	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
4	11	Port mode (input)	P4_11	0	1	N/A	N/A	1/0	N/A	N/A	1/0	1/0	N/A	N/A	0	0	0
		Port mode (output)	P4_11	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	N/A	N/A	N/A
		ALT1-IN	$\overline{P\_WAIT}$	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	0	0	1
		ALT2-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-IN	SCL0[a]	1	0	0	1	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0	1	0
		ALT4-IN	SCK0F[b]	1	N/A	1	1	N/A	1	N/A	1/0	1/0	N/A	N/A	1	0	0
		ALT1-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT2-OUT	TXD0F	1	0	1	0	N/A	N/A	1/0	N/A	N/A	0	1/0	N/A	N/A	N/A
		ALT3-OUT	SCL0[a]	1	0	0	1	N/A	N/A	1	N/A	N/A	0	1	N/A	N/A	N/A
		ALT4-OUT	SCK0F[b]	1	N/A	1	1	N/A	1	1/0	N/A	N/A	0	1/0	N/A	N/A	N/A

Table 8-38 Register setting for each pin function (24/63)

n	m	Function	Pin name	Feature							Characteristics						
Group	Bit			PMCnm	PMnm	PFCnm	PFCEnm	PIBCnm	PIPCnm	PBDCnm	PUnm	PDnm	PDSCnm	PODCnm	PISnm	PISEnm	PISAnm
4	12	Port mode (input)	P4_12	0	1	N/A	N/A	1/0	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		Port mode (output)	P4_12	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT1-IN	INTP23	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		ALT2-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT1-OUT	$\overline{P\_HLDAK}$	1	0	0	0	N/A	N/A	1/0	N/A	N/A	N/A	1/0	N/A	N/A	N/A
		ALT2-OUT	$\overline{DMAAK2}$	1	0	1	0	N/A	N/A	1/0	N/A	N/A	N/A	1/0	N/A	N/A	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT4-OUT	SO0F	1	N/A	1	1	N/A	1	1/0	N/A	N/A	0	1/0	N/A	N/A	N/A
4	13	Port mode (input)	P4_13	0	1	N/A	N/A	1/0	N/A	N/A	1/0	1/0	N/A	N/A	0	0	0
		Port mode (output)	P4_13	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	N/A	N/A	N/A
		ALT1-IN	$\overline{P\_HLDRQ}$	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	0	0	1
		ALT2-IN	RXD0F	1	1	1	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	1	0	0
		ALT3-IN	SDA0[a]	1	0	0	1	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0	1	0
		ALT4-IN	SI0F	1	1	1	1	N/A	N/A	N/A	1/0	1/0	N/A	N/A	1	0	0
		ALT1-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT2-OUT	$\overline{DMATC2}$	1	0	1	0	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT3-OUT	SDA0[a]	1	0	0	1	N/A	N/A	1	N/A	N/A	0	1	N/A	N/A	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
5	0	Port mode (input)	P5_0	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		Port mode (output)	P5_0	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	–
		ALT1-IN	INTP20	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT2-IN	ETH_CRS	1	1	1	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT3-IN	TA3_I0	1	1	0	1	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT4-IN	CSI0F_RYI	1	1	1	1	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT1-OUT	P_LLDQM	1	N/A	0	0	N/A	1	1/0	N/A	N/A	1	1/0	–	–	–
		ALT2-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT3-OUT	TA3_O0	1	0	0	1	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	–
		ALT4-OUT	CSI0F_RYO	1	N/A	1	1	N/A	1	1/0	N/A	N/A	0	1/0	–	–	–

Table 8-38 Register setting for each pin function (25/63)

n	m	Function	Pin name	Feature							Characteristics						
Group	Bit			PMCnm	PMnm	PFCnm	PFCEnm	PIBCnm	PIPCnm	PBDCnm	PUnm	PDnm	PDSCnm	PODCnm	PISnm	PISEnm	PISAnm
5	1	Port mode (input)	P5_1	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		Port mode (output)	P5_1	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	–
		ALT1-IN	INTP21	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT2-IN	ETH_COL	1	1	1	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT3-IN	TA3_I1	1	1	0	1	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT4-IN	CSI0F_SSI	1	1	1	1	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT1-OUT	P_LUDQM	1	N/A	0	0	N/A	1	1/0	N/A	N/A	1	1/0	–	–	–
		ALT2-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT3-OUT	TA3_O1	1	0	0	1	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	–
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
5	2	Port mode (input)	P5_2	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		Port mode (output)	P5_2	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	–
		ALT1-IN	INTP22	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT2-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT3-IN	TA3_I2	1	1	0	1	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT1-OUT	P_ULDQM	1	N/A	0	0	N/A	1	1/0	N/A	N/A	1	1/0	–	–	–
		ALT2-OUT	ETH_TXD3	1	0	1	0	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	–
		ALT3-OUT	TA3_O2	1	0	0	1	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	–
		ALT4-OUT	CSI0F_CS0	1	0	1	1	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	–
5	3	Port mode (input)	P5_3	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		Port mode (output)	P5_3	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	–
		ALT1-IN	INTP23	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT2-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT3-IN	TA3_I3	1	1	0	1	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT1-OUT	P_UUDQM	1	N/A	0	0	N/A	1	1/0	N/A	N/A	1	1/0	–	–	–
		ALT2-OUT	ETH_TXD2	1	0	1	0	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	–
		ALT3-OUT	TA3_O3	1	0	0	1	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	–
		ALT4-OUT	CSI0F_CS1	1	0	1	1	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	–

Table 8-38 Register setting for each pin function (26/63)

n	m	Function	Pin name	Feature							Characteristics						
Group	Bit			PMCnm	PMnm	PFCnm	PFCEnm	PIBCnm	PIPCnm	PBDCnm	PUnm	PDnm	PDSCnm	PODCnm	PISnm	PISEnm	PISAnm
5	4	Port mode (input)	P5_4	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		Port mode (output)	P5_4	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	–
		ALT1-IN	INTP24	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT2-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT3-IN	TA3_I4	1	1	0	1	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT1-OUT	$\overline{\text{P\_REFRQ}}$	1	0	0	0	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–
		ALT2-OUT	ETH_TXD1	1	0	1	0	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	–
		ALT3-OUT	TA3_O4	1	0	0	1	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	–
		ALT4-OUT	CSI0F_CS2	1	0	1	1	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	–
5	5	Port mode (input)	P5_5	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		Port mode (output)	P5_5	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	–
		ALT1-IN	INTP25	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT2-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT3-IN	TA3_I5	1	1	0	1	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT1-OUT	$\overline{\text{P\_SDRAS}}$	1	N/A	0	0	N/A	1	1/0	N/A	N/A	1	1/0	–	–	–
		ALT2-OUT	ETH_TXD0	1	0	1	0	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	–
		ALT3-OUT	TA3_O5	1	0	0	1	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	–
		ALT4-OUT	CSI0F_CS3	1	0	1	1	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	–
5	6	Port mode (input)	P5_6	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		Port mode (output)	P5_6	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	–
		ALT1-IN	INTP26	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT2-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT3-IN	TA3_I6	1	1	0	1	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT1-OUT	P_SDCAS	1	N/A	0	0	N/A	1	1/0	N/A	N/A	1	1/0	–	–	–
		ALT2-OUT	ETH_TXEN	1	0	1	0	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	–
		ALT3-OUT	TA3_O6	1	0	0	1	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	–
		ALT4-OUT	CSI0F_CS4	1	0	1	1	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	–

Table 8-38 Register setting for each pin function (27/63)

n	m	Function	Pin name	Feature							Characteristics						
Group	Bit			PMCnm	PMnm	PFCnm	PFCEnm	PIBCnm	PIPCnm	PBDCnm	PUnm	PDnm	PDSCnm	PODCnm	PISnm	PISEnm	PISAnm
5	7	Port mode (input)	P5_7	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		Port mode (output)	P5_7	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	–
		ALT1-IN	INTP27	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT2-IN	ETH_TXCLK	1	1	1	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT3-IN	TA3_I7	1	1	0	1	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT1-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT2-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT3-OUT	TA3_O7	1	0	0	1	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	–
		ALT4-OUT	CSI0F_CS5	1	0	1	1	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	–
5	8	Port mode (input)	P5_8	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		Port mode (output)	P5_8	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	–
		ALT1-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT2-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT3-IN	TA3_I8	1	1	0	1	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT1-OUT	P_SDCKE	1	N/A	0	0	N/A	1	1/0	N/A	N/A	1	1/0	–	–	–
		ALT2-OUT	ETH_TXER	1	0	1	0	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	–
		ALT3-OUT	TA3_O8	1	0	0	1	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	–
		ALT4-OUT	CSI0F_CS6	1	0	1	1	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	–
5	9	Port mode (input)	P5_9	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		Port mode (output)	P5_9	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	–
		ALT1-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT2-IN	ETH_RXER	1	1	1	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT3-IN	TA3_I9	1	1	0	1	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT1-OUT	P_CS4	1	N/A	0	0	N/A	1	1/0	N/A	N/A	1	1/0	–	–	–
		ALT2-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT3-OUT	TA3_O9	1	0	0	1	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	–
		ALT4-OUT	CSI0F_CS7	1	0	1	1	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	–

Table 8-38 Register setting for each pin function (28/63)

n	m	Function	Pin name	Feature							Characteristics						
Group	Bit			PMCnm	PMnm	PFCnm	PFCEnm	PIBCnm	PIPCnm	PBDCnm	PUnm	PDnm	PDSCnm	PODCnm	PISnm	PISEnm	PISAnm
5	10	Port mode (input)	P5_10	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		Port mode (output)	P5_10	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	–
		ALT1-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT2-IN	ETH_RXCLK	1	1	1	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT3-IN	TA3_I10	1	1	0	1	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT1-OUT	$\overline{P\_BUSRQ}$	1	0	0	0	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–
		ALT2-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT3-OUT	TA3_O10	1	0	0	1	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	–
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
5	11	Port mode (input)	P5_11	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		Port mode (output)	P5_11	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	–
		ALT1-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT2-IN	ETH_RXDV	1	1	1	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT3-IN	TA3_I11	1	1	0	1	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT1-OUT	$\overline{P\_SDWE}$	1	N/A	0	0	N/A	1	1/0	N/A	N/A	1	1/0	–	–	–
		ALT2-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT3-OUT	TA3_O11	1	0	0	1	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	–
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
5	12	Port mode (input)	P5_12	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		Port mode (output)	P5_12	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	–
		ALT1-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT2-IN	ETH_RXD1	1	1	1	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT3-IN	TA3_I12	1	1	0	1	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT1-OUT	P_BCYST	1	N/A	0	0	N/A	1	1/0	N/A	N/A	1	1/0	–	–	–
		ALT2-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT3-OUT	TA3_O12	1	0	0	1	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	–
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–

Table 8-38 Register setting for each pin function (29/63)

n	m	Function	Pin name	Feature							Characteristics						
Group	Bit			PMCnm	PMnm	PFCnm	PFCEnm	PIBCnm	PIPCnm	PBDCnm	PUnm	PDnm	PDSCnm	PODCnm	PISnm	PISEnm	PISAnm
5	13	Port mode (input)	P5_13	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		Port mode (output)	P5_13	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	–
		ALT1-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT2-IN	ETH_RXD1	1	1	1	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT3-IN	TA3_I13	1	1	0	1	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT1-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT2-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT3-OUT	TA3_O13	1	0	0	1	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	–
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
5	14	Port mode (input)	P5_14	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		Port mode (output)	P5_14	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	–
		ALT1-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT2-IN	ETH_RXD2	1	1	1	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT3-IN	TA3_I14	1	1	0	1	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT4-IN	TJ_I0	1	1	1	1	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT1-OUT	$\overline{\text{DMAAK0}}$	1	0	0	0	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–
		ALT2-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT3-OUT	TA3_O14	1	0	0	1	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	–
		ALT4-OUT	TJ_O0	1	0	1	1	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	–
5	15	Port mode (input)	P5_15	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		Port mode (output)	P5_15	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	–
		ALT1-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT2-IN	ETH_RXD3	1	1	1	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT3-IN	TA3_I15	1	1	0	1	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT4-IN	TJ_I1	1	1	1	1	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT1-OUT	DMAAK1	1	0	0	0	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–
		ALT2-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT3-OUT	TA3_O15	1	0	0	1	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	–
		ALT4-OUT	TJ_O1	1	0	1	1	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	–

Table 8-38 Register setting for each pin function (30/63)

n	m	Function	Pin name	Feature							Characteristics						
Group	Bit			PMCnm	PMnm	PFCnm	PFCEnm	PIBCnm	PIPCnm	PBDCnm	PUnm	PDnm	PDSCnm	PODCnm	PISnm	PISEnm	PISAnm
6	0	Port mode (input)	P6_0	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		Port mode (output)	P6_0	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	–
		ALT1-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT2-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT4-IN	TJ_I2	1	1	1	1	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT1-OUT	DMATC0	1	0	0	0	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–
		ALT2-OUT	ETH_MDC	1	0	1	0	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	–
		ALT3-OUT	P_A24	1	N/A	0	1	N/A	1	1/0	N/A	N/A	1	1/0	–	–	–
		ALT4-OUT	TJ_O2	1	0	1	1	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	–
6	1	Port mode (input)	P6_1	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		Port mode (output)	P6_1	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	–
		ALT1-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT2-IN	ETH_MDIO[a]	1	N/A	1	0	N/A	1	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT4-IN	TJ_I3	1	1	1	1	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT1-OUT	DMATC1	1	0	0	0	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–
		ALT2-OUT	ETH_MDIO[a]	1	N/A	1	0	N/A	1	1/0	N/A	N/A	0	1/0	–	–	–
		ALT3-OUT	P_A25	1	N/A	0	1	N/A	1	1/0	N/A	N/A	1	1/0	–	–	–
		ALT4-OUT	TJ_O3	1	0	1	1	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	–
7	0	Port mode (input)	P7_0	0	1	N/A	N/A	1	N/A	N/A	N/A	N/A	N/A	N/A	–	–	1
		Port mode (output)	P7_0	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	N/A
		ALT1-IN	S_D0[a]	1	N/A	0	0	N/A	1	N/A	N/A	N/A	N/A	N/A	–	–	1
		ALT2-IN	TA2_I0	1	1	1	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	1
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT1-OUT	S_D0[a]	1	N/A	0	0	N/A	1	1/0	N/A	N/A	1	1/0	–	–	N/A
		ALT2-OUT	TA2_O0	1	0	1	0	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A

Table 8-38 Register setting for each pin function (31/63)

n	m	Function	Pin name	Feature							Characteristics						
Group	Bit			PMCnm	PMnm	PFCnm	PFCEnm	PIBCnm	PIPCnm	PBDCnm	PUnm	PDnm	PDSCnm	PODCnm	PISnm	PISEnm	PISAnm
7	1	Port mode (input)	P7_1	0	1	N/A	N/A	1	N/A	N/A	N/A	N/A	N/A	N/A	–	–	1
		Port mode (output)	P7_1	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	N/A
		ALT1-IN	S_D1[a]	1	N/A	0	0	N/A	1	N/A	N/A	N/A	N/A	N/A	–	–	1
		ALT2-IN	TA2_I1	1	1	1	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	1
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT1-OUT	S_D1[a]	1	N/A	0	0	N/A	1	1/0	N/A	N/A	1	1/0	–	–	N/A
		ALT2-OUT	TA2_O1	1	0	1	0	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
7	2	Port mode (input)	P7_2	0	1	N/A	N/A	1	N/A	N/A	N/A	N/A	N/A	N/A	–	–	1
		Port mode (output)	P7_2	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	N/A
		ALT1-IN	S_D2[a]	1	N/A	0	0	N/A	1	N/A	N/A	N/A	N/A	N/A	–	–	1
		ALT2-IN	TA2_I2	1	1	1	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	1
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT1-OUT	S_D2[a]	1	N/A	0	0	N/A	1	1/0	N/A	N/A	1	1/0	–	–	N/A
		ALT2-OUT	TA2_O2	1	0	1	0	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
7	3	Port mode (input)	P7_3	0	1	N/A	N/A	1	N/A	N/A	N/A	N/A	N/A	N/A	–	–	1
		Port mode (output)	P7_3	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	N/A
		ALT1-IN	S_D3[a]	1	N/A	0	0	N/A	1	N/A	N/A	N/A	N/A	N/A	–	–	1
		ALT2-IN	TA2_I3	1	1	1	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	1
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT1-OUT	S_D3[a]	1	N/A	0	0	N/A	1	1/0	N/A	N/A	1	1/0	–	–	N/A
		ALT2-OUT	TA2_O3	1	0	1	0	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A

Table 8-38 Register setting for each pin function (32/63)

n	m	Function	Pin name	Feature							Characteristics						
Group	Bit			PMCnm	PMnm	PFCnm	PFCEnm	PIBCnm	PIPCnm	PBDCnm	PUnm	PDnm	PDSCnm	PODCnm	PISnm	PISEnm	PISAnm
7	4	Port mode (input)	P7_4	0	1	N/A	N/A	1	N/A	N/A	N/A	N/A	N/A	N/A	–	–	1
		Port mode (output)	P7_4	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	N/A
		ALT1-IN	S_D4[a]	1	N/A	0	0	N/A	1	N/A	N/A	N/A	N/A	N/A	–	–	1
		ALT2-IN	TA2_I4	1	1	1	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	1
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT1-OUT	S_D4[a]	1	N/A	0	0	N/A	1	1/0	N/A	N/A	1	1/0	–	–	N/A
		ALT2-OUT	TA2_O4	1	0	1	0	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
7	5	Port mode (input)	P7_5	0	1	N/A	N/A	1	N/A	N/A	N/A	N/A	N/A	N/A	–	–	1
		Port mode (output)	P7_5	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	N/A
		ALT1-IN	S_D5[a]	1	N/A	0	0	N/A	1	N/A	N/A	N/A	N/A	N/A	–	–	1
		ALT2-IN	TA2_I5	1	1	1	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	1
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT1-OUT	S_D5[a]	1	N/A	0	0	N/A	1	1/0	N/A	N/A	1	1/0	–	–	N/A
		ALT2-OUT	TA2_O5	1	0	1	0	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
7	6	Port mode (input)	P7_6	0	1	N/A	N/A	1	N/A	N/A	N/A	N/A	N/A	N/A	–	–	1
		Port mode (output)	P7_6	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	N/A
		ALT1-IN	S_D6[a]	1	N/A	0	0	N/A	1	N/A	N/A	N/A	N/A	N/A	–	–	1
		ALT2-IN	TA2_I6	1	1	1	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	1
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT1-OUT	S_D6[a]	1	N/A	0	0	N/A	1	1/0	N/A	N/A	1	1/0	–	–	N/A
		ALT2-OUT	TA2_O6	1	0	1	0	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A

Table 8-38 Register setting for each pin function (33/63)

n	m	Function	Pin name	Feature							Characteristics						
Group	Bit			PMCnm	PMnm	PFCnm	PFCEnm	PIBCnm	PIPCnm	PBDCnm	PUnm	PDnm	PDSCnm	PODCnm	PISnm	PISEnm	PISAnm
7	7	Port mode (input)	P7_7	0	1	N/A	N/A	1	N/A	N/A	N/A	N/A	N/A	N/A	–	–	1
		Port mode (output)	P7_7	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	N/A
		ALT1-IN	S_D7[a]	1	N/A	0	0	N/A	1	N/A	N/A	N/A	N/A	N/A	–	–	1
		ALT2-IN	TA2_I7	1	1	1	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	1
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT1-OUT	S_D7[a]	1	N/A	0	0	N/A	1	1/0	N/A	N/A	1	1/0	–	–	N/A
		ALT2-OUT	TA2_O7	1	0	1	0	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
7	8	Port mode (input)	P7_8	0	1	N/A	N/A	1	N/A	N/A	N/A	N/A	N/A	N/A	–	–	1
		Port mode (output)	P7_8	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	N/A
		ALT1-IN	S_D8[a]	1	N/A	0	0	N/A	1	N/A	N/A	N/A	N/A	N/A	–	–	1
		ALT2-IN	TA2_I8	1	1	1	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	1
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT1-OUT	S_D8[a]	1	N/A	0	0	N/A	1	1/0	N/A	N/A	1	1/0	–	–	N/A
		ALT2-OUT	TA2_O8	1	0	1	0	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
7	9	Port mode (input)	P7_9	0	1	N/A	N/A	1	N/A	N/A	N/A	N/A	N/A	N/A	–	–	1
		Port mode (output)	P7_9	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	N/A
		ALT1-IN	S_D9[a]	1	N/A	0	0	N/A	1	N/A	N/A	N/A	N/A	N/A	–	–	1
		ALT2-IN	TA2_I9	1	1	1	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	1
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT1-OUT	S_D9[a]	1	N/A	0	0	N/A	1	1/0	N/A	N/A	1	1/0	–	–	N/A
		ALT2-OUT	TA2_O9	1	0	1	0	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A

Table 8-38 Register setting for each pin function (34/63)

n	m	Function	Pin name	Feature							Characteristics						
Group	Bit			PMCnm	PMnm	PFCnm	PFCEnm	PIBCnm	PIPCnm	PBDCnm	PUnm	PDnm	PDSCnm	PODCnm	PISnm	PISEnm	PISAnm
7	10	Port mode (input)	P7_10	0	1	N/A	N/A	1	N/A	N/A	N/A	N/A	N/A	N/A	–	–	1
		Port mode (output)	P7_10	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	N/A
		ALT1-IN	S_D10[a]	1	N/A	0	0	N/A	1	N/A	N/A	N/A	N/A	N/A	–	–	1
		ALT2-IN	TA2_I10	1	1	1	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	1
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT1-OUT	S_D10[a]	1	N/A	0	0	N/A	1	1/0	N/A	N/A	1	1/0	–	–	N/A
		ALT2-OUT	TA2_O10	1	0	1	0	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
7	11	Port mode (input)	P7_11	0	1	N/A	N/A	1	N/A	N/A	N/A	N/A	N/A	N/A	–	–	1
		Port mode (output)	P7_11	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	N/A
		ALT1-IN	S_D11[a]	1	N/A	0	0	N/A	1	N/A	N/A	N/A	N/A	N/A	–	–	1
		ALT2-IN	TA2_I11	1	1	1	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	1
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT1-OUT	S_D11[a]	1	N/A	0	0	N/A	1	1/0	N/A	N/A	1	1/0	–	–	N/A
		ALT2-OUT	TA2_O11	1	0	1	0	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
7	12	Port mode (input)	P7_12	0	1	N/A	N/A	1	N/A	N/A	N/A	N/A	N/A	N/A	–	–	1
		Port mode (output)	P7_12	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	N/A
		ALT1-IN	S_D12[a]	1	N/A	0	0	N/A	1	N/A	N/A	N/A	N/A	N/A	–	–	1
		ALT2-IN	TA2_I12	1	1	1	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	1
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT1-OUT	S_D12[a]	1	N/A	0	0	N/A	1	1/0	N/A	N/A	1	1/0	–	–	N/A
		ALT2-OUT	TA2_O12	1	0	1	0	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A

Table 8-38 Register setting for each pin function (35/63)

n	m	Function	Pin name	Feature							Characteristics						
Group	Bit			PMCnm	PMnm	PFCnm	PFCEnm	PIBCnm	PIPCnm	PBDCnm	PUnm	PDnm	PDSCnm	PODCnm	PISnm	PISEnm	PISAnm
7	13	Port mode (input)	P7_13	0	1	N/A	N/A	1	N/A	N/A	N/A	N/A	N/A	N/A	–	–	1
		Port mode (output)	P7_13	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	N/A
		ALT1-IN	S_D13[a]	1	N/A	0	0	N/A	1	N/A	N/A	N/A	N/A	N/A	–	–	1
		ALT2-IN	TA2_I13	1	1	1	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	1
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT1-OUT	S_D13[a]	1	N/A	0	0	N/A	1	1/0	N/A	N/A	1	1/0	–	–	N/A
		ALT2-OUT	TA2_O13	1	0	1	0	N/A	1	1/0	N/A	N/A	0	1/0	–	–	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
7	14	Port mode (input)	P7_14	0	1	N/A	N/A	1	N/A	N/A	N/A	N/A	N/A	N/A	–	–	1
		Port mode (output)	P7_14	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	N/A
		ALT1-IN	S_D14[a]	1	N/A	0	0	N/A	1	N/A	N/A	N/A	N/A	N/A	–	–	1
		ALT2-IN	TA2_I14	1	1	1	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	1
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT1-OUT	S_D14[a]	1	N/A	0	0	N/A	1	1/0	N/A	N/A	1	1/0	–	–	N/A
		ALT2-OUT	TA2_O14	1	0	1	0	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
7	15	Port mode (input)	P7_15	0	1	N/A	N/A	1	N/A	N/A	N/A	N/A	N/A	N/A	–	–	1
		Port mode (output)	P7_15	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	N/A
		ALT1-IN	S_D15[a]	1	N/A	0	0	N/A	1	N/A	N/A	N/A	N/A	N/A	–	–	1
		ALT2-IN	TA2_I15	1	1	1	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	1
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT1-OUT	S_D15[a]	1	N/A	0	0	N/A	1	1/0	N/A	N/A	1	1/0	–	–	N/A
		ALT2-OUT	TA2_O15	1	0	1	0	N/A	1	1/0	N/A	N/A	0	1/0	–	–	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A

Table 8-38 Register setting for each pin function (36/63)

n	m	Function	Pin name	Feature							Characteristics						
Group	Bit			PMCnm	PMnm	PFCnm	PFCEnm	PIBCnm	PIPCnm	PBDCnm	PUnm	PDnm	PDSCnm	PODCnm	PISnm	PISEnm	PISAnm
8	0	Port mode (input)	P8_0	0	1	N/A	N/A	1	N/A	N/A	N/A	N/A	N/A	N/A	–	–	1
		Port mode (output)	P8_0	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	N/A
		ALT1-IN	S_D16[a]	1	N/A	0	0	N/A	1	N/A	N/A	N/A	N/A	N/A	–	–	1
		ALT2-IN	TA0_I0	1	1	1	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	1
		ALT3-IN	TE0_TI0	1	1	0	1	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	1
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT1-OUT	S_D16[a]	1	N/A	0	0	N/A	1	1/0	N/A	N/A	1	1/0	–	–	N/A
		ALT2-OUT	TA0_O0	1	0	1	0	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
8	1	Port mode (input)	P8_1	0	1	N/A	N/A	1	N/A	N/A	N/A	N/A	N/A	N/A	–	–	1
		Port mode (output)	P8_1	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	N/A
		ALT1-IN	S_D17[a]	1	N/A	0	0	N/A	1	N/A	N/A	N/A	N/A	N/A	–	–	1
		ALT2-IN	TA0_I1	1	1	1	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	1
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT1-OUT	S_D17[a]	1	N/A	0	0	N/A	1	1/0	N/A	N/A	1	1/0	–	–	N/A
		ALT2-OUT	TA0_O1	1	0	1	0	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
8	2	Port mode (input)	P8_2	0	1	N/A	N/A	1	N/A	N/A	N/A	N/A	N/A	N/A	–	–	1
		Port mode (output)	P8_2	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	N/A
		ALT1-IN	S_D18[a]	1	N/A	0	0	N/A	1	N/A	N/A	N/A	N/A	N/A	–	–	1
		ALT2-IN	TA0_I2	1	1	1	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	1
		ALT3-IN	TE0_TI1	1	1	0	1	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	1
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT1-OUT	S_D18[a]	1	N/A	0	0	N/A	1	1/0	N/A	N/A	1	1/0	–	–	N/A
		ALT2-OUT	TA0_O2	1	0	1	0	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A

Table 8-38 Register setting for each pin function (37/63)

n	m	Function	Pin name	Feature							Characteristics						
Group	Bit			PMCnm	PMnm	PFCnm	PFCEnm	PIBCnm	PIPCnm	PBDCnm	PUnm	PDnm	PDSCnm	PODCnm	PISnm	PISEnm	PISAnm
8	3	Port mode (input)	P8_3	0	1	N/A	N/A	1	N/A	N/A	N/A	N/A	N/A	N/A	–	–	1
		Port mode (output)	P8_3	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	N/A
		ALT1-IN	S_D19[a]	1	N/A	0	0	N/A	1	N/A	N/A	N/A	N/A	N/A	–	–	1
		ALT2-IN	TA0_I3	1	1	1	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	1
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT1-OUT	S_D19[a]	1	N/A	0	0	N/A	1	1/0	N/A	N/A	1	1/0	–	–	N/A
		ALT2-OUT	TA0_O3	1	0	1	0	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
8	4	Port mode (input)	P8_4	0	1	N/A	N/A	1	N/A	N/A	N/A	N/A	N/A	N/A	–	–	1
		Port mode (output)	P8_4	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	N/A
		ALT1-IN	S_D20[a]	1	N/A	0	0	N/A	1	N/A	N/A	N/A	N/A	N/A	–	–	1
		ALT2-IN	TA0_I4	1	1	1	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	1
		ALT3-IN	TE0_TA	1	1	0	1	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	1
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT1-OUT	S_D20[a]	1	N/A	0	0	N/A	1	1/0	N/A	N/A	1	1/0	–	–	N/A
		ALT2-OUT	TA0_O4	1	0	1	0	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
8	5	Port mode (input)	P8_5	0	1	N/A	N/A	1	N/A	N/A	N/A	N/A	N/A	N/A	–	–	1
		Port mode (output)	P8_5	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	N/A
		ALT1-IN	S_D21[a]	1	N/A	0	0	N/A	1	N/A	N/A	N/A	N/A	N/A	–	–	1
		ALT2-IN	TA0_I5	1	1	1	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	1
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT1-OUT	S_D21[a]	1	N/A	0	0	N/A	1	1/0	N/A	N/A	1	1/0	–	–	N/A
		ALT2-OUT	TA0_O5	1	0	1	0	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A

Table 8-38 Register setting for each pin function (38/63)

n	m	Function	Pin name	Feature							Characteristics						
Group	Bit			PMCnm	PMnm	PFCnm	PFCEnm	PIBCnm	PIPCnm	PBDCnm	PUnm	PDnm	PDSCnm	PODCnm	PISnm	PISEnm	PISAnm
8	6	Port mode (input)	P8_6	0	1	N/A	N/A	1	N/A	N/A	N/A	N/A	N/A	N/A	–	–	1
		Port mode (output)	P8_6	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	N/A
		ALT1-IN	S_D22[a]	1	N/A	0	0	N/A	1	N/A	N/A	N/A	N/A	N/A	–	–	1
		ALT2-IN	TA0_I6	1	1	1	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	1
		ALT3-IN	TE0_TB	1	1	0	1	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	1
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT1-OUT	S_D22[a]	1	N/A	0	0	N/A	1	1/0	N/A	N/A	1	1/0	–	–	N/A
		ALT2-OUT	TA0_O6	1	0	1	0	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
8	7	Port mode (input)	P8_7	0	1	N/A	N/A	1	N/A	N/A	N/A	N/A	N/A	N/A	–	–	1
		Port mode (output)	P8_7	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	N/A
		ALT1-IN	S_D23[a]	1	N/A	0	0	N/A	1	N/A	N/A	N/A	N/A	N/A	–	–	1
		ALT2-IN	TA0_I7	1	1	1	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	1
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT1-OUT	S_D23[a]	1	N/A	0	0	N/A	1	1/0	N/A	N/A	1	1/0	–	–	N/A
		ALT2-OUT	TA0_O7	1	0	1	0	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
8	8	Port mode (input)	P8_8	0	1	N/A	N/A	1	N/A	N/A	N/A	N/A	N/A	N/A	–	–	1
		Port mode (output)	P8_8	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	N/A
		ALT1-IN	S_D24[a]	1	N/A	0	0	N/A	1	N/A	N/A	N/A	N/A	N/A	–	–	1
		ALT2-IN	TA0_I8	1	1	1	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	1
		ALT3-IN	TE0_TZ	1	1	0	1	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	1
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT1-OUT	S_D24[a]	1	N/A	0	0	N/A	1	1/0	N/A	N/A	1	1/0	–	–	N/A
		ALT2-OUT	TA0_O8	1	0	1	0	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A

Table 8-38 Register setting for each pin function (39/63)

n	m	Function	Pin name	Feature							Characteristics						
Group	Bit			PMCnm	PMnm	PFCnm	PFCEnm	PIBCnm	PIPCnm	PBDCnm	PUnm	PDnm	PDSCnm	PODCnm	PISnm	PISEnm	PISAnm
8	9	Port mode (input)	P8_9	0	1	N/A	N/A	1	N/A	N/A	N/A	N/A	N/A	N/A	–	–	1
		Port mode (output)	P8_9	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	N/A
		ALT1-IN	S_D25[a]	1	N/A	0	0	N/A	1	N/A	N/A	N/A	N/A	N/A	–	–	1
		ALT2-IN	TA0_I9	1	1	1	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	1
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT1-OUT	S_D25[a]	1	N/A	0	0	N/A	1	1/0	N/A	N/A	1	1/0	–	–	N/A
		ALT2-OUT	TA0_O9	1	0	1	0	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
8	10	Port mode (input)	P8_10	0	1	N/A	N/A	1	N/A	N/A	N/A	N/A	N/A	N/A	–	–	1
		Port mode (output)	P8_10	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	N/A
		ALT1-IN	S_D26[a]	1	N/A	0	0	N/A	1	N/A	N/A	N/A	N/A	N/A	–	–	1
		ALT2-IN	TA0_I10	1	1	1	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	1
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT1-OUT	S_D26[a]	1	N/A	0	0	N/A	1	1/0	N/A	N/A	1	1/0	–	–	N/A
		ALT2-OUT	TA0_O10	1	N/A	1	0	N/A	1	1/0	N/A	N/A	0	1/0	–	–	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
8	11	Port mode (input)	P8_11	0	1	N/A	N/A	1	N/A	N/A	N/A	N/A	N/A	N/A	–	–	1
		Port mode (output)	P8_11	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	N/A
		ALT1-IN	S_D27[a]	1	N/A	0	0	N/A	1	N/A	N/A	N/A	N/A	N/A	–	–	1
		ALT2-IN	TA0_I11	1	1	1	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	1
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT1-OUT	S_D27[a]	1	N/A	0	0	N/A	1	1/0	N/A	N/A	1	1/0	–	–	N/A
		ALT2-OUT	TA0_O11	1	N/A	1	0	N/A	1	1/0	N/A	N/A	0	1/0	–	–	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A

Table 8-38 Register setting for each pin function (40/63)

n	m	Function	Pin name	Feature							Characteristics						
Group	Bit			PMCnm	PMnm	PFCnm	PFCEnm	PIBCnm	PIPCnm	PBDCnm	PUnm	PDnm	PDSCnm	PODCnm	PISnm	PISEnm	PISAnm
8	12	Port mode (input)	P8_12	0	1	N/A	N/A	1	N/A	N/A	N/A	N/A	N/A	N/A	–	–	1
		Port mode (output)	P8_12	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	N/A
		ALT1-IN	S_D28[a]	1	N/A	0	0	N/A	1	N/A	N/A	N/A	N/A	N/A	–	–	1
		ALT2-IN	TA0_I12	1	1	1	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	1
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT1-OUT	S_D28[a]	1	N/A	0	0	N/A	1	1/0	N/A	N/A	1	1/0	–	–	N/A
		ALT2-OUT	TA0_O12	1	N/A	1	0	N/A	1	1/0	N/A	N/A	0	1/0	–	–	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
8	13	Port mode (input)	P8_13	0	1	N/A	N/A	1	N/A	N/A	N/A	N/A	N/A	N/A	–	–	1
		Port mode (output)	P8_13	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	N/A
		ALT1-IN	S_D29[a]	1	N/A	0	0	N/A	1	N/A	N/A	N/A	N/A	N/A	–	–	1
		ALT2-IN	TA0_I13	1	1	1	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	1
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT1-OUT	S_D29[a]	1	N/A	0	0	N/A	1	1/0	N/A	N/A	1	1/0	–	–	N/A
		ALT2-OUT	TA0_O13	1	N/A	1	0	N/A	1	1/0	N/A	N/A	0	1/0	–	–	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
8	14	Port mode (input)	P8_14	0	1	N/A	N/A	1	N/A	N/A	N/A	N/A	N/A	N/A	–	–	1
		Port mode (output)	P8_14	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	N/A
		ALT1-IN	S_D30[a]	1	N/A	0	0	N/A	1	N/A	N/A	N/A	N/A	N/A	–	–	1
		ALT2-IN	TA0_I14	1	1	1	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	1
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT1-OUT	S_D30[a]	1	N/A	0	0	N/A	1	1/0	N/A	N/A	1	1/0	–	–	N/A
		ALT2-OUT	TA0_O14	1	N/A	1	0	N/A	1	1/0	N/A	N/A	0	1/0	–	–	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A

Table 8-38 Register setting for each pin function (41/63)

n	m	Function	Pin name	Feature							Characteristics						
Group	Bit			PMCnm	PMnm	PFCnm	PFCEnm	PIBCnm	PIPCnm	PBDCnm	PUnm	PDnm	PDSCnm	PODCnm	PISnm	PISEnm	PISAnm
8	15	Port mode (input)	P8_15	0	1	N/A	N/A	1	N/A	N/A	N/A	N/A	N/A	N/A	–	–	1
		Port mode (output)	P8_15	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	–	–	N/A
		ALT1-IN	S_D31[a]	1	N/A	0	0	N/A	1	N/A	N/A	N/A	N/A	N/A	–	–	1
		ALT2-IN	TA0_I15	1	1	1	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	1
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT1-OUT	S_D31[a]	1	N/A	0	0	N/A	1	1/0	N/A	N/A	1	1/0	–	–	N/A
		ALT2-OUT	TA0_O15	1	N/A	1	0	N/A	1	1/0	N/A	N/A	0	1/0	–	–	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	N/A
9	0	Port mode (input)	P9_0	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		Port mode (output)	P9_0	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT1-IN	INTP10	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		ALT2-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-IN	TA3_I0	1	1	0	1	N/A	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT1-OUT	S_SDCKE	1	N/A	0	0	N/A	1	1/0	N/A	N/A	N/A	1/0	N/A	N/A	N/A
		ALT2-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-OUT	TA3_O0	1	0	0	1	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT4-OUT	CSI2F_CS0	1	0	1	1	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
9	1	Port mode (input)	P9_1	0	1	N/A	N/A	1	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	1
		Port mode (output)	P9_1	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	N/A	N/A	N/A
		ALT1-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT2-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT1-OUT	S_BUSCLK	1	0	0	0	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT2-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT4-OUT	CSI2F_CS1	1	0	1	1	N/A	N/A	1/0	N/A	N/A	0	1/0	N/A	N/A	N/A

Table 8-38 Register setting for each pin function (42/63)

n	m	Function	Pin name	Feature							Characteristics						
Group	Bit			PMCnm	PMnm	PFCnm	PFCEnm	PIBCnm	PIPCnm	PBDCnm	PUnm	PDnm	PDSCnm	PODCnm	PISnm	PISEnm	PISAnm
9	2	Port mode (input)	P9_2	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		Port mode (output)	P9_2	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT1-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT2-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-IN	TA3_I1	1	1	0	1	N/A	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT1-OUT	$\overline{S\_SDCAS}$	1	N/A	0	0	N/A	1	1/0	N/A	N/A	N/A	1/0	N/A	N/A	N/A
		ALT2-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-OUT	TA3_O1	1	0	0	1	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT4-OUT	CSI2F_CS2	1	0	1	1	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
9	3	Port mode (input)	P9_3	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		Port mode (output)	P9_3	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT1-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT2-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-IN	TA3_I2	1	1	0	1	N/A	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT1-OUT	$\overline{S\_SDRAS}$	1	N/A	0	0	N/A	1	1/0	N/A	N/A	N/A	1/0	N/A	N/A	N/A
		ALT2-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-OUT	TA3_O2	1	0	0	1	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT4-OUT	CSI2F_CS3	1	0	1	1	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
9	4	Port mode (input)	P9_4	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		Port mode (output)	P9_4	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT1-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT2-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-IN	TA3_I3	1	1	0	1	N/A	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT1-OUT	S_LLDQM	1	N/A	0	0	N/A	1	1/0	N/A	N/A	N/A	1/0	N/A	N/A	N/A
		ALT2-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-OUT	TA3_O3	1	0	0	1	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT4-OUT	CSI2F_CS4	1	0	1	1	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A

Table 8-38 Register setting for each pin function (43/63)

n	m	Function	Pin name	Feature							Characteristics						
Group	Bit			PMCnm	PMnm	PFCnm	PFCEnm	PIBCnm	PIPCnm	PBDCnm	PUnm	PDnm	PDSCnm	PODCnm	PISnm	PISEnm	PISAnm
9	5	Port mode (input)	P9_5	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		Port mode (output)	P9_5	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT1-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT2-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-IN	TA3_I4	1	1	0	1	N/A	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT1-OUT	S_LUDQM	1	N/A	0	0	N/A	1	1/0	N/A	N/A	N/A	1/0	N/A	N/A	N/A
		ALT2-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-OUT	TA3_O4	1	0	0	1	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT4-OUT	CSI2F_CS5	1	0	1	1	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
9	6	Port mode (input)	P9_6	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		Port mode (output)	P9_6	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT1-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT2-IN	ESO0	1	1	1	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT1-OUT	S_ULDQM	1	N/A	0	0	N/A	1	1/0	N/A	N/A	N/A	1/0	N/A	N/A	N/A
		ALT2-OUT	$\overline{\text{S\_DMATC2}}$	1	0	1	0	N/A	N/A	1/0	N/A	N/A	N/A	1/0	N/A	N/A	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT4-OUT	CSI2F_CS6	1	0	1	1	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
9	7	Port mode (input)	P9_7	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		Port mode (output)	P9_7	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT1-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT2-IN	ESO1	1	1	1	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT1-OUT	S_UUDQM	1	N/A	0	0	N/A	1	1/0	N/A	N/A	N/A	1/0	N/A	N/A	N/A
		ALT2-OUT	$\overline{\text{S\_DMATC3}}$	1	0	1	0	N/A	N/A	1/0	N/A	N/A	N/A	1/0	N/A	N/A	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT4-OUT	CSI2F_CS7	1	0	1	1	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A

Table 8-38 Register setting for each pin function (44/63)

n	m	Function	Pin name	Feature							Characteristics						
Group	Bit			PMCnm	PMnm	PFCnm	PFCEnm	PIBCnm	PIPCnm	PBDCnm	PUnm	PDnm	PDSCnm	PODCnm	PISnm	PISEnm	PISAnm
9	8	Port mode (input)	P9_8	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		Port mode (output)	P9_8	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT1-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT2-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT4-IN	CSI4_RYI	1	N/A	1	1	N/A	1	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		ALT1-OUT	$\overline{S\_LLWR}$	1	N/A	0	0	N/A	1	1/0	N/A	N/A	N/A	1/0	N/A	N/A	N/A
		ALT2-OUT	S_DMAAK2	1	0	1	0	N/A	N/A	1/0	N/A	N/A	N/A	1/0	N/A	N/A	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT4-OUT	CSI4_RYO	1	N/A	1	1	N/A	1	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
9	9	Port mode (input)	P9_9	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		Port mode (output)	P9_9	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT1-IN	S_DMARQ2	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	1
		ALT2-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT1-OUT	$\overline{S\_LUWR}$	1	N/A	0	0	N/A	1	1/0	N/A	N/A	N/A	1/0	N/A	N/A	N/A
		ALT2-OUT	S_DMATC0	1	0	1	0	N/A	N/A	1/0	N/A	N/A	N/A	1/0	N/A	N/A	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT4-OUT	SO4	1	N/A	1	1	N/A	1	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
9	10	Port mode (input)	P9_10	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	0	0	0
		Port mode (output)	P9_10	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	N/A	N/A	N/A
		ALT1-IN	ESO2	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	0	0	0
		ALT2-IN	RXD4	1	1	1	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	1	0	0
		ALT3-IN	SDA4[a]	1	0	0	1	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0	1	0
		ALT4-IN	SI4	1	1	1	1	N/A	N/A	N/A	1/0	1/0	N/A	N/A	1	0	0
		ALT1-OUT	$\overline{S\_ULWR}$	1	N/A	0	0	N/A	1	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT2-OUT	S_DMAAK3	1	0	1	0	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT3-OUT	SDA4[a]	1	0	0	1	N/A	N/A	1	N/A	N/A	0	1	N/A	N/A	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

Table 8-38 Register setting for each pin function (45/63)

n	m	Function	Pin name	Feature							Characteristics						
Group	Bit			PMCnm	PMnm	PFCnm	PFCEnm	PIBCnm	PIPCnm	PBDCnm	PUnm	PDnm	PDSCnm	PODCnm	PISnm	PISEnm	PISAnm
9	11	Port mode (input)	P9_11	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	0	0	0
		Port mode (output)	P9_11	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	N/A	N/A	N/A
		ALT1-IN	$\overline{S\_DMARQ3}$	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	0	0	1
		ALT2-IN	ESO3	1	1	1	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	0	0	0
		ALT3-IN	SCL4[a]	1	0	0	1	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0	1	0
		ALT4-IN	SCK4[b]	1	N/A	1	1	N/A	1	N/A	1/0	1/0	N/A	N/A	1	0	0
		ALT1-OUT	$\overline{S\_UUWR}$	1	N/A	0	0	N/A	1	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT2-OUT	TXD4	1	0	1	0	N/A	N/A	1/0	N/A	N/A	0	1/0	N/A	N/A	N/A
		ALT3-OUT	SCL4[a]	1	0	0	1	N/A	N/A	1	N/A	N/A	0	1	N/A	N/A	N/A
		ALT4-OUT	SCK4[b]	1	N/A	1	1	N/A	1	1/0	N/A	N/A	0	1/0	N/A	N/A	N/A
9	12	Port mode (input)	P9_12	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		Port mode (output)	P9_12	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT1-IN	INTP11	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		ALT2-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-IN	TA3_I5	1	1	0	1	N/A	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT1-OUT	$\overline{S\_RD}$	1	N/A	0	0	N/A	1	1/0	N/A	N/A	N/A	1/0	N/A	N/A	N/A
		ALT2-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-OUT	TA3_O5	1	0	0	1	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
9	13	Port mode (input)	P9_13	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		Port mode (output)	P9_13	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT1-IN	INTP12	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		ALT2-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-IN	TA3_I6	1	1	0	1	N/A	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		ALT4-IN	CSI1_SSI	1	1	1	1	N/A	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		ALT1-OUT	S_WR	1	N/A	0	0	N/A	1	1/0	N/A	N/A	N/A	1/0	N/A	N/A	N/A
		ALT2-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-OUT	TA3_O6	1	0	0	1	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

Table 8-38 Register setting for each pin function (46/63)

n	m	Function	Pin name	Feature							Characteristics						
Group	Bit			PMCnm	PMnm	PFCnm	PFCEnm	PIBCnm	PIPCnm	PBDCnm	PUnm	PDnm	PDSCnm	PODCnm	PISnm	PISEnm	PISAnm
9	14	Port mode (input)	P9_14	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		Port mode (output)	P9_14	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT1-IN	INTP13	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		ALT2-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-IN	TA3_I7	1	1	0	1	N/A	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		ALT4-IN	CSI1_RYI	1	N/A	1	1	N/A	1	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		ALT1-OUT	$\overline{S\_LLBE}$	1	N/A	0	0	N/A	1	1/0	N/A	N/A	N/A	1/0	N/A	N/A	N/A
		ALT2-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-OUT	TA3_O7	1	0	0	1	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT4-OUT	CSI1_RYO	1	N/A	1	1	N/A	1	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
9	15	Port mode (input)	P9_15	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		Port mode (output)	P9_15	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT1-IN	INTP14	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		ALT2-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT1-OUT	$\overline{S\_LUBE}$	1	N/A	0	0	N/A	1	1/0	N/A	N/A	N/A	1/0	N/A	N/A	N/A
		ALT2-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT4-OUT	SO1	1	N/A	1	1	N/A	1	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
10	0	Port mode (input)	P10_0	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	0	0	0
		Port mode (output)	P10_0	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	N/A	N/A	N/A
		ALT1-IN	INTP15	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	0	0	0
		ALT2-IN	RXD1	1	1	1	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	1	0	0
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT4-IN	SI1	1	1	1	1	1	N/A	N/A	1/0	1/0	N/A	N/A	1	0	0
		ALT1-OUT	$\overline{S\_ULBE}$	1	N/A	0	0	N/A	1	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT2-OUT	S_DMATC0	1	0	1	0	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

Table 8-38 Register setting for each pin function (47/63)

n	m	Function	Pin name	Feature							Characteristics						
Group	Bit			PMCnm	PMnm	PFCnm	PFCEnm	PIBCnm	PIPCnm	PBDCnm	PUnm	PDnm	PDSCnm	PODCnm	PISnm	PISEnm	PISAnm
10	1	Port mode (input)	P10_1	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	0	0	0
		Port mode (output)	P10_1	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	N/A	N/A	N/A
		ALT1-IN	INTP16	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	0	0	0
		ALT2-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT4-IN	SCK1[b]	1	N/A	1	1	N/A	1	N/A	1/0	1/0	N/A	N/A	1	0	0
		ALT1-OUT	S_UUBE	1	N/A	0	0	N/A	1	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT2-OUT	TXD1	1	0	1	0	N/A	N/A	1/0	N/A	N/A	0	1/0	N/A	N/A	N/A
		ALT3-OUT	S_DMATC1	1	0	0	1	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT4-OUT	SCK1[b]	1	N/A	1	1	N/A	1	1/0	N/A	N/A	0	1/0	N/A	N/A	N/A
10	2	Port mode (input)	P10_2	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		Port mode (output)	P10_2	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT1-IN	INTP17	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		ALT2-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-IN	TA3_I8	1	1	0	1	N/A	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		ALT4-IN	CSI2F_SSI	1	1	1	1	N/A	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		ALT1-OUT	S_WE	1	N/A	0	0	N/A	1	1/0	N/A	N/A	N/A	1/0	N/A	N/A	N/A
		ALT2-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-OUT	TA3_O8	1	0	0	1	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
10	3	Port mode (input)	P10_3	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		Port mode (output)	P10_3	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT1-IN	INTP18	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		ALT2-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-IN	TA3_I9	1	1	0	1	N/A	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		ALT4-IN	CSI4_SSI	1	1	1	1	N/A	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		ALT1-OUT	S_BCYST	1	N/A	0	0	N/A	1	1/0	N/A	N/A	N/A	1/0	N/A	N/A	N/A
		ALT2-OUT	S_DMATC1	1	0	1	0	N/A	N/A	1/0	N/A	N/A	N/A	1/0	N/A	N/A	N/A
		ALT3-OUT	TA3_O9	1	0	0	1	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

Table 8-38 Register setting for each pin function (48/63)

n	m	Function	Pin name	Feature							Characteristics						
Group	Bit			PMCnm	PMnm	PFCnm	PFCEnm	PIBCnm	PIPCnm	PBDCnm	PUnm	PDnm	PDSCnm	PODCnm	PISnm	PISEnm	PISAnm
10	4	Port mode (input)	P10_4	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		Port mode (output)	P10_4	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT1-IN	INTP19	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		ALT2-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-IN	TA3_I10	1	1	0	1	N/A	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		ALT4-IN	CSI2_SSI	1	1	1	1	N/A	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		ALT1-OUT	$\overline{S\_CS0}$	1	N/A	0	0	N/A	1	1/0	N/A	N/A	N/A	1/0	N/A	N/A	N/A
		ALT2-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-OUT	TA3_O10	1	0	0	1	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
10	5	Port mode (input)	P10_5	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		Port mode (output)	P10_5	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT1-IN	INTP24	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		ALT2-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-IN	TA3_I11	1	1	0	1	N/A	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		ALT4-IN	CSI2_RYI	1	N/A	1	1	N/A	1	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		ALT1-OUT	$\overline{S\_CS1}$	1	N/A	0	0	N/A	1	1/0	N/A	N/A	N/A	1/0	N/A	N/A	N/A
		ALT2-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-OUT	TA3_O11	1	0	0	1	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT4-OUT	CSI2_RYO	1	N/A	1	1	N/A	1	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
10	6	Port mode (input)	P10_6	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		Port mode (output)	P10_6	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT1-IN	INTP25	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		ALT2-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-IN	TA3_I12	1	1	0	1	N/A	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		ALT4-IN	CSI1F_SSI	1	1	1	1	N/A	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		ALT1-OUT	S_CS2	1	N/A	0	0	N/A	1	1/0	N/A	N/A	N/A	1/0	N/A	N/A	N/A
		ALT2-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-OUT	TA3_O12	1	0	0	1	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

Table 8-38 Register setting for each pin function (49/63)

n	m	Function	Pin name	Feature							Characteristics						
Group	Bit			PMCnm	PMnm	PFCnm	PFCEnm	PIBCnm	PIPCnm	PBDCnm	PUnm	PDnm	PDSCnm	PODCnm	PISnm	PISEnm	PISAnm
10	7	Port mode (input)	P10_7	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		Port mode (output)	P10_7	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT1-IN	INTP26	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		ALT2-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT4-IN	CSI1F_RYI	1	N/A	1	1	N/A	1	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		ALT1-OUT	$\overline{S\_CS3}$	1	N/A	0	0	N/A	1	1/0	N/A	N/A	N/A	1/0	N/A	N/A	N/A
		ALT2-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-OUT	$\overline{S\_SDCS}$	1	N/A	0	1	N/A	1	1/0	N/A	N/A	N/A	1/0	N/A	N/A	N/A
		ALT4-OUT	CSI1F_RYO	1	N/A	1	1	N/A	1	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
10	8	Port mode (input)	P10_8	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	0	0	0
		Port mode (output)	P10_8	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	N/A	N/A	N/A
		ALT1-IN	$\overline{S\_WAIT}$	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	0	0	1
		ALT2-IN	RXD2	1	1	1	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	1	0	0
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT4-IN	SI2	1	1	1	1	N/A	N/A	N/A	1/0	1/0	N/A	N/A	1	0	0
		ALT1-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT2-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
10	9	Port mode (input)	P10_9	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		Port mode (output)	P10_9	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT1-IN	INTP27	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		ALT2-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-IN	TA3_I13	1	1	0	1	N/A	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT1-OUT	S_HLDAK	1	0	0	0	N/A	N/A	1/0	N/A	N/A	N/A	1/0	N/A	N/A	N/A
		ALT2-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-OUT	TA3_O13	1	0	0	1	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT4-OUT	SO2	1	N/A	1	1	N/A	1	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A

Table 8-38 Register setting for each pin function (50/63)

n	m	Function	Pin name	Feature							Characteristics						
Group	Bit			PMCnm	PMnm	PFCnm	PFCEnm	PIBCnm	PIPCnm	PBDCnm	PUnm	PDnm	PDSCnm	PODCnm	PISnm	PISEnm	PISAnm
10	10	Port mode (input)	P10_10	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	0	0	0
		Port mode (output)	P10_10	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	N/A	N/A	N/A
		ALT1-IN	$\overline{S\_HLDRQ}$	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	0	0	1
		ALT2-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-IN	TA3_I14	1	1	0	1	N/A	N/A	N/A	1/0	1/0	N/A	N/A	0	0	0
		ALT4-IN	SCK2[b]	1	N/A	1	1	N/A	1	N/A	1/0	1/0	N/A	N/A	1	0	0
		ALT1-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT2-OUT	TXD2	1	0	1	0	N/A	N/A	1/0	N/A	N/A	0	1/0	N/A	N/A	N/A
		ALT3-OUT	TA3_O14	1	0	0	1	N/A	N/A	1/0	N/A	N/A	0	1/0	N/A	N/A	N/A
		ALT4-OUT	SCK2[b]	1	N/A	1	1	N/A	1	1/0	N/A	N/A	0	1/0	N/A	N/A	N/A
10	11	Port mode (input)	P10_11	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		Port mode (output)	P10_11	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT1-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT2-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-IN	TA3_I15	1	1	0	1	N/A	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		ALT4-IN	CSI2F_RYI	1	N/A	1	1	N/A	1	N/A	1/0	1/0	N/A	N/A	N/A	N/A	0
		ALT1-OUT	$\overline{S\_REFRQ}$	1	0	0	0	N/A	N/A	1/0	N/A	N/A	N/A	1/0	N/A	N/A	N/A
		ALT2-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-OUT	TA3_O15	1	0	0	1	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT4-OUT	CSI2F_RYO	1	N/A	1	1	N/A	1	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
11	0	Port mode (input)	P11_0	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		Port mode (output)	P11_0	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–
		ALT1-IN	TE1_TI0	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT2-IN	TA1_I0	1	1	1	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT1-OUT	S_A1	1	N/A	0	0	N/A	1	1/0	N/A	N/A	N/A	1/0	–	–	–
		ALT2-OUT	TA1_O0	1	0	1	0	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–

Table 8-38 Register setting for each pin function (51/63)

n	m	Function	Pin name	Feature							Characteristics						
Group	Bit			PMCnm	PMnm	PFCnm	PFCEnm	PIBCnm	PIPCnm	PBDCnm	PUnm	PDnm	PDSCnm	PODCnm	PISnm	PISEnm	PISAnm
11	1	Port mode (input)	P11_1	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		Port mode (output)	P11_1	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–
		ALT1-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT2-IN	TA1_I1	1	1	1	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT1-OUT	S_A2	1	N/A	0	0	N/A	1	1/0	N/A	N/A	N/A	1/0	–	–	–
		ALT2-OUT	TA1_O1	1	0	1	0	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
11	2	Port mode (input)	P11_2	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		Port mode (output)	P11_2	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–
		ALT1-IN	TE1_TI1	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT2-IN	TA1_I2	1	1	1	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT1-OUT	S_A3	1	N/A	0	0	N/A	1	1/0	N/A	N/A	N/A	1/0	–	–	–
		ALT2-OUT	TA1_O2	1	0	1	0	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
11	3	Port mode (input)	P11_3	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		Port mode (output)	P11_3	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–
		ALT1-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT2-IN	TA1_I3	1	1	1	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT1-OUT	S_A4	1	N/A	0	0	N/A	1	1/0	N/A	N/A	N/A	1/0	–	–	–
		ALT2-OUT	TA1_O3	1	0	1	0	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–

Table 8-38 Register setting for each pin function (52/63)

n	m	Function	Pin name	Feature							Characteristics						
Group	Bit			PMCnm	PMnm	PFCnm	PFCEnm	PIBCnm	PIPCnm	PBDCnm	PUnm	PDnm	PDSCnm	PODCnm	PISnm	PISEnm	PISAnm
11	4	Port mode (input)	P11_4	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		Port mode (output)	P11_4	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–
		ALT1-IN	TE1_AI	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT2-IN	TA1_I4	1	1	1	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT1-OUT	S_A5	1	N/A	0	0	N/A	1	1/0	N/A	N/A	N/A	1/0	–	–	–
		ALT2-OUT	TA1_O4	1	0	1	0	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
11	5	Port mode (input)	P11_5	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		Port mode (output)	P11_5	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–
		ALT1-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT2-IN	TA1_I5	1	1	1	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT1-OUT	S_A6	1	N/A	0	0	N/A	1	1/0	N/A	N/A	N/A	1/0	–	–	–
		ALT2-OUT	TA1_O5	1	0	1	0	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
11	6	Port mode (input)	P11_6	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		Port mode (output)	P11_6	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–
		ALT1-IN	TE1_BI	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT2-IN	TA1_I6	1	1	1	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT1-OUT	S_A7	1	N/A	0	0	N/A	1	1/0	N/A	N/A	N/A	1/0	–	–	–
		ALT2-OUT	TA1_O6	1	0	1	0	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–

Table 8-38 Register setting for each pin function (53/63)

n	m	Function	Pin name	Feature							Characteristics						
Group	Bit			PMCnm	PMnm	PFCnm	PFCEnm	PIBCnm	PIPCnm	PBDCnm	PUnm	PDnm	PDSCnm	PODCnm	PISnm	PISEnm	PISAnm
11	7	Port mode (input)	P11_7	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		Port mode (output)	P11_7	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–
		ALT1-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT2-IN	TA1_I7	1	1	1	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT1-OUT	S_A8	1	N/A	0	0	N/A	1	1/0	N/A	N/A	N/A	1/0	–	–	–
		ALT2-OUT	TA1_O7	1	0	1	0	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
11	8	Port mode (input)	P11_8	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		Port mode (output)	P11_8	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–
		ALT1-IN	TE1_ZI	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT2-IN	TA1_I8	1	1	1	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT1-OUT	S_A9	1	N/A	0	0	N/A	1	1/0	N/A	N/A	N/A	1/0	–	–	–
		ALT2-OUT	TA1_O8	1	0	1	0	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
11	9	Port mode (input)	P11_9	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		Port mode (output)	P11_9	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–
		ALT1-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT2-IN	TA1_I9	1	1	1	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT1-OUT	S_A10	1	N/A	0	0	N/A	1	1/0	N/A	N/A	N/A	1/0	–	–	–
		ALT2-OUT	TA1_O9	1	0	1	0	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–

Table 8-38 Register setting for each pin function (54/63)

n	m	Function	Pin name	Feature							Characteristics						
Group	Bit			PMCnm	PMnm	PFCnm	PFCEnm	PIBCnm	PIPCnm	PBDCnm	PUnm	PDnm	PDSCnm	PODCnm	PISnm	PISEnm	PISAnm
11	10	Port mode (input)	P11_10	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		Port mode (output)	P11_10	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–
		ALT1-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT2-IN	TA1_I10	1	1	1	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT1-OUT	S_A11	1	N/A	0	0	N/A	1	1/0	N/A	N/A	N/A	1/0	–	–	–
		ALT2-OUT	TA1_O10	1	N/A	1	0	N/A	1	1/0	N/A	N/A	1	1/0	–	–	–
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT4-OUT	CSI1F_CS0	1	0	1	1	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–
11	11	Port mode (input)	P11_11	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		Port mode (output)	P11_11	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–
		ALT1-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT2-IN	TA1_I11	1	1	1	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT1-OUT	S_A12	1	N/A	0	0	N/A	1	1/0	N/A	N/A	N/A	1/0	–	–	–
		ALT2-OUT	TA1_O11	1	N/A	1	0	N/A	1	1/0	N/A	N/A	1	1/0	–	–	–
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT4-OUT	CSI1F_CS1	1	0	1	1	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–
11	12	Port mode (input)	P11_12	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		Port mode (output)	P11_12	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–
		ALT1-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT2-IN	TA1_I12	1	1	1	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT1-OUT	S_A13	1	N/A	0	0	N/A	1	1/0	N/A	N/A	N/A	1/0	–	–	–
		ALT2-OUT	TA1_O12	1	N/A	1	0	N/A	1	1/0	N/A	N/A	1	1/0	–	–	–
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT4-OUT	CSI1F_CS2	1	0	1	1	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–

Table 8-38 Register setting for each pin function (55/63)

n	m	Function	Pin name	Feature							Characteristics						
Group	Bit			PMCnm	PMnm	PFCnm	PFCEnm	PIBCnm	PIPCnm	PBDCnm	PUnm	PDnm	PDSCnm	PODCnm	PISnm	PISEnm	PISAnm
11	13	Port mode (input)	P11_13	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		Port mode (output)	P11_13	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–
		ALT1-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT2-IN	TA1_I13	1	1	1	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT1-OUT	S_A14	1	N/A	0	0	N/A	1	1/0	N/A	N/A	N/A	1/0	–	–	–
		ALT2-OUT	TA1_O13	1	N/A	1	0	N/A	1	1/0	N/A	N/A	1	1/0	–	–	–
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT4-OUT	CSI1F_CS3	1	0	1	1	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–
11	14	Port mode (input)	P11_14	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		Port mode (output)	P11_14	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–
		ALT1-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT2-IN	TA1_I14	1	1	1	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT1-OUT	S_A15	1	N/A	0	0	N/A	1	1/0	N/A	N/A	N/A	1/0	–	–	–
		ALT2-OUT	TA1_O14	1	N/A	1	0	N/A	1	1/0	N/A	N/A	1	1/0	–	–	–
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT4-OUT	CSI1F_CS4	1	0	1	1	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–
11	15	Port mode (input)	P11_15	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		Port mode (output)	P11_15	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–
		ALT1-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT2-IN	TA3_I15	1	1	1	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	–	–	–
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT1-OUT	S_A16	1	N/A	0	0	N/A	1	1/0	N/A	N/A	N/A	1/0	–	–	–
		ALT2-OUT	TA3_O15	1	N/A	1	0	N/A	1	1/0	N/A	N/A	1	1/0	–	–	–
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–	–	–
		ALT4-OUT	CSI1F_CS5	1	0	1	1	N/A	N/A	1/0	N/A	N/A	1	1/0	–	–	–

Table 8-38 Register setting for each pin function (56/63)

n	m	Function	Pin name	Feature							Characteristics						
Group	Bit			PMCnm	PMnm	PFCnm	PFCEnm	PIBCnm	PIPCnm	PBDCnm	PUnm	PDnm	PDSCnm	PODCnm	PISnm	PISEnm	PISAnm
12	0	Port mode (input)	P12_0	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	0	0	–
		Port mode (output)	P12_0	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	N/A	N/A	–
		ALT1-IN	INTP0	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	1	0	–
		ALT2-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–
		ALT1-OUT	S_A17	1	N/A	0	0	N/A	1	1/0	N/A	N/A	1	1/0	N/A	N/A	–
		ALT2-OUT	ADCNV0	1	0	1	0	N/A	N/A	1/0	N/A	N/A	0	1/0	N/A	N/A	–
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–
		ALT4-OUT	CSI1F_CS6	1	0	1	1	N/A	N/A	1/0	N/A	N/A	0	1/0	N/A	N/A	–
12	1	Port mode (input)	P12_1	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	0	0	–
		Port mode (output)	P12_1	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	N/A	N/A	–
		ALT1-IN	INTP1	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	1	0	–
		ALT2-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–
		ALT1-OUT	S_A18	1	N/A	0	0	N/A	1	1/0	N/A	N/A	1	1/0	N/A	N/A	–
		ALT2-OUT	ADCNV1	1	0	1	0	N/A	N/A	1/0	N/A	N/A	0	1/0	N/A	N/A	–
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–
		ALT4-OUT	CSI1F_CS7	1	0	1	1	N/A	N/A	1/0	N/A	N/A	0	1/0	N/A	N/A	–
12	2	Port mode (input)	P12_2	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	0	0	–
		Port mode (output)	P12_2	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	N/A	N/A	–
		ALT1-IN	INTP2	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	1	0	–
		ALT2-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–
		ALT4-IN	CSI0_SSI	1	1	1	1	N/A	N/A	N/A	1/0	1/0	N/A	N/A	0	0	–
		ALT1-OUT	S_A19	1	N/A	0	0	N/A	1	1/0	N/A	N/A	1	1/0	N/A	N/A	–
		ALT2-OUT	ADCNV2	1	0	1	0	N/A	N/A	1/0	N/A	N/A	0	1/0	N/A	N/A	–
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–

Table 8-38 Register setting for each pin function (57/63)

n	m	Function	Pin name	Feature							Characteristics						
Group	Bit			PMCnm	PMnm	PFCnm	PFCEnm	PIBCnm	PIPCnm	PBDCnm	PUnm	PDnm	PDSCnm	PODCnm	PISnm	PISEnm	PISAnm
12	3	Port mode (input)	P12_3	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	0	0	–
		Port mode (output)	P12_3	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	N/A	N/A	–
		ALT1-IN	INTP3	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	1	0	–
		ALT2-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–
		ALT4-IN	CSI0_RYI	1	N/A	1	1	N/A	1	N/A	1/0	1/0	N/A	N/A	0	0	–
		ALT1-OUT	S_A20	1	N/A	0	0	N/A	1	1/0	N/A	N/A	1	1/0	N/A	N/A	–
		ALT2-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–
		ALT4-OUT	CSI0_RYO	1	N/A	1	1	N/A	1	1/0	N/A	N/A	0	1/0	N/A	N/A	–
12	4	Port mode (input)	P12_4	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	0	0	–
		Port mode (output)	P12_4	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	N/A	N/A	–
		ALT1-IN	INTP4	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	1	0	–
		ALT2-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–
		ALT1-OUT	S_A21	1	N/A	0	0	N/A	1	1/0	N/A	N/A	1	1/0	N/A	N/A	–
		ALT2-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–
		ALT4-OUT	SO0	1	N/A	1	1	N/A	1	1/0	N/A	N/A	0	1/0	N/A	N/A	–
12	5	Port mode (input)	P12_5	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	0	0	–
		Port mode (output)	P12_5	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	N/A	N/A	–
		ALT1-IN	INTP5	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	0	0	–
		ALT2-IN	RXD0	1	1	1	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	1	0	–
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–
		ALT4-IN	SI0	1	1	1	1	N/A	N/A	N/A	1/0	1/0	N/A	N/A	1	0	–
		ALT1-OUT	S_A22	1	N/A	0	0	N/A	1	1/0	N/A	N/A	1	1/0	N/A	N/A	–
		ALT2-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–

Table 8-38 Register setting for each pin function (58/63)

n	m	Function	Pin name	Feature							Characteristics						
Group	Bit			PMCnm	PMnm	PFCnm	PFCEnm	PIBCnm	PIPCnm	PBDCnm	PUnm	PDnm	PDSCnm	PODCnm	PISnm	PISEnm	PISAnm
12	6	Port mode (input)	P12_6	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	0	0	–
		Port mode (output)	P12_6	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	N/A	N/A	–
		ALT1-IN	INTP6	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	0	0	–
		ALT2-IN	ADTRG11	1	1	1	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	0	0	–
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–
		ALT4-IN	SCK0[b]	1	N/A	1	1	N/A	1	N/A	1/0	1/0	N/A	N/A	1	0	–
		ALT1-OUT	S_A23	1	N/A	0	0	N/A	1	1/0	N/A	N/A	1	1/0	N/A	N/A	–
		ALT2-OUT	TXD0	1	0	1	0	N/A	N/A	1/0	N/A	N/A	0	1/0	N/A	N/A	–
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–
		ALT4-OUT	SCK0[b]	1	N/A	1	1	N/A	1	1/0	N/A	N/A	0	1/0	N/A	N/A	–
12	7	Port mode (input)	P12_7	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	–
		Port mode (output)	P12_7	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	–
		ALT1-IN	INTP7	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	N/A	N/A	–
		ALT2-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–
		ALT1-OUT	S_A24	1	N/A	0	0	N/A	1	1/0	N/A	N/A	N/A	1/0	N/A	N/A	–
		ALT2-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–
		ALT4-OUT	SO2F	1	N/A	1	1	N/A	1	1/0	N/A	N/A	1	1/0	N/A	N/A	–
12	8	Port mode (input)	P12_8	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	0	0	–
		Port mode (output)	P12_8	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	N/A	N/A	–
		ALT1-IN	INTP8	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	0	0	–
		ALT2-IN	RXD2F	1	1	1	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	1	0	–
		ALT3-IN	SDA2[a]	1	0	0	1	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0	1	–
		ALT4-IN	SI2F	1	1	1	1	N/A	N/A	N/A	1/0	1/0	N/A	N/A	1	0	–
		ALT1-OUT	S_A25	1	N/A	0	0	N/A	1	1/0	N/A	N/A	1	1/0	N/A	N/A	–
		ALT2-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–
		ALT3-OUT	SDA2[a]	1	0	0	1	N/A	N/A	1	N/A	N/A	0	1	N/A	N/A	–
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	–

Table 8-38 Register setting for each pin function (59/63)

n	m	Function	Pin name	Feature							Characteristics						
Group	Bit			PMCnm	PMnm	PFCnm	PFCEnm	PIBCnm	PIPCnm	PBDCnm	PUnm	PDnm	PDSCnm	PODCnm	PISnm	PISEnm	PISAnm
12	9	Port mode (input)	P12_9	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	0	0	–
		Port mode (output)	P12_9	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	N/A	N/A	–
		ALT1-IN	INTP9	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	0	0	–
		ALT2-IN	ADTRG21	1	1	1	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	0	0	–
		ALT3-IN	SCL2[a]	1	0	0	1	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0	1	–
		ALT4-IN	SCK2F[b]	1	N/A	1	1	N/A	1	N/A	1/0	1/0	N/A	N/A	1	0	–
		ALT1-OUT	S_A26	1	N/A	0	0	N/A	1	1/0	N/A	N/A	1	1/0	N/A	N/A	–
		ALT2-OUT	TXD2F	1	0	1	0	N/A	N/A	1/0	N/A	N/A	0	1/0	N/A	N/A	–
		ALT3-OUT	SCL2[a]	1	0	0	1	N/A	N/A	1	N/A	N/A	0	1	N/A	N/A	–
		ALT4-OUT	SCK2F[b]	1	N/A	1	1	N/A	1	1/0	N/A	N/A	0	1/0	N/A	N/A	–
13	0	Port mode (input)	P13_0	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	0	0	0
		Port mode (output)	P13_0	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	N/A	N/A	N/A
		ALT1-IN	UCLK	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	1	0	0
		ALT2-IN	ADTRG01	1	1	1	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	0	0	0
		ALT3-IN	S_DMARQ0	1	1	0	1	N/A	N/A	N/A	1/0	1/0	N/A	N/A	0	0	1
		ALT4-IN	CSI5_SSI	1	1	1	1	N/A	N/A	N/A	1/0	1/0	N/A	N/A	0	0	0
		ALT1-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT2-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
13	1	Port mode (input)	P13_1	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	0	0	0
		Port mode (output)	P13_1	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	N/A	N/A	N/A
		ALT1-IN	INTP0	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	1	0	0
		ALT2-IN	ADTRG00	1	1	1	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	0	0	0
		ALT3-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT4-IN	CSI5_RYI	1	N/A	1	1	N/A	1	N/A	1/0	1/0	N/A	N/A	0	0	0
		ALT1-OUT	S_CS3	1	N/A	0	0	N/A	1	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT2-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-OUT	S_DMAAK0	1	0	0	1	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT4-OUT	CSI5_RYO	1	N/A	1	1	N/A	1	1/0	N/A	N/A	0	1/0	N/A	N/A	N/A

Table 8-38 Register setting for each pin function (60/63)

n	m	Function	Pin name	Feature							Characteristics						
Group	Bit			PMCnm	PMnm	PFCnm	PFCEnm	PIBCnm	PIPCnm	PBDCnm	PUnm	PDnm	PDSCnm	PODCnm	PISnm	PISEnm	PISAnm
13	2	Port mode (input)	P13_2	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	0	0	0
		Port mode (output)	P13_2	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	N/A	N/A	N/A
		ALT1-IN	INTP1	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	1	0	0
		ALT2-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-IN	S_DMARQ1	1	1	0	1	N/A	N/A	N/A	1/0	1/0	N/A	N/A	0	0	1
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT1-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT2-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-OUT	PPON	1	0	0	1	N/A	N/A	1/0	N/A	N/A	0	1/0	N/A	N/A	N/A
		ALT4-OUT	SO1F	1	N/A	1	1	N/A	1	1/0	N/A	N/A	0	1/0	N/A	N/A	N/A
13	3	Port mode (input)	P13_3	0	1	N/A	N/A	1	N/A	N/A	1/0	1/0	N/A	N/A	0	0	0
		Port mode (output)	P13_3	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	0	1/0	N/A	N/A	N/A
		ALT1-IN	INTP2	1	1	0	0	N/A	N/A	N/A	1/0	1/0	N/A	N/A	1	0	0
		ALT2-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-IN	OCI	1	1	0	1	1	N/A	N/A	1/0	1/0	N/A	N/A	1	0	0
		ALT4-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT1-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT2-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-OUT	S_DMAAK1	1	0	0	1	N/A	N/A	1/0	N/A	N/A	1	1/0	N/A	N/A	N/A
		ALT4-OUT	SO5	1	N/A	1	1	N/A	1	1/0	N/A	N/A	0	1/0	N/A	N/A	N/A
13	4	Port mode (input)	P13_4	0	1	N/A	N/A	1	N/A	N/A	N/A	N/A	N/A	N/A	0	0	N/A
		Port mode (output)	P13_4	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	N/A	1/0	N/A	N/A	N/A
		ALT1-IN	INTP3	1	1	0	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	1	0	N/A
		ALT2-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-IN	SCL1[a]	1	0	0	1	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0	1	N/A
		ALT4-IN	SCK1F[b]	1	N/A	1	1	N/A	1	N/A	N/A	N/A	N/A	N/A	1	0	N/A
		ALT1-OUT	CAN0TXD	1	0	0	0	N/A	N/A	1/0	N/A	N/A	N/A	1/0	N/A	N/A	N/A
		ALT2-OUT	TXD1F	1	0	1	0	N/A	N/A	1/0	N/A	N/A	N/A	1/0	N/A	N/A	N/A
		ALT3-OUT	SCL1[a]	1	0	0	1	N/A	N/A	1	N/A	N/A	N/A	1	N/A	N/A	N/A
		ALT4-OUT	SCK1F[b]	1	N/A	1	1	N/A	1	1/0	N/A	N/A	N/A	1/0	N/A	N/A	N/A

Table 8-38 Register setting for each pin function (61/63)

n	m	Function	Pin name	Feature							Characteristics						
Group	Bit			PMCnm	PMnm	PFCnm	PFCEnm	PIBCnm	PIPCnm	PBDCnm	PUnm	PDnm	PDSCnm	PODCnm	PISnm	PISEnm	PISAnm
13	5	Port mode (input)	P13_5	0	1	N/A	N/A	1	N/A	N/A	N/A	N/A	N/A	N/A	0	0	N/A
		Port mode (output)	P13_5	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	N/A	1/0	N/A	N/A	N/A
		ALT1-IN	CAN0RXD	1	1	0	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0	1	N/A
		ALT2-IN	RXD1F	1	1	1	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	1	0	N/A
		ALT3-IN	SDA1[a]	1	0	0	1	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0	1	N/A
		ALT4-IN	SI1F	1	1	1	1	N/A	N/A	N/A	N/A	N/A	N/A	N/A	1	0	N/A
		ALT1-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT2-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-OUT	SDA1[a]	1	0	0	1	N/A	N/A	1	N/A	N/A	N/A	1	N/A	N/A	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
13	6	Port mode (input)	P13_6	0	1	N/A	N/A	1	N/A	N/A	N/A	N/A	N/A	N/A	0	0	N/A
		Port mode (output)	P13_6	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	N/A	1/0	N/A	N/A	N/A
		ALT1-IN	INTP4	1	1	0	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	1	0	N/A
		ALT2-IN	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-IN	SCL5[a]	1	0	0	1	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0	1	N/A
		ALT4-IN	SCK5[b]	1	N/A	1	1	N/A	1	N/A	N/A	N/A	N/A	N/A	1	0	N/A
		ALT1-OUT	CAN1TXD	1	0	0	0	N/A	N/A	1/0	N/A	N/A	N/A	1/0	N/A	N/A	N/A
		ALT2-OUT	TXD5	1	0	1	0	N/A	N/A	1/0	N/A	N/A	N/A	1/0	N/A	N/A	N/A
		ALT3-OUT	SCL5[a]	1	0	0	1	N/A	N/A	1	N/A	N/A	N/A	1	N/A	N/A	N/A
		ALT4-OUT	SCK5[b]	1	N/A	1	1	N/A	1	1/0	N/A	N/A	N/A	1/0	N/A	N/A	N/A
13	7	Port mode (input)	P13_7	0	1	N/A	N/A	1	N/A	N/A	N/A	N/A	N/A	N/A	0	0	N/A
		Port mode (output)	P13_7	0	0	N/A	N/A	N/A	N/A	1/0	N/A	N/A	N/A	1/0	N/A	N/A	N/A
		ALT1-IN	CAN1RXD	1	1	0	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0	1	N/A
		ALT2-IN	RXD5	1	1	1	0	N/A	N/A	N/A	N/A	N/A	N/A	N/A	1	0	N/A
		ALT3-IN	SDA5[a]	1	0	0	1	N/A	N/A	N/A	N/A	N/A	N/A	N/A	0	1	N/A
		ALT4-IN	SI5	1	1	1	1	N/A	N/A	N/A	N/A	N/A	N/A	N/A	1	0	N/A
		ALT1-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT2-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A
		ALT3-OUT	SDA5[a]	1	0	0	1	N/A	N/A	1	N/A	N/A	N/A	1	N/A	N/A	N/A
		ALT4-OUT	–	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

Table 8-38 Register setting for each pin function (62/63)

n	m	Function	Pin name	Feature							Characteristics						
Group	Bit			PMCnm	PMnm	PFCnm	PFCEnm	PIBCnm	PIPCnm	PBDCnm	PUnm	PDnm	PDSCnm	PODCnm	PISnm	PISEnm	PISAnm
14	0	Port mode (input)	P14_0[c]	–	–	–	–	1	–	–	–	–	–	–	–	–	–
		Port mode (output)	–	–	–	–	–	N/A	–	–	–	–	–	–	–	–	–
		ALT1-IN	ANI06	–	–	–	–	0	–	–	–	–	–	–	–	–	–
		ALT2-IN	–	–	–	–	–	N/A	–	–	–	–	–	–	–	–	–
		ALT3-IN	–	–	–	–	–	N/A	–	–	–	–	–	–	–	–	–
		ALT4-IN	–	–	–	–	–	N/A	–	–	–	–	–	–	–	–	–
		ALT1-OUT	–	–	–	–	–	N/A	–	–	–	–	–	–	–	–	–
		ALT2-OUT	–	–	–	–	–	N/A	–	–	–	–	–	–	–	–	–
		ALT3-OUT	–	–	–	–	–	N/A	–	–	–	–	–	–	–	–	–
		ALT4-OUT	–	–	–	–	–	N/A	–	–	–	–	–	–	–	–	–
14	1	Port mode (input)	P14_1[c]	–	–	–	–	1	–	–	–	–	–	–	–	–	–
		Port mode (output)	–	–	–	–	–	N/A	–	–	–	–	–	–	–	–	–
		ALT1-IN	ANI07	–	–	–	–	0	–	–	–	–	–	–	–	–	–
		ALT2-IN	–	–	–	–	–	N/A	–	–	–	–	–	–	–	–	–
		ALT3-IN	–	–	–	–	–	N/A	–	–	–	–	–	–	–	–	–
		ALT4-IN	–	–	–	–	–	N/A	–	–	–	–	–	–	–	–	–
		ALT1-OUT	–	–	–	–	–	N/A	–	–	–	–	–	–	–	–	–
		ALT2-OUT	–	–	–	–	–	N/A	–	–	–	–	–	–	–	–	–
		ALT3-OUT	–	–	–	–	–	N/A	–	–	–	–	–	–	–	–	–
		ALT4-OUT	–	–	–	–	–	N/A	–	–	–	–	–	–	–	–	–
14	2	Port mode (input)	P14_2[c]	–	–	–	–	1	–	–	–	–	–	–	–	–	–
		Port mode (output)	–	–	–	–	–	N/A	–	–	–	–	–	–	–	–	–
		ALT1-IN	ANI08	–	–	–	–	0	–	–	–	–	–	–	–	–	–
		ALT2-IN	–	–	–	–	–	N/A	–	–	–	–	–	–	–	–	–
		ALT3-IN	–	–	–	–	–	N/A	–	–	–	–	–	–	–	–	–
		ALT4-IN	–	–	–	–	–	N/A	–	–	–	–	–	–	–	–	–
		ALT1-OUT	–	–	–	–	–	N/A	–	–	–	–	–	–	–	–	–
		ALT2-OUT	–	–	–	–	–	N/A	–	–	–	–	–	–	–	–	–
		ALT3-OUT	–	–	–	–	–	N/A	–	–	–	–	–	–	–	–	–
		ALT4-OUT	–	–	–	–	–	N/A	–	–	–	–	–	–	–	–	–

Table 8-38 Register setting for each pin function (63/63)

n	m	Function	Pin name	Feature							Characteristics						
Group	Bit			PMCnm	PMnm	PFCnm	PFCEnm	PIBCnm	PIPCnm	PBDCnm	PUnm	PDnm	PDSCnm	PODCnm	PISnm	PISEnm	PISAnm
14	3	Port mode (input)	P14_3[c]	–	–	–	–	1	–	–	–	–	–	–	–	–	–
		Port mode (output)	–	–	–	–	–	N/A	–	–	–	–	–	–	–	–	–
		ALT1-IN	ANI09	–	–	–	–	0	–	–	–	–	–	–	–	–	–
		ALT2-IN	–	–	–	–	–	N/A	–	–	–	–	–	–	–	–	–
		ALT3-IN	–	–	–	–	–	N/A	–	–	–	–	–	–	–	–	–
		ALT4-IN	–	–	–	–	–	N/A	–	–	–	–	–	–	–	–	–
		ALT1-OUT	–	–	–	–	–	N/A	–	–	–	–	–	–	–	–	–
		ALT2-OUT	–	–	–	–	–	N/A	–	–	–	–	–	–	–	–	–
		ALT3-OUT	–	–	–	–	–	N/A	–	–	–	–	–	–	–	–	–
		ALT4-OUT	–	–	–	–	–	N/A	–	–	–	–	–	–	–	–	–
14	4	Port mode (input)	P14_4[c]	–	–	–	–	1	–	–	–	–	–	–	–	–	–
		Port mode (output)	–	–	–	–	–	N/A	–	–	–	–	–	–	–	–	–
		ALT1-IN	ANI010	–	–	–	–	0	–	–	–	–	–	–	–	–	–
		ALT2-IN	–	–	–	–	–	N/A	–	–	–	–	–	–	–	–	–
		ALT3-IN	–	–	–	–	–	N/A	–	–	–	–	–	–	–	–	–
		ALT4-IN	–	–	–	–	–	N/A	–	–	–	–	–	–	–	–	–
		ALT1-OUT	–	–	–	–	–	N/A	–	–	–	–	–	–	–	–	–
		ALT2-OUT	–	–	–	–	–	N/A	–	–	–	–	–	–	–	–	–
		ALT3-OUT	–	–	–	–	–	N/A	–	–	–	–	–	–	–	–	–
		ALT4-OUT	–	–	–	–	–	N/A	–	–	–	–	–	–	–	–	–
14	5	Port mode (input)	P14_5[c]	–	–	–	–	1	–	–	–	–	–	–	–	–	–
		Port mode (output)	–	–	–	–	–	N/A	–	–	–	–	–	–	–	–	–
		ALT1-IN	ANI011	–	–	–	–	0	–	–	–	–	–	–	–	–	–
		ALT2-IN	–	–	–	–	–	N/A	–	–	–	–	–	–	–	–	–
		ALT3-IN	–	–	–	–	–	N/A	–	–	–	–	–	–	–	–	–
		ALT4-IN	–	–	–	–	–	N/A	–	–	–	–	–	–	–	–	–
		ALT1-OUT	–	–	–	–	–	N/A	–	–	–	–	–	–	–	–	–
		ALT2-OUT	–	–	–	–	–	N/A	–	–	–	–	–	–	–	–	–
		ALT3-OUT	–	–	–	–	–	N/A	–	–	–	–	–	–	–	–	–
		ALT4-OUT	–	–	–	–	–	N/A	–	–	–	–	–	–	–	–	–

a) The settings for input and output must be specified. For registers that have the specified setting for input or output, specify the appropriate value.

b) Specify the setting for input or output according to the function to use.

c) Be sure to turn on the A/D converter by setting the ADCA0CTL1.ADCA0GPS, even if this pin is used for port input.

8.5 Port Filters

The input signals at some pins are passed through a filter to remove noise and glitches. The microcontroller supports both analog and digital filters.

The first section provides an overview, which port input signals are equipped with which kind of filter, their control registers an bits and the register addresses.

A detailed description of the analog and digital filter functions and their control registers follow.

8.5.1 Analog and digital filter assignments

Table 8-39 "Input signals that have analog filters and their control registers" lists the input signals that are equipped with an analog filter and their control registers.

Table 8-40 "Input signals that have digital filters and their control registers 1" and *Table 8-41 "Input signals that have digital filters and their control registers 2"* list the input signals that are equipped with a digital filter and their control registers.

Table 8-42 "Input signals that have analog/digital common noise filters and their control registers" lists the input signals that are equipped with an analog/digital common noise filter.

Table 8-39 Input signals that have analog filters and their control registers

Signal	FCLAn		
	Register		Address
NMI	FCLA15	CTL0	FF41 4F00$_H$
INTP0		CTL1	FF41 4F04$_H$
INTP1		CTL2	FF41 4F08$_H$
INTP2		CTL3	FF41 4F0C$_H$
INTP3		CTL4	FF41 4F10$_H$
INTP4		CTL5	FF41 4F14$_H$
ESO0	FCLA16	CTL0	FF41 5000$_H$
ESO1		CTL1	FF41 5004$_H$
ESO2		CTL2	FF41 5008$_H$
ESO3		CTL3	FF41 500C$_H$

Table 8-40 Input signals that have digital filters and their control registers 1 (1/3)

Signal	DNFAn			
	Register	Address	Filter enable bit	
TA0_I0	DNFA0CTL DNFA0EN DNFA0ENH DNFA0ENL	FF41 0000$_H$ FF41 0004$_H$ FF41 0008$_H$ FF41 000C$_H$	DNFA0EN.DNFA0	NFEN0
TA0_I1				NFEN1
TA0_I2				NFEN2
TA0_I3				NFEN3
TA0_I4				NFEN4
TA0_I5				NFEN5
TA0_I6				NFEN6
TA0_I7				NFEN7
TA0_I8				NFEN8
TA0_I9				NFEN9
TA0_I10				NFEN10
TA0_I11				NFEN11
TA0_I12				NFEN12
TA0_I13				NFEN13
TA0_I14				NFEN14
TA0_I15				NFEN15
TA1_I0	DNFA2CTL DNFA2EN DNFA2ENH DNFA2ENL	FF41 0200$_H$ FF41 0204$_H$ FF41 0208$_H$ FF41 020C$_H$	DNFA2EN.DNFA0	NFEN0
TA1_I1				NFEN1
TA1_I2				NFEN2
TA1_I3				NFEN3
TA1_I4				NFEN4
TA1_I5				NFEN5
TA1_I6				NFEN6
TA1_I7				NFEN7
TA1_I8				NFEN8
TA1_I9				NFEN9
TA1_I10				NFEN10
TA1_I11				NFEN11
TA1_I12				NFEN12
TA1_I13				NFEN13
TA1_I14				NFEN14
TA1_I15				NFEN15

Table 8-40 Input signals that have digital filters and their control registers 1 (2/3)

Signal	DNFAn			
	Register	Address	Filter enable bit	
TA2_I0	DNFA4CTL DNFA4EN DNFA4ENH DNFA4ENL	FF41 0400$_H$ FF41 0404$_H$ FF41 0408$_H$ FF41 040C$_H$	DNFA4EN.DNFA4	NFEN0
TA2_I1				NFEN1
TA2_I2				NFEN2
TA2_I3				NFEN3
TA2_I4				NFEN4
TA2_I5				NFEN5
TA2_I6				NFEN6
TA2_I7				NFEN7
TA2_I8				NFEN8
TA2_I9				NFEN9
TA2_I10				NFEN10
TA2_I11				NFEN11
TA2_I12				NFEN12
TA2_I13				NFEN13
TA2_I14				NFEN14
TA2_I15				NFEN15
TA3_I0	DNFA6CTL DNFA6EN DNFA6ENH DNFA6ENL	FF41 0600$_H$ FF41 0604$_H$ FF41 0608$_H$ FF41 060C$_H$	DNFA6EN.DNFA6	NFEN0
TA3_I1				NFEN1
TA3_I2				NFEN2
TA3_I3				NFEN3
TA3_I4				NFEN4
TA3_I5				NFEN5
TA3_I6				NFEN6
TA3_I7				NFEN7
TA3_I8				NFEN8
TA3_I9				NFEN9
TA3_I10				NFEN10
TA3_I11				NFEN11
TA3_I12				NFEN12
TA3_I13				NFEN13
TA3_I14				NFEN14
TA3_I15				NFEN15
TE0_TI0	DNFA8CTL DNFA8EN DNFA8ENH DNFA8ENL	FF41 0800$_H$ FF41 0804$_H$ FF41 080C$_H$	DNFA8EN.DNFA8	NFEN0
TE0_TI1				NFEN1
TE0_AI				NFEN2
TE0_BI				NFEN3
TE0_ZI				NFEN4

Table 8-40 Input signals that have digital filters and their control registers 1 (3/3)

Signal	DNFAn			
	Register	Address	Filter enable bit	
TE1_TI0	DNFA9CTL DNFA9EN DNFA9ENL	FF41 0900$_H$ FF41 0904$_H$ FF41 090C$_H$	DNFA9EN.DNFA9	NFEN0
TE1_TI1				NFEN1
TE1_AI				NFEN2
TE1_BI				NFEN3
TE1_ZI				NFEN4
TJ_I0	DNFA10CTL DNFA10EN DNFA10ENL	FF41 0A00$_H$ FF41 0A04$_H$ FF41 0A0C$_H$	DNFA10EN.DNFA10	NFEN0
TJ_I1				NFEN1
TJ_I2				NFEN2
TJ_I3				NFEN3

Table 8-41 Input signals that have digital filters and their control registers 2

Signal	DNFAn				FCLAn		
	Register	Address	Filter enable bit		Register		Address
INTP5	DNFA12CTL DNFA12EN DNFA12ENH DNFA12ENL	FF41 0C00$_H$ FF41 0C04$_H$ FF41 0C0C$_H$	DNFA12EN.DNFA12	NFEN0	FCLA12	CTL0	FF41 4C00$_H$
INTP6				NFEN1		CTL1	FF41 4C04$_H$
INTP7				NFEN2		CTL2	FF41 4C08$_H$
INTP8				NFEN3		CTL3	FF41 4C0C$_H$
INTP9				NFEN4		CTL4	FF41 4C10$_H$
INTP10				NFEN5		CTL5	FF41 4C14$_H$
INTP11				NFEN6		CTL6	FF41 4C18$_H$
INTP12				NFEN7		CTL7	FF41 4C1C$_H$
INTP13	DNFA13CTL DNFA13EN DNFA13ENH DNFA13ENL	FF41 0D00$_H$ FF41 0D04$_H$ FF41 0D0C$_H$	DNFA13EN.DNFA13	NFEN0	FCLA13	CTL0	FF41 4D00$_H$
INTP14				NFEN1		CTL1	FF41 4D04$_H$
INTP15				NFEN2		CTL2	FF41 4D08$_H$
INTP16				NFEN3		CTL3	FF41 4D0C$_H$
INTP17				NFEN4		CTL4	FF41 4D10$_H$
INTP18				NFEN5		CTL5	FF41 4D14$_H$
INTP19				NFEN6		CTL6	FF41 4D18$_H$
INTP20				NFEN7		CTL7	FF41 4D1C$_H$
INTP21	DNFA14CTL DNFA14EN DNFA14ENH DNFA14ENL	FF41 0E00$_H$ FF41 0E04$_H$ FF41 0E0C$_H$	DNFA14EN.DNFA14	NFEN0	FCLA14	CTL0	FF41 4E00$_H$
INTP22				NFEN1		CTL1	FF41 4E04$_H$
INTP23				NFEN2		CTL2	FF41 4E08$_H$
INTP24				NFEN3		CTL3	FF41 4E0C$_H$
INTP25				NFEN4		CTL4	FF41 4E10$_H$
INTP26				NFEN5		CTL5	FF41 4E14$_H$
INTP27				NFEN6		CTL6	FF41 4E18$_H$

Table 8-42 Input signals that have analog/digital common noise filters and their control registers

Signal	DNFAn				FCLAn		
	Register	Address	Filter enable bit[a]		Register		Address
ADTRG00	DNFA11CTL DNFA11EN DNFA11ENL	FF41 0B00$_H$ FF41 0B04$_H$ FF41 0B0C$_H$	DNFA11EN.DNFA11	NFEN0	FCLA11	CTL0	FF41 4B00$_H$
ADTRG01				NFEN1		CTL1	FF41 4B04$_H$
ADTRG10				NFEN2		CTL2	FF41 4B08$_H$
ADTRG11				NFEN3		CTL3	FF41 4B0C$_H$
ADTRG20				NFEN4		CTL4	FF41 4B10$_H$
ADTRG21				NFEN5		CTL5	FF41 4B14$_H$

a) The noise filter functions as a digital filter when DNFAnEN.DNFAnNFENm = 1, or as an analog filter when DNFAnEN.DNFAnNFENm = 0.

8.5.2 Functional Description

Two types of filters are used to generate two kinds of signals, according to the purpose of the input signal.

Event output Event output signals are used to transmit an external event. The waveform of the input signal after passing through a filter is not retained as the signal to be transmitted internally, because level and edge detection are performed on the signals after they pass through the filter. An external interrupt is a typical example of an event signal.

Alternative function Alternative function signals are used to input an input signal to an alternative function block. When an input signal passes through a filter, noise and spikes are eliminated. The waveform of the internal signal after passing through a filter is retained as the signal to be transmitted internally. A timer signal input to measure the frequency is a typical example of an alternative function signal.

(1) Analog filters

The block diagram of the analog filters is shown in the diagram below.

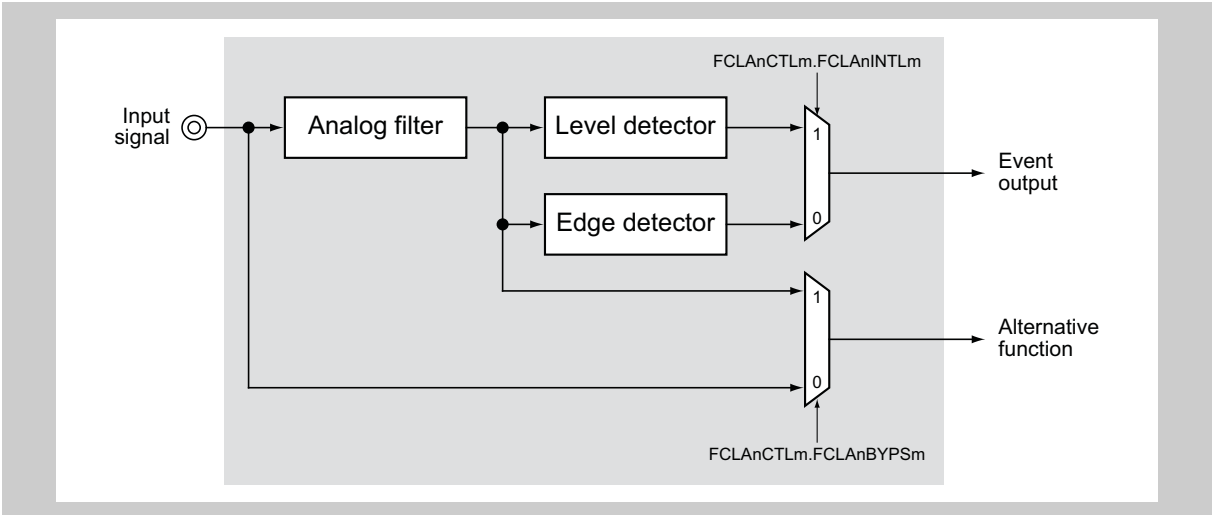


Figure 8-6 Analog filter block diagram

(a) Event signals

When an input signal passes through an analog filter, noise and spikes are eliminated. After passing through an analog filter, the level or edge of the signal is used to judge whether an event has occurred, and an event output signal is generated if an event is detected.

The detection mode is selected by the control bit FCLAnCTLm.FCLAnINTLm:

- FCLAnINTLm = 0: edge detection mode
  The detection of a rising or falling edge can be activated separately by FCLAnCTLm.FCLAnINTRm and FCLAnCTLm.FCLAnINTFm respectively.
- FCLAnINTLm = 1: level detection mode
  The detection of a high level or low level can be selected by FCLAnCTLm.FCLAnINTRm.

The table below summarizes the detection conditions of the analog filter.

Table 8-43 Analog filter event detection conditions

FCLAnINTLm	FCLAnINTFm	FCLAnINTRm	Edge detection	Level detection
0	0	0	no detection	not active
	0	1	rising edge	
	1	0	falling edge	
	1	1	both edges	
1	X	0	not active	low level
		1		high level

(b) Signals for alternative functions

The input signal is passed through an analog filter to eliminate noise and spikes.

Bypass In case the alternative function requires the original signal at its input, the analog filter can be bypassed by setting FCLAnCTLm.FCLAnBYPSm = 1.

Wake-up Analog filtered alternative function signals have wake-up capability. That means, an input signal can wake-up the microcontroller from stand-by modes, even if the clock supply is stopped.
Note that edge and level detection are not operable, if the clock is stopped in stand-by. Thus the event output signals can not issue a wake-up.

(c) Analog filter characteristic

The characteristics of the analog filter as well as of the level and edge detectors are specified in the Electrical Target Specification document.

(d) Analog filters control registers

For each input signal, that is equipped with an analog filter, a dedicated control register FCLAnCTLm is provided.

The registers are ordered in groups of 8 registers with the same index n. The register index m ranges from 0 to 7:

FCLA group n: FCLAnCTL0 to FCLAnCTL7

The assignment of the input signals to the control registers and their addresses is given in *Table 8-39 "Input signals that have analog filters and their control registers"* in *8.5.1 "Analog and digital filter assignments"*.

(2) Digital filters

The block diagram of the analog filters is shown in the diagram below.

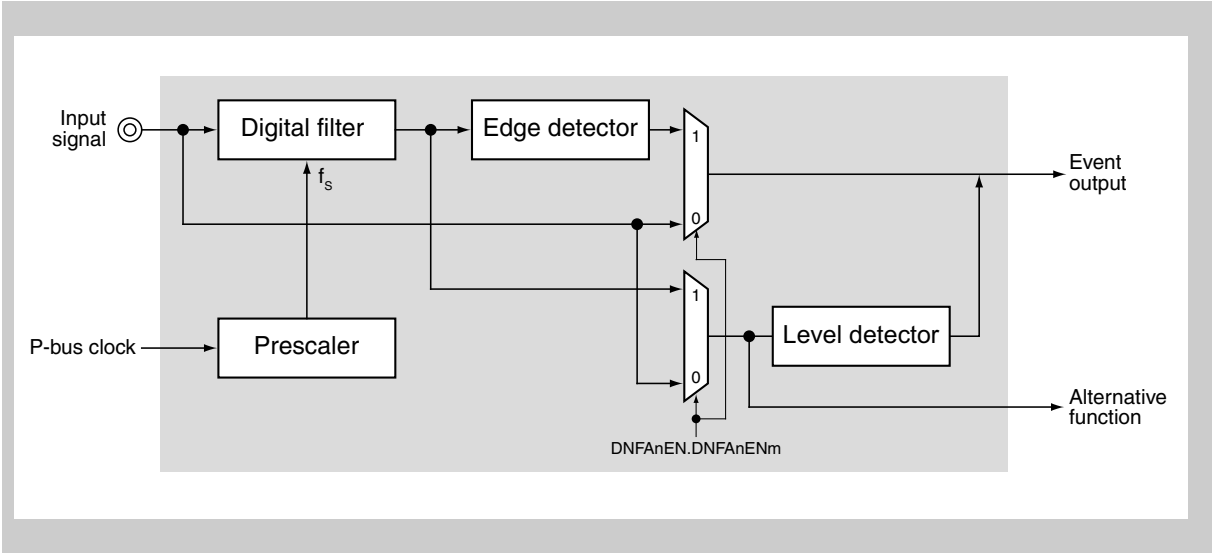


Figure 8-7 Digital filter block diagram

(a) Digital filter function

The input signal is sampled at the sampling frequency f_S.

If the level of the input signal remains unchanged for the specified number of samples, the signal level is judged to be valid, and the output signal changes accordingly.

If the level of the input signal changes before the specified number of samples have been obtained, the input signal is judged to be noise or a spike, and the output signal does not change.

Caution If noise occurs during sampling, the noise is sampled. Therefore, if noise is input during sampling and its level remains unchanged for the specified number of samples, the output signal changes to the level of the noise.

The pulse length used to judge whether an input signal is noise depends on the sampling frequency and the specified number of samples.

Either of the following parameters can be specified:

- Use DNFAnCTL.DNFAnPRS[2:0] to select the sampling frequency, using the following expression:
  $f_s = f_{PCLK} / 2^{DNFAnPRS[1:0]}$
  where f_{PCLK} is the frequency of the P-bus clock.
  (Refer to *"Clock Controller" on page xx* for details about the P-bus clock.)
- Use DNFAnCTL.DNFAnNFSTS[1:0] to specify the number of samples (from 2 to 5).

External signal pulses shorter than the following are always suppressed:

(Number of samples - 1) x $1/f_s$

External signal pulses longer than the following are always judged to be valid and passed to the filter output:

(Number of samples) x $1/f_s$

External signal pulses within the range below are sometimes suppressed, and sometimes judged to be valid:

(Number of samples -1) x $1/f_s$ to (Number of samples) x $1/f_s$

The filter operation is illustrated in the figure below with DNFAnNFSTS[1:0] = 01_B (3 samples).

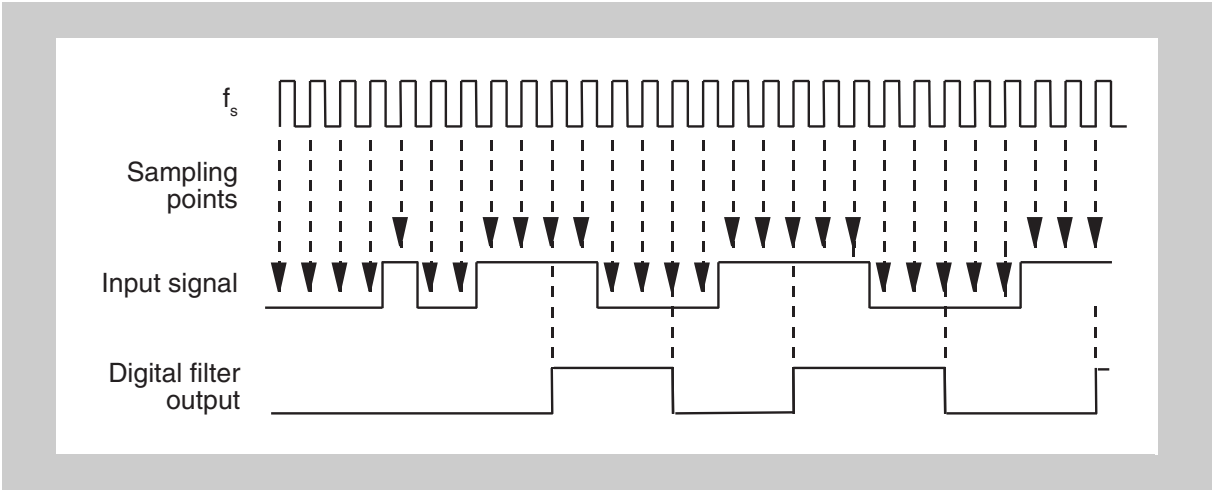


Figure 8-8 Digital filter function

Digital filter groups

Input signals with digital filters are ordered in groups of up to 16 signals. The digital filter characteristics, specified by DNFAnCTL.DNFAnPRS[2:0] and DNFAnCTL.DNFAnNFSTS[1:0], apply to the filters of the entire group. However the digital filter for each signal can be enabled and disabled separately by using DNFAnEN.DNFAnNFENm.

Cautions

1. To input the output signal of the digital filter to the alternative function pin, enable the digital filter by setting DNFAnEN.DNAFnENm, and then switch the port pin to the alternative function pin after the lapse of time below:

   DNFAnNFSTS[1:0] x $1/f_s$ + 4x $1/f_{PCLK}$

2. To use the event output signal of the digital filter as an interrupt, enable the digital filter in the interrupt prohibit state by setting DNFAnEN.DNAFnENm. Next, after the lapse of time below following the enabling of the digital filter, clear the interrupt request flag and then enable interrupts.

   DNFAnNFSTS[1:0] x $1/f_s$ + 5 x $1/f_{PCLK}$

(b) Event signals

The input signal is passed through a digital filter to eliminate noise and spikes. After passing through a digital filter, the level of the signal is used to judge whether an event has occurred, and an event output signal is generated if an event is detected.

The edge detection can be configured to detect a rising or falling edge separately by the control bits FCLAnCTLm.FCLAnINTRm and FCLAnCTLm.FCLAnINTFm respectively.

The table below summarizes the detection conditions of the digital filter.

Table 8-44 Digital filter event detection conditions

FCLAnINTFm	FCLAnINTRm	Edge detection
0	0	no detection
0	1	rising edge
1	0	falling edge
1	1	both edges

(c) Signals for alternative functions

The input signal is passed through the digital filter to eliminate noise and spikes.

Bypass All digital filter functions can be completely disabled by setting DNFAEN.DNFAnFNENm = 0. In this case the input signal is output as an event output and to the alternative function.

Note Since digital noise elimination requires the clock supply the P-bus clock to operate, digitally filtered signals can not serve as a wake-up event, if the P-bus clock is stopped during the stand-by mode.

(d) Digital filters control registers

The digital noise elimination control register DNFAnCTL and digital noise elimination enable register DNFAnEN are provided for each group of up to 16 digital filters to enable all filters to be set up in a group (n = group number).

While the filter settings specified by the DNFAnCTL register apply to the entire corresponding group, the control bit DNFAnNFENm in the DNFAnEN register enables and disables each filter separately. The register index m is in a range from 0 to 15:

The edge detection setup is done via the filter control register FCLAnCTLm.

The FCLAnCTLm registers are ordered in groups of 8 registers with the same index n. The register index m is in the range from 0 to 7:

The assignment of the input signals to the control registers and their addresses is given in *Table 8-40 “Input signals that have digital filters and their control registers 1”* and *Table 8-41 “Input signals that have digital filters and their control registers 2”* in *8.5.1 “Analog and digital filter assignments”*.

Caution Do not change any control register settings, while the concerned digital filter filter is enabled by setting DNFAnEN.DNFAnFNENm to 1. Otherwise an unintended filter output may be generated.

(3) Filter control registers

The analog and digital filters are controlled and operated by the following registers:

Table 8-45 Filter registers overview

Register Name	Shortcut	Address
Filter control register m	FCLAnCTLm	The addresses are given in the tables in *8.5.1 “Analog and digital filter assignments”*.
Digital noise elimination control register	DNFAnCTL	
Digital noise elimination enable register	DNFAnEN	

(a) FCLAnCTLm – Filter control register

This register controls the analog and digital filter operation.

Since the control options for analog and digital filters are partly different, the register description for both is given separately.

Access This register can be read/written in 8-bit or 1-bit units.

Address The assignment of the input signals to the FCLAnCTLm registers and their addresses is given in the tables in *8.5.1 "Analog and digital filter assignments" on page 306*.

Initial Value 00_H. This register is initialized by any reset.

Control for analog filters For analog filters, FCLAnCTLm provides the following control options:

7	6	5	4	3	2	1	0
FCLAn BYPSm	0	0	0	0	FCLAn INTLm	FCLAn INTFm	FCLAn INTRm
R/W	R	R	R	R	R/W	R/W	R/W

Table 8-46 FCLAnCTLm register contents for analog filters

Bit position	Bit name	Function
7	FCLAn BYPSm	Filter bypass control for alternative function 0: filter bypass disabled 1: filter bypass enabled
2	FCLAn INTLm	Level/edge detection mode selection 0: edge detection mode 1: level detection mode
1	FCLAn INTFm	• In level detection mode (FCLAnINTLm = 1): FCLAnINTFm has no effect • In edge detection mode (FCLAnINTLm = 0): falling edge detection control 0: falling edge detection disabled 1: falling edge detection enabled
0	FCLAn INTRm	• In level detection mode (FCLAnINTLm = 1): detection level selection 0: low level detection 1: high level detection • In edge detection mode (FCLAnINTLm = 0): rising edge detection control 0: rising edge detection disabled 1: rising edge detection enabled

Control for digital filters

For digital filters FCLAnCTLm provides the following control options:

7	6	5	4	3	2	1	0
0[a]	0	0	0	0	FCLAn INTLm	FCLAn INTFm	FCLAn INTRm
R	R	R	R	R	R	R/W	R/W

a) The default value “0” of these bits must not be changed.

Cautions

1. The default value “0” of bits 7 and 2 must not be changed.
2. When using an input pin that has a digital filter, be sure to enable digital noise elimination by using the DNFAnEN register. If digital noise elimination is not enabled, the operation is not guaranteed.

Table 8-47 FCLAnCTLm register contents for digital filters

Bit position	Bit name	Function
2	FCLAn INTLm	Level/edge detection mode selection 0: edge detection mode 1: level detection mode
1	FCLAn INTFm	Falling edge detection control 0: falling edge detection disabled 1: falling edge detection enabled
0	FCLAn INTRm	Rising edge detection control 0: rising edge detection disabled 1: rising edge detection enabled

Depending on a channel, some bits of the FCLAnCTLm register can be used as usual while the others must be set to their initial value. *Table 8-48 “Usable bits of the FCLAnCTLm register”* shows the details:

Table 8-48 Usable bits of the FCLAnCTLm register (1/2)

Bit position	7	6	5	4	3	2	1	0
Bit name	FCLAn BYPSm	0	0	0	0	FCLAn INTLm	FCLAn INTFm	FCLAn INTRm
FCLA11CTL0	0	0	0	0	0	0	0/1	0/1
FCLA11CTL1	0	0	0	0	0	0	0/1	0/1
FCLA11CTL2	0	0	0	0	0	0	0/1	0/1
FCLA11CTL3	0	0	0	0	0	0	0/1	0/1
FCLA11CTL4	0	0	0	0	0	0	0/1	0/1
FCLA11CTL5	0	0	0	0	0	0	0/1	0/1
FCLA12CTL0	0	0	0	0	0	0/1	0/1	0/1
FCLA12CTL1	0	0	0	0	0	0/1	0/1	0/1
FCLA12CTL2	0	0	0	0	0	0/1	0/1	0/1
FCLA12CTL3	0	0	0	0	0	0/1	0/1	0/1
FCLA12CTL4	0	0	0	0	0	0/1	0/1	0/1
FCLA12CTL5	0	0	0	0	0	0/1	0/1	0/1
FCLA12CTL6	0	0	0	0	0	0/1	0/1	0/1

Table 8-48 Usable bits of the FCLAnCTLm register (2/2)

Bit position	7	6	5	4	3	2	1	0
Bit name	FCLAn BYPSm	0	0	0	0	FCLAn INTLm	FCLAn INTFm	FCLAn INTRm
FCLA12CTL7	0	0	0	0	0	0/1	0/1	0/1
FCLA13CTL0	0	0	0	0	0	0/1	0/1	0/1
FCLA13CTL1	0	0	0	0	0	0/1	0/1	0/1
FCLA13CTL2	0	0	0	0	0	0/1	0/1	0/1
FCLA13CTL3	0	0	0	0	0	0/1	0/1	0/1
FCLA13CTL4	0	0	0	0	0	0/1	0/1	0/1
FCLA13CTL5	0	0	0	0	0	0/1	0/1	0/1
FCLA13CTL6	0	0	0	0	0	0/1	0/1	0/1
FCLA13CTL7	0	0	0	0	0	0/1	0/1	0/1
FCLA14CTL0	0	0	0	0	0	0/1	0/1	0/1
FCLA14CTL1	0	0	0	0	0	0/1	0/1	0/1
FCLA14CTL2	0	0	0	0	0	0/1	0/1	0/1
FCLA14CTL3	0	0	0	0	0	0/1	0/1	0/1
FCLA14CTL4	0	0	0	0	0	0/1	0/1	0/1
FCLA14CTL5	0	0	0	0	0	0/1	0/1	0/1
FCLA14CTL6	0	0	0	0	0	0/1	0/1	0/1
FCLA15CTL0	0	0	0	0	0	0	0/1	0/1
FCLA15CTL1	0	0	0	0	0	0	0/1	0/1
FCLA15CTL2	0	0	0	0	0	0	0/1	0/1
FCLA15CTL3	0	0	0	0	0	0	0/1	0/1
FCLA15CTL4	0	0	0	0	0	0	0/1	0/1
FCLA15CTL5	0	0	0	0	0	0	0/1	0/1
FCLA16CTL0	0	0	0	0	0	0	0/1	0/1
FCLA16CTL1	0	0	0	0	0	0	0/1	0/1
FCLA16CTL2	0	0	0	0	0	0	0/1	0/1
FCLA16CTL3	0	0	0	0	0	0	0/1	0/1

Cautions

1. Specify a value as required for bits indicated by *0/1*.
2. Be sure to clear the bits indicated by *0* in the table.

(b) DNFAnCTL – Digital noise elimination control register

This register specifies the filter characteristics of the digital noise elimination filter.

Access This register can be read/written in 8-bit units.

Address The assignment of the input signals to the DNFAnCTL registers and their addresses is given in *Table 8-40 "Input signals that have digital filters and their control registers 1"* and *Table 8-41 "Input signals that have digital filters and their control registers 2"* in *8.5.1 "Analog and digital filter assignments"*.

Initial Value 00_H. This register is initialized by any reset.

7	6	5	4	3	2	1	0
0	DNFAnNFSTS[1:0]		0	0	DNFAnPRS[2:0]		
R	R/W	R/W	R	R	R/W	R/W	R/W

Table 8-49 DNFAnCTL register contents

Bit position	Bit name	Function
6 to 5	DNFAn NFSTS[1:0]	The number of samples used to judge whether the input signal pulse is valid
2 to 0	DNFAn PRS[2:0]	Digital filter sampling clock selection

DNFAnNFSTS[1:0]	Number of samples
00_B	2 samples
01_B	3 samples
10_B	4 samples
11_B	5 samples

DNFAnPRS[2:0]	Sampling clock frequency
000_B	f_{PCLK} / 1
001_B	f_{PCLK} / 2
010_B	f_{PCLK} / 4
011_B	f_{PCLK} / 8
100_B	f_{PCLK} / 16
101_B	f_{PCLK} / 32
110_B	f_{PCLK} / 64
111_B	f_{PCLK} / 128

(c) DNFAnEN – Digital noise elimination enable register

This register enables and disables digital noise elimination for a certain input signal.

Access This register can be read/written in 16-bit units.
To read or write this register in 8-bit or 1-bit units, access DNFAnENH[7:0] and DNFAnENL[7:0], which are mirror bits of the higher byte DNFAnNFEN[15:8] and lower byte DNFAnNFEN[7:0], respectively.

Address The assignment of the input signals to DNFAnEN registers and their addresses is given in the table *"Input signals with digital filters"* in the previous section *"Analog and digital filter assignments"*.

Initial Value 0000_H. This register is initialized by any reset.

15	14	13	12	11	10	9	8
DNFAn NFEN15	DNFAn NFEN14	DNFAn NFEN13	DNFAn NFEN12	DNFAn NFEN11	DNFAn NFEN10	DNFAn NFEN9	DNFAn NFEN8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7	6	5	4	3	2	1	0
DNFAn NFEN7	DNFAn NFEN6	DNFAn NFEN5	DNFAn NFEN4	DNFAn NFEN3	DNFAn NFEN2	DNFAn NFEN1	DNFAn NFEN0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 8-50 DNFAnEN register contents

Bit position	Bit name	Function
15 to 0	DNFAn NFEN[15:0]	Digital noise elimination control 0: digital noise elimination disabled 1: digital noise elimination enabled

Depending on a channel, some bits of the DNFAnEN register can be used as usual while the others are disabled even if set. *Table 8-51 "Usable bits of the DNFAnEN register"* shows the details:

Table 8-51 Usable bits of the DNFAnEN register (1/2)

8-bit register name	DNFAnENH								DNFAnENL							
Bit position	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	DNFAnNFEN15	DNFAnNFEN14	DNFAnNFEN13	DNFAnNFEN12	DNFAnNFEN11	DNFAnNFEN10	DNFAnNFEN9	DNFAnNFEN8	DNFAnNFEN7	DNFAnNFEN6	DNFAnNFEN5	DNFAnNFEN4	DNFAnNFEN3	DNFAnNFEN2	DNFAnNFEN1	DNFAnNFEN0
DNFA0EN	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
DNFA2EN	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
DNFA4EN	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
DNFA6EN	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
DNFA8EN	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1

Table 8-51 Usable bits of the DNFAnEN register (2/2)

8-bit register name	DNFAnENH								DNFAnENL							
Bit position	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit name	DNFAnNFEN15	DNFAnNFEN14	DNFAnNFEN13	DNFAnNFEN12	DNFAnNFEN11	DNFAnNFEN10	DNFAnNFEN9	DNFAnNFEN8	DNFAnNFEN7	DNFAnNFEN6	DNFAnNFEN5	DNFAnNFEN4	DNFAnNFEN3	DNFAnNFEN2	DNFAnNFEN1	DNFAnNFEN0
DNFA9EN	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
DNFA10EN	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
DNFA11EN	0	0	0	0	0	0	0	0	0	0	1/0	1/0	1/0	1/0	1/0	1/0
DNFA12EN	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
DNFA13EN	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
DNFA14EN	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1

Cautions

1. Specify a value as required for bits indicated by *0/1*.
2. Be sure to set the bits indicated by *1* in the table.
3. Bits indicated by *0* in the table are fixed to 0 by hardware. Even if such a bit is set, therefore, it is ignored.

Chapter 9 Interrupt Functions

9.1 Features

The phenomenon of forcing a branch operation from a currently running program to another program, due to a specific cause, is called an exception. This microcontroller supports the following types of exceptions.

For details about interrupt functions, see the *V850E2 Architecture User's Manual*.

Table 9-1 Exception cause list

Exception name	Symbol	Cause group	Priority group	Exception level
CPU initialization	RESET	Reset input	P1	–
FE level non-maskable interrupt	FENMI	FENMI input	P2	FE
System error exception	SYSERR	SYSERR input (4 causes)	P3	FE
Peripheral device protection exception	PPI	Peripheral device protection violation	P4	FE
Timing monitoring exception	TSI	Timing monitoring violation	P5	FE
FE level maskable interrupt	FEINT	FEINT input	P6	FE
Floating-point operation exception (imprecise)	FPI	FPU instruction	P7	EI
EI level maskable interrupt	EIINT	Maskable interrupt input	P8	EI
Execution protection exception	MIP	Execution protection violation	P9	FE
Memory error exception	MEP	Instruction access error input	P10	FE
Data protection exception	MDP	Data protection violation	P11	FE
Floating-point operation exception (precise)	FPP	FPU instruction		EI
Coprocessor unusable exception	UCPOP	Coprocessor instruction		FE
Reserved instruction exception	RIEX	Reserved instruction		FE
FE level software exception	FETRAPEX	FETRAP instruction (vector = 1_H to F_H)		FE
EI level software exception	EITRAP0	TRAP0n instruction (vector = 00_H to $0F_H$)		EI
EI level software exception	EITRAP1	TRAP1n instruction (vector = 10_H to $1F_H$)		EI
System call exception	SYSCALLEX	SYSCALL instruction (vector = 00_H to FF_H)		EI

Priority order Priority group P1 has the highest, P11 the lowest priority.

Interrupts The following three types of exceptions in *Table 9-1 "Exception cause list"* are called interrupts.

- FE level non-maskable interrupt (FENMI)
  - FENMI is served immediately, even if another FE level interrupt - FENMI or FEINT - is in service (CPU system register PSW.NP = 1).
  - Resume disabled, recover disabled (for error report)
- FE level maskable interrupt (FEINT)
  - FEINT is served immediately, if no other FE level interrupt - FENMI or FEINT - is in service (CPU system register PSW.NP = 0). Thus PSW.NP = 1 would mask an FEINT.
  - Resume enabled, recover enabled
  - Highest priority interrupt (except FENMI)
- EI level maskable interrupt (EIINT)
  - EIINT may only be served, if no FE level interrupt - FENMI or FEINT - is in service (CPU system register PSW.NP = 0).
  - Resume enabled, recover enabled
  - Interrupt masking can be specified by each interrupt channel
  - 16 interrupt priority levels can be specified for each interrupt channel.
  - This chapter describes EIINT corresponding to interrupt channel n into EIINTn.

Resume: Indicates whether execution restart from the last position at which program execution was interrupted is possible.

Recover: Indicates whether recovery to the processor status (status of processor resources including general-purpose registers and system registers) at the time of program execution interruption is possible.

The following table shows the number of valid interrupts in the V850E2/MN4.

Table 9-2 Number of valid interrupts

Interrupt cause	μPD70F3510	μPD70F3512	μPD70F3514 μPD70F3515
FE level non-maskable interrupt (FENMI)	1 (including one external interrupt)	1 (including one external interrupt)	1 (including one external interrupt)
FE level non-maskable interrupt (FEINT)	1	1	2
EI level non-maskable interrupt (EIINT)	207 (including 28 external interrupts)	217 (including 28 external interrupts)	222 (including 28 external interrupts)

These interrupt causes are described below.

9.2 Interrupt Causes

9.2.1 FE level non-maskable interrupts

Priority group The FE level non-maskable interrupts have the priority P2.

Return PC An FE non-maskable interrupt does not allow to resume or recover.

Control register FE level NMI control register

See *9.3.8 "FNC: FE level NMI control register"*.

Return instruction The return instruction from FE level non-maskable interrupt service routines is FERET.

Table 9-3 FE level non-maskable interrupt requests

Interrupt			Interrupt request		Unit	Priority group	Exception case code	Handler address 0000...
Symbol	Control register		Name	Cause				
	Name	Address FFFF...						
FENMI	FNC	$645C_H$	WDTA0NMI	Watchdog Timer 0 error NMI interrupt	WDTA0	P2	00020_H	0020_H
			WDTA1NMI	Watchdog Timer 1 error NMI interrupt	WDTA1			

9.2.2 FE level maskable interrupts

Priority group The FE level maskable interrupts have the priority P6.

Return PC The program counter (PC) value, set after returning from any interrupt service routine by the RETI instruction is always the next address.

Control register FE level maskable interrupt control register

See *9.3.9 "FIC: FE level INT control register"*.

Return instruction The return instruction from FE level maskable interrupt service routines is FERET.

Table 9-4 FE level maskable interrupt requests

Interrupt			Interrupt request		Unit	Priority group	Exception cause code	Handler address 0000...
Symbol	Control register		Name	Cause				
	Name	Address FFFF...						
FEINT	FIC	$645E_H$	NMI0	NMI	Pin	P6	00001_H	0010_H

9.2.3 EI level maskable interrupts

Interrupt naming

The composition of the interrupt request signal names, their assigned interrupt control registers and the bits in these registers follow special rules.

In the following the name of the interrupt request is represented by *<name>*.

- Interrupt request name: **INT***<name>*
  The prefix "**INT**" is put in front of *<name>*.
- Interrupt request control register: **IC***<name>*
  The prefix "**IC**" is put in front of *<name>*.
  The 16 bit of the 16-bit register **IC***<name>* can also be accessed byte-wise with the following names:
  - low byte (bits[7:0]): **IC***<name>***L** at the address of **IC***<name>*
    The suffix "**L**" is appended to the register name **IC***<name>*.
  - high byte (bits[15:8]): **IC***<name>***H** at the address of **IC***<name>* + 1
    The suffix "**H**" is appended to the register name **IC***<name>*.
- Interrupt control register bit names: **RF***<name>*, **MK***<name>*, **P3***<name>*, **P2***<name>*, **P1***<name>*, **P0***<name>*
  The bit prefix "**RF**", "**MK**","**P3**","**P2**","**P1**","**P0**" is appended to the interrupt *<name>*.
- Each interrupt request is assigned to a certain interrupt channel number n = 0 to 255.
  The functional description of the Interrupt Controller in this chapter refers to general names of the interrupt requests and their control registers.
  If **INT***<name>* is assigned to the interrupt channel number *n*, throughout this chapter
  - the interrupt request is named **EIINT***n*
  - the assigned interrupt control register is named **EIC***n*
  - the interrupt control bits are named **EIRF***n*, **EIMK***n*, **EIP3***n*, **EIP2***n*, **EIP1***n*, **EIP0***n*.

Example

The interrupt request of the second TAUA0 channel (*<name>* = *TAUA0I2*) is named

INT*TAUA0I2*

The related interrupt control registers are

IC*TAUA0I2*, **IC***TAUA0I2L*, **IC***TAUA0I2H*

The bits in this register are

RF*TAUA0I2*, **MK***TAUA0I2*, **P3***TAUA0I2*, **P2***TAUA0I2*, **P1***TAUA0I2*, **P0***TAUA0I2*

The interrupt channel for **INT***TAUA0I2* is n = *53* (refer to the table below). Thus in the functional description of the Interrupt Controller this interrupt is referred to as

EIINT*53*,

the related control register as

EIC*53,*

and their control register bit names

EIRF*53*, **EIMK***53*, **EIP3***53*, **EIP2***53*, **EIP1***53*, **EIP0***53*.

The following tables list the references between the interrupt channel number *n*, the assigned V850E2/MN4 interrupt requests and control register names.

Priority group The EI level maskable interrupts have the priority P8.

Return PC The program counter (PC) value, set after returning from any interrupt service routine by the EIRET instruction is always the next address.

Control register FE level maskable interrupt control register

See *9.3.1 "EICn (n = 0 to 255): EI level interrupt control registers"*.

Return instruction EIRET instruction

Table 9-5 EI level maskable interrupt requests (1/14)

Interrupt			Interrupt request			Default priority	Exception cause code	Handler Address 0000...	Target product		
Channel	Control register		Name	Cause	Unit				μPD70F3510	μPD70F3512	μPD70F3514,70F3515
	Name	Address FFFF...									
0	EIC0	6000_H	–	Inter-CPU interrupt 0	CPU	1	0080_H	0080_H	×	×	○
1	EIC1	6002_H	–	Inter-CPU interrupt 1	CPU	2	0090_H	0090_H	×	×	○
2	EIC2	6004_H	–	PE guard error	CPU	3	$00A0_H$	$00A0_H$	×	×	○
3	Reserved	6006_H	Reserved			4	$00B0_H$	$00B0_H$	×	×	×
4	Reserved	6008_H	Reserved			5	$00C0_H$	$00C0_H$	×	×	×
5	Reserved	$600A_H$	Reserved			6	$00D0_H$	$00D0_H$	×	×	×
6	Reserved	$600C_H$	Reserved			7	$00E0_H$	$00E0_H$	×	×	×
7	Reserved	$600E_H$	Reserved			8	$00F0_H$	$00F0_H$	×	×	×
8	Reserved	6010_H	Reserved			9	0100_H	0100_H	×	×	×
9	Reserved	6012_H	Reserved			10	0110_H	0110_H	×	×	×
10	ICWDTA0	6014_H	INTWDTA0	WDTA0 interval timer	WDTA0	11	0120_H	0120_H	○	○	○
11	ICWDTA1	6016_H	INTWDTA1	WDTA1 interval timer	WDTA1	12	0130_H	0130_H	×	×	○
12	ICOSTM0	6018_H	INTOSTM0	OS timer underflow (OSTM0TIT)	OSTM	13	0140_H	0140_H	○	○	○
13	ICOSTM1	$601A_H$	INTOSTM1	OS timer underflow (OSTM1TIT)	OSTM	14	0150_H	0150_H	×	×	○
14	ICP0	$601C_H$	INTP0	INTP0 pin valid edge input/ INTDTS0	Pin/ DTS	15	0160_H	0160_H	○	○	○
15	ICP1	$601E_H$	INTP1	INTP1 pin valid edge input/ INTDTS1	Pin/ DTS	16	0170_H	0170_H	○	○	○
16	ICP2	6020_H	INTP2	INTP2 pin valid edge input/ INTDTS2	Pin/ DTS	17	0180_H	0180_H	○	○	○
17	ICP3	6022_H	INTP3	INTP3 pin valid edge input/ INTDTS3	Pin/ DTS	18	0190_H	0190_H	○	○	○
18	ICP4	6024_H	INTP4	INTP4 pin valid edge input/ INTDTS4	Pin/ DTS	19	$01A0_H$	$01A0_H$	○	○	○
19	ICP5	6026_H	INTP5	INTP5 pin valid edge input/ INTDTS5	Pin/ DTS	20	$01B0_H$	$01B0_H$	○	○	○
20	ICP6	6028_H	INTP6	INTP6 pin valid edge input/ INTDTS6	Pin/ DTS	21	$01C0_H$	$01C0_H$	○	○	○

Table 9-5 EI level maskable interrupt requests (2/14)

Interrupt			Interrupt request			Default priority	Exception cause code	Handler Address 0000...	Target product		
Channel	Control register		Name	Cause	Unit				μPD70F3510	μPD70F3512	μPD70F3514,70F3515
	Name	Address FFFF...									
21	ICP7	$602A_H$	INTP7	INTP7 pin valid edge input/ INTDTS7	Pin/ DTS	22	$01D0_H$	$01D0_H$	○	○	○
22	ICP8	$602C_H$	INTP8	INTP8 pin valid edge input/ INTDTS8	Pin/ DTS	23	$01E0_H$	$01E0_H$	○	○	○
23	ICP9	$602E_H$	INTP9	INTP9 pin valid edge input/ INTDTS9	Pin/ DTS	24	$01F0_H$	$01F0_H$	○	○	○
24	ICP10	6030_H	INTP10	INTP10 pin valid edge input/ INTDTS10	Pin/ DTS	25	0200_H	0200_H	○	○	○
25	ICP11	6032_H	INTP11	INTP11 pin valid edge input/ INTDTS11	Pin/ DTS	26	0210_H	0210_H	○	○	○
26	ICP12	6034_H	INTP12	INTP12 pin valid edge input/ INTDTS12	Pin/ DTS	27	0220_H	0220_H	○	○	○
27	ICP13	6036_H	INTP13	INTP13 pin valid edge input/ INTDTS13	Pin/ DTS	28	0230_H	0230_H	○	○	○
28	ICP14	6038_H	INTP14	INTP14 pin valid edge input/ INTDTS14	Pin/ DTS	29	0240_H	0240_H	○	○	○
29	ICP15	$603A_H$	INTP15	INTP15 pin valid edge input/ INTDTS15	Pin/ DTS	30	0250_H	0250_H	○	○	○
30	ICP16	$603C_H$	INTP16	INTP15 pin valid edge input/ INTDTS16	Pin/ DTS	31	0260_H	0260_H	○	○	○
31	ICP17	$603E_H$	INTP17	INTP17 pin valid edge input/ INTDTS17	Pin/ DTS	32	0270_H	0270_H	○	○	○
32	ICP18	6040_H	INTP18	INTP18 pin valid edge input/ INTDTS18	Pin/ DTS	33	0280_H	0280_H	○	○	○
33	ICP19	6042_H	INTP19	INTP19 pin valid edge input/ INTDTS19	Pin/ DTS	34	0290_H	0290_H	○	○	○
34	ICP20	6044_H	INTP20	INTP20 pin valid edge input/ INTDTS20	Pin/ DTS	35	$02A0_H$	$02A0_H$	○	○	○
35	ICP21	6046_H	INTP21	INTP21 pin valid edge input/ INTDTS21	Pin/ DTS	36	$02B0_H$	$02B0_H$	○	○	○
36	ICP22	6048_H	INTP22	INTP22 pin valid edge input/ INTDTS22	Pin/ DTS	37	$02C0_H$	$02C0_H$	○	○	○
37	ICP23	$604A_H$	INTP23	INTP23 pin valid edge input/ INTDTS23	Pin/ DTS	38	$02D0_H$	$02D0_H$	○	○	○
38	ICP24	$604C_H$	INTP24	INTP24 pin valid edge input/ INTDTS24	Pin/ DTS	39	$02E0_H$	$02E0_H$	○	○	○
39	ICP25	$604E_H$	INTP25	INTP25 pin valid edge input/ INTDTS25	Pin/ DTS	40	$02F0_H$	$02F0_H$	○	○	○
40	ICP26	6050_H	INTP26	INTP25 pin valid edge input/ INTDTS26	Pin/ DTS	41	0300_H	0300_H	○	○	○

Table 9-5 EI level maskable interrupt requests (3/14)

Interrupt			Interrupt request			Default priority	Exception cause code	Handler Address 0000...	Target product		
Channel	Control register		Name	Cause	Unit				μPD70F3510	μPD70F3512	μPD70F3514,70F3515
	Name	Address FFFF...									
41	ICP27	6052_H	INTP27	INTP27 pin valid edge input/ INTDTS27	Pin/ DTS	42	0310_H	0310_H	○	○	○
42	ICESO0	6054_H	INTESO0	ESO0 pin valid edge input	Pin	43	0320_H	0320_H	○	○	○
43	ICESO1	6056_H	INTESO1	ESO1 pin valid edge input	Pin	44	0330_H	0330_H	○	○	○
44	ICESO2	6058_H	INTESO2	ESO2 pin valid edge input	Pin	45	0340_H	0340_H	○	○	○
45	ICESO3	$605A_H$	INTESO3	ESO3 pin valid edge input	Pin	46	0350_H	0350_H	○	○	○
46	ICADCA0ERR	$605C_H$	INTADCA0ERR	A/D conversion result error	ADCA0	47	0360_H	0360_H	○	○	○
47	ICADCA0I0	$605E_H$	INTADCA0I0	A/D conversion completion (group 0)/INTDTS28	ADCA0 /DTS	48	0370_H	0370_H	○	○	○
48	ICADCA0I1	6060_H	INTADCA0I1	A/D conversion completion (group 1)/INTDTS29	ADCA0 /DTS	49	0380_H	0380_H	○	○	○
49	ICADCA0I2	6062_H	INTADCA0I2	A/D conversion completion (group 2)/INTDTS30	ADCA0 /DTS	50	0390_H	0390_H	○	○	○
50	Reserved	6064_H	Reserved			51	$03A0_H$	$03A0_H$	×	×	×
51	ICTAUA0I0	6066_H	INTTAUA0I0	TAUA0 channel 0 function dependent/INTDTS32	TAUA0/ DTS	52	$03B0_H$	$03B0_H$	○	○	○
52	ICTAUA0I1	6068_H	INTTAUA0I1	TAUA0 channel 1 function dependent/INTDTS33	TAUA0/ DTS	53	$03C0_H$	$03C0_H$	○	○	○
53	ICTAUA0I2	$606A_H$	INTTAUA0I2	TAUA0 channel 2 function dependent/INTDTS34	TAUA0/ DTS	54	$03D0_H$	$03D0_H$	○	○	○
54	ICTAUA0I3	$606C_H$	INTTAUA0I3	TAUA0 channel 3 function dependent/INTDTS35	TAUA0/ DTS	55	$03E0_H$	$03E0_H$	○	○	○
55	ICTAUA0I4	$606E_H$	INTTAUA0I4	TAUA0 channel 4 function dependent/INTDTS36	TAUA0/ DTS	56	$03F0_H$	$03F0_H$	○	○	○
56	ICTAUA0I5	6070_H	INTTAUA0I5	TAUA0 channel 5 function dependent/INTDTS37	TAUA0/ DTS	57	0400_H	0400_H	○	○	○
57	ICTAUA0I6	6072_H	INTTAUA0I6	TAUA0 channel 6 function dependent/INTDTS38	TAUA0/ DTS	58	0410_H	0410_H	○	○	○
58	ICTAUA0I7	6074_H	INTTAUA0I7	TAUA0 channel 7 function dependent/INTDTS39	TAUA0/ DTS	59	0420_H	0420_H	○	○	○
59	ICTAUA0I8	6076_H	INTTAUA0I8	TAUA0 channel 8 function dependent/INTDTS40	TAUA0/ DTS	60	0430_H	0430_H	○	○	○
60	ICTAUA0I9	6078_H	INTTAUA0I9	TAUA0 channel 9 function dependent/INTDTS41	TAUA0/ DTS	61	0440_H	0440_H	○	○	○
61	ICTAUA0I10	$607A_H$	INTTAUA0I10	TAUA0 channel 10 function dependent/INTDTS42	TAUA0/ DTS	62	0450_H	0450_H	○	○	○
62	ICTAUA0I11	$607C_H$	INTTAUA0I11	TAUA0 channel 11 function dependent/INTDTS43	TAUA0/ DTS	63	0460_H	0460_H	○	○	○
63	ICTAUA0I12	$607E_H$	INTTAUA0I12	TAUA0 channel 12 function dependent/INTDTS44	TAUA0/ DTS	64	0470_H	0470_H	○	○	○

Table 9-5 EI level maskable interrupt requests (4/14)

Interrupt			Interrupt request			Default priority	Exception cause code	Handler Address 0000...	Target product		
Channel	Control register		Name	Cause	Unit				μPD70F3510	μPD70F3512	μPD70F3514,70F3515
	Name	Address FFFF...									
64	ICTAUA0I13	6080_H	INTTAUA0I13	TAUA0 channel 13 function dependent/INTDTS45	TAUA0/ DTS	65	0480_H	0480_H	○	○	○
65	ICTAUA0I14	6082_H	INTTAUA0I14	TAUA0 channel 14 function dependent/INTDTS46	TAUA0/ DTS	66	0490_H	0490_H	○	○	○
66	ICTAUA0I15	6084_H	INTTAUA0I15	TAUA0 channel 15 function dependent/INTDTS47	TAUA0/ DTS	67	$04A0_H$	$04A0_H$	○	○	○
67	ICTAUA1I0	6086_H	INTTAUA1I0	TAUA1 channel 0 function dependent/INTDTS48	TAUA1/ DTS	68	$04B0_H$	$04B0_H$	○	○	○
68	ICTAUA1I1	6088_H	INTTAUA1I1	TAUA1 channel 1 function dependent/INTDTS49	TAUA1/ DTS	69	$04C0_H$	$04C0_H$	○	○	○
69	ICTAUA1I2	$608A_H$	INTTAUA1I2	TAUA1 channel 2 function dependent/INTDTS50	TAUA1/ DTS	70	$04D0_H$	$04D0_H$	○	○	○
70	ICTAUA1I3	$608C_H$	INTTAUA1I3	TAUA1 channel 3 function dependent/INTDTS51	TAUA1/ DTS	71	$04E0_H$	$04E0_H$	○	○	○
71	ICTAUA1I4	$608E_H$	INTTAUA1I4	TAUA1 channel 4 function dependent/INTDTS52	TAUA1/ DTS	72	$04F0_H$	$04F0_H$	○	○	○
72	ICTAUA1I5	6090_H	INTTAUA1I5	TAUA1 channel 5 function dependent/INTDTS53	TAUA1/ DTS	73	0500_H	0500_H	○	○	○
73	ICTAUA1I6	6092_H	INTTAUA1I6	TAUA1 channel 6 function dependent/INTDTS54	TAUA1/ DTS	74	0510_H	0510_H	○	○	○
74	ICTAUA1I7	6094_H	INTTAUA1I7	TAUA1 channel 7 function dependent/INTDTS55	TAUA1/ DTS	75	0520_H	0520_H	○	○	○
75	ICTAUA1I8	6096_H	INTTAUA1I8	TAUA1 channel 8 function dependent/INTDTS56	TAUA1/ DTS	76	0530_H	0530_H	○	○	○
76	ICTAUA1I9	6098_H	INTTAUA1I9	TAUA1 channel 9 function dependent/INTDTS57	TAUA1/ DTS	77	0540_H	0540_H	○	○	○
77	ICTAUA1I10	$609A_H$	INTTAUA1I10	TAUA1 channel 10 function dependent/INTDTS58	TAUA1/ DTS	78	0550_H	0550_H	○	○	○
78	ICTAUA1I11	$609C_H$	INTTAUA1I11	TAUA1 channel 11 function dependent/INTDTS59	TAUA1/ DTS	79	0560_H	0560_H	○	○	○
79	ICTAUA1I12	$609E_H$	INTTAUA1I12	TAUA1 channel 12 function dependent/INTDTS60	TAUA1/ DTS	80	0570_H	0570_H	○	○	○
80	ICTAUA1I13	$60A0_H$	INTTAUA1I13	TAUA1 channel 13 function dependent/INTDTS61	TAUA1/ DTS	81	0580_H	0580_H	○	○	○
81	ICTAUA1I14	$60A2_H$	INTTAUA1I14	TAUA1 channel 14 function dependent/INTDTS62	TAUA1/ DTS	82	0590_H	0590_H	○	○	○
82	ICTAUA1I15	$60A4_H$	INTTAUA1I15	TAUA1 channel 15 function dependent/INTDTS63	TAUA1/ DTS	83	$05A0_H$	$05A0_H$	○	○	○
83	ICTAUA2I0	$60A6_H$	INTTAUA2I0	TAUA2 channel 0 function dependent	TAUA2	84	$05B0_H$	$05B0_H$	○	○	○

Table 9-5 EI level maskable interrupt requests (5/14)

Interrupt			Interrupt request			Default priority	Exception cause code	Handler Address 0000...	Target product		
Channel	Control register		Name	Cause	Unit				μPD70F3510	μPD70F3512	μPD70F3514,70F3515
	Name	Address FFFF...									
84	ICTAUA2I1	$60A8_H$	INTTAUA2I1	TAUA2 channel 1 function dependent	TAUA2	85	$05C0_H$	$05C0_H$	○	○	○
85	ICTAUA2I2	$60AA_H$	INTTAUA2I2	TAUA2 channel 2 function dependent	TAUA2	86	$05D0_H$	$05D0_H$	○	○	○
86	ICTAUA2I3	$60AC_H$	INTTAUA2I3	TAUA2 channel 3 function dependent	TAUA2	87	$05E0_H$	$05E0_H$	○	○	○
87	ICTAUA2I4	$60AE_H$	INTTAUA2I4	TAUA2 channel 4 function dependent	TAUA2	88	$05F0_H$	$05F0_H$	○	○	○
88	ICTAUA2I5	$60B0_H$	INTTAUA2I5	TAUA2 channel 5 function dependent	TAUA2	89	0600_H	0600_H	○	○	○
89	ICTAUA2I6	$60B2_H$	INTTAUA2I6	TAUA2 channel 6 function dependent	TAUA2	90	0610_H	0610_H	○	○	○
90	ICTAUA2I7	$60B4_H$	INTTAUA2I7	TAUA2 channel 7 function dependent	TAUA2	91	0620_H	0620_H	○	○	○
91	ICTAUA2I8	$60B6_H$	INTTAUA2I8	TAUA2 channel 8 function dependent	TAUA2	92	0630_H	0630_H	○	○	○
92	ICTAUA2I9	$60B8_H$	INTTAUA2I9	TAUA2 channel 9 function dependent	TAUA2	93	0640_H	0640_H	○	○	○
93	ICTAUA2I10	$60BA_H$	INTTAUA2I10	TAUA2 channel 10 function dependent	TAUA2	94	0650_H	0650_H	○	○	○
94	ICTAUA2I11	$60BC_H$	INTTAUA2I11	TAUA2 channel 11 function dependent	TAUA2	95	0660_H	0660_H	○	○	○
95	ICTAUA2I12	$60BE_H$	INTTAUA2I12	TAUA2 channel 12 function dependent/INTDTS64	TAUA2/ DTS	96	0670_H	0670_H	○	○	○
96	ICTAUA2I13	$60C0_H$	INTTAUA2I13	TAUA2 channel 13 function dependent/INTDTS65	TAUA2/ DTS	97	0680_H	0680_H	○	○	○
97	ICTAUA2I14	$60C2_H$	INTTAUA2I14	TAUA2 channel 14 function dependent/INTDTS66	TAUA2/ DTS	98	0690_H	0690_H	○	○	○
98	ICTAUA2I15	$60C4_H$	INTTAUA2I15	TAUA2 channel 15 function dependent/INTDTS67	TAUA2/ DTS	99	$06A0_H$	$06A0_H$	○	○	○
99	ICTAUA3I0	$60C6_H$	INTTAUA3I0	TAUA3 channel 0 function dependent/INTDTS68	TAUA3/ DTS	100	$06B0_H$	$06B0_H$	○	○	○
100	ICTAUA3I1	$60C8_H$	INTTAUA3I1	TAUA3 channel 1 function dependent/INTDTS69	TAUA3/ DTS	101	$06C0_H$	$06C0_H$	○	○	○
101	ICTAUA3I2	$60CA_H$	INTTAUA3I2	TAUA3 channel 2 function dependent/INTDTS70	TAUA3/ DTS	102	$06D0_H$	$06D0_H$	○	○	○
102	ICTAUA3I3	$60CC_H$	INTTAUA3I3	TAUA3 channel 3 function dependent/INTDTS71	TAUA3/ DTS	103	$06E0_H$	$06E0_H$	○	○	○
103	ICTAUA3I4	$60CE_H$	INTTAUA3I4	TAUA3 channel 4 function dependent/INTDTS72	TAUA3/ DTS	104	$06F0_H$	$06F0_H$	○	○	○

Table 9-5 EI level maskable interrupt requests (6/14)

Interrupt			Interrupt request			Default priority	Exception cause code	Handler Address 0000...	Target product		
Channel	Control register		Name	Cause	Unit				μPD70F3510	μPD70F3512	μPD70F3514,70F3515
	Name	Address FFFF...									
104	ICTAUA3I5	$60D0_H$	INTTAUA3I5	TAUA3 channel 5 function dependent/INTDTS73	TAUA3/ DTS	105	0700_H	0700_H	○	○	○
105	ICTAUA3I6	$60D2_H$	INTTAUA3I6	TAUA3 channel 6 function dependent/INTDTS74	TAUA3/ DTS	106	0710_H	0710_H	○	○	○
106	ICTAUA3I7	$60D4_H$	INTTAUA3I7	TAUA3 channel 7 function dependent/INTDTS75	TAUA3/ DTS	107	0720_H	0720_H	○	○	○
107	ICTAUA3I8	$60D6_H$	INTTAUA3I8	TAUA3 channel 8 function dependent/INTDTS76	TAUA3/ DTS	108	0730_H	0730_H	○	○	○
108	ICTAUA3I9	$60D8_H$	INTTAUA3I9	TAUA3 channel 9 function dependent/INTDTS77	TAUA3/ DTS	109	0740_H	0740_H	○	○	○
109	ICTAUA3I10	$60DA_H$	INTTAUA3I10	TAUA3 channel 10 function dependent/INTDTS78	TAUA3/ DTS	110	0750_H	0750_H	○	○	○
110	ICTAUA3I11	$60DC_H$	INTTAUA3I11	TAUA3 channel 11 function dependent/INTDTS79	TAUA3/ DTS	111	0760_H	0760_H	○	○	○
111	ICTAUA3I12	$60DE_H$	INTTAUA3I12	TAUA3 channel 12 function dependent/INTDTS80	TAUA3/ DTS	112	0770_H	0770_H	○	○	○
112	ICTAUA3I13	$60E0_H$	INTTAUA3I13	TAUA3 channel 13 function dependent/INTDTS81	TAUA3/ DTS	113	0780_H	0780_H	○	○	○
113	ICTAUA3I14	$60E2_H$	INTTAUA3I14	TAUA3 channel 14 function dependent/INTDTS82	TAUA3/ DTS	114	0790_H	0790_H	○	○	○
114	ICTAUA3I15	$60E4_H$	INTTAUA3I15	TAUA3 channel 15 function dependent/INTDTS83	TAUA3/ DTS	115	$07A0_H$	$07A0_H$	○	○	○
115	ICTAUJ0I0	$60E6_H$	INTTAUJ0I0	TAUAJ channel 0 function dependent/INTDTS84	TAUJ0/ DTS	116	$07B0_H$	$07B0_H$	○	○	○
116	ICTAUJ0I1	$60E8_H$	INTTAUJ0I1	TAUAJ channel 1 function dependent/INTDTS85	TAUJ0/ DTS	117	$07C0_H$	$07C0_H$	○	○	○
117	ICTAUJ0I2	$60EA_H$	INTTAUJ0I2	TAUAJ channel 2 function dependent/INTDTS86	TAUJ0/ DTS	118	$07D0_H$	$07D0_H$	○	○	○
118	ICTAUJ0I3	$60EC_H$	INTTAUJ0I3	TAUAJ channel 3 function dependent/INTDTS87	TAUJ0/ DTS	119	$07E0_H$	$07E0_H$	○	○	○
119	ICENCA0IOV	$60EE_H$	INTENCA0IOV	TMENC0 overflow/INTDTS88	ENCA0 /DTS	120	$07F0_H$	$07F0_H$	○	○	○
120	ICENCA0IUD	$60F0_H$	INTENCA0IUD	TMENC0 underflow/INTDTS89	ENCA0 /DTS	121	0800_H	0800_H	○	○	○
121	ICENCA0I0	$60F2_H$	INTENCA0I0	TMENC0 compare 0 match/ capture 0 input/INTDTS90	ENCA0 /DTS	122	00081_H	0810_H	○	○	○
122	ICENCA0I1	$60F4_H$	INTENCA0I1	TMENC0 compare 1 match/ capture 1input/INTDTS91	ENCA0 /DTS	123	0820_H	0820_H	○	○	○
123	ICENCA0IEC	$60F6_H$	INTENCA0IEC	ENCA0 clear by encoder input (Z phase)/INTDTS92	ENCA0 /DTS	124	0830_H	0830_H	○	○	○

Table 9-5 EI level maskable interrupt requests (7/14)

Interrupt			Interrupt request			Default priority	Exception cause code	Handler Address 0000...	Target product		
Channel	Control register		Name	Cause	Unit				μPD70F3510	μPD70F3512	μPD70F3514,70F3515
	Name	Address FFFF...									
124	ICENCA1IOV	$60F8_H$	INTENCA1IOV	TMENC1 overflow/INTDTS93	ENCA1/DTS	125	0840_H	0840_H	○	○	○
125	ICENCA1IUD	$60FA_H$	INTENCA1IUD	TMENC1 underflow/INTDTS94	ENCA1/DTS	126	0850_H	0850_H	○	○	○
126	ICENCA1I0	$60FC_H$	INTENCA1I0	TMENC1 compare 0 match/ capture 0 input/INTDTS95	ENCA1/DTS	127	0860_H	0860_H	○	○	○
127	ICENCA1I1	$60FE_H$	INTENCA1I1	TMENC1 compare 1 match/ capture 1 input/ INTDTS96	ENCA1/DTS	128	0870_H	0870_H	○	○	○
128	ICENCA1IEC	6100_H	INTENCA1IEC	ENCA1 clear by encoder input (Z phase)/INTDTS97	ENCA1/DTS	129	0880_H	0880_H	○	○	○
129	Reserved	6102_H	Reserved			130	0890_H	0890_H	×	×	×
130	Reserved	6104_H	Reserved			131	$08A0_H$	$08A0_H$	×	×	×
131	ICTAPA0IPEK0	6106_H	INTTAPA0IPEK0	TAUA0 peak interrupt/ INTDTS98	TAPA0/DTS	132	$08B0_H$	$08B0_H$	○	○	○
132	ICTAPA0IVLY0	6108_H	INTTAPA0IVLY0	TAUA0 valley interrupt/ INTDTS99	TAPA0/DTS	133	$08C0_H$	$08C0_H$	○	○	○
133	Reserved	$610A_H$	Reserved			134	$08D0_H$	$08D0_H$	×	×	×
134	ICTAPA2ADOUT0	$610C_H$	INTTAPA2ADOUT0	TAUA0 A/D conversion trigger output signal (trigger group 0)/ INTDTS100	TAPA2/DTS	135	$08E0_H$	$08E0_H$	○	○	○
135	ICTAPA0ADOUT0	$610E_H$	INTTAPA0ADOUT0	TAUA0 A/D conversion trigger output signal (trigger group 1)/ INTDTS101	TAPA0/DTS	136	$08F0_H$	$08F0_H$	○	○	○
136	ICTAPA0ADOUT1	6110_H	INTTAPA0ADOUT1	TAUA0 A/D conversion trigger output signal (trigger group 2)/ INTDTS102	TAPA0/DTS	137	0900_H	0900_H	○	○	○
137	ICTAPA1IPEK0	6112_H	INTTAPA1IPEK0	TAUA1 peak interrupt/ INTDTS103	TAPA1/DTS	138	0910_H	0910_H	○	○	○
138	ICTAPA1IVLY0	6114_H	INTTAPA1IVLY0	TAUA1 valley interrupt/ INTDTS104	TAPA1/DTS	139	0920_H	0920_H	○	○	○
139	Reserved	6116_H	Reserved			140	0930_H	0930_H	×	×	×
140	ICTAPA3ADOUT0	6118_H	INTTAPA3ADOUT0	TAUA1 A/D conversion trigger output signal (trigger group 0)/ INTDTS105	TAPA3/DTS	141	0940_H	0940_H	○	○	○
141	ICTAPA1ADOUT0	$611A_H$	INTTAPA1ADOUT0	TAUA1 A/D conversion trigger output signal (trigger group 1)/ INTDTS106	TAPA1/DTS	142	0950_H	0950_H	○	○	○
142	ICTAPA1ADOUT1	$611C_H$	INTTAPA1ADOUT1	TAUA1 conversion trigger output signal (trigger group 2)/ INTDTS107	TAPA1/DTS	143	0960_H	0960_H	○	○	○

Table 9-5 EI level maskable interrupt requests (8/14)

Interrupt			Interrupt request			Default priority	Exception cause code	Handler Address 0000...	Target product		
Channel	Control register		Name	Cause	Unit				μPD70F3510	μPD70F3512	μPD70F3514,70F3515
	Name	Address FFFF...									
143	ICCSIH0IRE	$611E_H$	INTCSIH0IRE	UARTJ0 reception error/ CSIH0 reception error/ IICB0 status detection	UARTJ0/ CSIH0/ IICB0/	144	0970_H	0970_H	○	○	○
144	ICCSIH0IR	6120_H	INTCSIH0IR	UARTJ0 reception/ CSIH0 reception function dependent/IICB0 transmission/ reception/INTDTS108	UARTJ0/ CSIH0/ IICB0/ DTS	145	0980_H	0980_H	○	○	○
145	ICCSIH0IC	6122_H	INTCSIH0IC	UARTJ0 transmission/CSIH0 transmission function dependent/INTDTS109	UARTJ0/ CSIH0/ DTS	146	0990_H	0990_H	○	○	○
146	ICCSIH0IJC	6124_H	INTCSIH0IJC	End of CSIH0 job/INTDTS110	CSIH0/ DTS	147	$09A0_H$	$09A0_H$	○	○	○
147	ICCSIH1IRE	6126_H	INTCSIH1IRE	FCN0 error/UARTJ1 reception error/CSIH1 reception error/ IICB1 status detection	FCN0/ UARTJ1/ CSIH1/ IICB1	148	$09B0_H$	$09B0_H$	○	○	○
148	ICCSIH1IR	6128_H	INTCSIH1IR	FCN0 reception completion/ UARTJ1 reception/ CSIH1 reception function dependent/IICB1 transmission/ reception/INTDTS111	FCN0/ UARTJ1/ CSIH1/ IICB1/ DTS	149	$09C0_H$	$09C0_H$	○	○	○
149	ICCSIH1IC	$612A_H$	INTCSIH1IC	FCN0 transmission completion/ UARTJ1 transmission/ CSIH1 transmission function dependent/INTDTS112	FCN0/ UARTJ1/ CSIH1/ DTS	150	$09D0_H$	$09D0_H$	○	○	○
150	ICCSIH1IJC	$612C_H$	INTCSIH1IJC	End of CSIH1 job/INTDTS113	CSIH1/ DTS	151	$09E0_H$	$09E0_H$	○	○	○
151	ICCSIH2IRE	$612E_H$	INTCSIH2IRE	UARTJ2 reception error/ IICB2 status detection/ CSIH2 reception error	CSIH2/ IICB2/ UARTJ2	152	$09F0_H$	$09F0_H$	○	○	○
152	ICCSIH2IR	6130_H	INTCSIH2IR	UARTJ2 reception/ IICB2 transmission/reception/ CSIH2 reception function dependent/INTDTS114	CSIH2/ UARTJ2/ DTS	153	$0A00_H$	$0A00_H$	○	○	○
153	ICCSIH2IC	6132_H	INTCSIH2IC	UARTJ2 transmission/ CSIH2 transmission function dependent/INTDTS115	CSIH2/ IICB2/ UARTJ2/ DTS	154	$0A10_H$	$0A10_H$	○	○	○
154	ICCSIH2IJC	6134_H	INTCSIH2IJC	End of CSIH2 job/INTDTS116	CSIH2/ DTS	155	$0A20_H$	$0A20_H$	○	○	○
155	ICCSIH3IRE	6136_H	INTCSIH3IRE	UARTJ3 reception error/ IICB3 status detection/ CSIH3 reception error	CSIH3/ IICB3/ UARTJ3	156	$0A30_H$	$0A30_H$	○	○	○

Table 9-5 EI level maskable interrupt requests (9/14)

Interrupt			Interrupt request			Default priority	Exception cause code	Handler Address 0000...	Target product		
Channel	Control register		Name	Cause	Unit				μPD70F3510	μPD70F3512	μPD70F3514,70F3515
	Name	Address FFFF...									
156	ICCSIH3IR	6138_H	INTCSIH3IR	UARTJ3 reception/ IICB3 transmission/reception/ CSIH3 reception function dependent/INTDTS117	CSIH3/ IICB3/ UARTJ3/ DTS	157	$0A40_H$	$0A40_H$	○	○	○
157	ICCSIH3IC	$613A_H$	INTCSIH3IC	CSIH3 transmission function dependent/UARTJ3 transmission/INTDTS118	CSIH3/ UARTJ3/ DTS	158	$0A50_H$	$0A50_H$	○	○	○
158	ICCSIH3IJC	$613C_H$	INTCSIH3IJC	End of CSIH3 job/INTDTS119	CSIH3/ DTS	159	$0A60_H$	$0A60_H$	○	○	○
159	ICCSIG0IRE	$613E_H$	INTCSIG0IRE	UARTE0 reception error/ CSIG0 reception error	UARTE0/ CSIG0	160	$0A70_H$	$0A70_H$	○	○	○
160	ICCSIG0IR	6140_H	INTCSIG0IR	UARTE0 reception/ CSIG0 reception completion	UARTE0/ CSIG0	161	$0A80_H$	$0A80_H$	○	○	○
161	ICCSIG0IC	6142_H	INTCSIG0IC	UARTE0 transmission/ CSIG0 transmission	UARTE0/ CSIG0	162	$0A90_H$	$0A90_H$	○	○	○
162	ICCSIG1IRE	6144_H	INTCSIG1IRE	UARTE1 reception error/ CSIG1 reception error	UARTE1/ CSIG1	163	$0AA0_H$	$0AA0_H$	○	○	○
163	ICCSIG1IR	6146_H	INTCSIG1IR	UARTE1 reception/ CSIG1 reception completion	UARTE1/ CSIG1	164	$0AB0_H$	$0AB0_H$	○	○	○
164	ICCSIG1IC	6148_H	INTCSIG1IC	UARTE1 transmission/ CSIG1 transmission	UARTE1/ CSIG1	165	$0AC0_H$	$0AC0_H$	○	○	○
165	ICCSIG2IRE	$614A_H$	INTCSIG2IRE	UARTE2 reception error/ CSIG2 reception error	UARTE2/ CSIG2	166	$0AD0_H$	$0AD0_H$	○	○	○
166	ICCSIG2IR	$614C_H$	INTCSIG2IR	UARTE2 reception/ CSIG2 reception completion	UARTE2/ CSIG2	167	$0AE0_H$	$0AE0_H$	○	○	○
167	ICCSIG2IC	$614E_H$	INTCSIG2IC	UARTE2 transmission/ CSIG2 transmission	UARTE2/ CSIG2	168	$0AF0_H$	$0AF0_H$	○	○	○
168	ICCSIG3IRE	6150_H	INTCSIG3IRE	UARTE3 reception error/ CSIG3 reception error	UARTE3/ CSIG3	169	$0B00_H$	$0B00_H$	○	○	○
169	ICCSIG3IR	6152_H	INTCSIG3IR	UARTE3 reception/ CSIG3 reception completion	UARTE3/ CSIG3	170	$0B10_H$	$0B10_H$	○	○	○
170	ICCSIG3IC	6154_H	INTCSIG3IC	UARTE3 transmission/ CSIG3 transmission	UARTE3/ CSIG3	171	$0B20_H$	$0B20_H$	○	○	○
171	ICCSIG4IRE	6156_H	INTCSIG4IRE	UARTE4 reception error/ IICB4 status detection/ CSIG4 reception error	UARTE4/ IICB4/ CSIG4	172	$0B30_H$	$0B30_H$	○	○	○
172	ICCSIG4IR	6158_H	INTCSIG4IR	UARTE4 reception/ IICB4 transmission/reception/ CSIG4 reception completion	UARTE4/ IICB4/ CSIG4	173	$0B40_H$	$0B40_H$	○	○	○
173	ICCSIG4IC	$615A_H$	INTCSIG4IC	UARTE4 transmission/ CSIG4 transmission	UARTE4/ CSIG4	174	$0B50_H$	$0B50_H$	○	○	○

Table 9-5 EI level maskable interrupt requests (10/14)

Interrupt			Interrupt request			Default priority	Exception cause code	Handler Address 0000...	Target product		
Channel	Control register		Name	Cause	Unit				μPD70F3510	μPD70F3512	μPD70F3514,70F3515
	Name	Address FFFF...									
174	ICCSIG5IRE	$615C_H$	INTCSIG5IRE	FCN1 error/UARTE5 reception error/IICB5 status detection/ CSIG5 reception error	FCN1/ UARTE5/ IICB5/ CSIG5	175	$0B60_H$	$0B60_H$	△	○	○
175	ICCSIG5IR	$615E_H$	INTCSIG5IR	FCN1 reception completion/ UARTE5 reception/ IICB5 transmission/reception/ CSIG5 reception completion	FCN1/ UARTE5/ IICB5/ CSIG5/	176	$0B70_H$	$0B70_H$	△	○	○
176	ICCSIG5IC	6160_H	INTCSIG5IC	FCN1 transmission completion/ UARTE5 transmission/ CSIG5 transmission	FCN1/ UARTE5/ CSIG5	177	$0B80_H$	$0B80_H$	△	○	○
177	ICFCNWUP	6162_H	INTFCNWUP	FCN0 wakeup/FCN1 wakeup	FCN0/ FCN1	178	$0B90_H$	$0B90_H$	×	○	○
178	ICETHA0SRX	6164_H	INTETHA0SRX	Ethernet reception packet read request	Ethernet	179	$0BA0_H$	$0BA0_H$	×	○	○
179	ICETHA0SCRX	6166_H	INTETHA0SCRX	Ethernet packet reception/ INTDTS120	Ethernet /DTS	180	$0BB0_H$	$0BB0_H$	×	○	○
180	ICETHA0SCTX	6168_H	INTETHA0SCTX	Ethernet packet transmission/ INTDTS121	Ethernet /DTS	181	$0BC0_H$	$0BC0_H$	×	○	○
181	ICETHA0RS	$616A_H$	INTETHA0RS	Ethernet reception status detection	Ethernet	182	$0BD0_H$	$0BD0_H$	×	○	○
182	ICETHA0TS	$616C_H$	INTETHA0TS	Ethernet transmission status detection	Ethernet	183	$0BE0_H$	$0BE0_H$	×	○	○
183	ICETHA0FS	$616E_H$	INTETHA0FS	Ethernet FIFO status detection	Ethernet	184	$0BF0_H$	$0BF0_H$	×	○	○
184	ICETHA0MAC	6170_H	INTETHA0MAC	Ethernet statistics counter overflow	Ethernet	185	$0C00_H$	$0C00_H$	×	○	○
185	ICETHA0SCRXTCH	6172_H	INTETHA0SCRXTCH	Ethernet transmission data calculation completion	Ethernet	186	$0C10_H$	$0C10_H$	×	○	○
186	ICETHA0SCTXTCH	6174_H	INTETHA0SCTXTCH	Ethernet transmission checksum writing	Ethernet	187	$0C20_H$	$0C20_H$	×	○	○
187	ICUSFA0I0	6176_H	INTUSFA0I0	Bridge interrupt	USBF	188	$0C30_H$	$0C30_H$	○	○	○
188	ICUSFA0I1	6178_H	INTUSFA0I1	EPC interrupt	USBF	189	$0C40_H$	$0C40_H$	○	○	○
189	ICUSFA0I2	$617A_H$	INTUSFA0I2	USBF resume	USBF	190	$0C50_H$	$0C50_H$	○	○	○
190	ICUSHA0I0	$617C_H$	INTUSHA0I0	USBH PCI cycle error	USBH	191	$0C60_H$	$0C60_H$	○	○	○
191	ICUSHA0I1	$617E_H$	INTUSHA0I1	USBH status detection	USBH	192	$0C70_H$	$0C70_H$	○	○	○
192	ICUSHA0PME	6180_H	INTUSHA0PME	USB HOST PME (power management)	USBH	193	$0C80_H$	$0C80_H$	○	○	○
193	ICDMA0	6182_H	INTDMA0	DMA channel 0 transfer completion	DMAC	194	$0C90_H$	$0C90_H$	○	○	○
194	ICDMA1	6184_H	INTDMA1	DMA channel 1 transfer completion	DMAC	195	$0CA0_H$	$0CA0_H$	○	○	○

Table 9-5 EI level maskable interrupt requests (11/14)

Interrupt			Interrupt request			Default priority	Exception cause code	Handler Address 0000...	Target product		
Channel	Control register		Name	Cause	Unit				μPD70F3510	μPD70F3512	μPD70F3514,70F3515
	Name	Address FFFF...									
195	ICDMA2	6186_H	INTDMA2	DMA channel 2 transfer completion	DMAC	196	$0CB0_H$	$0CB0_H$	○	○	○
196	ICDMA3	6188_H	INTDMA3	DMA channel 3 transfer completion	DMAC	197	$0CC0_H$	$0CC0_H$	○	○	○
197	ICDMA4	$618A_H$	INTDMA4	DMA channel 4 transfer completion	DMAC	198	$0CD0_H$	$0CD0_H$	○	○	○
198	ICDMA5	$618C_H$	INTDMA5	DMA channel 5 transfer completion	DMAC	199	$0CE0_H$	$0CE0_H$	○	○	○
199	ICDMA6	$618E_H$	INTDMA6	DMA channel 6 transfer completion	DMAC	200	$0CF0_H$	$0CF0_H$	○	○	○
200	ICDMA7	6190_H	INTDMA7	DMA channel 7 transfer completion	DMAC	201	$0D00_H$	$0D00_H$	○	○	○
201	ICDMA8	6192_H	INTDMA8	DMA channel 8 transfer completion	DMAC	202	$0D10_H$	$0D10_H$	○	○	○
202	ICDMA9	6194_H	INTDMA9	DMA channel 9 transfer completion	DMAC	203	$0D20_H$	$0D20_H$	○	○	○
203	ICDMA10	6196_H	INTDMA10	DMA channel 10 transfer completion	DMAC	204	$0D30_H$	$0D30_H$	○	○	○
204	ICDMA11	6198_H	INTDMA11	DMA channel 11 transfer completion	DMAC	205	$0D40_H$	$0D40_H$	○	○	○
205	ICDMA12	$619A_H$	INTDMA12	DMA channel 12 transfer completion	DMAC	206	$0D50_H$	$0D50_H$	○	○	○
206	ICDMA13	$619C_H$	INTDMA13	DMA channel 13 transfer completion	DMAC	207	$0D60_H$	$0D60_H$	○	○	○
207	ICDMA14	$619E_H$	INTDMA14	DMA channel 14 transfer completion	DMAC	208	$0D70_H$	$0D70_H$	○	○	○
208	ICDMA15	$61A0_H$	INTDMA15	DMA channel 15 transfer completion	DMAC	209	$0D80_H$	$0D80_H$	○	○	○
209	ICDMACT0	$61A2_H$	INTDMACT0	DMA channel 0 transfer count match (for next address specification)	DMAC	210	$0D90_H$	$0D90_H$	○	○	○
210	ICDMACT1	$61A4_H$	INTDMACT1	DMA channel 1 transfer count match (for next address specification)	DMAC	211	$0DA0_H$	$0DA0_H$	○	○	○
211	ICDMACT2	$61A6_H$	INTDMACT2	DMA channel 2transfer count match (for next address specification)	DMAC	212	$0DB0_H$	$0DB0_H$	○	○	○
212	ICDMACT3	$61A8_H$	INTDMACT3	DMA channel 3 transfer count match (for next address specification)	DMAC	213	$0DC0_H$	$0DC0_H$	○	○	○

Table 9-5 EI level maskable interrupt requests (12/14)

Interrupt			Interrupt request			Default priority	Exception cause code	Handler Address 0000...	Target product		
Channel	Control register		Name	Cause	Unit				μPD70F3510	μPD70F3512	μPD70F3514,70F3515
	Name	Address FFFF...									
213	ICDMACT4	$61AA_H$	INTDMACT4	DMA channel 4 transfer count match (for next address specification)	DMAC	214	$0DD0_H$	$0DD0_H$	○	○	○
214	ICDMACT5	$61AC_H$	INTDMACT5	DMA channel 5 transfer count match (for next address specification)	DMAC	215	$0DE0_H$	$0DE0_H$	○	○	○
215	ICDMACT6	$61AE_H$	INTDMACT6	DMA channel 6 transfer count match (for next address specification)	DMAC	216	$0DF0_H$	$0DF0_H$	○	○	○
216	ICDMACT7	$61B0_H$	INTDMACT7	DMA channel 7 transfer count match (for next address specification)	DMAC	217	$0E00_H$	$0E00_H$	○	○	○
217	ICDMACT8	$61B2_H$	INTDMACT8	DMA channel 8 transfer count match (for next address specification)	DMAC	218	$0E10_H$	$0E10_H$	○	○	○
218	ICDMACT9	$61B4_H$	INTDMACT9	DMA channel 9 transfer count match (for next address specification)	DMAC	219	$0E20_H$	$0E20_H$	○	○	○
219	ICDMACT10	$61B6_H$	INTDMACT10	DMA channel 10 transfer count match (for next address specification)	DMAC	220	$0E30_H$	$0E30_H$	○	○	○
220	ICDMACT11	$61B8_H$	INTDMACT11	DMA channel 11 transfer count match (for next address specification)	DMAC	221	$0E40_H$	$0E40_H$	○	○	○
221	ICDMACT12	$61BA_H$	INTDMACT12	DMA channel 12 transfer count match (for next address specification)	DMAC	222	$0E50_H$	$0E50_H$	○	○	○
222	ICDMACT13	$61BC_H$	INTDMACT13	DMA channel 13 transfer count match (for next address specification)	DMAC	223	$0E60_H$	$0E60_H$	○	○	○
223	ICDMACT14	$61BE_H$	INTDMACT14	DMA channel 14 transfer count match (for next address specification)	DMAC	224	$0E70_H$	$0E70_H$	○	○	○
224	ICDMACT15	$61C0_H$	INTDMACT15	DMA channel 15 transfer count match (for next address specification)	DMAC	225	$0E80_H$	$0E80_H$	○	○	○
225	ICHDMAERR	$61C2_H$	INTHDMAERR	S_DMA error response	S_DMAC	226	$0E90_H$	$0E90_H$	○	○	○
226	ICHDMA0	$61C4_H$	INTHDMA0	S_DMA channel 0 transfer completion (external)	S_DMAC	227	$0EA0_H$	$0EA0_H$	○	○	○
227	ICHDMA1	$61C6_H$	INTHDMA1	S_DMA channel 1 transfer completion (external)	S_DMAC	228	$0EB0_H$	$0EB0_H$	○	○	○
228	ICHDMA2	$61C8_H$	INTHDMA2	S_DMA channel 2 transfer completion (external)	S_DMAC	229	$0EC0_H$	$0EC0_H$	○	○	○

Table 9-5 EI level maskable interrupt requests (13/14)

Interrupt			Interrupt request			Default priority	Exception cause code	Handler Address 0000...	Target product		
Channel	Control register		Name	Cause	Unit				μPD70F3510	μPD70F3512	μPD70F3514,70F3515
	Name	Address FFFF...									
229	ICHDMA3	$61CA_H$	INTHDMA3	S_DMA channel 3 transfer completion (external)	S_DMAC	230	$0ED0_H$	$0ED0_H$	○	○	○
230	ICHDMA4	$61CC_H$	INTHDMA4	S_DMA channel 4 transfer completion (Caused by DMA channel 0 completion)	S_DMAC	231	$0EE0_H$	$0EE0_H$	○	○	○
231	ICHDMA5	$61CE_H$	INTHDMA5	S_DMA channel 5 transfer completion (Caused by DMA channel 1 completion)	S_DMAC	232	$0EF0_H$	$0EF0_H$	○	○	○
232	ICHDMA6	$61D0_H$	INTHDMA6	S_DMA channel 6 transfer completion (Caused by DMA channel 2 completion)	S_DMAC	233	$0F00_H$	$0F00_H$	○	○	○
233	ICHDMA7	$61D2_H$	INTHDMA7	S_DMA channel 7 transfer completion (Caused by DMA channel 3 completion)	S_DMAC	234	$0F10_H$	$0F10_H$	○	○	○
234	Reserved	$61D4_H$	Reserved			235	$0F20_H$	$0F20_H$	×	×	×
235	Reserved	$61D6_H$	Reserved			236	$0F30_H$	$0F30_H$	×	×	×
236	Reserved	$61D8_H$	Reserved			237	$0F40_H$	$0F40_H$	×	×	×
237	Reserved	$61DA_H$	Reserved			238	$0F50_H$	$0F50_H$	×	×	×
238	Reserved	$61DC_H$	Reserved			239	$0F60_H$	$0F60_H$	×	×	×
239	Reserved	$61DE_H$	Reserved			240	$0F70_H$	$0F70_H$	×	×	×
240	Reserved	$61E0_H$	Reserved			241	$0F80_H$	$0F80_H$	×	×	×
241	Reserved	$61E2_H$	Reserved			242	$0F90_H$	$0F90_H$	×	×	×
242	Reserved	$61E4_H$	Reserved			243	$0FA0_H$	$0FA0_H$	×	×	×
243	Reserved	$61E6_H$	Reserved			244	$0FB0_H$	$0FB0_H$	×	×	×
244	Reserved	$61E8_H$	Reserved			245	$0FC0_H$	$0FC0_H$	×	×	×
245	Reserved	$61EA_H$	Reserved			246	$0FD0_H$	$0FD0_H$	×	×	×
246	Reserved	$61EC_H$	Reserved			247	$0FE0_H$	$0FE0_H$	×	×	×
247	Reserved	$61EE_H$	Reserved			248	$0FF0_H$	$0FF0_H$	×	×	×
248	Reserved	$61F0_H$	Reserved			249	1000_H	1000_H	×	×	×
249	Reserved	$61F2_H$	Reserved			250	1010_H	1010_H	×	×	×
250	Reserved	$61F4_H$	Reserved			251	1020_H	1020_H	×	×	×
251	Reserved	$61F6_H$	Reserved			252	1030_H	1030_H	×	×	×
252	Reserved	$61F8_H$	Reserved			253	1040_H	1040_H	×	×	×

Table 9-5 EI level maskable interrupt requests (14/14)

Interrupt			Interrupt request			Default priority	Exception cause code	Handler Address 0000...	Target product		
Channel	Control register		Name	Cause	Unit				μPD70F3510	μPD70F3512	μPD70F3514,70F3515
	Name	Address FFFF...									
253	Reserved	$61FA_H$	Reserved			254	1050_H	1050_H	×	×	×
254	Reserved	$61FC_H$	Reserved			255	1060_H	1060_H	×	×	×
255	Reserved	$61FE_H$	Reserved			256	1070_H	1070_H	×	×	×

Notes 1. ○: Can be used.
×: Cannot be used.
△: Not all functions can be used.

2. Default priority
The default priority number indicates the priority order when EI level maskable interrupt requests that have the same priority occur at the same time. 0 indicates the highest priority.

9.3 Interrupt Controller Control Registers

9.3.1 EICn (n = 0 to 255): EI level interrupt control registers

These registers, each of which is for a channel of EI level maskable interrupt (EIINT), are used to set a condition to control each channel. The values of bits 15 to 13, bits 11 to 8, and bits 6 to 4 are always 0.

Update the EICn register by a byte write access or a halfword access.

Access This register can be read/written in 16-bit units. Note, however, that the EICn register can be read and written in 8-bit or 1-bit units if its upper 8 bits and lower 8 bits are used as the EICnH and EICnL registers, respectively.

Address $FFFF\ 6000_H + n \times 02_H$

Initial Value $008F_H$. This register is initialized by any reset.

Caution Do not access EICn registers of interrupt channels, not listed in the above interrupt request tables.

15	14	13	12	11	10	9	8
0	0	0	EIRFn	0	0	0	0
R	R	R	R/W	R	R	R	R

7	6	5	4	3	2	1	0
EIMKn	0	0	0	EIP3n	EIP2n	EIP1n	EIP0n
R/W	R	R	R	R/W	R/W	R/W	R/W

Table 9-6 EICn register contents

Bit position	Bit name	Function
12	EIRFn	Interrupt request flag The EIRFn bit can be written from a program. Setting the EIRFn bit (1) generates an EI level maskable interrupt (EIINTn), just as when an interrupt request is acknowledged. 0: No interrupt request (initial value) 1: Interrupt request
7	EIMKn	Interrupt mask bit. When the EIMKn bit is set, the interrupt request set to the interrupt request flag (EIRFn) is masked, so that the interrupt request is not issued from that channel to the CPU core. From a channel with the EIMKn bit set, interrupt pending status is not displayed by the ICSR.PMF bit. The EIMKn bit does not mask a signal input from an interrupt input pin itself and, therefore, the corresponding interrupt request flag is set even when the EIMKn bit is set. The setting of the corresponding bit of the interrupt mask register (IMR) is also reflected. 0: Enables interrupt servicing. 1: Disables interrupt servicing (initial value).
3:0	EIP3n to EIP0n	These bits specify 16 levels of interrupt priorities. The highest priority is 0 and the lowest is 15. If two or more interrupt requests of EI level are generated at the same time, the interrupt source having the higher priority specified by these bits is selected and reported to the CPU core. If the priority specified by the EIP3n to EIP0n bits is the same, the source having the lower channel number is selected by a fixed priority.

9.3.2 IMRm (m = 0 to 15): EI level interrupt mask registers

These registers are a collection of the EIMKn bits of the EICn register. Each bit of the IMRm register reflects the setting of the corresponding EIMKn bit. Setting the IMRm register reflects on the corresponding EIMKn bit.

Access When this register is accessed in 16-bit, 8-bit, or 1-bit units, the upper 8 bits [15:8] and lower 8 bits [7:0] correspond to the IMRmH and IMRmL registers, respectively.

Address

IMR0: FFFF 6400$_H$ IMR1: FFFF 6402$_H$
IMR2: FFFF 6404$_H$ IMR3: FFFF 6406$_H$
IMR4: FFFF 6408$_H$ IMR5: FFFF 640A$_H$
IMR6: FFFF 640C$_H$ IMR7: FFFF 640E$_H$
IMR8: FFFF 6410$_H$ IMR9: FFFF 6412$_H$
IMR10: FFFF 6414$_H$ IMR11: FFFF 6416$_H$
IMR12: FFFF 6418$_H$ IMR13: FFFF 641A$_H$
IMR14: FFFF 641C$_H$ IMR15: FFFF 641E$_H$

Initial Value FFFF$_H$. This register is initialized by any reset.

Caution Be sure to set the EIMKn bit corresponding to interrupt channels not listed in the above interrupt request tables to 1.

15	14	13	12	11	10	9	8
EIMK m × 16+15	EIMK m × 16+14	EIMK m × 16+13	EIMK m × 16+12	EIMK m × 16+11	EIMK m × 16+10	EIMK m × 16+9	EIMK m × 16+8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
EIMK m × 16+7	EIMK m × 16+6	EIMK m × 16+5	EIMK m × 16+4	EIMK m × 16+3	EIMK m × 16+2	EIMK m × 16+1	EIMK m × 16+0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 9-7 IMRm register contents

Bit position	Bit name	Function
15:0	EIMK15 to EIMK0	These bits mask an interrupt from channels 0 to 15 of EI level maskable interrupt (EIINT). 0: Enables interrupt servicing. 1: Disables interrupt servicing.

9.3.3 ISPR: In-service priority register

This register holds the interrupt priority of EI level maskable interrupt (EIINT) that is being processed by the CPU core. When a response of acknowledging an interrupt request is received from the CPU core, the bit of this register corresponding to the priority of that interrupt request is set. When it is reported by the CPU core that interrupt servicing has been completed, the bit of this register having the highest priority of those that are set is automatically cleared. The bit of this register is not cleared if execution has returned from an FE level interrupt. If multiple interrupts of EI level maskable interrupt (EIINT) are generated, the bits of this register corresponding to the priorities of the interrupts that have been acknowledged are sequentially set, and the history of the priorities of the multiple interrupts is held.

After 1 has been simultaneously written in 16-bit units to the ISPC register, all the bits of the ISPR register can be cleared by writing 0 to the 16 bits of the ISPR register at the same time. Each bit of this register cannot be cleared or set by software. Once one of the bits has been cleared, it is impossible to restore its original value. When this register is accessed in 8-bit units, the upper 8 bits [15:8] and lower 8 bits [7:0] correspond to the ISPRH and ISPRL registers, respectively.

Access This register is read-only, in 16-bit or 8-bit units.
When this register is accessed in 8-bit units, the upper 8 bits [15:8] and lower 8 bits [7:0] correspond to the ISPRH and ISPRL registers, respectively.

Address FFFF 6440$_H$

Initial Value 0000$_H$. This register is initialized by any reset.

15	14	13	12	11	10	9	8
ISPR15	ISPR14	ISPR13	ISPR12	ISPR11	ISPR10	ISPR9	ISPR8
R	R	R	R	R	R	R	R

7	6	5	4	3	2	1	0
ISPR7	ISPR6	ISPR5	ISPR4	ISPR3	ISPR2	ISPR1	ISPR0
R	R	R	R	R	R	R	R

Table 9-8 ISPR register contents

Bit position	Bit name	Function
15:0	ISPR15 to ISPR0	These bits indicate the priority of the interrupt being acknowledged. 0: Interrupt request of the priority corresponding to a specified bit position is not acknowledged. 1: Interrupt request of the priority corresponding to a specified bit position is being processed by the CPU core.

9.3.4 PMR: Priority mask register

This register specifies an interrupt priority by which an interrupt request flag of EI level maskable interrupt (EIINT) is to be masked. It disables all at once the interrupt requests from the EIINT channel for which the interrupt priority specified by this register is set.

The position of each bit of this register corresponds to an interrupt priority. For example, if 1 is set to bit 0, channel of interrupt priority 0 can be masked.

Access This register can be read/written in 16-bit, 8-bit, or 1-bit units. When this register is accessed in 8-bit or 1-bit units, the upper 8 bits [15:8] and lower 8 bits [7:0] correspond to the PMRH and PMRL registers, respectively.

Address FFFF 6448_H

Initial Value 0000_H. This register is initialized by any reset.

15	14	13	12	11	10	9	8
PMR15	PMR14	PMR13	PMR12	PMR11	PMR10	PMR9	PMR8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7	6	5	4	3	2	1	0
PMR7	PMR6	PMR5	PMR4	PMR3	PMR2	PMR1	PMR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 9-9 PMR register contents

Bit position	Bit name	Function
15:0	PMR15 to PMR0	These bits specify an interrupt priority by which an interrupt request flag. 0: Enables interrupt servicing of the priority corresponding to a specified bit position (initial value). 1: Disables interrupt servicing of the priority corresponding to a specified bit position.

9.3.5 ISPC: In-service priority clear register

When 1 is written to all the bits of this ISPC register and then 0 is written to all the bits of the ISPR register, all the bits of the ISPR register can be cleared to 0. At the same time, all the processing modes of FE level NMI, FE level maskable interrupt (FEINT), and EI level maskable interrupt (EIINT) of the ICSR register are canceled. As a result, the internal mode registers of INTC for interrupt servicing, which indicate that an interrupt request is being processed by the CPU core, are cleared. The contents of these registers that once have been cleared cannot be restored by software.

When the ISPR register is cleared by writing 0 to all the bits of the ISPR register, the value of the ISPC register is also automatically cleared to 0. The value that is to be read from all the bits of this register when the register is read is 1 if 1 has been written to all the bits, or 0 after reset or after the ISPR register has been cleared. The value of the bits of this register does not change even if the register is written when all its bits are not 1 or 0 at the same time. If 0 is written to all the bits while they are 1, the value of all the bits of the ISPC register is cleared to 0, and the value of the ISPR register does not change.

Access This register can be read/written only in 16-bit units.

Address FFFF 6450_H

Initial Value 0000_H. This register is initialized by any reset.

15	14	13	12	11	10	9	8
ISPC15	ISPC14	ISPC13	ISPC12	ISPC11	ISPC10	ISPC9	ISPC8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7	6	5	4	3	2	1	0
ISPC7	ISPC6	ISPC5	ISPC4	ISPC3	ISPC2	ISPC1	ISPC0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 9-10 ISPC register contents

Bit position	Bit name	Function
15:0	ISPC15 to ISPC0	1 or 0 is read from all these bits. ISPR can be cleared if 0 is written to all the bits of ISPR when 1 is read from all the bits of this register.

9.3.6 SCR: Selected channel hold register

This register holds the channel number of the EI level maskable interrupt (EIINT) acknowledged by the CPU. It is read-only in 16-bit or 8-bit units, and cannot be written by software. The value of this register is updated when an interrupt vector is reported to the CPU core. Note that this register is overwritten when multiple interrupt requests of EI level INT are acknowledged. This register is not updated when an interrupt request of FE level is acknowledged.

Access This register is read-only, in 16-bit or 8-bit units. When this register is accessed in 8-bit units, the upper 8 bits [15:8] and lower 8 bits [7:0] correspond to the SCRH and SCRL registers, respectively.

Address FFFF 6458_H

Initial Value 0000_H. This register is initialized by any reset.

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R

7	6	5	4	3	2	1	0
SCR7	SCR6	SCR5	SCR4	SCR3	SCR2	SCR1	SCR0
R	R	R	R	R	R	R	R

Table 9-11 SCR register contents

Bit position	Bit name	Function
7:0	SCR7 to SCR0	Holds the channel number of the maskable interrupt that has been acknowledged by the CPU. The value of these bits is updated when an interrupt vector is reported to the CPU core. It is overwritten when multiple interrupts of EI level maskable interrupt (EIINT) are acknowledged. These bits are not updated when an FE level interrupt is acknowledged. Nothing happens if this register is accessed for write.

9.3.7 ICSR: Interrupt controller status register

This register indicates the operation status of the interrupt controller. Especially, bits 2 to 0 of this register serve as a mode register of interrupt servicing.

Access This register is read-only, in 16-bit, 8-bit, or 1-bit units. When this register is accessed in 8-bit or 1-bit units, the upper 8 bits [15:8] and lower 8 bits [7:0] correspond to the ICSRH and ICSRL registers, respectively.

Address FFFF 645A$_H$

Initial Value 0000$_H$. This register is initialized by any reset.

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	PMF
R	R	R	R	R	R	R	R

7	6	5	4	3	2	1	0
0	FNR	FIR	EIR	0	FNE	FIE	EIE
R	R	R	R	R	R	R	R

Table 9-12 ISCR register contents

Bit position	Bit name	Function
8	PMF	Indicates 1 if the request flag of a channel of EI level INT that has the interrupt priority prohibited by the setting of PMR from being serviced is set.
6	FNR	Indicates 1 if an FE level non-maskable interrupt (FENMI) has been issued to the CPU.
5	FIR	Indicates 1 if an FE level maskable interrupt (FEINT) has been issued to the CPU.
4	EIR	Indicates 1 if an EI level maskable interrupt (EIINT) has been issued to the CPU.
2	FNE	Indicates 1 if an CPU is processing the FE level non-maskable interrupt (FENMI).
1	FIE	Indicates 1 if an CPU is processing the FE level maskable interrupt (FEINT).
0	EIE	Indicates 1 if an CPU is processing the EI level maskable interrupt (EIINT).

9.3.8 FNC: FE level NMI control register

This register is used to set a condition to control the FE level non-maskable interrupt (FENMI).

Access This register is read-only, in 16-bit, 8-bit, or 1-bit units. When this register is accessed in 8-bit or 1-bit units, the upper 8 bits [15:8] correspond to the FNCH register.

Address FFFF 645C$_H$

Initial Value 0000$_H$. This register is initialized by any reset.

15	14	13	12	11	10	9	8
0	0	0	FNRF	0	0	0	0
R	R	R	R	R	R	R	R

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R

Table 9-13 FNC register contents

Bit position	Bit name	Function
12	FNRF	Interrupt request flag 0: No interrupt request (initial value) 1: Interrupt request

9.3.9 FIC: FE level INT control register

This register is used to set a condition to control the FE level maskable interrupt (FEINT).

Access This register is read-only, in 16-bit, 8-bit, or 1-bit units. When this register is accessed in 8-bit or 1-bit units, the upper 8 bits [15:8] correspond to the FICH register.

Address FFFF 645E$_H$

Initial Value 0000$_H$. This register is initialized by any reset.

15	14	13	12	11	10	9	8
0	0	0	FIRF	0	0	0	0
R	R	R	R	R	R	R	R

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R

Table 9-14 FNC register contents

Bit position	Bit name	Function
12	FIRF	Interrupt request flag 0 : No interrupt request (initial value) 1 : Interrupt request

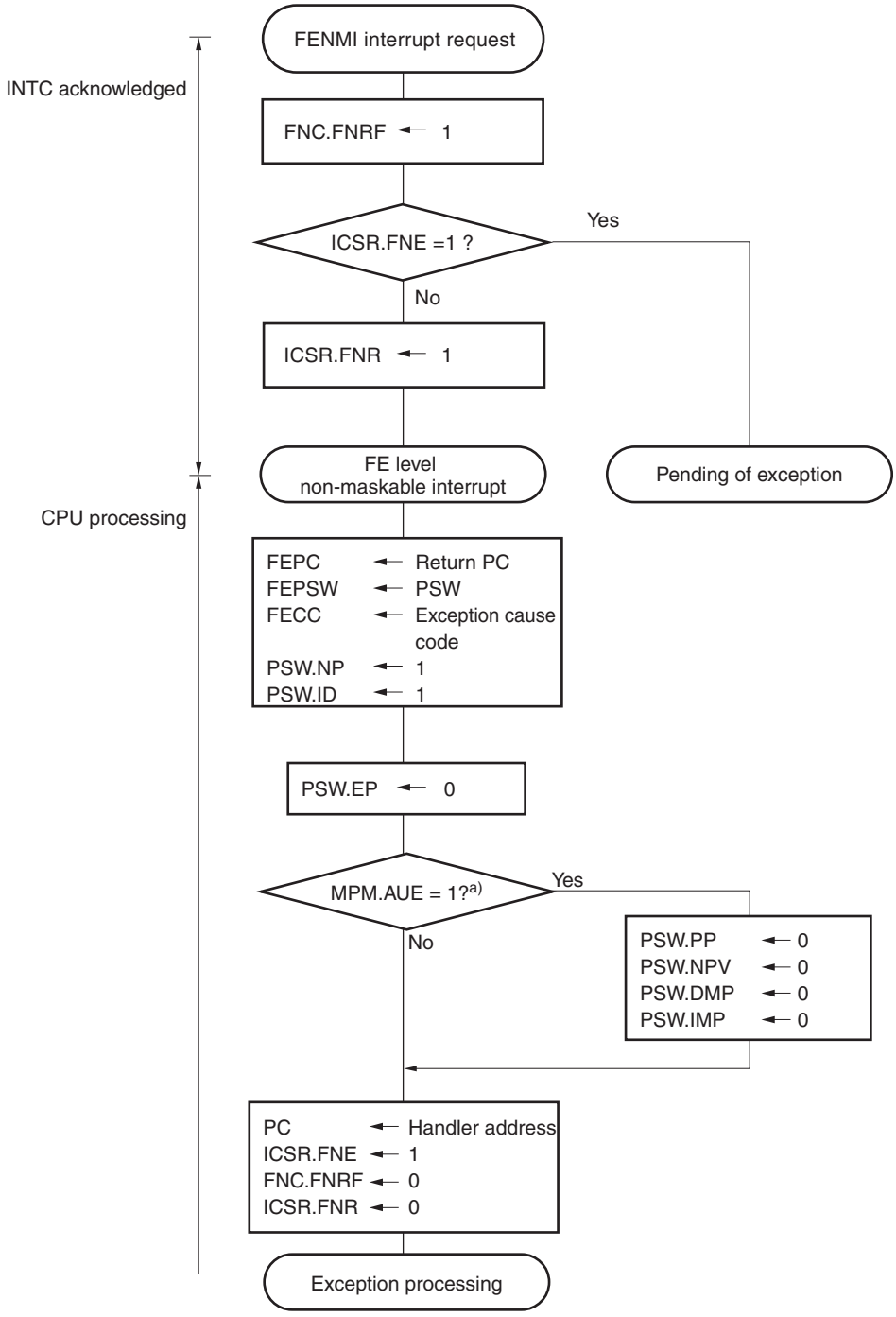
9.4 Interrupt Acknowledgment and Restoring

This section describes the operation during interrupt acknowledgment and restoring from interrupt servicing.

9.4.1 FE level non-maskable interrupt caused by FENMI interrupt request

When an FENMI interrupt is requested, an FE level non-maskable interrupt is generated in CPU. This FE level non-maskable interrupt is used when a fatal system error occurs.

Caution Upon acknowledgment of the FENMI interrupt, generation of the next FENMI, FEINT, or EIINT interrupt is pended until the FERET instruction is executed (interrupt request is acknowledged and held.)
FENMI can be acknowledged even when the NP bit is set to 1. Therefore, if the FENMI interrupt occurs during the processing of an FEINT exception, PPI exception, or other FE level exceptions, the save address is lost and cannot be restored. After a FENMI interrupt is requested and the required processing has been completed, execute a system reset, etc. Return to the original processing is not possible.



a) If a processor protection exception (MDP or MIP) occurs, the PSW.PP, NPV, DMP, and IMP bits are always cleared (0), regardless of the status of MPM.AUE.

Figure 9-1 Processing upon occurrence of FENMI interrupt request

9.4.2 Restore from FE level non-maskable interrupt (FENMI)

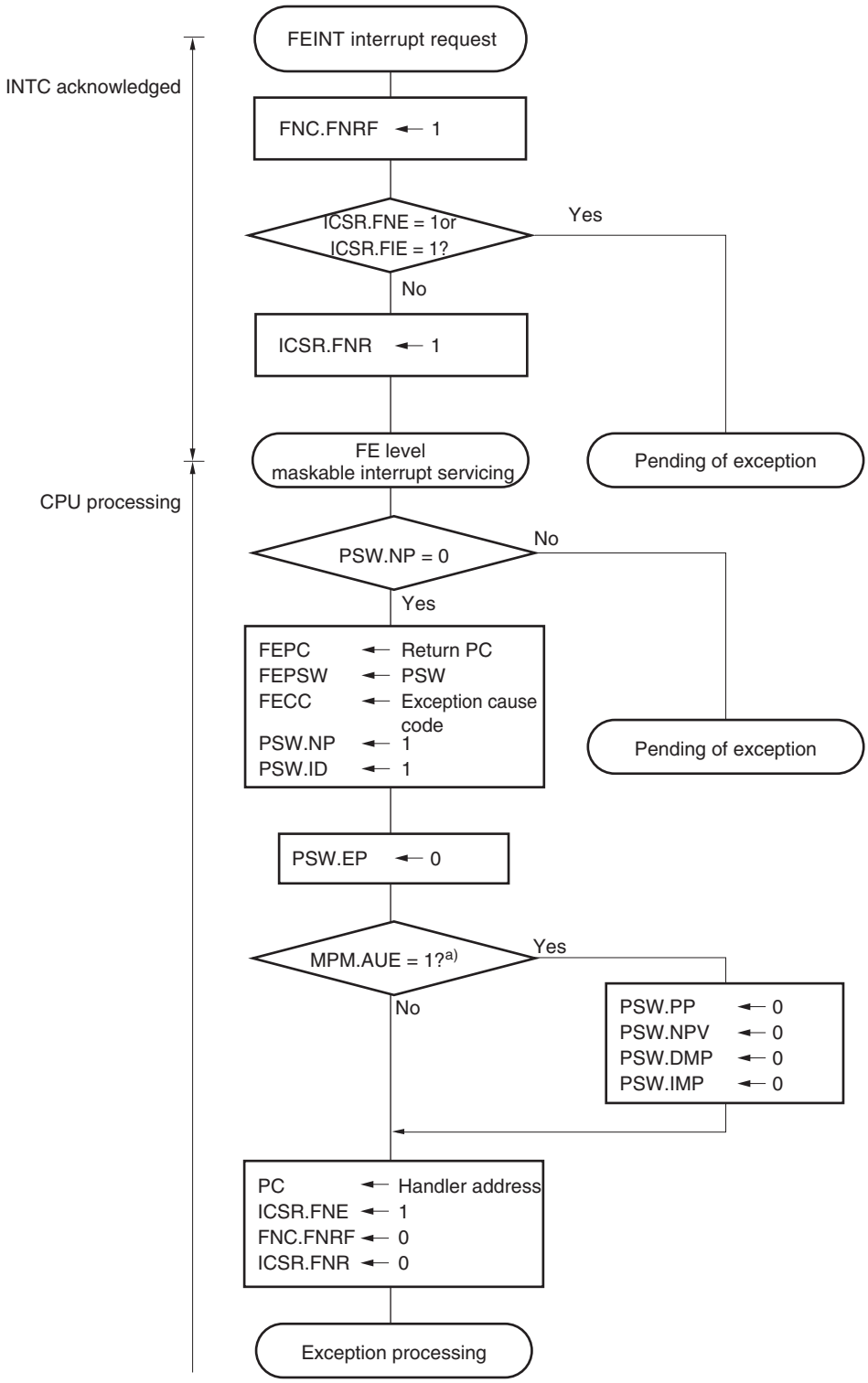
An FE level non-maskable interrupt (FENMI) cannot be restored since it is an interrupt used in cases such as when a fatal system error occurs.

Execute a system reset after exception processing.

9.4.3 FE level maskable interrupt caused by FEINT interrupt request

When an FEINT interrupt is requested by the FEINT pin, an FE level maskable interrupt is generated. This interrupt is a recoverable FE level interrupt.

Upon acknowledgment of the FEINT interrupt, generation of the next FEINT or EIINT interrupt is pended until the FERET instruction is executed. Interrupt request is acknowledged and held.



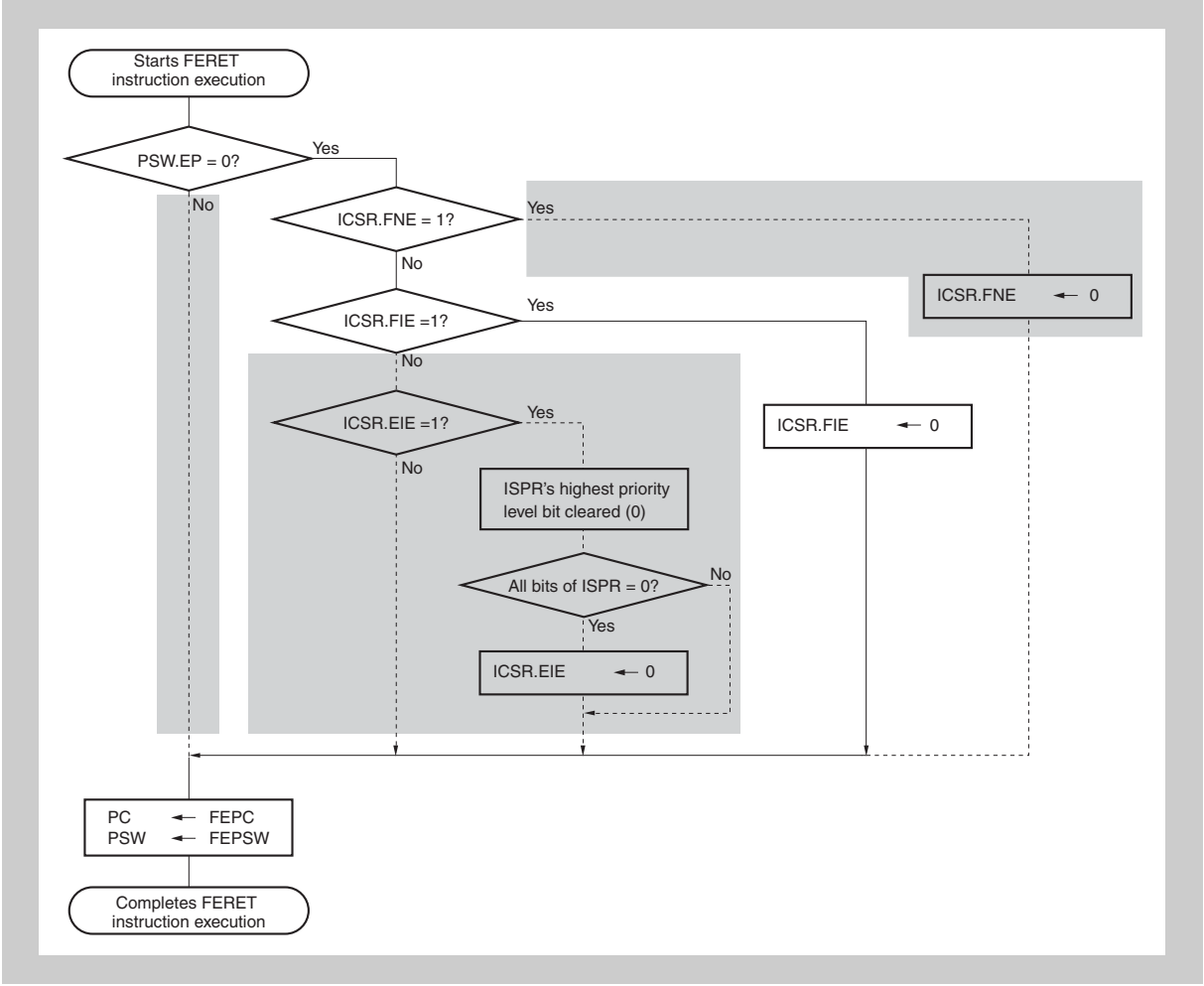
a) If a processor protection exception (MDP or MIP) occurs, the PSW.PP, NPV, DMP, and IMP bits are always cleared (0), regardless of the status of MPM.AUE.

Figure 9-2 Processing upon occurrence of FEINT interrupt request

9.4.4 Restore from FE level maskable interrupt (FEINT) servicing

Restore from FE level maskable interrupt (FEINT) servicing is performed using the FERET instructions. Execution of the FERET instruction while the PSW.EP bit status is cleared (0) causes restore processing from the FE level maskable interrupt interrupt (FEINT). Completely restoring from interrupt servicing when the PSW.EP bit is "1" is not possible (clearing of the ICSR, ISPR, and other registers is not performed). For return from FE level maskable interrupts, execute the FERET instruction with the PSW.EP bit always cleared (0).

Caution In the V850E2M CPU core, although RETI instructions are provided for backward compatibility with V850E1 and V850E2 architectures, but their use is, in principle, prohibited. Replace all RETI instructions other than existing programs that cannot be modified with EIRET or FERET instructions.



Note Shaded portions indicate processing which do not branch during restoring from an FE level maskable interrupt (FEINT) servicing.

Figure 9-3 Restore from FE level maskable interrupt (FEINT) servicing

9.4.5 EI level maskable interrupt caused by EIINT interrupt request

When an EI level maskable interrupt is requested, an EIINT interrupt is requested to CPU (the transition to the interrupt handler occurs from the setting of the IMR register of the INTC). This interrupt is a recoverable EI level interrupt.

In the case of an EIINT interrupt, the channel number where the interrupt input occurred is set to the SCR register. As a result, the channel number can be easily known when wishing to share the same interrupt vectors among several channels.

Caution Upon acknowledgment of the EI level interrupt, the priority level of the currently acknowledged interrupt is registered to the ISPR register (in-service priority register). Then, until execution of the EIRET instruction, interrupt with a priority level lower than that of this ISPR register are not generated. Interrupt request is acknowledged and held.
Registration of the priority level of the currently acknowledged interrupt and deletion of the priority level of the interrupt during EIRET to/from the ISPR register are automatically performed by the hardware. Write to the ISPR register cannot be performed by software. Write operations are ignored.

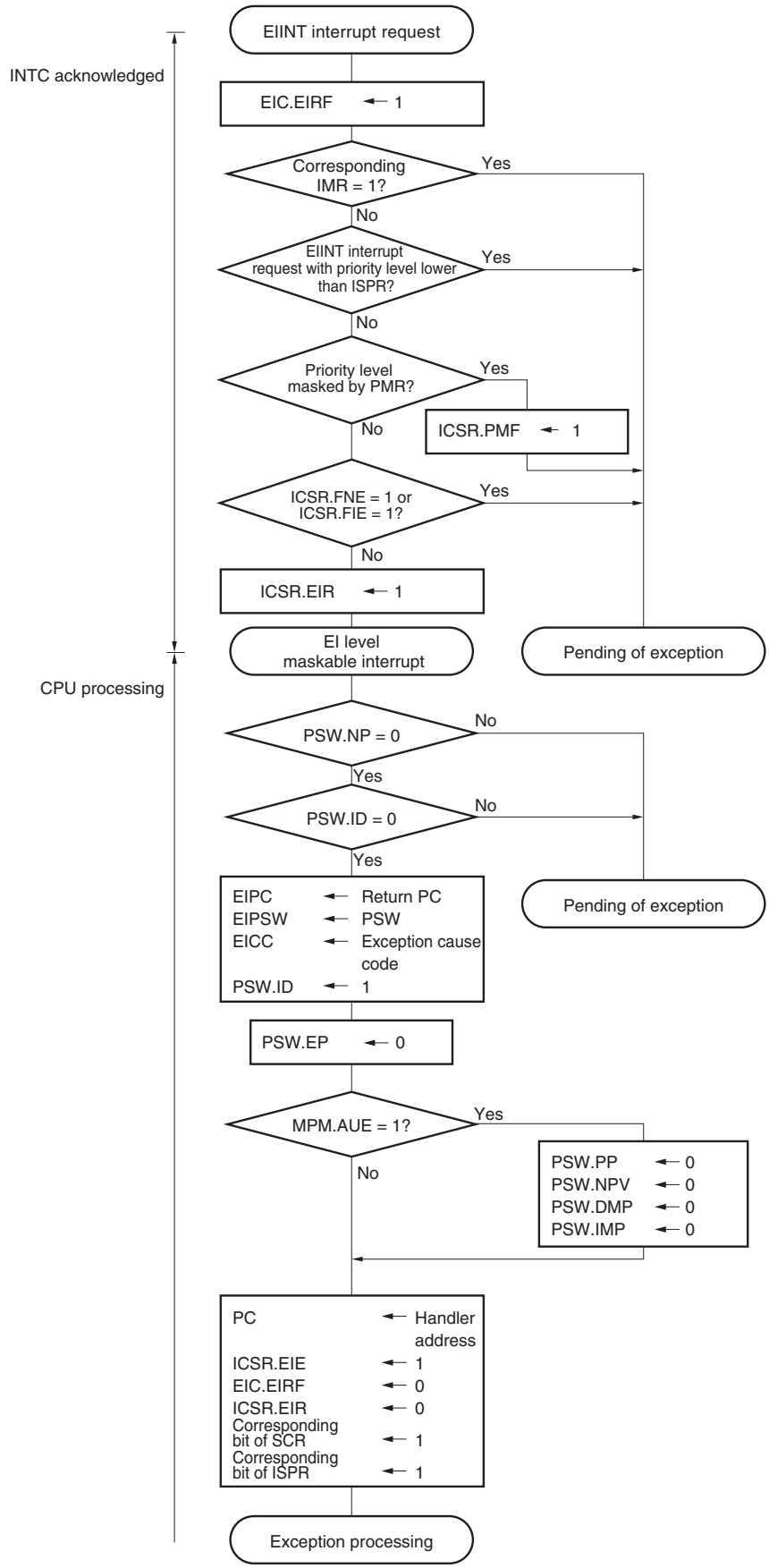
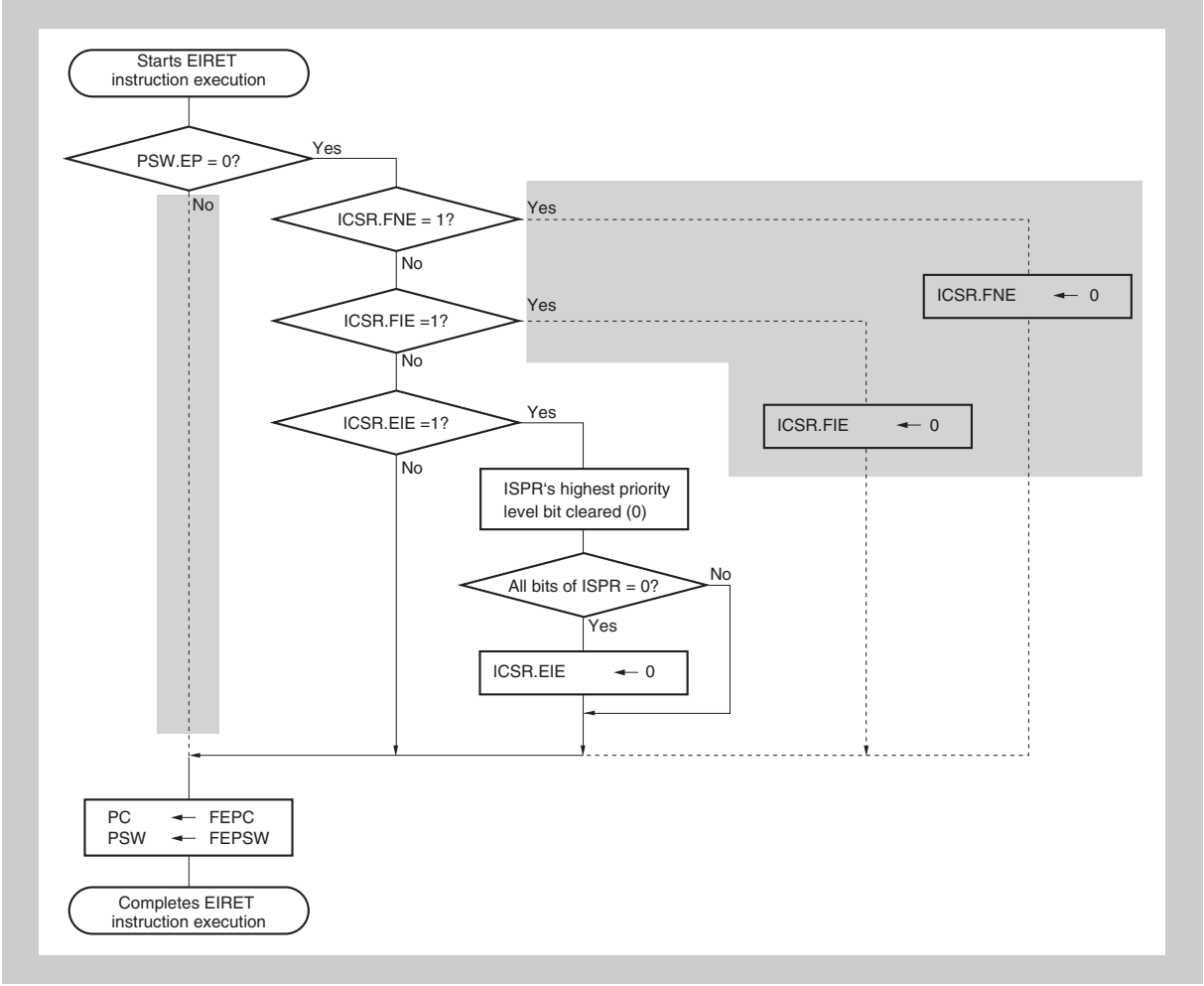


Figure 9-4 Processing upon occurrence of EIINT interrupt request

9.4.6 Restore from EI level maskable interrupt (EIINT)

Restore from EI level maskable interrupt (EIINT) is performed using the EIRET instruction. Execution of the EIRET instruction while the PSW.EP bit status is cleared (0) causes restore processing from the interrupt. Completely restoring from interrupt servicing when the PSW.EP bit is "1" is not possible (clearing of the ICSR, ISPR, and other registers is not performed). For return from EI level maskable interrupt, execute the EIRET instruction with the PSW.EP bit always cleared (0).

Caution In the V850E2M CPU core, although RETI instructions are provided for backward compatibility with V850E1 and V850E2 architectures, but their use is, in principle, prohibited. Replace all RETI instructions other than existing programs that cannot be modified with EIRET or FERET instructions.



Note Shaded portions indicate processing which do not branch during restoring from an EI level maskable interrupt (EIINT) servicing.

Figure 9-5 Restore from EI level maskable interrupt (EIINT)

9.5 Interrupt Operation

9.5.1 Mask function of EI level maskable interrupt (EIINT)

Interrupt masking can be specified for each respective interrupt channel of EIINT. Interrupt masking is performed by doing the following register settings.

EICn.EIMKn	Operation
1	Masks interrupt.
0	Enables interrupt.

The EICn.EIMKn bits can be read and written from the EIMKn bits corresponding to IMRm registers. They share the same registers.

[Operation example] (1) When "1" is written to IMRm.EIMKn bits, interrupts are prohibited for the corresponding channels.

(2) When the EICn.EIMKn bits are read, "1" is read.

Caution For EIMKn bit, the processing after interrupt hold is masked. Even if the EIMKn bit is set to 1, interrupt request acknowledgment and hold are performed. Therefore, even if software interrupts are requested for interrupts for which interrupt prohibit has been specified with EIMK, no interrupt occurs. Moreover, when EIMKn bit is again set to 0 while an interrupt request is held, that interrupts occur at that timing. To delete an interrupt request that is already being held, clear (0) the corresponding EIRFn bit.

9.5.2 Interrupt priority level judgment

When FENMI, FEINT, and EIINT interrupts are input, priorities including other exceptions are determined, and the exception with the highest priority (including interrupts) is requested. Exceptions requested at the same time (including interrupts) are processed in a pre-allocated priority order (the default priority order). The priority orders of FENMI, FEINT, and EIINT interrupts are as follows.

FENMI > FEINT > EIINT

(See *Table 9-1 "Exception cause list" on page 322* and the *V850E2M Architecture User's Manual* for other exceptions.)

For EIINT interrupts, the priority level can be set independently for each interrupt source. The priority level is specified with EIC0-EIC255.EIP3-EIP0. Priority levels from 0 to 15 can be set. 0 is the highest priority level, and 15 the lowest. In the case of multiple EIINT interrupts with the same priority level, the interrupt with the lowest interrupt channel number has priority.

Table 9-15 Example of EIINT interrupt priority level settings and priority levels

EIINT	EIP3 to EIP0 setting	Priority level during operation
EIINT0	3	10
EIINT1	4	11
EIINT2	0	1
EIINT3	0	2
EIINT4	1	3
EIINT5	2	6
EIINT6	2	7
EIINT7	1	4
EIINT8	1	5
EIINT9	2	8
EIINT10	2	9

During interrupt servicing, the interrupt controller also processes multiple interrupts acknowledging other interrupts.When multiple EIINT interrupts are requested at the same time, the interrupt to be acknowledged is determined with the following procedure.

(1) Comparison with the priority level as the interrupt currently being serviced

Interrupts with the same or lower priority level as the interrupt currently being serviced are held.

The priority level of the interrupt currently being serviced is held in the ISPR register.

Interrupts with a higher priority level than the interrupt currently being serviced are go on to the next priority judgment stage.

(2) Masking through priority mask register (PMR)

Only interrupts enabled by the PMR register go on to the next priority judgment stage.

(3) Of the requested interrupt sources, that with the highest priority level is selected.

When interrupts are being simultaneously requested from multiple sources, of the interrupt sources with the highest priority level, that with the smallest interrupt channel number is selected.

(4) Interrupt hold by CPU

Interrupt acknowledgment is pended according to the state of the NP and ID bits of the PSW register. At this time, priority judgment among EIINT interrupts, and priority judgment among EIINT interrupts, FEINT interrupts, and FENMI interrupts is performed even while interrupt acknowledgment is pended, and the interrupt with the highest priority is selected upon realization of the acknowledgment condition.

Example When a priority level 5 EIINT interrupt has already been requested and interrupt generation is pended because the value of the PSW.ID bit is "1", a subsequent priority level 3 EIINT interrupt is requested. Then, if the PSW.ID bit is cleared (0), the priority level 3 EIINT interrupt is generated.

Multiple interrupt servicing in which an interrupt is acknowledged while another interrupt is being serviced is shown below.

Figure 9-6 “Example of processing in which another interrupt request signal is issued while an interrupt is being serviced (1/2)”,*Figure 9-7 “Example of processing in which another interrupt request signal is issued while an interrupt is being serviced (2/2)”*.

When an interrupt request signal is acknowledged, the PSW.ID flag is automatically set to 1. Therefore, when multiple interrupts are to be used, clear the ID flag to 0 beforehand (for example, by placing the EI instruction in the interrupt service program) to set the interrupt enable mode.

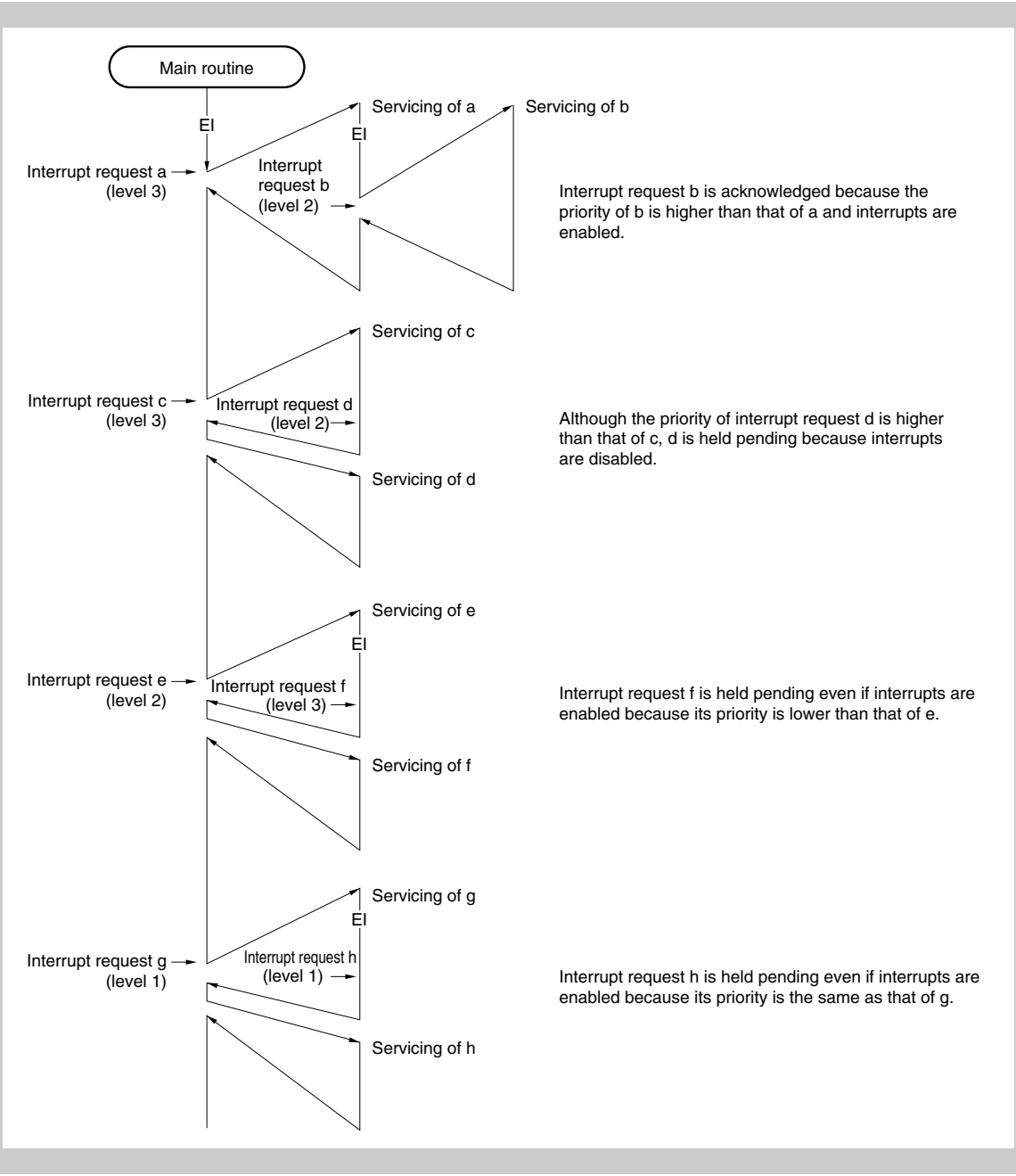


Figure 9-6 Example of processing in which another interrupt request signal is issued while an interrupt is being serviced (1/2)

Caution To perform multiple interrupt servicing, the values of the EIPC and EIPSW registers must be saved before executing the EI instruction. When returning from multiple interrupt servicing, restore the values of EIPC and EIPSW after executing the DI instruction.

Notes

1. a to u in the figure are the temporary names of interrupt request signals shown for the sake of explanation.
2. The default priority in the figure indicates the relative priority between two interrupt request signals.

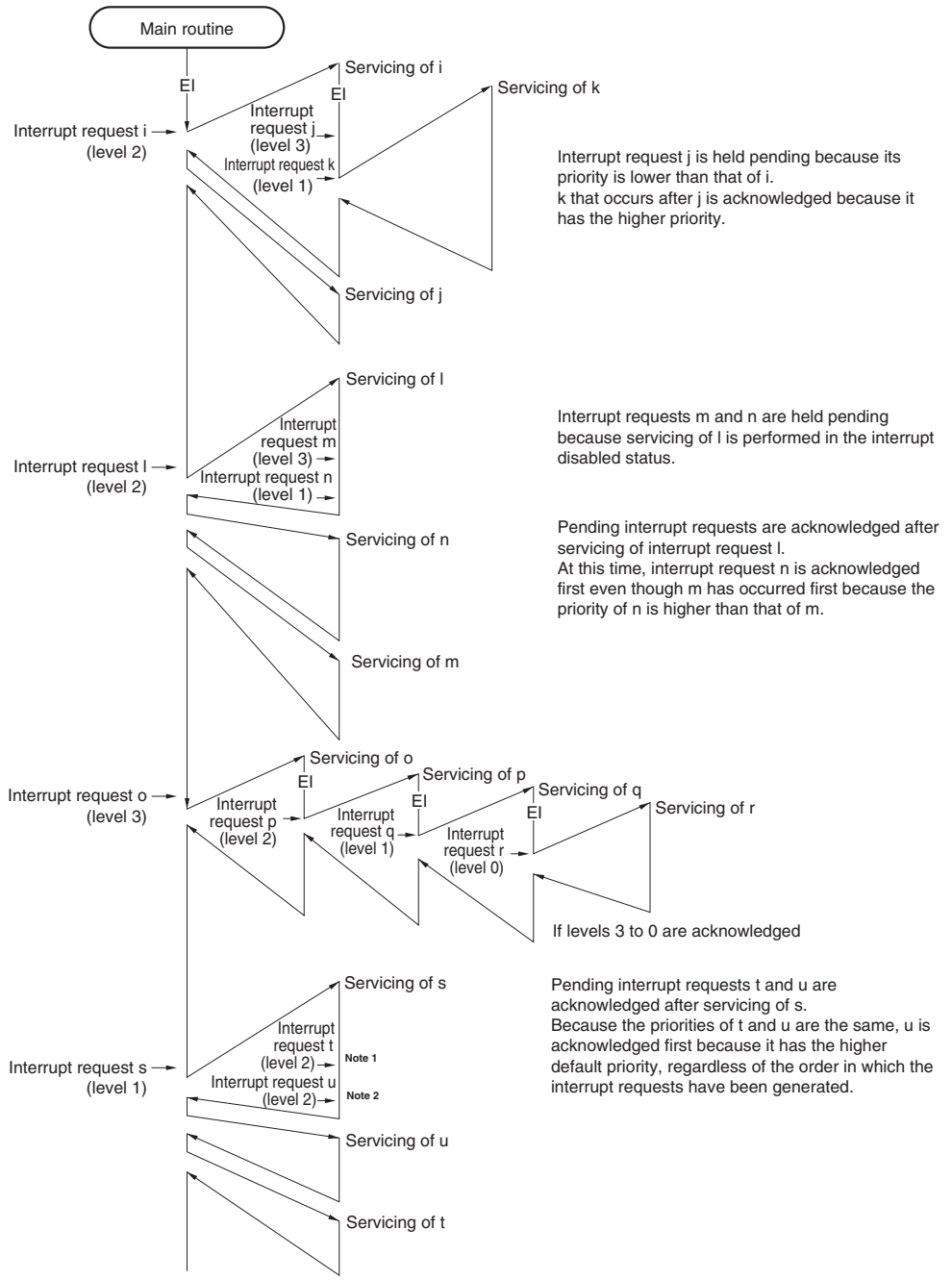


Figure 9-7 Example of processing in which another interrupt request signal is issued while an interrupt is being serviced (2/2)

Notes
1. Lower default priority.
2. Higher default priority.

Caution To perform multiple interrupt servicing, the values of the EIPC and EIPSW registers must be saved before executing the EI instruction. When returning from multiple interrupt servicing, restore the values of EIPC and EIPSW after executing the DI instruction.

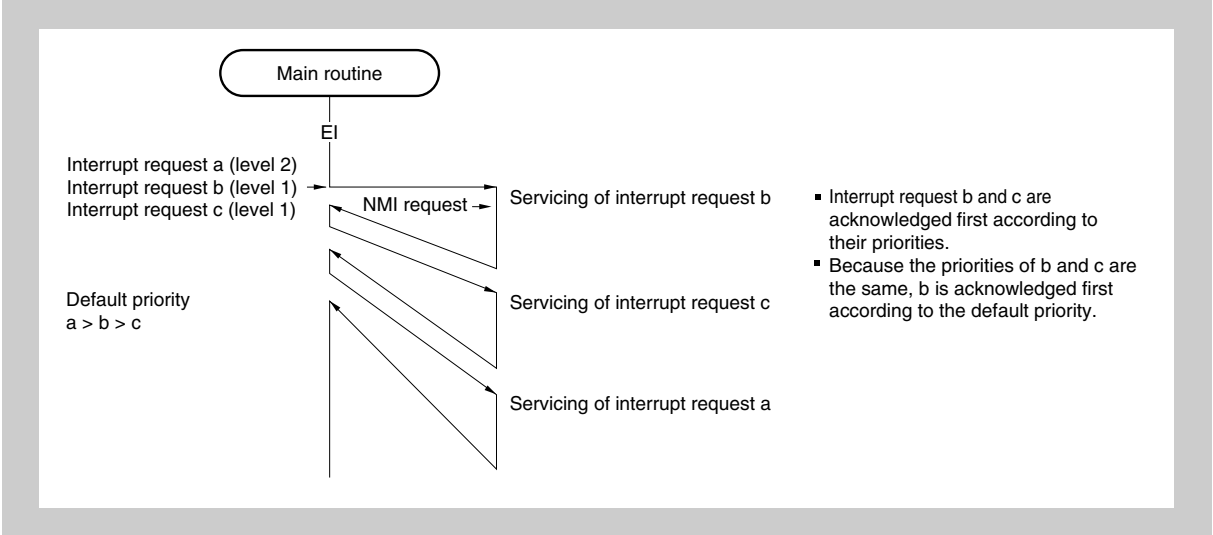


Figure 9-8 Example of servicing interrupt request signals simultaneously generated

Caution To perform multiple interrupt servicing, the values of the EIPC and EIPSW registers must be saved before executing the EI instruction. When returning from multiple interrupt servicing, restore the values of EIPC and EIPSW after executing the DI instruction.

Notes

1. a to c in the figure are the temporary names of interrupt request signals shown for the sake of explanation.
2. The default priority in the figure indicates the relative priority between two interrupt request signals.

9.5.3 Priority mask function

The priority mask function prohibits in batch EIINT interrupts of the specified interrupt priority level.

The interrupt masking priority level is specified with the PMR register. Masking and acknowledgment can be set for each priority level.

The following operations are possible using this function.

- Temporary prohibition of interrupts that have a priority level that is lower than a given priority level
- Temporary prohibition of interrupts that have a given priority level

PMR.PMRm	Operation
0	Acknowledges requests from priority level m interrupt source.
1	Masks requests from priority level m interrupt source.

Note m = 0 to 15

The PMR register prohibits interrupt occurrence. Interrupt request is acknowledged and held even while the interrupt occurrence is prohibited.

The presence of EIINT interrupts held pending with this function can be checked with the next section, *9.4.4 “Restore from FE level maskable interrupt (FEINT) servicing” on page 353*.

9.5.4 Pended interrupt report function

The state of the currently pended interrupt can be checked with the pended interrupt report function.

This function allows checking of the following states.

- When interrupts that are masked only by the priority mask function (PMR) exist
  The ICSR.PMF bit is set to 1.
  The ICSR.PMF bit is not set to 1 only when interrupts that are priority masked through ISPR register or interrupts masked through EIMK bit exist. Thus, the existence of priority requests pended through the priority mask function can be checked while interrupts are prohibited through priority masking.
- When EI level maskable interrupt request is not output to the CPU
  The ICSR.EIR bit is set to 1.
  By looking at the ICSR.EIR bit in the interval during which PSW.ID = 1, it is possible to check whether an EIINT interrupt request exists.
- When FE level maskable interrupt request is not output to the CPU
  ICSR.FIR bit is set to 1.
  By looking at the ICSR.FIR bit in the interval during which PSW.NP = 1, it is possible to check whether a FEINT interrupt request exists.

9.5.5 In-service priority clear function

This function initializes the internal status of the interrupt controller. It operates when the ISPC register is accessed. The following operations are possible using this function.

- Clear all contents of ISPR register
- Clear ICSR.EIE, FIE, and FNE bits

All the bits of ISPR register can be cleared to 0 by writing "1" to all the bits of this register and then writing "0" to all the bits of ISPR register. Moreover, the ICSR.EIE, FIE, and FNE bits, which all indicates the state in which an interrupt request is being processed in the CPU core, are all be cleared.

The value of this register is automatically cleared to 0 by writing 0 to all the bits of ISPR. The values of the bits of ISPR remain unchanged in the case of write access that is not performed simultaneously to all the bits.

9.6 Exception Handler Address Switching Function

Interrupt handler addresses can be switched by software.

For details, refer to *6.4 "Exception Handler Address Switching Function*" in *PART 2* of the *V850E2M Architecture User's Manual*.

Chapter 10 DMA Function

This chapter contains a generic description of the DMAC/DTS (DMA).

The first section describes all properties specific to the V850E2/MN4.

The subsequent sections describe the features that apply to all implementations.

10.1 V850E2/MN4 DMA Features

Instances

This microcontroller has the following number of DMAs.

Table 10-1 Instances of DMAC

	DMA
Instances	16

Instances index n

Throughout this chapter, the individual instances of DMAC are identified by the index “n” (n = 0 to 15).

(1) DMAC start sources

The DMAC start source can be selected by specifying DTFRn.IFCn[6:0].

The table below lists the DMA start sources that can be selected using the DTFRn register.

For details about the interrupts, see *Table 9-5 "EI level maskable interrupt requests" on page 326*.

For details about the transfer sources of the DTS function, see *10.12 "DTSFSL Function" on page 478*.

Table 10-2 DMA start sources (0 to 63)

DTFRn.IFCn[6:0]	Interrupt to start DMA
0	INTP0
1	INTP1
2	INTP2
3	INTP3
4	INTP4
5	INTP5
6	INTP6
7	INTP7
8	INTP8
9	INTP9
10	INTP10
11	INTP11
12	INTP12
13	INTP13
14	INTP14
15	INTP15
16	INTP16
17	INTP17
18	INTP18
19	INTP19
20	INTP20
21	INTP21
22	INTP22
23	INTP23
24	INTP24
25	INTP25
26	INTP26
27	INTP27
28	INTADCA0I0
29	INTADCA0I1
30	INTADCA0I2
31	Reserved

DTFRn.IFCn[6:0]	Interrupt to start DMA
32	INTTAUA0I0
33	INTTAUA0I1
34	INTTAUA0I2
35	INTTAUA0I3
36	INTTAUA0I4
37	INTTAUA0I5
38	INTTAUA0I6
39	INTTAUA0I7
40	INTTAUA0I8
41	INTTAUA0I9
42	INTTAUA0I10
43	INTTAUA0I11
44	INTTAUA0I12
45	INTTAUA0I13
46	INTTAUA0I14
47	INTTAUA0I15
48	INTTAUA1I0
49	INTTAUA1I1
50	INTTAUA1I2
51	INTTAUA1I3
52	INTTAUA1I4
53	INTTAUA1I5
54	INTTAUA1I6
55	INTTAUA1I7
56	INTTAUA1I8
57	INTTAUA1I9
58	INTTAUA1I10
59	INTTAUA1I11
60	INTTAUA1I12
61	INTTAUA1I13
62	INTTAUA1I14
63	INTTAUA1I15

Table 10-3 DMA start sources (64 to 127)

DTFRn.IFCn[6:0]	Interrupt to start DMA
64	INTTAUA2I12
65	INTTAUA2I13
66	INTTAUA2I14
67	INTTAUA2I15
68	INTTAUA3I0
69	INTTAUA3I1
70	INTTAUA3I2
71	INTTAUA3I3
72	INTTAUA3I4
73	INTTAUA3I5
74	INTTAUA3I6
75	INTTAUA3I7
76	INTTAUA3I8
77	INTTAUA3I9
78	INTTAUA3I10
79	INTTAUA3I11
80	INTTAUA3I12
81	INTTAUA3I13
82	INTTAUA3I14
83	INTTAUA3I15
84	INTTAUJ0I0
85	INTTAUJ0I1
86	INTTAUJ0I2
87	INTTAUJ0I3
88	INTENCA0IOV
89	INTENCA0IUD
90	INTENCA0I0
91	INTENCA0I1
92	INTENCA0IEC
93	INTENCA1IOV
94	INTENCA1IUD
95	INTENCA1I0

DTFRn.IFCn[6:0]	Interrupt to start DMA
96	INTENCA1I1
97	INTENCA1IEC
98	Reserved
99	Reserved
100	INTTAPA2ADOUT0
101	INTTAPA0ADOUT0
102	INTTAPA0ADOUT1
103	Reserved
104	Reserved
105	INTTAPA3ADOUT0
106	INTTAPA1ADOUT0
107	INTTAPA1ADOUT1
108	INTCSIH0IR (IICB0 only)
109	Reserved
110	INTCSIH1IR (IICB1 only)
111	Reserved
112	INTCSIH2IR (IICB2 only)
113	Reserved
114	INTCSIH3IR (IICB3 only)
115	Reserved
116	INTCSIG0IR
117	INTCSIG0IC
118	INTCSIG1IR
119	INTCSIG1IC
120	INTCSIG2IR
121	INTCSIG2IC
122	INTCSIG3IR
123	INTCSIG3IC
124	INTCSIG4IR
125	INTCSIG4IC
126	INTCSIG5IR
127	INTCSIG5IC

10.2 Terms

The terms used in this chapter are defined as follows.

Table 10-4 Definition of terms

Term	Function
DMA transfer	Period from the start of the first DMA cycle to assertion of INTDMA
DMA cycle	Period of transferring one unit of data (since a read cycle of the internal system bus has been started and until a write cycle is completed. In the case of 128-bit transfer, until the read cycle is completed four times and the write cycle is completed four times)
Hardware DMA transfer request	DMA transfer request by external pins
Software DMA transfer request	DMA transfer request by internal register (DTSnSR bit of DTSn register)
DMA transfer request	Hardware DMA transfer request or software DMA transfer request
Transfer information (TI)	Information required for DMA transfer, such as transfer address, transfer data size, and transfer count. In particular, the DTS's transfer information is called TI (Transfer Information).
TI fetch	When DTS reads transfer information from internal RAM
TI write back	When DTS writes transfer information back internal RAM
TI fetch cycle	Period during which DTS reads transfer information from internal RAM
DTS cycle	Period during which DTS performs a transfer according to the TI
TI write back cycle	Period during which DTS writes transfer information back to internal RAM
Single transfer	In the case of DMAC, one DMA cycle is executed per transfer request. In the case of DTS, one cycle of DTS operations that include a TI fetch cycle, DTS cycle, and TI write back cycle is executed per transfer request.
Single step transfer	This function is used only with DMAC. The number of transfers set in the transfer count setting register (DTC) is executed per software DMA transfer request. Since the bus is released for each transfer, the CPU is able to generate interrupts. If a higher-priority transfer request occurs during execution of a single step transfer, the single step transfer is aborted while the higher-priority transfer request is executed.
Block transfer	This function is used only with DTS. For each transfer request, one TI fetch cycle, the DTS cycles for the number specified in the transfer count setting register (TI-A's bits [31:16]), and one TI write back cycle are executed. Since the bus is released for each transfer, the CPU is able to generate interrupts. If a higher-priority DTS request occurs during execution of a block transfer, DTS ignores (cannot acknowledge) the request until the block transfer is completed.

10.3 General

Direct memory access (DMA) function is used to access data without going via the CPU.

The V850E2/MN4 incorporates five types of DMA subsystem units: DTFR, DTSFSL, DMAC (two units), DTS, and DMAT. DMAC can be used to perform high-speed data transfers.

DTS provides slower data transfers, chain functions, etc.

DMAT drives the internal system bus for data access. DTFR and DTSFSL have functions that select DMA transfer factors from among interrupt requests.

10.3.1 DMA controller (DMAC) function

- Registers to store transfer information (transfer address, transfer size, etc.) and registers to control DMAC are included.
- When a DMA transfer request is accepted, a transfer request is output to DMAT, according to the contained transfer information.
- Hardware DMA transfer requests, DMA acknowledge signals, and DMA transfer completion interrupts are input and output.
- Write back information is written back to registers.

10.3.2 DMA trigger factor register (DTFR) function

- DMA transfer factors are selected from among interrupt signals. (16 channels are selected from 128-channel interrupt signals.)

10.3.3 Data transfer service (DTS) function

- A register that controls DTS is included (transfer information is set in internal RAM).
- When a DTS transfer request is accepted, transfer information set in internal RAM is read, and a transfer request is output to DMAT, according to the transfer information.
- Hardware DTS transfer requests, DTS acknowledge signals, and DTS transfer completion interrupts are output.
- Write back information (next time's transfer information) is written back to internal RAM.

10.3.4 DTS factor selector (DTSFSL) function

- DTS transfer factors are selected from among interrupt signals. (Up to 128 transfer factors can be assigned.)
- Up to four priority levels can be set.
- Various interrupt sources from the DTS and interrupt signals from peripherals are selected.

Table 10-5 Transfer target spaces (DMAC)

Transfer source	Transfer destination					
	P-bus peripheral I/O	External memory (primary memory controller)	Internal RAM1	Internal RAM2[a]	Internal flash memory	H-bus peripheral I/O
P-bus peripheral I/O	√	√	√	√	×	√
External memory (primary memory controller)	√	√	√	√	×	√
Internal RAM1	√	√	√	√	×	√
Internal RAM2[a]	√	√	√	√	×	√
Internal flash memory	√	√	√	√	×	√
H-bus peripheral I/O	√	√	√	√	×	√

a) Available for dual core products (μPD70F3514 and 70F3515) only.

Caution Be careful not to specify the same address as the transfer source and transfer destination. Doing so might corrupt the transfer data.

Table 10-6 Transfer target spaces (DTS)

Transfer source	Transfer destination					
	P-bus peripheral I/O	External memory (primary memory controller)	Internal RAM1	Internal RAM2[a]	Internal flash memory	H-bus peripheral I/O
P-bus peripheral I/O	√	√	√	×	×	√
External memory (primary memory controller)	√	√	√	×	×	√
Internal RAM1	√	√	√	×	×	√
Internal RAM2[a]	×	×	×	×	×	×
Internal flash memory	√	√	√	×	×	√
H-bus peripheral I/O	√	√	√	×	×	√

a) Available for dual core products (μPD70F3514 and 70F3515) only.

10.3.5 DMA access memory map

The memory map that shows areas that can be accessed by DMA is shown below.

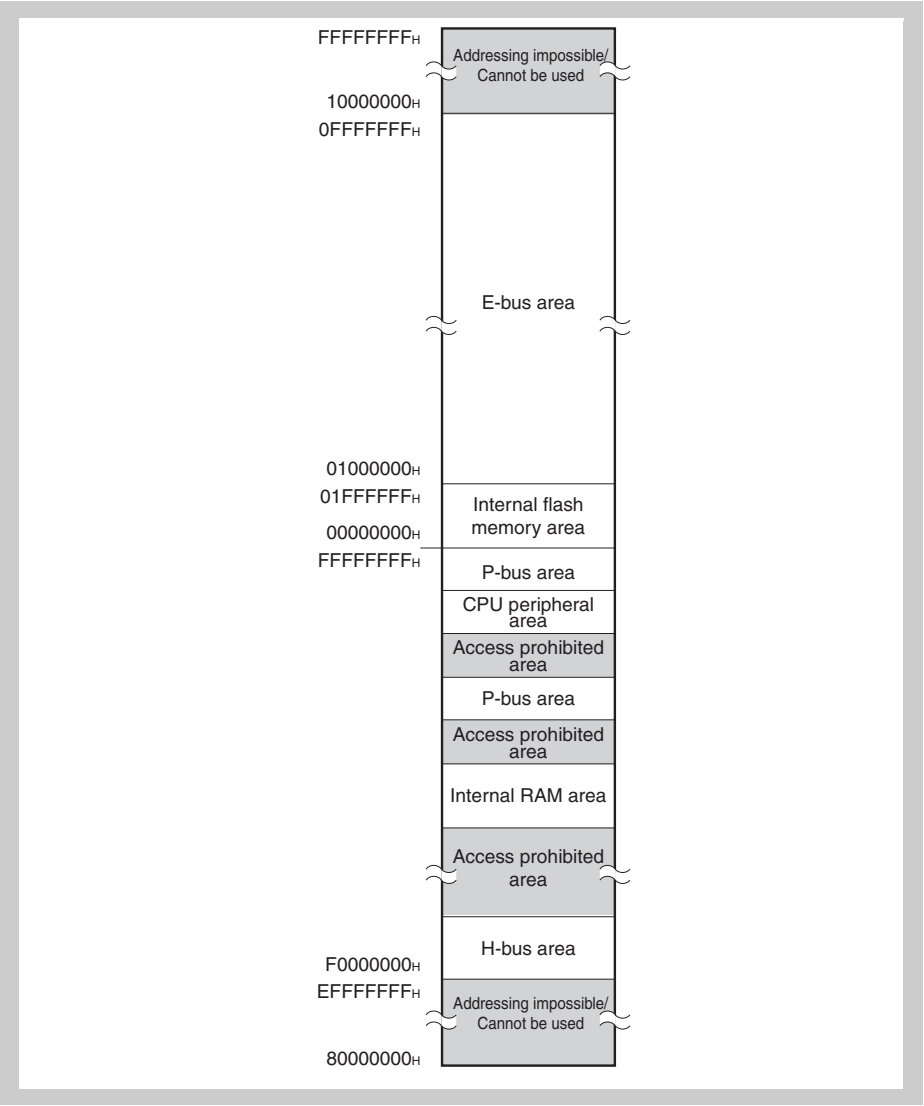


Figure 10-1 Memory map from DMA point of view

10.3.6 Prioritization of channels

The following describes how the prioritization of the DMA subsystem's various transfer channels is determined. Prioritization is determined via two stages: the first stage is when the priority level in each group among DMAC0, DMAC1, and DTS is determined separately.

Among DMAC0, the priority is set as CH0 > CH1 > CH2 > CH3 > CH4 > CH5 > CH6 > CH7, in which CH0 has the highest priority.

Among DMAC1, the priority is set as CH8 > CH9 > CH10 > CH11 > CH12 > CH13 > CH14 > CH15, in which CH8 has the highest priority.

Among DTS, where four levels of priority settings are possible, the priority is determined using the default priority channels: CH000 > CH001 > CH002... > CH127.

10.3.7 Arbitration of transfer requests

Since the DMA subsystem includes three transfer controllers (DMAC0, DMAC1, and DTS), these transfer requests are arbitrated internally. This arbitration is always performed using the fixed prioritization: DMAC0 > DMAC1 > DTS. DMAC0's priority level is the highest.

However, when DMAC0, DMAC1, or DTS completes one transfer request, the request is temporarily withdrawn, so the following phenomena occur.

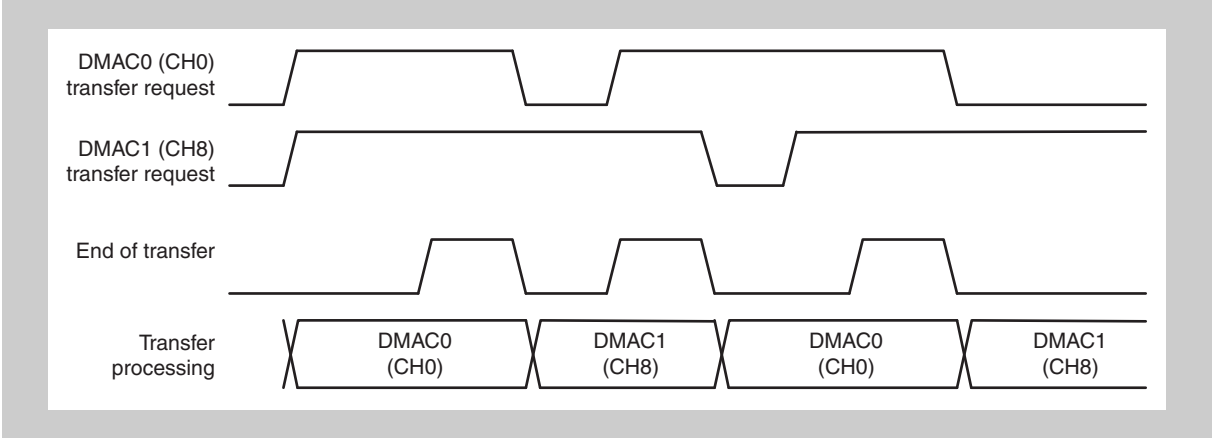


Figure 10-2 Alternation of DMAC0 and DMAC1

Even when CH0's transfer requests occur continuously, the DMAC temporarily withdraws transfer requests. As a result, DMAC1's requests take priority.

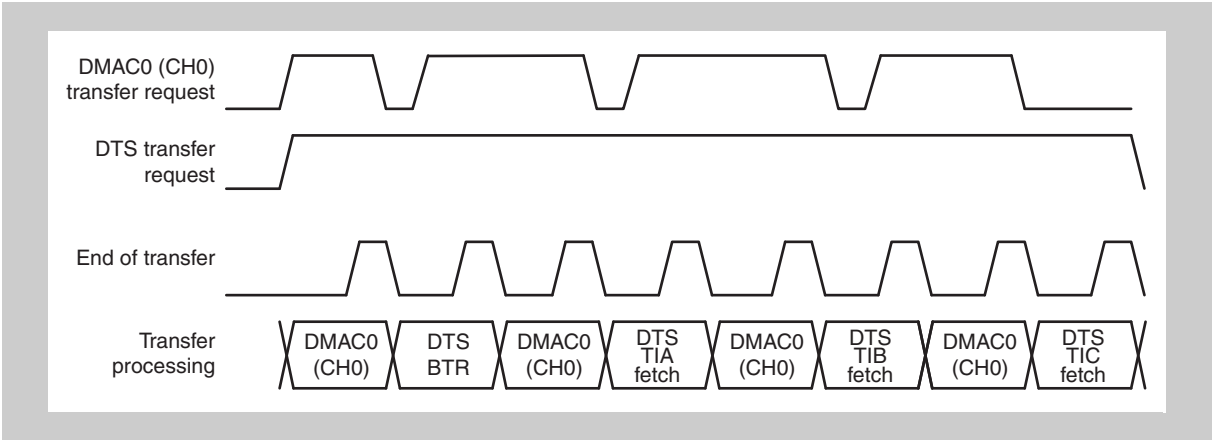


Figure 10-3 Alternation of DMAC0 and DTS

Even when CH0's transfer requests occur continuously, the DMAC0 temporarily withdraws transfer requests. As a result, DTS's requests take priority. After a base table read (BTR) cycle, DTS continues to output TIA fetch requests, but since DMAC0 has a higher priority level than DTS, DMAC0 takes priority as a result of arbitration. When DMAC0's transfer is completed, DMAC0 temporarily withdraws requests, and DTS's request is executed.

10.4 DMAC Function

10.4.1 Features

Number of channels

8 channels x 2 groups

Transfer data size

8 bits, 16 bits, 32 bits, and 128 bits

Transfer data

Fixed to little endian

Misaligned data not supported

Maximum transfer count

32,768 (2^{15}) times (MSB of the 16-bit register is used for the next address function.)

Channel priority control

Fixed priority (highest priority (CH0) → lowest priority (CH15))

Subject to transfer

Internal flash memory, internal RAM, external memory area, P-bus peripheral I/O area, and H-bus peripheral I/O area

DMA transfer cycle output function

$\overline{\text{DMAAK}}$[0:5] pin outputs (CH5 to CH0)

DMA last transfer cycle output function

$\overline{\text{DMAAK}}$[0:5] pin outputs (CH5 to CH0)

Transfer type

2-cycle transfer (dual address transfer)

The address at both the transfer source and destination is accessed. Two bus cycles are required to execute transfer once (read cycle + write cycle). Because the bus is not locked between the read cycle and write cycle, the CPU cycle may interrupt. When a 128-bit access is made, the write cycle is executed four times after the read cycle has been executed four times. Because the bus is not locked between the read cycles and between the write cycles, the CPU cycle may interrupt.

Transfer mode

- Single transfer mode (when hardware DMA transfer request is generated)

  When a hardware DMA transfer request is generated, the bus mastership is acquired, and the bus is always released after transfer has been executed once. If another hardware DMA transfer request is generated after that, transfer is executed once again. This operation is repeated until transfer has been executed the number of times specified by the transfer count register (DTC).

- Single-step transfer mode (when software DMA transfer request is generated)

  When a software DMA transfer request is generated, the bus mastership is acquired, and the bus is released each time transfer has been executed once. Once a software DMA transfer request has been acknowledged, this operation is repeated until transfer has been executed the number of times specified by the transfer count register (DTC).

Transfer address control	Incremental, decremental, and fixed
Transfer error support	When the data from the transfer source contains an error, or if an error occurs at the transfer source, DMA transfer is aborted, and a SysError exception is output for the CPU.
DMA transfer request	A hardware DMA transfer request or a software DMA transfer request can be selected for each channel (by setting the DTRS register). The software DMA transfer request can be set by software (by setting the DTS register). This register also has a status bit (DTS register) that indicates that a hardware DMA transfer request has been generated. The function for notifying the INTC about peripheral I/O interrupts not assigned to DMA transfer requests depends on the system configuration. Refer to the product's user's manual.
Transfer count match interrupt output function	This function has a transfer count compare register (DTCC) for each channel and outputs an interrupt signal (INTCT15 to INTCT0) upon a match with the transfer count register (DTC) of each channel.
Transfer completion interrupt output function	This function outputs a transfer completion interrupt signal (INTDMA15 to INTDMA0) when DMA transfer of each channel has been completed the number of times specified by the transfer count register (DTC).
Next address setting function	This function has a register for each channel for setting the transfer address and transfer count (Current) of the DMA transfer currently being executed, and a register for setting the transfer address and transfer count (Next) of the DMA transfer to be executed next, following completion of the DMA transfer currently being executed. This function also has a bit for each register for setting whether to copy Next to Current upon completion of DMA transfer.
DMA transfer abort function	This function supports aborting DMA transfer by software.

10.4.2 DMAC setting registers

Table 10-7 DMAC setting registers (1/10)

Address	Symbol	Function register name	R/W	Operable bits				Initial value
				1	8	16	32	
$FFFF7300_H$	DTRC0	DMA transfer request control register 0	R/W	√	√			00_H
$FFFF7310_H$	DTRS0	DMA transfer request select register CH0				√		0000_H
$FFFF7314_H$	DSA0	DMA source address register CH0					√	00000000_H
$FFFF7314_H$	DSA0L	DMA source address register LCH0				√		0000_H
$FFFF7316_H$	DSA0H	DMA source address register HCH0				√		0000_H
$FFFF7318_H$	DSC0	DMA source chip select register CH0				√		0001_H
$FFFF731C_H$	DNSA0	DMA next source address register CH0					√	00000000_H
$FFFF731C_H$	DNSA0L	DMA next source address register LCH0				√		0000_H
$FFFF731E_H$	DNSA0H	DMA next source address register HCH0				√		0000_H
$FFFF7320_H$	DNSC0	DMA next source chip select register CH0				√		0001_H
$FFFF7324_H$	DDA0	DMA destination address register CH0					√	00000000_H
$FFFF7324_H$	DDA0L	DMA destination address register LCH0				√		0000_H
$FFFF7326_H$	DDA0H	DMA destination address register HCH0				√		0000_H
$FFFF7328_H$	DDC0	DMA destination chip select register CH0				√		0001_H
$FFFF732C_H$	DNDA0	DMA next destination address register CH0					√	00000000_H
$FFFF732C_H$	DNDA0L	DMA next destination address register LCH0				√		0000_H
$FFFF732E_H$	DNDA0H	DMA next destination address register HCH0				√		0000_H
$FFFF7330_H$	DNDC0	DMA next destination chip select register CH0				√		0001_H
$FFFF7332_H$	DTC0	DMA transfer count register CH0				√		0000_H
$FFFF7334_H$	DNTC0	DMA next transfer count register CH0				√		0000_H
$FFFF7336_H$	DTCC0	DMA transfer count compare register CH0				√		0000_H
$FFFF7338_H$	DTCT0	DMA transfer control register CH0				√		0000_H
$FFFF733A_H$	DTS0	DMA transfer status register CH0		√	√			00_H
$FFFF7340_H$	DTRS1	DMA transfer request select register CH1				√		0000_H
$FFFF7344_H$	DSA1	DMA source address register CH1					√	00000000_H
$FFFF7344_H$	DSA1L	DMA source address register LCH1				√		0000_H
$FFFF7346_H$	DSA1H	DMA source address register HCH1				√		0000_H
$FFFF7348_H$	DSC1	DMA source chip select register CH1				√		0001_H
$FFFF734C_H$	DNSA1	DMA next source address register CH1					√	00000000_H
$FFFF734C_H$	DNSA1L	DMA next source address register LCH1				√		0000_H
$FFFF734E_H$	DNSA1H	DMA next source address register HCH1				√		0000_H
$FFFF7350_H$	DNSC1	DMA next source chip select register CH1				√		0001_H
$FFFF7354_H$	DDA1	DMA destination address register CH1					√	00000000_H
$FFFF7354_H$	DDA1L	DMA destination address register LCH1				√		0000_H
$FFFF7356_H$	DDA1H	DMA destination address register HCH1				√		0000_H
$FFFF7358_H$	DDC1	DMA destination chip select register CH1				√		0001_H

Table 10-7 DMAC setting registers (2/10)

Address	Symbol	Function register name	R/W	Operable bits				Initial value
				1	8	16	32	
$FFFF735C_H$	DNDA1	DMA next destination address register CH1	R/W				√	00000000_H
$FFFF735C_H$	DNDA1L	DMA next destination address register LCH1				√		0000_H
$FFFF735E_H$	DNDA1H	DMA next destination address register HCH1				√		0000_H
$FFFF7360_H$	DNDC1	DMA next destination chip select register CH1				√		0001_H
$FFFF7362_H$	DTC1	DMA transfer count register CH1				√		0000_H
$FFFF7364_H$	DNTC1	DMA next transfer count register CH1				√		0000_H
$FFFF7366_H$	DTCC1	DMA transfer count compare register CH1				√		0000_H
$FFFF7368_H$	DTCT1	DMA transfer control register CH1				√		0000_H
$FFFF736A_H$	DTS1	DMA transfer status register CH1		√	√			00_H
$FFFF7370_H$	DTRS2	DMA transfer request select register CH2				√		0000_H
$FFFF7374_H$	DSA2	DMA source address register CH2					√	00000000_H
$FFFF7374_H$	DSA2L	DMA source address register LCH2				√		0000_H
$FFFF7376_H$	DSA2H	DMA source address register HCH2				√		0000_H
$FFFF7378_H$	DSC2	DMA source chip select register CH2				√		0001_H
$FFFF737C_H$	DNSA2	DMA next source address register CH2					√	00000000_H
$FFFF737C_H$	DNSA2L	DMA next source address register LCH2				√		0000_H
$FFFF737E_H$	DNSA2H	DMA next source address register HCH2				√		0000_H
$FFFF7380_H$	DNSC2	DMA next source chip select register CH2				√		0001_H
$FFFF7384_H$	DDA2	DMA destination address register CH2					√	00000000_H
$FFFF7384_H$	DDA2L	DMA destination address register LCH2				√		0000_H
$FFFF7386_H$	DDA2H	DMA destination address register HCH2				√		0000_H
$FFFF7388_H$	DDC2	DMA destination chip select register CH2				√		0001_H
$FFFF738C_H$	DNDA2	DMA next destination address register CH2					√	00000000_H
$FFFF738C_H$	DNDA2L	DMA next destination address register LCH2				√		0000_H
$FFFF738E_H$	DNDA2H	DMA next destination address register HCH2				√		0000_H
$FFFF7390_H$	DNDC2	DMA next destination chip select register CH2				√		0001_H
$FFFF7392_H$	DTC2	DMA transfer count register CH2				√		0000_H
$FFFF7394_H$	DNTC2	DMA next transfer count register CH2				√		0000_H
$FFFF7396_H$	DTCC2	DMA transfer count compare register CH2				√		0000_H
$FFFF7398_H$	DTCT2	DMA transfer control register CH2				√		0000_H
$FFFF739A_H$	DTS2	DMA transfer status register CH2		√	√			00_H
$FFFF73A0_H$	DTRS3	DMA transfer request select register CH3				√		0000_H
$FFFF73A4_H$	DSA3	DMA source address register CH3					√	00000000_H
$FFFF73A4_H$	DSA3L	DMA source address register LCH3				√		0000_H
$FFFF73A6_H$	DSA3H	DMA source address register HCH3				√		0000_H
$FFFF73A8_H$	DSC3	DMA source chip select register CH3				√		0001_H

Table 10-7 DMAC setting registers (3/10)

Address	Symbol	Function register name	R/W	Operable bits				Initial value
				1	8	16	32	
$FFFF73AC_H$	DNSA3	DMA next source address register CH3	R/W				√	00000000_H
$FFFF73AC_H$	DNSA3L	DMA next source address register LCH3				√		0000_H
$FFFF73AE_H$	DNSA3H	DMA next source address register HCH3				√		0000_H
$FFFF73B0_H$	DNSC3	DMA next source chip select register CH3				√		0001_H
$FFFF73B4_H$	DDA3	DMA destination address register CH3					√	00000000_H
$FFFF73B4_H$	DDA3L	DMA destination address register LCH3				√		0000_H
$FFFF73B6_H$	DDA3H	DMA destination address register HCH3				√		0000_H
$FFFF73B8_H$	DDC3	DMA destination chip select register CH3				√		0001_H
$FFFF73BC_H$	DNDA3	DMA next destination address register CH3					√	00000000_H
$FFFF73BC_H$	DNDA3L	DMA next destination address register LCH3				√		0000_H
$FFFF73BE_H$	DNDA3H	DMA next destination address register HCH3				√		0000_H
$FFFF73C0_H$	DNDC3	DMA next destination chip select register CH3				√		0001_H
$FFFF73C2_H$	DTC3	DMA transfer count register CH3				√		0000_H
$FFFF73C4_H$	DNTC3	DMA next transfer count register CH3				√		0000_H
$FFFF73C6_H$	DTCC3	DMA transfer count compare register CH3				√		0000_H
$FFFF73C8_H$	DTCT3	DMA transfer control register CH3				√		0000_H
$FFFF73CA_H$	DTS3	DMA transfer status register CH3		√	√			00_H
$FFFF73D0_H$	DTRS4	DMA transfer request select register CH4				√		0000_H
$FFFF73D4_H$	DSA4	DMA source address register CH4					√	00000000_H
$FFFF73D4_H$	DSA4L	DMA source address register LCH4				√		0000_H
$FFFF73D6_H$	DSA4H	DMA source address register HCH4				√		0000_H
$FFFF73D8_H$	DSC4	DMA source chip select register CH4				√		0001_H
$FFFF73DC_H$	DNSA4	DMA next source address register CH4					√	00000000_H
$FFFF73DC_H$	DNSA4L	DMA next source address register LCH4				√		0000_H
$FFFF73DE_H$	DNSA4H	DMA next source address register HCH4				√		0000_H
$FFFF73E0_H$	DNSC4	DMA next source chip select register CH4				√		0001_H
$FFFF73E4_H$	DDA4	DMA destination address register CH4					√	00000000_H
$FFFF73E4_H$	DDA4L	DMA destination address register LCH4				√		0000_H
$FFFF73E6_H$	DDA4H	DMA destination address register HCH4				√		0000_H
$FFFF73E8_H$	DDC4	DMA destination chip select register CH4				√		0001_H
$FFFF73EC_H$	DNDA4	DMA next destination address register CH4					√	00000000_H
$FFFF73EC_H$	DNDA4L	DMA next destination address register LCH4				√		0000_H
$FFFF73EE_H$	DNDA4H	DMA next destination address register HCH4				√		0000_H
$FFFF73F0_H$	DNDC4	DMA next destination chip select register CH4				√		0001_H
$FFFF73F2_H$	DTC4	DMA transfer count register CH4				√		0000_H
$FFFF73F4_H$	DNTC4	DMA next transfer count register CH4				√		0000_H

Table 10-7 DMAC setting registers (4/10)

Address	Symbol	Function register name	R/W	Operable bits				Initial value
				1	8	16	32	
FFFF73F6$_H$	DTCC4	DMA transfer count compare register CH4	R/W			√		0000$_H$
FFFF73F8$_H$	DTCT4	DMA transfer control register CH4				√		0000$_H$
FFFF73FA$_H$	DTS4	DMA transfer status register CH4		√	√			00$_H$
FFFF7400$_H$	DTRS5	DMA transfer request select register CH5				√		0000$_H$
FFFF7404$_H$	DSA5	DMA source address register CH5					√	00000000$_H$
FFFF7404$_H$	DSA5L	DMA source address register LCH5				√		0000$_H$
FFFF7406$_H$	DSA5H	DMA source address register HCH5				√		0000$_H$
FFFF7408$_H$	DSC5	DMA source chip select register CH5				√		0001$_H$
FFFF740C$_H$	DNSA5	DMA next source address register CH5					√	00000000$_H$
FFFF740C$_H$	DNSA5L	DMA next source address register LCH5				√		0000$_H$
FFFF740E$_H$	DNSA5H	DMA next source address register HCH5				√		0000$_H$
FFFF7410$_H$	DNSC5	DMA next source chip select register CH5				√		0001$_H$
FFFF7414$_H$	DDA5	DMA destination address register CH5					√	00000000$_H$
FFFF7414$_H$	DDA5L	DMA destination address register LCH5				√		0000$_H$
FFFF7416H	DDA5H	DMA destination address register HCH5				√		0000$_H$
FFFF7418$_H$	DDC5	DMA destination chip select register CH5				√		0001$_H$
FFFF741C$_H$	DNDA5	DMA next destination address register CH5					√	00000000$_H$
FFFF741C$_H$	DNDA5L	DMA next destination address register LCH5				√		0000$_H$
FFFF741E$_H$	DNDA5H	DMA next destination address register HCH5				√		0000$_H$
FFFF7420$_H$	DNDC5	DMA next destination chip select register CH5				√		0001$_H$
FFFF7422$_H$	DTC5	DMA transfer count register CH5				√		0000$_H$
FFFF7424$_H$	DNTC5	DMA next transfer count register CH5				√		0000$_H$
FFFF7426$_H$	DTCC5	DMA transfer count compare register CH5				√		0000$_H$
FFFF7428$_H$	DTCT5	DMA transfer control register CH5				√		0000$_H$
FFFF742A$_H$	DTS5	DMA transfer status register CH5		√	√			00$_H$
FFFF7430$_H$	DTRS6	DMA transfer request select register CH6				√		0000$_H$
FFFF7434$_H$	DSA6	DMA source address register CH6					√	00000000$_H$
FFFF7434$_H$	DSA6L	DMA source address register LCH6				√		0000$_H$
FFFF7436$_H$	DSA6H	DMA source address register HCH6				√		0000$_H$
FFFF7438$_H$	DSC6	DMA source chip select register CH6				√		0001$_H$
FFFF743C$_H$	DNSA6	DMA next source address register CH6					√	00000000$_H$
FFFF743C$_H$	DNSA6L	DMA next source address register LCH6				√		0000$_H$
FFFF743E$_H$	DNSA6H	DMA next source address register HCH6				√		0000$_H$
FFFF7440$_H$	DNSC6	DMA next source chip select register CH6				√		0001$_H$
FFFF7444$_H$	DDA6	DMA destination address register CH6					√	00000000$_H$
FFFF7444$_H$	DDA6L	DMA destination address register LCH6				√		0000$_H$
FFFF7446$_H$	DDA6H	DMA destination address register HCH6				√		0000$_H$
FFFF7448$_H$	DDC6	DMA destination chip select register CH6				√		0001$_H$

Table 10-7 DMAC setting registers (5/10)

Address	Symbol	Function register name	R/W	Operable bits				Initial value
				1	8	16	32	
FFFF744C$_H$	DNDA6	DMA next destination address register CH6	R/W				√	00000000$_H$
FFFF744C$_H$	DNDA6L	DMA next destination address register LCH6				√		0000$_H$
FFFF744E$_H$	DNDA6H	DMA next destination address register HCH6				√		0000$_H$
FFFF7450$_H$	DNDC6	DMA next destination chip select register CH6				√		0001$_H$
FFFF7452$_H$	DTC6	DMA transfer count register CH6				√		0000$_H$
FFFF7454$_H$	DNTC6	DMA next transfer count register CH6				√		0000$_H$
FFFF7456$_H$	DTCC6	DMA transfer count compare register CH6				√		0000$_H$
FFFF7458$_H$	DTCT6	DMA transfer control register CH6				√		0000$_H$
FFFF745A$_H$	DTS6	DMA transfer status register CH6		√	√			00$_H$
FFFF7460$_H$	DTRS7	DMA transfer request select register CH7				√		0000$_H$
FFFF7464$_H$	DSA7	DMA source address register CH7					√	00000000$_H$
FFFF7464$_H$	DSA7L	DMA source address register LCH7				√		0000$_H$
FFFF7466$_H$	DSA7H	DMA source address register HCH7				√		0000$_H$
FFFF7468$_H$	DSC7	DMA source chip select register CH7				√		0001$_H$
FFFF746C$_H$	DNSA7	DMA next source address register CH7					√	00000000$_H$
FFFF746C$_H$	DNSA7L	DMA next source address register LCH7				√		0000$_H$
FFFF746E$_H$	DNSA7H	DMA next source address register HCH7				√		0000$_H$
FFFF7470$_H$	DNSC7	DMA next source chip select register CH7				√		0001$_H$
FFFF7474$_H$	DDA7	DMA destination address register CH7					√	00000000$_H$
FFFF7474$_H$	DDA7L	DMA destination address register LCH7				√		0000$_H$
FFFF7476$_H$	DDA7H	DMA destination address register HCH7				√		0000$_H$
FFFF7478$_H$	DDC7	DMA destination chip select register CH7				√		0001$_H$
FFFF747C$_H$	DNDA7	DMA next destination address register CH7					√	00000000$_H$
FFFF747C$_H$	DNDA7L	DMA next destination address register LCH7				√		0000$_H$
FFFF747E$_H$	DNDA7H	DMA next destination address register HCH7				√		0000$_H$
FFFF7480$_H$	DNDC7	DMA next destination chip select register CH7				√		0001$_H$
FFFF7482$_H$	DTC7	DMA transfer count register CH7				√		0000$_H$
FFFF7484$_H$	DNTC7	DMA next transfer count register CH7				√		0000$_H$
FFFF7486$_H$	DTCC7	DMA transfer count compare register CH7				√		0000$_H$
FFFF7488$_H$	DTCT7	DMA transfer control register CH7				√		0000$_H$
FFFF748A$_H$	DTS7	DMA transfer status register CH7		√	√			00$_H$
FFFF7500$_H$	DTRC1	DMA transfer request control register 1		√	√			00$_H$
FFFF7510$_H$	DTRS8	DMA transfer request select register CH8				√		0000$_H$
FFFF7514$_H$	DSA8	DMA source address register CH8					√	00000000$_H$
FFFF7514$_H$	DSA8L	DMA source address register LCH8				√		0000$_H$
FFFF7516$_H$	DSA8H	DMA source address register HCH8				√		0000$_H$

Table 10-7 DMAC setting registers (6/10)

Address	Symbol	Function register name	R/W	Operable bits				Initial value
				1	8	16	32	
$FFFF7518_H$	DSC8	DMA source chip select register CH8	R/W			√		0001_H
$FFFF751C_H$	DNSA8	DMA next source address register CH8					√	00000000_H
$FFFF751C_H$	DNSA8L	DMA next source address register LCH8				√		0000_H
$FFFF751E_H$	DNSA8H	DMA next source address register HCH8				√		0000_H
$FFFF7520_H$	DNSC8	DMA next source chip select register CH8				√		0001_H
$FFFF7524_H$	DDA8	DMA destination address register CH8					√	00000000_H
$FFFF7524_H$	DDA8L	DMA destination address register LCH8				√		0000_H
$FFFF7526_H$	DDA8H	DMA destination address register HCH8				√		0000_H
$FFFF7528_H$	DDC8	DMA destination chip select register CH8				√		0001_H
$FFFF752C_H$	DNDA8	DMA next destination address register CH8					√	00000000_H
$FFFF752C_H$	DNDA8L	DMA next destination address register LCH8				√		0000_H
$FFFF752E_H$	DNDA8H	DMA next destination address register HCH8				√		0000_H
$FFFF7530_H$	DNDC8	DMA next destination chip select register CH8				√		0001_H
$FFFF7532_H$	DTC8	DMA transfer count register CH8				√		0000_H
$FFFF7534_H$	DNTC8	DMA next transfer count register CH8				√		0000_H
$FFFF7536_H$	DTCC8	DMA transfer count compare register CH8				√		0000_H
$FFFF7538_H$	DTCT8	DMA transfer control register CH8				√		0000_H
$FFFF753A_H$	DTS8	DMA transfer status register CH8		√	√			00_H
$FFFF7540_H$	DTRS9	DMA transfer request select register CH9				√		0000_H
$FFFF7544_H$	DSA9	DMA source address register CH9					√	00000000_H
$FFFF7544_H$	DSA9L	DMA source address register LCH9				√		0000_H
$FFFF7546_H$	DSA9H	DMA source address register HCH9				√		0000_H
$FFFF7548_H$	DSC9	DMA source chip select register CH9				√		0001_H
$FFFF754C_H$	DNSA9	DMA next source address register CH9					√	00000000_H
$FFFF754C_H$	DNSA9L	DMA next source address register LCH9				√		0000_H
$FFFF754E_H$	DNSA9H	DMA next source address register HCH9				√		0000_H
$FFFF7550_H$	DNSC9	DMA next source chip select register CH9				√		0001_H
$FFFF7554_H$	DDA9	DMA destination address register CH9					√	00000000_H
$FFFF7554_H$	DDA9L	DMA destination address register LCH9				√		0000_H
$FFFF7556_H$	DDA9H	DMA destination address register HCH9				√		0000_H
$FFFF7558_H$	DDC9	DMA destination chip select register CH9				√		0001_H
$FFFF755C_H$	DNDA9	DMA next destination address register CH9					√	00000000_H
$FFFF755C_H$	DNDA9L	DMA next destination address register LCH9				√		0000_H
$FFFF755E_H$	DNDA9H	DMA next destination address register HCH9				√		0000_H
$FFFF7560_H$	DNDC9	DMA next destination chip select register CH9				√		0001_H
$FFFF7562_H$	DTC9	DMA transfer count register CH9				√		0000_H

Table 10-7 DMAC setting registers (7/10)

Address	Symbol	Function register name	R/W	Operable bits				Initial value
				1	8	16	32	
FFFF7564$_H$	DNTC9	DMA next transfer count register CH9	R/W			√		0000$_H$
FFFF7566$_H$	DTCC9	DMA transfer count compare register CH9				√		0000$_H$
FFFF7568$_H$	DTCT9	DMA transfer control register CH9				√		0000$_H$
FFFF756A$_H$	DTS9	DMA transfer status register CH9		√	√			00$_H$
FFFF7570$_H$	DTRS10	DMA transfer request select register CH10				√		0000$_H$
FFFF7574$_H$	DSA10	DMA source address register CH10					√	00000000$_H$
FFFF7574$_H$	DSA10L	DMA source address register LCH10				√		0000$_H$
FFFF7576$_H$	DSA10H	DMA source address register HCH10				√		0000$_H$
FFFF7578$_H$	DSC10	DMA source chip select register CH10				√		0001$_H$
FFFF757C$_H$	DNSA10	DMA next source address register CH10					√	00000000$_H$
FFFF757C$_H$	DNSA10L	DMA next source address register LCH10				√		0000$_H$
FFFF757E$_H$	DNSA10H	DMA next source address register HCH10				√		0000$_H$
FFFF7580$_H$	DNSC10	DMA next source chip select register CH10				√		0001$_H$
FFFF7584$_H$	DDA10	DMA destination address register CH10					√	00000000$_H$
FFFF7584$_H$	DDA10L	DMA destination address register LCH10				√		0000$_H$
FFFF7586$_H$	DDA10H	DMA destination address register HCH10				√		0000$_H$
FFFF7588$_H$	DDC10	DMA destination chip select register CH10				√		0001$_H$
FFFF758C$_H$	DNDA10	DMA next destination address register CH10					√	00000000$_H$
FFFF758C$_H$	DNDA10L	DMA next destination address register LCH10				√		0000$_H$
FFFF758E$_H$	DNDA10H	DMA next destination address register HCH10				√		0000$_H$
FFFF7590$_H$	DNDC10	DMA next destination chip select register CH10				√		0001$_H$
FFFF7592$_H$	DTC10	DMA transfer count register CH10				√		0000$_H$
FFFF7594$_H$	DNTC10	DMA next transfer count register CH10				√		0000$_H$
FFFF7596$_H$	DTCC10	DMA transfer count compare register CH10				√		0000$_H$
FFFF7598$_H$	DTCT10	DMA transfer control register CH10				√		0000$_H$
FFFF759A$_H$	DTS10	DMA transfer status register CH10		√	√			00$_H$
FFFF75A0$_H$	DTRS11	DMA transfer request select register CH11				√		0000$_H$
FFFF75A4$_H$	DSA11	DMA source address register CH11					√	00000000$_H$
FFFF75A4$_H$	DSA11L	DMA source address register LCH11				√		0000$_H$
FFFF75A6$_H$	DSA11H	DMA source address register HCH11				√		0000$_H$
FFFF75A8$_H$	DSC11	DMA source chip select register CH11				√		0001$_H$
FFFF75AC$_H$	DNSA11	DMA next source address register CH11					√	00000000$_H$
FFFF75AC$_H$	DNSA11L	DMA next source address register LCH11				√		0000$_H$
FFFF75AE$_H$	DNSA11H	DMA next source address register HCH11				√		0000$_H$
FFFF75B0$_H$	DNSC11	DMA next source chip select register CH11				√		0001$_H$
FFFF75B4$_H$	DDA11	DMA destination address register CH11					√	00000000$_H$
FFFF75B4$_H$	DDA11L	DMA destination address register LCH11				√		0000$_H$
FFFF75B6$_H$	DDA11H	DMA destination address register HCH11				√		0000$_H$

Table 10-7 DMAC setting registers (8/10)

Address	Symbol	Function register name	R/W	Operable bits				Initial value
				1	8	16	32	
FFFF75B8$_H$	DDC11	DMA destination chip select register CH11	R/W			√		0001$_H$
FFFF75BC$_H$	DNDA11	DMA next destination address register CH11					√	00000000$_H$
FFFF75BC$_H$	DNDA11L	DMA next destination address register LCH11				√		0000$_H$
FFFF75BE$_H$	DNDA11H	DMA next destination address register HCH11				√		0000$_H$
FFFF75C0$_H$	DNDC11	DMA next destination chip select register CH11				√		0001$_H$
FFFF75C2$_H$	DTC11	DMA transfer count register CH11				√		0000$_H$
FFFF75C4$_H$	DNTC11	DMA next transfer count register CH11				√		0000$_H$
FFFF75C6$_H$	DTCC11	DMA transfer count compare register CH11				√		0000$_H$
FFFF75C8$_H$	DTCT11	DMA transfer control register CH11				√		0000$_H$
FFFF75CA$_H$	DTS11	DMA transfer status register CH11		√	√			00$_H$
FFFF75D0$_H$	DTRS12	DMA transfer request select register CH12				√		0000$_H$
FFFF75D4$_H$	DSA12	DMA source address register CH12					√	00000000$_H$
FFFF75D4$_H$	DSA12L	DMA source address register LCH12				√		0000$_H$
FFFF75D6$_H$	DSA12H	DMA source address register HCH12				√		0000$_H$
FFFF75D8$_H$	DSC12	DMA source chip select register CH12				√		0001$_H$
FFFF75DC$_H$	DNSA12	DMA next source address register CH12					√	00000000$_H$
FFFF75DC$_H$	DNSA12L	DMA next source address register LCH12				√		0000$_H$
FFFF75DE$_H$	DNSA12H	DMA next source address register HCH12				√		0000$_H$
FFFF75E0$_H$	DNSC12	DMA next source chip select register CH12				√		0001$_H$
FFFF75E4$_H$	DDA12	DMA destination address register CH12					√	00000000$_H$
FFFF75E4$_H$	DDA12L	DMA destination address register LCH12				√		0000$_H$
FFFF75E6$_H$	DDA12H	DMA destination address register HCH12				√		0000$_H$
FFFF75E8$_H$	DDC12	DMA destination chip select register CH12				√		0001$_H$
FFFF75EC$_H$	DNDA12	DMA next destination address register CH12					√	00000000$_H$
FFFF75EC$_H$	DNDA12L	DMA next destination address register LCH12				√		0000$_H$
FFFF75EE$_H$	DNDA12H	DMA next destination address register HCH12				√		0000$_H$
FFFF75F0$_H$	DNDC12	DMA next destination chip select register CH12				√		0001$_H$
FFFF75F2$_H$	DTC12	DMA transfer count register CH12				√		0000$_H$
FFFF75F4$_H$	DNTC12	DMA next transfer count register CH12				√		0000$_H$
FFFF75F6$_H$	DTCC12	DMA transfer count compare register CH12				√		0000$_H$
FFFF75F8$_H$	DTCT12	DMA transfer control register CH12				√		0000$_H$
FFFF75FA$_H$	DTS12	DMA transfer status register CH12		√	√			00$_H$
FFFF7600$_H$	DTRS13	DMA transfer request select register CH13				√		0000$_H$

Table 10-7 DMAC setting registers (9/10)

Address	Symbol	Function register name	R/W	Operable bits				Initial value
				1	8	16	32	
$FFFF7604_H$	DSA13	DMA source address register CH13	R/W				√	00000000_H
$FFFF7604_H$	DSA13L	DMA source address register LCH13				√		0000_H
$FFFF7606_H$	DSA13H	DMA source address register HCH13				√		0000_H
$FFFF7608_H$	DSC13	DMA source chip select register CH13				√		0001_H
$FFFF760C_H$	DNSA13	DMA next source address register CH13					√	00000000_H
$FFFF760C_H$	DNSA13L	DMA next source address register LCH13				√		0000_H
$FFFF760E_H$	DNSA13H	DMA next source address register HCH13				√		0000_H
$FFFF7610_H$	DNSC13	DMA next source chip select register CH13				√		0001_H
$FFFF7614_H$	DDA13	DMA destination address register CH13					√	00000000_H
$FFFF7614_H$	DDA13L	DMA destination address register LCH13				√		0000_H
$FFFF7616_H$	DDA13H	DMA destination address register HCH13				√		0000_H
$FFFF7618_H$	DDC13	DMA destination chip select register CH13				√		0001_H
$FFFF761C_H$	DNDA13	DMA next destination address register CH13					√	00000000_H
$FFFF761C_H$	DNDA13L	DMA next destination address register LCH13				√		0000_H
$FFFF761E_H$	DNDA13H	DMA next destination address register HCH13				√		0000_H
$FFFF7620_H$	DNDC13	DMA next destination chip select register CH13				√		0001_H
$FFFF7622_H$	DTC13	DMA transfer count register CH13				√		0000_H
$FFFF7624_H$	DNTC13	DMA next transfer count register CH13				√		0000_H
$FFFF7626_H$	DTCC13	DMA transfer count compare register CH13				√		0000_H
$FFFF7628_H$	DTCT13	DMA transfer control register CH13				√		0000_H
$FFFF762A_H$	DTS13	DMA transfer status register CH13		√	√			00_H
$FFFF7630_H$	DTRS14	DMA transfer request select register CH14				√		0000_H
$FFFF7634_H$	DSA14	DMA source address register CH14					√	00000000_H
$FFFF7634_H$	DSA14L	DMA source address register LCH14				√		0000_H
$FFFF7636_H$	DSA14H	DMA source address register HCH14				√		0000_H
$FFFF7638_H$	DSC14	DMA source chip select register CH14				√		0001_H
$FFFF763C_H$	DNSA14	DMA next source address register CH14					√	00000000_H
$FFFF763C_H$	DNSA14L	DMA next source address register LCH14				√		0000_H
$FFFF763E_H$	DNSA14H	DMA next source address register HCH14				√		0000_H
$FFFF7640_H$	DNSC14	DMA next source chip select register CH14				√		0001H
$FFFF7644_H$	DDA14	DMA destination address register CH14					√	00000000_H
$FFFF7644_H$	DDA14L	DMA destination address register LCH14				√		0000_H
$FFFF7646_H$	DDA14H	DMA destination address register HCH14				√		0000_H
$FFFF7648_H$	DDC14	DMA destination chip select register CH14				√		0001_H

Table 10-7 DMAC setting registers (10/10)

Address	Symbol	Function register name	R/W	Operable bits				Initial value
				1	8	16	32	
FFFF764C$_H$	DNDA14	DMA next destination address register CH14	R/W				√	00000000$_H$
FFFF764C$_H$	DNDA14L	DMA next destination address register LCH14				√		0000$_H$
FFFF764E$_H$	DNDA14H	DMA next destination address register HCH14				√		0000$_H$
FFFF7650$_H$	DNDC14	DMA next destination chip select register CH14				√		0001$_H$
FFFF7652$_H$	DTC14	DMA transfer count register CH14				√		0000$_H$
FFFF7654$_H$	DNTC14	DMA next transfer count register CH14				√		0000$_H$
FFFF7656$_H$	DTCC14	DMA transfer count compare register CH14				√		0000$_H$
FFFF7658$_H$	DTCT14	DMA transfer control register CH14				√		0000$_H$
FFFF765A$_H$	DTS14	DMA transfer status register CH14		√	√			00$_H$
FFFF7660$_H$	DTRS15	DMA transfer request select register CH15				√		0000$_H$
FFFF7664$_H$	DSA15	DMA source address register CH15					√	00000000$_H$
FFFF7664$_H$	DSA15L	DMA source address register LCH15				√		0000$_H$
FFFF7666$_H$	DSA15H	DMA source address register HCH15				√		0000$_H$
FFFF7668$_H$	DSC15	DMA source chip select register CH15				√		0001$_H$
FFFF766C$_H$	DNSA15	DMA next source address register CH15					√	00000000$_H$
FFFF766C$_H$	DNSA15L	DMA next source address register LCH15				√		0000$_H$
FFFF766E$_H$	DNSA15H	DMA next source address register HCH15				√		0000$_H$
FFFF7670$_H$	DNSC15	DMA next source chip select register CH15				√		0001$_H$
FFFF7674$_H$	DDA15	DMA destination address register CH15					√	00000000$_H$
FFFF7674$_H$	DDA15L	DMA destination address register LCH15				√		0000$_H$
FFFF7676$_H$	DDA15H	DMA destination address register HCH15				√		0000$_H$
FFFF7678$_H$	DDC15	DMA destination chip select register CH15				√		0001$_H$
FFFF767C$_H$	DNDA15	DMA next destination address register CH15					√	00000000$_H$
FFFF767C$_H$	DNDA15L	DMA next destination address register LCH15				√		0000$_H$
FFFF767E$_H$	DNDA15H	DMA next destination address register HCH15				√		0000$_H$
FFFF7680$_H$	DNDC15	DMA next destination chip select register CH15				√		0001$_H$
FFFF7682$_H$	DTC15	DMA transfer count register CH15				√		0000$_H$
FFFF7684$_H$	DNTC15	DMA next transfer count register CH15				√		0000$_H$
FFFF7686$_H$	DTCC15	DMA transfer count compare register CH15				√		0000$_H$
FFFF7688$_H$	DTCT15	DMA transfer control register CH15				√		0000$_H$
FFFF768A$_H$	DTS15	DMA transfer status register CH15		√	√			00$_H$

Caution If an unmapped address is accessed, a write access is ignored and “0” is returned in response to a read access.

10.4.3 Enabling or disabling writing control registers

The following control registers cannot be written while DMA transfer is enabled. All these registers can always be read, however.

Table 10-8 Enabling/disabling writing control registers

Always writable	DTRCx, DNSAnL, DNSAnH, DNSCn, DNDAnL, DNDAnH, DNDCn, DNTCn, DTSn
Writing prohibited while DMA transfer is enabled (DTSn.DTSnDTE = 1) (Operation is not guaranteed if these registers are written.)	DTRSn, DSAnL, DSAnH, DSCn, DDAnL, DDAnH, DDCn, DTCn, DTCCn, DTCTn

Note n = 0 to 15, x = 0, 1

10.5 DMA Control Registers

10.5.1 DTRCx (x = 0, 1): DMA transfer request control register

Access This register can be read or written in 8- or 1-bit units.

Address DTRC0: FFFF 7300_H, DTRC1: FFFF 7500_H

Initial Value 00_H

7	6	5	4	3	2	1	0
DTRCx ERR	0	0	0	0	0	0	DTRCx ADS
R/W	R	R	R	R	R	R	R/W

Table 10-9 DTRCx register contents

Bit position	Bit name	Function
7	DTRCxERR	DMA transfer error status This bit indicates that an error response has been received from the transfer target during DMA transfer. If an error response is received, the DTRCxERR and DTRCxADS bits are set and a SysError exception is generated by the CPU. To clear this bit, write “0” to it. 0: No DMA transfer error 1: DMA transfer error
0	DTRCxADS	DMA transfer aborted This bit indicates that DMA transfer has been aborted by a transfer stop request. In addition, the current DMA transfer can be aborted if the user writes “1” to this bit. 0: DMA transfer not aborted 1: DMA transfer aborted/DMA transfer abort request

DTRC0 and DRTC1 correspond to DMA channels 0 to 7 and 8 to 15, respectively.

10.5.2 DTRSn (n = 0 to 15): DMA transfer request select register

Access This register can be read or written in 16-bit units.

Address DTRS15: FFFF 7660_H, DTRS14: FFFF 7630_H, DTRS13: FFFF 7600_H, DTRS12: FFFF $75D0_H$, DTRS11: FFFF $75A0_H$, DTRS10: FFFF 7570_H, DTRS9: FFFF 7540_H, DTRS8: FFFF 7510_H, DTRS7: FFFF 7460_H, DTRS6: FFFF 7430_H, DTRS5: FFFF 7400_H, DTRS4: FFFF $73D0_H$, DTRS3: FFFF $73A0_H$, DTRS2: FFFF 7370_H, DTRS1: FFFF 7340_H, DTRS0: FFFF 7310_H

Initial Value 0000_H

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R

7	6	5	4	3	2	1	0
0	0	0	0	DTR3	DTR2	DTR1	DTR0
R	R	R	R	R/W	R/W	R/W	R/W

Table 10-10 DTRSn register contents

Bit position	Bit name	Function
3:0	DTR3 to DTR0	DMA transfer request assignment These bits specify assignment of a DMA transfer request to channel n.

DTR3	DTR2	DTR1	DTR0	DMA transfer request
0	0	0	0	Software DMA transfer request
0	0	0	1	Hardware DMA transfer request
Other than above				Setting prohibited

Cautions

1. Writing these bits is prohibited while DMA transfer is enabled (DTSn.DTSnDTE bit = 1). If they are written, the operation is not guaranteed.
2. The operation is not guaranteed if DTR[3:0] are set to a prohibited status.

10.5.3 DSAnL (n = 0 to 15): DMA source address register L

Access This register can be read or written in 16-bit units.

Address DSA15L: FFFF 7664_H, DSA14L: FFFF 7634_H, DSA13L: FFFF 7604_H, DSA12L: FFFF $75D4_H$, DSA11L: FFFF $75A4_H$, DSA10L: FFFF 7574_H, DSA9L: FFFF 7544_H, DSA8L: FFFF 7514_H, DSA7L: FFFF 7464_H, DSA6L: FFFF 7434_H, DSA5L: FFFF 7404_H, DSA4L: FFFF $73D4_H$, DSA3L: FFFF $73A4_H$, DSA2L: FFFF 7374_H, DSA1L: FFFF 7344_H, DSA0L: FFFF 7314_H

Initial Value 0000_H

15	14	13	12	11	10	9	8
SA15	SA14	SA13	SA12	SA11	SA10	SA9	SA8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7	6	5	4	3	2	1	0
SA7	SA6	SA5	SA4	SA3	SA2	SA1	SA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 10-11 DSAnL register contents

Bit position	Bit name	Function
15:0	SA15 to SA0	DMA source address These bits set the lower 16 bits of the transfer source address of channel n. If this register is referenced during DMA transfer, the address from which data is to be transferred next can be read. When referencing this register, it is recommended to access this register together with DSAnH in 32-bit units. If the NSAV bit of the DNSAnH register is not set (to “1”) when DMA transfer has been completed, the values of these bits return to the values when DMA transfer was started.

Cautions

1. Writing these bits is prohibited while DMA transfer is enabled (DTSn.DTSnDTE bit = 1). If they are written, the operation is not guaranteed.
2. Set an address by accessing in 32-bit units while the DTSnDTE bit is “0” in order to avoid data being transferred from an address that has not been completely set.
3. DMA transfer of misaligned data is not supported. The lower 4 bits of an address corresponding to the transfer data size are as follows (x indicates any bit).

   The operation is not guaranteed if a setting other than the following is made.

Data size	SA3	SA2	SA1	SA0
8 bits	×	×	×	×
16 bits	×	×	×	0
32 bits	×	×	0	0
128 bits	0	0	0	0

10.5.4 DSAnH (n = 0 to 15): DMA source address register H

Access This register can be read or written in 16-bit units.

Address DSA15H: FFFF 7666_H, DSA14H: FFFF 7636_H, DSA13H: FFFF 7606_H, DSA12H: FFFF $75D6_H$, DSA11H: FFFF $75A6_H$, DSA10H: FFFF 7576_H, DSA9H: FFFF 7546_H, DSA8H: FFFF 7516_H, DSA7H: FFFF 7466_H, DSA6H: FFFF 7436_H, DSA5H: FFFF 7406_H, DSA4H: FFFF $73D6_H$, DSA3H: FFFF $73A6_H$, DSA2H: FFFF 7376_H, DSA1H: FFFF 7346_H, DSA0H: FFFF 7316_H

Initial Value 0000_H

15	14	13	12	11	10	9	8
0	0	0	SA28	SA27	SA26	SA25	SA24
R	R	R	R/W	R/W	R/W	R/W	R/W

7	6	5	4	3	2	1	0
SA23	SA22	SA21	SA20	SA19	SA18	SA17	SA16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 10-12 DSAnH register contents

Bit position	Bit name	Function
12:0	SA28 to SA16	DMA source address These bits set the higher 13 bits of the transfer source address of channel n. If this register is referenced during DMA transfer, the address from which data is to be transferred next can be read. When referencing this register, it is recommended to access this register together with DSAnL in 32-bit units. If the NSAV bit of the DNSAnH register is not set (to “1”) when DMA transfer has been completed, the values of these bits return to the values when DMA transfer was started.

Cautions

1. Writing these bits is prohibited while DMA transfer is enabled (DTSn.DTSnDTE bit = 1). If they are written, the operation is not guaranteed.
2. Set an address by accessing in 32-bit units while the DTSnDTE bit is “0” in order to avoid data being transferred from an address that has not been completely set.

10.5.5 DSCn (n = 0 to 15): DMA source chip select register

Access This register can be read or written in 16-bit units.

Address DSC15: FFFF 7668$_H$, DSC14: FFFF 7638$_H$, DSC13: FFFF 7608$_H$, DSC12: FFFF 75D8$_H$, DSC11: FFFF 75A8$_H$, DSC10: FFFF 7578$_H$, DSC9: FFFF 7548$_H$, DSC8: FFFF 7518$_H$, DSC7: FFFF 7468$_H$, DSC6: FFFF 7438$_H$, DSC5: FFFF 7408$_H$, DSC4: FFFF 73D8$_H$, DSC3: FFFF 73A8$_H$, DSC2: FFFF 7378$_H$, DSC1: FFFF 7348$_H$, DSC0: FFFF 7318$_H$

Initial Value 0001$_H$

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R

7	6	5	4	3	2	1	0
0	0	0	0	0	SCS1	SCS0	SCSE
R	R	R	R	R	R/W	R/W	R/W

Table 10-13 DSCn register contents

Bit position	Bit name	Function
2 1 0	SCS1 SCS0 SCSE	DMA source chip select These bits specify an area to be selected as the transfer source of channel n.

SCS1	SCS0	SCSE	Selected area
0	0	1	External memory area, P-bus peripheral I/O area, and H-bus peripheral I/O area
0	1	0	Internal flash memory and internal RAM
Other than above			Setting prohibited

Cautions

1. Writing these bits is prohibited while DMA transfer is enabled (DTSn.DTSnDTE bit = 1). If they are written, the operation is not guaranteed.
2. Set the SCS0 and SCSE bits so that only one of them is “1”. If both of these bits are set to 1, the operation is not guaranteed.
3. Be sure to set the SCS1 bit to “0”.

10.5.6 DNSAnL (n = 0 to 15): DMA next source address register L

Access This register can be read or written in 16-bit units.

Address DNSA15L: FFFF 766C$_H$, DNSA14L: FFFF 763C$_H$, DNSA13L: FFFF 760C$_H$, DNSA12L: FFFF 75DC$_H$, DNSA11L: FFFF 75AC$_H$, DNSA10L: FFFF 757C$_H$, DNSA9L: FFFF 754C$_H$, DNSA8L: FFFF 751C$_H$, DNSA7L: FFFF 746C$_H$, DNSA6L: FFFF 743C$_H$, DNSA5L: FFFF 740C$_H$, DNSA4L: FFFF 73DC$_H$, DNSA3L: FFFF 73AC$_H$, DNSA2L: FFFF 737C$_H$, DNSA1L: FFFF 734C$_H$, DNSA0L: FFFF 731C$_H$

Access 0000$_H$

15	14	13	12	11	10	9	8
NAS15	NAS14	NAS13	NAS12	NAS11	NAS10	NAS9	NAS8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7	6	5	4	3	2	1	0
NAS7	NAS6	NAS5	NAS4	NAS3	NAS2	NAS1	NAS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 10-14 DNSAnL register contents

Bit position	Bit name	Function
15:0	NSA15 to NSA0	DMA next source address These bits set the lower 16 bits of the transfer source address when data is next transferred from channel n.

Caution DMA transfer of misaligned data is not supported. The lower 4 bits of the address corresponding to data size are as follows (× indicates any bit).

The operation is not guaranteed if a setting other than the following is made.

Data size	NSA3	NSA2	NSA1	NSA0
8 bits	×	×	×	×
16 bits	×	×	×	0
32 bits	×	×	0	0
128 bits	0	0	0	0

10.5.7 DNSAnH (n = 0 to 15): DMA next source address register H

Access This register can be read or written in 16-bit units.

Address DNSA15H: FFFF $766E_H$, DNSA14H: FFFF $763E_H$, DNSA13H: FFFF $760E_H$, DNSA12H: FFFF $75DE_H$, DNSA11H: FFFF $75AE_H$, DNSA10H: FFFF $757E_H$, DNSA9H: FFFF $754E_H$, DNSA8H: FFFF $751E_H$, DNSA7H: FFFF $746E_H$, DNSA6H: FFFF $743E_H$, DNSA5H: FFFF $740E_H$, DNSA4H: FFFF $73DE_H$, DNSA3H: FFFF $73AE_H$, DNSA2H: FFFF $737E_H$, DNSA1H: FFFF $734E_H$, DNSA0H: FFFF $731E_H$

Initial Value 0000_H

15	14	13	12	11	10	9	8
NSAV	0	0	NSA28	NSA27	NSA26	NSA25	NSA24
R/W	R	R	R/W	R/W	R/W	R/W	R/W

7	6	5	4	3	2	1	0
NSA23	NSA22	NSA21	NSA20	NSA19	NSA18	NSA17	NSA16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 10-15 DNSAnH register contents

Bit position	Bit name	Function
15	NSAV	DMA next source address valid This bit controls whether to copy an address from the DMA next source address register to the DMA source address register when DMA transfer has been completed. It is cleared when the address has been copied. 0: Does not copy/copying completed 1: Copies/copying not completed
12:0	NSA28 to NSA16	DMA next source address These bits specify the higher 13 bits of the transfer source address for the next transfer of channel n.

10.5.8 DNSCn (n = 0 to 15): DMA next source chip select register

Access This register can be read or written in 16-bit units.

Address DNSC15: FFFF 7670_H, DNSC14: FFFF 7640_H, DNSC13: FFFF 7610_H, DNSC12: FFFF $75E0_H$, DNSC11: FFFF $75B0_H$, DNSC10: FFFF 7580_H, DNSC9: FFFF 7550_H, DNSC8: FFFF 7520_H, DNSC7: FFFF 7470_H, DNSC6: FFFF 7440_H, DNSC5: FFFF 7410_H, DNSC4: FFFF $73E0_H$, DNSC3: FFFF $73B0_H$, DNSC2: FFFF 7380_H, DNSC1: FFFF 7350_H, DNSC0: FFFF 7320_H

Initial Value 0001_H

15	14	13	12	11	10	9	8
NSCV	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R

7	6	5	4	3	2	1	0
0	0	0	0	0	NSCS1	NSCS0	NSCSE
R	R	R	R	R	R/W	R/W	R/W

Table 10-16 DNSCn register contents

Bit position	Bit name	Function
15	NSCV	DMA next source address select valid This bit controls whether to copy a chip select signal from the DMA next source chip select register to the DMA source chip select register when DMA transfer has been completed. It is cleared when the chip select signal has been copied. 0: Does not copy/copying completed 1: Copies/copying not completed
2 1 0	NSCS1 NSCS0 NSCSE	DMA next source chip select These bits specify an area to be selected as the transfer source for the next transfer of channel n.

NSCS1	NSCS0	NSCSE	Selected area
0	0	1	External memory area, P-bus peripheral I/O area, and H-bus peripheral I/O area
0	1	0	Internal flash memory and internal RAM
Other than above			Setting prohibited

Cautions

1. Set the NSCS0 and NSCSE bits so that only one of them is 1. If both of these bits are set to 1, the operation is not guaranteed.
2. Be sure to set the NCSC1 bit to “0”.

10.5.9 DDAnL (n = 0 to 15): DMA destination address register L

Access This register can be read or written in 16-bit units.

Address DDA15L: FFFF 7674_H, DDA14L: FFFF 7644_H, DDA13L: FFFF 7614_H, DDA12L: FFFF $75E4_H$, DDA11L: FFFF $75B4_H$, DDA10L: FFFF 7584_H, DDA9L: FFFF 7554_H, DDA8L: FFFF 7524_H, DDA7L: FFFF 7474_H, DDA6L: FFFF 7444_H, DDA5L: FFFF 7414_H, DDA4L: FFFF $73E4_H$, DDA3L: FFFF $73B4_H$, DDA2L: FFFF 7384_H, DDA1L: FFFF 7354_H, DDA0L: FFFF 7324_H

Initial Value 0000_H

15	14	13	12	11	10	9	8
DA15	DA14	DA13	DA12	DA11	DA10	DA9	DA8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7	6	5	4	3	2	1	0
DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 10-17 DDAnL register contents

Bit position	Bit name	Function
15:0	DA15 to DA0	DMA destination address These bits specify the lower 16 bits of the transfer destination address of channel n. If this register is referenced during DMA transfer, the address to which data is to be transferred next can be read. When referencing this register, it is recommended to access this register together with DDAnH in 32-bit units. If the NDAV bit of the DNDAnH register is not set (to “1”) when DMA transfer has been completed, the values of these bits return to the values when DMA transfer was started.

Cautions

1. Writing these bits is prohibited while DMA transfer is enabled (DTSn.DTSnDTE bit = 1). If they are written, the operation is not guaranteed.
2. Set an address by accessing in 32-bit units while the DTSnDTE bit is “0” in order to avoid data being transferred from an address that has not been completely set.
3. If an error occurs in the transfer target in the read cycle of DMA transfer, the write cycle is not executed but the destination address is updated.
4. DMA transfer of misaligned data is not supported. The lower 4 bits of an address corresponding to the transfer data size are as follows (× indicates any bit). The operation is not guaranteed if a setting other than the following is made.

Data size	DA3	DA2	DA1	DA0
8 bits	×	×	×	×
16 bits	×	×	×	0
32 bits	×	×	0	0
128 bits	0	0	0	0

10.5.10 DDAnH (n = 0 to 15): DMA destination address register H

Access This register can be read or written in 16-bit units.

Address DDA15H: FFFF 7676_H, DDA14H: FFFF 7646_H, DDA13H: FFFF 7616_H, DDA12H: FFFF $75E6_H$, DDA11H: FFFF $75B6_H$, DDA10H: FFFF 7586_H, DDA9H: FFFF 7556_H, DDA8H: FFFF 7526_H, DDA7H: FFFF 7476_H, DDA6H: FFFF 7446_H, DDA5H: FFFF 7416_H, DDA4H: FFFF $73E6_H$, DDA3H: FFFF $73B6_H$, DDA2H: FFFF 7386_H, DDA1H: FFFF 7356_H, DDA0H: FFFF 7326_H

Initial Value 0000_H

15	14	13	12	11	10	9	8
0	0	0	DA28	DA27	DA26	DA25	DA24
R	R	R	R/W	R/W	R/W	R/W	R/W

7	6	5	4	3	2	1	0
DA23	DA22	DA21	DA20	DA19	DA18	DA17	DA16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 10-18 DDAnH register contents

Bit position	Bit name	Function
12:0	DA28 to DA16	DMA destination address These bits specify the higher 13 bits of the transfer destination address of channel n. If this register is referenced during DMA transfer, the address to which data is to be transferred next can be read. When referencing this register, it is recommended to access this register together with DDAnL in 32-bit units. If the NDAV bit of the DNDAnH register is not set (to “1”) when DMA transfer has been completed, the values of these bits return to the values when DMA transfer was started.

Cautions

1. Writing these bits is prohibited while DMA transfer is enabled (DTSn.DTSnDTE bit = 1). If they are written, the operation is not guaranteed.
2. Set an address by accessing in 32-bit units while the DTSnDTE bit is “0” in order to avoid data being transferred from an address that has not been completely set.
3. If an error occurs in the transfer target in the read cycle of DMA transfer, the write cycle is not executed but the destination address is updated.

10.5.11 DDCn (n = 0 to 15): DMA destination chip select register

Access This register can be read or written in 16-bit units.

Address DDC15: FFFF 7678$_H$, DDC14: FFFF 7648$_H$, DDC13: FFFF 7618$_H$, DDC12: FFFF 75E8$_H$, DDC11: FFFF 75B8$_H$, DDC10: FFFF 7588$_H$, DDC9: FFFF 7558$_H$, DDC8: FFFF 7528$_H$, DDC7: FFFF 7478$_H$, DDC6: FFFF 7448$_H$, DDC5: FFFF 7418$_H$, DDC4: FFFF 73E8$_H$, DDC3: FFFF 73B8$_H$, DDC2: FFFF 7388$_H$, DDC1: FFFF 7358$_H$, DDC0: FFFF 7328$_H$

Initial Value 0001$_H$

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R

7	6	5	4	3	2	1	0
0	0	0	0	0	DCS1	DCS0	DCSE
R	R	R	R	R	R/W	R/W	R/W

Table 10-19 DDCn register contents

Bit position	Bit name	Function
2 1 0	DCS1 DCS0 DCSE	DMA destination chip select These bits specify an area to be selected as the transfer destination of channel n.

DCS1	DCS0	DCSE	Selected area
0	0	1	External memory area, P-bus peripheral I/O area, and H-bus peripheral I/O area
0	1	0	Internal flash memory and internal RAM
Other than above			Setting prohibited

Cautions

1. Writing these bits is prohibited while DMA transfer is enabled (DTSn.DTSnDTE bit = 1). If they are written, the operation is not guaranteed.
2. Set the DCS0 and DCSE bits so that only one of them is “1”. If both of these bits are set to 1, the operation is not guaranteed.
3. Be sure to set the DCS1 bit to “0”.

10.5.12 DNDAnL (n = 0 to 15): DMA next destination address register L

Access This register can be read or written in 16-bit units.

Address DNDA15L: FFFF 767C$_H$, DNDA14L: FFFF 764C$_H$, DNDA13L: FFFF 761C$_H$, DNDA12L: FFFF 75EC$_H$, DNDA11L: FFFF 75BC$_H$, DNDA10L: FFFF 758C$_H$, DNDA9L: FFFF 755C$_H$, DNDA8L: FFFF 752C$_H$, DNDA7L: FFFF 747C$_H$, DNDA6L: FFFF 744C$_H$, DNDA5L: FFFF 741C$_H$, DNDA4L: FFFF 73EC$_H$, DNDA3L: FFFF 73BC$_H$, DNDA2L: FFFF 738C$_H$, DNDA1L: FFFF 735C$_H$, DNDA0L: FFFF 732C$_H$

Initial Value 0000$_H$

15	14	13	12	11	10	9	8
NDA15	NDA14	NDA13	NDA12	NDA11	NDA10	NDA9	NDA8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7	6	5	4	3	2	1	0
NDA7	NDA6	NDA5	NDA4	NDA3	NDA2	NDA1	NDA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 10-20 DNDAnL register contents

Bit position	Bit name	Function
15:0	NDA15 to NDA0	DMA next destination address These bits specify the lower 16 bits of the transfer destination address when the next transfer of channel n is executed.

Caution DMA transfer of misaligned data is not supported. The lower 4 bits of an address corresponding to the transfer data size are as follows (× indicates any bit).

The operation is not guaranteed if a setting other than the following is made.

Data size	NDA3	NDA2	NDA1	NDA0
8 bits	×	×	×	×
16 bits	×	×	×	0
32 bits	×	×	0	0
128 bits	0	0	0	0

10.5.13 DNDAnH (n = 0 to 15): DMA next destination address register H

Access This register can be read or written in 16-bit units.

Address DNDA15H: FFFF 767E$_H$, DNDA14H: FFFF 764E$_H$, DNDA13H: FFFF 761E$_H$, DNDA12H: FFFF 75EE$_H$, DNDA11H: FFFF 75BE$_H$, DNDA10H: FFFF 758E$_H$, DNDA9H: FFFF 755E$_H$, DNDA8H: FFFF 752E$_H$, DNDA7H: FFFF 747E$_H$, DNDA6H: FFFF 744E$_H$, DNDA5H: FFFF 741E$_H$, DNDA4H: FFFF 73EE$_H$, DNDA3H: FFFF 73BE$_H$, DNDA2H: FFFF 738E$_H$, DNDA1H: FFFF 735E$_H$, DNDA0H: FFFF 732E$_H$

Initial Value 0000$_H$

15	14	13	12	11	10	9	8
NDAV	0	0	NDA28	NDA27	NDA26	NDA25	NDA24
R/W	R	R	R/W	R/W	R/W	R/W	R/W

7	6	5	4	3	2	1	0
NDA23	NDA22	NDA21	NDA20	NDA19	NDA18	NDA17	NDA16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 10-21 DNDAnH register contents

Bit position	Bit name	Function
15	NDAV	DMA next destination address valid This bit specifies whether to copy an address from the DMA next destination address register to the DMA destination address register when DMA transfer has been completed. It is cleared when the address has been copied. 0: Does not copy/copying completed 1: Copies/copying not completed
12:0	NDA28 to NDA16	DMA next destination address These bits specify the higher 13 bits of the transfer destination address when the next transfer of channel n is executed.

10.5.14 DNDCn (n = 0 to 15): DMA next destination chip select register

Access This register can be read or written in 16-bit units.

Address DNDC15: FFFF 7680$_H$, DNDC14: FFFF 7650$_H$, DNDC13: FFFF 7620$_H$, DNDC12: FFFF 75F0$_H$, DNDC11: FFFF 75C0$_H$, DNDC10: FFFF 7590$_H$, DNDC9: FFFF 7560$_H$, DNDC8: FFFF 7530$_H$, DNDC7: FFFF 7480$_H$, DNDC6: FFFF 7450$_H$, DNDC5: FFFF 7420$_H$, DNDC4: FFFF 73F0$_H$, DNDC3: FFFF 73C0$_H$, DNDC2: FFFF 7390$_H$, DNDC1: FFFF 7360$_H$, DNDC0: FFFF 7330$_H$

Initial Value 0001$_H$

15	14	13	12	11	10	9	8
NDCV	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R

7	6	5	4	3	2	1	0
0	0	0	0	0	NDCS1	NDCS0	NDCSE
R	R	R	R	R	R/W	R/W	R/W

Table 10-22 DNDCn register contents

Bit position	Bit name	Function
15	NDCV	DMA next destination chip select valid This bit specifies whether to copy a chip select signal from the DMA next destination chip select register to the DMA destination chip select register when DMA transfer is completed. It is cleared when the chip select signal is copied. 0: Does not copy/copying completed 1: Copies/copying not completed
2 1 0	NDCS1 NDCS0 NDCSE	DMA next destination chip select These bits specify an area to be selected as the transfer destination for the next transfer of channel n.

NDCS1	NDCS0	NDCSE	Selected area
0	0	1	External memory area, P-bus peripheral I/O area, and H-bus peripheral I/O area
0	1	0	Internal flash memory and internal RAM
Other than above			Setting prohibited

Cautions

1. Set the NDCS0 and NDCSE bits so that only one of them is “1”. If both of these bits are set to 1, the operation is not guaranteed.
2. Be sure to set the NDCS1 bit to “0”.

10.5.15 DTCn (n = 0 to 15): DMA transfer count register

Access This register can be read or written in 16-bit units.

Address DCT15: FFFF 7682_H, DCT14: FFFF 7652_H, DCT13: FFFF 7622_H, DCT12: FFFF $75F2_H$, DCT11: FFFF $75C2_H$, DCT10: FFFF 7592_H, DCT9: FFFF 7562_H, DCT8: FFFF 7532_H, DCT7: FFFF 7482_H, DCT6: FFFF 7452_H, DCT5: FFFF 7422_H, DCT4: FFFF $73F2_H$, DCT3: FFFF $73C2_H$, DCT2: FFFF 7392_H, DCT1: FFFF 7362_H, DCT0: FFFF 7332_H

Initial Value 0000_H

15	14	13	12	11	10	9	8
0	DTC14	DTC13	DTC12	DTC11	DTC10	DTC9	DTC8
R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7	6	5	4	3	2	1	0
DTC7	DTC6	DTC5	DTC4	DTC3	DTC2	DTC1	DTC0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 10-23 DTCn register contents

Bit position	Bit name	Function
14:0	DTC14 to DTC0	DMA transfer count These bits specify the number of times of DMA transfers (DMA transfer count) for channel n. When this register is referenced during DMA transfer, the remaining number of times DMA transfer to be executed can be read. If the NTCV bit of the DNTCn register is not set (to “1”), these bits hold the values when DMA transfer has been completed (0000H).

DTC[14:0]	The operation
0000H	Transfer executed 32,768 times or until completion of transfer
0001H	Transfer executed once or transfer to be executed once
...	...
7FFFH	Transfer executed 32,767 times or 32,767 times of transfer to be executed

Cautions

1. Writing these bits is prohibited while DMA transfer is enabled (DTSn.DTSnDTE bit = 1). If they are written, the operation is not guaranteed.
2. If an error occurs in the transfer target in the read cycle of DMA transfer, the write cycle is not executed but the destination address is updated.

10.5.16 DNTCn (n = 0 to 15): DMA next transfer count register

Access This register can be read or written in 16-bit units.

Address DNCT15: FFFF 7684_H, DNCT14: FFFF 7654_H, DNCT13: FFFF 7624_H, DNCT12: FFFF $75F4_H$, DNCT11: FFFF $75C4_H$, DNCT10: FFFF 7594_H, DNCT9: FFFF 7564_H, DNCT8: FFFF 7534_H, DNCT7: FFFF 7484_H, DNCT6: FFFF 7454_H, DNCT5: FFFF 7424_H, DNCT4: FFFF $73F4_H$, DNCT3: FFFF $73C4_H$, DNCT2: FFFF 7394_H, DNCT1: FFFF 7364_H, DNCT0: FFFF 7334_H

Initial Value 0000_H

15	14	13	12	11	10	9	8
NTCV	NDTC14	NDTC13	NDTC12	NDTC11	NDTC10	NDTC9	NDTC8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7	6	5	4	3	2	1	0
NDTC7	NDTC6	NDTC5	NDTC4	NDTC3	NDTC2	NDTC1	NDTC0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 10-24 DNTCn register contents

Bit position	Bit name	Function
15	NTCV	DMA next transfer count valid This bit controls whether to copy the DMA transfer count from the DMA next transfer count register to the DMA count register when DMA transfer has been completed. It is cleared when the DMA transfer count has been copied. 0: Does not copy/copying completed 1: Copies/copying not completed
14:0	NDTC14 to NDTC0	DMA next transfer count These bits specify the DMA next transfer count for channel n.

DTC[14:0]	The operation
0000H	Transfer executed 32,768 times
0001H	Transfer executed once
...	...
7FFFH	Transfer executed 32,767 times

10.5.17 DTCCn (n = 0 to 15): DMA transfer count compare register

Access This register can be read or written in 16-bit units.

Address DTCC15: FFFF 7686_H, DTCC14: FFFF 7656_H, DTCC13: FFFF 7626_H, DTCC12: FFFF $75F6_H$, DTCC11: FFFF $75C6_H$, DTCC10: FFFF 7596_H, DTCC9: FFFF 7566_H, DTCC8: FFFF 7536_H, DTCC7: FFFF 7486_H, DTCC6: FFFF 7456_H, DTCC5: FFFF 7426_H, DTCC4: FFFF $73F6_H$, DTCC3: FFFF $73C6_H$, DTCC2: FFFF 7396_H, DTCC1: FFFF 7366_H, DTCC0: FFFF 7336_H

Initial Value 0000_H

15	14	13	12	11	10	9	8
0	DTCC14	DTCC13	DTCC12	DTCC11	DTCC10	DTCC9	DTCC8
R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7	6	5	4	3	2	1	0
DTCC7	DTCC6	DTCC5	DTCC4	DTCC3	DTCC2	DTCC1	DTCC0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 10-25 DTCCn register contents

Bit position	Bit name	Function
14:0	DTCC14 to DTCC0	DMA transfer count comparison These bits specify the DMA transfer count to be compared with the value of the DMA transfer count register of channel n. If the values of these registers match, an interrupt is generated. This interrupt is used as a trigger for setting the next address. In other words, an interrupt is generated if DMA transfer has been executed the same number of times as the value set in this register.

DTC[14:0]	The operation
0000H	Not compared
0001H	Interrupt is generated when DTC = 0001H.
...	...
7FFFH	Interrupt is generated when DTC = 7FFFH.

Caution Writing these bits is prohibited while DMA transfer is enabled (DTSn.DTSnDTE bit = 1). If they are written, the operation is not guaranteed.

10.5.18 DTCTn (n = 0 to 15): DMA transfer control register

Access This register can be read or written in 16-bit units.

Address DTCT15: FFFF 7688_H, DTCT14: FFFF 7658_H, DTCT13: FFFF 7628_H, DTCT12: FFFF $75F8_H$, DTCT11: FFFF $75C8_H$, DTCT10: FFFF 7598_H, DTCT9: FFFF 7568_H, DTCT8: FFFF 7538_H, DTCT7: FFFF 7488_H, DTCT6: FFFF 7458_H, DTCT5: FFFF 7428_H, DTCT4: FFFF $73F8_H$, DTCT3: FFFF $73C8_H$, DTCT2: FFFF 7398_H, DTCT1: FFFF 7368_H, DTCT0: FFFF 7338_H

Initial Value 0000_H

15	14	13	12	11	10	9	8
0	DS1	DS0	MLE	0	0	0	0
R	R/W	R/W	R/W	R	R	R	R

7	6	5	4	3	2	1	0
SACM1	SACM0	DACM1	DACM0	0	0	0	DSM
R/W	R/W	R/W	R/W	R	R	R	R/W

Table 10-26 DTCTn register contents (1/2)

Bit position	Bit name	Function
14 13	DS1 DS0	DMA transfer data size These bits specify the DMA transfer data size of channel n. **DS1 / DS0 / Transfer data size** 0 / 0 / 8 bits 0 / 1 / 16 bits 1 / 0 / 32 bits 1 / 1 / 128 bits
12	MLE	Multi-link enable This bit specifies whether to acknowledge the next DMA transfer request, even if the DTSnTC bit is not cleared (to “0”) after DMA transfer has been completed. If this bit is set (to “1”), the DTSn.DTSnDTE bit is not cleared upon completion of DMA transfer. Even if the DTSnTC bit is not cleared, DMA transfer is executed if a DMA transfer request is issued. 0: Clears DTSnDTE bit upon completion of DMA transfer. 1: Does not clear DTSnDTE bit upon completion of DMA transfer.
7 6	SACM1 SACM0	DMA transfer source address counting direction These bits specify the direction in which the transfer source address of channel n is to be counted. **SACM1 / SACM0 / Counting direction** 0 / 0 / Incremented 0 / 1 / Decremented 1 / 0 / Fixed 1 / 1 / Setting prohibited

Table 10-26 DTCTn register contents (2/2)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>5
4</td><td>DACM1
DACM0</td><td>DMA transfer destination address counting direction
These bits specify the direction in which the transfer destination address of channel n is to be counted.
<table>
<tr><th>DACM1</th><th>DACM0</th><th>Counting direction</th></tr>
<tr><td>0</td><td>0</td><td>Incremented</td></tr>
<tr><td>0</td><td>1</td><td>Decremented</td></tr>
<tr><td>1</td><td>0</td><td>Fixed</td></tr>
<tr><td>1</td><td>1</td><td>Setting prohibited</td></tr>
</table></td></tr>
<tr><td>0</td><td>DSM</td><td>DMA signal mode
This bit specifies the output timing for the $\overline{\text{DMAAK}}$[0:5] and $\overline{\text{DMATC}}$[0:5] output pins.
0: Read cycle
1: Write cycle
The DSM bit is provided only for the DTCD0 to DTCD5 registers. The DTCD6 to DTCD15 registers do not have this bit.</td></tr>
</table>

Cautions

1. Writing these bits is prohibited while DMA transfer is enabled (DTSn.DTSnDTE bit = 1). If they are written, the operation is not guaranteed.
2. The operation cannot be guaranteed if the SACM[1:0] and DACM[1:0] bits are set to a prohibited status
3. Be sure to set bits 11 and 0 to “0”.

10.5.19 DTSn (n = 0 to 15): DMA transfer status register

Access This register can be read or written in 8- or 1-bit units.

Address DTS15: FFFF 768A$_H$, DTS14: FFFF 765A$_H$, DTS13: FFFF 762A$_H$, DTS12: FFFF 75FA$_H$, DTS11: FFFF 75CA$_H$, DTS10: FFFF 759A$_H$, DTS9: FFFF 756A$_H$, DTS8: FFFF 753A$_H$, DTS7: FFFF 748A$_H$, DTS6: FFFF 745A$_H$, DTS5: FFFF 742A$_H$, DTS4: FFFF 73FA$_H$, DTS3: FFFF 73CA$_H$, DTS2: FFFF 739A$_H$, DTS1: FFFF 736A$_H$, DTS0: FFFF 733A$_H$

Initial Value 00$_H$

7	6	5	4	3	2	1	0
DTSnTC	DTSnDT	0	0	DTSnER	DTSnDR	DTSnSR	DTSnDTE
R/W	R/W	R	R	R	R	R/W	R/W

Table 10-27 DTSn register contents (1/2)

Bit position	Bit name	Function
7	DTSnTC	DMA transfer end status This bit indicates that DMA transfer has been completed. Write “0” to this bit to clear it after reading “1” from it. It is recommended to write this bit using bit manipulation such as CLR1. 0: DMA transfer not completed 1: DMA transfer completed
6	DTSnDT	DT DMA transfer status This bit indicates that a DMA transfer request has been acknowledged and that DMA transfer is in progress. It is not set (to “1”) when only a DMA transfer request is issued. This bit is cleared (to “0”) when DMA transfer has been completed. If the DTSnDTE bit is “0”, this bit can be cleared by the user. (It can also be written at the same time as the DTSnDTE bit.) 0: DMA transfer request acknowledged 1: DMA transfer in progress
3	DTSnER	DMA transfer error flag This bit indicates that a DMA transfer error has occurred in channel n. It is cleared (to “0”) when the DTRCx.DTRCxERR bit is cleared. Note that this bit is read-only. 0: No DMA transfer error 1: DMA transfer error
2	DTSnDR	Hardware DMA transfer request flag This bit indicates that channel n has a hardware DMA transfer request. It is cleared (to “0”) when the hardware DMA transfer request is deasserted. This bit operates regardless of the status of the DTSnDTE bit. It is not set (to “1”) by a software DMA transfer request, or by a hardware DMA transfer request when a software DMA transfer request is selected by the DMA transfer request select register. Note that this bit is read-only. 0: No hardware DMA transfer request 1: Hardware DMA transfer request
1	DTSnSR	Software DMA transfer request This bit selects a software DMA transfer request. If a software DMA transfer request is selected by the DMA transfer request select register, writing “1” to this bit and the DTSnDTE bit starts DMA transfer. This bit is automatically cleared (to “0”) when DMA transfer has been completed. Writing “0” to this bit aborts DMA transfer. 0: No software DMA transfer request 1: Software DMA transfer request

Table 10-27 DTSn register contents (2/2)

Bit position	Bit name	Function
0	DTSnDTE	DMA transfer enable This bit enables or disables DMA transfer. DMA transfer is executed if “1” is written to this bit and a DMA transfer request is issued. This bit is automatically cleared (to “0”) if the MLE bit is “0” when DMA transfer has been completed. DMA transfer is aborted if “0” is written to this bit during DMA transfer. 0: Disables DMA transfer 1: Enables DMA transfer

10.6 DMAC Function Details

10.6.1 DMAC transfer setting flow

Figure 10-4 “DMAC transfer setting flow” shows the flow for setting DMAC transfer.

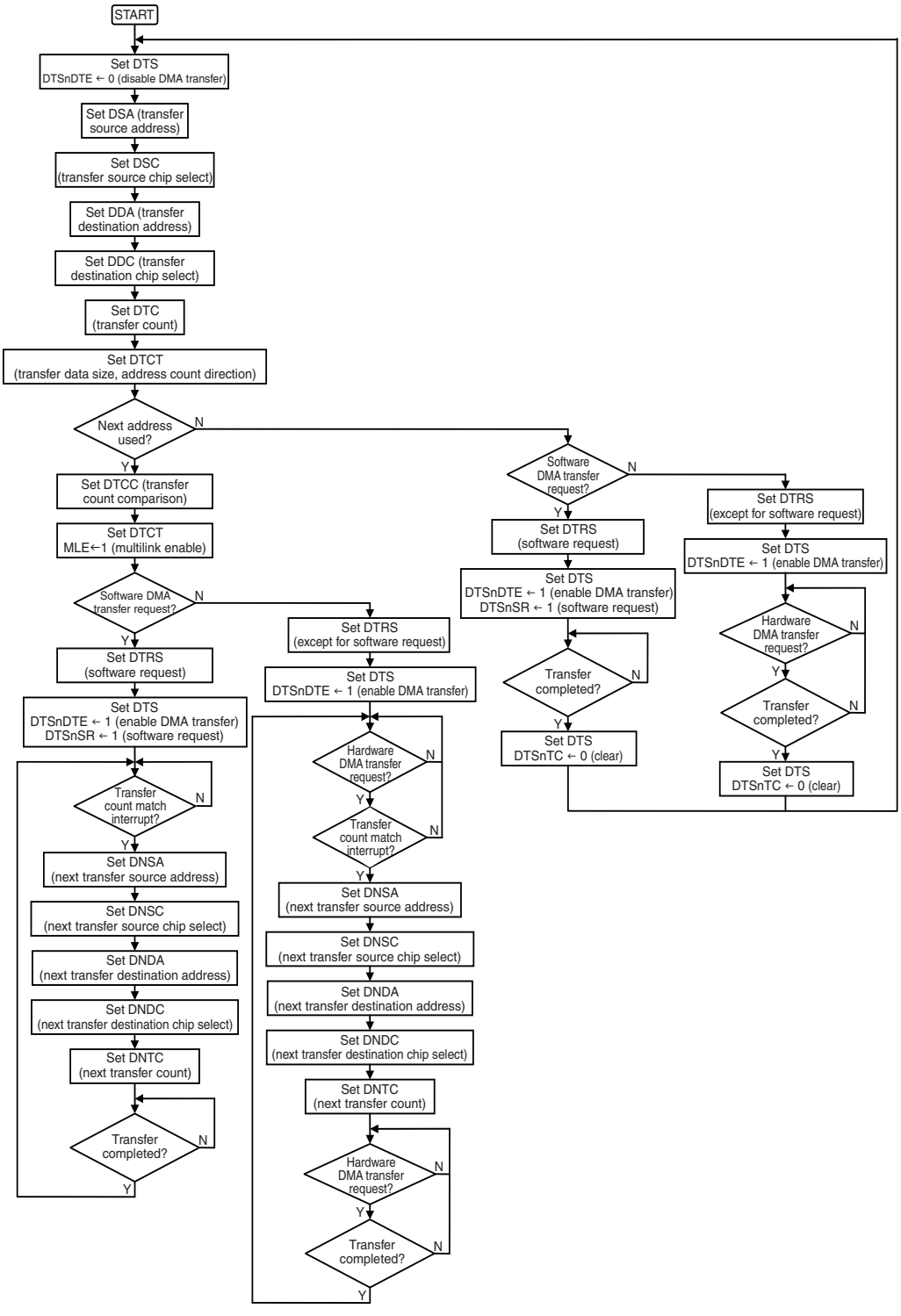


Figure 10-4 DMAC transfer setting flow

10.6.2 DMAC transfer modes

A single-transfer mode and a single-step transfer mode are supported as transfer modes.

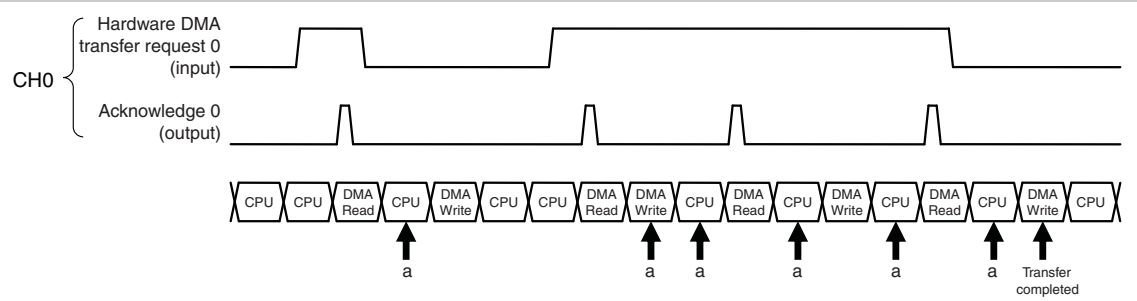
In either mode, transfer is executed in two cycles (dual address transfer) and therefore, a read cycle and a write cycle are generated each time transfer is executed. In the case of 128-bit transfer, the read cycle is generated four times and the write cycle is generated four times, in that order.

Note that the bus is not locked. Consequently, a CPU cycle may interrupt between the read and write cycles, and between the four read cycles and four write cycles during 128-bit transfer.

(1) Single transfer mode (when hardware DMA transfer request is generated)

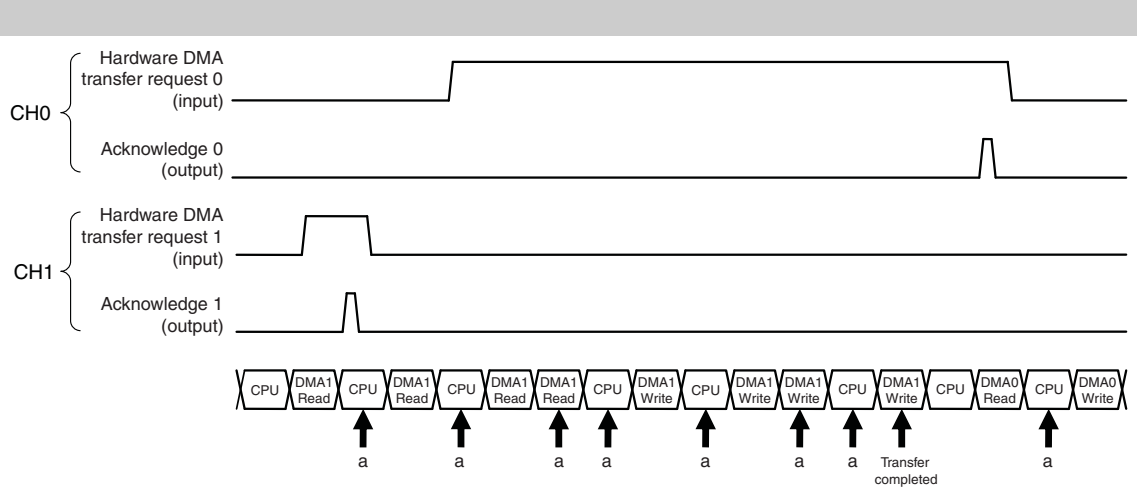
When a hardware DMA transfer request is acknowledged, data of the transfer data size (8, 16, 32, or 128 bits) is transferred. Each time transfer has been executed, the bus is released and the DMA controller waits for a DMA transfer request. At this time, the acknowledge n signal that indicates that the hardware DMA transfer request has been acknowledged is also output (n = 15 to 0).

Each time a hardware DMA transfer request has been acknowledged, transfer is executed once. This operation is repeated the number of times specified by the DMA transfer count register n (DTCn) (n = 15 to 0).



a: The bus is always released. When the CPU requests the bus mastership, the CPU acquires the bus mastership.

Figure 10-5 Example of single transfer (8/16/32 bits)



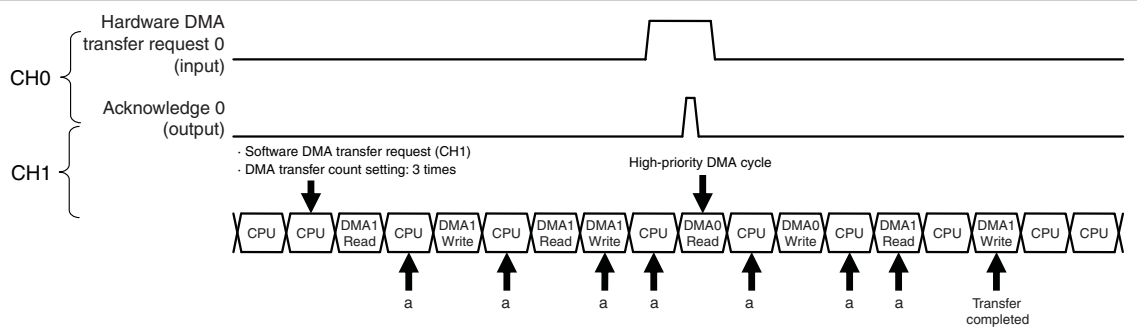
a: The bus is always released. When the CPU requests the bus mastership, the CPU acquires the bus mastership.

Figure 10-6 Example of single transfer (128 bits, DMA channel priority: CH0 (high) > CH1 (low))

(2) Single-step transfer mode (when software DMA transfer request is generated)

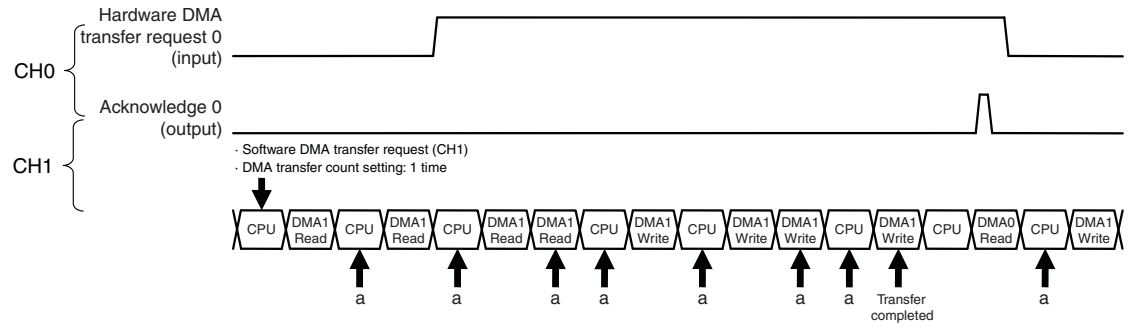
When a software DMA transfer request is acknowledged, data of the transfer data size (8, 16, 32, or 128 bits) is transferred. Each time transfer has been executed, the bus is released. At this time, the acknowledge n signal that indicates that a hardware DMA transfer request has been acknowledged is not output (n = 15 to 0).

Once a software DMA transfer request has been acknowledged, this operation is repeated the number of times specified by the DMA transfer count register n (DTCn) (n = 15 to 0). Because the priority is identified each time transfer is executed, the DMA cycle of a channel having the higher priority may interrupt.



a: The bus is always released. When the CPU requests the bus mastership, the CPU acquires the bus mastership.

Figure 10-7 Example of single-step transfer (8/16/32 bits, DMA channel priority: CH0 (high) > CH1 (low))



a: The bus is always released. When the CPU requests the bus mastership, the CPU acquires the bus mastership.

Figure 10-8 Example of single-step transfer (128 bits, DMA channel priority: CH0 (high) > CH1 (low))

10.6.3 DMAC channel priority control

The priority of each channel is fixed and is as follows.

DMAC0 > DMAC1

DMAC0: CH0 > CH1 > CH2 > CH3 > CH4 > CH5 > CH6 > CH7

DMAC1: CH8 > CH9 > CH10 > CH11 > CH12 > CH13 > CH14 > CH15

If another DMA transfer request with a high priority is generated, the DMA transfer request with the higher priority always takes precedence. When a software DMA transfer request is generated, the bus is also released each time a DMA cycle has been completed. If a DMA transfer request with a high priority is generated, therefore, the DMA transfer request with the higher priority always takes precedence.

An example where another DMA transfer request with a high priority is generated when DMA transfer is executed is shown below.

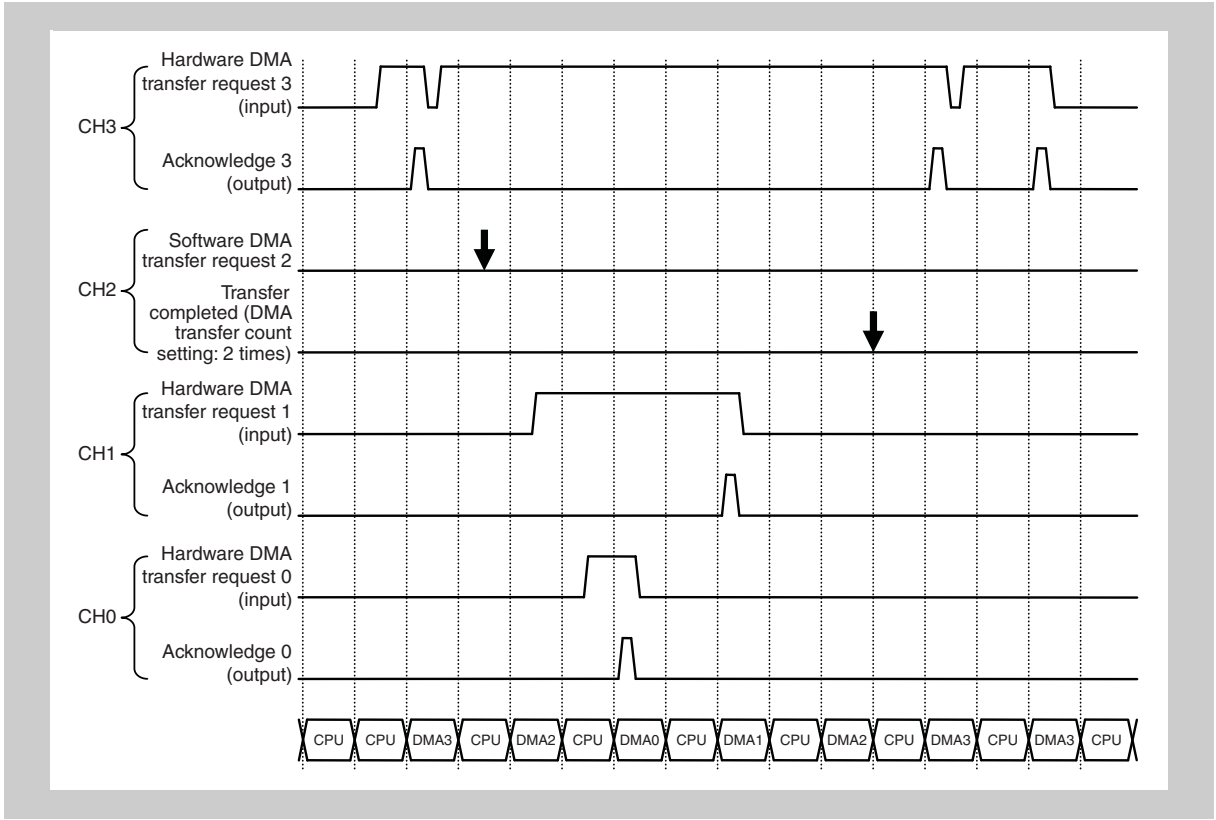


Figure 10-9 Example of priority control

10.6.4 Valid DMA transfer request conditions

Whether a DMA transfer request of channel n is acknowledged depends on the setting of the DTRCxERR and DTRCxADS bits of the DMA transfer request control register (DTRCx), the MLE bit of the DMA transfer control register (DTCTn), and the DTSnTC and DTSnDTE bits of the DMA transfer status register (DTSn). *Table 10-28 "Valid DMA transfer request conditions of channel n"* shows the relationship between the setting of each of the above bits and whether a DMA transfer request is acknowledged.

Table 10-28 Valid DMA transfer request conditions of channel n

Register.bit name	DTSn. DTSnDTE	DTSn. DTSnTC	DTCTn.MLE	DTRCx. DTRCxERR	DTRCx. DTRCxADS	DMA transfer request
When DMA transfer is disabled	0	×	×	×	×	Invalid
When DMA transfer error occurs	×	×	×	1	×	Invalid
When DMA transfer is aborted	×	×	×	×	1	Invalid
When DMA transfer is completed (multilink disabled)	×	1	0	×	×	Invalid
When DMA transfer is completed/not completed (multilink enabled)	1	×	1	0	0	Valid
When DMA transfer is enabled	1	0	0	0	0	Valid

Note n = 0 to 15

10.6.5 Next address function

(1) Next address setting register

This register is used to set beforehand the transfer information to be transferred next during DMA transfer. This information is copied to the corresponding register at the start of the last DMA cycle. The following registers are provided.

- DMA next source address register (DNSAnH/DNSAnL)
- DMA next source chip select register (DNSCn)
- DMA next destination address register (DNDAnH/DNDAnL)
- DMA next destination chip select register (DNDCn)
- DMA next transfer count register (DNTCn)

Each one of these registers has a valid bit in the most significant position (address register: most significant bit on H side), and whether to copy the transfer information to the current register at the start of the last DMA cycle can be selected. When the transfer information to be transferred next is copied to the current register, the valid bit is cleared.

(2) Processing upon DMA transfer completion when using next address function

Normally, upon completion of DMA transfer, the DMA transfer enable bit (DTSnDTE) is cleared at the same time the DMA transfer completion status bit (DTSnTC) of the DMA transfer status register (DTSn) is set, and subsequent DMA transfer requests are no longer acknowledged. However, if the multilink enable bit (MLE) is set, DTSnDTE is not cleared and DMA transfer requests can be acknowledged even if DTSnTC is set. Therefore, when using the next address function, it is possible to eliminate the need to clear DTSnTC upon DMA transfer completion and set DTSnDTE by setting MLE.

(3) Timing at which next address is set

The next address setting register can always be rewritten. However, to prevent a conflict between copying to the current register and a write operation by the user, complete setting the next address setting register before the last DMA cycle starts.

Use of the DMA transfer count match interrupt is recommended as the trigger for setting the next address setting register. In this case, set the DMA transfer count compare register (DTCCn) so as to secure the time required for setting the next address setting register.

10.6.6 Aborting/resuming DMA transfer

(1) Aborting or resuming DMA transfer for all channels through software

By setting the DMA transfer abort bit (DTRCxADS) of the DMA transfer request control register (DTRCx), the next DMA transfer and those that follow can be aborted. During a DMA cycle, the next DMA transfer is aborted after the ongoing DMA cycle has been completed. Note that the DMA transfer enable bit (DTSnDTE) and the software DMA transfer request bit (DTSnSR) of the DMA transfer status register (DTSn) are not cleared.

To resume the aborted DMA transfer, clear the DTRCxADS bit. If a DMA transfer is requested at that point, the transfer of the channel having the highest priority at that time is executed. To end DMA transfer, clear the DMA transfer request with the DTSnDTE bit cleared.

(2) Aborting or resuming DMA transfer by using DMA transfer enable bit (DTSnDTE)

By clearing the DMA transfer enable bit (DTSnDTE) of the DMA transfer status register (DTSn), the next DMA transfer and those that follow can be aborted. During a DMA cycle, the next DMA transfer is aborted after the ongoing DMA cycle is completed. Note that the software DMA transfer request bit (DTSnSR) of DTS is not cleared.

To resume the aborted DMA transfer, set the DTSnDTE bit. If another channel is not executing DMA transfer at that point, the priority is identified as usual. If another channel is executing DMA transfer, the priority is identified after that transfer has been completed. To end DMA transfer, clear the DMA transfer request with the DTSnDTE bit cleared.

(3) Aborting or resuming DMA transfer by using software DMA transfer request bit (DTSnSR)

By clearing the software DMA transfer request bit (DTSnSR) of the DMA transfer status register, the next DMA transfer and those that follow can be aborted (DTSn). During a DMA cycle, the next DMA transfer is aborted after the ongoing DMA cycle has been completed.

To resume the aborted DMA transfer, set the DTSnSR bit. If another channel is not executing DMA transfer at that point, the priority is identified as usual. If another channel is executing DMA transfer, the priority is identified after that transfer has been completed.

10.6.7 Error response support

(1) Aborting DMA transfer by error response

When an error occurs at the DMA transfer source or transfer destination, the DMA transfer abort bit (DTRCxADS) of the DMA transfer request control register (DTRCx) is set to abort the next and subsequent DMA transfers. At the same time, the DMA transfer error status bit (DTRCxERR) is set and a SysError exception is generated by the CPU. The user can learn in which channel the error has occurred, by using the DMA transfer error flag (DTSnER) of the DMA transfer status register (DTSn), when the user has confirmed that DTRCxERR has been set.

In this case, note that, if an error response is acknowledged in the read cycle, the write cycle is not executed, but the transfer address and the transfer count are updated.

(2) Canceling transfer abort by error response

DMA transfer abort can be canceled by clearing the DMA transfer abort bit (DTRCxADS) and DMA transfer error status bit (DTRCxERR) of the DMA transfer request control register (DTRCx).

Clear the DMA transfer enable bit (DTSnDTE) of the DMA transfer status register (DTSn) in advance, so that DMA transfer is not resumed after its abort has been canceled. In the case of a software DMA transfer request, also clear the software DMA transfer request bit (DTSnSR).

10.7 DTFR Function

The DMA trigger factor register (DTFR) selects DMA trigger factors from among interrupt signals, and requests DMAC for DMA transfer. DTFRn (n = 15 to 0) registers are included for selecting the signals to be used for DMA transfer requests from among the 128 input interrupt signals.

10.7.1 Features

Number of transfer factors

DMA transfer requests (for 16 channels) are selected from among 128 interrupt signals.

DMAC interface

The DMA transfer request signal n (n = 15 to 0) is output.

The DMA transfer request signal n is cleared by an acknowledge signal from DMA.

CPU interface

The last transfer signal from DMA is output as a CPU interrupt signal.

Clearing of transfer request

A function that clears transfer request signals sent to DMA through register access is provided.

Confirmation of transfer request

A function that checks transfer request signals sent to DMA through register access is provided.

10.8 DTFR Control Registers

10.8.1 DTFRn (n = 0 to 15): DTFRn register

Access This register can be read or written in 16-bit units.

Address DTFR0: FFFF 7B00$_H$, DTFR1: FFFF 7B02$_H$, DTFR2: FFFF 7B04$_H$, DTFR3: FFFF 7B06$_H$, DTFR4: FFFF 7B08$_H$, DTFR5: FFFF 7B0A$_H$, DTFR6: FFFF7B0C$_H$, DTFR7: FFFF 7B0E$_H$, DTFR8: FFFF 7B10$_H$, DTFR9: FFFF 7B12$_H$, DTFR10: FFFF 7B14$_H$, DTFR11: FFFF 7B16$_H$, DTFR12: FFFF 7B18$_H$, DTFR13: FFFF 7B1A$_H$, DTFR14: FFFF 7B1C$_H$, DTFR15: FFFF 7B1E$_H$

Initial Value 0000$_H$

15	14	13	12	11	10	9	8
REQEN	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R

7	6	5	4	3	2	1	0
0	IFCn6-0						
R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 10-29 DTFRn register contents

Bit position	Bit name	Function
15	REQEN	This bit enables or disables operation of the DMA source selector of channel n. 1: Enables operation of source selector 0: Stops operation of source selector. Does not issue DMA transfer request (DMARQ). The settings of IFCn6 to IFCn0 are valid. Requests are always sampled.
6:0	IFCn6 to IFCn0	These bits select the transfer source. The set values are shown in *Table 10-2 "DMA start sources (0 to 63)"*.

10.8.2 DRQCLR: DMA request clear register

Access This register can be read or written in 16-bit units.

Address FFFF 7B40$_H$

Initial Value 0000$_H$

15	14	13	12	11	10	9	8
RQCR15	RQCR14	RQCR13	RQCR12	RQCR11	RQCR10	RQCR9	RQCR8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7	6	5	4	3	2	1	0
RQCR7	RQCR6	RQCR5	RQCR4	RQCR3	RQCR2	RQCR1	RQCR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 10-30 DRQCLR register contents

Bit position	Bit name	Function
15:0	RQCR15 to RQCR0	Setting “1” to one of these bits clear the corresponding transfer request held in channel n to “0”.

Note Writing “0” to bits 15 to 0 is ignored.

10.8.3 DRQSTR: DMA request check register

Access This register is read-only, in 16-bit units.

Address FFFF 7B44$_H$

Initial Value 0000$_H$

15	14	13	12	11	10	9	8
RQST15	RQST14	RQST13	RQST12	RQST11	RQST10	RQST9	RQST8
R	R	R	R	R	R	R	R

7	6	5	4	3	2	1	0
RQST7	RQST6	RQST5	RQST4	RQST3	RQST2	RQST1	RQST0
R	R	R	R	R	R	R	R

Table 10-31 DRQSTR register contents

Bit position	Bit name	Function
15:0	RQST15 to RQST0	DMA transfer request status flag 1: Request issued (DMA transfer request signal n is “1”.) 0: No request (DMA transfer request signal n is “0”.)

10.9 DTS Function

The data transfer service (DTS) does not include registers for setting transfer information (TI). Consequently, each time a transfer request is generated, TI corresponding to the request number is fetched (read from internal RAM), and the transfer request for the TI is then sent to DMAT to execute data transfers, etc.

10.9.1 Features

Number of channels	Number of channels is freely set (channels 1 to 128)
Transfer data size	8 bits, 16 bits, and 32 bits
Transfer data	Fixed to little endian Misaligned data not supported
Maximum transfer count	When the DTS transfer count register is used as a 16-bit register: 65535 times When the DTS transfer count register is split into two 8-bit registers: 255 × n: (n indicates no limitation) Therefore, the number of transfers is not limited.
Channel priority control	No priority control function is provided (control must be done outside of DTS).
Subject to transfer	Internal flash memory Internal RAM External memory area P-bus peripheral I/O area H-bus peripheral I/O area
Chain function	This function enables DTS to be started multiple times by one request. After writing back TI, other TI is continuously fetched, so that data transfer (etc.) can be executed.

Transfer type

- 2-cycle transfer (dual address transfer) function

  After TI is fetched, data is read from the transfer source, and then the read data is written to the transfer destination. Afterward, updated TI is written back. This operation is the same as a DMA transfer, except for the TI fetch and write back. Since the bus remains unlocked during all cycles, a CPU cycle may interrupt.

- Compare function

  After TI is fetched, data is read from the comparison source and the comparison destination, and these two data are then compared. Afterward, updated TI is written back. A true/false judgment of the compare results can then be used to select whether to output an interrupt to the CPU or to continue processing using the chain function.

- Flag check function (bit check function)

  This function is used to check whether any data bit is “1” or “0”.

  After TI is fetched, mask data for flag checking and flagged check data are read, then any bit is determined. Afterward, the updated TI is written back. A true/false judgment of the flag check results can then be used to select whether to output an interrupt to the CPU or to continue processing using the chain function.

Transfer count register split mode

- Split

  In this mode, the 16-bit transfer count register is split into upper (8 bits) and lower (8 bits) sides which are used separately.

  The lower side is used as a transfer count register and the upper side is used as a transfer count hold register.

  Although the maximum transfer count is 255 times, when the lower side is zero, the upper side's values can be copied to the lower side, so transfers can be performed continuously without resetting the transfer count.

- Combined

  The entire 16 bits of the transfer count register are used. The maximum transfer count is 65535 times.

Transfer mode

- Single transfer mode

  When a DTS transfer request is generated, one DTS cycle is performed. Afterward, when a DTS transfer request is generated, another DTS cycle is performed. This operation is repeated until the transfer count register value reaches “0”.

- Block transfer mode

  When a DTS transfer request is generated, the number of DTS cycles specified in the transfer count register is performed.

  Other DTS transfer requests are not accepted until the number of DTS cycles specified in the transfer count register is completed. However, because the bus is not locked, a CPU cycle may interrupt.

Transfer address control

- Incremental
- Decremental
- Fixed

Transfer error support When an error response from the internal system bus occurs, the DTS cycle is aborted and an error flag is set. Also, a SysError exception is generated.

Interrupt output function An interrupt source signal is output (by the DTSFSL function) when the number of DTS transfers specified in the bus cycle count register has been completed, or depending on a flag check result and a compare result.

DTS transfer abort function This function supports aborting DTS transfer by software, but not by hardware.

10.10 DTS Control Registers

DTS uses the following control registers to control operation settings.

The DTS's registers include registers that can be accessed from the CPU (register group A) and internal registers that cannot be accessed from the CPU (register group B). Transfer information read from internal RAM is stored in the DTS internal register.

Register group B is comprised of DTS0SAR, DTS0DAR, DTS0TCEA, DTS0CIR, DTS0SCS, DTS0DCS, and DTS0ECSRA.

Caution The DTS register can only be accessed from CPU1, not CPU2.

Table 10-32 List of DTS control registers

	Address[a]	Symbol	Function register name	R/W	Operable bits			Initial value
					8	16	32	
Register group A	$FFFF7C00_H$	DTS0TSR	DTS transfer status register	R	√	–	–	00_H
	$FFFF7C04_H$	DTS0TRC	DTS transfer request control register	R/W	√	–	–	00_H
	$FFFF7C08_H$	DTS0ICR	DTS initialization control register	R/W	–	√	–	0000_H
	$FFFF7C08_H$	DTS0ICH	DTS initialization channel number register	R/W	√	–	–	00_H
	$FFFF7C09_H$	DTS0ITR	DTS initialization channel trigger register	R/W	√	–	–	00_H
	$FFFF7C0C_H$	DTS0BTR	DTS base table register	R/W	–	–	√	00000000_H
	$FFFF7C10_H$	DTS0BVR	DTS base vector register	R/W	–	–	√	00000000_H
	$FFFF7C14_H$	DTS0ACR	DTS active channel register	R	–	√	–	0000_H
	$FFFF7C18_H$	DTS0HST	DTS TI hold status register	R	√	–	–	00_H
	$FFFF7C20_H$	DTS0HC	DTS TI hold channel number register	R	–	–	√	00000000_H
	$FFFF7C20_H$	DTS0HC0	DTS hold channel register 0	R	√	–	–	00_H
	$FFFF7C21_H$	DTS0HC1	DTS hold channel register 1	R	√	–	–	00_H
	$FFFF7C22_H$	DTS0HC2	DTS hold channel register 2	R	√	–	–	00_H
	$FFFF7C23_H$	DTS0HC3	DTS hold channel register 3	R	√	–	–	00_H
Register group B	–	DTS0SAR	DTS source address register	–	–	–	–	00000000_H
	–	DTS0DAR	DTS destination address register	–	–	–	–	00000000_H
	–	DTS0TCEA	DTS transfer counter or else address register	–	–	–	–	0000_H
	–	DTS0CIR	DTS control information register	–	–	–	–	0000_H
	–	DTS0SCS	DTS source address count size register	–	–	–	–	00_H
	–	DTS0DCS	DTS destination address count size register	–	–	–	–	00_H
	–	DTS0ECSRA	DTS extension address count size/ repeat address register	–	–	–	–	00000000_H

a) If an address that is not mapped to DTS is accessed, write operations are ignored and a zero is returned when read.

10.10.1 DTS0TSR: DTS transfer status register [register group A]

Access This register is read-only, in 8-bit units.

Address FFFF 7C00$_H$

Initial Value 00$_H$

7	6	5	4	3	2	1	0
0	0	0	0	0	0	DTS0STPE	DTS0STPU
R	R	R	R	R	R	R	R

Table 10-33 DTS0TSR register contents

Bit position	Bit name	Function
1	DTS0STPE	DTS STоPped by Error response (DTS transfer abort by error response) This bit indicates whether a DTS transfer is aborted by an error response received from the transfer target. When an error response is received, “1” is set to this bit and a SysError exception is generated to abort the DTS transfer. When this bit is “1”, no transfers are accepted. This is a read-only bit. To clear this bit, write “1” to the DTS0ECL bit in the DTS0TRC register.
0	DTS0STPU	DTS STоPped by User request (DTS transfer abort by user request) This bit indicates whether a DTS transfer is aborted by a user request. “1” is set to this bit by writing “1” to the DTS0UST bit in the DTS0TRC register, and transfer is aborted. When this bit is “1”, no transfers are accepted. This is a read-only bit. To clear this bit, write “1” to the DTS0UCL bit in the DTS0TRC register.

10.10.2 DTS0TRC: DTS transfer request control register [register group A]

Access This register can be read or written in 8-bit units.

Address FFFF 7C04$_H$

Initial Value 00$_H$

7	6	5	4	3	2	1	0
DTS0UST	0	0	0	0	0	DTS0ECL	DTS0UCL
W	R	R	R	R	R	W	W

Table 10-34 DTS0TRC register contents

Bit position	Bit name	Function
7	DTS0UST	DTS User request STop Trigger (DTS abort request trigger bit) This is the transfer interrupt request trigger for all DTS channels. When “1” is written to this bit, the DTS0STPU bit in DTS0TSR is set to “1” and DTS transfer is aborted. This bit is always “0” when read, and writing “0” to this bit is ignored. When this bit and DTS0UCL are set to “1” at the same time, this bit takes priority.
1	DTS0ECL	DTS dts0stpE CLear Trigger (DTS0STPE clear request trigger bit) This is the clear request trigger bit for the DTS0STPE bit in DTS0TSR. When this bit is set to “1”, the DTS0STPE bit in DTS0TSR is cleared to “0”, and DTS transfers can be accepted. This bit is always “0” when read, and writing “0” to this bit is ignored.
0	DTS0UCL	DTS dts0stpU CLear Trigger (DTS0STPU clear request trigger bit) This is the clear request trigger bit for the DTS0STPU bit in DTS0TSR. When this bit is set to “1”, the DTS0STPU bit in DTS0TSR is cleared to “0”, and DTS transfers can be accepted. If a transfer has been aborted, the aborted transfer is restarted. This bit is always “0” when read, and writing “0” to this bit is ignored.

10.10.3 DTS0ICR: DTS initialization control register [register group A]

Access This register can be read or written in 16-bit units.
However, when the higher 8 bits and lower 8 bits of the DTS0ICR register are used as the DTS0ITR register and DTS0ICH register, respectively, this register can be read or written in 8-bit units.

Address DTS0ICR: FFFF $7C08_H$

DTS0ITR: FFFF $7C09_H$, DTS0ICH: FFFF $7C08_H$

Initial Value 0000_H

- DTS0ICR

15	14	13	12	11	10	9	8
0	0	0	0	0	DTS0ICS	DTS0HIT	DTS0TIT
R	R	R	R	R	R	W	W

7	6	5	4	3	2	1	0
0	DTS0ICH6	DTS0ICH5	DTS0ICH4	DTS0ICH3	DTS0ICH2	DTS0ICH1	DTS0ICH0
R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- DTS0ITR

7	6	5	4	3	2	1	0
0	0	0	0	0	DTS0ICS	DTS0HIT	DTS0TIT
R	R	R	R	R	R	W	W

- DTS0ICH

7	6	5	4	3	2	1	0
0	DTS0ICH6	DTS0ICH5	DTS0ICH4	DTS0ICH3	DTS0ICH2	DTS0ICH1	DTS0ICH0
R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 10-35 DTS0ICR register contents

Bit position	Bit name	Function
10	DTS0ICS	DTS Initialization Continuous Status (transfer abort status flag) This status flag indicates whether the DTS transfer is being aborted. When this flag is “1”, writing to this register is prohibited (writing to this register is ignored). When DTS0HIT or DTS0TIT in this register is set to “1”, this bit is set to “1”. When the transfer abort processing requested by using DTS0HIT or DTS0TIT in this register is completed, this bit is cleared. 0: Normal status 1: Transfer is being aborted.
9	DTS0HIT	DTS Hold Initialization Trigger (transfer abort and TI hold buffer initialization trigger bit) This is the trigger bit used to request aborting the current DTS transfer and clearing of all TI held in DTS. When “1” is written to this bit, all the held TI is cleared once the DTS cycle is completed. While waiting for the DTS cycle to be completed, the DTS0ICS bit is set to “1”. This bit is always “0” when read.
8	DTS0TIT	DTS Transfer Initialization Trigger (transfer abort trigger bit) This is the trigger bit used to request aborting the DTS transfer for the channel specified by the DTS0ICH6 to DTS0ICH0 bits in this register. When “1” is written to this bit the channel for current DTS transfer is compared with the channel specified by the DTS0ICH bit, once the DTS cycle is completed. If the two match, all subsequent transfers are stopped. While waiting for the DTS cycle to be completed, the DTS0ICS bit is set to “1”. If TI hold is set for the specified channel, the TI hold is also cleared. This bit is always “0” when read.
6:0	DTS0ICH6 to DTS0ICH0	DTS Initialization CHannel (initialization stop request channel) This bit specifies the channel for which initialization is to be aborted by the DTS0TIT bit. “1” can be written to these bits at the same time as when writing “1” to the DTS0HIT and DTS0TIT bits in this register.

10.10.4 DTS0BTR: DTS base table register [register group A]

Access This register can be read or written in 32-bit units.

Address FFFF 7C0C$_H$

Initial Value 0000 0000$_H$

31	30	29	28	27	26	25	24
0	0	0	DTS0BTR28				
R	R	R	R/W	R/W	R/W	R/W	R/W

23	22	21	20	19	18	17	16
DTS0BTR23	DTS0BTR22	DTS0BTR21	DTS0BTR20	DTS0BTR19	DTS0BTR18	DTS0BTR17	DTS0BTR16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8
DTS0BTR15	DTS0BTR14	DTS0BTR13	DTS0BTR12	DTS0BTR11	DTS0BTR10	DTS0BTR9	DTS0BTR8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7	6	5	4	3	2	1	0
DTS0BTR7	DTS0BTR6	DTS0BTR5	DTS0BTR4	DTS0BTR3	DTS0BTR2	DTS0BTR1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Table 10-36 DTS0BTR register contents

Bit position	Bit name	Function
28:1	–	This is a register for setting the start address of the table that is used by DTS to detect the TI position. Set the internal RAM area. Bits 31 to 29 and 0 are fixed to “0”. DTS uses DTS0BVR and this register to calculate the TI position corresponding to channels.

10.10.5 DTS0BVR: DTS base vector register [register group A]

Access This register can be read or written in 32-bit units.

Address FFFF 7C10$_H$

Initial Value 0000 0000$_H$

31	30	29	28	27	26	25	24
0	0	0	DTS0BVR28	DTS0BVR27	DTS0BVR26	DTS0BVR25	DTS0BVR24
R	R	R	R/W	R/W	R/W	R/W	R/W

23	22	21	20	19	18	17	16
DTS0BVR23	DTS0BVR22	DTS0BVR21	DTS0BVR20	DTS0BVR19	DTS0BVR18	DTS0BVR17	DTS0BVR16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8
DTS0BVR15	DTS0BVR14	DTS0BVR13	DTS0BVR12	DTS0BVR11	DTS0BVR10	DTS0BVR9	DTS0BVR8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7	6	5	4	3	2	1	0
DTS0BVR7	DTS0BVR6	DTS0BVR5	DTS0BVR4	DTS0BVR3	DTS0BVR2	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R	R

Table 10-37 DTS0BVR register contents

Bit position	Bit name	Function
28:2	–	This is a register for setting the start address of the vector that is used by DTS to detect the TI position. Set the internal RAM area. Bits 31 to 29, 1, and 0 are fixed to “0”. When a DTS request is generated, a 16-bit vector address is read from the table address calculated as “DTS0BTR + channel No. x 2”. The read value is added to this register's value to determine the TI's address.

10.10.6 DTS0ACR: DTS active channel register [register group A]

Access This register is read-only, in 16-bit units.

Address FFFF 7C14$_H$

Initial Value 0000$_H$

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	DTS0ACT
R	R	R	R	R	R	R	R

7	6	5	4	3	2	1	0
0	DTS0CH						
R	R	R	R	R	R	R	R

Table 10-38 DTS0ACR register contents

Bit position	Bit name	Function
8	DTS0ACT	DTS ACTive (DTS execution status) This bit indicates DTS's execution status. 0: DTS is not operating. 1: DTS is operating.
6:0	DTS0CH	CHannel (active channel) These bits indicate the active channel when the DTS0ACT bit is “1”. When DTS0ACT bit is “0”, the channel used for the previous execution is retained.

10.10.7 DTS0HST: DTS TI hold status register [register group A]

Access This register is read-only, in 8-bit units.

Address FFFF 7C18$_H$

Initial Value 00$_H$

7	6	5	4	3	2	1	0
0	0	0	0	DTS0TON3	DTS0TON2	DTS0TON1	DTS0TON0
R	R	R	R	R	R	R	R

Table 10-39 DTS0HST register contents

Bit position	Bit name	Function
3	DTS0TON3	DTS Ti-hold ON 3 (TI hold 3 ON) This bit checks whether TI is held in TI hold buffer 3. 0: TI is not held. 1: TI is held.
2	DTS0TON2	DTS Ti-hold ON 2 (TI hold 2 ON) This bit checks whether TI is held in TI hold buffer 2. 0: TI is not held. 1: TI is held.
1	DTS0TON1	DTS Ti-hold ON 1 (TI hold 1 ON) This bit checks whether TI is held in TI hold buffer 1. 0: TI is not held. 1: TI is held.
0	DTS0TON0	DTS Ti-hold ON 0 (TI hold 0 ON) This bit checks whether TI is held in TI hold buffer 0. 0: TI is not held. 1: TI is held.

10.10.8 DTS0HC: DTS TI hold channel number register [register group A]

Access This register is read-only, in 32-bit units.
However, when bits 30 to 24, 22 to 16, 14 to 8, and 6 to 0 in the DTS0HC register are used as the DTS0HC3, DTS0HC2, DTS0HC1, and DTS0HC0 registers respectively, these registers are read-only, in 8-bit units.

Address DTS0HC: FFFF 7C20$_H$
DTS0HC0: FFFF 7C20$_H$, DTS0HC1: FFFF 7C21$_H$, DTS0HC2: FFFF 7C22$_H$, DTS0HC3: FFFF 7C23$_H$

Initial Value 0000 0000$_H$

- DTS0HC

31	30	29	28	27	26	25	24
0	DTS0HC30	DTS0HC29	DTS0HC28	DTS0HC27	DTS0HC26	DTS0HC25	DTS0HC24
R	R	R	R	R	R	R	R

23	22	21	20	19	18	17	16
0	DTS0HC22	DTS0HC21	DTS0HC20	DTS0HC19	DTS0HC18	DTS0HC17	DTS0HC16
R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8
0	DTS0HC14	DTS0HC13	DTS0HC12	DTS0HC11	DTS0HC10	DTS0HC9	DTS0HC8
R	R	R	R	R	R	R	R

7	6	5	4	3	2	1	0
0	DTS0HC6	DTS0HC5	DTS0HC4	DTS0HC3	DTS0HC2	DTS0HC1	DTS0HC
R	R	R	R	R	R	R	R

- DTS0HC3

7	6	5	4	3	2	1	0
0	DTS0HC30	DTS0HC29	DTS0HC28	DTS0HC27	DTS0HC26	DTS0HC25	DTS0HC24
R	R	R	R	R	R	R	R

- DTS0HC2

7	6	5	4	3	2	1	0
0	DTS0HC22	DTS0HC21	DTS0HC20	DTS0HC19	DTS0HC18	DTS0HC17	DTS0HC16
R	R	R	R	R	R	R	R

- DTS0HC1

7	6	5	4	3	2	1	0
0	DTS0HC14	DTS0HC13	DTS0HC12	DTS0HC11	DTS0HC10	DTS0HC9	DTS0HC8
R	R	R	R	R	R	R	R

- DTS0HC0

7	6	5	4	3	2	1	0
0	DTS0HC6	DTS0HC5	DTS0HC4	DTS0HC3	DTS0HC2	DTS0HC1	DTS0HC
R	R	R	R	R	R	R	R

Table 10-40 DTS0HC register contents

Bit position	Bit name	Function
30:24	DTS0HC30 to DTS0HC24	These bits store the channel number of the TI that is held in TI hold buffer 3.
22:16	DTS0HC22 to DTS0HC16	These bits store the channel number of the TI that is held in TI hold buffer 2.
14:8	DTS0HC14 to DTS0HC8	These bits store the channel number of the TI that is held in TI hold buffer 1.
6:0	DTS0HC6 to DTS0HC0	These bits store the channel number of the TI that is held in TI hold buffer 0.

10.10.9 DTS0SAR: DTS source address register [register group B]

This register is a DTS internal register. It is neither write-accessible nor read-accessible from the CPU.

Access —

Address —

Initial Value $0000\ 0000_H$

31	30	29	28	27	26	25	24
IMM31	IMM30	IMM29	DTS0SAR[28:24] (IMM28-24)				

23	22	21	20	19	18	17	16
DTS0SAR[23:16] (IMM23-16)							

15	14	13	12	11	10	9	8
DTS0SAR[15:8] (IMM15-8)							

7	6	5	4	3	2	1	0
DTS0SAR[7:0] (IMM7-0)							

Bit position	Bit name	Function
31:29	IMM31 to IMM29	IMMediate (immediate bits) When using the immediate function, these bits and DTS0SAR bits are used in combination as a 32-bit register.
28:0	DTS0SAR28 to DTS0SAR0	DTS Source Address Register (source address register) These bits comprise the source address register. When using the immediate function, they function as IMM28 to IMM0.

Caution DTS transfer of misaligned data is not supported. The lower 2 bits of an address corresponding to the transfer data size are as follows (× indicates any bit).

Operation is not guaranteed if a setting other than the following is made.

Data size	DTS0SAR1	DTS0SAR0
8 bits	×	×
16 bits	×	0
32 bits	0	0

10.10.10 DTS0DAR: DTS destination address register [register group B]

This register is a DTS internal register. It is neither write-accessible nor read-accessible from the CPU.

Access —

Address —

Initial Value $0000\ 0000_H$

31	30	29	28	27	26	25	24
0	0	0	DTS0DAR[28:24]				

23	22	21	20	19	18	17	16
DTS0DAR[23:16]							

15	14	13	12	11	10	9	8
DTS0DAR[15:8]							

7	6	5	4	3	2	1	0
DTS0DAR[7:0]							

Table 10-41 DTS0DAR register contents

Bit position	Bit name	Function
28:0	DTS0DAR28 to DTS0DAR0	Destination Address Register (destination address register) These bits comprise the destination address register. Bits 31 to 29 are fixed to “0”.

Caution DTS transfer of misaligned data is not supported. The lower 2 bits of an address corresponding to the transfer data size are as follows (× indicates any bit).

Operation is not guaranteed if a setting other than the following is made.

Data size	DTS0DAR1	DTS0DAR0
8 bits	×	×
16 bits	×	0
32 bits	0	0

10.10.11 DTS0TCEA: DTS transfer counter or else address register [register group B]

This register is a DTS internal register. It is neither write-accessible nor read-accessible from the CPU.

Access —

Address —

Initial Value 0000_H

(1) Transfer count 255 mode

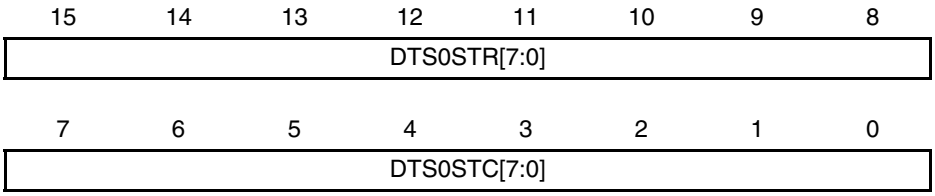


Table 10-42 DTS0TCEA register contents for transfer count 255 mode

Bit position	Bit name	Function
15:8	DTS0STR7 to DTS0STR0	Short Transfer Number Register (transfer count hold register) During transfer count 255 mode, this register holds the transfer count. If a DTS request is generated when the DTS0STC bit is “0”, the contents of this register are copied to the DTS0STC bit.
7:0	DTS0STC7 to DTS0STC0	Short Transfer Counter (short transfer count register) During transfer count 255 mode, this register counts the transfer count. The count is decremented once per DTS cycle. When 1 is set, one DTS is triggered, and when FFH is set, 255 DTS cycles are triggered. When the value of this register reaches “0”, an interrupt occurs (or the chain function can be selected). Continued execution of transfers can be selected. If a DTS request is generated when this register is “0”, the contents of the DTS0STR bits are copied to this register, and another decrementation count is performed.

(2) Transfer count 65535 mode

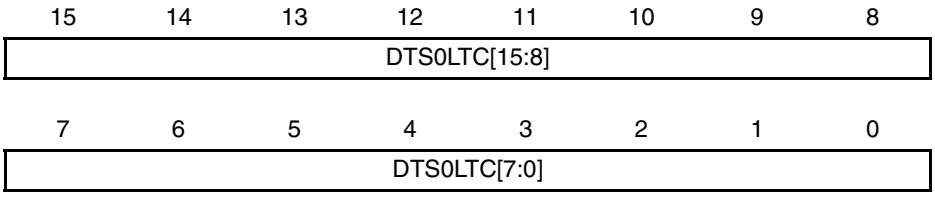


Table 10-43 DTS0TCEA register contents for transfer count 65,535 mode

Bit position	Bit name	Function
15 to 0	DTS0LTC15 to DTS0LTC0	Long Transfer Counter (long transfer count register) During transfer count 65535 mode, this register counts the transfer count. The count is decremented once per DTS cycle. When 1 is set, one DTS is triggered, and when FFFFH is set, 65,535 DTS cycles are triggered. When the value of this register reaches “0”, an interrupt occurs (or the chain function can be selected).

(3) Extended chain mode

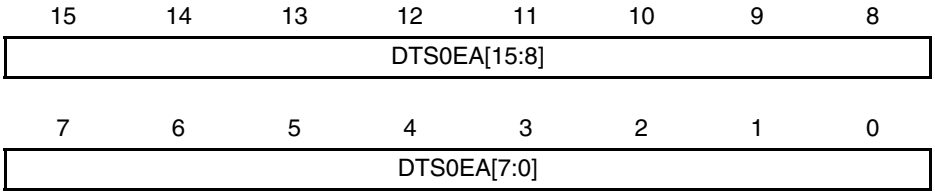


Table 10-44 DTS0TCEA register contents for extended chain mode

Bit position	Bit name	Function
15:0	DTS0EA15 to DTS0EA0	Else Address (ELSE address) During extended chain mode, this register specifies the position of TI that is read when the condition is false (ELSE). To specify the TI position, set an offset using this register from the current position (TI-A address where the extended chain is specified). The value is specified as a signed 16-bit value from –32,768 to 32,760 (lower 3 bits are 3'b000).

10.10.12 DTS0SCS: DTS source address count size register [register group B]

This register is a DTS internal register. It is neither write-accessible nor read-accessible from the CPU. When DTS0IMM in DTS0CIR is “1”, the following operations are performed.

- When the data size indicated by DTS0DS1 and DTS0DS0 in DTS0CIR is 32 bits:

  DTS0SCS becomes “0”.

- When the data size indicated by DTS0DS1 and DTS0DS0 in DTS0CIR is 16 bits:

  DTS0SAR (IMM15 to IMM0) is counted (incremented/decremented) by DTS0SCS.

- When the data size indicated by DTS0DS1 and DTS0DS0 in DTS0CIR is 8 bits:

  DTS0SAR (IMM7 to IMM0) is counted (incremented/decremented) by DTS0SCS.

However, when in SG mode, this register's settings are ignored.

Access —

Address —

Initial Value 00_H

7	6	5	4	3	2	1	0
0	0	0	0	0	DTS0SCS2	DTS0SCS1	DTS0SCS0

Table 10-45 DTS0SCS register contents

Bit position	Bit name	Function
2 1 0	DTS0SCS2 DTS0SCS1 DTS0SCS0	Specifies the count size of the source address. 000: 0 (FIX) 001: +1 010: +2 011: +4 101: –1 110: –2 111: –4

10.10.13 DTS0DCS: DTS destination address count size register [register group B]

This register is a DTS internal register. It is neither write-accessible nor read-accessible from the CPU.

This register's settings are ignored when in SG mode.

Access —

Address —

Initial Value 00_H

7	6	5	4	3	2	1	0
0	0	0	0	0	DTS0DCS2	DTS0DCS1	DTS0DCS0

Table 10-46 DTS0DCS register contents

Bit position	Bit name	Function
2 1 0	DTS0DCS2 DTS0DCS1 DTS0DCS0	Specifies the count size of the destination address. 000: 0 (FIX) 001: +1 010: +2 011: +4 101: –1 110: –2 111: –4

10.10.14 DTS0CIR: DTS control information register [register group B]

This register is a DTS internal register. It is neither write-accessible nor read-accessible from the CPU.

Access —

Address —

Initial Value 0000_H

15	14	13	12	11	10	9	8
DTS0CAF	DTS0DPE1	DTS0SPE1	DTS0DISI	DTS0CM1	DTS0CM0	DTS0DS1	DTS0DS0

7	6	5	4	3	2	1	0
DTS0BEN	DTS0IMM	DTS0TTYP5	DTS0TTYP4	DTS0TTYP3	DTS0TTYP2	DTS0TTYP1	DTS0TTYP0

Table 10-47 DTS0CIR register contents (1/2)

Bit position	Bit name	Function
15	DTS0CAF	Condition Accumulation Flag This is the flag check and comp check result accumulation flag. When the results of a flag check or comp check (comparison) are false, DTS0CAF is set to “1” and this value is written back to TI-A. If this bit is “1” when TI-A is read, then “1” must be written back. Condition for setting (to “1”): • When compare result is false. • When flag check result is false. • When DTS0CAF read data is “1” (bit 15 is “1” at TI-A fetch). In other words, when “1” is set to bit 15 of TI-A, the value of DTS0CAF remains “1”. Condition for clearing (to “0”): • When the DTS0CAF bit read data is “0” (bit 15 is “0” at TI-A fetch).
14	DTS0DPE1	Destination select PE1 (destination PE1 select) This bit specifies the address range on the destination side. 0: Selects external memory area, P-bus peripheral I/O area, and H-bus peripheral I/O area. 1: Selects internal flash memory and internal RAM.
13	DTS0SPE1	Source select PE1 (source PE1 select) This bit specifies the address range on the source side. 0: Selects external memory area, P-bus peripheral I/O area, and H-bus peripheral I/O area. 1: Selects internal flash memory and internal RAM.
12	DTS0DISI	DISable Interruption (interrupt output disabled) This bit prohibits interrupt requests for DTSFSL. 0: Interrupt enabled. 1: Interrupt disabled.
11 10	DTS0CM1 DTS0CM0	Chain Mode (chain mode) 00: Does not perform a chain of transfers. [if (counter==0) then Interrupt else no operation] 01: Performs a chain of transfers when counter = 0. [if (counter==0) then chain & Interrupt else no operation] 10: Performs a chain of transfers if true. [if (true) then chain else Interrupt] 11: Always perform a chain of transfers.

Table 10-47 DTS0CIR register contents (2/2)

Bit position	Bit name	Function
9 8	DTS0DS1 DTS0DS0	Data Size (data size) These bits set the data size for transfer, compare, and flag check. 00: 8 bits 01: 16 bits 10: 32 bits 11: Setting prohibited
7	DTS0BEN	Block Enable This is the block enable signal. 0: Single mode 1: Block mode
6	DTS0IMM	IMMediate enable (immediate function) This bit enables/disables the immediate function. When the immediate function is enabled, DTS0SAR functions as the immediate register. 0: Immediate disabled 1: Immediate enabled
5 4 3 2 1 0	DTS0TTYP5 DTS0TTYP4 DTS0TTYP3 DTS0TTYP2 DTS0TTYP1 DTS0TTYP0	Transfer TYPe (transfer type select) These bits select the transfer type. When a transfer type is selected, a special function and transfer count are selected at the same time. See *10.11.5 "Transfer types"* for the list of transfer types.

10.10.15 DTS0ECSRA: DTS extension address count size/repeat address registers [register group B]

These registers are DTS internal registers. They are neither write-accessible nor read-accessible from the CPU.

Access —

Address —

Initial Value $0000\ 0000_H$

(1) SG (Scatter & Gather) mode

31	30	29	28	27	26	25	24
DTS0EDCS[15:8]							

23	22	21	20	19	18	17	16
DTS0EDCS[7:0]							

15	14	13	12	11	10	9	8
DTS0ESCS[15:8]							

7	6	5	4	3	2	1	0
DTS0ESCS[7:0]							

Table 10-48 DTS0ECSRA register contents for SG (Scatter & Gather) mode

Bit position	Bit name	Function
31:16	DTS0EDCS15 to DTS0EDCS0	These bits specify the count size for the destination address. Any signed 16-bit value from –32,768 to 32,767 can be specified. This register is valid only when SG mode is selected by the DTS's special function. When not in SG mode, the count size is determined by DTS0DCS.
15:0	DTS0ESCS15 to DTS0ESCS0	This bit specifies the count size for the source address. Any signed 16-bit value from –32,768 to 32,767 can be specified. This register is valid only when SG mode is selected by the DTS's special function. When not in SG mode, the count size is determined by DTS0SCS.

(2) SREP (Source Repeat) mode

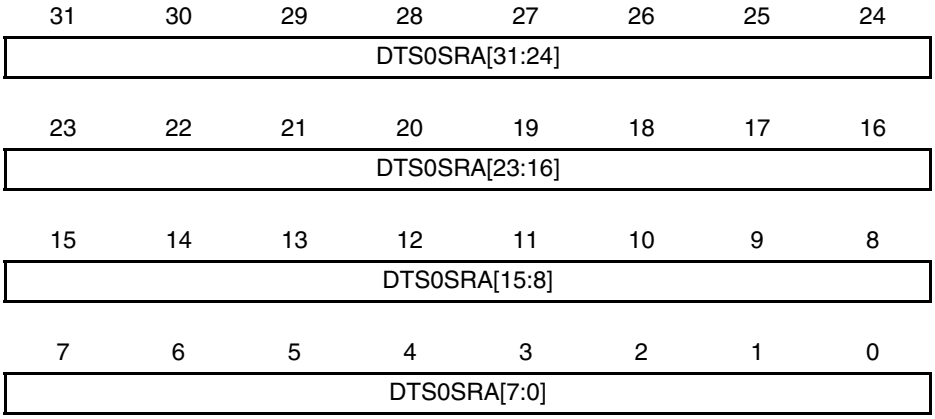


Table 10-49 DTS0ECSRA register contents for SREP (Source Repeat) mode

Bit position	Bit name	Function
31 to 0	DTS0SRA31 to DTS0SRA0	These bits comprise an area where the repeat address is held. The source repeat becomes valid only when SREP is selected by the DTS's special function.

(3) DREP (Destination Repeat) mode

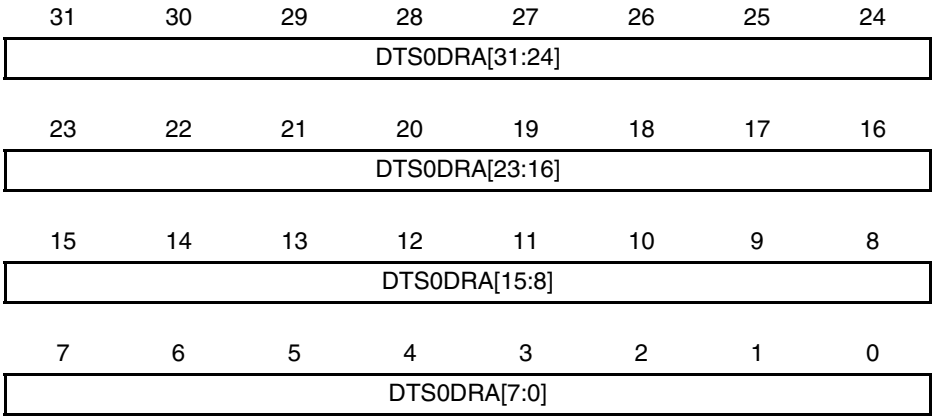


Table 10-50 DTS0ECSRA register contents for DREP (Destination Repeat) mode

Bit position	Bit name	Function
31 to 0	DTS0DRA31 to DTS0DRA0	These bits comprise an area where the repeat address is held. The destination repeat becomes valid only when DREP is selected by the DTS's special function.

10.11 DTS Function Details

10.11.1 Transfer information (TI)

DTS internal registers (register group B) cannot be operated directly from an external source. The DTS reads transfer information from internal RAM, and then sets values to the internal registers. This transfer information is called TI.

TI is configured of either 96 bits (TI-A to TI-C) or 128 bits (TI-A to TI-D), and the setting in DTS0CIR.DTS0TTYP5 to DTS0CIR.DTS0TTYP0 determines which type is used.

“TI0” refers to a set of “TI-A, TI-B, and TI-C” or a set of “TI-A, TI-B, TI-C, and TI-D”. “TI0, TI1, TI2” indicates there are three sets of “TI-A, TI-B, and TI-C” or “TI-A, TI-B, TI-C, and TI-D”.

(1) Configuration of TI

Each TI should be 32-bit aligned (the lower 2 bits of the address are 00).

The 128 bits from TI-A to TI-D should be allocated to internal RAM1 as one set.

Caution TI can only be allocated to internal RAM1, not internal RAM2.

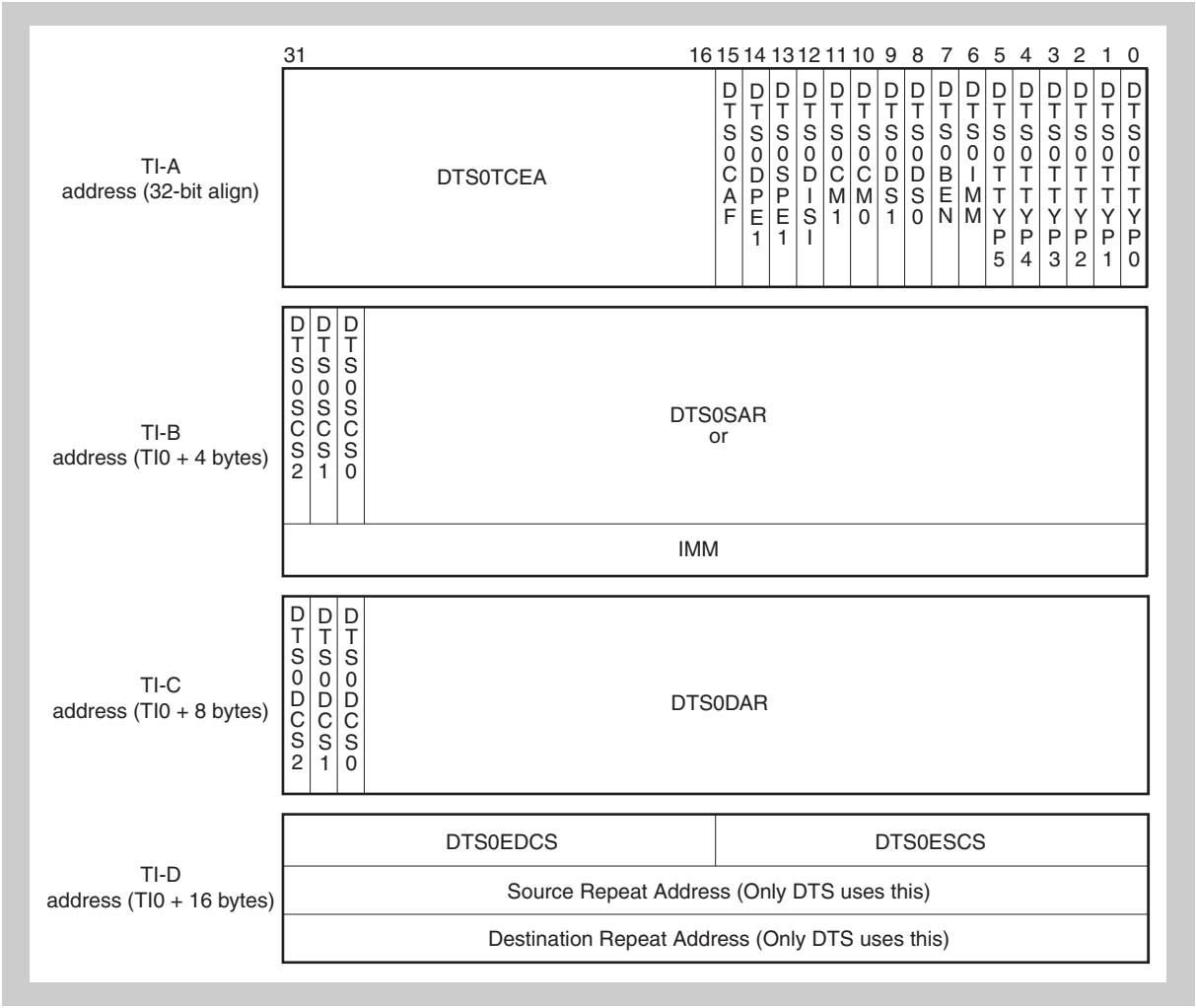


Figure 10-10 Configuration of TI

(2) Address where TI is allocated

The DTS uses DTS0BTR and DTS0BVR to determine the TI address corresponding to a channel.

TI can be freely allocated anywhere in the range from the address specified by DTS0BVR to the address to which FFFCH is added.

The method for determining DTS's TI address is as follows.

1. A 16-bit offset address is read from the table address [table address = DTS0BTR + channel number x 2] (the lower 2 bits of the offset address are ignored).
2. TI is read from the vector address [TI address = DTS0BVR + offset address]. In this case as well, the lower 2 bits are fixed to 00.

In the following figure, in which the request from CH3 is used as an example, the data for table address ($1EC0\_1000_H$ + 4 x 2) is 0320_H and the vector address is DTS0BVR + 0320_H.

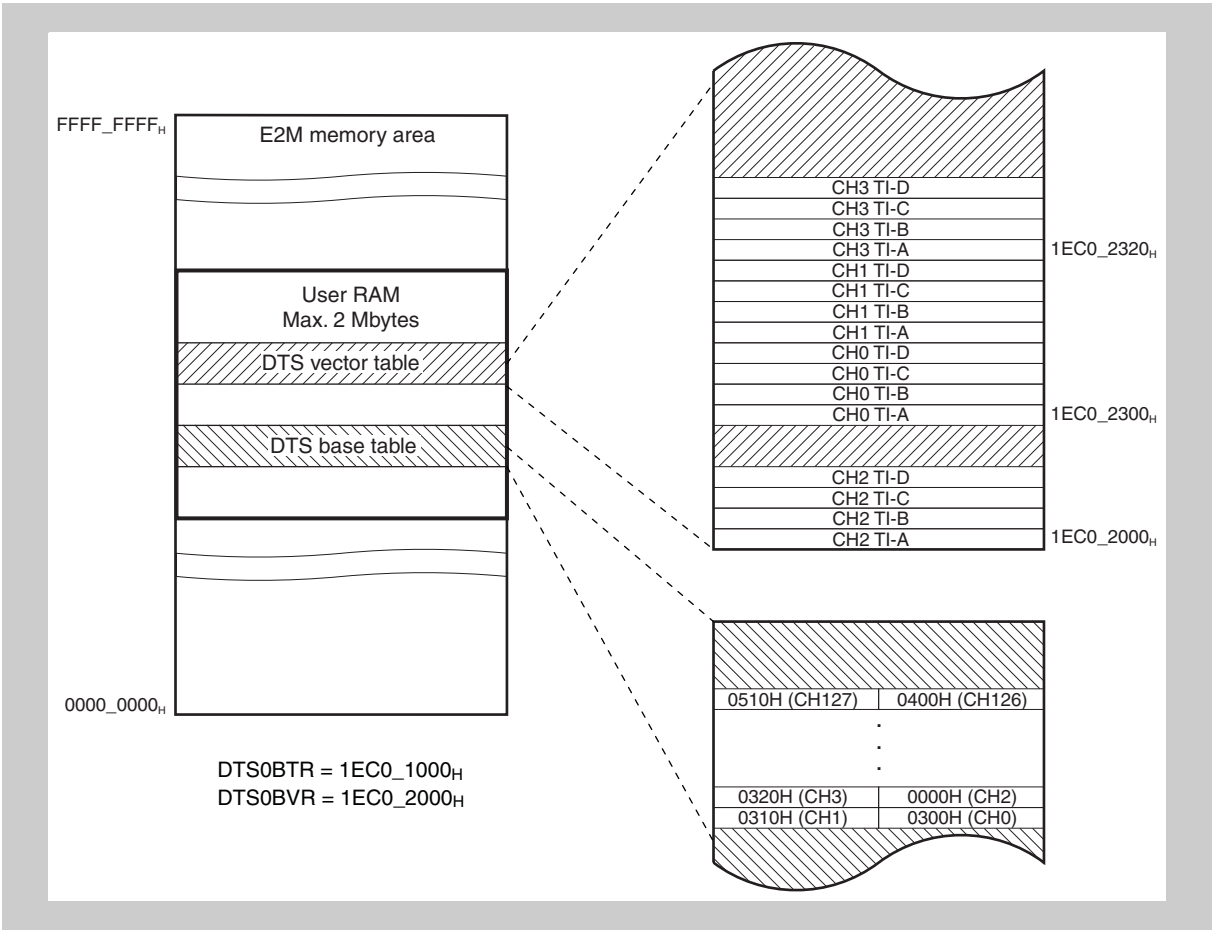


Figure 10-11 Example of address where TI is allocated

(3) Base table

The DTS references a base table to find the TI's position. This base table must be created by the user. The address set by DTS0BTR is the base table's start address, and 16 bits are allocated for each address, starting from channel 0. Data set to the table is offset from the DTS0BVR value (the offset range is 0 to 65,524 when using TI-A, TI-B, and TI-C and 0 to 65,520 when using TI-A, TI-B, TI-C, and TI-D).

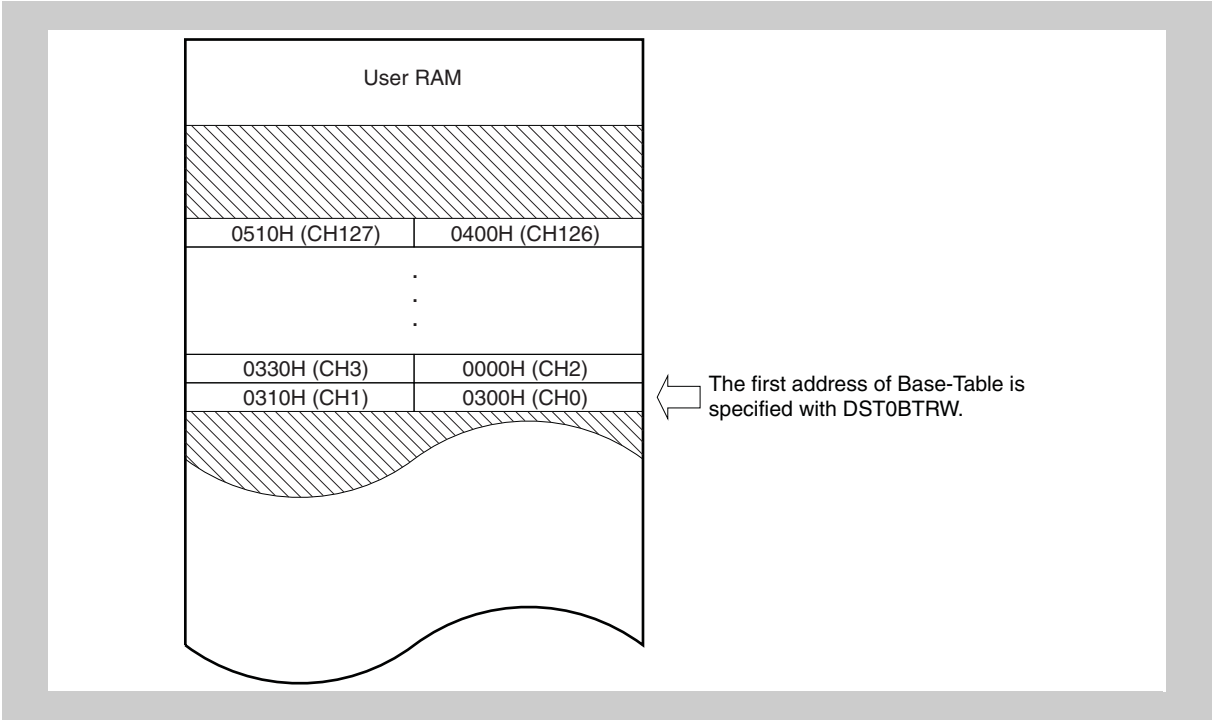


Figure 10-12 Description of base table

10.11.2 DTS transfer setting flow

Figure 10-13 "DTS transfer setting flow" shows the flow for setting DTS transfer by hardware when DTSEN is enabled after TI is set.

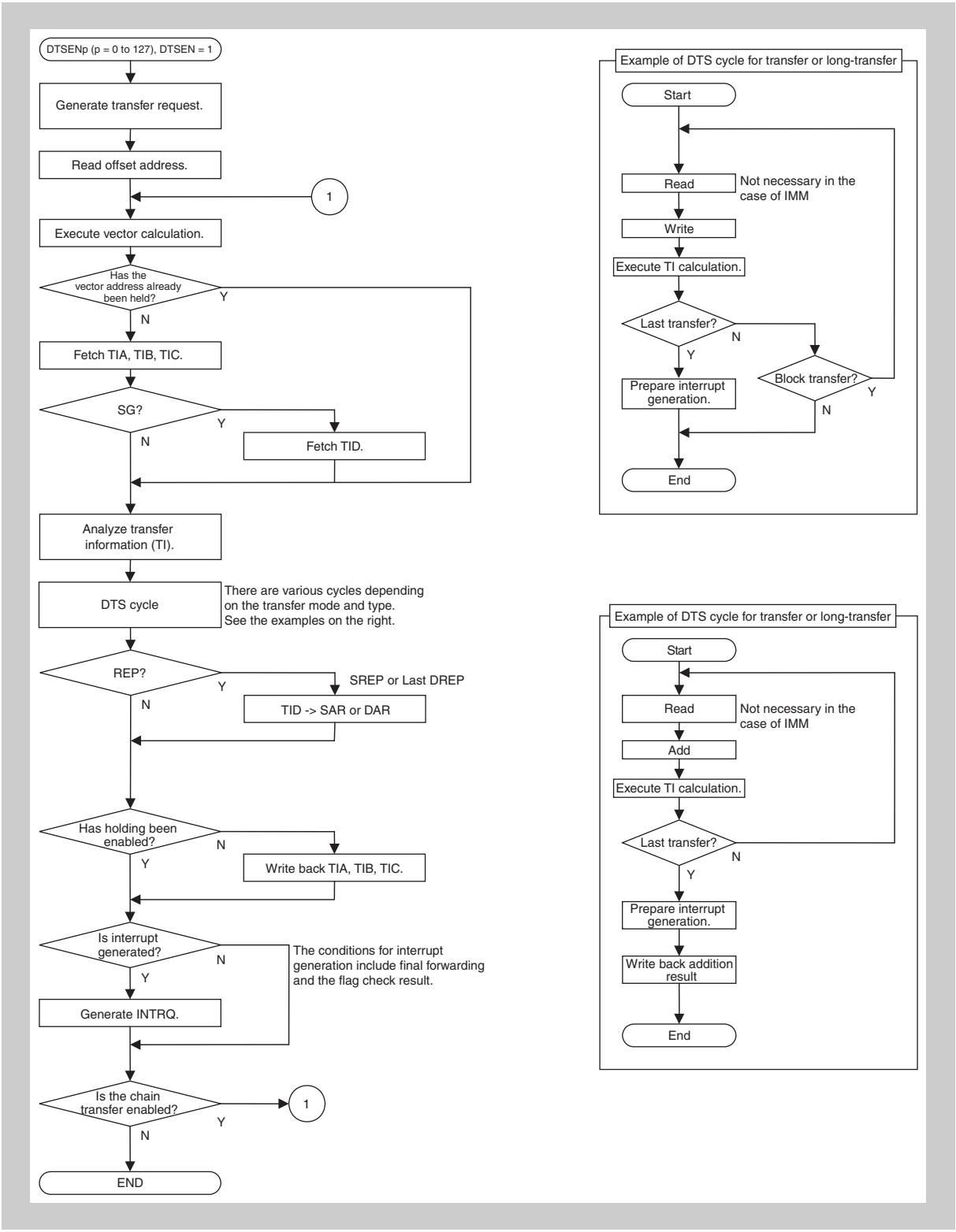


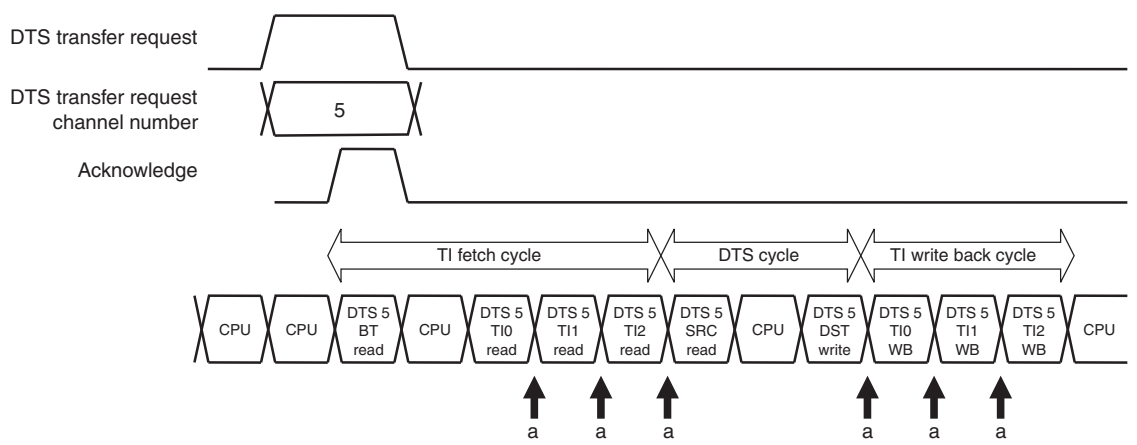
Figure 10-13 DTS transfer setting flow

10.11.3 Basic operations of DTS

The DTS fetches the information required for transfers (TI) from internal RAM, then executes transfers according to this TI, after which it updates the TI and writes it back to internal RAM.

(1) TI fetch and write back

For each DTS request, one round of “TI fetch cycle → DTS cycle → TI write back cycle” occurs. Since the bus is not locked, a CPU cycle may interrupt.



a) The bus is always released. When the CPU requests the bus mastership, the CPU acquires the bus mastership.

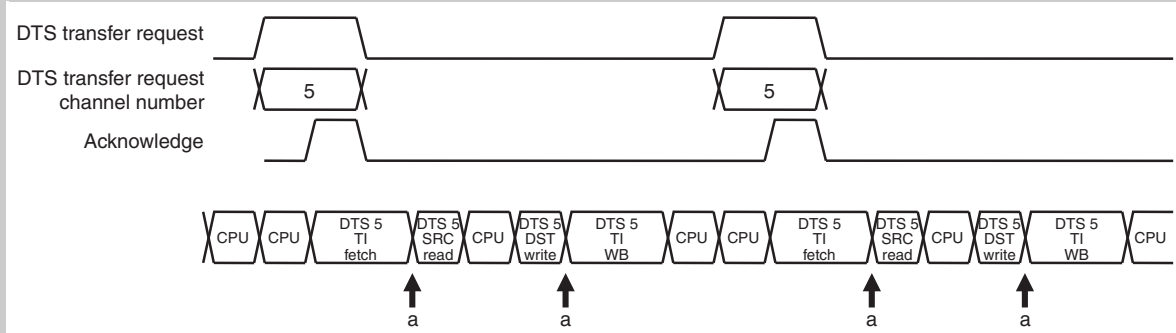
Figure 10-14 Example of TI fetch and write back

10.11.4 Transfer modes

A single-transfer mode and a block transfer mode are supported as transfer modes.

(1) Single transfer mode

When a DTS request is generated in single transfer mode, one round of “TI fetch → DTS cycle → TI write back” is performed and then transfer request wait status is set. When the next DTS request is generated, these cycles are executed again starting from a TI fetch. These operations are repeated until the value of DTS0STC (or DTS0LTC when in 65535 mode) reaches “0”. Since the bus is not locked, a CPU cycle may interrupt.



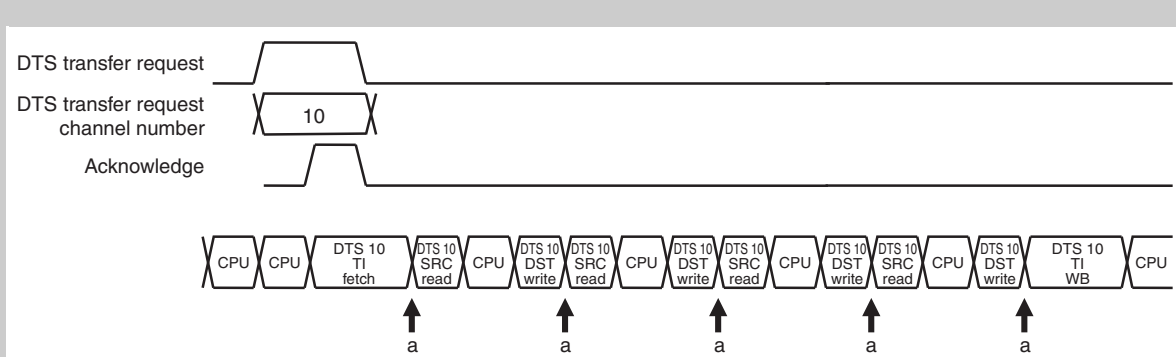
a) The bus is always released. When the CPU requests the bus mastership, the CPU acquires the bus mastership.

Figure 10-15 Example of single transfer

(2) Block transfer mode

When a DTS request is generated in block transfer mode, TI is fetched and then the DTS cycle is repeated until the value of DTS0STC (or DTS0LTC when in 65535 mode) reaches “0”, after which the TI is written back. Since the bus is not locked, a CPU cycle may interrupt.

Other DTS requests cannot be received during a block transfer (they are held pending in DTSFSL).



a) The bus is always released. When the CPU requests the bus mastership, the CPU acquires the bus mastership.

Figure 10-16 Example of block transfer

10.11.5 Transfer types

Transfer (2-cycle transfer), flag check, compare, and add are supported as transfer types. Although transfer types are specified by DTS0TTYP5 to DTS0TTYP0 in DTS0CIR, at the same time the transfer count and the special function[a] are also set.

The DTS transfer types are listed below. The operation of each transfer type varies depending on the setting of the DTS0IMM bit in DTS0CIR.

a) For a description of the special function, see *10.11.6 "Special functions" on page 468*.

Table 10-51 List of DTS transfer types (1/2)

DTS0TTYP5 to DTS0TTYP0			MAX transfer count	Transfer type	Special function	TI-D
0	000	00	255	Transfer	None	Not used
		01			SREP	Used
		10			DREP	
		11			SG	
0	001	00		Add	None	Not used
		01			SREP	Used
		10			DREP	
		11			SG	
0	010	00		flag[0]	None	Not used
		01			SREP	Used
		10			DREP	
		11			(SG)	
0	011	00		flag[1]	None	Not used
		01			SREP	Used
		10			DREP	
		11			(SG)	
0	100	00		comp[=]	None	Not used
		01			SREP	Used
		10			DREP	
		11			(SG)	
0	101	00		comp[!=]	None	Not used
		01			SREP	Used
		10			DREP	
		11			(SG)	
0	110	00		comp[<]	None	Not used
		01			SREP	Used
		10			DREP	
		11			(SG)	
0	111	00		comp[>]	None	Not used
		01			SREP	Used
		10			DREP	
		11			(SG)	

Table 10-51 List of DTS transfer types (2/2)

DTS0TTYP5 to DTS0TTYP0			MAX transfer count	Transfer type	Special function	TI-D
1	000	00	65536	Long transfer	None	Not used
		01			(SREP)	Used
		10			(DREP)	
		11			SG	
1	001	00		Long add	None	Not used
		01			(SREP)	Used
		10			(DREP)	
		11			SG	
1	010	××	—	flag[0]	Extended chain	Not used
	011			flag[1]		
	100			comp[=]		
	101			comp[!=]		
	110			comp[<]		
	111			comp[>]		

(1) Transfer: DTS0TTYP5 to DTS0TTYP2 are 0000 or 1000

- When DTS0TTYP5 to DTS0TTYP2 are 0000 and DTS0CIR.DTS0IMM is “0”

  When the transfer type is set to “transfer”, the DTS cycle is used for data transfer. DTS0TCEA is split into DTS0STR and DTS0STC, and DTS0STC can be used to specify a transfer count value up to 255 times. When the value of DTS0STC reaches “0”, the value of DTS0STR is copied to DTS0STC and another transfer can be performed.

  The chain mode specification can be 00 (no chain), 01 (transfer count 0 chain), or 11 (always chain), and SREP (Source Repeat), DREP (Destination Repeat), or SG (Scatter & Gather) can be selected as the special function.

Operation Data read from address SAR is written to address DAR.

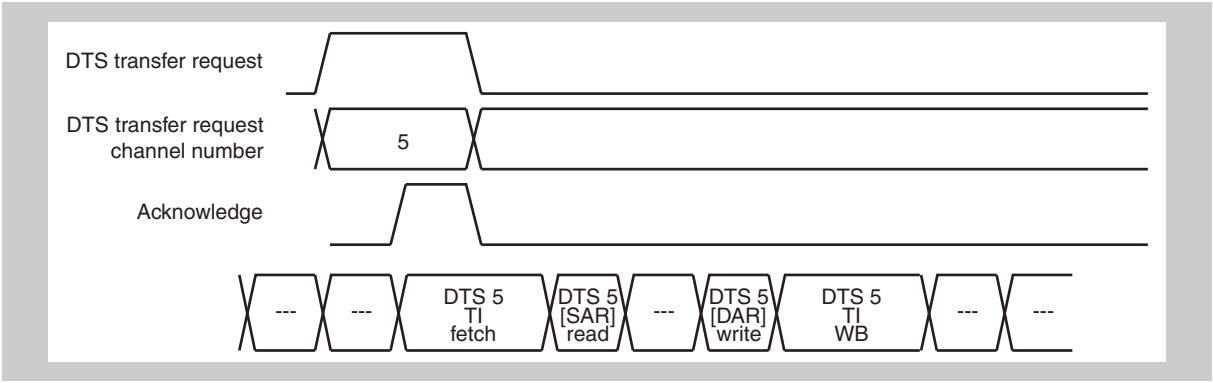


Figure 10-17 When DTS0TTYP5 to DTS0TYYP2 are 0000 and DTS0CIR.DTS0IMM is “0”

- When DTS0TTYP5 to DTS0TYYP2 are 0000 and immediate function is enabled (DTS0CIR.DTS0IMM is “1”)

  When DTS0IMM is “1”, DTS0SAR becomes IMM and the DTS0SAR value is written to address DTS0DAR.

  DTS0TCEA is split into DTS0STR and DTS0STC, and DTS0STC can be used to specify a transfer count value up to 255 times. When the value of DTS0STC reaches “0”, the value of DTS0STR is copied to DTS0STC and another transfer can be performed.

  The chain mode specification can be 00, 01, or 11, and DREP or SG can be selected as the special function.

Operation The SAR data is written to address DAR without modification.

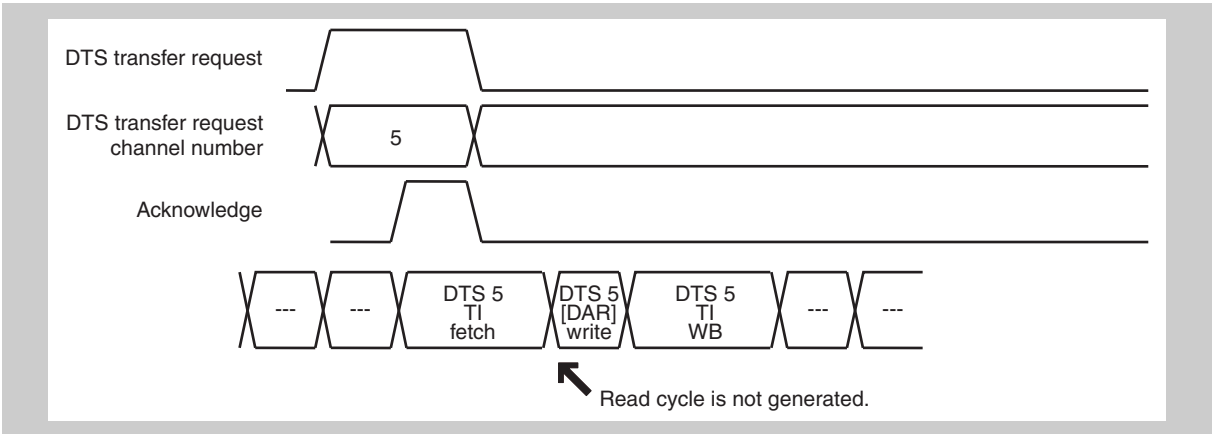


Figure 10-18 When DTS0TTYP5 to DTS0TYYP2 are 0000 and immediate function is enabled (DTS0CIR.DTS0IMM is “1”)

- When DTS0TTYP5 to DTS0TTYP2 are 1000 and DTS0CIR.DTS0IMM is “0”

  When the transfer type is set to “long transfer”, the DTS cycle is used for data transfer. DTS0TCEA becomes DTS0LTC and a transfer count up to 65,535 times can be specified. Once the value of DTS0LTC reaches “0”, no more transfer requests can be accepted. Before executing a transfer, set DTS0LTC (TIA's DTS0TCEA) again.

  The chain mode specification can be 00, 01, or 11, and SG can be selected as the special function.

Operation Data read from address SAR is written to address DAR.

The transfer cycle is the same as *Figure 10-17 “When DTS0TTYP5 to DTS0TYYP2 are 0000 and DTS0CIR.DTS0IMM is “0””*.

- When DTS0TTYP5 to DTS0TYYP2 are 1000 and immediate function is enabled (DTS0CIR.DTS0IMM is “1”)

  When DTS0IMM is “1”, DTS0SAR becomes IMM and the DTS0SAR value is written to address DTS0DAR.

  DTS0TCEA becomes DTS0LTC and a transfer count up to 65,535 times can be specified. When the value of DTS0LTC reaches “0”, no more transfer requests can be accepted. Before executing a transfer, set DTS0LTC (TIA's DTS0TCEA) again.

  The chain mode specification can be 00, 01, or 11, and SG can be selected as the special function.

Operation The SAR data is written to address DAR without modification.

The transfer cycle is the same as *Figure 10-18 “When DTS0TTYP5 to DTS0TYYP2 are 0000 and immediate function is enabled (DTS0CIR.DTS0IMM is “1”)”*.

(2) Add: When DTS0TTYP5 to DTS0TTYP2 are 0001 or 1001

- When DTS0TTYP5 to DTS0TTYP2 are 0001 and DTS0CIR.DTS0IMM is “0”

  When the transfer type is set to “add”, the DTS cycle becomes an accumulation cycle.

  DTS0TCEA is split into DTS0STR and DTS0STC, and DTS0STC can be used to specify a transfer count value up to 255 times.

  When the value of DTS0STC reaches “0”, the value of DTS0STR is copied to DTS0STC and another transfer can be performed.

  The chain mode specification can be 00 or 01, and SREP, DREP, or SG can be selected as the special function.

  DTS0BEN in DTS0CIR is ignored and block transfer mode is always set.

Operation Data is read from address SAR until DTS0STC reaches “0”.
The read data is added each time as unsigned data.
When DTS0STC reaches “0”, the addition result is written to address DAR.

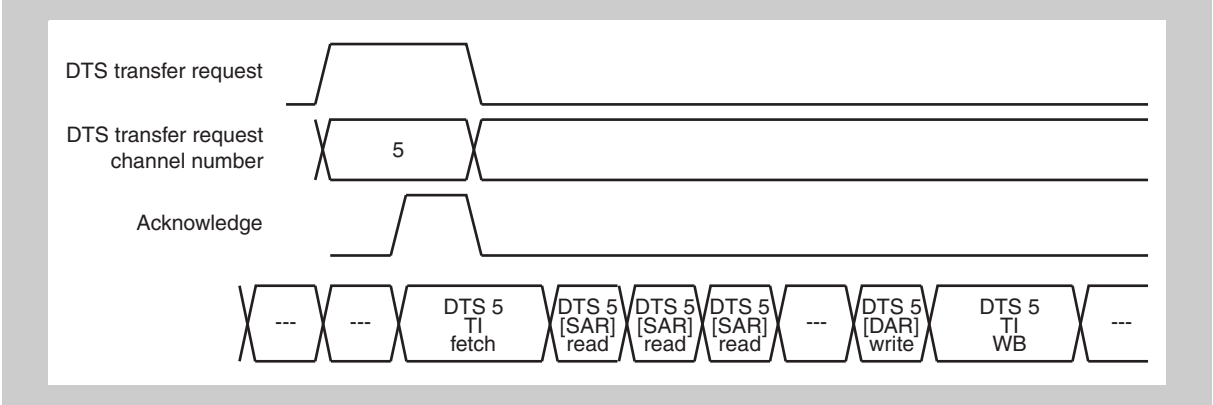


Figure 10-19 When DTS0TTYP5 to DTSTTYP2 are 0001 and DTS0CIR.DTS0IMM is “0”

The cumulative value is stored to a 32-bit register regardless of the data size, and when the transfer count reaches “0”, data is written to DAR in 32-bit units. Any data exceeding 32 bits is truncated.

Accordingly, when the transfer size is 32 bits, if an overflow or borrow occurs the calculation result will not be correct. When the transfer size is 16 bits or 8 bits, an overflow or borrow will not occur if the maximum transfer count of 65,535 times is specified.

- When DTS0TTYP5 to DTS0TYYP2 are 0001 and immediate function is enabled (DTS0CIR.DTS0IMM is “1”)

  When DTS0IMM is “1”, DTS0SAR becomes IMM.

  DTS0TCEA is split into DTS0STR and DTS0STC, and DTS0STC can be used to specify a transfer count value up to 255 times.

  DTS0BEN in DTS0CIR is ignored and block transfer mode is always set. In other words, the DTS0SAR value is added the number of times specified by DTS0STC, and when the value of DTS0STC reaches “0”, the result is written to address DAR.

  The chain mode specification can be 00 or 01, and DREP or SG can be selected as the special function.

- When DTS0TTYP5 to DTS0TTYP2 are 1001 and DTS0CIR.DTS0IMM is “0”

  When the transfer type is set to “long add”, the DTS cycle becomes an accumulation cycle. DTS0TCEA becomes DTS0LTC and a transfer count up to 65,535 times can be specified. When the value of DTS0LTC reaches “0”, no more transfer requests can be accepted.

  Before executing a transfer, set DTS0LTC (TIA's DTS0TCEA) again.

  The chain mode specification can be 00 or 01, and SG can be selected as the special function.

  DTS0BEN in DTS0CIR is ignored and block transfer mode is always set.

- When DTS0TTYP5 to DTS0TYYP2 are 1001 and immediate function is enabled (DTS0CIR.DTS0IMM is “1”)

  When DTS0IMM is “1”, DTS0SAR becomes IMM.

  DTS0TCEA becomes DTS0LTC and a transfer count up to 65,535 times can be specified.

  DTS0BEN in DTS0CIR is ignored and block transfer mode is always set. In other words, the DTS0SAR value is added the number of times specified by DTS0LTC, and when the value of DTS0LTC reaches “0”, the addition result is written to address DAR.

  The chain mode specification can be 00, or 01, and SG can be selected as the special function.

(3) flag[0]: When DTS0TTYP5 to DTS0TTYP2 are 0010 or 1010

- When DTS0TTYP5 to DTS0TTYP2 are 0010 and DTS0CIR.DTS0IMM is “0”

  When the transfer type is set to “flag [0]”, the DTS cycle becomes a flag check [0] cycle.

  DTS0TCEA is split into DTS0STR and DTS0STC, and DTS0STC can be used to specify a transfer count value up to 255 times.

  When the value of DTS0STC reaches “0”, the value of DTS0STR is copied to DTS0STC and another transfer can be performed.

  A special function (SREP, DREP, or SG) and all chain modes can be selected.

  The function of flag check [0] is to check whether an arbitrary bit in the data read from address DAR is “0”.

  If at least one bit is “0”, the check result is “true”. The check result can be used as a chain condition or as interrupt output.

Operation First, the check bit specification data is read from address SAR, and then the data to be checked is read from address DAR to execute a flag check.

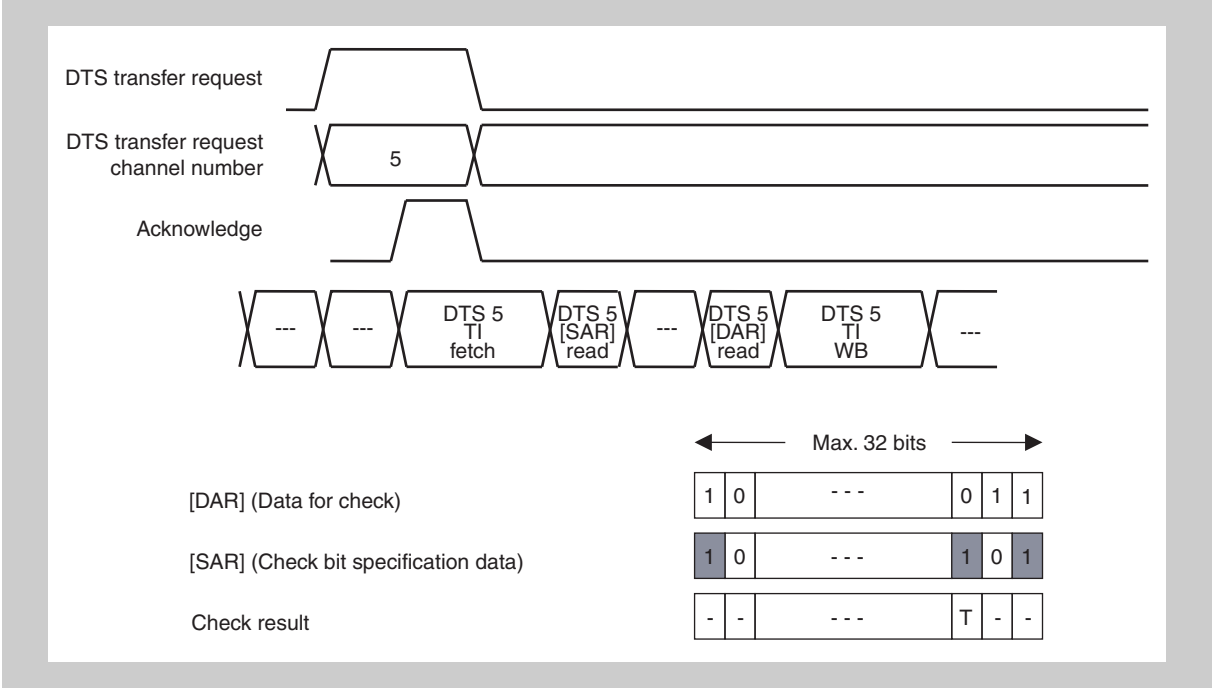


Figure 10-20 When DTS0TTYP5 to DTS0TTYP2 are 0010 and DTS0CIR.DTS0IMM is “0”

- When DTS0TTYP5 to DTS0TYYP2 are 0010 and immediate function is enabled (DTS0CIR.DTS0IMM is “1”)

  When DTS0IMM is “1”, DTS0SAR becomes IMM, and DTS0SAR becomes the check bit.

  DTS0TCEA is split into DTS0STR and DTS0STC, and DTS0STC can be used to specify a transfer count value up to 255 times.

  When the value of DTS0STC reaches “0”, the value of DTS0STR is copied to DTS0STC and another transfer can be performed.

  A special function (DREP or SG) and all chain modes can be selected.

  The function of flag check [0] is to check whether an arbitrary bit in the data read from address DAR is “0”.

  If at least one bit is “0”, the check result is “true”.

  The check result can be used as a chain condition or as interrupt output.

Operation The data to be checked is read from address DAR, and a flag check is executed using the SAR value as the check bit specification data.

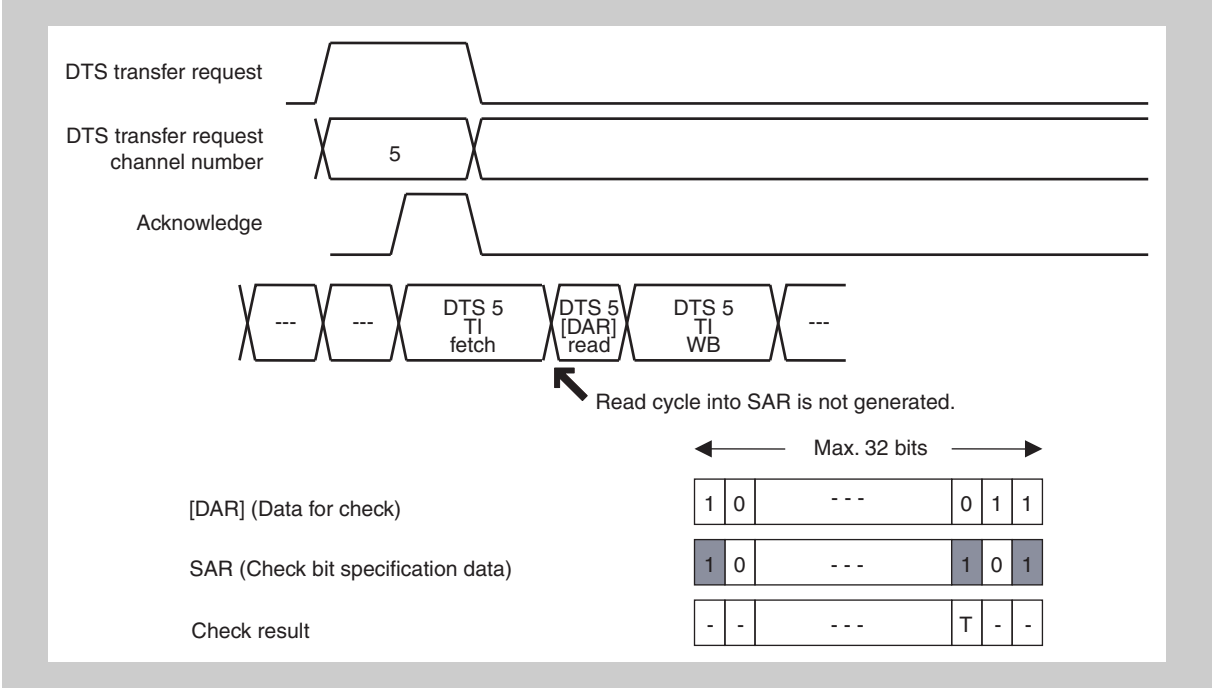


Figure 10-21 When DTS0TTYP5 to DTS0TYYP2 are 0010 and immediate function is enabled (DTS0CIR.DTS0IMM is “1”)

- When DTS0TTYP5 to DTS0TTYP2 are 1010 and DTS0CIR.DTS0IMM is “0”

  When the transfer type is set to “extended flag[0]”, the DTS cycle becomes a flag check [0] cycle.

  DTS0TCEA becomes DTS0EA, functioning as a register that specifies an offset address used when the check result is “false”.

  Accordingly, the transfer count cannot be specified. When the check result is “true”, TI is fetched from the current TI fetch address added to 10H to perform a chain of transfers. When the check result is “false”, TI is fetched from the current TI fetch address added to the value of DTS0EA to perform a chain of transfers.

  Neither the chain mode nor a special function can be selected.

Operation First, the check bit specification data is read from address SAR, and then the data to be checked is read from address DAR to execute a flag check.
A chain of transfers is always performed.

- When DTS0TTYP5 to DTS0TYYP2 are 1010 and immediate function is enabled (DTS0CIR.DTS0IMM is “1”)

  When DTS0IMM is “1”, DTS0SAR becomes IMM, and DTS0SAR becomes the check bit.

  DTS0TCEA becomes DTS0EA, functioning as a register that specifies an offset address used when the check result is “false”.

  Accordingly, the transfer count cannot be specified. When the check result is “true”, TI is fetched from the current TI fetch address added to 10H to perform a chain of transfers. When the check result is “false”, TI is fetched from the current TI fetch address added to the value of DTS0EA to perform a chain of transfers.

  Neither the chain mode nor a special function can be selected.

Operation The data to be checked is read from address DAR, and a flag check is executed using the SAR value as the check bit specification data.
A chain of transfers is always performed.

(4) flag[1]: When DTS0TTYP5 to DTS0TTYP2 are 0011 or 1011

- When DTS0TTYP5 to DTS0TTYP2 are 0011 and DTS0CIR.DTS0IMM is “0”

  When the transfer type is set to “flag [1]”, the DTS cycle becomes a flag check [1] cycle.

  DTS0TCEA is split into DTS0STR and DTS0STC, and DTS0STC can be used to specify a transfer count value up to 255 times.

  When the value of DTS0STC reaches “0”, the value of DTS0STR is copied to DTS0STC and another transfer can be performed.

  A special function (SREP, DREP, or SG) and all chain modes can be selected. The function of flag check [1] is to check whether an arbitrary bit in the data read from address DAR is “1”.

  If at least one bit is “1”, the check result is “true”.

  The check result can be used as a chain condition or as interrupt output.

Operation First, the check bit specification data is read from address SAR, and then the data to be checked is read from address DAR to execute a flag check.

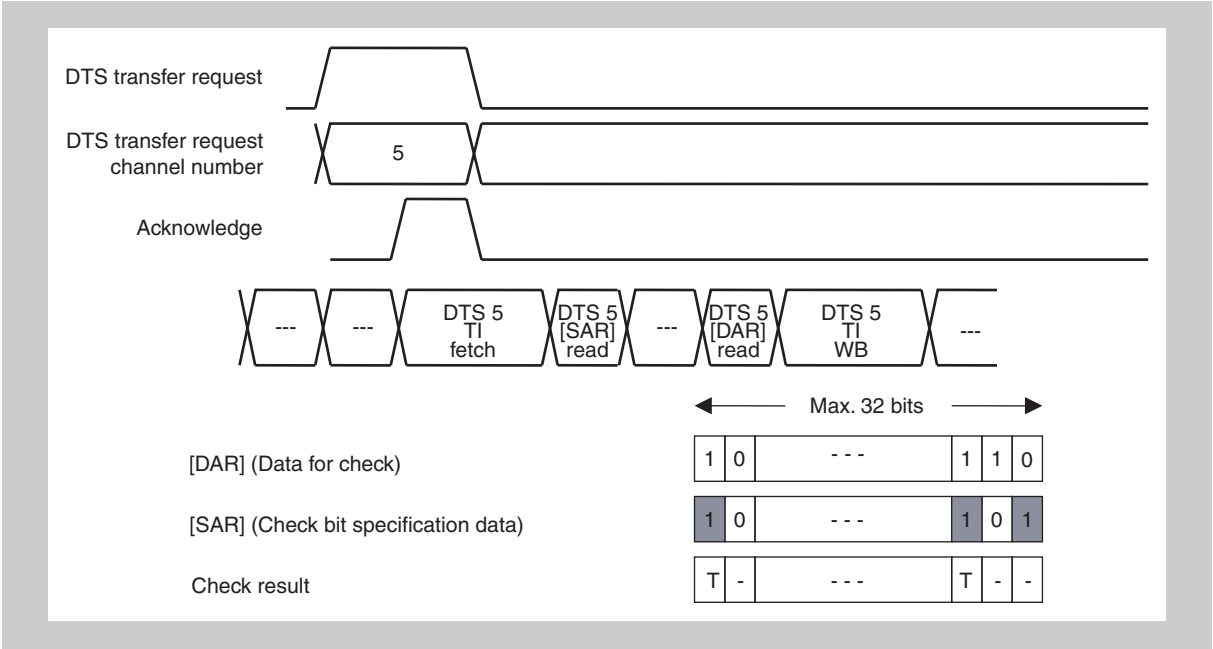


Figure 10-22 When DTS0TTYP5 to DTS0TTYP2 are 0011 and DTS0CIR.DTS0IMM is “0”

- When DTS0TTYP5 to DTS0TYYP2 are 0011 and immediate function is enabled (DTS0CIR.DTS0IMM is “1”)

  When DTS0IMM is “1”, DTS0SAR becomes IMM, and DTS0SAR becomes the check bit.

  DTS0TCEA is split into DTS0STR and DTS0STC, and DTS0STC can be used to specify a transfer count value up to 255 times.

  When the value of DTS0STC reaches “0”, the value of DTS0STR is copied to DTS0STC and another transfer can be performed.

  A special function (DREP or SG) and all chain modes can be selected.

  The function of flag check [1] is to check whether an arbitrary bit in the data read from address DAR is “1”.

  If at least one bit is “1”, the check result is “true”. The check result can be used as a chain condition or as interrupt output.

Operation The data to be checked is read from address DAR, and a flag check is executed using the SAR value as the check bit specification data.

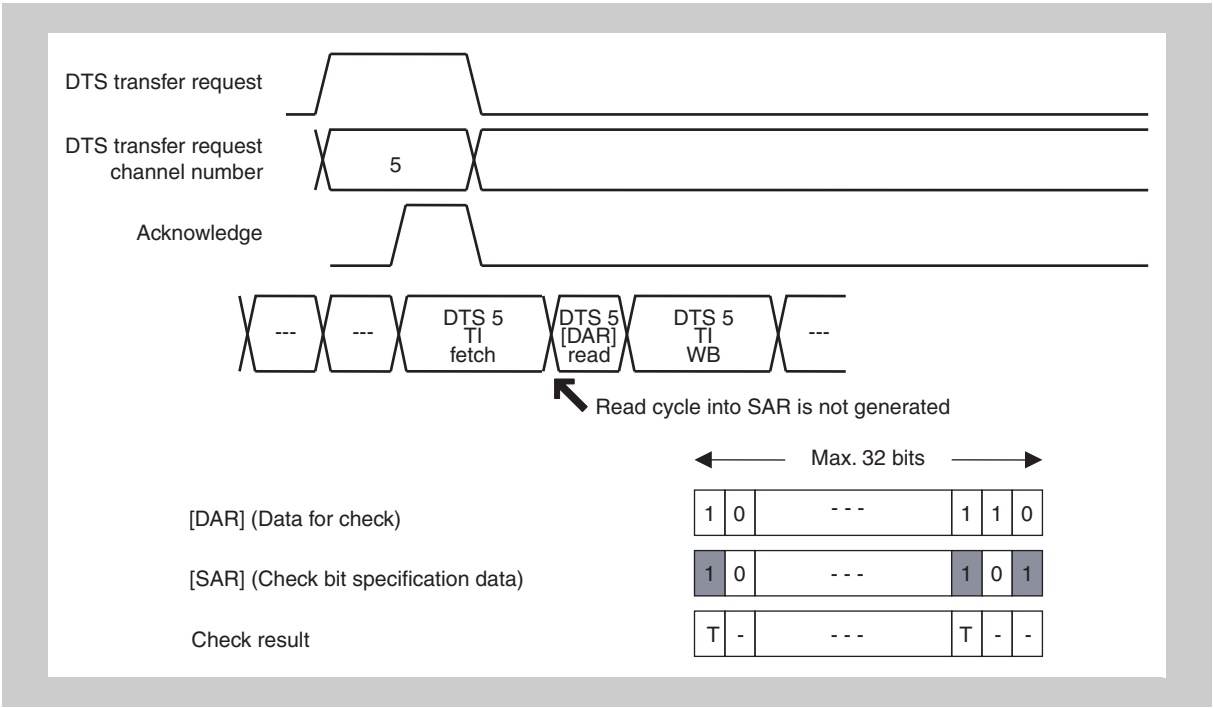


Figure 10-23 When DTS0TTYP5 to DTS0TYYP2 are 0011 and immediate function is enabled (DTS0CIR.DTS0IMM is “1”)

- When DTS0TTYP5 to DTS0TTYP2 are 1011 and DTS0CIR.DTS0IMM is “0”

  When the transfer type is set to “extended flag[1]”, the DTS cycle becomes a flag check [1] cycle.

  DTS0TCEA becomes DTS0EA, functioning as a register that specifies an offset address used when the check result is “false”.

  Accordingly, the transfer count cannot be specified. When the check result is “true”, TI is fetched from the current TI fetch address added to 10H to perform a chain of transfers. When the check result is “false”, TI is fetched from the current TI fetch address added to the value of DTS0EA to perform a chain of transfers.

  Neither the chain mode nor a special function can be selected.

Operation First, the check bit specification data is read from address SAR, and then the data to be checked is read from address DAR to execute a flag check.
A chain of transfers is always performed.

- When DTS0TTYP5 to DTS0TYYP2 are 1011 and immediate function is enabled (DTS0CIR.DTS0IMM is “1”)

  When DTS0IMM is “1”, DTS0SAR becomes IMM, and DTS0SAR becomes the check bit.

  DTS0TCEA becomes DTS0EA, functioning as a register that specifies an offset address used when the check result is “false”.

  Accordingly, the transfer count cannot be specified. When the check result is “true”, TI is fetched from the current TI fetch address added to 10H to perform a chain of transfers. When the check result is “false”, TI is fetched from the current TI fetch address added to the value of DTS0EA to perform a chain of transfers.

  Neither the chain mode nor a special function can be selected.

Operation The data to be checked is read from address DAR, and a flag check is executed using the SAR value as the check bit specification data.
A chain of transfers is always performed.

(5) comp[=]: When DTS0TTYP5 to DTS0TTYP2 are 0100 or 1100

- When DTS0TTYP5 to DTS0TTYP2 are 0100 and DTS0CIR.DTS0IMM is “0”

  When the transfer type is set to “comp[=]”, the DTS cycle becomes a comparison [=] cycle. DTS0TCEA is split into DTS0STR and DTS0STC, and DTS0STC can be used to specify a transfer count value up to 255 times.

  When the value of DTS0STC reaches “0”, the value of DTS0STR is copied to DTS0STC and another transfer can be performed.

  A special function (SREP, DREP, or SG) and all chain modes can be selected.

  With comp[=], data read from address SAR is compared with data read from address DAR, and if these data match, the compare result is “true”.

  The check result can be used as a chain condition or as interrupt output.

Operation First, data is read from address SAR, then data is read from address DAR. Next, whether [SAR] == [DAR] is checked.

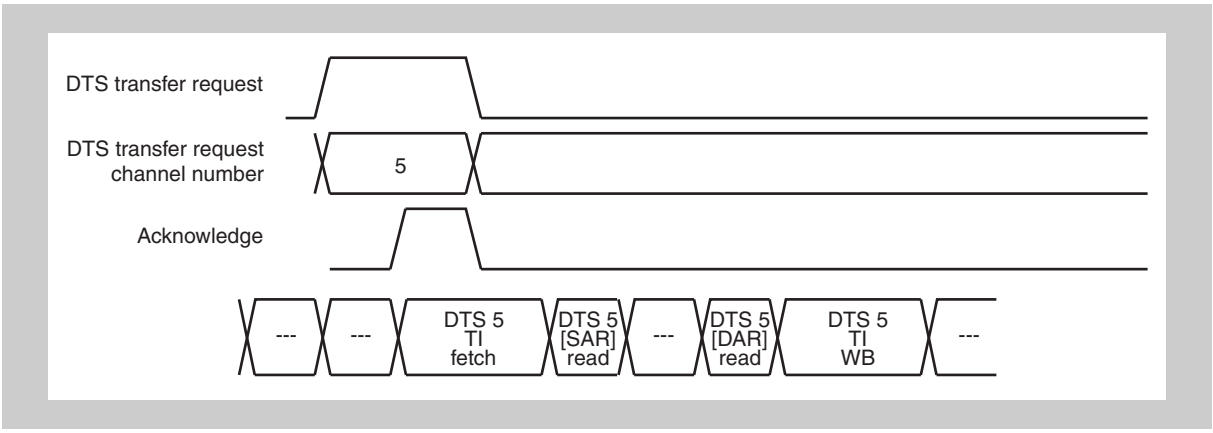


Figure 10-24 When DTS0TTYP5 to DTS0TTYP2 are 0100 and DTS0CIR.DTS0IMM is “0”

- When DTS0TTYP5 to DTS0TYYP2 are 0100 and immediate function is enabled (DTS0CIR.DTS0IMM is “1”)

  When DTS0IMM is “1”, DTS0SAR becomes IMM, and DTS0SAR becomes the data to be compared.

  DTS0TCEA is split into DTS0STR and DTS0STC, and DTS0STC can be used to specify a transfer count value up to 255 times.

  When the value of DTS0STC reaches “0”, the value of DTS0STR is copied to DTS0STC and another transfer can be performed.

  A special function (DREP or SG) and all chain modes can be selected.

  The check result can be used as a chain condition or as interrupt output.

Operation Data is read from address DAR, and whether SAR == [DAR] is checked.

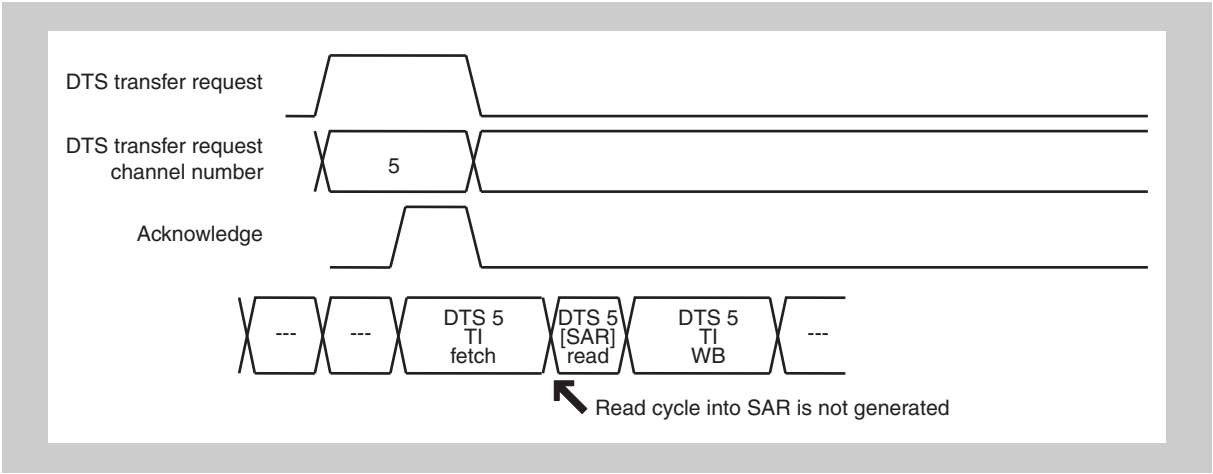


Figure 10-25 When DTS0TTYP5 to DTS0TYYP2 are 0100 and immediate function is enabled (DTS0CIR.DTS0IMM is “1”)

- When DTS0TTYP5 to DTS0TTYP2 are 1100 and DTS0CIR.DTS0IMM is “0”

  When the transfer type is set to “extended comp[=]”, the DTS cycle becomes a comparison [=] cycle.

  DTS0TCEA becomes DTS0EA, functioning as a register that specifies an offset address used when the check result is “false”.

  Accordingly, the transfer count cannot be specified. When the check result is “true”, TI is fetched from the current TI fetch address added to 10H to perform a chain of transfers. When the check result is “false”, TI is fetched from the current TI fetch address added to the value of DTS0EA to perform a chain of transfers.

  Neither the chain mode nor a special function can be selected.

Operation First, data is read from address SAR, then data is read from address DAR. Next, whether [SAR] == [DAR] is checked.
A chain of transfers is always performed.

- When DTS0TTYP5 to DTS0TYYP2 are 1100 and immediate function is enabled (DTS0CIR.DTS0IMM is “1”)

  When DTS0IMM is “1”, DTS0SAR becomes IMM, and DTS0SAR becomes the data to be compared.

  DTS0TCEA becomes DTS0EA, functioning as a register that specifies an offset address used when the check result is “false”.

  Accordingly, the transfer count cannot be specified. When the check result is “true”, TI is fetched from the current TI fetch address added to 10H to perform a chain of transfers. When the check result is “false”, TI is fetched from the current TI fetch address added to the value of DTS0EA to perform a chain of transfers.

  Neither the chain mode nor a special function can be selected.

Operation The data to be compared is read from address DAR, and then compared with the SAR value.
A chain of transfers is always performed.

(6) comp[!=]: When DTS0TTYP5 to DTS0TTYP2 are 0101 or 1101

- When DTS0TTYP5 to DTS0TTYP2 are 0101 and DTS0CIR.DTS0IMM is “0”

  When the transfer type is set to “comp[!=]”, the DTS cycle becomes a comparison [!=] cycle.

  DTS0TCEA is split into DTS0STR and DTS0STC, and DTS0STC can be used to specify a transfer count value up to 255 times.

  When the value of DTS0STC reaches “0”, the value of DTS0STR is copied to DTS0STC and another transfer can be performed.

  A special function (SREP, DREP, or SG) and all chain modes can be selected.

  With comp[!=], data read from address SAR is compared with data read from address DAR, and if these data do not match, the compare result is “true”.

  The check result can be used as a chain condition or as interrupt output.

Operation First, data is read from address SAR, then data is read from address DAR. Next, whether [SAR] != [DAR] is checked.

The transfer cycle is the same as *Figure 10-24 “When DTS0TTYP5 to DTS0TTYP2 are 0100 and DTS0CIR.DTS0IMM is “0””*.

- When DTS0TTYP5 to DTS0TYYP2 are 0101 and immediate function is enabled (DTS0CIR.DTS0IMM is “1”)

  When DTS0IMM is “1”, DTS0SAR becomes IMM, and DTS0SAR becomes the data to be compared.

  DTS0TCEA is split into DTS0STR and DTS0STC, and DTS0STC can be used to specify a transfer count value up to 255 times.

  When the value of DTS0STC reaches “0”, the value of DTS0STR is copied to DTS0STC and another transfer can be performed.

  A special function (DREP or SG) and all chain modes can be selected.

  The check result can be used as a chain condition or as interrupt output.

Operation Data is read from address DAR, and whether SAR != [DAR] is checked.

The transfer cycle is the same as *Figure 10-25 “When DTS0TTYP5 to DTS0TYYP2 are 0100 and immediate function is enabled (DTS0CIR.DTS0IMM is “1”)”*.

- When DTS0TTYP5 to DTS0TTYP2 are 1101 and DTS0CIR.DTS0IMM is “0”

  When the transfer type is set to “extended comp[!=]”, the DTS cycle becomes a comparison [!=] cycle.

  DTS0TCEA becomes DTS0EA, functioning as a register that specifies an offset address used when the check result is “false”.

  Accordingly, the transfer count cannot be specified. When the check result is “true”, TI is fetched from the current TI fetch address added to 10H to perform a chain of transfers. When the check result is “false”, TI is fetched from the current TI fetch address added to the value of DTS0EA to perform a chain of transfers.

  Neither the chain mode nor a special function can be selected.

Operation First, data is read from address SAR, then data is read from address DAR. Next, whether [SAR] != [DAR] is checked.
A chain of transfers is always performed.

- When DTS0TTYP5 to DTS0TYYP2 are 1101 and immediate function is enabled (DTS0CIR.DTS0IMM is “1”)

  When DTS0IMM is “1”, DTS0SAR becomes IMM, and DTS0SAR becomes the data to be compared.

  DTS0TCEA becomes DTS0EA, functioning as a register that specifies an offset address used when the check result is “false”.

  Accordingly, the transfer count cannot be specified. When the check result is “true”, TI is fetched from the current TI fetch address added to 10H to perform a chain of transfers. When the check result is “false”, TI is fetched from the current TI fetch address added to the value of DTS0EA to perform a chain of transfers.

  Neither the chain mode nor a special function can be selected.

Operation The data to be compared is read from address DAR, and then compared with the SAR value.
A chain of transfers is always performed.

(7) comp[<]: When DTS0TTYP5 to DTS0TTYP2 are 0110 or 1110

- When DTS0TTYP5 to DTS0TTYP2 are 0110 and DTS0CIR.DTS0IMM is “0”

  When the transfer type is set to “comp[<]”, the DTS cycle becomes a comparison [<] cycle.

  DTS0TCEA is split into DTS0STR and DTS0STC, and DTS0STC can be used to specify a transfer count value up to 255 times.

  When the value of DTS0STC reaches “0”, the value of DTS0STR is copied to DTS0STC and another transfer can be performed.

  A special function (SREP, DREP, or SG) and all chain modes can be selected.

  With comp[<], data read from address SAR is compared with data read from address DAR, and if the data read from address SAR is smaller, the compare result is “true”.

  The check result can be used as a chain condition or as interrupt output.

Operation First, data is read from address SAR, then data is read from address DAR. Next, whether [(unsigned) SAR] < [(unsigned) DAR] is checked.

The transfer cycle is the same as *Figure 10-24 “When DTS0TTYP5 to DTS0TTYP2 are 0100 and DTS0CIR.DTS0IMM is “0””*.

- When DTS0TTYP5 to DTS0TTYP2 are 0110 and immediate function is enabled (DTS0CIR.DTS0IMM is “1”)

  When DTS0IMM is “1”, DTS0SAR becomes IMM, and DTS0SAR becomes the data to be compared.
  DTS0TCEA is split into DTS0STR and DTS0STC, and DTS0STC can be used to specify a transfer count value up to 255 times.
  When the value of DTS0STC reaches “0”, the value of DTS0STR is copied to DTS0STC and another transfer can be performed.
  A special function (DREP or SG) and all chain modes can be selected.
  The check result can be used as a chain condition or as interrupt output.

Operation Data is read from address DAR, and whether (unsigned) SAR < [(unsigned) DAR] is checked.

The transfer cycle is the same as *Figure 10-25 “When DTS0TTYP5 to DTS0TYYP2 are 0100 and immediate function is enabled (DTS0CIR.DTS0IMM is “1”)”*.

- When DTS0TTYP5 to DTS0TTYP2 are 1110 and DTS0CIR.DTS0IMM is “0”

  When the transfer type is set to “extended comp[<]”, the DTS cycle becomes a comparison [<] cycle.
  DTS0TCEA becomes DTS0EA, functioning as a register that specifies an offset address used when the check result is “false”.
  Accordingly, the transfer count cannot be specified. When the check result is “true”, TI is fetched from the current TI fetch address added to 10H to perform a chain of transfers. When the check result is “false”, TI is fetched from the current TI fetch address added to the value of DTS0EA to perform a chain of transfers.
  Neither the chain mode nor a special function can be selected.

Operation First, data is read from address SAR, then data is read from address DAR. Next, whether [(unsigned) SAR] < [(unsigned) DAR] is checked.

A chain of transfers is always performed.

- When DTS0TTYP5 to DTS0TTYP2 are 1110 and immediate function is enabled (DTS0CIR.DTS0IMM is “1”)

  When DTS0IMM is “1”, DTS0SAR becomes IMM, and DTS0SAR becomes the data to be compared.
  DTS0TCEA becomes DTS0EA, functioning as a register that specifies an offset address used when the check result is “false”.
  Accordingly, the transfer count cannot be specified. When the check result is “true”, TI is fetched from the current TI fetch address added to 10H to perform a chain of transfers. When the check result is “false”, TI is fetched from the current TI fetch address added to the value of DTS0EA to perform a chain of transfers.
  Neither the chain mode nor a special function can be selected.

Operation The data to be compared is read from address DAR, and then compared with the SAR value.
A chain of transfers is always performed.

(8) comp[>]: When DTS0TTYP5 to DTS0TTYP2 are 0111 or 1111

- When DTS0TTYP5 to DTS0TTYP2 are 0111 and DTS0CIR.DTS0IMM is “0”

  When the transfer type is set to “comp[>]”, the DTS cycle becomes a comparison [>] cycle.

  DTS0TCEA is split into DTS0STR and DTS0STC, and DTS0STC can be used to specify a transfer count value up to 255 times.

  When the value of DTS0STC reaches “0”, the value of DTS0STR is copied to DTS0STC and another transfer can be performed.

  A special function (SREP, DREP, or SG) and all chain modes can be selected.

  With comp[>], data read from address SAR is compared with data read from address DAR, and if the data read from address SAR is greater, the compare result is “true”.

  The check result can be used as a chain condition or as interrupt output.

Operation First, data is read from address SAR, then data is read from address DAR. Next, whether [(unsigned) SAR] > [(unsigned) DAR] is checked.

The transfer cycle is the same as *Figure 10-24 “When DTS0TTYP5 to DTS0TTYP2 are 0100 and DTS0CIR.DTS0IMM is “0””*.

- When DTS0TTYP5 to DTS0TTYP2 are 0111 and immediate function is enabled (DTS0CIR.DTS0IMM is “1”)

  When DTS0IMM is “1”, DTS0SAR becomes IMM, and DTS0SAR becomes the data to be compared.

  DTS0TCEA is split into DTS0STR and DTS0STC, and DTS0STC can be used to specify a transfer count value up to 255 times.

  When the value of DTS0STC reaches “0”, the value of DTS0STR is copied to DTS0STC and another transfer can be performed.

  A special function (DREP or SG) and all chain modes can be selected.

  The check result can be used as a chain condition or as interrupt output.

Operation Data is read from address DAR, and whether (unsigned) SAR > [(unsigned) DAR] is checked.

The transfer cycle is the same as *Figure 10-25 “When DTS0TTYP5 to DTS0TYYP2 are 0100 and immediate function is enabled (DTS0CIR.DTS0IMM is “1”)”*.

- When DTS0TTYP5 to DTS0TTYP2 are 1111 and DTS0CIR.DTS0IMM is “0”

  When the transfer type is set to “extended comp[>]”, the DTS cycle becomes a comparison [>] cycle.

  DTS0TCEA becomes DTS0EA, functioning as a register that specifies an offset address used when the check result is “false”.

  Accordingly, the transfer count cannot be specified. When the check result is “true”, TI is fetched from the current TI fetch address added to 10H to perform a chain of transfers. When the check result is “false”, TI is fetched from the current TI fetch address added to the value of DTS0EA to perform a chain of transfers.

  Neither the chain mode nor a special function can be selected.

Operation First, data is read from address SAR, then data is read from address DAR. Next, whether [(unsigned) SAR] > [(unsigned) DAR] is checked.
A chain of transfers is always performed.

- When DTS0TTYP5 to DTS0TTYP2 are 1111 and immediate function is enabled (DTS0CIR.DTS0IMM is “1”)

  When DTS0IMM is “1”, DTS0SAR becomes IMM, and DTS0SAR becomes the data to be compared.

  DTS0TCEA becomes DTS0EA, functioning as a register that specifies an offset address used when the check result is “false”.

  Accordingly, the transfer count cannot be specified. When the check result is “true”, TI is fetched from the current TI fetch address added to 10H to perform a chain of transfers. When the check result is “false”, TI is fetched from the current TI fetch address added to the value of DTS0EA to perform a chain of transfers.

  Neither the chain mode nor a special function can be selected.

Operation The data to be compared is read from address DAR, and then compared with the SAR value.
A chain of transfers is always performed.

10.11.6 Special functions

SREP, DREP, and SG are supported as special functions.

The special function is selected using DTS0TTYP1 and DTS0TTYP0 in DTS0CIR.

(1) SREP (Source address REPeat)

When the value of DTS0STC reaches “0”, this function returns DTS0SAR to its initial value. The TI-D area is used.

Set a value equal to the DTS0SAR value in advance to the TI-D area.

DTS cycles are executed until DTS0STC reaches “0”, then when the last transfer occurs the TI-D area's value is read to DTS0SAR, and a write back is performed to achieve SREP. With SREP, when DTS0STC reaches “0”, a write back of TIB is always performed (it is never skipped).

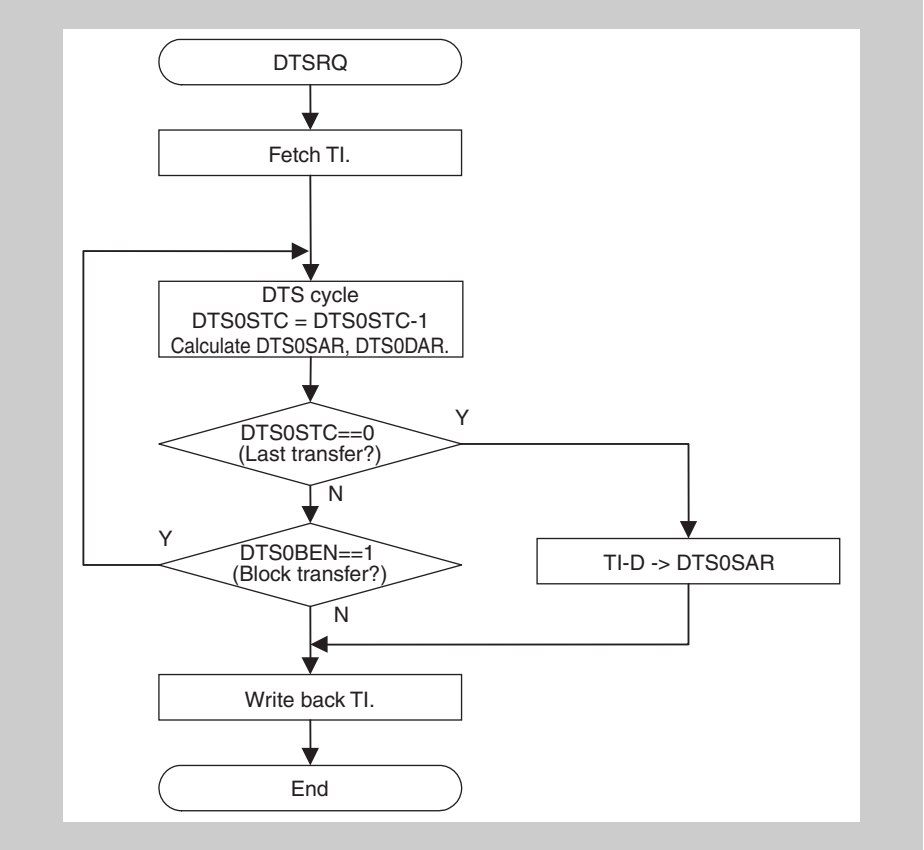


Figure 10-26 SREP processing flow

(2) DREP (Destination address REPeat)

When the value of DTS0STC reaches “0”, this function returns DTS0DAR to its initial value. The TI-D area is used.

The value is returned using the same method as for SREP, and the TI-D value is written back to DTS0DAR.

With DREP, when DTS0STC reaches “0”, a write back of TIC is always performed (it is never skipped).

Note SREP and DREP cannot be set at the same time (since the TI-D area is used).

(3) SG (Scatter & Gather)

In this mode, the count sizes of the source address and destination address are specified as signed 16-bit values. The TI-D area is used. When in modes other than SG, the DTS's address count size can only be specified using DTS0SCS and DTS0DCS, as a value in the range from –4 to +8, but in SG mode DTS0ESCS and DTS0EDCS are used, which enables the count size to become much larger (data in the TI-D area is read to DTS internal registers DTS0ESCS and DTS0EDCS).

10.11.7 Transfer count

The transfer count is MAX255 times or MAX65535 times. In either case, when the specified number of transfers is completed, an interrupt request is output to DTSFSL.

(1) MAX255 times

Although the maximum transfer count is 255 times, the transfer count can be recovered when the transfer count (DTS0STC) reaches “0”, so in fact unlimited transfers are supported.

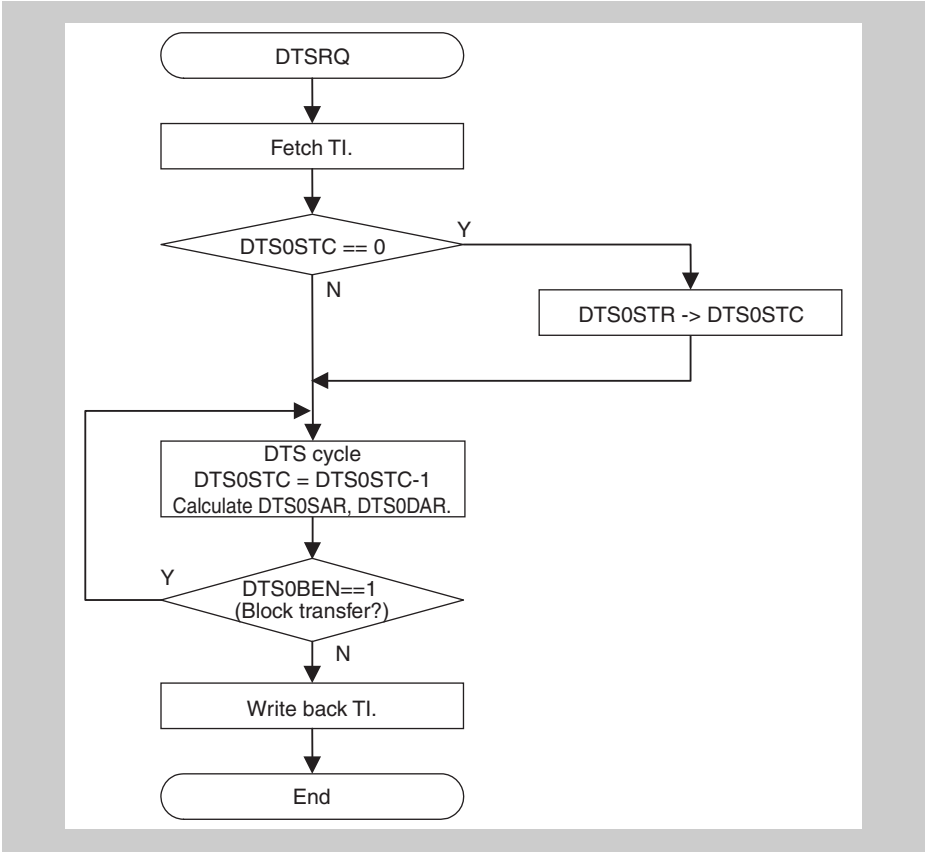


Figure 10-27 Transfer count processing flow

(2) MAX65535 times

When DTS0TCEA is used as a 16-bit counter, a transfer count of up to 65,535 times can be specified.

10.11.8 Chain function

DTS includes a chain function that performs multiple data transfers in succession.

The condition for the chain function can be set by using DTS0CM1 and DTS0CM0 in DTS0CIR. Also, DTS0TTYP4 to DTS0TTYP0 in DTS0CIR can be used to select an extended chain function.

When special flag[0], special flag[1], or special comp[=, !=, <, >] has been selected, the settings in DTS0CM1 and DTS0CM0 are ignored. After TI fetch → DTS cycle → TI write back is completed, the chain function performs a TI fetch, achieving a chain of transfers. The fetch address for this TI fetch operation is the address TI-C + 4 when only TI-A to TI-C are used, and is the address TI-D + 4 when TI-A to TI-D are used (consecutive addresses in internal RAM).

Table 10-52 List of chain functions (1/2)

Transfer mode	Special mode	Settings of DTS0CM1 and DTS0CM0 in DTS0CIR	Operation
transfer	—	00 (No chain)	Does not perform chain of transfers.
		01 (Last transfer)	Performs chain of transfers when DTS0STC (or DTS0LTC when 65535) is 0.
		10 (Chain if true)	Does not perform chain of transfers.
		11 (Always chain)	Performs chain of transfers following DTS cycle. Same as “01” during block transfer.
add	—	00 (No chain)	Does not perform chain of transfers.
		01 (Last transfer)	Performs chain of transfers when DTS0STC (or DTS0LTC when 65535) is 0.
		10 (Chain if true)	Does not perform chain of transfers.
		11 (Always chain)	Performs chain of transfers following DTS cycle. Same as “01” during block transfer.
flag	None SREP DREP	00 (No chain)	Does not perform chain of transfers.
		01 (Last transfer)	Performs chain of transfers when DTS0STC is 0.
		10 (Chain if true)	Performs chain of transfers only when condition is true. An interrupt is generated when it is false.
		11 (Always chain)	Performs chain of transfers following DTS cycle. Same as “01” during block transfer.
	Extended chain	Ignored	Always performs chain of transfers. When condition is true, TI is fetched from current TI fetch address added to 10H to perform chain of transfers (same manner as normal chain transfer). When condition is false, TI is fetched from current TI fetch address added to ±DTS0TCEA to perform chain of transfers.

Table 10-52 List of chain functions (2/2)

Transfer mode	Special mode	Settings of DTS0CM1 and DTS0CM0 in DTS0CIR	Operation
comp	None SREP DREP	00 (No chain)	Does not perform chain of transfers.
		01 (Last transfer)	Performs chain of transfers when DTS0STC is 0.
		10 (Chain if true)	Performs chain of transfers only when condition is true. An interrupt is generated when it is false.
		11 (Always chain)	Performs chain of transfers following DTS cycle. Same as “01” during block transfer.
	Extended chain	Ignored	Always performs chain of transfers. When condition is true, TI is fetched from current TI fetch address added to 10H to perform chain of transfers (same manner as normal chain transfer). When condition is false, TI is fetched from current TI fetch address added to ±DTS0TCEA to perform chain of transfers.

10.11.9 TI hold function

The DTSHENm (m = 0 to 3) register of DTSFSL can be used to enable the TI hold function (ON).

(1) Single transfer when TI hold is ON

When the TI hold function is set to ON in single transfer mode, TI fetch → DTS cycle is executed for one DTS request, then transfer request wait status is set (TI is set to TI hold status). After this, if a DTS request occurs for another channel, TI fetch → DTS cycle → TI write back is executed for that channel. If a DTS request occurs for the same channel as the TI hold, processing goes as far as reading the BT (base table), but since the TI has already been set to TI hold status, a TI fetch is not executed when the DTS cycle occurs.

This operation is repeated until the value of DTS0STC (or DTS0LTC when in 65535 mode) reaches “0”.

When DTS0STC (or DTS0LTC) reaches “0”, TI hold is set to OFF and the TI is written back.

Since the bus is not locked, a CPU cycle may interrupt.

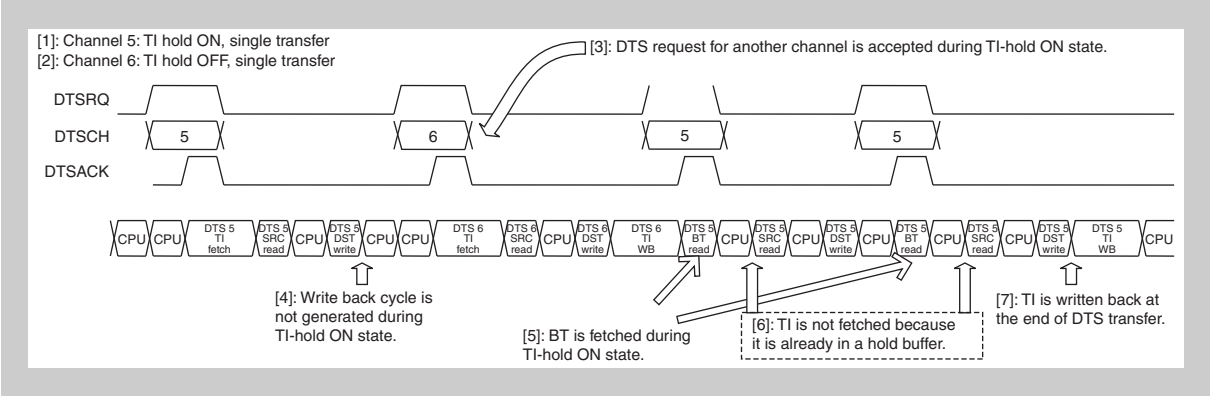


Figure 10-28 TI hold function

(2) Number of TI hold buffers

Four sets of TI hold buffers are provided. The hold instruction for these hold buffers is specified using the DTSHENm (m = 3 to 0) register of DTSFSL.

10.11.10 Interrupt output function

DTS is able to output interrupts to the CPU via DTSFSL. DTS outputs an interrupt source signal to DTSFSL, and DTSFSL then decodes this signal and transmits it to the CPU as an interrupt signal.

The conditions for enabling the interrupt source signal (DTSINTRQ) involves the transfer count, flag check result, and compare result.

With regard to the transfer count, when only one transfer remains (the last transfer), DTSINTRQ activates upon completion of this transfer.

With regard to the flag check and compare results , if either of their judgment results is false, DTSINTRQ activates upon completion of the transfer.

The interrupt source signal (DTSINTRQ) can be masked using the DTS0DISI bit in DTS0CIR.

DTSINTRQ does not become active when the DTS0DISI bit in DTS0CIR is “1”.

10.11.11 Interrupt and chain of transfers during block transfer

In the case of a block transfer, judgments concerning interrupts and chain of transfers are performed only once when the number of transfers specified by DTS0TCEA has been completed.

With regard to interrupts, an interrupt is always output when the specified number of transfers has been completed (the counter has reached “0”). (Note that interrupt masking is possible using DTS0DISI in DTS0CIR.)

With regard to a chain of transfers, when DTS0CM1 and DTS0CM0 in DTS0CIR are “01” (chain when counter is 0) or “11” (always chain), a chain of transfers is always performed. When these bits are “10” (chain if true), a chain of transfers is performed only when all the flag check and compare results are true (and a chain of transfers is not performed if any result is false).

DTS0TCF in DTS0CIR becomes “1” if any of the flag check and compare results is false.

10.11.12 TI write back skip function

DTS skips any TI write back that is not necessary. The conditions for skipping write back are as follows.

Table 10-53 Write back skip conditions

	DTS0CIR				Transfer count	SCS	SAM
	DS	IMM	Type	Special function			
TIB write back skip conditions	—	—	—	SG	—	—	0
	—	0	Not add	SREP	Some remain	0	—
	—	0	—	DREP/ None	—	0	—
	32 bits	1	Not add	SREP	Some remain	—	—
	8/16 bits	1	Not add	SREP	Some remain	0	—
	32 bits	1	—	DREP/ None	—	—	—
	8/16 bits	1	—	DREP/ None	—	0	—

	DTS0CIR		Transfer count	SCS	SAM
	Type	Special function			
TIB write back skip conditions	—	SG	—	—	0
	—	SREP	—	0	—
	Not add	DREP	Some remain	0	—
	—	None	—	0	—

10.11.13 DTS channel priority control

The priority function can set up to four priority levels using DTSFSL.

10.11.14 Valid DTS transfer request conditions

When the DTSEN bit in DTSEN000 to DTSEN127 of DTSFSL is set to “0”, a transfer request to the DTS is not generated for the corresponding channel, but sampling of DTS transfer factors [127:0] is performed. When sampled, if the DTSEN bit is set to “1”, a DTS transfer request is generated.

When the DTSEN bit is “1”, if INTIN occurs when the DTS is transferring data or is stopped either by a transfer error response or by a user, the INITIN is held pending in DTSFSL until the DTS is able to accept it.

Table 10-54 Valid DTS transfer request conditions

DTS	DTS	DTSFSL	Status of DTS	Transfer request by DTS transfer factors [127:0]
DTS0TSR.bit1	DTS0TSR.bit0	DTSEN		
×	×	0	—	Invalid (held)
0	0	1	Acceptable	Valid (if next request occurs during a transfer, it is held)
×	1	1	Not acceptable	Held in DTSFSL
1	×	1	Not acceptable	Held in DTSFSL

10.11.15 Aborting/resuming DTS transfer

Table 10-55 DTS transfer abort timing

Transfer mode	Transfer count	Aborted after:
Single transfer with TI hold OFF	Last transfer	TI write back
	Some remain	
Block transfer with TI hold OFF	Last transfer	TI write back
	Some remain	DTS cycle
Single transfer with TI hold ON	Last transfer	TI write back
	Some remain	DTS cycle
Block transfer with TI hold ON	Last transfer	TI write back
	Some remain	DTS cycle

(1) Aborting or resuming DTS transfer for all channels through hardware (NMI)

Not supported

(2) Aborting or resuming DTS transfer for all channels through software

By setting the DTS transfer abort request trigger bit (DTS0UST) in the DTS transfer request control register (DTS0TRC), the next DTS transfer and those that follow can be aborted. If this occurs during a DTS cycle, the next DTS transfer is aborted according to the condition shown in *Table 10-55 “DTS transfer abort timing”*.

To resume the aborted DTS transfer, set the DTS0UCL bit in the DTS transfer request control register (DTS0TRC), and then clear the DTS0STPU bit. If a DTS transfer is requested at that point, the DTS transfer is resumed.

(3) Aborting or resuming DTS transfer by using DTS transfer enable bit (DTSEN)

This function is performed by the DTSFSL side.

10.11.16 Error response support

(1) Aborting DTS transfer by error response

When DMAT receives an error response from the internal system bus, the DTS cycle end signal and the the internal system bus transfer error detection signal are asserted at the same time. When the transfer error detection signal is detected at high level, DTS sets the DTS0STPE bit in the DTS transfer status register (DTS0TSR) to “1” to abort the DTS transfer immediately. Subsequent DTS transfer requests are not accepted. Meanwhile, the DTS transfer error interrupt signal (DTSERR) is asserted for one clock cycle. This abort occurs as soon as an error response is received, regardless of the transfer mode and TI hold ON/OFF status.

The user can learn in which channel the error has occurred, by reading the DTS active channel register, after confirming that DTS0STPE has been set.

Notes
1. If an error response is received on a channel with TI hold ON, TI write back is not performed.
2. It is not possible to determine the cycle (TI fetch cycle, etc.) where the error response occurred.

(2) Canceling transfer abort caused by error response

To cancel a DTS transfer abort caused by an error response, check the DTS active channel register (DTS0ACR), use the DTS0TIT and DTS0ICH bits in the DTS initialization control register (DTS0ICR) to clear the TI for the channel in which the error has occurred, and then set the DTS0ECL bit in the DTS transfer request control register (DTS0TRC) to “1” to clear DTS0STPE in the DTS transfer status register (DTS0TSR) to “0”.

10.12 DTSFSL Function

The DTS factor selector (DTSFSL) function selects DTS trigger factors from among interrupt signals. Four priority levels can be set for DTS trigger factors. Factors that are not used with DTS can be input to the interrupt controller via DTS transfer-factor-through outputs (IRQ[127:0]).

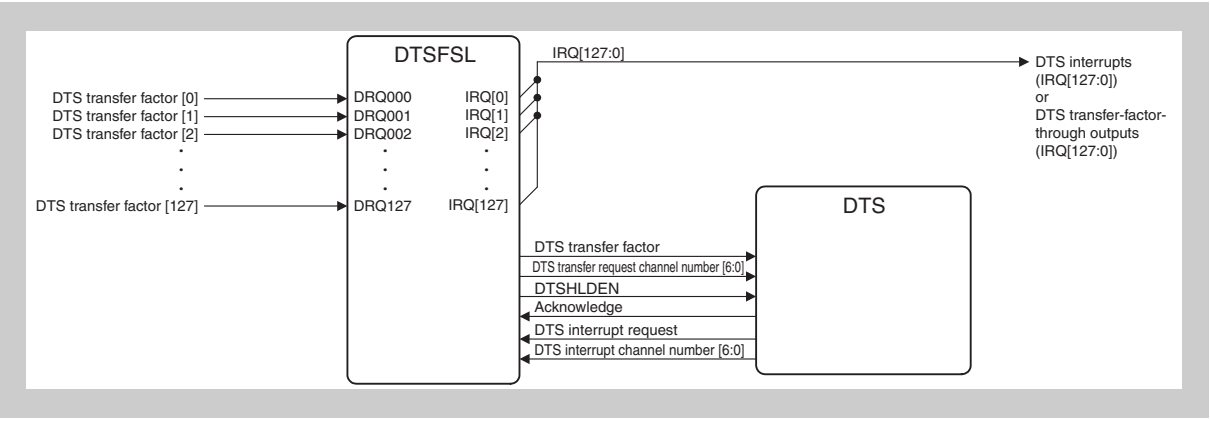


Figure 10-29 Connections between DTSFSL and DTS

10.12.1 Features

A maximum of up to 128 factors can be assigned as DTS transfer factors.

Any of four priority levels can be set for each of these factors.

Based on an interrupt request source from the DTS, an interrupt is output (shared output) to the CPU via a DTS interrupt (IRQ[127:0]) corresponding to the particular channel.

Note The types of interrupt request sources from the DTS are “all transfer counts ended”, “flag check result is false”, and “compare result is false”.

Factors that are not used with DTS are input to the interrupt controller via DTS transfer-factor-through outputs (IRQ[127:0]) (shared output).

Table 10-56 DTS transfer sources (1/4)

Channel	Interrupt to start DTS	Corresponding DTSENx register	
		Address	Symbol
0	INTP0	$FFFF7D00_H$	DTSEN000
1	INTP1	$FFFF7D02_H$	DTSEN001
2	INTP2	$FFFF7D04_H$	DTSEN002
3	INTP3	$FFFF7D06_H$	DTSEN003
4	INTP4	$FFFF7D08_H$	DTSEN004
5	INTP5	$FFFF7D0A_H$	DTSEN005
6	INTP6	$FFFF7D0C_H$	DTSEN006
7	INTP7	$FFFF7D0E_H$	DTSEN007
8	INTP8	$FFFF7D10_H$	DTSEN008

Table 10-56 DTS transfer sources (2/4)

Channel	Interrupt to start DTS	Corresponding DTSENx register	
		Address	Symbol
9	INTP9	$FFFF7D12_H$	DTSEN009
10	INTP10	$FFFF7D14_H$	DTSEN010
11	INTP11	$FFFF7D16_H$	DTSEN011
12	INTP12	$FFFF7D18_H$	DTSEN012
13	INTP13	$FFFF7D1A_H$	DTSEN013
14	INTP14	$FFFF7D1C_H$	DTSEN014
15	INTP15	$FFFF7D1E_H$	DTSEN015
16	INTP16	$FFFF7D20_H$	DTSEN016
17	INTP17	$FFFF7D22_H$	DTSEN017
18	INTP18	$FFFF7D24_H$	DTSEN018
19	INTP19	$FFFF7D26_H$	DTSEN019
20	INTP20	$FFFF7D28_H$	DTSEN020
21	INTP21	$FFFF7D2A_H$	DTSEN021
22	INTP22	$FFFF7D2C_H$	DTSEN022
23	INTP23	$FFFF7D2E_H$	DTSEN023
24	INTP24	$FFFF7D30_H$	DTSEN024
25	INTP25	$FFFF7D32_H$	DTSEN025
26	INTP26	$FFFF7D34_H$	DTSEN026
27	INTP27	$FFFF7D36_H$	DTSEN027
28	INTADCA0I0	$FFFF7D38_H$	DTSEN028
29	INTADCA0I1	$FFFF7D3A_H$	DTSEN029
30	INTADCA0I2	$FFFF7D3C_H$	DTSEN030
31	Reserved	$FFFF7D3E_H$	DTSEN031
32	INTTAUA0I0	$FFFF7D40_H$	DTSEN032
33	INTTAUA0I1	$FFFF7D42_H$	DTSEN033
34	INTTAUA0I2	$FFFF7D44_H$	DTSEN034
35	INTTAUA0I3	$FFFF7D46_H$	DTSEN035
36	INTTAUA0I4	$FFFF7D48_H$	DTSEN036
37	INTTAUA0I5	$FFFF7D4A_H$	DTSEN037
38	INTTAUA0I6	$FFFF7D4C_H$	DTSEN038
39	INTTAUA0I7	$FFFF7D4E_H$	DTSEN039
40	INTTAUA0I8	$FFFF7D50_H$	DTSEN040
41	INTTAUA0I9	$FFFF7D52_H$	DTSEN041
42	INTTAUA0I10	$FFFF7D54_H$	DTSEN042
43	INTTAUA0I11	$FFFF7D56_H$	DTSEN043
44	INTTAUA0I12	$FFFF7D58_H$	DTSEN044
45	INTTAUA0I13	$FFFF7D5A_H$	DTSEN045
46	INTTAUA0I14	$FFFF7D5C_H$	DTSEN046
47	INTTAUA0I15	$FFFF7D5E_H$	DTSEN047
48	INTTAUA1I0	$FFFF7D60_H$	DTSEN048

Table 10-56 DTS transfer sources (3/4)

Channel	Interrupt to start DTS	Corresponding DTSENx register	
		Address	Symbol
49	INTTAUA1I1	$FFFF7D62_H$	DTSEN049
50	INTTAUA1I2	$FFFF7D64_H$	DTSEN050
51	INTTAUA1I3	$FFFF7D66_H$	DTSEN051
52	INTTAUA1I4	$FFFF7D68_H$	DTSEN052
53	INTTAUA1I5	$FFFF7D6A_H$	DTSEN053
54	INTTAUA1I6	$FFFF7D6C_H$	DTSEN054
55	INTTAUA1I7	$FFFF7D6E_H$	DTSEN055
56	INTTAUA1I8	$FFFF7D70_H$	DTSEN056
57	INTTAUA1I9	$FFFF7D72_H$	DTSEN057
58	INTTAUA1I10	$FFFF7D74_H$	DTSEN058
59	INTTAUA1I11	$FFFF7D76_H$	DTSEN059
60	INTTAUA1I12	$FFFF7D78_H$	DTSEN060
61	INTTAUA1I13	$FFFF7D7A_H$	DTSEN061
62	INTTAUA1I14	$FFFF7D7C_H$	DTSEN062
63	INTTAUA1I15	$FFFF7D7E_H$	DTSEN063
64	INTTAUA2I12	$FFFF7D80_H$	DTSEN064
65	INTTAUA2I13	$FFFF7D82_H$	DTSEN065
66	INTTAUA2I14	$FFFF7D84_H$	DTSEN066
67	INTTAUA2I15	$FFFF7D86_H$	DTSEN067
68	INTTAUA3I0	$FFFF7D88_H$	DTSEN068
69	INTTAUA3I1	$FFFF7D8A_H$	DTSEN069
70	INTTAUA3I2	$FFFF7D8C_H$	DTSEN070
71	INTTAUA3I3	$FFFF7D8E_H$	DTSEN071
72	INTTAUA3I4	$FFFF7D90_H$	DTSEN072
73	INTTAUA3I5	$FFFF7D92_H$	DTSEN073
74	INTTAUA3I6	$FFFF7D94_H$	DTSEN074
75	INTTAUA3I7	$FFFF7D96_H$	DTSEN075
76	INTTAUA3I8	$FFFF7D98_H$	DTSEN076
77	INTTAUA3I9	$FFFF7D9A_H$	DTSEN077
78	INTTAUA3I10	$FFFF7D9C_H$	DTSEN078
79	INTTAUA3I11	$FFFF7D9E_H$	DTSEN079
80	INTTAUA3I12	$FFFF7DA0_H$	DTSEN080
81	INTTAUA3I13	$FFFF7DA2_H$	DTSEN081
82	INTTAUA3I14	$FFFF7DA4_H$	DTSEN082
83	INTTAUA3I15	$FFFF7DA6_H$	DTSEN083
84	INTTAUJ0I0	$FFFF7DA8_H$	DTSEN084
85	INTTAUJ0I1	$FFFF7DAA_H$	DTSEN085
86	INTTAUJ0I2	$FFFF7DAC_H$	DTSEN086
87	INTTAUJ0I3	$FFFF7DBE_H$	DTSEN087
88	INTENCA0IOV	$FFFF7DB0_H$	DTSEN088

Table 10-56 DTS transfer sources (4/4)

Channel	Interrupt to start DTS	Corresponding DTSENx register	
		Address	Symbol
89	INTENCA0IUD	FFFF7DB2$_H$	DTSEN089
90	INTENCA0I0	FFFF7DB4$_H$	DTSEN090
91	INTENCA0I1	FFFF7DB6$_H$	DTSEN091
92	INTENCA0IEC	FFFF7DB8$_H$	DTSEN092
93	INTENCA1IOV	FFFF7DBA$_H$	DTSEN093
94	INTENCA1IUD	FFFF7DBC$_H$	DTSEN094
95	INTENCA1I0	FFFF7DBE$_H$	DTSEN095
96	INTENCA1I1	FFFF7DC0$_H$	DTSEN096
97	INTENCA1IEC	FFFF7DC2$_H$	DTSEN097
98	INTTAPA0IPEK0	FFFF7DC4$_H$	DTSEN098
99	INTTAPA0IVLY0	FFFF7DC6$_H$	DTSEN099
100	INTTAPA2ADOUT0	FFFF7DC8$_H$	DTSEN100
101	INTTAPA0ADOUT0	FFFF7DCA$_H$	DTSEN101
102	INTTAPA0ADOUT1	FFFF7DCC$_H$	DTSEN102
103	INTTAPA1IPEK0	FFFF7DCE$_H$	DTSEN103
104	INTTAPA1IVLY0	FFFF7DD0$_H$	DTSEN104
105	INTTAPA3ADOUT0	FFFF7DD2$_H$	DTSEN105
106	INTTAPA1ADOUT0	FFFF7DD4$_H$	DTSEN106
107	INTTAPA1ADOUT1	FFFF7DD6$_H$	DTSEN107
108	INTCSIH0IR	FFFF7DD8$_H$	DTSEN108
109	INTCSIH0IC	FFFF7DDA$_H$	DTSEN109
110	INTCSIH0IJC	FFFF7DDC$_H$	DTSEN110
111	INTCSIH1IR	FFFF7DDE$_H$	DTSEN111
112	INTCSIH1IC	FFFF7DE0$_H$	DTSEN112
113	INTCSIH1IJC	FFFF7DE2$_H$	DTSEN113
114	INTCSIH2IR	FFFF7DE4$_H$	DTSEN114
115	INTCSIH2IC	FFFF7DE6$_H$	DTSEN115
116	INTCSIH2IJC	FFFF7DE8$_H$	DTSEN116
117	INTCSIH3IR	FFFF7DEA$_H$	DTSEN117
118	INTCSIH3IC	FFFF7DEC$_H$	DTSEN118
119	INTCSIH3IJC	FFFF7DEE$_H$	DTSEN119
120	INTETHA0SCRX	FFFF7DF0$_H$	DTSEN120
121	INTETHA0SCTX	FFFF7DF2$_H$	DTSEN121
122	Reserved	FFFF7DF4$_H$	DTSEN122
123	Reserved	FFFF7DF6$_H$	DTSEN123
124	Reserved	FFFF7DF8$_H$	DTSEN124
125	Reserved	FFFF7DFA$_H$	DTSEN125
126	Reserved	FFFF7DFC$_H$	DTSEN126
127	Reserved	FFFF7DFE$_H$	DTSEN127

10.13 DTSFSL Control Registers

10.13.1 DTSENx (x = 000 to 127): DTS transfer enable register

Access This register can be read or written in 8-bit units.

Address See *Table 10-56 "DTS transfer sources"*.

Initial Value 00_H

7	6	5	4	3	2	1	0
DRQ	0	DRQCLR	0	0	DTSEN	DTSPR1	DTSPR0
R	R/W	R/W	R	R	R/W	R/W	R/W

Table 10-57 DTSENx register contents

Bit position	Bit name	Function
7	DRQ	This bit indicates whether a transfer request to the DTS is being held pending. This is a read-only bit, and writing to this bit is ignored. 0: No DTS transfer requests 1: DTS transfer requests exists (wait status) Condition for setting (to “1”): When the rising edge of DTS transfer source x is detected Condition for clearing (to “0”): When “1” is written to the DRQCLR bit, or when the DTS accepts a transfer request
5	DRQCLR	Writing “1” to this bit clears the DRQ bit to “0”. Writing “0” to this bit is ignored.
2	DTSEN	DTS transfer request enable bit for DTS transfer source x 0: Does not use DTS transfer source x as DTS transfer request (DTS request disabled) 1: Uses DTS transfer source x as DTS transfer request (DTS request enabled)
1 0	DTSPR1 DTSPR0	These bits specify the priority order. “00” specifies the highest priority level and “11” specifies the lowest priority level.

10.13.2 DTSHENm (m = 0 to 3): DTS hold enable register

Access This register can be read or written in 16-bit units.

Address DTSHEN0: FFFF 7E00$_H$, DTSHEN1: FFFF 7E02$_H$, DTSHEN2: FFFF 7E04$_H$, DTSHEN1: FFFF 7E06$_H$

Initial Value 0000$_H$

15	14	13	12	11	10	9	8
HEN	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R

7	6	5	4	3	2	1	0
0	dtsch6	dtsch5	dtsch4	dtsch3	dtsch2	dtsch1	dtsch0
R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 10-58 DTSHENm register contents

Bit position	Bit name	Function
15	HEN	This bit enables/disables operation of the DTS hold enable register. 1: Enables operation of hold enable register. 0: Disables operation of hold enable register.
6:0	dtsch6 to dtsch0	These bits select the target channel for hold. Settings to these bits are valid only when HEN is “1”. The set values are listed in the table below.

Table 10-59 List of channels subject to hold selected by setting dtsch6 to dtsch0

dtsch[6:0]	Channel to be held	dtsch[6:0]	Channel to be held	dtsch[6:0]	Channel to be held	dtsch[6:0]	Channel to be held
000_0000	CH000	010_0000	CH032	100_0000	CH064	110_0000	CH096
000_0001	CH001	010_0001	CH033	100_0001	CH065	110_0001	CH097
000_0010	CH002	010_0010	CH034	100_0010	CH066	110_0010	CH098
000_0011	CH003	010_0011	CH035	100_0011	CH067	110_0011	CH099
000_0100	CH004	010_0100	CH036	100_0100	CH068	110_0100	CH100
000_0101	CH005	010_0101	CH037	100_0101	CH069	110_0101	CH101
000_0110	CH006	010_0110	CH038	100_0110	CH070	110_0110	CH102
000_0111	CH007	010_0111	CH039	100_0111	CH071	110_0111	CH103
000_1000	CH008	010_1000	CH040	100_1000	CH072	110_1000	CH104
000_1001	CH009	010_1001	CH041	100_1001	CH073	110_1001	CH105
000_1010	CH010	010_1010	CH042	100_1010	CH074	110_1010	CH106
000_1011	CH011	010_1011	CH043	100_1011	CH075	110_1011	CH107
000_1100	CH012	010_1100	CH044	100_1100	CH076	110_1100	CH108
000_1101	CH013	010_1101	CH045	100_1101	CH077	110_1101	CH109
000_1110	CH014	010_1110	CH046	100_1110	CH078	110_1110	CH110
000_1111	CH015	010_1111	CH047	100_1111	CH079	110_1111	CH111
001_0000	CH016	011_0000	CH048	101_0000	CH080	111_0000	CH112
001_0001	CH017	011_0001	CH049	101_0001	CH081	111_0001	CH113
001_0010	CH018	011_0010	CH050	101_0010	CH082	111_0010	CH114
001_0011	CH019	011_0011	CH051	101_0011	CH083	111_0011	CH115
001_0100	CH020	011_0100	CH052	101_0100	CH084	111_0100	CH116
001_0101	CH021	011_0101	CH053	101_0101	CH085	111_0101	CH117
001_0110	CH022	011_0110	CH054	101_0110	CH086	111_0110	CH118
001_0111	CH023	011_0111	CH055	101_0111	CH087	111_0111	CH119
001_1000	CH024	011_1000	CH056	101_1000	CH088	111_1000	CH120
001_1001	CH025	011_1001	CH057	101_1001	CH089	111_1001	CH121
001_1010	CH026	011_1010	CH058	101_1010	CH090	111_1010	CH122
001_1011	CH027	011_1011	CH059	101_1011	CH091	111_1011	CH123
001_1100	CH028	011_1100	CH060	101_1100	CH092	111_1100	CH124
001_1101	CH029	011_1101	CH061	101_1101	CH093	111_1101	CH125
001_1110	CH030	011_1110	CH062	101_1110	CH094	111_1110	CH126
001_1111	CH031	011_1111	CH063	101_1111	CH095	111_1111	CH127

10.13.3 DTSSELm (m = 0 to 15): DTS transfer end interrupt select register

This register is used to switch between a DTS transfer end interrupt and internal or external interrupt.

Access This register can be read or written in 32-bit units.

Address

DTSSEL0: FFFF FE00$_H$	DTSSEL1: FFFF FE04$_H$
DTSSEL2: FFFF FE08$_H$	DTSSEL3: FFFF FE0C$_H$
DTSSEL4: FFFF FE10$_H$	DTSSEL5: FFFF FE14$_H$
DTSSEL6: FFFF FE18$_H$	DTSSEL7: FFFF FE1C$_H$
DTSSEL8: FFFF FE20$_H$	DTSSEL9: FFFF FE24$_H$
DTSSEL10: FFFF FE28$_H$	DTSSEL11: FFFF FE2C$_H$
DTSSEL12: FFFF FE30$_H$	DTSSEL13: FFFF FE34$_H$
DTSSEL14: FFFF FE38$_H$	DTSSEL15: FFFF FE3C$_H$

Initial Value 0000 0000$_H$. This register is initialized by any reset.

Cautions

1. Be sure to clear the DSEL[127:0] bits corresponding to the channels that are not listed or indicated as *Reserved* in *Table 10-59 "List of channels subject to hold selected by setting dtsch6 to dtsch0" on page 484*.
2. When changing a DTSSELm.DSEL[127:0] bit, make sure that the interrupt request signal corresponding to the DTS transfer source to be changed will not be generated.
3. Set a DTS transfer end interrupt to the same DTS transfer source.

<R>

31	30	29	28	27	26	25	24
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
7	6	5	4	3	2	1	0
DSEL m×8+7	DSEL m×8+6	DSEL m×8+5	DSEL m×8+4	DSEL m×8+3	DSEL m×8+2	DSEL m×8+1	DSEL m×8+0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 10-60 DTSSELm register contents

Bit position	Bit name	Function
7:0	DSELm×8+7 to DSELm×8+0	These bits select an interrupt request signal that triggers DTS transfer. 0: Selects the internal or external interrupt source. 1: Selects the DTS transfer end interrupt. To start a DTS transfer, enable transfer requests by using DTS transfer enable register n (DTSENn) of the DTSFSL, and select the DTS transfer end interrupt by setting the DTSSELm.DSEL[127:0] bit. To cancel the DTS transfer, disable transfer requests by using DTS transfer enable register n (DTSENn) of the DTSFSL, and select the internal/external interrupt source by clearing the DTSSELm.DSEL[127:0] bit. If the DTSSELm.DSEL[127:0] bit is cleared to select the internal or external interrupt source while transfer requests are enabled, or if the DTSSELm.DSEL[127:0] bit is set to select the DTS transfer end interrupt while transfer requests are disabled, the interrupt request signal will not reach the CPU even if an interrupt corresponding to the source assigned to DTS channel n is generated.

Chapter 11 Primary Memory Controller (PMEMC)

11.1 Overview

The external memory access function of this microcontroller provides an interface for connecting memory, ASICs, etc., outside this microcontroller. This microcontroller allows the connection of the following 2 types of memory.

- SRAM
- SDRAM

This microcontroller provides 4 chip select areas (CS1 to CS4), and the bus size and wait time can be set independently for each chip select area.The bus cycle type of the CS4 area is fixed to SDRAM type, and the eighth consecutive read access can be performed by making the burst read available.

In addition to an external wait function, this microcontroller has various programmable wait functions that can be set for each chip select area, allowing the connection of various types of memory.

This microcontroller has also an external bus hold function that allows the creation of a memory system with a multi-master configuration.

Input/output signals

The following table shows the input/output signals of the primary memory controller.

Table 11-1 Input/output signals of primary memory controller (1/2)

P_MEMC signal	Function	Connected to
MDIxx MADxx	Data bus	Port P_D[0:31]
BENZxx	Byte enable signal output (D0 to D7)	Port $\overline{\text{P\_LLBE}}$
	Byte enable signal output (D8 toD15)	Port $\overline{\text{P\_LUBE}}$
	Byte enable signal output (D16 to D23)	Port $\overline{\text{P\_ULBE}}$
	Byte enable signal output (D24 to D31)	Port $\overline{\text{P\_UUBE}}$
RDZ	Read strobe signal output	Port $\overline{\text{P\_RD}}$
DVCLK	Bus clock output	Port P_BUSCLK
WRZ	Write strobe signal output	Port $\overline{\text{P\_WR}}$
RWZ	Read/write status signal output	Port $\overline{\text{P\_RW}}$
xxWRZ	Write strobe signal output (D0 to D7)	Port $\overline{\text{P\_LLWR}}$
	Write strobe signal output (D8 to D15)	Port $\overline{\text{P\_LUWR}}$
	Write strobe signal output (D16 to D23)	Port $\overline{\text{P\_ULWR}}$
	Write strobe signal output (D24 to D31)	Port $\overline{\text{P\_UUWR}}$
DQMxx	I/O mask signal output for SDRAM (D0 to D7)	Port P_LLDQM
	I/O mask signal output for SDRAM (D8 to D15)	Port P_LUDQM
	I/O mask signal output for SDRAM (D16 to D23)	Port P_ULDQM
	I/O mask signal output for SDRAM (D24 to D31)	Port P_UUDQM
REFRQZ	Refresh request	Port $\overline{\text{P\_REFRQ}}$

Table 11-1 Input/output signals of primary memory controller (2/2)

P_MEMC signal	Function	Connected to
SDRASZ	Row address strobe signal output for SDRAM	Port $\overline{\text{P\_SDRAS}}$
SDCASZ	Column address strobe signal output for SDRAM	Port $\overline{\text{P\_SDCAS}}$
CKE	SDRAM clock enable output signal	Port P_SDCKE
CPBUSRQ	CPU bus request report output	Port P_BUSRQ
SDWEZ	Data write enable output for SDRAM	Port $\overline{\text{P\_SDWE}}$
LBAZ	Bus cycle start for SRAM cycle	Port $\overline{\text{P\_BCYST}}$
MAOxx	Address bus for external memory	Port P_A[0:25]
CSZxx	Chip select signal output	Port $\overline{\text{P\_CS}}$[1:4]
WAITZ	External wait request input	Port $\overline{\text{P\_WAIT}}$
HLDAKZ	Bus hold acknowledge output	Port $\overline{\text{P\_HLDAK}}$
HLDRQZ	Bus hold request input	Port $\overline{\text{P\_HLDRQ}}$

11.1.1 Operation mode, connectable memory types

(1) Separate bus mode

This is an operation mode that connects address output and data input/output to external memory using independent signal lines. By using this mode, any of the following three types of memory can be connected to individual chip select areas.

- SRAM
- SDRAM[a]

11.1.2 Chip select output function

The external bus area of the memory space is divided into four chip select areas, and a chip select signal can be output for each chip select area. The allocation of these chip select areas is fixed by the system and cannot be changed through programming.

11.1.3 Bus sizing function

The bus size can be selected from 8 bits, 16 bits, or 32 bits for each chip select area. To execute access when the data size exceeds the selected bus width, divide the data into sizes smaller than the bus width by using the bus sizing function.

11.1.4 Data endian setting function

The data endian (little endian/big endian) can be specified for the chip select areas. However, since the software development tools made by Renesas Electronics (assemblers and debuggers) only support little endian, instruction fetch operations in big endian are not possible.

The initial status of all chip select areas is little endian.

Misaligned access to these areas is prohibited when big endian has been selected.

11.1.5 Programmable wait setting functions

This microcontroller has the following wait functions, which can be set for each chip select area.

- Programmable data wait
- Data hold wait
- Idle cycle function
- RAS latency (in SDRAM)
- CAS latency (in SDRAM)

a) SDRAM can only be selected for a specific chip select area (CS4).

11.1.6 External wait function

When SRAM, or during write access to page-ROM memory or Sync-Flash memory, data waits of any width can be inserted from outside from the WAITZ pin. The WAITZ pin is sampled just before the data output cycle, and the data latch timing can be delayed by any amount.

11.1.7 External wait error detection function

This microcontroller has a function to forcibly cancel a wait state if an external wait is continuously input for 128 clock cycles and generate a SYSERROR interrupt, in order to prevent a system hang-up if an external wait is continuously input due to a defect of the external wait pin.

11.1.8 Bus hold function

A bus hold request from outside can be input through the HLDRQZ pin.

This microcontroller also has the CPBUSRQ pin, which is used to report the fact that there was a transfer request from inside this microcontroller during the bus hold period. By controlling bus hold requests from the outside using this pin function, it is possible to create a multi-master system that minimizes the effect of an external bus hold on the execution of CPU processes.

11.1.9 DMA cycle output function

This pin function indicates that an external bus cycle has been generated by the DMA.

This microcontroller has a DMA equipped with 6 channels.

11.2 Registers

These registers are used to control the primary memory controller.

Note The clock counts shown in this section indicate the clock count when operating on the external bus clock unless otherwise specified.

Table 11-2 External memory access control registers

Address	Register name	Symbol	R/W	Manipulable bit unit			Initial value
				1	8	16	
$FFFF7200_H$	Bus size configuration register	BSC	R/W			√	$AAAA_H$
$FFFF7202_H$	Data endian configuration register	DEC	R/W			√	0000_H
$FFFF7208_H$	Data wait configuration register 0	DWC0	R/W			√	7777_H
$FFFF720C_H$	Data hold wait configuration register	DHC	R/W			√	0000_H
$FFFF7214_H$	Idle cycle configuration register 0	ICC0	R/W			√	0000_H
$FFFF7216_H$	Idle cycle configuration register 1	ICC1	R/W			√	0000_H
$FFFF721A_H$	External wait error configuration register	EWC	R/W			√	0000_H
$FFFF7220_H$	SDRAM enable control register	SEN	R/W		√		00_H
$FFFF7222_H$	SDRAM configuration register	SDCR	R/W			√	$30C0_H$
$FFFF7224_H$	SDRAM status register	STR	R		√		00_H
$FFFF7226_H$	SDRAM refresh control register	RFS	R/W			√	0000_H

11.2.1 Bus size configuration register (BSC)

The BSC register is used to set the bus size of the external bus for each chip select area.

Access This register can be read or written in 16-bit units.

Address $FFFF7200_H$

Initial Value $AAAA_H$.

Caution Only bits 9 to 2 can be set. Be sure to set bits 15, 13, 11, and 1 to “1”, and bits 14, 12, 10, and 0 to “0”.

15	14	13	12	11	10	9	8
BS71	BS70	BS61	BS60	BS51	BS50	BS41	BS40
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
BS31	BS30	BS21	BS20	BS11	BS10	BS01	BS00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 11-3 BSC register contents

Bit position	Bit name	Function
15:14, 13:12, 11:10, 9:8, 7:6, 5:4, 3:2, 1:0	BSn1, BSn0	Bus size setting bits These bits set the bus width of each chip select area.

BSn1	BSn0	Bus size
0	0	8 bits
0	1	16 bits
1	0	32 bits
1	1	Setting prohibited

11.2.2 Data endian configuration register (DEC)

The DEC register is used to set the endianness of the external bus.

Access This register can be read or written in 16-bit units.

Address $FFFF7202_H$

Initial Value 0000_H. This register is initialized by any reset.

Cautions

1. Only bits 8, 6, 4, and 2 can be set. Be sure to set bits 15 to 9, 7, 5, 3, 1, and 0 to “0”.
2. Misaligned access to these areas is prohibited when big endian has been selected.

15	14	13	12	11	10	9	8
0	DE7	0	DE6	0	DE5	0	DE4
R	R/W	R	R/W	R	R/W	R	R/W
7	6	5	4	3	2	1	0
0	DE3	0	DE2	0	DE1	0	DE0
R	R/W	R	R/W	R	R/W	R	R/W

Table 11-4 DEC register contents

Bit position	Bit name	Function
14, 12, 10, 8, 6, 4, 2, 0	DEn	Data endian setting bits These bits set the endian of each chip select area. 0: Little endian 1: Big endian

11.2.3 Data wait configuration registers 0 (DWC0)

The DWC0 register is used to set the number of data wait states of the external bus.

The value set to the DWC0 register becomes valid in the following bus cycles.

- Data transfer cycle of SRAM bus cycle type

Access These registers can be read or written in 16-bit units.

Address FFFF7208$_H$

Initial Value 7777$_H$

Caution Only bits 15 to 4 can be set. Be sure to set bit 3 to “0”, and bits 2 to 0 to “1”.

	15	14	13	12	11	10	9	8
DWC0	DW33	DW32	DW31	DW30	DW23	DW22	DW21	DW20
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	7	6	5	4	3	2	1	0
	DW13	DW12	DW11	DW10	DW03	DW02	DW01	DW00
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 11-5 DWC0 and DWC1 register contents

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>15 to 12,
11 to 8,
7 to 4,
3 to 0</td><td>DWn3,
DWn2,
DWn1,
DWn0</td><td>Data wait setting bits
These bits set the number of data wait states for each chip select area.
<table>
<tr><th>DWn3</th><th>DWn2</th><th>DWn1</th><th>DWn0</th><th>Number of data wait states</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>No data wait</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1 clock cycle</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>2 clock cycles</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>3 clock cycles</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>4 clock cycles</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>5 clock cycles</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td><td>6 clock cycles</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td>7 clock cycles</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>8 clock cycles</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td><td>9 clock cycles</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>0</td><td>10 clock cycles</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td><td>11 clock cycles</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>0</td><td>12 clock cycles</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>1</td><td>13 clock cycles</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>0</td><td>14 clock cycles</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>15 clock cycles</td></tr>
</table></td></tr>
</table>

11.2.4 Data hold wait configuration register (DHC)

The DHC register is used to set the number of extended data hold wait states for each chip select area in the write cycle of the external bus.

The number of data hold wait states determined by the set value of the DHC register + 1 cycle is inserted in a write cycle.

Access This register can be read or written in 16-bit units.

Address $FFFF720C_H$

Initial Value 0000_H. This register is initialized by any reset.

Caution Bits 7 to 2 can be set. Be sure to clear bits 15 to 8, 1, and 0 to “0”.

15	14	13	12	11	10	9	8
DH71	DH70	DH61	DH60	DH51	DH50	DH41	DH40
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
DH31	DH30	DH21	DH20	DH11	DH10	DH01	DH00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 11-6 DHC register contents

Bit position	Bit name	Function
15:14, 13:12, 11:10, 9:8, 7:6, 5:4, 3:2, 1:0	DHn1, DHn0	Data hold wait setting bits These bits set the number of data hold wait states for each chip select area.

DHn1	DHn0	Number of data hold wait states
0	0	No data hold wait
0	1	1 clock cycle
1	0	2 clock cycles
1	1	3 clock cycles

11.2.5 Idle cycle configuration registers 0 and 1 (ICC0, ICC1)

The ICC registers are used to set the number of idle cycles of the external bus. The number of idle cycles can be set for each chip select area and in the read cycle or write cycle.

Access These registers can be read or written in 16-bit units.

Address $FFFF7214_H$: ICC0

$FFFF7216_H$: ICC1

Initial Value 0000_H. These registers are initialized by any reset.

Cautions

1. The number of idle cycles set by the ICCm register (m = 0, 1) is invalid during the burst read cycle and bus sizing cycle.
2. Bits 15 to 4 of the ICC0 register can be set. Be sure to set bits 3 to 0 to “0”.
3. Bits 3 to 0 of the ICC1 register can be set. Be sure to set bits 15 to 4 to “0”.

ICC0

15	14	13	12	11	10	9	8
WIC31	WIC30	RIC31	RIC30	WIC21	WIC20	RIC21	RIC20
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
WIC11	WIC10	RIC11	RIC10	WIC01	WIC00	RIC01	RIC00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ICC1

15	14	13	12	11	10	9	8
WIC71	WIC70	RIC71	RIC70	WIC61	WIC60	RIC61	RIC60
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
WIC51	WIC50	RIC51	RIC50	WIC41	WIC40	RIC41	RIC40
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 11-7 ICC0 and ICC1 register contents

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>13:12,
9:8
5:4
1:0</td><td>RICn1,
RICn0</td><td>Idle cycle setting bits after read cycle
These bits set the number of idle cycles for each chip select area after a read cycle.
<table>
<tr><th>RICn1</th><th>RICn0</th><th>Number of idle cycles</th></tr>
<tr><td>0</td><td>0</td><td>No idle cycle</td></tr>
<tr><td>0</td><td>1</td><td>1 clock cycle</td></tr>
<tr><td>1</td><td>0</td><td>2 clock cycles</td></tr>
<tr><td>1</td><td>1</td><td>3 clock cycles</td></tr>
</table>
Setting the RICn1 and RICn0 bits is enabled for read accesses in all the bus modes and of all the bus cycle types.</td></tr>
<tr><td>15:14
11:10
7:6
3:2</td><td>WICn1
WICn0</td><td>Idle cycle setting bits after write cycle
These bits set the number of idle cycles for each chip select area after a write cycle.
<table>
<tr><th>WICn1</th><th>WICn0</th><th>Number of idle cycles</th></tr>
<tr><td>0</td><td>0</td><td>No idle cycle</td></tr>
<tr><td>0</td><td>1</td><td>1 clock cycle</td></tr>
<tr><td>1</td><td>0</td><td>2 clock cycles</td></tr>
<tr><td>1</td><td>1</td><td>3 clock cycles</td></tr>
</table>
Setting the WICn1 and WICn0 bits is enabled for write accesses in all the bus modes and of all the bus cycle types.</td></tr>
</table>

11.2.6 External wait error configuration register (EWC)

The EWC register is used to enable or disable the external wait error function for each chip select area.

Setting the EWC register is enabled in the following bus cycles.

- SRAM bus cycle type (separate bus mode)

Access This register can be read or written in 16-bit units.

Address FFFF721A$_H$

Initial Value 0000$_H$. This register is initialized by any reset.

Cautions

1. Only bits 6, 4, and 2 can be specified. Be sure to clear bits 14, 12, 10, 8, and 0.
2. For how to enable system error exceptions, see *(a) "System error control register (SEG_CONT)"* in *4.12.3 (2)*.
3. When using the bus hold function of the primary memory controller, set this register to 0000$_H$.

15	14	13	12	11	10	9	8
0	EW7	0	EW6	0	EW5	0	EW4
R	R/W	R	R/W	R	R/W	R	R/W
7	6	5	4	3	2	1	0
0	EW3	0	EW2	0	EW1	0	EW0
R	R/W	R	R/W	R	R/W	R	R/W

Table 11-8 EWC register contents

Bit position	Bit name	Function
14, 12, 10, 8, 6, 4, 2, 0	EWn	External wait error setting bits These bits enable or disable an external wait error in each chip select area. 0: Disables external wait error. 1: Enables external wait error. When this function is enabled, this microcontroller forcibly cancels the wait state when it has detected the external wait signal for the duration of 128 consecutive clock cycles, and the CPU generates the SYSERROR exception.

11.2.7 SDRAM enable control register (SEN)

The SEN register controls whether SDRAM controller operations are enabled or disabled.

Access This register can be read or written in 8-bit units.

Address $FFFF7220_H$

Initial Value 00_H. This register is initialized by any reset.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	SE
R	R	R	R	R	R	R	R/W

Table 11-9 SEN register contents

Bit position	Bit name	Function
0	SE	This bet sets whether SDRAM controller operations are enabled or disabled. 0: Operation stopped (No external bus cycles) 1: Operation enabled

11.2.8 SDRAM configuration register (SDCR)

The SDCR register is used to set the number of waits and address multiplexed width when SDRAM is accessed. When this register is written, a register write operation is started.

Access This register can be read or written in 16-bit units.

Address $FFFF7222_H$

Initial Value $30C0_H$. This register is initialized by any reset.

Cautions

1. Before a register write operation is performed, there are no SDRAM read or write cycles. Before accessing SDRAM, read the STR register values and make sure that the WCF bit is set to 1.
2. Before setting the SDCR register, set the SDRAM refresh control register (RFS).
3. After accessing SDRAM, to again write to the SDCR register, first clear the SE bit of the SEN register to 0.
4. When the SE bit is cleared to 0, be sure to write to the SDCR register before setting the SE bit to 1 again. At this time, if the SDCR register value does not need to be changed, write the same values.
5. Be sure to clear bits 11, 10, and 8 to “0”.
6. Perform a write to SDRAM configuration register only once after reset release. After the write is performed, keep the value set.
   In case the value is changed, the normal access to SDRAM may not be ensured.

15	14	13	12	11	10	9	8
BST	LTM2	LTM1	LTM0	0	0	PDM	0
R/W	R/W	R/W	R/W	R	R	R/W	R
7	6	5	4	3	2	1	0
BCW1	BCW0	SSO1	SSO0	RAW1	RAW0	SAW1	SAW0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 11-10 SDCR register contents (1/2)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>15</td><td>BST</td><td>This bit enables or disables the burst read function.
0: Burst read function disabled (initial value)
1: Burst read function enabled</td></tr>
<tr><td>14 to 12</td><td>LTM2 to LTM0</td><td>These bits set the CAS latency value when reading.
<table>
<tr><th>LTM2</th><th>LTM1</th><th>LTM0</th><th>CAS latency</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>2</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>3</td></tr>
<tr><td colspan="3">Other than above</td><td>Setting prohibited</td></tr>
</table></td></tr>
<tr><td>9</td><td>PDM</td><td>This bit is set when the power down mode for SDRAM is used. When the PDM bit is set to 1, if the SDRAM is not being accessed, the CKE signal level is set to low level and SDRAM enters the power down mode.
0: Do not use power down mode (initial value)
1: Use power down mode</td></tr>
<tr><td>7, 6</td><td>BCW1, BCW0</td><td>These bits set the number of wait states from a bank active command to a read or write command, or from a precharge command to a bank active command.
<table>
<tr><th>BCW1</th><th>BCW0</th><th>Number of wait states</th></tr>
<tr><td>0</td><td>0</td><td>Setting prohibited</td></tr>
<tr><td>0</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>2</td></tr>
<tr><td>1</td><td>1</td><td>3 (initial value)</td></tr>
</table></td></tr>
<tr><td>5, 4</td><td>SSO1, SSO0</td><td>These bits set the address shift width during page judgment.
If the data bus width is set to 16 or 32 bits, the system does not use the lower addresses (A0 or A1, A0).
<table>
<tr><th>SSO1</th><th>SSO0</th><th>Address shift width</th></tr>
<tr><td>0</td><td>0</td><td>0 bits (data bus width: 8 bits) (initial value)</td></tr>
<tr><td>0</td><td>1</td><td>1 bit (data bus width: 16 bits)</td></tr>
<tr><td>1</td><td>0</td><td>2 bits (data bus width: 32 bits)</td></tr>
<tr><td>1</td><td>1</td><td>Setting prohibited</td></tr>
</table></td></tr>
<tr><td>3, 2</td><td>RAW1, RAW0</td><td>These bits set row address width.
<table>
<tr><th>RAW1</th><th>RAW0</th><th>Row address width</th></tr>
<tr><td>0</td><td>0</td><td>11 bits (initial value)</td></tr>
<tr><td>0</td><td>1</td><td>12 bit</td></tr>
<tr><td>1</td><td>0</td><td>13 bits</td></tr>
<tr><td>1</td><td>1</td><td>14 bits</td></tr>
</table></td></tr>
</table>

Table 11-10 SDCR register contents (2/2)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>1, 0</td><td>SAW1,
SAW0</td><td>These bits set the address multiplexed width (column address width) for when SDRAM is accessed.
<table>
<tr><th>SAW1</th><th>SAW0</th><th>Address multiplexed width
(column address width)</th></tr>
<tr><td>0</td><td>0</td><td>8 bits (initial value)</td></tr>
<tr><td>0</td><td>1</td><td>9 bits</td></tr>
<tr><td>1</td><td>0</td><td>10 bits</td></tr>
<tr><td>1</td><td>1</td><td>11 bits</td></tr>
</table>
</td></tr>
</table>

Table 11-11 Row address output

Bit setting		Address pins																		
SAW1	SAW0	MAO 28 to MAO 18	MAO 17	MAO 16	MAO 15	MAO 14	MAO 13	MAO 12	MAO 11	MAO 10	MAO 9	MAO 8	MAO 7	MAO 6	MAO 5	MAO 4	MAO 3	MAO 2	MAO 1	MAO 0
0	0	a28 to a18	a25	a24	a23	a22	a21	a20	a19	a18	a17	a16	a15	a14	a13	a12	a11	a10	a9	a8
0	1	a28 to a18	a26	a25	a24	a23	a22	a21	a20	a19	a18	a17	a16	a15	a14	a13	a12	a11	a10	a9
1	0	a28 to a18	a27	a26	a25	a24	a23	a22	a21	a20	a19	a18	a17	a16	a15	a14	a13	a12	a11	a10
1	1	a28 to a18	a28	a27	a26	a25	a24	a23	a22	a21	a20	a19	a18	a17	a16	a15	a14	a13	a12	a11

Table 11-12 Column address output

(a) When any bank precharge command is issued

Bit setting		Address pins																		
SSO 1	SSO 0	MAO 28 to MAO 18	MAO 17	MAO 16	MAO 15	MAO 14	MAO 13	MAO 12	MAO 11	MAO 10	MAO 9	MAO 8	MAO 7	MAO 6	MAO 5	MAO 4	MAO 3	MAO 2	MAO 1	MAO 0
0	0	a28 to a18	a17	a16	a15	a14	a13	a12	a11	1	a9	a8	a7	a6	a5	a4	a3	a2	a1	a0
0	1	a28 to a18	a17	a16	a15	a14	a13	a12	1	a10	a9	a8	a7	a6	a5	a4	a3	a2	a1	a0
1	0	a28 to a18	a17	a16	a15	a14	a13	1	a11	a10	a9	a8	a7	a6	a5	a4	a3	a2	a1	a0

(b) When register write command is issued

Bit setting		Address pins																		
SSO 1	SSO 0	MAO 28 to MAO 18	MAO 17	MAO 16	MAO 15	MAO 14	MAO 13	MAO 12	MAO 11	MAO 10	MAO 9	MAO 8	MAO 7	MAO 6	MAO 5	MAO 4	MAO 3	MAO 2	MAO 1	MAO 0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	LTM2	LTM1	LTM0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	LTM2	LTM1	LTM0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	LTM2	LTM1	LTM0	0	0	0	0	0	0

(c) When read or write command is issued

Bit setting		Address pins																		
SSO 1	SSO 0	MAO 28 to MAO 18	MAO 17	MAO 16	MAO 15	MAO 14	MAO 13	MAO 12	MAO 11	MAO 10	MAO 9	MAO 8	MAO 7	MAO 6	MAO 5	MAO 4	MAO 3	MAO 2	MAO 1	MAO 0
0	0	a28 to a18	a17	a16	a15	a14	a12	a11	a10	1[a]	a9	a8	a7	a6	a5	a4	a3	a2	a1	a0
0	1	a28 to a18	a17	a16	a15	a14	a12	a11	1[a]	a10	a9	a8	a7	a6	a5	a4	a3	a2	a1	a0
1	0	a28 to a18	a17	a16	a15	a14	a12	1[a]	a11	a10	a9	a8	a7	a6	a5	a4	a3	a2	a1	a0

a) The value is 1 when read and 0 when written.

11.2.9 SDRAM status register (STR)

The STR register indicates the SDRAM status.

Access This register is read-only, in 8-bit units.

Address $FFFF7224_H$

Initial Value 00_H. This register is initialized by any reset.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	WBF	WCF
R	R	R	R	R	R	R	R

Table 11-13 STR register contents

Bit position	Bit name	Function
1	WBF	This bit indicates whether data is stored in the SDRAMC write buffer. 0: No data in write buffer (initial value) 1: Data in write buffer
0	WCF	After the SDCR register is set, this bit indicates that a register write command for the SDRAM is complete. If the SE bit of the SEN register is cleared to 0, the WCF bit is also cleared to 0. 0: Settings not complete (initial value) 1: Settings complete

11.2.10 SDRAM refresh control register (RFS)

The RFS register is used to set the CBR refresh cycle and self refresh cycle for SDRAM. This register can also be used to set the refresh cycle when SDRAM is initially set (when a register write operation is executed).

Access This register can be read or written in 16-bit units.

Address $FFFF7226_H$

Initial Value 0000_H. This register is initialized by any reset.

Caution To change the setting of the RFS register during SDRAMC operation (SE bit = 1), use the following procedure.

1. Clear the SE bit of the SEN register to 0.
2. Clear the REN bit to 0.
3. While specifying new values for the RIN12 to RIN0 bits, set the REN bit to 1.
4. Set the SDCR register.
5. Make sure that the WCF bit of the STR register is 1, and then set the SE bit of the SEN register to 1.

When switching the refresh interval, set a value that enables the refresh to be in time even during the switch.

15	14	13	12	11	10	9	8
REN	0	0	RIN12	RIN11	RIN10	RIN9	RIN8
R/W	R	R	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
RIN7	RIN6	RIN5	RIN4	RIN3	RIN2	RIN1	RIN0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 11-14 RFS register contents

Bit position	Bit name	Function
15	REN	This bit sets whether to enable refreshing. 0: Refreshing disabled (initial setting) 1: Refreshing enabled
12-0	RIN12 to RIN0	These bits set the refresh interval factor.

RIN12	RIN11	...	RIN2	RIN1	RIN0	Interval factor (Ifac)
0	0	0	0	0	0	1 (initial value)
0	0	0	0	0	1	2
0	0	0	0	1	0	3
0	0	0	0	1	1	4
...	...	...	...	...	...	...
1	1	1	1	1	0	8,191
1	1	1	1	1	1	8,192

Table 11-15 Example of SDRAM Refresh Intervals

Target refresh interval setting (μs)	Interval factor (Ifac)[a]
	When f_{BUSCLK} = 50 MHz
15	750 (15.0)
30	1,500 (30.0)
60	3,000 (60.0)

a) The values in parentheses are the calculated refresh intervals (μs).

<Calculating the interval factor>

Ifac = X (μs) / (1 / f_{BUSCLK} (MHz)): Disregard any fractional component.

X: Target refresh interval setting

f_{BUSCLK}: SDRAM operating frequency

11.3 Bus Cycle Type Setting Function

In the separate bus mode, the type of bus cycle listed below can be used.

- SRAM bus cycle type

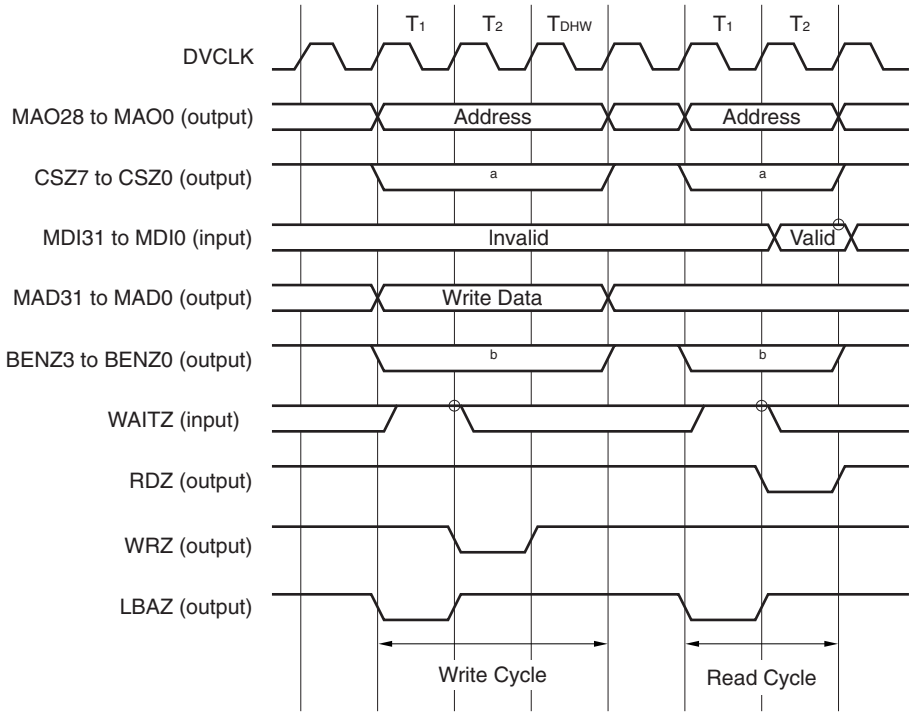
Note that the CS4 area is fixed to the SDRAM bus cycle type and cannot be changed.

11.3.1 SRAM bus cycle type

The SRAM cycle is the most basic bus cycle in the separate bus mode. This microcontroller accesses the external memory using address and chip select signals, and the read strobe signal during the read cycle or the write strobe signal during the write cycle.

In the separate bus mode, this bus cycle is selected for all the chip select areas via initialization through reset.

The read strobe/write strobe cycle can be extended by up to 15 clock cycles with the DWC register. External wait is also enabled through these strobes



a) Indicates the chip select area to be accessed.
b) The data size differs according to the bus size set to the BSC register. For details, refer to *11.9 "Data Flow" on page 539*.

Figure 11-1 SRAM cycle (separate bus mode, read/write)

The valid wait functions in the SRAM cycle are listed below.

- Programmable data wait of 15 clock cycles max. through DWC register
- Data wait through external pin
- Insertion of idle cycle through ICC register
- Data hold wait of up to 3 clock cycles through DHC register

The LBAZ pin becomes active during the SRAM cycle in order to enable access to the Sync-Flash memory or page-ROM memory in the SRAM cycle. When only SRAM memory is used, the LBAZ pin is not used.

11.3.2 SDRAM bus cycle type

This bus cycle is used when connecting SDR-SDRAM that conforms to JEDEC (it is not supported for Mobile SDRAM or SGRAM).

Only the CS4 area is allocated as an SDRAM area. The maximum supported SDRAM memory size is 512 Mb. The row address width corresponds to bits 11, 12, 13, and 14, and the column address width corresponds to bits 8, 9, and 10. The maximum number of banks is four. Both little endian and big endian are supported.

When writing to SDRAM, single write access (burst length: 1) is always used. By buffering data into the SDRAM-controller internal buffer in response to a write cycle request from the CPU, the CPU can immediately perform the following processing. The buffer consists of eight stages, and, regardless of the SDRAM data bus width, can accumulate the data from up to eight writes. If even one item of data is accumulated in the write buffer, the WBF flag of the SDRAM status register is set to 1, and, when the write buffer empties, this flag is cleared to 0. Write access to SDRAM continues sequentially until the write buffer empties.

When reading from SDRAM, the burst read function can be used. This function is enabled when the BST bit of the SDRAM configuration register is set to 1. When this function is enabled, eight sequential read accesses are made in response to a read cycle request from the CPU. If the data bus width is 8 bits, 64 bits including the requested address are sequentially accessed; if the data bus width is 16 bits, 128 bits including the requested address are sequentially accessed; and, if the data bus width is 32 bits, 256 bits including the requested address are sequentially accessed. The read data is saved in the SDRAM-controller internal buffer, and then, when there is a read access request from the CPU, the address is compared to the address of the buffered data and, if the addresses are the same, the data is returned from the buffer without starting an external bus cycle. Any write access to SDRAM discards the read buffer data unconditionally. Also, read requests from the CPU are held pending until the eight data reads finish.

Regardless of whether the burst read function is enabled or disabled, when DRAM is accessed for reading, the SDRAM controller issues active commands followed by auto precharge read commands, regardless of which bank or page is accessed.

The wait function enabled for the SDRAM bus cycle type is as follows:

- Idle cycle insertion by the ICC register (but only the last cycle is enabled)

Table 11-16 List of Address Bus Connecting to SDRAM

Row address (RAW)	Column address (SAW)	Size (SSO)	MAO 15	MAO 14	MAO 13	MAO 12	MAO 11	MAO 10	...	MAO 2	MAO 1	MAO 0	Bank address[a]
11 bits(00)	8 bits(00)	8 bits(00)	-	-	-	-	-	a10	…	a2	a1	a0	MAO21or later
11 bits(00)	8 bits(00)	16 bits(01)	-	-	-	-	a10	a9	…	a1	a0	-	MAO21or later
11 bits(00)	8 bits(00)	32 bits(10)	-	-	-	a10	a9	a8	…	a0	-	-	MAO21or later
12 bits(01)	8 bits(00)	8 bits(00)	-	-	-	-	a11	a10	…	a2	a1	a0	MAO21or later
12 bits(01)	8 bits(00)	16 bits(01)	-	-	-	a11	a10	a9	…	a1	a0	-	MAO21or later
12 bits(01)	8 bits(00)	32 bits(10)	-	-	a11	a10	a9	a8	…	a0	-	-	MAO22or later
13 bits(10)	8 bits(00)	8 bits(00)	-	-	-	a12	a11	a10	…	a2	a1	a0	MAO21or later
13 bits(10)	8 bits(00)	16 bits(01)	-	-	a12	a11	a10	a9	…	a1	a0	-	MAO22or later
13 bits(10)	8 bits(00)	32 bits(10)	-	a12	a11	a10	a9	a8	…	a0	-	-	MAO23or later
14 bits(11)	8 bits(00)	8 bits(00)	-	-	a13	a12	a11	a10	…	a2	a1	a0	MAO22or later
14 bits(11)	8 bits(00)	16 bits(01)	-	a13	a12	a11	a10	a9	…	a1	a0	-	MAO23or later
14 bits(11)	8 bits(00)	32 bits(10)	a13	a12	a11	a10	a9	a8	…	a0	-	-	MAO24or later
11 bits(00)	9 bits(01)	8 bits(00)	-	-	-	-	-	a10	…	a2	a1	a0	MAO20or later
11 bits(00)	9 bits(01)	16 bits(01)	-	-	-	-	a10	a9	…	a1	a0	-	MAO21or later
11 bits(00)	9 bits(01)	32 bits(10)	-	-	-	a10	a9	a8	…	a0	-	-	MAO22or later
12 bits(01)	9 bits(01)	8 bits(00)	-	-	-	-	a11	a10	…	a2	a1	a0	MAO21or later
12 bits(01)	9 bits(01)	16 bits(01)	-	-	-	a11	a10	a9	…	a1	a0	-	MAO22or later
12 bits(01)	9 bits(01)	32 bits(10)	-	-	a11	a10	a9	a8	…	a0	-	-	MAO23or later
13 bits(10)	9 bits(01)	8 bits(00)	-	-	-	a12	a11	a10	…	a2	a1	a0	MAO22or later
13 bits(10)	9 bits(01)	16 bits(01)	-	-	a12	a11	a10	a9	…	a1	a0	-	MAO23or later
13 bits(10)	9 bits(01)	32 bits(10)	-	a12	a11	a10	a9	a8	…	a0	-	-	MAO24or later
14 bits(11)	9 bits(01)	8 bits(00)	-	-	a13	a12	a11	a10	…	a2	a1	a0	MAO23or later
14 bits(11)	9 bits(01)	16 bits(01)	-	a13	a12	a11	a10	a9	…	a1	a0	-	MAO24or later
14 bits(11)	9 bits(01)	32 bits(10)	a13	a12	a11	a10	a9	a8	…	a0	-	-	MAO25or later
11 bits(00)	10 bits(10)	8 bits(00)	-	-	-	-	-	a10	…	a2	a1	a0	MAO21or later
11 bits(00)	10 bits(10)	16 bits(01)	-	-	-	-	a10	a9	…	a1	a0	-	MAO22or later
11 bits(00)	10 bits(10)	32 bits(10)	-	-	-	a10	a9	a8	…	a0	-	-	MAO23or later
12 bits(01)	10 bits(10)	8 bits(00)	-	-	-	-	a11	a10	…	a2	a1	a0	MAO22or later
12 bits(01)	10 bits(10)	16 bits(01)	-	-	-	a11	a10	a9	…	a1	a0	-	MAO23or later
12 bits(01)	10 bits(10)	32 bits(10)	-	-	a11	a10	a9	a8	…	a0	-	-	MAO24or later
13 bits(10)	10 bits(10)	8 bits(00)	-	-	-	a12	a11	a10	…	a2	a1	a0	MAO23 or later
13 bits(10)	10 bits(10)	16 bits(01)	-	-	a12	a11	a10	a9	…	a1	a0	-	MAO24 or later
13 bits(10)	10 bits(10)	32 bits(10)	-	a12	a11	a10	a9	a8	…	a0	-	-	MAO25 or later
14 bits(11)	10 bits(10)	8 bits(00)	-	-	a13	a12	a11	a10	…	a2	a1	a0	MAO24 or later
14 bits(11)	10 bits(10)	16 bits(01)	-	a13	a12	a11	a10	a9	…	a1	a0	-	MAO25 or later
14 bits(11)	10 bits(10)	32 bits(10)	a13	a12	a11	a10	a9	a8	…	a0	-	-	MAO26 or later
11 bits(00)	11 bits(11)	8 bits(00)	-	-	-	-	-	a10	…	a2	a1	a0	MAO22 or later
11 bits(00)	11 bits(11)	16 bits(01)	-	-	-	-	a10	a9	…	a1	a0	-	MAO23 or later
11 bits(00)	11 bits(11)	32 bits(10)	-	-	-	a10	a9	a8	…	a0	-	-	MAO24 or later
12 bits(01)	11 bits(11)	8 bits(00)	-	-	-	-	a11	a10	…	a2	a1	a0	MAO23 or later
12 bits(01)	11 bits(11)	16 bits(01)	-	-	-	a11	a10	a9	…	a1	a0	-	MAO24 or later
12 bits(01)	11 bits(11)	32 bits(10)	-	-	a11	a10	a9	a8	…	a0	-	-	MAO25 or later
13 bits(10)	11 bits(11)	8 bits(00)	-	-	-	a12	a11	a10	…	a2	a1	a0	MAO24 or later
13 bits(10)	11 bits(11)	16 bits(01)	-	-	a12	a11	a10	a9	…	a1	a0	-	MAO25 or later
13 bits(10)	11 bits(11)	32 bits(10)	-	a12	a11	a10	a9	a8	…	a0	-	-	MAO26 or later
14 bits(11)	11 bits(11)	8 bits(00)	-	-	a13	a12	a11	a10	…	a2	a1	a0	MAO25 or later
14 bits(11)	11 bits(11)	16 bits(01)	-	a13	a12	a11	a10	a9	…	a1	a0	-	MAO26 or later
14 bits(11)	11 bits(11)	32 bits(10)	a13	a12	a11	a10	a9	a8	…	a0	-	-	MAO27 or later

a) The space address to be accesses is output as that of bank address output signal. For example, when the address output bus that can be used as bank address is MAO21 or later, the output of MAO21 becomes "1" when address space where MAO21 becomes "1" is accessed.

Note MAOxx : Address output bus, axx :Connecting address pins to be assumed, -: No use

11.4 Bus Control Function

11.4.1 Chip select output function

The connected external memory area is managed divided into 4 chip select areas up to CSn (n = 1 to 4), as shown in *Figure 11-2 “External memory map”*.

When a bus cycle is generated for the external bus, This microcontroller makes the CSZn (n = 1 to 4) output pins corresponding to the access target address active (low level), along with outputting the access target address from the MAO[23:0] pins.

The various settings for the external bus, such as the bus size and number of wait/idle states, can all be made for each chip select area.

By using these functions, different types of memory can be connected for each chip select area.

The allocation of the chip select areas is fixed by the system and cannot be changed through programming.

The memory map is shown next.

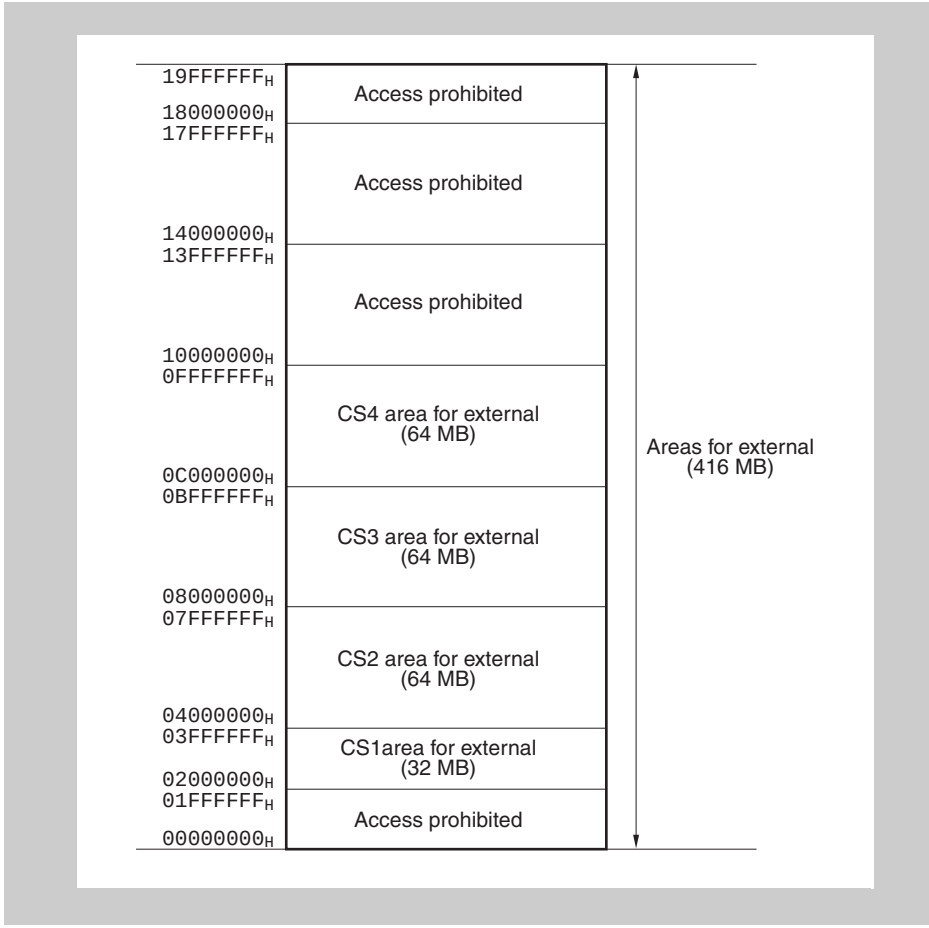
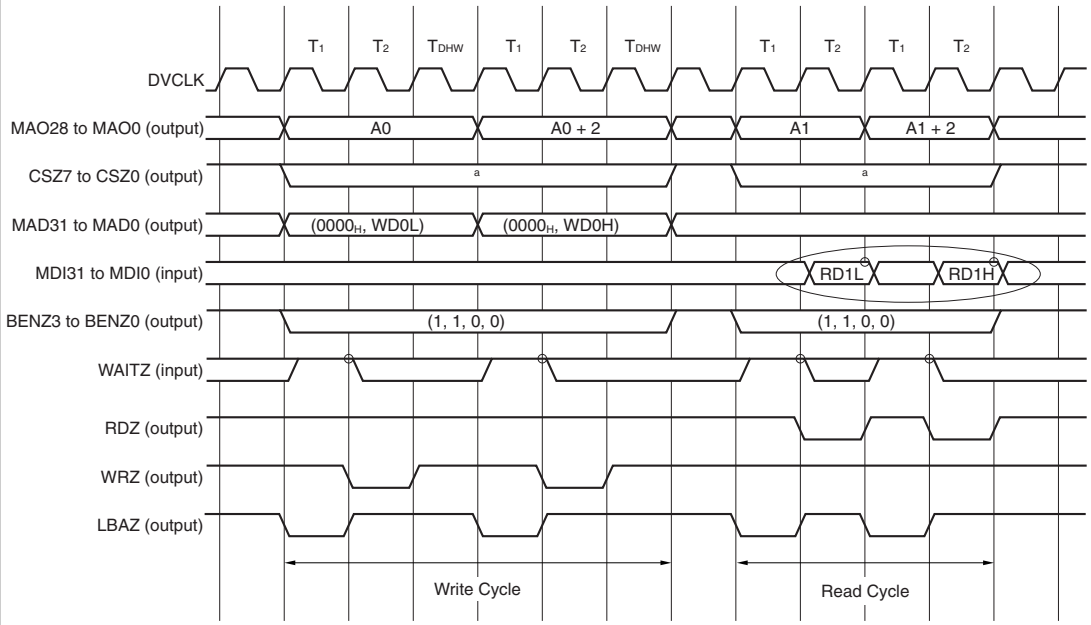


Figure 11-2 External memory map

11.4.2 Bus size setting function

Access requests from the CPU (or DMA) are executed after being divided in accordance with the bit width of the external bus of the access destination.

The bit width of the external bus can be selected from 32, 16, and 8 bits for each chip select area by setting the BSC register.



a) Indicates the chip select area to be accessed.

Figure 11-3 SRAM cycle for 16-bit bus size

11.4.3 Data endian setting function

Either little endian or big endian can be selected as the data endian of the external bus interface. This setting can be made for each chip select area with the DEC register. Initial setting through input pins is not possible. The initial status is little endian for all the chip select areas.

Access to a chip select area for which big endian has been specified as the data endian is performed in big endian.

This function can be used only for the SRAM access type.

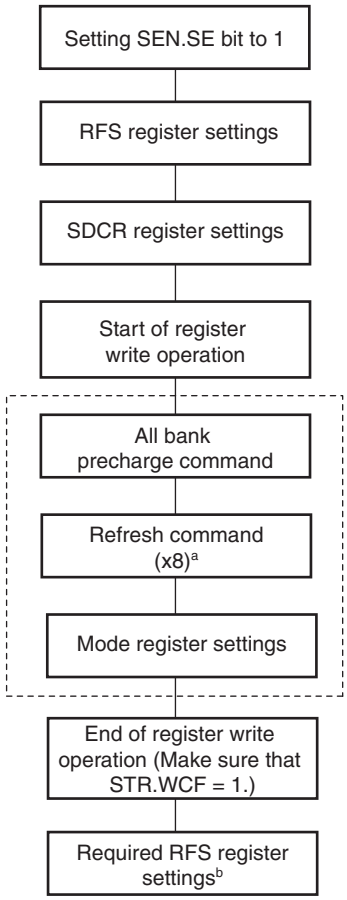
Cautions

1. In this microcontroller, instruction fetch operation with big endian is not supported.
2. Misaligned access to these areas is prohibited when big endian has been selected.

Note For details about the data flow for each external bus size and data size, refer to *11.9 “Data Flow” on page 539*.

11.4.4 SDRAM setting function

To use SDRAM, first set the SE bit of the SDRAM enable control register (SEN) to 1, and then set the SDRAM refresh control register (RFS). When the SDCR register is written to, a register write operation is performed. Before the execution of a register write operation, no SDRAM read/write cycles occur. When the register write operation ends, the STR.WCF bit is set to 1. Before starting to access SDRAM, be sure to read the STR register values, and make sure that the WCF bit is set.

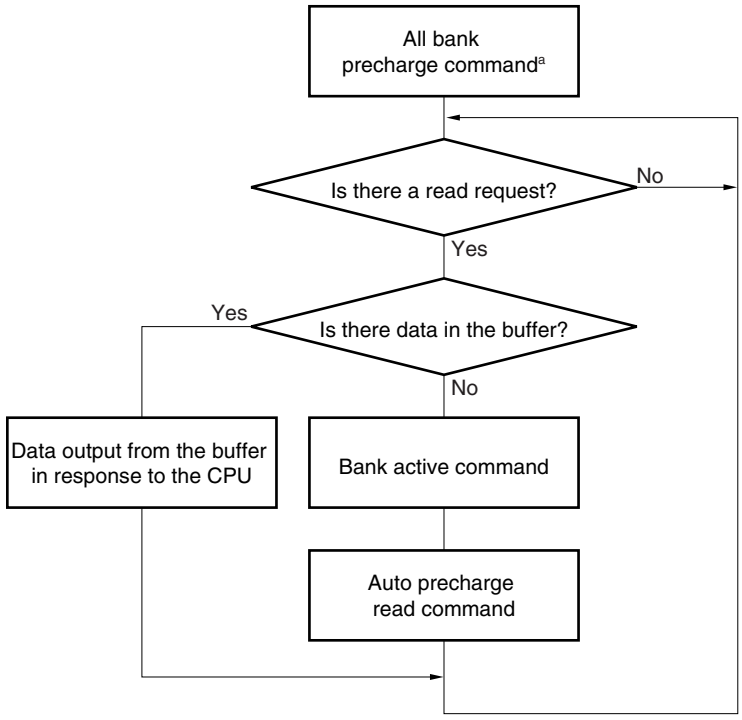


a) Each refresh command interval during a register write operation and the interval from the eighth refresh command to the setting of the mode register are the interval set with the RFS.RIN bit.
b) Set the REN bit to 1. Also change the value of RIN [12:0] if the refresh interval during a register write operation does not match the refresh interval during normal access.

Figure 11-4 Initial register settings - register write operation flow

(1) SDRAM read access

When accessing SDRAM for reading, the burst read function can be used. When this function is enabled, eight sequential read accesses are made, regardless of the data bus width. Read requests from the CPU are held pending until the eight data reads are complete. The read data is saved in the SDRAM controller internal buffer, and then, when there is a read access request from the CPU, the address is compared to the address of the buffered data and, if the addresses are the same, the data is returned from the buffer without starting an external bus cycle. Any write access to SDRAM discards the read buffer data unconditionally.



a) After powering on, after a bus hold, or during a refresh, an all bank precharge command is always issued for SDRAM.

Figure 11-5 SDRAM read access status transitions

(2) Read buffer operation

The size of the SDRAM read buffer can be up to 256 bits. If the burst read function is enabled, this buffer can accumulate the data from up to eight reads. The method for storing data in the buffer differs depending on the data bus width. Operations for each data bus width are described below. If the write buffer is written to even once, or when the system goes into the bus hold state, all the data in the read buffer is deleted.

(a) Data bus width (8 bits)

During 8-bit access, 64 bits, including the address of the data to be read, are stored in the buffer. If the lower 4 bits of the address of the target data are in the range from 0_H to 7_H, 0_H is used as the starting address for reading, and the data from eight burst read accesses (up to 7_H) is stored in the buffer. If the lower 4 bits of the address of the target data are in the range from 8_H to F_H, 8_H is used as the starting address for reading, and the data from eight burst read accesses (up to F_H) is stored in the buffer.

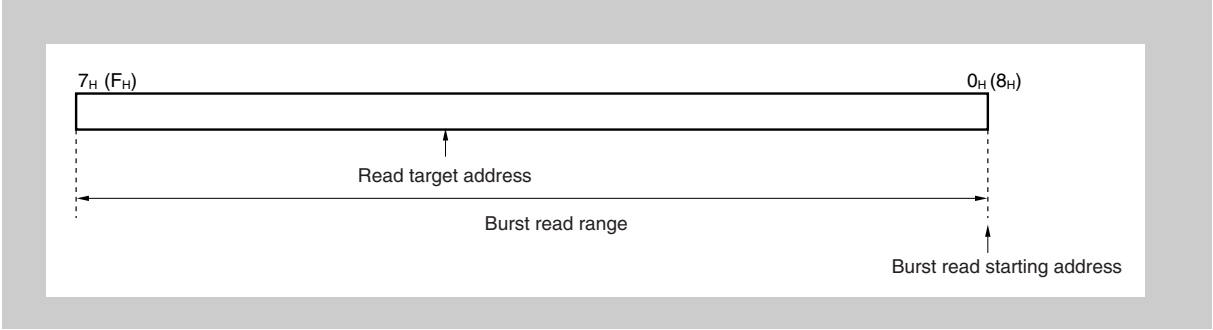


Figure 11-6 Data bus width (8 bits)

(b) Data bus width (16 bits)

During 16-bit access, 128 bits, including the address of the data to be read, are stored in the buffer. The lower 4 bits of the accessed address are monitored, and, within the 128-bit data range that includes the address of the target data, 0_H is used as the starting address for reading and the data from eight burst read accesses (in the range up to F_H) is stored in the buffer.

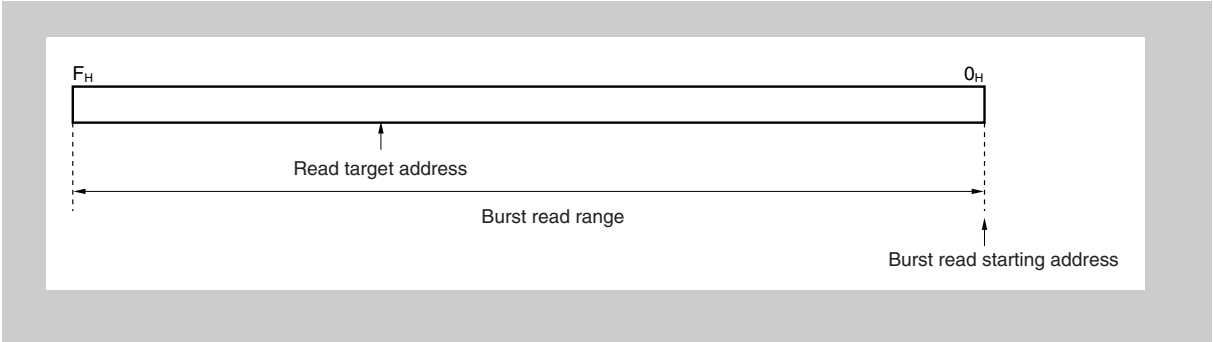


Figure 11-7 Data bus width (16 bits)

(c) Data bus width (32 bits)

During 32-bit access, 256 bits, including the address of the data to be read, are stored in the buffer. The lower 5 bits of the accessed address are monitored, and, within the 256-bit data range that includes the address of the target data, 00_H is used as the starting address for reading and the data from eight burst read accesses (in the range up to $1F_H$) is stored in the buffer.

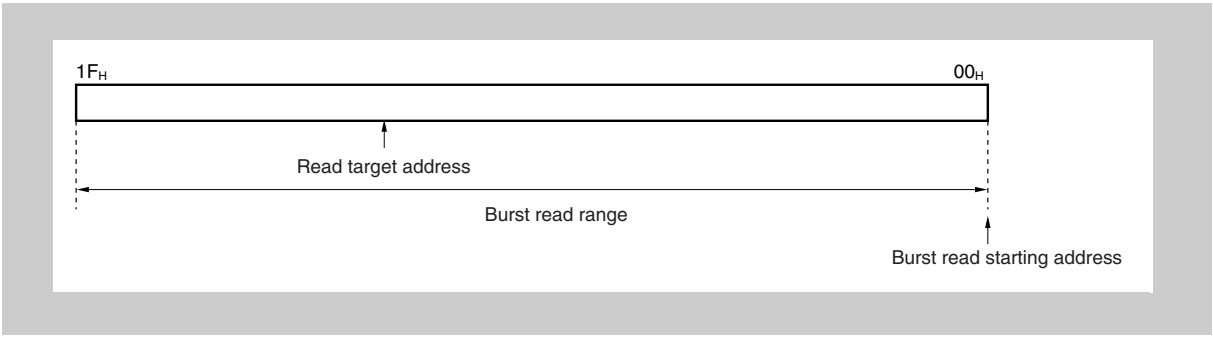
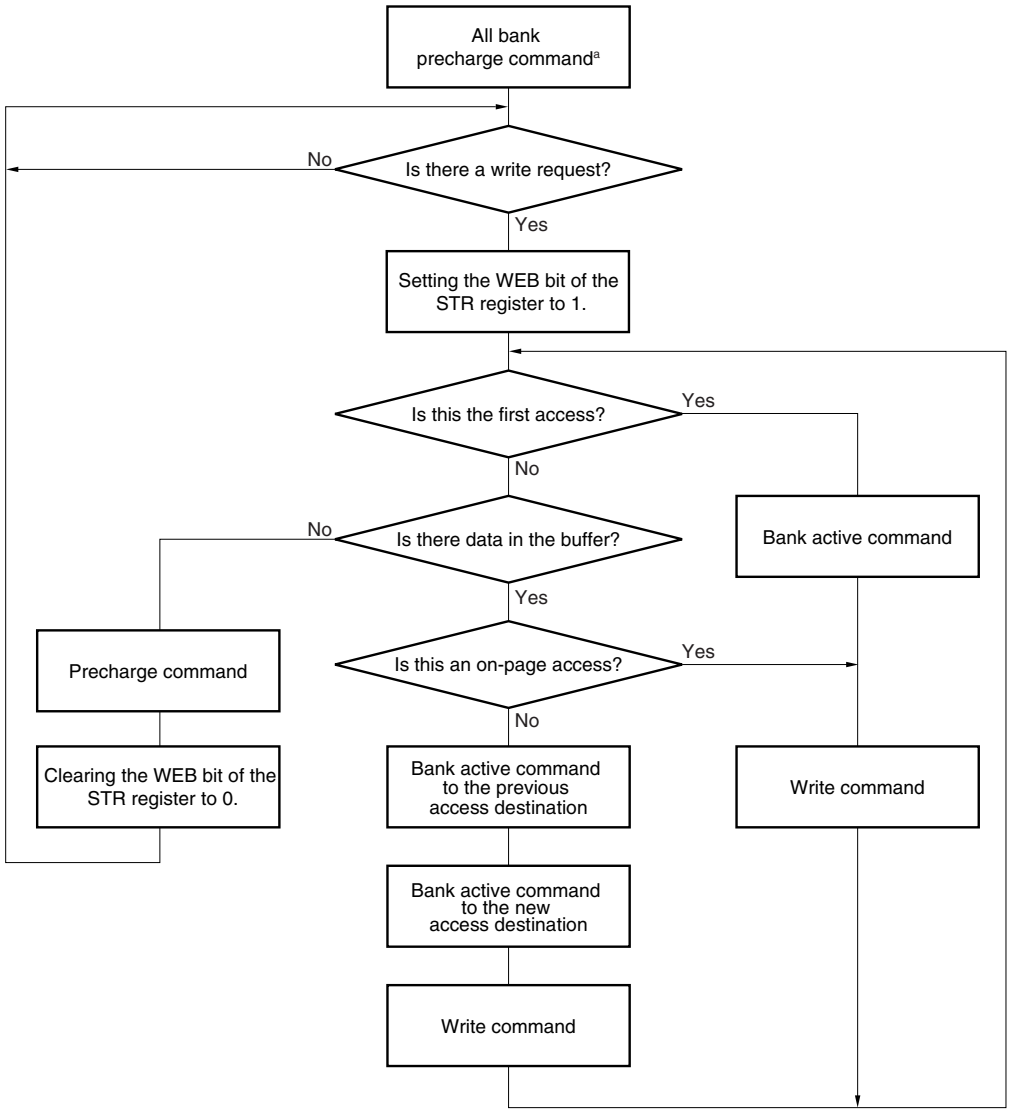


Figure 11-8 Data bus width (32 bits)

(3) SDRAM write access

When writing to SDRAM, single write access (burst length: 1) is always used. By buffering data into the SDRAM controller internal buffer in response to a write cycle request from the CPU, the CPU can immediately perform the following processing. The buffer consists of eight stages, and, regardless of the SDRAM data bus width, can accumulate the data from up to eight writes. Write access to SDRAM continues sequentially until the write buffer empties.



a) After powering on, after a bus hold, or during a refresh, an all bank precharge command is always issued for SDRAM.

Figure 11-9 SDRAM write access status transitions

(4) Write buffer operation

The size of the SDRAM write buffer, which is an 8-stage FIFO buffer, is 256 bits. When the write buffer runs out of space, write access from the CPU is held pending. Also, read access from the CPU is held pending until there is no data in the write buffer.

(5) CBR refresh function

The SDRAM controller starts a CBR refresh cycle at the refresh interval specified by the RFS register. When a CBR refresh command is issued, if an external bus cycle is occurring, the CBR refresh command is held pending until this cycle ends. Also, if an external bus hold request conflicts with a CBR refresh request, the external bus hold request is prioritized. If a refresh is requested during an external bus hold, the REFRQZ signal can be activated to indicate that a refresh has been requested for the external bus master. If a refresh cycle must be entered during an external bus hold, this signal can be used to cancel the bus hold request.

Note Because the refresh counter performs free-running operations, if a CBR refresh command is held pending, the interval until the next CBR refresh cycle is shortened.

(6) Self refresh function

When the CPU enters standby mode, the SDRAM controller performs the following procedure to start an SDRAM self refresh cycle:

<1> If the primary memory controller is accessing the external bus, after waiting for the access to end, MEMC sends a standby request signal to the SDRAM controller. (At this time, the SDRAM controller does not access SDRAM.)

→ Go to <4>.

If the primary memory controller is not accessing the external bus, Primary memory controller immediately sends a standby request signal to the SDRAM controller.

<2> If the SDRAM controller is accessing SDRAM for reading, MEMC waits for the end of read access to SDRAM.

- If the burst read function is enabled, MEMC waits for the completion of eight read accesses.
- If the timing at which a standby request conflicts with a read request to SDRAM, the reading of SDRAM is prioritized.

<3> If the SDRAM controller is accessing SDRAM for writing, MEMC waits for the end of the write access to SDRAM.

- All data accumulated in the write buffer is written to SDRAM.
- If the timing at which a standby request signal is activated conflicts with a write request to SDRAM, the write request is prioritized.

<4> The all bank precharge command, NOP command, and self refresh command are issued.

<5> SDRAM enters the self refresh status.

<6> Standby mode is entered.

<7> Processing to release the self refresh begins. (The NOP command is issued, and (BCW set value x 4) clock waits are inserted.)

<8> The SDRAM self refresh status is released.

<9> The system returns to the normal status.

If an external bus hold request conflicts with a request to enter standby mode, the request to enter standby mode is prioritized.

11.5 Wait Functions

Wait functions listed below.

Table 11-17 Wait functions

Wait function		Data wait		Data hold wait	Idle
		Programmable	External wait		
SRAM bus cycle type	Read	√	√	-	√
	Write	√	√	√	√
SDRAM bus cycle type	Read	–	–	–	√
	Write	–	–	–	√
Setting registers		DWC0	–	DHC	ICC0 ICC1
Max. number of waits		15	–	3	3

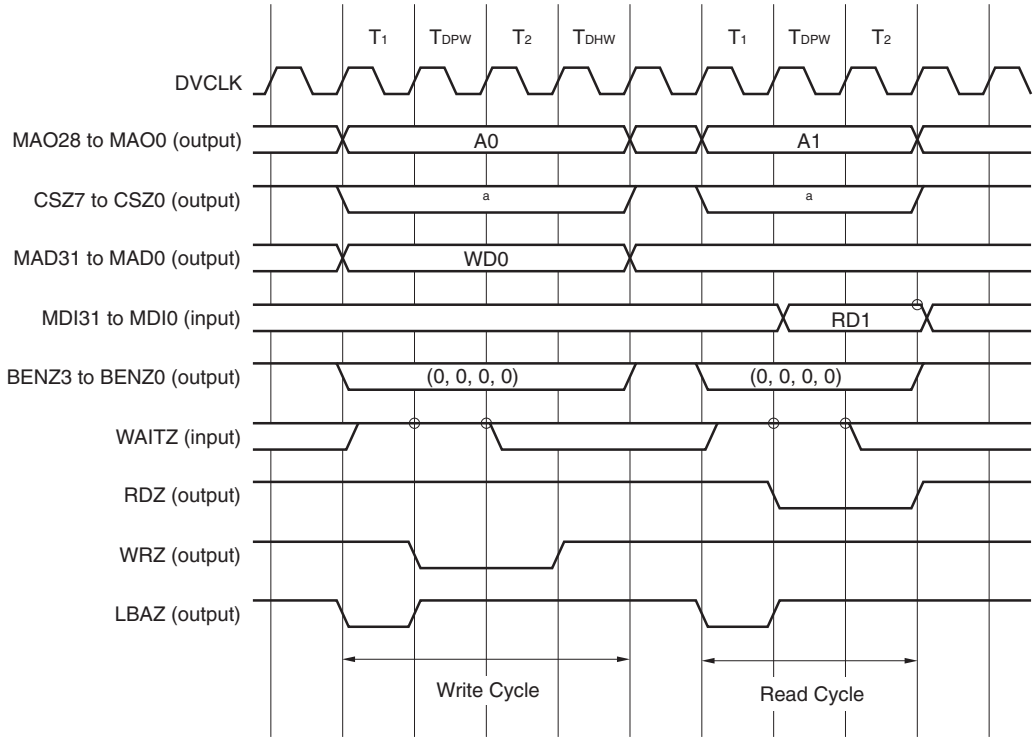
11.5.1 Programmable data wait function

This wait function is for delaying the data latch timing by extending the read strobe and write strobe periods.

This function is enabled during all write accesses and at the first data transfer timing in the SRAM.

Up to 15 cycles can be inserted.

Setting individual chip select areas with the DWC0 register is possible.



a) Indicates the chip select area to be accessed.

Figure 11-10 Programmable data wait in separate bus mode

11.5.2 External wait function

If the separate bus mode or the multiplexed bus mode is selected as the bus cycle type, data waits of any length can be inserted from the WAITZ pin.

The WAITZ pin input level is sampled immediately after completion of the T_A, T_1 cycles and the T_{DPW}, T_{DEW} cycles.

Data wait cycles obtained by ORing the programmable data wait set by data wait control registers 0 and 1 (DWC0 registers) and the external wait specification set by the WAITZ pin input, are inserted.

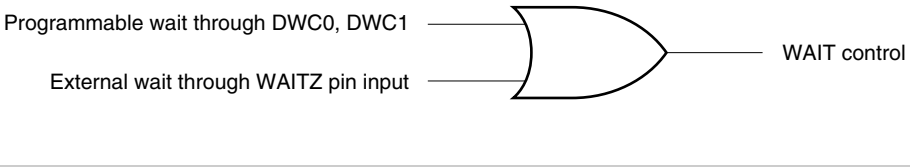
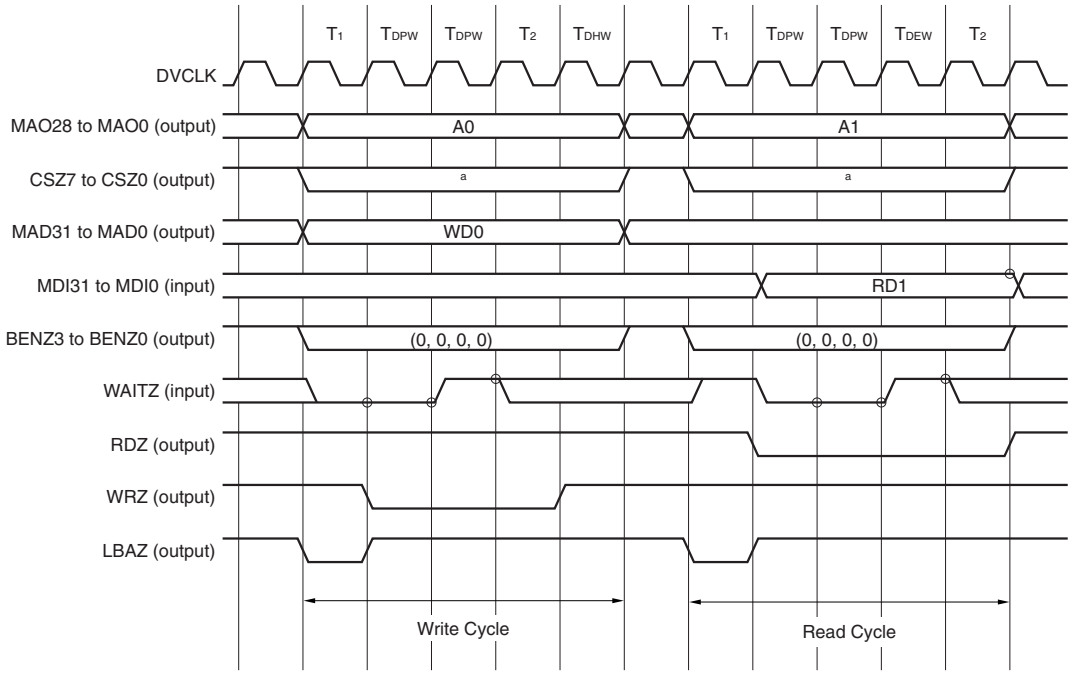


Figure 11-11 Internal data wait generator



a) Indicates the chip select area to be accessed.

Figure 11-12 Relationship between external data wait and programmable data wait (When DWC = 2)

11.5.3 External wait error detection function

This microcontroller has a function to forcibly cancel the data wait if an external wait is continuously input for 128 cycles and report the error response to the CPU (or DMA) that issued the access request, by setting the EWn bit of the external wait error setting register (EWC register). A SysError exception occurs in the CPU at this time.

Using this function, even if an unforeseen bug occurs at the WAITZ input pin, the system does not hang up and exception processing for detecting the anomaly can be executed.

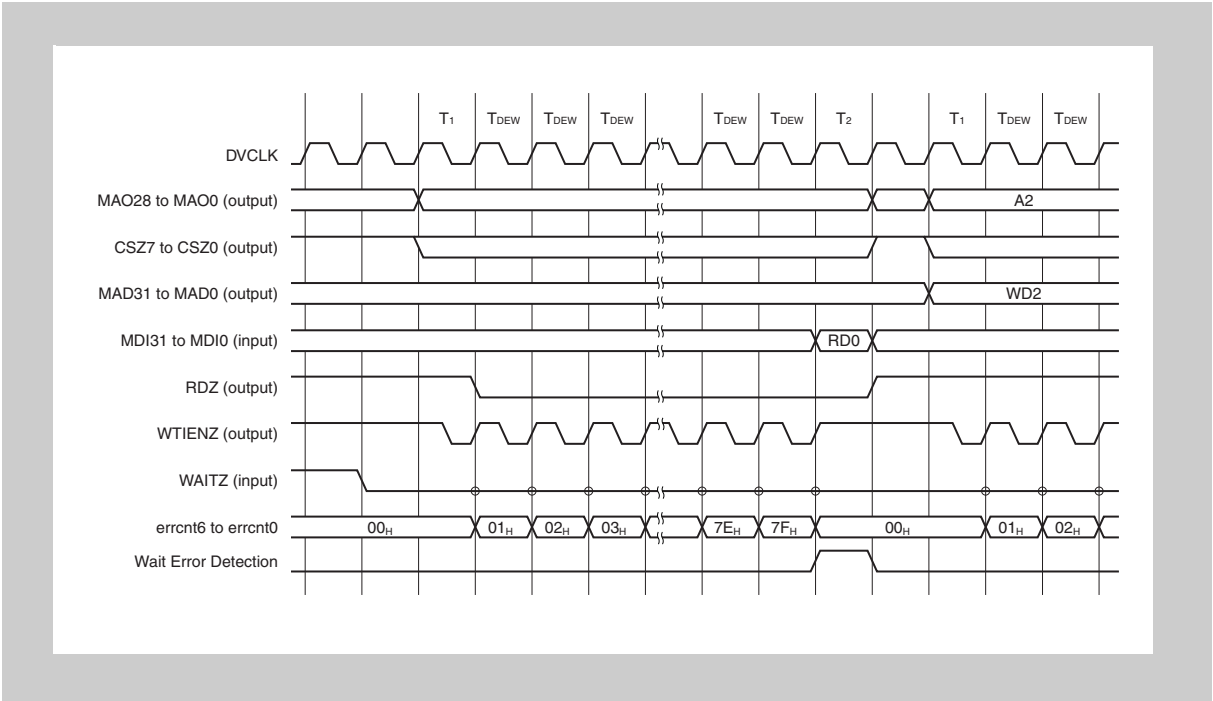


Figure 11-13 Operation timing upon external wait error detection

Following the occurrence of an error at the CPU that issued the access request, the external wait error detection circuit is initialized, and all transfer requests, including data wait requests from the WAITZ input pin, are processed normally.

At this time, when an external wait is continuously input for 128 cycles again, data wait is forcibly released, and an error is reported to the CPU that issued the access request.

11.5.4 Data setup wait function

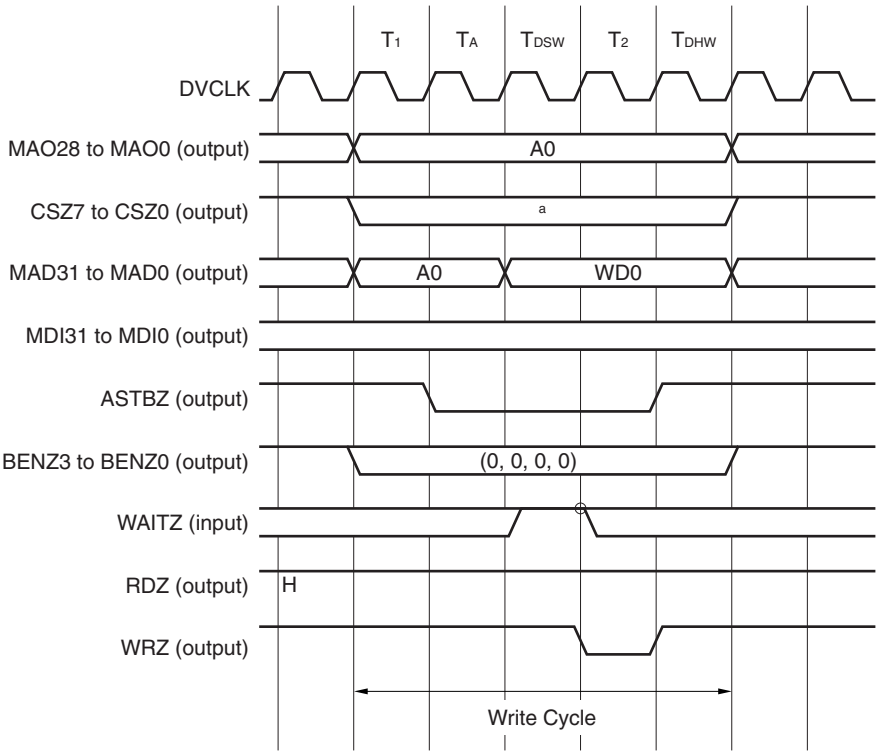
This function inserts a wait prior to the transfer state in order to secure the setup time for the data write strobe.

This function is enabled only during write cycles in the multiplexed bus mode.

Up to 3 cycles can be inserted.

The number of wait cycles to be inserted can be set for each chip select area with the DSC register.

The initial status is no wait for any of the chip select areas.



a) Indicates the chip select area to be accessed.

Figure 11-14 Data setup wait (separate bus mode)

11.5.5 Idle insertion function

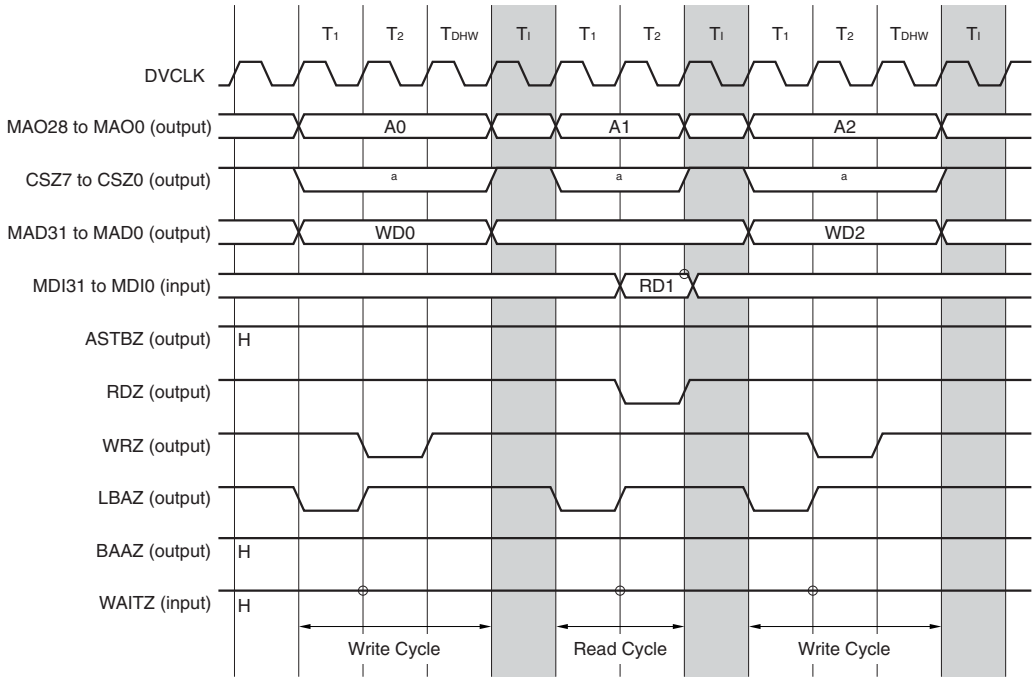
This function inserts an idle state after the last state of each cycle in order to prevent bus conflicts between cycles.

Up to 3 cycles can be inserted for all memory types.

This function can be set independently after a read cycle or after a write cycle for each chip select area by setting the ICC0 and ICC1 registers.

The initial status is no idle cycle for any of the chip select areas.

Caution The interval from completion of a bus cycle until the occurrence of subsequent bus cycles from the CPU (or DMA) lasts 1 cycle, regardless of the idle cycle setting. Therefore, a 1-cycle interval occurs between bus cycles even if the setting is no idle cycle.



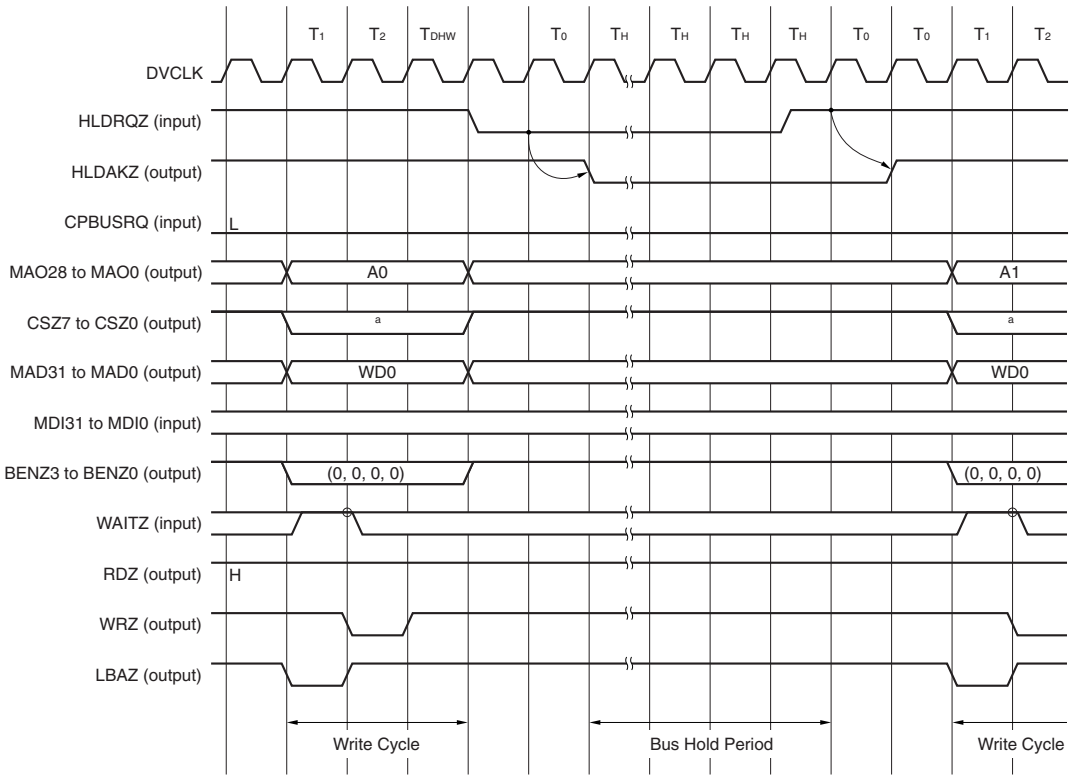
a) Indicates the chip select area to be accessed.

Figure 11-15 Idle cycle

11.6 Bus Hold Function

When the HLDRQZ pin input signal becomes active (low level) and a bus hold request is received from outside, this microcontroller enters the bus hold cycle following completion of the external bus cycle currently being executed. Upon completion of the transition to the bus hold cycle, the active level is output from the HLDAKZ output pin.

When the HLDRQZ pin becomes inactive (high level), the external bus goes into the normal operation mode.



a) Indicates the chip select area to be accessed.

Figure 11-16 Bus hold function

During execution of bus cycles involving lock, such as read-modify-write from the CPU, the transition to the bus hold cycle is held pending until the bus cycle for that lock is completed.

Sampling of the HLDRQZ input pin is done at the same timing as the rising edge of the operation clock of the external bus.

When an external bus hold request is issued, the bank information internally held by the primary memory controller is deleted, and the primary memory controller cycle after the external bus hold is released always starts from the issuance of a bank active command. Also, if the external bus master accesses SDRAM, it must always start from the issuance of an all bank precharge command.

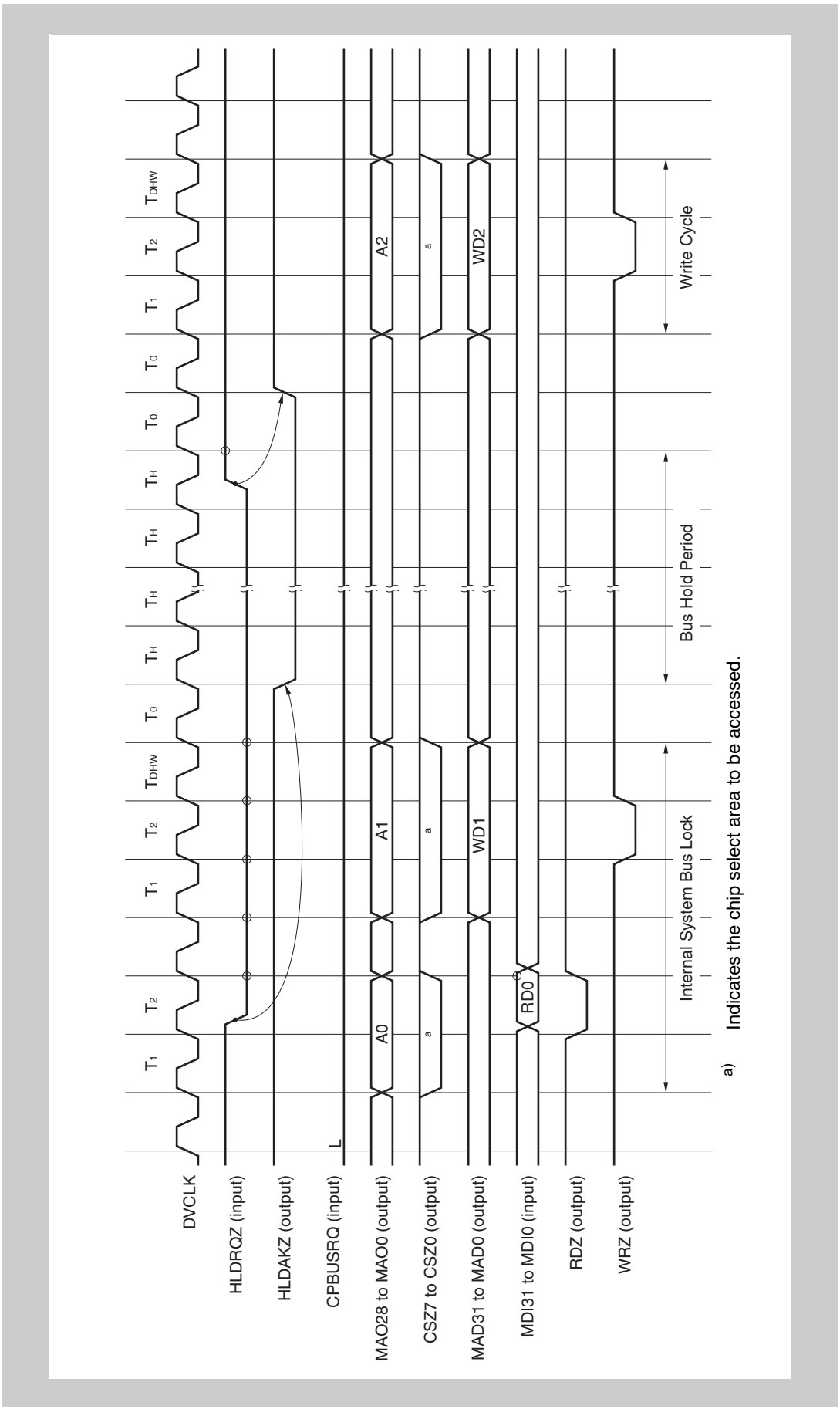


Figure 11-17 Delay of bus hold cycle through internal system bus lock

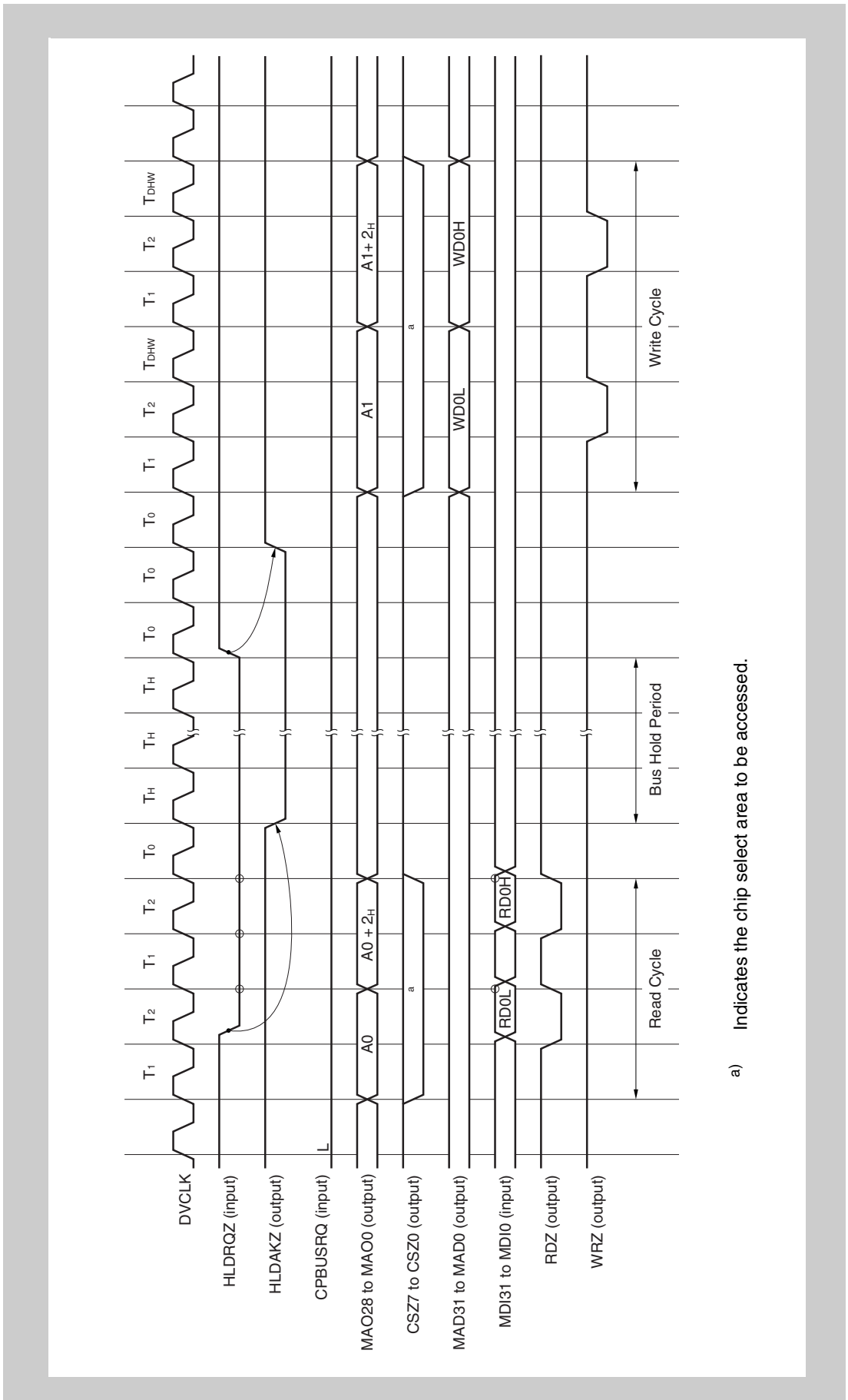
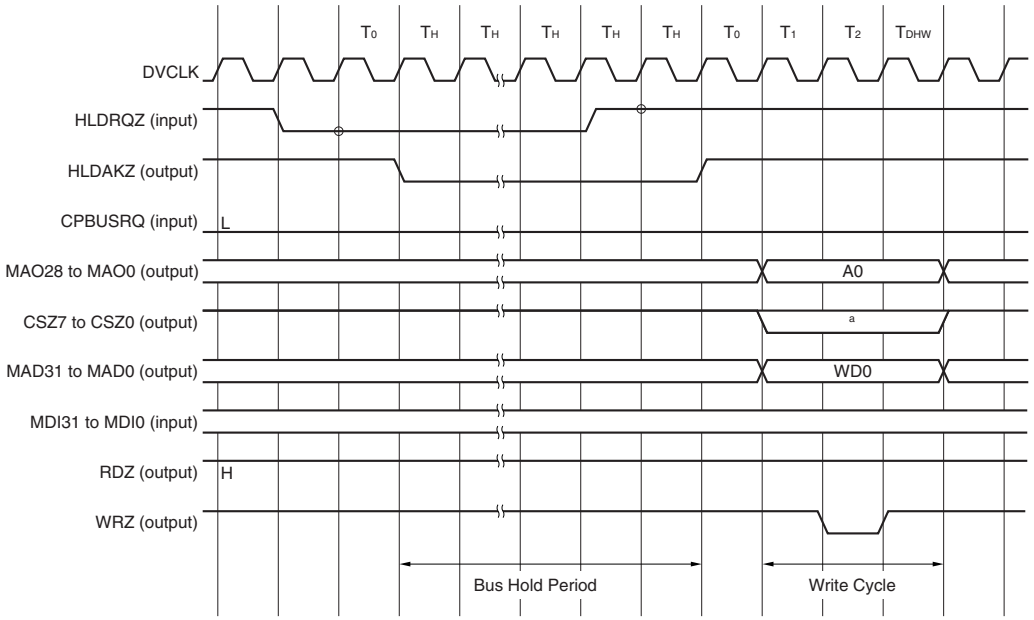


Figure 11-18 Delay of bus hold cycle through bus sizing

If a bus hold request and a transfer request from CPU occur at the same time, a wait response is returned for the transfer request and execution of the transition to the bus hold cycle is given priority.



a) Indicates the chip select area to be accessed.

Figure 11-19 Operation during conflict between bus hold request and transfer request

11.7 DMA Cycle Output Function

When transfer is output from the DMA to the external bus, this microcontroller can output the DMA transfer cycle signal and DMA last transfer signal.

By using this function, whether or not the external bus cycle currently generated for a device outside the chip is caused by the DMA, and by DMA transfer of which channel if it is caused by the DMA, can be reported.

This microcontroller performs output the DMA transfer cycle signal from the DMACH[7:0] pins, and the DMA last transfer signal from the DMALAST[7:0] pins in synchronization with the external bus cycle.

For details about the DMACH[7:0] occurrence timing, DMALAST[7:0] output timing, and the output settings, refer to *Chapter 10 DMA Function*.

Caution This microcontroller allows the connection of a DMA subsystem equipped with 16 channels, but this pin function supports only an 8-channel connection.

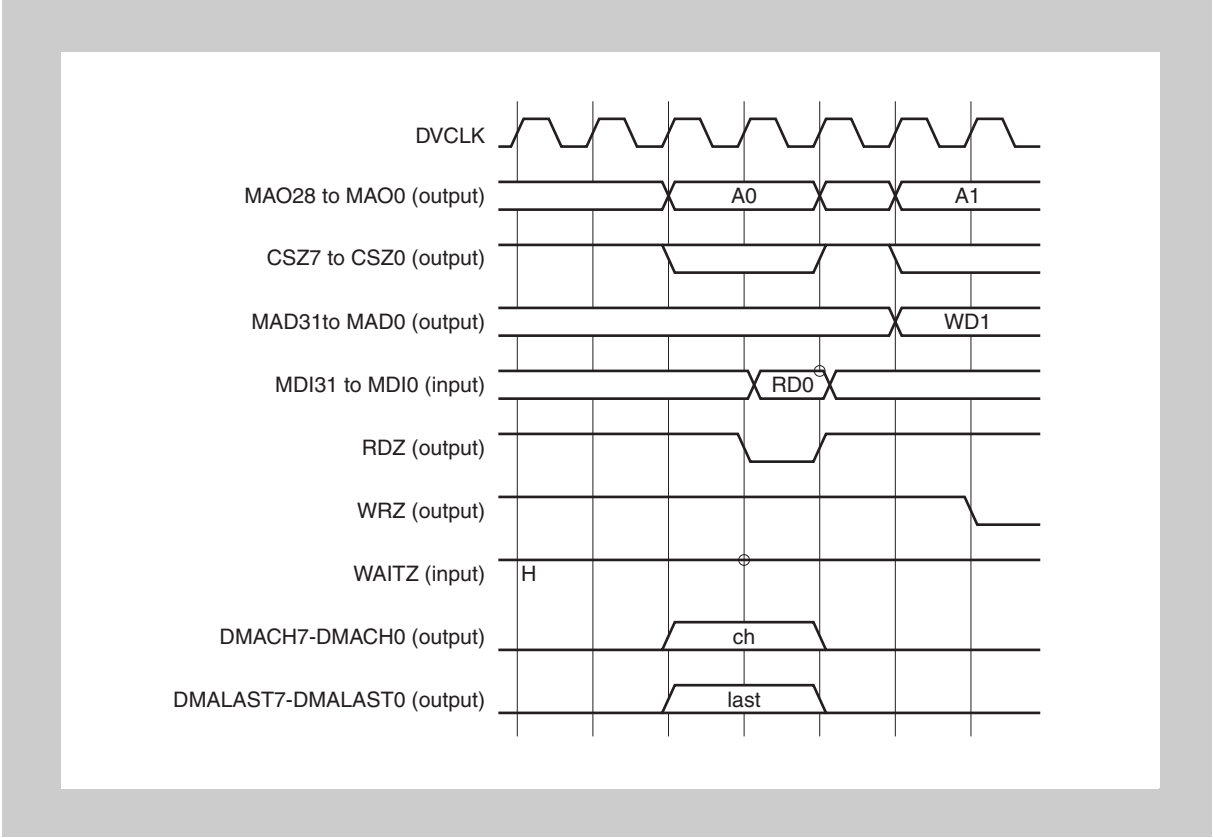


Figure 11-20 DMA channel output timing

11.8 Memory Connection Examples

11.8.1 SRAM connection example

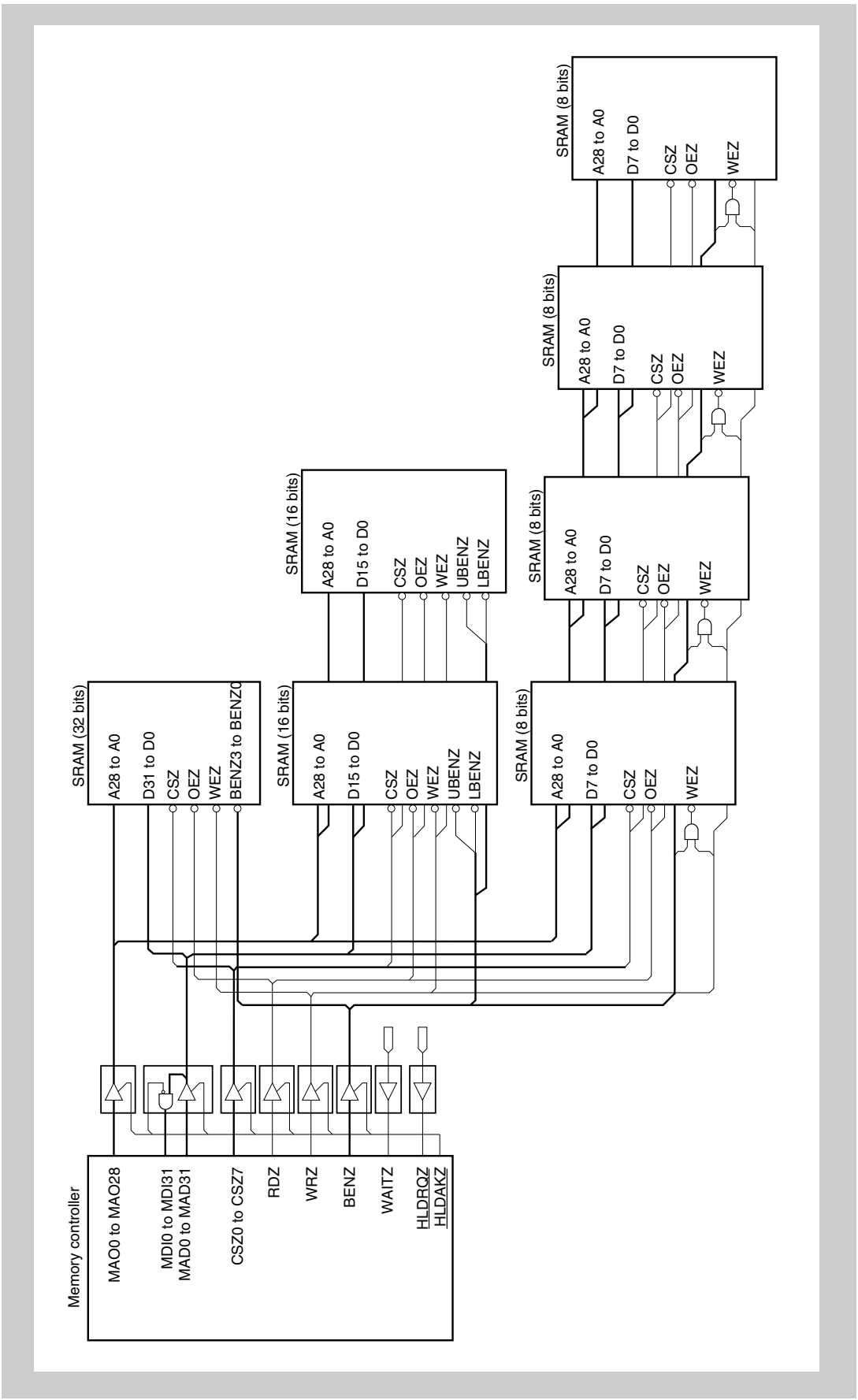


Figure 11-21 SRAM connection example

11.8.2 SDRAM connection example

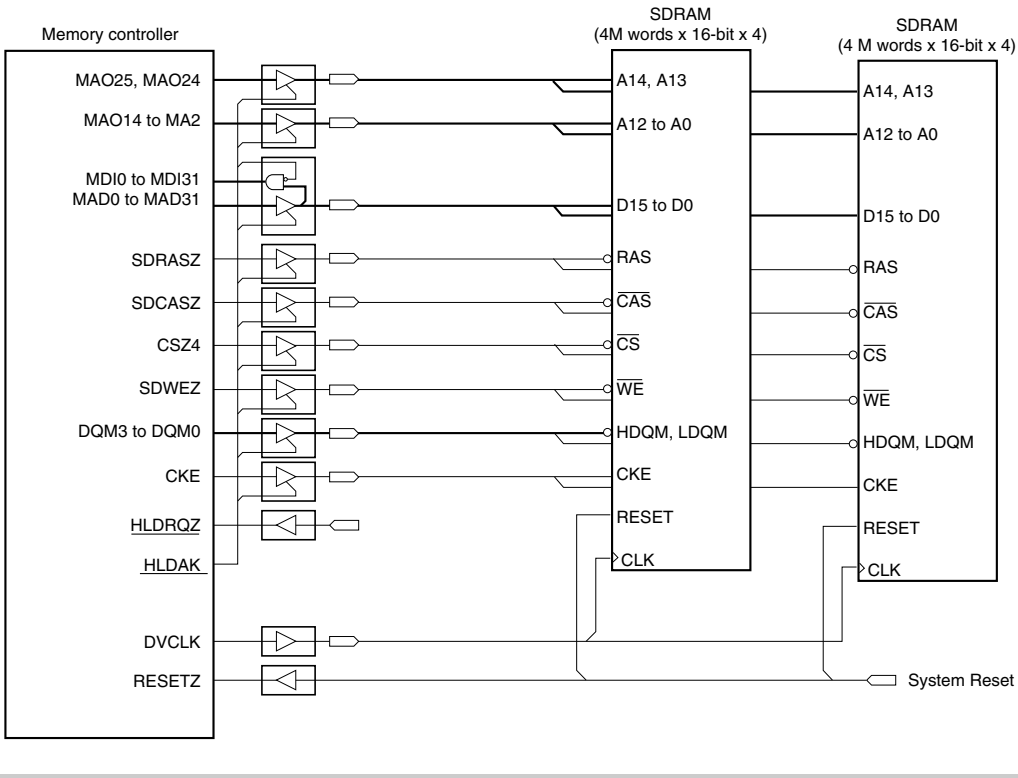


Figure 11-22 512 Mb SDRAM connection example (256-bit SDRAM (4M words × 16 bits × 4 banks) × 2)

(1) Address output and SDRAM connection

This section describes the settings of SDRAM configuration register n (SDCRn), the physical addresses, the address output from memory controller, and the connection between memory controller and SDRAM for each data bus width (8 bits, 16 bits, and 32 bits).

(a) 8-bit data bus width

The following shows an example of how to connect 64 Mb of SDRAM (2M words × 8 bits × 4 banks) for an 8-bit data bus width.

- SDCRn register settings

  SSO1, SSO0 = 00: Data bus width = 8 bits
  RAW1, RAW0 = 01: Row address width = 12 bits
  SAW1, SAW0 = 01: Column address width = 9 bits

- Physical address

  A22, A21: Bank address
  A20 to A9: Row address
  A8 to A0: Column address

- Address output from memory controller

  A22, A21: Bank address
  A11 to A0: Row address (12 bits), column address (9 bits)

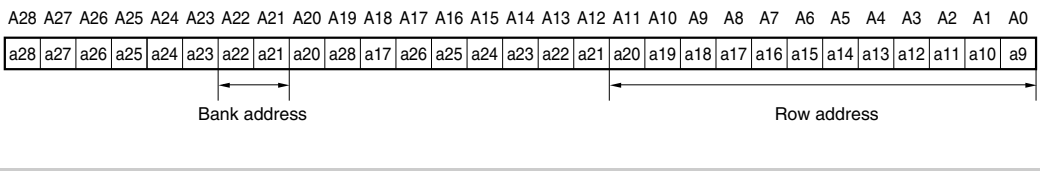
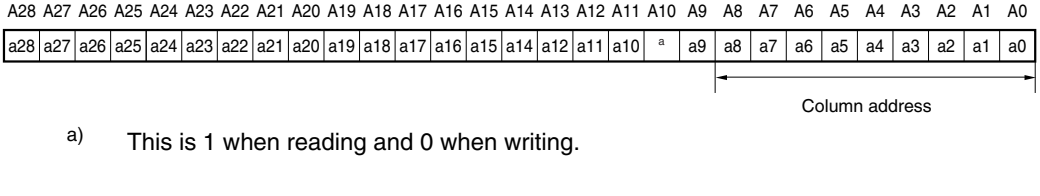


Figure 11-23 Row address, bank address output when active command issued (8 bits)



a) This is 1 when reading and 0 when writing.

Figure 11-24 Column address output when read/write command issued (8 bits)

- Connection of memory controller and SDRAM

  A22, A21 (memory controller): BA0 (A13), BA1 (A12) (SDRAM)
  A11 to A0 (memory controller): A11 to A0 (SDRAM)

(b) 16-bit data bus width

The following shows an example of how to connect 512 Mb of SDRAM (8M words × 16 bits × 4 banks) for a 16-bit data bus width.

- SDCRn register settings

  SSO1, SSO0 = 01: Data bus width = 16 bits
  RAW1, RAW0 = 10: Row address width = 13 bits
  SAW1, SAW0 = 10: Column address width = 10 bits

- Physical address

  A25, A24: Bank address
  A23 to A11: Row address
  A10 to A1: Column address

- Address output from memory controller

  A25, A24: Bank address
  A13 to A1: Row address (13 bits), column address (10 bits)

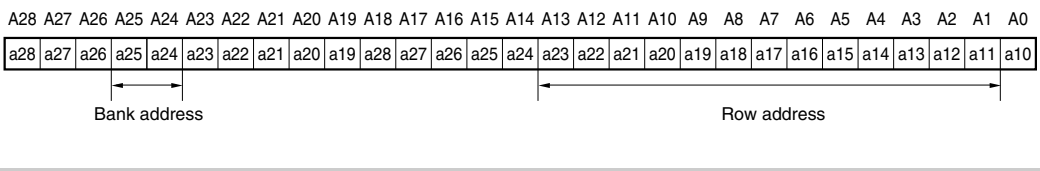
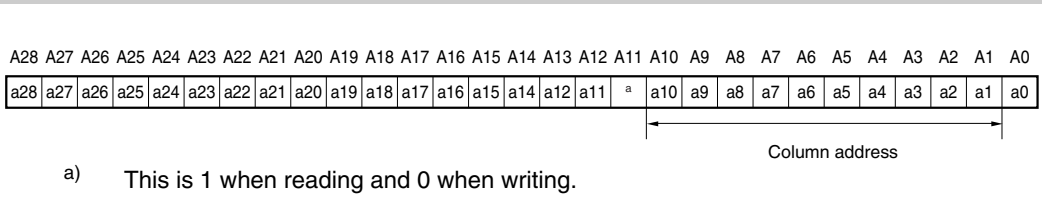


Figure 11-25 Row address, bank address output when active command issued (16 bits)



a) This is 1 when reading and 0 when writing.

Figure 11-26 Column address output when read/write command issued (16 bits)

- Connection of memory controller and SDRAM

  A25, A24 (memory controller): BA0 (A14), BA1 (A13) (SDRAM)
  A13 to A1 (memory controller): A12 to A0 (SDRAM)

(c) 32-bit data bus width

The following shows an example of how to connect 512 Mb of SDRAM (256M words SDRAM (4M words × 16 bits × 4 banks) × 2) for a 32-bit data bus width.

- SDCRn register settings

  SSO1, SSO0 = 10: Data bus width = 32 bits
  RAW1, RAW0 = 10: Row address width = 13 bits
  SAW1, SAW0 = 01: Column address width = 9 bits

- Physical address

  A25, A24: Bank address
  A23 to A11: Row address
  A10 to A2: Column address

- Address output from memory controller

  A25, A24: Bank address
  A14 to A2: Row address (13 bits), column address (9 bits)

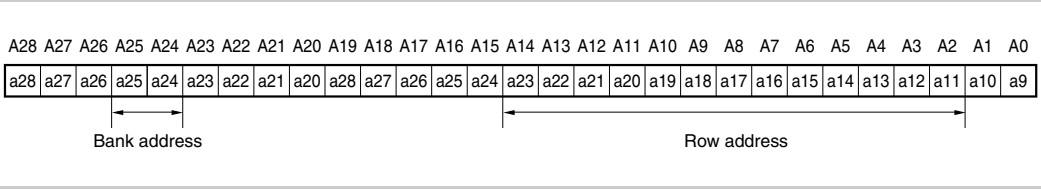


Figure 11-27 Row address, bank address output when active command issued (32 bits)

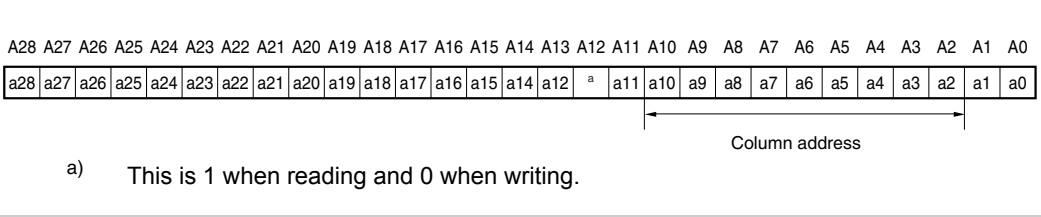


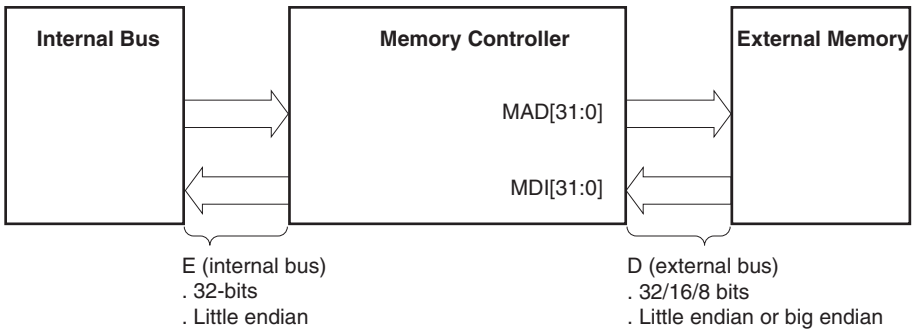
Figure 11-28 Column address output when read/write command issued (32 bits)

- Connection of memory controller and SDRAM

  A23, A22 (memory controller): BA0 (A14), BA1 (A13) (SDRAM)
  A13 to A2 (memory controller): A12 to A0 (SDRAM)

11.9 Data Flow

The data transfer flow to external memory differs according to factors such as the data width, endian specification, external bus width, and start address.



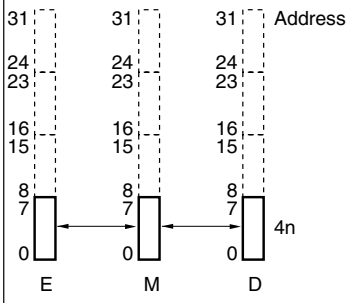
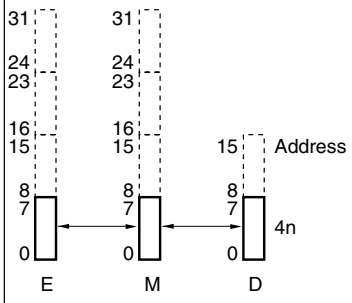
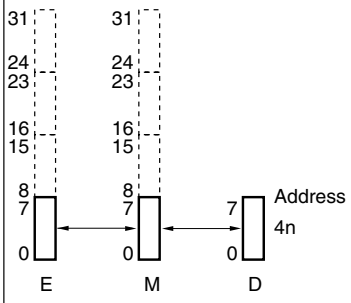
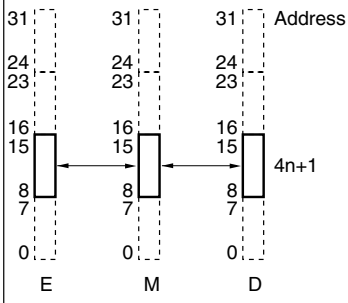
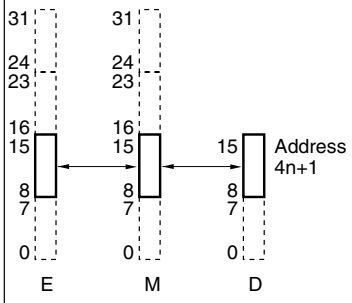
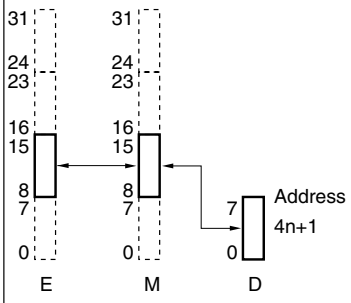
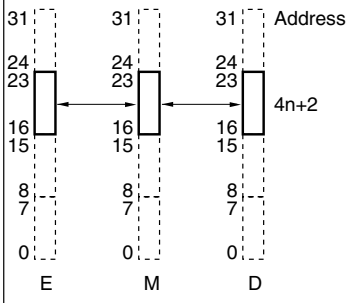
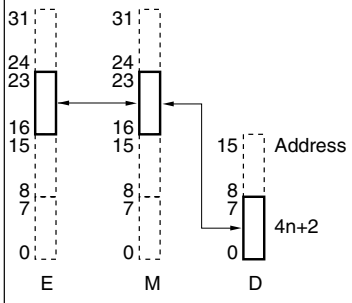
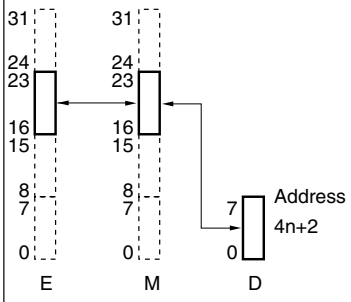
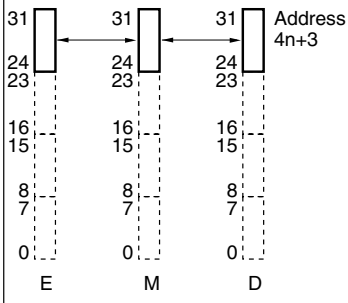
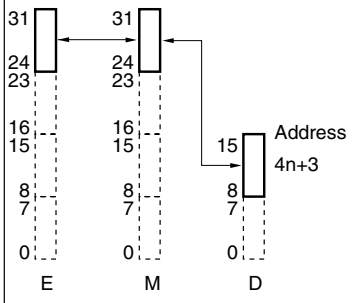
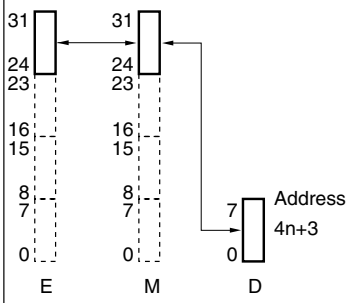
Note B: Byte data
HW: Halfword data
W: Word data

Figure 11-29 Internal bus, memory controller, external bus data flow

The data flows for various conditions are shown on the following pages.

11.9.1 Data flow during byte access

Table 11-18 Data flow during byte access (little endian)

Accessed Address	Data Transfer Flow		
	External Data Bus: 32 Bits	External Data Bus: 16 Bits	External Data Bus: 8 Bits
4n			
4n+1			
4n+2			
4n+3			

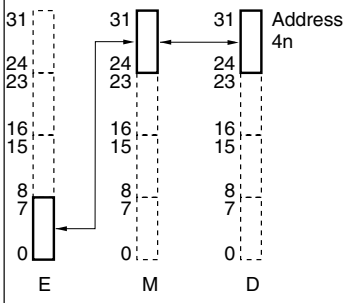
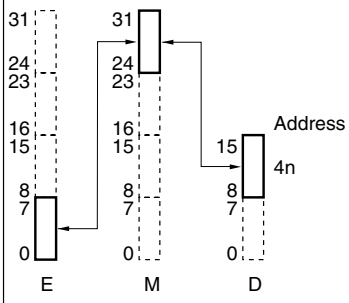
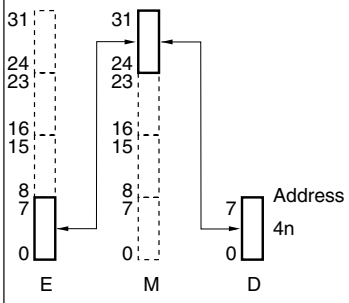
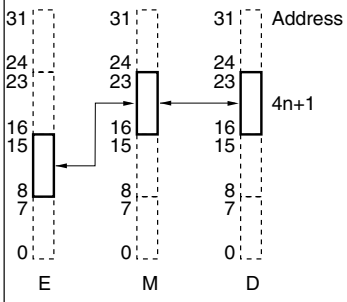
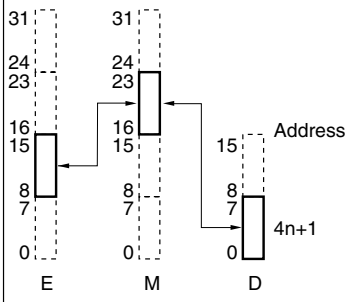
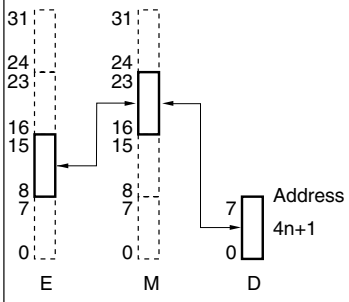
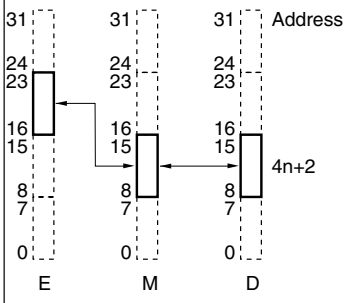
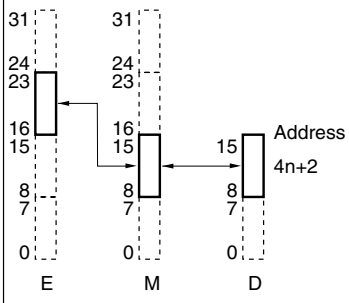
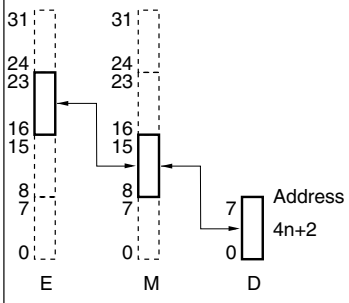
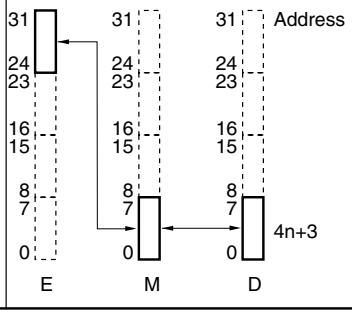
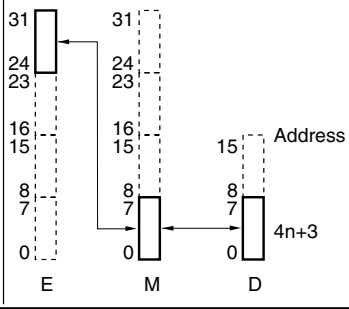
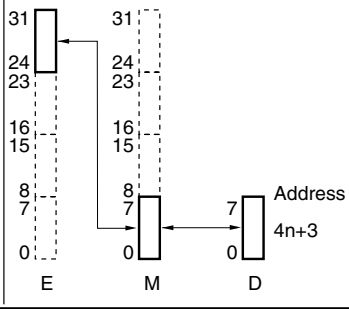
Note E:Internal bus

M: MEMC data buffer

D: External data bus

n = 0, 1, 2, 3, …

Table 11-19 Data flow during byte access (big endian)

Accessed Address	Data Transfer Flow		
	External Data Bus: 32 Bits	External Data Bus: 16 Bits	External Data Bus: 8 Bits
4n			
4n+1			
4n+2			
4n+3			

Note E:Internal bus

M: MEMC data buffer

D: External data bus

n = 0, 1, 2, 3, …

Table 11-20 Data flow during halfword access (little endian) (1/2)

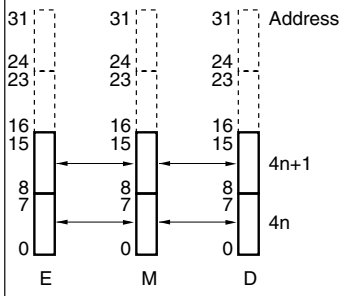
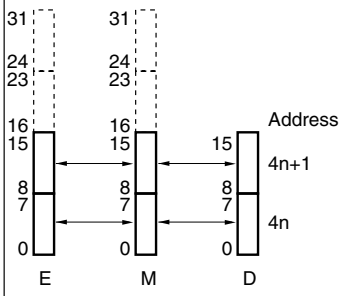
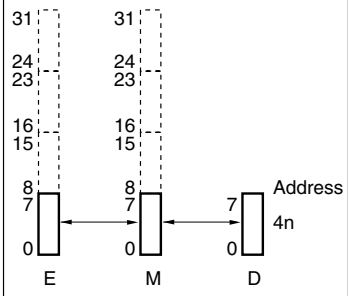
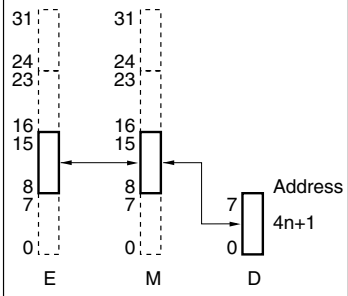
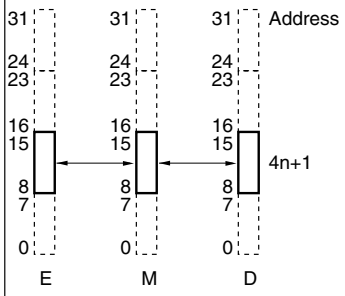
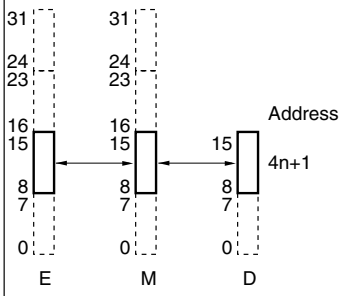
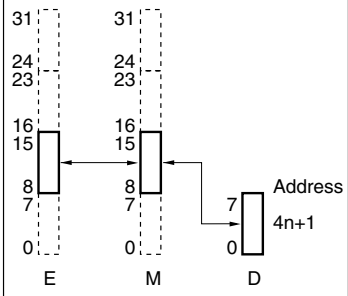
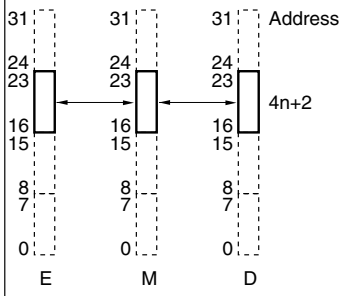
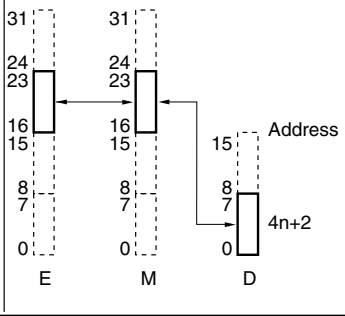
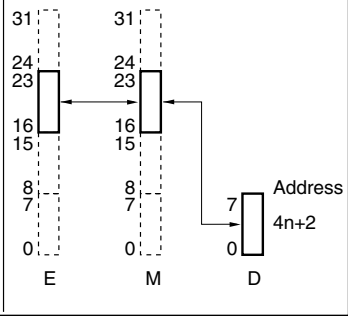
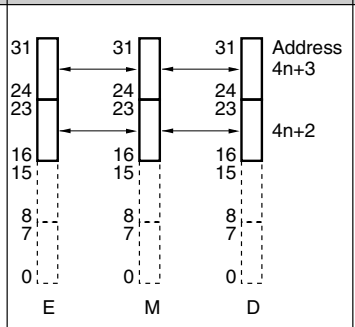
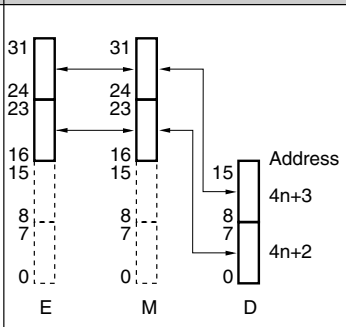
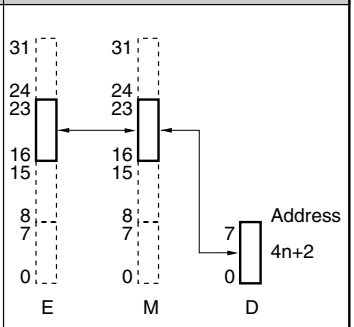
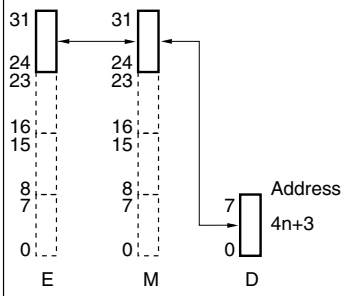
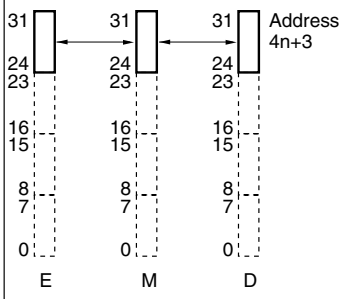
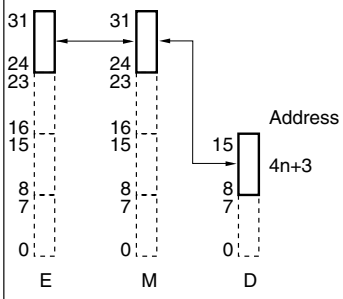
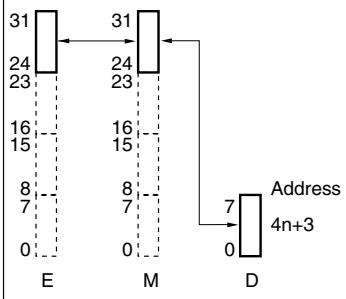
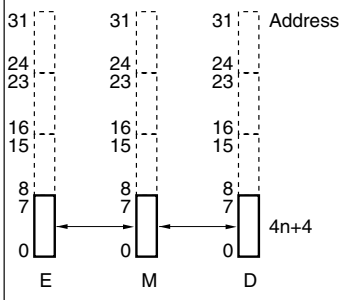
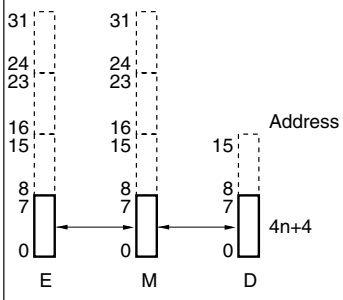
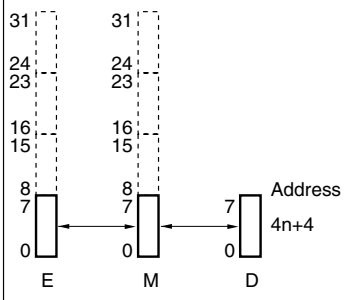
Accessed Address		Data Transfer Flow		
		External Data Bus: 32 Bits	External Data Bus: 16 Bits	External Data Bus: 8 Bits
4n	1st			
	2nd	—	—	
4n+1	1st			
	2nd			

Table 11-20 Data flow during halfword access (little endian) (2/2)

Accessed Address		Data Transfer Flow		
		External Data Bus: 32 Bits	External Data Bus: 16 Bits	External Data Bus: 8 Bits
4n+2	1st			
	2nd	—	—	
4n+3	1st			
	2nd			

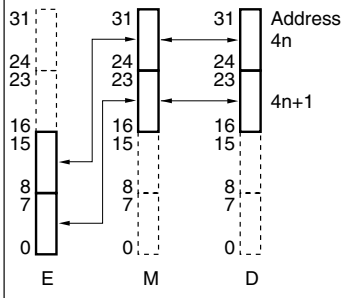
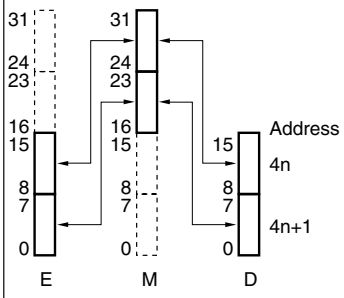
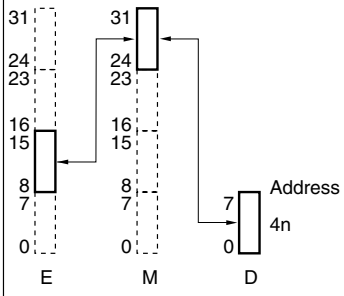
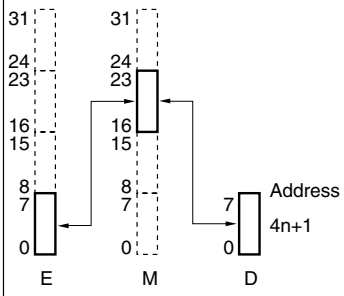
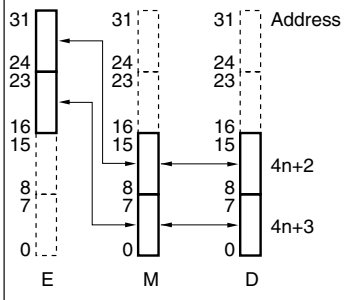
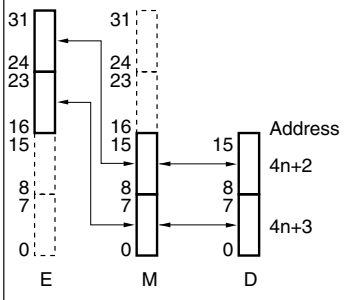
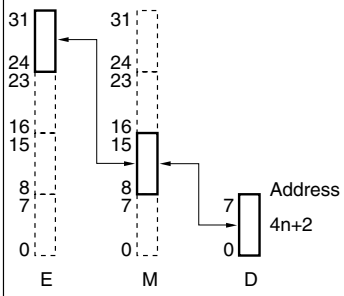
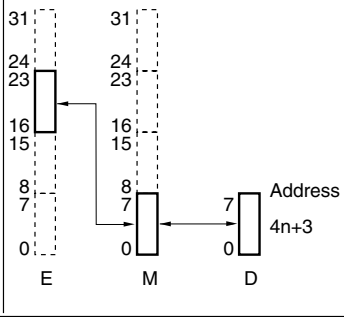
Note E:Internal bus

M: MEMC data buffer

D: External data bus

n = 0, 1, 2, 3, …

Table 11-21 Data flow during halfword access (big endian)

Accessed Address		Data Transfer Flow		
		External Data Bus: 32 Bits	External Data Bus: 16 Bits	External Data Bus: 8 Bits
4n	1st			
	2nd	———	———	
4n+2	1st			
	2nd	———	———	

Notes 1. E:Internal bus

M:MEMC data buffer

D:External data bus

n = 0, 1, 2, 3, …

2. Accesses with the start address starting from 4n+1 or 4n+3 are prohibited.

Table 11-22 Data flow during word access (little endian) (1/4)

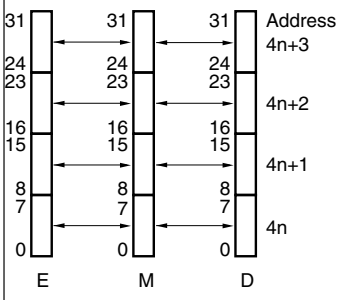
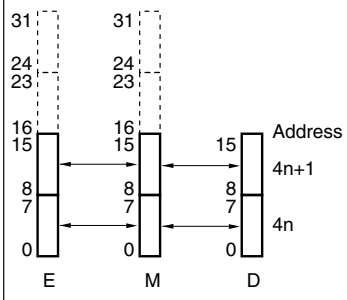
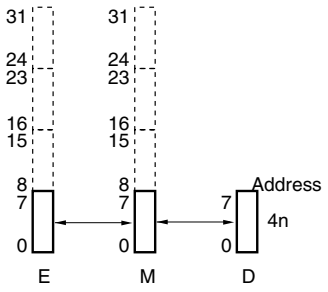
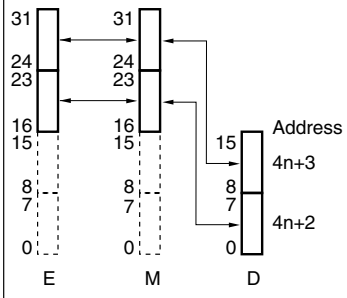
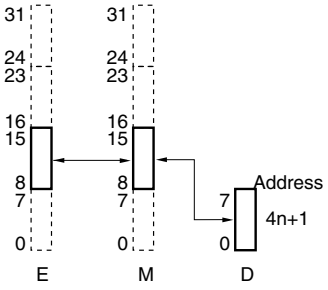
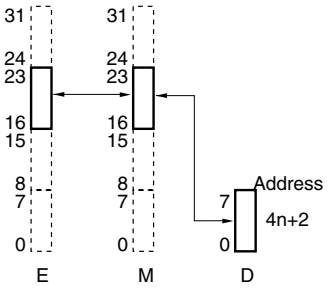
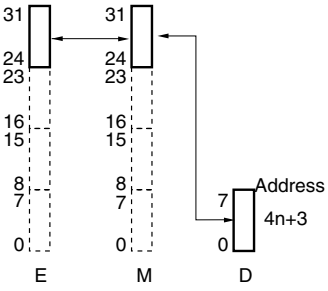
Accessed Address		Data Transfer Flow		
		External Data Bus: 32 Bits	External Data Bus: 16 Bits	External Data Bus: 8 Bits
4n	1st			
	2nd	—		
	3rd	—	—	
	4th	—	—	

Table 11-22 Data flow during word access (little endian) (2/4)

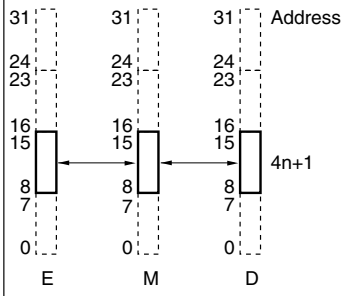
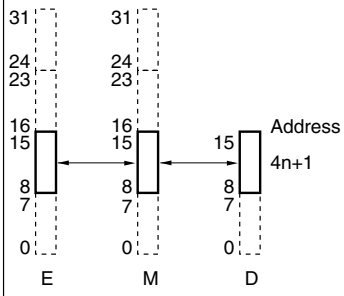
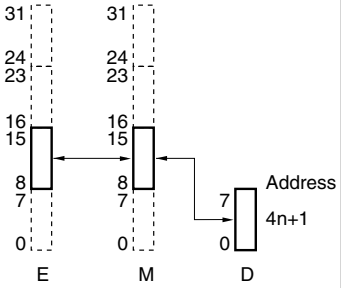
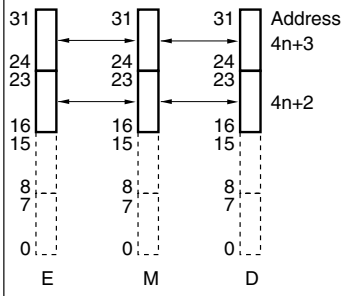
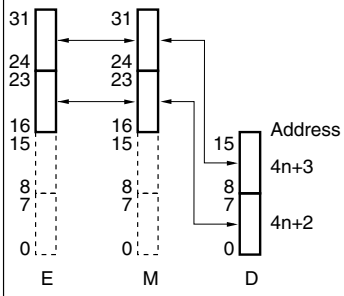
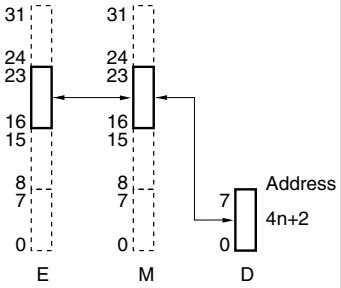
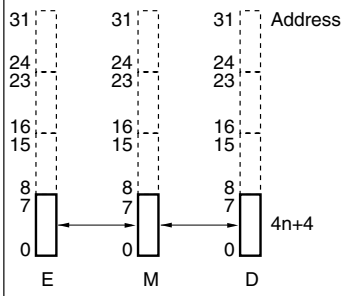
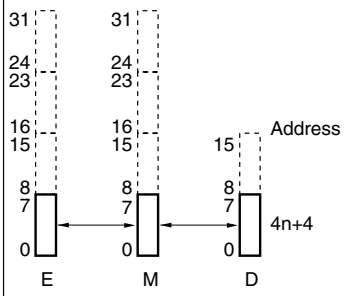
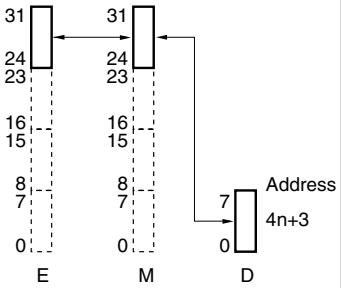
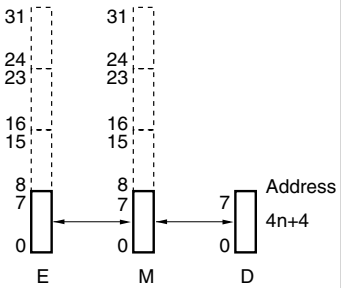
Accessed Address		Data Transfer Flow		
		External Data Bus: 32 Bits	External Data Bus: 16 Bits	External Data Bus: 8 Bits
4n+1	1st			
	2nd			
	3rd			
	4th	———	———	

Table 11-22 Data flow during word access (little endian) (3/4)

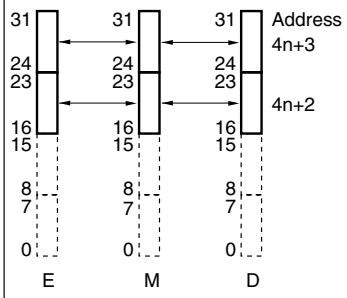
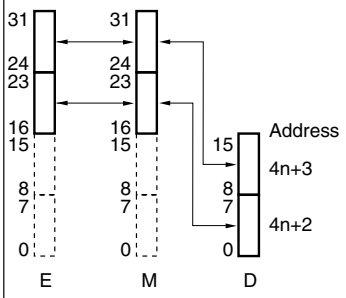
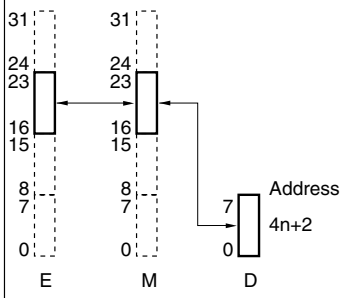
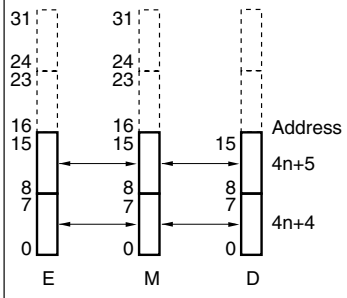
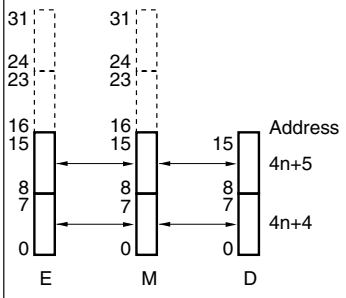
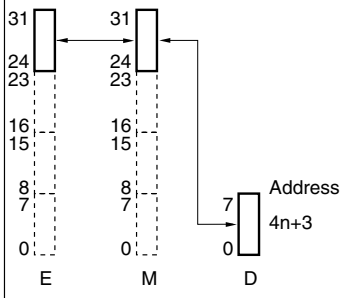
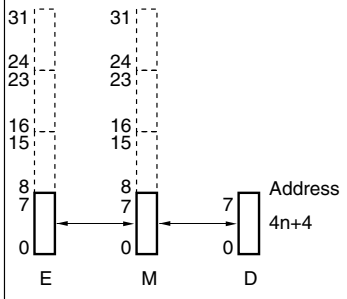
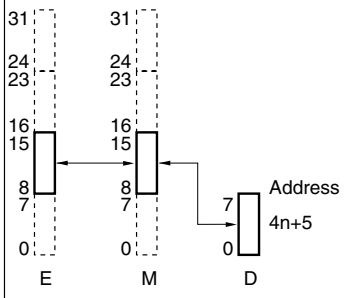
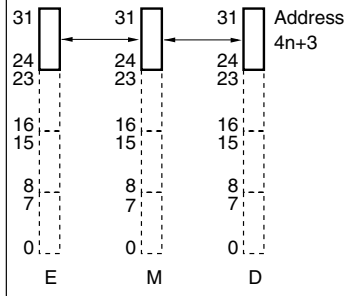
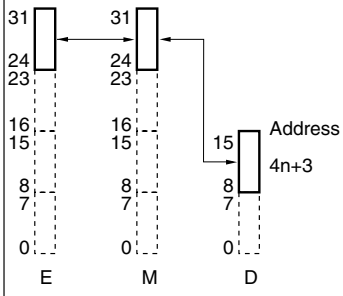
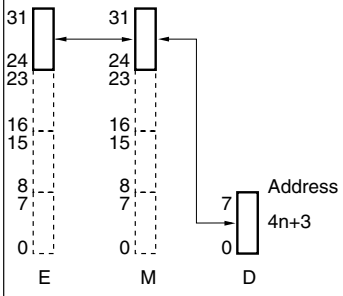
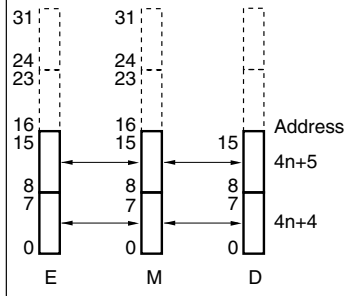
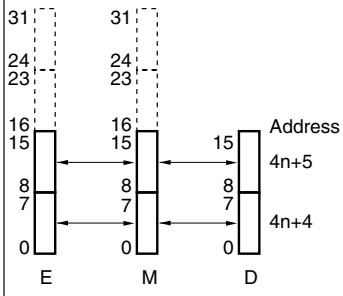
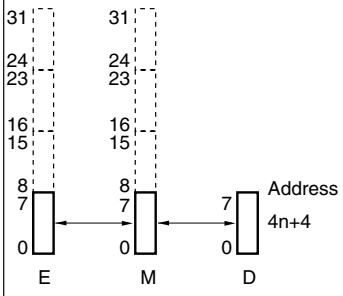
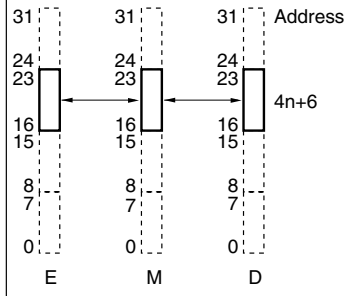
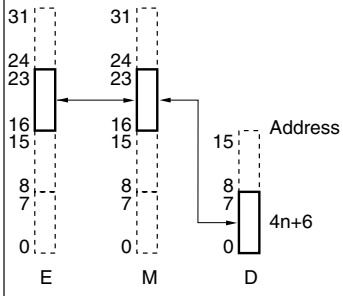
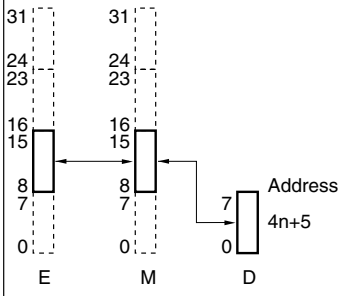
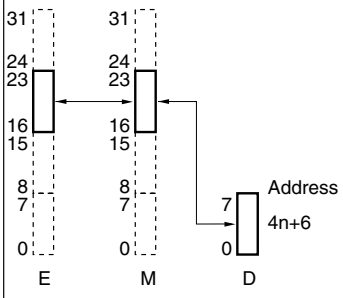
Accessed Address		Data Transfer Flow		
		External Data Bus: 32 Bits	External Data Bus: 16 Bits	External Data Bus: 8 Bits
4n+2	1st			
	2nd			
	3rd	—	—	
	4th	—	—	

Table 11-22 Data flow during word access (little endian) (4/4)

Accessed Address		Data Transfer Flow		
		External Data Bus: 32 Bits	External Data Bus: 16 Bits	External Data Bus: 8 Bits
4n+3	1st			
	2nd			
	3rd			
	4th	———	———	

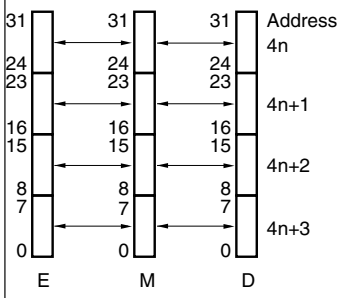
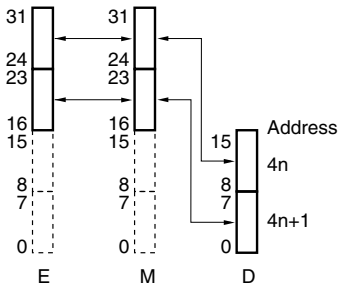
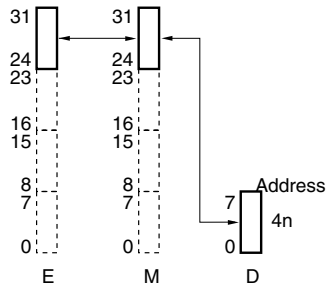
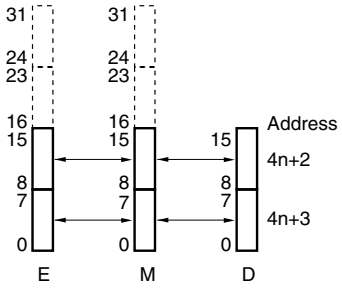
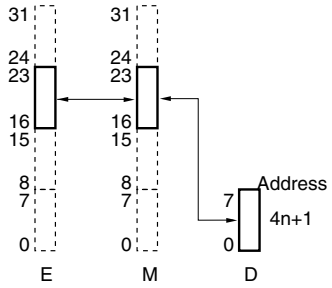
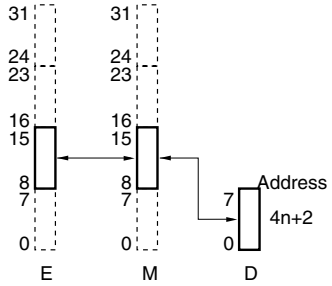
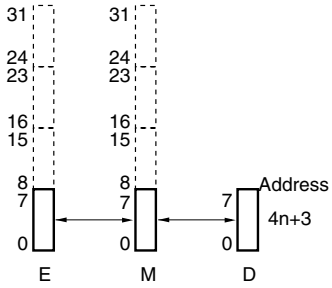
Note E:Internal bus

M: MEMC data buffer

D: External data bus

n = 0, 1, 2, 3, …

Table 11-23 Data flow during word access (big endian)

Accessed Address		Data Transfer Flow		
		External Data Bus: 32 Bits	External Data Bus: 16 Bits	External Data Bus: 8 Bits
4n	1st			
	2nd	———		
	3rd	———	———	
	4th	———	———	

Note 1. E:Internal bus
M:MEMC data buffer
D:External data bus
n = 0, 1, 2, 3, …

2. Accesses with the start address starting from 4n+1 or 4n+3 are prohibited.

Chapter 12 Timer Array Unit A (TAUA)

This chapter describes the Timer Array Unit A (TAUA).

The first section describes all V850E2/MN4 specific properties, such as instances, register base addresses, input/output signal names, etc.
The subsequent sections describe the features that apply to all implementations.

12.1 V850E2/MN4 TAUA Features

Instances

This microcontroller has following number of instances of TAUA.

Table 12-1 Instances of TAUA

TAUA	
Instance	4
Name	TAUA0 to TAUA3

Instances index n

Throughout this chapter, the individual instances of a TAUA is identified by the index "n" (n = 0 to 3), for example, TAUAnTOM for the TAUAn channel output mode register.

Channel index m

TAUA has 16 channels. Throughout this chapter, the individual channels are identified by the index “m” (m = 0 to 15), thus a certain channel is denoted as CHm.
The even numbered channels (m = 0, 2, 4, 6, 8, 10, 12, 14) are denoted as CHm_even.
The odd numbered channels (m = 1, 3, 5, 7, 9, 11, 13, 15) are denoted as CHm_odd.

Register addresses

All TAUAn register addresses are given as address offsets to the individual base address <TAUAn_base_OS> or <TAUAn_base_USER>.
The base addresses <TAUAn_base_OS> and <TAUAn_base_USER> of each TAUAn are listed in the following table:

Table 12-2 Register base addresses <TAUAn_base>

TAUAn instance	Base address	Address
TAUA0	<TAUAn_base_USER>	FFFF C400$_H$
	<TAUAn_base_OS>	FF80 8000$_H$
TAUA1	<TAUAn_base_USER>	FFFF C800$_H$
	<TAUAn_base_OS>	FF80 9000$_H$
TAUA2	<TAUAn_base_USER>	FFFF CC00$_H$
	<TAUAn_base_OS>	FF80 A000$_H$
TAUA3	<TAUAn_base_USER>	FFFF D000$_H$
	<TAUAn_base_OS>	FF80 B000$_H$

Clock supply TAUA provides one clock input:

Table 12-3 TAUAn clock supply

TAUAn instance	TAUAn clock	Connected to
TAUA0	PCLK	f_{PCLK}
TAUA1	PCLK	f_{PCLK}
TAUA2	PCLK	f_{PCLK}
TAUA3	PCLK	f_{PCLK}

Interrupts and DMA/DTS TAUA can generate the following interrupt and DMA/DTS requests:

Table 12-4 TAUAn interrupt and DMA/DTS requests

TAUAn signals	Function	Connected to
TAUA0:		
INTTAUA0I[0:15]	Channel 0 to 15 interrupt	Interrupt Controller INTTAUA0I[0:15] DMA Controller trigger 32 to 47 DTS Controller trigger 32 to 47 PIC: INTTAUA0I[15:0]
TAUA1:		
INTTAUA1I[0:15]	Channel 0 to 15 interrupt	Interrupt Controller INTTAUA1I[0:15] DMA Controller trigger 48 to 63 DTS Controller trigger 48 to 63 PIC: INTTAUA1I[15:0]
TAUA2:		
INTTAUA2I[0:11]	Channel 0 to 11 interrupt	Interrupt Controller INTTAUA2I[0:11]
INTTAUA2I[12:15]	Channel 12 to 15 interrupt	Interrupt Controller INTTAUA2I[12:15] DMA Controller trigger 64 to 67 DTS Controller trigger 64 to 67
TAUA3:		
INTTAUA3I[0:15]	Channel 0 to 15 interrupt	Interrupt Controller INTTAUA3I[0:15] DMA Controller trigger 68 to 83 DTS Controller trigger 68 to 83

I/O signals The I/O signals of TAUA are listed in the following table.

Table 12-5 TAUAn I/O signals

TAUA signal	Function	Connected to
TAUA0:		
TAUA0TTIN[0:15]	Channel [0:15] input	Port TA0_I[0:15]
TAUA0TTOUT[0:15]	Channel [0:15] output	Port TA0_O[0:15] PIC (TAUA0UDC[8, 2, 0])
TAUA0TSST[0:15]	Channel [0:15] start trigger input	PIC0 TAUA0TSST[0:15]
TAUA1:		
TAUA1TTIN[0:15]	Channel [0:15] input	Port TA1_I[0:15]
TAUA1TTOUT[0:15]	Channel [0:15] output	Port TA1_O[0:15] PIC (TAUA1UDC[8, 2, 0])
TAUA1TSST[0:15]	Channel [0:15] start trigger input	PIC0 TAUA1TSST[0:15]
TAUA2:		
TAUA2TTIN[0:15]	Channel [0:15] input	Port TA2_I[0:15]
TAUA2TTOUT[0:15]	Channel [0:15] output	Port TA2_O[0:15]
TAUA2TSST[12:15]	Channel [0:15] start trigger input	PIC (TAUJ1TSST[0:3])
TAUA3:		
TAUA3TTIN[0:15]	Channel [0:15] input	Port TA3_I[0:15]
TAUA3TTOUT[0:15]	Channel [0:15] output	Port TA3_O[0:15]

All TAUAn interrupt and I/O signals are sketched in the following figure.

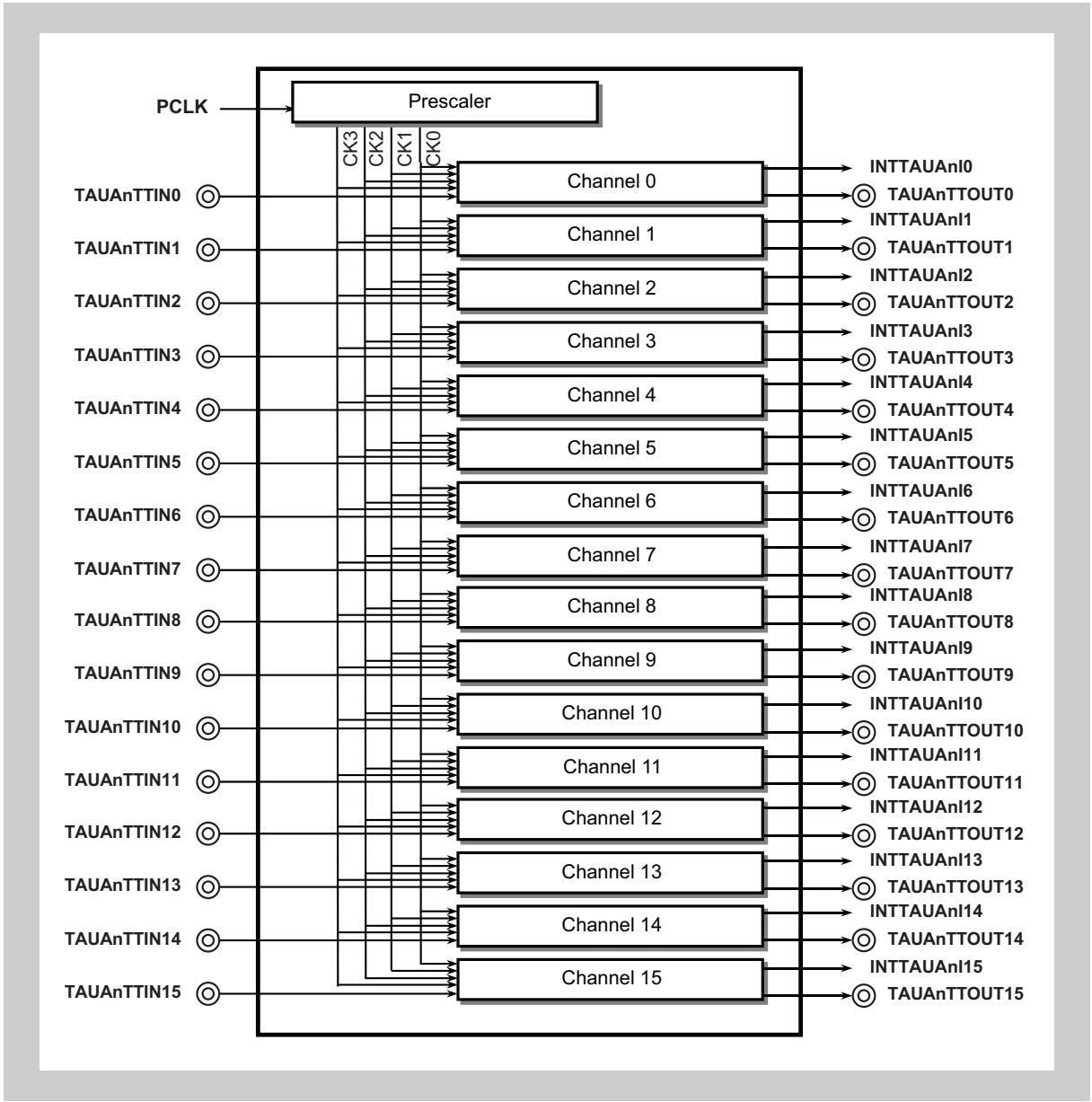


Figure 12-1 TAUA I/O and interrupt signals

12.2 Functional Overview

Features summary

The TAUA has the following functions:

- 16 channels
- 16-bit counter and 16-bit data register per channel
- Independent channel operation
- Synchronous channel operation (master and slave operation)
- Generation of different types of output signal
- Real-time output
- Counter can be triggered by external signal
- Interrupt generation

The following figure shows the main components of the TAUA:

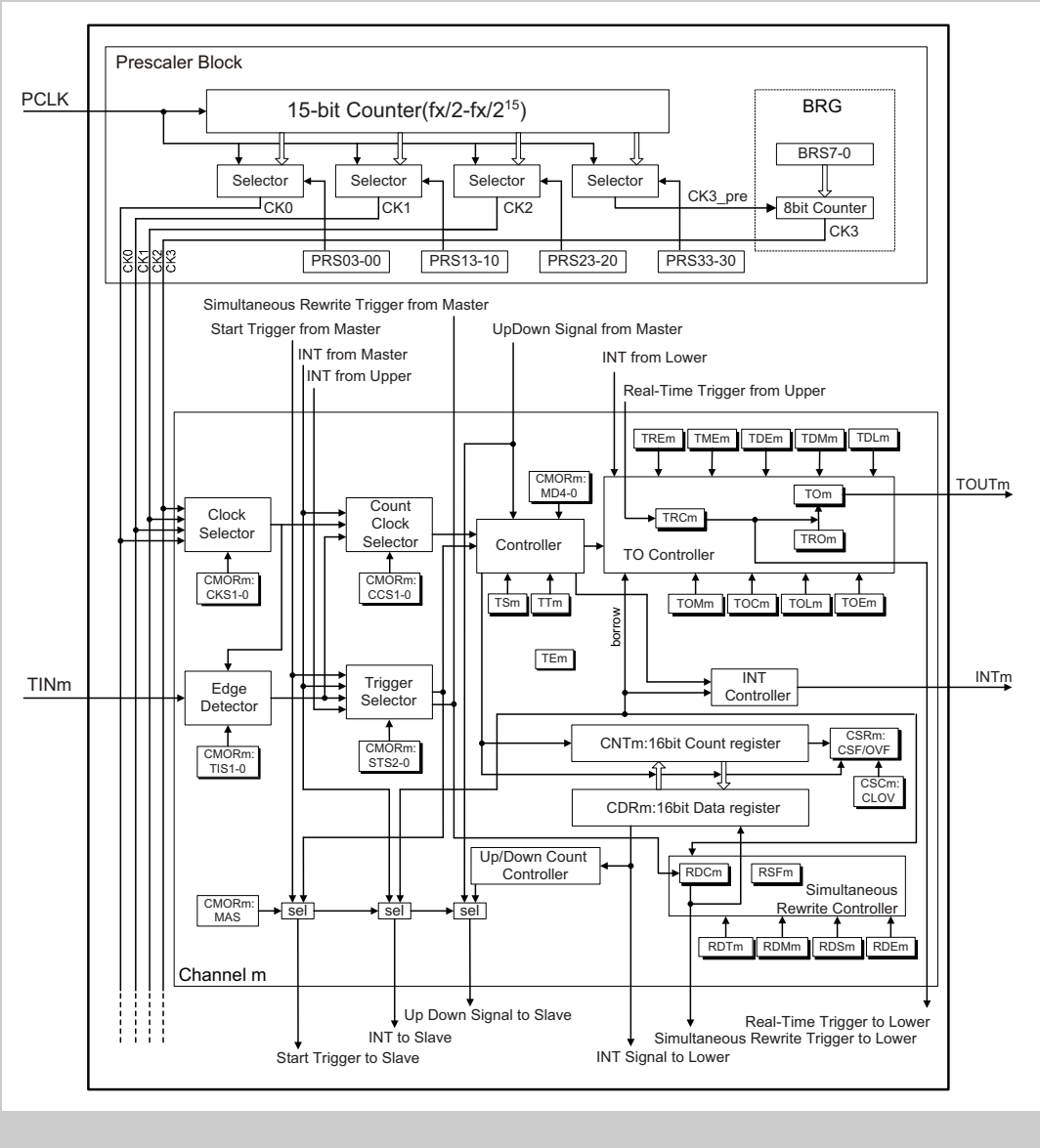


Figure 12-2 Block diagram of the TAUA

The prefix “TAUAn” has been omitted from the register names for the sake of clarity in the above figure.

12.2.1 Terms

In this chapter, the following terms are used:

- **Independent / synchronous channel operation**

  Independent or synchronous channel operation describes the dependency of channels on each other:

  – If a channel operates independent of all other channels, this is called independent channel operation.

  – If a channel operates depending on other channels, this is called synchronous channel operation.

- **Channel group**

  In synchronous channel operation, all channels that depend on each other are referred to as a “channel group”.

  A channel group has one master channel and one or more slave channels.

- **Operation mode**

  An operation mode can be selected for every channel m. The operation mode defines the *basic* operation and features of a channel.

  In synchronous channel operation, every channel in the channel group can operate in a different operation mode.

  Examples are “Capture Mode”, “Event Count Mode”, and “Interval Timer Mode”.

- **Channel output mode**

  The channel output mode defines the operation of TAUAnTTOUTm

  – of a single channel (independent output operation) or

  – of all channels in a channel group (synchronous output operation).

  Examples are “Independent Channel Output Mode 1” and “Synchronous Channel Output Mode 2 with Dead Time Output”.

- **Channel operation function**

  The channel operation function defines the *complete* function and all features

  – of a single channel (independent channel operation) or

  – of all channels in a channel group (synchronous channel operation).

- **Upper / lower channel**

  Depending on the channel number m, a neighboring channel can be referred to as “upper” or “lower” channel:

  – Upper channel: Channel with a smaller channel number

  – Lower channel: Channel with a higher channel number

  Example:
  For channel 5, channel 3 is an upper channel and channel 9 is a lower channel.

12.3 Functional Description

TAUA is used to perform various count or timer operations and to output a signal which depends on the result of the operation. It contains one prescaler for count clock generation and 16 channels, each equipped with a 16 bit counter TAUAnCNTm and a 16-bit data register TAUAnCDRm to hold the start or compare value of the counter.

It also contains several control and status registers.

Independent and synchronous operation

Every channel can operate in different operation modes, either independently or in combination with other channels (synchronously). If a channel group has one master channel and one or more slave channels, the slave channels depend on the master channel.

When a channel is operated independently, its operation mode and functions are not affected by those of other channels. When a channel is operated synchronously it is either a master or a slave. A master channel can have multiple slaves, and the state of one channel affects that of the other channels. For example, this means that one channel can control when another starts to count, is reset, etc.

The following describes the functional blocks:

Prescaler

The prescaler provides up to 4 clock signals (CK0 to CK3) that can be used as count clocks for all channels.

For count clocks CK0 to CK2, a clock obtained by dividing PCLK by 2^0 to 2^{15} using the prescaler can be selected. For the fourth count clock CK3, a division factor that is not a power of 2 can be specified by using BRG.

Clock and count clock selection

For every channel, the count clock selector selects which of the following is used as the clock source:

- One of the clocks CK0 to CK3 (selected by the clock selector)
- INTTAUAnIm from master channel
- TAUAnTTINm input signal valid edge

Controller

The controller controls the main operations of the counter:

- Operation mode (selected by bits TAUAnCMORm.TAUAnMD[4:0])
- Counter start enable (TAUAnTS.TAUAnTSm) and counter stop (TAUAnTT.TAUAnTTm)

  When counter start is enabled, status flag TAUAnTE.TAUAnTEm is set.
- Count direction (up/down) (can be controlled by master channel)

Trigger selector

Depending on the selected operation mode, the counter starts automatically when it is enabled (TAUAnTE.TAUAnTEm = 1), or it waits for an external start trigger signal. Any of the following signals can be used as the start trigger:

- Synchronous channel start trigger input TAUAnTSSTm
- TAUAnTTINm input valid edge
- INTTAUAnIm from the master or any upper channel
- Up/down output trigger signal of the master channel
- Dead-time output signal of the TAUAnTTOUTm generation unit.

Simultaneous rewrite controller

Simultaneous rewrite control is a function that can be used in synchronous operation modes. The data registers (TAUAnCDRm) of all channels in a

channel group can be rewritten at any time. The simultaneous rewrite controller ensures that new data register values of all channels become effective at the same time.

TAUAnTO Controller The output control of every channel enables the generation of various output signal forms such as PWM signals or triangular waves.

12.3.1 Timer operation functions

The functions below can be achieved by operating TAUA independently on each channel or by operating it on a combination of multiple channels.

Table 12-6 TAUA Operation Functions (1/2)

Independent Channel Operation Function	Synchronous Channel Operation Function
Independent channel operation functions	Synchronous channel operation functions
Interval timer function	PWM output function
TAUAnTTINm input interval timer function	Trigger start PWM output function
Delay count function	Delay pulse output function
One-pulse output function	A/D conversion trigger output function type 1
Independent channel signal measurement functions	Synchronous PWM signal functions triggered by an external signal
TAUAnTTINm input pulse interval measurement function	One-shot pulse output function
TAUAnTTINm input signal width measurement function	Offset trigger output function
Overflow interrupt output function (during TAUAnTTINm width measurement)	Synchronous Triangle PWM output function
TAUAnTTINm input period count detection function	Triangle PWM output function
Overflow interrupt output function (during TAUAnTTINm input period count detection)	Triangle PWM output function with dead time
TAUAnTTINm input pulse interval judgment function	A/D conversion trigger output function type 2
TAUAnTTINm input signal width judgment function	Synchronous non-complementary modulation output function and synchronous complementary modulation output function
Independent channel real-time functions	Non-complementary modulation output function type 1
Real-time output function type 1	Non-complementary modulation output function type 2
Real-time output function type 2	Complementary modulation output function
Independent channel simultaneous rewrite functions	Other independent channel functions
Simultaneous rewrite trigger generation function type 1	Interrupt culling function
Simultaneous rewrite trigger generation function type 2	
Independent channel one-phase PWM function	
One-phase PWM output function	
Other independent channel functions	

Table 12-6 TAUA Operation Functions (2/2)

Independent Channel Operation Function	Synchronous Channel Operation Function
External event count function	
Clock divide function	
TAUAnTTINm input position detection function	

12.4 General Operating Procedure

The following lists the general operation procedure for the TAUAn:

After reset release, the operation of each channel is stopped. Clock supply is started and writing to each register is enabled. The control register of TAUAnTTOUTm is also initialized and outputs a low level.

1. Set the TAUAnTPS and TAUAnBRS registers to specify the clock frequency of CK0 to CK3.
2. Configure the desired TAUAn function:
   – Set the operation mode
   – Set the channel output mode
   – Set any other control bits
3. Enable the counter by setting the TAUAnTS.TAUAnTSm bit to 1.

   The counter starts to count immediately, or when an appropriate trigger is detected, depending on the bit settings.
4. During counting, if desired, and if possible for the configured function, stop the counter or perform a forced restart operation.
5. Stop the function by setting the TAUAnTT.TAUAnTTm bit to 1.

Note A detailed description of the required control bits and the operation of the individual functions is given in *12.14 “Independent Channel Operation Functions” on page 597* and *12.21 “Synchronous Channel Operation Functions” on page 719*.

12.5 Operation Modes

The TAUA contains 12 operation modes.

One operation mode can be set for each channel. It is specified using the TAUAnCMOR.TAUAnMD[4:0] bits.

Note For registers and bits, some values are fixed according to the operation function, and others are selected by the user.

For details about the settings for registers and bits, refer to the chapters describing each operation function.

12.6 Concepts of Synchronous Channel Operation

In synchronous channel operation, multiple channels depend on each other, or are affected by changes in another channel. Therefore, several rules apply for the use of synchronous channel functions. These rules are detailed in *12.6.1 "Rules"*.

Two special features for synchronous channel operation are detailed in the following subchapters:

- *12.6.2 "Simultaneous start and stop of synchronous channel counters" on page 564*
- *12.7 "Simultaneous Rewrite" on page 565*

12.6.1 Rules

Number of masters and slaves

- Only even channels (CH0, CH2, CH4, …) can be set as master channels. Any channel except CH0 can be set as a slave channel.
- Only channels lower than the master channel can be set as slave channels, and several slave channels can be set for one master channel.
  Example: If CH2 is a master channel, CH3 and the lower channels (CH4, CH5, …) can be set as slave channels.
- If multiple master channels are used, slave channels cannot cross the master channels.
  Example: If CH0 and CH4 are master channels, CH1 to CH3 can be set as slave channels for CH0, but CH5 to CH15 cannot.

Operation clock

- The same operation clock must be set for the corresponding slave channel and the master channel. Specify the same value for the TAUAnCMORm.TAUAnCKS[1:0] bits of the synchronized master and slave channels.

The basic concepts of master/slave usage and operation clocks are illustrated in the following figure.

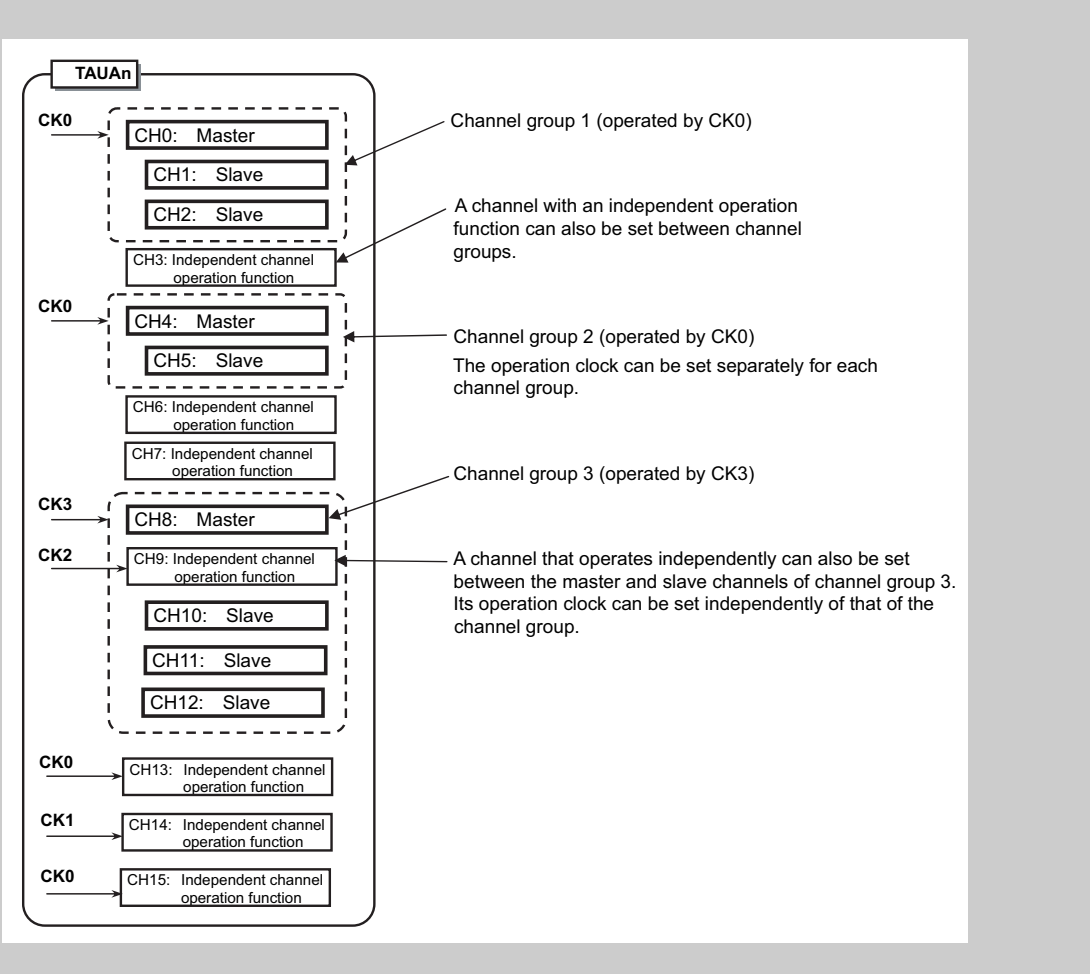


Figure 12-3 Grouping of the channels and assignment of operation clocks

Control trigger signals of the master and slave channels

- A master channel can output a control trigger signal to its slave channels.
- Slave channels can use the control trigger signal of the master channel but cannot transfer their control trigger signals to the lower channels.
- A master channel cannot use the control trigger signal of an upper master channel.

12.6.2 Simultaneous start and stop of synchronous channel counters

Channels that are operated synchronously can be started and stopped simultaneously, both within a unit, and between units.

(1) Simultaneous start and stop within a unit

- To simultaneously start synchronized channels, the TAUAnTS.TAUAnTSm bits of the channels must be set at the same time.
- To simultaneously stop synchronized channels, the TAUAnTT.TAUAnTTm bits of the channels must be set at the same time.

Setting a TAUAnTS.TAUAnTSm bit to 1 sets the corresponding TAUAnTE.TAUAnTEm bit to 1, enabling counting. When the counter starts counting depends on the operation mode.

12.7 Simultaneous Rewrite

12.7.1 Introduction

Simultaneous rewrite describes the ability to change the compare/start value and the output logic of multiple channels at the same time.

The corresponding data and control registers (TAUAnCDRm and TAUAnTOLm) can nevertheless be written at any time. The new value does not affect the counter operation or the output signal until simultaneous rewrite is triggered.

Simultaneous rewrite can be triggered by:

- The counter on the master channel or upper channel (depending on the selected operation mode) reaching a certain value
- INTTAUAnIm being issued on the upper channel specified by TAUAnRDC.TAUAnRDCm

There are four methods for simultaneous rewrite. These are listed in the following table, along with how to specify them and when they cause simultaneous rewrite to be triggered.

Table 12-7 Simultaneous rewrite methods and when they are triggered

Method	Simultaneous rewrite triggered when	TAUAn RDE. TAUAn RDEm	TAUAn RDS. TAUAn RDSm	TAUAn RDM. TAUAn RDMm
-	No simultaneous rewrite	0	0	0
A	The master channel (re)starts counting	1	0	0
B	The slave channel starts counting down at the upper peak of a triangular cycle	1	0	1
C1	INTTAUAnIm is generated on an upper channel specified by TAUAnRDC.TAUAnRDCm	1	1	0
C2	INTTAUAnIm is generated on an upper channel specified by TAUAnRDC.TAUAnRDCm that in turn is triggered by an external signal	1	1	0

The following table lists which of these four methods is available for each channel operation function. For more information about the individual channel operation functions, see the corresponding sections in *12.14 “Independent Channel Operation Functions” on page 597* and *12.21 “Synchronous Channel Operation Functions” on page 719*.

Table 12-8 Channel operation functions and methods they use

Function	-	A	B	C1	C2
Simultaneous Rewrite Trigger Output Function Type 1				X	
PWM Output Function		X		X	
One-Shot Pulse Output Function		X			
Trigger Start PWM Output Function		X			X
Offset Trigger Output Function	X				
Delay Pulse Output Function		X			
Triangle PWM Output Function			X	X	
Triangle PWM Output Function with Dead Time			X	X	
Interrupt Culling Function		X	X	X	
AD Conversion Trigger Output Function Type 1		X		X	
AD Conversion Trigger Output Function Type 2			X	X	
Non-Complementary Modulation Output Function Type 1		X		X	
Non-Complementary Modulation Output Function Type 2			X	X	
Complementary Modulation Output Function			X	X	

12.7.2 How to control simultaneous rewrite

The following figure shows the general procedure for simultaneous rewrite. The three main blocks (Initial setup, Start counter & count operation, and Simultaneous rewrite) are explained afterwards.

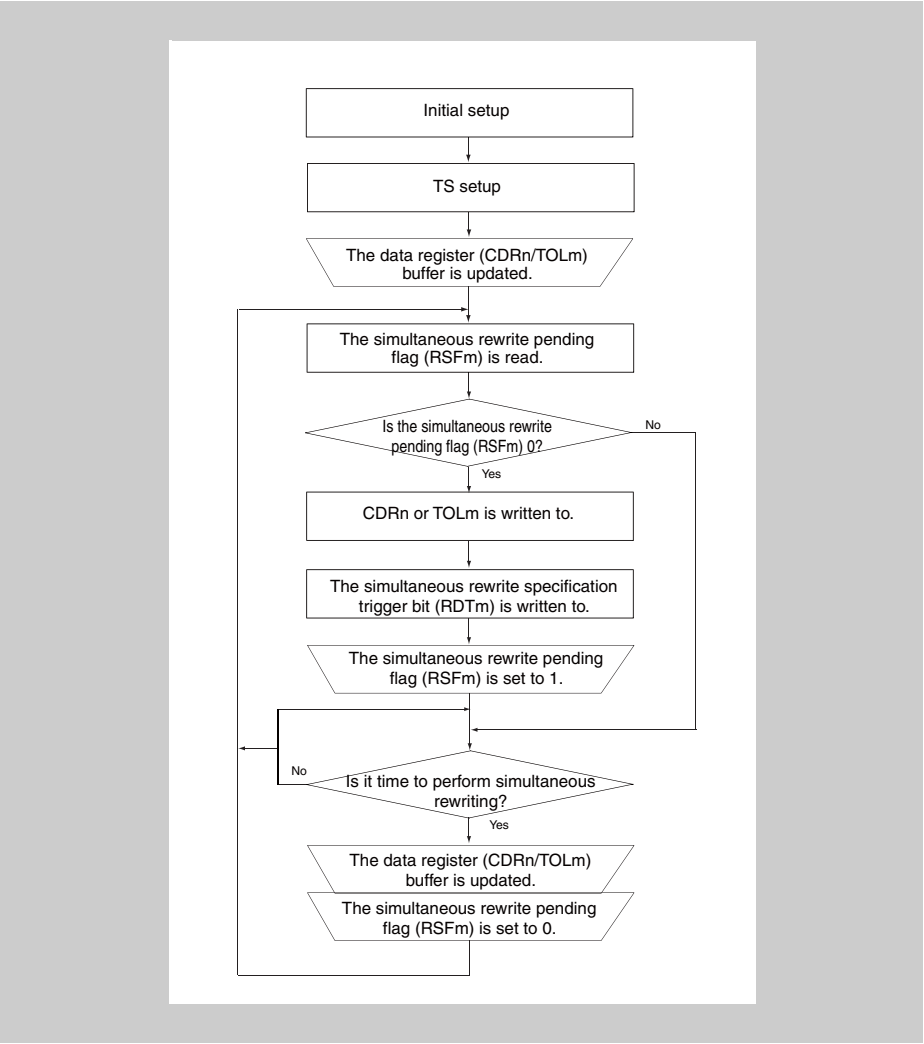


Figure 12-4 General procedure for simultaneous rewrite

(1) Initial settings

- To enable simultaneous rewrite in channel m, set TAUAnRDE.TAUAnRDEm = 1
- To select the type of simultaneous rewrite, set TAUAnRDM.TAUAnRDMm and TAUAnRDS.TAUAnRDSm according to the values in *Table 12-7 "Simultaneous rewrite methods and when they are triggered" on page 565*
- Specify the simultaneous rewrite trigger generation channel for TAUAnRDC.TAUAnRDCm. (Prerequisite: TAUAnRDS.TAUAnRDSm is specified as an upper channel.)

(2) Start counter and count operation

- To start all the TAUAnCNTm counters in the channel group, set the corresponding TAUAnTS.TAUAnTSm bits to 1. TAUAnTOL.TAUAnTOLm and the values in the data registers (TAUAnCDRm) are loaded to the corresponding TAUAnTOL.TAUAnTOLm buffer (TAUAnTOL.TAUAnTOLm buf) and data buffer registers (TAUAnCDRm buf) and the counters start.
- Setting the reload data trigger bit (TAUAnRDT.TAUAnRDTm) to 1 sets the reload flag (TAUAnRSF.RSFm) to 1, enabling simultaneous rewrite. TAUAnRSF.RSFm remains 1 until simultaneous rewrite has taken place.
- When the specified trigger for simultaneous rewrite is detected, the TAUAnRSF.TAUAnRSFm bit is checked to see if simultaneous rewrite is enabled (TAUAnRSF.TAUAnRSFm = 1). If it is, simultaneous rewrite is carried out. If such writing is disabled, simultaneous rewriting is not performed, and the system awaits the detection of the next simultaneous rewrite trigger.

(3) Simultaneous rewrite

- When the simultaneous rewrite trigger is detected and simultaneous rewrite is enabled (TAUAnRSF.TAUAnRSFm = 1), the current values of the data registers are copied to their buffers. These values are then loaded to the corresponding counters and the values are applied the next time the counter starts or restarts.
- When simultaneous rewriting finishes, the TAUAnRSF.TAUAnRSFm bit is cleared to 0, and the system awaits the next simultaneous rewrite trigger.

12.7.3 Other general rules of simultaneous rewrite

The following rules also apply:

- TAUAnRDE.TAUAnRDEm, TAUAnRDS.TAUAnRDSm, TAUAnRDM.TAUAnRDMm, and TAUAnRDC.TAUAnRDCm cannot be changed while the counter is in operation (TAUAnTE.TAUAnTEm = 1).
- TAUAnTOL.TAUAnTOLm can only be rewritten during operation when in PWM output function or triangle PWM output function. For all other output functions, TAUAnTOL.TAUAnTOLm must be written before the counter starts. If it is rewritten in another function, TAUAnTTOUTm outputs an invalid wave.
- When an upper channel is used as the channel issuing the simultaneous rewrite trigger (TAUAnRDS.TAUAnRDSm = 1), the TAUAnRDC.TAUAnRDCm bit controls all the lower channels. This means that if the TAUAnRDC.TAUAnRDCm bits of CH2 and CH7 are set to 1 and the TAUAnRDC.TAUAnRDCm bits of other channels are set to 0, CH2 and CH7 serve as simultaneous rewrite trigger generation channels. CH2 controls the lower channels CH3 to CH6, and CH7 controls the lower channels CH8 to CH15.
- If simultaneous rewrite is enabled and an upper channel is selected for the simultaneous rewrite trigger (TAUAnRDE.TAUAnRDEm and TAUAnRDS.TAUAnRDSm = 1) but no upper channel is set (TAUAnRDC.TAUAnRDC[15:0] = 0), simultaneous rewrite cannot take place.

12.7.4 Types of simultaneous rewrite

In the following section the four simultaneous rewrite methods are explained using timing diagrams.

(1) Simultaneous rewrite when the master channel (re)starts counting (method A)

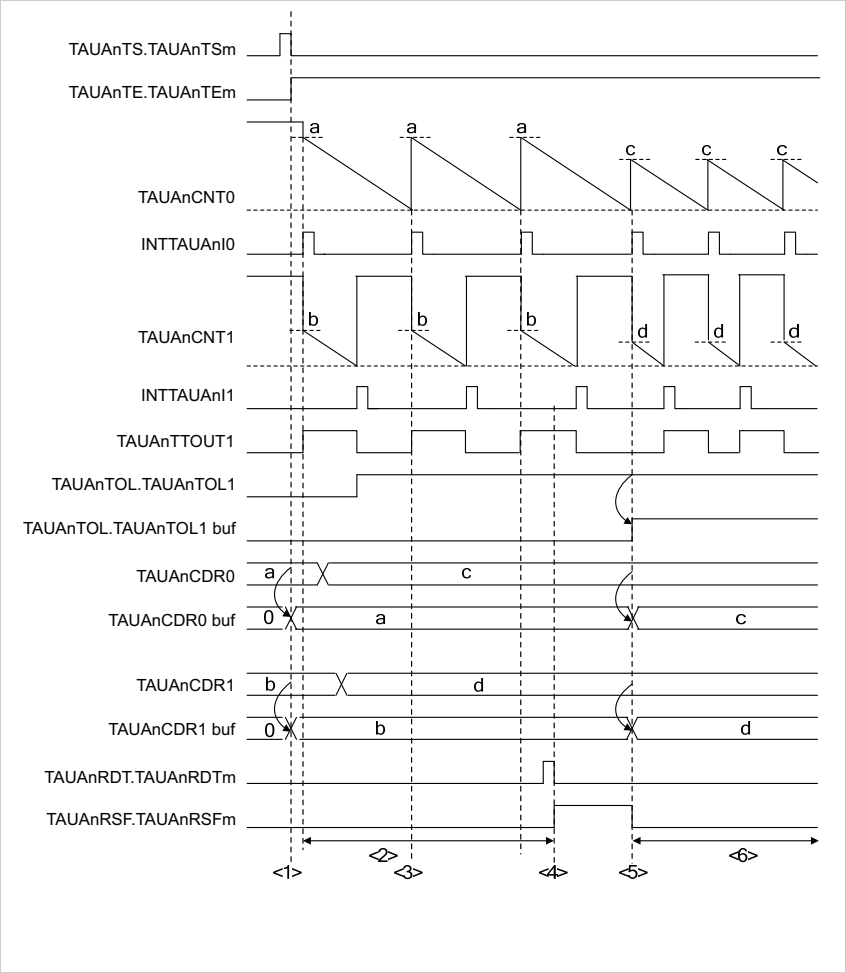


Figure 12-5 Simultaneous rewrite when the master channel (re)starts counting

Setup:

- CH0 is the master channel, counting down, CH1 represents an arbitrary slave channel, and simultaneous rewrite method A is applied.

Description:

1. When TAUAnTS.TAUAnTSm is set to 1, the value of TAUAnCDRm is copied to the TAUAnCDRm buffer, and the value of TAUAnTOL.TAUAnTOLm is copied to the TAUAnTOL.TAUAnTOLm buffer.
2. The TAUAnCDRm and TAUAnTOL.TAUAnTOLm registers can be written at any time.
3. CH0 restarts counting, but simultaneous rewrite does not occur because it is disabled (TAUAnRSF.TAUAnRSFm = 0).
4. The reload data trigger bit (TAUAnRDT.TAUAnRDTm) is set to 1 which sets the status flag (TAUAnRSF.TAUAnRSFm = 1), enabling simultaneous rewrite.
5. Because simultaneous rewriting is enabled, it is performed before counting on channel 0 resumes. The TAUAnCDRm value is loaded to the TAUAnCDRm buffer, and the TAUAnTOL.TAUAnTOLm value is loaded to the TAUAnTOL.TAUAnTOLm buffer.
6. The counters count down and await the next simultaneous rewrite trigger. The values of TAUAnCDRm and TAUAnTOL.TAUAnTOLm can be changed again.

(2) Simultaneous rewrite at the peak of a triangular cycle of the slave channel (method B)

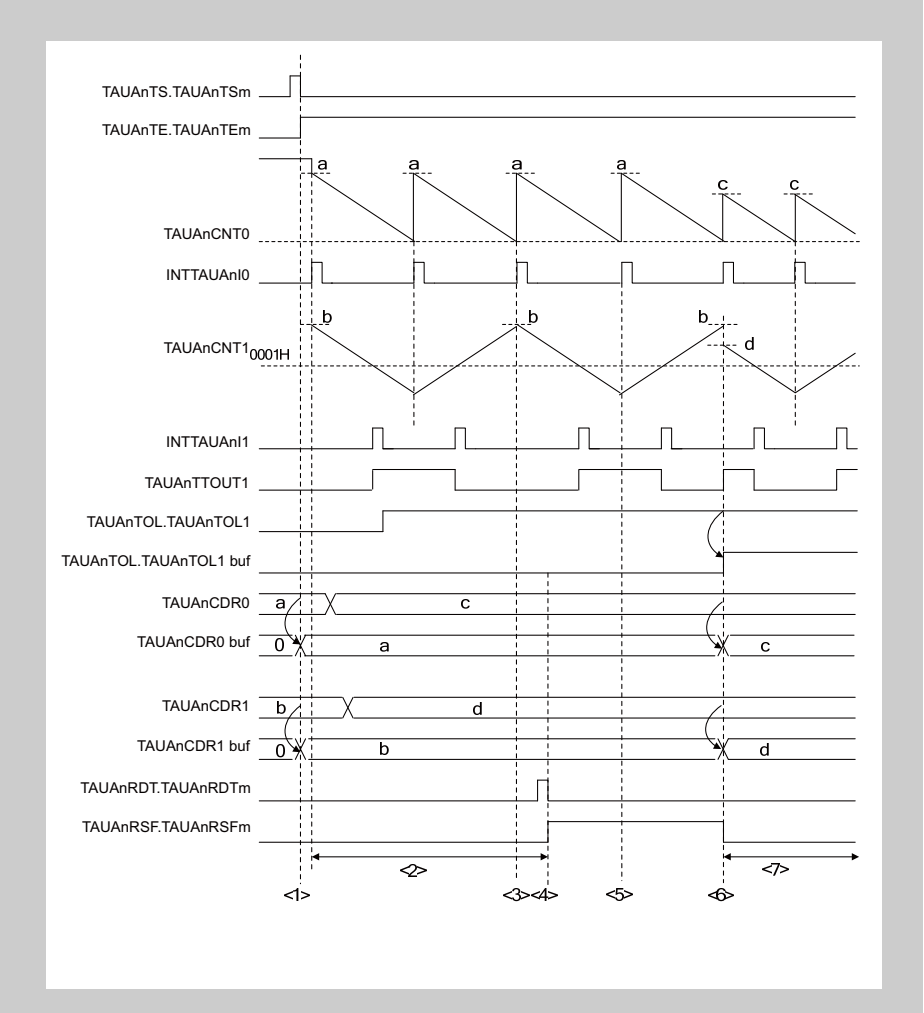


Figure 12-6 Simultaneous rewrite at the peak of a triangular cycle of the slave channel

Setup:

- CH0 is the master channel, counting up and down, CH1 represents an arbitrary slave channel, and simultaneous rewrite method B is applied.

Description:

1. When TAUAnTS.TAUAnTSm is set to 1, the value of TAUAnCDRm is copied to the TAUAnCDRm buffer.
2. The TAUAnCDRm and TAUAnTOL registers can be written at any time.
3. Simultaneous rewrite does not occur because it is disabled (TAUAnRSF.TAUAnRSFm = 0).
4. The reload data trigger bit (TAUAnRDT.TAUAnRDTm) is set to 1 which sets the status flag (TAUAnRSF.TAUAnRSFm = 1), enabling simultaneous rewrite.
5. Simultaneous rewriting does not occur at the valley of a triangular wave cycle.
6. Simultaneous rewriting is performed at the peak of a triangular wave cycle. The TAUAnCDRm value is loaded to the TAUAnCDRm buffer, and the TAUAnTOL.TAUAnTOLm value is loaded to the TAUAnTOL.TAUAnTOLm buffer.
7. The counters count down and await the next simultaneous rewrite trigger. The values of TAUAnCDRm and TAUAnTOL.TAUAnTOLm can be changed again.

(3) Simultaneous rewrite when INTTAUAnIm is generated on an upper channel specified by TAUAnRDC.TAUAnRDCm (method C1)

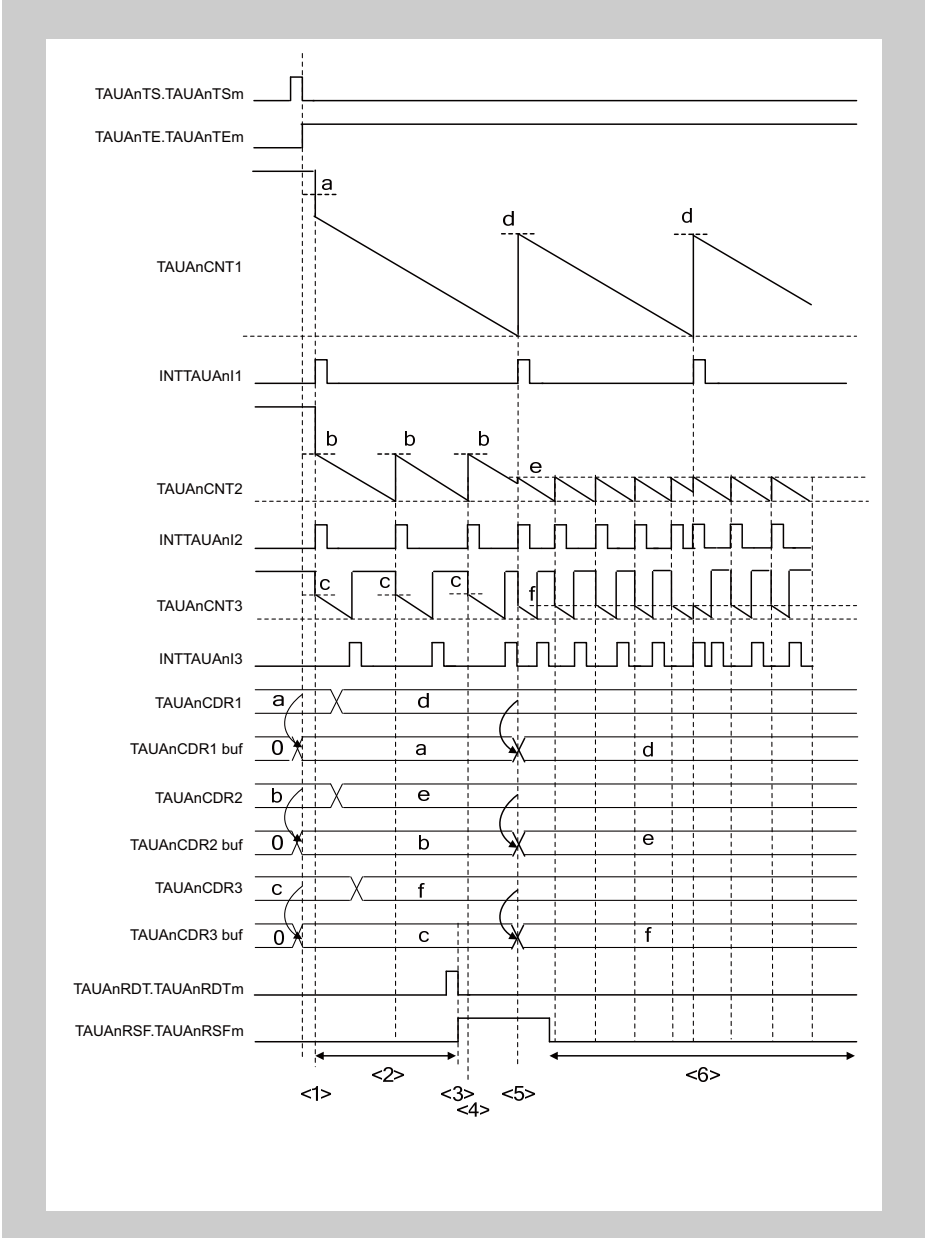


Figure 12-7 Simultaneous rewrite when INTTAUAnIm is generated on an upper channel specified by TAUAnRDC.TAUAnRDCm

Setup:

- CH1 is an upper channel, counting down. CH2 is a master channel, CH3 is the slave channel, and simultaneous rewrite method C1 is applied. The TAUAnRDC register specifies the channel to generate the simultaneous rewrite trigger.

Description:

1. When TAUAnTS.TAUAnTSm is set to 1, the value of TAUAnCDRm is copied to the TAUAnCDRm buffer.
2. The TAUAnCDRm register can be written at any time.
3. The reload data trigger bit (TAUAnRDT.TAUAnRDTm) is set to 1 which sets the status flag (TAUAnRSF.TAUAnRSFm = 1), enabling simultaneous rewrite.
4. Even though simultaneous rewrite is enabled, it does not take place because it is only triggered by an interrupt on channel 1.
5. Simultaneous rewriting is triggered by INT1, which is caused by counter 1 reaching 0000H. The TAUAnCDRm values are loaded to the corresponding TAUAnCDRm buffers.
6. The counter counts down and awaits the next simultaneous rewrite trigger. The values of the TAUAnCDRm registers can be changed again.

(4) Simultaneous rewrite when INTTAUAnIm is generated on an upper channel specified by TAUAnRDC.TAUAnRDCm that in turn is triggered by an external signal (method C2)

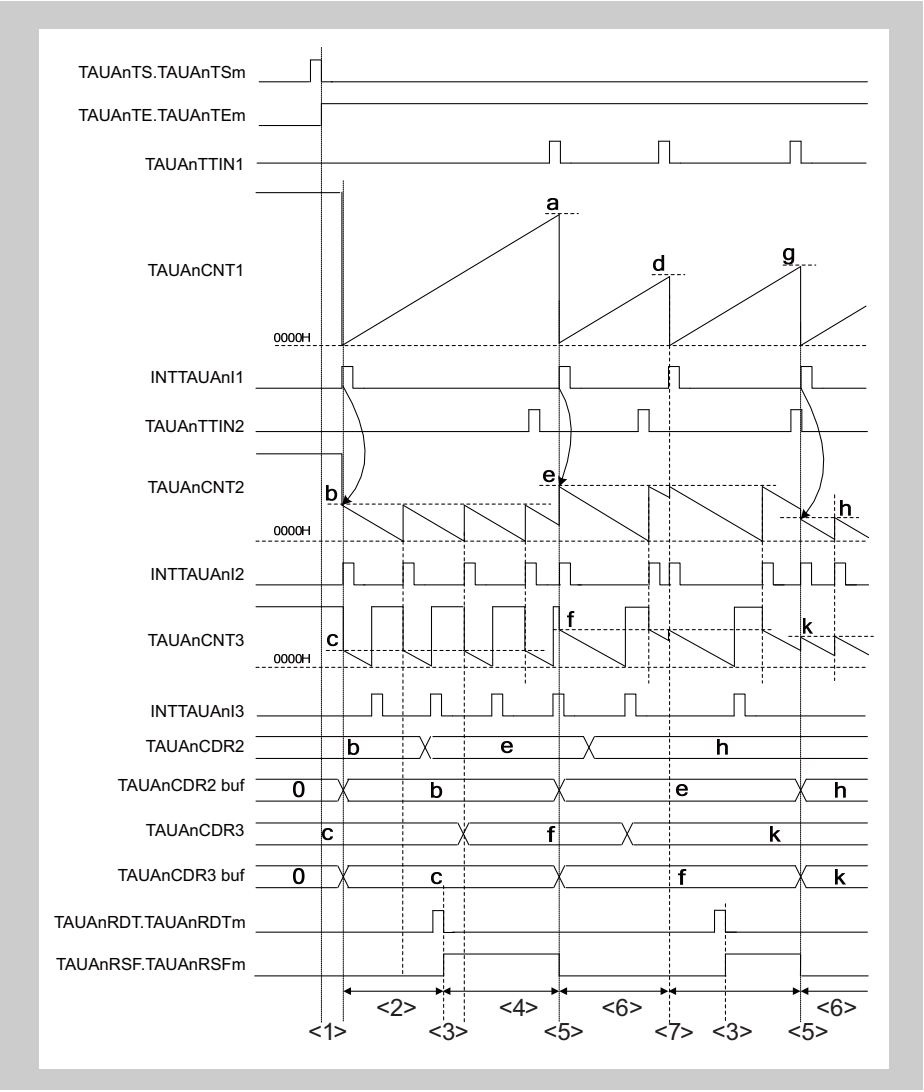


Figure 12-8 Simultaneous rewrite when INTTAUAnIm is generated on an upper channel specified by TAUAnRDC.TAUAnRDCm that in turn is triggered by an external signal

Setup:

- CH1 is an upper channel, counting down. CH2 is a master channel, CH3 is the slave channel, and synchronous channel operation method C2 is applied. The TAUAnRDC register specifies which upper channel is monitored for an INTTAUAnIm trigger.

Description:

1. When TAUAnTS.TAUAnTSm is set to 1, the value of TAUAnCDRm is copied to the TAUAnCDRm buffer.
2. The TAUAnCDRm register can be written at any time.
3. The reload data trigger bit (TAUAnRDT.TAUAnRDTm) is set to 1 which sets the status flag (TAUAnRSF.TAUAnRSFm = 1), enabling simultaneous rewrite.
4. Even though simultaneous rewrite is enabled, it does not take place because it is only triggered by an interrupt on channel 1.
5. Simultaneous rewriting is triggered by INT1, which is caused by the external signal TIN1. The TAUAnCDRm values are loaded to the corresponding TAUAnCDRm buffers.
6. The counters count down and await the next simultaneous rewrite trigger. The values of the TAUAnCDRm registers can be changed again.
7. An external signal occurs at TIN2 but simultaneous rewrite does not take place because it is disabled (TAUAnRSF.TAUAnRSFmTAUAnRSF.TAUAnRSFm = 0).

12.8 Channel Output Modes

The output of the TAUAnTTOUTm pin can be controlled in two ways, the latter of which can be further split into individual modes:

- By software (TAUAnTOE.TAUAnTOEm = 0)

  When controlled by software, the output register bit (TAUAnTO.TAUAnTOm) can be written and the value of the bit is output from to the output pin (TAUAnTTOUTm).

- By TAUA signals (TAUAnTOE.TAUAnTOEm = 1)

  When operated by TAUA signals, the output level of TAUAnTTOUTm is set or reset or toggled by internal signals. The value of TAUAnTO.TAUAnTOm is updated accordingly to reflect the value of TAUAnTTOUTm.

  – Independently (TAUAnTOM.TAUAnTOMm = 0)

    When operated independently, the output of the TAUAnTTOUTm pin is only affected by settings of channel m. Therefore, independent channel operation must be selected (TAUAnTOM.TAUAnTOMm = 0).

  – Synchronously (TAUAnTOM.TAUAnTOMm = 1)

    When operated synchronously, the output of the TAUAnTTOUTm pin is affected by settings of channel m and those of other channels. Therefore, synchronous channel operation must be selected for all participating channels (TAUAnTOM.TAUAnTOMm = 1).

The TAUAnTO.TAUAnTOm bit can always be read to determine the current value of TAUAnTTOUTm, regardless of whether the pin is controlled by software, operated independently, or operated synchronously.

Control bits

The settings of the control bits required to select a specific channel output mode are listed in *Table 12-9 “Channel output modes” on page 578*.

The channel output modes are described in detail in

- *12.8.2 “Channel output modes controlled independently by TAUAn signals” on page 580*
- *12.8.3 “Channel output modes controlled synchronously by TAUAn signals” on page 582*.

Collective TAUAnTOm bit manipulation

Whether to apply settings to the TAUAnTOm bits is controlled using the TAUAnTOE.TAUAnTOEm bits.

When writing to the TAUAnTO register, TAUAnTOm settings are only written to channels for which the TAUAnTOE.TAUAnTOEm bit is cleared to 0. The TAUAnTOm settings are not applied for channels for which the corresponding TAUAnTOE.TAUAnTOEm bit is set to 1.

Note

The TAUAnTO.TAUAnTOm bits are allocated so that the bit numbers correspond to the channel numbers.

Output logic

Positive logic or inverted logic of the output is specified by control bit TAUAnTOL.TAUAnTOLm.

The value of the TAUAnTOL.TAUAnTOLm bit must be set before the counter is started. It can only be changed during operation in PWM output function or triangle PWM output function. Otherwise, changes to TAUAnTOL.TAUAnTOLm result in an undefined TAUAnTTOUTm signal.

Refer to *12.7 “Simultaneous Rewrite” on page 565*.

The various channel output modes and the channel output control bits are listed in the following table.

Table 12-9 Channel output modes

Channel output mode	TAUAn TOE. TAUAn TOEm	TAUAn TOM. TAUAn TOMm	TAUAn TOC. TAUAn TOCm	TAUAn TDE. TAUAn TDEm	TAUAn TRE. TAUAn TREm	TAUAn TME. TAUAn TMEm	TAUAn TDM. TAUAn TDMm
By software							
Independent Channel Output Mode Controlled by Software	0	x					
By TAUA signals, independently							
Independent Channel Output Mode 1	1	0	0	0	0	0	0
with Real-Time Output					1		
Independent Channel Output Mode 2			1		0		
By TAUA signals, synchronously							
Synchronous Channel Output Mode 1	1	1	0	0	0	0	0
with Non-Complementary Modulation Output					1	1	
Synchronous Channel Output Mode 2			1	0	0	0	0
with Dead Time Output				1			
with One-Phase PWM Output							1
with Complementary Modulation Output					1	1	0
with Non-Complementary Modulation Output			1	0			

- All combinations not listed in this table are forbidden.
- Bits marked with an x can be set to any value.

Note The following bits cannot be changed during count operation (TAUAnTE.TAUAnTE = 1):

- TAUAnTOE.TAUAnTOEm,
- TAUAnTOM.TAUAnTOMm,
- TAUAnTOC.TAUAnTOCm,
- TAUAnTDE.TAUAnTDEm,
- TAUAnTRE.TAUAnTREm, and
- TAUAnTDM.TAUAnTDMm

The following bits cannot be changed during count operation (TAUAnTE.TAUAnTEm = 1) except in channel output modes with modulation output:

- TAUAnTME.TAUAnTMEm,
- TAUAnTDL.TAUAnTDLm

12.8.1 General procedure for specifying a channel output mode

The following steps describe the general procedure for specifying a TAUAnTTOUTm channel output mode. The prerequisite is that timer output operation is disabled (TAUAnTOE.TAUAnTOEm = 0).

1. Set TAUAnTO.TAUAnTOm to specify the initial level of the TAUAnTTOUTm output.
2. Set the channel output mode using *Table 12-9 “Channel output modes” on page 578* and the output logic using the TAUAnTOL.TAUAnTOLm bit.
3. Start the counter (TAUAnTS.TAUAnTSm = 1).

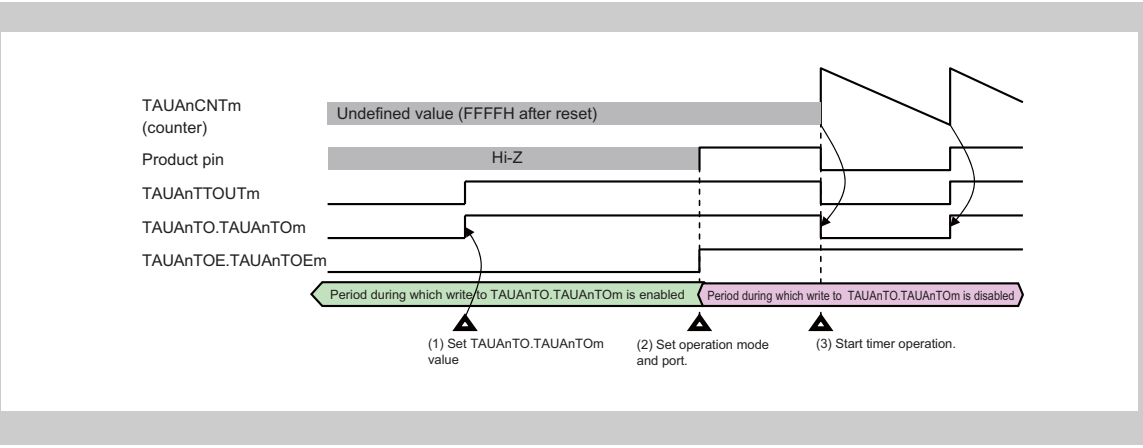


Figure 12-9 General procedure for specifying a TAUAnTTOUTm channel output mode

12.8.2 Channel output modes controlled independently by TAUAn signals

This chapter lists the channel output modes that are controlled independently by TAUAn signals. The control bits used to specify a mode are listed in *Table 12-9 “Channel output modes” on page 578*.

(1) Independent Channel Output Mode 1

Set/reset conditions In this output mode, TAUAnTTOUTm toggles when INTTAUAnIm is detected. The value of TAUAnTOL.TAUAnTOLm is ignored.

Prerequisites None, other than those in *Table 12-9 “Channel output modes” on page 578*.

(2) Independent Channel Output Mode 1 with Real-Time Output

In this output mode, the value of the TAUAnTRO.TAUAnTROm bit of the trigger channel is output to TAUAnTTOUTm. The trigger channel is specified by setting the corresponding TAUAnTRC.TAUAnTRCm bit to 1. It controls all lower channels for which TAUAnTRC.TAUAnTRCm = 0.

Set/reset conditions The value of the TAUAnTRO.TAUAnTROm bit is only sent to TAUAnTTOUTm when an interrupt INTTAUAnIm occurs on the trigger channel. The interrupt is generated either:

- at certain specified intervals or
- on detection of a valid TAUAnTTINm input edge / counter start

The type of trigger is set using the TAUAnCMORm.TAUAnMD[4:1] bits.

Prerequisites Both master and slave channels can be set as a trigger generation channel. A channel for which TAUAnTRC.TAUAnTRCm is set to 1 serves as the trigger generation channel even if TAUAnTRE.TAUAnTREm is set to 0. Real-time output cannot take place if there is no channel for which TAUAnTRE.TAUAnTREm is set to 1 or if TAUAnTRCm.TRC0 = 0.

This can be seen in the following figure.

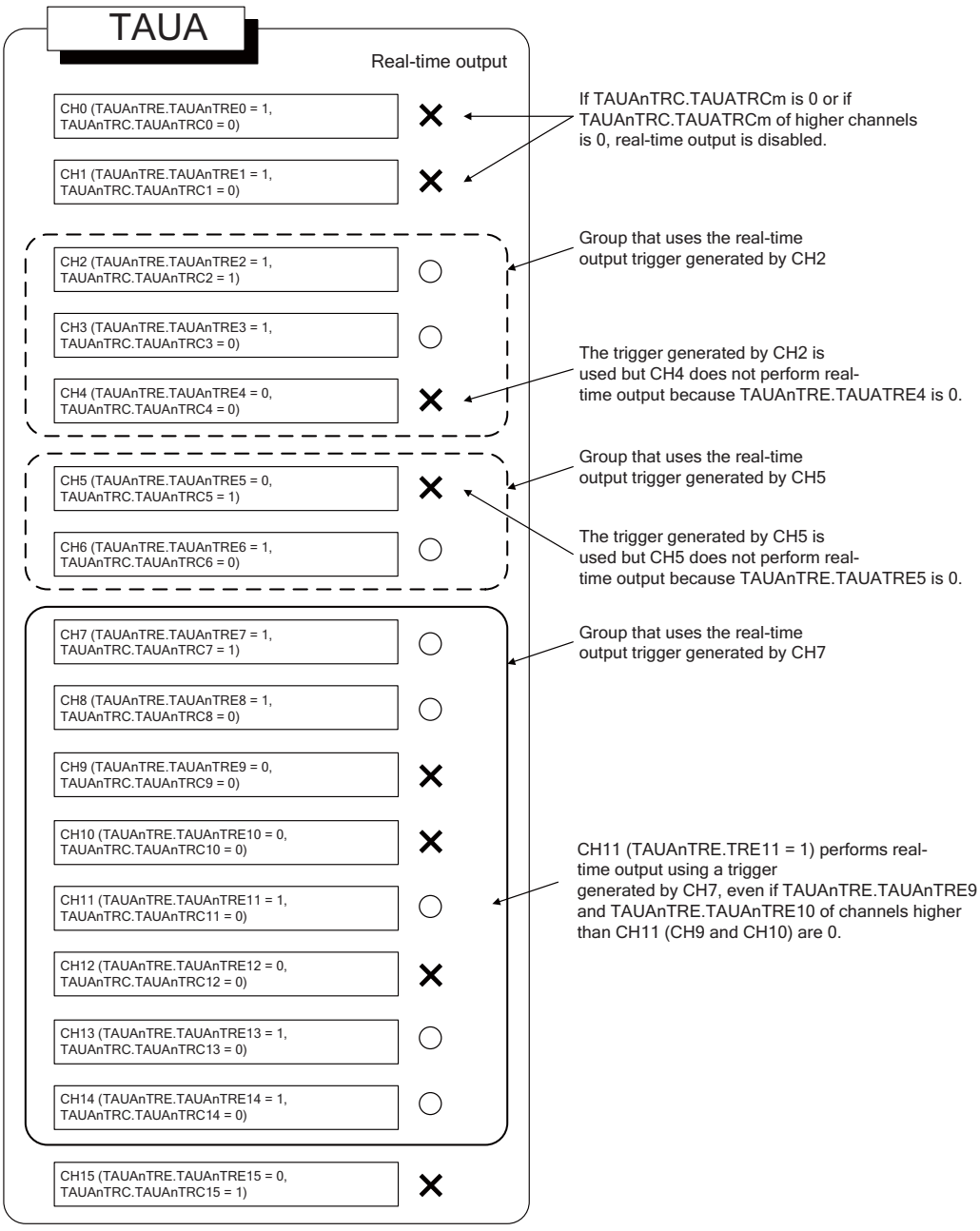


Figure 12-10 Real-time output

(3) Independent Channel Output Mode 2

Set/reset conditions In this output mode, TAUAnTTOUTm is set when INTTAUAnIm occurs upon count start and reset when INTTAUAnIm occurs due to a match between TAUAnCNTm and TAUAnCDRm.

Prerequisites None, other than those in *Table 12-9 “Channel output modes” on page 578*.

12.8.3 Channel output modes controlled synchronously by TAUAn signals

This chapter lists the channel output modes that are controlled synchronously by TAUAn signals. The control bits used to specify a mode are listed in *Table 12-9 "Channel output modes" on page 578*.

(1) Synchronous Channel Output Mode 1

Set/reset conditions

In this output mode, INTTAUAnIm of the master channel serves as the set signal and INTTAUAnIm of the slave channel as the reset signal. If INTTAUAnIm of the master channel and INTTAUAnIm of the slave channel are generated at the same time, INTTAUAnIm of the slave channel (reset signal) has priority over INTTAUAnIm (set signal) of the master channel, i.e. the master channel is ignored.

Prerequisites

None, other than those in *Table 12-9 "Channel output modes" on page 578*.

(2) Synchronous Channel Output Mode 1 with Non-Complementary Modulation Output

Set/reset conditions

In this output mode, TAUAnTTOUTm outputs the result of an AND operation between the PWM output of a channel and the real-time output bit (TAUAnTRO.TAUAnTROm).

The phase period to which the dead time is added is specified using the TAUAnTDL.TAUAnTDLm bit; for positive phase set TAUAnTDL.TAUAnTDLm = 0 and for negative phase set TAUAnTDL.TAUAnTDLm = 1.

Prerequisites

A set of at least three channels is required to generate the PWM output. The master channel and slave 1 generate the period, and slave 2 generates the duty cycle. In typical applications, a further 5 slave channels are also used that operate in the same manner as slave 2.

Only the PWM output and the real-time output bit of the same channel can be combined.

TAUAnTRO.TAUAnTROm, TAUAnTME.TAUAnTMEm, and TAUAnTDL.TAUAnTDLm can only be changed during count operation.

- If TAUAnTME.TAUAnTMEm is changed, the new value of TAUAnTME.TAUAnTMEm is applied upon detection of INTTAUAnIm on the specified channel.
- If TAUAnTME.TAUAnTMEm and TAUAnTDL.TAUAnTDLm are changed, the new values are applied upon detection of INTTAUAnIm on the master channel.

(3) Synchronous Channel Output Mode 2

In this output mode, the operation mode must be set to Up Down Count mode. The result is a triangle PWM wave at TAUAnTTOUTm. For details refer to *12.24.1 "Triangle PWM Output Function" on page 776*.

Set/reset conditions

TAUAnCNTm of the slave channel counts down and up alternatively. When it passes 0001_H it generates an interrupt, causing TAUAnTTOUTm to toggle.

Prerequisites

A set of two channels is required to generate the triangle PWM output. TAUAnTTOUTm must be set to 0 before the function starts.

(4) Synchronous Channel Output Mode 2 with Dead Time Output

In this output mode, a dead time delay is added to TAUAnTTOUTm. The set/reset conditions are shown in the following figure.

Set/reset conditions

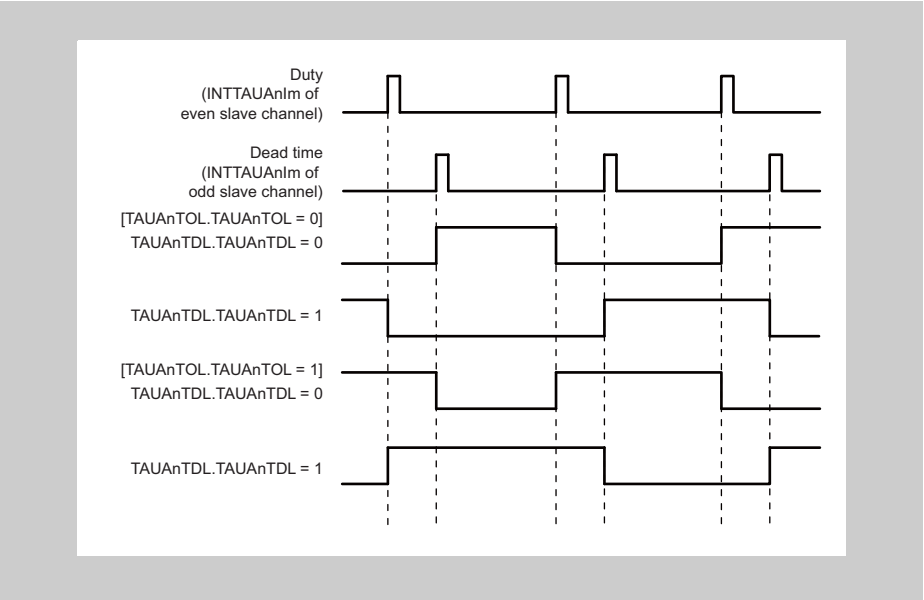


Figure 12-11 Set/reset conditions for Synchronous Channel Output Mode 2 with Dead Time Output

The edge to which the dead time is added is specified using the TAUAnTDL.TAUAnTDLm bit; for rising edge set TAUAnTDL.TAUAnTDLm = 0 and for falling edge set TAUAnTDL.TAUAnTDLm = 1.

Prerequisites Dead time control requires a set of three channels, each operating in the following modes:

- One master channel

  The master channel must be set to Interval Timer Mode

- One even slave channel

  The even slave channel must be set to Up Down Count Mode

- One odd slave channel (even channel + 1)

  The odd slave channel must be set to One Count Mode

The values of the following bits must be the same for the odd channel and the even channel:

- TAUAnTOE.TAUAnTOEm,
- TAUAnTME.TAUAnTMEm,
- TAUAnTRE.TAUAnTREm,
- TAUAnTOM.TAUAnTOMm,
- TAUAnTOC.TAUAnTOCm,
- TAUAnTDE.TAUAnTDEm, and
- TAUAnTDM.TAUAnTDMm

(5) Synchronous Channel Output Mode 2 with One-Phase PWM Output

In this output mode, a dead time delay is added to TAUAnTTOUTm. The set/reset conditions are shown in the following figure.

Set/reset conditions

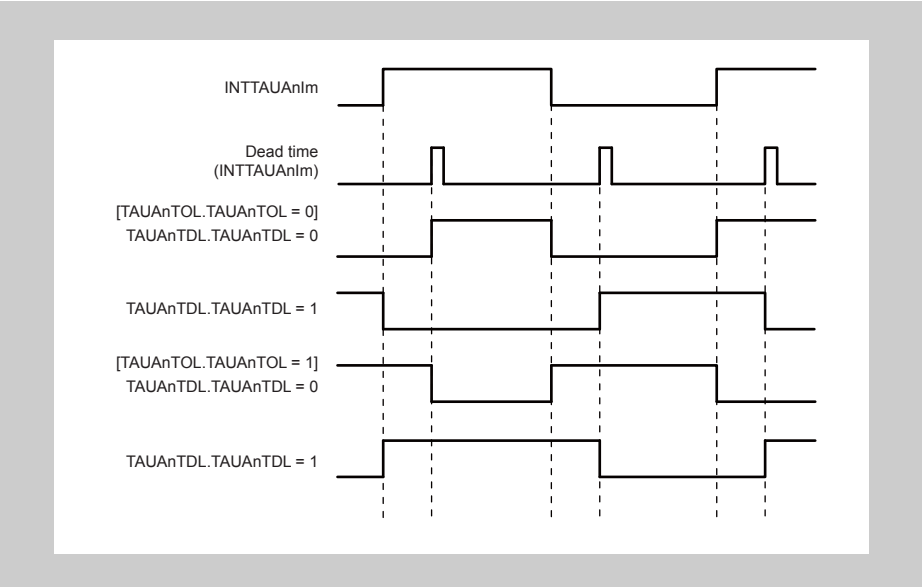


Figure 12-12 Set/reset conditions for Synchronous Channel Output Mode 2 with One-Phase PWM Output

The edge to which the dead time is added is specified using the TAUAnTDL.TAUAnTDLm bit; for rising edge set TAUAnTDL.TAUAnTDLm = 0 and for falling edge set TAUAnTDL.TAUAnTDLm = 1.

Prerequisites One-phase PWM output control requires a set of two channels:

- One even slave channel
- One odd slave channel (even channel + 1)

  The odd slave channel must be set to “One Count Mode”

The values of the following bits must be the same for the odd channel and the even channel:

- TAUAnTOE.TAUAnTOEm,
- TAUAnTME.TAUAnTMEm,
- TAUAnTRE.TAUAnTREm,
- TAUAnTOM.TAUAnTOMm,
- TAUAnTOC.TAUAnTOCm,
- TAUAnTDE.TAUAnTDEm, and
- TAUAnTDM.TAUAnTDMm

(6) Synchronous Channel Output Mode 2 with Complementary Modulation Output

Set/reset conditions

In this output mode, TAUAnTTOUTm outputs a PWM signal, a high signal, or a low signal depending on the value of the real-time output bit (TAUAnTRO.TAUAnTROm), the modulation output bit (TAUAnTME.TAUAnTMEm), and the output level bit (TAUAnTOL.TAUAnTOLm) of a pair of slave channels.

For details refer to *12.25.3 “Complementary Modulation Output Function” on page 831*.

Prerequisites

A set of at least four channels is required for this mode. The master channel and slave 1 generate the period, slave channel 2 generates the duty cycle, and slave 3 generates the dead time. Slave 2 and 3 are a pair. In typical applications, a further 4 channels are also used that operate in the same manner as slaves 2 and 3 respectively.

TAUAnTRO.TAUAnTROm, TAUAnTME.TAUAnTMEm, and TAUAnTDL.TAUAnTDLm can only be changed during count operation.

- If TAUAnTME.TAUAnTMEm is changed during operation, the new TAUAnTME.TAUAnTMEm value is applied upon detection of INTTAUAnIm at the specified channel.
- If TAUAnTME.TAUAnTMEm and TAUAnTDL.TAUAnTDLm are changed, the new values are applied upon detection of INTTAUAnIm on an even slave channel.

(7) Synchronous Channel Output Mode 2 with Non-Complementary Modulation Output

The difference to "Synchronous Channel Output Mode 1 with Non-Complementary Modulation Output" is the PWM wave shape.

It is a rectangular wave with mode 1 while it is a triangular wave with mode 2.

12.9 Start Timing of Operating Modes

This chapter describes when the counters of the different operating modes start after the TAUAnTS.TAUAnTSm bit is set to 1.

In all modes, the value of the data register and whether or not an interrupt is issued depends on the individual mode and corresponding register settings.

Caution The timing for starting counting in this section is only for reference. The timing for starting counting actually varies according to the count clock timing.

12.9.1 Interval Timer Mode, Judge Mode, Capture Mode, Up Down Count Mode

The counter starts at the start of the next count clock cycle after TAUAnTS.TAUAnTSm is set to 1. The value of data register is also loaded at the point the counter starts.

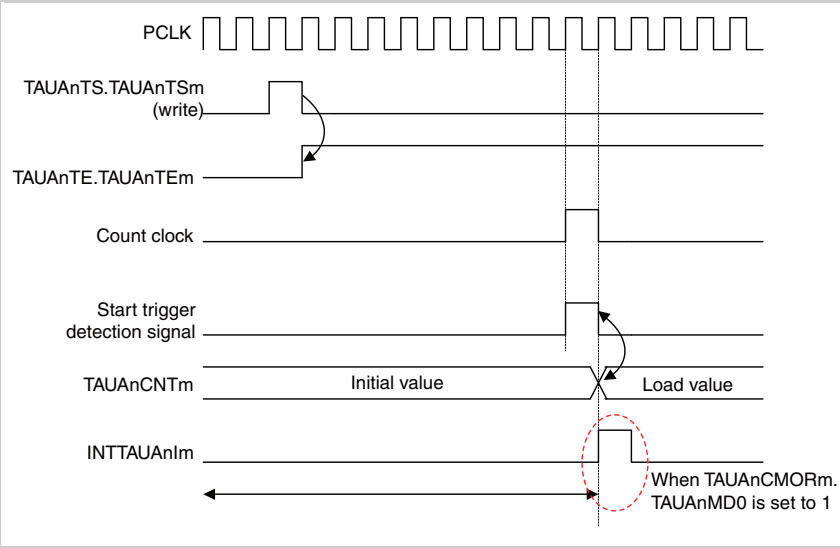


Figure 12-13 Start timing of Interval Timer Mode, Judge Mode, Capture Mode, Up Down Count Mode

12.9.2 Event Mode

The value of the data register is loaded as soon as TAUAnTS.TAUAnTSm is set to 1. The counter also starts immediately. The value of the data register increments at the start of each subsequent count clock cycle.

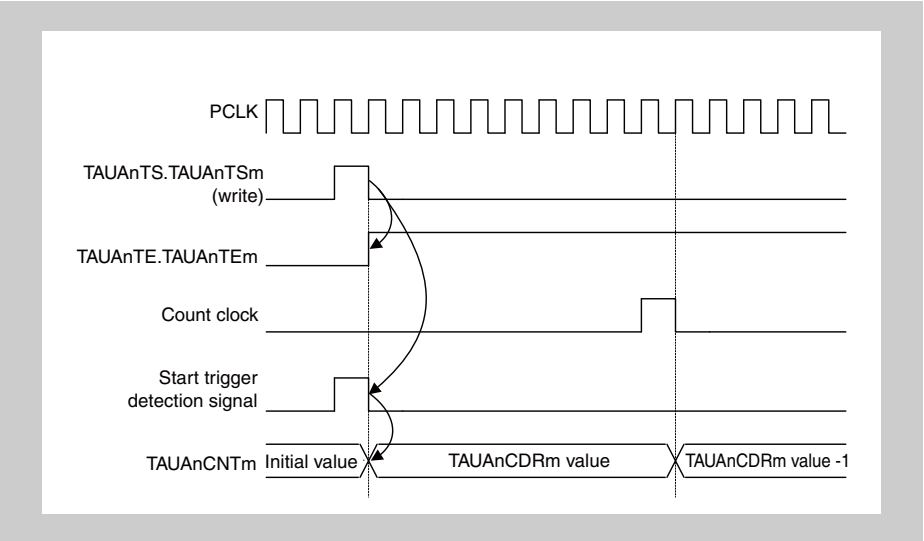


Figure 12-14 Start timing of Event Mode

12.9.3 All other operating modes

In all other operating modes, the count clock cycles are ignored with regard to starting the counter. The counter is only triggered by detection of a valid TAUAnTTINm edge. The value of data register is also loaded at the point the counter starts. Nevertheless, the count clock cycles determine the frequency with which all operations take place.

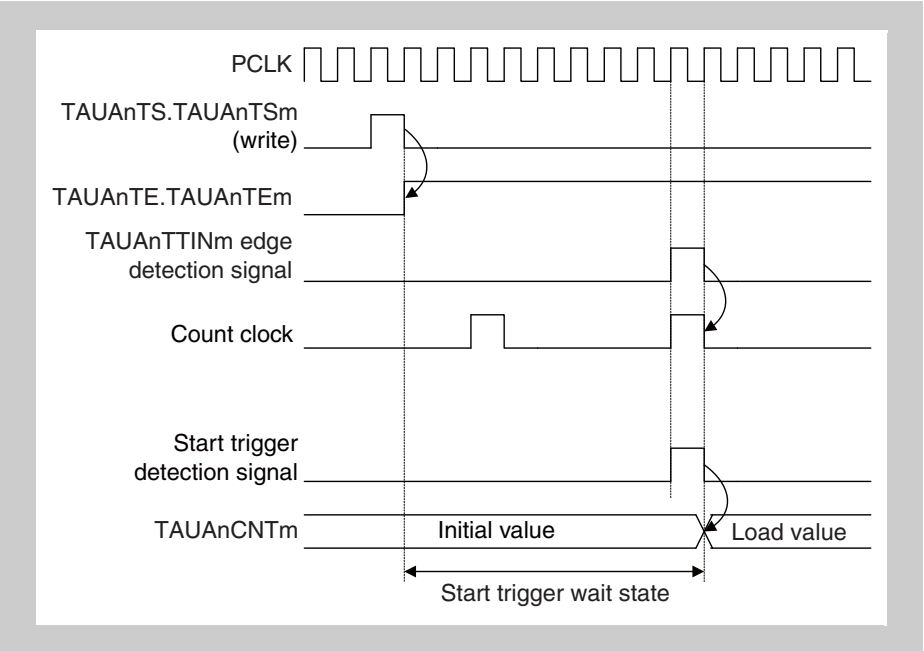


Figure 12-15 Start timing of all other operating modes

12.10 TAUAnTTOUTm Output and INTTAUAnIm Generation when Counter Starts or Restarts

When the counter starts, it is possible to specify whether an INTTAUAnIm is generated using the TAUAnCMORn.TAUAnMD0 bit. The effect of the bit depends on the selected mode, as shown in the following table. The effects of INTTAUAnIm on TAUAnTTOUTm depend on the selected channel operation function.

Table 12-10 Effect of TAUAnCMORm.TAUAnMD0 bit on generation of INTTAUAnIm when counter is triggered

Mode	TAUAnCMORm. TAUAnMD0 bit	INTTAUAnIm generated when counter starts
Interval Timer Mode Capture Mode Count Capture Mode	0	No
	1	Yes
Capture & One Count Mode Capture & Gate Count Mode Event Count Mode Up Down Count Mode	0	No
One Count Mode Gate Count Mode	0/1	No, regardless of setting of TAUAnCMORn.TAUAnMD0 bit.
Pulse One Count Mode		Yes, regardless of setting of TAUAnCMORn.TAUAnMD0 bit.

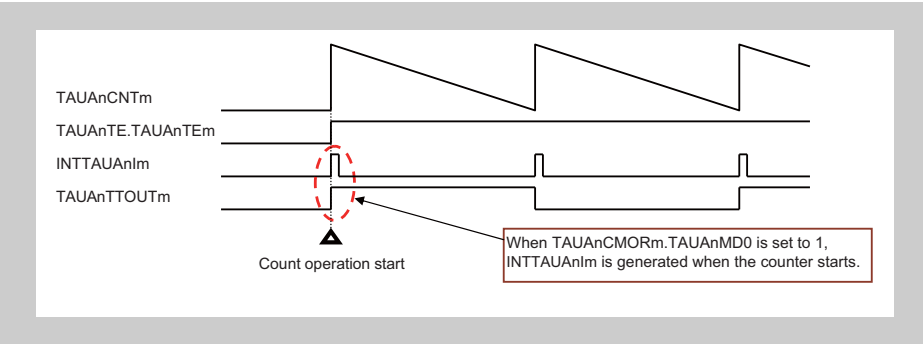


Figure 12-16 INTTAUAnIm generated when counter starts

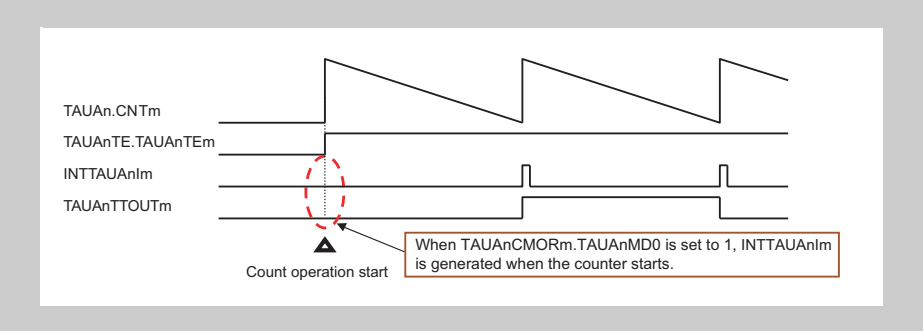


Figure 12-17 INTTAUAnIm not generated when counter starts

12.11 Interrupt Generation upon Overflow

Certain independent functions that count up, overflow without generating an interrupt when they reach $FFFF_H$. This section describes how it is possible to generate an interrupt, by combining a channel operating in one of these modes with a channel in a different operation mode which counts down.

The appropriate operation mode for the second channel depends on the operation mode of the first channel. Nevertheless, the principle is the same for all combinations:

- Find a operation mode for the second channel that counts down in such a manner, that it reaches 0000_H at the same time as the first channel overflows (TAUAnCNTm = $FFFF_H$).
- Set TAUAnCDRm of the second channel to $FFFF_H$
- The two channels must count at the same speed (i.e. they must have the same count clock)
- Both channels are triggered by the same TAUAnTTINm input

Result: The down-counter of the second channel reaches 0000_H at exactly the same time as the up-counter of the first channel overflows (TAUAnCNTm = $FFFF_H$). Thus the second channel generates the desired interrupt.

The following sections list the operating modes that count down that are required to match specific operating modes that count up, as well as example timing diagrams.

12.11.1 Capture Mode

Applies to
- TAUAnTTINm Input Pulse Interval Measurement Function

Combine with
Interval Timer Mode

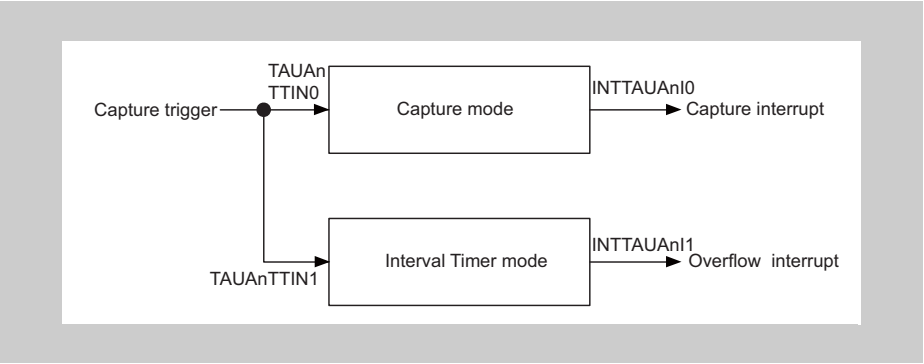


Figure 12-18 Combination of Capture Mode and Interval Timer Mode

Timing diagram

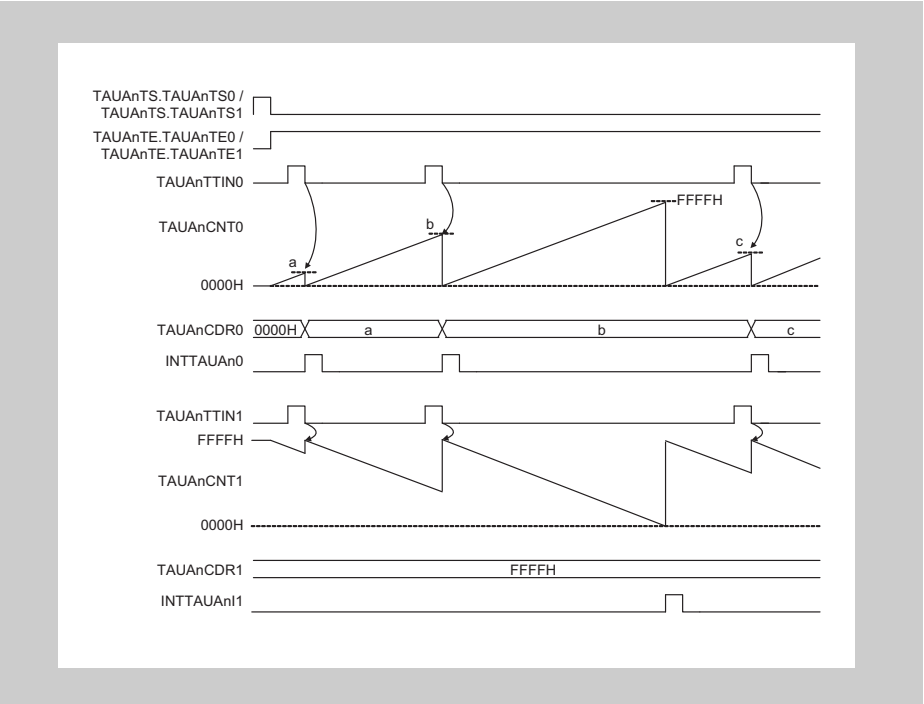


Figure 12-19 Interrupt generation via combination of Capture Mode and Interval Timer Mode

12.11.2 Capture and One Count Mode

Applies to
- TAUAnTTINm Input Signal Width Measurement Function

Combine with
One Count Mode

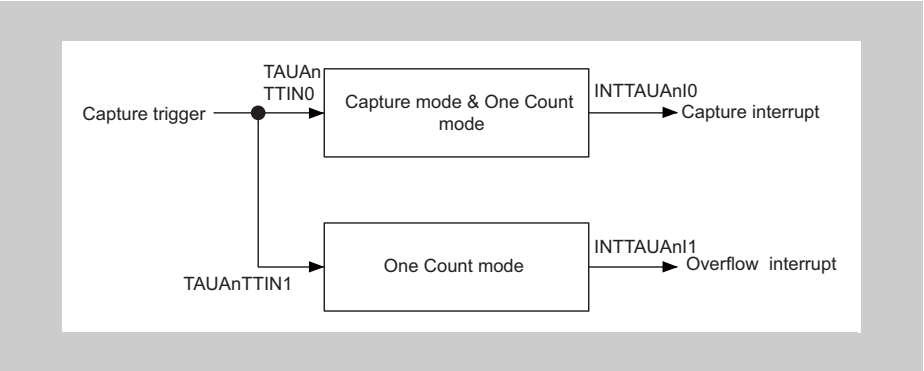


Figure 12-20 Combination of Capture and One Count Mode and One Count Mode

Timing diagram

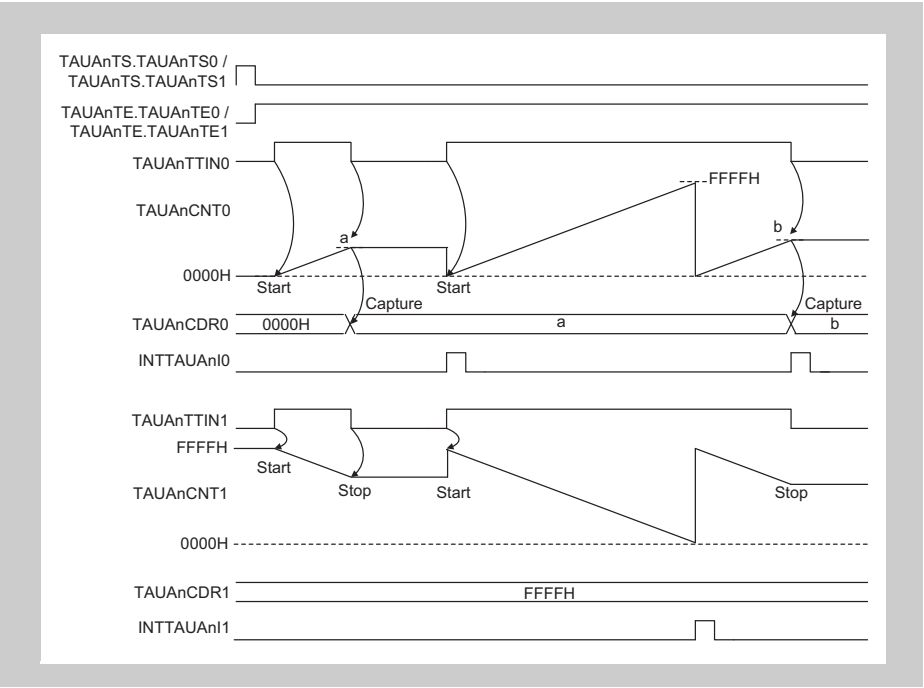


Figure 12-21 Interrupt generation via combination of Capture and One Count Mode and One Count Mode

12.11.3 Count Capture Mode

Applies to • TAUAnTTINm Input Position Detection Function

Combine with Interval Timer Mode

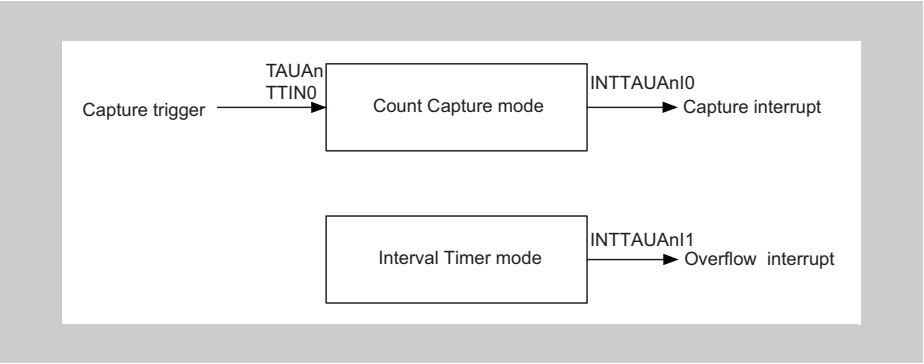


Figure 12-22 Combination of Count Capture Mode and Interval Timer Mode

Timing diagram

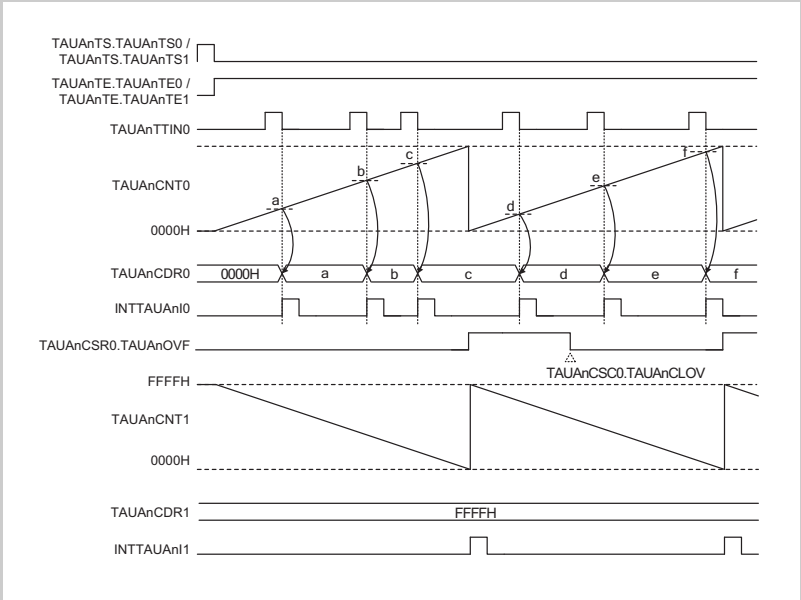


Figure 12-23 Interrupt generation via combination of Count Capture Mode and Interval Timer Mode

In the above timing diagram, TAUAnCSRm.TAUAnOVF is set to 1 when TAUAnCNTm overflows.

TAUAnCSRm.TAUAnOVF is cleared by writing 1 to TAUAnCSCm.TAUAnCLOV.

12.11.4 Capture and Gate Count Mode

Applies to • TAUAnTTINm Input Period Count Detection Function

Combine with Gate Count Mode

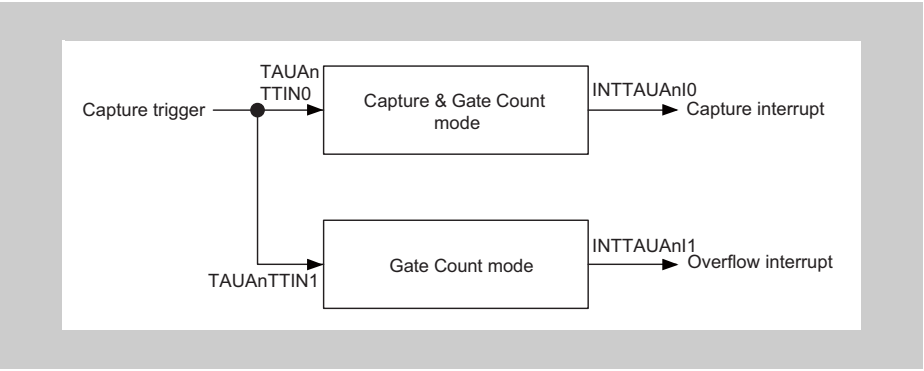


Figure 12-24 Combination of Capture and Gate Count Mode and Gate Count Mode

Timing diagram

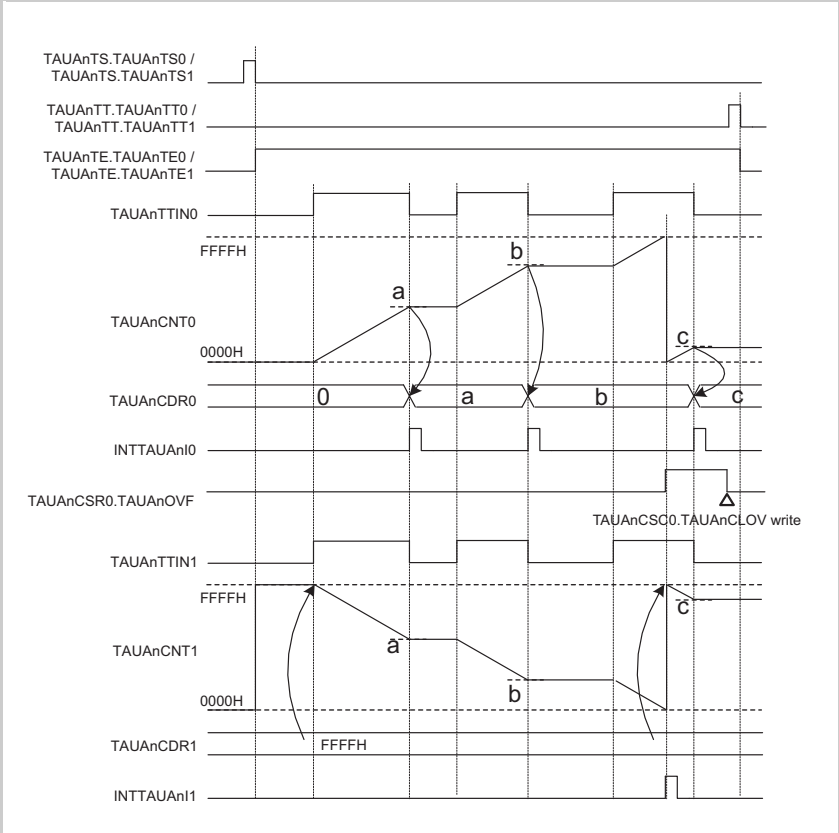


Figure 12-25 Interrupt generation via combination of Capture and Gate Count Mode and Gate Count Mode

In the above timing diagram, TAUAnCSRm.TAUAnOVF is set to 1 when TAUAnCNTm overflows. TAUAnCSRm.TAUAnOVF is cleared by writing 1 to TAUAnCSCm.TAUAnCLOV.

12.12 TAUAnTTINm Edge Detection

Edge detection is based on the operation clock. This means that an edge can only be detected at the next rising edge of the operation clock. This can lead to a maximum delay of one operation clock cycle.

The following figure shows an edge detection timing example.

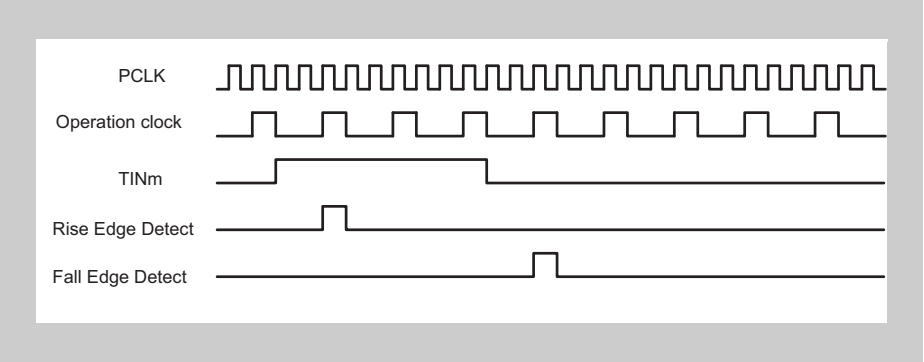


Figure 12-26 Basic edge detection timing

Figure 12-26 "Basic edge detection timing" shows the approximate operation timing. In actuality, there is delay due to the noise filter and synchronizer between the TAUAnIm pin and TAUAn.

12.13 Assigning DMA Window Addresses

DMA (direct memory access) can be used to store values in the TAUAnDWRj registers, for example the current values of the TAUAnCDRm register and the TAUAnTOL register.

For example, the following figure shows how to rewrite the period register, duty registers, and TAUAnTOL register to the TAUAnDWRj registers.

1. Specify the addresses of the selected registers (TAUAnCDR0, TAUAnCDR2, TAUAnCDR4, TAUAnCDR6, TAUAnCDR9, TAUAnCDR13, TAUAnTOL.TAUAnTOLm, and TAUAnRDT.TAUAnRDTm) to the TAUAnDAS registers.
2. The TAUAnDMA window address function then loads the values of the selected registers to the corresponding TAUAnDWRj registers.

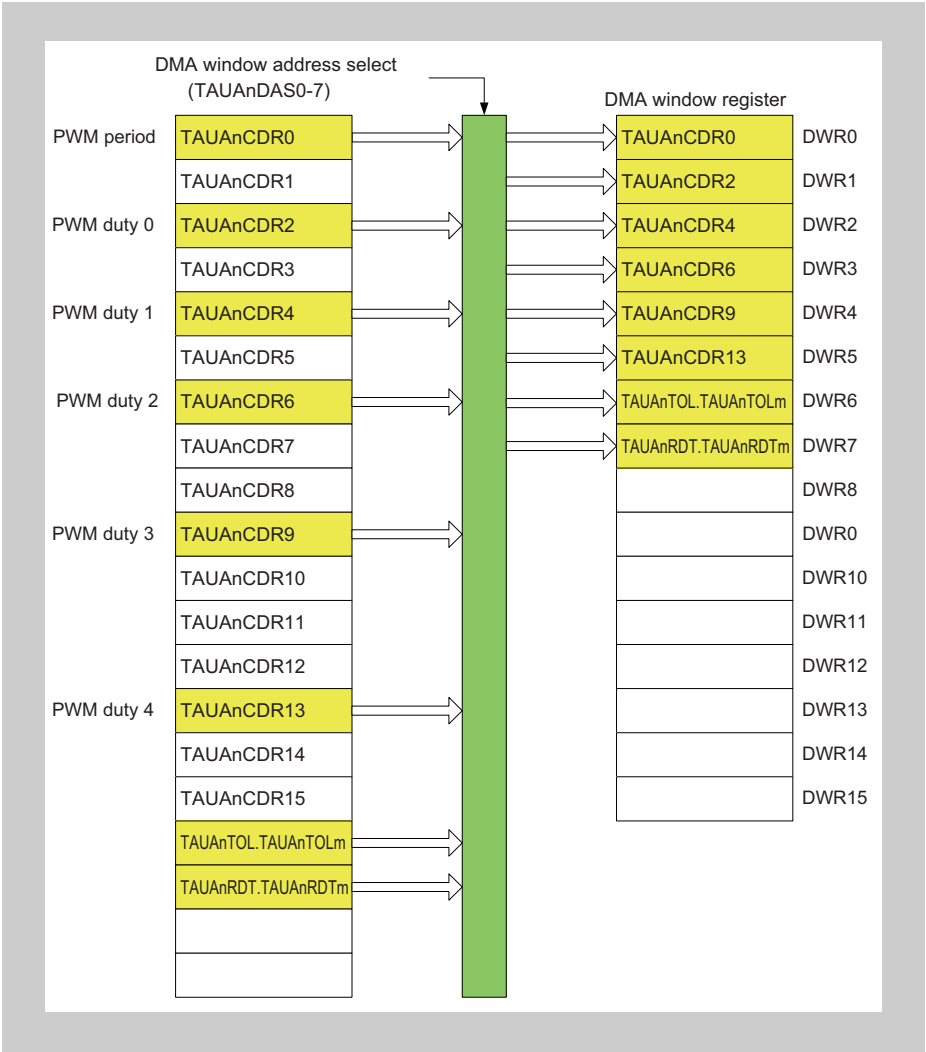


Figure 12-27 Assigning DMA window addresses

Note When using the simultaneous rewrite function for a data register, it is recommended to allocate the TAUAnRDT register at the end of the specified area.

12.14 Independent Channel Operation Functions

The following sections list the independent channel operation functions provided by TAUA. For a general overview of independent channel operation, see *12.3 “Functional Description” on page 557*.

12.15 Independent Channel Interrupt Functions

This chapter describes functions that generate interrupts at regular intervals or with a specified delay.

- *12.15.1 “Interval Timer Function”*
- *12.15.2 “TAUAnTTINm Input Interval Timer Function”*
- *12.15.3 “Delay Count Function”*
- *12.15.4 “One-Pulse Output Function”*

12.15.1 Interval Timer Function

(1) Overview

Summary This function is used as a reference timer for generating timer interrupts (INTTAUAnIm) at regular intervals. When an interrupt is generated, the TAUAnTTOUTm signal toggles, resulting in a square wave.

Prerequisites
- The operation mode must be set to Interval Timer Mode. Refer to *Table 12-11 "TAUAnCMORm settings for Interval Timer Function" on page 600*.
- The channel output mode must be set to Independent Channel Output Mode 1. Refer to *12.8 "Channel Output Modes" on page 577*.

Description The counter is started by setting the channel trigger bit (TAUAnTS.TAUAnTSm) to 1. This in turn sets TAUAnTE.TAUAnTEm = 1, enabling count operation. The current value of TAUAnCDRm is loaded to TAUAnCNTm and the counter starts to count down from this value.

When the counter reaches 0000_H, INTTAUAnIm is generated and the TAUAnTTOUTm signal toggles. Next, TAUAnCNTm loads the value of TAUAnCDRm, and then subsequently continues operation.

The value of TAUAnCDRm can be rewritten at any time, and the changed value of TAUAnCDRm is applied the next time the counter starts to count down.

The counter can be stopped by setting TAUAnTT.TAUAnTTm to 1, which in turn sets TAUAnTE.TAUAnTEm to 0. TAUAnCNTm and TAUAnTTOUTm stop but retain their values. The counter can be reset by setting TAUAnTS.TAUAnTSm to 1. The counter can also be forcibly restarted (without stopping it first) by setting TAUAnTS.TAUAnTSm to 1 during operation.

Conditions If the TAUAnCMORm.TAUAnMD0 bit is set to 0, the first interrupt after a start or restart is not generated, and therefore TAUAnTTOUTm does not toggle. This results in an inverted TAUAnTTOUTm signal compared to when TAUAnCMORm.TAUAnMD0 is set to 1. For details refer to *12.10 "TAUAnTTOUTm Output and INTTAUAnIm Generation when Counter Starts or Restarts" on page 589*.

(2) Equations

INTTAUAnIm cycle = count clock cycle x (TAUAnCDRm + 1)

TAUAnTTOUTm square wave cycle = count clock cycle x (TAUAnCDRm + 1) x 2

Caution The minimum interval timer interrupt cycle is 1/(PCLK/2).

(3) Block diagram and general timing diagram

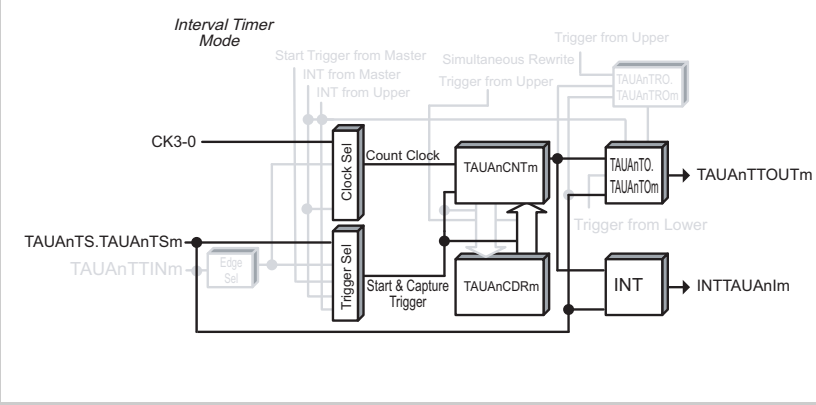


Figure 12-28 Block diagram for Interval Timer Function

The following settings apply to the general timing diagram:

- INTTAUAnIm is generated at operation start (TAUAnCMORm.TAUAnMD0 = 1)

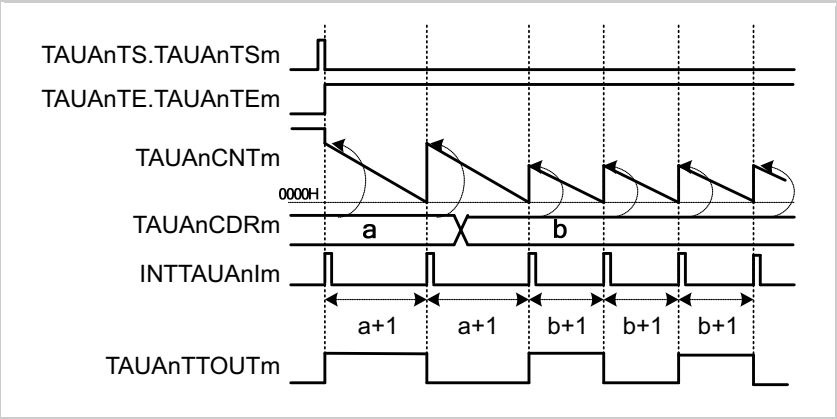


Figure 12-29 General timing diagram for Interval Timer Function

(4) Register settings

(a) TAUAnCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn CKS[1:0]		TAUAn CCS[1:0]		TAUAn MAS	TAUAn STS[2:0]			TAUAn COS[1:0]		-	TAUAn MD[4:1]				TAUAn MD0

Table 12-11 TAUAnCMORm settings for Interval Timer Function

Bit name	Setting
TAUAnCKS[1:0]	00: Operation clock = CK0 01: Operation clock = CK1 10: Operation clock = CK2 11: Operation clock = CK3
TAUAnCCS[1:0]	00: Operation clock is used as the count clock
TAUAnMAS	0: Not used, so set to 0
TAUAnSTS[2:0]	000: Counter triggered by software trigger
TAUAnCOS[1:0]	00: Not used, so set to 00
TAUAnMD[4:1]	0000: Interval Timer Mode
TAUAnMD0	0: INTTAUAnIm not generated and TAUAnTTOUTm does not toggle at operation start or restart 1: Generates INTTAUAnIm and toggles TAUAnTTOUTm at operation start or restart

(b) TAUAnCMURm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-														TAUAnTIS[1:0]	

Table 12-12 TAUAnCMURm settings for Interval Timer Function

Bit name	Setting
TAUAnTIS[1:0]	00: Not used, so set to 00

(c) Channel output mode

Table 12-13 Control bit settings for Independent Channel Output Mode 1

Bit name	Setting
TAUAnTOE.TAUAnTOEm	1: Enables Independent Channel Output Mode
TAUAnTOM.TAUAnTOMm	0: Independent channel output
TAUAnTOC.TAUAnTOCm	0: Operation mode 1 (= Toggle mode if TAUAnTOM.TAUAnTOMm = 0)
TAUAnTOL.TAUAnTOLm	0: Positive logic
TAUAnTDE.TAUAnTDEm	0: Disables dead time operation
TAUAnTDM.TAUAnTDMm	0: When dead time operation is disabled (TAUAnTDE.TAUAnTDEm = 0), set these bits to 0
TAUAnTDL.TAUAnTDLm	
TAUAnTRE.TAUAnTREm	0: Disables real-time output
TAUAnTRO.TAUAnTROm	0: When real-time output is disabled (TAUAnTRE.TAUAnTREm = 0), set these bits to 0
TAUAnTRC.TAUAnTRCm	
TAUAnTME.TAUAnTMEm	0: Disables modulation

Note The channel output mode can also be set to Channel Output Mode Controlled by Software by setting TAUAnTOE.TAUAnTOEm = 0. TAUAnTTOUTm can then be controlled independently of the interrupts. For details refer to *12-9 “Channel output modes” on page 578*.

(d) Simultaneous rewrite

The simultaneous rewrite registers (TAUAnRDE, TAUAnRDS, TAUAnRDM, and TAUAnRDC) cannot be used with the Interval Timer Function. Therefore, these registers must be set to 0.

Table 12-14 Simultaneous rewrite settings for Interval Timer Function

Bit name	Setting
TAUAnRDE.TAUAnRDEm	0: Disables simultaneous rewrite
TAUAnRDS.TAUAnRDSm	0: When simultaneous rewrite is disabled (TAUAnRDE.TAUAnRDEm = 0), set these bits to 0
TAUAnRDM.TAUAnRDMm	
TAUAnRDC.TAUAnRDCm	

(5) Operating procedure for Interval Timer Function

Table 12-15 Operating procedure for Interval Timer Function

	Operation	Status of TAUAn
Initial channel setting	Set the TAUAnCMORm register and TAUAnCMURm registers as described in *Table 12-11 "TAUAnCMORm settings for Interval Timer Function" on page 600* and *Table 12-12 "TAUAnCMURm settings for Interval Timer Function" on page 600* Set the value of the TAUAnCDRm register Set the channel output mode by setting the control bits as described in *Table 12-13 "Control bit settings for Independent Channel Output Mode 1" on page 601*	Channel operation is stopped.
Start operation	Set TAUAnTS.TAUAnTSm to 1. TAUAnTS.TAUAnTSm is a trigger bit, so it is automatically cleared to 0.	TAUAnTE.TAUAnTEm is set to 1 and the counter starts. TAUAnCNTm loads the TAUAnCDRm value. When TAUAnCMORm.TAUAnMD0 = 1, INTTAUAnIm is generated and TAUAnTTOUTm toggles.
During operation	The TAUAnCDRm register value can be changed at any time. The TAUAnCNTm register can be read at all times.	TAUAnCNTm counts down. When the counter reaches 0000_H: • TAUAnCNTm reloads the TAUAnCDRm value, and then continues count operation. • INTTAUAnIm is generated and TAUAnTTOUTm toggles.
Stop operation	Set TAUAnTT.TAUAnTTm to 1. TAUAnTT.TAUAnTTm is a trigger bit, so it is automatically cleared to 0.	TAUAnTE.TAUAnTEm is cleared to 0 and the counter stops. TAUAnCNTm and TAUAnTTOUTm stop and retain their current values.

Restart

(6) Specific timing diagrams

(a) TAUAnCDRm = 0000_H, count clock = PCLK/2

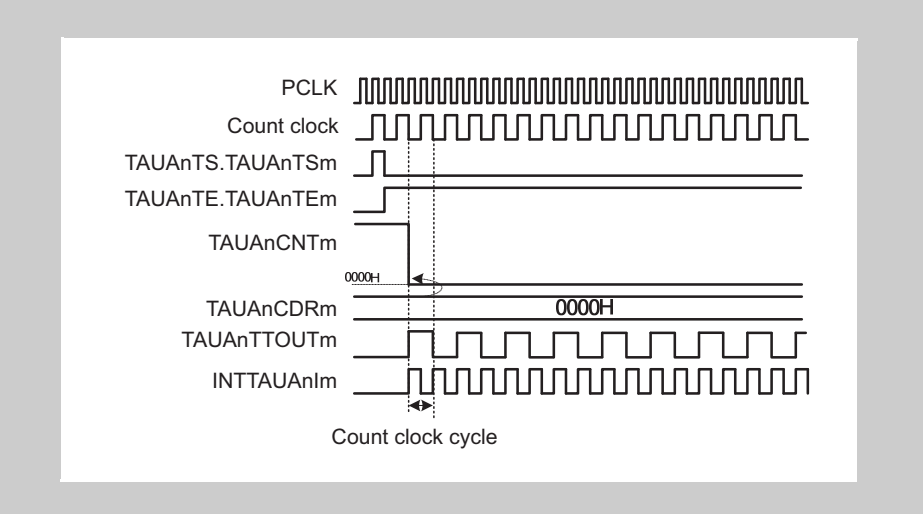


Figure 12-30 TAUAnCDRm = 0000_H, count clock = PCLK/2

- If TAUAnCDRm = 0000_H and the count clock = PCLK/2[1], the TAUAnCDRm value is loaded to TAUAnCNTm every count clock, meaning that TAUAnCNTm is always 0000_H.
- INTTAUAnIm is generated every count clock, resulting in TAUAnTTOUTm toggling every count clock.

(b) TAUAnCDRm = 0000_H, count clock = PCLK

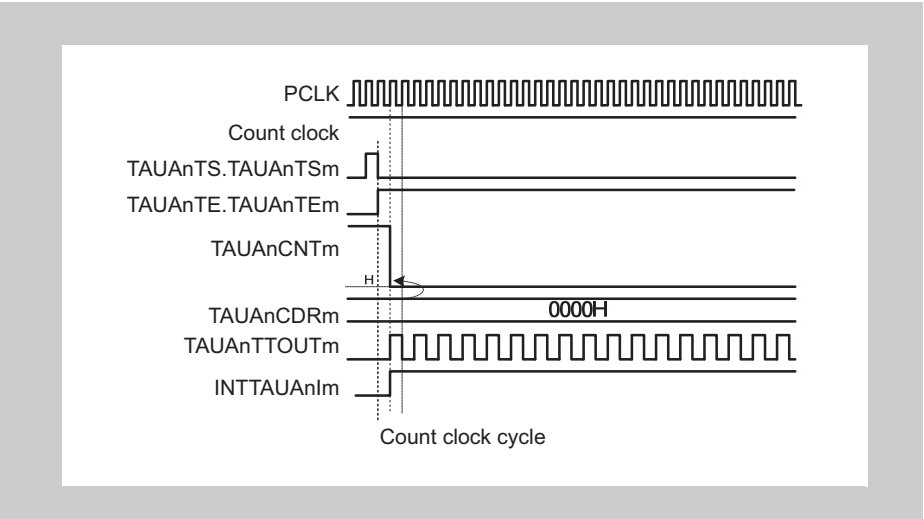


Figure 12-31 TAUAnCDRm = 0000_H, count clock = PCLK

- If TAUAnCDRm = 0000_H and the count clock = PCLK, the TAUAnCDRm value is loaded to TAUAnCNTm every PCLK clock, meaning that TAUAnCNTm is always 0000_H.
- INTTAUAnIm is generated continuously, resulting in TAUAnTTOUTm toggling every PCLK clock.

(c) Operation stop and restart

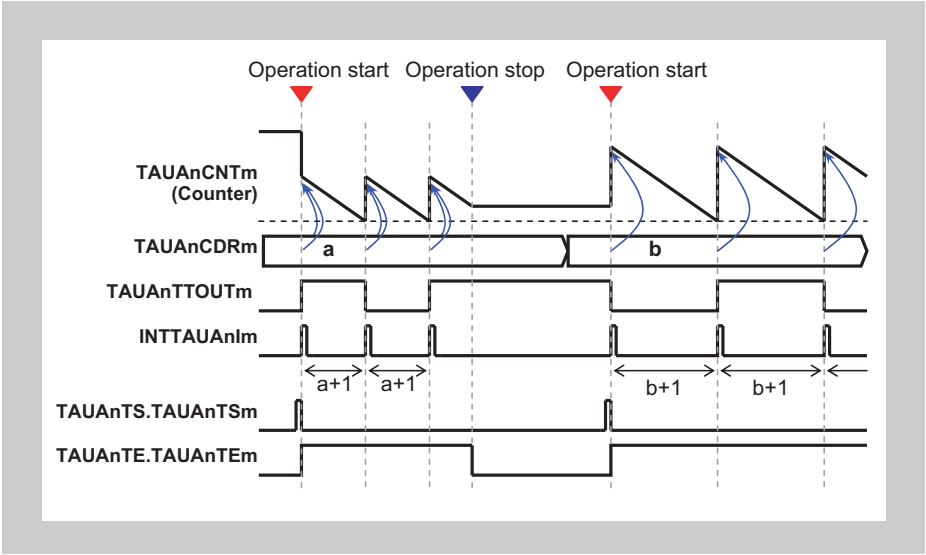


Figure 12-32 Operation stop and restart, TAUAnCMORm.TAUAnMD0 = 1

- The counter can be stopped by setting TAUAnTT.TAUAnTTm to 1, which in turn sets TAUAnTE.TAUAnTEm to 0.
- TAUAnCNTm and TAUAnTTOUTm stop but retain their values.
- The counter can be restarted by setting TAUAnTS.TAUAnTSm to 1.

(d) Forced restart

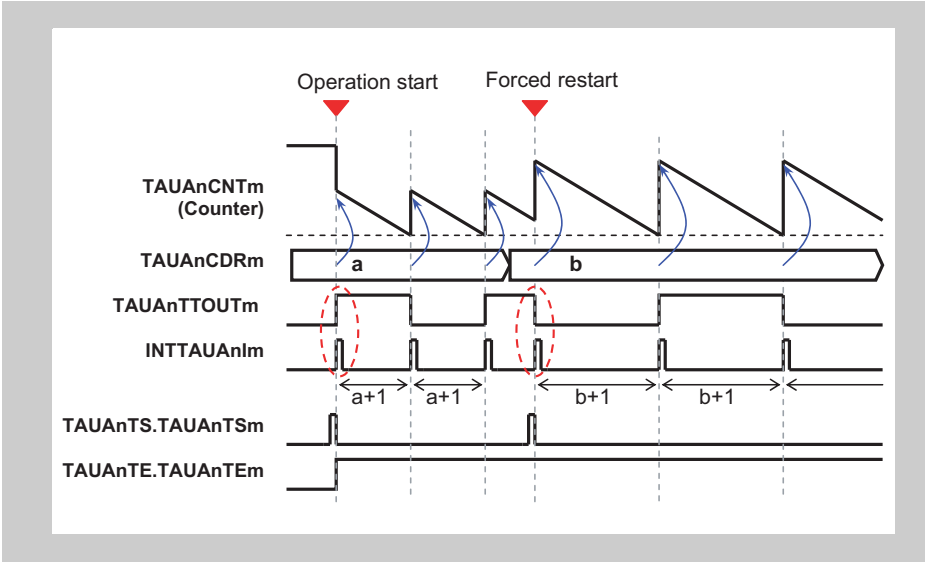


Figure 12-33 Forced restart operation, TAUAnCMORm.TAUAnMD0 = 1

- The counter can be forcibly restarted (without stopping it first) by setting TAUAnTS.TAUAnTSm to 1 during operation.
- If the TAUAnCMORm.TAUAnMD0 bit is set to 1, the first interrupt after a start or restart is generated.

12.15.2 TAUAnTTINm Input Interval Timer Function

(1) Overview

Summary This function is used as a reference timer for generating timer interrupts (INTTAUAnIm) at regular intervals or when a valid TAUAnTTINm input edge is detected. When an interrupt is generated, the TAUAnTTOUTm signal toggles, resulting in a square wave.

Prerequisites

- The operation mode must be set to Interval Timer Mode. Refer to *Table 12-16 "TAUAnCMORm settings for TAUAnTTINm Input Interval Timer Function" on page 607*.
- The channel output mode must be set to Independent Channel Output Mode 1. Refer to *12.8 "Channel Output Modes" on page 577*.

Description This function operates in an identical manner to the Interval Timer Function (see *12.15.1 "Interval Timer Function" on page 598*), except that this function is restarted by a valid TAUAnTTINm input edge. The type of edge used as the trigger is specified using the TAUAnCMURm.TAUAnTIS[1:0] bits. Either rising edge, falling edge, or rising and falling edge can be selected.

(2) Equations

INTTAUAnIm cycle = count clock cycle x (TAUAnCDRm + 1)

TAUAnTTOUTm square wave cycle = count clock cycle x (TAUAnCDRm + 1) x 2

(3) Block diagram and general timing diagram

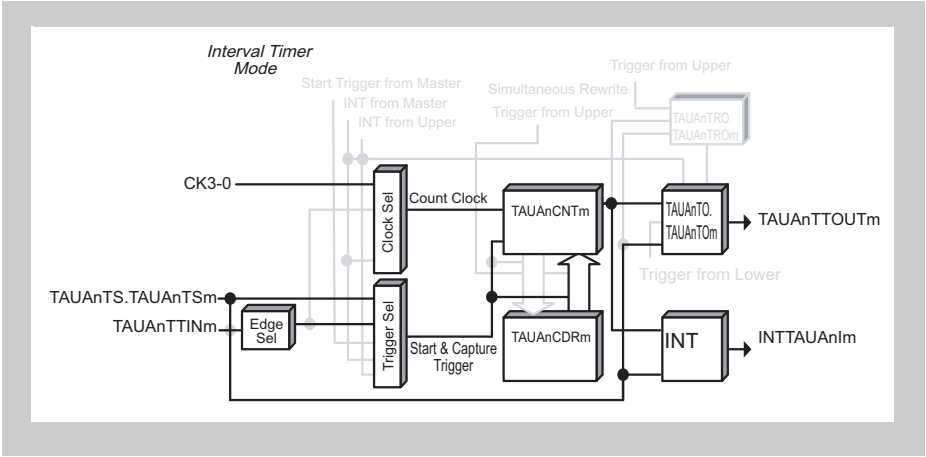


Figure 12-34 Block diagram for TAUAnTTINm Input Interval Timer Function

The following settings apply to the general timing diagram:

- INTTAUAnIm is generated at operation start (TAUAnCMORm.TAUAnMD0 = 1)
- Rising edge detection (TAUAnCMURm.TAUAnTIS[1:0] = 01_B)

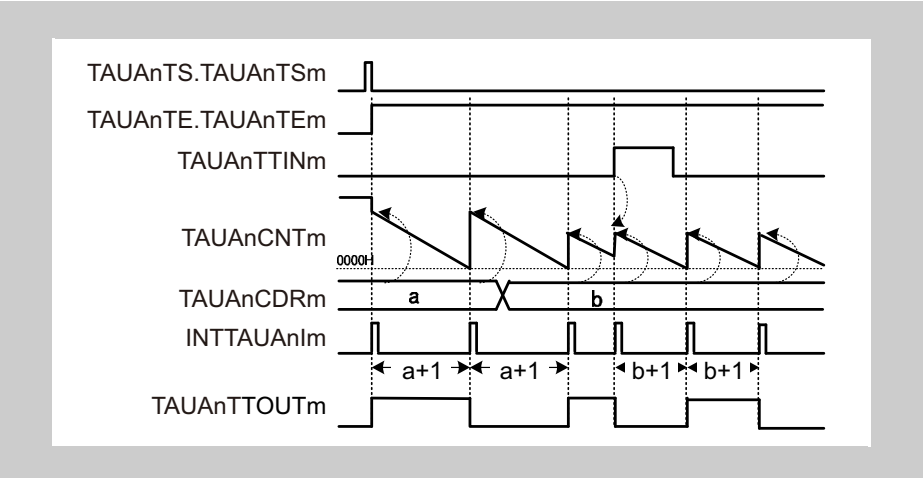


Figure 12-35 General timing diagram for TAUAnTTINm Input Interval Timer Function

(4) Register settings

(a) TAUAnCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn CKS[1:0]		TAUAn CCS[1:0]		TAUAn MAS	TAUAn STS[2:0]			TAUAn COS[1:0]		-	TAUAnMD[4:1]				TAUAn MD0

Table 12-16 TAUAnCMORm settings for TAUAnTTINm Input Interval Timer Function

Bit name	Setting
TAUAnCKS[1:0]	00: Operation clock = CK0 01: Operation clock = CK1 10: Operation clock = CK2 11: Operation clock = CK3
TAUAnCCS[1:0]	00: Operation clock is used as the count clock
TAUAnMAS	0: Not used, so set to 0
TAUAnSTS[2:0]	001: Valid TAUAnTTINm input edge signal is used as the external start trigger
TAUAnCOS[1:0]	00: Not used, so set to 00
TAUAnMD[4:1]	0000: Interval Timer Mode
TAUAnMD0	0: INTTAUAnIm not generated and TAUAnTTOUTm does not toggle at operation start 1: Generates INTTAUAnIm and toggles TAUAnTTOUTm at operation start

(b) TAUAnCMURm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-														TAUAnTIS[1:0]	

Table 12-17 TAUAnCMURm settings for TAUAnTTINm Input Interval Timer Function

Bit name	Setting
TAUAnTIS[1:0]	00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection

(c) Channel output mode

Table 12-18 Control bit settings for Independent Channel Output Mode 1

Bit name	Setting
TAUAnTOE.TAUAnTOEm	1: Enables Independent Channel Output Mode
TAUAnTOM.TAUAnTOMm	0: Independent channel output
TAUAnTOC.TAUAnTOCm	0: Operation mode 1 (= Toggle mode if TAUAnTOM.TAUAnTOMm = 0)
TAUAnTOL.TAUAnTOLm	0: Positive logic
TAUAnTDE.TAUAnTDEm	0: Disables dead time operation
TAUAnTDM.TAUAnTDMm	0: When dead time operation is disabled (TAUAnTDE.TAUAnTDEm = 0), set these bits to 0
TAUAnTDL.TAUAnTDLm	
TAUAnTRE.TAUAnTREm	0: Disables real-time output
TAUAnTRO.TAUAnTROm	0: When real-time output is disabled (TAUAnTRE.TAUAnTREm = 0), set these bits to 0
TAUAnTRC.TAUAnTRCm	
TAUAnTME.TAUAnTMEm	0: Disables modulation

Note The channel output mode can also be set to Channel Output Mode Controlled by Software by setting TAUAnTOE.TAUAnTOEm = 0. TAUAnTTOUTm can then be controlled independently of the interrupts. For details refer to *12-9 "Channel output modes" on page 578*.

(d) Simultaneous rewrite

The simultaneous rewrite registers (TAUAnRDE, TAUAnRDS, TAUAnRDM, and TAUAnRDC) cannot be used with the TAUAnTTINm Input Interval Timer Function. Therefore, these registers must be set to 0.

Table 12-19 Simultaneous rewrite settings for TAUAnTTINm Input Interval Timer Function

Bit name	Setting
TAUAnRDE.TAUAnRDEm	0: Disables simultaneous rewrite
TAUAnRDS.TAUAnRDSm	0: When simultaneous rewrite is disabled (TAUAnRDE.TAUAnRDEm = 0), set these bits to 0
TAUAnRDM.TAUAnRDMm	
TAUAnRDC.TAUAnRDCm	

(5) Operating procedure for TAUAnTTINm Input Interval Timer Function

Table 12-20 Operating procedure for TAUAnTTINm Input Interval Timer Function

	Operation	Status of TAUAn
Initial channel setting	Set the TAUAnCMORm register and TAUAnCMURm registers as described in *Table 12-16 "TAUAnCMORm settings for TAUAnTTINm Input Interval Timer Function" on page 607* and *Table 12-17 "TAUAnCMURm settings for TAUAnTTINm Input Interval Timer Function" on page 607* Set the value of the TAUAnCDRm register Set the channel output mode by setting the control bits as described in *Table 12-18 "Control bit settings for Independent Channel Output Mode 1" on page 608*	Channel operation is stopped.
Start operation	Set TAUAnTS.TAUAnTAUAnTSm to 1. TAUAnTS.TAUAnTAUAnTSm is a trigger bit, so it is automatically cleared to 0.	TAUAnTE.TAUAnTEm is set to 1 and the counter starts. TAUAnCNTm loads the TAUAnCDRm value. When TAUAnCMORm.TAUAnMD0 = 1, INTTAUAnIm is generated and TAUAnTTOUTm toggles.
During operation	The values of the TAUAnCMURm.TAUAnTAUAnTIS[1:0] bits and the TAUAnCDRm register can be changed at any time. The TAUAnCNTm register can be read at all times. Detection of TAUAnTTINm edge	TAUAnCNTm counts down. When the counter reaches 0000_H: • TAUAnCNTm reloads the TAUAnCDRm value, and then continues count operation. • INTTAUAnIm is generated and TAUAnTTOUTm toggles. When a TAUAnTTINm input valid edge is detected during count operation, the TAUAnCDRm value and continues count operation continues. Afterwards, this procedure is repeated.
Stop operation	Set TAUAnTT.TAUAnTTm to 1. TAUAnTT.TAUAnTTm is a trigger bit, so it is automatically cleared to 0.	TAUAnTE.TAUAnTEm is cleared to 0 and the counter stops. TAUAnCNTm and TAUAnTTOUTm stop and retain their current values.

Restart

(6) Specific timing diagrams

The timing diagrams in *12.15.1 "Interval Timer Function" on page 598* also apply, except for this function the counter can also be restarted by a valid TAUAnTTINm input edge.

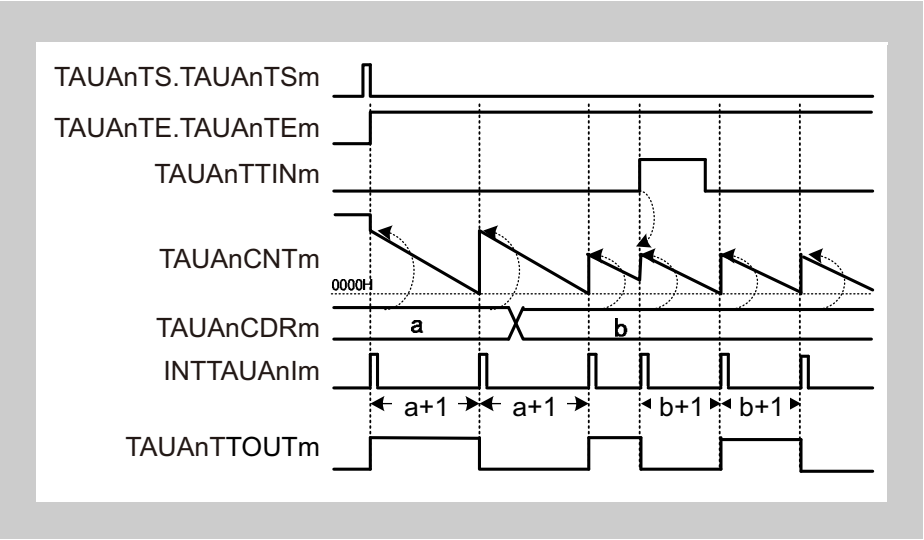


Figure 12-36 Counter triggered by rising TAUAnTTINm input edge (TAUAnCMURm.TAUAnTIS[1:0] = 01_B), TAUAnCMORm.TAUAnMD0 = 1

- If a valid TAUAnTTINm input edge is detected, an interrupt is generated which causes TAUAnTTOUTm to toggle. In this example, the valid edge is a rising edge (TAUAnCMURm.TAUAnTIS[1:0] = 01_B).

12.15.3 Delay Count Function

(1) Overview

Summary This function generates interrupts (INTTAUAnIm), which have a defined delay to the TAUAnTTINm input signal. TAUAnTTINm input signal pulses that occur within the delay period are ignored.

Prerequisites
- The operation mode must be set to One Count Mode. Refer to *Table 12-21 "TAUAnCMORm settings for Delay Count Function" on page 613*.
- TAUAnTTOUTm is not used for this function.
- The start trigger must be disabled during counting (TAUAnCMORn.TAUAnMD0 = 0).

Description The counter is enabled by setting the channel trigger bit (TAUAnTS.TAUAnTSm) to 1. This in turn sets TAUAnTE.TAUAnTEm = 1, enabling count operation.

The counter starts when a valid TAUAnTTINm input start edge is detected. The value of TAUAnCDRm is loaded to TAUAnCNTm and the counter starts to count down from the TAUAnCDRm value.

When the counter reaches 0000_H an interrupt is generated. The counter returns to $FFFF_H$ and awaits the next valid TAUAnTTINm input edge.

When the counter is counting down, further TAUAnTTINm input signals are ignored, i.e., the counter does not reset.

The value of TAUAnCDRm can be rewritten at any time, and the changed value of TAUAnCDRm is applied the next time the counter starts to count down.

Conditions The type of edge used as the trigger is specified by the TAUAnCMURm.TAUAnTIS[1:0] bits:
- If TAUAnCMURm.TAUAnTIS[1:0] = 00_B, falling edges trigger the counter.
- If TAUAnCMURm.TAUAnTIS[1:0] = 01_B, rising edges trigger the counter.
- If TAUAnCMURm.TAUAnTIS[1:0] = 10_B, rising and falling edges trigger the counter.

(2) Equations

Delay between TAUAnTTINm and INTTAUAnIm =

count clock cycle × (TAUAnCDRm + 1)

(3) Block diagram and general timing diagram

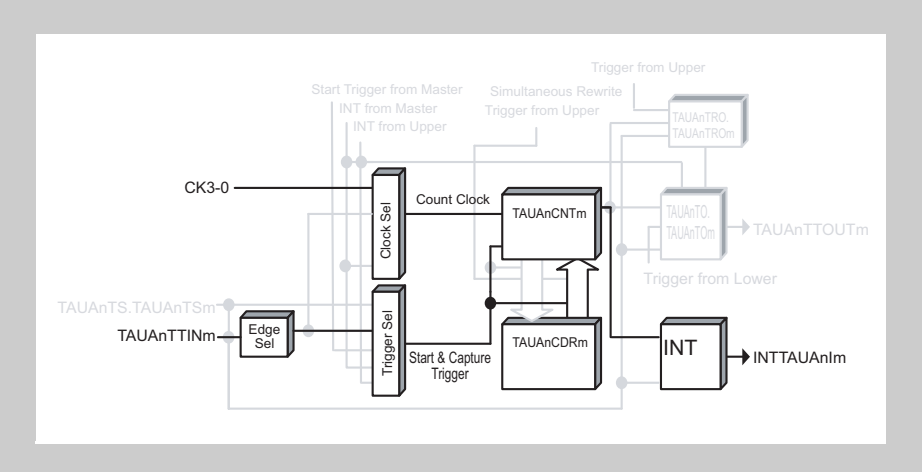


Figure 12-37 Block diagram for Delay Count Function

The following settings apply to the general timing diagram:

- Falling edge detection (TAUAnCMURm.TAUAnTIS[1:0] = 00_B)

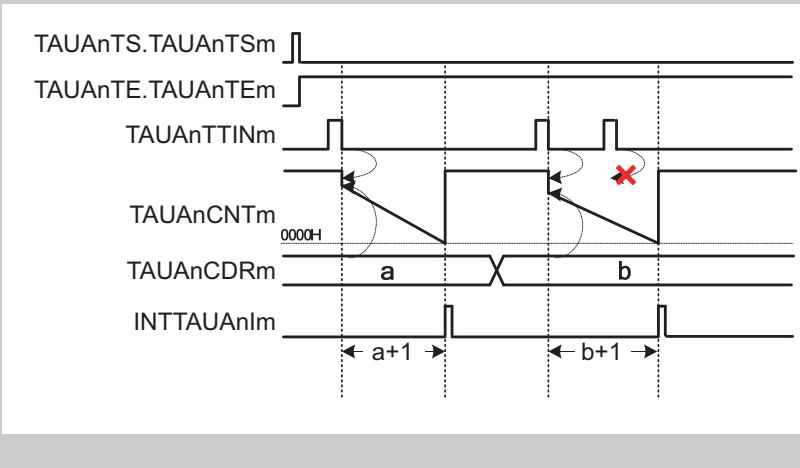


Figure 12-38 General timing diagram for Delay Count Function

(4) Register settings

(a) TAUAnCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn CKS[1:0]		TAUAn CCS[1:0]		TAUAn MAS	TAUAnSTS[2:0]			TAUAn COS[1:0]		-	TAUAn MD[4:1]				TAUAn MD0

Table 12-21 TAUAnCMORm settings for Delay Count Function

Bit name	Setting
TAUAnCKS[1:0]	00: Operation clock = CK0 01: Operation clock = CK1 10: Operation clock = CK2 11: Operation clock = CK3
TAUAnCCS[1:0]	00: Operation clock is used as the count clock
TAUAnMAS	0: Not used, so set to 0
TAUAnSTS[2:0]	001: Valid TAUAnTTINm input edge signal is used as the external start trigger
TAUAnCOS[1:0]	00: Not used, so set to 00
TAUAnMD[4:1]	0100: One Count Mode
TAUAnMD0	0: Disables the start trigger during operation

(b) TAUAnCMURm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-														TAUAnTIS[1:0]	

Table 12-22 TAUAnCMURm settings for Delay Count Function

Bit name	Setting
TAUAnTIS[1:0]	00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection (low width measurement)

(c) Channel output mode

Because the channel output mode is not used by this function, clear TAUAnTOE.TAUAnTOEm. However, the channel output mode can be used in Independent Channel Output Mode Controlled by Software.

(d) Simultaneous rewrite

The simultaneous rewrite registers (TAUAnRDE, TAUAnRDS, TAUAnRDM, and TAUAnRDC) cannot be used with the Delay Count Function. Therefore, these registers must be set to 0.

Table 12-23 Simultaneous rewrite settings for Delay Count Function

Bit name	Setting
TAUAnRDEm	0: Disables simultaneous rewrite
TAUAnRDSm	0: When simultaneous rewrite is disabled (TAUAnRDE.TAUAnRDEm = 0), set these bits to 0
TAUAnRDMm	
TAUAnRDCm	

(5) Operating procedure for Delay Count Function

Table 12-24 Operating procedure for Delay Count Function

	Operation	Status of TAUAn
Initial channel setting	Set the TAUAnCMORm register and TAUAnCMURm registers as described in *Table 12-21 "TAUAnCMORm settings for Delay Count Function" on page 613* and *Table 12-22 "TAUAnCMURm settings for Delay Count Function" on page 613* Set the value of the TAUAnCDRm register	Channel operation is stopped.
Start operation	Set TAUAnTS.TAUAnTSm to 1. TAUAnTS.TAUAnTSm is a trigger bit, so it is automatically cleared to 0. Detection of TAUAnTTINm start edge	TAUAnTE.TAUAnTEm is set to 1 and TAUAnCNTm waits for detection of the TAUAnTTINm start edge. When a start edge is detected, TAUAnCNTm loads the TAUAnCDRm value.
During operation	The values set in the TAUAnCDRm register can be changed at any time. The TAUAnCNTm register can be read at all times.	TAUAnCNTm counts down. When the counter reaches 0000_H: INTTAUAnIm is generated. TAUAnCNTm stops counting, returns to $FFFF_H$, and waits for a trigger. If a trigger occurs while TAUAnCNTm is counting, the trigger is ignored. Afterwards, this procedure is repeated.
Stop operation	Set TAUAnTT.TAUAnTTm to 1. TAUAnTT.TAUAnTTm is a trigger bit, so it is automatically cleared to 0.	TAUAnTE.TAUAnTEm is cleared to 0 and the counter stops. TAUAnCNTm stops and retains its value.

Restart

12.15.4 One-Pulse Output Function

(1) Overview

Summary This function generates an interrupt (INTTAUAnIm) when a valid TAUAnTTINm input edge is detected and also a specific interval later. TAUAnTTINm input signal pulses that occur within the defined interval are ignored. When an interrupt is generated, the TAUAnTTOUTm signal toggles, resulting in a square wave.

Prerequisites

- The operation mode must be set to Pulse One Count Mode. Refer to *Table 12-25 "TAUAnCMORm settings for One-Pulse Output Function" on page 618*
- The channel output mode must be set to Independent Channel Output Mode 1. Refer to *12.8 "Channel Output Modes" on page 577*.
- Trigger detection must be disabled during counting (TAUAnCMORn.TAUAnMD0 = 0).

Description The counter is enabled by setting the channel trigger bit (TAUAnTS.TAUAnTSm) to 1. This in turn sets TAUAnTE.TAUAnTEm = 1, enabling count operation.

The counter starts when a valid TAUAnTTINm input edge is detected. The value of TAUAnCDRm is loaded to TAUAnCNTm and the counter starts to count down from the TAUAnCDRm value. An interrupt is generated and TAUAnTTOUTm becomes active.

When the counter reaches 0001_H an interrupt is generated and TAUAnTTOUTm becomes inactive. The counter stops at 0000_H and awaits the next valid TAUAnTTINm input edge.

When the counter is counting down, further TAUAnTTINm input signals are ignored, i.e. the counter does not reset.

The value of TAUAnCDRm can be rewritten at any time, and the changed value of TAUAnCDRm is applied the next time the counter starts to count down.

Conditions The type of edge used as the trigger is specified by the TAUAnCMURm.TAUAnTIS[1:0] bits:

- If TAUAnCMURm.TAUAnTIS[1:0] = 00_B, falling edges trigger the counter.
- If TAUAnCMURm.TAUAnTIS[1:0] = 01_B, rising edges trigger the counter.
- If TAUAnCMURm.TAUAnTIS[1:0] = 10_B, rising and falling edges trigger the counter.

(2) Equations

Interval between TAUAnTTINm and INTTAUAnIm = TAUAnTTOUTm (timer output) width = count clock cycle × TAUAnCDRm

(3) Block diagram and general timing diagram

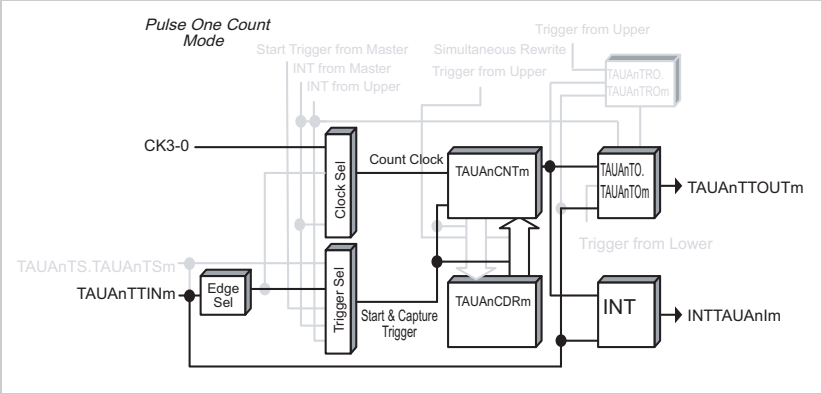


Figure 12-39 Block diagram for One-Pulse Output Function

The following settings apply to the general timing diagram:

- Falling edge detection (TAUAnCMURm.TAUAnTIS[1:0] = 00_B)

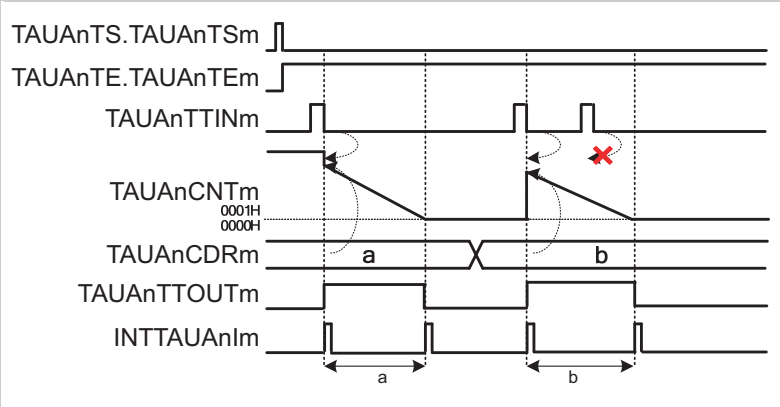


Figure 12-40 General timing diagram for One-Pulse Output Function

(4) Register settings

(a) TAUAnCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn CKS[1:0]		TAUAn CCS[1:0]		TAUAn MAS	TAUAn STS[2:0]			TAUAn COS[1:0]		-	TAUAn MD[4:1]				TAUAn MD0

Table 12-25 TAUAnCMORm settings for One-Pulse Output Function

Bit name	Setting
TAUAnCKS[1:0]	00: Operation clock = CK0 01: Operation clock = CK1 10: Operation clock = CK2 11: Operation clock = CK3
TAUAnCCS[1:0]	00: Operation clock is used as the count clock
TAUAnMAS	0: Not used, so set to 0
TAUAnSTS[2:0]	001: Valid TAUAnTTINm input edge signal is used as the external start trigger
TAUAnCOS[1:0]	00: Not used, so set to 00
TAUAnMD[4:1]	1010: Pulse One Count Mode
TAUAnMD0	0: Disables the start trigger during operation

(b) TAUAnCMURm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-															TAUAnTIS[1:0]

Table 12-26 TAUAnCMURm settings for One-Pulse Output Function

Bit name	Setting
TAUAnTIS[1:0]	00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection (low width measurement)

(c) Channel output mode

Table 12-27 Control bit settings for Independent Channel Output Mode 1

Bit name	Setting
TAUAnTOEm	1: Enables Independent Channel Output Mode
TAUAnTOMm	0: Independent channel output
TAUAnTOCm	1: Set/reset mode
TAUAnTOLm	0: Positive logic 1: Inverted logic
TAUAnTDEm	0: Disables dead time operation
TAUAnTDMm	0: When dead time operation is disabled (TAUAnTDE.TAUAnTDEm = 0), set these bits to 0
TAUAnTDLm	
TAUAnTREm	0: Disables real-time output
TAUAnTROm	0: When real-time output is disabled (TAUAnTRE.TAUAnTREm = 0), set these bits to 0
TAUAnTRCm	
TAUAnTMEm	0: Disables modulation

Note The channel output mode can also be set to Channel Output Mode Controlled by Software by setting TAUAnTOE.TAUAnTOEm = 0. TAUAnTTOUTm can then be controlled independently of the interrupts. For details refer to *Table 12-9 “Channel output modes” on page 578*.

(d) Simultaneous rewrite

The simultaneous rewrite registers (TAUAnRDE, TAUAnRDS, TAUAnRDM, and TAUAnRDC) cannot be used with the One-Pulse Output Function. Therefore, these registers must be set to 0.

Table 12-28 Simultaneous rewrite settings for One-Pulse Output Function

Bit name	Setting
TAUAnRDEm	0: Disables simultaneous rewrite
TAUAnRDSm	0: When simultaneous rewrite is disabled (TAUAnRDE.TAUAnRDEm = 0), set these bits to 0
TAUAnRDMm	
TAUAnRDCm	

(5) Operating procedure for One-Pulse Output Function

Table 12-29 Operating procedure for One-Pulse Output Function

	Operation	Status of TAUAn
Initial channel setting	Set the TAUAnCMORm register and TAUAnCMURm registers as described in *Table 12-25 "TAUAnCMORm settings for One-Pulse Output Function" on page 618* and *Table 12-26 "TAUAnCMURm settings for One-Pulse Output Function" on page 618* Set the value of the TAUAnCDRm register Set the channel output mode by setting the control bits as described in *Table 12-27 "Control bit settings for Independent Channel Output Mode 1" on page 619*	Channel operation is stopped.
Start operation	Set TAUAnTS.TAUAnTSm to 1. TAUAnTS.TAUAnTSm is a trigger bit, so it is automatically cleared to 0. Detection of TAUAnTTINm start edge	TAUAnTE.TAUAnTEm is set to 1 and TAUAnCNTm waits for detection of the TAUAnTTINm start edge. When a start edge is detected, TAUAnCNTm loads the TAUAnCDRm value.
During operation	The value of TAUAnCDRm can be changed at any time. The TAUAnCNTm register can be read at all times.	INTTAUAnIm is generated when TAUAnCNTm starts and TAUAnTTOUTm is set to its active level. TAUAnCNTm counts down. When the counter reaches 0001_H: • INTTAUAnIm is generated. • TAUAnTTOUTm is set to its inactive level. TAUAnCNTm stops counting and waits for a trigger. If a trigger occurs while TAUAnCNTm is counting, the trigger is ignored. Afterwards, this procedure is repeated. Afterwards, this procedure is repeated.
Stop operation	Set TAUAnTT.TAUAnTTm to 1. TAUAnTT.TAUAnTTm is a trigger bit, so it is automatically cleared to 0.	TAUAnTE.TAUAnTEm is cleared to 0 and the counter stops. TAUAnCNTm and TAUAnTTOUTm stop and retain their current values.

Restart

12.16 Independent Channel Signal Measurement Functions

This chapter describes functions that measure the widths of an individual TAUAnTTINm pulse or the total width of successive TAUAnTTINm pulses. It also describes functions that measure the interval of the signal or that compare the width of a pulse with a reference value.

- *12.16.1 “TAUAnTTINm Input Pulse Interval Measurement Function”*
- *12.16.2 “TAUAnTTINm Input Signal Width Measurement Function”*
- *12.16.3 “Overflow Interrupt Output Function (During TAUAnTTINm Width Measurement)”*
- *12.16.4 “TAUAnTTINm Input Period Count Detection Function”*
- *12.16.5 “Overflow Interrupt Output Function (During TAUAnTTINm Input Period Count Detection)”*
- *12.16.6 “TAUAnTTINm Input Pulse Interval Judgment Function”*
- *12.16.7 “TAUAnTTINm Input Signal Width Judgment Function”*

12.16.1 TAUAnTTINm Input Pulse Interval Measurement Function

(1) Overview

Summary This function captures the count value and uses this value and the overflow bit TAUAnCSRm.TAUAnOVF to measure the interval of the TAUAnTTINm input signal.

Prerequisites
- The operation mode must be set to Capture Mode. Refer to *Table 12-31 "TAUAnCMORm settings for TAUAnTTINm Input Pulse Interval Measurement Function" on page 624*.
- TAUAnTTOUTm is not used for this function.

Description The counter is started by setting the channel trigger bit (TAUAnTS.TAUAnTSm) to 1. This in turn sets TAUAnTE.TAUAnTEm = 1, enabling count operation. The counter TAUAnCNTm starts counting up from 0000_H. When a valid TAUAnTTINm edge is detected, the value of TAUAnCNTm is captured, transferred to TAUAnCDRm, and an interrupt INTTAUAnIm is generated. The counter resets to 0000_H and subsequently continues operation.

If the counter reaches $FFFF_H$ before a valid TAUAnTTINm edge is detected, it overflows. The counter is reset to 0000_H and subsequently continues operation. The values transferred to TAUAnCDRm and TAUAnCSRm.TAUAnOVF respectively depend on the values of bits TAUAnCMORm.TAUAnCOS[1:0]:

Table 12-30 Effects of an overflow

TAUAnCMORm. TAUAnCOS[1:0]	When overflow occurs		When a valid TAUAnTTINm input is then detected	
	TAUAnCDRm	TAUAnCSRm. TAUAnOVF	TAUAnCDRm and TAUAnCNTm	TAUAnCSRm. TAUAnOVF
00	Unchanged	0	TAUAnCNTm loaded to TAUAnCDRm	1
01		1		
10	Set to $FFFF_H$	0	TAUAnCNTm set to 0, TAUAnCDRm unchanged	0
11		1		

When TAUAnCMORm.TAUAnCOS[0] is 1, the overflow bit TAUAnCSRm.TAUAnOVF can only be cleared by setting TAUAnCSCm.TAUAnCLOV to 1.

The combination of the value of TAUAnCDRm and TAUAnCSRm.TAUAnOVF can be used to deduce the interval of the TAUAnTTINm signal. However, if an overflow occurs multiple times before a valid TAUAnTTINm input is detected, the overflow bit TAUAnCSRm.TAUAnOVF cannot indicate this.

The function can be stopped by setting TAUAnTT.TAUAnTTm = 1, which in turn sets TAUAnTE.TAUAnTEm = 0. TAUAnCNTm stops but retains its value. While the function is stopped, TAUAnTTINm input valid edge detection and TAUAnCNTm capture are not performed.

The function can be restarted by setting TAUAnTS.TAUAnTSm = 1. The counter is reset to 0000_H and subsequently continues operation. The counter can also be forcibly restarted (without stopping it first) by setting TAUAnTS.TAUAnTSm = 1 during operation.

Conditions If the TAUAnCMORm.TAUAnMD0 bit is set to 0, the first interrupt after a start or restart is not generated. For details refer to *12.10 "TAUAnTTOUTm Output and INTTAUAnIm Generation when Counter Starts or Restarts" on page 589*.

Note When TAUAnCMORm.TAUAnCOS[1] = 1, the value of TAUAnCNTm is *not* loaded to TAUAnCDRm when the first valid TAUAnTTINm input edge occurs after an overflow. However, an interrupt is generated.

(2) Equations

TAUAnTTINm input pulse interval = count clock cycle x [(TAUAnCSRm.TAUAnOVF x ($FFFF_H$ + 1)) + TAUAnCDRm capture value + 1]

(3) Block diagram and general timing diagram

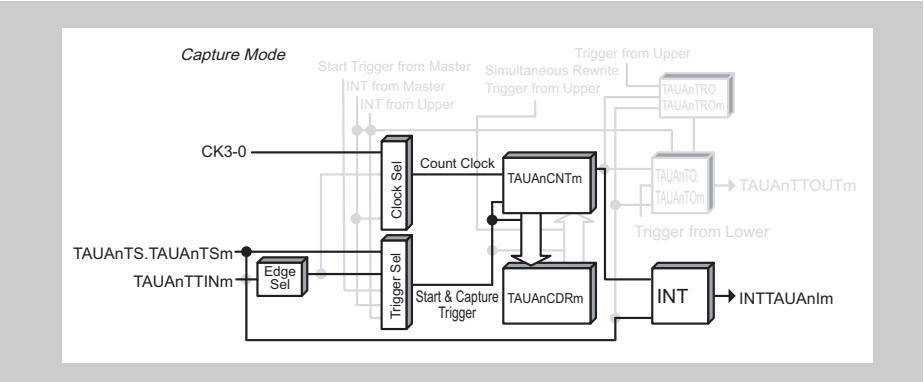


Figure 12-41 Block diagram for TAUAnTTINm Input Pulse Interval Measurement Function

The following settings apply to the general timing diagram:

- INTTAUAnIm not generated at operation start (TAUAnCMORm.TAUAnMD0 = 0)
- Falling edge detection (TAUAnCMURm.TAUAnTIS[1:0] = 00_B)
- When a valid TAUAnTTINm input is detected after an overflow TAUAnCDRm is changed and TAUAnCSRm.TAUAnOVF is set to 1 (TAUAnCMORm.TAUAnCOS[1:0] = 00_B)

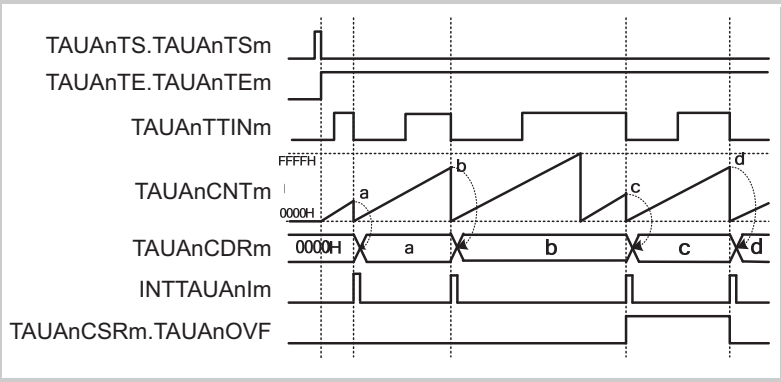


Figure 12-42 General timing diagram for TAUAnTTINm Input Pulse Interval Measurement Function

(4) Register settings

(a) TAUAnCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn CKS[1:0]		TAUAn CCS[1:0]		TAUAn MAS	TAUAnSTS[2:0]			TAUAn COS[1:0]		-	TAUAnMD[4:1]				TAUAn MD0

Table 12-31 TAUAnCMORm settings for TAUAnTTINm Input Pulse Interval Measurement Function

Bit name	Setting
TAUAnCKS[1:0]	00: Operation clock = CK0 01: Operation clock = CK1 10: Operation clock = CK2 11: Operation clock = CK3
TAUAnCCS[1:0]	00: Operation clock is used as the count clock
TAUAnMAS	0: Not used, so set to 0
TAUAnSTS[2:0]	001: Valid edge of the TAUAnTTINm input signal is the external capture trigger
TAUAnCOS[1:0]	See *Table 12-30 “Effects of an overflow” on page 622*.
TAUAnMD[4:1]	0010: Capture Mode
TAUAnMD0	0: INTTAUAnIm not generated at operation start 1: Generates INTTAUAnIm at operation start

(b) TAUAnCMURm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-														TAUAnTIS[1:0]	

Table 12-32 TAUAnCMURm settings for TAUAnTTINm Input Pulse Interval Measurement Function

Bit name	Setting
TAUAnTIS[1:0]	00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection

(c) Channel output mode

Because the channel output mode is not used by this function, clear TAUAnTOE.TAUAnTOEm. However, the channel output mode can be used in Independent Channel Output Mode Controlled by Software.

(d) Simultaneous rewrite

The simultaneous rewrite registers (TAUAnRDE, TAUAnRDS, TAUAnRDM, and TAUAnRDC) cannot be used with the TAUAnTTINm Input Pulse Interval Measurement Function. Therefore, these registers must be set to 0.

Table 12-33 Simultaneous rewrite settings for TAUAnTTINm Input Pulse Interval Measurement Function

Bit name	Setting
TAUAnRDE.TAUAnRDEm	0: Disables simultaneous rewrite
TAUAnRDS.TAUAnRDSm	0: When simultaneous rewrite is disabled (TAUAnRDE.TAUAnRDEm = 0), set these bits to 0
TAUAnRDM.TAUAnRDMm	
TAUAnRDC.TAUAnRDCm	

(5) Operating procedure for TAUAnTTINm Input Pulse Interval Measurement Function

Table 12-34 Operating procedure for TAUAnTTINm Input Pulse Interval Measurement Function

	Operation	Status of TAUAn
Initial channel setting	Set the TAUAnCMORm register and TAUAnCMURm registers as described in *Table 12-31 "TAUAnCMORm settings for TAUAnTTINm Input Pulse Interval Measurement Function" on page 624* and *Table 12-32 "TAUAnCMURm settings for TAUAnTTINm Input Pulse Interval Measurement Function" on page 624* Set the value of the TAUAnCDRm register	Channel operation is stopped.
Start operation	Set TAUAnTS.TAUAnTSm to 1. TAUAnTS.TAUAnTSm is a trigger bit, so it is automatically cleared to 0.	TAUAnTE.TAUAnTEm is set to 1 and the counter starts. TAUAnCNTm is cleared to 0000_H. INTTAUAnIm is generated when TAUAnCMORm.TAUAnMD0 is set to 1.
During operation	Detection of TAUAnTTINm edges. The TAUAnCMURm.TAUAnTIS[1:0] bits can be changed at any time. The TAUAnCDRm and TAUAnCSRm registers can be read at any time. Writing 1 to the TAUAnCSCm.TAUAnCLOV bit is possible. (The TAUAnCSRm.TAUAnOVF bit is cleared to 0.)	TAUAnCNTm starts to count up from 0000_H. When a TAUAnTTINm valid edge is detected: • TAUAnCNTm transfers (captures) its value to TAUAnCDRm, and returns to 0000_H. • INTTAUAnIm is then generated. Afterwards, this procedure is repeated.
Stop operation	Set TAUAnTT.TAUAnTTm to 1. TAUAnTT.TAUAnTTm is a trigger bit, so it is automatically cleared to 0.	TAUAnTE.TAUAnTEm is cleared to 0 and the counter stops. TAUAnCNTm stops and both it and TAUAnCSRm.TAUAnOVF retain their current values.

Restart

(6) Specific timing diagrams: overflow behavior

(a) TAUAnCMORm.TAUAnCOS[1:0] = 00_B

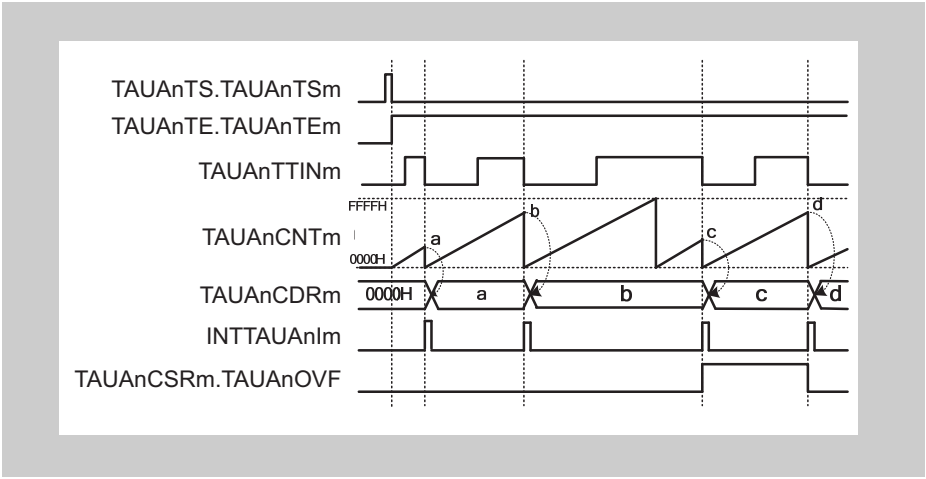


Figure 12-43 TAUAnCMORm.TAUAnCOS[1:0] = 00_B, TAUAnCMORm.TAUAnMD0 = 0, TAUAnCMURm.TAUAnTIS[1:0] = 00_B

- When an overflow occurs, the value of TAUAnCDRm remains unchanged and TAUAnCSRm.TAUAnOVF remains = 0.
- Upon detection of the next valid TAUAnTTINm input edge, the value of TAUAnCNTm is loaded to TAUAnCDRm and TAUAnCSRm.TAUAnOVF is set to 1.
- Upon detecting the next valid TAUAnTTINm input edge while no overflow has occurred, TAUAnCSRm.TAUAnOVF is cleared to 0.

(b) TAUAnCMORm.TAUAnCOS[1:0] = 01_B

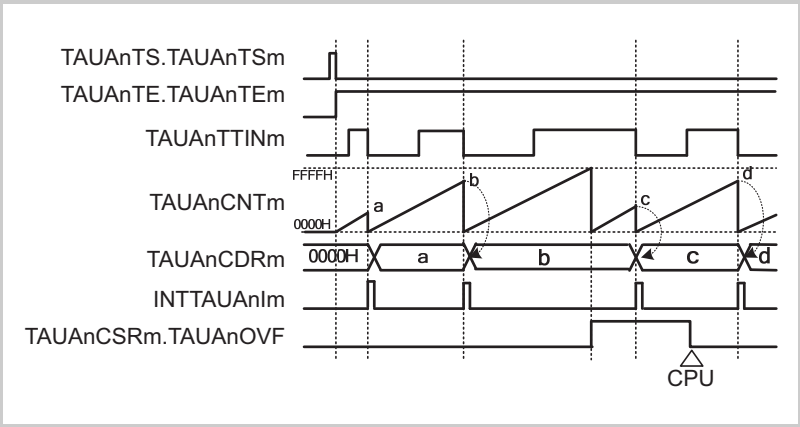


Figure 12-44 TAUAnCMORm.TAUAnCOS[1:0] = 01_B, TAUAnCMORm.TAUAnMD0 = 0, TAUAnCMURm.TAUAnTIS[1:0] = 00_B

- When an overflow occurs, the value of TAUAnCDRm remains unchanged and TAUAnCSRm.TAUAnOVF is set to 1.
- Upon detection of the next valid TAUAnTTINm input edge, the value of TAUAnCNTm is loaded to TAUAnCDRm.
- TAUAnCSRm.TAUAnOVF is only cleared by a CPU command (setting the TAUAnCSCm.TAUAnCLOV bit to 1).

(c) TAUAnCMORm.TAUAnCOS[1:0] = 10_B

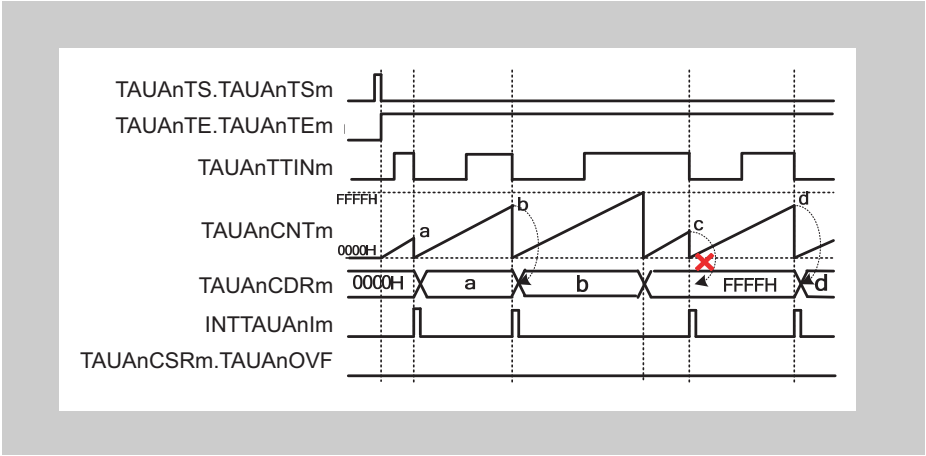


Figure 12-45 TAUAnCMORm.TAUAnCOS[1:0] = 10_B, TAUAnCMORm.TAUAnMD0 = 0, TAUAnCMURm.TAUAnTIS[1:0] = 00_B

- When an overflow occurs, TAUAnCDRm is set to $FFFF_H$ and TAUAnCSRm.TAUAnOVF remains = 0.
- Upon detection of the next valid TAUAnTTINm input edge, TAUAnCNTm is reset to 0, but TAUAnCDRm and TAUAnCSRm.TAUAnOVF remain unchanged.
- Thus, the next TAUAnTTINm input valid edge after the overflow is ignored.

(d) TAUAnCMORm.TAUAnCOS[1:0] = 11_B

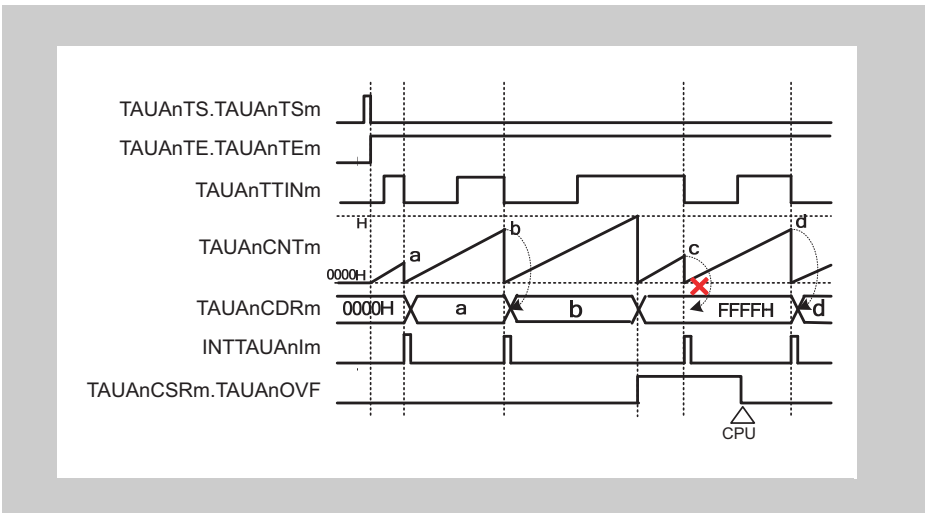


Figure 12-46 TAUAnCMORm.TAUAnCOS[1:0] = 11_B, TAUAnCMORm.TAUAnMD0 = 0, TAUAnCMURm.TAUAnTIS[1:0] = 00_B

- When an overflow occurs, TAUAnCDRm is set to $FFFF_H$, and TAUAnCSRm.TAUAnOVF is set to 1.
- Upon detection of the next valid TAUAnTTINm input edge, TAUAnCNTm is reset to 0, but TAUAnCDRm and TAUAnCSRm.TAUAnOVF remain unchanged.
- Thus, the next TAUAnTTINm input valid edge after the overflow is ignored.
- TAUAnCSRm.TAUAnOVF is cleared by setting TAUAnCSCm.TAUAnCLOV = 1.

12.16.2 TAUAnTTINm Input Signal Width Measurement Function

(1) Overview

Summary This function measures the width of a TAUAnTTINm input signal.

Prerequisites

- The operation mode must be set to Capture & One Count Mode. Refer to *Table 12-36 "TAUAnCMORm settings for TAUAnTTINm Input Signal Width Measurement Function" on page 631*.
- TAUAnTTOUTm is not used for this function.
- TAUAnCMORm.TAUAnMD0 must be set to 0.

Description The counter is started by setting the channel trigger bit (TAUAnTS.TAUAnTSm) to 1. This in turn sets TAUAnTE.TAUAnTEm = 1, enabling count operation. When a valid TAUAnTTINm start edge is detected, the counter TAUAnCNTm starts counting up from 0000_H. When a valid TAUAnTTINm stop edge is detected, the value of TAUAnCNTm is captured, transferred to TAUAnCDRm, and an interrupt INTTAUAnIm is generated. The counter retains its value and awaits the next valid TAUAnTTINm input start edge.

If the counter reaches $FFFF_H$ before a valid TAUAnTTINm stop edge is detected, it overflows. The counter is reset to 0000_H and subsequently continues operation. The values transferred to TAUAnCDRm and TAUAnCSRm.TAUAnOVF respectively depend on the values of bits TAUAnCMORm.TAUAnCOS[1:0].

Table 12-35 Effects of an overflow

TAUAnCMORm.COS[1:0]	When overflow occurs		When a valid TAUAnTTINm input stop edge is detected	
	TAUAnCDRm	TAUAnCSRm.TAUAnOVF	TAUAnCDRm and TAUAnCNTm	TAUAnCSRm.TAUAnOVF
00	Unchanged	0	TAUAnCNTm loaded to TAUAnCDRm	1
01		1		
10	Set to $FFFF_H$	0	TAUAnCNTm stops counting TAUAnCDRm unchanged	0
11		1		

When TAUAnCMORm.TAUAnCOS[0] is 1, the overflow bit TAUAnCSRm.TAUAnOVF can only be cleared by setting TAUAnCSCm.TAUAnCLOV to 1.

The combination of the value of TAUAnCDRm and TAUAnCSRm.TAUAnOVF can be used to deduce the width of the TAUAnTTINm signal. However, if an overflow occurs multiple times before a valid TAUAnTTINm input is detected, the overflow bit TAUAnCSRm.TAUAnOVF cannot indicate this.

This function cannot be forcibly restarted.

Note When TAUAnCMORm.TAUAnCOS[2] = 1_B, the value of TAUAnCNTm is *not* loaded to TAUAnCDRm when the first valid TAUAnTTINm input edge occurs after an overflow. However, an interrupt is generated.

(2) Equations

TAUAnTTINm input signal width = count clock cycle x [(TAUAnCSRm.TAUAnOVF x ($FFFF_H$ + 1)) + TAUAnCDRm capture value + 1]

(3) Block diagram and general timing diagram

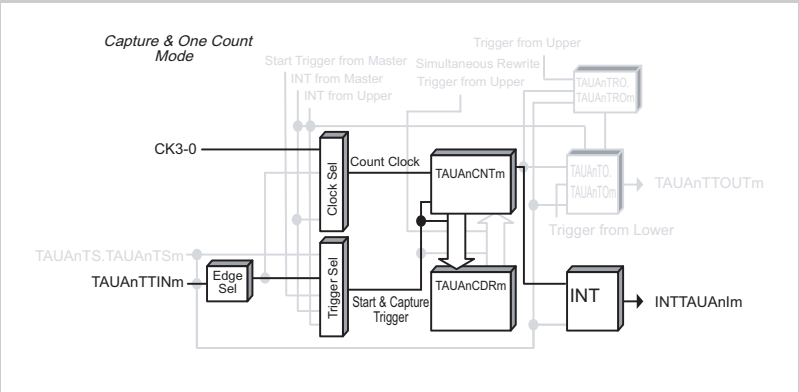


Figure 12-47 Block diagram for TAUAnTTINm Input Signal Width Measurement Function

The following settings apply to the general timing diagram:

- Rising and falling edge detection = high width measurement (TAUAnCMURm.TAUAnTIS[1:0] = 11_B)
- When a valid TAUAnTTINm input is detected after an overflow TAUAnCDRm is changed and TAUAnCSRm.TAUAnOVF is set to 1 (TAUAnCMORm.TAUAnCOS[1:0] = 00_B)

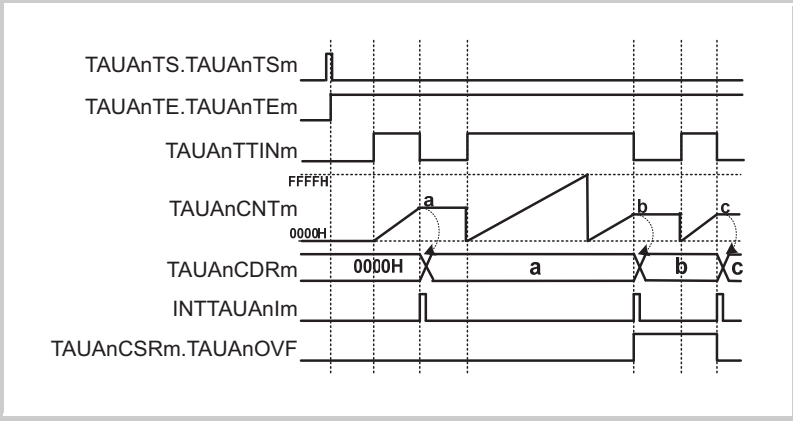


Figure 12-48 General timing diagram for TAUAnTTINm Input Signal Width Measurement Function

(4) Register settings

(a) TAUAnCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn CKS[1:0]		TAUAn CCS[1:0]		TAUAn MAS	TAUAnSTS[2:0]			TAUAn COS[1:0]		-	TAUAnMD[4:1]				TAUAn MD0

Table 12-36 TAUAnCMORm settings for TAUAnTTINm Input Signal Width Measurement Function

Bit name	Setting
TAUAnCKS[1:0]	00: Operation clock = CK0 01: Operation clock = CK1 10: Operation clock = CK2 11: Operation clock = CK3
TAUAnCCS[1:0]	00: Operation clock is used as the count clock
TAUAnMAS	0: Not used, so set to 0
TAUAnSTS[2:0]	010: Valid edge of the TAUAnTTINm input signal is the external start trigger and the reverse edge is the stop trigger
TAUAnCOS[1:0]	See *Table 12-35 "Effects of an overflow" on page 629*.
TAUAnMD[4:1]	0110: Capture & One Count Mode
TAUAnMD0	0: Disables the start trigger during operation

(b) TAUAnCMURm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-														TAUAnTIS[1:0]	

Table 12-37 TAUAnCMURm settings for TAUAnTTINm Input Signal Width Measurement Function

Bit name	Setting
TAUAnTIS[1:0]	10:Rising and falling edge detection (low width measurement) 11:Rising and falling edge detection (high width measurement)

(c) Channel output mode

Because the channel output mode is not used by this function, clear TAUAnTOE.TAUAnTOEm. However, it can be used in Independent Channel Output Mode Controlled by Software.

(d) Simultaneous rewrite

The simultaneous rewrite registers (TAUAnRDE, TAUAnRDS, TAUAnRDM, and TAUAnRDC) cannot be used with the TAUAnTTINm Input Signal Width Measurement Function. Therefore, these registers must be set to 0.

Table 12-38 Simultaneous rewrite settings for TAUAnTTINm Input Signal Width Measurement Function

Bit name	Setting
TAUAnRDE.TAUAnRDEm	0: Disables simultaneous rewrite
TAUAnRDS.TAUAnRDSm	0: When simultaneous rewrite is disabled (TAUAnRDE.TAUAnRDEm = 0), set these bits to 0
TAUAnRDM.TAUAnRDMm	
TAUAnRDC.TAUAnRDCm	

(5) Operating procedure for TAUAnTTINm Input Signal Width Measurement Function

Table 12-39 Operating procedure for TAUAnTTINm Input Signal Width Measurement Function

	Operation	Status of TAUAn
Initial channel setting	Set the TAUAnCMORm register and TAUAnCMURm registers as described in *Table 12-36 "TAUAnCMORm settings for TAUAnTTINm Input Signal Width Measurement Function" on page 631* and *Table 12-37 "TAUAnCMURm settings for TAUAnTTINm Input Signal Width Measurement Function" on page 631* Set the value of the TAUAnCDRm register	Channel operation is stopped.
Start operation	Set TAUAnTS.TAUAnTSm to 1. TAUAnTS.TAUAnTSm is a trigger bit, so it is automatically cleared to 0.	TAUAnTE.TAUAnTEm is set to 1 and TAUAnCNTm waits for detection of the TAUAnTTINm start edge. When a TAUAnTTINm start is detected, TAUAnCNTm starts to count up.
During operation	Detection of TAUAnTTINm edges. The TAUAnCDRm, TAUAnCNTm, and TAUAnCSRm registers can be read at any time. The TAUAnCSC.CLOV bit can be set to 1.	TAUAnCNTm starts to count up from 0000_H. When a TAUAnTTINm valid edge is detected: • TAUAnCNTm transfers (captures) its value to TAUAnCDRm, and retains its value. • INTTAUAnIm is then generated. Afterwards, this procedure is repeated.
Stop operation	Set TAUAnTT.TAUAnTTm to 1. TAUAnTT.TAUAnTTm is a trigger bit, so it is automatically cleared to 0.	TAUAnTE.TAUAnTEm is cleared to 0 and the counter stops. TAUAnCNTm stops and both it and TAUAnCSRm.TAUAnOVF retain their current values.

Restart

(6) Specific timing diagrams: overflow behavior

(a) TAUAnCMORm.TAUAnCOS[1:0] = 00_B

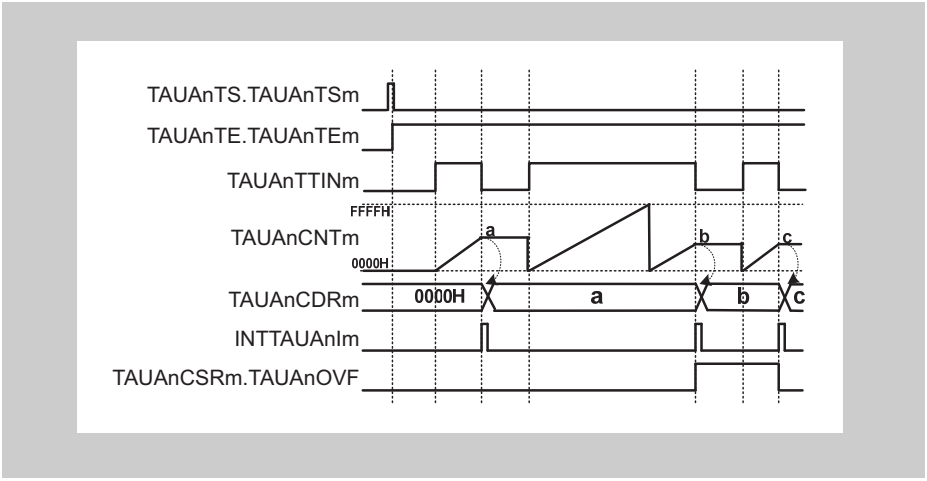


Figure 12-49 TAUAnCMORm.TAUAnCOS[1:0] = 00_B, TAUAnCMORm.TAUAnMD0 = 0, TAUAnCMURm.TAUAnTIS[1:0] = 11_B

- When an overflow occurs, the value of TAUAnCDRm remains unchanged and TAUAnCSRm.TAUAnOVF remains = 0.
- Upon detection of the next valid TAUAnTTINm input edge, the value of TAUAnCNTm is loaded to TAUAnCDRm and TAUAnCSRm.TAUAnOVF is set to 1.
- Upon detecting the next valid TAUAnTTINm input edge while no overflow has occurred, TAUAnCSRm.TAUAnOVF is cleared to 0.

(b) TAUAnCMORm.TAUAnCOS[1:0] = 01_B

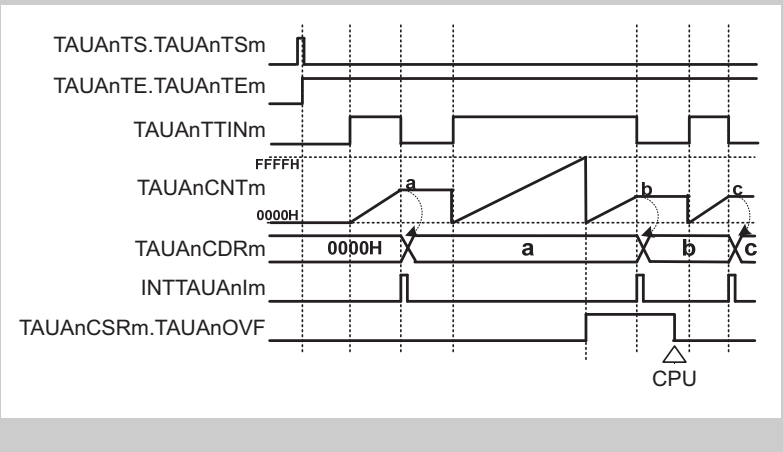


Figure 12-50 TAUAnCMORm.TAUAnCOS[1:0] = 01_B, TAUAnCMORm.TAUAnMD0 = 0, TAUAnCMURm.TAUAnTIS[1:0] = 11_B

- When an overflow occurs, the value of TAUAnCDRm remains unchanged and TAUAnCSRm.TAUAnOVF is set to 1.
- Upon detection of the next valid TAUAnTTINm input edge, the value of TAUAnCNTm is loaded to TAUAnCDRm.
- TAUAnCSRm.TAUAnOVF is only cleared by a CPU command (setting TAUAnCSCm.TAUAnCLOV = 1).

(c) TAUAnCMORm.TAUAnCOS[1:0] = 10_B

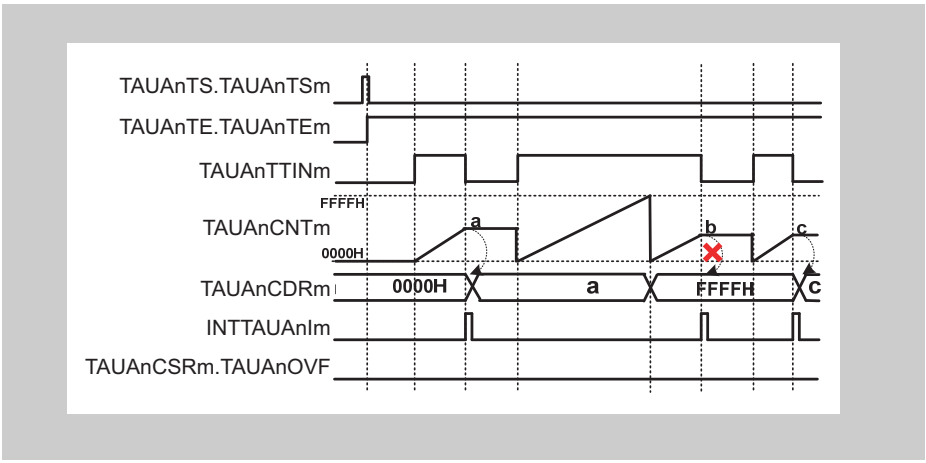


Figure 12-51 TAUAnCMORm.TAUAnCOS[1:0] = 10_B, TAUAnCMORm.TAUAnMD0 = 0, TAUAnCMURm.TAUAnTIS[1:0] = 11_B

- When an overflow occurs, TAUAnCDRm is set to $FFFF_H$ and TAUAnCSRm.TAUAnOVF remains = 0.
- Upon detection of the next valid TAUAnTTINm input edge, TAUAnCNTm is reset to 0, but TAUAnCDRm and TAUAnCSRm.TAUAnOVFTAUAnCSRm.TAUAnOVF remain unchanged.
- Thus, the next TAUAnTTINm input valid edge after the overflow is ignored.

(d) TAUAnCMORm.TAUAnCOS[1:0] = 11_B

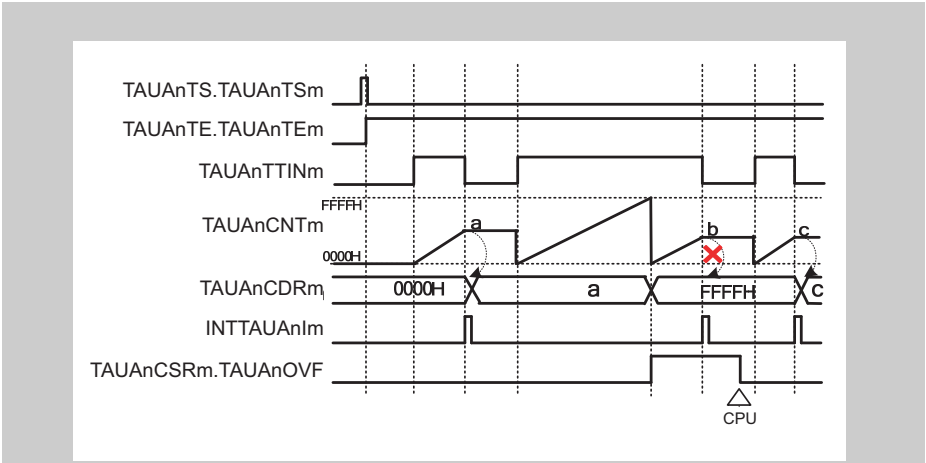


Figure 12-52 TAUAnCMORm.TAUAnCOS[1:0] = 11_B, TAUAnCMORm.TAUAnMD0 = 0, TAUAnCMURm.TAUAnTIS[1:0] = 11_B

- When an overflow occurs, TAUAnCDRm is set to $FFFF_H$, and TAUAnCSRm.TAUAnOVF is set to 1.
- Upon detection of the next valid TAUAnTTINm input edge, TAUAnCNTm is reset to 0, but TAUAnCDRm and TAUAnCSRm.TAUAnOVF remain unchanged.
- Thus, the next TAUAnTTINm input valid edge after the overflow is ignored.
- TAUAnCSRm.TAUAnOVF is cleared by setting TAUAnCSCm.TAUAnCLOV = 1.

12.16.3 Overflow Interrupt Output Function (During TAUAnTTINm Width Measurement)

(1) Overview

Summary This function measures the width of an individual TAUAnTTINm input signal. An interrupt is generated if (FFFFH + 1) is exceeded following TAUAnTTINm input.

Prerequisites

- The operation mode must be set to One Count Mode. Refer to *Table 12-40 "TAUAnCMORm settings for Overflow Interrupt Output Function (During TAUAnTTINm Width Measurement)" on page 638*.
- TAUAnTTOUTm is not used for this function.
- The value of TAUAnCDRm must be set to $FFFF_H$.

Description The counter is enabled by setting the channel trigger bit (TAUAnTS.TAUAnTSm) to 1. This in turn sets TAUAnTE.TAUAnTEm = 1, enabling count operation.

The counter starts when a valid TAUAnTTINm input start edge is detected. $FFFF_H$ is loaded to TAUAnCNTm and the counter starts to count down.

When a valid stop edge is detected, the counter stops and retains the current value.

When the next TAUAnTTINm input start edge is detected, TAUAnCNTm loads $FFFF_H$ and starts to count down.

If the counter reaches 0000_H before a stop edge is detected, an interrupt is generated.

Conditions The valid start and stop edges are specified by the TAUAnCMURm.TAUAnTIS[1:0] bits:

- If TAUAnCMURm.TAUAnTIS[1:0] = 10_B, the TAUAnTTINm input low width is measured. The start trigger is a falling edge and the stop trigger is a rising edge.
- If TAUAnCMURm.TAUAnTIS[1:0] = 11_B, the TAUAnTTINm input high width is measured. The start trigger is a rising edge and the stop trigger is a falling edge.

Note The counter cannot be restarted during operation.

(2) Block diagram and general timing diagram

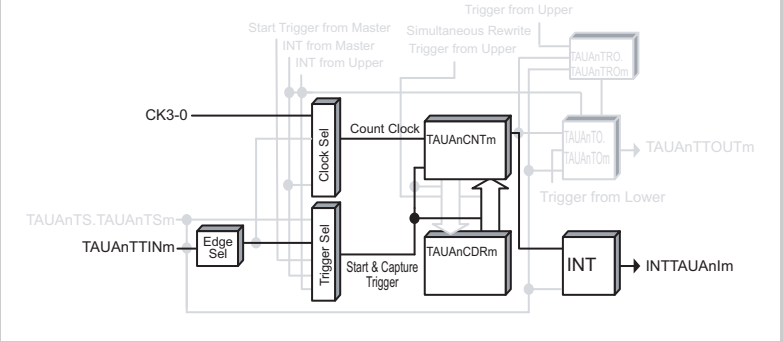


Figure 12-53 Block diagram for Overflow Interrupt Output Function (During TAUAnTTINm Width Measurement)

The following settings apply to the general timing diagram:

- Rising and falling edge detection = high width measurement (TAUAnCMURm.TAUAnTIS[1:0] = 11_B)

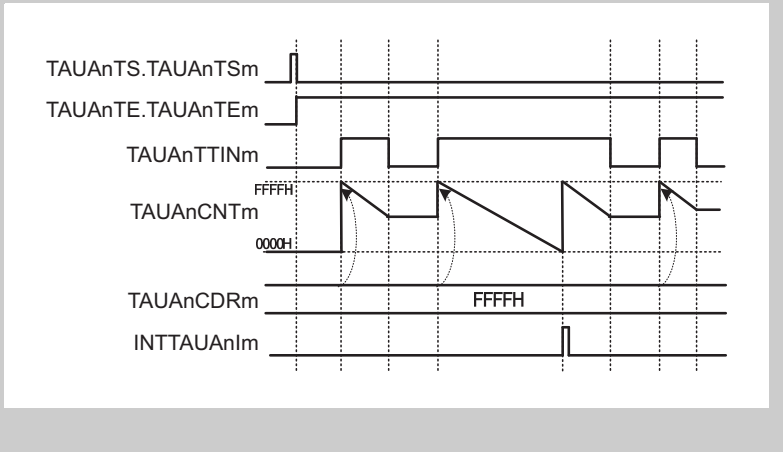


Figure 12-54 General timing diagram for Overflow Interrupt Output Function (During TAUAnTTINm Width Measurement)

(3) Register settings

(a) TAUAnCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn CKS[1:0]		TAUAn CCS[1:0]		TAUAn MAS	TAUAnSTS[2:0]			TAUAn COS[1:0]		-	TAUAnMD[4:1]				TAUAn MD0

Table 12-40 TAUAnCMORm settings for Overflow Interrupt Output Function (During TAUAnTTINm Width Measurement)

Bit name	Setting
TAUAnCKS[1:0]	00: Operation clock = CK0 01: Operation clock = CK1 10: Operation clock = CK2 11: Operation clock = CK3
TAUAnCCS[1:0]	00: Operation clock is used as the count clock
TAUAnMAS	0: Not used, so set to 0
TAUAnSTS[2:0]	010: Valid edge of the TAUAnTTINm input signal is the external start trigger and the reverse edge is the stop trigger
TAUAnCOS[1:0]	00: Not used, so set to 00
TAUAnMD[4:1]	0100: One Count Mode
TAUAnMD0	0: Disables the start trigger during operation

(b) TAUAnCMURm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-														TIS[1:0]	

Table 12-41 TAUAnCMURm settings Overflow Interrupt Output Function (During TAUAnTTINm Width Measurement)

Bit name	Setting
TIS[1:0]	10: Rising and falling edge detection (Low width measurement) 11: Rising and falling edge detection (High width measurement)

(c) Channel output mode

Because channel output mode is not used by this function, TAUAnTOE.TAUAnTOEm is cleared. However, it can be used in Independent Channel Output Mode Controlled by Software.

(d) Simultaneous rewrite

The simultaneous rewrite registers (TAUAnRDE, TAUAnRDS, TAUAnRDM, and TAUAnRDC) cannot be used with the Overflow Interrupt Output Function (During TAUAnTTINm Width Measurement). Therefore, these registers must be set to 0.

(4) Operating procedure for Overflow Interrupt Output Function

Table 12-42 Simultaneous rewrite settings for Overflow Interrupt Output Function (During TAUAnTTINm Width Measurement)

Bit name	Setting
TAUAnRDEm	0: Disables simultaneous rewrite
TAUAnRDSm	0: When simultaneous rewrite is disabled (TAUAnRDE.TAUAnRDEm = 0), set these bits to 0
TAUAnRDMm	
TAUAnRDCm	

(During TAUAnTTINm Width Measurement)

Table 12-43 Operating procedure for Overflow Interrupt Output Function (During TAUAnTTINm Width Measurement)

	Operation	Status of TAUAn
Initial channel setting	Set the TAUAnCMORm register and TAUAnCMURm registers as described in *Table 12-40 "TAUAnCMORm settings for Overflow Interrupt Output Function (During TAUAnTTINm Width Measurement)" on page 638* and *Table 12-41 "TAUAnCMURm settings Overflow Interrupt Output Function (During TAUAnTTINm Width Measurement)" on page 638* Set the value of the TAUAnCDRm register	Channel operation is stopped.
Start operation	Set TAUAnTS.TAUAnTSm to 1. TAUAnTS.TAUAnTSm is a trigger bit, so it is automatically cleared to 0. Detection of TAUAnTTINm start edge	TAUAnTE.TAUAnTEm is set to 1 and TAUAnCNTm waits for detection of the start edge. When a start edge is detected, TAUAnCNTm loads the TAUAnCDRm value ($FFFF_H$).
During operation	The TAUAnCNTm register can be read at any time. Detection of TAUAnTTINm edges.	TAUAnCNTm counts down. When the counter reaches 0000_H: • INTTAUAnIm is generated. When a reverse edge of TAUAnTTINm is detected during count operation: • TAUAnCNTm stops counting and waits for a trigger. Afterwards, this procedure is repeated.
Stop operation	Set TAUAnTT.TAUAnTTm to 1. TAUAnTT.TAUAnTTm is a trigger bit, so it is automatically cleared to 0.	TAUAnTE.TAUAnTEm is cleared to 0 and the counter stops. TAUAnCNTm stops and retains its current value.

Restart

12.16.4 TAUAnTTINm Input Period Count Detection Function

(1) Overview

Summary This function measures the cumulative width of a TAUAnTTINm input signal.

Prerequisites
- The operation mode must be set to Capture & Gate Count Mode. Refer to *Table 12-44 "TAUAnCMORm settings for TAUAnTTINm Input Period Count Detection Function" on page 643*.
- TAUAnTTOUTm is not used for this function.

Description The counter is enabled by setting the channel trigger bit (TAUAnTS.TAUAnTSm) to 1. This in turn sets TAUAnTE.TAUAnTEm = 1, enabling count operation. The counter awaits a valid TAUAnTTINm input edge.

When a valid TAUAnTTINm input start edge is detected, the counter starts to count from 0000_H.

When a valid TAUAnTTINm input stop edge is detected, the current TAUAnCNTm value is loaded to TAUAnCDRm and an interrupt (INTTAUAnIm) is generated. The counter stops and retains its value until the next valid TAUAnTTINm input start edge is detected.

When the next valid TAUAnTTINm input start edge is detected, the counter resumes counting, starting with its value upon stopping.

If the counter reaches $FFFF_H$ the bit TAUAnCSRm.TAUAnOVF is set to 1 and the counter restarts from 0000_H. The value of TAUAnCSRm.TAUAnOVF is reset by the CPU by setting TAUAnCSCm.TAUAnCLOV = 1.

Note The input TAUAnTTINm is sampled at the frequency of the operation clock, specified by the TAUAnCMORm.TAUAnCKS[1:0] bits.

Conditions The valid start and stop edges are specified by the TAUAnCMURm.TAUAnTIS[1:0] bits:
- If TAUAnCMURm.TAUAnTIS[1:0] = 10_B, the TAUAnTTINm input low period is counted. The start trigger is a falling edge and the stop trigger is a rising edge.
- If TAUAnCMURm.TAUAnTIS[1:0] = 11_B, the TAUAnTTINm input high period is counted. The start trigger is a rising edge and the stop trigger is a falling edge.

(2) Equations

Cumulative TAUAnTTINm input width =
count clock cycle × (($FFFF_H$ × TAUAnCSRm.TAUAnOVF) + (TAUAnCDRm capture value + 1))

<R> **Caution** If a capture signal in input when the counter value reaches the upper limit (0FFFFH), the overflow flag (OVF) that should be set in response to the occurrence of an overflow in the next count clock cycle will not be set. The OVF flag is set normally at all other times.

(3) Block diagram and general timing diagram

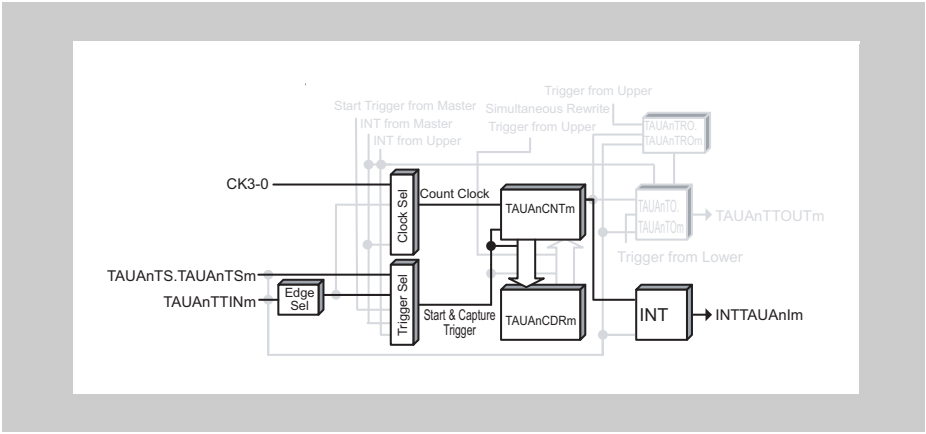


Figure 12-55 Block diagram for TAUAnTTINm Input Period Count Detection Function

The following settings apply to the general timing diagram:

- Rising and falling edge detection = high width measurement (TAUAnCMURm.TAUAnTIS[1:0] = 11_B)

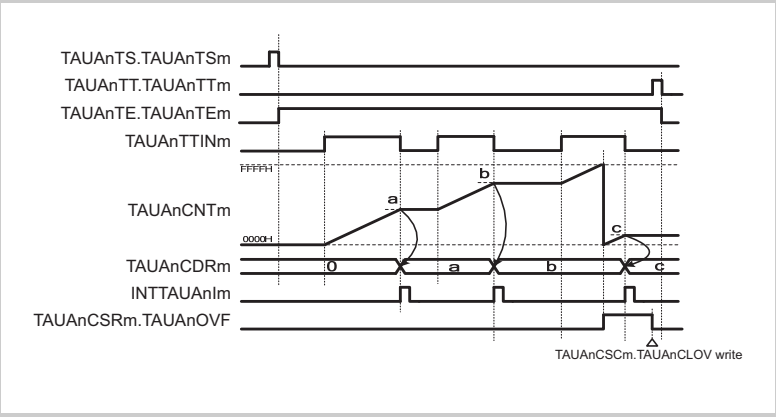


Figure 12-56 General timing diagram for TAUAnTTINm Input Period Count Detection Function

(4) Register settings

(a) TAUAnCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn CKS[1:0]		TAUAn CCS[1:0]		TAUAn MAS	TAUAnSTS[2:0]			TAUAn COS[1:0]		-	TAUAnMD[4:1]				TAUAn MD0

Table 12-44 TAUAnCMORm settings for TAUAnTTINm Input Period Count Detection Function

Bit name	Setting
TAUAnCKS[1:0]	00: Operation clock = CK0 01: Operation clock = CK1 10: Operation clock = CK2 11: Operation clock = CK3
TAUAnCCS[1:0]	00: Operation clock is used as the count clock
TAUAnMAS	0: Not used, so set to 0
TAUAnSTS[2:0]	010: Valid edge of the TAUAnTTINm input signal is the external start trigger and the reverse edge is the stop trigger
TAUAnCOS[1:0]	01: Overflow (TAUAnCSRm.TAUAnOVF) set upon counter overflow and cleared by a CPU instruction
TAUAnMD[4:1]	1101: Capture & Gate Count Mode
TAUAnMD0	0: Disables the start trigger during operation

(b) TAUAnCMURm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-														TIS[1:0]	

Table 12-45 TAUAnCMURm settings for TAUAnTTINm Input Period Count Detection Function

Bit name	Setting
TAUAnTIS[1:0]	10: Rising and falling edge detection (Low width measurement) 11: Rising and falling edge detection (High width measurement)

(c) Channel output mode

Because the channel output mode is not used by this function, clear TAUAnTOE.TAUAnTOEm. However, it can be used in Independent Channel Output Mode Controlled by Software.

(d) Simultaneous rewrite

The simultaneous rewrite registers (TAUAnRDE, TAUAnRDS, TAUAnRDM, and TAUAnRDC) cannot be used with the TAUAnTTINm Input Period Count Detection Function. Therefore, these registers must be set to 0.

Table 12-46 Simultaneous rewrite settings for TAUAnTTINm Input Period Count Detection Function

Bit name	Setting
TAUAnRDEm	0: Disables simultaneous rewrite
TAUAnRDSm	0: When simultaneous rewrite is disabled (TAUAnRDE.TAUAnRDEm = 0), set these bits to 0
TAUAnRDMm	
TAUAnRDCm	

(5) Operating procedure for TAUAnTTINm Input Period Count Detection Function

Table 12-47 Operating procedure for TAUAnTTINm Input Period Count Detection Function

	Operation	Status of TAUAn
Initial channel setting	Set the TAUAnCMORm register and TAUAnCMURm registers as described in *Table 12-44 "TAUAnCMORm settings for TAUAnTTINm Input Period Count Detection Function" on page 643* and *Table 12-45 "TAUAnCMURm settings for TAUAnTTINm Input Period Count Detection Function" on page 643* Set the value of the TAUAnCDRm register	Channel operation is stopped.
Start operation	Set TAUAnTS.TAUAnTSm to 1. TAUAnTS.TAUAnTSm is a trigger bit, so it is automatically cleared to 0. Detection of TAUAnTTINm start edge	TAUAnTE.TAUAnTEm is set to 1 and TAUAnCNTm waits for detection of the TAUAnTTINm start edge. When a start edge is detected, TAUAnCNTm is cleared to 0000_H and TAUAnCNTm starts to count up.
During operation	Detection of TAUAnTTINm edges. The TAUAnCDRm, TAUAnCNTm, and TAUAnCSRm registers can be read at any time. The TAUAnCSCm.TAUAnCLOV bit can be set to 1.	When a TAUAnTTINm start edge (rising edge for high width measurement, falling edge for low width measurement) is detected, TAUAnCNTm starts to count up from the stop value. When TAUAnCNTm detects a stop edge (falling edge for high width measurement, rising edge for low width measurement), it transfers the value to TAUAnCDRm and INTTAUAnIm is generated. Counting stops at the "value transferred to TAUAnCDRm + 1" value and TAUAnCNTm waits for detection of the TAUAnTTINm start edge. If the TAUAnCNTm reaches $FFFF_H$, the counter overflows and TAUAnCSRm.TAUAnOVF is set to 1. Afterwards, this procedure is repeated.
Stop operation	Set TAUAnTT.TAUAnTTm to 1. TAUAnTT.TAUAnTTm is a trigger bit, so it is automatically cleared to 0.	TAUAnTE.TAUAnTEm is cleared to 0 and the counter stops. TAUAnCNTm stops and it and TAUAnCSRm.TAUAnOVF retain their current values.

Restart

(6) Specific timing diagrams

(a) Operation stop and restart

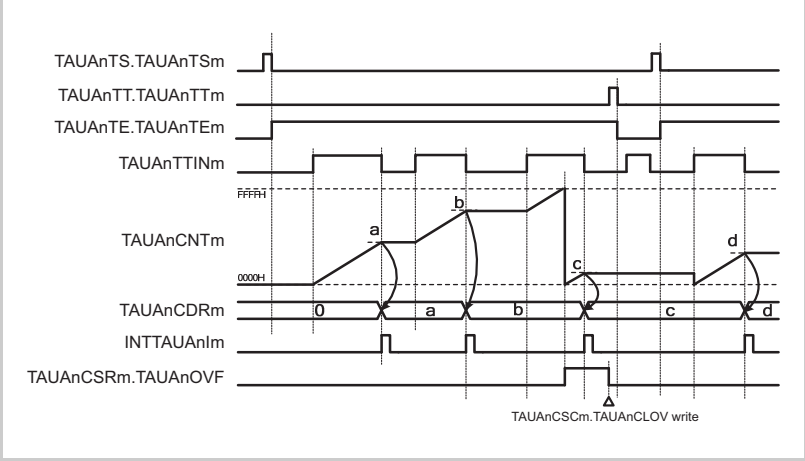


Figure 12-57 Operation stop and restart, TAUAnCMURm.TAUAnTIS[1:0] = 11_B

- The counter can be stopped by setting TAUAnTT.TAUAnTTm to 1, which in turn sets TAUAnTE.TAUAnTEm to 0.
- TAUAnCNTm stops and the current value is retained.
- If the counter is stopped, valid TAUAnTTINm input edges are ignored.
- The counter can be restarted by setting TAUAnTS.TAUAnTSm to 1. TAUAnCNTm restarts to count from 0000_H.

12.16.5 Overflow Interrupt Output Function (During TAUAnTTINm Input Period Count Detection)

(1) Overview

Summary This function measures the cumulative width of a TAUAnTTINm input signal. If the cumulative TAUAnTTINm input width is longer than $FFFF_H$, an interrupt can be generated to output an overflow interrupt.

Prerequisites

- The operation mode must be set to Gate Count Mode. Refer to *Table 12-48 "TAUAnCMORm settings for Overflow Interrupt Output Function (During TAUAnTTINm Input Period Count Detection)" on page 649*.
- TAUAnTTOUTm is not used for this function.
- The value of TAUAnCDRm must be set to $FFFF_H$.

Description The counter is enabled by setting the channel trigger bit (TAUAnTS.TAUAnTSm) to 1. This in turn sets TAUAnTE.TAUAnTEm = 1, enabling count operation.

The counter starts when a valid TAUAnTTINm input start edge is detected. $FFFF_H$ is loaded to TAUAnCNTm and the counter starts to count down.

When a valid stop edge is detected, the counter stops and retains the current value. The counter awaits the next TAUAnTTINm input start edge and then continues to count down from the current value.

When the counter reaches 0000_H an interrupt is generated. $FFFF_H$ is loaded to TAUAnCNTm and the counter continues to count down until a TAUAnTTINm input stop edge is detected.

Conditions The valid start and stop edges are specified by the TAUAnCMURm.TAUAnTIS[1:0] bits:

- If TAUAnCMURm.TAUAnTIS[1:0] = 10_B, the TAUAnTTINm input low period is counted. The start trigger is a falling edge and the stop trigger is a rising edge.
- If TAUAnCMURm.TAUAnTIS[1:0] = 11_B, the TAUAnTTINm input high period is counted. The start trigger is a rising edge and the stop trigger is a falling edge.

Note The counter cannot be restarted during operation.

(2) Block diagram and general timing diagram

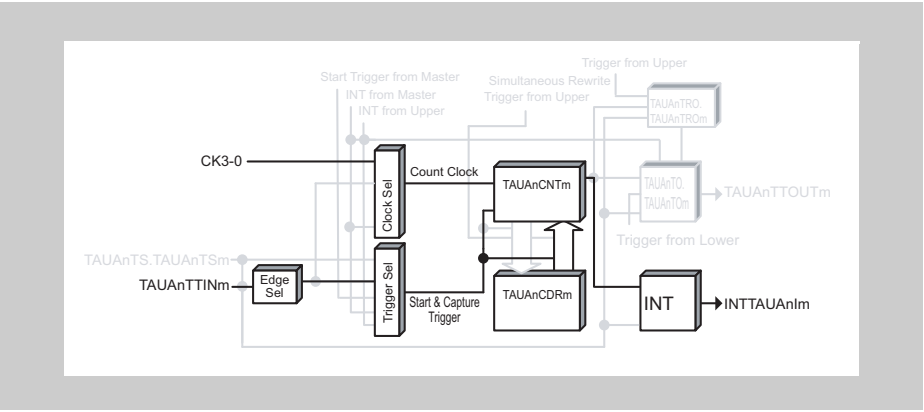


Figure 12-58 Block diagram for Overflow Interrupt Output Function (During TAUAnTTINm Input Period Count Detection)

The following settings apply to the general timing diagram:

- Rising and falling edge detection = high width measurement (TAUAnCMURm.TAUAnTIS[1:0] = 11_B)

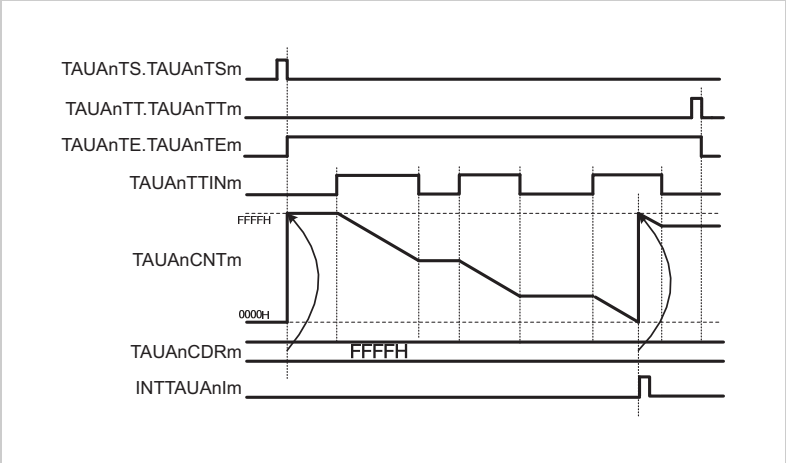


Figure 12-59 General timing diagram for Overflow Interrupt Output Function (During TAUAnTTINm Input Period Count Detection)

(3) Register settings

(a) TAUAnCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn CKS[1:0]		TAUAn CCS[1:0]		TAUAn MAS	TAUAnSTS[2:0]			TAUAn COS[1:0]		-	TAUAnMD[4:1]				TAUAn MD0

Table 12-48 TAUAnCMORm settings for Overflow Interrupt Output Function (During TAUAnTTINm Input Period Count Detection)

Bit name	Setting
TAUAnCKS[1:0]	00: Operation clock = CK0 01: Operation clock = CK1 10: Operation clock = CK2 11: Operation clock = CK3
TAUAnCCS[1:0]	00: Operation clock is used as the count clock
TAUAnMAS	0: Not used, so set to 0
TAUAnSTS[2:0]	010: Valid edge of the TAUAnTTINm input signal is the external start trigger and the reverse edge is the stop trigger
TAUAnCOS[1:0]	00: Not used, so set to 00
TAUAnMD[4:1]	1100: Gate Count Mode
TAUAnMD0	0: INTTAUAnIm not generated at operation start

(b) TAUAnCMURm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-														TAUAnTIS[1:0]	

Table 12-49 TAUAnCMURm settings for Overflow Interrupt Output Function (During TAUAnTTINm Input Period Count Detection)

Bit name	Setting
TAUAnTIS[1:0]	10: Rising and falling edge detection (Low width measurement) 11: Rising and falling edge detection (High width measurement)

(c) Channel output mode

Because the channel output mode is not used by this function, clear TAUAnTOE.TAUAnTOEm. However, it can be used in Independent Channel Output Mode Controlled by Software.

(d) Simultaneous rewrite

The simultaneous rewrite registers (TAUAnRDE, TAUAnRDS, TAUAnRDM, and TAUAnRDC) cannot be used with the Overflow Interrupt Output Function (During TAUAnTTINm Input Period Count Detection). Therefore, these registers must be set to 0.

Table 12-50 Simultaneous rewrite settings for Overflow Interrupt Output Function (During TAUAnTTINm Input Period Count Detection)

Bit name	Setting
TAUAnRDEm	0: Disables simultaneous rewrite
TAUAnRDSm	0: When simultaneous rewrite is disabled (TAUAnRDE.TAUAnRDEm = 0), set these bits to 0
TAUAnRDMm	
TAUAnRDCm	

(4) Operating procedure for Overflow Interrupt Output Function (during TAUAnTTINm input period count detection)

Table 12-51 Operating procedure for Overflow Interrupt Output Function (during TAUAnTTINm input period count detection)

	Operation	Status of TAUAn
Initial channel setting	Set the TAUAnCMORm register and TAUAnCMURm registers as described in *Table 12-48 "TAUAnCMORm settings for Overflow Interrupt Output Function (During TAUAnTTINm Input Period Count Detection)" on page 649* and *Table 12-49 "TAUAnCMURm settings for Overflow Interrupt Output Function (During TAUAnTTINm Input Period Count Detection)" on page 649* Set the value of the TAUAnCDRm register	Channel operation is stopped.
Start operation	Set TAUAnTS.TAUAnTSm to 1. TAUAnTS.TAUAnTSm is a trigger bit, so it is automatically cleared to 0. Detection of TAUAnTTINm start edge	TAUAnTE.TAUAnTEm is set to 1 and TAUAnCNTm waits for detection of the start edge. When a start edge is detected, TAUAnCNTm loads the TAUAnCDRm value ($FFFF_H$).
During operation	The TAUAnCNTm register can be read at all times.	TAUAnCNTm counts down. When the counter reaches 0000_H: • INTTAUAnIm is generated. • TAUAnCNTm loads the TAUAnCDRm value ($FFFF_H$), and then continues counting down. When a reverse edge of TAUAnTTINm is detected during count operation: • TAUAnCNTm counts down from the stop value. Afterwards, this procedure is repeated.
Stop operation	Set TAUAnTT.TAUAnTTm to 1. TAUAnTT.TAUAnTTm is a trigger bit, so it is automatically cleared to 0.	TAUAnTE.TAUAnTEm is cleared to 0 and the counter stops. TAUAnCNTm stops and retains its current value.

Restart

12.16.6 TAUAnTTINm Input Pulse Interval Judgment Function

(1) Overview

Summary This function outputs the result of a comparison between the count value (TAUAnCNTm) and the value in the channel data register (TAUAnCDRm) when a TAUAnTTINm input pulse occurs. An interrupt signal INTTAUAnIm is generated if the result of the comparison is true.

Prerequisites
- The operation mode must be set to Judge Mode. Refer to *Table 12-52 "TAUAnCMORm settings for TAUAnTTINm Input Pulse Interval Judgment Function" on page 654*.
- TAUAnTTOUTm is not used for this function.

Description The counter is started by setting the channel trigger bit (TAUAnTS.TAUAnTSm) to 1. This in turn sets TAUAnTE.TAUAnTEm = 1, enabling count operation. The current value of TAUAnCDRm is loaded to TAUAnCNTm and the counter starts to count down from this value.

When a TAUAnTTINm valid edge is detected or TAUAnTS.TAUAnTSm is set to 1, the function compares the current values of TAUAnCNTm and TAUAnCDRm. An interrupt signal INTTAUAnIm is generated if the result of the comparison is true. TAUAnCNTm reloads the value of TAUAnCDRm, and then subsequently continues operation, regardless of the result of the comparison.

If the counter reaches 0000_H before a TAUAnTTINm valid edge is detected, TAUAnCNTm overflows and is set to $FFFF_H$. It then continues to count down.

The value of TAUAnCDRm can be rewritten at any time, and the changed value of TAUAnCDRm is applied the next time the function starts to count down.

Conditions The TAUAnCMORm.TAUAnMD0 bit specifies the type of comparison:
- If TAUAnCMORm.TAUAnMD0 = 0, INTTAUAnIm is generated when TAUAnCNTm ≤ TAUAnCDRm.
- If TAUAnCMORm.TAUAnMD0 = 1, INTTAUAnIm is generated when TAUAnCNTm > TAUAnCDRm.

(2) Block diagram and general timing diagram

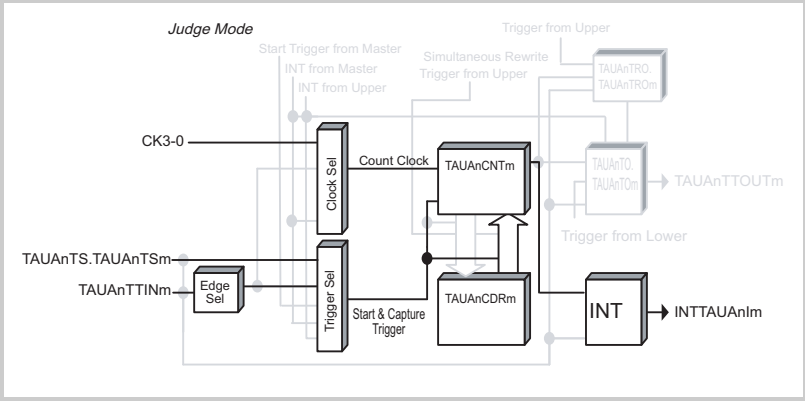


Figure 12-60 Block diagram for TAUAnTTINm Input Pulse Interval Judgment Function

The following settings apply to the general timing diagram:

- Falling edge detection (TAUAnCMURm.TAUAnTIS[1:0] = 00_B)

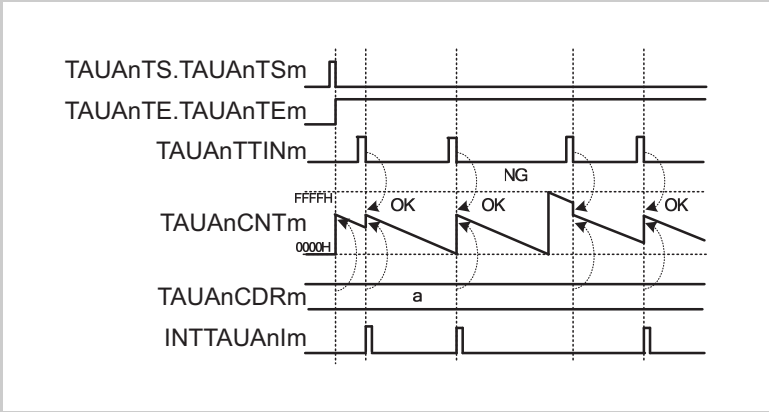


Figure 12-61 General timing diagram for TAUAnTTINm Input Pulse Interval Judgment Function

(3) Register settings

(a) TAUAnCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn CKS[1:0]		TAUAn CCS[1:0]		TAUAn MAS	TAUAnSTS[2:0]			TAUAn COS[1:0]		-	TAUAnMD[4:1]				TAUAn MD0

Table 12-52 TAUAnCMORm settings for TAUAnTTINm Input Pulse Interval Judgment Function

Bit name	Setting
TAUAnCKS[1:0]	00: Operation clock = CK0 01: Operation clock = CK1 10: Operation clock = CK2 11: Operation clock = CK3
TAUAnCCS[1:0]	00: Operation clock is used as the count clock
TAUAnMAS	0: Not used, so set to 0
TAUAnSTS[2:0]	001: Valid edge of the TAUAnTTINm input signal is the external start trigger
TAUAnCOS[1:0]	00: Not used, so set to 00
TAUAnMD[4:1]	0001: Judge Mode
TAUAnMD0	0: INTTAUAnIm is generated when TAUAnCNTm ≤ TAUAnCDRm 1: INTTAUAnIm is generated when TAUAnCNTm > TAUAnCDRm

(b) TAUAnCMURm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-														TAUAnTIS[1:0]	

Table 12-53 TAUAnCMURm settings for TAUAnTTINm Input Pulse Interval Judgment Function

Bit name	Setting
TAUAnTIS[1:0]	00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection

(c) Channel output mode

Because the channel output mode is not used by this function, clear TAUAnTOE.TAUAnTOEm. However, it can be used in Independent Channel Output Mode Controlled by Software.

(d) Simultaneous rewrite

The simultaneous rewrite registers (TAUAnRDE, TAUAnRDS, TAUAnRDM, and TAUAnRDC) cannot be used with the TAUAnTTINm Input Pulse Interval Judgment Function. Therefore, these registers must be set to 0.

Table 12-54 Simultaneous rewrite settings for TAUAnTTINm Input Pulse Interval Judgment Function

Bit name	Setting
TAUAnRDE.TAUAnRDEm	0: Disables simultaneous rewrite
TAUAnRDS.TAUAnRDSm	0: When simultaneous rewrite is disabled (TAUAnRDE.TAUAnRDEm = 0), set these bits to 0
TAUAnRDM.TAUAnRDMm	
TAUAnRDC.TAUAnRDCm	

(4) Operating procedure for TAUAnTTINm Input Pulse Interval Judgment Function

Table 12-55 Operating procedure for TAUAnTTINm Input Pulse Interval Judgment Function

	Operation	Status of TAUAn
Initial channel setting	Set the TAUAnCMORm register and TAUAnCMURm registers as described in *Table 12-52 "TAUAnCMORm settings for TAUAnTTINm Input Pulse Interval Judgment Function" on page 654* and *Table 12-53 "TAUAnCMURm settings for TAUAnTTINm Input Pulse Interval Judgment Function" on page 654* Set the value of the TAUAnCDRm register	Channel operation is stopped.
Start operation	Set TAUAnTS.TAUAnTSm to 1. TAUAnTS.TAUAnTSm is a trigger bit, so it is automatically cleared to 0.	TAUAnTE.TAUAnTEm is set to 1 and the counter starts. TAUAnCNTm loads the TAUAnCDRm value.
During operation	Detection of TAUAnTTINm edges. The value of TAUAnCDRm can be changed at any time. The TAUAnCNTm register can be read at any time.	TAUAnCNTm counts down. When a TAUAnTTINm input edge is detected • TAUAnCNTm reloads TAUAnCDRm, and then continues count operation. • TAUAnCNTm compares the values and judges the condition according to the TAUAnCMORm.TAUAnMD0 setting. • If the condition is satisfied, INTTAUAnIm is generated. Afterwards, this procedure is repeated.
Stop operation	Set TAUAnTT.TAUAnTTm to 1. TAUAnTT.TAUAnTTm is a trigger bit, so it is automatically cleared to 0.	TAUAnTE.TAUAnTEm is cleared to 0 and the counter stops. TAUAnCNTm stops and retains its current value.

Restart

12.16.7 TAUAnTTINm Input Signal Width Judgment Function

(1) Overview

Summary This function outputs the result of a comparison between the count value (TAUAnCNTm) and the value in the channel data register (TAUAnCDRm) at a valid stop edge of a TAUAnTTINm input signal. An interrupt signal INTTAUAnIm is generated if the result of the comparison is true.

Prerequisites
- The operation mode must be set to Judge & One Count Mode. Refer to *Table 12-56 "TAUAnCMORm settings for TAUAnTTINm Input Signal Width Judgment Function" on page 659*.
- TAUAnTTOUTm is not used for this function.

Description The counter is started by setting the channel trigger bit (TAUAnTS.TAUAnTSm) to 1. This in turn sets TAUAnTE.TAUAnTEm = 1, enabling count operation. When a valid TAUAnTTINm input start edge is detected, the current value of TAUAnCDRm is loaded to TAUAnCNTm and the counter starts to count down from this value.

When a TAUAnTTINm valid stop edge is detected, the function compares the current values of TAUAnCNTm and TAUAnCDRm. An interrupt signal INTTAUAnIm is generated if the result of the comparison is true. The counter TAUAnCNTm retains its value until the next TAUAnTTINm valid start edge is detected, regardless of the result of the comparison.

If the counter reaches 0000_H before a valid TAUAnTTINm stop edge is detected, TAUAnCNTm overflows and is set to $FFFF_H$. It then continues to count down.

The value of TAUAnCDRm can be rewritten at any time, and the changed value of TAUAnCDRm is applied the next time the function starts to count down.

Conditions
- The TAUAnCMORm.TAUAnMD0 bit specifies the type of comparison:
  - If TAUAnCMORm.TAUAnMD0 = 0, INTTAUAnIm is generated when TAUAnCNTm ≤ TAUAnCDRm.
  - If TAUAnCMORm.TAUAnMD0 = 1, INTTAUAnIm is generated when TAUAnCNTm > TAUAnCDRm.
- The TAUAnCMURm.TAUAnTIS[1:0] bits specify the type of width measurement:
  - For high width measurement (TAUAnCMURm.TAUAnTIS[1:0] = 11_B), the start edge is a rising TAUAnTTINm edge and the stop edge is a falling TAUAnTTINm edge.
  - For low width measurement (TAUAnCMURm.TAUAnTIS[1:0] = 10_B), the start edge is a falling TAUAnTTINm edge and the stop edge is a rising TAUAnTTINm edge.
- Forced restart is not possible for this function.

(2) Block diagram and general timing diagram

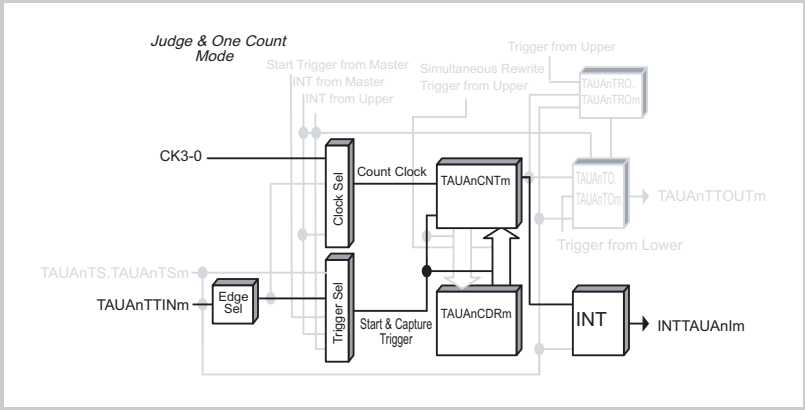


Figure 12-62 Block diagram for TAUAnTTINm Input Signal Width Judgment Function

The following settings apply to the general timing diagram:

- INTTAUAnIm is generated when TAUAnCNTm ≤ TAUAnCDRm (TAUAnCMORm.TAUAnMD0 = 0)
- TAUAnTTINm valid start edge = rising edge, TAUAnTTINm valid stop edge = falling edge (TAUAnCMURm.TAUAnTIS[1:0] = 11_B)

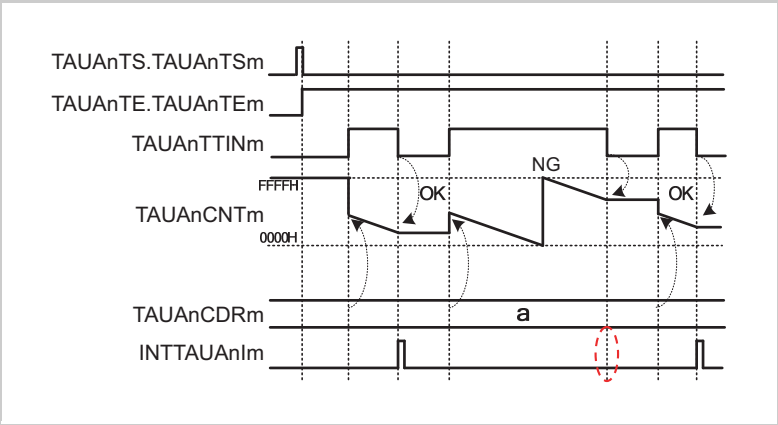


Figure 12-63 General timing diagram for TAUAnTTINm Input Signal Width Judgment Function

(3) Register settings

(a) TAUAnCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn CKS[1:0]		TAUAn CCS[1:0]		TAUAn MAS	TAUAnSTS[2:0]			TAUAn COS[1:0]		-	TAUAnMD[4:1]				TAUAn MD0

Table 12-56 TAUAnCMORm settings for TAUAnTTINm Input Signal Width Judgment Function

Bit name	Setting
TAUAnCKS[1:0]	00: Operation clock = CK0 01: Operation clock = CK1 10: Operation clock = CK2 11: Operation clock = CK3
TAUAnCCS[1:0]	00: Operation clock is used as the count clock
TAUAnMAS	0: Not used, so set to 0
TAUAnSTS[2:0]	010: Valid edge of the TAUAnTTINm input signal is the external start trigger and the reverse edge is the stop trigger
TAUAnCOS[1:0]	00: Not used, so set to 00
TAUAnMD[4:1]	0111: Judge & One Count Mode
TAUAnMD0	0: INTTAUAnIm is generated when TAUAnCNTm ≤ TAUAnCDRm 1: INTTAUAnIm is generated when TAUAnCNTm > TAUAnCDRm

(b) TAUAnCMURm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-														TAUAnTIS[1:0]	

Table 12-57 TAUAnCMURm settings for TAUAnTTINm Input Signal Width Judgment Function

Bit name	Setting
TAUAnTIS[1:0]	10: Rising and falling edge detection (low width measurement) 11: Rising and falling edge detection (high width measurement)

(c) Channel output mode

Because the channel output mode is not used by this function, clear TAUAnTOE.TAUAnTOEm. However, it can be used in Independent Channel Output Mode Controlled by Software.

(d) Simultaneous rewrite

The simultaneous rewrite registers (TAUAnRDE, TAUAnRDS, TAUAnRDM, and TAUAnRDC) cannot be used with the TAUAnTTINm Input Signal Width Judgment Function. Therefore, these registers must be set to 0.

Table 12-58 Simultaneous rewrite settings for TAUAnTTINm Input Signal Width Judgment Function

Bit name	Setting
TAUAnRDE.TAUAnRDEm	0: Disables simultaneous rewrite
TAUAnRDS.TAUAnRDSm	0: When simultaneous rewrite is disabled (TAUAnRDE.TAUAnRDEm = 0), set these bits to 0
TAUAnRDM.TAUAnRDMm	
TAUAnRDC.TAUAnRDCm	

(4) Operating procedure for TAUAnTTINm Input Signal Width Judgment Function

Table 12-59 Operating procedure for TAUAnTTINm Input Signal Width Judgment Function

	Operation	Status of TAUAn
Initial channel setting	Set the TAUAnCMORm register and TAUAnCMURm registers as described in *Table 12-56 "TAUAnCMORm settings for TAUAnTTINm Input Signal Width Judgment Function" on page 659* and *Table 12-57 "TAUAnCMURm settings for TAUAnTTINm Input Signal Width Judgment Function" on page 659* Set the value of the TAUAnCDRm register	Channel operation is stopped.
Start operation	Set TAUAnTS.TAUAnTSm to 1. TAUAnTS.TAUAnTSm is a trigger bit, so it is automatically cleared to 0. Detection of TAUAnTTINm start edge	TAUAnTE.TAUAnTEm is set to 1 and TAUAnCNTm waits for detection of the TAUAnTTINm start edge. When a TAUAnTTINm start edge is detected, TAUAnCNTm loads the TAUAnCDRm value.
During operation	Detection of TAUAnTTINm edges The value of TAUAnCDRm can be changed at any time. The TAUAnCNTm register can be read at any time.	TAUAnCNTm counts down. When a TAUAnTTINm stop edge is detected: • TAUAnCNTm stops and waits for detection of the TAUAnTTINm start edge. • TAUAnCNTm compares the values and judges the condition according to the TAUAnCMORm.TAUAnMD0. • If the condition is satisfied, INTTAUAnIm is generated Afterwards, this procedure is repeated.
Stop operation	Set TAUAnTT.TAUAnTTm to 1. TAUAnTT.TAUAnTTm is a trigger bit, so it is automatically cleared to 0.	TAUAnTE.TAUAnTEm is cleared to 0 and the counter stops. TAUAnCNTm stops and retains its current value.

Restart

12.17 Independent Channel Real-Time Functions

This chapter describes functions that output the value of the TAUAnTRO.TAUAnTROm bit in real time:

- *12.17.1 “Real-Time Output Function Type 1”*
- *12.17.2 “Real-Time Output Function Type 2”*

12.17.1 Real-Time Output Function Type 1

(1) Overview

Summary

This function outputs the value of the TAUAnTRO.TAUAnTROm bit from TAUAnTTOUTm when a specified channel generates an interrupt (INTTAUAnIm). In this function, the interrupt is generated at certain specified intervals.

The upper channel generates the real-time output trigger (TAUAnTRC.TAUAnTRCm = 1), and the lower channel receives this trigger and performs real-time output (TAUAnTRC.TAUAnTRCm = 0).

Prerequisites

- One channel using the TAUAnTTOUTm control of one of more other channels
- The operation mode of the upper channel must be set to Interval Timer Mode. Refer to *Table 12-60 "TAUAnCMORm settings for Real-Time Output Function Type 1" on page 665*.
- The operation mode of the lower channel(s) can be set as desired.
- The channel output mode of all the channels must be set to Independent Channel Output Mode 1 with Real-Time Output. Refer to *12.8 "Channel Output Modes" on page 577*.
- Real-time output must be enabled for the upper channel (TAUAnTRE.TAUAnTREm = 1).

Description

The counter of the upper channel is started by setting the channel trigger bit (TAUAnTS.TAUAnTSm) to 1. This in turn sets TAUAnTE.TAUAnTEm, enabling count operation. The current value of the data register of the upper channel (TAUAnCDRm) is loaded to the counter (TAUAnCNTm) and the counter starts to count down from this value.

When the counter of the upper channel reaches 0000_H, INTTAUAnIm is generated and TAUAnTTOUTm outputs the current value of the real-time output bit (TAUAnTRO.TAUAnTROm) in all the channels for which TAUAnTRE.TAUAnTREm is set. TAUAnCNTm reloads the TAUAnCDRm value, and then subsequently continues operation.

The TAUAnTTOUTm signal only changes when an interrupt is generated, and then only when its value is different to current value of TAUAnTRO.TAUAnTROm at the moment that the interrupt is generated.

Conditions

- The channel that detects the generation of INTTAUAnIm is specified by setting TAUAnTRC.TAUAnTRCm to 1 for the corresponding channel. The TAUAnTRC.TAUAnTRCm bit must be cleared to 0 for all other channels that do not generate the real-time output trigger.
- If real-time output of a lower channel is disabled (TAUAnTRE.TAUAnTREm = 0) or the channel itself is used as the rewrite trigger (TAUAnTRC.TAUAnTRCm = 1), the value of that channel's TAUAnTRO.TAUAnTROm bit is output when INTTAUAnIm is generated by that channel.
- If real-time output of a lower channel is enabled (TAUAnTRE.TAUAnTREm = 1) and TAUAnTRC.TAUAnTRCm = 0, the value of that channel's TAUAnTRO.TAUAnTROm bit is output when INTTAUAnIm is generated by an upper channel.
- If the TAUAnCMORm.TAUAnMD0 bit is set to 0, the first interrupt after a start or restart is not output. For details refer to *12.10 "TAUAnTTOUTm Output and INTTAUAnIm Generation when Counter Starts or Restarts" on page 589*.

(2) Equations

INTTAUAnIm generation cycle = count clock cycle × (TAUAnCDRm value + 1)

(3) Block diagram and general timing diagram

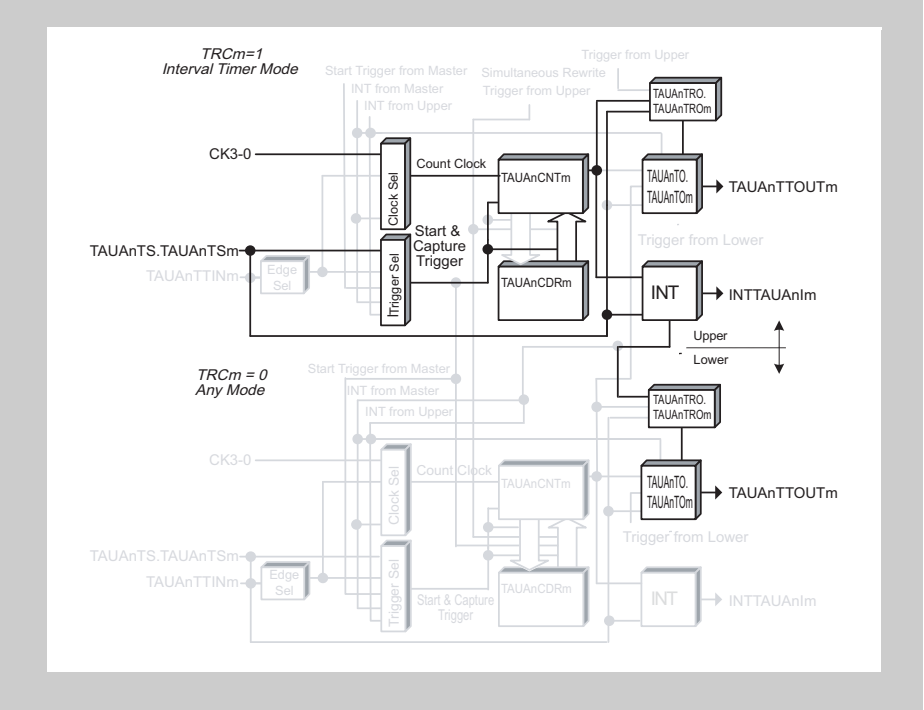


Figure 12-64 Block diagram for Real-Time Output Function Type 1

The following settings apply to the general timing diagram:

- INTTAUAnIm is generated at operation start (TAUAnCMORm.TAUAnMD0 = 1)

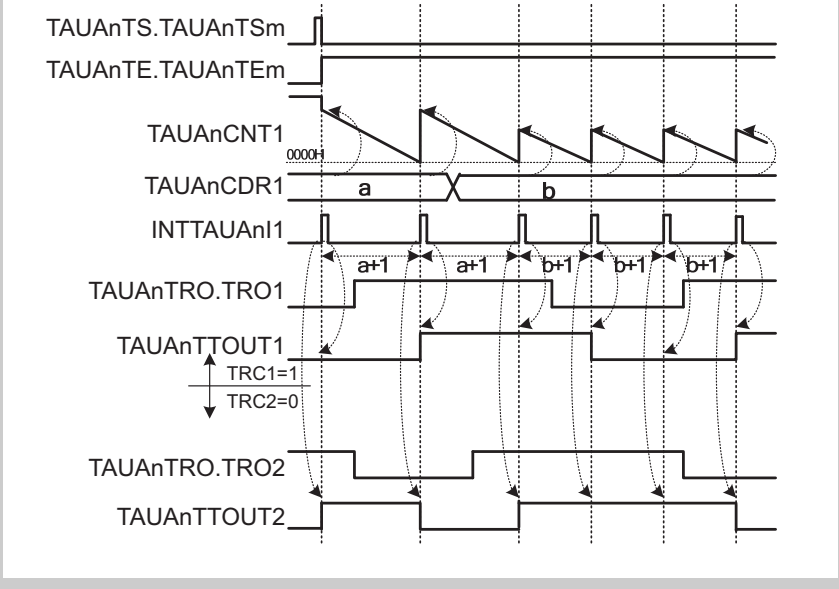


Figure 12-65 General timing diagram for Real-Time Output Function Type 1

(4) Register settings for the upper channel

(a) TAUAnCMORm for the upper channel

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn CKS[1:0]		TAUAn CCS[1:0]		TAUAn MAS	TAUAnSTS[2:0]			TAUAn COS[1:0]		-	TAUAnMD[4:1]				TAUAn MD0

Table 12-60 TAUAnCMORm settings for Real-Time Output Function Type 1

Bit name	Setting
TAUAnCKS[1:0]	00: Operation clock = CK0 01: Operation clock = CK1 10: Operation clock = CK2 11: Operation clock = CK3
TAUAnCCS[1:0]	00: Operation clock is used as the count clock
TAUAnMAS	0: Not used, so set to 0
TAUAnSTS[2:0]	000: Counter triggered by software trigger
TAUAnCOS[1:0]	00: Not used, so set to 00
TAUAnMD[4:1]	0000: Interval Timer Mode
TAUAnMD0	0: INTTAUAnIm is not generated at operation start. 1: INTTAUAnIm is generated at operation start.

(b) TAUAnCMURm for the upper channel

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-														TAUAnTIS[1:0]	

Table 12-61 TAUAnCMURm settings for Real-Time Output Function Type 1

Bit name	Setting
TAUAnTIS[1:0]	00: Not used, so set to 00

(c) Channel output mode for the upper channel

Table 12-62 Control bit settings for Independent Channel Output Mode 1 with Real-Time Output

Bit name	Setting
TAUAnTOE.TAUAnTOEm	1: Enables Independent Channel Output Mode
TAUAnTOM.TAUAnTOMm	0: Independent channel output
TAUAnTOC.TAUAnTOCm	0: Operation mode 1 (= Toggle mode if TAUAnTOM.TAUAnTOMm = 0)
TAUAnTOL.TAUAnTOLm	0: Positive logic
TAUAnTDE.TAUAnTDEm	0: Disables dead time operation
TAUAnTDM.TAUAnTDMm	0: When dead time operation is disabled (TAUAnTDE.TAUAnTDEm = 0), set these bits to 0
TAUAnTDL.TAUAnTDLm	
TAUAnTRE.TAUAnTREm	1: Enables real-time output
TAUAnTRO.TAUAnTROm	0: Real-time output is low 1: Real-time output is high
TAUAnTRC.TAUAnTRCm	1: Channel m generates its own real-time trigger
TAUAnTME.TAUAnTMEm	0: Disables modulation

(d) Simultaneous rewrite for the upper channel

The simultaneous rewrite registers (TAUAnRDE, TAUAnRDS, TAUAnRDM, and TAUAnRDC) cannot be used with the Real-Time Output Function Type 1. Therefore, these registers must be set to 0.

Table 12-63 Simultaneous rewrite settings for Real-Time Output Function Type 1

Bit name	Setting
TAUAnRDE.TAUAnRDEm	0: Disables simultaneous rewrite
TAUAnRDS.TAUAnRDSm	0: When simultaneous rewrite is disabled (TAUAnRDE.TAUAnRDEm = 0), set these bits to 0
TAUAnRDM.TAUAnRDMm	
TAUAnRDC.TAUAnRDCm	

(5) Register settings for the lower channel(s)

(a) TAUAnCMORm for the lower channel(s)

The TAUAnCMORm register of the lower channel(s) can be set arbitrarily.

(b) TAUAnCMURm for the lower channel(s)

The TAUAnCMURm register of the lower channel(s) can be set arbitrarily.

(c) Channel output mode for the lower channel(s)

Table 12-64 Control bit settings for the lower channel in Independent Channel Output Mode 1 with Real-Time Output

Bit name	Setting
TAUAnTOE.TAUAnTOEm	1: Enables Independent Channel Output Mode
TAUAnTOM.TAUAnTOMm	0: Independent channel output
TAUAnTOC.TAUAnTOCm	0: Operation mode 1 (= Toggle mode if TAUAnTOM.TAUAnTOMm = 0)
TAUAnTOL.TTAUAnOLm	0: Positive logic
TAUAnTDE.TAUAnTDEm	0: Disables dead time operation
TAUAnTDM.TAUAnTDMm	0: When dead time operation is disabled (TAUAnTDE.TAUAnTDEm = 0), set these bits to 0
TAUAnTDL.TAUAnTDLm	0: When dead time operation is disabled (TAUAnTDE.TAUAnTDEm = 0), set these bits to 0
TAUAnTRE.TAUAnTREm	1: Enables real-time output
TAUAnTRO.TAUAnTROm	0: Real-time output is low 1: Real-time output is high
TAUAnTRC.TAUAnTRCm	0: The next upper channel generates the real-time trigger for channel m
TAUAnTME.TAUAnTMEm	0: Disables modulation

(d) Simultaneous rewrite for the lower channel(s)

The simultaneous rewrite registers of the lower channel(s) can be set arbitrarily.

(6) Operating procedure for Real-Time Output Function Type 1

Table 12-65 Operating procedure for Real-Time Output Function Type 1

	Operation	Status of TAUAn
Initial channel setting	Set the TAUAnCMORm register and TAUAnCMURm registers for the upper channel as described in *Table 12-60 "TAUAnCMORm settings for Real-Time Output Function Type 1" on page 665* and *Table 12-61 "TAUAnCMURm settings for Real-Time Output Function Type 1" on page 665* Set the TAUAnCMORm register and TAUAnCMURm registers for the lower channel as described in *(5) "Register settings for the lower channel(s)" on page 667* Set the value of the TAUAnCDRm register (only for the channels for which TAUAnTRC.TAUAnTRCm is set) Set the channel output mode for the upper channel by setting the control bits as described in *Table 12-62 "Control bit settings for Independent Channel Output Mode 1 with Real-Time Output" on page 666* Set the channel output mode for the lower channel by setting the control bits as described in *Table 12-64 "Control bit settings for the lower channel in Independent Channel Output Mode 1 with Real-Time Output" on page 667*	Channel operation is stopped.
Start operation	Set TAUAnTS.TAUAnTSm = 1 in the channel where TAUAnTRC.TAUAnTRCm is set to 1. TAUAnTS.TAUAnTSm is a trigger bit, so it is automatically cleared to 0.	[Channel where TAUAnTRC.TAUAnTRCm is set to 1] TAUAnTE.TAUAnTEm is set to 1 and the counter starts. TAUAnCNTm loads the TAUAnCDRm value. INTTAUAnIm is generated when TAUAnCMORm.TAUAnMD0 is set to 1.
During operation	The TAUAnCDRm register and TAUAnTRO.TAUAnTROm can be changed at any time. The TAUAnCNTm register can be read at any time.	TAUAnCNTm counts down. When the counter reaches 0000_H: • TAUAnCNTm reloads the TAUAnCDRm value, and then continues count operation. • INTTAUAnIm is generated. • TAUAnTTOUTm outputs the current value of the real-time output bit TAUAnTRO.TAUAnTROm. The operation is then repeated
Stop operation	Set TAUAnTT.TAUAnTTm to 1. TAUAnTT.TAUAnTTm is a trigger bit, so it is automatically cleared to 0.	TAUAnTE.TAUAnTEm is cleared to 0 and the counter stops. TAUAnCNTm stops and both it and TAUAnTTOUTm retain their current values.

Restart

(7) Specific timing diagrams

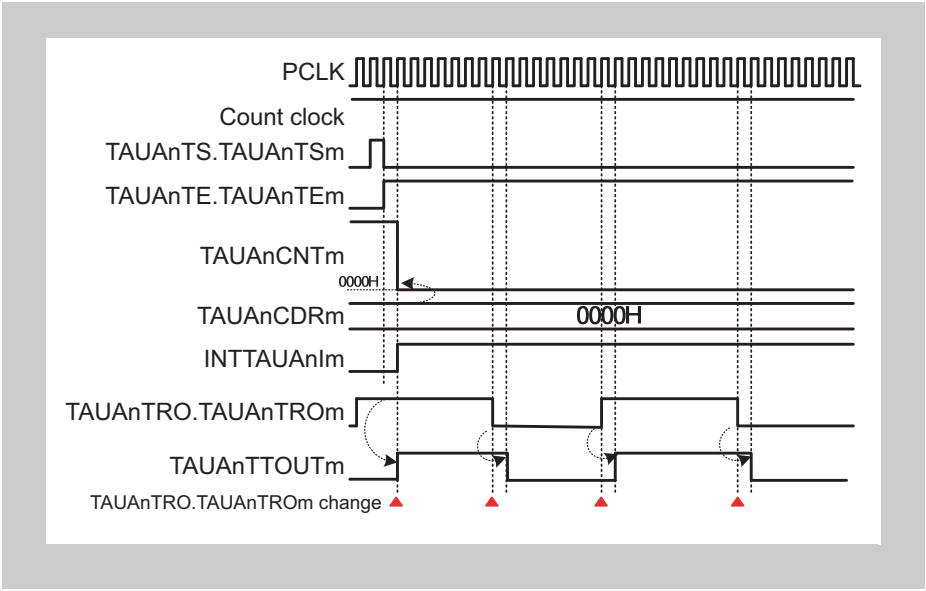


Figure 12-66 TAUAnCDRm = 0000_H, TAUAnCMORm.TAUAnMD0 = 1

- The value of TAUAnTTOUTm follows the TAUAnTRO.TAUAnTROm settings with once PCLK cycle delay.

12.17.2 Real-Time Output Function Type 2

(1) Overview

Summary

This function outputs the value of the TAUAnTRO.TAUAnTROm bit from TAUAnTTOUTm when a specified channel generates an interrupt (INTTAUAnIm). In this function, the interrupt is generated when a valid TAUAnTTINm input edge is detected or the function starts.

The upper channel generates the real-time output trigger (TAUAnTRC.TAUAnTRCm = 1), and the lower channel receives this trigger and performs real-time output (TAUAnTRC.TAUAnTRCm = 0).

Prerequisites

- One channel using the TAUAnTTOUTm control of one of more other channels
- The operation mode of the upper channel must be set to Capture Mode. Refer to *Table 12-66 "TAUAnCMORm settings for Real-Time Output Function Type 2" on page 672*.
- The operation mode of the lower channel(s) can be set as desired.
- The channel output mode of all the channels must be set to Independent Channel Output Mode 1 with Real-Time Output. Refer to *12.8 "Channel Output Modes" on page 577*.
- Real-time output must be enabled for the upper channel (TAUAnTRE.TAUAnTREm = 1).

Description

The counter of the upper channel is started by setting the channel trigger bit (TAUAnTS.TAUAnTSm) to 1. This in turn sets TAUAnTE.TAUAnTEm, enabling count operation. The counter of the upper channel starts to count up.

When a valid TAUAnTTINm input edge is generated on the upper channel, an interrupt is generated, and TAUAnTTOUTm outputs the current value of the real-time output bit (TAUAnTRO.TAUAnTROm) on all channels (only for channels for which TAUAnTRE.TAUAnTREm = 1).

The TAUAnTTOUTm signal only changes when an interrupt is generated, and then only when its value is different to current value of TAUAnTRO.TAUAnTROm at the moment that the interrupt is generated.

Conditions

- The channel that detects the generation of INTTAUAnIm is specified by setting TAUAnTRC.TAUAnTRCm to 1 for the corresponding channel. The TAUAnTRC.TAUAnTRCm bit must be cleared to 0 for all other channels that do not generate the real-time output trigger.
- If real-time output of a lower channel is disabled (TAUAnTRE.TAUAnTREm = 0) or the channel itself is used as the rewrite trigger (TAUAnTRC.TAUAnTRCm = 1), the value of that channel's TAUAnTRO.TAUAnTROm bit is output when INTTAUAnIm is generated by that channel.
- If real-time output of a lower channel is enabled (TAUAnTRE.TAUAnTREm = 1) and TAUAnTRC.TAUAnTRCm = 0, the value of that channel's TAUAnTRO.TAUAnTROm bit is output when INTTAUAnIm is generated by an upper channel.
- If the TAUAnCMORm.TAUAnMD0 bit is set to 0, the first interrupt after a start or restart is not output. For details refer to *12.10 "TAUAnTTOUTm Output and INTTAUAnIm Generation when Counter Starts or Restarts" on page 589*.

(2) Block diagram and general timing diagram

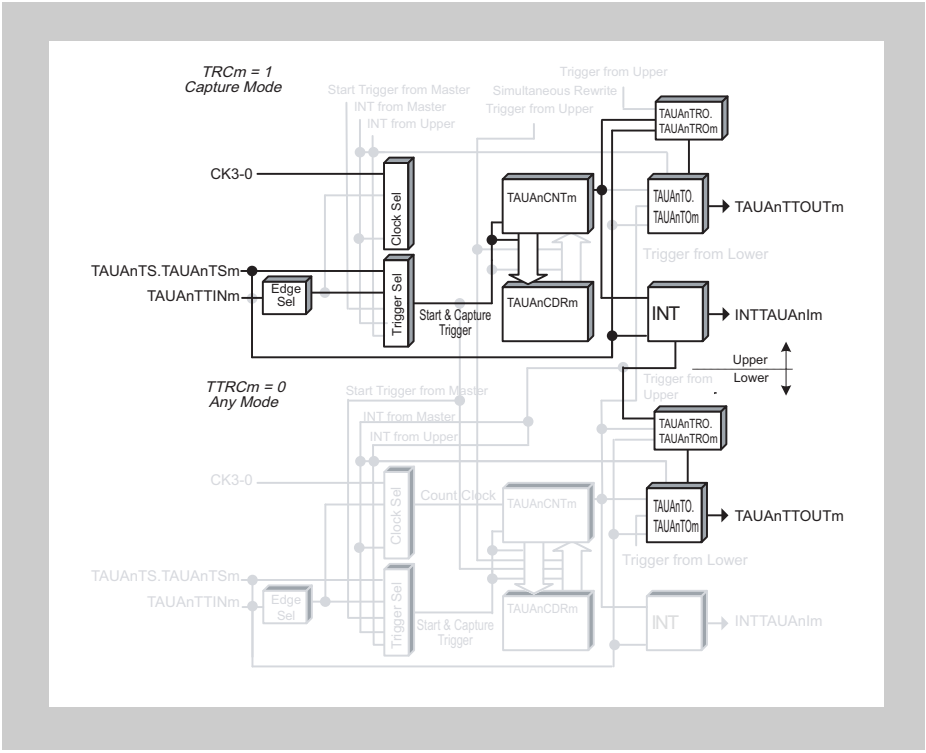


Figure 12-67 Block diagram for Real-Time Output Function Type 2

The following settings apply to the general timing diagram:

- INTTAUAnIm not output at operation start (TAUAnCMORm.TAUAnMD0 = 0)

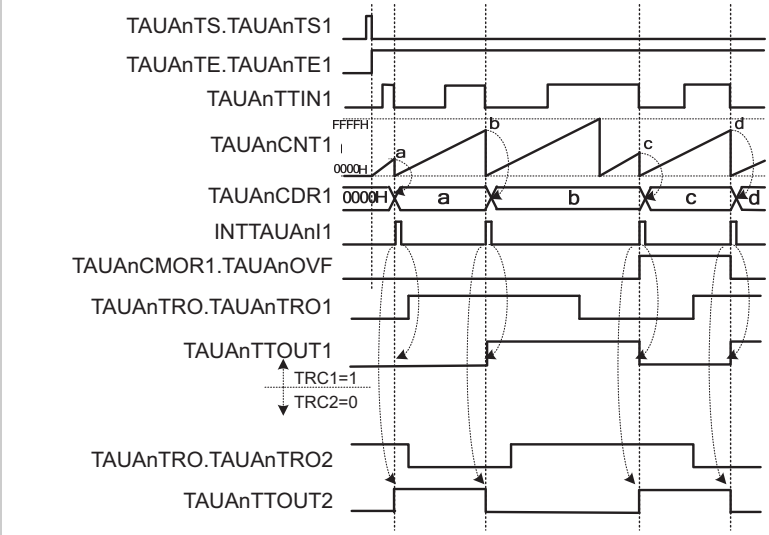


Figure 12-68 General timing diagram for Real-Time Output Function Type 2

(3) Register settings for the upper channel

(a) TAUAnCMORm for the upper channel

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn CKS[1:0]		TAUAn CCS[1:0]		TAUAn MAS	TAUAnSTS[2:0]			TAUAn COS[1:0]		-	TAUAnMD[4:1]				TAUAn MD0

Table 12-66 TAUAnCMORm settings for Real-Time Output Function Type 2

Bit name	Setting
TAUAnCKS[1:0]	00: Operation clock = CK0 01: Operation clock = CK1 10: Operation clock = CK2 11: Operation clock = CK3
TAUAnCCS[1:0]	00: Operation clock is used as the count clock
TAUAnMAS	0: Not used, so set to 0
TAUAnSTS[2:0]	000: Valid edge of the TAUAnTTINm input signal is the external start trigger
TAUAnCOS[1:0]	00: Not used, so set to 00
TAUAnMD[4:1]	0010: Capture Mode
TAUAnMD0	0: INTTAUAnIm is not generated at operation start. 1: INTTAUAnIm is generated at operation start.

(b) TAUAnCMURm for the upper channel

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-														TAUAnTIS[1:0]	

Table 12-67 TAUAnCMURm settings for Real-Time Output Function Type 2

Bit name	Setting
TAUAnTIS[1:0]	00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection

(c) Channel output mode for the upper channel

Table 12-68 Control bit settings for Independent Channel Output Mode 1 with Real-Time Output

Bit name	Setting
TAUAnTOE.TAUAnTOEm	1: Enables Independent Channel Output Mode
TAUAnTOM.TAUAnTOMm	0: Independent channel output
TAUAnTOC.TAUAnTOCm	0: Operation mode 1 (= Toggle mode if TAUAnTOM.TAUAnTOMm = 0)
TAUAnTOL.TAUAnTOLm	0: Positive logic
TAUAnTDE.TAUAnTDEm	0: Disables dead time operation
TAUAnTDM.TAUAnTDMm	0: When dead time operation is disabled (TAUAnTDE.TAUAnTDEm = 0), set these bits to 0
TAUAnTDL.TAUAnTDLm	
TAUAnTRE.TAUAnTREm	1: Enables real-time output
TAUAnTRO.TAUAnTROm	0: Real-time output is low 1: Real-time output is high
TAUAnTRC.TAUAnTRCm	1: Channel m generates its own real-time trigger
TAUAnTME.TAUAnTMEm	0: Disables modulation

(d) Simultaneous rewrite for the upper channel

The simultaneous rewrite registers (TAUAnRDE, TAUAnRDS, TAUAnRDM, and TAUAnRDC) cannot be used with the Real-Time Output Function Type 2. Therefore, these registers must be set to 0.

Table 12-69 Simultaneous rewrite settings for Real-Time Output Function Type 2

Bit name	Setting
TAUAnRDE.TAUAnRDEm	0: Disables simultaneous rewrite
TAUAnRDS.TAUAnRDSm	0: When simultaneous rewrite is disabled (TAUAnRDE.TAUAnRDEm = 0), set these bits to 0
TAUAnRDM.TAUAnRDMm	
TAUAnRDC.TAUAnRDCm	

(4) Register settings for the lower channel(s)

(a) TAUAnCMORm for the lower channel(s)

The TAUAnCMORm register of the lower channel(s) can be set arbitrarily.

(b) TAUAnCMURm for the lower channel(s)

The TAUAnCMURm register of the lower channel(s) can be set arbitrarily.

(c) Channel output mode for the lower channel(s)

Table 12-70 Control bit settings for lower channel in Independent Channel Output Mode 1 with Real-Time Output

Bit name	Setting
TAUAnTOE.TAUAnTOEm	1: Enables Independent Channel Output Mode
TAUAnTOM.TAUAnTOMm	0: Independent channel output
TAUAnTOC.TAUAnTOCm	0: Operation mode 1 (= Toggle mode if TAUAnTOM.TAUAnTOMm = 0)
TAUAnTOL.TAUAnTOLm	0: Positive logic
TAUAnTDE.TAUAnTDEm	0: Disables dead time operation
TAUAnTDM.TAUAnTDMm	0: When dead time operation is disabled (TAUAnTDE.TAUAnTDEm = 0), set these bits to 0
TAUAnTDL.TAUAnTDLm	
TAUAnTRE.TAUAnTREm	0: Disables real-time output 1: Enables real-time output
TAUAnTRO.TAUAnTROm	0: Real-time output is low 1: Real-time output is high
TAUAnTRC.TAUAnTRCm	0: The upper channel generates the real-time trigger for channel m 1: Channel m generates its own real-time trigger
TAUAnTME.TAUAnTMEm	0: Disables modulation

(d) Simultaneous rewrite for the lower channel(s)

The simultaneous rewrite registers of the lower channel(s) can be set arbitrarily.

(5) Operating procedure for Real-Time Output Function Type 2

Table 12-71 Operating procedure for Real-Time Output Function Type 2

	Operation	Status of TAUAn
Initial channel setting	Set the TAUAnCMORm register and TAUAnCMURm registers for the upper channel as described in *Table 12-66 "TAUAnCMORm settings for Real-Time Output Function Type 2" on page 672* and *Table 12-67 "TAUAnCMURm settings for Real-Time Output Function Type 2" on page 672* Set the TAUAnCMORm register and TAUAnCMURm registers for the lower channel as described in *(4) "Register settings for the lower channel(s)" on page 674* Set the channel output mode for the upper channel by setting the control bits as described in *Table 12-68 "Control bit settings for Independent Channel Output Mode 1 with Real-Time Output" on page 673* Set the channel output mode for the lower channel by setting the control bits as described in *Table 12-70 "Control bit settings for lower channel in Independent Channel Output Mode 1 with Real-Time Output" on page 674*	Channel operation is stopped.
Start operation	Set TAUAnTS.TAUAnTSm of the channel where TAUAnTRC.TAUAnTRCm is set to 1. TAUAnTS.TAUAnTSm is a trigger bit, so it is automatically cleared to 0.	[Channel where TAUAnTRC.TAUAnTRCm is set to 1] TAUAnTE.TAUAnTEm is set to 1 and the counter starts. TAUAnCNTm is cleared to 0000_H. When TAUAnCMORm.TAUAnMD0 is set to 1, INTTAUAnIm is generated.
During operation	TAUAnTRO.TAUAnTROm can be changed at any time.	TAUAnCNTm starts to count up from 0000_H. When a TAUAnTTINm input valid edge is detected: • INTTAUAnIm is generated. • TAUAnCSRm.TAUAnOVF is cleared to 0. TAUAnTTOUTm outputs the current value of the real-time output bit TAUAnTRO.TAUAnTROm. Afterwards, this procedure is repeated.
Stop operation	Set TAUAnTT.TAUAnTTm to 1. TAUAnTT.TAUAnTTm is a trigger bit, so it is automatically cleared to 0.	TAUAnTE.TAUAnTEm is cleared to 0 and the counter stops. TAUAnCNTm stops and both it, TAUAnCSRm.TAUAnOVF, and TAUAnTTOUTm retain their current values.

Restart

12.18 Independent Channel Simultaneous Rewrite Functions

This chapter describes functions that carry out simultaneous rewrite:

- *12.18.1 “Simultaneous Rewrite Trigger Generation Function Type 1”*
- *12.18.2 “Simultaneous Rewrite Trigger Generation Function Type 2”*

12.18.1 Simultaneous Rewrite Trigger Generation Function Type 1

(1) Overview

Summary

This function generates an interrupt on a specific channel that can be used by lower channels as a simultaneous rewrite trigger. The interrupt is generated at regular intervals.

The upper channel generates the simultaneous rewrite trigger (TAUAnRDC.TAUAnRDCm = 1), and the lower channel receives this trigger and performs simultaneous rewriting (TAUAnRDC.TAUAnRDCm = 0).

Prerequisites

- At least two channels lower than a channel used as an upper channel, each with simultaneous rewriting enabled (TAUAnRDE.TAUAnRDEm = 1)
- The operation mode of the upper channel must be set to Interval Timer Mode. Refer to *Table 12-72 "TAUAnCMORm settings for Simultaneous Rewrite Trigger Generation Function Type 1" on page 680*.
- For details about which operation modes can be specified for lower channels, refer to *Table 12-7 "Simultaneous rewrite methods and when they are triggered" on page 565*.
- TAUAnTTOUTm is not used for any channel in this function.

Description

The counter is enabled by setting the channel trigger bit of the upper channel or lower channel (TAUAnTS.TAUAnTSm) to 1. This in turn sets TAUAnTE.TEm to 1, enabling counting. The current value of the data register buffer of the upper channel (TAUAnCDRm buf) is written to the counter (TAUAnCNTm) and the counter starts to count down from this value. The counter(s) of the lower channel(s) start to count as specified by their selected operating modes.

When a counter reaches 0000_H, an interrupt is generated from the channel. The TAUAnCNTm loads the current value of the corresponding TAUAnCDRm buffer, and then subsequently continues operation.

If the channel where the interrupt occurs is specified as the trigger channel for simultaneous rewrite (TAUAnRDC.TAUAnRDCm = 1) and is an upper channel, simultaneous rewrite takes place on all lower channels in which simultaneous rewrite is currently possible (TAUAnRSF.TAUAnRSFm = 1).

The values of the data registers are copied to the corresponding data register buffers. Each time a counter starts to count down, it reads the value in the data register buffer and counts down from this value.

The value of a data register can be changed at any time, but it is only transferred to the corresponding data register buffer when simultaneous rewrite occurs.

Conditions

- The channel that detects the generation of INTTAUAnIm is specified by setting TAUAnRDC.TAUAnRDCm to 1 for the corresponding channel. The TAUAnRDC.TAUAnRDCm bit must be 0 for all other channels in which simultaneous rewrite should take place.
- If the TAUAnCMORm.TAUAnMD0 bit is set to 0, the first interrupt after a start or restart is not generated. For details refer to *12.10 "TAUAnTTOUTm Output and INTTAUAnIm Generation when Counter Starts or Restarts" on page 589*.

(2) Equations

Simultaneous rewrite trigger generation cycle = count clock cycle × (TAUAnCDRm + 1)

To control simultaneous rewrite, the following condition must be satisfied:

For PMW:

TAUAnCDRm = [(value of TAUAnCDRm of master channel subject to simultaneous rewrite + 1) × number of interrupts] - 1

For triangle PWM:

TAUAnCDRm = [(value of TAUAnCDRm of master channel subject to simultaneous rewriting + 1) × 2 × number of interrupts] - 1

That is, the ratio of TAUAnCDRm + 1 and TAUAnCDRm_master + 1 must be an integer. This integer corresponds to the number of interrupts.

Note that, for triangle PWM, the cycle is doubled.

(3) Block diagram and general timing diagram

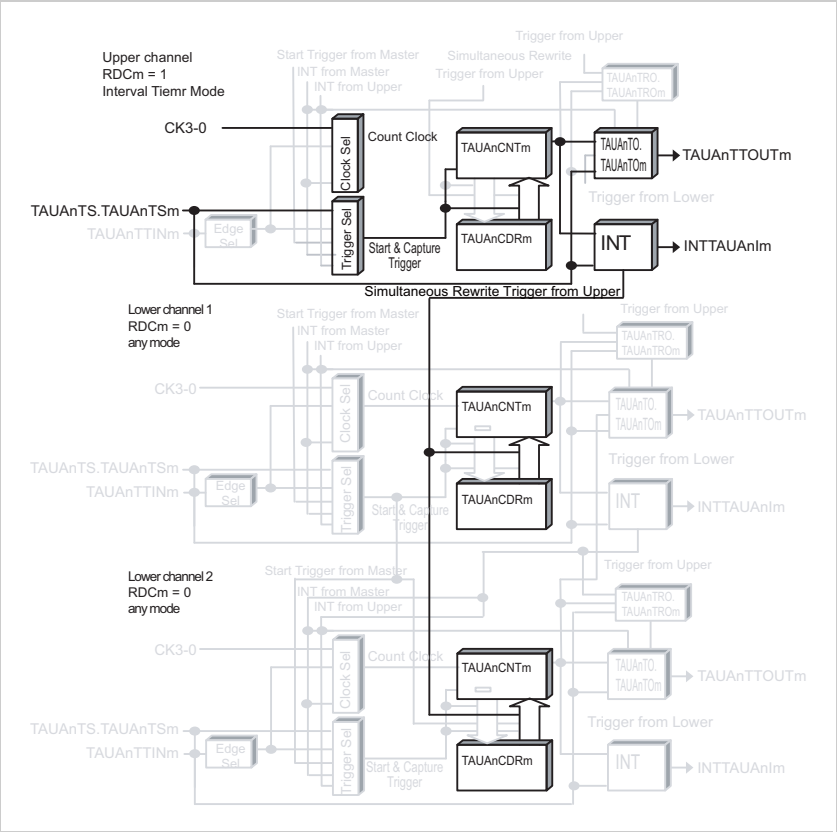


Figure 12-69 Block diagram for Simultaneous Rewrite Trigger Generation Function Type 1

The following settings apply to the general timing diagram:

- INTTAUAnIm generated at operation start (TAUAnCMORm.TAUAnMD0 = 1)

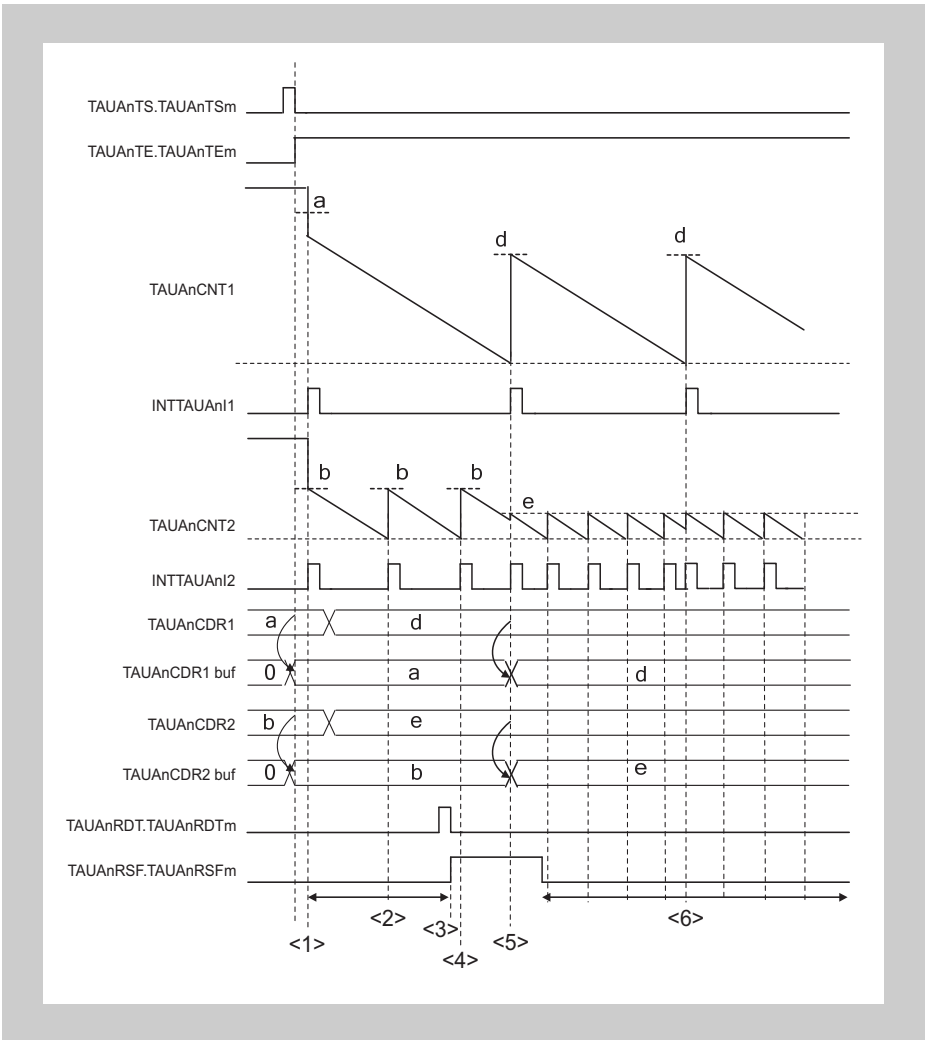


Figure 12-70 General timing diagram for Simultaneous Rewrite Trigger Generation Function Type 1

Description:

1. When TAUAnTS.TAUAnTSm is set to 1, the value of TAUAnCDRm is copied to the TAUAnCDRm buffer.
2. The TAUAnCDRm register can always be written to.
3. The reload data trigger bit (TAUAnRDT.TAUAnRDTm) is set to 1, which sets the status flag (TAUAnRSF.TAUAnRSFm = 1), enabling simultaneous rewriting.
4. Even though simultaneous rewriting is enabled, it does not take place because it is only triggered by an interrupt on channel 1.
5. Simultaneous rewriting is triggered by INT1, which is caused by counter 1 reaching 0000H. The TAUAnCDRm values are loaded to the corresponding TAUAnCDRm buffers.
6. The counter counts down and awaits the next simultaneous rewrite trigger. The values of the TAUAnCDRm registers can be changed again.

(4) Register settings for the upper channel

(a) TAUAnCMORm for the upper channel

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn CKS[1:0]		TAUAn CCS[1:0]		TAUAn MAS	TAUAnSTS[2:0]			TAUAn COS[1:0]		-	TAUAnMD[4:1]				TAUAn MD0

Table 12-72 TAUAnCMORm settings for Simultaneous Rewrite Trigger Generation Function Type 1

Bit name	Setting
TAUAnCKS[1:0]	00: Operation clock = CK0 01: Operation clock = CK1 10: Operation clock = CK2 11: Operation clock = CK3
TAUAnCCS[1:0]	00: Operation clock is used as the count clock
TAUAnMAS	0: Not used, so set to 0
TAUAnSTS[2:0]	000: Counter triggered by software trigger
TAUAnCOS[1:0]	00: Not used, so set to 00
TAUAnMD[4:1]	0000: Interval Timer Mode
TAUAnMD0	0: INTTAUAnIm not generated at operation start 1: Generates INTTAUAnIm at operation start

(b) TAUAnCMURm for the upper channel

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-														TAUAnTIS[1:0]	

Table 12-73 TAUAnCMURm settings for Simultaneous Rewrite Trigger Generation Function Type 1

Bit name	Setting
TAUAnTIS[1:0]	00: Not used, so set to 00

(c) Channel output mode for the upper channel

Because the channel output mode is not used by this function, clear TAUAnTOE.TAUAnTOEm. However, it can be used in Independent Channel Output Mode Controlled by Software.

(d) Simultaneous rewrite for the upper channel

Table 12-74 Simultaneous rewrite settings for Simultaneous Rewrite Trigger Generation Function Type 1

Bit name	Setting
TAUAnRDE.TAUAnRDEm	1: Enables simultaneous rewrite
TAUAnRDS.TAUAnRDSm	1: Selects an upper channel as the control channel for simultaneous rewrite
TAUAnRDM.TAUAnRDMm	0: The signal that controls simultaneous rewrite is loaded when the master channel starts counting
TAUAnRDC.TAUAnRDCm	1: Channel is monitored for an INTTAUAnIm signal that is used as the simultaneous rewrite trigger

(5) Register settings for the lower channel(s)

(a) TAUAnCMORm for the lower channel(s)

For the TAUAnCMORm registers of lower channels, use the TAUAnCMORm register settings of operation modes that can be specified. (For details, refer to *Table 12-7 “Simultaneous rewrite methods and when they are triggered” on page 565*.)

(b) TAUAnCMURm for the lower channel(s)

For the TAUAnCMURm registers of lower channels, use the TAUAnCMURm register settings of operation modes that can be specified. (For details, refer to *Table 12-7 “Simultaneous rewrite methods and when they are triggered” on page 565*.)

(c) Channel output mode for the lower channel(s)

Output is possible according to the operation mode setting (master or slave) of a lower channel.

(d) Simultaneous rewrite for the lower channel(s)

Table 12-75 Simultaneous rewrite settings for the lower channel in Simultaneous Rewrite Trigger Generation Function Type 1

Bit name	Setting
TAUAnRDE.TAUAnRDEm	1: Enables simultaneous rewrite
TAUAnRDS.TAUAnRDSm	1: Selects an upper channel as the control channel for simultaneous rewrite
TAUAnRDM.TAUAnRDMm	0: The signal that controls simultaneous rewrite is loaded when the master channel starts counting
TAUAnRDC.TAUAnRDCm	0: Channel is not monitored for an INTTAUAnIm signal that is used as the simultaneous-rewrite trigger

(6) Operating procedure for Simultaneous Rewrite Trigger Generation Function Type 1

Table 12-76 Operating procedure for Simultaneous Rewrite Trigger Generation Function Type 1

	Operation	Status of TAUAn
Initial channel setting	Set the TAUAnCMORm register and TAUAnCMURm registers for the upper channel as described in *Table 12-72 "TAUAnCMORm settings for Simultaneous Rewrite Trigger Generation Function Type 1" on page 680* and *Table 12-73 "TAUAnCMURm settings for Simultaneous Rewrite Trigger Generation Function Type 1" on page 680* Set the TAUAnCMORm register and TAUAnCMURm registers for the lower channel as described in *(5) "Register settings for the lower channel(s)" on page 681* Set the value of the TAUAnCDRm register	Channel operation is stopped.
Start operation	Set TAUAnTS.TAUAnTSm to 1. TAUAnTS.TAUAnTSm is a trigger bit, so it is automatically cleared to 0.	TAUAnTE.TAUAnTEm is set to 1 and the counter starts. TAUAnCNTm loads the TAUAnCDRm value. When TAUAnCMORm.TAUAnMD0 = 1, INTTAUAnIm is generated.
During operation	TAUAnRDT.TAUAnRDTm can be changed. TAUAnRSF.TAUAnRSFm can be read at all times.	TAUAnCNTm counts down. When the counter reaches 0000_H: • TAUAnCNTm reloads the TAUAnCDRm value, and then continues count operation. • INTTAUAnIm is generated. Simultaneous rewrite is controlled when INTTAUAnIm is generated from the channel where TAUAnRDC.TAUAnRDCm is set to 1. Afterwards, this procedure is repeated.
Stop operation	Set TAUAnTT.TAUAnTTm to 1. TAUAnTT.TAUAnTTm is a trigger bit, so it is automatically cleared to 0.	TAUAnTE.TAUAnTEm is cleared to 0 and the counter stops. TAUAnCNTm stops and retains the current value.

Restart

12.18.2 Simultaneous Rewrite Trigger Generation Function Type 2

(1) Overview

Summary

This function generates an interrupt on a specific channel that can be used by lower channels as a simultaneous rewrite trigger. The interrupt is triggered by a valid TAUAnTTINm input edge or the function starting.

The upper channel generates the simultaneous rewrite trigger (TAUAnRDC.TAUAnRDCm = 1), and the lower channel receives this trigger and performs simultaneous rewriting (TAUAnRDC.TAUAnRDCm = 0).

Prerequisites

- At least two channels used as lower channels than upper channels, each with simultaneous rewriting enabled (TAUAnRDE.TAUAnRDEm = 1)
- The operation mode of the upper channel must be set to Capture Mode. Refer to *Table 12-77 “TAUAnCMORm settings for Simultaneous Rewrite Trigger Generation Function Type 2” on page 687*.
- For details about which operation modes can be specified for lower channels, refer to *Table 12-7 “Simultaneous rewrite methods and when they are triggered” on page 565*.
- The channel output mode of the upper channel must be set to Independent Channel Output Mode Controlled by Software. Refer to *12.8 “Channel Output Modes” on page 577*.
- The channel output mode of the lower channel(s) can be set as desired.

Description

The counter is enabled by setting the channel trigger bit of the upper channel or lower channel (TAUAnTS.TAUAnTSm) to 1. This in turn sets TAUAnTE.TEm to 1, enabling counting. The counter of the upper channel starts to count up, and the counter(s) of the lower channel(s) start to count as specified by their selected operating modes.

When a valid TAUAnTTINm input edge is generated on the upper channel, an interrupt is generated, and an interrupt is generated upon trigger detection on the lower channel.

TAUAnRDC.TAUAnRDCm = 1 on the upper channel, therefore simultaneous rewrite takes place on all lower channels in which simultaneous rewrite is currently possible (TAUAnRSF.TAUAnRSFm = 1).

The values of the data registers are copied to the corresponding data register buffers.

The value of a data register can be changed at any time, but it is only transferred to the corresponding data register buffer when simultaneous rewrite occurs.

Conditions

- The channel that detects the generation of INTTAUAnIm is specified by setting TAUAnRDC.TAUAnRDCm to 1 for the corresponding channel. The TAUAnRDC.TAUAnRDCm bit must be 0 for all other channels on which simultaneous rewriting must take place.
- If the TAUAnCMORm.TAUAnMD0 bit is set to 1, an interrupt is generated when the function starts. For details refer to *12.10 “TAUAnTTOUTm Output and INTTAUAnIm Generation when Counter Starts or Restarts” on page 589*.

(2) Block diagram and general timing diagram

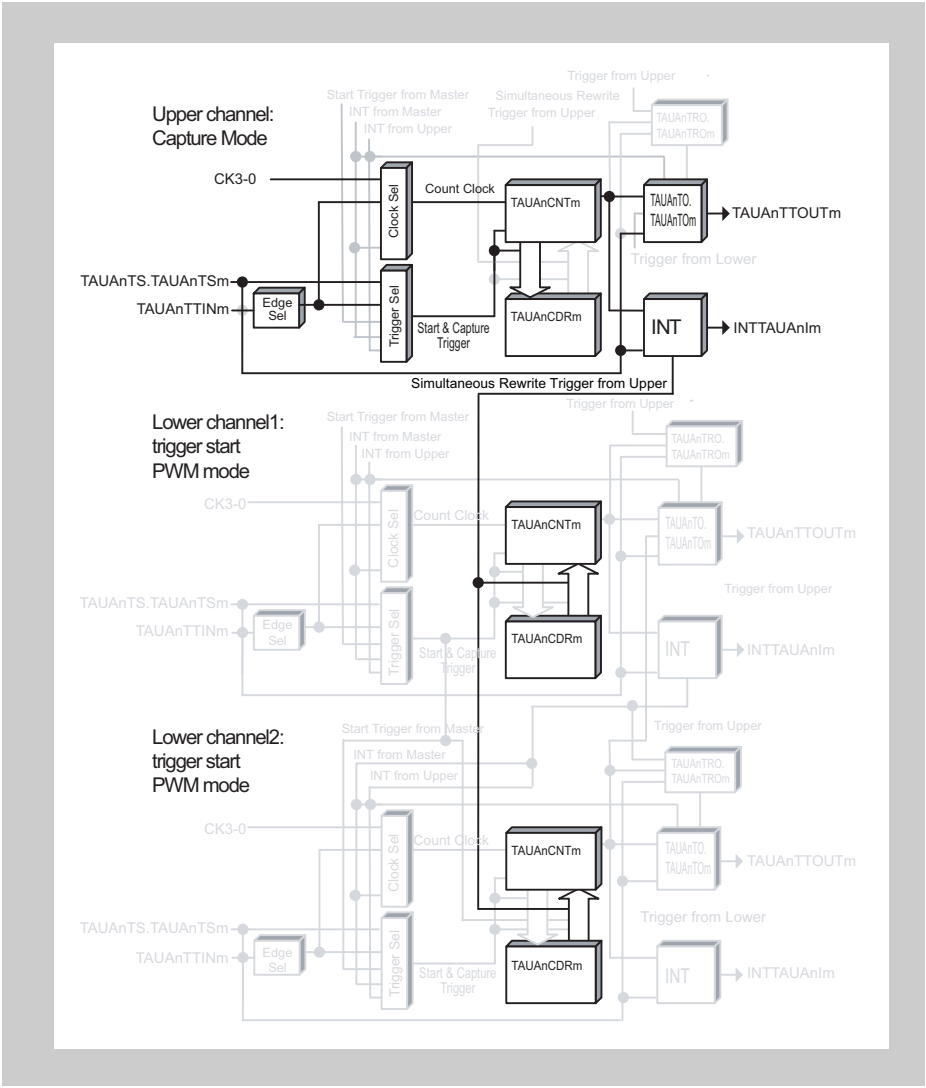


Figure 12-71 Block diagram for Simultaneous Rewrite Trigger Generation Function Type 2

The following settings apply to the general timing diagram:

- INTTAUAnIm not generated at operation start (TAUAnCMORm.TAUAnMD0 = 0)
- Falling edge detection (TAUAnCMURm.TAUAnTIS[1:0] = 00_B)
- Upper channel (channel 1) generates simultaneous rewrite trigger

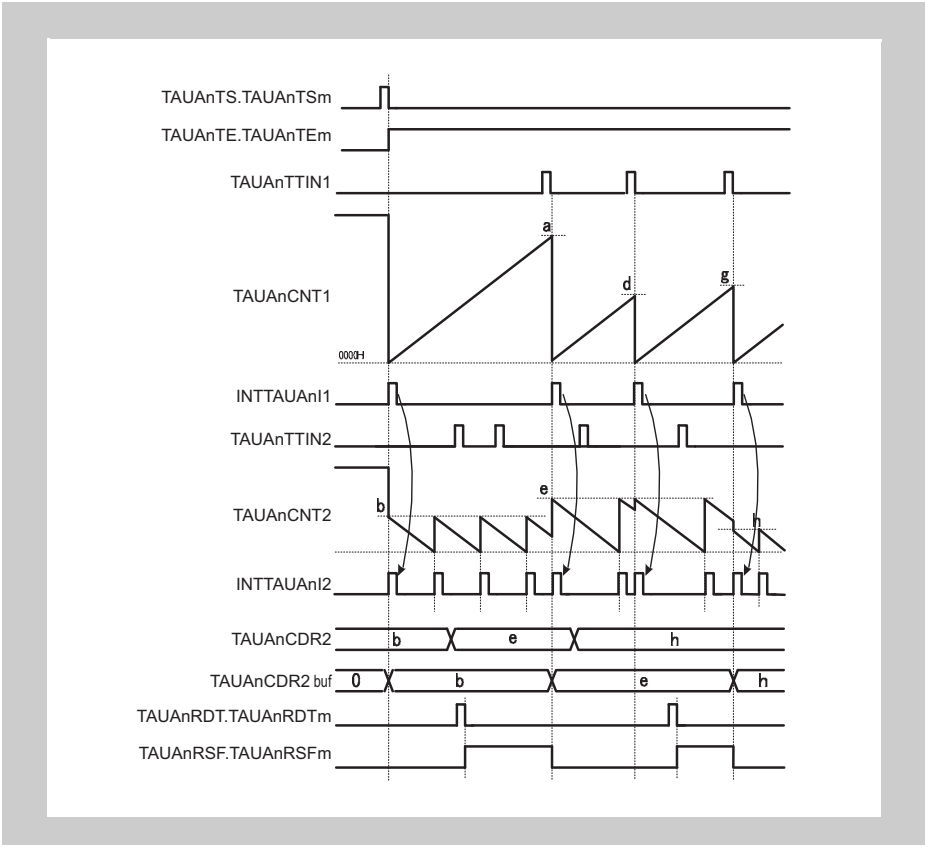


Figure 12-72 General timing diagram for Simultaneous Rewrite Trigger Generation Function Type 2

(3) Register settings for the upper channel

(a) TAUAnCMORm for the upper channel

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn CKS[1:0]		TAUAn CCS[1:0]		TAUAn MAS	TAUAnSTS[2:0]			TAUAn COS[1:0]		-	TAUAnMD[4:1]				TAUAn MD0

Table 12-77 TAUAnCMORm settings for Simultaneous Rewrite Trigger Generation Function Type 2

Bit name	Setting
TAUAnCKS[1:0]	00: Operation clock = CK0 01: Operation clock = CK1 10: Operation clock = CK2 11: Operation clock = CK3
TAUAnCCS[1:0]	00: Operation clock is used as the count clock
TAUAnMAS	0: Not used, so set to 0
TAUAnSTS[2:0]	001:Valid edge of the TAUAnTTINm input signal is the external capture trigger
TAUAnCOS[1:0]	00: Not used, so set to 00
TAUAnMD[4:1]	0010: Capture Mode
TAUAnMD0	0: INTTAUAnIm not generated at operation start 1: Generates INTTAUAnIm at operation start

(b) TAUAnCMURm for the upper channel

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-														TAUAnTIS[1:0]	

Table 12-78 TAUAnCMURm settings for Simultaneous Rewrite Trigger Generation Function Type 2

Bit name	Setting
TAUAnTIS[1:0]	00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection (low width measurement) 11: Rising and falling edge detection (high width measurement)

(c) Channel output mode for the upper channel

The channel output mode is not used by this function. However, it can be used in Independent Channel Output Mode Controlled by Software.

(d) Simultaneous rewrite for the upper channel

Table 12-79 Simultaneous rewrite settings for Simultaneous Rewrite Trigger Generation Function Type 2

Bit name	Setting
TAUAnRDE.TAUAnRDEm	1: Enables simultaneous rewrite
TAUAnRDS.TAUAnRDSm	1: Selects an upper channel as the control channel for simultaneous rewrite
TAUAnRDM.TAUAnRDMm	0: The signal that controls simultaneous rewrite is loaded when the master channel starts counting
TAUAnRDC.TAUAnRDCm	1: Channel is monitored for an INTTAUAnIm signal that is used as the simultaneous rewrite trigger

(4) Register settings for the lower channel(s)

(a) TAUAnCMORm for the lower channel(s)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn CKS[1:0]		TAUAn CCS[1:0]		TAUAn MAS	TAUAnSTS[2:0]			TAUAn COS[1:0]		-	TAUAn MD[4:1]				TAUAn MD0

Table 12-80 TAUAnCMORm settings for Simultaneous Rewrite Trigger Generation Function Type 2

Bit name	Setting
TAUAnCKS[1:0]	00:Prescaler output CK0 01:Prescaler output CK1 10:Prescaler output CK2 11:Prescaler output CK3 The value of the TAUAnCKS[1:0] bits of the master and slave channels must be the same.
TAUAnCCS[1:0]	00:Use the operation clock as the count clock.
TAUAnMAS	1: The channel is the master channel.
TAUAnSTS[2:0]	001:Use the valid TAUAnTTINm input edge signal as the start trigger.
TAUAnCOS[1:0]	00:These are not used, so set them to 00.
TAUAnMD[4:1]	0000: Interval timer mode
TAUAnMD0	1: Generates INTTAUAnIm at operation start

(b) TAUAnCMURm for the lower channel(s)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-														TAUAnTIS[1:0]	

Table 12-81 TAUAnCMURm settings for Simultaneous Rewrite Trigger Generation Function Type 2

Bit name	Setting
TAUAnTIS[1:0]	00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection 11: Setting prohibited

(c) Channel output mode for the lower channel(s)

Output is possible according to the trigger start PWM mode setting.

(d) Simultaneous rewrite for the lower channel(s)

Table 12-82 Simultaneous rewrite settings for the lower channel in Simultaneous Rewrite Trigger Generation Function Type 2

Bit name	Setting
TAUAnRDE.TAUAnRDEm	1: Enables simultaneous rewrite
TAUAnRDS.TAUAnRDSm	1: Selects an upper channel as the control channel for simultaneous rewrite
TAUAnRDM.TAUAnRDMm	0: The signal that controls simultaneous rewrite is loaded when the master channel starts counting
TAUAnRDC.TAUAnRDCm	0: Channel is not monitored for an INTTAUAnIm signal that is used as the simultaneous-rewrite trigger

(5) Operating procedure for simultaneous rewrite trigger generation function type 2

Table 12-83 Operating procedure for simultaneous rewrite trigger generation function type 2

	Operation	Status of TAUAn
Initial channel setting	Set the TAUAnCMORm register and TAUAnCMURm registers for the upper channel as described in *Table 12-77 "TAUAnCMORm settings for Simultaneous Rewrite Trigger Generation Function Type 2" on page 687* and *Table 12-78 "TAUAnCMURm settings for Simultaneous Rewrite Trigger Generation Function Type 2" on page 687* Set up the TAUAnCMORm and TAUAnCMURm registers of the lower channel as shown in *Table 12-80 "TAUAnCMORm settings for Simultaneous Rewrite Trigger Generation Function Type 2" on page 688* and *Table 12-81 "TAUAnCMURm settings for Simultaneous Rewrite Trigger Generation Function Type 2" on page 689*. Set the value of the TAUAnCDRm register.	Channel operation is stopped.
Start operation	Set TAUAnTS.TAUAnTSm to 1. TAUAnTS.TAUAnTSm is a trigger bit, so it is automatically cleared to 0.	TAUAnTE.TAUAnTEm is set to 1 and the counter starts. TAUAnCNTm is cleared to 0000_H. INTTAUAnIm is generated when TAUAnCMORm.TAUAnMD0 is set to 1.
During operation	TAUAnRDT.TAUAnRDTm register can be set at any time. TAUAnRSF.TAUAnRSFm register can be read at any time.	TAUAnCNTm counts up from 0000_H. When a TAUAnTTINm valid edge is detected: • TAUAnCNTm transfers (captures) its value to TAUAnCDRm and returns to 0000_H. • INTTAUAnIm is generated. Simultaneous rewrite is controlled when INTTAUAnIm is generated from the channel where TAUAnRDC.TAUAnRDCm is set to 1. Afterwards, this procedure is repeated.
Stop operation	Set TAUAnTT.TAUAnTTm to 1. TAUAnTT.TAUAnTTm is a trigger bit, so it is automatically cleared to 0.	TAUAnTE.TAUAnTEm is cleared to 0 and the counter stops. TAUAnCNTm stops and retains the current value.

Restart

12.19 Independent Channel One-Phase PWM Function

This chapter describes the One-Phase PWM Function:

- *12.19.1 “One-Phase PWM Output Function”*

12.19.1 One-Phase PWM Output Function

(1) Overview

Summary This function adds dead time to a TAUAnTTINm input signal. The resulting PWM signal is output via TAUAnTTOUTm of the channel and TAUAnTTOUTm of (an) upper channel(s).

Prerequisites

- Two (or more) channels, each with dead time control enabled (TAUAnTDE.TAUAnTDEm = 1)
- The operation mode of the lower channel must be set to One Count Mode. Refer to *Table 12-84 "TAUAnCMORm settings for One-Phase PWM Output Function" on page 695*.
- The operation mode of the upper channel(s) can be set as desired.
- The channel output mode of the upper and lower channels must be set to Synchronous Channel Output Mode 2 with One-Phase PWM Output. Refer to *12.8 "Channel Output Modes" on page 577*.

Description The counter is enabled by setting the channel trigger bit (TAUAnTS.TAUAnTSm) to 1. This in turn sets TAUAnTE.TAUAnTEm = 1, enabling count operation.

The counter starts when a valid TAUAnTTINm input start edge is detected. The value of TAUAnCDRm is written to TAUAnCNTm and the counter starts to count down from the TAUAnCDRm value.

When the counter reaches 0000_H an interrupt is generated. The counter returns to $FFFF_H$ and awaits the next valid TAUAnTTINm input start edge.

Conditions

- The TAUAnCMURm.TAUAnTIS[1:0] bits specify the type of width measurement:
  - TAUAnCMURm.TAUAnTIS[1:0] = 10B:
    Detect both rising and falling edges as valid (low-width measurement).
  - TAUAnCMURm.TAUAnTIS[1:0] = 11B:
    Detect both rising and falling edges as valid (high-width measurement).
- The TAUAnTDL.TAUAnTDLm bit specifies the behavior of TAUAnTTOUTm for each channel when an interrupt or valid TAUAnTTINm edge is detected on the lower channel:
  - If TAUAnTDL.TAUAnTDLm = 0, an interrupt is the trigger for setting TAUAnTTOUTm, and a valid TAUAnTTINm edge is the trigger for resetting TAUAnTTOUTm.
  - If TAUAnTDL.TAUAnTDLm = 1, a valid TAUAnTTINm edge is the trigger for setting TAUAnTTOUTm, and an interrupt is the trigger for resetting TAUAnTTOUTm.
- Forced restart is not possible for this function.

(2) Block diagram and general timing diagram

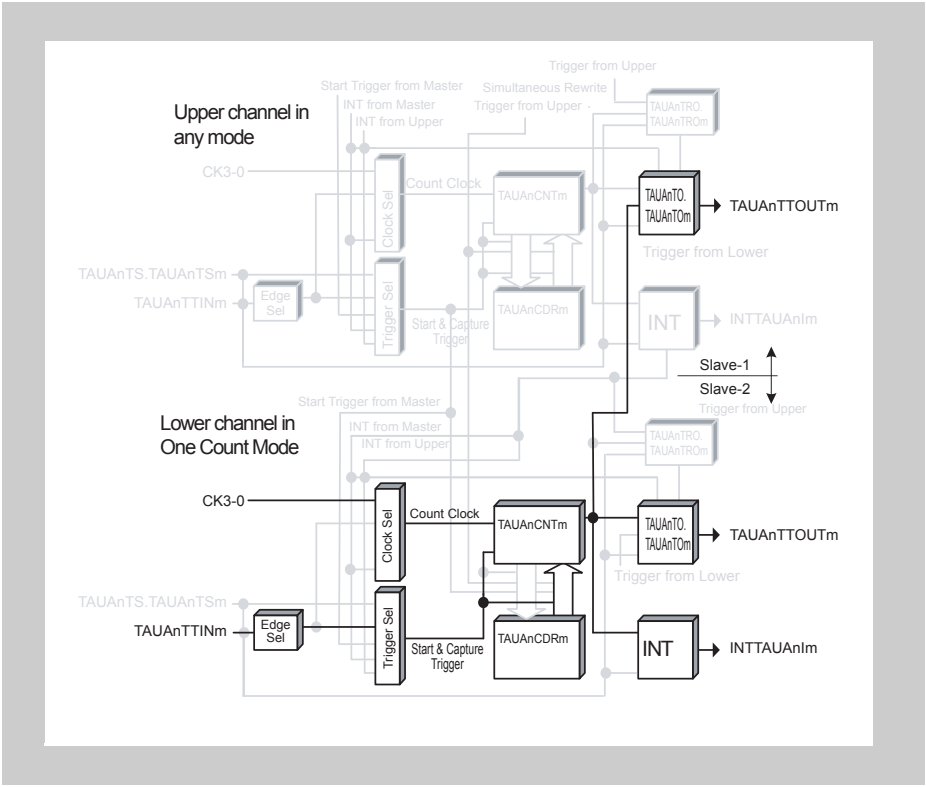


Figure 12-73 Block diagram for One-Phase PWM Output Function

The following settings apply to the general timing diagram:

- Rising and falling edge detection = high width measurement (TAUAnCMURm.TAUAnTIS[1:0] = 11_B)

For these settings, signals are assumed to be active when at high level.

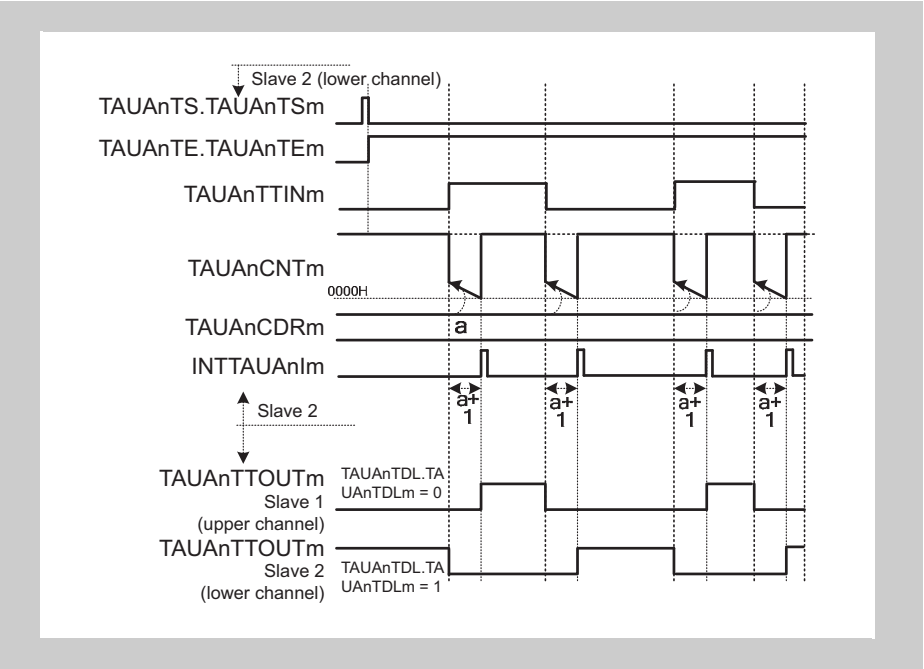


Figure 12-74 General timing diagram for One-Phase PWM Output Function

(3) Register settings for the lower channel

(a) TAUAnCMORm for the lower channel

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn CKS[1:0]		TAUAn CCS[1:0]		TAUAn MAS	TAUAnSTS[2:0]			TAUAn COS[1:0]		-	TAUAnMD[4:1]				TAUAn MD0

Table 12-84 TAUAnCMORm settings for One-Phase PWM Output Function

Bit name	Setting
TAUAnCKS[1:0]	00: Operation clock = CK0 01: Operation clock = CK1 10: Operation clock = CK2 11: Operation clock = CK3
TAUAnCCS[1:0]	00: Operation clock is used as the count clock
TAUAnMAS	0: Not used, so set to 0
TAUAnSTS[2:0]	001: Valid edge of the TAUAnTTINm input signal is the external start trigger
TAUAnCOS[1:0]	00: Not used, so set to 00
TAUAnMD[4:1]	0100: One Count Mode
TAUAnMD0	1: Enables start trigger detection during counting

(b) TAUAnCMURm for the lower channel

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-														TAUAnTIS[1:0]	

Table 12-85 TAUAnCMURm settings for One-Phase PWM Output Function

Bit name	Setting
TAUAnTIS[1:0]	10: Rising and falling edge detection (low width measurement) 11: Rising and falling edge detection (high width measurement)

(c) Channel output mode for the lower channel

Table 12-86 Control bit settings for Synchronous Channel Output Mode 2 with One-Phase PWM Output

Bit name	Setting
TAUAnTOE.TAUAnTOEm	1: Enables Independent Channel Output Mode
TAUAnTOM.TAUAnTOMm	1: Synchronous channel output
TAUAnTOC.TAUAnTOCm	1: Operation mode 2
TAUAnTOL.TAUAnTOLm	0: Positive logic 1: Inverted logic
TAUAnTDE.TAUAnTDEm	1: Enables dead time operation
TAUAnTDM.TAUAnTDMm	1: Dead time is added upon detection of a TAUAnTTINm input edge from a lower odd channel
TAUAnTDL.TAUAnTDLm	0: An interrupt is the TAUAnTTOUTm set trigger and a valid TAUAnTTINm edge is the TAUAnTTOUTm reset trigger 1: A valid TAUAnTTINm edge is the TAUAnTTOUTm set trigger and an interrupt is the TAUAnTTOUTm reset trigger
TAUAnTRE.TAUAnTREm	0: Disables real-time output
TAUAnTRO.TAUAnTROm	0: When real-time output is disabled (TAUAnTRE.TAUAnTREm = 0), set these bits to 0
TAUAnTRC.TAUAnTRCm	
TAUAnTME.TAUAnTMEm	0: Disables modulation

Caution For TAUAnTDL.TAUAnTDLm, specify the setting that is opposite that of the upper channel.

(d) Simultaneous rewrite for the lower channel

The simultaneous rewrite registers (TAUAnRDE, TAUAnRDS, TAUAnRDM, and TAUAnRDC) cannot be used with the One-Phase PWM Output Function. Therefore, these registers must be set to 0.

Table 12-87 Simultaneous rewrite settings for One-Phase PWM Output Function

Bit name	Setting
TAUAnRDE.TAUAnRDEm	0: Disables simultaneous rewrite
TAUAnRDS.TAUAnRDSm	0: When simultaneous rewrite is disabled (TAUAnRDE.TAUAnRDEm = 0), set these bits to 0
TAUAnRDM.TAUAnRDMm	
TAUAnRDC.TAUAnRDCm	

(4) Register settings for the upper channel(s)

(a) TAUAnCMORm for the upper channel(s)

The TAUAnCMORm register of the upper channel(s) can be set arbitrarily.

(b) TAUAnCMURm for the upper channel(s)

The TAUAnCMURm register of the upper channel(s) can be set arbitrarily.

(c) Channel output mode for the upper channel(s)

Table 12-88 Control bit settings for upper channel(s) for Synchronous Channel Output Mode 2 with One-Phase PWM Output

Bit name	Setting
TAUAnTOE.TAUAnTOEm	1: Enables Independent Channel Output Mode
TAUAnTOM.TAUAnTOMm	1: Synchronous channel output
TAUAnTOC.TAUAnTOCm	1: Operation mode 2
TAUAnTOL.TAUAnTOLm	0: Positive logic 1: Inverted logic
TAUAnTDE.TAUAnTDEm	1: Enables dead time operation
TAUAnTDM.TAUAnTDMm	1: Dead time is added upon detection of a TAUAnTTINm input edge from a lower odd channel
TAUAnTDL.TAUAnTDLm	0: An interrupt on the lower channel is the TAUAnTTOUTm set trigger and a valid TAUAnTTINm edge on the lower channel is the TAUAnTTOUTm reset trigger 1: A valid TAUAnTTINm edge on the lower channel is the TAUAnTTOUTm set trigger and an interrupt on the lower channel is the TAUAnTTOUTm reset trigger
TAUAnTRE.TAUAnTREm	0: Disables real-time output
TAUAnTRO.TAUAnTROm	0: When real-time output is disabled (TAUAnTRE.TAUAnTREm = 0), set these bits to 0
TAUAnTRC.TAUAnTRCm	
TAUAnTME.TAUAnTMEm	0: Disables modulation

Caution For TAUAnTDL.TAUAnTDLm, specify the setting that is opposite that of the lower channel.

(d) Simultaneous rewrite for the upper channel(s)

The simultaneous rewrite registers of the upper channel(s) can be set arbitrarily.

(5) Operating procedure for One-phase PWM Output Function

Table 12-89 Operating procedure for One-phase PWM Output Function

	Operation	Status of TAUAn
Initial channel setting	Set the TAUAnCMORm register and TAUAnCMURm registers for the lower channel as described in *Table 12-84 "TAUAnCMORm settings for One-Phase PWM Output Function" on page 695* and *Table 12-85 "TAUAnCMURm settings for One-Phase PWM Output Function" on page 695* Set the TAUAnCMORm register and TAUAnCMURm registers for the upper channel as described in *(4) "Register settings for the upper channel(s)" on page 697* Set the value of the TAUAnCDRm register Set the channel output mode for the upper and lower channel by setting the control bits as described in *Table 12-86 "Control bit settings for Synchronous Channel Output Mode 2 with One-Phase PWM Output" on page 696*	Channel operation is stopped.
Start operation	Set TAUAnTOE.TAUAnTOEm (slave channels 1 and 2) to 1 (at operation restart only). Set TAUAnTS.TAUAnTSm = 1 for slave channel 2. TAUAnTS.TAUAnTSm is a trigger bit, so it is automatically cleared to 0. Detection of TAUAnTTINm start edge	TAUAnTE.TAUAnTEm is set to 1 (slave channel 2) and TAUAnCNTm waits for TAUAnTTINm start edge detection. TAUAnCNTm loads the TAUAnCDRm value.
During operation	The value of the TAUAnCDRm register can be changed at any time. The TAUAnCNTm register can be read at any time.	TAUAnCNTm of slave channel 2 counts down. When it reaches 0000_H: • INTTAUAnIm is generated. • TAUAnCNTm stops counting. The TAUAnTTOUTm level is changed through the TAUAnTTINm edge detection signal and the INTTAUAnIm signal from slave channel 2 to output a one-phase PWM waveform with dead time. Afterwards, this procedure is repeated.
Stop operation	Set TAUAnTT.TAUAnTTm = 1 for slave channel 2. TAUAnTT.TAUAnTTm is a trigger bit, so it is automatically cleared to 0.	TAUAnTE.TAUAnTEm is cleared to 0 and the counter stops. TAUAnCNTm stops and retains the current value.

Restart

12.20 Other Independent Channel Functions

This chapter describes a function that generates an interrupt when a certain number of TAUAnTTINm pulses has occurred, a function that divides the frequency of TAUAnTTINm, and a function that measures the duration between the function start and a TAUAnTTINm input signal:

- *12.20.1 “External Event Count Function”*
- *12.20.2 “Clock Divide Function”*
- *12.20.3 “TAUAnTTINm Input Position Detection Function”*

12.20.1 External Event Count Function

(1) Overview

Summary This function is used as an event timer. It generates an interrupt (INTTAUAnIm) when a specific number of TAUAnTTINm input pulses has occurred.

Prerequisites
- The operation mode must be set to Event Count Mode. Refer to *Table 12-90 "TAUAnCMORm settings for External Event Count Function" on page 702*.
- TAUAnTTOUTm is not used for this function.

Description The counter is enabled by setting the channel trigger bit (TAUAnTS.TAUAnTSm) to 1. This in turn sets TAUAnTE.TAUAnTEm = 1, enabling count operation. When the counter starts, the current value of TAUAnCDRm is loaded to TAUAnCNTm.

When a valid TAUAnTTINm input edge is detected, the value of TAUAnCNTm reduces by 1. TAUAnCNTm retains this value until a valid TAUAnTTINm input edge is detected or the counter is restarted.

When the counter value reaches 0000_H, INTTAUAnIm is generated. TAUAnCNTm loads the TAUAnCDRm value, and then subsequently continues operation.

The counter can be stopped by setting TAUAnTT.TAUAnTTm to 1, which in turn sets TAUAnTE.TAUAnTEm to 0. TAUAnCNTm stops and retains its value. The counter can be restarted by setting TAUAnTS.TAUAnTSm to 1. The counter can also be restarted without stopping it first (forced restart) by setting TAUAnTS.TAUAnTSm to 1 during operation.

The value of TAUAnCDRm can be rewritten at any time, and the changed value of TAUAnCDRm is applied the next time the counter starts to count down.

Conditions The type of edge used as the trigger is specified by the TAUAnCMURm.TAUAnTIS[1:0] bits:
- If TAUAnCMURm.TAUAnTIS[1:0] = 00_B, the falling edges are counted.
- If TAUAnCMURm.TAUAnTIS[1:0] = 01_B, the rising edges are counted.
- If TAUAnCMURm.TAUAnTIS[1:0] = 10_B, the rising and falling edges are counted.

(2) Equations

Number of valid edges,
detected before INTTAUAnIm is generated = TAUAnCDRm + 1

(3) Block diagram and general timing diagram

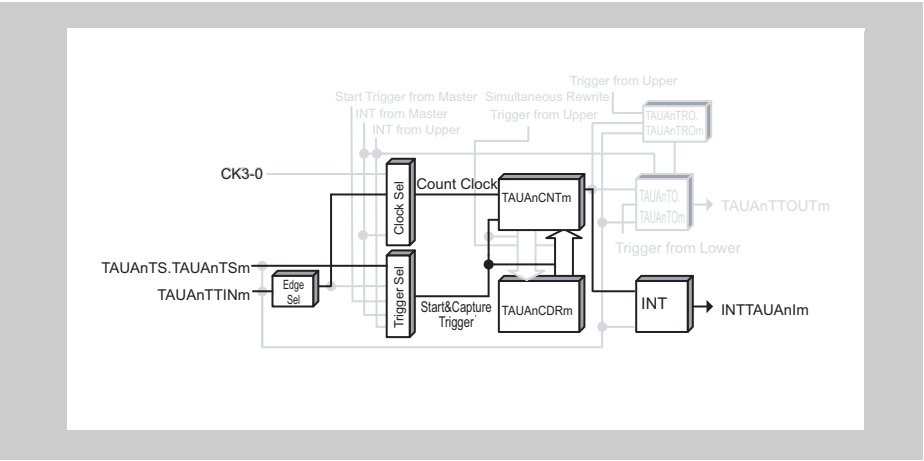


Figure 12-75 Block diagram for External Event Count Function

The following settings apply to the general timing diagram:

- Rising edge detection (TAUAnCMURm.TAUAnTIS[1:0] = 01_B)

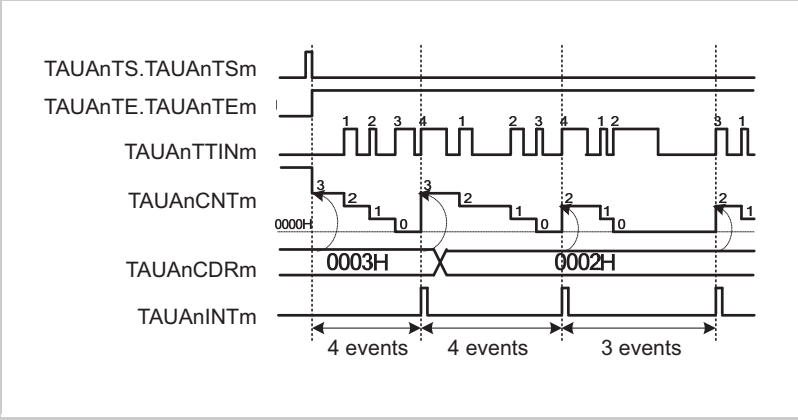


Figure 12-76 General timing diagram for External Event Count Function

(4) Register settings

(a) TAUAnCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn CKS[1:0]		TAUAn CCS[1:0]		TAUAn MAS	TAUAnSTS[2:0]			TAUAn COS[1:0]		-	TAUAnMD[4:1]				TAUAn MD0

Table 12-90 TAUAnCMORm settings for External Event Count Function

Bit name	Setting
TAUAnCKS[1:0]	00: Operation clock = CK0 01: Operation clock = CK1 10: Operation clock = CK2 11: Operation clock = CK3
TAUAnCCS[1:0]	01: Valid TAUAnTTINm input edge is used as the count clock
TAUAnMAS	0: Not used, so set to 0
TAUAnSTS[2:0]	000: Counter triggered by software trigger
TAUAnCOS[1:0]	00: Not used, so set to 00
TAUAnMD[4:1]	0011: Event Count Mode
TAUAnMD0	0: INTTAUAnIm not generated at operation start

(b) TAUAnCMURm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-														TAUAnTIS[1:0]	

Table 12-91 TAUAnCMURm settings for External Event Count Function

Bit name	Setting
TAUAnTIS[1:0]	00: Falling edge 01: Rising edge 10: Rising and falling edge

(c) Channel output mode

The channel output mode is not used by this function. However, it can be used in Independent Channel Output Mode Controlled by Software.

(d) Simultaneous rewrite

The simultaneous rewrite registers (TAUAnRDE, TAUAnRDS, TAUAnRDM, and TAUAnRDC) cannot be used with the External Event Count Function. Therefore, these registers must be set to 0.

Table 12-92 Simultaneous rewrite settings for External Event Count Function

Bit name	Setting
TAUAnRDE.TAUAnRDEm	0: Disables simultaneous rewrite
TAUAnRDS.TAUAnRDSm	0: When simultaneous rewrite is disabled (TAUAnRDE.TAUAnRDEm = 0), set these bits to 0
TAUAnRDM.TAUAnRDMm	
TAUAnRDC.TAUAnRDCm	

(5) Operating procedure for External Event Count Function

Table 12-93 Operating procedure for External Event Count Function

	Operation	Status of TAUAn
Initial channel setting	Set the TAUAnCMORm register and TAUAnCMURm registers as described in *Table 12-90 "TAUAnCMORm settings for External Event Count Function" on page 702* and *Table 12-91 "TAUAnCMURm settings for External Event Count Function" on page 702* Set the value of the TAUAnCDRm register	Channel operation is stopped.
Start operation	Set TAUAnTS.TAUAnTSm to 1. TAUAnTS.TAUAnTSm is a trigger bit, so it is automatically cleared to 0.	TAUAnTE.TAUAnTEm is set to 1 and the counter starts. TAUAnCNTm loads the TAUAnCDRm value, and then waits for detection of the TAUAnTTINm input edge.
During operation	Detection of TAUAnTTINm edges. The value of TAUAnCDRm can be changed at any time. The TAUAnCNTm register can be read at any tme.	TAUAnCNTm performs count-down operation each time a TAUAnTTINm input edge is detected. When the counter reaches 0000_H: • TAUAnCNTm loads the TAUAnCDRm value, and then continues count operation. • INTTAUAnIm is generated. Afterwards, this procedure is repeated.
Stop operation	Set TAUAnTT.TAUAnTTm to 1. TAUAnTT.TAUAnTTm is a trigger bit, so it is automatically cleared to 0.	TAUAnTE.TAUAnTEm is cleared to 0 and the counter stops. TAUAnCNTm stops and retains its current value.

Restart

(6) Specific timing diagrams

(a) TAUAnCDRm = 0000_H

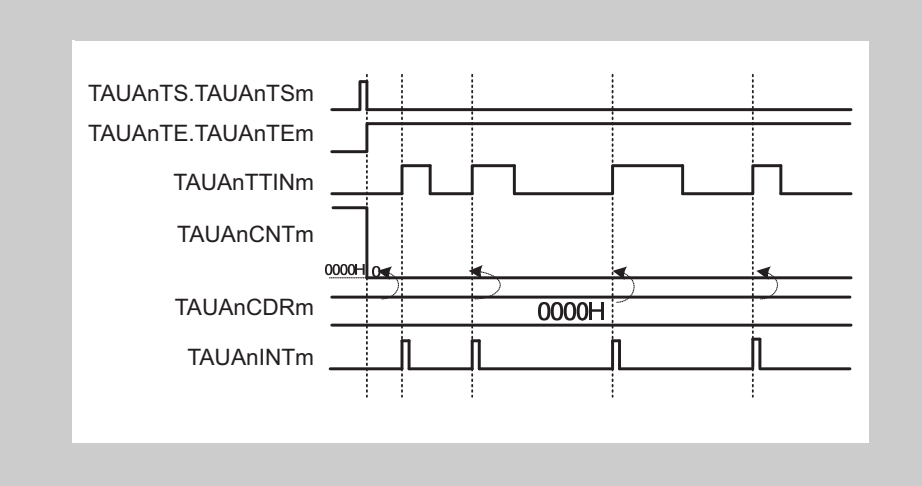


Figure 12-77 TAUAnCDRm = 0000H, TAUAnCMURm.TAUAnTIS[1:0] = 01_B

- If 0000_H = TAUAnCDRm, 0000_H is loaded to TAUAnCNTm every time a valid TAUAnTTINm input edge is detected.

  This means, INTTAUAnIm is generated every time a valid TAUAnTTINm input edge is detected.

(b) Operation stop and restart

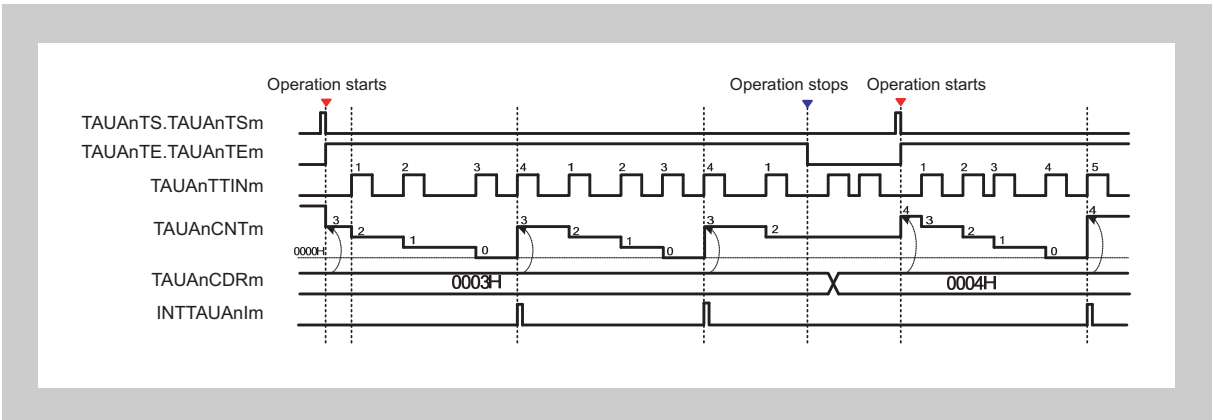


Figure 12-78 Operation stop and restart, TAUAnCMURm.TAUAnTIS[1:0] = 01_B

- The counter can be stopped by setting TAUAnTT.TAUAnTTm to 1, which in turn sets TAUAnTE.TAUAnTEm to 0.
- TAUAnCNTm stops and the current value is retained. TAUAnTTINm continues and TAUAnCNTm ignores the valid edge.
- The counter can be restarted by setting TAUAnTS.TAUAnTSm to 1. TAUAnCNTm loads the TAUAnCDRm value, and then restarts count operation.

(c) Forced restart

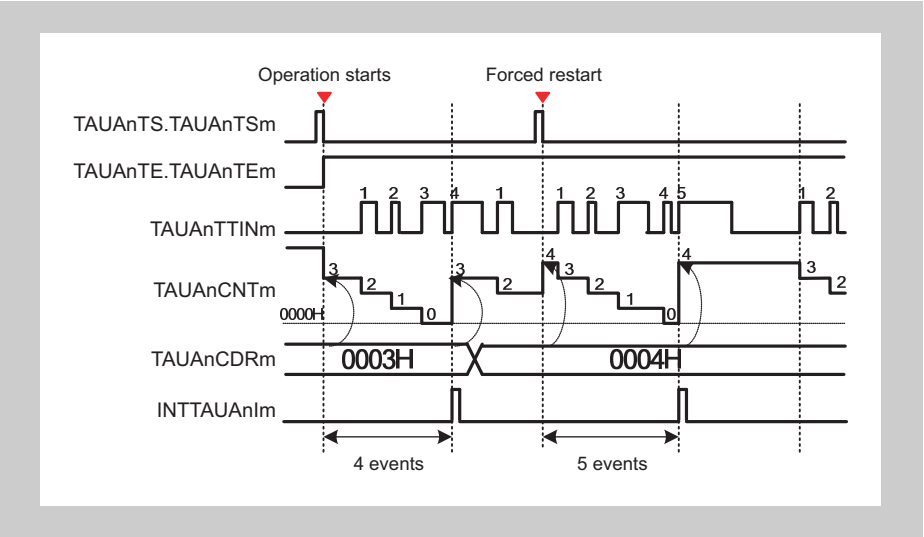


Figure 12-79 Forced restart, TAUAnCMURm.TAUAnTIS[1:0] = 01_B

A forced restart applies a change to TAUAnCDRm immediately.

- The counter can be restarted (without stopping it first), by setting TAUAnTS.TAUAnTSm to 1 during operation.
- The value of TAUAnCDRm is loaded to TAUAnCNTm and the counter awaits the next valid TAUAnTTINm input edge.

12.20.2 Clock Divide Function

(1) Overview

Summary This function is used as a frequency divider. The frequency of the input signal TAUAnTTINm is divided by a factor related to TAUAnCDRm, and the resulting signal is output to TAUAnTTOUTm.

Prerequisites

- TAUAnTTINm must have a fixed frequency.
- The operation mode must be set to Interval Timer Mode. Refer to *Table 12-94 "TAUAnCMORm settings for Clock Divide Function" on page 709*.
- The channel output mode must be set to Independent Channel Output Mode 1. Refer to *12.8 "Channel Output Modes" on page 577*.

Description The counter is started by setting the channel trigger bit (TAUAnTS.TAUAnTSm) to 1. This in turn sets TAUAnTE.TAUAnTEm = 1, enabling count operation. The current value of TAUAnCDRm is loaded to TAUAnCNTm and the counter starts to count down from this value, using TAUAnTTINm as the count clock.

When the counter value reaches 0000_H, INTTAUAnIm is generated and the TAUAnTTOUTm signal toggles. TAUAnCNTm loads the TAUAnCDRm value, and then subsequently continues operation.

The value of TAUAnCDRm can be rewritten at any time, and the changed value of TAUAnCDRm is applied the next time the function starts to count down.

The counter can be stopped by setting TAUAnTT.TAUAnTTm = 1, which in turn sets TAUAnTE.TAUAnTEm = 0. TAUAnCNTm and TAUAnTTOUTm stop but retain their values. The function can be restarted by setting TAUAnTS.TAUAnTSm = 1. The counter can also be forcibly restarted (without stopping it first) by setting TAUAnTS.TAUAnTSm = 1 during operation.

Conditions If the TAUAnCMORm.TAUAnMD0 bit is set to 0, the first interrupt after a start or restart is not generated, and therefore TAUAnTTOUTm does not toggle. This results in an inverted TAUAnTTOUTm signal compared to when TAUAnCMORm.TAUAnMD0 is set to 1. For details refer to *12.10 "TAUAnTTOUTm Output and INTTAUAnIm Generation when Counter Starts or Restarts" on page 589*.

Note The input TAUAnTTINm is sampled at the frequency of the operation clock, specified by TAUAnCMORm.TAUAnCKS[1:0] bits. As a result, the output cycle of TAUAnTTOUTm has an error of ± 1 operation clock cycle.

(2) Equations

- When rising edge detection is selected:
  TAUAnTTOUTm frequency = TAUAnTTINm frequency / [(TAUAnCDRm + 1) x 2]
- When falling edge detection is selected:
  TAUAnTTOUTm frequency = TAUAnTTINm frequency / [(TAUAnCDRm + 1) x 2]
- When rising and falling edge detection is selected:
  TAUAnTTOUTm frequency = TAUAnTTINm frequency / (TAUAnCDRm + 1)

(3) Block diagram and general timing diagram

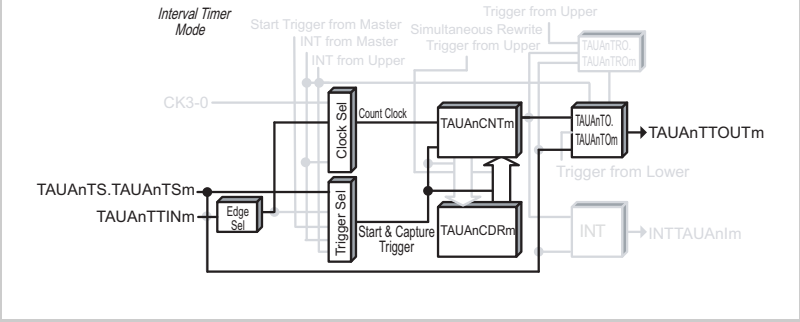


Figure 12-80 Block diagram for Clock Divide Function

The following settings apply to the general timing diagram:

- INTTAUAnIm generated at operation start (TAUAnCMORm.TAUAnMD0 = 1)
- Rising edge detection (TAUAnCMURm.TAUAnTIS[1:0] = 01_B)

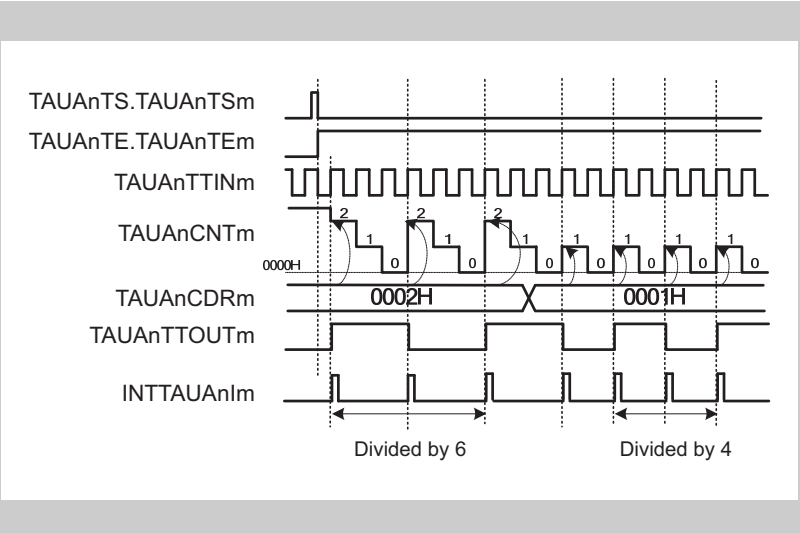


Figure 12-81 General timing diagram for Clock Divide Function

(4) Register settings

(a) TAUAnCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn CKS[1:0]		TAUAn CCS[1:0]		TAUAn MAS	TAUAn STS[2:0]			TAUAn COS[1:0]		-	TAUAnMD[4:1]				TAUAn MD0

Table 12-94 TAUAnCMORm settings for Clock Divide Function

Bit name	Setting
TAUAnCKS[1:0]	00: Operation clock = CK0 01: Operation clock = CK1 10: Operation clock = CK2 11: Operation clock = CK3
TAUAnCCS[1:0]	01: Valid TAUAnTTINm input edge is used as the count clock
TAUAnMAS	0: Not used, so set to 0
TAUAnSTS[2:0]	000: Counter triggered by software trigger
TAUAnCOS[1:0]	00: Not used, so set to 00
TAUAnMD[4:1]	0000: Interval Timer Mode
TAUAnMD0	0: INTTAUAnIm not generated and TAUAnTTOUTm does not toggle at operation start 1: Generates INTTAUAnIm and toggles TAUAnTTOUTm at operation start

(b) TAUAnCMURm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-														TAUAnTIS[1:0]	

Table 12-95 TAUAnCMURm settings for Clock Divide Function

Bit name	Setting
TAUAnTIS[1:0]	00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection

(c) Channel output mode

Table 12-96 Control bit settings for Independent Channel Output Mode 1

Bit name	Setting
TAUAnTOE.TAUAnTOEm	1: Enables Independent Channel Output Mode
TAUAnTOM.TAUAnTOMm	0: Independent channel output
TAUAnTOC.TAUAnTOCm	0: Operation mode 1 (= Toggle mode if TAUAnTOM.TAUAnTOMm = 0)
TAUAnTOL.TAUAnTOLm	0: Positive logic
TAUAnTDE.TAUAnTDEm	0: Disables dead time operation
TAUAnTDM.TAUAnTDMm	0: When dead time operation is disabled (TAUAnTDE.TAUAnTDEm = 0), set these bits to 0
TAUAnTDL.TAUAnTDLm	
TAUAnTRE.TAUAnTREm	0: Disables real-time output
TAUAnTRO.TAUAnTROm	0: When real-time output is disabled (TAUAnTRE.TAUAnTREm = 0), set these bits to 0
TAUAnTRC.TAUAnTRCm	
TAUAnTME.TAUAnTMEm	0: Disables modulation

(d) Simultaneous rewrite

The simultaneous rewrite registers (TAUAnRDE, TAUAnRDS, TAUAnRDM, and TAUAnRDC) cannot be used with the Clock Divide Function. Therefore, these registers must be set to 0.

Table 12-97 Simultaneous rewrite settings for Clock Divide function

Bit name	Setting
TAUAnRDE.TAUAnRDEm	0: Disables simultaneous rewrite
TAUAnRDS.TAUAnRDSm	0: When simultaneous rewrite is disabled (TAUAnRDE.TAUAnRDEm = 0), set these bits to 0
TAUAnRDM.TAUAnRDMm	
TAUAnRDC.TAUAnRDCm	

(5) Operating procedure for Clock Divide Function

Table 12-98 Operating procedure for Clock Divide Function

	Operation	Status of TAUAn
Initial channel setting	Set the TAUAnCMORm register and TAUAnCMURm registers as described in *Table 12-94 "TAUAnCMORm settings for Clock Divide Function" on page 709* and *Table 12-95 "TAUAnCMURm settings for Clock Divide Function" on page 709* Set the value of the TAUAnCDRm register Set the channel output mode by setting the control bits as described in *Table 12-96 "Control bit settings for Independent Channel Output Mode 1" on page 710*	Channel operation is stopped.
Start operation	Set TAUAnTS.TAUAnTSm to 1. TAUAnTS.TAUAnTSm is a trigger bit, so it is automatically cleared to 0.	TAUAnTE.TAUAnTEm is set to 1 and the counter starts. TAUAnCNTm loads the TAUAnCDRm value. When TAUAnCMORm.TAUAnMD0 is set to 1, INTTAUAnIm is generated and TAUAnTTOUTm toggles.
During operation	The value of TAUAnCDRm can be changed at any time. The TAUAnCNTm register can be read at all times.	When a TAUAnTTINm input edge is detected, TAUAnCNTm counts down. When the counter reaches 0000_H: • TAUAnCNTm loads the TAUAnCDRm value, and then continues count operation. • INTTAUAnIm is generated. • TAUAnTTOUTm toggles. Afterwards, this procedure is repeated.
Stop operation	Set TAUAnTT.TAUAnTTm to 1. TAUAnTT.TAUAnTTm is a trigger bit, so it is automatically cleared to 0.	TAUAnTE.TAUAnTEm is cleared to 0 and the counter stops. TAUAnCNTm stops and both it and TAUAnTTOUTm retain their current values.

Restart

(6) Specific timing diagrams

(a) TAUAnCDRm = 0000_H

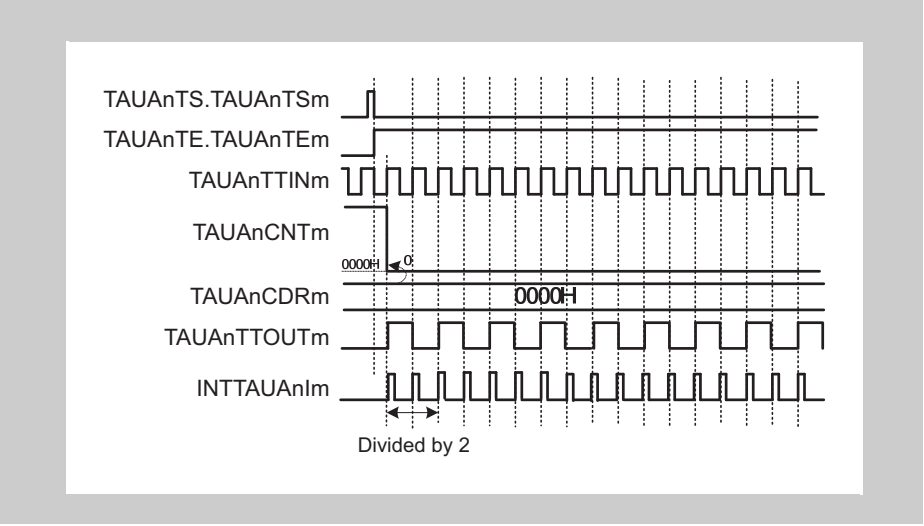


Figure 12-82 TAUAnCDRm = 0000_H, TAUAnCMORm.TAUAnMD0 = 1, TAUAnCMURm.TAUAnTIS[1:0] = 01_B

- If TAUAnCDRm is 0000_H, TAUAnCNTm is also always 0000_H.
- INTTAUAnIm is generated every count clock, resulting in TAUAnTTOUTm toggling every count clock.

Figure 12-82 "TAUAnCDRm = 0000_H, TAUAnCMORm.TAUAnMD0 = 1, TAUAnCMURm.TAUAnTIS[1:0] = 01_B" shows the approximate operation timing. In actuality, because there is delay due to the noise filter and synchronizer between the TAUAnIm pin and TAUAn, there is delay between TINm detection and TOUTm output.

(b) Restart

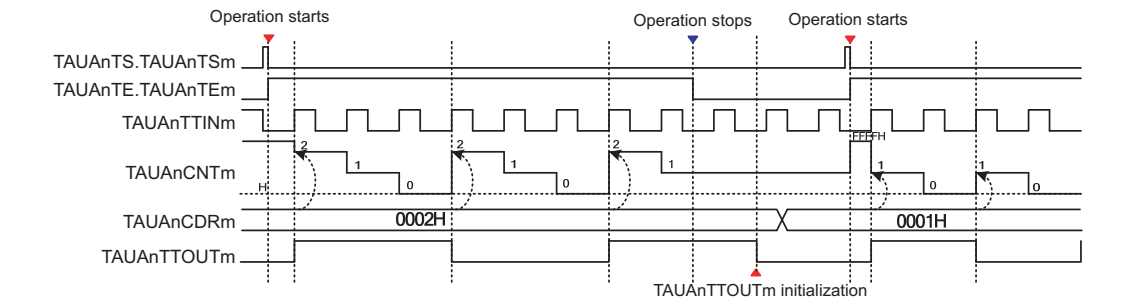


Figure 12-83 Restart, TAUAnCMORm.TAUAnMD0 = 1, TAUAnCMURm.TAUAnTIS[1:0] = 01_B

To reset the value of TAUAnTTOUTm:

- Set TAUAnTOE.TAUAnTOEm = 0 when the counter is stopped (TAUAnTE.TAUAnTEm = 0)
- Then write the either 0 or 1 to TAUAnTO.TAUAnTOm to set the new start value of TAUAnTTOUTm

12.20.3 TAUAnTTINm Input Position Detection Function

(1) Overview

Summary This function measures the duration between the function start and a TAUAnTTINm input signal.

Prerequisites
- The operation mode must be set to Count Capture Mode. Refer to *Table 12-99 "TAUAnCMORm settings for TAUAnTTINm Input Position Detection Function" on page 715*.
- TAUAnTTOUTm is not used for this function.

Description The counter is enabled by setting the channel trigger bit (TAUAnTS.TAUAnTSm) to 1. This in turn sets TAUAnTE.TAUAnTEm = 1, enabling count operation. The counter starts to count from 0000_H. When a valid TAUAnTTINm input stop edge is detected, the current TAUAnCNTm value is loaded to TAUAnCDRm and an interrupt (INTTAUAnIm) is generated. The counter continues counting.

When the counter reaches $FFFF_H$, the bit TAUAnCSRm.TAUAnOVF is set to 1 and the counter restarts from 0000_H. TAUAnCSRm.TAUAnOVF is cleared by setting TAUAnCSCm.CLOV.

Conditions If the TAUAnCMORm.TAUAnMD0 bit is set to 0, the first interrupt after a start or restart is not generated. For details refer to *12.10 "TAUAnTTOUTm Output and INTTAUAnIm Generation when Counter Starts or Restarts" on page 589*.

(2) Equations

Function duration at a TAUAnTTINm input pulse =

count clock cycle × [($FFFF_H$+1 × TAUAnCSRm.TAUAnOVF) + (TAUAnCDRm capture value + 1)]

<R> **Caution** If a capture signal in input when the counter value reaches the upper limit (0FFFFH), the overflow flag (OVF) that should be set in response to the occurrence of an overflow in the next count clock cycle will not be set. The OVF flag is set normally at all other times.

(3) Block diagram and general timing diagram

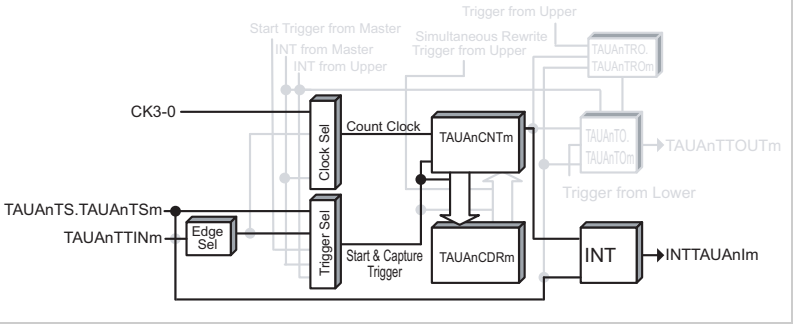


Figure 12-84 Block diagram for TAUAnTTINm Input Position Detection Function

The following settings apply to the general timing diagram:

- INTTAUAnIm not generated at operation start (TAUAnCMORm.TAUAnMD0 = 0)
- Falling edge detection (TAUAnCMURm.TAUAnTIS[1:0] = 00_B)

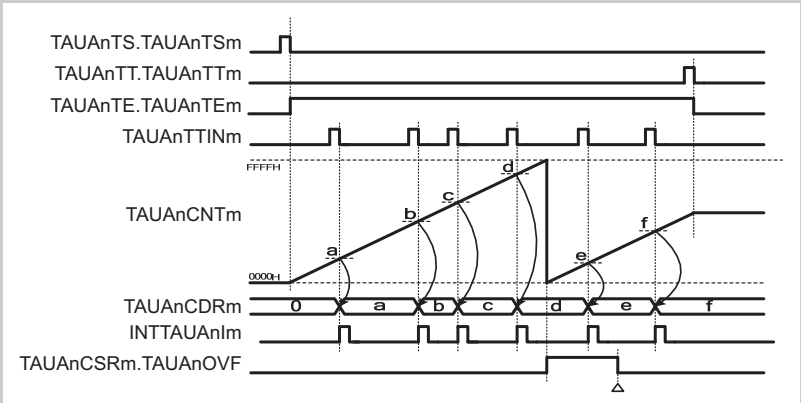


Figure 12-85 General timing diagram for TAUAnTTINm Input Position Detection Function

(4) Register settings

(a) TAUAnCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn CKS[1:0]		TAUAn CCS[1:0]		TAUAn MAS	TAUAnSTS[2:0]			TAUAn COS[1:0]		-	TAUAnMD[4:1]				TAUAn MD0

Table 12-99 TAUAnCMORm settings for TAUAnTTINm Input Position Detection Function

Bit name	Setting
TAUAnCKS[1:0]	00: Operation clock = CK0 01: Operation clock = CK1 10: Operation clock = CK2 11: Operation clock = CK3
TAUAnCCS[1:0]	00: Operation clock is used as the count clock
TAUAnMAS	0: Not used, so set to 0
TAUAnSTS[2:0]	001: Valid TAUAnTTINm input edge signal is used as the external capture trigger
TAUAnCOS[1:0]	01: Overflow (TAUAnCSRm.OVF) set upon counter overflow and cleared by a CPU instruction
TAUAnMD[4:1]	1011: Count Capture Mode
TAUAnMD0	0: INTTAUAnIm not generated at operation start 1: Generates INTTAUAnIm at operation start

(b) TAUAnCMURm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-														TAUAnTIS[1:0]	

Table 12-100 TAUAnCMURm settings for TAUAnTTINm Input Position Detection Function

Bit name	Setting
TAUAnTIS[1:0]	00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection

(c) Channel output mode

The channel output mode is not used by this function. However, it can be used in Independent Channel Output Mode Controlled by Software.

(d) Simultaneous rewrite

The simultaneous rewrite registers (TAUAnRDE, TAUAnRDS, TAUAnRDM, and TAUAnRDC) cannot be used with the TAUAnTTINm Input Position Detection Function. Therefore, these registers must be set to 0.

Table 12-101 Simultaneous rewrite settings for TAUAnTTINm Input Position Detection Function

Bit name	Setting
TAUAnRDE.TAUAnRDEm	0: Disables simultaneous rewrite
TAUAnRDS.TAUAnRDSm	0: When simultaneous rewrite is disabled (TAUAnRDE.TAUAnRDEm = 0), set these bits to 0
TAUAnRDM.TAUAnRDMm	
TAUAnRDC.TAUAnRDCm	

(5) Operating procedure for TAUAnTTINm Input Position Detection Function

Table 12-102 Operating procedure for TAUAnTTINm Input Position Detection Function

	Operation	Status of TAUAn
Initial channel setting	Set the TAUAnCMORm register and TAUAnCMURm registers as described in *Table 12-99 "TAUAnCMORm settings for TAUAnTTINm Input Position Detection Function" on page 715* and *Table 12-100 "TAUAnCMURm settings for TAUAnTTINm Input Position Detection Function" on page 715* Set the value of the TAUAnCDRm register	Channel operation is stopped.
Start operation	Set TAUAnTS.TAUAnTSm to 1. TAUAnTS.TAUAnTSm is a trigger bit, so it is automatically cleared to 0.	TAUAnTE.TAUAnTEm is set to 1 and the counter starts. INTTAUAnIm is generated when TAUAnCMORm.TAUAnMD0 is set to 1.
During operation	The TAUAnCMURm.TAUAnTIS[1:0] bits can be changed at any time. The TAUAnCDRm and TAUAnCSRm registers can be read at any time. The TAUAnCSC.CLOV bit can be set to 1.	TAUAnCNTm starts to count up from 0000_H. When a TAUAnTTINm valid edge is detected: • TAUAnCNTm transfers (captures) its value to TAUAnCDRm. • INTTAUAnIm is output. • The counter value is not cleared to 0000_H and TAUAnCNTm continues count operation. Afterwards, this procedure is repeated.
Stop operation	Set TAUAnTT.TAUAnTTm to 1. TAUAnTT.TAUAnTTm is a trigger bit, so it is automatically cleared to 0.	TAUAnTE.TAUAnTEm is cleared to 0 and the counter stops. TAUAnCNTm stops and both it and TAUAnCSRm.OVF retain their current values.

Restart

(6) Specific timing diagrams

(a) Operation stop and restart

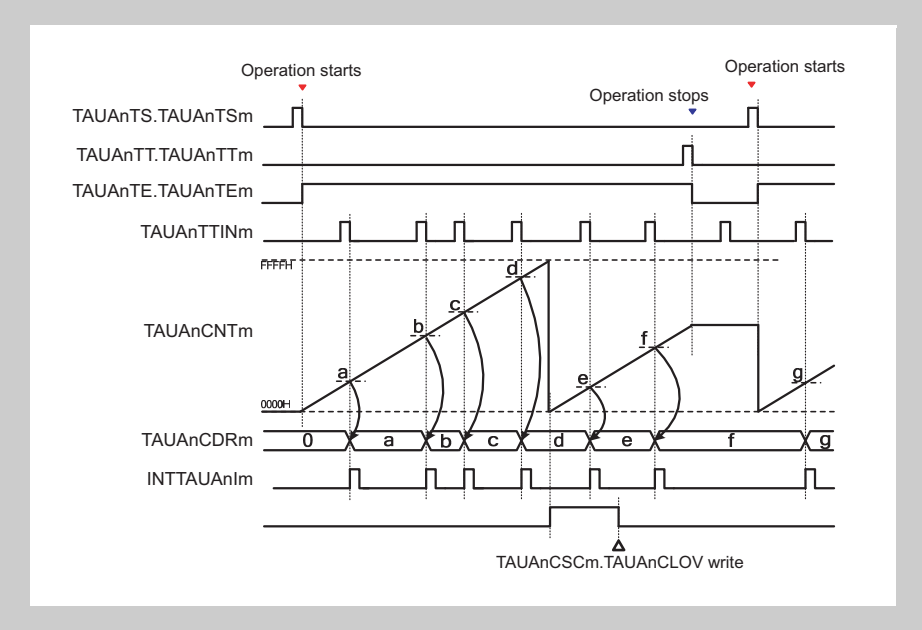


Figure 12-86 Operation stop and restart, TAUAnCMORm.TAUAnMD0 = 0, TAUAnCMURm.TAUAnTIS[1:0] = 00_B

- The counter can be stopped by setting TAUAnTT.TAUAnTTm to 1, which in turn sets TAUAnTE.TAUAnTEm to 0.
- TAUAnCNTm stops and the current value is retained.
- If the counter is stopped, valid TAUAnTTINm input edges are ignored.
- The counter can be restarted by setting TAUAnTS.TAUAnTSm to 1. TAUAnCNTm restarts to count from 0000_H.

12.21 Synchronous Channel Operation Functions

This section lists all the synchronous channel operation functions provided by TAUA. For a general overview of synchronous channel operation, see *12.3 “Functional Description” on page 557*.

12.22 Synchronous PWM Signal Functions Triggered at Regular Intervals

This chapter describes functions that generate PWM signals at regular intervals.

- *12.22.1 “PWM Output Function”*
- *12.22.2 “Trigger Start PWM Output Function”*
- *12.22.3 “Delay Pulse Output Function”*
- *12.22.4 “AD Conversion Trigger Output Function Type 1”*

12.22.1 PWM Output Function

(1) Overview

Summary This function generates multiple PWM outputs by using a master and multiple slave channels. It enables the pulse cycle (frequency) and the duty cycle of the TAUAnTTOUTm to be set. The pulse cycle is set in the master channel. The duty cycle is set in the slave channel.

Prerequisites

- Two channels
- The operation mode of the master channel must be set to Interval Timer Mode. Refer to *Table 12-103 “TAUAnCMORm settings for the master channel of the PWM Output Function” on page 723*.
- The operation mode of the slave channel(s) must be set to One Count Mode. Refer to *Table 12-106 “TAUAnCMORm settings for the slave channel of the PWM Output Function” on page 725*.
- TAUAnTTOUTm is not used for the master channel of this function.
- The channel output mode of the slave channel(s) must be set to Synchronous Channel Output Mode 1 (*12.8 “Channel Output Modes” on page 577*).

Description The counters are started by setting the channel trigger bits (TAUAnTS.TAUAnTSm) to 1. This in turn sets TAUAnTE.TAUAnTEm = 1, enabling count operation. The current value of TAUAnCDRm is loaded to TAUAnCNTm and the counters start to count down from these values. INTTAUAnIm is generated on the master channel, and PWM output is achieved by setting and resetting TAUAnTTOUTm (slave).

- Master channel:

  When the counter of the master channel reaches 0000_H, pulse cycle time has elapsed and INTTAUAnIm is generated. TAUAnCNTm loads the TAUAnCDRm value, and then counts down.

- Slave channel(s)

  The INTTAUAnIm of the master channel triggers the counter of the slave channel(s). The current value of TAUAnCDRm (slave) is loaded to TAUAnCNTm (slave) and the counter starts to count down from this value. The TAUAnTTOUTm signal becomes active.

  When the counter reaches 0000_H, i.e. duty time has elapsed, INTTAUAnIm is generated and the TAUAnTTOUTm signal becomes inactive. The counter returns to $FFFF_H$ and awaits the next INTTAUAnIm of the master channel, and thus the start of the next pulse cycle.

The counter can be stopped by setting TAUAnTT.TAUAnTTm to 1 for the master and slave channel(s), which in turn sets TAUAnTE.TAUAnTEm to 0. TAUAnCNTm and TAUAnTTOUTm of master and slave channel(s) stop but retain their values. The counters can be restarted by setting TAUAnTS.TAUAnTSm to 1.

Conditions Simultaneous rewrite can be used with this function. Please refer to *12.7 “Simultaneous Rewrite” on page 565*

(2) Equations

Pulse cycle = (TAUAnCDRm (master) + 1) x count clock cycle

Duty cycle [%] = (TAUAnCDRm (slave) / (TAUAnCDRm (master) + 1)) x 100

– Duty cycle = 0 %

TAUAnCDRm (slave) = 0000_H

– Duty cycle = 100 %

TAUAnCDRm (slave) ≥ TAUAnCDRm (master) + 1

(3) Block diagram and general timing diagram

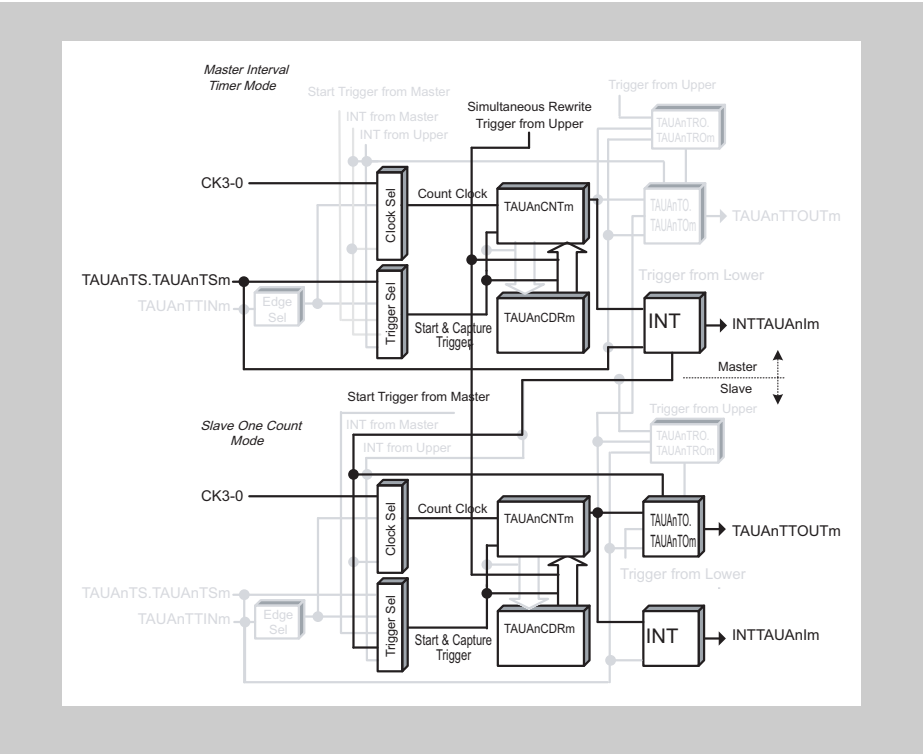


Figure 12-87 Block diagram for PWM Output Function

The following settings apply to the general timing diagram:

- Slave channel: Positive logic (TAUAnTOL.TAUAnTOLm = 0)

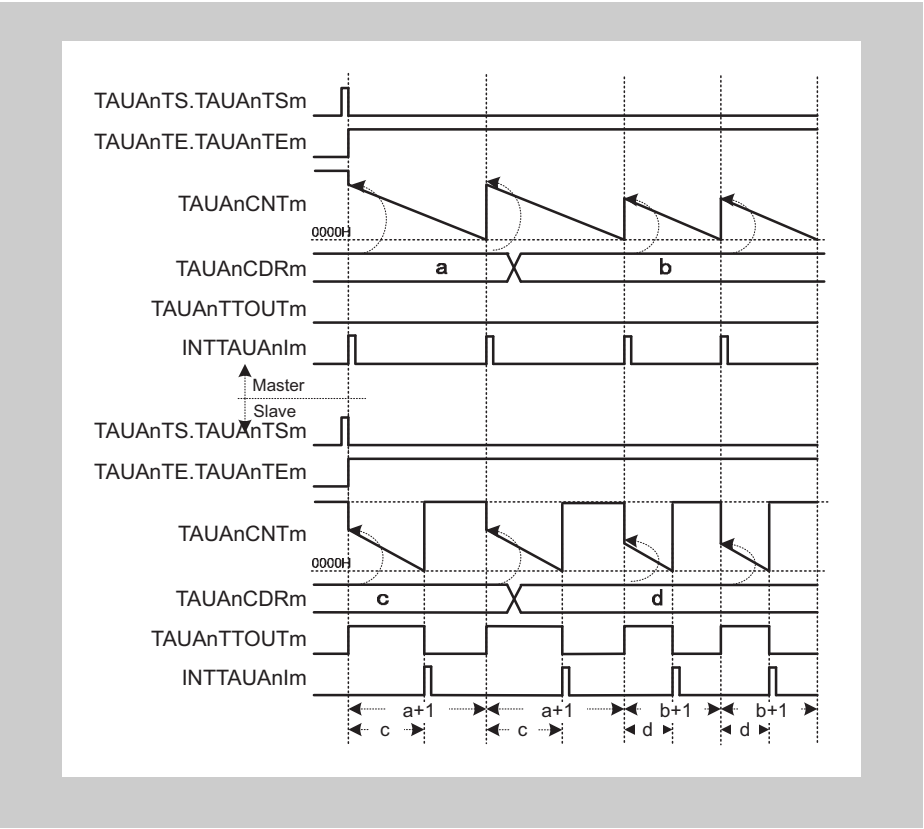


Figure 12-88 General timing diagram for PWM Output Function

Note The interval between the slave channel starting to count and an interrupt being generated is the value of corresponding TAUAnCDRm, whereas for the master channel the interval is the corresponding TAUAnCDRm + 1.

(4) Register settings for the master channel

(a) TAUAnCMORm for the master channel

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn CKS[1:0]		TAUAn CCS[1:0]		TAUAn MAS	TAUAnSTS[2:0]			TAUAn COS[1:0]		-	TAUAnMD[4:1]				TAUAn MD0

Table 12-103 TAUAnCMORm settings for the master channel of the PWM Output Function

Bit name	Setting
TAUAnCKS[1:0]	00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 The value of the TAUAnCKS[1:0] bit of the master and slave channel(s) must be identical.
TAUAnCCS[1:0]	00: Operation clock is used as the count clock
TAUAnMAS	1: Channel is master channel
TAUAnSTS[2:0]	000: Counter triggered by software trigger
TAUAnCOS[1:0]	00: Not used, so set to 00
TAUAnMD[4:1]	0000: Interval Timer Mode
TAUAnMD0	1: Generates INTTAUAnIm at operation start

(b) TAUAnCMURm for the master channel

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-														TAUAnTIS[1:0]	

Table 12-104 TAUAnCMURm settings for the master channel of the PWM Output Function

Bit name	Setting
TAUAnTIS[1:0]	00: These are not used, so set them to 00.

(c) Channel output mode for the master channel

The channel output mode is not used by this function. However, it can be used by other functions or in Independent Channel Output Mode Controlled by Software.

(d) Simultaneous rewrite for the master channel

The simultaneous rewrite settings of the master and slave channel must be identical.

Table 12-105 Simultaneous rewrite settings for the master channel of the PWM Output Function

Bit name	Setting
TAUAnRDE.TAUAnRDEm	1: Enables simultaneous rewrite
TAUAnRDS.TAUAnRDSm	0: The master channel is monitored for the simultaneous rewrite trigger 1: An upper channel outside the channel group is monitored for the simultaneous rewrite trigger
TAUAnRDM.TAUAnRDMm	0: The simultaneous rewrite trigger signal is generated when the master channel starts counting
TAUAnRDC.TAUAnRDCm	0: Channel is not monitored for an INTTAUAnIm signal that is used as the simultaneous rewrite trigger. If TAUAnRDS.TAUAnRDSm = 0, the master channel is monitored for the simultaneous rewrite trigger regardless of the value of this bit.

Note If the TAUAnRDS.TAUAnRDSm bit is 1, there must be a channel higher than the master channel that operates with simultaneous rewrite trigger output function type 1.

- Set up the operation as follows:

  Channel setting for simultaneous rewrite trigger output function type 1: TAUAnRDCm = 1, TAUAnRDS = 1

  Note that the TAUAnCDR setting for this channel is as follows:

  = ((TAUAnCDR setting of master channel subject to simultaneous rewriting +1) × number of interrupts) - 1

- Master channel: TAUAnRDCm = 0, TAUAnRDS = 1
- Slave channel: TAUAnRDCm = 0, TAUAnRDS = 1

(5) Register settings for the slave channel(s)

(a) TAUAnCMORm for the slave channel(s)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn CKS[1:0]		TAUAn CCS[1:0]		TAUAn MAS	TAUAnSTS[2:0]			TAUAn COS[1:0]		-	TAUAnMD[4:1]				TAUAn MD0

Table 12-106 TAUAnCMORm settings for the slave channel of the PWM Output Function

Bit name	Setting
TAUAnCKS[1:0]	00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 The value of the TAUAnCKS[1:0] bit of the master and slave channel(s) must be identical.
TAUAnCCS[1:0]	00: Operation clock is used as the count clock
TAUAnMAS	0: Channel is a slave channel
TAUAnSTS[2:0]	100: INTTAUAnIm of the master channel is the start trigger
TAUAnCOS[1:0]	00: Not used, so set to 00
TAUAnMD[4:1]	0100: One Count Mode
TAUAnMD0	1: Enables the start trigger during operation

(b) TAUAnCMURm for the slave channel(s)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-														TAUAnTIS[1:0]	

Table 12-107 TAUAnCMURm settings for the slave channel of the PWM Output Function

Bit name	Setting
TAUAnTIS[1:0]	00: These are not used, so set them to 00.

(c) Channel output mode for the slave channel(s)

Table 12-108 Control bit settings for Independent Channel Output Mode 1

Bit name	Setting
TAUAnTOE.TAUAnTOEm	1: Enables Independent Channel Output Mode
TAUAnTOM.TAUAnTOMm	1: Synchronous channel operation
TAUAnTOC.TAUAnTOCm	0: Operation mode 1
TAUAnTOL.TAUAnTOLm	0: Positive logic 1: Inverted logic
TAUAnTDE.TAUAnTDEm	0: Disables dead time operation
TAUAnTDM.TAUAnTDMm	0: When dead time operation is disabled (TAUAnTDE.TAUAnTDEm = 0), set these bits to 0
TAUAnTDL.TAUAnTDLm	
TAUAnTRE.TAUAnTREm	0: Disables real-time output
TAUAnTRO.TAUAnTROm	0: When real-time output is disabled (TAUAnTRE.TAUAnTREm = 0), set these bits to 0
TAUAnTRC.TAUAnTRCm	
TAUAnTME.TAUAnTMEm	0: Disables modulation

(d) Simultaneous rewrite for the slave channel(s)

The simultaneous rewrite settings of the master and slave channel must be identical.

Table 12-109 Simultaneous rewrite settings for the slave channel of the PWM Output Function

Bit name	Setting
TAUAnRDE.TAUAnRDEm	1: Enables simultaneous rewrite
TAUAnRDS.TAUAnRDSm	0: The master channel is monitored for the simultaneous rewrite trigger 1: An upper channel outside the channel group is monitored for the simultaneous rewrite trigger
TAUAnRDM.TAUAnRDMm	0: The simultaneous rewrite trigger signal is generated when the master channel starts counting
TAUAnRDC.TAUAnRDCm	0: Channel is not monitored for an INTTAUAnIm signal that is used as the simultaneous rewrite trigger. If TAUAnRDS.TAUAnRDSm = 0, the master channel is monitored for the simultaneous rewrite trigger regardless of the value of this bit.

(6) Operating procedure for PWM Output Function

Table 12-110 Operating procedure for PWM Output Function

	Operation	Status of TAUAn
Initial channel setting	Master channel: set the TAUAnCMORm and TAUAnCMURm registers and the channel output mode as described in *(4) "Register settings for the master channel" on page 723* Slave channel: set the TAUAnCMORm and TAUAnCMURm registers and the channel output mode as described in *(5) "Register settings for the slave channel(s)" on page 725* Set the values of the TAUAnCDRm registers of all channels	Channel operation is stopped.
Start operation	Set TAUAnTS.TAUAnTSm of the master and slave channels to 1 simultaneously. TAUAnTS.TAUAnTSm is a trigger bit, so it is automatically cleared to 0.	TAUAnTE.TAUAnTEm (master and slave channels) is set to 1 and the counters of the master and slave channels start. INTTAUAnIm is generated on the master channel and TAUAnTTOUTm (slave) becomes active.
During operation	TAUAnCDRm can be changed at any time. TAUAnCNTm and TAUAnRSF.TAUAnRSFm can be read at any time. TAUAnRDT.TAUAnRDTm can be changed during operation.	TAUAnCNTm of the master channel loads TAUAnCDRm, and then counts down. When the counter reaches 0000_H: • INTTAUAnIm (master) is generated. • TAUAnCNTm (master) loads the TAUAnCDRm value, and then continues count operation. • TAUAnCNTm (slave) loads the TAUAnCDRm value, and then counts down. • TAUAnTTOUTm (slave) becomes active. When TAUAnCNTm (slave) reaches 0000_H: • INTTAUAnIm (slave) is generated. • TAUAnTTOUTm (slave) becomes inactive.
Stop operation	Set TAUAnTT.TAUAnTTm of the master and slave channels to 1 simultaneously. TAUAnTT.TAUAnTTm is a trigger bit, so it is automatically cleared to 0.	TAUAnTE.TAUAnTEm is cleared to 0 and the counter stops. TAUAnCNTm and TAUAnTTOUTm stop and retain their current values.

Restart

(7) Specific timing diagrams

(a) Duty cycle = 0%

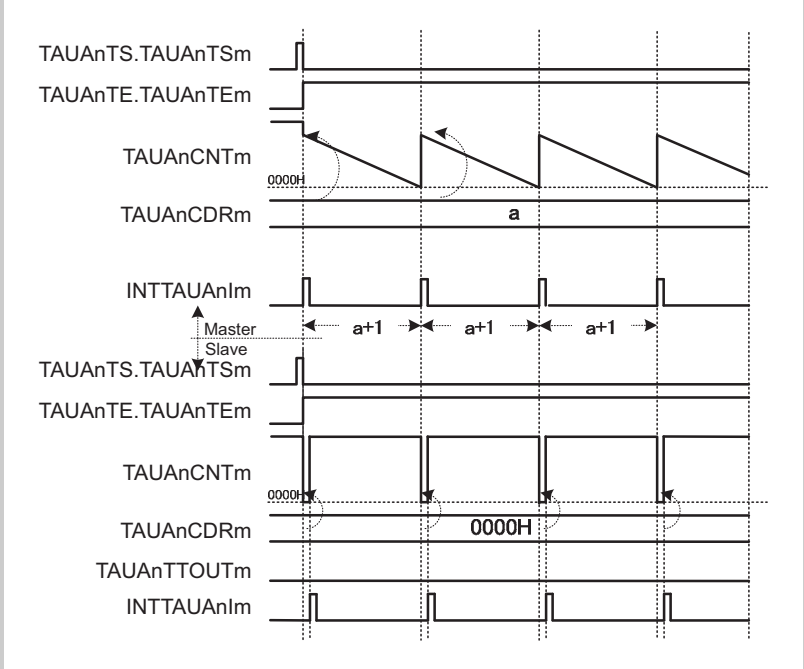


Figure 12-89 TAUAnCDRm (slave) = 0000_H, positive logic (TAUAnTOL.TAUAnTOLm (slave) = 0)

- Every time the master channel generates an interrupt (INTTAUAnIm), 0000_H is loaded to TAUAnCNTm (slave). Therefore, TAUAnCNTm (slave) cannot start to count and TAUAnTTOUTm remains at not active state.
- TAUAnCNTm (slave) loads the TAUAnCDRm value and an interrupt is generated.

(b) Duty cycle = 100%

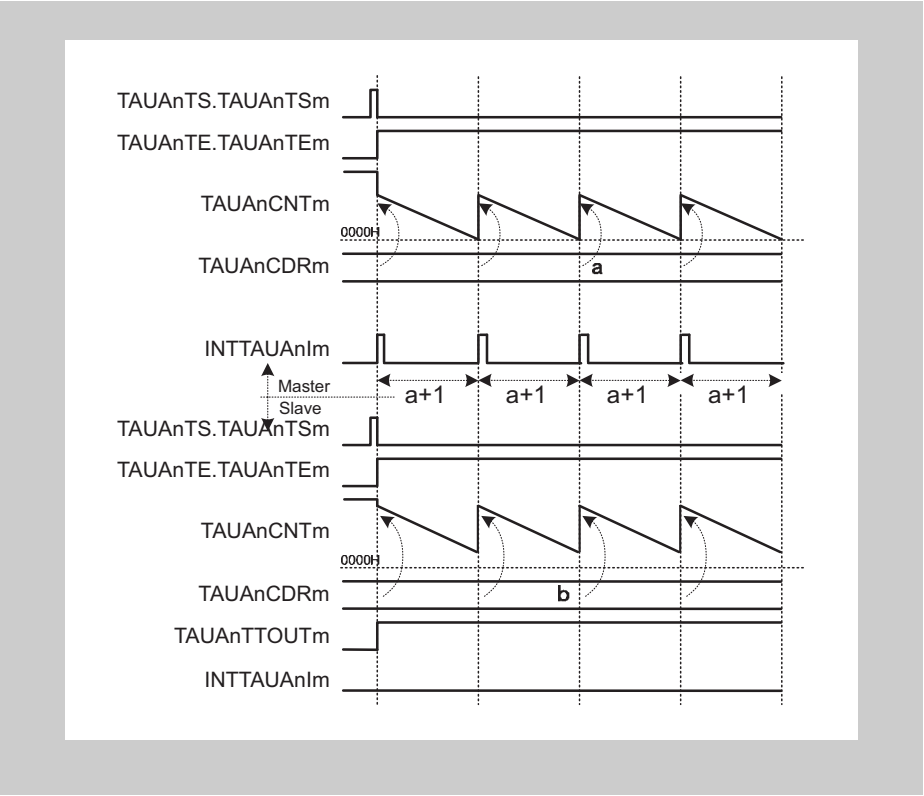


Figure 12-90 TAUAnCDRm (slave) ≥ TAUAnCDRm (master) + 1, positive logic (TAUAnTOL.TAUAnTOLm (slave) = 0)

- If the value TAUAnCDRm (slave) is higher than the value TAUAnCDRm (master), the counter of the slave channel cannot reach 0000_H and interrupts are not generated. The TAUAnTTOUTm remains at active state.

(c) Stop and restart operation

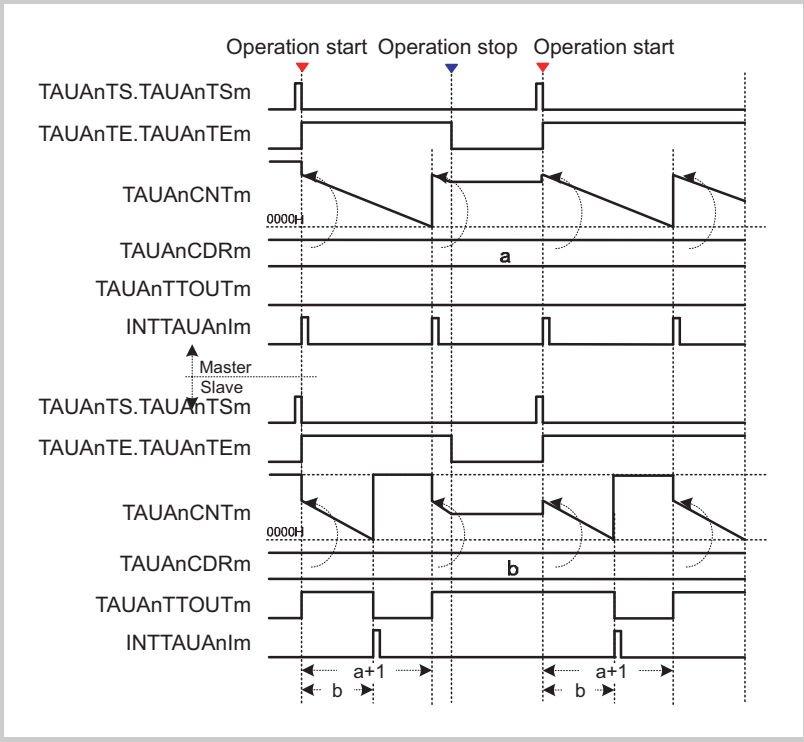


Figure 12-91 Stop and restart operation, positive logic (TAUAnTOL.TAUAnTOLm (slave) = 0)

- The counter can be stopped by setting TAUAnTT.TAUAnTTm of the master and slave channel(s) to 1, which in turn sets TAUAnTE.TAUAnTEm to 0.
- TAUAnCNTm and TAUAnTTOUTm of all channels stop and the current values are retained. No interrupts are generated.
- The counter can be restarted by setting TAUAnTS.TAUAnTSm of master and slave channel(s) to 1. TAUAnCNTm loads the TAUAnCDRm value of master and slave channels, and then starts counting down from this value.

12.22.2 Trigger Start PWM Output Function

(1) Overview

Summary This function generates a PWM output using a master and a slave channel. It enables the pulse cycle (frequency) and the duty cycle of the TAUAnTTOUTm to be set. The duty cycle is specified using the master channel. The pulse width is specified using the slave channel. The Trigger Start PWM Output Function is identical to PWM Output Function except that the master channel of this function can be reset by a valid TAUAnTTINm input edge.

Prerequisites

- Two channels
- The operation mode of the master channel must be set to Interval Timer Mode. Refer to *Table 12-111 "TAUAnCMORm settings for the master channel of the Trigger Start PWM Output Function" on page 734*.
- The operation mode of the slave channel must be set to One Count Mode. Refer to *Table 12-114 "TAUAnCMORm settings for the slave channel of the Trigger Start PWM Output Function" on page 736*.
- The channel output mode of the slave channel must be set to Synchronous Channel Output Mode 1. Refer to *12.8 "Channel Output Modes" on page 577*.
- TAUAnTTOUTm is not used for the master channel of this function.

Description The counters (master and slave) are started by setting the channel trigger bits (TAUAnTS.TAUAnTSm) to 1. This in turn sets TAUAnTE.TAUAnTEm, enabling count operation.

TAUAnCNTm loads the current value of TAUAnCDRm, and then starts counting down from this value. INTTAUAnIm is generated on the master channel, and PWM output is achieved by setting and resetting TAUAnTTOUTm (slave).

- Master channel:

  The current value of TAUAnCDRm is loaded to the counter (TAUAnCNTm), INTTAUAnIm is generated and the counter starts to count down from this value.

  When the counter reaches 0000_H, pulse cycle time has elapsed, INTTAUAnIm is generated and the TAUAnCNTm (master and slave) load the current TAUAnCDRm values.

  If a valid TAUAnTTINm input edge is detected, the counter of the master channel loads the current TAUAnCDRm value, restarts counting down, and then generates an interrupt.

- Slave channel:

  When the slave detects an interrupt from the master channel, it starts to count down from the current value of TAUAnCDRm. The TAUAnTTOUTm signal becomes active.

  When the counter reaches 0000_H, duty time has elapsed, INTTAUAnIm is generated and the TAUAnTTOUTm signal is reset. The counter returns to $FFFF_H$ and awaits the next INTTAUAnIm of the master channel.

The counter can be stopped by setting TAUAnTT.TAUAnTTm to 1 for the master and slave channel, which in turn sets TAUAnTE.TAUAnTEm to 0. TAUAnCNTm and TAUAnTTOUTm of master and slave channel stop but retain their values. The counters can be restarted by setting TAUAnTS.TAUAnTSm to 1.

Conditions Simultaneous rewrite can be used with this function. Please refer to *12.7 "Simultaneous Rewrite" on page 565*

(2) Equations

Pulse cycle = (TAUAnCDRm (master) + 1) x count clock cycle

Duty cycle [%] = [TAUAnCDRm (slave) / (TAUAnCDRm (master) + 1)] x 100

– Duty cycle = 0 %

TAUAnCDRm (slave) = 0000_H

– Duty cycle = 100 %

TAUAnCDRm (slave) ≥ TAUAnCDRm (master) + 1

(3) Block diagram and general timing diagram

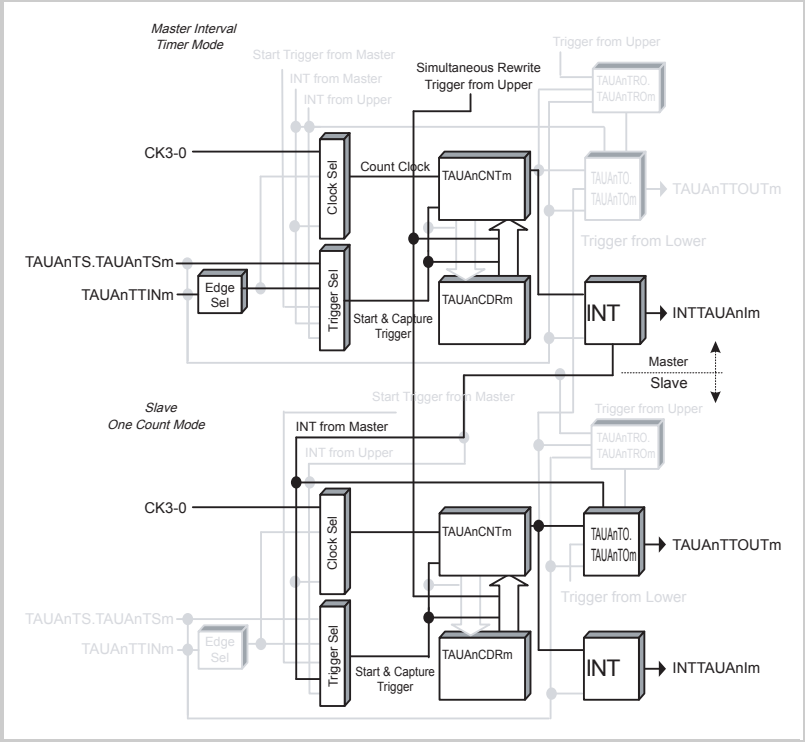


Figure 12-92 Block diagram for Trigger Start PWM Output Function

The following settings apply to the general timing diagram:

- Rising edge detection (TAUAnCMURm.TAUAnTIS[1:0] = 01_B)
- Positive logic (TAUAnTOL.TAUAnTOLm (slave) = 0)

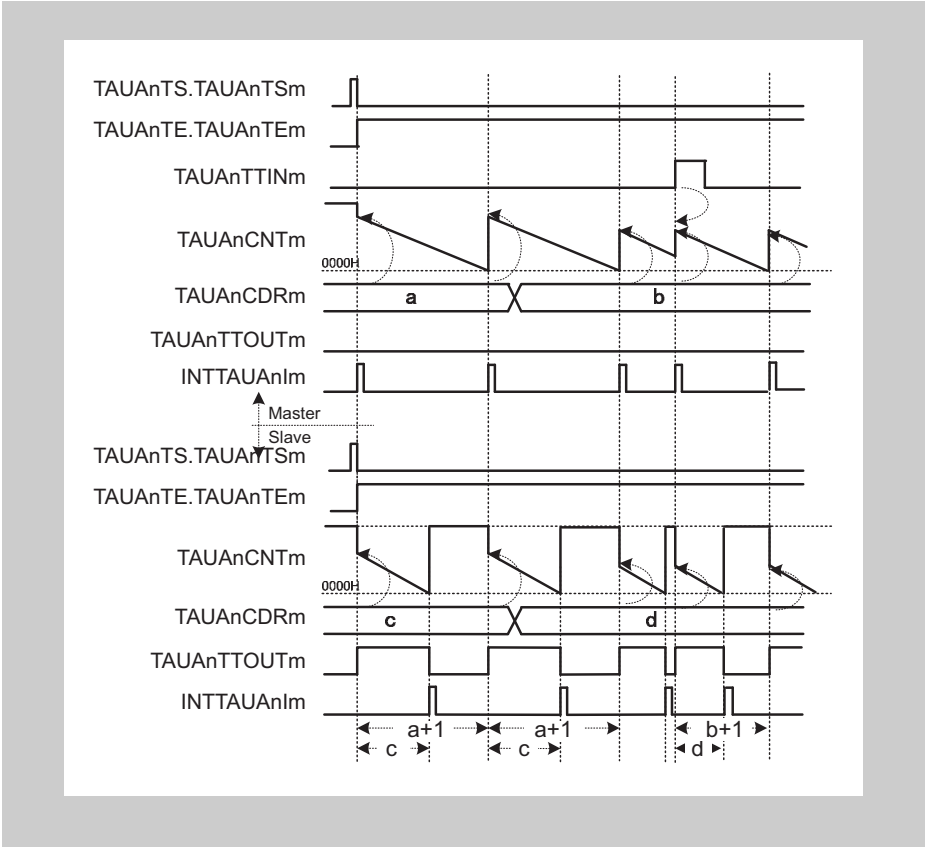


Figure 12-93 General timing diagram for Trigger Start PWM Output Function

(4) Register settings for the master channel

(a) TAUAnCMORm for the master channel

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn CKS[1:0]		TAUAn CCS[1:0]		TAUAn MAS	TAUAnSTS[2:0]			TAUAn COS[1:0]		-	TAUAnMD[4:1]				TAUAn MD0

Table 12-111 TAUAnCMORm settings for the master channel of the Trigger Start PWM Output Function

Bit name	Setting
TAUAnCKS[1:0]	00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 The value of the TAUAnCKS[1:0] bit of the master and slave channel must be identical.
TAUAnCCS[1:0]	00: Operation clock is used as the count clock
TAUAnMAS	1: Channel is master channel
TAUAnSTS[2:0]	001: Valid TAUAnTTINm input edge signal is used as the start trigger
TAUAnCOS[1:0]	00: Not used, so set to 00
TAUAnMD[4:1]	0000: Interval Timer Mode
TAUAnMD0	1: Generates INTTAUAnIm at operation start

(b) TAUAnCMURm for the master channel

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-														TAUAnTIS[1:0]	

Table 12-112 TAUAnCMURm settings for the master channel of the Trigger Start PWM Output Function

Bit name	Setting
TAUAnTIS[1:0]	00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection 11: Setting prohibited

(c) Channel output mode for the master channel

The channel output mode is not used by this function. However, it can be used by other functions or in Independent Channel Output Mode Controlled by Software.

(d) Simultaneous rewrite for the master channel

The simultaneous rewrite settings of the master and slave channel must be identical.

Table 12-113 Simultaneous rewrite settings for the master channel of the Trigger Start PWM Output Function

Bit name	Setting
TAUAnRDE.TAUAnRDEm	1: Enables simultaneous rewrite
TAUAnRDS.TAUAnRDSm	0: The master channel is monitored for the simultaneous rewrite trigger 1: An upper channel outside the channel group is monitored for the simultaneous rewrite trigger
TAUAnRDM.TAUAnRDMm	0: The simultaneous rewrite trigger signal is generated when the master channel starts counting
TAUAnRDC.TAUAnRDCm	0: Channel is not monitored for an INTTAUAnIm signal that is used as the simultaneous rewrite trigger. If TAUAnRDS.TAUAnRDSm = 0, the master channel is monitored for the simultaneous rewrite trigger regardless of the value of this bit.

Note If the TAUAnRDS.TAUAnRDSm bit is 1, there must be a channel higher than the master channel that operates with simultaneous rewrite trigger output function type 2.

- Channel setting for simultaneous rewrite trigger output function type 2: TAUAnRDCm = 1, TAUAnRDS = 1
- Master channel: TAUAnRDCm = 0, TAUAnRDS = 1
- Slave channel: TAUAnRDCm = 0, TAUAnRDS = 1

(5) Register settings for the slave channel

(a) TAUAnCMORm for the slave channel

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn CKS[1:0]		TAUAn CCS[1:0]		TAUAn MAS	TAUAnSTS[2:0]			TAUAn COS[1:0]		-	TAUAnMD[4:1]				TAUAn MD0

Table 12-114 TAUAnCMORm settings for the slave channel of the Trigger Start PWM Output Function

Bit name	Setting
TAUAnCKS[1:0]	00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 The value of the TAUAnCKS[1:0] bit of the master and slave channel must be identical.
TAUAnCCS[1:0]	00: Operation clock is used as the count clock
TAUAnMAS	0: Channel is a slave channel
TAUAnSTS[2:0]	100: INTTAUAnIm of the master channel is the start trigger
TAUAnCOS[1:0]	00: Not used, so set to 00
TAUAnMD[4:1]	0100: One Count Mode
TAUAnMD0	1: Enables the start trigger during operation

(b) TAUAnCMURm for the slave channel

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-														TAUAnTIS[1:0]	

Table 12-115 TAUAnCMURm settings for the slave channel of the Trigger Start PWM Output Function

Bit name	Setting
TAUAnTIS[1:0]	00: These are not used, so set them to 00.

(c) Channel output mode for the slave channel

Table 12-116 Control bit settings for Independent Channel Output Mode 1

Bit name	Setting
TAUAnTOE.TAUAnTOEm	1: Enables Independent Channel Output Mode
TAUAnTOM.TAUAnTOMm	1: Synchronous channel operation
TAUAnTOC.TAUAnTOCm	0: Operation mode 1
TAUAnTOL.TAUAnTOLm	0: Positive logic 1: Inverted logic
TAUAnTDE.TAUAnTDEm	0: Disables dead time operation
TAUAnTDM.TAUAnTDMm	0: When dead time operation is disabled (TAUAnTDE.TAUAnTDEm = 0), set these bits to 0
TAUAnTDL.TAUAnTDLm	
TAUAnTRE.TAUAnTREm	0: Disables real-time output
TAUAnTRO.TAUAnTROm	0: When real-time output is disabled (TAUAnTRE.TAUAnTREm = 0), set these bits to 0
TAUAnTRC.TAUAnTRCm	
TAUAnTME.TAUAnTMEm	0: Disables modulation

(d) Simultaneous rewrite for the slave channel

The simultaneous rewrite settings of the master and slave channel must be identical.

Table 12-117 Simultaneous rewrite settings for the slave channel of the Trigger Start PWM Output Function

Bit name	Setting
TAUAnRDE.TAUAnRDEm	1: Enables simultaneous rewrite
TAUAnRDS.TAUAnRDSm	0: The master channel is monitored for the simultaneous rewrite trigger 1: An upper channel outside the channel group is monitored for the simultaneous rewrite trigger
TAUAnRDM.TAUAnRDMm	0: The simultaneous rewrite trigger signal is generated when the master channel starts counting
TAUAnRDC.TAUAnRDCm	0: Channel is not monitored for an INTTAUAnIm signal that is used as the simultaneous rewrite trigger. If TAUAnRDS.TAUAnRDSm = 0, the master channel is monitored for the simultaneous rewrite trigger regardless of the value of this bit.

(6) Operating procedure for Trigger Start PWM Output Function

Table 12-118 Operating procedure for Trigger Start PWM Output Function

	Operation	Status of TAUAn
Initial channel setting	Master channel: set the TAUAnCMORm and TAUAnCMURm registers and the channel output mode as described in *(4) "Register settings for the master channel" on page 734* Slave channel: set the TAUAnCMORm and TAUAnCMURm registers and the channel output mode as described in *(5) "Register settings for the slave channel" on page 736* Set the values of the TAUAnCDRm registers of all channels	Channel operation is stopped.
Start operation	Set TAUAnTS.TAUAnTSm of the master and slave channels to 1 simultaneously. TAUAnTS.TAUAnTSm is a trigger bit, so it is automatically cleared to 0.	TAUAnTE.TAUAnTEm (master and slave channels) is set to 1 and the counters of the master and slave channels start. INTTAUAnIm is generated on the master channel.
During operation	TAUAnCDRm can be changed at any time. TAUAnCNTm and TAUAnRSF.TAUAnRSFm can be read at any time. TAUAnRDT.TAUAnRDTm can be changed during operation.	TAUAnCNTm of the master channel loads TAUAnCDRm, and then counts down. When the counter reaches 0000_H: • INTTAUAnIm (master) is generated. • TAUAnCNTm (master) loads the TAUAnCDRm value, and then continues count operation. • TAUAnCNTm (slave) loads the TAUAnCDRm value, and then starts counting down. • TAUAnTTOUTm (slave) becomes active. When TAUAnCNTm of the slave = 0000_H: • INTTAUAnIm (slave) is generated. • TAUAnTTOUTm (slave) becomes inactive. If a TAUAnTTINm input is detected on the master channel while the TAUAnCNTm of the master channel is counting down: • TAUAnCNTm (master and slave) loads the TAUAnCDRm value, and then counts down. • INTTAUAnIm (master) is generated. • TAUAnTTOUTm (slave) becomes active.
Stop operation	Set TAUAnTT.TAUAnTTm of the master and slave channels to 1 simultaneously. TAUAnTT.TAUAnTTm is a trigger bit, so it is automatically cleared to 0.	TAUAnTE.TAUAnTEm is cleared to 0 and the counter stops. TAUAnCNTm and TAUAnTTOUTm stop and retain their current values.

Restart

(7) Specific timing diagrams

(a) Duty cycle = 0%

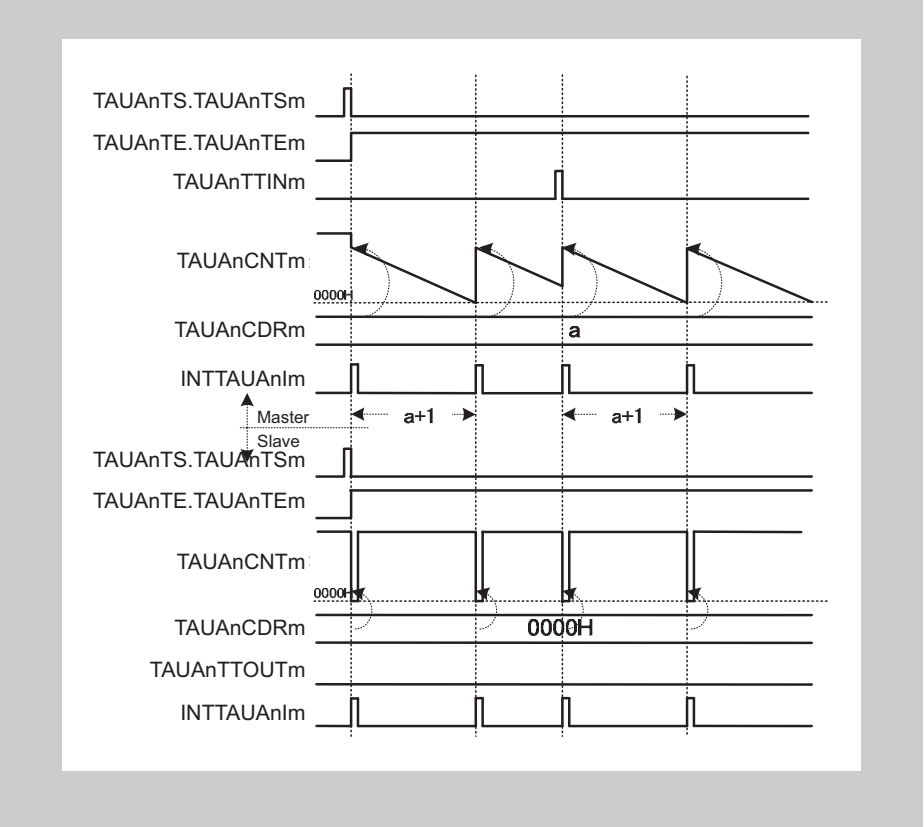


Figure 12-94 TAUAnCDRm (slave) = 0000_H, positive logic (TAUAnTOL.TAUAnTOLm (slave) = 0) falling edge detection (TAUAnCMURm.TAUAnTIS[1:0] = 00_B)

- Every time the master channel generates an interrupt (INTTAUAnIm), 0000_H is loaded to TAUAnCNTm (slave). Therefore, TAUAnCNTm (slave) cannot start to count and TAUAnTTOUTm remains at not active state.
- TAUAnCNTm (slave) generates an interrupt every time the value of TAUAnCDRm is loaded.

  The detection of a valid TAUAnTTINm input edge has no effect on TAUAnTTOUTm (slave).

(b) Duty cycle = 100%

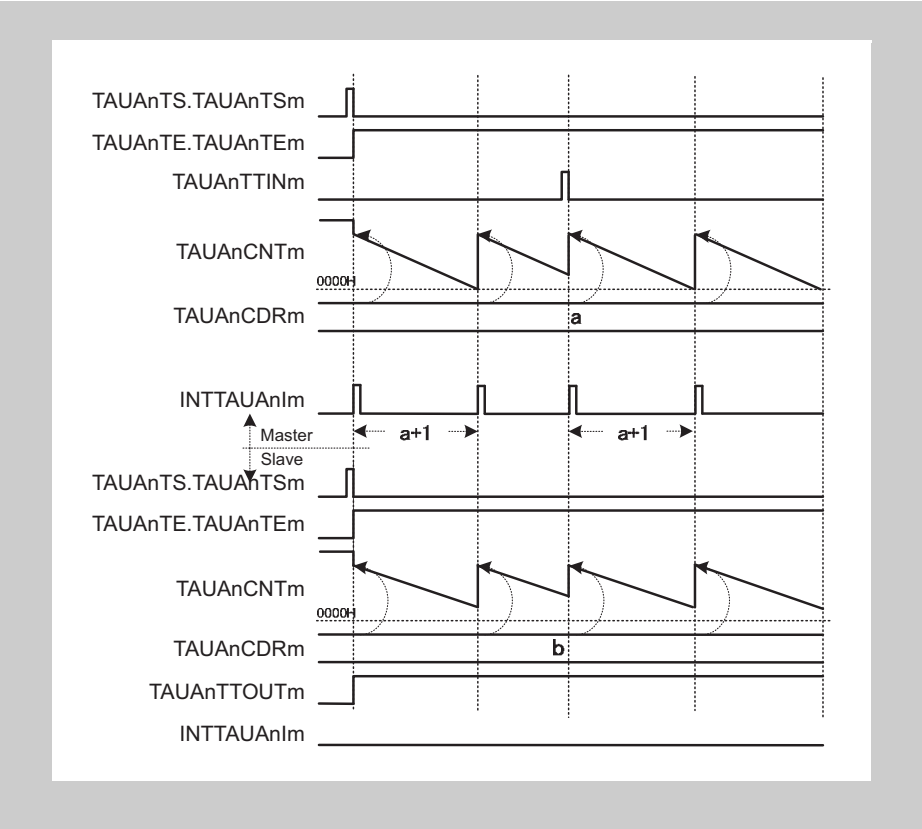


Figure 12-95 TAUAnCDRm (slave) ≥ TAUAnCDRm (master) + 1, positive logic (TAUAnTOL.TAUAnTOLm (slave) = 0) falling edge detection (TAUAnCMURm.TAUAnTIS[1:0] = 00_B)

- If the value TAUAnCDRm (slave) is higher than the value TAUAnCDRm (master), the counter of the slave channel cannot reach 0000_H and no interrupt is generated.

  The TAUAnTTOUTm remains at active state.

  The detection of a valid TAUAnTTINm input edge has no effect on TAUAnTTOUTm (slave).

(c) TAUAnTTINm detection and active slave counter

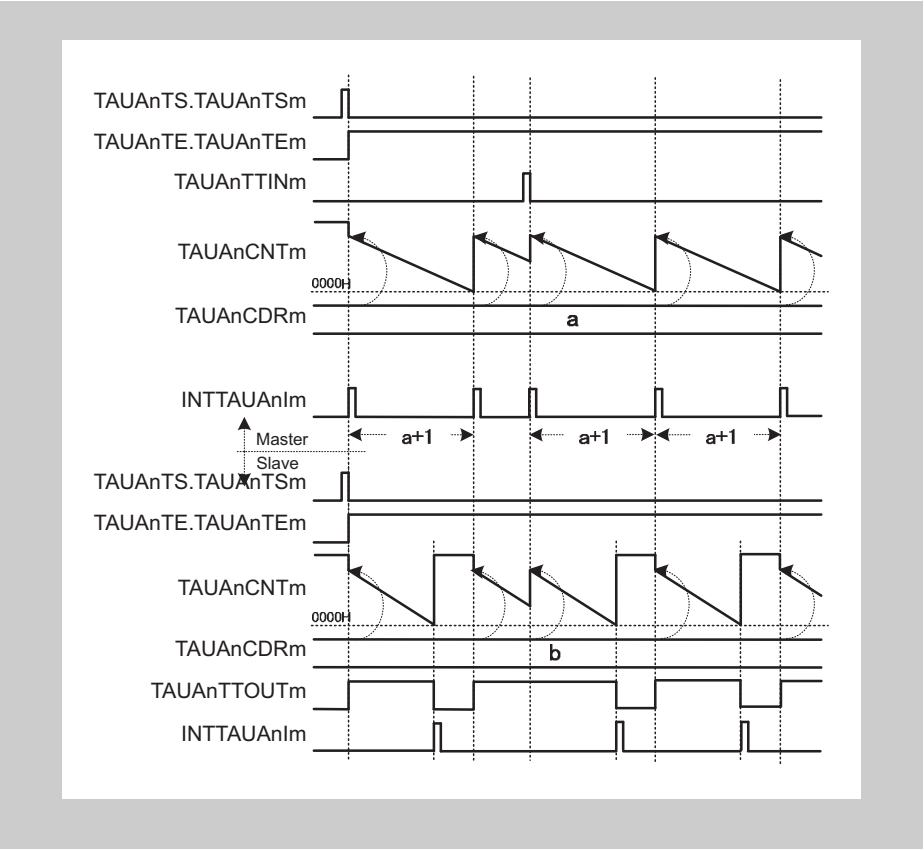


Figure 12-96 Positive logic (TAUAnTOL.TAUAnTOLm (slave) = 0), falling edge detection (TAUAnCMURm.TAUAnTIS[1:0] = 00_B)

- If TAUAnCNTm (slave) loads the value of TAUAnCDRm (slave) while it is counting down, TAUAnTTOUTm does not change and extends the duty cycle.

  The duty cycle does not correspond to the value of the data register of the slave channel.

12.22.3 Delay Pulse Output Function

(1) Overview

Summary

This function outputs two signals. The reference signal has a defined pulse width and pulse cycle specified using the master channel and slave channel 1. Slave channels 2 and 3 output the reference signal with a specified delay. The delay signal is identical to the reference signal, but delayed by amount specified in slave channel 2.

The signal values are specified in the following way:

- The pulse cycle is specified using the master channel.
- The duty cycle of the reference signal is specified using slave channel 1. The duty cycle of the delay signal is specified using slave channel 3.
- The delay is specified in slave channel 2.

Prerequisites

- Four channels
- The operation mode of the master channel must be set to Interval Timer Mode. Refer to *Table 12-119 “TAUAnCMORm settings for the master channel of the Delay Pulse Output Function” on page 746*.
- The operation mode of slave channel 1 and 2 must be set to One Count Mode. Refer to *Table 12-122 “TAUAnCMORm settings for slave channel 1 of the Delay Pulse Output Function” on page 748*.
- The operation mode of slave channel 3 must be set to Pulse One Count Mode. Refer to *Table 12-126 “TAUAnCMORm settings for slave channel 2 of the Delay Pulse Output Function” on page 750*.
- TAUAnTTOUTm is not used for the master channel and slave channel 2.
- The channel output mode of slave channel 1 must be set to Synchronous Channel Output Mode 1 (refer to *12.8 “Channel Output Modes” on page 577*).
- The channel output mode of slave channel 3 must be set to Independent Channel Output Mode 2 (refer to *12.8 “Channel Output Modes” on page 577*).

Description

The counters of the channel group are started by setting the channel trigger bit (TAUAnTS.TAUAnTSm) to 1. This in turn sets TAUAnTE.TAUAnTEm, enabling count operation.

- Master channel:

  The current value of TAUAnCDRm is loaded to TAUAnCNTm and the counter starts to count down from this value. INTTAUAnIm is generated on the master channel.

  When the counter of the master channel reaches 0000_H, pulse cycle time has elapsed and INTTAUAnIm is generated. The counter reloads the TAUAnCDRm value, and then counts down.

- Slave channels 1 and 2:

  When the slave channels 1 and 2 detect an interrupt from the master channel, they start to count down from the current value of TAUAnCDRm. The TAUAnTTOUTm signal (slave 1) becomes active.

– Slave channel 1:

When the counter of slave channel 1 reaches 0000_H, duty time has elapsed, INTTAUAnIm is generated and the TAUAnTTOUTm signal is reset. The counter returns to $FFFF_H$ and awaits the next INTTAUAnIm of the master channel.

– Slave channel 2:

When the counter of slave channel 2 reaches 0000_H, delay time has elapsed and INTTAUAnIm is generated. The counter returns to $FFFF_H$ and awaits the next INTTAUAnIm of the master channel.

INTTAUAnIm (slave 2) triggers the counter of slave channel 3

- Slave channel 3:

When slave channel 3 detects an interrupt from slave channel 2, it starts to count down from the current value of TAUAnCDRm. INTTAUAnIm is generated and the TAUAnTTOUTm signal (slave 3) is set.

When the counter of slave channel 3 reaches 0000_H, duty time has elapsed, INTTAUAnIm is generated and the TAUAnTTOUTm signal is reset.

The output from slave channel 3 is the delayed PWM pulse

The counter can be stopped by setting TAUAnTT.TAUAnTTm to 1 for the master and slave channels, which in turn sets TAUAnTE.TAUAnTEm to 0. TAUAnCNTm and TAUAnTTOUTm of master and slave channels stop but retain their values. The counters can be restarted by setting TAUAnTS.TAUAnTSm to 1.

Conditions Simultaneous rewrite can be used with this function. Please refer to *12.7 "Simultaneous Rewrite" on page 565*

Equations

Pulse cycle = (TAUAnCDRm (master) + 1) × count clock cycle

Duty width 1 = (TAUAnCDRm (slave 1)) × count clock cycle

Delay width = (TAUAnCDRm (slave 2) + 1) × count clock cycle

Duty width 2 = (TAUAnCDRm (slave 3)) × count clock cycle

However, the duty width setting is assumed to be in the following range:

0000H ≤ TAUAnCDRm (slave 2) < TAUAnCDRm (master)

Notes

1. The output waveform of TAUAnTOUTm (slave 3) is the output waveform of TAUAnTOUTm (slave 1) delayed by the amount of delay generated by slave 2. It is not possible to delay the waveform by a pulse cycle or more.
2. If the TAUAnINTm signal for slave 2 is generated during slave 3 counting, slave 3 resumes operation. Therefore, the output waveform of TAUAnTOUTm (slave 3) stays at the active level. (In this case, it is not possible for TOUTn (slave channel 3) to output a waveform that delays the TOUTn (slave channel 1) reference pulse.)

(2) Block diagram and general timing diagram

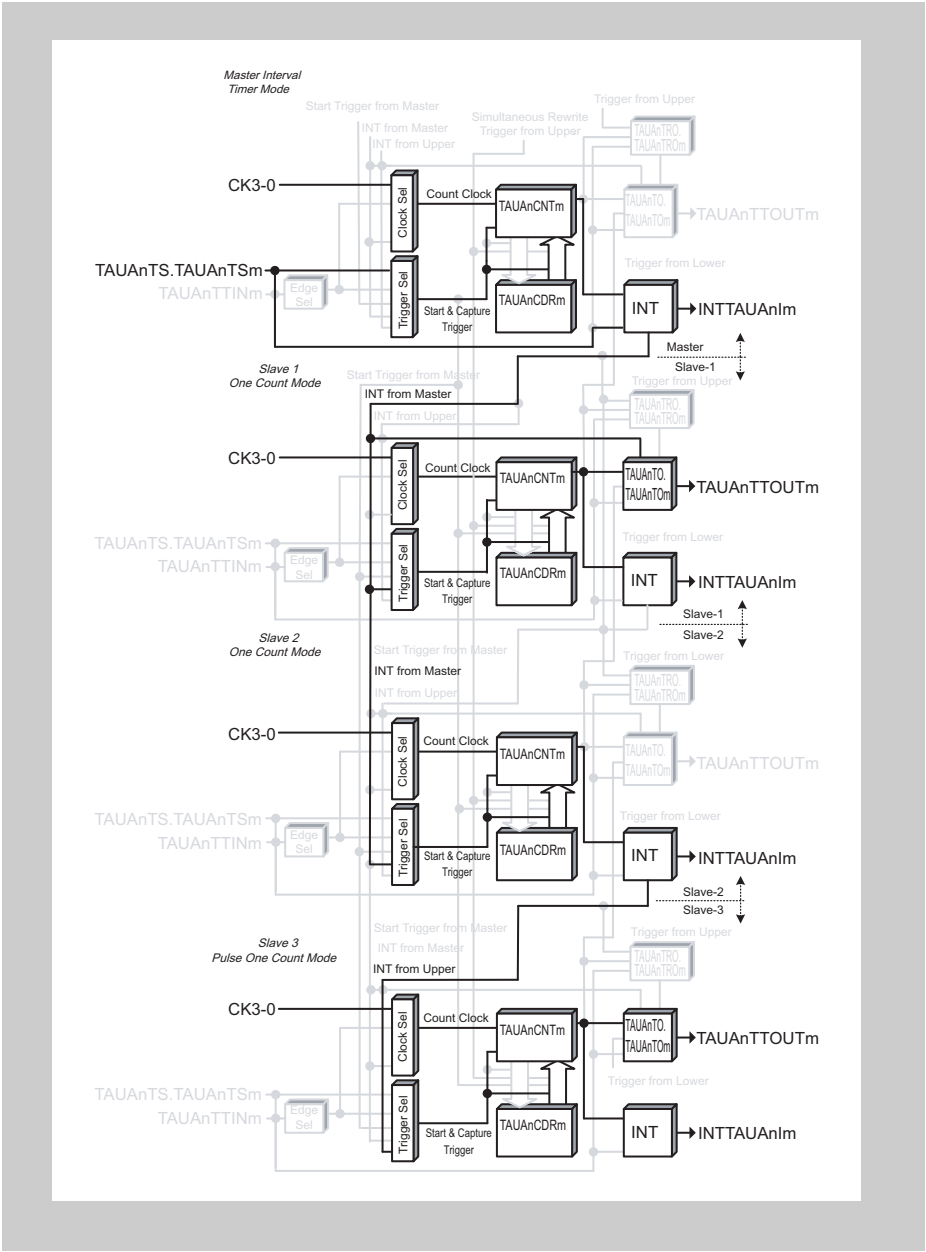


Figure 12-97 Block diagram for Delay Pulse Output Function

The following settings apply to the general timing diagram:

- All channels
  - INTTAUAnIm is generated at operation start (TAUAnCMORm.TAUAnMD0 = 1)

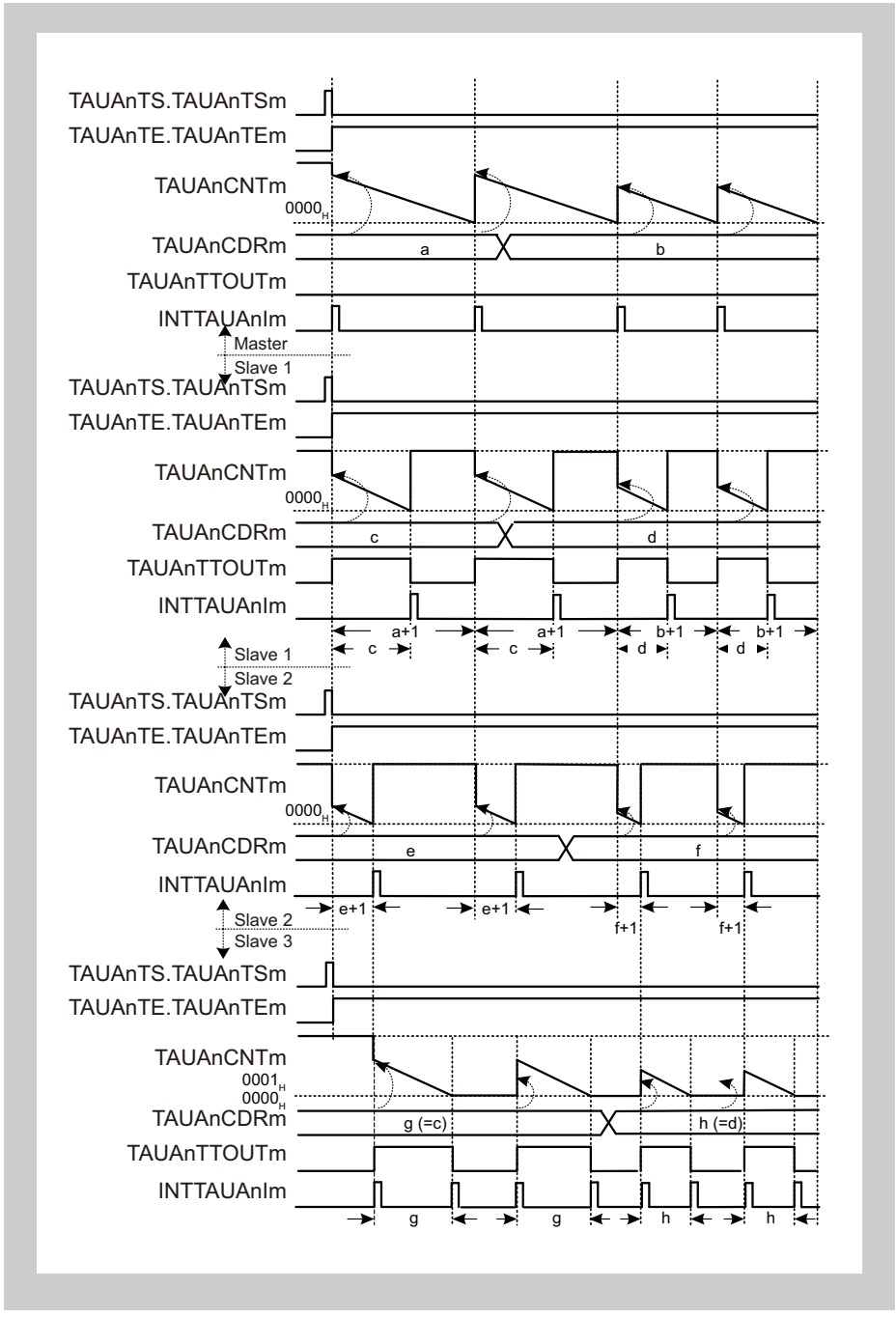


Figure 12-98 General timing diagram for Delay Pulse Output Function

(3) Register settings for the master channel

(a) TAUAnCMORm for the master channel

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn CKS[1:0]		TAUAn CCS[1:0]		TAUAn MAS	TAUAnSTS[2:0]			TAUAn COS[1:0]		-	TAUAn MD[4:1]				TAUAn MD0

Table 12-119 TAUAnCMORm settings for the master channel of the Delay Pulse Output Function

Bit name	Setting
TAUAnCKS[1:0]	00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 The value of the TAUAnCKS[1:0] bit of the master and slave channel must be identical.
TAUAnCCS[1:0]	00: Operation clock is used as the count clock
TAUAnMAS	1: Channel is master channel
TAUAnSTS[2:0]	000: Counter triggered by software trigger
TAUAnCOS[1:0]	00: Not used, so set to 00
TAUAnMD[4:1]	0000: Interval Timer Mode
TAUAnMD0	1: Generates INTTAUAnIm at operation start

(b) TAUAnCMURm for the master channel

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-														TAUAnTIS[1:0]	

Table 12-120 TAUAnCMURm settings for the master channel of the Delay Pulse Output Function

Bit name	Setting
TAUAnTIS[1:0]	00: These are not used, so set them to 00.

(c) Channel output mode for the master channel

Because the channel output mode is not used by the master channel of this function, clear TAUAnTOE.TAUAnTOEm. However, it can be used by other functions or in Independent Channel Output Mode Controlled by Software.

(d) Simultaneous rewrite for the master channel

The simultaneous rewrite settings of the master and slave channels must be identical.

Table 12-121 Simultaneous rewrite settings for the master channel of the Delay Pulse Output Function

Bit name	Setting
TAUAnRDE.TAUAnRDEm	1: Enables simultaneous rewrite
TAUAnRDS.TAUAnRDSm	0: The master channel is the control channel for simultaneous rewrite
TAUAnRDM.TAUAnRDMm	0: The simultaneous rewrite trigger signal is generated when the master channel starts counting
TAUAnRDC.TAUAnRDCm	0: Channel is not monitored for an INTTAUAnIm signal that is used as the simultaneous rewrite trigger. If TAUAnRDS.TAUAnRDSm = 0, the master channel is monitored for the simultaneous rewrite trigger regardless of the value of this bit.

(4) Register settings for slave channel 1

(a) TAUAnCMORm for slave channel 1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn CKS[1:0]		TAUAn CCS[1:0]		TAUAn MAS	TAUAnSTS[2:0]			TAUAn COS[1:0]		-	TAUAnMD[4:1]				TAUAn MD0

Table 12-122 TAUAnCMORm settings for slave channel 1 of the Delay Pulse Output Function

Bit name	Setting
TAUAnCKS[1:0]	00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 The value of the TAUAnCKS[1:0] bit of the master and slave channel must be identical.
TAUAnCCS[1:0]	00: Operation clock is used as the count clock
TAUAnMAS	0: Channel is a slave channel
TAUAnSTS[2:0]	100: INTTAUAnIm of the master channel is the start trigger
TAUAnCOS[1:0]	00: Not used, so set to 00
TAUAnMD[4:1]	0100: One Count Mode
TAUAnMD0	1: Enables the start trigger during operation

(b) TAUAnCMURm for slave channel 1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-														TAUAnTIS[1:0]	

Table 12-123 TAUAnCMURm settings for slave channel 1 of the Delay Pulse Output Function

Bit name	Setting
TAUAnTIS[1:0]	00: These are not used, so set them to 00.

(c) Channel output mode for slave channel 1

Table 12-124 Control bit settings for slave channel 1 of the Synchronous Channel Output Mode 1

Bit name	Setting
TAUAnTOE.TAUAnTOEm	1: Enables Independent Channel Output Mode
TAUAnTOM.TAUAnTOMm	1: Synchronous channel operation
TAUAnTOC.TAUAnTOCm	1: Set/reset mode
TAUAnTOL.TAUAnTOLm	0: Positive logic 1: Inverted logic
TAUAnTDE.TAUAnTDEm	0: Disables dead time operation
TAUAnTDM.TAUAnTDMm	0: When dead time operation is disabled (TAUAnTDE.TAUAnTDEm = 0), set these bits to 0
TAUAnTDL.TAUAnTDLm	
TAUAnTRE.TAUAnTREm	0: Disables real-time output
TAUAnTRO.TAUAnTROm	0: When real-time output is disabled (TAUAnTRE.TAUAnTREm = 0), set these bits to 0
TAUAnTRC.TAUAnTRCm	
TAUAnTME.TAUAnTMEm	0: Disables modulation

(d) Simultaneous rewrite for slave channel 1

The simultaneous rewrite settings of the master and slave channels must be identical.

Table 12-125 Simultaneous rewrite settings for slave channel 1 of the Delay Pulse Output Function

Bit name	Setting
TAUAnRDE.TAUAnRDEm	1: Enables simultaneous rewrite
TAUAnRDS.TAUAnRDSm	0: The master channel is the control channel for simultaneous rewrite
TAUAnRDM.TAUAnRDMm	0: The simultaneous rewrite trigger signal is generated when the master channel starts counting
TAUAnRDC.TAUAnRDCm	0: Channel is not monitored for an INTTAUAnIm signal that is used as the simultaneous rewrite trigger. If TAUAnRDS.TAUAnRDSm = 0, the master channel is monitored for the simultaneous rewrite trigger regardless of the value of this bit.

(5) Register settings for slave channel 2

(a) TAUAnCMORm for slave channel 2

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn CKS[1:0]		TAUAn CCS[1:0]		TAUAn MAS	TAUAn STS[2:0]			TAUAn COS[1:0]		-	TAUAnMD[4:1]				TAUAn MD0

Table 12-126 TAUAnCMORm settings for slave channel 2 of the Delay Pulse Output Function

Bit name	Setting
TAUAnCKS[1:0]	00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 The value of the TAUAnCKS[1:0] bit of the master and slave channel must be identical.
TAUAnCCS[1:0]	00: Operation clock is used as the count clock
TAUAnMAS	0: Channel is a slave channel
TAUAnSTS[2:0]	100: INTTAUAnIm of the master channel is the start trigger
TAUAnCOS[1:0]	00: Not used, so set to 00
TAUAnMD[4:1]	0100: One Count Mode
TAUAnMD0	1: Enables the start trigger during operation

(b) TAUAnCMURm for slave channel 2

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-														TAUAnTIS[1:0]	

Table 12-127 TAUAnCMURm settings for slave channel 2 of the Delay Pulse Output Function

Bit name	Setting
TAUAnTIS[1:0]	00: These are not used, so set them to 00.

(c) Channel output mode for slave channel 2

Because the channel output mode is not used by this function, clear TAUAnTOE.TAUAnTOEm. However, it can be used by other functions or in Independent Channel Output Mode Controlled by Software.

(d) Simultaneous rewrite for slave channel 2

The simultaneous rewrite settings of the master and slave channels must be identical.

Table 12-128 Simultaneous rewrite settings for slave channel 2 of the Delay Pulse Output Function

Bit name	Setting
TAUAnRDE.TAUAnRDEm	1: Enables simultaneous rewrite
TAUAnRDS.TAUAnRDSm	0: The master channel is the control channel for simultaneous rewrite
TAUAnRDM.TAUAnRDMm	0: The simultaneous rewrite trigger signal is generated when the master channel starts counting
TAUAnRDC.TAUAnRDCm	0: Channel is not monitored for an INTTAUAnIm signal that is used as the simultaneous rewrite trigger. If TAUAnRDS.TAUAnRDSm = 0, the master channel is monitored for the simultaneous rewrite trigger regardless of the value of this bit.

(6) Register settings for slave channel 3

(a) TAUAnCMORm for slave channel 3

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn CKS[1:0]		TAUAn CCS[1:0]		TAUAn MAS	TAUAnSTS[2:0]			TAUAn COS[1:0]		-	TAUAnMD[4:1]				TAUAn MD0

Table 12-129 TAUAnCMORm settings for slave channel 3 of the Delay Pulse Output Function

Bit name	Setting
TAUAnCKS[1:0]	00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 The value of the TAUAnCKS[1:0] bit of the master and slave channel must be identical.
TAUAnCCS[1:0]	00: Operation clock is used as the count clock
TAUAnMAS	0: Channel is a slave channel
TAUAnSTS[2:0]	101: INTTAUAnIm of the upper channel (m-1) is the start trigger, regardless of the master setting
TAUAnCOS[1:0]	00: Not used, so set to 00
TAUAnMD[4:1]	1010: Pulse One Count Mode
TAUAnMD0	1: Enables the start trigger during operation

(b) TAUAnCMURm for slave channel 3

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-														TAUAnTIS[1:0]	

Table 12-130 TAUAnCMURm settings for slave channel 3 of the Delay Pulse Output Function

Bit name	Setting
TAUAnTIS[1:0]	00: These are not used, so set them to 00.

(c) Channel output mode for slave channel 3

Table 12-131 Control bit settings for Independent Channel Output Mode 2

Bit name	Setting
TAUAnTOE.TAUAnTOEm	1: Enables Independent Channel Output Mode
TAUAnTOM.TAUAnTOMm	0: Independent channel output
TAUAnTOC.TAUAnTOCm	1: Operation mode 2
TAUAnTOL.TAUAnTOLm	0: Positive logic 1: Inverted logic
TAUAnTDE.TAUAnTDEm	0: Disables dead time operation
TAUAnTDM.TAUAnTDMm	0: When dead time operation is disabled (TAUAnTDE.TAUAnTDEm = 0), set these bits to 0
TAUAnTDL.TAUAnTDLm	
TAUAnTRE.TAUAnTREm	0: Disables real-time output
TAUAnTRO.TAUAnTROm	0: When real-time output is disabled (TAUAnTRE.TAUAnTREm = 0), set these bits to 0
TAUAnTRC.TAUAnTRCm	
TAUAnTME.TAUAnTMEm	0: Disables modulation

(d) Simultaneous rewrite for slave channel 3

The simultaneous rewrite settings of the master and slave channels must be identical.

Table 12-132 Simultaneous rewrite settings for slave channel 3 of the Delay Pulse Output Function

Bit name	Setting
TAUAnRDE.TAUAnRDEm	1: Enables simultaneous rewrite
TAUAnRDS.TAUAnRDSm	0: The master channel is the control channel for simultaneous rewrite
TAUAnRDM.TAUAnRDMm	0: The simultaneous rewrite trigger signal is generated when the master channel starts counting
TAUAnRDC.TAUAnRDCm	0: Channel is not used to detect an INTTAUAnIm signal that is used as the simultaneous rewrite trigger. If TAUAnRDS.TAUAnRDSm = 0, the master channel is used to detect the simultaneous rewrite trigger regardless of the value of this bit.

(7) Operating procedure for Delay Pulse Output Function

Table 12-133 Operating procedure for Delay Pulse Output Function (1/2)

	Operation	Status of TAUAn
Initial channel setting	Master channel: set the TAUAnCMORm and TAUAnCMURm registers and the channel output mode as described in *(3) "Register settings for the master channel" on page 746* Slave channel 1: set the TAUAnCMORm and TAUAnCMURm registers and the channel output mode as described in *(4) "Register settings for slave channel 1" on page 748* Slave channel 2: set the TAUAnCMORm and TAUAnCMURm registers and the channel output mode as described in *(5) "Register settings for slave channel 2" on page 750* Slave channel 3: set the TAUAnCMORm and TAUAnCMURm registers and the channel output mode as described in *(6) "Register settings for slave channel 3" on page 752* Set the values of the TAUAnCDRm registers of all channels	Channel operation is stopped.

Table 12-133 Operating procedure for Delay Pulse Output Function (2/2)

	Operation	Status of TAUAn
Start operation	Set TAUAnTS.TAUAnTSm of the master and slave channels to 1 simultaneously. TAUAnTS.TAUAnTSm is a trigger bit, so it is automatically cleared to 0.	TAUAnTE.TAUAnTEm (master and slave channels) is set to 1 and the counters of the master channel and slave channels 1 and 2 start. INTTAUAnIm is generated on the master channel and TAUAnTTOUTm (slave 1) is set.
During operation	TAUAnCDRm can be changed at any time. TAUAnCNTm and TAUAnRSF.TAUAnRSFm can be read at any time. TAUAnRDT.TAUAnRDTm can be changed during operation.	TAUAnCNTm loads the TAUAnCDRm value of the master channel and slave channels 1 and 2, and then counts down. When the counter of the master channel reaches 0000_H: • INTTAUAnIm (master) is generated. • TAUAnCNTm (master) reloads the TAUAnCDRm value, and then continues count operation. • TAUAnCNTm (slave 1 and slave 2) reload the TAUAnCDRm value and start counting down. • TAUAnTTOUTm (slave 1) is set. When TAUAnCNTm (slave 1) reaches 0000_H: • INTTAUAnIm (slave 1) is generated. • TAUAnTTOUTm (slave 1) is reset. When TAUAnCNTm (slave 2) reaches 0000_H: • INTTAUAnIm (slave 2) is generated. • TAUAnTTOUTm (slave 3) is set. • TAUAnCNTm (slave 3) reloads the TAUAnCDRm value, and then starts counting down. When TAUAnCNTm (slave 3) reaches 0000_H: • INTTAUAnIm (slave 3) is generated. • TAUAnTTOUTm (slave 3) is reset.
Stop operation	Set TAUAnTT.TAUAnTTm of the master and slave channels to 1 simultaneously. TAUAnTT.TAUAnTTm is a trigger bit, so it is automatically cleared to 0.	TAUAnTE.TAUAnTEm is cleared to 0 and the counter stops. TAUAnCNTm and TAUAnTTOUTm stop and retain their current values.

Restart

12.22.4 AD Conversion Trigger Output Function Type 1

(1) Overview

Summary This function is identical to *12.22.1 “PWM Output Function” on page 720*except that TAUAnTTOUTm is not output.

This is achieved by setting the channel output mode of the slave to Independent Channel Output Mode Controlled by Software.

(2) Block diagram and general timing diagram

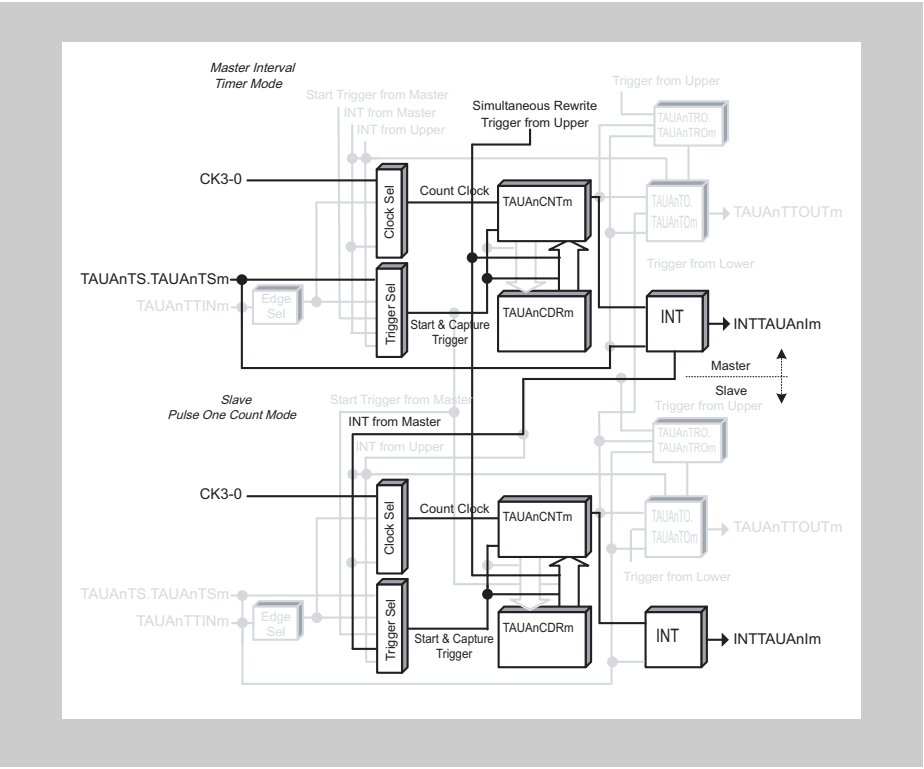


Figure 12-99 Block diagram for AD Conversion Trigger Output Function Type 1

(3) General timing diagram

The following settings apply to the general timing diagram:

- Slave channel: Positive logic (TAUAnTOL.TAUAnTOLm = 0)

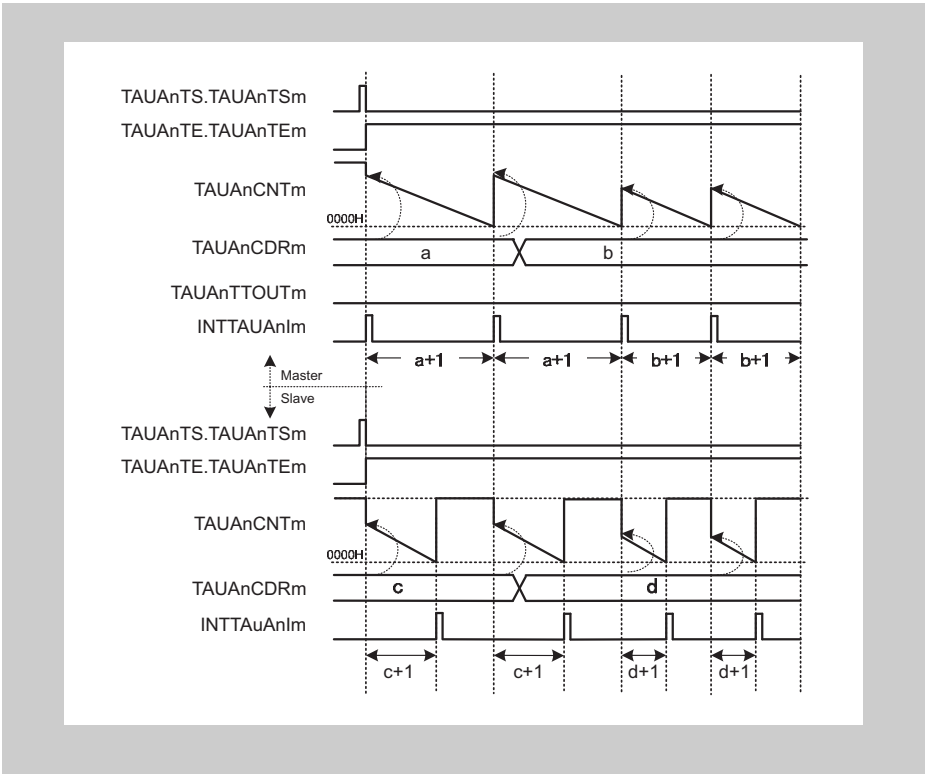


Figure 12-100 General timing diagram for AD Conversion Trigger Output Function Type 1

12.23 Synchronous PWM Signal Functions Triggered by an External Signal

This chapter describes functions that generate PWM signals and which are triggered by an external signal.

- *12.23.1 “One-Shot Pulse Output Function”*
- *12.23.2 “Offset Trigger Output Function”*

12.23.1 One-Shot Pulse Output Function

(1) Overview

Summary This function outputs a signal pulse with a defined pulse width and a specific delay time compared to an external input signal pulse by using a master and a slave channel. The delay time is specified using the master channel. The pulse width is specified using the slave channel.

Prerequisites

- Two channels
- The operation mode of the master channel must be set to One Count Mode. Refer to *Table 12-134 "TAUAnCMORm settings for the master channel of the One-Shot Pulse Output Function" on page 762*.
- The operation mode of the slave channel must be set to Pulse One Count Mode. Refer to *Table 12-137 "TAUAnCMORm settings for the slave channel of the One-Shot Pulse Output Function" on page 764*.
- TAUAnTTOUTm is not used for the master channel of this function.
- The channel output mode of the slave channel must be set to Synchronous Channel Output Mode 2 (refer to *12.8 "Channel Output Modes" on page 577*).
- TAUAnTTINm (master) has to be detected while TAUAnCNTm (master) and TAUAnCNTm (slave) await a trigger. Furthermore, the slave is only triggered by an interrupt from the master channel and not by TAUAnTTINm (slave).

Description The counters are enabled by setting the channel trigger bits (TAUAnTS.TAUAnTSm) of the master and slave channels to 1. This in turn sets TAUAnTE.TAUAnTEm, enabling count operation.

- Master channel:

  When the next valid TAUAnTTINm input edge is detected, the current value of TAUAnCDRm is loaded to TAUAnCNTm. The counter starts to count down from this value. If TAUAnCMORm.TAUAnMD0 = 0, a trigger (TAUAnTTINm) which is detected within the delay time is ignored.
  When the counter of the master channel reaches 0000_H, INTTAUAnIm is generated. The counter returns to $FFFF_H$ and awaits the next valid TAUAnTTINm input edge.

- Slave channel

  The INTTAUAnIm of the master channel triggers the counter of the slave channel. The current value of TAUAnCDRm (slave) is loaded to TAUAnCNTm (slave) and the counter starts to count down from this value. An interrupt is generated and the TAUAnTTOUTm signal is set.
  When the counter reaches 0001_H, INTTAUAnIm is generated and the TAUAnTTOUTm signal is reset. The counter stops at 0000_H and awaits the next INTTAUAnIm of the master channel.

The counter can be stopped by setting TAUAnTT.TAUAnTTm to 1 for the master and slave channel, which in turn sets TAUAnTE.TAUAnTEm to 0. TAUAnCNTm and TAUAnTTOUTm of master and slave channel stop but retain their values. The counters can be restarted by setting TAUAnTS.TAUAnTSm to 1.

The counter of the master channel can be restarted without stopping it first (forced restart) by setting TAUAnTS.TAUAnTSm to 1 during operation.

Conditions

- If TAUAnCMORm.TAUAnMD0 of the master channel is set to 0, during counting detected TAUAnTTINm input edges are ignored.
- Simultaneous rewrite can be used with this function. Please refer to *12.7 "Simultaneous Rewrite" on page 565*

Equations

Delay to input pulse = (TAUAnCDRm (master) + 1) × count clock cycle

Pulse width = (TAUAnCDRm (slave)) × count clock cycle

(2) Block diagram and general timing diagram

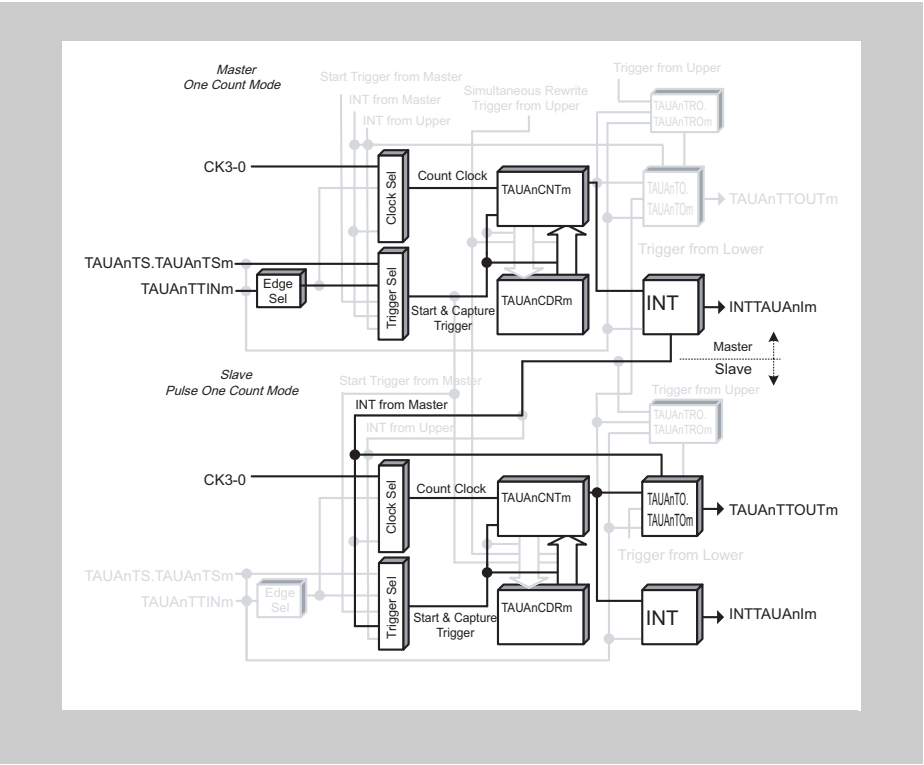


Figure 12-101 Block diagram for One-Shot Pulse Output Function

The following settings apply to the general timing diagram:

- Start trigger detection disabled during counting (TAUAnCMORm.TAUAnMD0 = 0)
- Falling edge detection (TAUAnCMURm.TAUAnTIS[1:0] = 00_B)

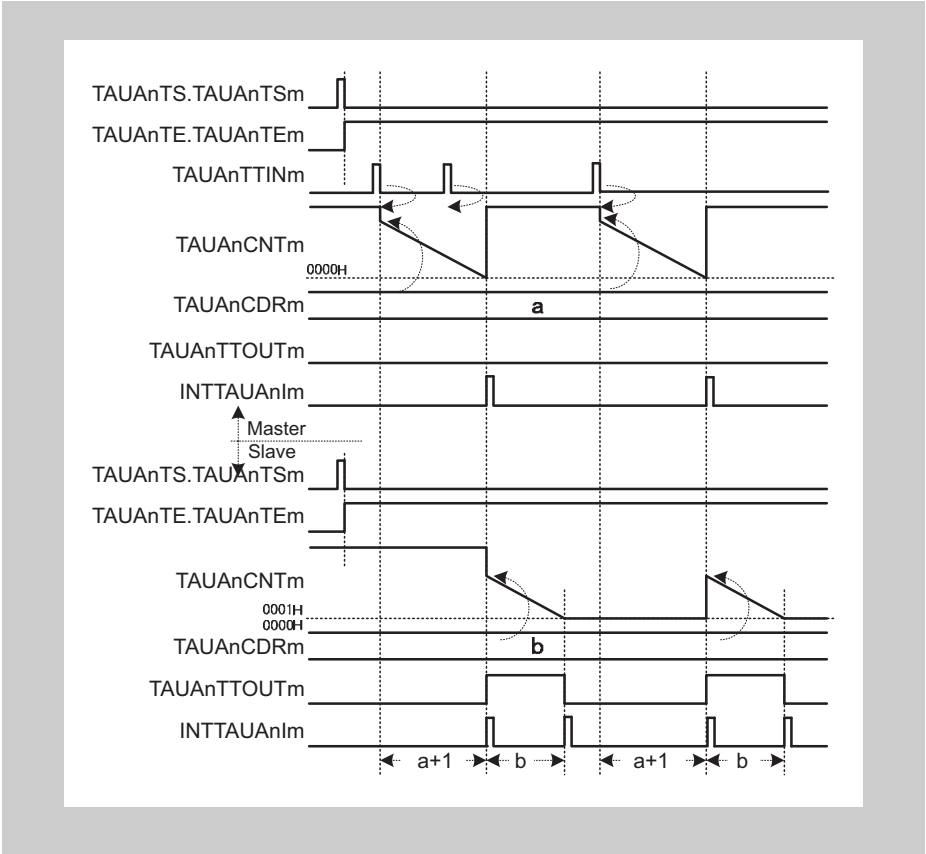


Figure 12-102 General timing diagram for One-Shot Pulse Output Function

(3) Register settings for the master channel

(a) TAUAnCMORm for the master channel

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn CKS[1:0]		TAUAn CCS[1:0]		TAUAn MAS	TAUAnSTS[2:0]			TAUAn COS[1:0]		-	TAUAnMD[4:1]				TAUAn MD0

Table 12-134 TAUAnCMORm settings for the master channel of the One-Shot Pulse Output Function

Bit name	Setting
TAUAnCKS[1:0]	00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 The value of the TAUAnCKS[1:0] bit of the master and slave channel must be identical.
TAUAnCCS[1:0]	00: Operation clock is used as the count clock
TAUAnMAS	1: Channel is master channel
TAUAnSTS[2:0]	001: Valid TAUAnTTINm input edge signal is used as the start trigger
TAUAnCOS[1:0]	00: Not used, so set to 00
TAUAnMD[4:1]	0100: One Count Mode
TAUAnMD0	0: Disables start trigger detection during counting 1: Enables start trigger detection during counting The value of the TAUAnMD0 bit of the master and slave channel must be identical.

(b) TAUAnCMURm for the master channel

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-														TAUAnTIS[1:0]	

Table 12-135 TAUAnCMURm settings for the master channel of the One-Shot Pulse Output Function

Bit name	Setting
TAUAnTIS[1:0]	00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection 11: Setting prohibited

(c) Channel output mode for the master channel

Because the channel output mode is not used by this function, clear TAUAnTOE.TAUAnTOEm. However, it can be used by other functions or in Independent Channel Output Mode Controlled by Software.

(d) Simultaneous rewrite for the master channel

The simultaneous rewrite settings of the master and slave channel must be identical.

Table 12-136 Simultaneous rewrite settings for the master channel of the One-Shot Pulse Output Function

Bit name	Setting
TAUAnRDE.TAUAnRDEm	1: Enables simultaneous rewrite
TAUAnRDS.TAUAnRDSm	0: The master channel is the control channel for simultaneous rewrite
TAUAnRDM.TAUAnRDMm	0: The simultaneous rewrite trigger signal is generated when the master channel starts counting
TAUAnRDC.TAUAnRDCm	0: Channel is not monitored for an INTTAUAnIm signal that is used as the simultaneous rewrite trigger. If TAUAnRDS.TAUAnRDSm = 0, the master channel is monitored for the simultaneous rewrite trigger regardless of the value of this bit.

(4) Register settings for the slave channel

(a) TAUAnCMORm for the slave channel

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn CKS[1:0]		TAUAn CCS[1:0]		TAUAn MAS	TAUAnSTS[2:0]			TAUAn COS[1:0]		-	TAUAnMD[4:1]				TAUAn MD0

Table 12-137 TAUAnCMORm settings for the slave channel of the One-Shot Pulse Output Function

Bit name	Setting
TAUAnCKS[1:0]	00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 The value of the TAUAnCKS[1:0] bit of the master and slave channel must be identical.
TAUAnCCS[1:0]	00: Operation clock is used as the count clock
TAUAnMAS	0: Channel is a slave channel
TAUAnSTS[2:0]	100: INTTAUAnIm of the master channel is the start trigger
TAUAnCOS[1:0]	00: Not used, so set to 00
TAUAnMD[4:1]	1010: Pulse One Count Mode
TAUAnMD0	0: Disables start trigger detection during counting 1: Enables start trigger detection during counting The value of the TAUAnMD0 bit of the master and slave channel must be identical.

(b) TAUAnCMURm for the slave channel

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-														TAUAnTIS[1:0]	

Table 12-138 TAUAnCMURm settings for the slave channel of the One-Shot Pulse Output Function

Bit name	Setting
TAUAnTIS[1:0]	00: These are not used, so set them to 00.

(c) Channel output mode for the slave channel

Table 12-139 Control bit settings for Synchronous Channel Output Mode 2

Bit name	Setting
TAUAnTOE.TAUAnTOEm	1: Enables Independent Channel Output Mode
TAUAnTOM.TAUAnTOMm	0: Independent channel output
TAUAnTOC.TAUAnTOCm	1: Operation mode 2
TAUAnTOL.TAUAnTOLm	0: Positive logic 1: Inverted logic
TAUAnTDE.TAUAnTDEm	0: Disables dead time operation
TAUAnTDM.TAUAnTDMm	0: When dead time operation is disabled (TAUAnTDE.TAUAnTDEm = 0), set these bits to 0
TAUAnTDL.TAUAnTDLm	0: When dead time operation is disabled (TAUAnTDE.TAUAnTDEm = 0), set these bits to 0
TAUAnTRE.TAUAnTREm	0: Disables real-time output
TAUAnTRO.TAUAnTROm	0: When real-time output is disabled (TAUAnTRE.TAUAnTREm = 0), set these bits to 0
TAUAnTRC.TAUAnTRCm	0: When real-time output is disabled (TAUAnTRE.TAUAnTREm = 0), set these bits to 0
TAUAnTME.TAUAnTMEm	0: Disables modulation

(d) Simultaneous rewrite for the slave channel

The simultaneous rewrite settings of the master and slave channel must be identical.

Table 12-140 Simultaneous rewrite settings for the slave channel of the One-Shot Pulse Output Function

Bit name	Setting
TAUAnRDE.TAUAnRDEm	1: Enables simultaneous rewrite
TAUAnRDS.TAUAnRDSm	0: The master channel is the control channel for simultaneous rewrite
TAUAnRDM.TAUAnRDMm	0: The simultaneous rewrite trigger signal is generated when the master channel starts counting
TAUAnRDC.TAUAnRDCm	0: Channel is not monitored for an INTTAUAnIm signal that is used as the simultaneous rewrite trigger. If TAUAnRDS.TAUAnRDSm = 0, the master channel is monitored for the simultaneous rewrite trigger regardless of the value of this bit.

(5) Operating procedure for One-Shot Pulse Output Function

Table 12-141 Operating procedure for One-Shot Pulse Output Function

	Operation	Status of TAUAn
Initial channel setting	Master channel: set the TAUAnCMORm and TAUAnCMURm registers and the channel output mode as described in *(3) "Register settings for the master channel" on page 762* Slave channel: set the TAUAnCMORm and TAUAnCMURm registers and the channel output mode as described in *(4) "Register settings for the slave channel" on page 764* Set the values of the TAUAnCDRm registers of all channels	Channel operation is stopped.
Start operation	Set TAUAnTS.TAUAnTSm of the master and slave channels to 1 simultaneously. TAUAnTS.TAUAnTSm is a trigger bit, so it is automatically cleared to 0.	TAUAnTE.TAUAnTEm (master and slave channels) is set to 1 and the master channel awaits a TAUAnTTINm input.
During operation	TAUAnCDRm can be changed at any time. TAUAnCNTm and TAUAnRSF.TAUAnRSFm can be read at any time. TAUAnRDT.TAUAnRDTm can be changed during operation.	When the valid edge of TAUAnTTINm input is detected, TAUAnCNTm loads the TAUAnCDRm value of the master channel, and then counts down. When the counter reaches 0000_H: • INTTAUAnIm (master) is generated. • TAUAnCNTm (master) reloads the TAUAnCDRm value, and then continues count operation. • TAUAnCNTm (slave) reloads the TAUAnCDRm value, and then starts counting down. • INTTAUAnIm (slave) is generated. • TAUAnTTOUTm (slave) is set. When TAUAnCNTm (slave) reaches 0001_H: • INTTAUAnIm (slave) is generated. • TAUAnTTOUTm (slave) is reset. If a TAUAnTTINm input is detected on the master channel while the counter is counting, the input is ignored when TAUAnCMORm.TAUAnMD0 = 0.
Stop operation	Set TAUAnTT.TAUAnTTm of the master and slave channels to 1 simultaneously. TAUAnTT.TAUAnTTm is a trigger bit, so it is automatically cleared to 0.	TAUAnTE.TAUAnTEm is cleared to 0 and the counter stops. TAUAnCNTm and TAUAnTTOUTm stop and retain their current values.

Restart

12.23.2 Offset Trigger Output Function

(1) Overview

Summary This function generates a PWM output using a master and a slave channel. It enables the pulse width (duration) of the TAUAnTTOUTm to be set. The pulse cycle is set by the detection of a valid input edge of the master channel. The pulse width is specified using the slave channel.

Prerequisites

- Two channels
- The operation mode of the master channel must be set to Capture Mode. Refer to *Table 12-142 "TAUAnCMORm settings for the master channel of the Offset Trigger Output Function" on page 770*.
- The operation mode of the slave channel must be set to One Count Mode. Refer to *Table 12-145 "TAUAnCMORm settings for the slave channel of the Offset Trigger Output Function" on page 772*.
- The channel output mode of the slave channel must be set to Independent Channel Output Mode 2 (refer to *12.8 "Channel Output Modes" on page 577*).
- TAUAnTTOUTm is not used for the master channel of this function.

Description The counters are started by setting the channel trigger bit (TAUAnTS.TAUAnTSm) to 1. This in turn sets TAUAnTE.TAUAnTEm, enabling count operation. The counter of the master channel (TAUAnCNTm) starts to count up from 0000_H.

- Master channel:

  When a valid TAUAnTTINm input edge is detected, the current value of the counter (TAUAnCNTm) is loaded to the data register of the master channel (TAUAnCDRm). INTTAUAnIm is generated and the counter restarts counting up from 0000_H.

- Slave channel:

  The INTTAUAnIm of the master channel sets the TAUAnTTOUTm (slave) signal and triggers the counter of the slave channel. The current value of TAUAnCDRm (slave) is loaded to TAUAnCNTm (slave) and the counter starts to count down from this value.
  When the counter reaches 0000_H, i.e. duty time has elapsed, INTTAUAnIm is generated and the TAUAnTTOUTm signal is reset. The counter returns to $FFFF_H$ and awaits the next INTTAUAnIm of the master channel.

The counter can be stopped by setting TAUAnTT.TAUAnTTm to 1 for the master and slave channel(s), which in turn sets TAUAnTE.TAUAnTEm to 0. TAUAnCNTm and TAUAnTTOUTm of master and slave channel(s) stop but retain their values. The counters can be restarted by setting TAUAnTS.TAUAnTSm to 1.

(2) Equations

Pulse width = (TAUAnCDRm (slave) + 1) x count clock cycle

Duty cycle [%] = [TAUAnCDRm (slave) / TAUAnTTINm cycle + 1)] x 100

– Duty cycle = 0 %

TAUAnCDRm (slave) = 0000_H

– Duty cycle = 100 %

TAUAnCDRm (slave) ≥ TAUAnTTINm cycle + 1

(3) Block diagram and general timing diagram

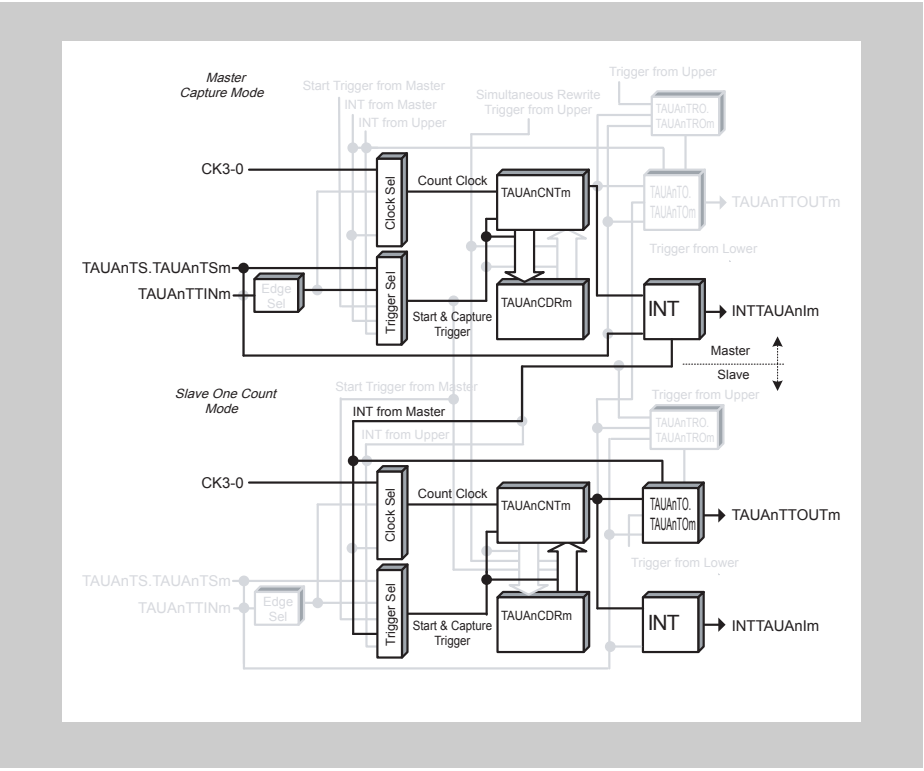


Figure 12-103 Block diagram for Offset Trigger Output Function

The following settings apply to the general timing diagram:

- Falling edge detection (TAUAnCMURm.TAUAnTIS[1:0] = 00_B)

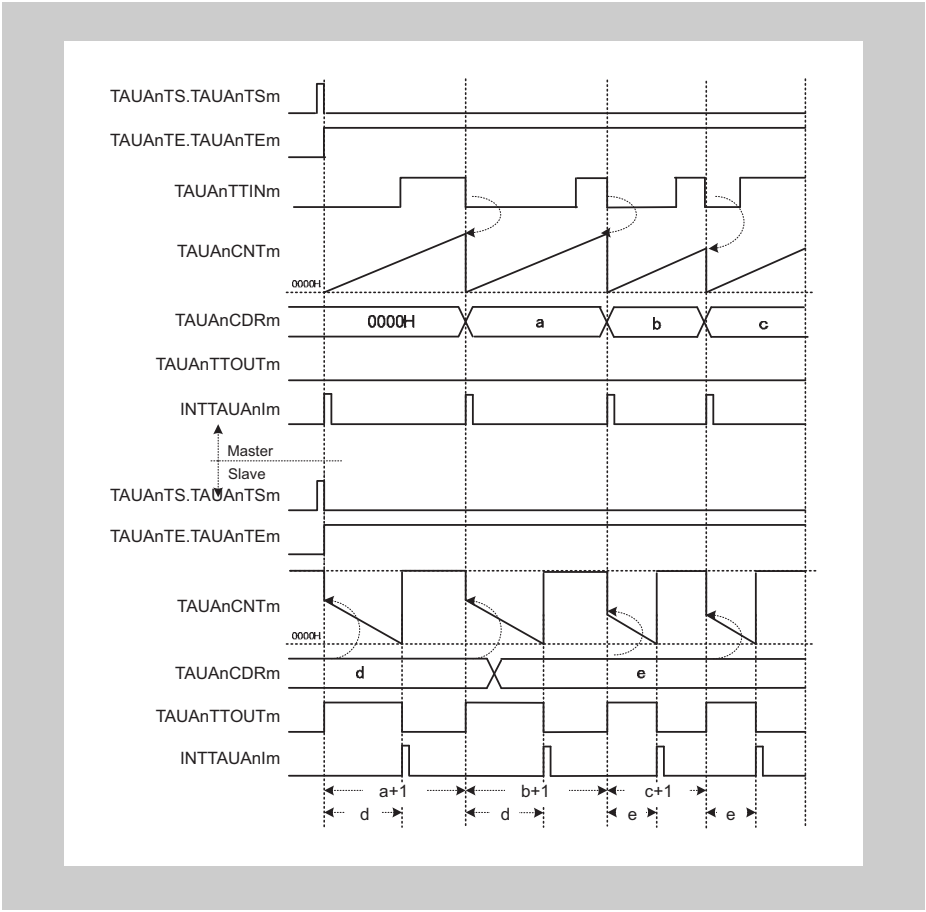


Figure 12-104 General timing diagram for Offset Trigger Output Function

(4) Register settings for the master channel

(a) TAUAnCMORm for the master channel

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn CKS[1:0]		TAUAn CCS[1:0]		TAUAn MAS	TAUAnSTS[2:0]			TAUAn COS[1:0]		-	TAUAnMD[4:1]				TAUAn MD0

Table 12-142 TAUAnCMORm settings for the master channel of the Offset Trigger Output Function

Bit name	Setting
TAUAnCKS[1:0]	00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 The value of the TAUAnCKS[1:0] bit of the master and slave channel(s) must be identical.
TAUAnCCS[1:0]	00: Operation clock is used as the count clock
TAUAnMAS	1: Channel is master channel
TAUAnSTS[2:0]	001: Valid TAUAnTTINm input edge signal is used as the start trigger
TAUAnCOS[1:0]	11: Capture register updated upon detection of a TAUAnTTINm input valid edge and upon counter overflow: - • TAUAnTTINm input valid edge: Counter value is written to TAUAnCDRm - • Overflow: $FFFF_H$ is written to TAUAnCDRm. The next TAUAnTTINm input valid edge detection is ignored. TAUAnCSRm.TAUAnOVF set by counter overflow and cleared by a CPU instruction.
TAUAnMD[4:1]	0010: Capture Mode
TAUAnMD0	1: Generates INTTAUAnIm at operation start

(b) TAUAnCMURm for the master channel

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-														TAUAnTIS[1:0]	

Table 12-143 TAUAnCMURm settings for the master channel of the Offset Trigger Output Function

Bit name	Setting
TAUAnTIS[1:0]	00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection 11: Setting prohibited

(c) Channel output mode for the master channel

Because the channel output mode is not used by this function, clear TAUAnTOE.TAUAnTOEm. However, it can be used by other functions or in Independent Channel Output Mode Controlled by Software.

(d) Simultaneous rewrite for the master channel

The simultaneous rewrite registers (TAUAnRDE, TAUAnRDS, TAUAnRDM, and TAUAnRDC) cannot be used with the Offset Trigger Output Function. Therefore, these registers must be set to 0.

Table 12-144 Simultaneous rewrite settings for the master channel of the Offset Trigger Output Function

Bit name	Setting
TAUAnRDE.TAUAnRDEm	0: Disables simultaneous rewrite
TAUAnRDS.TAUAnRDSm	When simultaneous rewrite is disabled (TAUAnTAUAnRDE.TAUAnRDEm = 0), set these bits to 0
TAUAnRDM.TAUAnRDMm	
TAUAnRDC.TAUAnRDCm	

(5) Register settings for the slave channel

(a) TAUAnCMORm for the slave channel

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn CKS[1:0]		TAUAn CCS[1:0]		TAUAn MAS	TAUAnSTS[2:0]			TAUAn COS[1:0]		-	TAUAnMD[4:1]				TAUAn MD0

Table 12-145 TAUAnCMORm settings for the slave channel of the Offset Trigger Output Function

Bit name	Setting
TAUAnCKS[1:0]	00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 The value of the TAUAnCKS[1:0] bit of the master and slave channel must be identical.
TAUAnCCS[1:0]	00: Operation clock is used as the count clock
TAUAnMAS	0: Channel is a slave channel
TAUAnSTS[2:0]	100: INTTAUAnIm of the master channel is the start trigger
TAUAnCOS[1:0]	00: Not used, so set to 00
TAUAnMD[4:1]	0100: One Count Mode
TAUAnMD0	1: Enables start trigger detection during counting

(b) TAUAnCMURm for the slave channel(s)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-														TAUAnTIS[1:0]	

Table 12-146 TAUAnCMURm settings for the slave channel of the Offset Trigger Output Function

Bit name	Setting
TAUAnTIS[1:0]	00: These are not used, so set them to 00.

(c) Channel output mode for the slave channel

Table 12-147 Control bit settings for Independent Channel Output Mode 2

Bit name	Setting
TAUAnTOE.TAUAnTOEm	1: Enables Independent Channel Output Mode
TAUAnTOM.TAUAnTOMm	0: Independent channel output operation
TAUAnTOC.TAUAnTOCm	1: Operation mode 2
TAUAnTOL.TAUAnTOLm	0: Positive logic 1: Inverted logic
TAUAnTDE.TAUAnTDEm	0: Disables dead time operation
TAUAnTDM.TAUAnTDMm	0: When dead time operation is disabled (TAUAnTDE.TAUAnTDEm = 0), set these bits to 0
TAUAnTDL.TAUAnTDLm	
TAUAnTRE.TAUAnTREm	0: Disables real-time output
TAUAnTRO.TAUAnTROm	0: When real-time output is disabled (TAUAnTRE.TAUAnTREm = 0), set these bits to 0
TAUAnTRC.TAUAnTRCm	
TAUAnTME.TAUAnTMEm	0: Disables modulation

(d) Simultaneous rewrite for the slave channel

The simultaneous rewrite registers (TAUAnRDE, TAUAnRDS, TAUAnRDM, and TAUAnRDC) cannot be used with the Offset Trigger Output Function. Therefore, these registers must be set to 0.

Table 12-148 Simultaneous rewrite settings for the slave channel of the Offset Trigger Output Function

Bit name	Setting
TAUAnRDE.TAUAnRDEm	0: Disables simultaneous rewrite
TAUAnRDS.TAUAnRDSm	When simultaneous rewrite is disabled (TAUAnTAUAnRDE.TAUAnRDEm = 0), set these bits to 0
TAUAnRDM.TAUAnRDMm	
TAUAnRDC.TAUAnRDCm	

(6) Operating procedure for Offset Trigger Output Function

Table 12-149 Operating procedure for Offset Trigger Output Function

	Operation	Status of TAUAn
Initial channel setting	Master channel: set the TAUAnCMORm and TAUAnCMURm registers and the channel output mode as described in *(4) "Register settings for the master channel" on page 770* Slave channel: set the TAUAnCMORm and TAUAnCMURm registers and the channel output mode as described in *(5) "Register settings for the slave channel" on page 772* Set the values of the TAUAnCDRm registers of all channels	Channel operation is stopped.
Start operation	Set TAUAnTS.TAUAnTSm of the master and slave channels to 1 simultaneously. TAUAnTS.TAUAnTSm is a trigger bit, so it is automatically cleared to 0.	TAUAnTE.TAUAnTEm (master and slave channels) is set to 1 and the counters of the master and slave channels start: • TAUAnCNTm (master) counts up. • TAUAnCNTm (slave) loads TAUAnCDRm, and then counts down. INTTAUAnIm is generated on the master channel and TAUAnTTOUTm (master) is set.
During operation	TAUAnCDRm can be changed at any time. TAUAnCSCm.TAUAnCLOV can be set to 1.	When TAUAnCNTm of the slave reaches 0000_H: • INTTAUAnIm (slave) is generated. • TAUAnTTOUTm (slave) is reset. When a TAUAnTTINm input is detected on the master channel: • INTTAUAnIm (master) is generated. • TAUAnCNTm (master) is reset to 0000_H and continues count operation. • TAUAnCNTm (slave) reloads the TAUAnCDRm value, and then counts down. • TAUAnTTOUTm (slave) is set.
Stop operation	Set TAUAnTT.TAUAnTTm of the master and slave channels to 1 simultaneously. TAUAnTT.TAUAnTTm is a trigger bit, so it is automatically cleared to 0.	TAUAnTE.TAUAnTEm is cleared to 0 and the counter stops. TAUAnCNTm and TAUAnTTOUTm stop and retain their current values.

Restart

12.24 Synchronous Triangle PWM Functions

This chapter describes functions that generate a triangle PWM output.

- *12.24.1 “Triangle PWM Output Function”*
- *12.24.2 “Triangle PWM Output Function with Dead Time”*
- *12.24.3 “AD Conversion Trigger Output Function Type 2”*

12.24.1 Triangle PWM Output Function

(1) Overview

Summary

This function generates multiple triangle PWM outputs by using a master and one or more slave channels. It enables the pulse cycle (frequency) and the duty cycle of TAUAnTTOUTm to be set using the master and slave channel(s) respectively.

Carrier cycles are generated on the master channel. The first pulse of the master channel controls the down status and the second pulse controls the up status of the slaves counter.

Prerequisites

- Two channels
- The operation mode of the master channel must be set to Interval Timer Mode. Refer to *Table 12-150 “TAUAnCMORm settings for the master channel of the Triangle PWM Output Function” on page 780*.
- The operation mode of the slave channel(s) must be set to Up Down Count Mode. Refer to *Table 12-154 “TAUAnCMORm settings for the slave channel of the Triangle PWM Output Function” on page 782*.
- The channel output mode of the master channel must be set to Independent Channel Output Mode 1 (refer to *12.8 “Channel Output Modes” on page 577*).
- The channel output mode of the slave channel(s) must be set to Synchronous Channel Output Mode 2 (refer to *12.8 “Channel Output Modes” on page 577*).
- The following settings establish TAUAnTTOUTm at high level for the down status of the carrier cycle.
  - If the TAUAnCMORm.TAUAnMD0 (master) bit is set to 0, TAUAnTO.TAUAnTOm must be set to 1 while TAUAnTOE.TAUAnTOEm is 0 (recommended).
  - If the TAUAnCMORm.TAUAnMD0 (master) bit is set to 1, TAUAnTO.TAUAnTOm must be set to 0 while TAUAnTOE.TAUAnTOEm is 0.

Description

The counters are started by setting the channel trigger bit (TAUAnTS.TAUAnTSm) to 1 for every channel. This in turn sets TAUAnTE.TAUAnTEm, enabling count operation. The values of TAUAnCDRm (master and slave) are loaded to TAUAnCNTm (master and slave) and the counters start to count down from these values. If the master channel TAUAnCMORm.TAUAnMD0 bit is set, an interrupt is generated and TAUAnTTOUTm signal of the master toggles.

- Master channel:

  When the counter of the master channel reaches 0000_H, pulse cycle time has elapsed, INTTAUAnIm is generated and the TAUAnTTOUTm signal toggles. TAUAnCNTm then reloads the TAUAnCDRm value, and then counts down.

- Slave channel:

  The INTTAUAnIm of the master channel triggers the counter of the slave channel:

  – If the slave counter currently counts down, it changes count direction.

  – If the slave counter currently counts up, the value of TAUAnCDRm is reloaded and the counter counts down.

  When the counter of the slave channel reaches 0001_H while counting up or down, INTTAUAnIm is generated and the TAUAnTTOUTm (slave) signal is set or reset.

  The counter continues to count down or up and awaits the next INTTAUAnIm of the master channel.

TAUAnTTOUTm can be switched between positive and negative phase setting TAUAnTOL.TAUAnTOLm during operation.

The counters can be stopped by setting TAUAnTT.TAUAnTTm to 1 for the master and slave channel(s), which in turn sets TAUAnTE.TAUAnTEm to 0. TAUAnCNTm and TAUAnTTOUTm of master and slave channel(s) stop but retain their values.

Conditions Simultaneous rewrite can be used with this function. Please refer to *12.7 "Simultaneous Rewrite" on page 565*

(2) Equations

Pulse cycle = (TAUAnCDRm (master) + 1) x count clock cycle

$0000_H \leq$ TAUAnCDR (master) $< FFFF_H$

Carrier cycle (down/up) = (TAUAnCDRm (master) + 1) × 2 × count clock cycle

Duty cycle [%] =
[(TAUAnCDRm (master) + 1 - TAUAnCDRm (slave))/ (TAUAnCDRm (master) + 1)] x 100

– Duty cycle = 100 %

  TAUAnCDRm (slave) = 0000_H

– Duty cycle = 0 %

  TAUAnCDRm (slave) ≥ TAUAnCDRm (master) + 1

(3) Block diagram and general timing diagram

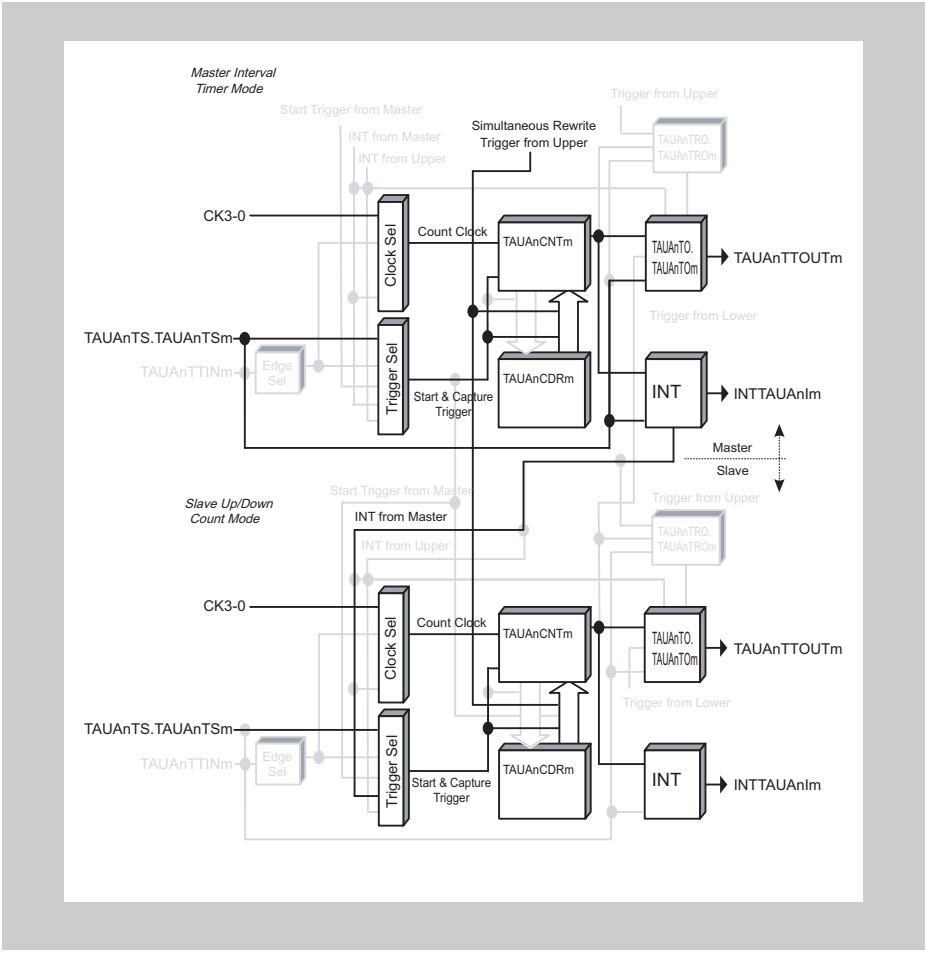


Figure 12-105 Block diagram for Triangle PWM Output Function

The following settings apply to the general timing diagram:

- Master channel
  - INTTAUAnIm is generated at operation start (TAUAnCMORm.TAUAnMD0 = 1)

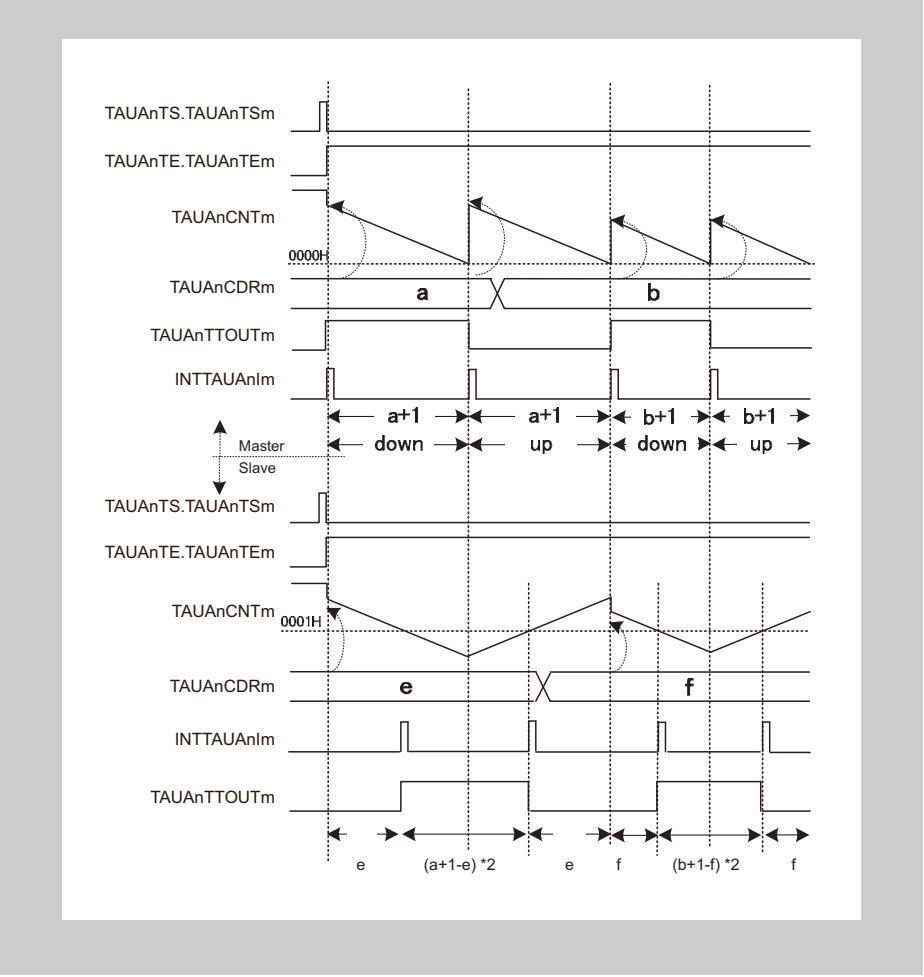


Figure 12-106 General timing diagram for Triangle PWM Output Function

(4) Register settings for the master channel

(a) TAUAnCMORm for the master channel

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn CKS[1:0]		TAUAn CCS[1:0]		MAS	TAUAnSTS[2:0]			TAUAn COS[1:0]		-	TAUAnMD[4:1]				TAUAn MD0

Table 12-150 TAUAnCMORm settings for the master channel of the Triangle PWM Output Function

Bit name	Setting
TAUAnCKS[1:0]	00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 The value of the TAUAnCKS[1:0] bit of the master and slave channel(s) must be identical.
TAUAnCCS[1:0]	00: Operation clock is used as the count clock
TAUAnMAS	1: Channel is master channel
TAUAnSTS[2:0]	000: Counter triggered by software trigger
TAUAnCOS[1:0]	00: Not used, so set to 00
TAUAnMD[4:1]	0000: Interval Timer Mode
TAUAnMD0	0: INTTAUAnIm not generated and TAUAnTTOUTm does not toggle at operation start 1: Generates INTTAUAnIm and toggles TAUAnTTOUTm at operation start

(b) TAUAnCMURm for the master channel

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-														TAUAnTIS[1:0]	

Table 12-151 TAUAnCMURm settings for the master channel of the Triangle PWM Output Function

Bit name	Setting
TAUAnTIS[1:0]	00: These are not used, so set them to 00.

(c) Channel output mode for the master channel

Table 12-152 Control bit settings for Independent Channel Output Mode 1

Bit name	Setting
TAUAnTOE.TAUAnTOEm	1: Enables Independent Channel Output Mode
TAUAnTOM.TAUAnTOMm	0: Independent channel output
TAUAnTOC.TAUAnTOCm	0: Operation mode 1 (= Toggle mode if TAUAnTOM.TAUAnTOMm = 0)
TAUAnTOL.TAUAnTOLm	0: Positive logic
TAUAnTDE.TAUAnTDEm	0: Disables dead time operation
TAUAnTDM.TAUAnTDMm	0: When dead time operation is disabled (TAUAnTDE.TAUAnTDEm = 0), set these bits to 0
TAUAnTDL.TAUAnTDLm	
TAUAnTRE.TAUAnTREm	0: Disables real-time output
TAUAnTRO.TAUAnTROm	0: When real-time output is disabled (TAUAnTRE.TAUAnTREm = 0), set these bits to 0
TAUAnTRC.TAUAnTRCm	
TAUAnTME.TAUAnTMEm	0: Disables modulation

(d) Simultaneous rewrite for the master channel

The simultaneous rewrite settings of the master and slave channel must be identical.

Table 12-153 Simultaneous rewrite settings for the master channel of the Triangle PWM Output Function

Bit name	Setting
TAUAnRDE.TAUAnRDEm	1: Enables simultaneous rewrite
TAUAnRDS.TAUAnRDSm	0: The master channel is monitored for the simultaneous rewrite trigger 1: An upper channel outside the channel group is monitored for the simultaneous rewrite trigger
TAUAnRDM.TAUAnRDMm	1: The simultaneous rewrite trigger signal is generated when the master channel starts counting and the corresponding slave channel is at the peak of a triangular wave
TAUAnRDC.TAUAnRDCm	0: Channel is not monitored for an INTTAUAnIm signal that is used as the simultaneous rewrite trigger. If TAUAnRDS.TAUAnRDSm = 0, the master channel is monitored for the simultaneous rewrite trigger regardless of the value of this bit.

Note If the TAUAnRDS.TAUAnRDSm bit is 1, there must be a channel higher than the master channel to which a simultaneous rewrite trigger signal is generated.

(5) Register settings for the slave channel(s)

(a) TAUAnCMORm for the slave channel(s)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn CKS[1:0]		TAUAn CCS[1:0]		TAUAn MAS	TAUAnSTS[2:0]			TAUAn COS[1:0]		-	TAUAnMD[4:1]				TAUAn MD0

Table 12-154 TAUAnCMORm settings for the slave channel of the Triangle PWM Output Function

Bit name	Setting
TAUAnCKS[1:0]	00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 The value of the TAUAnCKS[1:0] bit of the master and slave channel(s) must be identical.
TAUAnCCS[1:0]	00: Operation clock is used as the count clock
TAUAnMAS	0: Channel is a slave channel
TAUAnSTS[2:0]	111: The up/down output trigger signal of the master channel
TAUAnCOS[1:0]	00: Not used, so set to 00
TAUAnMD[4:1]	1001: Up Down Count Mode
TAUAnMD0	0: INTTAUAnIm not generated at operation start

(b) TAUAnCMURm for the slave channel(s)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-														TAUAnTIS[1:0]	

Table 12-155 TAUAnCMURm settings for the slave channel of the Triangle PWM Output Function

Bit name	Setting
TAUAnTIS[1:0]	00: These are not used, so set them to 00.

(c) Channel output mode for the slave channel(s)

Table 12-156 Control bit settings for Synchronous Channel Output Mode 2

Bit name	Setting
TAUAnTOE.TAUAnTOEm	1: Enables Independent Channel Output Mode
TAUAnTOM.TAUAnTOMm	1: Synchronous channel operation
TAUAnTOC.TAUAnTOCm	1: Operation mode 2
TAUAnTOL.TAUAnTOLm	0: Positive logic 1: Inverted logic
TAUAnTDE.TAUAnTDEm	0: Disables dead time operation
TAUAnTDM.TAUAnTDMm	0: When dead time operation is disabled (TAUAnTDE.TAUAnTDEm = 0), set these bits to 0
TAUAnTDL.TAUAnTDLm	
TAUAnTRE.TAUAnTREm	0: Disables real-time output
TAUAnTRO.TAUAnTROm	0: When real-time output is disabled (TAUAnTRE.TAUAnTREm = 0), set these bits to 0
TAUAnTRC.TAUAnTRCm	
TAUAnTME.TAUAnTMEm	0: Disables modulation

(d) Simultaneous rewrite for the slave channel(s)

The simultaneous rewrite settings of the master and slave channel must be identical.

Table 12-157 Simultaneous rewrite settings for the slave channel of the Triangle PWM Output Function

Bit name	Setting
TAUAnRDE.TAUAnRDEm	1: Enables simultaneous rewrite
TAUAnRDS.TAUAnRDSm	0: The master channel is monitored for the simultaneous rewrite trigger 1: An upper channel is monitored for the simultaneous rewrite trigger
TAUAnRDM.TAUAnRDMm	1: The simultaneous rewrite trigger signal is generated when the master channel starts counting and the corresponding slave channel is at the peak of a triangular wave
TAUAnRDC.TAUAnRDCm	0: Channel is not monitored for an INTTAUAnIm signal that is used as the simultaneous rewrite trigger. If TAUAnRDS.TAUAnRDSm = 0, the master channel is monitored for the simultaneous rewrite trigger regardless of the value of this bit.

(6) Operating procedure for Triangle PWM Output Function

Table 12-158 Operating procedure for Triangle PWM Output Function

	Operation	Status of TAUAn
Initial channel setting	Master channel: set the TAUAnCMORm and TAUAnCMURm registers and the channel output mode as described in *(4) "Register settings for the master channel" on page 780* Slave channel: set the TAUAnCMORm and TAUAnCMURm registers and the channel output mode as described in *(5) "Register settings for the slave channel(s)" on page 782* Set the values of the TAUAnCDRm registers of all channels	Channel operation is stopped.
Start operation	Set TAUAnTS.TAUAnTSm of the master and slave channels to 1 simultaneously. TAUAnTS.TAUAnTSm is a trigger bit, so it is automatically cleared to 0.	TAUAnTE.TAUAnTEm (master and slave channels) is set to 1 and the counters of the master and slave channels start. If TAUAnCMORm.TAUAnMD0 is set to 1 on the master channel, INTTAUAnIm (master) is generated.
During operation	TAUAnCDRm can be changed at any time. TAUAnCNTm and TAUAnRSF.TAUAnRSFm can be read at any time. TAUAnRDT.TAUAnRDTm can be changed during operation.	TAUAnCNTm loads the TAUAnCDRm value of the master channel or slave channel, and then counts down. When the counter of the master channel reaches 0000_H: • INTTAUAnIm (master) is generated. • TAUAnTTOUTm (master) toggles. • TAUAnCNTm (master) reloads the TAUAnCDRm value, and then continues count operation. • TAUAnCNTm (slave) reloads the TAUAnCDRm value or counts in the reverse direction. When TAUAnCNTm of the slave = 0001_H: • INTTAUAnIm (slave) is generated. • TAUAnTTOUTm (slave) is set (in count-down status) or reset (in count-up status).
Stop operation	Set TAUAnTT.TAUAnTTm of the master and slave channels to 1 simultaneously. TAUAnTT.TAUAnTTm is a trigger bit, so it is automatically cleared to 0.	TAUAnTE.TAUAnTEm is cleared to 0 and the counter stops. TAUAnCNTm and TAUAnTTOUTm stop and retain their current values.

Restart

(7) Specific timing diagrams

(a) Duty cycle = 0%

The following settings apply to the general timing diagram:

- Master channel:
  - INTTAUAnIm is generated at operation start (TAUAnCMORm.TAUAnMD0 = 1)
  - TAUAnCDRm = a = 5_H
- Slave channel:
  - TAUAnCDRm = 6_H

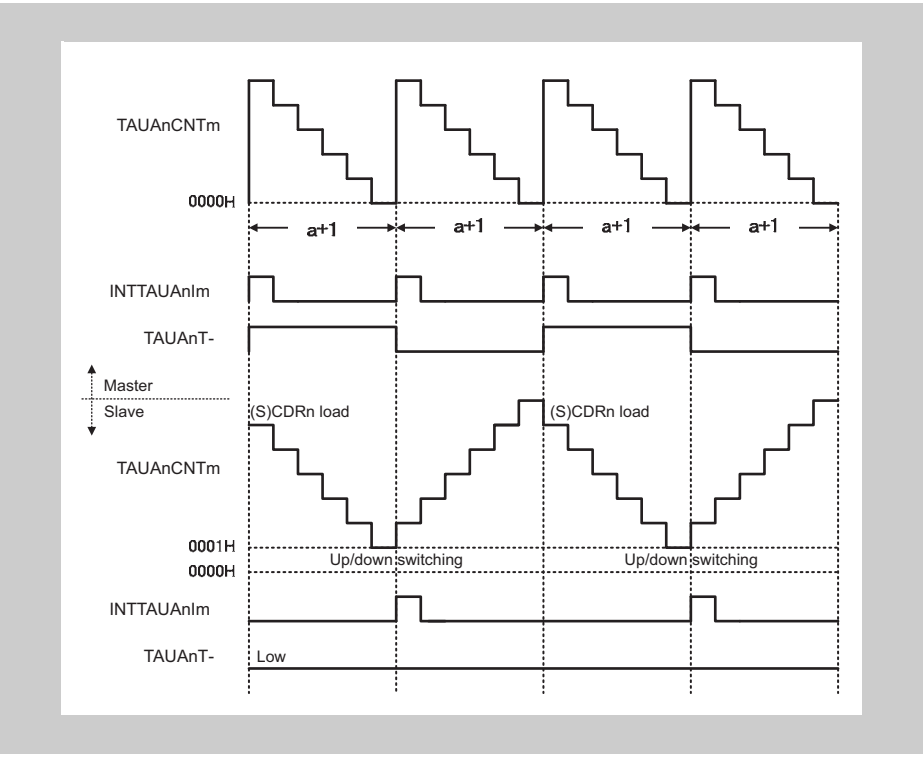


Figure 12-107 TAUAnCDRm (slave) ≥ TAUAnCDRm (master) + 1

- If TAUAnCDRm (slave) ≥ TAUAnCDRm (master) the counter of slave channel cannot reach 0001_H during *counting down*. The set signal is never detected, so TAUAnTTOUTm remains at low state.

(b) Duty cycle = 100%

The following settings apply to the general timing diagram:

- Master channel:
  - INTTAUAnIm is generated at operation start (TAUAnCMORm.TAUAnMD0 = 1)
  - TAUAnCDRm = a = 5_H
- Slave channel:
  - TAUAnCDRm = 0_H

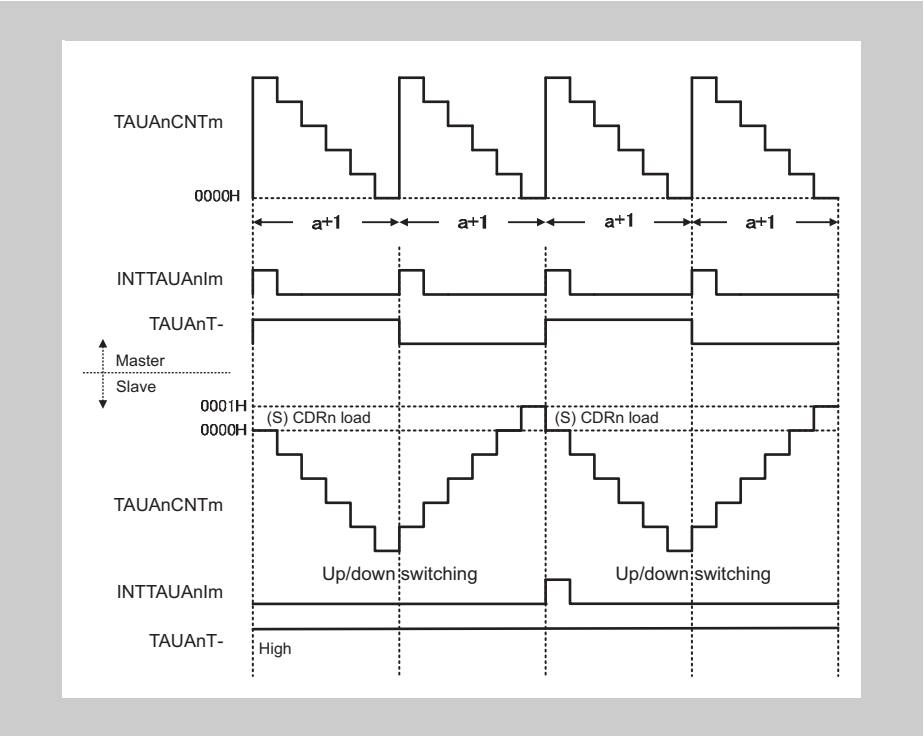


Figure 12-108 TAUAnCDRm (slave) = 0000_H

- If TAUAnCDRm (slave) =0000_H the counter of slave channel cannot reach 0001_H during *counting up*. The reset signal is never detected, so TAUAnTTOUTm remains at high state.

12.24.2 Triangle PWM Output Function with Dead Time

(1) Overview

Summary This function generates multiple triangle PWM outputs with a defined dead time by using a master and two or more slave channels. The resulting PWM signals are output via TAUAnTTOUTm of the slave channel 3. It enables the pulse cycle (frequency) and the duty cycle of TAUAnTTOUTm to be set using the master and slave channel(s) respectively.

Carrier cycles are generated on the master channel. The first pulse cycle controls the down status of the slave counter, and the second pulse cycle controls the up status.

An interrupt on slave 2 causes TAUAnTTOUTm of the slave channels to be set or reset. Depending on the settings of TAUAnTDL.TAUAnTDLm, delay time is added to positive or negative logic side of the signal (i.e. whether TAUAnTTOUTm is set/reset immediately or after dead time has elapsed). The duration of the dead time is specified by slave channel 3.

Prerequisites

- Three channels. For slave channels 2 and 3, select the even channel CH (a) and the odd channel CH (a + 1).
- The operation mode of the master channel must be set to Interval Timer Mode. Refer to *Table 12-160 “TAUAnCMORm settings for the master channel of the Triangle PWM Output Function with Dead Time” on page 792*.
- Slave channel 1 is not used for this function. This ensures that slave channel 2 is an odd channel, and slave channel 3 is an even channel.
- The operation mode of slave channel 2 must be set to Up Down Mode. Refer to *Table 12-164 “TAUAnCMORm settings for slave channel 2 of the Triangle PWM Output Function with Dead Time” on page 794*. Furthermore, slave channel 2 must be an even channel.
- The operation mode of slave channel 3 must be set to One Count Mode. Refer to *Table 12-168 “TAUAnCMORm settings for slave channel 3 of the Triangle PWM Output Function with Dead Time” on page 796*. Furthermore, slave channel 3 must be an odd channel.
- The channel output mode of the master channel must be set to Independent Channel Output Mode 1 (refer to *12.8 “Channel Output Modes” on page 577*).
- The channel output mode of the slave channels 2 and 3 must be set to Synchronous Channel Output Mode 2 with Dead Time Output (refer to *12.8 “Channel Output Modes” on page 577*).
- The following settings establish TAUAnTTOUTm at high level for the down status of the carrier cycle.
  - If the TAUAnCMORm.TAUAnMD0 (master) bit is set to 0, TAUAnTO.TAUAnTOm must be set to 1 while TAUAnTOE.TAUAnTOEm is 0 (recommended).
  - If the TAUAnCMORm.TAUAnMD0 (master) bit is set to 1, TAUAnTO.TAUAnTOm must be set to 0 while TAUAnTOE.TAUAnTOEm is 0.

Note Slave channel 1 is not used for Triangle PWM Output Function with Dead Time.

Description

The counters are started by setting the channel trigger bits (TAUAnTS.TAUAnTSm) to 1. This in turn sets TAUAnTE.TAUAnTEm, enabling count operation. The current values of TAUAnCDRm is loaded to TAUAnCNTm and the counters start to count down from these values. If the master channel TAUAnCMORm.TAUAnMD0 bit is set, an interrupt is generated and TAUAnTTOUTm signal of the master toggles.

- Master channel:

  When the counter of the master channel reaches 0000_H, INTTAUAnIm is generated and the TAUAnTTOUTm signal toggles. The counter reloads the TAUAnCDRm value, and then counts down.

- Slave channel 2:

  The INTTAUAnIm of the master channel triggers the counter of the slave channel 2:

  – If the slave counter currently counts down, it changes count direction.

  – If the slave counter currently counts up, the value of TAUAnCDRm is reloaded and the counter counts down.

  The counter continues to count down or up and awaits the next INTTAUAnIm of the master channel.

- Slave channel 3:

  INTTAUAnIm of slave channel 2 triggers the counter of slave channel 3. The current value of TAUAnCDRm (slave 3) is loaded to TAUAnCNTm (slave 3) and the counter starts to count down from this value.

  When the counter reaches 0000_H, INTTAUAnIm is generated. The counter returns to $FFFF_H$ and awaits the next INTTAUAnIm of slave channel 2.

The TAUAnTDL.TAUAnTDLm settings of the corresponding channel specify whether it is set/reset immediately, or after dead time has elapsed, as shown in *Table 12-159 “Behavior of TAUAnTTOUTm when an interrupt occurs on slave channel 2” on page 789*.

The TAUAnTOL.TAUAnTOLm settings specify whether set corresponds to a high signal (TAUAnTOL.TAUAnTOLm = 0) or a low signal (TAUAnTOL.TAUAnTOLm = 1).

The counter can be stopped by setting TAUAnTT.TAUAnTTm to 1 for the master and slave channel(s), which in turn sets TAUAnTE.TAUAnTEm to 0. TAUAnCNTm and TAUAnTTOUTm of master and slave channel(s) stop but retain their values.

TAUAnCDRm value of slave channel 2 can be set to 0000_H to output 100 % TAUAnTTOUTm.

Conditions Simultaneous rewrite can be used with this function. Please refer to *12.7 "Simultaneous Rewrite" on page 565*

TAUAnTOL.TAUAnTOLm and TAUAnTDL.TAUAnTDLm bits should be set before the counter starts, and slave channels 2 and 3 should have opposite TAUAnTOL.TAUAnTOLm settings or opposite TAUAnTDL.TAUAnTDLm settings.

Table 12-159 Behavior of TAUAnTTOUTm when an interrupt occurs on slave channel 2

TAUAnTDL. TAUAnTDLm	Count direction of slave channel 2 when interrupt is generated	TAUAnTTOUTm set/reset timing
0	Down	Set after dead time has elapsed
	Up	Reset immediately
1	Down	Set immediately
	Up	Reset after dead time has elapsed

(2) Equations

Pulse cycle = (TAUAnCDRm (master) + 1) x count clock cycle

$0000_H \leq$ TAUAnCDR (master) $< FFFF_H$

Carrier cycle (down/up) = (TAUAnCDRm (master) + 1) × 2 × count clock cycle

PWM signal width (positive phase) = [((TAUAnCDRm (master) + 1 - TAUAnCDRm (slave 2) × 2) - (TAUAnCDRm (slave 3) + 1)] × count clock cycle

PWM signal width (negative phase) = [((TAUAnCDRm (master) + 1 - TAUAnCDRm (slave 2) × 2) + (TAUAnCDRm (slave 3) + 1)] × count clock cycle

(3) Block diagram and general timing diagram

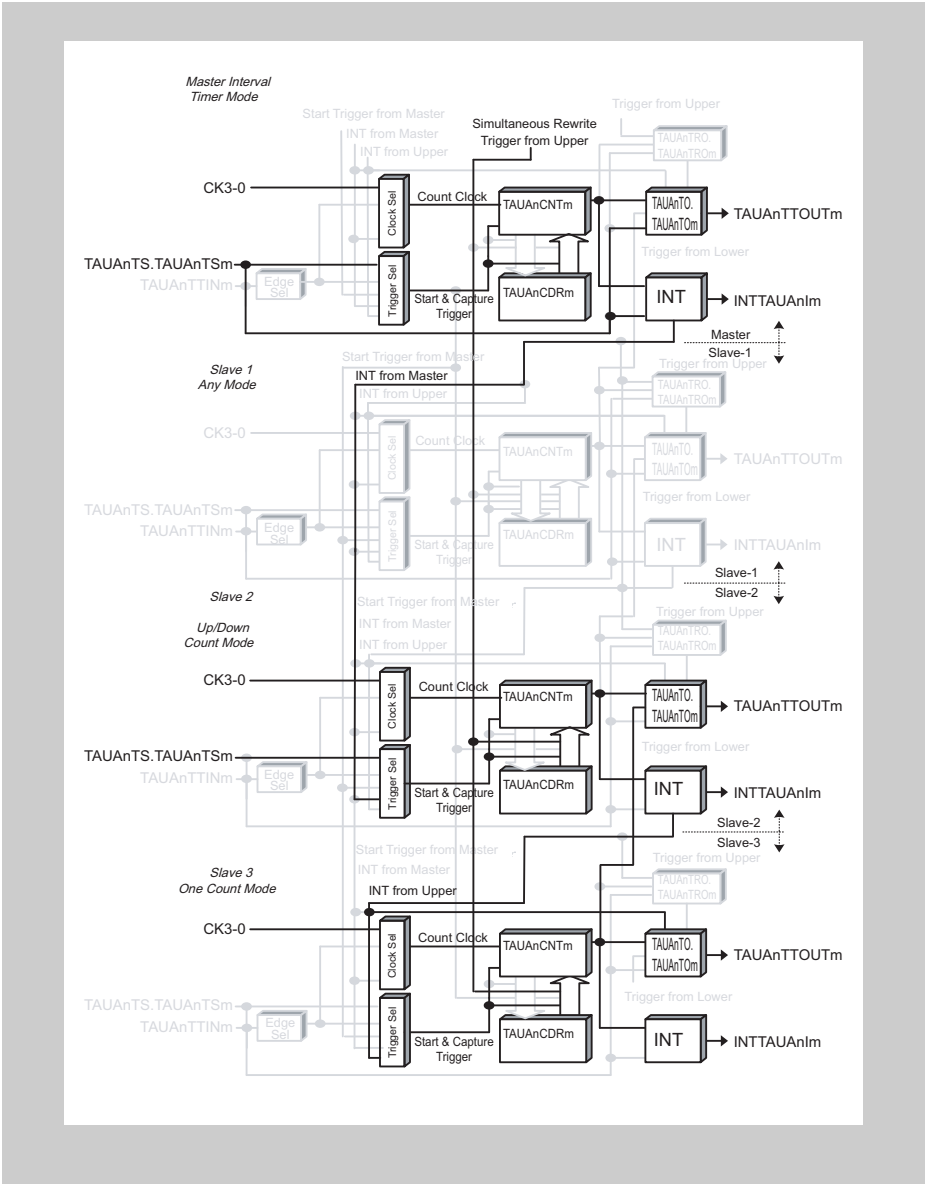


Figure 12-109 Block diagram for Triangle PWM Output Function with Dead Time

The following settings apply to the general timing diagram:

- Master channel:
  - INTTAUAnIm is generated at operation start (TAUAnCMORm.TAUAnMD0 = 1)
- Slave channel 2:
  - INTTAUAnIm not generated at operation start (TAUAnCMORm.TAUAnMD0 = 0)
  - TAUAnTDL.TAUAnTDLm = 0
  - Positive logic (TAUAnTOL.TAUAnTOLm = 0)
- Slave channel 3:
  - INTTAUAnIm is generated at operation start (TAUAnCMORm.TAUAnMD0 = 1)
  - TAUAnTDL.TAUAnTDLm = 1
  - Negative logic (TAUAnTOL.TAUAnTOLm = 1)

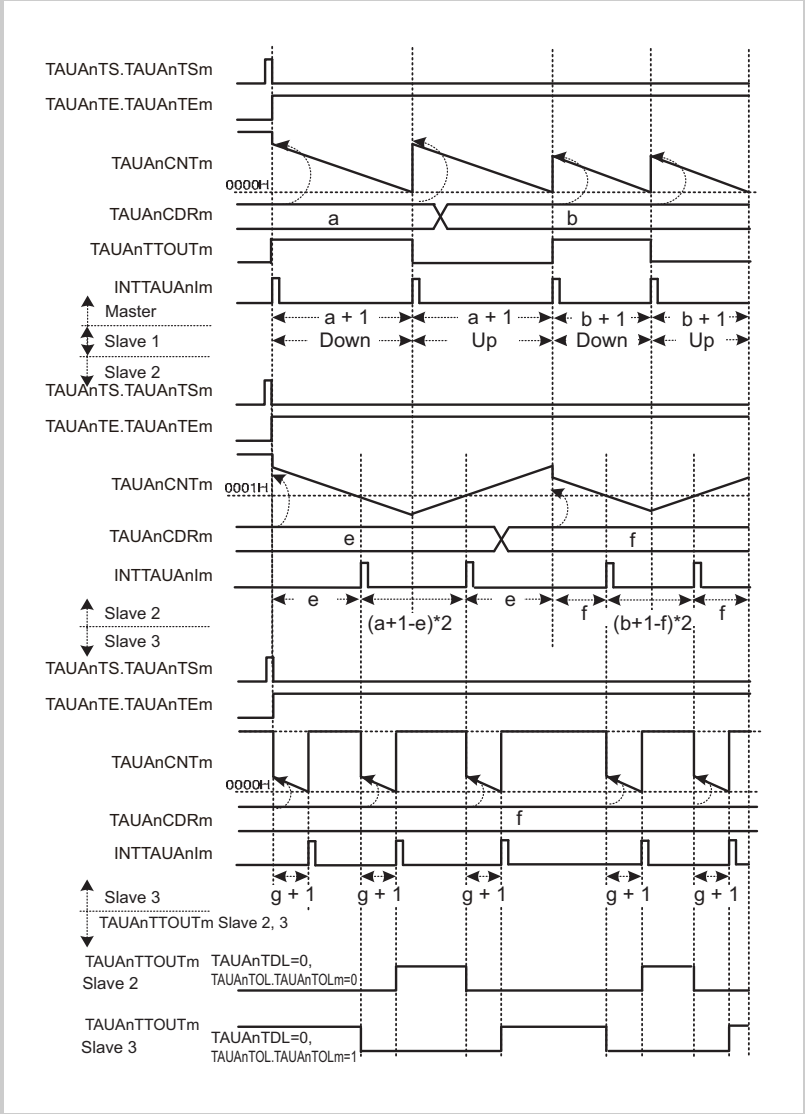


Figure 12-110 General timing diagram for Triangle PWM Output Function with Dead Time

(4) Register settings for the master channel

(a) TAUAnCMORm for the master channel

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn CKS[1:0]		TAUAn CCS[1:0]		TAUAn MAS	TAUAn STS[2:0]			TAUAn COS[1:0]		-	TAUAn MD[4:1]				TAUAn MD0

Table 12-160 TAUAnCMORm settings for the master channel of the Triangle PWM Output Function with Dead Time

Bit name	Setting
TAUAnCKS[1:0]	00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 The value of the TAUAnCKS[1:0] bit of the master and slave channel(s) must be identical.
TAUAnCCS[1:0]	00: Operation clock is used as the count clock
TAUAnMAS	1: Channel is master channel
TAUAnSTS[2:0]	000: Counter triggered by software trigger
TAUAnCOS[1:0]	00: Not used, so set to 00
TAUAnMD[4:1]	0000: Interval Timer Mode
TAUAnMD0	0: INTTAUAnIm not generated and TAUAnTTOUTm does not toggle at operation start 1: Generates INTTAUAnIm and toggles TAUAnTTOUTm at operation start

(b) TAUAnCMURm for the master channel

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-														TAUAnTIS[1:0]	

Table 12-161 TAUAnCMURm settings for the master channel of the Triangle PWM Output Function with Dead Time

Bit name	Setting
TAUAnTIS[1:0]	00: These are not used, so set them to 00.

(c) Channel output mode for the master channel

Table 12-162 Control bit settings for Independent Channel Output Mode 1

Bit name	Setting
TAUAnTOE.TAUAnTOEm	1: Enables Independent Channel Output Mode
TAUAnTOM.TAUAnTOMm	0: Independent channel output
TAUAnTOC.TAUAnTOCm	0: Operation mode 1 (= Toggle mode if TAUAnTOM.TAUAnTOMm = 0)
TAUAnTOL.TAUAnTOLm	0: Positive logic
TAUAnTDE.TAUAnTDEm	0: Disables dead time operation
TAUAnTDM.TAUAnTDMm	0: When dead time operation is disabled (TAUAnTDE.TAUAnTDEm = 0), set these bits to 0
TAUAnTDL.TAUAnTDLm	
TAUAnTRE.TAUAnTREm	0: Disables real-time output
TAUAnTRO.TAUAnTROm	0: When real-time output is disabled (TAUAnTRE.TAUAnTREm = 0), set these bits to 0
TAUAnTRC.TAUAnTRCm	
TAUAnTME.TAUAnTMEm	0: Disables modulation

(d) Simultaneous rewrite for the master channel

The simultaneous rewrite settings of the master and slave channels must be identical.

Table 12-163 Simultaneous rewrite settings for the master channel of the Triangle PWM Output Function with Dead Time

Bit name	Setting
TAUAnRDE.TAUAnRDEm	1: Enables simultaneous rewrite
TAUAnRDS.TAUAnRDSm	0: The master channel is monitored for the simultaneous rewrite trigger 1: An upper channel outside the channel group is monitored for the simultaneous rewrite trigger
TAUAnRDM.TAUAnRDMm	1: The simultaneous rewrite trigger signal is generated when the master channel starts counting and the corresponding slave channel is at the peak of a triangular wave
TAUAnRDC.TAUAnRDCm	0: Channel is not monitored for an INTTAUAnIm signal that is used as the simultaneous rewrite trigger. If TAUAnRDS.TAUAnRDSm = 0, the master channel is monitored for the simultaneous rewrite trigger regardless of the value of this bit.

Note If the TAUAnRDS.TAUAnRDSm bit is 1, there must be a channel higher than the master channel to which a simultaneous rewrite trigger signal is generated.

(5) Register settings for slave channel 2

(a) TAUAnCMORm for slave channel 2

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn CKS[1:0]		TAUAn CCS[1:0]		TAUAn MAS	TAUAnSTS[2:0]			TAUAn COS[1:0]		-	TAUAnMD[4:1]				TAUAn MD0

Table 12-164 TAUAnCMORm settings for slave channel 2 of the Triangle PWM Output Function with Dead Time

Bit name	Setting
TAUAnCKS[1:0]	00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 The value of the TAUAnCKS[1:0] bit of the master and slave channel(s) must be identical.
TAUAnCCS[1:0]	00: Operation clock is used as the count clock
TAUAnMAS	0: Channel is a slave channel
TAUAnSTS[2:0]	111: The up/down output trigger signal of the master channel
TAUAnCOS[1:0]	00: Not used, so set to 00
TAUAnMD[4:1]	1001: Up Down Count Mode
TAUAnMD0	0: INTTAUAnIm not generated at operation start

(b) TAUAnCMURm for slave channel 2

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-														TAUAnTIS[1:0]	

Table 12-165 TAUAnCMURm settings for slave channel 2 of the Triangle PWM Output Function with Dead Time

Bit name	Setting
TAUAnTIS[1:0]	00: These are not used, so set them to 00.

(c) Channel output mode for slave channel 2

Table 12-166 Control bit settings for Synchronous channel Output Mode 2 with Dead Time Output

Bit name	Setting
TAUAnTOE.TAUAnTOEm	1: Enables Independent Channel Output Mode
TAUAnTOM.TAUAnTOMm	1: Synchronous channel operation
TAUAnTOC.TAUAnTOCm	1: Operation mode 2
TAUAnTOL.TAUAnTOLm	0: Positive logic 1: Inverted logic
TAUAnTDE.TAUAnTDEm	1: Enables dead time operation
TAUAnTDM.TAUAnTDMm	0: Dead time is added upon detection of an interrupt on the upper even channel, if the TAUAnTDL.TAUAnTDLm condition is also fulfilled.
TAUAnTDL.TAUAnTDLm	0: Dead time is added to the positive phase 1: Dead time is added to the negative phase
TAUAnTRE.TAUAnTREm	0: Disables real-time output
TAUAnTRO.TAUAnTROm	0: When real-time output is disabled (TAUAnTRE.TAUAnTREm = 0), set these bits to 0
TAUAnTRC.TAUAnTRCm	
TAUAnTME.TAUAnTMEm	0: Disables modulation

Caution For TAUAnTDL.TAUAnTDLm, specify the setting that is opposite that of the odd channel.

(d) Simultaneous rewrite for slave channel 2

The simultaneous rewrite settings of the master and slave channels must be identical.

Table 12-167 Simultaneous rewrite settings for slave channel 2 of the Triangle PWM Output Function

Bit name	Setting
TAUAnRDE.TAUAnRDEm	1: Enables simultaneous rewrite
TAUAnRDS.TAUAnRDSm	0: The master channel is monitored for the simultaneous rewrite trigger 1: An upper channel outside the channel group is monitored for the simultaneous rewrite trigger
TAUAnRDM.TAUAnRDMm	1: The simultaneous rewrite trigger signal is generated when the master channel starts counting and the corresponding slave channel is at the peak of a triangular wave
TAUAnRDC.TAUAnRDCm	0: Channel is not monitored for an INTTAUAnIm signal that is used as the simultaneous rewrite trigger. If TAUAnRDS.TAUAnRDSm = 0, the master channel is monitored for the simultaneous rewrite trigger regardless of the value of this bit.

(6) Register settings for slave channel 3

(a) TAUAnCMORm for slave channel 3

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn CKS[1:0]		TAUAn CCS[1:0]		TAUAn MAS	TAUAnSTS[2:0]			TAUAn COS[1:0]		-	TAUAnMD[4:1]				TAUAn MD0

Table 12-168 TAUAnCMORm settings for slave channel 3 of the Triangle PWM Output Function with Dead Time

Bit name	Setting
TAUAnCKS[1:0]	00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 The value of the TAUAnCKS[1:0] bit of the master and slave channel(s) must be identical.
TAUAnCCS[1:0]	00: Operation clock is used as the count clock
TAUAnMAS	0: Channel is a slave channel
TAUAnSTS[2:0]	110: Dead time trigger
TAUAnCOS[1:0]	00: Not used, so set to 00
TAUAnMD[4:1]	0100: One Count Mode
TAUAnMD0	1: Enables start trigger detection during counting

(b) TAUAnCMURm for slave channel 3

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-														TAUAnTIS[1:0]	

Table 12-169 TAUAnCMURm settings for slave channel 3 of the Triangle PWM Output Function with Dead Time

Bit name	Setting
TAUAnTIS[1:0]	00: These are not used, so set them to 00.

(c) Channel output mode for slave channel 3

Table 12-170 Control bit settings for Synchronous channel Output Mode 2 with Dead Time Output

Bit name	Setting
TAUAnTOE.TAUAnTOEm	1: Enables Independent Channel Output Mode
TAUAnTOM.TAUAnTOMm	1: Synchronous channel operation
TAUAnTOC.TAUAnTOCm	1: Operation mode 2
TAUAnTOL.TAUAnTOLm	0: Positive logic 1: Inverted logic
TAUAnTDE.TAUAnTDEm	1: Enables dead time operation
TAUAnTDM.TAUAnTDMm	0: Dead time is added upon detection of an interrupt on the upper even channel, if the TAUAnTDL.TAUAnTDLm condition is also fulfilled.
TAUAnTDL.TAUAnTDLm	0: Dead time is added to the positive phase 1: Dead time is added to the negative phase
TAUAnTRE.TAUAnTREm	0: Disables real-time output
TAUAnTRO.TAUAnTROm	0: When real-time output is disabled (TAUAnTRE.TAUAnTREm = 0), set these bits to 0
TAUAnTRC.TAUAnTRCm	
TAUAnTME.TAUAnTMEm	0: Disables modulation

Caution For TAUAnTDL.TAUAnTDLm, specify the setting that is opposite that of the even channel.

(d) Simultaneous rewrite for slave channel 3

The simultaneous rewrite settings of the master and slave channel must be identical.

Table 12-171 Simultaneous rewrite settings for slave channel 3 of the Triangle PWM Output Function

Bit name	Setting
TAUAnRDE.TAUAnRDEm	1: Enables simultaneous rewrite
TAUAnRDS.TAUAnRDSm	0: The master channel is monitored for the simultaneous rewrite trigger 1: An upper channel outside the channel group is monitored for the simultaneous rewrite trigger
TAUAnRDM.TAUAnRDMm	1: The simultaneous rewrite trigger signal is generated when the master channel starts counting and the corresponding slave channel is at the peak of a triangular wave
TAUAnRDC.TAUAnRDCm	0: Channel is not monitored for an INTTAUAnIm signal that is used as the simultaneous rewrite trigger. If TAUAnRDS.TAUAnRDSm = 0, the master channel is monitored for the simultaneous rewrite trigger regardless of the value of this bit.

(7) Operating procedure for Triangle PWM Output Function with Dead Time

Table 12-172 Operating procedure for Triangle PWM Output Function with Dead Time

	Operation	Status of TAUAn
Initial channel setting	Master channel: set the TAUAnCMORm and TAUAnCMURm registers and the channel output mode as described in *(4) "Register settings for the master channel" on page 792* Slave channel 2: set the TAUAnCMORm and TAUAnCMURm registers and the channel output mode as described in *(5) "Register settings for slave channel 2" on page 794* Slave channel 3: set the TAUAnCMORm and TAUAnCMURm registers and the channel output mode as described in *(6) "Register settings for slave channel 3" on page 796* Set the values of the TAUAnCDRm registers of all channels	Channel operation is stopped.
Start operation	Set TAUAnTS.TAUAnTSm of the master and slave channels to 1 simultaneously. TAUAnTS.TAUAnTSm is a trigger bit, so it is automatically cleared to 0.	TAUAnTE.TAUAnTEm (master and slave channels) is set to 1 and the counters of the master and slave channels start. INTTAUAnIm is generated on the master channel, if TAUAnCMORm.TAUAnMD0 of the the master channel is set.
During operation	TAUAnCDRm can be changed at any time. TAUAnCNTm and TAUAnRSF.TAUAnRSFm can be read at any time. TAUAnRDT.TAUAnRDTm can be changed during operation.	TAUAnCNTm loads the TAUAnCDRm value of the master channel and slave channel 2, and then counts down. When the counter of the master channel reaches 0000_H: • INTTAUAnIm (master) is generated. • TAUAnCNTm (master) reloads the value of TAUAnCDRm, and then continues counting. • TAUAnCNTm (slave 2) reloads the value of TAUAnCDRm, or starts counting in the opposite direction. When TAUAnCNTm (slave 2) reaches 0001_H: • INTTAUAnIm (slave 2) is generated. • TAUAnCNTm loads the TAUAnCDRm value of slave channel 3, and then counts down. When TAUAnCNTm of slave channel 3 = 0000_H: • INTTAUAnIm is generated.
Stop operation	Set TAUAnTT.TAUAnTTm of the master and slave channels to 1 simultaneously. TAUAnTT.TAUAnTTm is a trigger bit, so it is automatically cleared to 0.	TAUAnTE.TAUAnTEm is cleared to 0 and the counter stops. TAUAnCNTm and TAUAnTTOUTm stop and retain their current values.

Restart →

(8) Specific timing diagrams

(a) Duty cycle = 0%

The following settings apply to the diagram below:

- Slave channel 2:
  - Positive logic (TAUAnTOL.TAUAnTOLm = 0)
- Slave channel 3:
  - Negative logic (TAUAnTOL.TAUAnTOLm = 1)

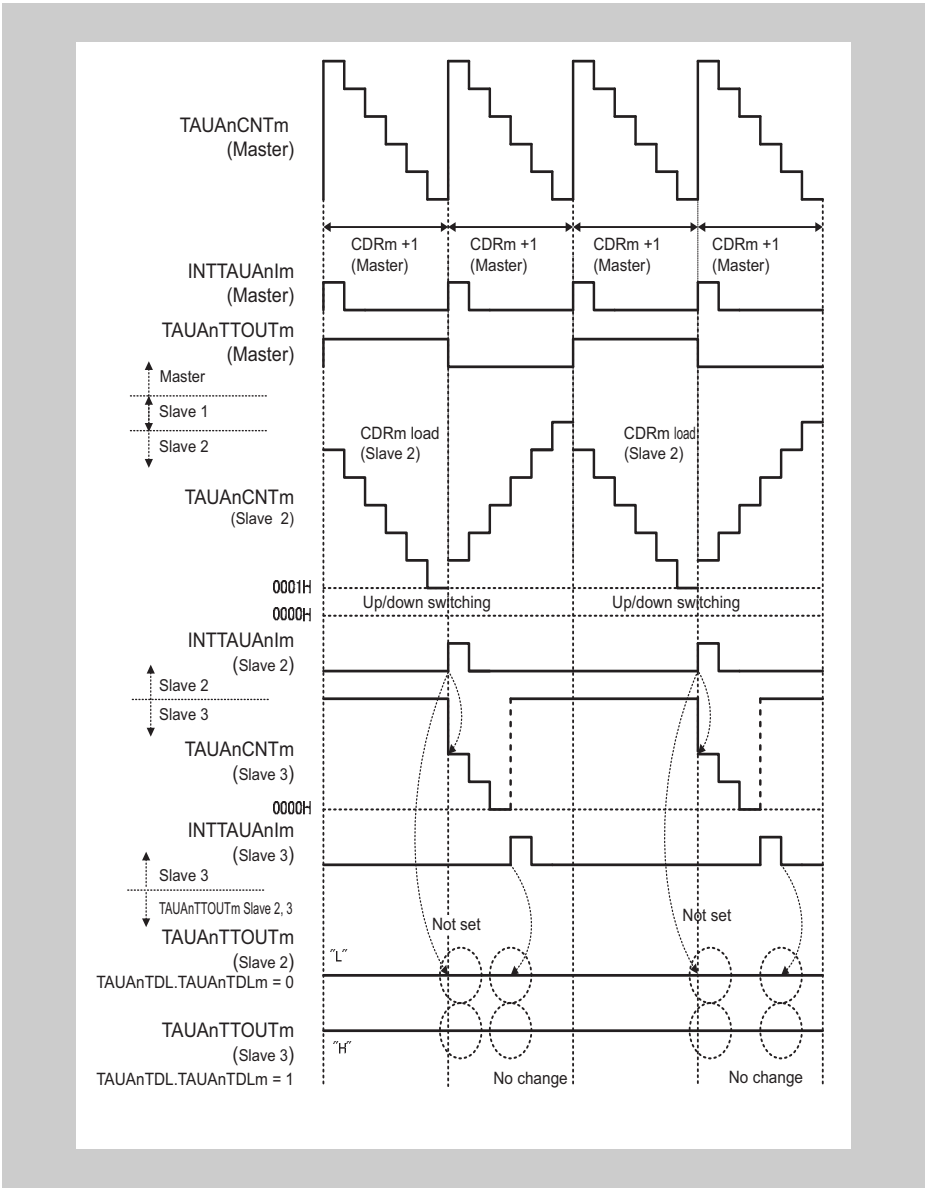


Figure 12-111 TAUAnCDRm (slave) ≥ TAUAnCDRm (master) + 1

- If TAUAnCDRm (slave 2) ≥ TAUAnCDRm (master) the counter of slave channel cannot reach 0000_H during counting down. Therefore TAUAnTTOUTm cannot be set or reset, and it remains at its initial state. The interrupt from slave channel 2 occurs during count up, therefore it is a reset signal.

(b) Duty cycle = 100%

The following settings apply to the diagram below:

- Slave channels 2:
  - Positive logic (TAUAnTOL.TAUAnTOLm = 0)
- Slave channels 3:
  - Negative logic (TAUAnTOL.TAUAnTOLm = 1)

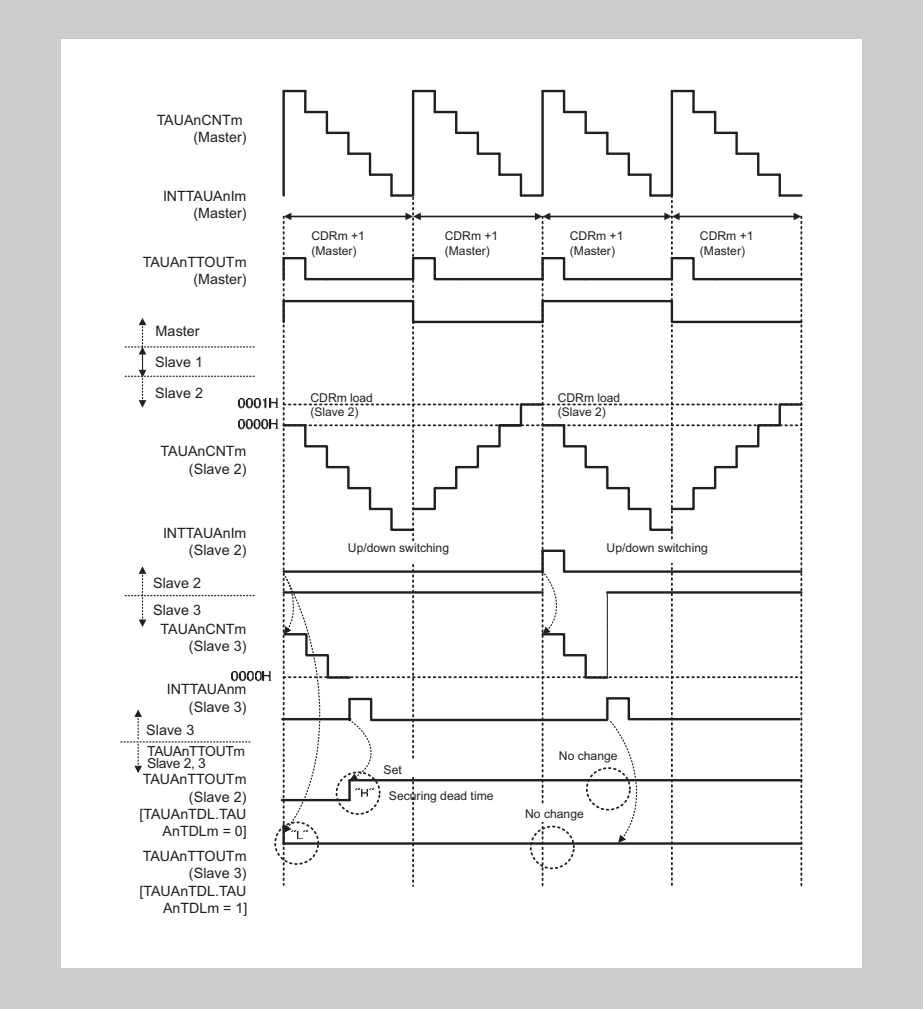


Figure 12-112 TAUAnCDRm (slave) = 0000_H

- If TAUAnCDRm (slave 2) =0000_H the counter of slave channel cannot reach 0001_H while counting up and therefore no INTTAUAnIm is generated while counting up.
  - The set conditions for a channel in which TAUAnTDL.TAUAnTDLm = 0 are met after dead time has elapsed. TAUAnTTOUTm is set/reset but remains in the new state because the reset conditions never occur for such a channel.
  - Slave channel 3 in the diagram above is set when the counter starts. However, the reset conditions for a channel in which TAUAnTDL.TAUAnTDLm = 1 never occur so TAUAnTTOUTm remains in its initial state for such a slave channel.

12.24.3 AD Conversion Trigger Output Function Type 2

(1) Overview

Summary This function is identical to *12.24.1 “Triangle PWM Output Function” on page 776*except that TAUAnTTOUTm is not output.

This is achieved by setting the channel output mode of the slave to Independent Channel Output Mode Controlled by Software.

(2) Block diagram and general timing diagram

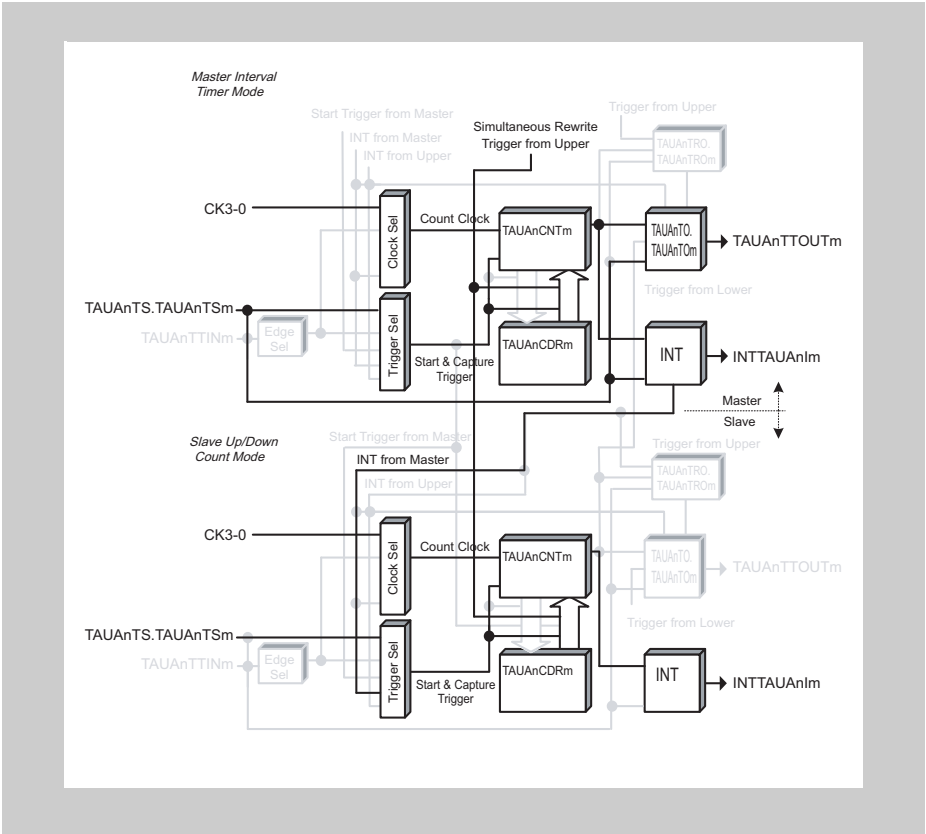


Figure 12-113 Block diagram for AD Conversion Trigger Output Function Type 2

The following settings apply to the general timing diagram:

- Master channel
  - INTTAUAnIm is generated at operation start (TAUAnCMORm.TAUAnMD0 = 1)

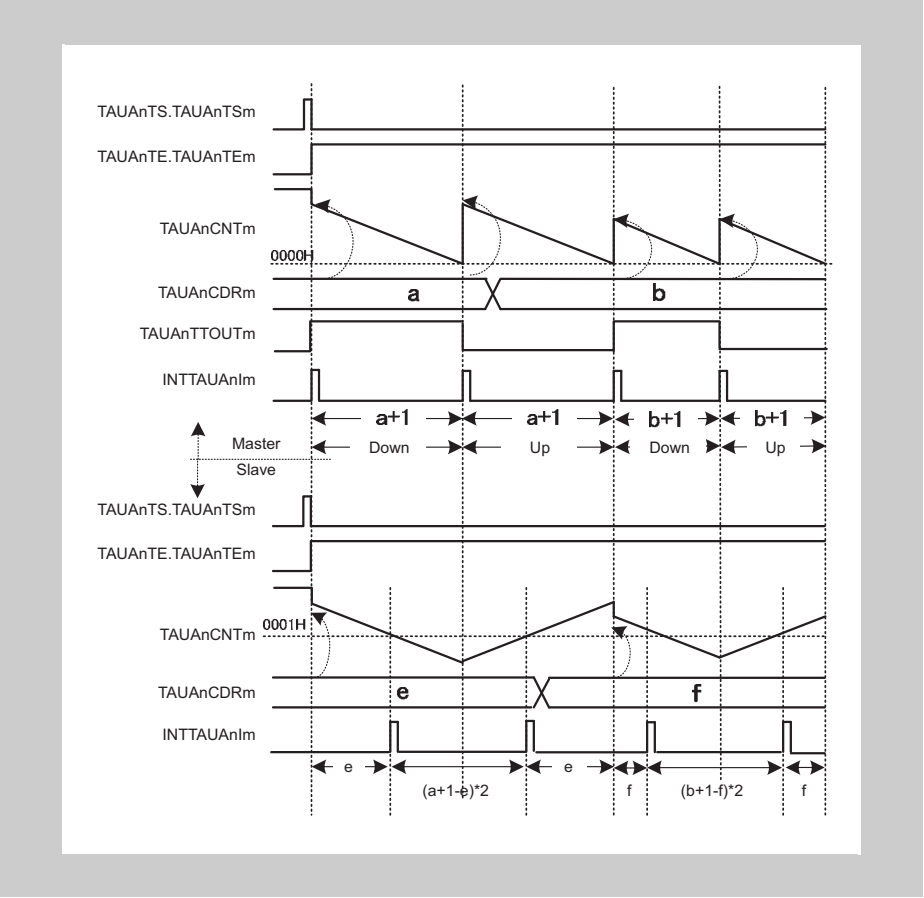


Figure 12-114 General timing diagram for AD Conversion Trigger Output Function Type 2

12.25 Synchronous Non-Complementary and Complementary Functions

This chapter describes functions that generate 6-phase triangle PWM using a master channel and 7 slaves.

- *12.25.1 “Non-Complementary Modulation Output Function Type 1”*
- *12.25.2 “Non-Complementary Modulation Output Function Type 2”*
- *12.25.3 “Complementary Modulation Output Function”*

12.25.1 Non-Complementary Modulation Output Function Type 1

(1) Overview

Summary This function outputs a PWM signal, a high signal, or a low signal from TAUAnTTOUTm depending on the value of the real-time output bits (TAUAnTRO.TAUAnTROm) and the modulation output enable bits (TAUAnTME.TAUAnTMEm) of a pair of slave channels. Three pairs of channels are typically used.

Prerequisites

- One master channel and seven slave channels
- The operation mode of the master channel must be set to Interval Timer Mode. Refer to *Table 12-174 "TAUAnCMORm settings for the master channel of Non-Complementary Modulation Output Function Type 1" on page 808*.
- The operation mode of slave channels 1 to 7 must be set to One Count Mode. Refer to *Table 12-177 "TAUAnCMORm settings for slave channel 1 of Non-Complementary Modulation Output Function Type 1" on page 810*.
- TAUAnTTOUTm is not used for the master channel of this function.
- TAUAnTTOUTm of slave channel 1 is not used for this function, but TAUAnTRC.TAUAnTRCm must be set to 1. Refer to *12.8 "Channel Output Modes" on page 577*.
- The channel output mode of slave channels 2 to 7 must be set to Simultaneous Channel Output Mode 1 with Non-Complementary Modulation Output. Refer to *12.8 "Channel Output Modes" on page 577*.
- TAUAnCDRm of slave channel 1 must be set to 0000_H.

Description The counters of the master and slave channels are started by setting the channel trigger bit (TAUAnTS.TAUAnTSm) to 1. This in turn sets TAUAnTE.TAUAnTEm, enabling count operation. The values of the data registers (TAUAnCDRm) are loaded to the counters (TAUAnCNTm) and the counters start to count down. When the counters reach 0000_H, INTTAUAnIm is generated.

- Slave channel 1:

  Because slave channel 1 is specified as the trigger channel for real-time output (TAUAnTRC.TAUAnTRCm = 1), when an interrupt is generated on slave channel 1 (for which TAUAnCDRm is fixed to 0), the real-time output bit (TAUAnTRO.TAUAnTROm) value of the channel that detects the generated interrupt of the corresponding channel is changed. After generating an interrupt, the counter returns to $FFFF_H$ and awaits the next interrupt from the master channel.

- Slave channel 2:

  Slave channel 2 generates a PWM output. The PWM output cycle is specified for the master channel, and the duty cycle is specified for slave channel 2. After generating an interrupt, the counter returns to $FFFF_H$ and awaits the next interrupt from the master channel.

Slave channels 3 to 7 behave analogously to slave channel 2.

The signal that is output from TAUAnTTOUTm depends on the value of the real-time output bit (TAUAnTRO.TAUAnTROm) and the modulation output bit (TAUAnTME.TAUAnTMEm) of the slave channel, as shown in *Table 12-173 "TAUAnTTOUTm output of a pair of slave channels in Non-Complementary Modulation Output Function Type 1" on page 805*.

Forced restart is not possible with this function. The counter can be stopped by setting TAUAnTT.TAUAnTTm to 1 for the master and slave channels, which in turn sets TAUAnTE.TAUAnTEm to 0. TAUAnCNTm and TAUAnTTOUTm of master and slave channels stop but retain their values. The counters can be restarted by setting TAUAnTS.TAUAnTSm to 1.

Conditions

- If TAUAnTME.TAUAnTMEm = 0 for slave channels 2 to 7:
  - If TAUAnTRO.TAUAnTROm of the channel is 1, TAUAnTTOUTm outputs a high signal
  - If TAUAnTRO.TAUAnTROm of the channel is 0, TAUAnTTOUTm outputs a low signal
- If TAUAnTME.TAUAnTMEm = 1 for slave channels 2 to 7:
  - If TAUAnTRO.TAUAnTROm of the channel is 1, TAUAnTTOUTm outputs the corresponding PWM of the channel
  - If TAUAnTRO.TAUAnTROm of the channel is 0, TAUAnTTOUTm outputs a low signal
- If TAUAnTOL.TAUAnTOLm = 1 the high and low signals output from TAUAnTTOUTm are inverted. The PWM signals remain unaffected. Only the initial setting can be used for TAUAnTOL.TAUAnTOLm (and this setting cannot be changed during operation).

Table 12-173 TAUAnTTOUTm output of a pair of slave channels in Non-Complementary Modulation Output Function Type 1

TAUAnTME.TAUAnTMEm	TAUAnTRO.TAUAnTROm	TAUAnTTOUTm outputs
0	0	Low
	1	High
1	0	Low
	1	PWMm

- Simultaneous rewrite can be used with this function. Please refer to *12.7 "Simultaneous Rewrite" on page 565*
- The value of TAUAnCDRm of slave channel 1 must be set to 0000_H so that the real-time output trigger occurs at the same time as the PWM is generated by slave channels 2 to 7.
- If TAUAnTOL.TAUAnTOLm is cleared to 0 on slave channels 2 to 7, TAUAnTO.TAUAnTOm is set to the low level before clearing TAUAnTE.TAUAnTEm to 0.
- If TAUAnTOL.TAUAnTOLm is set to 1 on slave channels 2 to 7, TAUAnTO.TAUAnTOm is set to the high level before clearing TAUAnTE.TAUAnTEm to 0.

(2) Equations

For slave channels 2 to 7:

PWM output cycle time = [TAUAnCDRm (master) + 1] x count clock

PWM output duty time = [TAUAnCDRm (slave)] x count clock

(3) Block diagram and general timing diagram

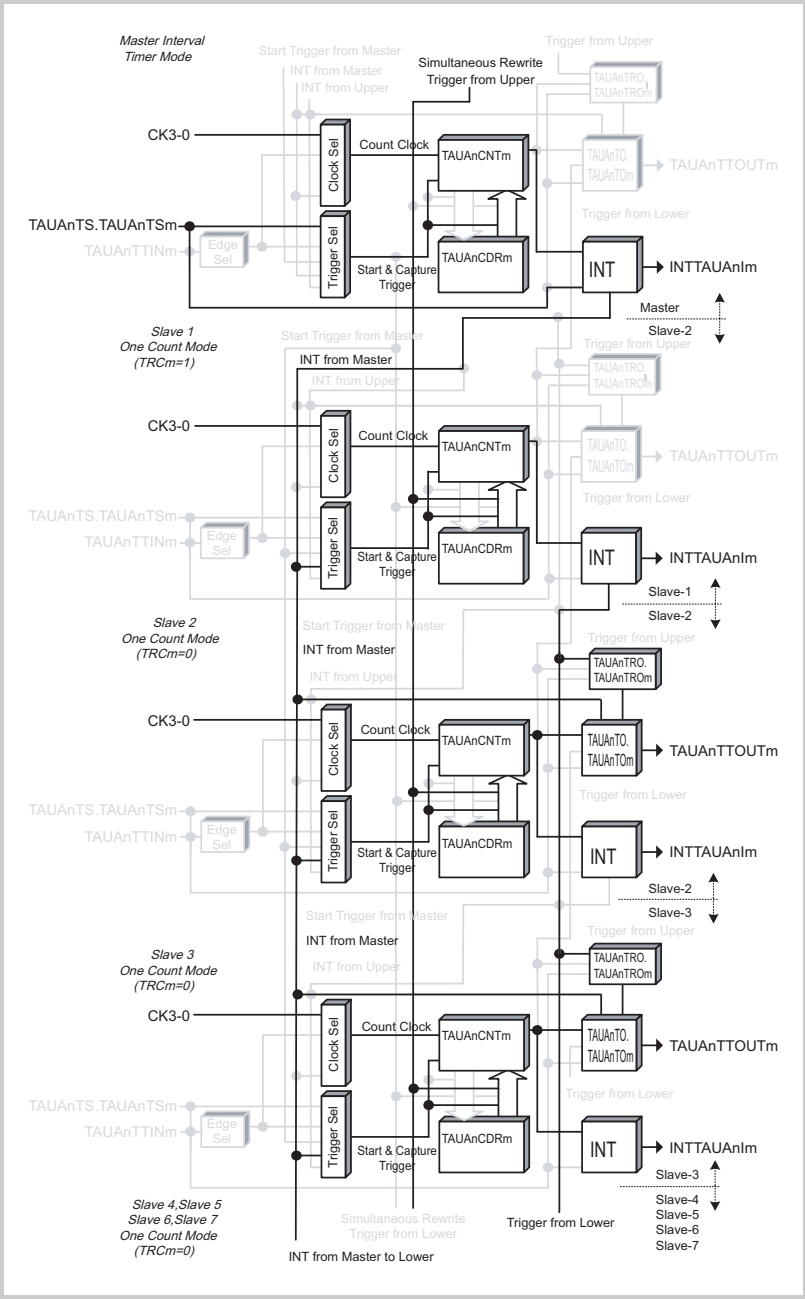


Figure 12-115 Block diagram for Non-Complementary Modulation Output Function Type 1

The following settings apply to the general timing diagram:

- Slave channels 2 to 7: positive logic (TAUAnTOL.TAUAnTOLm = 0)

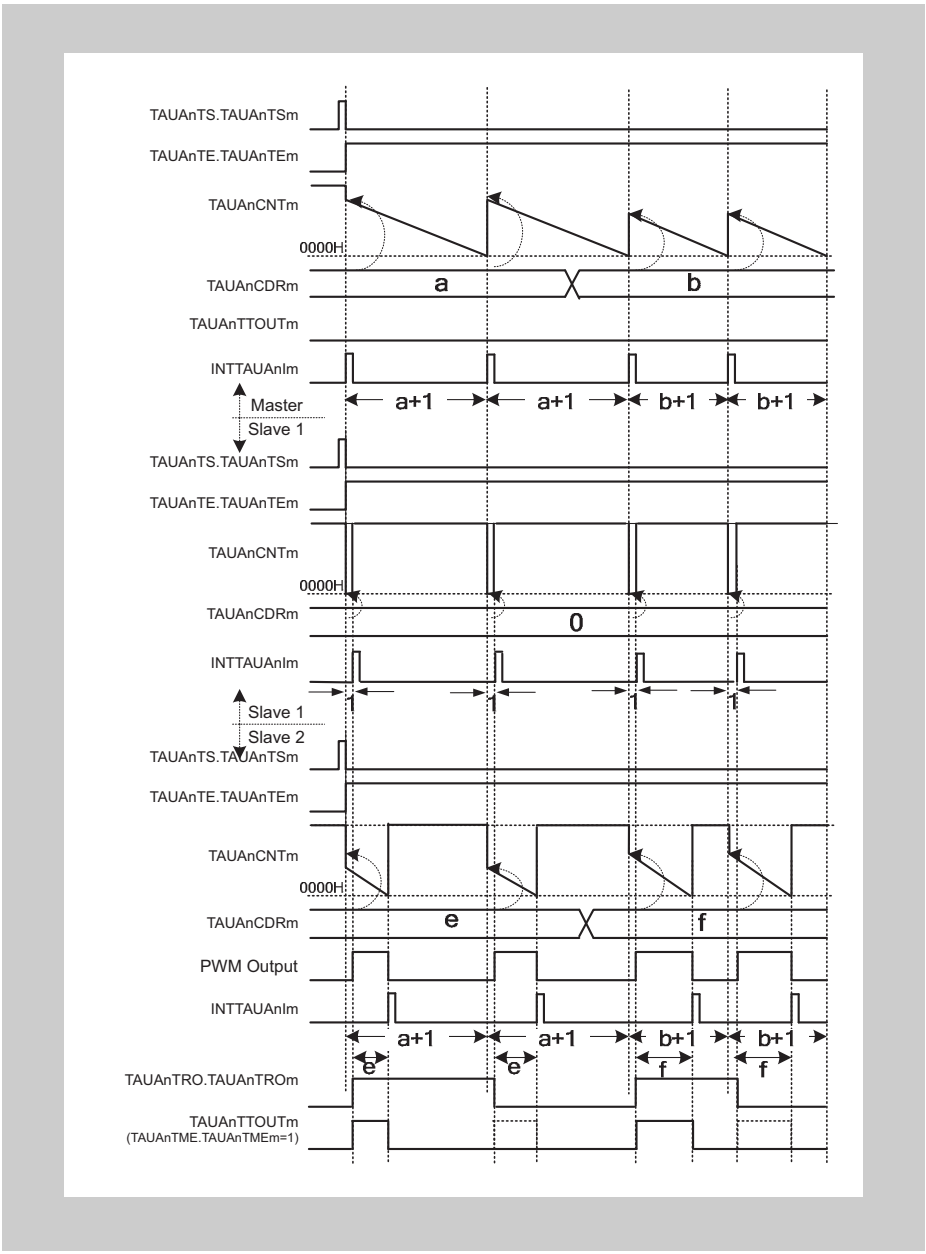


Figure 12-116 General timing diagram for Non-Complementary Modulation Output Function Type 1

(4) Register settings for the master channel

(a) TAUAnCMORm for the master channel

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn CKS[1:0]		TAUAn CCS[1:0]		TAUAn MAS	TAUAnSTS[2:0]			TAUAn COS[1:0]		-	TAUAnMD[4:1]				TAUAn MD0

Table 12-174 TAUAnCMORm settings for the master channel of Non-Complementary Modulation Output Function Type 1

Bit name	Setting
TAUAnCKS[1:0]	00: Operation clock = CK0 01: Operation clock = CK1 10: Operation clock = CK2 11: Operation clock = CK3 The value of the TAUAnCKS[1:0] bit of the master and slave channels must be identical.
TAUAnCCS[1:0]	00: Operation clock is used as the count clock
TAUAnMAS	1: Channel is master channel
TAUAnSTS[2:0]	000: Counter triggered by software trigger
TAUAnCOS[1:0]	00: Not used, so set to 00
TAUAnMD[4:1]	0000: Interval Timer Mode
TAUAnMD0	1: Generates INTTAUAnIm at operation start or restart

(b) TAUAnCMURm for the master channel

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-														TAUAnTIS[1:0]	

Table 12-175 TAUAnCMURm settings for the master channel of Non-Complementary Modulation Output Function Type 1

Bit name	Setting
TAUAnTIS[1:0]	00: Not used, so set to 00

(c) Channel output mode of the master channel

Because the channel output mode is not used by this function, clear TAUAnTOE.TAUAnTOEm. However, it can be used by other functions or in Independent Channel Output Mode Controlled by Software.

(d) Simultaneous rewrite for the master channel

The simultaneous rewrite settings of the master and slave channels must be identical.

Table 12-176 Simultaneous rewrite settings for the master channel of Non-Complementary Modulation Output Function Type 1

Bit name	Setting
TAUAnRDE.TAUAnRDEm	1: Enables simultaneous rewrite
TAUAnRDS.TAUAnRDSm	0: The master channel is monitored for the simultaneous rewrite trigger 1: An upper channel outside the channel group is monitored for the simultaneous rewrite trigger
TAUAnRDM.TAUAnRDMm	0: The simultaneous rewrite trigger signal is generated when the master channel starts counting
TAUAnRDC.TAUAnRDCm	0: Channel is not monitored for an INTTAUAnIm signal that is used as the simultaneous rewrite trigger. If TAUAnRDS.TAUAnRDSm = 0, the master channel is monitored for the simultaneous rewrite trigger regardless of the value of this bit.

Note If the TAUAnRDS.TAUAnRDSm bit is 1, there must be a channel higher than the master channel to which a simultaneous rewrite trigger signal is generated.

(5) Register settings for slave channel 1

(a) TAUAnCMORm for slave channel 1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn CKS[1:0]		TAUAn CCS[1:0]		TAUAn MAS	TAUAnSTS[2:0]			TAUAn COS[1:0]		-	TAUAnMD[4:1]				TAUAn MD0

Table 12-177 TAUAnCMORm settings for slave channel 1 of Non-Complementary Modulation Output Function Type 1

Bit name	Setting
TAUAnCKS[1:0]	00: Operation clock = CK0 01: Operation clock = CK1 10: Operation clock = CK2 11: Operation clock = CK3 The value of the TAUAnCKS[1:0] bit of the master and slave channels must be identical.
TAUAnCCS[1:0]	00: Operation clock is used as the count clock
TAUAnMAS	0: Channel is a slave channel
TAUAnSTS[2:0]	100: INTTAUAnIm of the master channel is the start trigger
TAUAnCOS[1:0]	00: Not used, so set to 00
TAUAnMD[4:1]	0100: One Count Mode
TAUAnMD0	1: Generates INTTAUAnIm and toggles TAUAnTTOUTm at operation start or restart

(b) TAUAnCMURm for slave channel 1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-														TAUAnTIS[1:0]	

Table 12-178 TAUAnCMURm settings for slave channel 1 Non-Complementary Modulation Output Function Type 1

Bit name	Setting
TAUAnTIS[1:0]	00: Not used, so set to 00

(c) Channel output mode

Because the channel output mode is not used by slave channel 1 of this function, clear TAUAnTOE.TAUAnTOEm. However, it can be used in Independent Channel Output Mode Controlled by Software.

Caution TAUAnTRC.TAUAnTRCm must be set to channel 1 to enable slave 1 to be used as the real-time output trigger.

(d) Simultaneous rewrite for slave channel 1

The simultaneous rewrite settings of the master and slave channels must be identical.

Table 12-179 Simultaneous rewrite settings for slave channel 1 Non-Complementary Modulation Output Function Type 1

Bit name	Setting
TAUAnRDE.TAUAnRDEm	1: Enables simultaneous rewrite
TAUAnRDS.TAUAnRDSm	0: The master channel is monitored for the simultaneous rewrite trigger 1: An upper channel outside the channel group is monitored for the simultaneous rewrite trigger
TAUAnRDM.TAUAnRDMm	0: The simultaneous rewrite trigger signal is generated when the master channel starts counting
TAUAnRDC.TAUAnRDCm	0: Channel is not monitored for an INTTAUAnIm signal that is used as the simultaneous rewrite trigger. If TAUAnRDS.TAUAnRDSm = 0, the master channel is monitored for the simultaneous rewrite trigger regardless of the value of this bit.

(6) Register settings for slave channels 2 to 7

(a) TAUAnCMORm for slave channels 2 to 7

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn CKS[1:0]		TAUAn CCS[1:0]		TAUAn MAS	TAUAnSTS[2:0]			TAUAn COS[1:0]		-	TAUAnMD[4:1]				TAUAn MD0

Table 12-180 TAUAnCMORm settings for slave channels 2 to 7 of the Non-Complementary Modulation Output Function Type 1

Bit name	Setting
TAUAnCKS[1:0]	00: Operation clock = CK0 01: Operation clock = CK1 10: Operation clock = CK2 11: Operation clock = CK3 The value of the TAUAnCKS[1:0] bit of the master and slave channels must be identical.
TAUAnCCS[1:0]	00: Operation clock is used as the count clock
TAUAnMAS	0: Channel is a slave channel
TAUAnSTS[2:0]	100: INTTAUAnIm of the master channel is the start trigger
TAUAnCOS[1:0]	00: Not used, so set to 00
TAUAnMD[4:1]	0100: One Count Mode
TAUAnMD0	1: Generates INTTAUAnIm and toggles TAUAnTTOUTm at operation start or restart

(b) TAUAnCMURm for slave channels 2 to 7

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-														TAUAnTIS[1:0]	

Table 12-181 TAUAnCMURm settings for slave channels 2 to 7 of the Non-Complementary Modulation Output Function Type 1

Bit name	Setting
TAUAnTIS[1:0]	00: Not used, so set to 00

(c) Channel output mode of slave channels 2 to 7

Table 12-182 Control bit settings for slave channels 2 to 7 of the Synchronous Channel Output Mode 1 with Non-Complementary Modulation Output

Bit name	Setting
TAUAnTOE.TAUAnTOEm	1: Enables Independent Channel Output Mode
TAUAnTOM.TAUAnTOMm	1: Synchronous channel output
TAUAnTOC.TAUAnTOCm	0: Operation mode 1
TAUAnTOL.TAUAnTOLm	0: Positive logic 1: Inverted logic
TAUAnTDE.TAUAnTDEm	0: Disables dead time operation
TAUAnTDM.TAUAnTDMm	0: When dead time operation is disabled (TAUAnTDE.TAUAnTDEm = 0), set these bits to 0
TAUAnTDL.TAUAnTDLm	
TAUAnTRE.TAUAnTREm	1: Enables real-time output
TAUAnTRO.TAUAnTROm	0: Real-time output is low 1: Real-time output is high
TAUAnTRC.TAUAnTRCm	0: The upper channel generates the real-time trigger for channel m
TAUAnTME.TAUAnTMEm	0: Disables modulation 1: Enables modulation

(d) Simultaneous rewrite for slave channels 2 to 7

The simultaneous rewrite settings of the master and slave channels must be identical.

Table 12-183 Simultaneous rewrite settings for slave channels 2 to 7 of the Non-Complementary Modulation Output Function Type 1

Bit name	Setting
TAUAnRDE.TAUAnRDEm	1: Enables simultaneous rewrite
TAUAnRDS.TAUAnRDSm	0: The master channel is monitored for the simultaneous rewrite trigger 1: An upper channel outside the channel group is monitored for the simultaneous rewrite trigger
TAUAnRDM.TAUAnRDMm	0: The simultaneous rewrite trigger signal is generated when the master channel starts counting
TAUAnRDC.TAUAnRDCm	0: Channel is not monitored for an INTTAUAnIm signal that is used as the simultaneous rewrite trigger. If TAUAnRDS.TAUAnRDSm = 0, the master channel is monitored for the simultaneous rewrite trigger regardless of the value of this bit.

(7) Operating Procedure for Non-Complementary Modulation Output Function Type 1

Table 12-184 Operating procedure for Non-Complementary Modulation Output Function Type 1 (1/2)

	Operation	Status of TAUAn
Initial channel setting	Master channel: set the TAUAnCMORm and TAUAnCMURm registers and the channel output mode as described in *(4) "Register settings for the master channel" on page 808* Slave channel 1: set the TAUAnCMORm and TAUAnCMURm registers and the channel output mode as described in *(5) "Register settings for slave channel 1" on page 810* Slave channel 2 to 7: set the TAUAnCMORm and TAUAnCMURm registers and the channel output mode as described in *(6) "Register settings for slave channels 2 to 7" on page 812* Set the values of the TAUAnCDRm registers of all channels: set the pulse cycle in TAUAnCDRm of the master channel, set TAUAnCDRm of slave channel 1 to 0000_H, and set the duty width in TAUAnCDRm of slave channels 2 to 7 Set TAUAnTRC.TAUAnTRCm = 1 for slave channel 1	Channel operation is stopped.

Table 12-184 Operating procedure for Non-Complementary Modulation Output Function Type 1 (2/2)

	Operation	Status of TAUAn
Start operation	Set TAUAnTS.TAUAnTSm of the master and slave channels to 1 simultaneously (for channel restart, only slaves 2 to 7). TAUAnTS.TAUAnTSm is a trigger bit, so it is automatically cleared to 0.	TAUAnTE.TAUAnTEm of the master and slave channels is set to 1 and the counters start to count down.
During operation	TAUAnCDRm, TAUAnTRO.TAUAnTROm, and TAUAnTME.TAUAnTMEm can be changed at any time. TAUAnCNTm and TAUAnRSF.TAUAnRSFm can be read at any time. TAUAnRDT.TAUAnRDTm can be changed during operation.	TAUAnCNTm loads the TAUAnCDRm value of the master channel and slave channels 1 to 7, and then counts down. When the counter of the master channel reaches 0000_H: • INTTAUAnIm is generated. • TAUAnCNTm reloads the TAUAnCDRm value, and then continues counting down. • The PWM output signals of slave channels 2 to 7 are set/reset. • TAUAnCNTm reloads the TAUAnCDRm value of slave channel 1, and then counts down. • TAUAnCNTm reloads the TAUAnCDRm value of slave channels 2 to 7, and then counts down. • When the counter of slave channel 1 or slave channels 2 to 7 reaches 0000_H: - INTTAUAnIm is generated. • When the counter of slave channels 2 to 7 reaches 0000_H: - The PWM output signals of slave channels 2 to 7 are set/reset. TAUAnTTOUTm of slave channels 2 to 7 outputs a PWM signal, a high signal, or a low signal depending on the value of the real-time output bits (TAUAnTRO.TAUAnTROm) and the modulation output bits (TAUAnTME.TAUAnTMEm) of a pair of slave channels.
Stop operation	Set TAUAnTT.TAUAnTTm of the master and slave channels to 1 simultaneously. TAUAnTT.TAUAnTTm is a trigger bit, so it is automatically cleared to 0.	TAUAnTE.TAUAnTEm is cleared to 0 and the counter stops. TAUAnCNTm and TAUAnTTOUTm stop and retain their current values.

Restart

(8) Specific timing diagrams

The following settings apply to the specific timing diagram:

- Slave channels 2 to 7: positive logic (TAUAnTOL.TAUAnTOLm = 0)

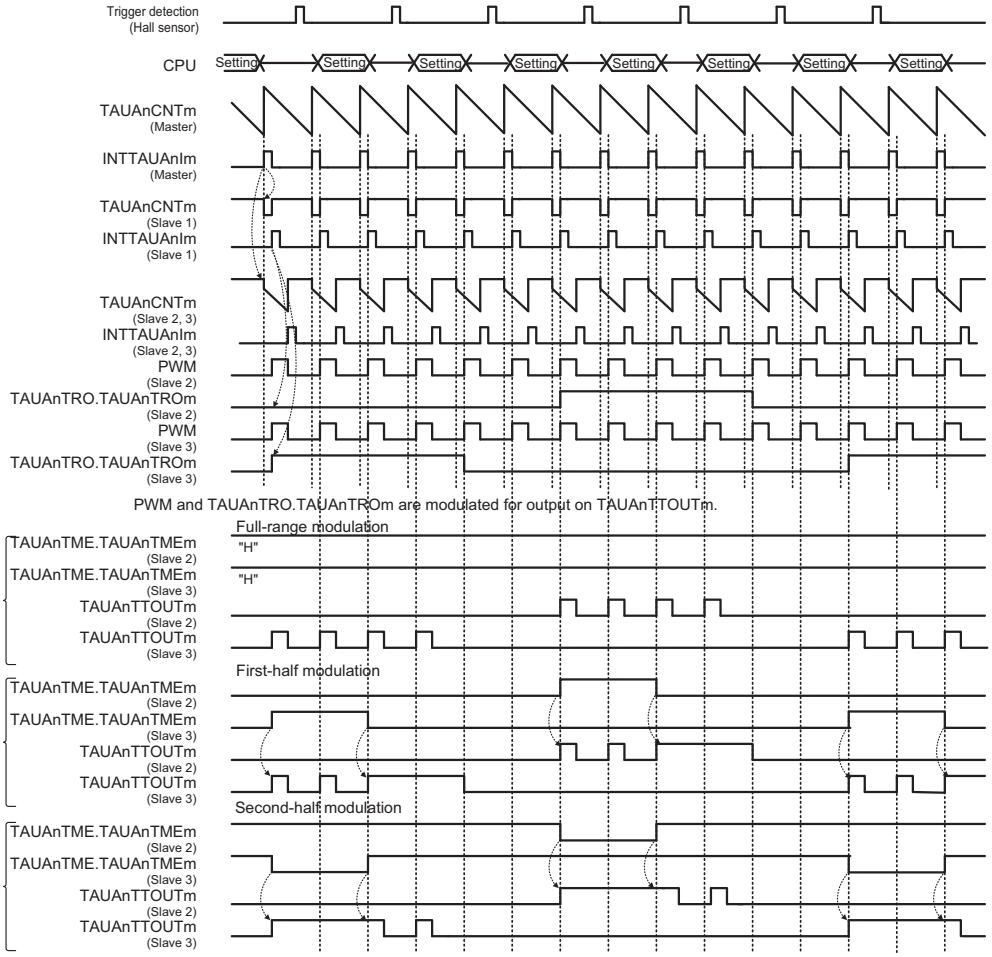


Figure 12-117 Specific timing diagram for Non-Complementary Modulation Output Function Type 1

The above timing diagram shows how full modulation, first-half modulation, and second-half modulation can be achieved by modifying the TAUAnTME.TAUAnTMEm bits of the lower slave channels during operation.

The “Setting” symbol indicates the time period during which the values of TAUAnCDRm, TAUAnTME.TAUAnTMEm and TAUAnTRO.TAUAnTROm can be changed.

The TAUAnTME.TAUAnTMEm setting is applied when counting starts or when a master channel cycle is detected. A modulation waveform is output from TAUAnTTOUTm according to the changed setting.

The values of the TAUAnTRO.TAUAnTROm bits are specified by software, but the new values are only applied when an interrupt occurs on slave channel 1.

12.25.2 Non-Complementary Modulation Output Function Type 2

(1) Overview

Summary This function outputs a PWM signal, a high signal, or a low signal from TAUAnTTOUTm depending on the value of the real-time output bits (TAUAnTRO.TAUAnTROm) and the modulation output enable bits (TAUAnTME.TAUAnTMEm) of a pair of slave channels. Three pairs of channels are typically used.

Prerequisites

- One master channel and seven slave channels
- The operation mode of the master channel must be set to Interval Timer Mode. Refer to *Table 12-186 “TAUAnCMORm settings for the master channel of Non-Complementary Modulation Output Function Type 2” on page 822*.
- The operation mode of slave channel 1 must be set to Event Count Mode. Refer to *Table 12-190 “TAUAnCMORm settings for slave channel 1 of Non-Complementary Modulation Output Function Type 2” on page 824*.
- The operation mode of slave channels 2 to 7 must be set to Up Down Count Mode. Refer to *Table 12-193 “TAUAnCMORm settings for slave channels 2 to 7 of Non-Complementary Modulation Output Function Type 2” on page 826*.
- TAUAnTTOUTm is not used for the master channel of this function.
- TAUAnTTOUTm of slave channel 1 is not used for this function, but TAUAnTRC.TAUAnTRCm must be set to 1. Refer to *12.8 “Channel Output Modes” on page 577*.
- The channel output mode of slave channels 2 to 7 must be set to Simultaneous Channel Output Mode 2 with Non-Complementary Modulation Output. Refer to *12.8 “Channel Output Modes” on page 577*.

Description The counters of the master and slave channels are started by setting the channel trigger bit (TAUAnTS.TAUAnTSm) to 1. This in turn sets TAUAnTE.TAUAnTEm, enabling count operation. The values of the data registers (TAUAnCDRm) are loaded to the counters (TAUAnCNTm).

- Master channel:

  The counter of the master channel starts to count down. When the counter reaches 0000_H, INTTAUAnIm is generated.

- Slave channel 1:

  When slave channel 1 detects an interrupt on the master channel, counter reduces by 1. When the counter reaches 0000_H, it awaits the next interrupt from the master channel. Next, the value of TAUAnCDRm is reloaded to TAUAnCNTm (slave 1), and then INTTAUAnIm is generated.

  Because slave channel 1 is specified as the trigger channel for real-time output (TAUAnTRC.TAUAnTRCm = 1), when an interrupt is generated on slave channel 1, the real-time output bit (TAUAnTRO.TAUAnTROm) value of the channel that detects the generated interrupt of the corresponding channel is changed.

- Slave channel 2:

  When an interrupt is detected from the master channel, the TAUAnCNTm counts in the reverse direction. If the interrupt is detected during up count, the TAUAnCNTm reloads the value of TAUAnCDRm, and then starts counting down.

When TAUAnCNTm = 0001_H, an interrupt is generated and the PWM output is set or reset.

The combination of the master channel and slave channel 2 generates a PWM output. The master channel generates the period of the PWM output, the slave channel the duty cycle.

Slave channels 3 to 7 behave analogously to slave channel 2.

The signal that is output from TAUAnTTOUTm depends on the value of the real-time output bit (TAUAnTRO.TAUAnTROm) and the modulation output bit (TAUAnTME.TAUAnTMEm) of the slave channel, as shown in *Table 12-185 "TAUAnTTOUTm output of a pair of slave channels in Non-Complementary Modulation Output Function Type 2" on page 818*.

Forced restart is not possible with this function. The counter can be stopped by setting TAUAnTT.TAUAnTTm to 1 for the master and slave channels, which in turn sets TAUAnTE.TAUAnTEm to 0. TAUAnCNTm and TAUAnTTOUTm of master and slave channels stop but retain their values. The counters can be restarted by setting TAUAnTS.TAUAnTSm to 1.

Conditions

- If TAUAnTME.TAUAnTMEm of the slave channel is 1:
  - If TAUAnTRO.TAUAnTROm of the channel is 1, TAUAnTTOUTm outputs the corresponding PWM of the channel
  - If TAUAnTRO.TAUAnTROm of the channel is 0, TAUAnTTOUTm outputs a low signal
- If TAUAnTME.TAUAnTMEm of the slave channel is 0:
  - If TAUAnTRO.TAUAnTROm of the channel is 1, TAUAnTTOUTm outputs a high signal
  - If TAUAnTRO.TAUAnTROm of the channel is 0, TAUAnTTOUTm outputs a low signal
- If TAUAnTOL.TAUAnTOLm = 1 the high and low signals output from TAUAnTTOUTm are inverted. The PWM signals remain unaffected. Only the initial setting can be used for TAUAnTOL.TAUAnTOLm (and this setting cannot be changed during operation).

Table 12-185 TAUAnTTOUTm output of a pair of slave channels in Non-Complementary Modulation Output Function Type 2

TAUAnTME.TAUAnTMEm	TAUAnTRO.TAUAnTROm	TAUAnTTOUTm outputs
0	0	Low
	1	High
1	0	Low
	1	PWMm

- Simultaneous rewrite can be used with this function. Please refer to *12.7 "Simultaneous Rewrite" on page 565*
- If TAUAnTOL.TAUAnTOLm is cleared to 0 on slave channels 2 to 7, TAUAnTO.TAUAnTOm is set to the low level before clearing TAUAnTE.TAUAnTEm to 0.
- If TAUAnTOL.TAUAnTOLm is set to 1 on slave channels 2 to 7, TAUAnTO.TAUAnTOm is set to the high level before clearing TAUAnTE.TAUAnTEm to 0.

(2) Equations

For slave channels 2 to 7:

PWM output cycle time
= [TAUAnCDRm (master) + 1] x count clock

PWM output duty time
= [TAUAnCDRm (master) + 1 - TAUAnCDRm (slave)] x 2 x count clock

(3) Block diagram and general timing diagram

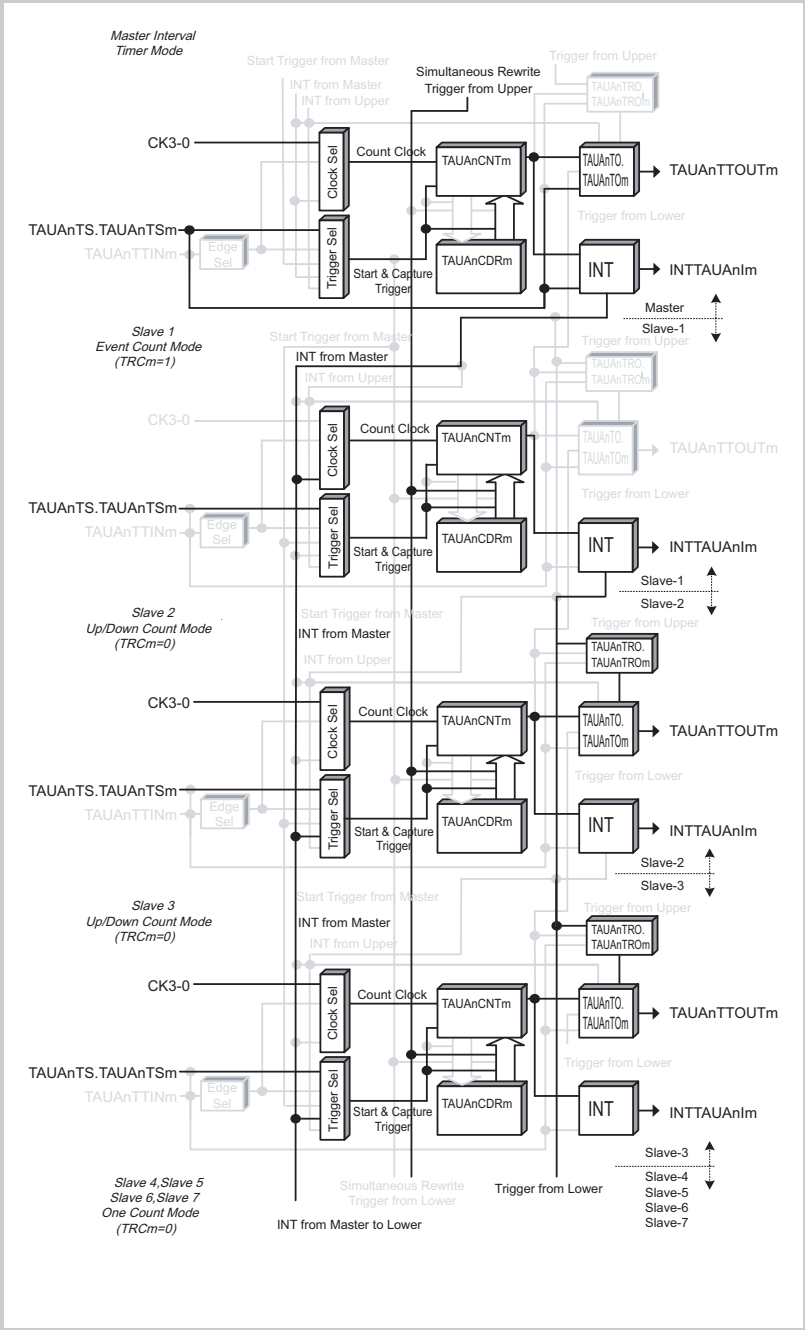


Figure 12-118 Block diagram for Non-Complementary Modulation Output Function Type 2

The following settings apply to the general timing diagram:

- Master channel: INTTAUAnIm not generated at operation start (TAUAnCMORm.TAUAnMD0 = 0)
- Slave channels 2 to 7: positive logic (TAUAnTOL.TAUAnTOLm = 0)

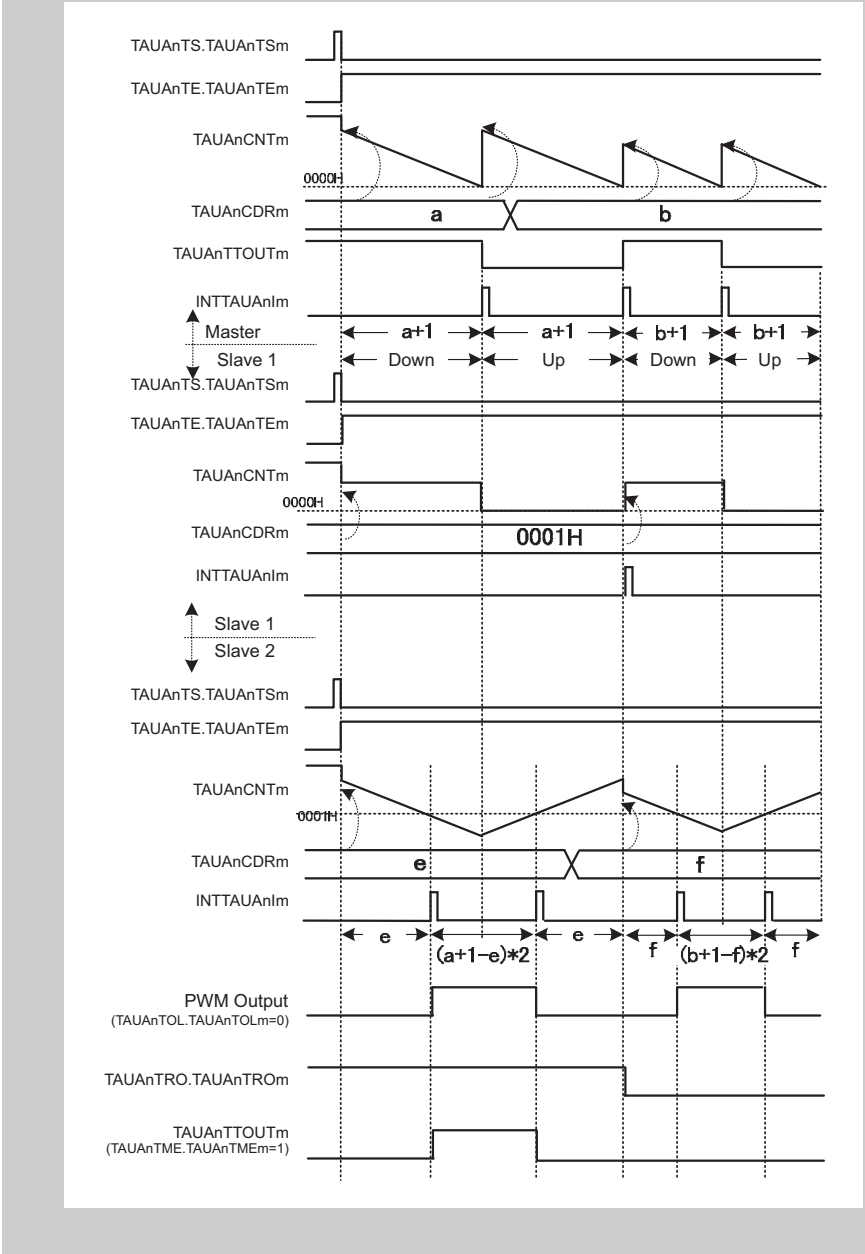


Figure 12-119 General timing diagram for Non-Complementary Modulation Output Function Type 2

(4) Register settings for the master channel

(a) TAUAnCMORm for the master channel

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn CKS[1:0]		TAUAn CCS[1:0]		TAUAn MAS	TAUAnSTS[2:0]			TAUAn COS[1:0]		-	TAUAnMD[4:1]				TAUAn MD0

Table 12-186 TAUAnCMORm settings for the master channel of Non-Complementary Modulation Output Function Type 2

Bit name	Setting
TAUAnCKS[1:0]	00: Operation clock = CK0 01: Operation clock = CK1 10: Operation clock = CK2 11: Operation clock = CK3 The value of the TAUAnCKS[1:0] bit of the master and slave channels must be identical.
TAUAnCCS[1:0]	00: Operation clock is used as the count clock
TAUAnMAS	1: Channel is master channel
TAUAnSTS[2:0]	000: Counter triggered by software trigger
TAUAnCOS[1:0]	00: Not used, so set to 00
TAUAnMD[4:1]	0000: Interval Timer Mode
TAUAnMD0	0: INTTAUAnIm not generated at operation start or restart 1: Generates INTTAUAnIm at operation start or restart

(b) TAUAnCMURm for the master channel

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-														TAUAnTIS[1:0]	

Table 12-187 TAUAnCMURm settings for the master channel of Non-Complementary Modulation Output Function Type 2

Bit name	Setting
TAUAnTIS[1:0]	00: Not used, so set to 00

(c) Channel output mode of the master channel

Table 12-188 Master Channel Control Bit Settings When Using Non-Complementary Modulation Output Function Type 2

Bit name	Setting
TAUAnTOE.TAUAnTOEm	1: Enables the independent channel output mode.
TAUAnTOM.TAUAnTOMm	0: Independent channel output
TAUAnTOC.TAUAnTOCm	0: Operation mode 1 (the toggle mode if TAUAnTOM.TAUAnTOMm is 0)
TAUAnTOL.TAUAnTOLm	0: Positive logic
TAUAnTDE.TAUAnTDEm	0: Disables dead time operation.
TAUAnTDM.TAUAnTDMm	0: When dead time operation is disabled (TAUAnTDE.TAUAnTDEm = 0), clear these bits to 0.
TAUAnTDL.TAUAnTDLm	
TAUAnTRE.TAUAnTREm	0: Disables real-time output.
TAUAnTRO.TAUAnTROm	0: The real-time output is low.
TAUAnTRC.TAUAnTRCm	0: The next upper channel generates the real-time trigger for channel m.
TAUAnTME.TAUAnTMEm	0: Disables modulation.

(d) Simultaneous rewrite for the master channel

The simultaneous rewrite settings of the master and slave channels must be identical.

Table 12-189 Simultaneous rewrite settings for the master channel of Non-Complementary Modulation Output Function Type 2

Bit name	Setting
TAUAnRDE.TAUAnRDEm	1: Enables simultaneous rewrite
TAUAnRDS.TAUAnRDSm	0: The master channel is monitored for the simultaneous rewrite trigger 1: An upper channel outside the channel group is monitored for the simultaneous rewrite trigger
TAUAnRDM.TAUAnRDMm	1: The simultaneous rewrite trigger signal is generated when the master channel starts counting and the corresponding slave channel is at the peak of a triangular wave
TAUAnRDC.TAUAnRDCm	0: When simultaneous rewrite is disabled (TAUAnTAUAnRDE.TAUAnRDEm = 0), set these bits to 0. If TAUAnRDS.TAUAnRDSm = 0, the master channel is monitored for the simultaneous rewrite trigger regardless of the value of this bit.

Note If the TAUAnRDS.TAUAnRDSm bit is 1, there must be a channel higher than the master channel to which a simultaneous rewrite trigger signal is generated.

(5) Register settings for slave channel 1

(a) TAUAnCMORm for slave channel 1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn CKS[1:0]		TAUAn CCS[1:0]		TAUAn MAS	TAUAnSTS[2:0]			TAUAn COS[1:0]		-	TAUAnMD[4:1]				TAUAn MD0

Table 12-190 TAUAnCMORm settings for slave channel 1 of Non-Complementary Modulation Output Function Type 2

Bit name	Setting
TAUAnCKS[1:0]	00: Operation clock = CK0 01: Operation clock = CK1 10: Operation clock = CK2 11: Operation clock = CK3 The value of the TAUAnCKS[1:0] bit of the master and slave channels must be identical.
TAUAnCCS[1:0]	11: INTTAUAnIm of the master channel is used as the count clock
TAUAnMAS	0: Channel is a slave channel
TAUAnSTS[2:0]	000: Counter triggered by software trigger 011: Simultaneous rewrite trigger
TAUAnCOS[1:0]	00: Not used, so set to 00
TAUAnMD[4:1]	0011: Event Count Mode
TAUAnMD0	0: INTTAUAnIm not generated at operation start or restart

(b) TAUAnCMURm for slave channel 1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-														TAUAnTIS[1:0]	

Table 12-191 TAUAnCMURm settings for slave channel 1 of Non-Complementary Modulation Output Function Type 2

Bit name	Setting
TAUAnTIS[1:0]	00: Not used, so set to 00

(c) Channel output mode

Because the channel output mode is not used by slave channel 1 of this function, clear TAUAnTOEn.TAUAnTOEn. However, it can be used in Independent Channel Output Mode Controlled by Software.

Caution TAUAnTRC.TAUAnTRCm must be set to 1 to enable slave channel 1 to be used as the real-time output trigger.

(d) Simultaneous rewrite for slave channel 1

The simultaneous rewrite settings of the master and slave channels must be identical.

Table 12-192 Simultaneous rewrite settings for slave channel 1 of Non-Complementary Modulation Output Function Type 2

Bit name	Setting
TAUAnRDE.TAUAnRDEm	1: Enables simultaneous rewrite
TAUAnRDS.TAUAnRDSm	0: The master channel is monitored for the simultaneous rewrite trigger 1: An upper channel outside the channel group is monitored for the simultaneous rewrite trigger
TAUAnRDM.TAUAnRDMm	1: The simultaneous rewrite trigger signal is generated when the master channel starts counting and the corresponding slave channel is at the peak of a triangular wave
TAUAnRDC.TAUAnRDCm	0: Channel is not monitored for an INTTAUAnIm signal that is used as the simultaneous rewrite trigger. If TAUAnRDS.TAUAnRDSm = 0, the master channel is monitored for the simultaneous rewrite trigger regardless of the value of this bit.

(6) Register settings for slave channels 2 to 7

(a) TAUAnCMORm for slave channels 2 to 7

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn CKS[1:0]		TAUAn CCS[1:0]		TAUAn MAS	TAUAnSTS[2:0]			TAUAn COS[1:0]		-	TAUAnMD[4:1]				TAUAn MD0

Table 12-193 TAUAnCMORm settings for slave channels 2 to 7 of Non-Complementary Modulation Output Function Type 2

Bit name	Setting
TAUAnCKS[1:0]	00: Operation clock = CK0 01: Operation clock = CK1 10: Operation clock = CK2 11: Operation clock = CK3 The value of the TAUAnCKS[1:0] bit of the master and slave channels must be identical.
TAUAnCCS[1:0]	00: Operation clock is used as the count clock
TAUAnMAS	0: Channel is a slave channel
TAUAnSTS[2:0]	111: The up/down output trigger signal TAUAnTUDSm of the master channel is the start trigger
TAUAnCOS[1:0]	00: Not used, so set to 00
TAUAnMD[4:1]	1001: Up Down Count Mode
TAUAnMD0	0: INTTAUAnIm not generated at operation start or restart

(b) TAUAnCMURm for slave channels 2 to 7

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-														TAUAnTIS[1:0]	

Table 12-194 TAUAnCMURm settings slave channels 2 to 7 of Non-Complementary Modulation Output Function Type 2

Bit name	Setting
TAUAnTIS[1:0]	00: Not used, so set to 00

(c) Channel output mode of slave channels 2 to 7

Table 12-195 Control bit settings slave channels 2 to 7 of Synchronous Channel Output Mode 2 with Non-Complementary Modulation Output

Bit name	Setting
TAUAnTOE.TAUAnTOEm	1: Enables Independent Channel Output Mode
TAUAnTOM.TAUAnTOMm	1: Synchronous channel output
TAUAnTOC.TAUAnTOCm	1: Operation mode 2
TAUAnTOL.TAUAnTOLm	0: Positive logic 1: Inverted logic
TAUAnTDE.TAUAnTDEm	0: Disables dead time operation
TAUAnTDM.TAUAnTDMm	0: When dead time operation is disabled (TAUAnTDE.TAUAnTDEm = 0), set these bits to 0
TAUAnTDL.TAUAnTDLm	
TAUAnTRE.TAUAnTREm	1: Enables real-time output
TAUAnTRO.TAUAnTROm	0: Real-time output is low 1: Real-time output is high
TAUAnTRC.TAUAnTRCm	0: The next upper channel generates the real-time trigger for channel m
TAUAnTME.TAUAnTMEm	0: Disables modulation 1: Enables modulation

(d) Simultaneous rewrite for slave channels 2 to 7

The simultaneous rewrite settings of the master and slave channels must be identical.

Table 12-196 Simultaneous rewrite settings slave channels 2 to 7 of Non-Complementary Modulation Output Function Type 2

Bit name	Setting
TAUAnRDE.TAUAnRDEm	1: Enables simultaneous rewrite
TAUAnRDS.TAUAnRDSm	0: The master channel is monitored for the simultaneous rewrite trigger 1: An upper channel outside the channel group is monitored for the simultaneous rewrite trigger
TAUAnRDM.TAUAnRDMm	1: The simultaneous rewrite trigger signal is generated when the master channel starts counting and the corresponding slave channel is at the peak of a triangular wave
TAUAnRDC.TAUAnRDCm	0: Channel is not monitored for an INTTAUAnIm signal that is used as the simultaneous rewrite trigger. If TAUAnRDS.TAUAnRDSm = 0, the master channel is monitored for the simultaneous rewrite trigger regardless of the value of this bit.

(7) Operating Procedure for Non-Complementary Modulation Output Function Type 2

Table 12-197 Operating procedure for Non-Complementary Modulation Output Function Type 2 (1/2)

	Operation	Status of TAUAn
Initial channel setting	Master channel: set the TAUAnCMORm and TAUAnCMURm registers and the channel output mode as described in *(4) "Register settings for the master channel" on page 822* Slave channel 1: set the TAUAnCMORm and TAUAnCMURm registers and the channel output mode as described in *(5) "Register settings for slave channel 1" on page 824* Slave channel 2 to 7: set the TAUAnCMORm and TAUAnCMURm registers and the channel output mode as described in *(6) "Register settings for slave channels 2 to 7" on page 826* Set the values of the TAUAnCDRm registers of all channels: set the pulse cycle in TAUAnCDRm of the master channel, using TAUAnCDRm of slave channel 1 set the number of interrupts on the master channel to be ignored before slave 1 generates an input, and set the duty width in TAUAnCDRm of slave channels 2 to 7 Set TAUAnTRC.TAUAnTRCm = 1 for slave channel 1	Channel operation is stopped.

Table 12-197 Operating procedure for Non-Complementary Modulation Output Function Type 2 (2/2)

	Operation	Status of TAUAn
Start operation	Set TAUAnTS.TAUAnTSm of the master and slave channels to 1 simultaneously (for channel restart, only slaves 2 to 7). TAUAnTS.TAUAnTSm is a trigger bit, so it is automatically cleared to 0.	TAUAnTE.TAUAnTEm of the master and slave channels is set to 1 and the counters start to count down.
During operation	TAUAnCDRm, TAUAnTRO.TAUAnTROm, and TAUAnTME.TAUAnTMEm can be changed at any time. TAUAnCNTm and TAUAnRSF.TAUAnRSFm can be read at any time. TAUAnRDT.TAUAnRDTm can be changed during operation.	TAUAnCNTm loads the TAUAnCDRm value of the master channel and slave channels 2 and 7, and then counts down. The TAUAnCDRm value of slave channel 1 is loaded, and a master channel interrupt is awaited. When the counter of the master channel reaches 0000_H: • INTTAUAnIm is generated. • TAUAnCNTm reloads the TAUAnCDRm value, and then continues counting down. • TAUAnCNTm of slave channel 1 reduces by 1 and awaits the next interrupt on the master channel. • TAUAnCNTm of slave channels 2 to 7 counts in the other direction. • When the counter of slave channel 1 reaches 0000_H it awaits the next interrupt from the master channel. When it is detected: - INTTAUAnIm is generated. • When the counter of slave channels 2 to 7 reaches 0001_H: - INTTAUAnIm is generated. - The PWM output signals of slave channels 2 to 7 are set/reset. TAUAnTTOUTm of slave channels 2 to 7 outputs a PWM signal, a high signal, or a low signal depending on the value of the real-time output bits (TAUAnTRO.TAUAnTROm) and the modulation output bits (TAUAnTME.TAUAnTMEm) of a pair of slave channels.
Stop operation	Set TAUAnTT.TAUAnTTm of the master and slave channels to 1 simultaneously. TAUAnTT.TAUAnTTm is a trigger bit, so it is automatically cleared to 0.	TAUAnTE.TAUAnTEm is cleared to 0 and the counter stops. TAUAnCNTm and TAUAnTTOUTm stop and retain their current values.

Restart

(8) Specific timing diagrams

The following settings apply to the general timing diagram:

- Slave channels 2 to 7: positive logic (TAUAnTOL.TAUAnTOLm = 0)

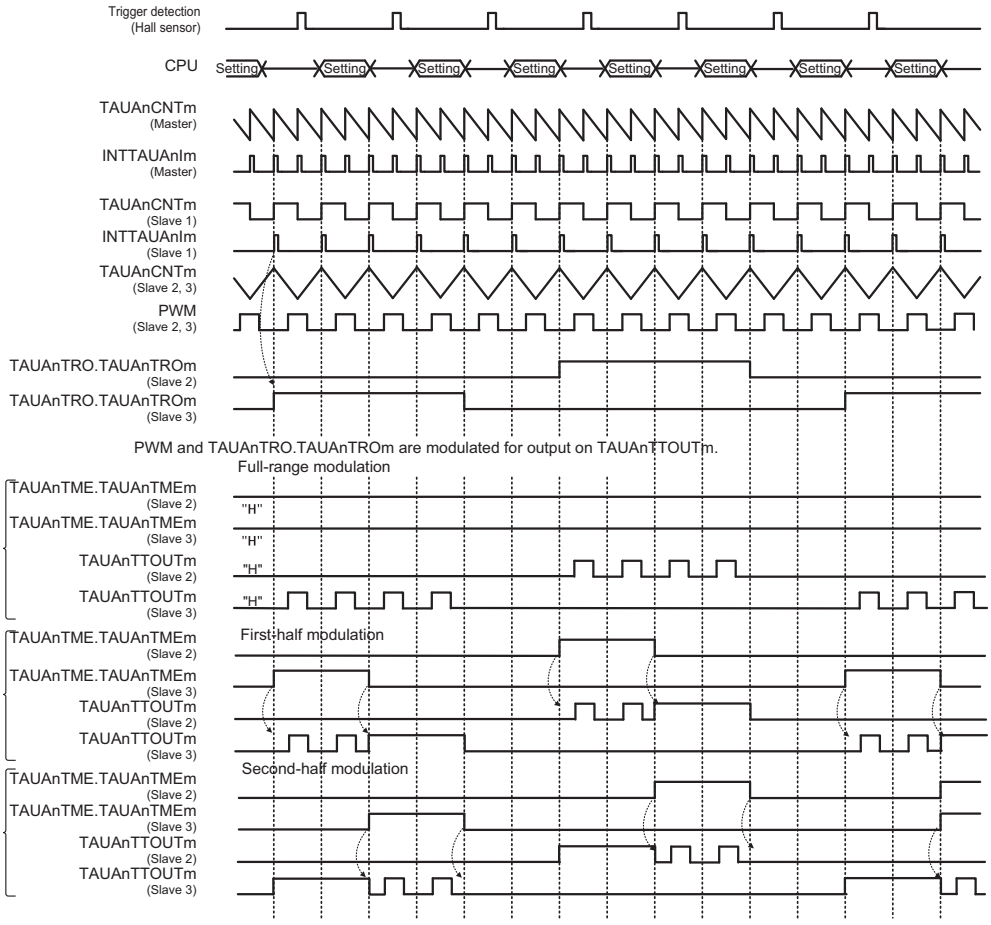


Figure 12-120 Specific timing diagram for Non-Complementary Modulation Output Function Type 2

The above timing diagram shows how full modulation, first-half modulation, and second-half modulation can be achieved by modifying the TAUAnTME.TAUAnTMEm bits of the lower slave channels during operation.

The “Setting” symbol indicates the time period during which the values of TAUAnCDRm, TAUAnTME.TAUAnTMEm and TAUAnTRO.TAUAnTROm can be changed.

The TAUAnTME.TAUAnTMEm setting is applied when counting starts or when a triangle PWM carrier cycle (at the peak interrupt timing) is detected.

The values of the TAUAnTRO.TAUAnTROm bits are specified by software, but the new values are only applied when an interrupt occurs on slave channel 1.

12.25.3 Complementary Modulation Output Function

(1) Overview

Summary This function outputs a PWM signal, a high signal, or a low signal from TAUAnTTOUTm depending on the value of the real-time output bits (TAUAnTRO.TAUAnTROm), the modulation output bits (TAUAnTME.TAUAnTMEm), and the output level bits (TAUAnTDL.TAUAnTDLm) of a pair of slave channels. Three pairs of channels are typically used.

Prerequisites

- One master channel and seven slave channels
- The operation mode of the master channel must be set to Interval Timer Mode. Refer to *Table 12-200 “TAUAnCMORm settings for the master channel of the Complementary Modulation Output Function” on page 837*.
- The operation mode of slave channel 1 must be set to Event Count Mode. Refer to *Table 12-204 “TAUAnCMORm settings for slave channel 1 of the Complementary Modulation Output Function” on page 839*.
- The operation mode of slave channels 2, 4, and 6 must be set to Up Down Count Mode. Refer to *Table 12-207 “TAUAnCMORm settings for slave channels 2, 4, and 6 of the Complementary Modulation Output Function” on page 841*.
- The operation mode of slave channels 3, 5, and 7 must be set to One Count Mode. Refer to *Table 12-211 “TAUAnCMORm settings for slave channels 3, 5, and 7 of the Complementary Modulation Output Function” on page 843*.
- The channel output mode of the master channel must be set to Independent Channel Output Mode 1. Refer to *12.8 “Channel Output Modes” on page 577*.
- TAUAnTTOUTm of slave channel 1 is not used for this function, but TAUAnTRC.TAUAnTRCm must be set to 1. Refer to *12.8 “Channel Output Modes” on page 577*.
- The channel output mode of slave channels 2 to 7 must be set to Simultaneous Channel Output Mode 2 with Complementary Modulation Output. Refer to *12.8 “Channel Output Modes” on page 577*.

Description

- Master channel:

  The counter of the master channel is started by setting the channel trigger bit (TAUAnTS.TAUAnTSm) to 1. This in turn sets TAUAnTE.TAUAnTEm, enabling count operation. The value of the data register of the master channel (TAUAnCDRm) is loaded to the counter (TAUAnCNTm) and the counter starts to count down from this value.

  When the counter of the master channel reaches 0000_H, INTTAUAnIm is generated. This triggers the counter of slave channel 1 to reduce by 1 and the counter of slave channel 2 to count in the reverse direction.

- Slave channel 1:

  When the counter reaches 0000_H, it awaits the next interrupt from the master channel. Next, TAUAnCNTm (slave 1) reloads the value of INTTAUAnIm, and then INTTAUAnIm is generated.

  Slave channel 1 is set as the channel that triggers real-time output (TAUAnTRC.TAUAnTRCm = 1). For each channel on which an interrupt generated on slave channel 1 is detected, the real-time output bit (TAUAnTRO.TAUAnTROm) value is applied. The value of the real-time output bits can be changed at any time by the application software, but the new values are only applied when the interrupt occurs on slave channel 1.

- Slave channel 2:

  When the counter of slave channel 2 reaches 0001_H, the counter of slave channel 3 starts to count down. When the counter of slave channel 3 reaches 0000_H it generates an interrupt.

- Slave channels 2 and 3:

  The combination of the master channel and slave channels 2 and 3 generates a PWM output. The master channel generates the period of the PWM output, slave channel 2 the duty cycle, and slave channel 3 generates the dead time.

- Slave channels 4 to 7:

  Slave channels 4 and 6 behave analogously to slave channel 2. Slave channels 5 and 7 behave analogously to slave channel 3.

The signal that is output from TAUAnTTOUTm depends on the value of the real-time output bits (TAUAnTRO.TAUAnTROm), the modulation output bits (TAUAnTME.TAUAnTMEm), and the output level bits (TAUAnTOL.TAUAnTOLm) of a pair of slave channels, as shown in *Table 12-173 "TAUAnTTOUTm output of a pair of slave channels in Non-Complementary Modulation Output Function Type 1" on page 805*.

However, the output of slave channels 2 and 3 must not be high simultaneously (for example to prevent a motor driver from short circuiting). To prevent both channels being high at the instant that one channel switches to low and the other switches to high, a delay must be added to the channel that switches to high so that this transition occurs later. This is achieved by adding dead time to either the positive or negative-phased PWM signal as specified by the value of the channel dead time output level register (TAUAnTDL.TAUAnTDLm).

The counter can be stopped by setting TAUAnTT.TAUAnTTm to 1 for the master and slave channels, which in turn sets TAUAnTE.TAUAnTEm to 0. TAUAnCNTm and TAUAnTTOUTm of master and slave channels stop but retain their values. The counters can be restarted by setting TAUAnTS.TAUAnTSm to 1.

Conditions

- If TAUAnTME.TAUAnTMEm of both channels in a pair is 1:
  - If the TAUAnTRO.TAUAnTROm value of one channel is 1, TAUAnTTOUTm performs PWM output for that channel.
  - If the TAUAnTRO.TAUAnTROm value of one channel is 0, TAUAnTTOUTm outputs a low level signal.
- If TAUAnTME.TAUAnTMEm of both channels in a pair is 0:
  - If TAUAnTRO.TAUAnTROm of a channel is 1, TAUAnTTOUTm outputs a high signal
  - If TAUAnTRO.TAUAnTROm of a channel is 0, TAUAnTTOUTm outputs a low signal
- If TAUAnTOL.TAUAnTOLm = 1, the high and low signals output from TAUAnTTOUTm are inverted. The PWM signal does not change in accordance with the TAUAnTOL.TAUAnTOLm setting.

Table 12-198 TAUAnTTOUTm output of a pair of slave channels in Complementary Modulation Output Function for TAUAnTOL.TAUAnTOLm = 0

TAUAnTME.TAUAnTME2	TAUAnTME.TAUAnTME3	TAUAnTRO.TAUAnTRO2	TAUAnTRO.TAUAnTRO3	TAUAnTTOUT2 outputs	TAUAnTTOUT3 outputs
0	0	0	0	LOW	LOW
		0	1	LOW	HIGH
		1	0	HIGH	LOW
		1	1	Prohibited	Prohibited
1	1	0	0	LOW	LOW
		0	1	~PWMm	PWMm
		1	0	PWMm	~PWMm
		1	1	Prohibited	Prohibited

Notes
1. ~PWM indicates an inverted PWM signal. PWM and -PWM are set up using TAUAnTDL.TAUAnTDLm.
2. All settings that are not listed in this table are not permitted.

- While TAUAnTRO.TAUAnTROm is set to 1 on one set of channels, if TAUAnTME.TAUAnTMEm is continuously set to 1, the modulation is entire modulation.
- If TAUAnTME.TAUAnTMEm is set to 1 during the first half of the period during which TAUAnTRO.TAUAnTROm is set to 1 on one set of channels, the modulation is first-half modulation.
- If TAUAnTME.TAUAnTMEm is set to 1 during the second half of the period during which TAUAnTRO.TAUAnTROm is set to 1 on one set of channels, the modulation is second-half modulation.
- If two channels simultaneously output a high level signal, the TAUAnTDL.TAUAnTDLm bit value determines whether the dead time is added to the positive-phase PWM signal or the negative-phase PWM signal.
  - If TAUAnTDL.TAUAnTDLm = 0, dead time is added to the positive PWM signal
  - If TAUAnTDL.TAUAnTDLm = 1 dead time is added to the negative PWM signal
  - The values of the TAUAnTDL.TAUAnTDLm bits must be manipulated by the application software during operation. To change TAUAnTDL.TAUAnTDLm, perform rewriting while TAUAnTRO.TAUAnTROm is 00B.

Table 12-199 TAUAnTDL.TAUAnTDLm settings for a pair of slave channels in Complementary Modulation Output Function for TAUAnTOL.TAUAnTOLm = 0

TAUAnTME. TAUAnTME2	TAUAnTME. TAUAnTME3	TAUAnTRO. TAUAnTRO2	TAUAnTRO. TAUAnTRO3	TAUAnTDL. TAUAnTDL2	TAUAnTDL. TAUAnTDL3
0	0	0	0	1	1
		0	1	1	0
		1	0	0	1
1	1	0	0	1	1
		0	1	1	0
		1	0	0	1

- The value of TAUAnCDRm of slave channel 1 must be set so that INTTAUAnIm is generated from slave 1 at the summit of a carrier cycle.
- Clear the TAUAnCMORm.TAUAnMD0 value of the master channel to 0.
- Simultaneous rewrite can be used with this function. Please refer to *12.7 "Simultaneous Rewrite" on page 565*.

(2) Equations

For slaves 2, 4, and 6

When TAUAnTOL.TAUAnTOLm = 0 and TAUAnCDR (slave 1):

PWM output cycle time = 2 x [TAUAnCDRm (master) + 1] x count clock

PWM output duty time = { [TAUAnCDRm (master) + 1 - TAUAnCDRm (slave 2)] x 2 - [TAUAnCDRm (slave 3) + 1] } x count clock

(3) Block diagram and general timing diagram

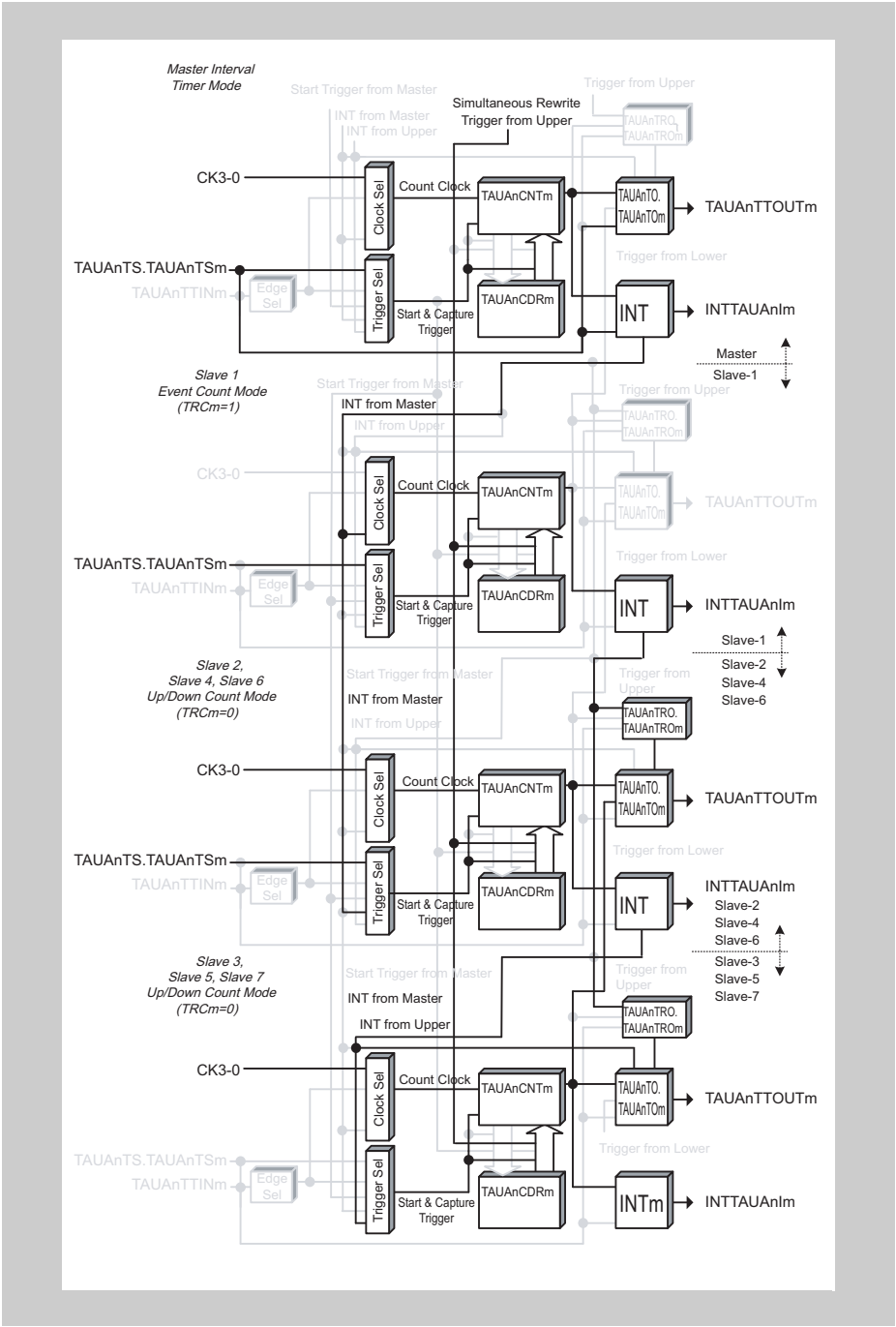


Figure 12-121 Block diagram for Complementary Modulation Output Function

The following settings apply to the general timing diagram:

- Master channel: INTTAUAnIm not generated at operation start (TAUAnCMORm.TAUAnMD0 = 0)
- Slave channels 2 to 7: positive logic (TAUAnTOL.TAUAnTOLm = 0)

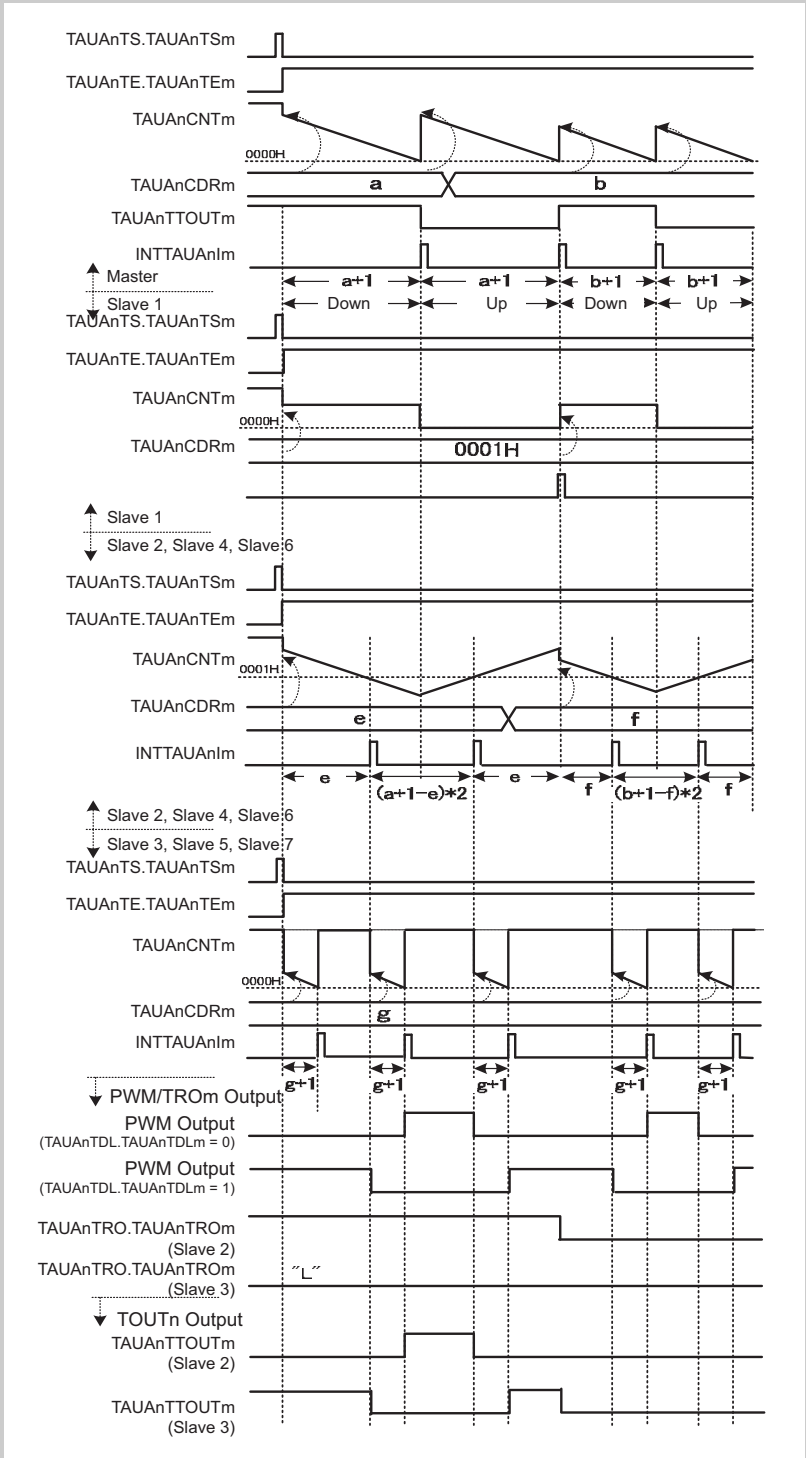


Figure 12-122 General timing diagram for Complementary Modulation Output Function

(4) Register settings for the master channel

(a) TAUAnCMORm for the master channel

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn CKS[1:0]		TAUAn CCS[1:0]		TAUAn MAS	TAUAnSTS[2:0]			TAUAn COS[1:0]		-	TAUAnMD[4:1]				TAUAn MD0

Table 12-200 TAUAnCMORm settings for the master channel of the Complementary Modulation Output Function

Bit name	Setting
TAUAnCKS[1:0]	00: Operation clock = CK0 01: Operation clock = CK1 10: Operation clock = CK2 11: Operation clock = CK3 The value of the TAUAnCKS[1:0] bit of the master and slave channels must be identical.
TAUAnCCS[1:0]	00: Operation clock is used as the count clock
TAUAnMAS	1: Channel is master channel
TAUAnSTS[2:0]	000: Counter triggered by software trigger
TAUAnCOS[1:0]	00: Not used, so set to 00
TAUAnMD[4:1]	0000: Interval Timer Mode
TAUAnMD0	0: INTTAUAnIm not generated and TAUAnTTOUTm does not toggle at operation start or restart 1: Generates INTTAUAnIm and toggles TAUAnTTOUTm at operation start or restart

(b) TAUAnCMURm for the master channel

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-														TAUAnTIS[1:0]	

Table 12-201 TAUAnCMURm settings for the master channel of the Complementary Modulation Output Function

Bit name	Setting
TAUAnTIS[1:0]	00: Not used, so set to 00

(c) Channel output mode

Table 12-202 Control bit settings for Independent Channel Output Mode 1

Bit name	Setting
TAUAnTOE.TAUAnTOEm	1: Enables Independent Channel Output Mode
TAUAnTOM.TAUAnTOMm	0: Independent channel output
TAUAnTOC.TAUAnTOCm	0: Operation mode 1 (= Toggle mode if TAUAnTOM.TAUAnTOMm = 0)
TAUAnTOL.TAUAnTOLm	0: Positive logic
TAUAnTDE.TAUAnTDEm	0: Disables dead time operation
TAUAnTDM.TAUAnTDMm	0: When dead time operation is disabled (TAUAnTDE.TAUAnTDEm = 0), set these bits to 0
TAUAnTDL.TAUAnTDLm	
TAUAnTRE.TAUAnTREm	0: Disables real-time output
TAUAnTRO.TAUAnTROm	0: When real-time output is disabled (TAUAnTRE.TAUAnTREm = 0), set these bits to 0
TAUAnTRC.TAUAnTRCm	
TAUAnTME.TAUAnTMEm	0: Disables modulation

(d) Simultaneous rewrite for the master channel

The simultaneous rewrite settings of the master and slave channels must be identical.

Table 12-203 Simultaneous rewrite settings for the master channel of the Complementary Modulation Output Function

Bit name	Setting
TAUAnRDE.TAUAnRDEm	1: Enables simultaneous rewrite
TAUAnRDS.TAUAnRDSm	0: The master channel is monitored for the simultaneous rewrite trigger 1: An upper channel outside the channel group is monitored for the simultaneous rewrite trigger
TAUAnRDM.TAUAnRDMm	1: The simultaneous rewrite trigger signal is generated when the master channel starts counting and the corresponding slave channel is at the peak of a triangular wave
TAUAnRDC.TAUAnRDCm	0: Channel is not monitored for an INTTAUAnIm signal that is used as the simultaneous rewrite trigger. If TAUAnRDS.TAUAnRDSm = 0, the master channel is monitored for the simultaneous rewrite trigger regardless of the value of this bit.

Note If the TAUAnRDS.TAUAnRDSm bit is 1, there must be a channel higher than the master channel to which a simultaneous rewrite trigger signal is generated.

(5) Register settings for slave channel 1

(a) TAUAnCMORm for slave channel 1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn CKS[1:0]		TAUAn CCS[1:0]		TAUAn MAS	TAUAnSTS[2:0]			TAUAn COS[1:0]		-	TAUAnMD[4:1]				TAUAn MD0

Table 12-204 TAUAnCMORm settings for slave channel 1 of the Complementary Modulation Output Function

Bit name	Setting
TAUAnCKS[1:0]	00: Operation clock = CK0 01: Operation clock = CK1 10: Operation clock = CK2 11: Operation clock = CK3 The value of the TAUAnCKS[1:0] bit of the master and slave channels must be identical.
TAUAnCCS[1:0]	11: INTTAUAnIm of the master channel is used as the count clock
TAUAnMAS	0: Channel is a slave channel
TAUAnSTS[2:0]	000: Counter triggered by software trigger 011: Simultaneous rewrite trigger
TAUAnCOS[1:0]	00: Not used, so set to 00
TAUAnMD[4:1]	0011: Event Count Mode
TAUAnMD0	0: INTTAUAnIm not generated at operation start or restart

(b) TAUAnCMURm for slave channel 1

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-														TAUAnTIS[1:0]	

Table 12-205 TAUAnCMURm settings for slave channel 1 of the Complementary Modulation Output Function

Bit name	Setting
TAUAnTIS[1:0]	00: Not used, so set to 00

(c) Channel output mode

Because the channel output mode is not used by slave channel 1 of this function, clear TAUAnTOE.TAUAnTOEm. However, it can be used in Independent Channel Output Mode Controlled by Software.

Caution TAUAnTRC.TAUAnTRCm must be set to 1 to enable slave 1 to be used as the real-time output trigger.

(d) Simultaneous rewrite for slave channel 1

The simultaneous rewrite settings of the master and slave channels must be identical.

Table 12-206 Simultaneous rewrite settings for slave channel 1 of the Complementary Modulation Output Function

Bit name	Setting
TAUAnRDE.TAUAnRDEm	1: Enables simultaneous rewrite
TAUAnRDS.TAUAnRDSm	0: The master channel is monitored for the simultaneous rewrite trigger 1: An upper channel outside the channel group is monitored for the simultaneous rewrite trigger
TAUAnRDM.TAUAnRDMm	1: The simultaneous rewrite trigger signal is generated when the master channel starts counting and the corresponding slave channel is at the peak of a triangular wave
TAUAnRDC.TAUAnRDCm	0: Channel is not monitored for an INTTAUAnIm signal that is used as the simultaneous rewrite trigger. If TAUAnRDS.TAUAnRDSm = 0, the master channel is monitored for the simultaneous rewrite trigger regardless of the value of this bit.

(6) Register settings for slave channels 2, 4, and 6

(a) TAUAnCMORm for slave channels 2, 4, and 6

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn CKS[1:0]		TAUAn CCS[1:0]		TAUAn MAS	TAUAnSTS[2:0]			TAUAn COS[1:0]		-	TAUAnMD[4:1]				TAUAn MD0

Table 12-207 TAUAnCMORm settings for slave channels 2, 4, and 6 of the Complementary Modulation Output Function

Bit name	Setting
TAUAnCKS[1:0]	00: Operation clock = CK0 01: Operation clock = CK1 10: Operation clock = CK2 11: Operation clock = CK3 The value of the TAUAnCKS[1:0] bit of the master and slave channels must be identical.
TAUAnCCS[1:0]	00: Operation clock is used as the count clock
TAUAnMAS	0: Channel is a slave channel
TAUAnSTS[2:0]	111: The up/down output trigger signal of the master channel
TAUAnCOS[1:0]	00: Not used, so set to 00
TAUAnMD[4:1]	1001: Up Down Count Mode
TAUAnMD0	0: INTTAUAnIm not generated at operation start or restart

(b) TAUAnCMURm for slave channels 2, 4, and 6

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-														TAUAnTIS[1:0]	

Table 12-208 TAUAnCMURm settings for slave channels 2, 4, and 6 of the Complementary Modulation Output Function

Bit name	Setting
TAUAnTIS[1:0]	00: Not used, so set to 00

(c) Channel output mode of slave channels 2, 4, and 6

Table 12-209 Control bit settings for Synchronous Channel Output Mode 2 with Complementary Modulation Output

Bit name	Setting
TAUAnTOE.TAUAnTOEm	1: Enables Independent Channel Output Mode
TAUAnTOM.TAUAnTOMm	1: Synchronous channel output
TAUAnTOC.TAUAnTOCm	1: Operation mode 2
TAUAnTOL.TAUAnTOLm	0: Positive logic 1: Inverted logic
TAUAnTDE.TAUAnTDEm	1: Enables dead time operation
TAUAnTDM.TAUAnTDMm	0: Dead time is added upon detection of a duty cycle at the upper even channel
TAUAnTDL.TAUAnTDLm	0: Dead time is added to the positive phase 1: Dead time is added to the negative phase
TAUAnTRE.TAUAnTREm	1: Enables real-time output
TAUAnTRO.TAUAnTROm	0: Real-time output is low 1: Real-time output is high
TAUAnTRC.TAUAnTRCm	0: The upper channel generates the real-time trigger for channel m
TAUAnTME.TAUAnTMEm	0: Disables modulation 1: Enables modulation

Caution For TAUAnTDL.TAUAnTDLm, specify the setting that is opposite that of the odd channel.

(d) Simultaneous rewrite for slave channels 2, 4, and 6

The simultaneous rewrite settings of the master and slave channel must be identical.

Table 12-210 Simultaneous rewrite settings for slave channels 2, 4, and 6 of the Complementary Modulation Output Function

Bit name	Setting
TAUAnRDE.TAUAnRDEm	1: Enables simultaneous rewrite
TAUAnRDS.TAUAnRDSm	0: The master channel is monitored for the simultaneous rewrite trigger 1: An upper channel outside the channel group is monitored for the simultaneous rewrite trigger
TAUAnRDM.TAUAnRDMm	1: The simultaneous rewrite trigger signal is generated when the master channel starts counting and the corresponding slave channel is at the peak of a triangular wave
TAUAnRDC.TAUAnRDCm	0: Channel is not monitored for an INTTAUAnIm signal that is used as the simultaneous rewrite trigger. If TAUAnRDS.TAUAnRDSm = 0, the master channel is monitored for the simultaneous rewrite trigger regardless of the value of this bit.

(7) Register settings for slave channels 3, 5, and 7

(a) TAUAnCMORm for slave channels 3, 5, and 7

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn CKS[1:0]		TAUAn CCS[1:0]		TAUAn MAS	TAUAnSTS[2:0]			TAUAn COS[1:0]		-	TAUAnMD[4:1]				TAUAn MD0

Table 12-211 TAUAnCMORm settings for slave channels 3, 5, and 7 of the Complementary Modulation Output Function

Bit name	Setting
TAUAnCKS[1:0]	00: Operation clock = CK0 01: Operation clock = CK1 10: Operation clock = CK2 11: Operation clock = CK3
TAUAnCCS[1:0]	00: Operation clock is used as the count clock
TAUAnMAS	0: Channel is a slave channel
TAUAnSTS[2:0]	110: Dead time trigger
TAUAnCOS[1:0]	00: Not used, so set to 00
TAUAnMD[4:1]	0100: One count mode
TAUAnMD0	1: Enables start trigger detection during counting

(b) TAUAnCMURm for slave channels 3, 5, and 7

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-														TAUAnTIS[1:0]	

Table 12-212 TAUAnCMURm settings for slave channels 3, 5, and 7 of the Complementary Modulation Output Function

Bit name	Setting
TAUAnTIS[1:0]	00: Not used, so set to 00

(c) Channel output mode of slave channels 3, 5, and 7

Table 12-213 Control bit settings for Synchronous Channel Output Mode 2 with Complementary Modulation Output

Bit name	Setting
TAUAnTOE.TAUAnTOEm	1: Enables Independent Channel Output Mode
TAUAnTOM.TAUAnTOMm	1: Synchronous channel output
TAUAnTOC.TAUAnTOCm	1: Operation mode 2
TAUAnTOL.TAUAnTOLm	0: Positive logic 1: Inverted logic
TAUAnTDE.TAUAnTDEm	1: Enables dead time operation
TAUAnTDM.TAUAnTDMm	0: Dead time is added upon detection of a duty cycle at the upper even channel
TAUAnTDL.TAUAnTDLm	0: Dead time is added to the positive phase 1: Dead time is added to the negative phase
TAUAnTRE.TAUAnTREm	1: Enables real-time output
TAUAnTRO.TAUAnTROm	0: Real-time output is low 1: Real-time output is high
TAUAnTRC.TAUAnTRCm	0: The upper channel generates the real-time trigger for channel m
TAUAnTME.TAUAnTMEm	0: Disables modulation 1: Enables modulation

Caution For TAUAnTDL.TAUAnTDLm, specify the setting that is opposite that of the even channel.

(d) Simultaneous rewrite for slave channels 3, 5, and 7

The simultaneous rewrite settings of the master and slave channel must be identical.

Table 12-214 Simultaneous rewrite settings for slave channels 3, 5, and 7 of the Complementary Modulation Output Function

Bit name	Setting
TAUAnRDE.TAUAnRDEm	1: Enables simultaneous rewrite
TAUAnRDS.TAUAnRDSm	0: The master channel is monitored for the simultaneous rewrite trigger 1: An upper channel outside the channel group is monitored for the simultaneous rewrite trigger
TAUAnRDM.TAUAnRDMm	1: The simultaneous rewrite trigger signal is generated when the master channel starts counting and the corresponding slave channel is at the peak of a triangular wave
TAUAnRDC.TAUAnRDCm	0: Channel is not monitored for an INTTAUAnIm signal that is used as the simultaneous rewrite trigger. If TAUAnRDS.TAUAnRDSm = 0, the master channel is monitored for the simultaneous rewrite trigger regardless of the value of this bit.

(8) Operating Procedure for Complementary Modulation Output Function

Table 12-215 Operating procedure for Complementary Modulation Output Function (1/2)

	Operation	Status of TAUAn
Initial channel setting	Master channel: set the TAUAnCMORm and TAUAnCMURm registers and the channel output mode as described in *(4) "Register settings for the master channel" on page 837* Slave channel 1: set the TAUAnCMORm and TAUAnCMURm registers and the channel output mode as described in *(5) "Register settings for slave channel 1" on page 839* Slave channels 2, 4, and 6: set the TAUAnCMORm and TAUAnCMURm registers and the channel output mode as described in *(6) "Register settings for slave channels 2, 4, and 6" on page 841* Slave channels 3, 5, and 7: set the TAUAnCMORm and TAUAnCMURm registers and the channel output mode as described in *(7) "Register settings for slave channels 3, 5, and 7"* Set the values of the TAUAnCDRm registers of all channels: set the pulse cycle in TAUAnCDRm of the master channel, set the number of interrupts on the master channel to be ignored using TAUAnCDRm of slave channel 1, and set the duty width in TAUAnCDRm of slave channels 2, 4, and 6, and set the dead time delay in slave channels 3, 5, and 7. Set TAUAnTRC.TAUAnTRCm = 1 for slave channel 1	Channel operation is stopped.

Table 12-215 Operating procedure for Complementary Modulation Output Function (2/2)

	Operation	Status of TAUAn
Start operation	Set TAUAnTS.TAUAnTSm of the master and slave channels to 1 simultaneously (for channel restart, only slaves 2 to 7). TAUAnTS.TAUAnTSm is a trigger bit, so it is automatically cleared to 0.	TAUAnTE.TAUAnTEm of the master and slave channels is set to 1 and the counters start to count down.
During operation	TAUAnCDRm, TAUAnTRO.TAUAnTROm, and TAUAnTME.TAUAnTMEm can be changed at any time. TAUAnCNTm and TAUAnRSF.TAUAnRSFm can be read at any time. TAUAnRDT.TAUAnRDTm can be changed during operation.	TAUAnCNTm loads the TAUAnCDRm value of the master channel and slave channels 2 and 7, and then counts down. The TAUAnCDRm value of slave channel 1 is loaded, and a master channel interrupt is awaited. When the counter of the master channel reaches 0000_H: • INTTAUAnIm is generated. • TAUAnCNTm reloads the TAUAnCDRm value, and then continues counting down. • TAUAnCNTm of slave channel 1 reduces by 1 and awaits the next interrupt on the master channel. • TAUAnCNTm of slave channels 2, 4 and 6 counts in the reverse direction. • When the counter of slave channel 1 reaches. 0000_H it awaits the next interrupt from the master channel. When it is detected: - TAUAnCNTm reloads the TAUAnCDRm value and awaits the next interrupt on the master channel. - INTTAUAnIm is generated. - TAUAnTRO.TAUAnTROm can change. • When the counter of slave channels 2, 4, and 6 reaches 0001_H: - INTTAUAnIm is generated. - The PWM output of slave channel m resets. - TAUAnCNTm loads the TAUAnCDRm value of slave channels 3, 5, and 7, and then counts down. • When the counter of slave channels 3, 5, and 7 reaches 0001_H: - INTTAUAnIm is generated. TAUAnTTOUTm of slave channels 2 to 7 outputs a PWM signal, a high signal, or a low signal depending on the value of the real-time output bits (TAUAnTRO.TAUAnTROm), the modulation output bits (TAUAnTME.TAUAnTMEm), and the output level bits (TAUAnTOL.TAUAnTOLm) of a pair of slave channels.
Stop operation	Set TAUAnTT.TAUAnTTm of the master and slave channels to 1 simultaneously. TAUAnTT.TAUAnTTm is a trigger bit, so it is automatically cleared to 0.	TAUAnTE.TAUAnTEm is cleared to 0 and the counter stops. TAUAnCNTm and TAUAnTTOUTm stop and retain their current values.

Restart

(9) Specific timing diagrams

The following settings apply to the timing diagram:

- Slave channels 2 to 7: positive logic (TAUAnTOL.TAUAnTOLm = 0)

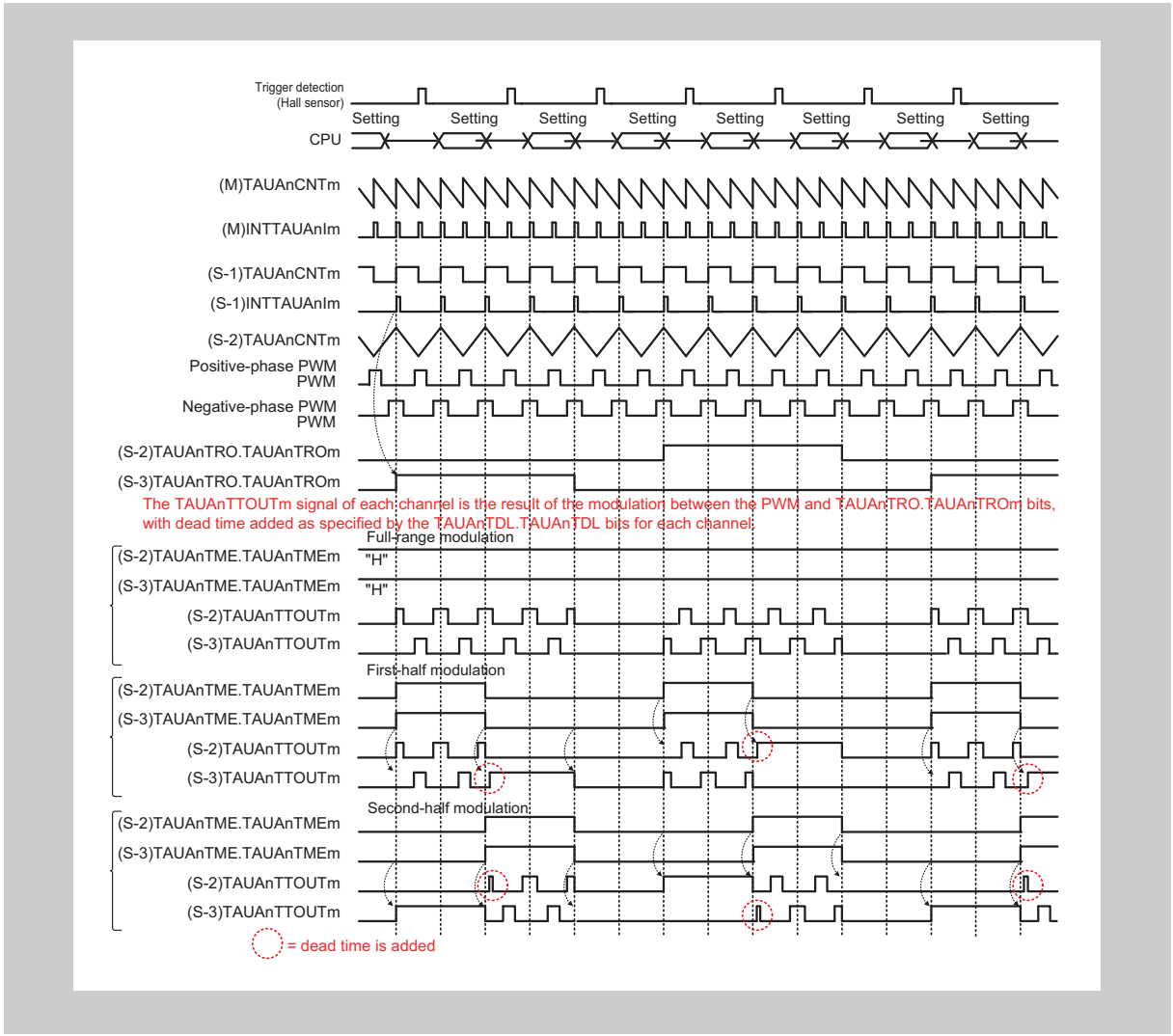


Figure 12-123 Specific timing diagram for Complementary Modulation Output Function

The above timing diagram shows how full modulation, first-half modulation, and second-half modulation can be achieved by modifying the TAUAnTME.TAUAnTMEm bits of the lower slave channels during operation.

The output of the channels 2 and 3 is the modulation of the PWM output and the value of the TAUAnTRO.TAUAnTROm bits. Thus the type of PWM signal (positive or negative) output by an individual channel varies depending on the value of these bits.

The TAUAnTRO.TAUAnTROm, TAUAnTME.TAUAnTMEm, and TAUAnTDL.TAUAnTDLm bit values are specified using software, but a newly specified value is not applied unless an interrupt is generated on slave channel 1.

Note Dead time is added to prevent the positive- and negative-phase PWM edges from changing simultaneously.

The “Setting” symbol indicates the time period during which the values of TAUAnCDRm, TAUAnTME.TAUAnTMEm, TAUAnTRO.TAUAnTROm, and TAUAnTDL.TAUAnTDLm can be changed.

12.26 Other Synchronous Channel Functions

This chapter describes a function that is used to ignore a specified number of interrupts on the master channel.

12.26.1 Interrupt Culling Function

(1) Overview

Summary This function divides the number of interrupts of the master channel by a specified value using a slave channel.

The Interrupt Culling Function is a sub function of the following functions:

- PWM Output Function

  (*12.22.1 "PWM Output Function" on page 720*)

- Triangle PWM Output Function

  (*12.24.1 "Triangle PWM Output Function" on page 776*)

- Triangle PWM Output Function with Dead Time

  (*12.24.2 "Triangle PWM Output Function with Dead Time" on page 787*)

Prerequisites

- Two channels
- The operation mode of the master channel must be set to Interval Timer Mode. Refer to *Table 12-216 "TAUAnCMORm settings for the master channel of the Interrupt Culling Function" on page 853*.
- The operation mode of the slave channel must be set to Event Count Mode. Refer to *Table 12-219 "TAUAnCMORm settings for the slave channel of the Interrupt Culling Function" on page 855*.
- TAUAnTTOUTm is not used for the slave channel of this function

Description The counters (master and slave) are started by setting the channel trigger bit (TAUAnTS.TAUAnTSm) to 1 for both channels. This in turn sets TAUAnTE.TAUAnTEm, enabling count operation. The current value of the data register of the master channel and slave channel (TAUAnCDRm) are loaded to the counter (TAUAnCNTm).

- Master channel:

  When the counter of the master channel reaches 0000_H, INTTAUAnIm is generated and TAUAnCNTm loads the TAUAnCDRm value.

- Slave channel:

  Every time the master channel generates an INTTAUAnIm, the counter of the slave channel reduces by one. When the counter reaches 0000_H, it awaits the next interrupt from the master channel. This causes TAUAnCNTm (slave) to load the value of TAUAnCDRm, and an INTTAUAnIm is generated.

Forced restart is not possible for this function. The counter can be stopped by setting TAUAnTT.TAUAnTTm to 1 for the master and slave channel, which in turn sets TAUAnTE.TAUAnTEm to 0. TAUAnCNTm and TAUAnTTOUTm of master and slave channel stop but retain their values.

Conditions Simultaneous rewrite can be used with this function. Please refer to *12.7 "Simultaneous Rewrite" on page 565*.

(2) Equations

Interrupt divider = TAUAnCDRm (slave channel)

– One INTTAUAnIm is generated for the number of INTTAUAnIm of the master channel defined as TAUAnCDRm (slave channel) + 1.

(3) Block diagram and general timing diagram

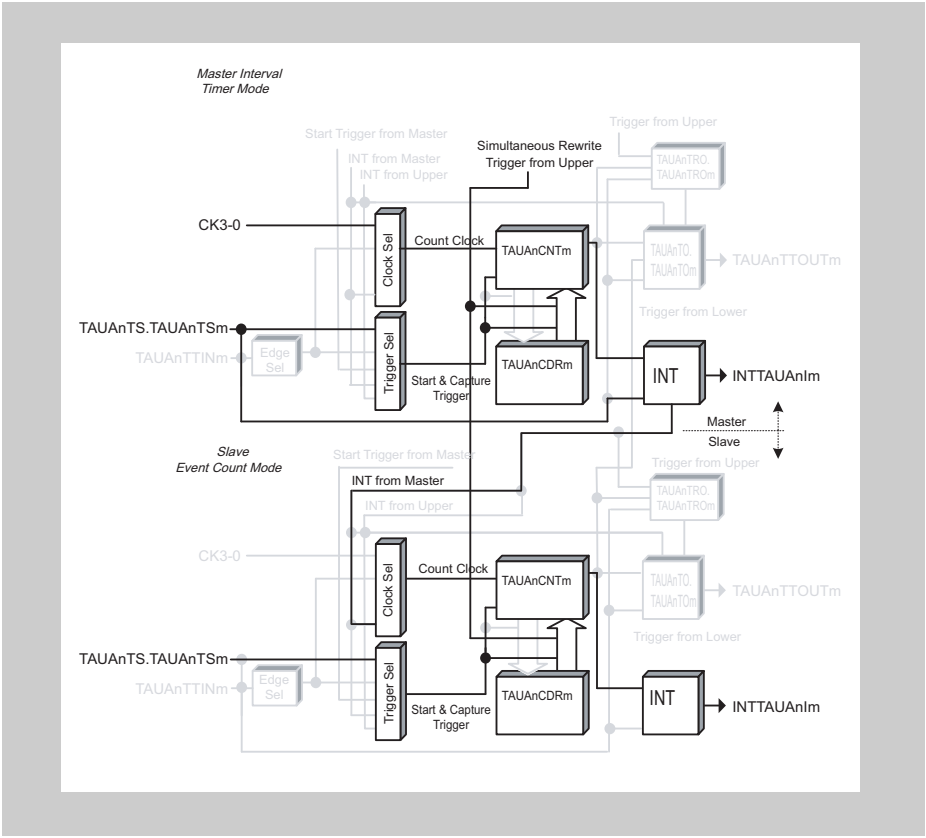


Figure 12-124 Block diagram for Interrupt Culling Function

The following settings apply to the general timing diagram:

Master channel:

- INTTAUAnIm is generated at operation start (TAUAnCMORm.TAUAnMD0 = 1)

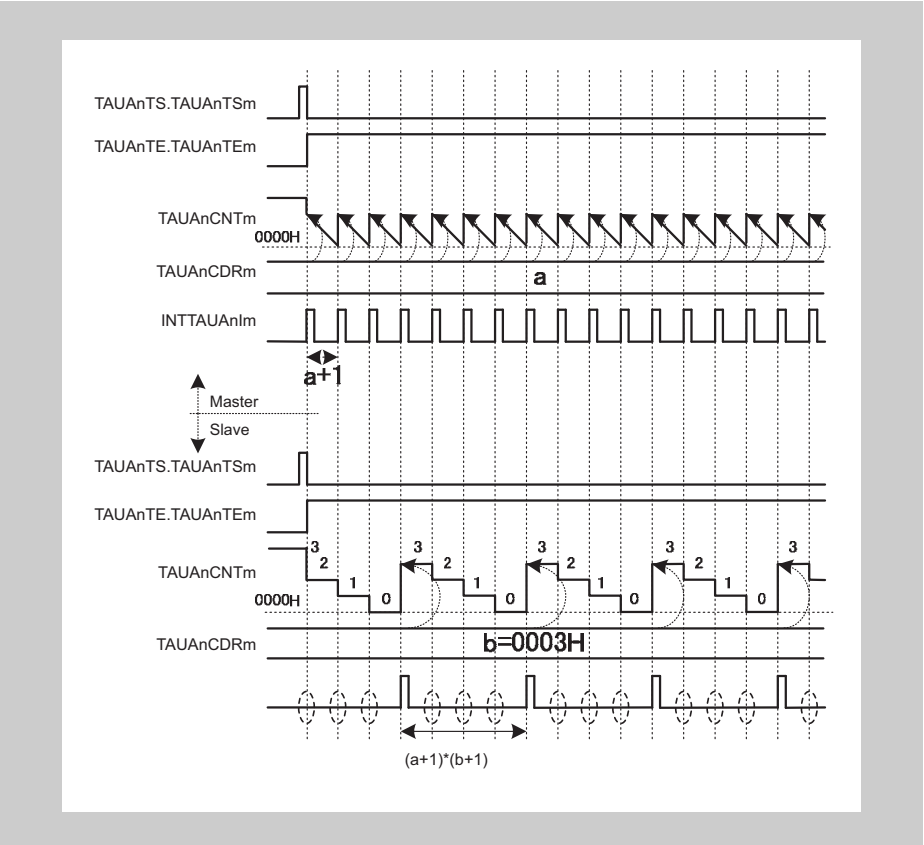


Figure 12-125 General timing diagram for Interrupt Culling Function

The simultaneous rewrite trigger signal is generated when the master channel starts counting and the corresponding slave channel is at the peak of a triangular wave.

(4) Register settings for the master channel

(a) TAUAnCMORm for the master channel

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn CKS[1:0]		TAUAn CCS[1:0]		TAUAn MAS	TAUAnSTS[2:0]			TAUAn COS[1:0]		-	TAUAnMD[4:1]				TAUAn MD0

Table 12-216 TAUAnCMORm settings for the master channel of the Interrupt Culling Function

Bit name	Setting
TAUAnCKS[1:0]	00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 The value of the TAUAnCKS[1:0] bit of the master and slave channel must be identical.
TAUAnCCS[1:0]	00: Operation clock is used as the count clock
TAUAnMAS	1: Channel is master channel
TAUAnSTS[2:0]	000: Counter triggered by software trigger
TAUAnCOS[1:0]	00: Not used, so set to 00
TAUAnMD[4:1]	0000: Interval Timer Mode
TAUAnMD0	0: INTTAUAnIm not output at operation startup 1: Generates INTTAUAnIm at operation startup

(b) TAUAnCMURm for the master channel

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-														TAUAnTIS[1:0]	

Table 12-217 TAUAnCMURm settings for the master channel of the Interrupt Culling Function

Bit name	Setting
TAUAnTIS[1:0]	00: These are not used, so set them to 00.

(c) Channel output mode for the master channel

Because the channel output mode is not used by this function, clear TAUAnTOE.TAUAnTOEn. However, it can be used by other functions or in Independent Channel Output Mode Controlled by Software.

(d) Simultaneous rewrite for the master channel

The simultaneous rewrite settings of the master and slave channel must be identical.

Table 12-218 Simultaneous rewrite settings for the master channel of the Interrupt Culling Function

Bit name	Setting
TAUAnRDE.TAUAnRDEm	1: Enables simultaneous rewrite
TAUAnRDS.TAUAnRDSm	0: The master channel is monitored for the simultaneous rewrite trigger
TAUAnRDM.TAUAnRDMm	0: The simultaneous rewrite trigger signal is generated when the master channel starts counting 1: The simultaneous rewrite trigger signal is generated when the master channel starts counting and the corresponding slave channel is at the peak of a triangular wave
TAUAnRDC.TAUAnRDCm	0: Channel is not monitored for an INTTAUAnIm signal that is used as the simultaneous rewrite trigger. If TAUAnRDS.TAUAnRDSm = 0, the master channel is monitored for the simultaneous rewrite trigger regardless of the value of this bit.

(5) Register settings for the slave channel

(a) TAUAnCMORm for the slave channel

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn CKS[1:0]		TAUAn CCS[1:0]		TAUAn MAS	TAUAnSTS[2:0]			TAUAn COS[1:0]		-	TAUAnMD[4:1]				TAUAn MD0

Table 12-219 TAUAnCMORm settings for the slave channel of the Interrupt Culling Function

Bit name	Setting
TAUAnCKS[1:0]	00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 The value of the TAUAnCKS[1:0] bit of the master and slave channel must be identical.
TAUAnCCS[1:0]	11: INTTAUAnIm of the master channel is used as the count clock
TAUAnMAS	0: Channel is a slave channel
TAUAnSTS[2:0]	000: Counter triggered by software trigger
TAUAnCOS[1:0]	00: Not used, so set to 00
TAUAnMD[4:1]	0011: Event Count Mode
TAUAnMD0	0: INTTAUAnIm not output at operation startup

(b) TAUAnCMURm for the slave channel

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-														TAUAnTIS[1:0]	

Table 12-220 TAUAnCMURm settings for the slave channel of the Interrupt Culling Function

Bit name	Setting
TAUAnTIS[1:0]	00: These are not used, so set them to 00.

(c) Channel output mode for the slave channel

Because the channel output mode is not used by this function, clear TAUAnTOE.TAUAnTOEm. However, it can be used by other functions or in Independent Channel Output Mode Controlled by Software.

(d) Simultaneous rewrite for the slave channel

The simultaneous rewrite settings of the master and slave channel must be identical.

Table 12-221 Simultaneous rewrite settings for the slave channel of the Interrupt Culling Function

Bit name	Setting
TAUAnRDE.TAUAnRDEm	1: Enables simultaneous rewrite
TAUAnRDS.TAUAnRDSm	0: The master channel is monitored for the simultaneous rewrite trigger
TAUAnRDM.TAUAnRDMm	0: The simultaneous rewrite trigger signal is generated when the master channel starts counting 1: The simultaneous rewrite trigger signal is generated when the master channel starts counting and the corresponding slave channel is at the peak of a triangular wave
TAUAnRDC.TAUAnRDCm	0: Channel is not monitored for an INTTAUAnIm signal that is used as the simultaneous rewrite trigger. If TAUAnRDS.TAUAnRDSm = 0, the master channel is monitored for the simultaneous rewrite trigger regardless of the value of this bit.

(6) Operating procedure for Interrupt Culling Function

Table 12-222 Operating procedure for Interrupt Culling Function

	Operation	Status of TAUAn
Initial channel setting	Master channel: set the TAUAnCMORm and TAUAnCMURm registers and the channel output mode as described in *(4) "Register settings for the master channel" on page 853* Slave channel: set the TAUAnCMORm and TAUAnCMURm registers and the channel output mode as described in *(5) "Register settings for the slave channel" on page 855* Set the values of the TAUAnCDRm registers of all channels	Channel operation is stopped.
Start operation	Set TAUAnTS.TAUAnTSm of the master and slave channels to 1 simultaneously. TAUAnTS.TAUAnTSm is a trigger bit, so it is automatically cleared to 0.	TAUAnTE.TAUAnTEm (master and slave channels) is set to 1 and the counters of the master and slave channels start. INTTAUAnIm is generated on the master channel.
During operation	TAUAnCDRm can be changed at any time. TAUAnCNTm and TAUAnRSF.TAUAnRSFm can be read at any time. TAUAnRDT.TAUAnRDTm can be changed during operation.	TAUAnCNTm of the master channel loads TAUAnCDRm, and then counts down. When the counter reaches 0000_H: • INTTAUAnIm (master) is generated. • TAUAnCNTm (master) loads the TAUAnCDRm value, and then continues count operation. • TAUAnCNTm of the slave channel counts down each time INTTAUAnIm is detected on the master channel. When TAUAnCNTm of the slave reaches 0000_H: • INTTAUAnIm (slave) is generated.
Stop operation	Set TAUAnTT.TAUAnTTm of the master and slave channels to 1 simultaneously. TAUAnTT.TAUAnTTm is a trigger bit, so it is automatically cleared to 0.	TAUAnTE.TAUAnTEm is cleared to 0 and the counter stops. TAUAnCNTm and TAUAnTTOUTm stop and retain their current values.

Restart

(7) Specific timing diagrams

(a) Number of interrupts (master) = number of interrupts (slave)

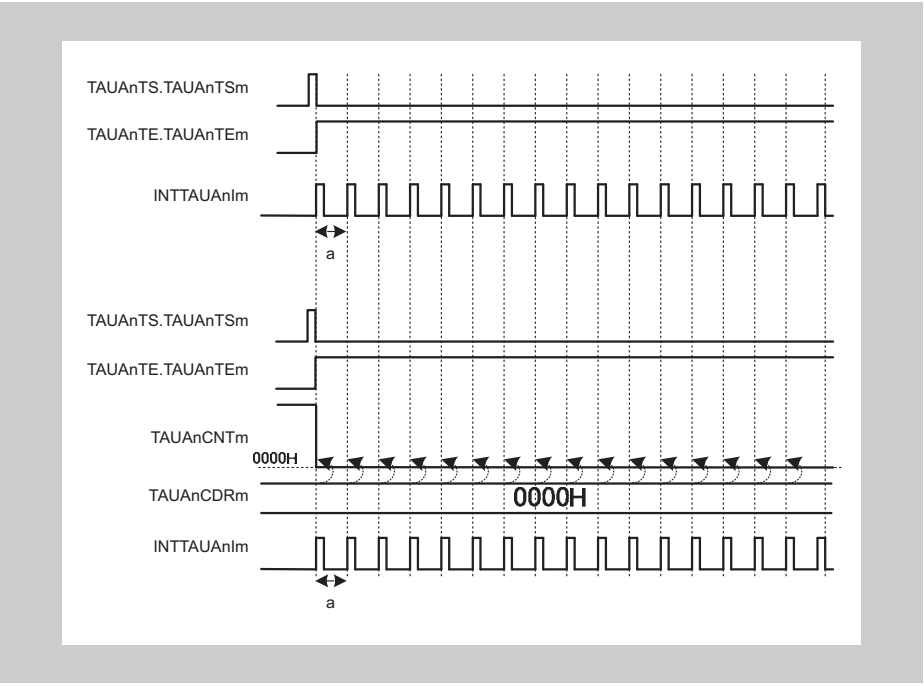


Figure 12-126 TAUAnCDRm (slave) = 0000_H

- If TAUAnCDRm is 0000_H, the TAUAnCDRm value of the slave channel is loaded to TAUAnCNTm each time INTTAUAnIm is detected on the master channel. In other words, TAUAnCNTm is always read as 0000_H.
- Therefore, the slave channel generates an interrupt at the same time as an interrupt is generated on the master channel.

12.27 Registers

This section describes all TAUA registers.

12.27.1 TAUAn registers overview

The TAUAn is controlled and operated by the registers in the following table. Where there is one register per channel, this is indicated by an “m”, where m runs from 0 to 15.

Table 12-223 TAUAn registers overview (1/2)

Register name	Shortcut	Address
TAUAn prescaler registers		
TAUAn prescaler clock select register	TAUAnTPS	<TAUAn_base_OS> + 240_H
TAUAn prescaler baud rate setting register	TAUAnBRS	<TAUAn_base_OS> + 244_H
TAUAn control registers		
TAUAn channel data register m	TAUAnCDRm	<TAUAn_base_USER> + m x 4_H
TAUAn channel counter register m	TAUAnCNTm	<TAUAn_base_USER> + 80_H + m x 4_H
TAUAn channel mode OS register m	TAUAnCMORm	<TAUAn_base_OS> + 200_H + m x 4_H
TAUAn channel mode user register m	TAUAnCMURm	<TAUAn_base_USER> + $C0_H$ + m x 4_H
TAUAn channel status register m	TAUAnCSRm	<TAUAn_base_USER> + 140_H + m x 4_H
TAUAn channel status clear trigger register m	TAUAnCSCm	<TAUAn_base_USER> + 180_H + m x 4_H
TAUAn channel start trigger register	TAUAnTS	<TAUAn_base_USER> + $1C4_H$
TAUAn channel enable status register	TAUAnTE	<TAUAn_base_USER> + $1C0_H$
TAUAn channel stop trigger register	TAUAnTT	<TAUAn_base_USER> + $1C8_H$
TAUAn output registers		
TAUAn channel output enable register	TAUAnTOE	<TAUAn_base_USER> + $5C_H$
TAUAn channel output register	TAUAnTO	<TAUAn_base_USER> + 58_H
TAUAn channel output mode register	TAUAnTOM	<TAUAn_base_OS> + 248_H
TAUAn channel output configuration register	TAUAnTOC	<TAUAn_base_OS> + $24C_H$
TAUAn channel output active level register	TAUAnTOL	<TAUAn_base_USER> + 040_H
TAUAn channel dead time output enable register	TAUAnTDE	<TAUAn_base_OS> + 250_H
TAUAn channel dead time output mode register	TAUAnTDM	<TAUAn_base_OS> + 254_H
TAUAn channel dead time output level register	TAUAnTDL	<TAUAn_base_USER> + 54_H
TAUAn channel real-time output register	TAUAnTRO	<TAUAn_base_USER> + $4C_H$
TAUAn channel real-time output enable register	TAUAnTRE	<TAUAn_base_OS> + 258_H
TAUAn channel real-time output control register	TAUAnTRC	<TAUAn_base_OS> + $25C_H$
TAUAn channel modulation output enable register	TAUAnTME	<TAUAn_base_USER> + 50_H
TAUAn reload data registers		
TAUAn channel reload data enable register	TAUAnRDE	<TAUAn_base_OS> + 260_H
TAUAn channel reload data mode register	TAUAnRDM	<TAUAn_base_OS> + 264_H

Table 12-223 TAUAn registers overview (2/2)

Register name	Shortcut	Address
TAUAn channel reload data control CH select register	TAUAnRDS	<TAUAn_base_OS> + 268_H
TAUAn channel reload data control register	TAUAnRDC	<TAUAn_base_OS> + $26C_H$
TAUAn channel reload data trigger register	TAUAnRDT	<TAUAn_basebase_USER> + 44_H
TAUAn channel reload status register	TAUAnRSF	<TAUAn_base_USER> + 48_H
TAUAn DMA window registers		
TAUAn DMA window address setting register 0	TAUAnDAS0	<TAUAn_base_OS> + 270_H
TAUAn DMA window address setting register 1	TAUAnDAS1	<TAUAn_base_OS> + 274_H
TAUAn DMA window address setting register 2	TAUAnDAS2	<TAUAn_base_OS> + 278_H
TAUAn DMA window address setting register 3	TAUAnDAS3	<TAUAn_base_OS> + $27C_H$
TAUAn DMA window address setting register 4	TAUAnDAS4	<TAUAn_base_OS> + 280_H
TAUAn DMA window address setting register 5	TAUAnDAS5	<TAUAn_base_OS> + 284_H
TAUAn DMA window address setting register 6	TAUAnDAS6	<TAUAn_base_OS> + 288_H
TAUAn DMA window address setting register 7	TAUAnDAS7	<TAUAn_base_OS> + $28C_H$
TAUAn DMA window register 0	TAUAnDWR0	<TAUAn_base_USER> + 100_H
TAUAn DMA window register 1	TAUAnDWR1	<TAUAn_base_USER> + 104_H
TAUAn DMA window register 2	TAUAnDWR2	<TAUAn_base_USER> + 108_H
TAUAn DMA window register 3	TAUAnDWR3	<TAUAn_base_USER> + $10C_H$
TAUAn DMA window register 4	TAUAnDWR4	<TAUAn_base_USER> + 110_H
TAUAn DMA window register 5	TAUAnDWR5	<TAUAn_base_USER> + 114_H
TAUAn DMA window register 6	TAUAnDWR6	<TAUAn_base_USER> + 118_H
TAUAn DMA window register 7	TAUAnDWR7	<TAUAn_base_USER> + $11C_H$
TAUAn DMA window register 8	TAUAnDWR8	<TAUAn_base_USER> + 120_H
TAUAn DMA window register 9	TAUAnDWR9	<TAUAn_base_USER> + 124_H
TAUAn DMA window register 10	TAUAnDWR10	<TAUAn_base_USER> + 128_H
TAUAn DMA window register 11	TAUAnDWR11	<TAUAn_base_USER> + $12C_H$
TAUAn DMA window register 12	TAUAnDWR12	<TAUAn_base_USER> + 130_H
TAUAn DMA window register 13	TAUAnDWR13	<TAUAn_base_USER> + 134_H
TAUAn DMA window register 14	TAUAnDWR14	<TAUAn_base_USER> + 138_H
TAUAn DMA window register 15	TAUAnDWR15	<TAUAn_base_USER> + $13C_H$

Note The <TAUAn_base_OS> and <TAUAn_base_USER> addresses of the registers are defined in the first section of this chapter under the keyword “Register addresses”.

12.27.2 TAUAn prescaler registers details

(1) TAUAnTPS - TAUAn prescaler clock select register

This register specifies the PCLK prescalers for clocks CK0, CK1, CK2, and CK3_PRE for all channels. CK3 is generated by dividing CK3_PRE by the factor specified in TAUAnBRS.

Access This register can be read/written in 16-bit units.

Address <TAUAn_base_OS> + 240_H

Initial Value $FFFF_H$. This register is initialized by any reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnPRS3[3:0]				TAUAnPRS2[3:0]				TAUAnPRS1[3:0]				TAUAnPRS0[3:0]			
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

Table 12-224 TAUAnTPS register contents (1/4)

Bit position	Bit name	Function
15 to 12	TAUAn PRS3[3:0]	Specifies the CK3_PRE clock. Clock CK3_PRE is the input clock of the BRG unit. The BRG unit supplies the CK3 operation clock for all channels. **PRS3[3:0] / CK3_PRE clock** 0000_B PCLK/2^0 0001_B PCLK/2^1 0010_B PCLK/2^2 0011_B PCLK/2^3 0100_B PCLK/2^4 0101_B PCLK/2^5 0110_B PCLK/2^6 0111_B PCLK/2^7 1000_B PCLK/2^8 1001_B PCLK/2^9 1010_B PCLK/2^{10} 1011_B PCLK/2^{11} 1100_B PCLK/2^{12} 1101_B PCLK/2^{13} 1110_B PCLK/2^{14} 1111_B PCLK/2^{15} These bits can only be rewritten when all counters using CK3 are stopped (TAUAnTE.TAUAnTEm = 0).

Table 12-224 TAUAnTPS register contents (2/4)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>11 to 8</td><td>TAUAn
PRS2[3:0]</td><td>Specifies the CK2 clock.
<table>
<tr><th>PRS2[3:0]</th><th>CK2 clock</th></tr>
<tr><td>0000_B</td><td>PCLK/2^0</td></tr>
<tr><td>0001_B</td><td>PCLK/2^1</td></tr>
<tr><td>0010_B</td><td>PCLK/2^2</td></tr>
<tr><td>0011_B</td><td>PCLK/2^3</td></tr>
<tr><td>0100_B</td><td>PCLK/2^4</td></tr>
<tr><td>0101_B</td><td>PCLK/2^5</td></tr>
<tr><td>0110_B</td><td>PCLK/2^6</td></tr>
<tr><td>0111_B</td><td>PCLK/2^7</td></tr>
<tr><td>1000_B</td><td>PCLK/2^8</td></tr>
<tr><td>1001_B</td><td>PCLK/2^9</td></tr>
<tr><td>1010_B</td><td>PCLK/2^{10}</td></tr>
<tr><td>1011_B</td><td>PCLK/2^{11}</td></tr>
<tr><td>1100_B</td><td>PCLK/2^{12}</td></tr>
<tr><td>1101_B</td><td>PCLK/2^{13}</td></tr>
<tr><td>1110_B</td><td>PCLK/2^{14}</td></tr>
<tr><td>1111_B</td><td>PCLK/2^{15}</td></tr>
</table>
These bits can only be rewritten when all counters using CK2 are stopped (TAUAnTE.TAUAnTEm = 0).</td></tr>
</table>

Table 12-224 TAUAnTPS register contents (3/4)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>7 to 4</td><td>TAUAn
PRS1[3:0]</td><td>Specifies the CK1 clock.
<table>
<tr><th>PRS1[3:0]</th><th>CK1 clock</th></tr>
<tr><td>0000_B</td><td>PCLK/2^0</td></tr>
<tr><td>0001_B</td><td>PCLK/2^1</td></tr>
<tr><td>0010_B</td><td>PCLK/2^2</td></tr>
<tr><td>0011_B</td><td>PCLK/2^3</td></tr>
<tr><td>0100_B</td><td>PCLK/2^4</td></tr>
<tr><td>0101_B</td><td>PCLK/2^5</td></tr>
<tr><td>0110_B</td><td>PCLK/2^6</td></tr>
<tr><td>0111_B</td><td>PCLK/2^7</td></tr>
<tr><td>1000_B</td><td>PCLK/2^8</td></tr>
<tr><td>1001_B</td><td>PCLK/2^9</td></tr>
<tr><td>1010_B</td><td>PCLK/2^{10}</td></tr>
<tr><td>1011_B</td><td>PCLK/2^{11}</td></tr>
<tr><td>1100_B</td><td>PCLK/2^{12}</td></tr>
<tr><td>1101_B</td><td>PCLK/2^{13}</td></tr>
<tr><td>1110_B</td><td>PCLK/2^{14}</td></tr>
<tr><td>1111_B</td><td>PCLK/2^{15}</td></tr>
</table>
These bits can only be rewritten when all counters using CK1 are stopped (TAUAnTE.TAUAnTEm = 0).</td></tr>
</table>

Table 12-224 TAUAnTPS register contents (4/4)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>3 to 0</td><td>TAUAn
PRS0[3:0]</td><td>Specifies the CK0 clock.
<table>
<tr><th>PRS0[3:0]</th><th>CK0 clock</th></tr>
<tr><td>0000_B</td><td>PCLK/2^0</td></tr>
<tr><td>0001_B</td><td>PCLK/2^1</td></tr>
<tr><td>0010_B</td><td>PCLK/2^2</td></tr>
<tr><td>0011_B</td><td>PCLK/2^3</td></tr>
<tr><td>0100_B</td><td>PCLK/2^4</td></tr>
<tr><td>0101_B</td><td>PCLK/2^5</td></tr>
<tr><td>0110_B</td><td>PCLK/2^6</td></tr>
<tr><td>0111_B</td><td>PCLK/2^7</td></tr>
<tr><td>1000_B</td><td>PCLK/2^8</td></tr>
<tr><td>1001_B</td><td>PCLK/2^9</td></tr>
<tr><td>1010_B</td><td>PCLK/2^{10}</td></tr>
<tr><td>1011_B</td><td>PCLK/2^{11}</td></tr>
<tr><td>1100_B</td><td>PCLK/2^{12}</td></tr>
<tr><td>1101_B</td><td>PCLK/2^{13}</td></tr>
<tr><td>1110_B</td><td>PCLK/2^{14}</td></tr>
<tr><td>1111_B</td><td>PCLK/2^{15}</td></tr>
</table>
These bits can only be rewritten when all counters using CK0 are stopped (TAUAnTE.TAUAnTEm = 0).</td></tr>
</table>

Note The TAUAn clock input PCLK is specified in the first section of this chapter under the keyword “Clock supply”.

(2) TAUAnBRS - TAUAn prescaler baud rate setting register

This register specifies the division factor of prescaler clock CK3.

CK3 is generated by dividing CK3_PRE by the factor specified in this register plus one. The PCLK prescaler for CK3_PRE is specified in TAUAnTPS.TAUAnPRS3[3:0].

Access This register can be read/written in 8-bit units.

Address <TAUAn_base_OS> + 244_H

Initial Value 00_H. This register is initialized by any reset.

7	6	5	4	3	2	1	0
TAUAnBRS[07:00]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 12-225 TAUAnBRS register contents

Bit position	Bit name	Function
7 to 0	TAUAn BRS[07:00]	Specifies the CK3_PRE clock division factor for generating CK3:

TAUAnBRS[07:00]	CK3 clock
$0000\ 0000_B$	CK3_PRE / 1
$0000\ 0001_B$	CK3_PRE / 2
$0000\ 0010_B$	CK3_PRE / 3
$0000\ 0011_B$	CK3_PRE / 4
...	...
$1111\ 1110_B$	CK3_PRE / 255
$1111\ 1111_B$	CK3_PRE / 256

12.27.3 TAUAn control registers details

(1) TAUAnCDRm - TAUAn channel data register

This register functions either as a compare register or as a capture register, depending on the operation mode specified in TAUAnCMORm.TAUAnMD[4:1].

Access This register can be read/written in 16-bit units.

- In capture mode, only reading is possible. Write operation is ignored.
- In compare mode, reading and writing is possible.

Address <TAUAn_base_USER> + m x 4_H

Initial Value 0000_H. This register is initialized by any reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnCDR[15:0]															
R/W															

Table 12-226 TAUAnCDRm register contents

Bit position	Bit name	Function
15 to 0	TAUAn CDR[15:0]	Data register for the capture/compare value.

(2) TAUAnCNTm - TAUAn channel counter register

This register is the channel m counter register.

Access This register can be read in 16-bit units.

Address <TAUAn_base_USER> + 80_H + m x 4_H

Initial Value 0000_H or $FFFF_H$ The initial value depends on the operation mode, see *Table 12-228 "TAUAnCNTm read values after the counter is re-enabled" on page 867*. This register is initialized by any reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAnCNT[15:0]															
R															

Table 12-227 TAUAnCNTm register contents

Bit position	Bit name	Function
15 to 0	TAUAn CNT[15:0]	16-bit counter value.

The read value depends on the counter, the operation mode change, and the values of the TAUAnTS.TAUAnTSm and TAUAnTT.TAUAnTTm bits.

The *initial* counter read value depends on the operation mode and how the counter was stopped:

- by a reset
- by a counter stop trigger (TAUAnTT.TAUAnTTm = 1)

The following table lists the initial counter read values after the counter has stopped (TAUAnTE.TAUAnTEm = 0) and re-enabled (TAUAnTS.TAUAnTSm = 1).

The table also contains the counter read value one count after the counter is enabled (TAUAnTS.TAUAnTSm = 1) for modes where the counter waits for a start trigger.

Table 12-228 TAUAnCNTm read values after the counter is re-enabled (1/2)

Mode name	Count method (up/down)	TAUAnCNTm value		
		After reset	After stop trigger	After one count
Interval Timer mode	Count down	$FFFF_H$	Stop value	-
Judge mode	Count down	$FFFF_H$	Stop value	-
Capture mode	Count up	0000_H	Stop value	-
Event Count mode	Count down	$FFFF_H$	Stop value	-
One Count mode	Count down	$FFFF_H$	Stop value	$FFFF_H$
Capture & One Count mode	Count up	0000_H	Stop value	Captured value + 1 (TAUAnCDRm)
Judge & One Count mode	Count down	$FFFF_H$	Stop value	TAUAnCNTm value - 1
Up Down Count mode	Count up/down	$FFFF_H$	Stop value	-
Pulse One Count mode	Count down	$FFFF_H$	Stop value	0000_H

Table 12-228 TAUAnCNTm read values after the counter is re-enabled (2/2)

Mode name	Count method (up/down)	TAUAnCNTm value		
		After reset	After stop trigger	After one count
Count Capture Mode	Count up	0000_H	Stop value	-
Gate Count Mode	Count down	$FFFF_H$	Stop value	Stop value
Capture & Gate Count Mode	Count up	0000_H	Stop value	Stop value

Note If the operation mode is changed while the counter is stopped, the initial counter value after counter restart is undefined. The operation mode is changed by register TAUAnCMORm.TAUAnMD[4:1].

(3) TAUAnCMORm - TAUAn channel mode OS register

This register controls channel m operation.

Access This register can be read or written in 16-bit units. Writing is only possible while the counter is stopped (TAUAnTE.TAUAnTEm = 0).

Address <TAUAn_base_OS> + 200_H + m x 4_H

Initial Value 0000_H. This register is initialized by any reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn CKS[1:0]		TAUAn CCS[1:0]		TAUAn MAS	TAUAnSTS[2:0]			TAUAn COS[1:0]		-	TAUAnMD[4:0]				
R/W		R/W		R/W	R/W			R/W		R	R/W				

Table 12-229 TAUAnCMORm register contents (1/4)

Bit position	Bit name	Function
15,14	TAUAn CKS[1:0]	Selects the operation clock. The operation clock is used for the TAUAnTTINm input edge detection circuit. It can also be used as the count clock depending on bits TAUAnCMORm.TAUAnCCS[1:0]. **TAUAn CKS1 / TAUAn CKS0 / Selected operation clock** 0 / 0 / CK0 0 / 1 / CK1 1 / 0 / CK2 1 / 1 / CK3
13,12	TAUAn CCS[1:0]	Selects the count clock for TAUAnCNTm counter: **TAUAn CCS1 / TAUAn CCS0 / Selected count clock** 0 / 0 / Operation clock as specified by TAUAnCMORm.TAUAnCKS[1:0]. 0 / 1 / Valid edge of TAUAnTTINm input signal 1 / 0 / Setting prohibited 1 / 1 / INTTAUAnIm signal of the master channel
11	TAUAnMAS	Specifies the channel as master or slave channel during synchronous channel operation: 0: Slave 1: Master This bit is only valid for even channels (CHm_even). For odd channels (CHm_odd), it is fixed to 0.

Table 12-229 TAUAnCMORm register contents (2/4)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>10 to 8</td><td>TAUAn STS[2:0]</td><td>Selects the external start trigger:
<table>
<tr><th>TAUAn STS2</th><th>TAUAn STS1</th><th>TAUAn STS0</th><th>Description</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>Software trigger</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>Valid edge of the TAUAnTTINm input signal. TAUAnCMURm.TAUAnTIS[1:0] specifies the valid edge.</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>Valid edge of the TAUAnTTINm input signal is the start trigger and the reverse edge is the stop (capture) trigger</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>Setting prohibited</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>INTTAUAnI of the master channel</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>INTTAUAnI of the upper channel (m-1), regardless of the master setting</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>Dead-time output signal of the TAUAnTTOUTm generation unit</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>Up/down output trigger signal TAUAnTUDSm of the master channel.</td></tr>
</table></td></tr>
</table>

Table 12-229 TAUAnCMORm register contents (3/4)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>7, 6</td><td>TAUAn COS[1:0]</td><td>Specifies when the capture register TAUAnCDRm and the overflow flag TAUAnCSRm.TAUAnOVF of channel m are updated.
These bits are only valid if channel m is in capture mode.
<table>
<tr><th>TAUAn COS1</th><th>TAUAn COS0</th><th>TAUAnCDRm</th><th>TAUAnCSRm.TAUAnOVF</th></tr>
<tr><td>0</td><td>0</td><td rowspan="2">Updated upon detection of a TAUAnTTINm input valid edge.</td><td>Updated (cleared or set) upon detection of a TAUAnTTINm input valid edge:
• If a counter overflow has occurred since the last valid edge detection, TAUAnCSRm.TAUAnOVF is set.
• If no counter overflow has occurred since the last valid edge detection, TAUAnCSR.OVF is cleared.</td></tr>
<tr><td>0</td><td>1</td><td>Set upon counter overflow and cleared by setting TAUAnCSCm.TAUAnCLOV.</td></tr>
<tr><td>1</td><td>0</td><td rowspan="2">Updated upon detection of a TAUAnTTINm input valid edge and upon counter overflow:
• TAUAnTTINm input valid edge: Counter value is written to TAUAnCDRm
• Overflow: $FFFF_H$ is written to TAUAnCDRm. The next TAUAnTTINm input valid edge detection is ignored.</td><td>Not set.</td></tr>
<tr><td>1</td><td>1</td><td>Set upon counter overflow and cleared by setting TAUAnCSCm.TAUAnCLOV.</td></tr>
</table></td></tr>
</table>

Table 12-229 TAUAnCMORm register contents (4/4)

Bit position	Bit name	Function
4 to 0	TAUAn MD[4:0]	Specifies the operation mode.

TAUAn MD4	TAUAn MD3	TAUAn MD2	TAUAn MD1	TAUAn MD0	Description
0	0	0	0	1/0	Interval Timer mode
0	0	0	1	1/0	Judge mode
0	0	1	0	1/0	Capture mode
0	0	1	1	0	Event Count mode
0	1	0	0	1/0	One Count mode
0	1	0	1	1/0	Setting prohibited
0	1	1	0	0	Capture & One Count mode
0	1	1	1	1/0	Judge & One Count mode
1	0	0	0	0	Setting prohibited
1	0	0	1	0	Up Down Count mode
1	0	1	0	1/0	Pulse One Count mode
1	0	1	1	1/0	Count Capture mode
1	1	0	0	0	Gate Count mode
1	1	0	1	0	Capture & Gate Count mode

Mode	Role of the TAUAnMD0 bit
Interval Timer mode Capture mode Count Capture mode	Specifies whether the INTTAUAnIm signal is output when the counter starts counting (when the start trigger is input). 0: No INTTAUAnIm generated 1: INTTAUAnIm generated
Event Count mode Up Down Count mode	This bit must be set to 0.
One Count mode Gate Count mode Pulse One Count mode	Enables/disables start trigger detection during counting. 0: Disabled 1: Enabled
Capture & One Count mode Capture & Gate Count mode	This bit must be set to 0.
Judge mode Judge One Count mode	Specifies when INTTAUAnIm is generated. 0: When TAUAnCNTm ≤ TAUAnCDRm 1: When TAUAnCNTm > TAUAnCDRm

(4) TAUAnCMURm - TAUAn channel mode user register

This register specifies the type of valid edge detection used for the TAUAnTTINm input.

Access This register can be read/written in 8-bit units.

Address <TAUAn_base_USER> + $C0_H$ + m x 4_H

Initial Value 00_H. This register is initialized by any reset.

7	6	5	4	3	2	1	0
-	-	-	-	-	-	TAUAnTIS[1:0]	
R	R	R	R	R	R	R/W	R/W

Table 12-230 TAUAnCMURm register contents

Bit position	Bit name	Function
1, 0	TAUAn TIS[1:0]	Specifies the valid edge of the TAUAnTTINm input: • Edge detection for TAUAnTTINm input signals is performed based on the operation clock selected by TAUAnCMORm.TAUAnCKS[1:0].

TAUAn TIS1	TAUAn TIS0	Description
0	0	Falling edge
0	1	Rising edge
1	0	Rising and falling edges (low-width measurement selection). Start trigger: falling edge Stop trigger (capture): rising edge
1	1	Rising and falling edges (high-width measurement selection). Start trigger: rising edge Stop trigger (capture): falling edge

(5) TAUAnCSRm - TAUAn channel status register

This register indicates the count direction and the overflow status of channel m's counter.

Access This register can be read in 8-bit units.

Address <TAUAn_base_USER> + 140_H + m x 4_H

Initial Value 00_H. This register is initialized by any reset.

7	6	5	4	3	2	1	0
-	-	-	-	-	-	TAUAnCSF	TAUAnOVF
R	R	R	R	R	R	R	R

Table 12-231 TAUAnCSRm register contents

Bit position	Bit name	Function
1	TAUAn CSF	Indicates the count direction: 0: Counts up 1: Counts down The read value of this bit is only valid in the following mode: • Up Down Count mode
0	TAUAn OVF	Indicates the counter overflow status: 0: No overflow occurred 1: Overflow occurred This bit is only used in the following modes: • Capture mode • Capture & One Count mode • Count Capture mode • Capture & Gate Count mode The function of this bit depends on the setting of control bits TAUAnCMORm.TAUAnCOS[1:0].

(6) TAUAnCSCm - TAUAn channel status clear register

This register is a trigger register for clearing the overflow flag TAUAnCSRm.TAUAnOVF of a channel m.

Access This register can be written in 8-bit units. It is always read as 00_H.

Address <TAUAn_base_USER> + 180_H + m x 4_H

Initial Value 00_H. This register is initialized by any reset.

7	6	5	4	3	2	1	0
-	-	-	-	-	-	0	TAUAnCLOV
R	R	R	R	R	R	R	W

Table 12-232 TAUAnCSCm register contents

Bit position	Bit name	Function
0	TAUAnCLOV	0: No function 1: Clears the overflow flag TAUAnCSRm.TAUAnOVF

(7) TAUAnTS - TAUAn channel start trigger register

This register enables the counter for each channel.

Access This register can be written in 16-bit units. It is always read as 0000_H.

Address <TAUAn_base_USER> + $1C4_H$

Initial Value 0000_H. This register is initialized by any reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn TS15	TAUAn TS14	TAUAn TS13	TAUAn TS12	TAUAn TS11	TAUAn TS10	TAUAn TS09	TAUAn TS08	TAUAn TS07	TAUAn TS06	TAUAn TS05	TAUAn TS04	TAUAn T03	TAUAn TS02	TAUAn TS01	TAUAn TS00
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 12-233 TAUAnTS register contents

Bit position	Bit name	Function
15 to 0	TAUAnTSm	Enables the counter for channel m: 0: No function 1: Enables the counter and sets TAUAnTE.TAUAnTEm = 1. TAUAnTE.TAUAnTEm = 1 only *enables* counter. Whether the counter *starts* depends on the selected operation mode.

(8) TAUAnTE - TAUAn channel enable status register

This register indicates whether counter is enabled or disabled.

Access This register can be read in 16-bit units.

Address <TAUAn_base_USER> + $1C0_H$

Initial Value 0000_H. This register is initialized by any reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn TE15	TAUAn TE14	TAUAn TE13	TAUAn TE12	TAUAn TE11	TAUAn TE10	TAUAn TE09	TAUAn TE08	TAUAn TE07	TAUAn TE06	TAUAn TE05	TAUAn TE04	TAUAn TE03	TAUAn TE02	TAUAn TE01	TAUAn TE00
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12-234 TAUAnTE register contents

Bit position	Bit name	Function
15 to 0	TAUAnTEm	Indicates whether counter for channel m is enabled or disabled: 0: Counter disabled 1: Counter enabled This bit is set when TAUAnTSSTm (the synchronous channel start trigger signal) trigger input is detected or when TAUAnTS.TAUAnTSm is set. Setting TAUAnTT.TAUAnTTm to 1 resets this bit to 0.

(9) TAUAnTT - TAUAn channel stop trigger register

This register stops the counter for each channel.

Access This register can be written in 16-bit units. It is always read as 0000_H.

Address <TAUAn_base_USER> + $1C8_H$

Initial Value 0000_H.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn TT15	TAUAn TT14	TAUAn TT13	TAUAn TT12	TAUAn TT11	TAUAn TT10	TAUAn TT09	TAUAn TT08	TAUAn TT07	TAUAn TT06	TAUAn TT05	TAUAn TT04	TAUAn TT03	TAUAn TT02	TAUAn TT01	TAUAn TT00
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 12-235 TAUAnTT register contents

Bit position	Bit name	Function
15 to 0	TAUAnTTm	Stops the counter of channel m: 0: No function 1: Stops the counter and reset TAUAnTE.TAUAnTEm. TAUAnCNTm, TAUAnTO.TAUAnTOm, and TAUAnTTOUTm retain their values from before the counter stopped.

12.27.4 TAUAn output registers details

(1) TAUAnTOE - TAUAn channel output enable register

This register enables and disables Independent Channel Output Mode Controlled by Software.

Access This register can be read/written in 16-bit units.

Address <TAUAn_base_USER> + $5C_H$

Initial Value 0000_H. This register is initialized by any reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn TOE15	TAUAn TOE14	TAUAn TOE13	TAUAn TOE12	TAUAn TOE11	TAUAn TOE10	TAUAn TOE09	TAUAn TOE08	TAUAn TOE07	TAUAn TOE06	TAUAn TOE05	TAUAn TOE04	TAUAn TOE03	TAUAn TOE02	TAUAn TOE01	TAUAn TOE00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 12-236 TAUAnTOE register contents

Bit position	Bit name	Function
15 to 0	TAUAnTOEm	Enables/disables Independent Channel Output Mode Controlled by Software: 0: Enables Independent Channel Output Mode Controlled by Software 1: Disables Independent Channel Output Mode Controlled by Software

(2) TAUAnTOM - TAUAn channel output mode register

This register specifies the output mode of each channel.

Access This register can be read/written in 16-bit units. Writing is only possible while the counter is stopped (TAUAnTE.TAUAnTEm = 0).

Address <TAUAn_base_USER> + 248_H

Initial Value 0000_H. This register is initialized by any reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn TOM15	TAUAn TOM14	TAUAn TOM13	TAUAn TOM12	TAUAn TOM11	TAUAn TOM10	TAUAn TOM09	TAUAn TOM08	TAUAn TOM07	TAUAn TOM06	TAUAn TOM05	TAUAn TOM04	TAUAn TOM03	TAUAn TOM02	TAUAn TOM01	TAUAn TOM00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 12-237 TAUAnTOM register contents

Bit position	Bit name	Function
15 to 0	TAUAnTOMm	Specifies the channel output mode: 0: Independent channel output mode 1: Synchronous channel output mode The output mode depends on several channel output control bits, as can be seen in *Table 12-9 “Channel output modes” on page 578*.

(3) TAUAnTOC - TAUAn channel output configuration register

This register specifies the output mode of each channel in combination with TAUAnTOMm.

Access This register can be read/written in 16-bit units. Writing is only possible while the counter is stopped (TAUAnTE.TAUAnTEm = 0).

Address <TAUAn_base_OSbase_USER>> + $24C_H$

Initial Value 0000_H. This register is initialized by any reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn TOC15	TAUAn TOC14	TAUAn TOC13	TAUAn TOC12	TAUAn TOC11	TAUAn TOC10	TAUAn TOC09	TAUAn TOC08	TAUAn TOC07	TAUAn TOC06	TAUAn TOC05	TAUAn TOC04	TAUAn TOC03	TAUAn TOC02	TAUAn TOC01	TAUAn TOC00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 12-238 TAUAnTOC register contents

Bit position	Bit name	Function
15 to 0	TAUAn TOCm	Specifies the output mode: 0: Operation mode 1 1: Operation mode 2 The output mode also depends on TAUAnTOM.TAUAnTOMm, as can be seen in the following table.

TOMm	TOCm	Description
0	0	Toggle mode: TAUAnTTOUTm toggles when INTTAUAnIm occurs.
	1	Set/reset mode: TAUAnTTOUTm set when INTTAUAnIm occurs upon count start and reset when INTTAUAnIm occurs due to detection of a match between TAUAnCNTm and TAUAnCDRm.
1	0	Synchronous Channel Operation Mode 1: TAUAnTTOUTm set when INTTAUAnI occurs on the master channel and reset when INTTAUAnI occurs on the slave channel.
	1	Synchronous Channel Operation Mode 2: TAUAnTTOUTm set when INTTAUAnIm occurs while the slave channel is counting down and reset when INTTAUAnIm occurs while the slave channel is counting up.

(4) TAUAnTDE - TAUAn channel dead time output enable register

This register enables/disables dead time operation for each channel.

Access This register can be read/written in 16-bit units. Writing is only possible while the counter is stopped (TAUAnTE.TAUAnTEm = 0).

Address <TAUAn_base_OS> + 250_H

Initial Value 0000_H. This register is initialized by any reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn TDE15	TAUAn TDE14	TAUAn TDE13	TAUAn TDE12	TAUAn TDE11	TAUAn TDE10	TAUAn TDE09	TAUAn TDE08	TAUAn TDE07	TAUAn TDE06	TAUAn TDE05	TAUAn TDE04	TAUAn TDE03	TAUAn TDE02	TAUAn TDE01	TAUAn TDE00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 12-239 TAUAnTDE register contents

Bit position	Bit name	Function
15 to 0	TAUAnTDEm	Enables/disables dead time control operation of channel m: 0: Disables dead time operation 1: Enables dead time operation The same settings must be set for the even and the odd slave channel that comprise a set. These bits only apply when: • TAUAnTOE.TAUAnTOEm, TAUAnTOM.TAUAnTOMm, and TAUAnTOC.TAUAnTOCm = 1.

(5) TAUAnTDM - TAUAn channel dead time output mode register

This register specifies when dead time is added during dead time output.

Access This register can be read/written in 16-bit units. Writing is only possible while the counter is stopped (TAUAnTE.TAUAnTEm = 0).

Address <TAUAn_base_OS> + 254_H

Initial Value 0000_H. This register is initialized by any reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn TDM15	TAUAn TDM14	TAUAn TDM13	TAUAn TDM12	TAUAn TDM11	TAUAn TDM10	TAUAn TDM09	TAUAn TDM08	TAUAn TDM07	TAUAn TDM06	TAUAn TDM05	TAUAn TDM04	TAUAn TDM03	TAUAn TDM02	TAUAn TDM01	TAUAn TDM00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 12-240 TAUAnTDM register contents

Bit position	Bit name	Function
15 to 0	TAUAnTDMm	Specifies when dead time is added during dead time output: 0: Upon detection of a duty cycle at the upper even channel (duty dead time output) 1: Upon detection of a TAUAnTTIN input edge at the lower odd channel (one-phase dead time output) The same settings must be set for the even and the odd slave channel that comprise a set. These bits only apply when: • TAUAnTOE.TAUAnTOEm, TAUAnTOM.TAUAnTOMm, TAUAnTOC.TAUAnTOCm, and TAUAnTDE.TAUAnTDEm = 1.

(6) TAUAnTDL - TAUAn channel dead time output level register

This register selects the phase period to which dead time is added.

Access This register can be read/written in 16-bit units.

Address <TAUAn_base_USER> + 54_H

Initial Value 0000_H. This register is initialized by any reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn TDL15	TAUAn TDL14	TAUAn TDL13	TAUAn TDL12	TAUAn TDL11	TAUAn TDL10	TAUAn TDL09	TAUAn TDL08	TAUAn TDL07	TAUAn TDL06	TAUAn TDL05	TAUAn TDL04	TAUAn TDL03	TAUAn TDL02	TAUAn TDL01	TAUAn TDL00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 12-241 TAUAnTDL register contents

Bit position	Bit name	Function
15 to 0	TAUAnTDLm	Selects the phase period to which dead time is added: 0: Positive phase period 1: Negative phase period These bits only apply when: • TAUAnTOE.TAUAnTOEm, TAUAnTOM.TAUAnTOMm, TAUAnTOC.TAUAnTOCm, and TAUAnTDE.TAUAnTDEm = 1.

(7) TAUAnTRE - TAUAn channel real-time output enable register

This register enables or disables real-time output.

Access This register can be read/written in 16-bit units. Writing is only possible while TAUAnTE.TAUAnTEm is 0.

Address <TAUAn_base_OS> + 258_H

Initial Value 0000_H. This register is initialized by any reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn TRE15	TAUAn TRE14	TAUAn TRE13	TAUAn TRE12	TAUAn TRE11	TAUAn TRE10	TAUAn TRE09	TAUAn TRE08	TAUAn TRE07	TAUAn TRE06	TAUAn TRE05	TAUAn TRE04	TAUAn TRE03	TAUAn TRE02	TAUAn TRE01	TAUAn TRE00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 12-242 TAUAnTRE register contents

Bit position	Bit name	Function
15 to 0	TAUAnTREm	Enables or disables real-time output of channel m. 0: Disables real-time output 1: Enables real-time output These bits only apply when TAUTnTOE.TAUAnTOEm = 1. When TAUAnTRE.TAUAnTREm = 0, TAUAnTTOUTm is not affected by real-time output. When TAUAnTRE.TAUAnTREm = 1, TAUAnTTOUTm outputs the value of the real-time output bit TAUAnTRO.TAUAnTROm, dependent on the timer operation.

(8) TAUAnTRC - TAUAn channel real-time control register

This register controls the real-time output trigger for each channel.

Access This register can be read/written in 16-bit units. Writing is only possible while TAUAnTE.TAUAnTEm is 0.

Address <TAUAn_base_OS> + $25C_H$

Initial Value 0000_H. This register is initialized by any reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn TRC15	TAUAn TRC14	TAUAn TRC13	TAUAn TRC12	TAUAn TRC11	TAUAn TRC10	TAUAn TRC09	TAUAn TRC08	TAUAn TRC07	TAUAn TRC06	TAUAn TRC05	TAUAn TRC04	TAUAn TRC03	TAUAn TRC02	TAUAn TRC01	TAUAn TRC00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 12-243 TAUAnTRC register contents

Bit position	Bit name	Function
15 to 0	TAUAnTRCm	Specifies which channel generates the real-time output trigger for channel m: 0: The next upper channel where this bit is set to 1 1: Channel m These bits only apply when TAUAnTRE.TAUAnTREm = 1.

(9) TAUAnTRO - TAUAn channel real-time output register

For this register, specify the value output to TAUAnTTOUTm.

Access This register can be read/written in 16-bit units.

Address <TAUAn_base_USER> + $4C_H$

Initial Value 0000_H. This register is initialized by any reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn TRO15	TAUAn TRO14	TAUAn TRO13	TAUAn TRO12	TAUAn TRO11	TAUAn TRO10	TAUAn TRO09	TAUAn TRO08	TAUAn TRO07	TAUAn TRO06	TAUAn TRO05	TAUAn TRO04	TAUAn TRO03	TAUAn TRO02	TAUAn TRO01	TAUAn TRO00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 12-244 TAUAnTRO register contents

Bit position	Bit name	Function
15 to 0	TAUAnTROm	Specifies the value output to TAUAnTTOUTm. When TAUAnTRE.TAUAnTRE is 0, the TAUAnTROm value is not output to TAUAnTTOUTm even if a real-time trigger is generated.

(10) TAUAnTME - TAUAn channel modulation output enable register

This register enables and disables modulation output for the timer output and real-time output.

Access This register can be read/written in 16-bit units.

Address <TAUAn_base_USER> + 50_H

Initial Value 0000_H. This register is initialized by any reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn TME15	TAUAn TME14	TAUAn TME13	TAUAn TME12	TAUAn TME11	TAUAn TME10	TAUAn TME09	TAUAn TME08	TAUAn TME07	TAUAn TME06	TAUAn TME05	TAUAn TME04	TAUAn TME03	TAUAn TME02	TAUAn TME01	TAUAn TME00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 12-245 TAUAnTME register contents

Bit position	Bit name	Function
15 to 0	TAUAnTMEm	Enables/disables modulation output for timer output and real-time output of channel m: 0: Disables modulation 1: Enables modulation These bits only apply when TAUAnTOE.TAUAnTOEm and TAUAnTRE.TAUAnTREm = 1.

12.27.5 TAUAn channel output level registers details

(1) TAUAnTO - TAUAn channel output register

This register specifies and reads the level of TAUAnTTOUTm.

Access This register can be read/written in 16-bit units.

Address <TAUAn_base_USER> + 58_H

Initial Value 0000_H. This register is initialized by any reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn TO15	TAUAn TO14	TAUAn TO13	TAUAn TO12	TAUAn TO11	TAUAn TO10	TAUAn TO09	TAUAn TO08	TAUAn TO07	TAUAn TO06	TAUAn TO05	TAUAn TO04	TAUAn TO03	TAUAn TO02	TAUAn TO01	TAUAn TO00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 12-246 TAUAnTO register contents

Bit position	Bit name	Function
15 to 0	TAUAnTOm	Specifies/reads the level of TAUAnTTOUTm: 0: Low 1: High Only TAUAnTOm bits for which Independent Channel Output function is disabled (TAUAnTOEm = 0) can be written.

(2) TAUAnTOL - TAUAn channel output level register

This register specifies the output logic of the channel output bit (TAUAnTO.TAUAnTOm).

Access This register can be read/written in 16-bit units.

Address <TAUAn_base_USER> + 40_H

Initial Value 0000_H. This register is initialized by any reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn TOL15	TAUAn TOL14	TAUAn TOL13	TAUAn TOL12	TAUAn TOL11	TAUAn TOL10	TAUAn TOL09	TAUAn TOL08	TAUAn TOL07	TAUAn TOL06	TAUAn TOL05	TAUAn TOL04	TAUAn TOL03	TAUAn TOL02	TAUAn TOL01	TAUAn TOL00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 12-247 TAUAnTOL register contents

Bit position	Bit name	Function
15 to 0	TAUAnTOLm	Specifies the output logic of the channel m output bit (TAUAnTO.TAUAnTOm): 0: Positive logic (active high) 1: Inverted logic (active low)

12.27.6 TAUAn simultaneous rewrite register details

(1) TAUAnRDE - TAUAn channel reload data enable register

This register enables and disables simultaneous rewriting of the data register TAUAnCDRm or TAUAnTOLm.

Access This register can be read/written in 16-bit units. Writing is only possible while TAUAnTE.TAUAnTEm is 0.

Address <TAUAn_base_OS> + 260_H

Initial Value 0000_H. This register is initialized by any reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn RDE15	TAUAn RDE14	TAUAn RDE13	TAUAn RDE12	TAUAn RDE11	TAUAn RDE10	TAUAn RDE09	TAUAn RDE08	TAUAn RDE07	TAUAn RDE06	TAUAn RDE05	TAUAn RDE04	TAUAn RDE03	TAUAn RDE02	TAUAn RDE01	TAUAn RDE00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 12-248 TAUAnRDE register contents

Bit position	Bit name	Function
15 to 0	TAUAnRDEm	Enables/disables simultaneous rewrite of the data register of channel m: 0: Disables simultaneous rewrite 1: Enabled simultaneous rewrite

(2) TAUAnRDS - TAUAn channel reload data control channel select register

This register selects the control channel for simultaneous rewrite.

Access This register can be read/written in 16-bit or 1-bit units. Writing is only possible while TAUAnTE.TAUAnTEm is 0.

Address <TAUAn_base_OS> + 268_H

Initial Value 0000_H. This register is initialized by any reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn RDS15	TAUAn RDS14	TAUAn RDS13	TAUAn RDS12	TAUAn RDS11	TAUAn RDS10	TAUAn RDS09	TAUAn RDS08	TAUAn RDS07	TAUAn RDS06	TAUAn RDS05	TAUAn RDS04	TAUAn RDS03	TAUAn RDS02	TAUAn RDS01	TAUAn RDS00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 12-249 TAUAnRDS register contents

Bit position	Bit name	Function
15 to 0	TAUAnRDSm	Specifies which channel is monitored for the simultaneous rewrite trigger: 0: Master channel 1: Another upper channel

(3) TAUAnRDM - TAUAn channel reload data mode register

This register selects when the signal that controls simultaneous rewrite is loaded.

Access This register can be read/written in 16-bit units. Writing is only possible while TAUAnTE.TAUAnTEm is 0.

Address <TAUAn_base_OS> + 264_H

Initial Value 0000_H. This register is initialized by any reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn RDM15	TAUAn RDM14	TAUAn RDM13	TAUAn RDM12	TAUAn RDM11	TAUAn RDM10	TAUAn RDM09	TAUAn RDM08	TAUAn RDM07	TAUAn RDM06	TAUAn RDM05	TAUAn RDM04	TAUAn RDM03	TAUAn RDM02	TAUAn RDM01	TAUAn RDM00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 12-250 TAUAnRDM register contents

Bit position	Bit name	Function
15 to 0	TAUAnRDMm	Selects when the signal that triggers simultaneous is generated: 0: When the master channel counter starts counting 1: At the top of a triangle wave cycle These bits only apply when TAUAnRDE.TAUAnRDEm = 1 and TAUAnRDS.RDSm = 0.

(4) TAUAnRDC - TAUAn channel reload data control register

This register specifies the channel that generates the INTTAUAnIm signal that triggers simultaneous rewrite.

Access This register can be read/written in 16-bit units. Writing is only possible while TAUAnTE.TAUAnTEm is 0.

Address <TAUAn_base_OS> + $26C_H$

Initial Value 0000_H. This register is initialized by any reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn RDC15	TAUAn RDC14	TAUAn RDC13	TAUAn RDC12	TAUAn RDC11	TAUAn RDC10	TAUAn RDC09	TAUAn RDC08	TAUAn RDC07	TAUAn RDC06	TAUAn RDC05	TAUAn RDC04	TAUAn RDC03	TAUAn RDC02	TAUAn RDC01	TAUAn RDC00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 12-251 TAUAnRDC register contents

Bit position	Bit name	Function
15 to 0	TAUAnRDCm	Specifies whether the channel generates the simultaneous rewrite trigger signal. 0: Have the channel generate the simultaneous rewrite trigger signal. 1: Do not have the channel generate the simultaneous rewrite trigger signal. These bits only apply when TAUAnRDS.TAUAnRDSm is 1.

(5) TAUAnRDT - TAUAn channel reload data trigger register

This register triggers the simultaneous rewrite pending state.

Access This register can be written in 16-bit units. It is always read as 0000_H.

Address <TAUAn_base_USER> + 44_H

Initial Value 0000_H. This register is initialized by any reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn RDT15	TAUAn RDT14	TAUAn RDT13	TAUAn RDT12	TAUAn RDT11	TAUAn RDT10	TAUAn RDT09	TAUAn RDT08	TAUAn RDT07	TAUAn RDT06	TAUAn RDT05	TAUAn RDT04	TAUAn RDT03	TAUAn RDT02	TAUAn RDT01	TAUAn RDT00
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 12-252 TAUAnRDT register contents

Bit position	Bit name	Function
15 to 0	TAUAnRDTm	Triggers the simultaneous rewrite pending state: 0: No function 1: Simultaneous rewrite pending state is triggered. The simultaneous rewrite pending flag (TAUAnRSFm) is set to 1. The system waits for the simultaneous rewrite trigger.

(6) TAUAnRSF - TAUAn channel reload status register

This flag register indicates the simultaneous rewriting status.

Access This register can be read in 16-bit units.

Address <TAUAn_base_USER> + 48_H

Initial Value 0000_H. This register is initialized by any reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn RSF15	TAUAn RSF14	TAUAn RSF13	TAUAn RSF12	TAUAn RSF11	TAUAn RSF10	TAUAn RSF09	TAUAn RSF08	TAUAn RSF07	TAUAn RSF06	TAUAn RSF05	TAUAn RSF04	TAUAn RSF03	TAUAn RSF02	TAUAn RSF01	TAUAn RSF00
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 12-253 TAUAnRSF register contents

Bit position	Bit name	Function
15 to 0	TAUAnRSFm	Indicates the simultaneous rewrite status: 0: Simultaneous rewriting was performed due to the generation of a simultaneous rewrite trigger. 1: Simultaneous rewriting is pending (TAUAnRDTm = 1).

12.27.7 TAUAn DMA window registers

(1) TAUAnDASi - TAUAn DMA window address setting register i (i = 0 to 7)

This register specifies addresses of the window registers used for DMA. Eight TAUAnDASi registers control sixteen TAUAnDWRj registers, i.e. each TAUAnDASi register controls two TAUAnDWRj registers independently.

Access This register can be read/written in 16-bit units.

Address <TAUAn_base_OSbase_USER>> + 270_H + i x 4_H

Initial Value 0000_H. This register is initialized by any reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUAn DAS *odd7*	TAUAn DAS *odd6*	TAUAn DAS *odd5*	TAUAn DAS *odd4*	TAUAn DAS *odd3*	TAUAn DAS *odd2*	TAUAn DAS *odd1*	TAUAn DAS *odd0*	TAUAn DAS *even7*	TAUAn DAS *even6*	TAUAn DAS *even5*	TAUAn DAS *even4*	TAUAn DAS *even3*	TAUAn DAS *even2*	TAUAn DAS *even1*	TAUAn DAS *even0*
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

odd = 01, 03, 05, 07, 09, 11, 13, 15

even = 00, 02, 04, 06, 08, 10, 12, 14

The following table shows the relationship between the TAUAnDASi control registers and the TAUAnDWRj registers:

Table 12-254 Relationship between TAUAnDASi registers and TAUAnDWRj registers

Control register	Control bits	Control target register TAUAnDWRj
TAUAnDAS0	Bits 7 to 0	TAUAnDWR0
TAUAnDAS0	Bits 15 to 8	TAUAnDWR1
TAUAnDAS1	Bits 7 to 0	TAUAnDWR2
TAUAnDAS1	Bits 15 to 8	TAUAnDWR3
TAUAnDAS2	Bits 7 to 0	TAUAnDWR4
TAUAnDAS2	Bits 15 to 8	TAUAnDWR5
TAUAnDAS3	Bits 7 to 0	TAUAnDWR6
TAUAnDAS3	Bits 15 to 8	TAUAnDWR7
TAUAnDAS4	Bits 7 to 0	TAUAnDWR8
TAUAnDAS4	Bits 15 to 8	TAUAnDWR9
TAUAnDAS5	Bits 7 to 0	TAUAnDWR10
TAUAnDAS5	Bits 15 to 8	TAUAnDWR11
TAUAnDAS6	Bits 7 to 0	TAUAnDWR12
TAUAnDAS6	Bits 15 to 8	TAUAnDWR13
TAUAnDAS7	Bits 7 to 0	TAUAnDWR14
TAUAnDAS7	Bits 15 to 8	TAUAnDWR15

Table 12-255 TAUAnDASi register contents

Bit position	Bit name	Function
15 to 8	TAUAnDAS *odd*[15:8]	00_H to $3C_H$: Specifies the CDR0 to CDR15 registers. 40_H: Specifies the TAUAnTOL register. 44_H: Specifies the TAUAnRDT register. 48_H: Specifies the TAUAnRSF register. $4C_H$: Specifies the TAUAnTRO register. 50_H: Specifies the TAUAnTME register. 54_H: Specifies the TAUAnTDL register. 58_H: Specifies the TAUAnTO register. $5C_H$: Specifies the TAUAnTOE register. 60_H to $7C_H$: Setting prohibited 80_H to BC_H: Specifies the CNT0 to CNT15 registers. $C0_H$ to FC_H: Setting prohibited
7 to 0	TAUAnDAS *even*[7:0]	00_H to $3C_H$: Specifies the CDR0 to CDR15 registers. 40_H: Specifies the TAUAnTOL register. 44_H: Specifies the TAUAnRDT register. 48_H: Specifies the TAUAnRSF register. $4C_H$: Specifies the TAUAnTRO register. 50_H: Specifies the TAUAnTME register. 54_H: Specifies the TAUAnTDL register. 58_H: Specifies the TAUAnTO register. $5C_H$: Specifies the TAUAnTOE register. 60_H to $7C_H$: Setting prohibited 80_H to BC_H: Specifies the CNT0 to CNT15 registers. $C0_H$ to FC_H: Setting prohibited

Be sure to fix bits [9:8] and [1:0] to “0”.

(2) TAUAnDWRj - TAUAn DMA window register j

This register is used for DMA (j = 0 to 15). TAUAnDWRj mirrors the addresses specified for the corresponding TAUAnDASi registers (where i is a value from 0 to 7). (See *(1) “TAUAnDASi - TAUAn DMA window address setting register i (i = 0 to 7)” on page 887*.)

Access This register can be read/written in 16-bit units.

Address <TAUAn_base_USER> + 0100_H + j x 4_H

Initial Value 0000_H. This register is initialized by any reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMA window address specified for TAUAnDASi (where i is a value from 0 to 7)															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Chapter 13 Timer Array Unit J (TAUJ)

This chapter contains a generic description of the Timer Array Unit J (TAUJ).

The first section describes all V850E2/MN4 specific properties, such as instances, register base addresses, input/output signal names, etc.
The subsequent sections describe the features that apply to all implementations.

13.1 V850E2/MN4 TAUJ Features

Instances

This microcontroller has following number of instances of the Timer Array Unit J.

Table 13-1 Instances of TAUJ

TAUJ	
Instance	1
Name	TAUJ0

Instances index n

Throughout this chapter, the individual instances of a Timer Array Unit J is identified by the index "n" (n = 0), for example, TAUJnTOM for the TAUJn channel output mode register.

Channel index m

The Timer Array Unit J has 4 channels. Throughout this chapter, the individual channels are identified by the index “m” (m = 0 to 3), thus a certain channel is denoted as CHm.
The even numbered channels (m = 0, 2) are denoted as CHm_even.
The odd numbered channels (m = 1, 3) are denoted as CHm_odd.

Register addresses

All TAUJ n register addresses are given as address offsets to the individual base address <TAUJ n_base>.
The <TAUJ n_base> address of each TAUJn are listed in the following table:

Table 13-2 Register base addresses <TAUJn_base>

TAUJn	Base address	Address
TAUJ0	<TAUJn_base_USER>	FFFF C200$_H$
	<TAUJn_base_OS>	FF81 1000$_H$

Clock supply

All Timer Array Units J provide one clock input.

Table 13-3 TAUJn clock supply

TAUJn instance	TAUJn clock	Connected to
TAUJ0	PCLK	f_{PCLK}

Interrupts and DMA/DTS

Timer Array Unit J can generate the following interrupt and DMA/DTS requests:

Table 13-4 TAUJn interrupt and DMA/DTS requests

TAUJn signals	Function	Connected to
TAUJ0:		
INTTAUJ0I0	Channel 0 interrupt	• Interrupt Controller INTTAUJ0I0 • DMA Controller trigger 84 • DTS Controller trigger 84
INTTAUJ0I1	Channel 1 interrupt	• Interrupt Controller INTTAUJ0I1 • DMA Controller trigger 85 • DTS Controller trigger 85
INTTAUJ0I2	Channel 2 interrupt	• Interrupt Controller INTTAUJ0I2 • DMA Controller trigger 86 • DTS Controller trigger 86
INTTAUJ0I3	Channel 3 interrupt	• Interrupt Controller INTTAUJ0I3 • DMA Controller trigger 87 • DTS Controller trigger 87

TAUJ H/W reset

The Timer Array Units J and their registers are initialized by the following reset signal:

Table 13-5 TAUJn reset signal

TAUJn	Function	Reset signal
TAUJn		
TAUJ0TTIN0 to TAUJ0TTIN3	Channel 0 to 3 input	Port TJ_I0-TJ_I3
TAUJ0TTOUT0 to TAUJ0TTOUT3	Channel 0 to 3 input	Port TJ_O0-TJ_O3

I/O signals

The I/O signals of the Timer Array Unit J are listed in the following table.

Table 13-6 TAUJn I/O signals

TAUJ signal	Function	Connected to
TAUJ0:		
TAUJnTSSTm	Synchronous channel start trigger input	PIC

13.2 Functional Overview

Features summary

The TAUJ has the following functions:

- 4 channels
- 32-bit counter and 32-bit data register per channel
- Independent channel operation
- Synchronous channel operation (master and slave operation)
- Generation of different types of output signal
- Counter can be triggered by external signal
- Interrupt generation

The following figure shows the main components of the TAUJ:

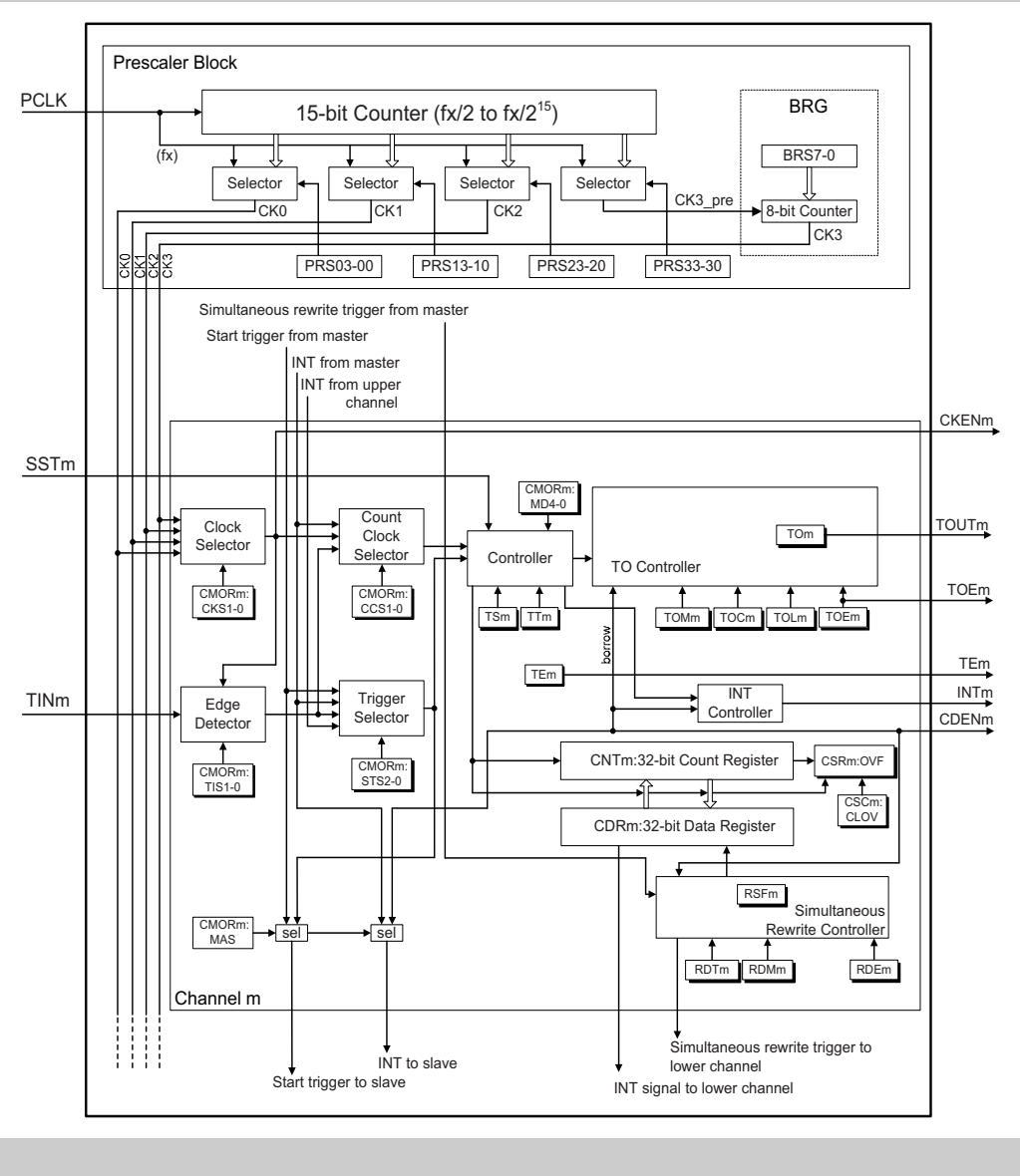


Figure 13-1 Block diagram of the TAUJ

The prefix *TAUJn* has been omitted from the register and block names in the above figure for the sake of clarity.

13.2.1 Terms

In this section, the following terms are used:

- **Independent / synchronous channel operation**

  Independent or synchronous channel operation describes the dependency of channels on each other:

  – If a channel operates independent of all other channels, this is called independent channel operation.

  – If a channel operates depending on other channels, this is called synchronous channel operation.

- **Channel group**

  In synchronous channel operation, all channels that depend on each other are referred to as a “channel group”.

  A channel group has one master channel and one or more slave channels.

- **Operation mode**

  An operation mode can be selected for every channel m. The operation mode defines the *basic* operation and features of a channel.

  In synchronous channel operation, every channel in the channel group can operate in a different operation mode.

  Examples are “Capture Mode”, and “Interval Timer Mode”.

- **Channel output mode**

  The channel output mode defines the operation of TAUJnTTOUTm

  – of a single channel (independent output operation) or

  – of all channels in a channel group (synchronous output operation).

  An example is “Independent Channel Output Mode 1”.

- **Channel operation function**

  The channel operation function defines the *complete* function and all features

  – of a single channel (independent channel operation) or

  – of all channels in a channel group (synchronous channel operation).

- **Upper / lower channel**

  Depending on the channel number m, a neighboring channel can be referred to as “upper” or “lower” channel:

  – Upper channel: Channel that has a smaller number

  – Lower channel: Channel that has a greater number

  Example:
  For channel 2, channel 1 is an upper channel and channel 3 is a lower channel.

13.3 Functional Description

TAUJ is used to perform various count or timer operations and to output a signal which depends on the result of the operation. It contains one prescaler for count clock generation and 4 channels, each equipped with a 32-bit counter TAUJnCNTm and a 32-bit data register TAUJnCDRm to hold the start or compare value of the counter.

It also contains several control and status registers.

Independent and synchronous operation

Every channel can operate in two operation modes, either independently or in combination with other channels (synchronously). If a channel group has one master channel and one or more slave channels, the slave channels depend on the master channel.

When a channel is operated independently, its operation mode and functions are not affected by those of other channels. When a channel is operated synchronously it is either a master or a slave. A master channel can have multiple slaves, and the state of one channel affects that of the other channels. For example, a channel can be used to control the timing for starting counting and resetting for another channel.

The following describes the functional blocks:

Prescaler

The prescaler provides up to 4 clock signals (CK0 to CK3) that can be used as count clocks for all channels.

For count clocks CK0 to CK2, a clock obtained by dividing PCLK by 2^0 to 2^{15} using the prescaler can be selected. For the fourth count clock CK3, a division factor that is not a power of 2 can be specified by using BRG.

Clock and count clock selection

For every channel, the count clock selector selects which of the following is used as the clock source:

- One of the clocks CK0 to CK3 (selected by the clock selector)
- INTTAUJnIm from master channel
- TAUJnTTINm input signal valid edge

Controller

The controller controls the main operations of the counter:

- Operation mode (selected by bits TAUJnCMORm.TAUJnMD[4:0])
- Counter start enable (TAUJnTS.TAUJnTSm) and counter stop (TAUJnTT.TAUJnTTm)

  When counter start is enabled, status flag TAUJnTE.TAUJnTEm is set.

Trigger selector

If the counter is enabled (TAUJnTE.TAUJnTEm = 1), it starts automatically or waits for an external start trigger signal, depending on the selected operation mode. Any of the following signals can be used as the start trigger:

- Synchronous channel start trigger input TAUJnTSSTm
  For details about the inter-unit simultaneous start method, refer to the simultaneous start control register in Chapter X "PIC Function". (Only include this information for a product that includes the PIC function.)
- TAUJnTTINm input signal valid edge
- INTTAUJnIm from the master channel

Simultaneous rewrite controller Simultaneous rewrite control is a function that can be used in synchronous operation modes. The data registers of all channels in a channel group (TAUJnCDRm) can be rewritten at any time. The simultaneous rewrite controller ensures that new data register values of all channels become effective at the same time.

TAUJnTO Controller The output control of every channel enables the generation of various output signal forms such as PWM signals.

13.3.1 Timer operation functions

The functions below can be achieved by operating TAUJ independently on each channel or by operating it on a combination of multiple channels.

Table 13-7 TAUJ operation functions

Independent channel operation function	Synchronous channel operation function
Independent channel operation functions	Synchronous channel operation function
Interval timer function	PWM output function
TAUJnTTINm input interval timer function	
Independent channel signal measurement functions	
Overflow interrupt output function (during TAUJnTTINm width measurement)	
TAUJnTTINm input period count detection function	
Overflow interrupt output function (during TAUJnTTINm input period count detection)	
TAUJnTTINm input pulse interval judgment function	
TAUJnTTINm input signal width judgment function	
Other independent channel function	
TAUJnTTINm input position detection function	

13.4 General Operating Procedure

The following lists the general operation procedure for the TAUJn:

After reset release, the operation of each channel is stopped. Writing to each register is enabled when clock supply is started. The control register of TAUJnTTOUTm is initialized and outputs a low level.

1. Set the TAUJnTPS and TAUJnBRS registers to specify the clock frequency of CK0 to CK3.
2. Configure the desired TAUJn function:
   – Set the operation mode
   – Set the channel output mode
   – Set any other control bits
3. Enable the counter by setting the TAUJnTS.TAUJnTSm bit to 1.

   The counter starts counting immediately, or when an appropriate trigger is detected, depending on the bit settings.
4. During counting, if desired, and if possible for the configured function, stop the counter or perform a forced restart operation.
5. Stop the function by setting the TAUJnTT.TAUJnTTm bit to 1.

Note A detailed description of the required control bits and the operation of the individual functions is given in the following sections:

13.14 “Independent Channel Interrupt Functions” on page 920

13.15 “Independent Channel Signal Measurement Functions” on page 934

13.16 “Other Independent Channel Function” on page 965

13.5 Operation Modes

The TAUJ contains 7 operation modes.

One operation mode can be set for each channel. It is specified using the TAUJnCMORm.TAUJnMD[4:0] bits.

Note For registers and bits, some values are fixed according to the operation function, and others are selected by the user.

For details about the settings for registers and bits, refer to the sections describing each operation function.

13.6 Concepts of Synchronous Channel Operation

In synchronous channel operation, multiple channels depend on each other, or are affected by changes in another channel. Therefore, several rules apply for the use of synchronous channel functions. These rules are detailed in *13.6.1 "Rules"*.

Two special features for synchronous channel operation are detailed in the following subsections:

- *13.6.2 "Simultaneous start and stop of synchronous channel counters" on page 900*
- *13.7 "Simultaneous Rewrite" on page 901*

13.6.1 Rules

Number of masters and slaves

- Only even channels (CH0, CH2) can be set as master channels. Any channel except CH0 can be set as a slave channel.
- Only channels lower than the master channel can be set as slave channels, and several slave channels can be set for one master channel.
  Example: If CH2 is a master channel, CH3 can be set as slave channel.
- If two master channels are used, slave channels cannot cross the master channels.
  Example: If CH0 and CH2 are master channels, CH1 can be set as a slave channel for CH0, but CH3 cannot.

Operation clock

- The same operation clock must be set for the corresponding slave channel and the master channel. Specify the same value for the TAUJnCMORm.TAUJnCKS[1:0] bits of the synchronized master and slave channels.

The basic concepts of master/slave usage and operation clocks are illustrated in the following figure.

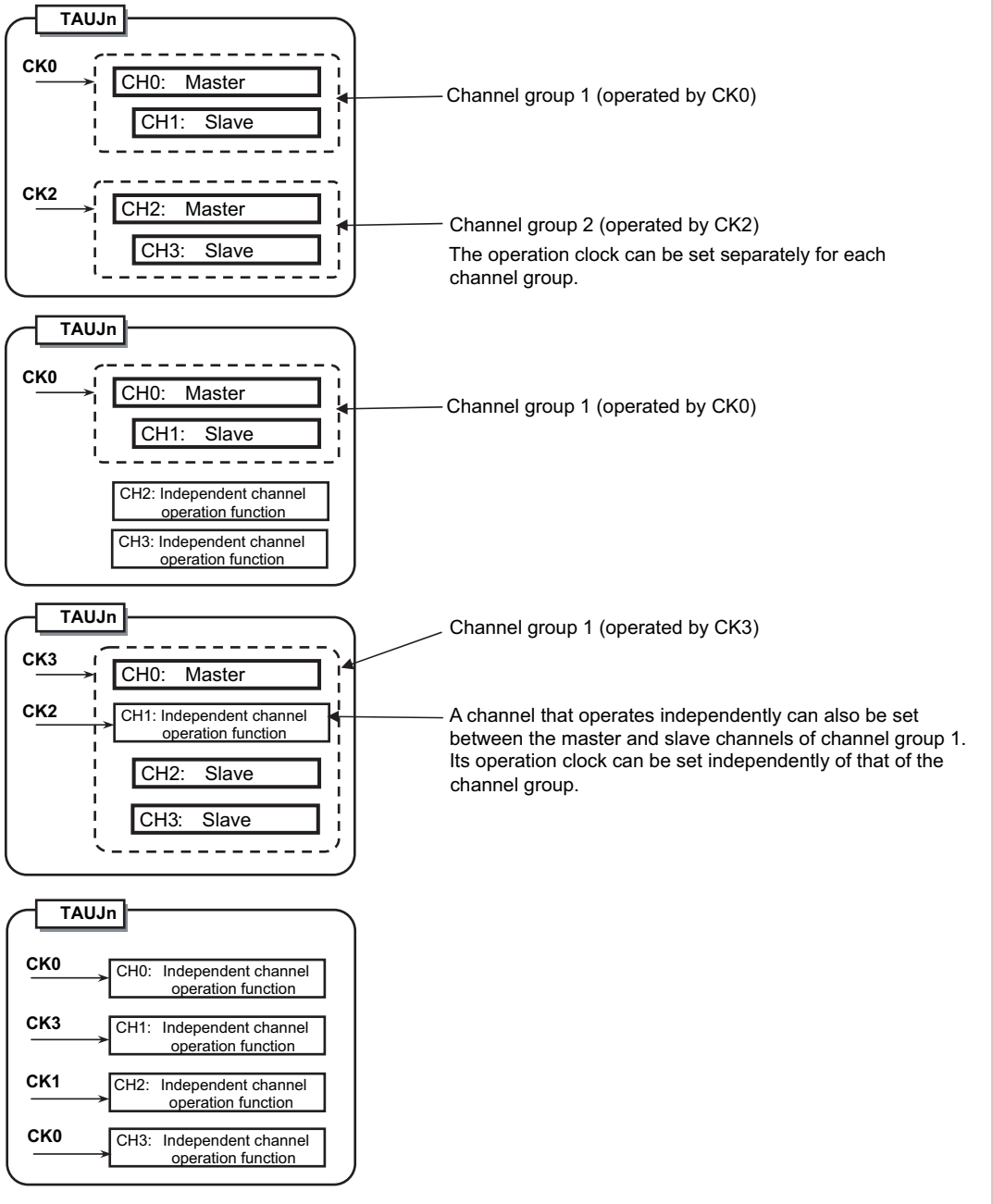


Figure 13-2 Grouping of the channels and assignment of operation clocks

Control trigger signals of the master and slave channels

- A master channel can output a control trigger signal to its slave channels.
- Slave channels can use the control trigger signal of the master channel but cannot transfer their control trigger signals to the lower channels.
- A master channel cannot use the control trigger signal of an upper master channel.

13.6.2 Simultaneous start and stop of synchronous channel counters

Channels that are operated synchronously can be started and stopped simultaneously, both within a unit, and between units.

(1) Simultaneous start and stop within a unit

- To simultaneously start synchronized channels, the TAUJnTS.TAUJnTSm bits of the channels must be set at the same time.
- To simultaneously stop synchronized channels, the TAUJnTT.TAUJnTTm bits of the channels must be set at the same time.

Setting a TAUJnTS.TAUJnTSm bit to 1 sets the corresponding TAUJnTE.TAUJnTEm bit to 1, enabling counting. When the counter starts counting depends on the operation mode.

(2) Simultaneous start between units

Counters in different units can also be started simultaneously if the corresponding counters are enabled before receiving the simultaneous trigger signal.

For details about the inter-unit simultaneous start method, see Chapter *16 “Peripheral Interconnection (PIC)”*.

13.7 Simultaneous Rewrite

13.7.1 Introduction

Simultaneous rewrite describes the ability to change the compare/start value and the output logic of multiple channels at the same time.

The corresponding data and control registers (TAUJnCDRm and TAUJnTOLm) can be written at any time. The new value does not affect the counter operation or the output signal until simultaneous rewrite is triggered.

Simultaneous rewrite is triggered when the counter on the master channel reaches a certain value.

The following table shows the settings for simultaneous rewrite (TAUJnRDM.TAUJnRDMm = 0).

Table 13-8 Simultaneous rewrite settings

Method	Simultaneous rewrite triggered when	TAUJnRDE.TAUJnRDEm
-	No simultaneous rewrite	0
A	The master channel (re)starts counting	1

13.7.2 How to control simultaneous rewrite

The following figure shows the general procedure for simultaneous rewrite. The three main blocks (Initial settings, Start counter & count operation, and Simultaneous rewrite) are explained afterwards.

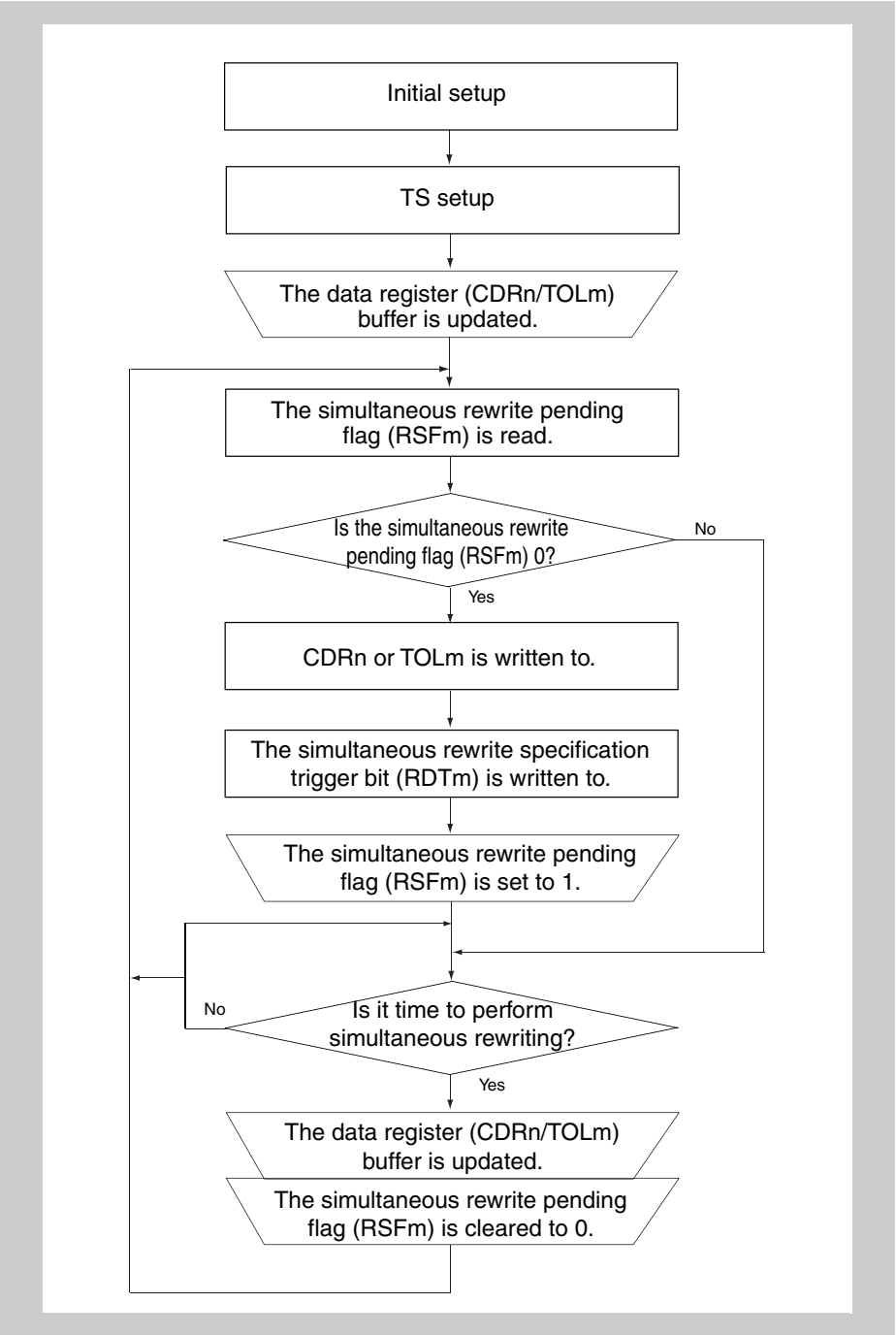


Figure 13-3 General procedure for simultaneous rewrite

(1) Initial settings

- To enable simultaneous rewrite in channel m, set TAUJnRDE.TAUJnRDEm = 1
- To select the type of simultaneous rewrite upon the start of counting by the master channel counter, specify the TAUJnRDM.TAUJnRDMm bits.

(2) Start counter and count operation

- To start all the TAUJnCNTm counters in the channel group, set the corresponding TAUJnTS.TAUJnTSm bits to 1. TAUJnTOL.TAUJnTOLm and the values in the data registers (TAUJnCDRm) are loaded to the corresponding TAUJnTOL.TAUJnTOLm buffer (TAUJnTOL.TAUJnTOLm buf) and data buffer registers (TAUJnCDRm buf) and the counters start.
- Setting the reload data trigger bit (TAUJnRDT.TAUJnRDTm) to 1 sets the reload flag (TAUJnRSF.TAUJnRSFm) to 1, enabling simultaneous rewrite. TAUJnRSF.TAUJnRSFm remains at 1 until simultaneous rewrite has taken place.
- When the specified trigger for simultaneous rewrite is detected, the TAUJnRSF.TAUJnRSFm bit is checked to see if simultaneous rewrite is enabled (TAUJnRSF.TAUJnRSFm = 1). If it is, simultaneous rewrite is carried out. If such writing is disabled, simultaneous rewriting is not performed, and the system awaits the detection of the next simultaneous rewrite trigger.

(3) Simultaneous rewrite

- When the simultaneous rewrite trigger is detected and simultaneous rewrite is enabled (TAUJnRSF.TAUJnRSFm = 1), the current values of the data registers are copied to their buffers. These values are then loaded to the corresponding counters and the values are applied the next time the counter starts or restarts.
- When simultaneous rewriting finishes, the TAUJnRSF.TAUJnRSFm bit is cleared to 0, and the system awaits the next simultaneous rewrite trigger.

13.7.3 Other general rules of simultaneous rewrite

The following rules also apply:

- TAUJnRDE.TAUJnRDEm and TAUJnRDM.TAUJnRDMm cannot be changed while the counter is in operation (TAUJnTE.TAUJnTEm = 1).
- TAUJnTOL.TAUJnTOLm can only be rewritten during operation when in PWM output function. For all other output functions, TAUJnTOL.TAUJnTOLm must be written before the counter starts. If it is rewritten in another function, TAUJnTTOUTm outputs an invalid wave.

13.7.4 Simultaneous rewrite procedure

Simultaneous rewrite is executed when the master channel starts or restarts counting.

The simultaneous rewrite procedure is described in the following figure.

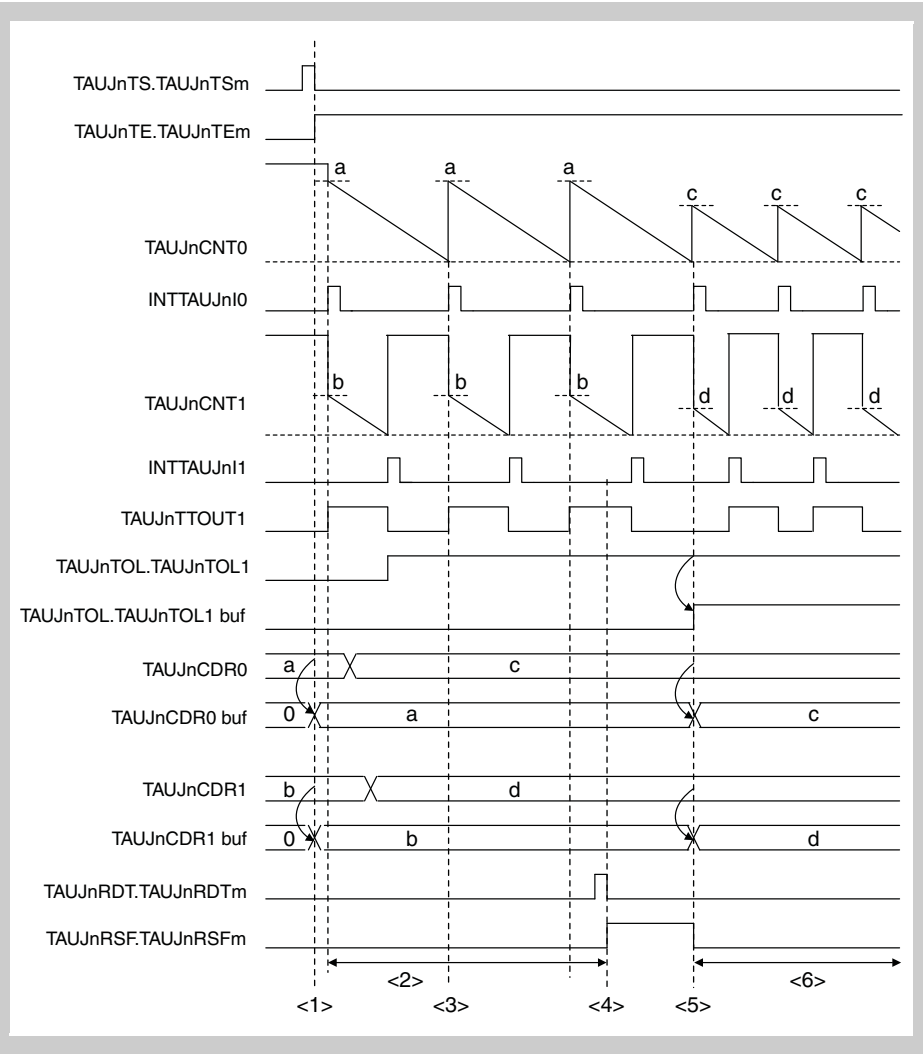


Figure 13-4 Simultaneous rewrite when the master channel (re)starts counting

Setup:

- CH0 is the master channel that counts down. CH1 represents an arbitrary slave channel. Simultaneous rewrite is executed when the master channel starts counting.

Description:

1. When TAUJnTS.TAUJnTSm is set to 1, the value of TAUJnCDRm is copied to the TAUJnCDRm buffer, and the value of TAUJnTOL.TAUJnTOLm is copied to the TAUJnTOL.TAUJnTOLm buffer.
2. The TAUJnCDRm and TAUJnTOL.TAUJnTOLm registers can be written at any time.
3. CH0 restarts counting, but simultaneous rewrite does not occur because it is disabled (TAUJnRSF.TAUJnRSFm = 0).
4. The reload data trigger bit (TAUJnRDT.TAUJnRDTm) is set to 1 which sets the status flag (TAUJnRSF.TAUJnRSFm = 1), enabling simultaneous rewrite.
5. Because simultaneous rewriting is enabled, it is performed before counting on channel 0 resumes. The TAUJnCDRm value is loaded to the TAUJnCDRm buffer, and the TAUJnTOL.TAUJnTOLm value is loaded to the TAUJnTOL.TAUJnTOLm buffer.
6. The counters count down and await the next simultaneous rewrite trigger. The values of TAUJnCDRm and TAUJnTOL.TAUJnTOLm can be changed again.

13.8 Channel Output Modes

The output of the TAUJnTTOUTm pin can be controlled in two ways, the latter of which can be further split into individual modes:

- By software (TAUJnTOE.TAUJnTOEm = 0)

  When controlled by software, the value written in the output register bit (TAUJnTO.TAUJnTOm) is output from the output pin (TAUJnTTOUTm).

- By TAUJ signals (TAUJnTOE.TAUJnTOEm = 1)

  When operated by TAUJ signals, the output level of TAUJnTTOUTm is set or reset or toggled by internal signals. The value of TAUJnTO.TAUJnTOm is updated accordingly to reflect the value of TAUJnTTOUTm.

  – Independently (TAUJnTOM.TAUJnTOMm = 0)

    When operated independently, the output of the TAUJnTTOUTm pin is only affected by settings of channel m. Therefore, independent channel operation must be specified (TAUJnTOM.TAUJnTOM = 0).

  – Synchronously (TAUJnTOM.TAUJnTOMm = 1)

    When operated synchronously, the output of the TAUJnTTOUTm pin is affected by settings of channel m and those of other channels. Therefore, synchronous channel operation must be selected for all participating channels (TAUJnTOM.TAUJnTOMm = 1).

The TAUJnTO.TAUJnTOm bit can always be read to determine the current value of TAUJnTTOUTm, regardless of whether the pin is controlled by software, operated independently, or operated synchronously.

Control bits

The settings of the control bits required to select a specific channel output mode are listed in *Table 13-9 “Channel output modes” on page 907*.

The channel output modes are described in detail in

- *13.8.2 “Channel output modes controlled independently by TAUJn signals” on page 909* to
- *13.8.3 “Channel output modes controlled synchronously by TAUJn signals” on page 910*.

Collective TAUJnTOm bit manipulation

Whether to apply settings to the TAUJnTOm bits is controlled using the TAUJnTOE.TAUJnTOEm bits.

When writing to the TAUJnTO register, TAUJnTOm settings are only written to bits (channels) for which the corresponding TAUJnTOE.TAUJnTOEm bit is cleared to 0. The TAUJnTOm settings are not applied for bits (channels) for which the corresponding TAUJnTOE.TAUJnTOEm bit is set to 1.

Note

The TAUJnTO.TAUJnTOm bits are allocated so that the bit numbers correspond to the channel numbers.

Output logic

Positive logic or inverted logic of the output is specified by control bit TAUJnTOL.TAUJnTOLm.

The value of the TAUJnTOL.TAUJnTOLm bit must be set before the counter is started. It can only be changed during operation in PWM output function. If the TAUJnTOL.TAUJnTOLm bit is changed after counter operation starts, the TAUJnTTOUTm signal output becomes undefined.

Refer to *13.7 “Simultaneous Rewrite” on page 901*.

The various channel output modes and the channel output control bits when TAUJnTOC.TAUJnTOCm = 0 are listed in the following table.

Table 13-9 Channel output modes

Channel output mode	TAUJnTOE. TAUJnTOEm	TAUJnTOM. TAUJnTOMm
By software		
Independent Channel Output Mode Controlled by Software	0	x
By TAUJ signals, independently		
Independent Channel Output Mode 1	1	0
By TAUJ signals, synchronously		
Synchronous Channel Output Mode 1	1	1

- The combinations not listed in this table are forbidden.
- The bit marked with an x can be set to any value.

Note The following bits cannot be changed during count operation (TAUJnTE.TAUJnTE = 1):

- TAUJnTOE.TAUJnTOEm
- TAUJnTOM.TAUJnTOMm
- TAUJnTOC.TAUJnTOCm

13.8.1 General procedure for specifying a channel output mode

The following steps describe the general procedure for specifying a TAUJnTTOUTm channel output mode. The prerequisite is that timer output operation is disabled (TAUJnTOE.TAUJnTOEm = 0).

1. Set TAUJnTO.TAUJnTOm to specify the initial level of the TAUJnTTOUTm output.
2. Set the channel output mode using *Table 13-9 “Channel output modes” on page 907* and the output logic using the TAUJnTOL.TAUJnTOLm bit.
3. Start the counter (TAUJnTS.TAUJnTSm = 1).

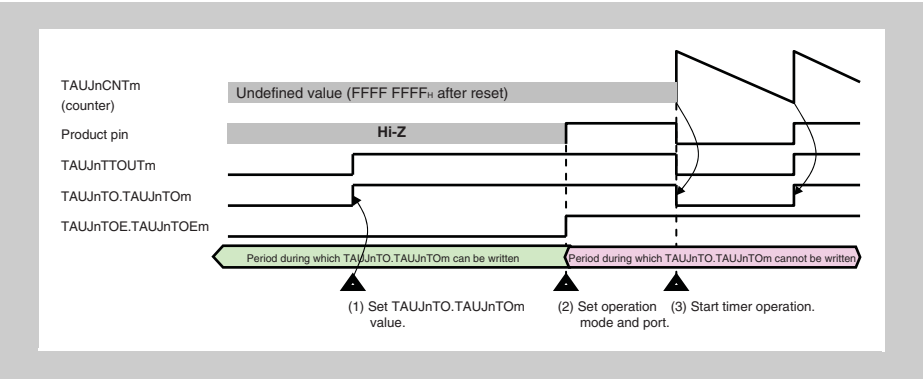


Figure 13-5 General procedure for specifying a TAUJnTTOUTm channel output mode

13.8.2 Channel output modes controlled independently by TAUJn signals

This section lists the channel output modes that are controlled independently by TAUJn signals. The control bits used to specify a mode are listed in *Table 13-9 “Channel output modes” on page 907*.

(1) Independent Channel Output Mode 1

Set/reset conditions In this output mode, TAUJnTTOUTm toggles when INTTAUJnIm is detected. The value of TAUJnTOL.TAUJnTOLm is ignored.

Prerequisites None, other than those in *Table 13-9 “Channel output modes” on page 907*.

13.8.3 Channel output modes controlled synchronously by TAUJn signals

This section lists the channel output modes that are controlled synchronously by TAUJn signals. The control bits used to specify a mode are listed in *Table 13-9 “Channel output modes” on page 907*.

(1) Synchronous Channel Output Mode 1

Set/reset conditions

In this output mode, INTTAUJnIm of the master channel serves as the set signal and INTTAUJnIm of the slave channel as the reset signal. If INTTAUJnIm of the master channel and INTTAUJnIm of the slave channel are generated at the same time, INTTAUJnIm of the slave channel (reset signal) has priority over INTTAUJnIm (set signal) of the master channel, i.e. the master channel is ignored.

Prerequisites

None, other than those in *Table 13-9 “Channel output modes” on page 907*.

13.9 Count Start Timing in Each Operating Mode

This section describes the timing for starting counting after the TAUJnTS.TAUJnTSm bit is set to 1, for each operation mode.

The value of the data register and whether an interrupt is generated depend on the individual mode and corresponding register settings.

Caution The timing for starting counting in this section is only for reference. The actual timing varies according to the count clock timing.

13.9.1 Interval Timer Mode, Capture Mode

The counter starts counting at the start of the next count clock cycle after TAUJnTS.TAUJnTSm is set to 1. The value of data register is also loaded at the point the counter starts.

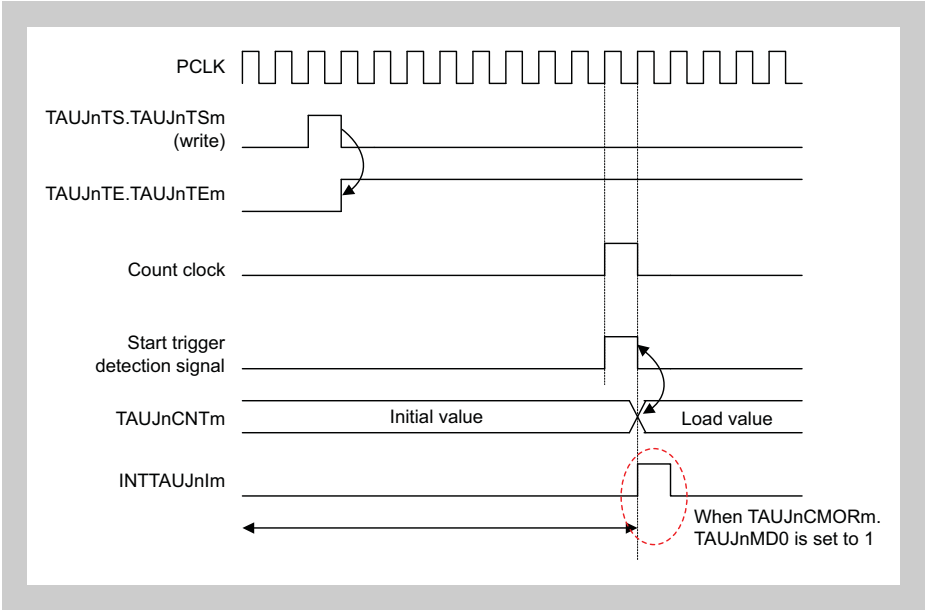


Figure 13-6 Start timing of Interval Timer Mode, Capture Mode

13.9.2 Other operating modes

In all other operating modes, the count clock cycles are ignored with regard to starting the counter. The counter operation is only triggered by detection of a valid TAUJnTTINm edge. The value of data register is also loaded at the point the counter starts. Although the count clock cycle does not affect the timing for starting the counter operation, it determines the frequency with which all operations take place.

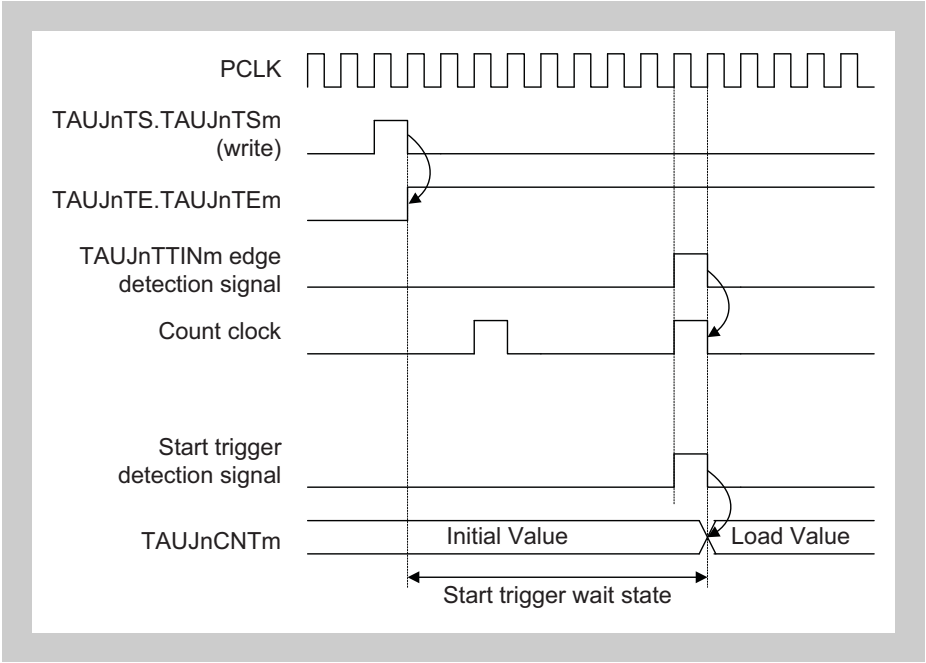


Figure 13-7 Start timing of all other operating modes

13.10 TAUJnTTOUTm Output and INTTAUJnIm Generation when Counter Starts or Restarts (by TAUJnMD0 Bit)

When the counter starts, it is possible to specify whether an INTTAUJnIm is generated using the TAUJnCMORm.TAUJnMD0 bit. The effect of the bit depends on the selected mode, as shown in the following table. The effects of INTTAUJnIm on TAUJnTTOUTm depend on the selected channel operation function.

Table 13-10 Effect of TAUJnCMORm.TAUJnMD0 bit on generation of INTTAUJnIm when counter is triggered

Mode	TAUJnCMORm. TAUJnMD0 bit	INTTAUJnIm generation upon start or restart of counting or upon trigger detection of TAUJnTTINm input signal
Interval Timer Mode Capture Mode Count Capture Mode	0	No
	1	Yes
Capture & One Count Mode Capture & Gate Count Mode	0	No
One Count Mode Gate Count Mode	0/1	No, regardless of the TAUJnCMOR.YAUJnMD0 bit setting

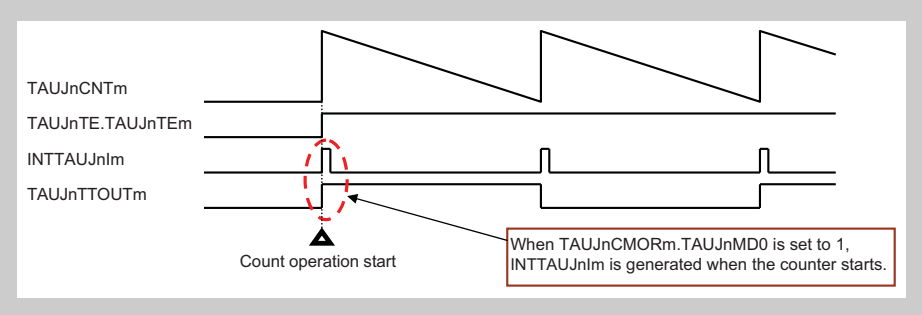


Figure 13-8 INTTAUJnIm generated when counter starts

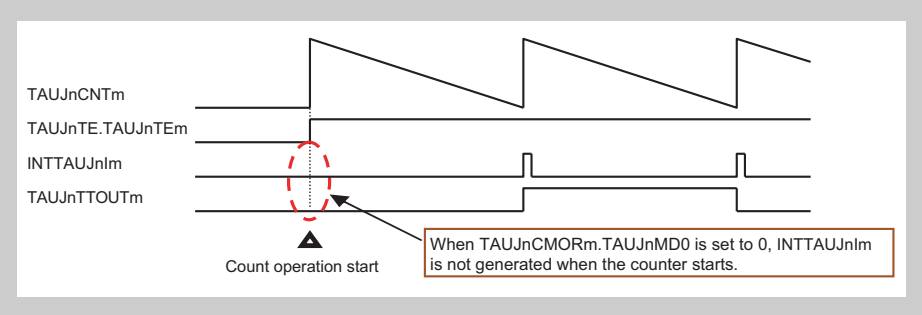


Figure 13-9 INTTAUJnIm not generated when counter starts

13.11 Interrupt Generation upon Overflow

For certain independent channel operation functions, no interrupt is generated even if the counter counts up to FFFF $FFFF_H$ and an overflow occurs. This section describes how it is possible to generate an interrupt, by combining a channel operating in one of these modes with a channel in a different operation mode which counts down.

The appropriate operation mode for the second channel depends on the operation mode of the first channel. Nevertheless, the principle is the same for all combinations:

- For the second channel, specify an operation mode in which the counter counts down and reaches 0000 0000_H at the same time that an overflow occurs on the first channel (TAUJnCNTm = FFFF $FFFF_H$).
- Set TAUJnCDRm of the second channel to FFFF $FFFF_H$
- The two channels must count at the same speed (i.e., they must have the same count clock).
- Both channels are triggered by the same TAUJnTTINm input
- The trigger detection settings (TAUJnCMORm.TAUJnSTS[2:0] and TAUJnCMURm.TAUJnTIS[1:0]) must be identical for both channels

Result: the down-counter of the second channel reaches 0000 0000_H at exactly the same time as the up-counter of the first channel overflows (TAUJnCNTm = FFFF $FFFF_H$). Thus the second channel generates the desired interrupt.

The following sections list the operating modes that count down that are required to match specific operating modes that count up, as well as example timing diagrams.

13.11.1 Capture Mode

Applies to • TAUJnTTINm Input Pulse Interval Measurement Function

Combine with Interval Timer Mode

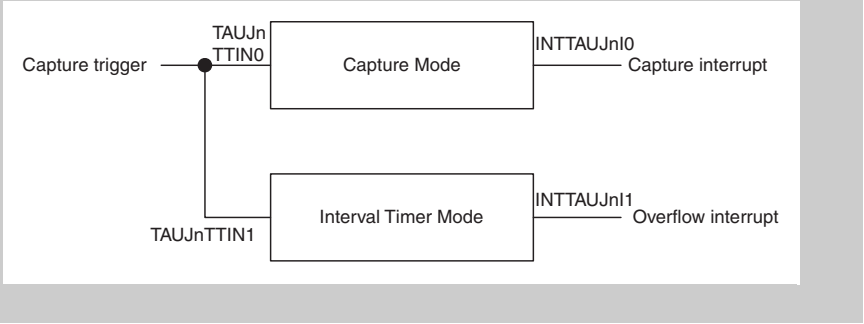


Figure 13-10 Combination of Capture Mode and Interval Timer Mode

Timing diagram

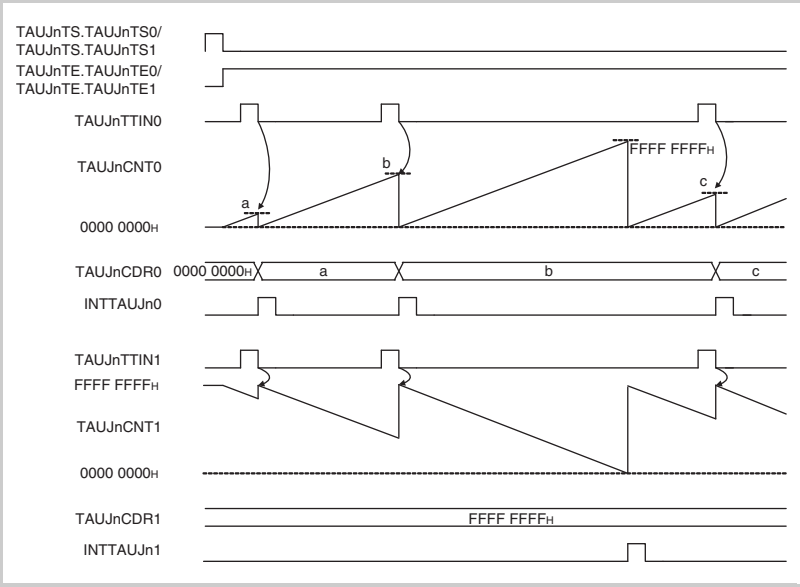


Figure 13-11 Interrupt generation via combination of Capture Mode and Interval Timer Mode

13.11.2 Capture and One Count Mode

Applies to
- TAUJnTTINm Input Signal Width Measurement Function

Combine with
One Count Mode

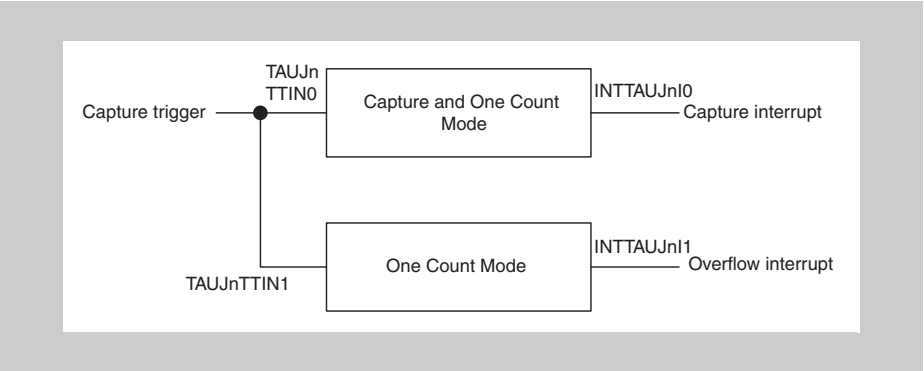


Figure 13-12 Combination of Capture and One Count Mode and One Count Mode

Timing diagram

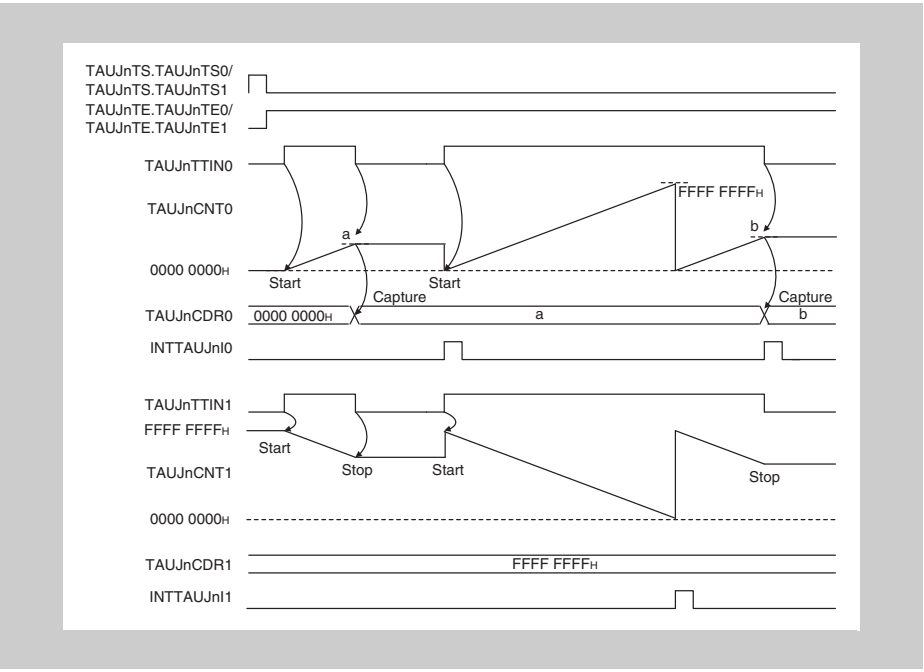


Figure 13-13 Interrupt generation via combination of Capture and One Count Mode and One Count Mode

13.11.3 Count Capture Mode

Applies to • TAUJnTTINm Input Position Detection Function

Combine with Interval Timer Mode

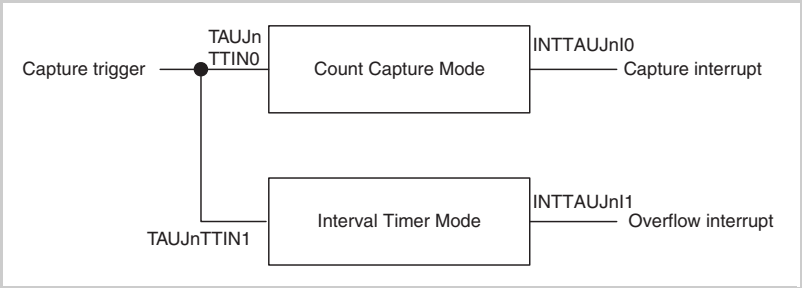


Figure 13-14 Combination of Count Capture Mode and Interval Timer Mode

Timing diagram

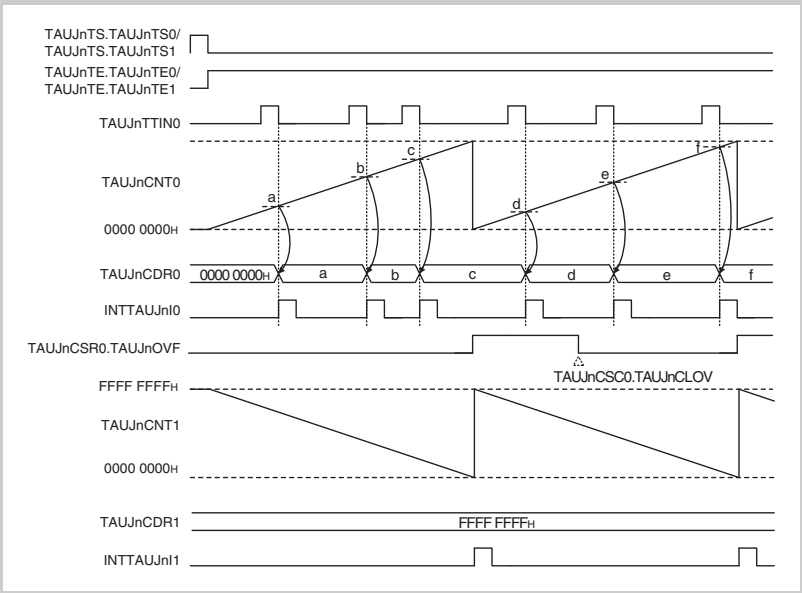


Figure 13-15 Interrupt generation via combination of Count Capture Mode and Interval Timer Mode

In the above timing diagram, TAUJnCSRm.TAUJnOVF is set to 1 when TAUJnCNTm overflows. TAUJnCSRm.TAUJnOVF is cleared by writing 1 to TAUJnCSCm.TAUJnCLOV.

13.11.4 Capture and Gate Count Mode

Applies to • TAUJnTTINm Input Period Count Detection Function

Combine with Gate Count Mode

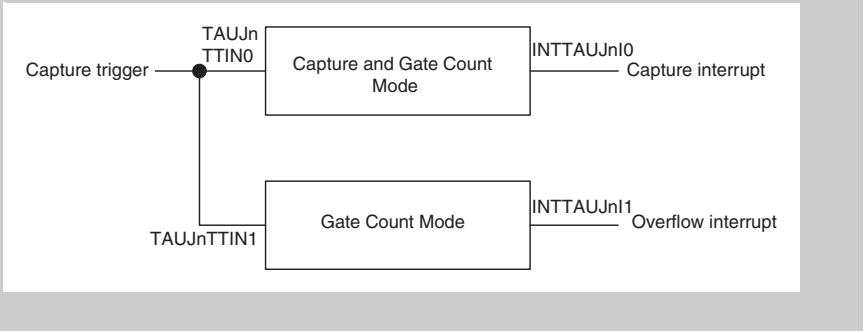


Figure 13-16 Combination of Capture and Gate Count Mode and Gate Count Mode

Timing diagram

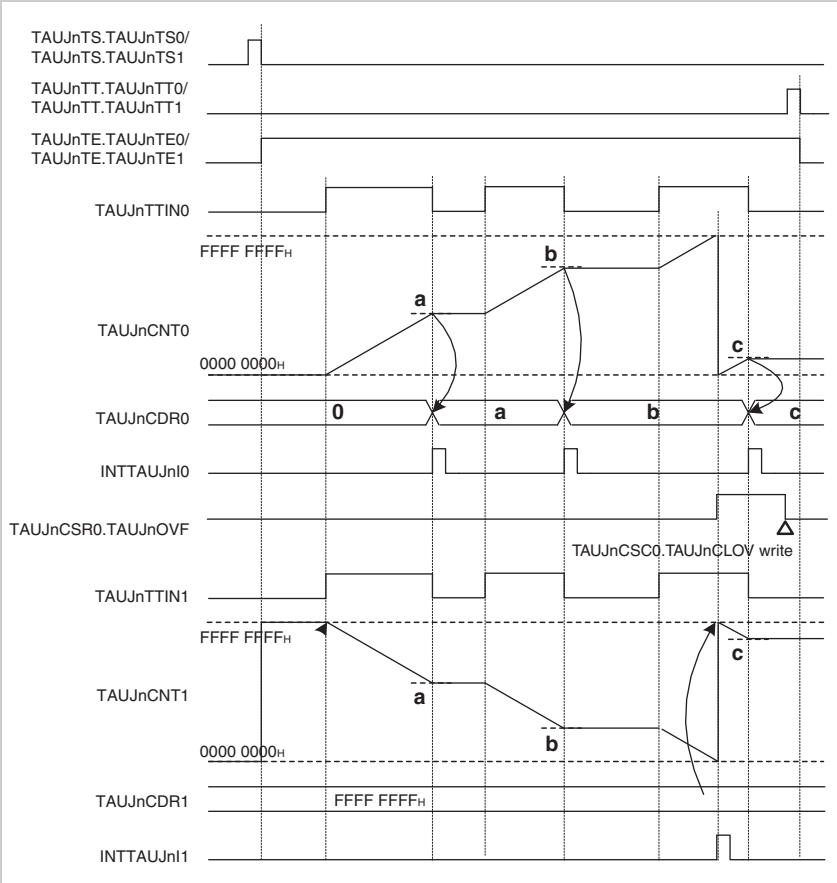


Figure 13-17 Interrupt generation via combination of Capture and Gate Count Mode and Gate Count Mode

In the above timing diagram, TAUJnCSRm.TAUJnOVF is set to 1 when TAUJnCNTm overflows. TAUJnCSRm.TAUJnOVF is cleared by writing 1 to TAUJnCSCm.TAUJnCLOV.

13.12 TAUJnTTINm Edge Detection

Edge detection is based on the operation clock. This means that an edge can only be detected at the next rising edge of the operation clock. This can lead to a maximum delay of one operation clock cycle.

The following figure shows an edge detection timing example.

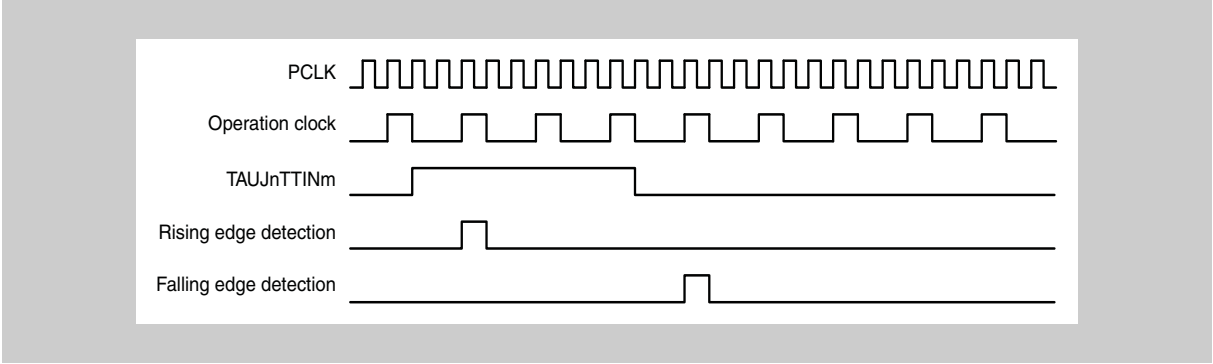


Figure 13-18 Basic edge detection timing

Figure 13-18 "Basic edge detection timing" shows the approximate operation timing. Actually, there is delay due to the noise filter and synchronizer between the TAUJnIm pin and TAUJn.

13.13 Independent Channel Operation Functions

The following sections list the independent channel operation functions provided by TAUJ. For a general overview of independent channel operation, see *13.3 “Functional Description” on page 894*.

13.14 Independent Channel Interrupt Functions

This section describes functions that generate interrupts at regular intervals.

- *13.14.1 “Interval Timer Function”*
- *13.14.2 “TAUJnTTINm Input Interval Timer Function”*

13.14.1 Interval Timer Function

(1) Overview

Summary This function is used as a reference timer for generating timer interrupts (INTTAUJnIm) at regular intervals. When an interrupt is generated, the TAUJnTTOUTm signal toggles, resulting in a square wave.

Prerequisites
- The operation mode must be set to Interval Timer Mode, refer to *Table 13-11 "TAUJnCMORm settings for Interval Timer Function" on page 923*.
- The channel output mode must be set to Independent Channel Output Mode 1, refer to *13.8 "Channel Output Modes" on page 906*.

Description The counter operation is enabled by setting the channel trigger bit (TAUJnTS.TAUJnTSm) to 1. This in turn sets TAUJnTE.TAUJnTEm = 1, enabling count operation. The current value of TAUJnCDRm is loaded to TAUJnCNTm and the counter starts counting down from this value.

When the counter reaches 0000 0000_H, INTTAUJnIm is generated and the TAUJnTTOUTm signal toggles. Next, TAUJnCNTm loads the value of TAUJnCDRm, and then subsequently continues operation.

The value of TAUJnCDRm can be rewritten at any time, and the changed value of TAUJnCDRm is applied the next time the counter starts counting down.

The counter can be stopped by setting TAUJnTT.TAUJnTTm to 1, which in turn sets TAUJnTE.TAUJnTEm to 0. TAUJnCNTm and TAUJnTTOUTm stop but retain their values. The counter can be reset by setting TAUJnTS.TAUJnTSm to 1. The counter can also be forcibly restarted (without stopping it first) by setting TAUJnTS.TAUJnTSm to 1 during operation.

Conditions If the TAUJnCMORm.TAUJnMD0 bit is set to 0, the first interrupt after a start or restart is not generated, and therefore TAUJnTTOUTm does not toggle. This results in an inverted TAUJnTTOUTm signal compared to when TAUJnCMORm.TAUJnMD0 is set to 1. For details refer to *13.10 "TAUJnTTOUTm Output and INTTAUJnIm Generation when Counter Starts or Restarts (by TAUJnMD0 Bit)" on page 913*.

(2) Equations

INTTAUJnIm cycle = count clock cycle x (TAUJnCDRm + 1)

TAUJnTTOUTm square wave cycle = count clock cycle x (TAUJnCDRm + 1) x 2

Caution The minimum interval timer interrupt cycle is 1/(PCLK/2).

(3) Block diagram and general timing diagram

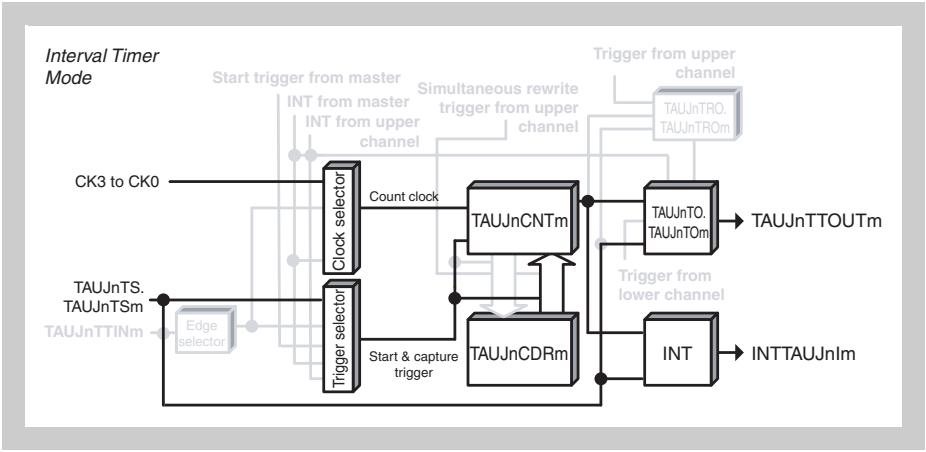


Figure 13-19 Block diagram for Interval Timer Function

The following settings apply to the general timing diagram:

- INTTAUJnIm is generated at operation start (TAUJnCMORm.TAUJnMD0 = 1)

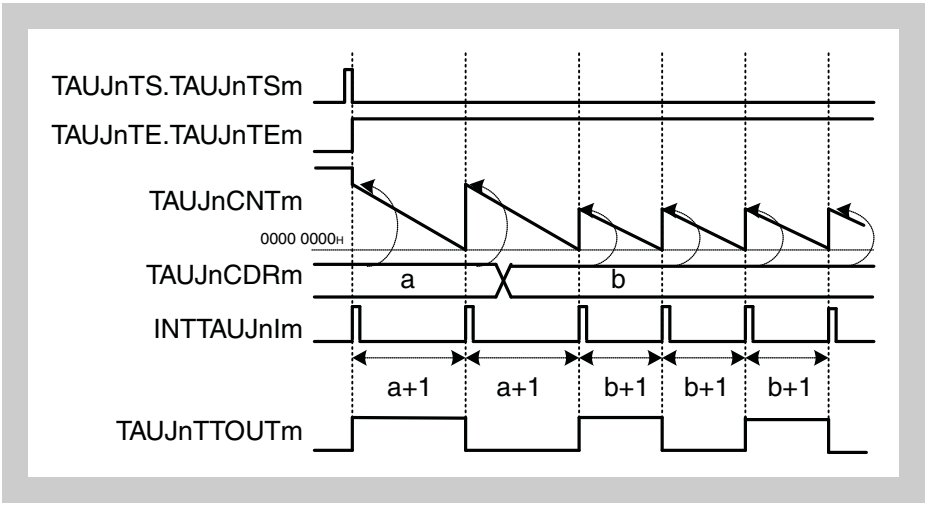


Figure 13-20 General timing diagram for Interval Timer Function

(4) Register settings

(a) TAUJnCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUJn CKS[1:0]		TAUJn CCS[1:0]		TAUJn MAS	TAUJnSTS[2:0]			TAUJn COS[1:0]		-	TAUJnMD[4:1]				TAUJn MD0

Table 13-11 TAUJnCMORm settings for Interval Timer Function

Bit name	Setting
TAUJnCKS[1:0]	00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
TAUJnCCS[1:0]	00: Operation clock is used as the count clock
TAUJnMAS	0: Not used, so set to 0
TAUJnSTS[2:0]	000: Counter triggered by software trigger
TAUJnCOS[1:0]	00: Not used, so set to 00
TAUJnMD[4:1]	0000: Interval Timer Mode
TAUJnMD0	0: INTTAUJnIm not generated and TAUJnTTOUTm does not toggle at operation start or restart 1: Generates INTTAUJnIm and toggles TAUJnTTOUTm at operation start or restart

(b) TAUJnCMURm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-														TAUJnTIS[1:0]	

Table 13-12 TAUJnCMURm settings for Interval Timer Function

Bit name	Setting
TAUJnTIS[1:0]	00: Not used, so set to 00

(c) Channel output mode

Table 13-13 Control bit settings for Independent Channel Output Mode 1

Bit name	Setting
TAUJnTOE.TAUJnTOEm	1: Enable the independent channel output mode.
TAUJnTOM.TAUJnTOMm	0: Independent channel output
TAUJnTOC.TAUJnTOCm	0: Operation mode 1 (= Toggle mode if TAUJnTOM.TAUJnTOMm = 0)
TAUJnTOL.TAUJnTOLm	0: Positive logic

Note The channel output mode can also be set to Channel Output Mode Controlled by Software by setting TAUJnTOE.TAUJnTOEm = 0. TAUJnTTOUTm can then be controlled independently of the interrupts. For details refer to *13.8 "Channel Output Modes" on page 906*.

(d) Simultaneous rewrite

The simultaneous rewrite registers (TAUJnRDE and TAUJnRDM) cannot be used with the Interval Timer Function. Therefore, these registers must be set to 0.

Table 13-14 Simultaneous rewrite settings for Interval Timer Function

Bit name	Setting
TAUJnRDE.TAUJnRDEm	0: Disables simultaneous rewrite
TAUJnRDM.TAUJnRDMm	0: When simultaneous rewrite is disabled (TAUJnRDE.TAUJnRDEm = 0), set these bits to 0

(5) Operating procedure for Interval Timer Function

Table 13-15 Operating procedure for Interval Timer Function

	Operation	Status of TAUJn
Initial channel setting	Set the TAUJnCMORm and TAUJnCMURm registers as described in *Table 13-11 "TAUJnCMORm settings for Interval Timer Function" on page 923* and *Table 13-12 "TAUJnCMURm settings for Interval Timer Function" on page 923*. Set the value of the TAUJnCDRm register. Set the channel output mode by setting the control bits as described in *Table 13-13 "Control bit settings for Independent Channel Output Mode 1" on page 924*.	Channel operation is stopped.
Start operation	Set TAUJnTS.TAUJnTSm to 1. TAUJnTS.TAUJnTSm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TEm is set to 1 and the counter starts. TAUJnCNTm loads the TAUJnCDRm value. When TAUJnCMORm.TAUJnMD0 = 1, INTTAUJnIm is generated and TAUJnTTOUTm toggles.
During operation	The TAUJnCDRm register value can be changed at any time. The TAUJnCNTm register can be read at all times.	TAUJnCNTm counts down. When the counter reaches 0000_H: • TAUJnCNTm reloads the TAUJnCDRm value, and then continues count operation. • INTTAUJnIm is generated and TAUJnTTOUTm toggles.
Stop operation	Set TAUJnTT.TAUJnTTm to 1. TAUJnTT.TAUJnTTm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is cleared to 0 and the counter stops. TAUJnCNTm and TAUJnTTOUTm stop and retain their current values.

Restart

(6) Specific timing diagrams

(a) TAUJnCDRm = 0000 0000$_H$, count clock = PCLK/2

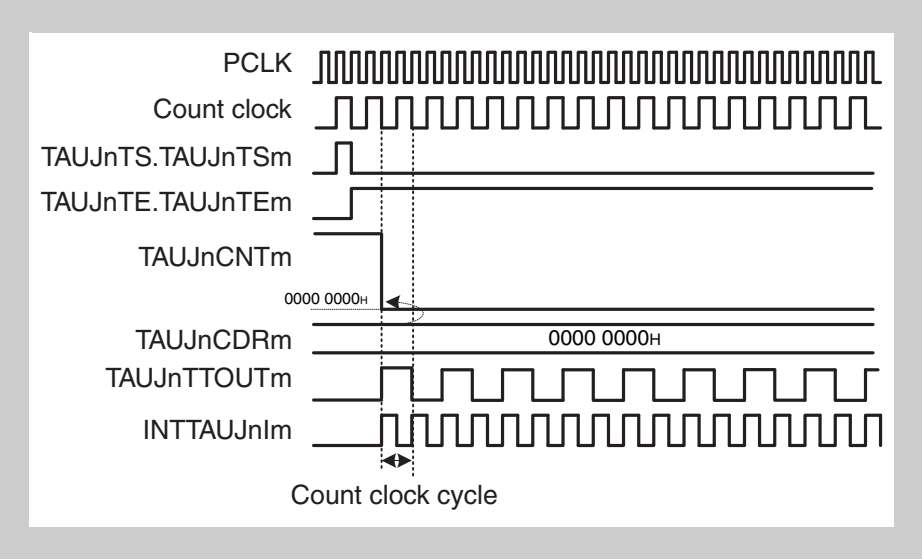


Figure 13-21 TAUJnCDRm = 0000 0000$_H$, count clock = PCLK/2

- If TAUJnCDRm = 0000 0000$_H$ and the count clock = PCLK/2[1], the TAUJnCDRm value is loaded to TAUJnCNTm every count clock, meaning that TAUJnCNTm is always 0000 0000$_H$.
- INTTAUJnIm is generated every count clock, resulting in TAUJnTTOUTm toggling every count clock.

(b) TAUJnCDRm = 0000 0000$_H$, count clock = PCLK

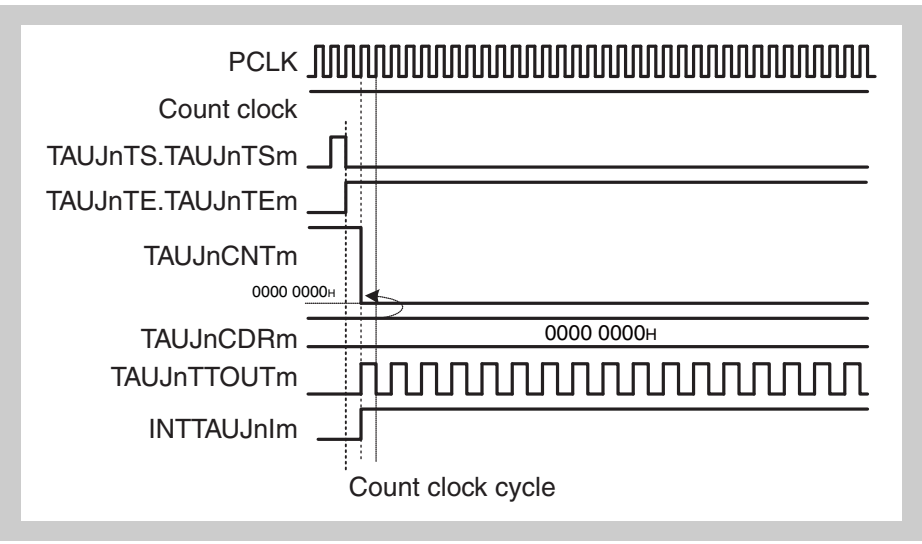


Figure 13-22 TAUJnCDRm = 0000 0000$_H$, count clock = PCLK

- If TAUJnCDRm = 0000 0000$_H$ and the count clock = PCLK, the TAUJnCDRm value is loaded to TAUJnCNTm every PCLK clock, meaning that TAUJnCNTm is always 0000 0000$_H$.
- INTTAUJnIm is generated continuously, resulting in TAUJnTTOUTm toggling every PCLK clock.

(c) Operation stop and restart

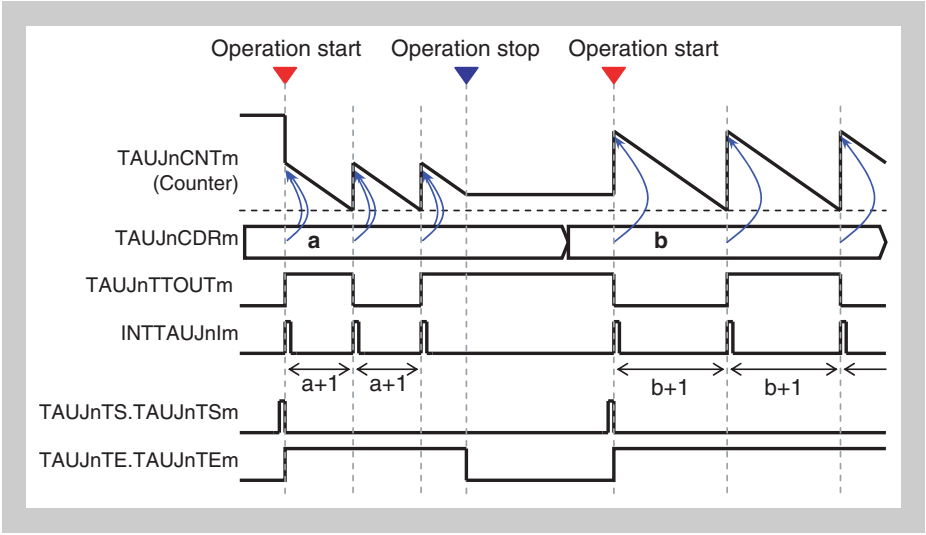


Figure 13-23 Operation stop and restart, TAUJnCMORm.TAUJnMD0 = 1

- The counter can be stopped by setting TAUJnTT.TAUJnTTm to 1, which in turn sets TAUJnTE.TAUJnTEm to 0.
- TAUJnCNTm and TAUJnTTOUTm stop but retain their values.
- The counter can be restarted by setting TAUJnTS.TAUJnTSm to 1.

(d) Forced restart

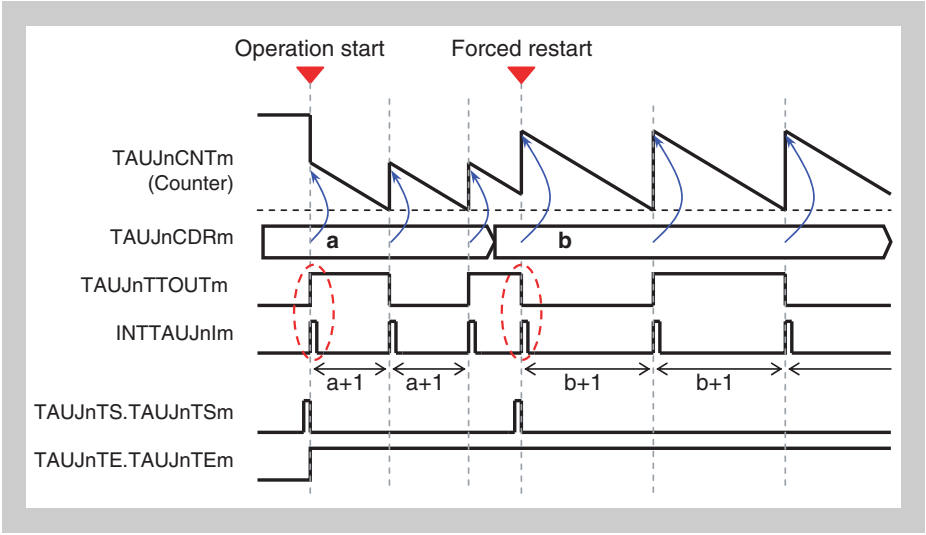


Figure 13-24 Forced restart operation, TAUJnCMORm.TAUJnMD0 = 1

- The counter can be forcibly restarted (without stopping it first) by setting TAUJnTS.TAUJnTSm to 1 during operation.
- If the TAUJnCMORm.TAUJnMD0 bit is set to 1, the first interrupt after a start or restart is generated.

13.14.2 TAUJnTTINm Input Interval Timer Function

(1) Overview

Summary This function is used as a reference timer for generating timer interrupts (INTTAUJnIm) at regular intervals or when a valid TAUJnTTINm input edge is detected. When an interrupt is generated, the TAUJnTTOUTm signal toggles, resulting in a square wave.

Prerequisites

- The operation mode must be set to Interval Timer Mode, refer to *Table 13-16 "TAUJnCMORm settings for TAUJnTTINm Input Interval Timer Function" on page 930*.
- The channel output mode must be set to Independent Channel Output Mode 1, refer to *13.8 "Channel Output Modes" on page 906*.

Description This function operates in an identical manner to the Interval Timer Function (see *13.14.1 "Interval Timer Function" on page 921*), except that this function is restarted by a valid TAUJnTTINm input edge. The type of edge used as the trigger is specified using the TAUJnCMURm.TAUJnTIS[1:0] bits. Either rising edge, falling edge, or rising and falling edge can be selected.

(2) Equations

INTTAUJnIm cycle = count clock cycle x (TAUJnCDRm + 1)

TAUJnTTOUTm square wave cycle = count clock cycle x (TAUJnCDRm + 1) x 2

(3) Block diagram and general timing diagram

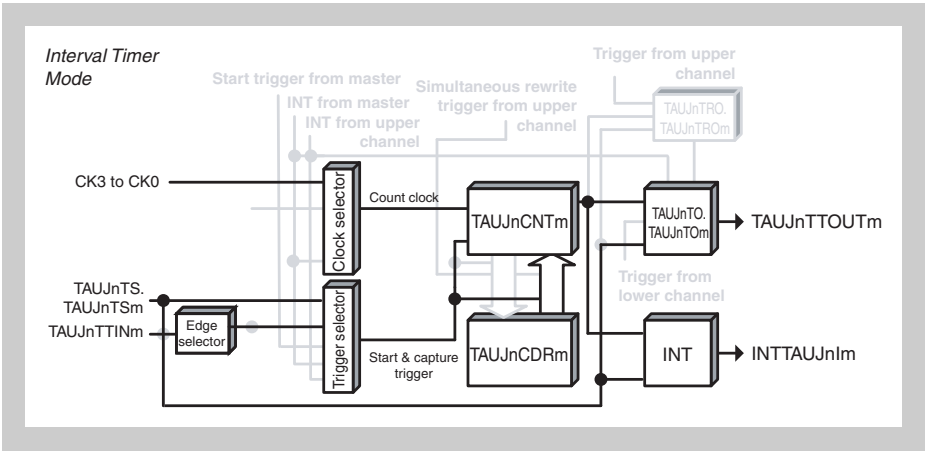


Figure 13-25 Block diagram for TAUJnTTINm Input Interval Timer Function

The following settings apply to the general timing diagram:

- INTTAUJnIm is generated at operation start (TAUJnCMORm.TAUJnMD0 = 1).
- Rising edge detection (TAUJnCMURm.TAUJnTIS[1:0] = 01_B)

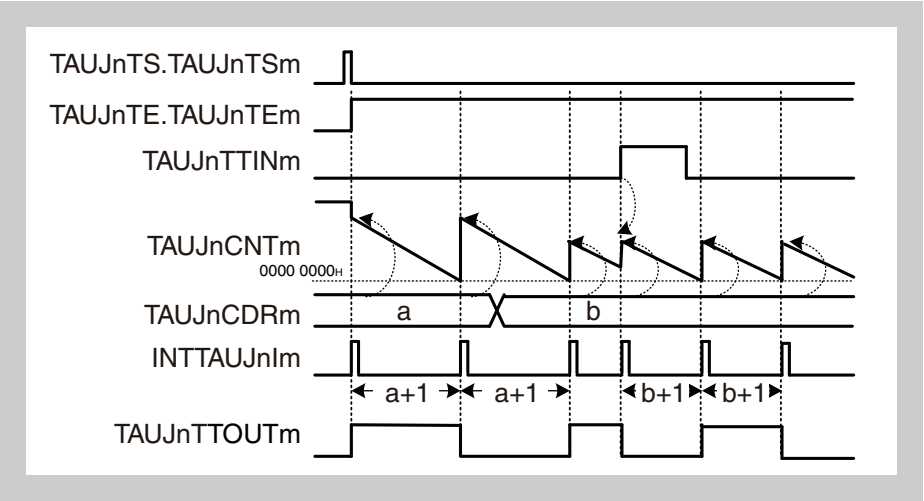


Figure 13-26 General timing diagram for TAUJnTTINm Input Interval Timer Function

(4) Register settings

(a) TAUJnCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUJn CKS[1:0]		TAUJn CCS[1:0]		TAUJn MAS	TAUJnSTS[2:0]			TAUJn COS[1:0]		-	TAUJnMD[4:1]				TAUJn MD0

Table 13-16 TAUJnCMORm settings for TAUJnTTINm Input Interval Timer Function

Bit name	Setting
TAUJnCKS[1:0]	00: Prescaler output= CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
TAUJnCCS[1:0]	00: Operation clock is used as the count clock
TAUJnMAS	0: Not used, so set to 0
TAUJnSTS[2:0]	001: Valid TAUJnTTINm input edge signal is used as the external start trigger
TAUJnCOS[1:0]	00: Not used, so set to 00
TAUJnMD[4:1]	0000: Interval Timer Mode
TAUJnMD0	0: INTTAUJnIm not generated and TAUJnTTOUTm does not toggle at operation start 1: Generates INTTAUJnIm and toggles TAUJnTTOUTm at operation start

(b) TAUJnCMURm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-														TAUJnTIS[1:0]	

Table 13-17 TAUJnCMURm settings for TAUJnTTINm Input Interval Timer Function

Bit name	Setting
TAUJnTIS[1:0]	00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection (low width measurement) 11: Rising and falling edge detection (high width measurement)

(c) Channel output mode

Table 13-18 Control bit settings for Independent Channel Output Mode 1

Bit name	Setting
TAUJnTOE.TAUJnTOEm	1: Enable the independent channel output mode.
TAUJnTO.TAUJnTOm	0: Low level 1: High level
TAUJnTOM.TAUJnTOMm	0: Independent channel output
TAUJnTOC.TAUJnTOCm	0: Operation mode 1 (= Toggle mode if TAUJnTOM.TAUJnTOMm = 0)
TAUJnTOL.TAUJnTOLm	0: Positive logic

Note The channel output mode can also be set to Channel Output Mode Controlled by Software by setting TAUJnTOE.TAUJnTOEm = 0. TAUJnTTOUTm can then be controlled independently of the interrupts. For details refer to *13.8 "Channel Output Modes" on page 906*.

(d) Simultaneous rewrite

The simultaneous rewrite registers (TAUJnRDE and TAUJnRDM) cannot be used with the TAUJnTTINm Input Interval Timer Function. Therefore, these registers must be set to 0.

Table 13-19 Simultaneous rewrite settings for TAUJnTTINm Input Interval Timer Function

Bit name	Setting
TAUJnRDE.TAUJnRDEm	0: Disables simultaneous rewrite
TAUJnRDM.TAUJnRDMm	0: When simultaneous rewrite is disabled (TAUJnRDE.TAUJnRDEm = 0), set these bits to 0

(5) Operating procedure for TAUJnTTINm Input Interval Timer Function

Table 13-20 Operating procedure for TAUJnTTINm Input Interval Timer Function

	Operation	Status of TAUJn
Initial channel setting	Set the TAUJnCMORm register and TAUJnCMURm registers as described in *Table 13-16 "TAUJnCMORm settings for TAUJnTTINm Input Interval Timer Function" on page 930* and *Table 13-17 "TAUJnCMURm settings for TAUJnTTINm Input Interval Timer Function" on page 930*. Set the value of the TAUJnCDRm register. Set the channel output mode by setting the control bits as described in *Table 13-18 "Control bit settings for Independent Channel Output Mode 1" on page 931*.	Channel operation is stopped.
Start operation	Set TAUJnTS.TAUJnTSm to 1. TAUJnTS.TAUJnTSm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is set to 1 and the counter starts. TAUJnCNTm loads the TAUJnCDRm value. When TAUJnCMORm.TAUJnMD0 = 1, INTTAUJnIm is generated and TAUJnTTOUTm toggles.
During operation	The values of the TAUJnCMURm.TAUJnTIS[1:0] and the TAUJnCDRm register can be changed at any time. The TAUJnCNTm register can be read at all times. Detection of TAUJnTTINm edge	TAUJnCNTm counts down. When the counter reaches 0000 0000_H: • TAUJnCNTm reloads the TAUJnCDRm value and continues count operation. • INTTAUJnIm is generated and TAUJnTTOUTm toggles. When a TAUJnTTINm input valid edge is detected during count operation, TAUJnCNTm reloads the TAUJnCDRm value and continues count operation. Afterwards, this procedure is repeated.
Stop operation	Set TAUJnTT.TAUJnTTm to 1. TAUJnTT.TAUJnTTm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is cleared to 0 and the counter stops. TAUJnCNTm and TAUJnTTOUTm stop and retain their current values.

Restart

(6) Specific timing diagrams

The timing diagrams in *13.14.1 "Interval Timer Function" on page 921* also apply, except for this function the counter can also be restarted by a valid TAUJnTTINm input edge.

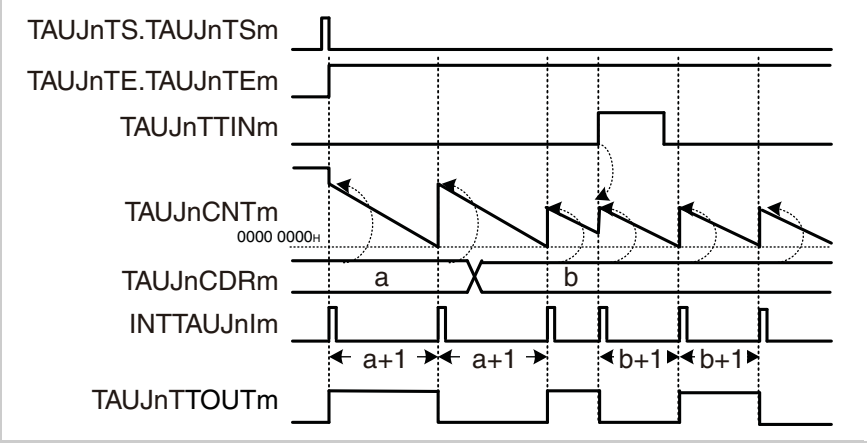


Figure 13-27 Counter triggered by rising TAUJnTTINm input edge (TAUJnCMURm.TAUJnTIS[1:0] = 01_B), TAUJnCMORm.TAUJnMD0 = 1

- If a valid TAUJnTTINm input edge is detected, an interrupt is generated which causes TAUJnTTOUTm to toggle. In this example, the valid edge is a rising edge (TAUJnCMURm.TAUJnTIS[1:0] = 01_B).

13.15 Independent Channel Signal Measurement Functions

This section describes functions that measure the widths of an individual TAUJnTTINm pulse or the total width of successive TAUJnTTINm pulses. It also describes functions that measure the interval of the signal or that compare the width of a pulse with a reference value.

- *13.15.1 “TAUJnTTINm Input Pulse Interval Measurement Function”*
- *13.15.2 “TAUJnTTINm Input Signal Width Measurement Function”*
- *13.15.3 “Overflow Interrupt Output Function (During TAUJnTTINm Width Measurement)”*
- *13.15.4 “TAUJnTTINm Input Period Count Detection Function”*
- *13.15.5 “Overflow Interrupt Output Function (During TAUJnTTINm Input Period Count Detection)”*

13.15.1 TAUJnTTINm Input Pulse Interval Measurement Function

(1) Overview

Summary This function captures the count value and uses this value and the overflow bit TAUJnCSRm.TAUJnOVF to measure the interval of the TAUJnTTINm input signal.

Prerequisites
- The operation mode must be set to Capture Mode, refer to *Table 13-22 "TAUJnCMORm settings for TAUJnTTINm Input Pulse Interval Measurement Function" on page 937*.
- TAUJnTTOUTm is not used for this function.

Description The counter operation is enabled by setting the channel trigger bit (TAUJnTS.TAUJnTSm) to 1. This in turn sets TAUJnTE.TAUJnTEm = 1, enabling count operation. The counter TAUJnCNTm starts counting up from 0000 0000_H. When a valid TAUJnTTINm edge is detected, the value of TAUJnCNTm is captured, transferred to TAUJnCDRm, and an interrupt INTTAUJnIm is generated. The counter resets to 0000 0000_H and subsequently continues operation.

If the counter reaches FFFF $FFFF_H$ before a valid TAUJnTTINm edge is detected, it overflows to 0000 0000_H. The counter is reset to 0000 0000_H and subsequently continues operation. The values transferred to TAUJnCDRm and TAUJnCSRm.TAUJnOVF respectively depend on the values of bits TAUJnCMORm.TAUJnCOS[1:0]:

Table 13-21 Effects of an overflow

TAUJnCMORm.COS[1:0]	When overflow occurs		When a valid TAUJnTTINm input is then detected	
	TAUJnCDRm	TAUJnCSRm.TAUJnOVF	TAUJnCDRm and TAUJnCNTm	TAUJnCSRm.TAUJnOVF
00	Unchanged	0	TAUJnCNTm loaded to TAUJnCDRm	1
01		1		
10	Set to FFFF $FFFF_H$	0	TAUJnCNTm set to 0, TAUJnCDRm unchanged	0
11		1		

When TAUJnCMORm.TAUJnCOS[0] is 1, the overflow bit TAUJnCSRm.TAUJnOVF can only be cleared by setting TAUJnCSCm.TAUJnCLOV to 1.

The combination of the value of TAUJnCDRm and TAUJnCSRm.TAUJnOVF can be used to deduce the interval of the TAUJnTTINm signal. However, if an overflow occurs multiple times before a valid TAUJnTTINm input is detected, the overflow bit TAUJnCSRm.TAUJnOVF cannot indicate this.

The function can be stopped by setting TAUJnTT.TAUJnTTm = 1, which in turn sets TAUJnTE.TAUJnTEm = 0. TAUJnCNTm stops and retains its value. While the function is stopped, TAUJnTTINm input valid edge detection and TAUJnCNTm capture are not performed.

The counter is reset to 0000 0000_H and subsequently continues operation. The counter can also be forcibly restarted (without stopping it first) by setting TAUJnTS.TAUJnTSm = 1 during operation.

Conditions If the TAUJnCMORm.TAUJnMD0 bit is set to 0, the first interrupt after a start or restart is not generated. For details refer to *13.10 "TAUJnTTOUTm Output and INTTAUJnIm Generation when Counter Starts or Restarts (by TAUJnMD0 Bit)" on page 913*.

Note When TAUJnCMORm.TAUJnCOS[1:0] = 11_B, the value of TAUJnCNTm is *not* loaded to TAUJnCDRm when the first valid TAUJnTTINm input edge occurs after an overflow. However, an interrupt is generated.

(2) Equations

TAUJnTTINm input pulse interval = count clock cycle x [(TAUJnCSRm.TAUJnOVF x (FFFF $FFFF_H$ + 1)) + TAUJnCDRm capture value + 1]

(3) Block diagram and general timing diagram

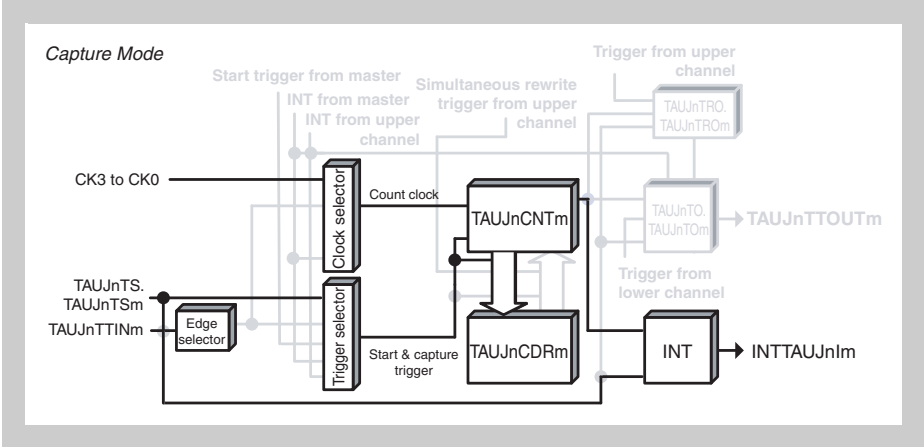


Figure 13-28 Block diagram for TAUJnTTINm Input Pulse Interval Measurement Function

The following settings apply to the general timing diagram:

- INTTAUJnIm not generated at operation start (TAUJnCMORm.TAUJnMD0 = 0)
- Falling edge detection (TAUJnCMURm.TAUJnTIS[1:0] = 00_B)
- When a valid TAUJnTTINm input is detected after an overflow TAUJnCDRm is changed and TAUJnCSRm.TAUJnOVF is set to 1 (TAUJnCMORm.TAUJnCOS[1:0] = 00_B)

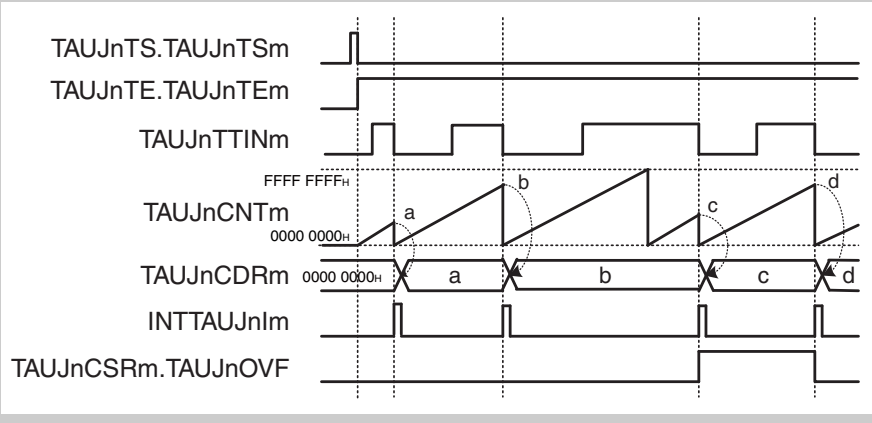


Figure 13-29 General timing diagram for TAUJnTTINm Input Pulse Interval Measurement Function

(4) Register settings

(a) TAUJnCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUJn CKS[1:0]		TAUJn CCS[1:0]		TAUJn MAS	TAUJnSTS[2:0]			TAUJn COS[1:0]		-	TAUJnMD[4:1]				TAUJn MD0

Table 13-22 TAUJnCMORm settings for TAUJnTTINm Input Pulse Interval Measurement Function

Bit name	Setting
TAUJnCKS[1:0]	00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
TAUJnCCS[1:0]	00: Operation clock is used as the count clock
TAUJnMAS	0: Not used, so set to 0
TAUJnSTS[2:0]	001: Valid edge of the TAUJnTTINm input signal is the external capture trigger
TAUJnCOS[1:0]	See *Table 13-21 “Effects of an overflow” on page 935*
TAUJnMD[4:1]	0010: Capture Mode
TAUJnMD0	0: INTTAUJnIm not generated at operation start 1: Generates INTTAUJnIm at operation start

(b) TAUJnCMURm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-														TAUJnTIS[1:0]	

Table 13-23 TAUJnCMURm settings for TAUJnTTINm Input Pulse Interval Measurement Function

Bit name	Setting
TAUJnTIS[1:0]	00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection (low width measurement) 11: Rising and falling edge detection (high width measurement)

(c) Channel output mode

The channel output mode is not used by this function. However, it can be used in Independent Channel Output Mode Controlled by Software.

(d) Simultaneous rewrite

The simultaneous rewrite registers (TAUJnRDE and TAUJnRDM) cannot be used with the TAUJnTTINm Input Pulse Interval Measurement Function. Therefore, these registers must be set to 0.

Table 13-24 Simultaneous rewrite settings for TAUJnTTINm Input Pulse Interval Measurement Function

Bit name	Setting
TAUJnRDE.TAUJnRDEm	0: Disables simultaneous rewrite
TAUJnRDM.TAUJnRDMm	0: When simultaneous rewrite is disabled (TAUJnRDE.TAUJnRDEm = 0), set these bits to 0

(5) Operating procedure for TAUJnTTINm Input Pulse Interval Measurement Function

Table 13-25 Operating procedure for TAUJnTTINm Input Pulse Interval Measurement Function

	Operation	Status of TAUJn
Initial channel setting	Set the TAUJnCMORm register and TAUJnCMURm registers as described in *Table 13-22 "TAUJnCMORm settings for TAUJnTTINm Input Pulse Interval Measurement Function" on page 937* and *Table 13-23 "TAUJnCMURm settings for TAUJnTTINm Input Pulse Interval Measurement Function" on page 937*. Set the value of the TAUJnCDRm register.	Channel operation is stopped.
Start operation	Set TAUJnTS.TAUJnTSm to 1. TAUJnTS.TAUJnTSm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is set to 1 and the counter starts. TAUJnCNTm is cleared to 0000 0000_H. INTTAUJnIm is generated when TAUJnCMORm.TAUJnMD0 is set to 1.
During operation	Detection of TAUJnTTINm edges. The TAUJnCMURm.TAUJnTIS[1:0] bits can be changed at any time. The TAUJnCDRm and TAUJnCSRm registers can be read at any time. Writing 1 to the TAUJnCSCm.TAUJnCLOV bit is possible. (The TAUJnCSRm.TAUJnOVF bit is cleared to 0.)	TAUJnCNTm starts counting up from 0000 0000_H. When a TAUJnTTINm valid edge is detected: • TAUJnCNTm transfers (captures) its value to TAUJnCDRm, and returns to 0000 0000_H. • INTTAUJnIm is then generated. Afterwards, this procedure is repeated.
Stop operation	Set TAUJnTT.TAUJnTTm to 1. TAUJnTT.TAUJnTTm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is cleared to 0 and the counter stops. TAUJnCNTm stops and both it and TAUJnCSRm.TAUJnOVF retain their current values.

Restart

(6) Specific timing diagrams: overflow behavior

(a) TAUJnCMORm.TAUJnCOS[1:0] = 00_B

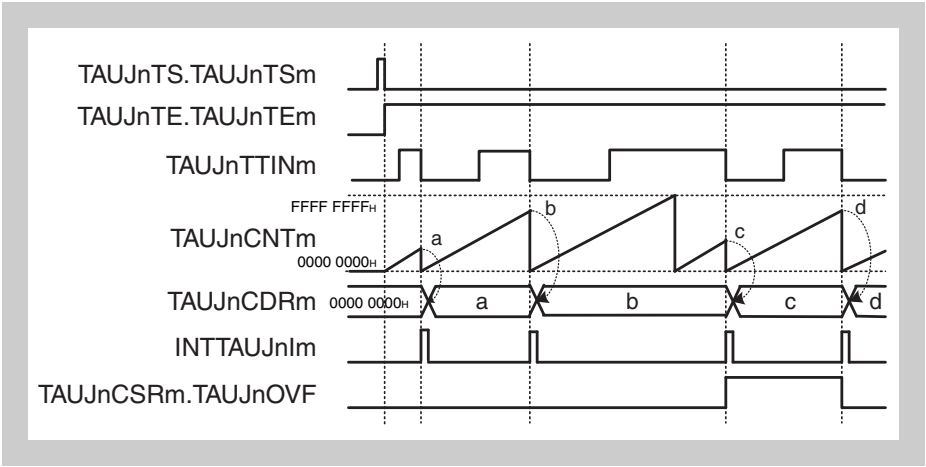


Figure 13-30 TAUJnCMORm.TAUJnCOS[1:0] = 00_B, TAUJnCMORm.TAUJnMD0 = 0, TAUJnCMURm.TAUJnTIS[1:0] = 00_B

- When an overflow occurs, the value of TAUJnCDRm remains unchanged and TAUJnCSRm.TAUJnOVF remains 0.
- Upon detection of the next valid TAUJnTTINm input edge, the value of TAUJnCNTm is loaded to TAUJnCDRm and TAUJnCSRm.TAUJnOVF is set to 1.
- Upon detecting the next valid TAUJnTTINm input edge while no overflow has occurred, TAUJnCSRm.TAUJnOVF is cleared to 0.

(b) TAUJnCMORm.TAUJnCOS[1:0] = 01_B

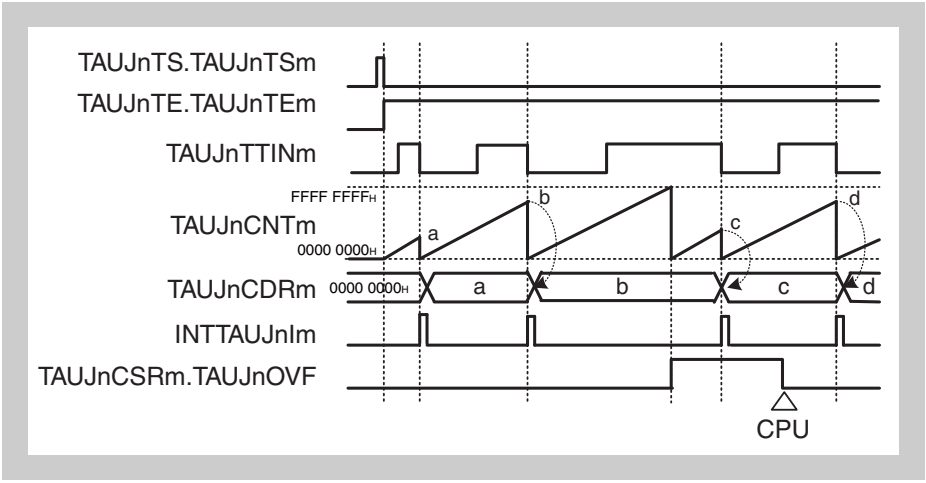


Figure 13-31 TAUJnCMORm.TAUJnCOS[1:0] = 01_B, TAUJnCMORm.TAUJnMD0 = 0, TAUJnCMURm.TAUJnTIS[1:0] = 00_B

- When an overflow occurs, the value of TAUJnCDRm remains unchanged and TAUJnCSRm.TAUJnOVF is set to 1.
- Upon detection of the next valid TAUJnTTINm input edge, the value of TAUJnCNTm is loaded to TAUJnCDRm.
- TAUJnCSRm.TAUJnOVF is only cleared by a CPU command (setting the TAUJnCSCm.TAUJnCLOV bit to 1).

(c) TAUJnCMORm.TAUJnCOS[1:0] = 10_B

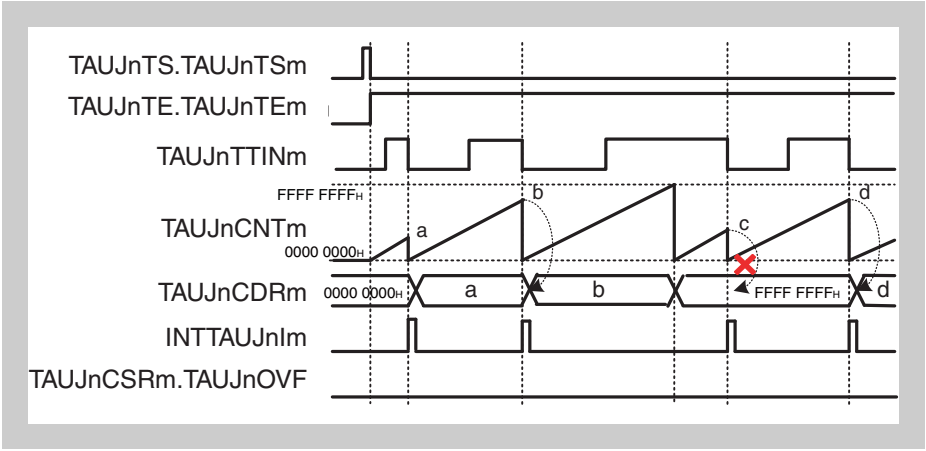


Figure 13-32 TAUJnCMORm.TAUJnCOS[1:0] = 10_B, TAUJnCMORm.TAUJnMD0 = 0, TAUJnCMURm.TAUJnTIS[1:0] = 00_B

- When an overflow occurs, TAUJnCDRm is set to FFFF FFFF$_H$ and TAUJnCSRm.TAUJnOVF remains 0.
- Upon detection of the next valid TAUJnTTINm input edge, TAUJnCNTm is reset to 0, but TAUJnCDRm and TAUJnCSRm.TAUJnOVF remain unchanged.
- Thus, the next TAUJnTTINm input valid edge after the overflow is ignored.

(d) TAUJnCMORm.TAUJnCOS[1:0] = 11_B

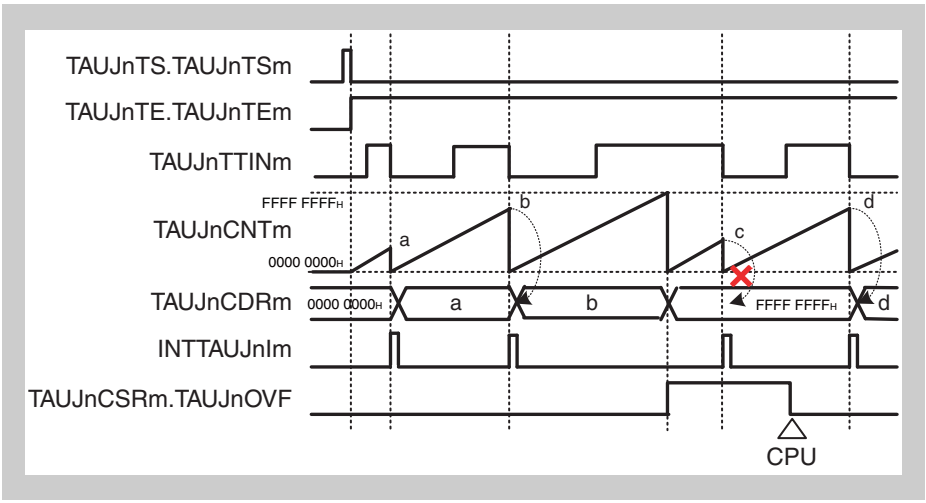


Figure 13-33 TAUJnCMORm.TAUJnCOS[1:0] = 11_B, TAUJnCMORm.TAUJnMD0 = 0, TAUJnCMURm.TAUJnTIS[1:0] = 00_B

- When an overflow occurs, TAUJnCDRm is set to FFFF FFFF$_H$, and TAUJnCSRm.TAUJnOVF is set to 1.
- Upon detection of the next valid TAUJnTTINm input edge, TAUJnCNTm is reset to 0, but TAUJnCDRm and TAUJnCSRm.TAUJnOVF remain unchanged.
- Thus, the next TAUJnTTINm input valid edge after the overflow is ignored.
- TAUJnCSRm.TAUJnOVF is cleared by setting TAUJnCSCm.TAUJnCLOV to 1.

13.15.2 TAUJnTTINm Input Signal Width Measurement Function

(1) Overview

Summary This function measures the width of a TAUJnTTINm input signal.

Prerequisites
- The operation mode must be set to Capture & One Count Mode, refer to *Table 13-27 "TAUJnCMORm settings for TAUJnTTINm Input Signal Width Measurement Function" on page 944*.
- TAUJnTTOUTm is not used for this function.
- TAUJnCMORm.TAUJnMD0 must be set to 0.

Description The counter operation is enabled by setting the channel trigger bit (TAUJnTS.TAUJnTSm) to 1. This in turn sets TAUJnTE.TAUJnTEm = 1, enabling count operation. When a valid TAUJnTTINm start edge is detected, the counter TAUJnCNTm starts counting up from 0000 0000_H. When a valid TAUJnTTINm stop edge is detected, the value of TAUJnCNTm is captured, transferred to TAUJnCDRm, and an interrupt INTTAUJnIm is generated. The counter retains its value and awaits the next valid TAUJnTTINm input start edge.

If the counter reaches FFFF $FFFF_H$ before a valid TAUJnTTINm stop edge is detected, it overflows. The counter is reset to 0000 0000_H and subsequently continues operation. The values transferred to TAUJnCDRm and TAUJnCSRm.TAUJnOVF respectively depend on the values of bits TAUJnCMORm.TAUJnCOS[1:0]:

Table 13-26 Effects of an overflow

TAUJnCMORm. COS[1:0]	When overflow occurs		When a valid TAUJnTTINm input stop edge is detected	
	TAUJnCDRm	TAUJnCSRm. TAUJnOVF	TAUJnCDRm and TAUJnCNTm	TAUJnCSRm. TAUJnOVF
00	Unchanged	0	TAUJnCNTm loaded to TAUJnCDRm	1
01		1		
10	Set to FFFF $FFFF_H$	0	TAUJnCNTm stops counting TAUJnCDRm unchanged	0
11		1		

When TAUJnCMORm.TAUJnCOS[0] is 1, the overflow bit TAUJnCSRm.TAUJnOVF can only be cleared by setting TAUJnCSCm.TAUJnCLOV to 1.

The combination of the value of TAUJnCDRm and TAUJnCSRm.TAUJnOVF can be used to deduce the width of the TAUJnTTINm signal. However, if an overflow occurs multiple times before a valid TAUJnTTINm input is detected, the overflow bit TAUJnCSRm.TAUJnOVF cannot indicate this.

This function cannot be forcibly restarted.

Note When TAUJnCMORm.TAUJnCOS[1] = 1, the value of TAUJnCNTm is not loaded to TAUJnCDRm when the first valid TAUJnTTINm input edge occurs after an overflow. However, an interrupt is generated.

(2) Equations

TAUJnTTINm input signal width = count clock cycle x [(TAUJnCSRm.TAUJnOVF x (FFFF $FFFF_H$ + 1)) + TAUJnCDRm capture value + 1]

(3) Block diagram and general timing diagram

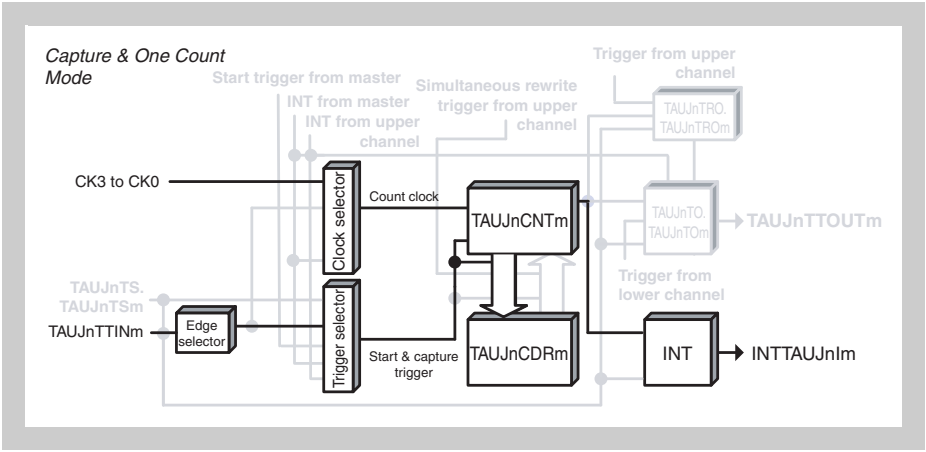


Figure 13-34 Block diagram for TAUJnTTINm Input Signal Width Measurement Function

The following settings apply to the general timing diagram:

- Rising and falling edge detection = high width measurement (TAUJnCMURm.TAUJnTIS[1:0] = 11_B)
- When a valid TAUJnTTINm input is detected after an overflow TAUJnCDRm is changed and TAUJnCSRm.TAUJnOVF is set to 1 (TAUJnCMORm.TAUJnCOS[1:0] = 00_B)

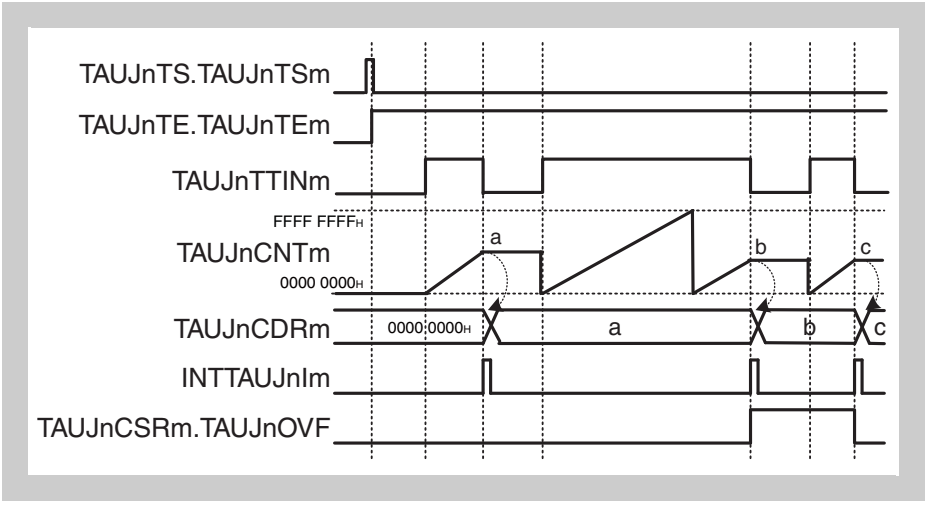


Figure 13-35 General timing diagram for TAUJnTTINm Input Signal Width Measurement Function

(4) Register settings

(a) TAUJnCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUJn CKS[1:0]		TAUJn CCS[1:0]		TAUJn MAS	TAUJnSTS[2:0]			TAUJn COS[1:0]		-	TAUJnMD[4:1]				TAUJn MD0

Table 13-27 TAUJnCMORm settings for TAUJnTTINm Input Signal Width Measurement Function

Bit name	Setting
TAUJnCKS[1:0]	00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
TAUJnCCS[1:0]	00: Operation clock is used as the count clock
TAUJnMAS	0: Not used, so set to 0
TAUJnSTS[2:0]	010: Valid edge of the TAUJnTTINm input signal is the external start trigger and the reverse edge is the stop trigger
TAUJnCOS[1:0]	See *Table 13-26 “Effects of an overflow” on page 942*
TAUJnMD[4:1]	0110: Capture & One Count Mode
TAUJnMD0	0: Disables the start trigger during operation

(b) TAUJnCMURm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-														TAUJnTIS[1:0]	

Table 13-28 TAUJnCMURm settings for TAUJnTTINm Input Signal Width Measurement Function

Bit name	Setting
TAUJnTIS[1:0]	10: Rising and falling edge detection (low width measurement) 11: Rising and falling edge detection (high width measurement)

(c) Channel output mode

Because the channel output mode is not used for this function, clear TAUJnTOE.TAUJnTOEm to 0. However, it can be used in Independent Channel Output Mode Controlled by Software.

(d) Simultaneous rewrite

The simultaneous rewrite registers (TAUJnRDE and TAUJnRDM) cannot be used with the TAUJnTTINm Input Signal Width Measurement Function. Therefore, these registers must be set to 0.

Table 13-29 Simultaneous rewrite settings for TAUJnTTINm Input Signal Width Measurement Function

Bit name	Setting
TAUJnRDE.TAUJnRDEm	0: Disables simultaneous rewrite
TAUJnRDM.TAUJnRDMm	0: When simultaneous rewrite is disabled (TAUJnRDE.TAUJnRDEm = 0), set these bits to 0

(5) Operating procedure for TAUJnTTINm Input Signal Width Measurement Function

Table 13-30 Operating procedure for TAUJnTTINm Input Signal Width Measurement Function

	Operation	Status of TAUJn
Initial channel setting	Set the TAUJnCMORm register and TAUJnCMURm registers as described in *Table 13-22 "TAUJnCMORm settings for TAUJnTTINm Input Pulse Interval Measurement Function" on page 937* and *Table 13-23 "TAUJnCMURm settings for TAUJnTTINm Input Pulse Interval Measurement Function" on page 937*. Set the value of the TAUJnCDRm register.	Channel operation is stopped.
Start operation	Set TAUJnTS.TAUJnTSm to 1. TAUJnTS.TAUJnTSm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is set to 1 and TAUJnCNTm waits for detection of the TAUJnTTINm start edge. When a TAUJnTTINm start is detected, TAUJnCNTm starts counting up.
During operation	Detection of TAUJnTTINm edges. The TAUJnCDRm, TAUJnCNTm, and TAUJnCSRm registers can be read at any time. The TAUJnCSCm.CLOV bit can be set to 1.	TAUJnCNTm starts counting up from 0000 0000_H. When a TAUJnTTINm valid edge is detected: • TAUJnCNTm transfers (captures) its value to TAUJnCDRm, and retains its value. • INTTAUJnIm is then generated. Afterwards, this procedure is repeated.
Stop operation	Set TAUJnTT.TAUJnTTm to 1. TAUJnTT.TAUJnTTm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is cleared to 0 and the counter stops. TAUJnCNTm stops and both it and TAUJnCSRm.TAUJnOVF retain their current values.

Restart

(6) Specific timing diagrams: overflow behavior

(a) TAUJnCMORm.TAUJnCOS[1:0] = 00_B

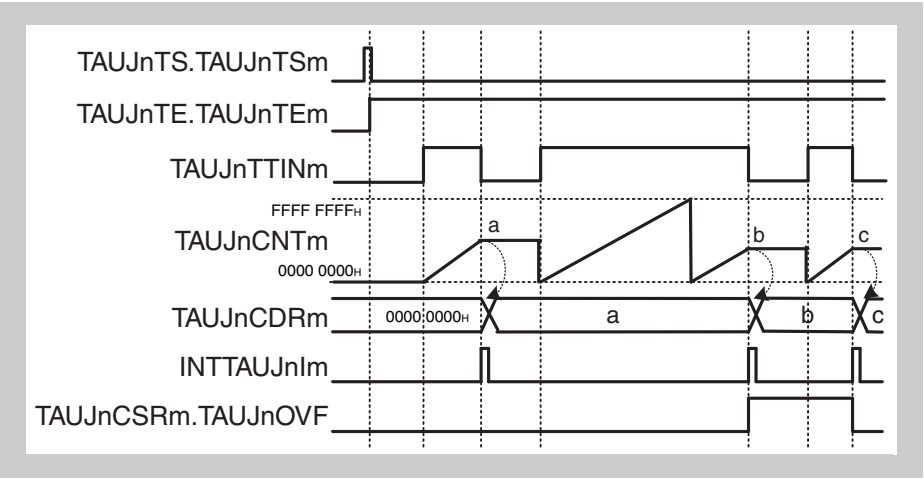


Figure 13-36 TAUJnCMORm.TAUJnCOS[1:0] = 00_B, TAUJnCMORm.TAUJnMD0 = 0, TAUJnCMURm.TAUJnTIS[1:0] = 11_B

- When an overflow occurs, the value of TAUJnCDRm remains unchanged and TAUJnCSRm.TAUJnOVF remains 0.
- Upon detection of the next valid TAUJnTTINm input edge, the value of TAUJnCNTm is loaded to TAUJnCDRm and TAUJnCSRm.TAUJnOVF is set.
- Upon detecting the next valid TAUJnTTINm input edge while no overflow has occurred, TAUJnCSRm.TAUJnOVF is cleared to 0.

(b) TAUJnCMORm.TAUJnCOS[1:0] = 01_B

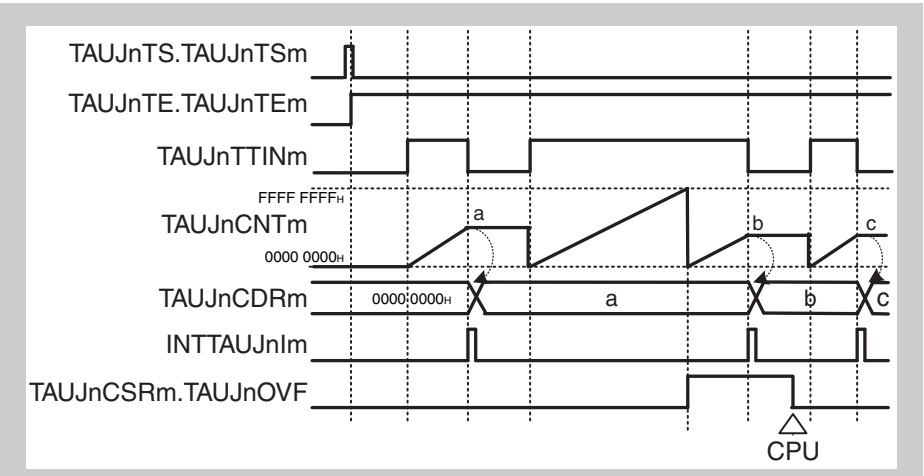


Figure 13-37 TAUJnCMORm.TAUJnCOS[1:0] = 01_B, TAUJnCMORm.TAUJnMD0 = 0, TAUJnCMURm.TAUJnTIS[1:0] = 11_B

- When an overflow occurs, the value of TAUJnCDRm remains unchanged and TAUJnCSRm.TAUJnOVF is set to 1.
- Upon detection of the next valid TAUJnTTINm input edge, the value of TAUJnCNTm is loaded to TAUJnCDRm.
- TAUJnCSRm.TAUJnOVF is only cleared by a CPU command (setting the TAUJnCSCm.TAUJnCLOV bit to 1).

(c) TAUJnCMORm.TAUJnCOS[1:0] = 10_B

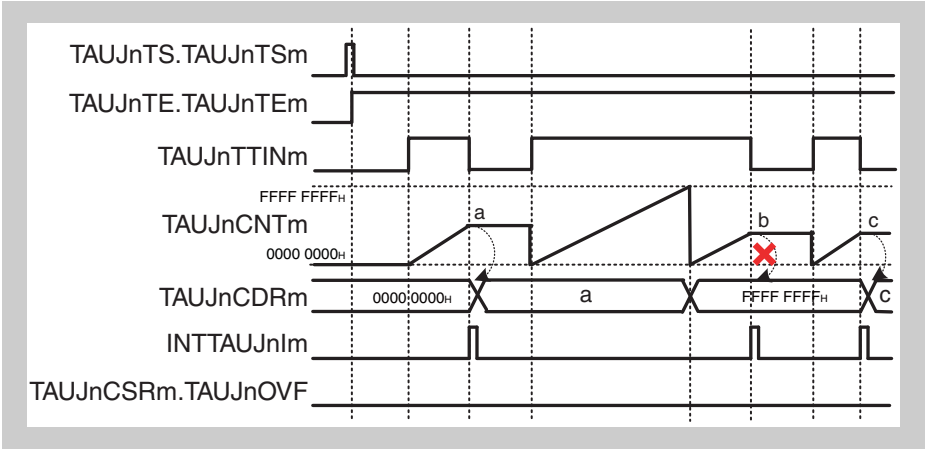


Figure 13-38 TAUJnCMORm.TAUJnCOS[1:0] = 10_B, TAUJnCMORm.TAUJnMD0 = 0, TAUJnCMURm.TAUJnTIS[1:0] = 11_B

- When an overflow occurs, TAUJnCDRm is set to FFFF FFFF$_H$ and TAUJnCSRm.TAUJnOVF remains 0.
- Upon detection of the next valid TAUJnTTINm input edge, TAUJnCNTm is reset to 0, but TAUJnCDRm and TAUJnCSRm.TAUJnOVF remain unchanged.
- Thus, the next TAUJnTTINm input valid edge after the overflow is ignored.

(d) TAUJnCMORm.TAUJnCOS[1:0] = 11_B

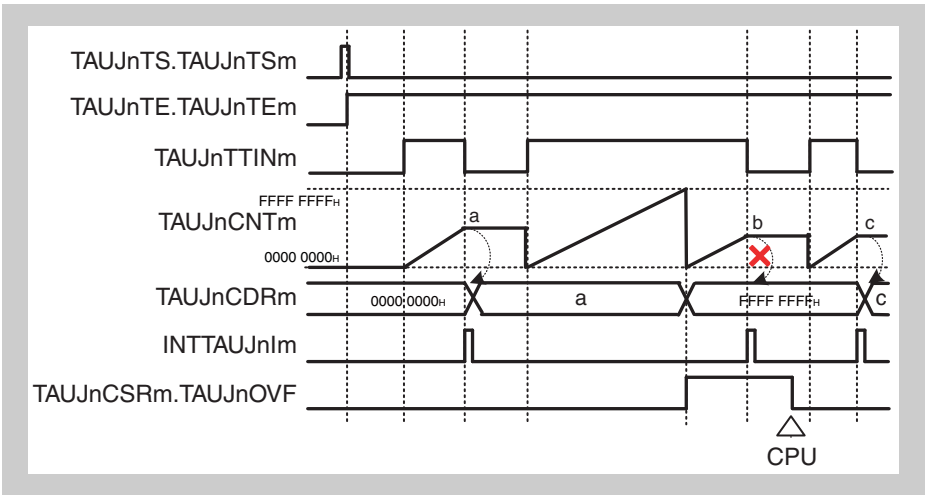


Figure 13-39 TAUJnCMORm.TAUJnCOS[1:0] = 11_B, TAUJnCMORm.TAUJnMD0 = 0, TAUJnCMURm.TAUJnTIS[1:0] = 11_B

- When an overflow occurs, TAUJnCDRm is set to FFFF FFFF$_H$, and TAUJnCSRm.TAUJnOVF is set to 1.
- Upon detection of the next valid TAUJnTTINm input edge, TAUJnCNTm is reset to 0, but TAUJnCDRm and TAUJnCSRm.TAUJnOVF remain unchanged.
- Thus, the next TAUJnTTINm input valid edge after the overflow is ignored.
- TAUJnCSRm.TAUJnOVF is cleared by setting TAUJnCSCm.TAUJnCLOV to 1.

13.15.3 Overflow Interrupt Output Function (During TAUJnTTINm Width Measurement)

(1) Overview

Summary This function measures the width of an individual TAUJnTTINm input signal. An interrupt is generated if (FFFF FFFF$_H$ + 1) is exceeded following TAUJnTTINm input.

Prerequisites
- The operation mode must be set to One Count Mode, refer to *Table 13-31 "TAUJnCMORm settings for Overflow Interrupt Output Function (During TAUJnTTINm Width Measurement)" on page 951*.
- TAUJnTTOUTm is not used for this function.
- The value of TAUJnCDRm must be set to FFFF FFFF$_H$.

Description The counter is enabled by setting the channel trigger bit (TAUJnTS.TAUJnTSm) to 1. This in turn sets TAUJnTE.TAUJnTEm = 1, enabling count operation.

The counter starts when a valid TAUJnTTINm input start edge is detected. FFFF FFFF$_H$ is loaded to TAUJnCNTm and the counter starts counting down.

When a valid stop edge is detected, the counter stops and retains the current value.

When the next TAUJnTTINm input start edge is detected, TAUJnCNTm loads FFFF FFFF$_H$ and starts counting down.

If the counter reaches 0000 0000$_H$ before a stop edge is detected, an interrupt is generated.

Conditions The valid start and stop edges are specified by the TAUJnCMURm.TAUJnTIS[1:0] bits:
- If TAUJnCMURm.TAUJnTIS[1:0] = 10$_B$, the TAUJnTTINm input low width is measured. The start trigger is a falling edge and the stop trigger is a rising edge.
- If TAUJnCMURm.TAUJnTIS[1:0] = 11$_B$, the TAUJnTTINm input high width is measured. The start trigger is a rising edge and the stop trigger is a falling edge.

Note The counter cannot be restarted during operation.

(2) Block diagram and general timing diagram

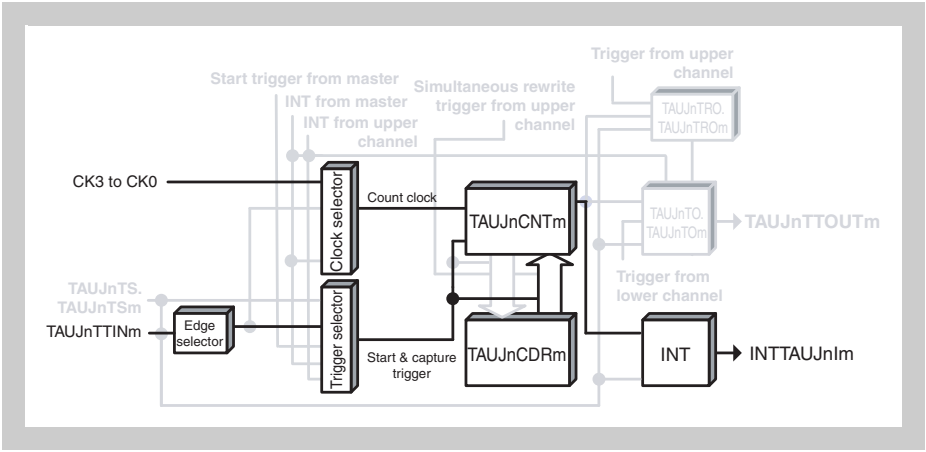


Figure 13-40 Block diagram for Overflow Interrupt Output Function (During TAUJnTTINm Width Measurement)

The following settings apply to the general timing diagram:

- Rising and falling edge detection = high width measurement (TAUJnCMURm.TAUJnTIS[1:0] = 11_B)

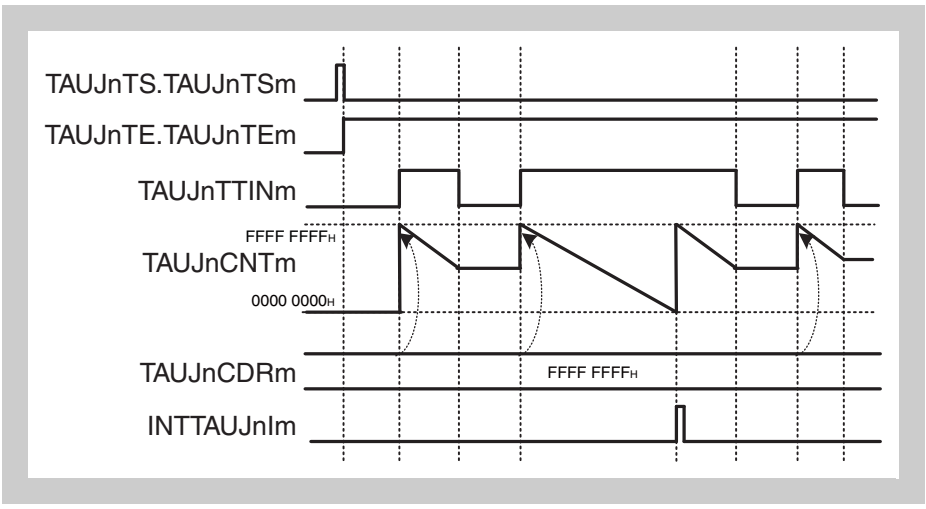


Figure 13-41 General timing diagram for Overflow Interrupt Output Function (During TAUJnTTINm Width Measurement)

(3) Register settings

(a) TAUJnCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUJn CKS[1:0]		TAUJn CCS[1:0]		TAUJn MAS	TAUJnSTS[2:0]			TAUJn COS[1:0]		-	TAUJnMD[4:1]				TAUJn MD0

Table 13-31 TAUJnCMORm settings for Overflow Interrupt Output Function (During TAUJnTTINm Width Measurement)

Bit name	Setting
TAUJnCKS[1:0]	00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
TAUJnCCS[1:0]	00: Operation clock is used as the count clock
TAUJnMAS	0: Not used, so set to 0
TAUJnSTS[2:0]	010: Valid edge of the TAUJnTTINm input signal is the external start trigger and the reverse edge is the stop trigger
TAUJnCOS[1:0]	00: Not used, so set to 00
TAUJnMD[4:1]	0100: One Count Mode
TAUJnMD0	0: Disables the start trigger during operation

(b) TAUJnCMURm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-														TAUJnTIS[1:0]	

Table 13-32 TAUJnCMURm settings Overflow Interrupt Output Function (During TAUJnTTINm Width Measurement)

Bit name	Setting
TAUJnTIS[1:0]	10: Rising and falling edge detection (low width measurement) 11: Rising and falling edge detection (high width measurement)

(c) Channel output mode

Because the channel output mode is not used for this function, clear TAUJnTOE.TAUJnTOEm to 0. However, it can be used in Independent Channel Output Mode Controlled by Software.

(d) Simultaneous rewrite

The simultaneous rewrite registers (TAUJnRDE and TAUJnRDM) cannot be used with the Overflow Interrupt Output Function (During TAUJnTTINm Width Measurement). Therefore, these registers must be set to 0.

(4) Operating procedure for Overflow Interrupt Output Function

Table 13-33 Simultaneous rewrite settings for Overflow Interrupt Output Function (During TAUJnTTINm Width Measurement)

Bit name	Setting
TAUJnRDE.TAUJnRDEm	0: Disables simultaneous rewrite
TAUJnRDM.TAUJnRDMm	0: When simultaneous rewrite is disabled (TAUJnRDE.TAUJnRDEm = 0), set these bits to 0

(During TAUJnTTINm Width Measurement)

Table 13-34 Operating procedure for Overflow Interrupt Output Function (During TAUJnTTINm Width Measurement)

	Operation	Status of TAUJn
Initial channel setting	Set the TAUJnCMORm register and TAUJnCMURm registers as described in *Table 13-31 "TAUJnCMORm settings for Overflow Interrupt Output Function (During TAUJnTTINm Width Measurement)" on page 951* and *Table 13-32 "TAUJnCMURm settings Overflow Interrupt Output Function (During TAUJnTTINm Width Measurement)" on page 951*. Set the value of the TAUJnCDRm register.	Channel operation is stopped.
Start operation	Set TAUJnTS.TAUJnTSm to 1. TAUJnTS.TAUJnTSm is a trigger bit, so it is automatically cleared to 0. Detection of TAUJnTTINm start edge	TAUJnTE.TAUJnTEm is set to 1 and TAUJnCNTm waits for detection of the start edge. When a start edge is detected, TAUJnCNTm loads the TAUJnCDRm value (FFFF FFFF$_H$).
During operation	The TAUJnCNTm register can be read at any time. Detection of TAUJnTTINm edges	TAUJnCNTm counts down. When the counter reaches 0000 0000$_H$: • INTTAUJnIm is generated. When a reverse edge of TAUJnTTINm is detected during count operation: • TAUJnCNTm stops counting and waits for a trigger. Afterwards, this procedure is repeated.
Stop operation	Set TAUJnTT.TAUJnTTm to 1. TAUJnTT.TAUJnTTm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is cleared to 0 and the counter stops. TAUJnCNTm stops and retains its current value.

Restart

13.15.4 TAUJnTTINm Input Period Count Detection Function

(1) Overview

Summary This function measures the cumulative width of a TAUJnTTINm input signal.

Prerequisites
- The operation mode must be set to Capture & Gate Count Mode, refer to *Table 13-35 "TAUJnCMORm settings for TAUJnTTINm Input Period Count Detection Function" on page 956*.
- TAUJnTTOUTm is not used for this function.

Description The counter is enabled by setting the channel trigger bit (TAUJnTS.TAUJnTSm) to 1. This in turn sets TAUJnTE.TAUJnTEm = 1, enabling count operation. The counter awaits a valid TAUJnTTINm input edge.

When a valid TAUJnTTINm input start edge is detected, the counter starts counting from 0000 0000_H.

When a valid TAUJnTTINm input stop edge is detected, the current TAUJnCNTm value is loaded to TAUJnCDRm and an interrupt (INTTAUJnIm) is generated. The counter stops and retains its value until the next valid TAUJnTTINm input start edge is detected.

When the next valid TAUJnTTINm input start edge is detected, the counter resumes counting, starting with its value upon stopping.

If the counter reaches FFFF $FFFF_H$, the TAUJnCSRm.TAUJnOVF bit is set to 1 and the counter restarts counting from 0000 0000_H. The value of TAUJnCSRm.TAUJnOVF is reset by the CPU by setting TAUJnCSCm.TAUJnCLOV = 1.

Note The input TAUJnTTINm is sampled at the frequency of the operation clock, specified by the TAUJnCMORm.TAUJnCKS[1:0] bits.

Conditions The valid start and stop edges are specified by the TAUJnCMURm.TAUJnTIS[1:0] bits:
- If TAUJnCMURm.TAUJnTIS[1:0] = 10_B, the TAUJnTTINm input low period is counted. The start trigger is a falling edge and the stop trigger is a rising edge.
- If TAUJnCMURm.TAUJnTIS[1:0] = 11_B, the TAUJnTTINm input high period is counted. The start trigger is a rising edge and the stop trigger is a falling edge.

(2) Equations

Cumulative TAUJnTTINm input width =
count clock cycle × ((FFFF $FFFF_H$ × TAUJnCSRm.TAUJnOVF) + (TAUJnCDRm capture value + 1))

<R> **Caution** If a capture signal in input when the counter value reaches the upper limit (0FFFF FFFFH), the overflow flag (OVF) that should be set in response to the occurrence of an overflow in the next count clock cycle will not be set. The OVF flag is set normally at all other times.

(3) Block diagram and general timing diagram

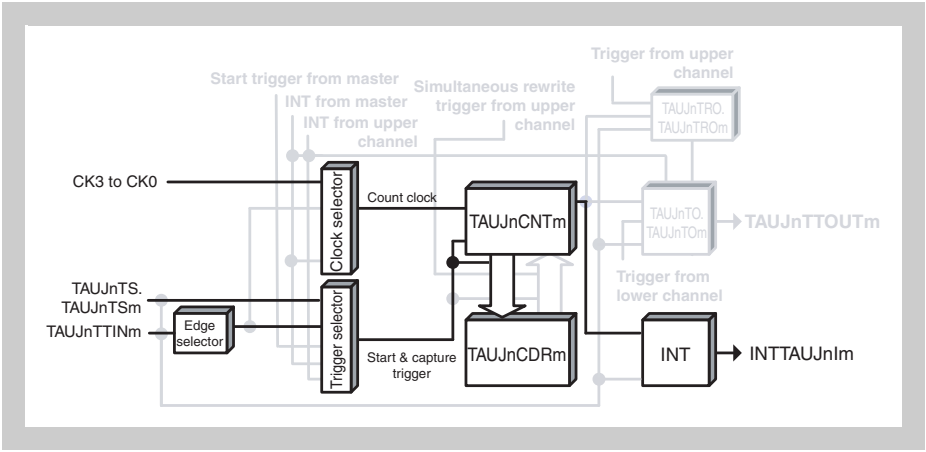


Figure 13-42 Block diagram for TAUJnTTINm Input Period Count Detection Function

The following settings apply to the general timing diagram:

- Rising and falling edge detection = high width measurement (TAUJnCMURm.TAUJnTIS[1:0] = 11_B)

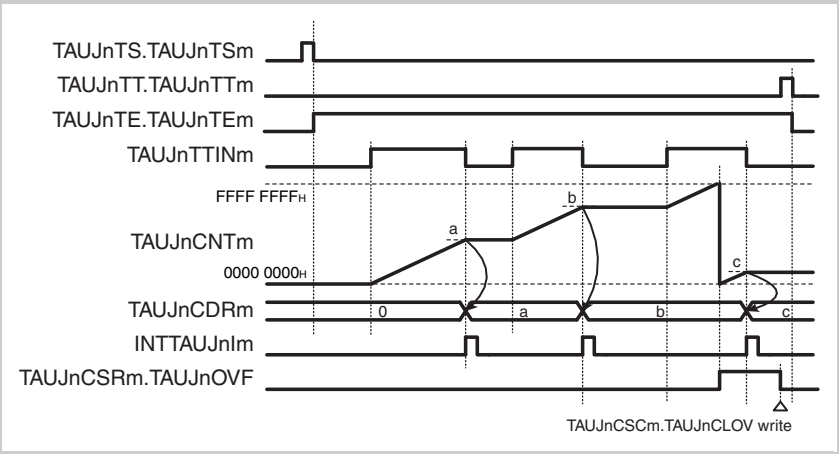


Figure 13-43 General timing diagram for TAUJnTTINm Input Period Count Detection Function

(4) Register settings

(a) TAUJnCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUJn CKS[1:0]		TAUJn CCS[1:0]		TAUJn MAS	TAUJnSTS[2:0]			TAUJn COS[1:0]		-	TAUJnMD[4:1]				TAUJn MD0

Table 13-35 TAUJnCMORm settings for TAUJnTTINm Input Period Count Detection Function

Bit name	Setting
TAUJnCKS[1:0]	00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
TAUJnCCS[1:0]	00: Operation clock is used as the count clock
TAUJnMAS	0: Not used, so set to 0
TAUJnSTS[2:0]	010: Valid edge of the TAUJnTTINm input signal is the external start trigger and the reverse edge is the stop trigger
TAUJnCOS[1:0]	01: Overflow (TAUJnCSRm.TAUJnOVF) set upon counter overflow and cleared by a CPU instruction
TAUJnMD[4:1]	1101: Capture & Gate Count Mode
TAUJnMD0	0: Disables the start trigger during operation

(b) TAUJnCMURm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-														TAUJnTIS[1:0]	

Table 13-36 TAUJnCMURm settings for TAUJnTTINm Input Period Count Detection Function

Bit name	Setting
TAUJnTIS[1:0]	10: Rising and falling edge detection (low width measurement) 11: Rising and falling edge detection (high width measurement)

(c) Channel output mode

Because the channel output mode is not used for this function, clear TAUJnTOE.TAUJnTOEm to 0. However, it can be used in Independent Channel Output Mode Controlled by Software.

(d) Simultaneous rewrite

The simultaneous rewrite registers (TAUJnRDE and TAUJnRDM) cannot be used with the TAUJnTTINm Input Period Count Detection Function. Therefore, these registers must be set to 0.

Table 13-37 Simultaneous rewrite settings for TAUJnTTINm Input Period Count Detection Function

Bit name	Setting
TAUJnRDE.TAUJnRDEm	0: Disables simultaneous rewrite
TAUJnRDM.TAUJnRDMm	0: When simultaneous rewrite is disabled (TAUJnRDE.TAUJnRDEm = 0), set these bits to 0

(5) Operating procedure for TAUJnTTINm Input Period Count Detection Function

Table 13-38 Operating procedure for TAUJnTTINm Input Period Count Detection Function

	Operation	Status of TAUJn
Initial channel setting	Set the TAUJnCMORm register and TAUJnCMURm registers as described in *Table 13-35 "TAUJnCMORm settings for TAUJnTTINm Input Period Count Detection Function" on page 956* and *Table 13-36 "TAUJnCMURm settings for TAUJnTTINm Input Period Count Detection Function" on page 956*. Set the value of the TAUJnCDRm register.	Channel operation is stopped.
Start operation	Set TAUJnTS.TAUJnTSm to 1. TAUJnTS.TAUJnTSm is a trigger bit, so it is automatically cleared to 0. Detection of TAUJnTTINm start edge	TAUJnTE.TAUJnTEm is set to 1 and TAUJnCNTm waits for detection of the TAUJnTTINm start edge. When a start edge is detected, TAUJnCNTm is cleared to 0000 0000_H and TAUJnCNTm starts counting up.
During operation	Detection of TAUJnTTINm edges The TAUJnCDRm, TAUJnCNTm, and TAUJnCSRm registers can be read at any time. The TAUJnCSCm.TAUJnCLOV bit can be set to 1.	When a TAUJnTTINm start edge (rising edge for high width measurement, falling edge for low width measurement) is detected, TAUJnCNTm starts counting up from the stop value. When TAUJnCNTm detects a stop edge (falling edge for high width measurement, rising edge for low width measurement), it transfers the value to TAUJnCDRm and INTTAUJnIm is generated. Counting stops at the "value transferred to TAUJnCDRm + 1" value and TAUJnCNTm waits for detection of the TAUJnTTINm start edge. If the TAUJnCNTm reaches FFFF $FFFF_H$, the counter overflows and TAUJnCSRm.TAUJnOVF is set to 1. Afterwards, this procedure is repeated.
Stop operation	Set TAUJnTT.TAUJnTTm to 1. TAUJnTT.TAUJnTTm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is cleared to 0 and the counter stops. TAUJnCNTm stops and it and TAUJnCSRm.TAUJnOVF retain their current values.

Restart

(6) Specific timing diagrams

(a) Operation stop and restart

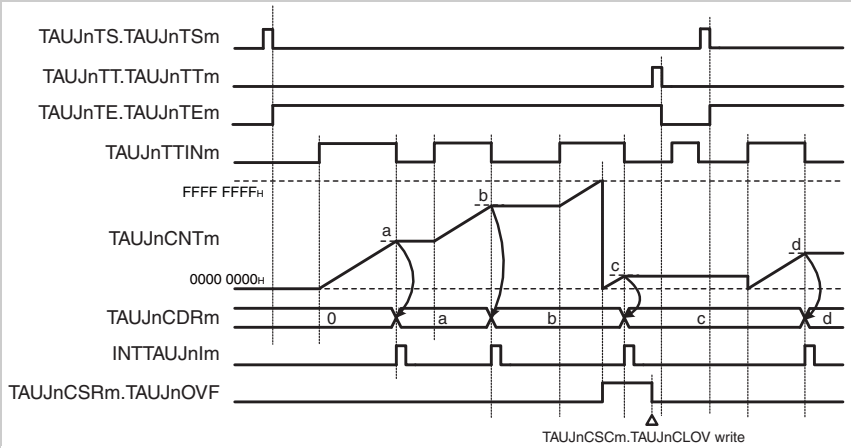


Figure 13-44 Operation stop and restart, TAUJnCMURm.TAUJnTIS[1:0] = 11_B

- The counter can be stopped by setting TAUJnTT.TAUJnTTm to 1, which in turn sets TAUJnTE.TAUJnTEm to 0.
- TAUJnCNTm stops and the current value is retained.
- If the counter is stopped, valid TAUJnTTINm input edges are ignored.
- The counter can be restarted by setting TAUJnTS.TAUJnTSm to 1. TAUJnCNTm restarts counting from 0000 0000_H.

13.15.5 Overflow Interrupt Output Function (During TAUJnTTINm Input Period Count Detection)

(1) Overview

Summary This function measures the cumulative width of a TAUJnTTINm input signal. If the cumulative TAUJnTTINm input width is longer than FFFF $FFFF_H$, an interrupt can be generated to output an overflow interrupt.

Prerequisites
- The operation mode must be set to Gate Count Mode, refer to *Table 13-39 "TAUJnCMORm settings for Overflow Interrupt Output Function (During TAUJnTTINm Input Period Count Detection)" on page 962*.
- TAUJnTTOUTm is not used for this function.
- The value of TAUJnCDRm must be set to FFFF $FFFF_H$.

Description The counter is enabled by setting the channel trigger bit (TAUJnTS.TAUJnTSm) to 1. This in turn sets TAUJnTE.TAUJnTEm = 1, enabling count operation.

The counter starts when a valid TAUJnTTINm input start edge is detected. FFFF $FFFF_H$ is loaded to TAUJnCNTm and the counter starts counting down.

When a valid stop edge is detected, the counter stops and retains the current value. The counter awaits the next TAUJnTTINm input start edge and then continues counting down from the current value.

When the counter reaches 0000 0000_H an interrupt is generated. FFFF $FFFF_H$ is loaded to TAUJnCNTm and the counter continues counting down until a TAUJnTTINm input stop edge is detected.

Conditions The valid start and stop edges are specified by the TAUJnCMURm.TAUJnTIS[1:0] bits:
- If TAUJnCMURm.TAUJnTIS[1:0] = 10_B, the TAUJnTTINm input low period is counted. The start trigger is a falling edge and the stop trigger is a rising edge.
- If TAUJnCMURm.TAUJnTIS[1:0] = 11_B, the TAUJnTTINm input high period is counted. The start trigger is a rising edge and the stop trigger is a falling edge.

Note The counter cannot be restarted during operation.

(2) Block diagram and general timing diagram

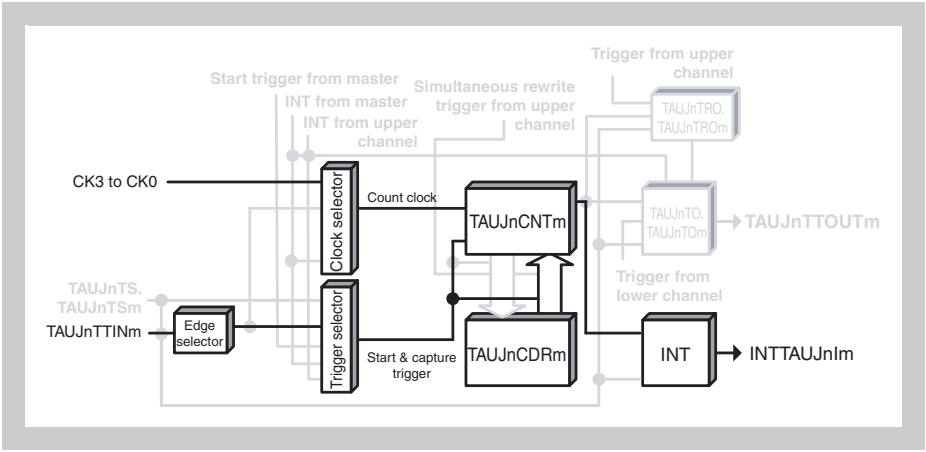


Figure 13-45 Block diagram for Overflow Interrupt Output Function (During TAUJnTTINm Input Period Count Detection)

The following settings apply to the general timing diagram:

- Rising and falling edge detection = high width measurement (TAUJnCMURm.TAUJnTIS[1:0] = 11_B)

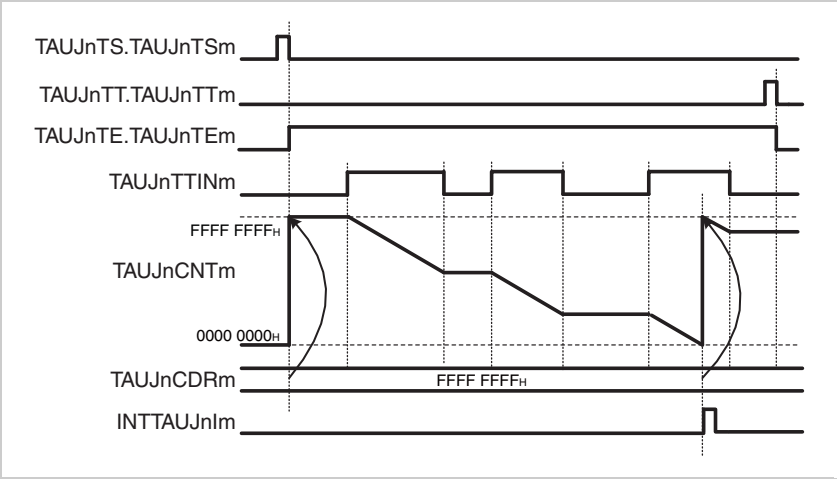


Figure 13-46 General timing diagram for Overflow Interrupt Output Function (During TAUJnTTINm Input Period Count Detection)

(3) Register settings

(a) TAUJnCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUJn CKS[1:0]		TAUJn CCS[1:0]		TAUJn MAS	TAUJnSTS[2:0]			TAUJn COS[1:0]		-	TAUJnMD[4:1]				TAUJn MD0

Table 13-39 TAUJnCMORm settings for Overflow Interrupt Output Function (During TAUJnTTINm Input Period Count Detection)

Bit name	Setting
TAUJnCKS[1:0]	00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
TAUJnCCS[1:0]	00: Operation clock is used as the count clock
TAUJnMAS	0: Not used, so set to 0
TAUJnSTS[2:0]	010: Valid edge of the TAUJnTTINm input signal is the external start trigger and the reverse edge is the stop trigger
TAUJnCOS[1:0]	00: Not used, so set to 00
TAUJnMD[4:1]	1100: Gate Count Mode
TAUJnMD0	0: INTTAUJnIm not generated at operation start

(b) TAUJnCMURm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-														TAUJnTIS[1:0]	

Table 13-40 TAUJnCMURm settings for Overflow Interrupt Output Function (During TAUJnTTINm Input Period Count Detection)

Bit name	Setting
TAUJnTIS[1:0]	10: Rising and falling edge detection (low width measurement) 11: Rising and falling edge detection (high width measurement)

(c) Channel output mode

Because the channel output mode is not used for this function, clear TAUJnTOE.TAUJnTOEm to 0. However, it can be used in Independent Channel Output Mode Controlled by Software.

(d) Simultaneous rewrite

The simultaneous rewrite registers (TAUJnRDE and TAUJnRDM) cannot be used with the Overflow Interrupt Output Function (During TAUJnTTINm Input Period Count Detection). Therefore, these registers must be set to 0.

Table 13-41 Simultaneous rewrite settings for Overflow Interrupt Output Function (During TAUJnTTINm Input Period Count Detection)

Bit name	Setting
TAUJnRDE.TAUJnRDEm	0: Disables simultaneous rewrite
TAUJnRDM.TAUJnRDMm	0: When simultaneous rewrite is disabled (TAUJnRDE.TAUJnRDEm = 0), set these bits to 0

(4) Operating procedure for Overflow Interrupt Output Function (during TAUJnTTINm input period count detection)

Table 13-42 Operating procedure for Overflow Interrupt Output Function (during TAUJnTTINm input period count detection)

	Operation	Status of TAUJn
Initial channel setting	Set the TAUJnCMORm register and TAUJnCMURm registers as described in *Table 13-39 "TAUJnCMORm settings for Overflow Interrupt Output Function (During TAUJnTTINm Input Period Count Detection)" on page 962* and *Table 13-40 "TAUJnCMURm settings for Overflow Interrupt Output Function (During TAUJnTTINm Input Period Count Detection)" on page 962*. Set the value of the TAUJnCDRm register.	Channel operation is stopped.
Start operation	Set TAUJnTS.TAUJnTSm to 1. TAUJnTS.TAUJnTSm is a trigger bit, so it is automatically cleared to 0. Detection of TAUJnTTINm start edge	TAUJnTE.TAUJnTEm is set to 1 and TAUJnCNTm waits for detection of the start edge. When a start edge is detected, TAUJnCNTm loads the TAUJnCDRm value (FFFF FFFF$_H$).
During operation	The TAUJnCNTm register can be read at all times.	TAUJnCNTm counts down. When the counter reaches 0000 0000$_H$: • INTTAUJnIm is generated. • TAUJnCNTm loads the TAUJnCDRm value (FFFF FFFF$_H$) and continues counting down. When a reverse edge of TAUJnTTINm is detected during count operation: • TAUJnCNTm counts down from the stop value. Afterwards, this procedure is repeated.
Stop operation	Set TAUJnTT.TAUJnTTm to 1. TAUJnTT.TAUJnTTm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is cleared to 0 and the counter stops. TAUJnCNTm stops and retains its current value.

Restart

13.16 Other Independent Channel Function

This section describes a function that measures the duration between the function start and a TAUJnTTINm input signal.

13.16.1 TAUJnTTINm Input Position Detection Function

(1) Overview

Summary This function measures the duration between the function start and a TAUJnTTINm input signal.

Prerequisites
- The operation mode must be set to Count Capture Mode, refer to *Table 13-43 "TAUJnCMORm settings for TAUJnTTINm Input Position Detection Function" on page 968*.
- TAUJnTTOUTm is not used for this function.

Description The counter operation is enabled by setting the channel trigger bit (TAUJnTS.TAUJnTSm) to 1. This in turn sets TAUJnTE.TAUJnTEm = 1, enabling count operation. The counter starts counting from 0000 0000_H. When a valid TAUJnTTINm input edge is detected, the current TAUJnCNTm value is loaded to TAUJnCDRm and an interrupt (INTTAUJnIm) is generated. The counter continues counting.

When the counter reaches FFFF $FFFF_H$, the TAUJnCSRm.TAUJnOVF bit is set to 1 and the counter restarts counting from 0000 0000_H.

TAUJnCSRm.TAUJnOVF is cleared by writing 1 to TAUJnCSCm.TAUJnCLOV.

Conditions If the TAUJnCMORm.TAUJnMD0 bit is set to 0, the first interrupt after a start or restart is not generated. For details refer to *13.10 "TAUJnTTOUTm Output and INTTAUJnIm Generation when Counter Starts or Restarts (by TAUJnMD0 Bit)" on page 913*.

(2) Equations

Function duration at a TAUJnTTINm input pulse =

count clock cycle × [(FFFF $FFFF_H$+1 × TAUJnCSRm.TAUJnOVF) + (TAUJnCDRm capture value + 1)]

<R> **Caution** If a capture signal in input when the counter value reaches the upper limit (0FFFF FFFFH), the overflow flag (OVF) that should be set in response to the occurrence of an overflow in the next count clock cycle will not be set. The OVF flag is set normally at all other times.

(3) Block diagram and general timing diagram

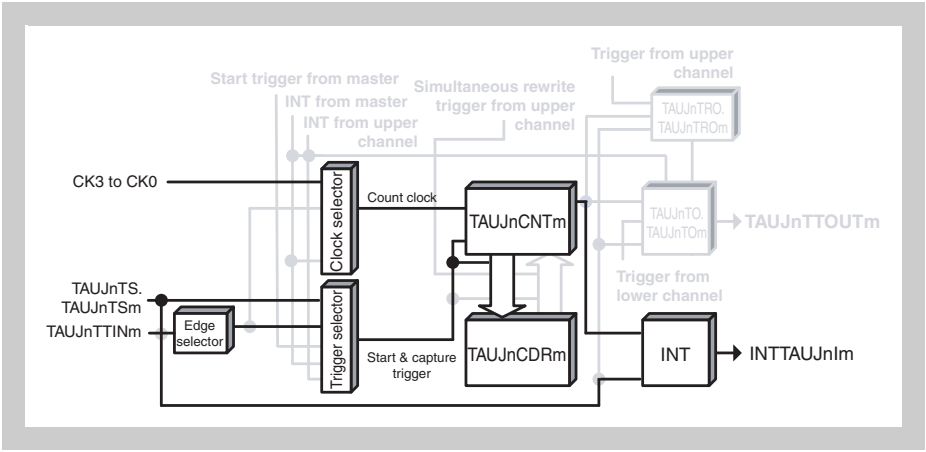


Figure 13-47 Block diagram for TAUJnTTINm Input Position Detection Function

The following settings apply to the general timing diagram:

- INTTAUJnIm not generated at operation start (TAUJnCMORm.TAUJnMD0 = 0)
- Falling edge detection (TAUJnCMURm.TAUJnTIS[1:0] = 00_B)

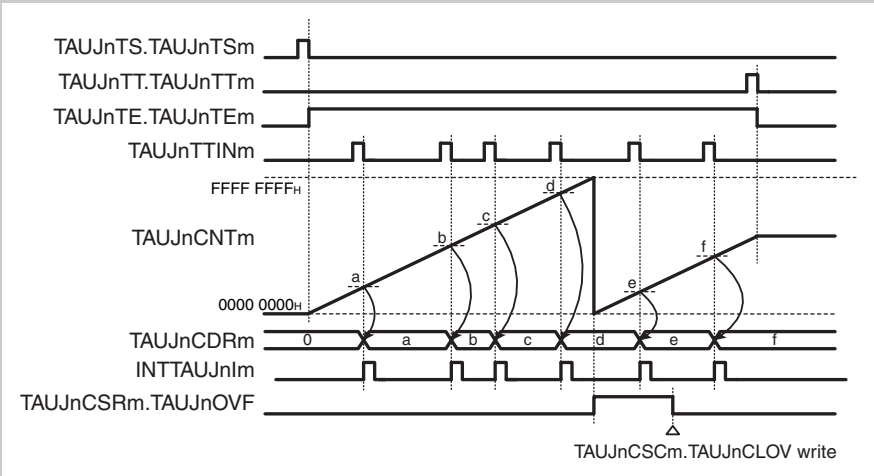


Figure 13-48 General timing diagram for TAUJnTTINm Input Position Detection Function

(4) Register settings

(a) TAUJnCMORm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUJn CKS[1:0]		TAUJn CCS[1:0]		TAUJn MAS	TAUJnSTS[2:0]			TAUJn COS[1:0]		-	TAUJnMD[4:1]				TAUJn MD0

Table 13-43 TAUJnCMORm settings for TAUJnTTINm Input Position Detection Function

Bit name	Setting
TAUJnCKS[1:0]	00: Prescaler output = CK0 01: Prescaler output = CK1 10: Prescaler output = CK2 11: Prescaler output = CK3
TAUJnCCS[1:0]	00: Operation clock is used as the count clock
TAUJnMAS	0: Not used, so set to 0
TAUJnSTS[2:0]	001: Valid TAUJnTTINm input edge signal is used as the external capture trigger
TAUJnCOS[1:0]	01: Overflow (TAUJnCSRm.TAUJnOVF) set upon counter overflow and cleared by a CPU instruction
TAUJnMD[4:1]	1011: Count Capture Mode
TAUJnMD0	0: INTTAUJnIm not generated at operation start 1: Generates INTTAUJnIm at operation start

(b) TAUJnCMURm

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-														TAUJnTIS[1:0]	

Table 13-44 TAUJnCMURm settings for TAUJnTTINm Input Position Detection Function

Bit name	Setting
TAUJnTIS[1:0]	00: Falling edge detection 01: Rising edge detection 10: Rising and falling edge detection (low width measurement) 11: Rising and falling edge detection (high width measurement)

(c) Channel output mode

The channel output mode is not used by this function. However, it can be used in Independent Channel Output Mode Controlled by Software.

(d) Simultaneous rewrite

The simultaneous rewrite registers (TAUJnRDE and TAUJnRDM) cannot be used with the TAUJnTTINm Input Position Detection Function. Therefore, these registers must be set to 0.

Table 13-45 Simultaneous rewrite settings for TAUJnTTINm Input Position Detection Function

Bit name	Setting
TAUJnRDE.TAUJnRDEm	0: Disables simultaneous rewrite
TAUJnRDM.TAUJnRDMm	0: When simultaneous rewrite is disabled (TAUJnRDE.TAUJnRDEm = 0), set these bits to 0

(5) Operating procedure for TAUJnTTINm Input Position Detection Function

Table 13-46 Operating procedure for TAUJnTTINm Input Position Detection Function

	Operation	Status of TAUJn
Initial channel setting	Set the TAUJnCMORm register and TAUJnCMURm registers as described in *Table 13-43 "TAUJnCMORm settings for TAUJnTTINm Input Position Detection Function" on page 968* and *Table 13-44 "TAUJnCMURm settings for TAUJnTTINm Input Position Detection Function" on page 968*. Set the value of the TAUJnCDRm register.	Channel operation is stopped.
Start operation	Set TAUJnTS.TAUJnTSm to 1. TAUJnTS.TAUJnTSm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is set to 1 and the counter starts. INTTAUJnIm is generated when TAUJnCMORm.TAUJnMD0 is set to 1.
During operation	The TAUJnCMURm.TAUJnTIS[1:0] bits can be changed at any time. The TAUJnCDRm and TAUJnCSRm registers can be read at any time. The TAUJnCSC.CLOV bit can be set to 1.	TAUJnCNTm starts counting up from $0000\ 0000_H$. When a TAUJnTTINm valid edge is detected: • TAUJnCNTm transfers (captures) its value to TAUJnCDRm. • INTTAUJnIm is output. • The counter value is not cleared to $0000\ 0000_H$ and TAUJnCNTm continues count operation. Afterwards, this procedure is repeated.
Stop operation	Set TAUJnTT.TAUJnTTm to 1. TAUJnTT.TAUJnTTm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is cleared to 0 and the counter stops. TAUJnCNTm stops and both it and TAUJnCSRm.TAUJnOVF retain their current values.

Restart

(6) Specific timing diagrams

(a) Operation stop and restart

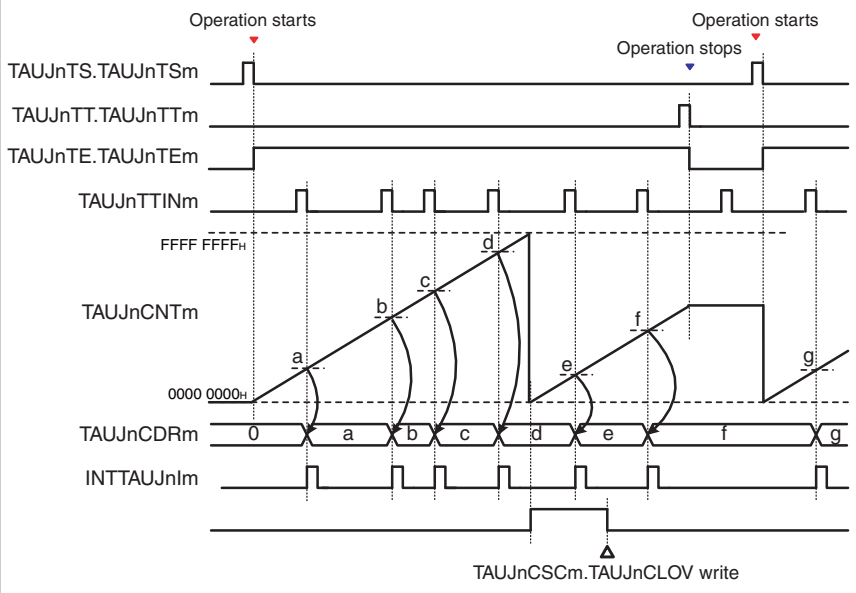


Figure 13-49 Operation stop and restart, TAUJnCMORm.TAUJnMD0 = 0, TAUJnCMURm.TAUJnTIS[1:0] = 00

- The counter can be stopped by setting TAUJnTT.TAUJnTTm to 1, which in turn sets TAUJnTE.TAUJnTEm to 0.
- TAUJnCNTm stops and the current value is retained.
- If the counter is stopped, valid TAUJnTTINm input edges are ignored.
- The counter can be restarted by setting TAUJnTS.TAUJnTSm to 1. TAUJnCNTm restarts counting from 0000 0000_H.

13.17 Synchronous PWM Signal Functions Triggered at Regular Intervals

This section describes a function that generates PWM signals at regular intervals. For a general overview of synchronous channel operation, see *13.3 “Functional Description” on page 894*.

13.17.1 PWM Output Function

(1) Overview

Summary This function generates multiple PWM outputs by using a master and multiple slave channels. It enables the pulse cycle (frequency) and the duty of the TAUJnTTOUTm to be set. The pulse cycle is set in the master channel. The duty is set in the slave channel.

Prerequisites

- Two channels
- The operation mode of the master channel must be set to Interval Timer Mode, refer to *Table 13-47 "TAUJnCMORm settings for the master channel of the PWM Output Function" on page 976*.
- The operation mode of the slave channel(s) must be set to One Count Mode, refer to *Table 13-50 "TAUJnCMORm settings for the slave channel of the PWM Output Function" on page 978*.
- TAUJnTTOUTm is not used for the master channel of this function.
- The channel output mode of the slave channel(s) must be set to Synchronous Channel Output Mode 1 (*13.8 "Channel Output Modes" on page 906*).

Description The counters are started by setting the channel trigger bits (TAUJnTS.TAUJnTSm) to 1. This in turn sets TAUJnTE.TAUJnTEm = 1, enabling count operation. The current value of TAUJnCDRm is loaded to TAUJnCNTm and the counters start counting down from these values. INTTAUJnIm is generated on the master channel, and PWM output is achieved by setting and resetting TAUJnTTOUTm (slave).

- Master channel:

  When the counter of the master channel reaches 0000 0000_H, pulse cycle time has elapsed and INTTAUJnIm is generated. TAUJnCNTm loads the TAUJnCDRm value, and then counts down.

- Slave channel(s)

  The INTTAUJnIm of the master channel triggers the counter operation of the slave channel(s). The current value of TAUJnCDRm (slave) is loaded to TAUJnCNTm (slave) and the counter starts counting down from this value. The TAUJnTTOUTm signal becomes active.

  When the counter reaches 0000 0000_H, i.e., duty time has elapsed, INTTAUJnIm is generated and the TAUJnTTOUTm signal becomes inactive. The counter returns to FFFF $FFFF_H$ and awaits the next INTTAUJnIm of the master channel, and thus the start of the next pulse cycle.

The counter can be stopped by setting TAUJnTT.TAUJnTTm to 1 for the master and slave channel(s), which in turn sets TAUJnTE.TAUJnTEm to 0. TAUJnCNTm and TAUJnTTOUTm of master and slave channel(s) stop but retain their values. The counters can be restarted by setting TAUJnTS.TAUJnTSm to 1.

Conditions Simultaneous rewrite can be used with this function. Refer to *13.7 "Simultaneous Rewrite" on page 901*.

(2) Equations

Pulse cycle = (TAUJnCDRm (master) + 1) x count clock cycle

Duty cycle [%] = (TAUJnCDRm (slave) / (TAUJnCDRm (master) + 1)) x 100

– Duty cycle = 0 %

TAUJnCDRm (slave) = 0000 0000_H

– Duty cycle = 100 %

TAUJnCDRm (slave) ≥ TAUJnCDRm (master) + 1

(3) Block diagram and general timing diagram

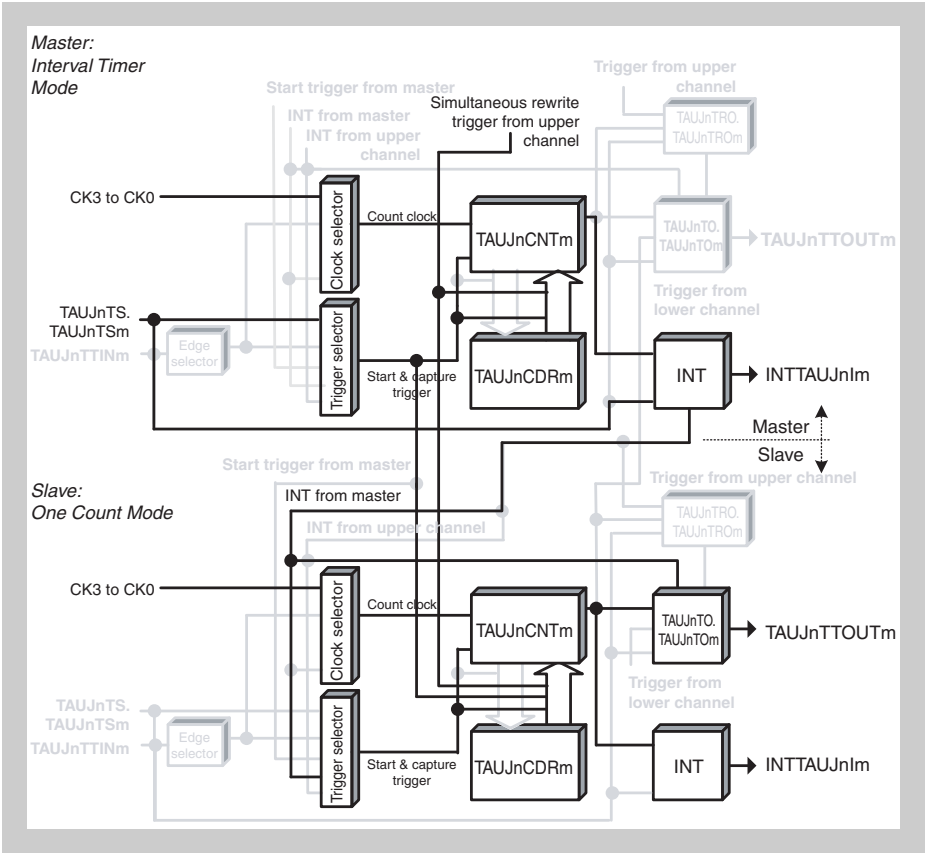


Figure 13-50 Block diagram for PWM Output Function

The following settings apply to the general timing diagram:

- Slave channel: Positive logic (TAUJnTOL.TAUJnTOLm = 0)

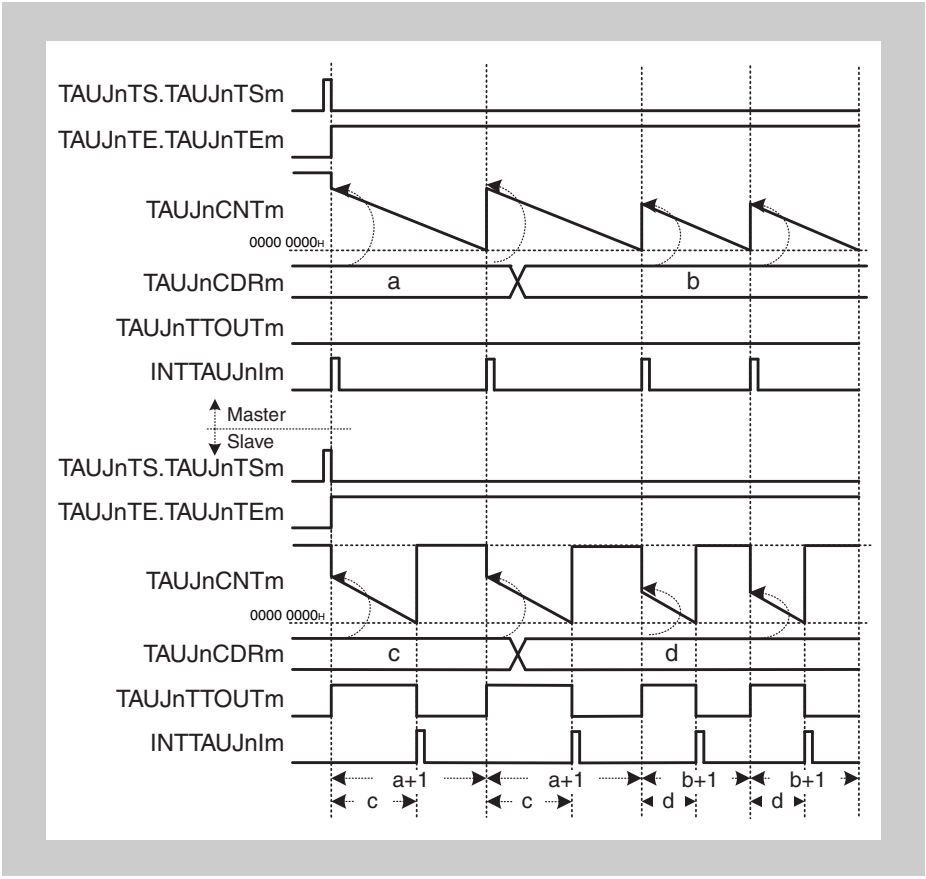


Figure 13-51 General timing diagram for PWM Output Function

Note The interval between the slave channel starting counting and an interrupt being generated is the value of corresponding TAUJnCDRm, whereas for the master channel the interval is the corresponding TAUJnCDRm + 1.

(4) Register settings for the master channel

(a) TAUJnCMORm for the master channel

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUJn CKS[1:0]		TAUJn CCS[1:0]		TAUJn MAS	TAUJnSTS[2:0]			TAUJn COS[1:0]		-	TAUJnMD[4:1]				TAUJn MD0

Table 13-47 TAUJnCMORm settings for the master channel of the PWM Output Function

Bit name	Setting
TAUJnCKS[1:0]	00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 The value of the TAUJnCKS[1:0] bit of the master and slave channel(s) must be identical.
TAUJnCCS[1:0]	00: Operation clock is used as the count clock
TAUJnMAS	1: Channel is master channel
TAUJnSTS[2:0]	000: Counter triggered by software trigger
TAUJnCOS[1:0]	00: Not used, so set to 00
TAUJnMD[4:1]	0000: Interval Timer Mode
TAUJnMD0	1: Generates INTTAUJnIm at operation start

(b) TAUJnCMURm for the master channel

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-														TAUJnTIS[1:0]	

Table 13-48 TAUJnCMURm settings for the master channel of the PWM Output Function

Bit name	Setting
TAUJnTIS[1:0]	00: Not used so set to 00

(c) Channel output mode for the master channel

The channel output mode is not used by this function. However, it can be used by other functions or in Independent Channel Output Mode Controlled by Software.

(d) Simultaneous rewrite for the master channel

The simultaneous rewrite settings of the master and slave channel must be identical.

Table 13-49 Simultaneous rewrite settings for the master channel of the PWM Output Function

Bit name	Setting
TAUJnRDE.TAUJnRDEm	1: Enables simultaneous rewrite
TAUJnRDM.TAUJnRDMm	0: The simultaneous rewrite trigger signal is generated when the master channel starts counting

Note If the TAUJnRDS.TAUJnRDSm bit is 1, there must be a channel higher than the master channel that operates with simultaneous rewrite trigger output function type 1.

Set up the operation as follows:

- Channel setting for simultaneous rewrite trigger output function type 1:
  TAUJnRDCm = 1, TAUJnRDS = 1
  Note that the TAUJnCDR setting for this channel is as follows:
  = ((TAUJnCDR setting of master channel subject to simultaneous rewriting + 1) × number of interrupts) – 1
- Master channel: TAUJnRDCm = 0, TAUJnRDS = 1
- Slave channel: TAUJnRDCm = 0, TAUJnRDS = 1

When the CDRn (slave) setting is greater than the CDRn (master) setting + 1, the duty ratio exceeds 100%, but 100% output is assumed.

(5) Register settings for the slave channel(s)

(a) TAUJnCMORm for the slave channel(s)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUJn CKS[1:0]		TAUJn CCS[1:0]		TAUJn MAS	TAUJnSTS[2:0]			TAUJn COS[1:0]		-	TAUJnMD[4:1]				TAUJn MD0

Table 13-50 TAUJnCMORm settings for the slave channel of the PWM Output Function

Bit name	Setting
TAUJnCKS[1:0]	00: Prescaler output CK0 01: Prescaler output CK1 10: Prescaler output CK2 11: Prescaler output CK3 The value of the TAUJnCKS[1:0] bit of the master and slave channel(s) must be identical.
TAUJnCCS[1:0]	00: Operation clock is used as the count clock
TAUJnMAS	0: Channel is a slave channel
TAUJnSTS[2:0]	100: INTTAUJnIm of the master channel is the start trigger
TAUJnCOS[1:0]	00: Not used, so set to 00
TAUJnMD[4:1]	0100: One Count Mode
TAUJnMD0	1: Enables the start trigger during operation

(b) TAUJnCMURm for the slave channel(s)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
-														TAUJnTIS[1:0]	

Table 13-51 TAUJnCMURm settings for the slave channel of the PWM Output Function

Bit name	Setting
TAUJnTIS[1:0]	00: Not used so set to 00

(c) Channel output mode for the slave channel(s)

Table 13-52 Control bit settings for Independent Channel Output Mode 1

Bit name	Setting
TAUJnTOE.TAUJnTOEm	1: Enable the independent channel output mode.
TAUJnTO.TAUJnTOm	0: Low level 1: High level
TAUJnTOM.TAUJnTOMm	1: Synchronous channel operation
TAUJnTOC.TAUJnTOCm	0: Operation mode 1
TAUJnTOL.TAUJnTOLm	0: Positive logic 1: Inverted logic

(d) Simultaneous rewrite for the slave channel(s)

The simultaneous rewrite settings of the master and slave channel must be identical.

Table 13-53 Simultaneous rewrite settings for the slave channel of the PWM Output Function

Bit name	Setting
TAUJnRDE.TAUJnRDEm	1: Enables simultaneous rewrite
TAUJnRDM.TAUJnRDMm	0: The simultaneous rewrite trigger signal is generated when the master channel starts counting

(6) Operating procedure for PWM Output Function

Table 13-54 Operating procedure for PWM Output Function

	Operation	Status of TAUJn
Initial channel setting	Master channel: set the TAUJnCMORm and TAUJnCMURm registers and the channel output mode as described in *4 "Register settings for the master channel" on page 976*. Slave channel: set the TAUJnCMORm and TAUJnCMURm registers and the channel output mode as described in *5 "Register settings for the slave channel(s)" on page 978*. Set the values of the TAUJnCDRm registers of all channels.	Channel operation is stopped.
Start operation	Set TAUJnTS.TAUJnTSm of the master and slave channels to 1 simultaneously. TAUJnTS.TAUJnTSm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm (master and slave channels) is set to 1 and the counters of the master and slave channels start. INTTAUJnIm is generated on the master channel and TAUJnTTOUTm (slave) is set.
During operation	TAUJnCDRm can be changed at any time. TAUJnCNTm and TAUJnRSF.TAUJnRSFm can be read at any time. TAUJnRDT.TAUJnRDTm can be changed during operation.	TAUJnCNTm of the master channel loads TAUJnCDRm and counts down. When the counter reaches 0000 0000_H: • INTTAUJnIm (master) is generated. • TAUJnCNTm (master) loads the TAUJnCDRm value, and then continues count operation. • TAUJnCNTm (slave) loads the TAUJnCDRm value, and then counts down. • TAUJnTTOUTm (slave) becomes active. When TAUJnCNTm (slave) reaches 0000 0000_H: • INTTAUJnIm (slave) is generated. • TAUJnTTOUTm (slave) becomes inactive.
Stop operation	Set TAUJnTT.TAUJnTTm of the master and slave channels to 1 simultaneously. TAUJnTT.TAUJnTTm is a trigger bit, so it is automatically cleared to 0.	TAUJnTE.TAUJnTEm is cleared to 0 and the counter stops. TAUJnCNTm and TAUJnTTOUTm stop and retain their current values.

Restart

(7) Specific timing diagrams

(a) Duty cycle = 0%

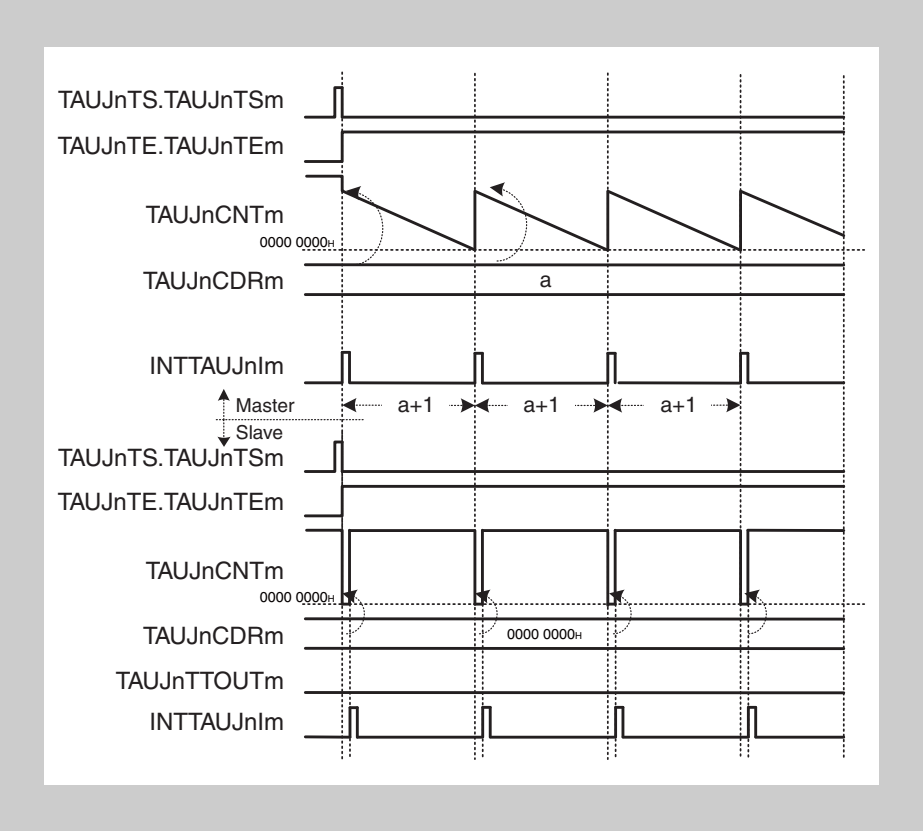


Figure 13-52 TAUJnCDRm (slave) = 0000 0000$_H$, positive logic (TAUJnTOL.TAUJnTOLm (slave) = 0)

- Every time the master channel generates an interrupt (INTTAUJnIm), 0000 0000$_H$ is loaded to TAUJnCNTm (slave). Therefore, TAUJnCNTm (slave) cannot start counting and TAUJnTTOUTm remains at not active state.
- TAUJnCNTm (slave) loads the TAUJnCDRm value and an interrupt is generated.

(b) Duty cycle = 100%

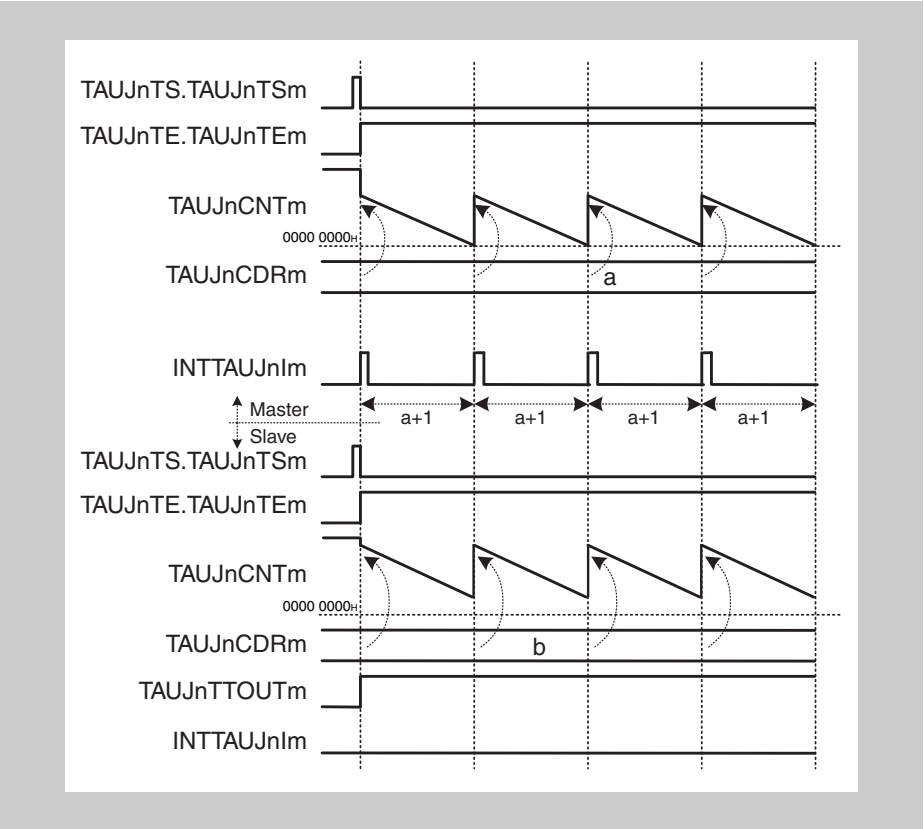


Figure 13-53 TAUJnCDRm (slave) ≥ TAUJnCDRm (master) + 1, positive logic (TAUJnTOL.TAUJnTOLm (slave) = 0)

- If the TAUJnCDRm (slave) value is higher than the TAUJnCDRm (master) value, the counter of the slave channel cannot reach 0000 0000_H and interrupts are therefore not generated. The TAUJnTTOUTm remains at active state.

(c) Stop and restart operation

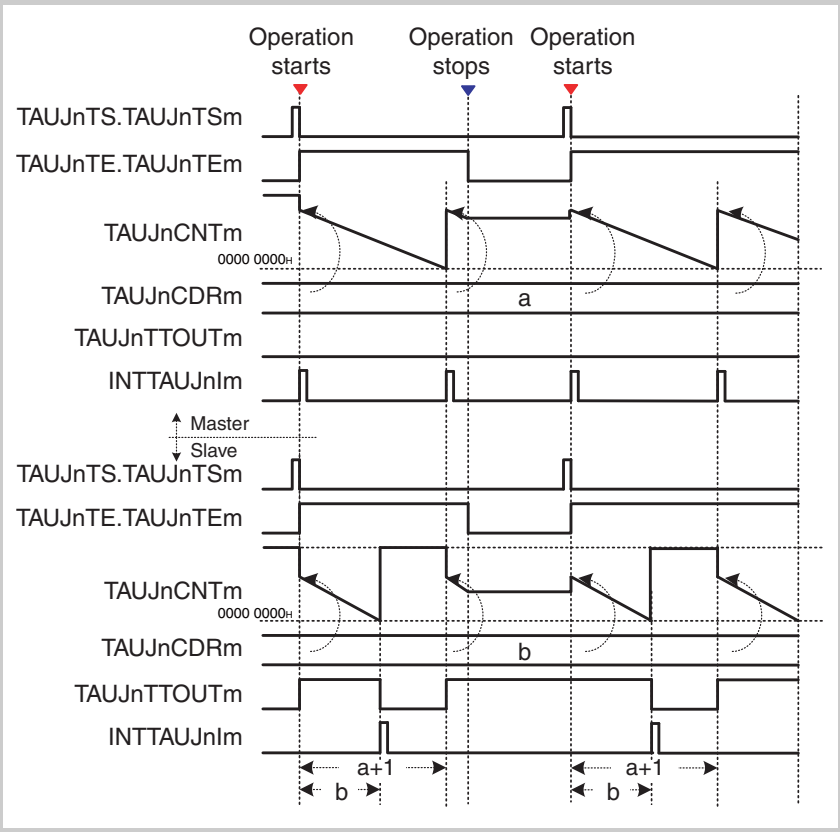


Figure 13-54 Stop and restart operation
Positive logic (TAUJnTOL.TAUJnTOLm (slave) = 0)

- The counter can be stopped by setting TAUJnTT.TAUJnTTm of the master and slave channel(s) to 1, which in turn sets TAUJnTE.TAUJnTEm to 0.
- TAUJnCNTm and TAUJnTTOUTm of all channels stop and the current values are retained. No interrupts are generated.
- The counter can be restarted by setting TAUJnTS.TAUJnTSm of master and slave channel(s) to 1.
  TAUJnCNTm loads the TAUJnCDRm value of master and slave channels, and then starts counting down from this value.

13.18 Registers

This section contains a description of all the registers of the 32-bit TAUJ.

13.18.1 TAUJn registers overview

The TAUJn is controlled and operated by the registers in the following table. Where there is one register per channel, this is indicated by an “m”, where m runs from 0 to 3.

Table 13-55 TAUJn registers overview

Register name	Shortcut	Address
TAUJn prescaler registers		
TAUJn prescaler clock select register	TAUJnTPS	<TAUJn_base> + 90_H
TAUJn prescaler baud rate setting register	TAUJnBRS	<TAUJn_base> + 94_H
TAUJn control registers		
TAUJn channel data register m	TAUJnCDRm	<TAUJn_base> + m x 4_H
TAUJn channel counter register m	TAUJnCNTm	<TAUJn_base> + 10_H + m x 4_H
TAUJn channel mode OS register m	TAUJnCMORm	<TAUJn_base> + 80_H + m x 4_H
TAUJn channel mode user register m	TAUJnCMURm	<TAUJn_base> + 20_H + m x 4_H
TAUJn channel status register m	TAUJnCSRm	<TAUJn_base> + 30_H + m x 4_H
TAUJn channel status clear trigger register m	TAUJnCSCm	<TAUJn_base> + 40_H + m x 4_H
TAUJn channel start trigger register	TAUJnTS	<TAUJn_base> + 54_H
TAUJn channel enable status register	TAUJnTE	<TAUJn_base> + 50_H
TAUJn channel stop trigger register	TAUJnTT	<TAUJn_base> + 58_H
TAUJn output registers		
TAUJn channel output enable register	TAUJnTOE	<TAUJn_base> + 60_H
TAUJn channel output register	TAUJnTO	<TAUJn_base> + $5C_H$
TAUJn channel output mode register	TAUJnTOM	<TAUJn_base> + 98_H
TAUJn channel output configuration register	TAUJnTOC	<TAUJn_base> + $9C_H$
TAUJn channel output active level register	TAUJnTOL	<TAUJn_base> + 64_H
TAUJn reload data registers		
TAUJn channel reload data enable register	TAUJnRDE	<TAUJn_base> + $A0_H$
TAUJn channel reload data mode register	TAUJnRDM	<TAUJn_base> + $A4_H$
TAUJn channel reload data trigger register	TAUJnRDT	<TAUJn_base> + 68_H
TAUJn channel reload status register	TAUJnRSF	<TAUJn_base> + $6C_H$

Note The <TAUJn_base> addresses of the registers are defined in the first section of this chapter under the keyword “Register addresses”.

13.18.2 TAUJn prescaler registers details

(1) TAUJnTPS - TAUJn prescaler clock select register

This register specifies the PCLK prescalers for clocks CK0, CK1, CK2, and CK3_PRE for all channels. CK3 is generated by dividing CK3_PRE by the factor specified in TAUJnBRS.

Access This register can be read/written in 16-bit units.

Address <TAUJn_base> + 90_H

Initial Value $FFFF_H$. This register is initialized by any reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUJnPRS3[3:0]				TAUJnPRS2[3:0]				TAUJnPRS1[3:0]				TAUJnPRS0[3:0]			
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

Table 13-56 TAUJnTPS register contents (1/3)

Bit position	Bit name	Function
15 to 12	TAUJnPRS3[3:0]	Specifies the CK3_PRE clock. Clock CK3_PRE is the input clock of the BRG unit. The BRG unit supplies the CK3 operation clock for all channels. These bits can only be rewritten when all counters using CK3 are stopped (TAUJnTE.TAUJnTEm = 0).

TAUJnPRS3[3:0]	CK3_PRE clock
0000_B	PCLK/2^0
0001_B	PCLK/2^1
0010_B	PCLK/2^2
0011_B	PCLK/2^3
0100_B	PCLK/2^4
0101_B	PCLK/2^5
0110_B	PCLK/2^6
0111_B	PCLK/2^7
1000_B	PCLK/2^8
1001_B	PCLK/2^9
1010_B	PCLK/2^{10}
1011_B	PCLK/2^{11}
1100_B	PCLK/2^{12}
1101_B	PCLK/2^{13}
1110_B	PCLK/2^{14}
1111_B	PCLK/2^{15}

Table 13-56 TAUJnTPS register contents (2/3)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>11 to 8</td><td>TAUJnPRS2[3:0]</td><td>Specifies the CK2 clock.
<table>
<tr><th>PRS2[3:0]</th><th>CK2 clock</th></tr>
<tr><td>0000_B</td><td>PCLK/2^0</td></tr>
<tr><td>0001_B</td><td>PCLK/2^1</td></tr>
<tr><td>0010_B</td><td>PCLK/2^2</td></tr>
<tr><td>0011_B</td><td>PCLK/2^3</td></tr>
<tr><td>0100_B</td><td>PCLK/2^4</td></tr>
<tr><td>0101_B</td><td>PCLK/2^5</td></tr>
<tr><td>0110_B</td><td>PCLK/2^6</td></tr>
<tr><td>0111_B</td><td>PCLK/2^7</td></tr>
<tr><td>1000_B</td><td>PCLK/2^8</td></tr>
<tr><td>1001_B</td><td>PCLK/2^9</td></tr>
<tr><td>1010_B</td><td>PCLK/2^{10}</td></tr>
<tr><td>1011_B</td><td>PCLK/2^{11}</td></tr>
<tr><td>1100_B</td><td>PCLK/2^{12}</td></tr>
<tr><td>1101_B</td><td>PCLK/2^{13}</td></tr>
<tr><td>1110_B</td><td>PCLK/2^{14}</td></tr>
<tr><td>1111_B</td><td>PCLK/2^{15}</td></tr>
</table>
These bits can only be rewritten when all counters using CK2 are stopped (TAUJnTE.TAUJnTEm = 0).</td></tr>
<tr><td>7 to 4</td><td>TAUJnPRS1[3:0]</td><td>Specifies the CK1 clock.
<table>
<tr><th>PRS1[3:0]</th><th>CK1 clock</th></tr>
<tr><td>0000_B</td><td>PCLK/2^0</td></tr>
<tr><td>0001_B</td><td>PCLK/2^1</td></tr>
<tr><td>0010_B</td><td>PCLK/2^2</td></tr>
<tr><td>0011_B</td><td>PCLK/2^3</td></tr>
<tr><td>0100_B</td><td>PCLK/2^4</td></tr>
<tr><td>0101_B</td><td>PCLK/2^5</td></tr>
<tr><td>0110_B</td><td>PCLK/2^6</td></tr>
<tr><td>0111_B</td><td>PCLK/2^7</td></tr>
<tr><td>1000_B</td><td>PCLK/2^8</td></tr>
<tr><td>1001_B</td><td>PCLK/2^9</td></tr>
<tr><td>1010_B</td><td>PCLK/2^{10}</td></tr>
<tr><td>1011_B</td><td>PCLK/2^{11}</td></tr>
<tr><td>1100_B</td><td>PCLK/2^{12}</td></tr>
<tr><td>1101_B</td><td>PCLK/2^{13}</td></tr>
<tr><td>1110_B</td><td>PCLK/2^{14}</td></tr>
<tr><td>1111_B</td><td>PCLK/2^{15}</td></tr>
</table>
These bits can only be rewritten when all counters using CK1 are stopped (TAUJnTE.TAUJnTEm = 0).</td></tr>
</table>

Table 13-56 TAUJnTPS register contents (3/3)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>3 to 0</td><td>TAUJnPRS0[3:0]</td><td>Specifies the CK0 clock.
<table>
<tr><th>PRS0[3:0]</th><th>CK0 clock</th></tr>
<tr><td>0000_B</td><td>PCLK/2^0</td></tr>
<tr><td>0001_B</td><td>PCLK/2^1</td></tr>
<tr><td>0010_B</td><td>PCLK/2^2</td></tr>
<tr><td>0011_B</td><td>PCLK/2^3</td></tr>
<tr><td>0100_B</td><td>PCLK/2^4</td></tr>
<tr><td>0101_B</td><td>PCLK/2^5</td></tr>
<tr><td>0110_B</td><td>PCLK/2^6</td></tr>
<tr><td>0111_B</td><td>PCLK/2^7</td></tr>
<tr><td>1000_B</td><td>PCLK/2^8</td></tr>
<tr><td>1001_B</td><td>PCLK/2^9</td></tr>
<tr><td>1010_B</td><td>PCLK/2^{10}</td></tr>
<tr><td>1011_B</td><td>PCLK/2^{11}</td></tr>
<tr><td>1100_B</td><td>PCLK/2^{12}</td></tr>
<tr><td>1101_B</td><td>PCLK/2^{13}</td></tr>
<tr><td>1110_B</td><td>PCLK/2^{14}</td></tr>
<tr><td>1111_B</td><td>PCLK/2^{15}</td></tr>
</table>
These bits can only be rewritten when all counters using CK0 are stopped (TAUJnTE.TAUJnTEm = 0).</td></tr>
</table>

Note The TAUJn clock input PCLK is specified in the first section of this chapter under the keyword “Clock supply”.

(2) TAUJnBRS - TAUJn prescaler baud rate setting register

This register specifies the division factor of prescaler clock CK3.

CK3 is generated by dividing CK3_PRE by the factor specified in this register plus one. The PCLK prescaler for CK3_PRE is specified in TAUJnTPS.TAUJnPRS3[3:0].

Access This register can be read/written in 8-bit units.

Address <TAUJn_base> + 94_H

Initial Value 00_H. This register is initialized by any reset.

7	6	5	4	3	2	1	0
TAUJBRS[07:00]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 13-57 TAUJnBRS register contents

Bit position	Bit name	Function
7 to 0	TAUJnBRS[07:00]	Specifies the CK3_PRE clock division factor for generating CK3.

TAUJnBRS[07:00]	CK3 clock
$0000\ 0000_B$	CK3_PRE / 1
$0000\ 0001_B$	CK3_PRE / 2
$0000\ 0010_B$	CK3_PRE / 3
$0000\ 0011_B$	CK3_PRE / 4
...	...
$1111\ 1110_B$	CK3_PRE / 255
$1111\ 1111_B$	CK3_PRE / 256

13.18.3 TAUJn control registers details

(1) TAUJnCDRm - TAUJn channel data register

This register functions either as a compare register or as a capture register, depending on the operation mode specified in TAUJnCMORm.TAUJnMD[4:0].

Access This register can be read/written in 32-bit units.

- In capture mode, only reading is possible. Write operation is ignored.
- In compare mode, reading and writing is possible.

Address <TAUJn_base> + m x 4_H

Initial Value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TAUJnCDR[31:16]															
R/W															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUJnCDR[15:0]															
R/W															

Table 13-58 TAUJnCDRm register contents

Bit position	Bit name	Function
31 to 0	TAUJnCDR[31:0]	Data register for the capture/compare value

(2) TAUJnCNTm - TAUJn channel counter register

This register is the channel m counter register.

Access This register can be read in 32-bit units.

Address <TAUJn_base> + 10_H + m x 4_H

Initial Value 0000 0000_H or FFFF $FFFF_H$. The initial value depends on the operation mode, see *Table 13-60 "TAUJnCNTm read values after the counter is re-enabled" on page 991*

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TAUJnCNT[31:16]															
R															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUJnCNT[15:0]															
R															

Table 13-59 TAUJnCNTm register contents

Bit position	Bit name	Function
31 to 0	TAUJnCNT[31:0]	32-bit counter value

The read value depends on the counter, the operation mode change, and the values of the TAUJnTS.TAUJnTSm and TAUJnTT.TAUJnTTm bits.

The *initial* counter read value depends on the operation mode and how the counter was stopped:

- by a reset
- by a counter stop trigger (TAUJnTT.TAUJnTTm = 1)

The following table lists the initial counter read values after the counter has stopped (TAUJnTE.TAUJnTEm= 0) and re-enabled (TAUJnTS.TAUJnTSm = 1).

The table also contains the counter read value one count after the counter is enabled (TAUJnTS.TAUJnTSm = 1) for modes where the counter waits for a start trigger.

Table 13-60 TAUJnCNTm read values after the counter is re-enabled

Mode name	Count method (up/down)	TAUJnCNTm value		
		When operation mode is changed after reset	After stop trigger	After one count
Interval Timer mode	Count down	FFFF FFFF$_H$	Stop value	-
Capture mode	Count up	0000 0000$_H$	Stop value	-
One Count mode	Count down	FFFF FFFF$_H$	Stop value	FFFF FFFF$_H$
Capture & One Count mode	Count up	0000 0000$_H$	Stop value	Captured value + 1 (TAUJnCDRm)
Count Capture Mode	Count up	0000 0000$_H$	Stop value	-
Gate Count Mode	Count down	FFFF FFFF$_H$	Stop value	Stop value
Capture & Gate Count Mode	Count up	0000 0000$_H$	Stop value	Stop value

Note If the operation mode is changed while the counter is stopped, the initial counter value after counter restart is undefined. The operation mode is changed by the TAUJnCMORm.TAUJnMD[4:1] bits.

(3) TAUJnCMORm - TAUJn channel mode OS register

This register controls channel m operation.

Access This register can be read or written in 16-bit units. Writing is only possible while the counter is stopped (TAUJnTE.TAUJnTEm = 0).

Address <TAUJn_base> + 80_H + m x 4_H

Initial Value 0000_H. This register is initialized by any reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAUJn CKS[1:0]		TAUJn CCS[1:0]		TAUJn MAS	TAUJnSTS[2:0]			TAUJn COS[1:0]		-	TAUJnMD[4:0]				
R/W		R/W		R/W	R/W			R/W		R	R/W				

Table 13-61 TAUJnCMORm register contents (1/3)

Bit position	Bit name	Function
15,14	TAUJn CKS[1:0]	Selects the prescaler output. The prescaler output is used for the TAUJnTTINm input edge detection circuit. It can also be used as the count clock depending on bits TAUJnCMORm.CCS[1:0]. TAUJn CKS1 / TAUJn CKS0 / Selected prescaler output 0 / 0 / CK0 0 / 1 / CK1 1 / 0 / CK2 1 / 1 / CK3
13,12	TAUJn CCS[1:0]	Selects the count clock for TAUJnCNTm counter. TAUJn CCS1 / TAUJn CCS0 / Selected count clock 0 / 0 / Prescaler output specified by TAUJnCMORm.TAUJnCKS[1:0] 0 / 1 / Valid edge of TAUJnTTINm input signal 1 / 0 / Setting prohibited 1 / 1
11	TAUJnMAS	Specifies the channel as master or slave channel during synchronous channel operation. 0: Slave 1: Master This bit is only valid for even channels (CHm_even). For odd channels (CHm_odd), it is fixed to 0.

Table 13-61 TAUJnCMORm register contents (2/3)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>10 to 8</td><td>TAUJn STS[2:0]</td><td>Selects the external start trigger.
<table>
<tr><th>TAUJn STS2</th><th>TAUJn STS1</th><th>TAUJn STS0</th><th>Description</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>Software trigger</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>Valid edge of the TAUJnTTINm input signal. TAUJnCMURm.TAUJnTIS[1:0] specifies the valid edge.</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>Valid edge of the TAUJnTTINm input signal is used as the start trigger, and the reverse edge is used as the stop trigger.</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>Setting prohibited</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>INT of the master channel</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td rowspan="3">Setting prohibited</td></tr>
<tr><td>1</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>1</td><td>1</td></tr>
</table></td></tr>
<tr><td>7, 6</td><td>TAUJn COS[1:0]</td><td>Specifies when the capture register TAUJnCDRm and the overflow flag TAUJnCSRm.TAUJnOVF of channel m are updated.
These bits are only valid if channel m is in capture mode.
<table>
<tr><th>TAUJn COS1</th><th>TAUJn COS0</th><th>TAUJnCDRm</th><th>TAUJnCSRm.TAUJnOVF</th></tr>
<tr><td>0</td><td>0</td><td rowspan="2">Updated upon detection of a TAUJnTTINm input valid edge.</td><td>Updated (cleared or set) upon detection of a TAUJnTTINm input valid edge:
• If a counter overflow has occurred since the last valid edge detection, TAUJnCSRm.TAUJnOVF is set.
• If no counter overflow has occurred since the last valid edge detection, TAUJnCSRm.TAUJnOVF is cleared.</td></tr>
<tr><td>0</td><td>1</td><td>Set upon counter overflow and cleared by setting TAUJnCSCm.TAUJnCLOV.</td></tr>
<tr><td>1</td><td>0</td><td rowspan="2">Updated upon detection of a TAUJnTTINm input valid edge and upon counter overflow:
• TAUJnTTINm input valid edge: Counter value is written to TAUJnCDRm.
• Overflow: FFFF FFFF$_H$ is loaded to TAUJnCDRm.
The next TAUJnTTINm input valid edge detection is ignored.</td><td>Not set.</td></tr>
<tr><td>1</td><td>1</td><td>Set upon counter overflow and cleared by setting TAUJnCSCm.TAUJnCLOV.</td></tr>
</table></td></tr>
</table>

Table 13-61 TAUJnCMORm register contents (3/3)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>4 to 0</td><td>TAUJn MD[4:0]</td><td>Specifies the operation mode.
<table>
<tr><th>TAUJn MD4</th><th>TAUJn MD3</th><th>TAUJnMD2</th><th>TAUJnMD1</th><th>TAUJnMD0</th><th>Description</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1/0</td><td>Interval Timer mode</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1/0</td><td>Setting prohibited</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>1/0</td><td>Capture mode</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>1/0</td><td>Setting prohibited</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>1/0</td><td>One Count mode</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>1/0</td><td>Setting prohibited</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>Capture & One Count mode</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td rowspan="4">1/0</td><td rowspan="4">Setting prohibited</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td><td>1/0</td><td>Count Capture mode</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>Gate Count mode</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>Capture & Gate Count mode</td></tr>
</table>
<table>
<tr><th>Mode</th><th>Role of the MD0 bit</th></tr>
<tr><td>Interval Timer mode
Capture mode
Count Capture mode</td><td>Specifies whether the INTTAUJnIm signal is output when the counter starts counting (when the start trigger is input).
0: Does not output INTTAUJnIm.
1: Outputs INTTAUJnIm</td></tr>
<tr><td>One Count mode
Gate Count mode</td><td>Enables/disables start trigger detection during counting:
0: Disables
1: Enables</td></tr>
<tr><td>Capture & One Count mode
Capture & Gate Count mode</td><td>This bit must be set to 0.</td></tr>
</table>
</td></tr>
</table>

(4) TAUJnCMURm - TAUJn channel mode user register

This register specifies the type of valid edge detection used for the TAUJnTTINm input.

Access This register can be read/written in 8-bit units.

Address <TAUJn_base> + 20_H + m x 4_H

Initial Value 00_H. This bit is initialized by any reset.

7	6	5	4	3	2	1	0
-	-	-	-	-	-	TAUJnTIS[1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 13-62 TAUJnCMURm register contents

Bit position	Bit name	Function
1, 0	TAUJnTIS[1:0]	Specifies the valid edge of the TAUJnTTINm input. • Edge detection for TAUJnTTINm input signals is performed based on the prescaler output selected by TAUJnCMORm.TAUJnCKS[1:0].

TAUJn TIS1	TAUJn TIS0	Description
0	0	Falling edge
0	1	Rising edge
1	0	Rising and falling edges (low-width measurement selection). Start trigger: falling edge Stop trigger (capture): rising edge
1	1	Rising and falling edges (high-width measurement selection). Start trigger: rising edge Stop trigger (capture): falling edge

(5) TAUJnCSRm - TAUJn channel status register

This register indicates the overflow status of channel m.

Access This register can be read in 8-bit units.

Address <TAUJn_base> + 30_H + m x 4_H

Initial Value 00_H. This bit is initialized by any reset.

7	6	5	4	3	2	1	0
-	-	-	-	-	-	-	TAUJnOVF
R	R	R	R	R	R	R	R

Table 13-63 TAUJnCSRm register contents

Bit position	Bit name	Function
0	TAUJnOVF	Indicates the counter overflow status: 0: No overflow occurred 1: Overflow occurred This bit is only used in the following modes: • Capture mode • Capture & One Count mode • Count Capture mode • Capture & Gate Count mode The function of this bit depends on the setting of control bits TAUJnCMORm.TAUJnCOS[1:0].

(6) TAUJnCSCm - TAUJn channel status clear register

This register is a trigger register for clearing the overflow flag TAUJnCSRm.TAUJnOVF of a channel m.

Access This register can be written in 8-bit units. It is always read as 00_H.

Address <TAUJn_base> + 40_H + m x 4_H

Initial Value 00_H. This bit is initialized by any reset.

7	6	5	4	3	2	1	0
-	-	-	-	-	-	0	TAUJnCLOV
R	R	R	R	R	R	R	W

Table 13-64 TAUJnCSCm register contents

Bit position	Bit name	Function
0	TAUJnCLOV	0: No function 1: Clears the overflow flag TAUJnCSRm.TAUJnOVF

(7) TAUJnTS - TAUJn channel start trigger register

This register enables the counter for each channel.

Access This register can be written in 8-bit units. It is always read as 00_H.

Address <TAUJn_base> + 54_H

Initial Value 00_H. This bit is initialized by any reset.

7	6	5	4	3	2	1	0
-	-	-	-	TAUJnTS03	TAUJnTS02	TAUJnTS01	TAUJnTS00
W	W	W	W	W	W	W	W

Table 13-65 TAUJnTS register contents

Bit position	Bit name	Function
3 to 0	TAUJnTSm	Enables the counter for channel m. 0: No function 1: Enables the counter and sets TAUJnTE.TAUJnTEm = 1. TAUJnTE.TAUJnTEm = 1 only *enables* counter. Whether the counter *starts* depends on the selected operation mode.

(8) TAUJnTE - TAUJn channel enable status register

This register indicates whether counter is enabled or disabled.

Access This register can be read in 8-bit units.

Address <TAUJn_base> + 50_H

Initial Value 00_H. This bit is initialized by any reset.

7	6	5	4	3	2	1	0
-	-	-	-	TAUJnTE03	TAUJnTE02	TAUJnTE01	TAUJnTE00
R	R	R	R	R	R	R	R

Table 13-66 TAUJnTE register contents

Bit position	Bit name	Function
3 to 0	TAUJnTEm	Indicates whether counter for channel m is enabled or disabled. 0: Counter disabled 1: Counter enabled This bit is set when TAUJnTSSTm (the synchronous channel start trigger signal) trigger input is detected or when TAUJnTS.TAUJnTSm is set. Setting TAUJnTT.TAUJnTTm to 1 resets this bit to 0.

(9) TAUJnTT - TAUJn channel stop trigger register

This register stops the counter for each channel.

Access This register can be written in 8-bit units. It is always read as 00_H.

Address <TAUJn_base> + 58_H

Initial Value 00_H.

7	6	5	4	3	2	1	0
-	-	-	-	TAUJnTT03	TAUJnTT02	TAUJnTT01	TAUJnTT00
W	W	W	W	W	W	W	W

Table 13-67 TAUJnTT register contents

Bit position	Bit name	Function
3 to 0	TAUJnTTm	Stops the counter of channel m. 0: No function 1: Stop the counter and reset TAUJnTE.TAUJnTEm. TAUJnCNTm, TAUJnTO.TAUJnTOm, and TAUJnTTOUTm retain their values from before the counter stopped.

13.18.4 TAUJn output registers details

(1) TAUJnTOE - TAUJn channel output enable register

This register enables and disables Independent Channel Output Mode Controlled by Software.

Access This register can be read/written in 8-bit units.

Address <TAUJn_base> + 60_H

Initial Value 00_H. This bit is initialized by any reset.

7	6	5	4	3	2	1	0
-	-	-	-	TAUJnTOE03	TAUJnTOE02	TAUJnTOE01	TAUJnTOE00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 13-68 TAUJnTOE register contents

Bit position	Bit name	Function
3 to 0	TAUJnTOEm	Enables or disables the independent timer output function. 0: Disables the independent timer output function. 1: Enables the independent timer output function.

(2) TAUJnTOM - TAUJn channel output mode register

This register specifies the output mode of each channel.

Access This register can be read/written in 8-bit units. Writing is only possible while the counter is stopped (TAUJnTE.TAUJnTEm = 0).

Address <TAUJn_base> + 98_H

Initial Value 00_H. This bit is initialized by any reset.

7	6	5	4	3	2	1	0
-	-	-	-	TAUJnTOM03	TAUJnTOM02	TAUJnTOM01	TAUJnTOM00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 13-69 TAUJnTOM register contents

Bit position	Bit name	Function
3 to 0	TAUJnTOMm	Specifies the channel output mode. 0: Independent channel output mode 1: Synchronous channel output mode The output mode depends on several channel output control bits, as can be seen in *Table 13-9 “Channel output modes” on page 907*.

(3) TAUJnTOC - TAUJn channel output configuration register

This register specifies the output mode of each channel in combination with TAUJnTOMm.

Access This register can be read/written in 8-bit units. Writing is only possible while the counter is stopped (TAUJnTE.TAUJnTEm = 0).

Address <TAUJn_base> + $9C_H$

Initial Value 00_H. This bit is initialized by any reset.

7	6	5	4	3	2	1	0
-	-	-	-	TAUJnTOC03	TAUJnTOC02	TAUJnTOC01	TAUJnTOC00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 13-70 TAUJnTOC register contents

Bit position	Bit name	Function
3 to 0	TAUJnTOCm	Specifies the output mode. 0: Operation mode 1 (toggle mode) 1: No function

13.18.5 TAUJn channel output level registers details

(1) TAUJnTO - TAUJn channel output register

This register specifies and reads the level of TAUJnTTOUTm.

Access This register can be read/written in 8-bit units.

Address <TAUJn_base> + $5C_H$

Initial Value 00_H. This bit is initialized by any reset.

7	6	5	4	3	2	1	0
-	-	-	-	TAUJnTO03	TAUJnTO02	TAUJnTO01	TAUJnTO00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 13-71 TAUJnTO register contents

Bit position	Bit name	Function
3 to 0	TAUJnTOm	Specifies and reads the level of TAUJnTTOUTm. 0: Low level 1: High level Only TAUJnTOm bits for which the independent channel output function is disabled (TAUJnTOEm = 0) can be written.

(2) TAUJnTOL - TAUJn channel output level register

This register specifies the output logic of the channel output bit (TAUJnTO.TAUJnTOm).

Access This register can be read/written in 8-bit units.

Address <TAUJn_base> + 64_H

Initial Value 00_H. This bit is initialized by any reset.

7	6	5	4	3	2	1	0
-	-	-	-	TAUJnTOL03	TAUJnTOL02	TAUJnTOL01	TAUJnTOL00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 13-72 TAUJnTOL register contents

Bit position	Bit name	Function
3 to 0	TAUJnTOLm	Specifies the output logic of the channel m output bit (TAUJnTO.TAUJnTOm). 0: Positive logic (active high) 1: Inverted logic (active low)

13.18.6 TAUJn simultaneous rewrite register details

(1) TAUJnRDE - TAUJn channel reload data enable register

This register enables and disables simultaneous rewrite of the data register TAUJnCDRm.

Access This register can be read/written in 8-bit units. Writing is only possible while TAUJnTE.TAUJnTEm is 0.

Address <TAUJn_base> + $A0_H$

Initial Value 00_H. This bit is initialized by any reset.

7	6	5	4	3	2	1	0
-	-	-	-	TAUJnRDE03	TAUJnRDE02	TAUJnRDE01	TAUJnRDE00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 13-73 TAUJnRDE register contents

Bit position	Bit name	Function
3 to 0	TAUJnRDEm	Enables and disables simultaneous rewrite of the data register of channel m. 0: Disables simultaneous rewrite 1: Enabled simultaneous rewrite

(2) TAUJnRDM - TAUJn channel reload data mode register

This register selects the timing for generating a simultaneous rewrite control signal.

Access This register can be read/written in 8-bit units. Writing is only possible while TAUJnTE.TAUJnTEm is 0.

Address <TAUJn_base> + $A4_H$

Initial Value 00_H. This bit is initialized by any reset.

7	6	5	4	3	2	1	0
-	-	-	-	TAUJnRDM03	TAUJnRDM02	TAUJnRDM01	TAUJnRDM00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 13-74 TAUJnRDM register contents

Bit position	Bit name	Function
3 to 0	TAUJnRDMm	Selects the timing for generating a simultaneous rewrite control signal. 0: When the master channel counter starts counting 1: No function These bits only apply when TAUJnRDE.TAUJnRDEm = 1.

(3) TAUJnRDT - TAUJn channel reload data trigger register

This register triggers the simultaneous rewrite pending state.

Access This register can be written in 8-bit units. It is always read as 00_H.

Address <TAUJn_base> + 68_H

Initial Value 00_H. This bit is initialized by any reset.

7	6	5	4	3	2	1	0
-	-	-	-	TAUJnRDT03	TAUJnRDT02	TAUJnRDT01	TAUJnRDT00
W	W	W	W	W	W	W	W

Table 13-75 TAUJnRDT register contents

Bit position	Bit name	Function
3 to 0	TAUJnRDTm	Triggers the simultaneous rewrite pending state. 0: No function 1: Simultaneous rewrite pending state is triggered. The simultaneous rewrite pending flag (TAUJnRSFm) is set to 1. The system waits for the simultaneous rewrite trigger.

(4) TAUJnRSF - TAUJn channel reload status register

This flag register indicates the simultaneous rewriting status.

Access This register can be read in 8-bit units.

Address <TAUJn_base> + $6C_H$

Initial Value 00_H. This bit is initialized by any reset.

7	6	5	4	3	2	1	0
-	-	-	-	TAUJnRSF03	TAUJnRSF02	TAUJnRSF01	TAUJnRSF00
R	R	R	R	R	R	R	R

Table 13-76 TAUJnRSF register contents

Bit position	Bit name	Function
3 to 0	TAUJnRSFm	Indicates the simultaneous rewrite status. 0: Simultaneous rewriting was performed due to the generation of a simultaneous rewrite trigger. 1: Simultaneous rewriting is pending (TAUJnRDTm = 1).

Chapter 14 Encoder Timer (ENCA)

This chapter contains a generic description of the Encoder Timer (ENCA).

The first section describes the properties specific to the V850E2/MN4, such as instances, register base addresses, and input/output signal names.
The subsequent sections describe the common features.

14.1 V850E2/MN4 ENCA Features

Instances

This microcontroller has the following number of instances of the Encoder Timer.

Table 14-1 Instances of ENCA

Encoder Timer	
Instance	2
Name	ENCA0, ENCA1

Instances index n

Throughout this chapter, the individual instances of a Encoder Timer is identified by the index "n" (n = 0, 1), for example, ENCAnCTL for the ENCAn control register.

Register addresses

All ENCAn register addresses are given as address offsets from the individual base address <ENCAn_base_OS> or <ENCAn_base_USER>.
The base addresses <ENCAn_base_OS> and <ENCAn_base_USER> of each ENCAn are listed in the following table.

Table 14-2 Register base addresses <ENCAn_base>

ENCAn instance	Base address	Address
ENCA0	ENCAn_base_OS	FF81 9000_H
	ENCAn_base_USER	FFFF $D800_H$
ENCA1	ENCAn_base_OS	FF81 $A000_H$
	ENCAn_base_USER	FFFF $D900_H$

Clock supply

All Encoder Timers provide one clock input.

Table 14-3 ENCAn clock supply

ENCAn instance	ENCAn clock	Connected to
ENCA0	PCLK	f_{PCLK}
ENCA1	PCLK	f_{PCLK}

Interrupts and DMA/DTS ENCAn can generate the following interrupts and DMA/DTS requests:

Table 14-4 ENCAn interrupts and DMA/DTS requests

ENCAn signals	Function	Connected to
ENCA0:		
ENCATIOV	ENCA0 overflow interrupt	• Interrupt Controller INTENCA0IOV • DMA controller trigger 88 • DTS controller trigger 88
ENCATIUD	ENCA0 underflow interrupt	• Interrupt Controller INTENCA0IUD • DMA controller trigger 89 • DTS controller trigger 89
ENCATINT0	ENCA0 compare match 0 or capture 0 interrupt	• Interrupt Controller INTENCA0I0 • DMA controller trigger 90 • DTS controller trigger 90
ENCATINT1	ENCA0 compare match 1or capture 1interrupt	• Interrupt Controller INTENCA0I1 • DMA controller trigger 91 • DTS controller trigger 91 • ADCA trigger (ADCA0TTIN004, ADCA0TTIN104, ADCA0TTIN204)
ENCATIEC	ENCA0 clear interrupt by encoder input (phase Z)	• Interrupt Controller INTENCA0IEC • DMA controller trigger 92 • DTS controller trigger 92
ENCA1:		
ENCATIOV	ENCA1 overflow interrupt	• Interrupt Controller INTENCA1IOV • DMA controller trigger 93 • DTS controller trigger 93
ENCATIUD	ENCA1 underflow interrupt	• Interrupt Controller INTENCA1IUD • DMA controller trigger 94 • DTS controller trigger 94
ENCATINT0	ENCA1 compare match 0 or capture 0 interrupt	• Interrupt Controller INTENCA1I0 • DMA controller trigger 95 • DTS controller trigger 95
ENCATINT1	ENCA1 compare match 1or capture 1interrupt	• Interrupt Controller INTENCA1I1 • DMA controller trigger 96 • DTS controller trigger 96 • ADCA trigger (ADCA0TTIN005, ADCA0TTIN105, ADCA0TTIN205)
ENCATIEC	ENCA1 clear interrupt by encoder input (phase Z)	• Interrupt Controller INTENCA1IEC • DMA controller trigger 97 • DTS controller trigger 97

Internal signals The internal signals of the Encoder Timers are listed in the following table.

Table 14-5 ENCAn interrupt requests

ENCAn signals	Function	Connected to
ENCA0:		
ENCATSST	ENCA0 simultanous start trigger input	PIC
ENCATSTT	ENCA0 simultanous stop trigger input	Unused
ENCA1:		
ENCATSST	ENCA1 simultanous start trigger input	PIC
ENCATSTT	ENCA1 simultanous stop trigger input	Unused

I/O signals The I/O signals of the Encoder Timers are listed in the following table.

Table 14-6 ENCAn I/O signals

ENCAn signal	Function	Connected to
ENCA0:		
ENCATTIN0	ENCA0 capture trigger input 0	Port TE0_TI0
ENCATTIN1	ENCA0 capture trigger input 1	Port TE0_TI1
ENCATAIN	ENCA0 encoder input (phase A)	Port TE0_AI
ENCATBIN	ENCA0 encoder input (phase B)	Port TE0_BI
ENCATZIN	ENCA0 encoder input (phase Z)	Port TE0_ZI
ENCATCSO	ENCA0 count status output	Reserved
ENCA1:		
ENCATTIN0	ENCA1 capture trigger input 0	Port TE1_TI0
ENCATTIN1	ENCA1 capture trigger input 1	Port TE1_TI1
ENCATAIN	ENCA1 encoder input (phase A)	Port TE1_AI
ENCATBIN	ENCA1 encoder input (phase B)	Port TE1_BI
ENCATZIN	ENCA1 encoder input (phase Z)	Port TE1_ZI
ENCATCSO	ENCA1 count status output	Reserved

14.2 Functional Overview

Features summary

- Generation of the counter control signal from the encoder input signal, and performs count operation in synchronization with PCLK.
- Capture function for capturing the counter value with an external trigger signal
- Compare function for compare match judgment with the counter value
- Two capture compare registers that can be set separately for capture operation and for compare operation
- Interrupt mask function for masking the interrupt signal output as a result of compare match judgment during compare operation
- Function for loading the value of the capture compare register to the counter upon underflow occurrence
- Encoder input signal can be applied to the timer counter clear condition
- Edge or level for clearing the encoder input signal of the timer counter clear condition can be selected
- Detection of counter overflow and underflow and output error flags and occurrence interrupts
- Five interrupts: 2 capture compare interrupts, 1 counter clear interrupt, 1 overflow interrupt, and 1 underflow interrupt.

14.2.1 Block Diagram

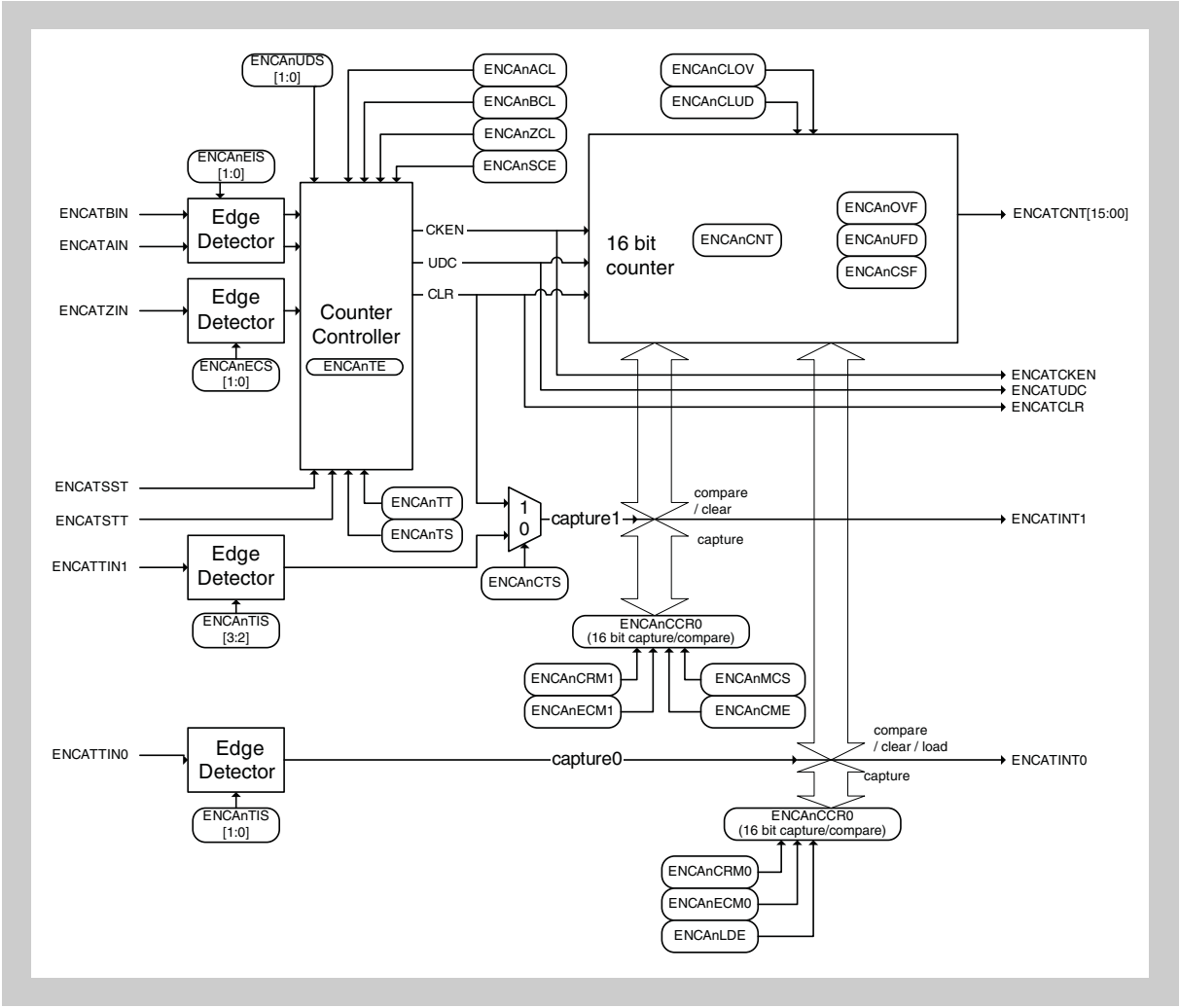


Figure 14-1 Encoder Timer block diagram

14.2.2 Preliminary knowledge for understanding basic specifications

As this macro has an encoder counter function, the rotary encoder is explained below.

Rotary Encoder

The rotary encoder outputs a pulse according to the amount of displacement of the axis of rotation. The output pulse is output as a waveform that has two phase differences, from the fact that the light passing through two slits (phase A, phase B) is transmitted or blocked, when the disc, on which an optical pattern is written, rotates, along with the displacement of the axis of rotation.

The rotation can be detected by counting the output pulses with a counter.

Moreover, there is a pulse called phase Z, which indicates the origin within 1 rotation of the encoder.

Errors can be corrected by clearing the counter to 0000_H by using the phase Z pulse.

Position measurement and speed measurement applications can be achieved by using the rotation measured with the timer counter.

14.3 ENCA Control Registers

The ENCAn is controlled and operated by means of the following registers:

Table 14-7 ENCAn register overview

Register name	Shortcut	Address
ENCA capture compare register 0	ENCAnCCR0	<ENCAn_base_USER>
ENCA capture compare register 1	ENCAnCCR1	<ENCAn_base_USER> + 04_H
ENCA counter register	ENCAnCNT	<ENCAn_base_USER> + 08_H
ENCA status flag register	ENCAnFLG	<ENCAn_base_USER> + $0C_H$
ENCA status flag clear register	ENCAnFGC	<ENCAn_base_USER> + 10_H
ENCA timer enable status register	ENCAnTE	<ENCAn_base_USER> + 14_H
ENCA timer start trigger register	ENCAnTS	<ENCAn_base_USER> + 18_H
ENCA timer stop trigger register	ENCAnTT	<ENCAn_base_USER_USER> + $1C_H$
ENCA I/O control register 0	ENCAnIOC0	<ENCAn_base_USER> + 20_H
ENCA control register	ENCAnCTL	<ENCAn_base_OS> + 40_H
ENCA I/O control register 1	ENCAnIOC1	<ENCAn_base_OS> + 44_H

Base address The base addresses <ENCAn_base_USER> and <ENCAn_base_OS> of the ENCAn is defined in the first section of this chapter under the key word “Register addresses”.

(1) ENCAnCTL – ENCA control register

This register is used to configure various operation settings of the Encoder Timer.

Access This register can be read/written in 16-bit units.
Writing to this register during operation is prohibited.

Address <ENCAn_base_OS> + 40_H

Initial value Reset input initializes this register to 0000_H.

15	14	13	12	11	10	9	8
ENCAn CME	ENCAn MCS	0	0	0	0	ENCAn CRM1	ENCAn CRM0
R/W	R/W	R	R	R	R	R/W	R/W

7	6	5	4	3	2	1	0
ENCAn CTS	0	0	ENCAn LDE	ENCAn ECM1	ENCAn ECM0	ENCAn UDS[1:0]	
R/W	R	R	R/W	R/W	R/W	R/W	R/W

Table 14-8 ENCAnCTL register contents (1/2)

Bit	Name	Function
15	ENCAnCME	Encoder clear mask enable bit This bit is used to enable/disable masking of compare match interrupt detection when the compare function is used. 0: Disables the compare match interrupt (ENCATINT1) mask function for the ENCAnCCR1 register 1: Enables the compare match interrupt (ENCATINT1) mask function for the ENCAnCCR1 register. This bit is valid only when ENCAnCRM1 = 0. When this bit is set to “1”, setting ENCAnECM1 to “1” is prohibited.
14	ENCAnMCS	Encoder mask clear select bit This bit is used to select the trigger for cancelling masking of compare match interrupt detection when the compare function is used. This bit is valid only when ENCAnCRM1 = 0. 0 Masking of compare match interrupt detection is cancelled when the ENCAnCCR1 register is written. 1: Masking of compare match interrupt detection is cancelled when one of the following three operations is performed. -Timer counter clear operation accompanying phase Z -Timer counter clear operation upon compare match between ENCAnCNT and ENCAnCCR0 when ENCAnECM0 = 1 -Loading from ENCAnCCR0 to timer counter upon underflow detection when ENCAnLDE = 1
9	ENCAnCRM1	ENCACCR1 register mode bit 0: ENCAnCCR1 used as compare register. 1: ENCAnCCR1 used as capture register.
8	ENCAnCRM0	ENCACCR0 register mode bit 0: ENCAnCCR0 used as compare register. 1: ENCAnCCR0 used as capture register.

Table 14-8 ENCAnCTL register contents (2/2)

Bit	Name	Function
7	ENCAnCTS	ENCAnCCR1 capture trigger select bit This is a trigger selection bit register for the capture operation to the ENCAnCCR1 register. This bit selects the capture trigger to the ENCAnCCR1 register. This bit is valid only when ENCAnCRM1 = 1. 0: Uses ENCATTIN1 of capture trigger 1 signal as the trigger for capturing to the ENCAnCCR1 register. 1: Uses ENCATZIN of the phase Z encoder input signal as the trigger for capturing to the ENCAnCCR1 register.
4	ENCAnLDE	ENCAn counter load enable bit This register is used to enable/disable setting value loading to the counter upon underflow occurrence. This bit is valid only when ENCAnCRM0 = 0. When ENCAnCRM0 = 1, loading of the ENCAnCCR0 register setting value to the counter upon occurrence of an underflow is not performed, regardless of the value of this bit. 0: Disable loading of ENCAnCCR0 register setting value to counter upon occurrence of a counter underflow. 1: Enable loading of ENCAnCCR0 register setting value to counter upon occurrence of a counter underflow.
3	ENCAnECM1	Encoder clear mode bit 1 This register is used to set the counter clear operation upon match between the counter value and ENCAnCCR1 setting value. This bit is valid only when ENCAnCRM1 = 0. 0: Does not clear the counter to 0000_H upon match of timer counter value and ENCAnCCR1 setting value. 1: Clears the counter to 0000_H upon match of timer counter value and ENCAnCCR1 setting value if the next count is a up-count.
2	ENCAnECM0	Encoder clear mode bit 0 This register is used to set the counter clear operation upon match between the counter value and ENCAnCCR0 setting value. This bit is valid only when ENCAnCRM0 = 0. 0: Does not clear the counter to 0000_H upon match of timer counter value and ENCAnCCR0 setting value. 1: Clears the counter to 0000_H upon match of timer counter value and ENCAnCCR0 setting value if the next count is a down-count.
1, 0	ENCAn UDS[1:0]	UPDOWN count selection bits 1 and 0 This is the counter up/down control register using ENCATAIN and ENCATBIN. 00: Upon detection of valid edge of ENCATAIN, - down-count when ENCATBIN = H, - up-count when ENCATBIN = L 01: Upon detection of valid edge of ENCATAIN, up-count, Upon detection of valid edge of ENCATBIN, down-count 10: At rising edge of ENCATAIN, down-count At falling edge of ENCATAIN, up-count However, count operation performed only when ENCATBIN = L. 11: Detection of both edges of ENCATAIN, ENCATBIN. Judgment of count operation combining both detected edge and level.

(2) ENCAnIOC0 – ENCA I/O control register 0

This register is used to select the input edge of capture triggers 0 and 1 (ENCAnTTIN0, ENCAnTTIN1).

Access This register can be read/written in 8-bit units.
Writing to this register during operation is prohibited.

Address <ENCAn_base_USER> + 20_H

Initial value Reset input initializes this register to 00_H.

7	6	5	4	3	2	1	0
0	0	0	0	ENCAnTIS[3:2]		ENCAnTIS[1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 14-9 ENCAnIOC0 register contents

Bit	Name	Function
3, 2	ENCAn TIS[3:2]	Input edge selection bits for capture trigger 1. These bits are valid only when the ENCAnCTL register's ENCAnCRM1 = 1 and ENCAnCTS = 0. All other settings of ENCAnCRM1 and ENCAnCTS are invalid. 00: No edge detection 01: Rising edge detection 10: Falling edge detection 11: Both edges detection
1, 0	ENCAn TIS[1:0]	Input edge selection bits for capture trigger 0. These bits are valid only when ENCAnCTL.ENCAnCRM0 = 1 00: No edge detection 01: Rising edge detection 10: Falling edge detection 11: Both edges detection

(3) ENCAnIOC1 – ENCA I/O control register 1

This register is used to perform the clear condition setting and edge selection upon encoder input.

Access This register can be read/written in 8-bit units.
Writing to this register during operation is prohibited.

Address <ENCAn_base_OS> + 44_H

Initial value Reset input initializes this register to 00_H.

7	6	5	4	3	2	1	0
ENCAn SCE	ENCAn ZCL	ENCAn BCL	ENCAn ACL	ENCAn ECS[1:0]		ENCAn EIS[1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 14-10 ENCAnIOC1 register contents (1/2)

Bit	Name	Function
7	ENCAnSCE	Encoder special-clear enable bit This is an encoder special clear enable bit. When cleared to 0, the counter is cleared upon phase Z edge detection. When set to 1, the counter is cleared using the level of ENCATZIN, ENCATAIN, and ENCATBIN (special clear). When setting this bit to 1, set ENCAnUDS1 and ENCAnUDS0 to {1, 0} or {1, 1}. The operation is not guaranteed if this bit is set to 1 with ENCAnUDS1 and ENCAnUDS0 set to {0, 0} or {0, 1}. 0: Clears the counter upon detection of ENCATZIN valid edge (set with ENCAnECS1 and ENCAnECS0). 1: Clears the counter upon detection of input level condition of ENCATZIN, ENCATBIN and ENCATAIN (set with ENCAnZCL bit, ENCAnBCL bit, and ENCAnACL bit).
6	ENCAnZCL	Input-Z clear condition selection bit This bit is used to set the condition for clearing the phase Z encoder input (ENCAnZIN) when using the encoder special clear function. This bit is valid only when ENCAnSCE = 1; it is invalid when ENCAnSCE = 0. 0: Clear condition: Low level 1: Clear condition: High level
5	ENCAnBCL	Input-B clear condition selection bit This bit is used to set the condition for clearing the phase B encoder input (ENCAnBIN) when using the encoder special clear function. This bit is valid only when ENCAnSCE = 1; it is invalid when ENCAnSCE = 0. 0: Clear condition: Low level 1: Clear condition: High level

Table 14-10 ENCAnIOC1 register contents (2/2)

Bit	Name	Function
4	ENCAnACL	Input-A clear condition selection bit This bit is used to set the condition for clearing the phase A encoder input (ENCAnAIN) when using the encoder special clear function. This bit is valid only when ENCAnSCE = 1; it is invalid when ENCAnSCE = 0. 0: Clear condition: Low level 1: Clear condition: High level
3, 2	ENCAn ECS[1:0]	Encoder clear input edge selection bits 1 and 0 These are the encoder clear input edge selection bits (phase Z). These bits are valid only when ENCAnSCE = 0; they are invalid when ENCAnSCE = 1. 00: No edge detection 01: Rising edge detection 10: Falling edge detection 11: Both edges detection
1, 0	ENCAn EIS[1:0]	Encoder input selection bits 1 and 0 These are the encoder input edge selection bits (phase A, phase B). These bits are valid when ENCAnUDS1 and ENCAnUDS0 = {0, 0} or {0, 1}, and are invalid when ENCAnUDS1 and ENCAnUDS0 = {1, 0} or {1, 1}. 00: No edge detection 01: Rising edge detection 10: Falling edge detection 11: Both edges detection

(4) ENCAnFLG – ENCA status flag register

This register is holds the status flags of the timer counter of the ENCAn.

Access This register can be read in 8-bit units.
Writing to this register during operation is prohibited.

Address <ENCAn_base_USER> + $0C_H$

Initial value Reset input initializes this register to 00_H.

7	6	5	4	3	2	1	0
					ENCAnCSF	ENCAnUDF	ENCAnOVF
R	R	R	R	R	R	R	R

Table 14-11 ENCAnFLG register contents

Bit	Name	Function
2	ENCAnCSF	Counter status flag This bit reflects the current timer counter operation. This bit is cleared at the start of count operation. 0: Timer counter in up-count status 1: Timer counter in down-count status
1	ENCAnUDF	Underflow flag This bit reflects the occurrence of an underflow during the timer counter operation. This bit is cleared at the start of count operation. 0: This flag is cleared upon any of the following events: - “1” is written to ENCAnFGC.ENCAnCLUD - ENCAnTS is set while ENCAnTE = 0 - “1” is written to XX.yy of PIC[a] 1: This flag is set to “1” upon occurrence of an underflow during the encoder timer count operation.
0	ENCAnOVF	Overflow flag This bit reflects the occurrence of an overflow during the timer counter operation. This bit is cleared at the start of count operation. 0: This flag is cleared upon any of the following events: - “1” is written to ENCAnFGC.ENCAnCLOV - ENCAnTS is set while ENCAnTE = 0 - “1” is written to XX.yy of PIC[a] 1: This flag is set to “1” upon occurrence of an overflow during the encoder timer count operation.

a) For details, see the chapter that describes PIC.

(5) ENCAnFGC – ENCA status flag clear register

This register is used to clear the timer counter status flags of ENCAnFLG.

Access This register can be written in 8-bit units.
This register always returns 0 when read.

Address <ENCAn_base_USER> + 10_H

Initial value Reset input initializes this register to 00_H.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	ENCAnCLUD	ENCAnCLOV
W	W	W	W	W	W	W	W

Table 14-12 ENCAnFGC register contents

Bit	Name	Function
1	ENCAnCLUD	Underflow flag clear This bit clears the underflow flag. 0: Writing is ignored. 1: Clears ENCAnUDF of the ENCAnFL register (clears underflow detection).
0	ENCAnCLOV	Overflow flag clear This bit clears the overflow flag. 0: Writing is ignored. 1: Clears ENCAnOVF of the ENCAnFL register (clears overflow detection).

(6) ENCAnCCR0 – ENCA capture compare register 0

This register is a 16-bit capture compare register 0.

Access This register can be written in 16-bit units when used as a compare register (ENCAnCTL.ENCAnCRM0 = 0).
When used as a capture register (ENCAnCTL.ENCAnCRM0 = 1), writing to this register during operation is invalid.

Address <ENCAn_base_USER>

Initial value Reset input initializes this register to 0000_H.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENCAnCCR0[15:0]															
R/W															

Table 14-13 ENCAnCCR0 register contents

Bit	Name	Function
15 to 0	ENCAn CCR0[15:0]	Capture compare register 0 Upon occurrence of an underflow, the setting value of this register may be loaded to the counter according to the ENCAnCTL.ENCAnLDE setting. See the description of the ENCAnLDE bit in ENCA control register ENCAnCTL for details. • if ENCAnCTL.ENCAnCRM0 = 0: ENCAnCCR0 is compare register Set the value to be compared with the timer counter value. • if ENCAnCTL.ENCAnCRM0 = 1: ENCAnCCR0 is capture register The captured timer counter value is stored.

(7) ENCAnCCR1 – ENCA capture compare register 1

This register is a 16-bit capture compare register 1.

Access This register can be written in 16-bit units when used as a compare register (ENCAnCTL.ENCAnCRM1 = 0).
When used as a capture register (ENCAnCTL.ENCAnCRM1 = 1), writing to this register during operation is invalid.

Address <ENCAn_base_USER> + 04_H

Initial value Reset input initializes this register to 0000_H.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENCAnCCR1[15:0]															
R/W															

Table 14-14 ENCAnCCR1 register contents

Bit	Name	Function
15 to 0	ENCAn CCR1[15:0]	Capture compare register 1 During capture operation, the trigger for capturing to this register differs according to the ENCAnCTL.ENCAnCTS setting. See the description of the ENCAnCTS bit in ENCA control register ENCAnCTL for details. • if ENCAnCTL.ENCAnCRM1 = 0: ENCAnCCR1 is compare register Set the value to be compared with the timer counter value. • if ENCAnCTL.ENCAnCRM1 = 1: ENCAnCCR1 is capture register The captured timer counter value is stored.

(8) ENCAnCNT – ENCA counter register

This register is the 16-bit timer counter register.

Access This register can be read/written in 16-bit units.
This register can be written only when the operation is stopped.

Address <ENCAn_base_USER> + 08_H

Initial value Reset input initializes this register to 0000_H.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENCAnCNT[15:0]															
R/W															

Table 14-15 ENCAnCNT register contents

Bit	Name	Function
15 to 0	ENCAn CNT[15:0]	Counter register • ENCAnTE.ENCAnTE status: 0 (initial setting): Count stop An arbitrary value can be set to timer counter. • ENCAnTE.ENCAnTE status: 0 → 1 (operation start)): Count operation start Up/down count operation is started with the set arbitrary value. • ENCAnTE.ENCAnTE status: 1 (operating): Counting Up/down count operation is performed. • ENCAnTE.ENCAnTE status: 1 → 0 (stopped): Count stop The counter value immediately before the operation was stopped is held, and the count operation is stopped.

(9) ENCAnTE – ENCA timer enable status register

This register indicates the operating status of the ENCAn.

Access This register can be read in 8-bit units.

Address <ENCAn_base_USER> + 14_H

Initial value Reset input initializes this register to 00_H.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	ENCAnTE
R	R	R	R	R	R	R	R

Table 14-16 ENCAnTE register contents

Bit	Name	Function
0	ENCAnTE	Timer status enable bit This is a status bit that indicates the operation enabled/stopped status of the ENCAn. This bit is cleared to “0” when “1” is written to ENCAnTT.ENCAnTT, or when xxxx.yy of PIC is set. This bit is set to “1” when “1” is written to ENCAnTS.ENCAnTS, or when xxxx.zz of PIC is set. 0: Operation stopped status 1: Operation enabled status

(10) ENCAnTS – ENCA timer start trigger register

This register provides the trigger bit for setting the ENCAn to the operation enabled state.

Access This register can be written in 8-bit units.
This register always returns 0 when read. This register can be written only when ENCAnTE.ENCAnTE is 0.

Address <ENCAn_base_USER> + 18_H

Initial value Reset input initializes this register to 00_H.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	ENCAnTS
R	R	R	R	R	R	R	W

Table 14-17 ENCAnTS register contents

Bit	Name	Function
0	ENCAnTS	Timer start trigger bit This is the trigger bit that sets the ENCAn to the operation enabled state. 0: Writing is ignored. 1: ENCAn is set to the operation enabled state by setting ENCAnTE.ENCAnTE = 1.

(11) ENCAnTT – ENCA timer stop trigger register

This register provides the trigger bit for setting the ENCAn to the operation stopped state.

Access This register can be written in 8-bit units.
This register always returns 0 when read.

Address <ENCAn_base_USER> + $1C_H$

Initial value Reset input initializes this register to 00_H.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	ENCAnTT
R	R	R	R	R	R	R	W

Table 14-18 ENCAnTT register contents

Bit	Name	Function
0	ENCAnTT	Timer stop trigger bit This is the trigger bit that sets the ENCAn to the operation stopped state. 0: Writing is ignored. 1: Clears ENCAnTE.ENCAnTE to “0”, to set the ENCAn to the count operation stopped state.

14.4 Functional Description

The ENCAn operates the timer counter with counter up/down control and clear control by encoder inputs (phase A, phase B, phase Z). The ENCAnCCR0 and ENCAnCCR1 registers can be used as dedicated compare registers or as dedicated capture registers.

14.4.1 Timer counter operation

The timer counter operations of the ENCAn are described below.

The figure below below shows the operation phases. See the corresponding section with the section number for detailed descriptions on each operation.

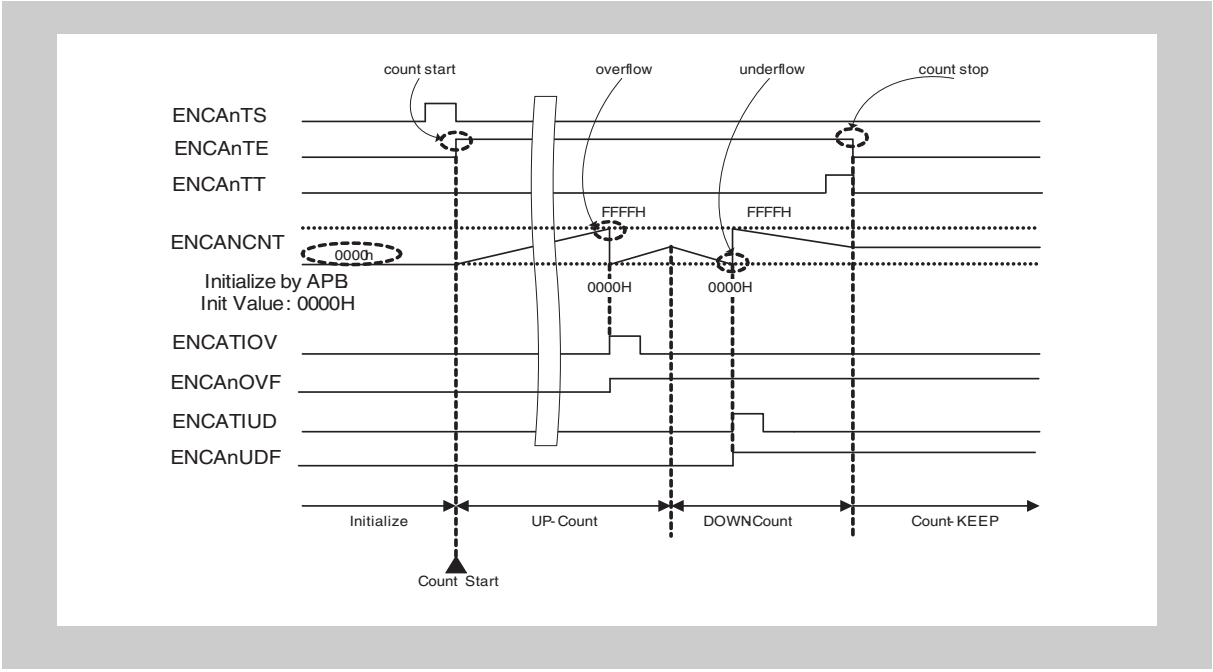


Figure 14-2 Timer counter initial value setting/start/stop

(1) Timer counter initial value setting

The initial value of the ENCAn counter register (ENCAnCNT) can be set in the counter operation stopped status (ENCAnTE = 0).

(2) Timer counter startup

By writing “1” to the timer start trigger bit (ENCAnTS), the timer status enable bit (ENCAnTE) is set to “1”, the count operation is enabled, and counting operation is performed upon detection of the valid edge of the encoder input.

(3) Overflow operation

An overflow occurs when up-counting is performed when the counter value is $FFFF_H$. If the counter value changes from $FFFF_H$ to 0000_H, an overflow interrupt (ENCATIOV) is generated, and the overflow flag (ENCAnOVF) is set to “1”. The overflow flag (ENCAnOVF) is cleared to “0” when “1” is set to the overflow flag clear bit (ENCAnCLOV).

(4) Underflow operation

An underflow occurs when down-counting is performed when the counter value is 0000_H. If the counter value changes from 0000_H to $FFFF_H$, an underflow interrupt (ENCATIUD) is generated, and the underflow flag (ENCAnUDF) is set to “1”. The underflow flag (ENCAnUDF) is cleared to “0” when “1” is set to the underflow flag clear bit (ENCAnCLUD).

(5) Timer counter stop

By writing “1” to the timer stop trigger bit (ENCAnTT), the timer status enable bit (ENCAnTE) is cleared to “0”, and the count operation is stopped. At this time, the timer counter is not reset to 0000_H and holds the value before count operation stop.

14.4.2 Up/down control of timer counter

Up/down control is performed by judging the phase of the encoder inputs (ENCTAIN, ENCTBIN) according to the settings of ENCAnUDS1 and ENCAnUDS0.

(1) When ENCAnUDS1 and ENCAnUDS0 = {0, 0}

ENCAnUDS1	ENCAnUDS0	Operation Description		
		ENCATAIN input	ENCTBIN input	Count operation
0	0	Rising edge	High level	Down
		Falling edge		
		Both edges		
		Rising edge	Low level	Up
		Falling edge		
		Both edges		

The valid edge for ENCATAIN is specified by setting ENCAnEIS1 and ENCAnEIS0.

Count operation is performed when the valid edges of ENCATAIN and ENCATBIN overlap.

The following timing chart shows the count operation when ENCAnUDS1 and ENCAnUDS0 = {0, 0}.

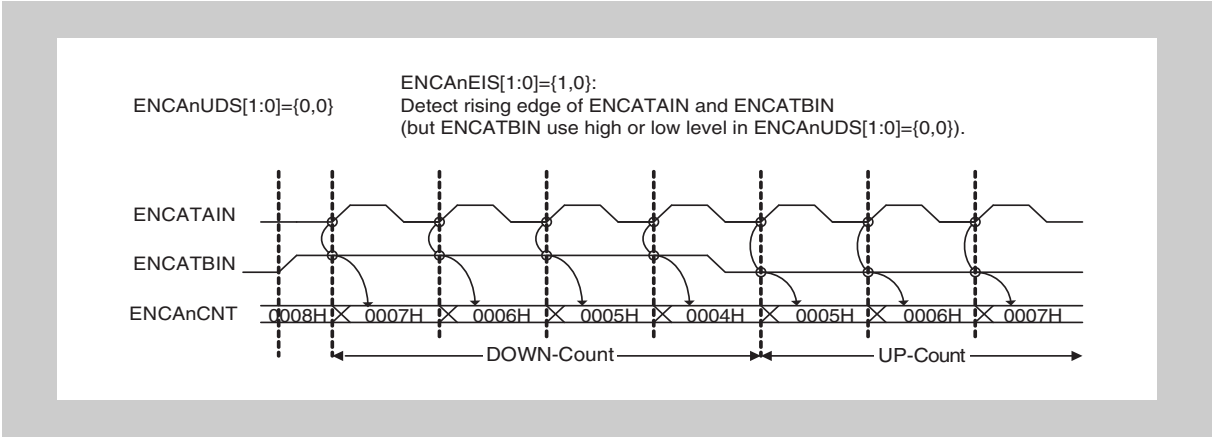


Figure 14-3 Count operation when ENCAnUDS1 and ENCAnUDS0 = {0, 0}

(2) When ENCAnUDS1 and ENCAnUDS0 = {0, 1}

ENCAnUDS1	ENCAnUDS0	Operation description		
		ENCATAIN input	ENCTBIN input	Count operation
0	1	Low level	Rising edge	Down
			Falling edge	
			Both edges	
		High level	Rising edge	
			Falling edge	
			Both edges	
		Rising edge	Low level	Up
		Falling edge		
		Both edges		
		Rising edge	High level	
		Falling edges		
		Both edges		
		Simultaneous input		Hold

The valid edges for ENCATAIN and ENCATBIN are specified by setting ENCAnEIS1 and ENCAnEIS0.

Count operation is performed when the valid edges of ENCATAIN and ENCATBIN overlap.

The following timing chart shows the count operation when ENCAnUDS1 and ENCAnUDS0 = {0, 1}.

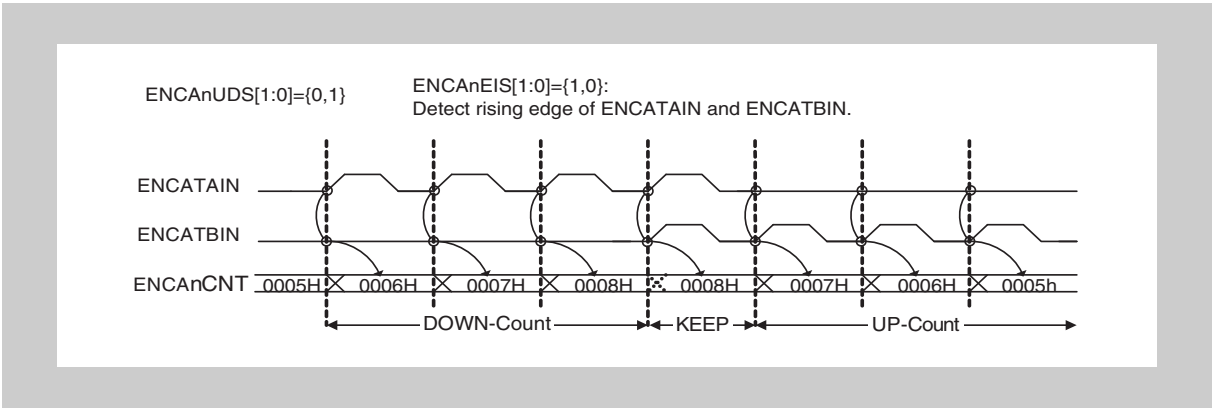


Figure 14-4 Count Operation when ENCAnUDS1 and ENCAnUDS0 = {0, 1}

(3) When ENCAnUDS1 and ENCAnUDS0 = {1, 0}

ENCAnUDS1	ENCAnUDS0	Operation description		
		ENCATAIN input	ENCTBIN input	Count operation
1	0	Rising edge	Low level	Down
		Rising edge	Falling edge	
		Falling edge	Low level	Up
		Falling edge	Falling edge	
		Low level	Rising edge	Hold
		Rising edge	Rising edge	
		High level	Rising edge	
		Falling edge	Rising edge	
		Low level	Falling edge	
		Rising edge	High level	
		High level	Falling edge	
		Falling edge	High level	

Valid edge specification for ENCATAIN and ENCATBIN (settings of ENCAnEIS1 and ENCAnEIS0) is invalid.

The following timing chart shows the count operation when ENCAnUDS1 and ENCAnUDS0 = {1, 0}.

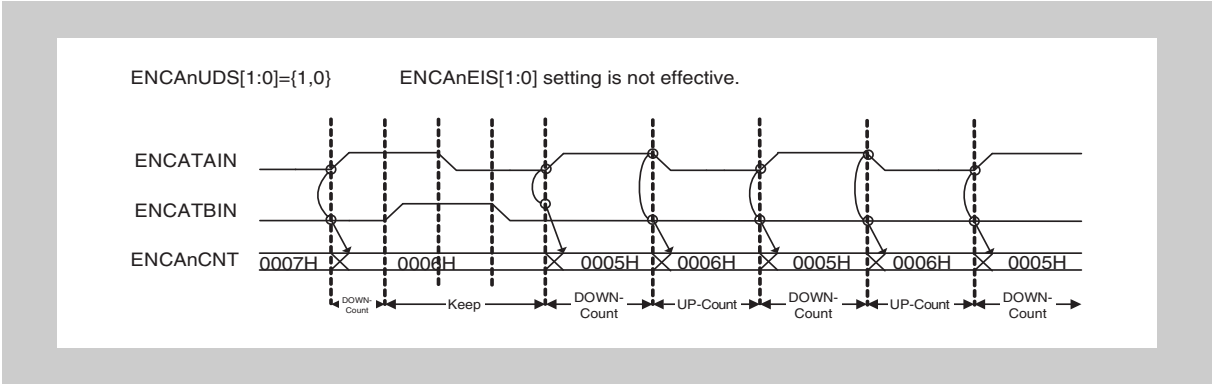


Figure 14-5 Count Operation when ENCAnUDS1 and ENCAnUDS0 = {1, 0}

(4) When ENCAnUDS1 and ENCAnUDS0 = {1, 1}

ENCAnUDS1	ENCAnUDS0	Operation description		
		ENCATAIN input	ENCTBIN input	Count operation
1	1	Low level	Falling edge	Down
		Rising edge	Low level	
		High level	Rising edge	
		Falling edge	High level	
		Rising edge	High level	Up
		High level	Falling edge	
		Falling edge	Low level	
		Low level	Rising edge	
		Simultaneous input		Hold

Valid edge specification for ENCATAIN and ENCATBIN (settings of ENCAnEIS1 and ENCAnEIS0) is invalid.

The count value is held when the valid edges of ENCATAIN and ENCATBIN overlap.

The following timing chart shows the count operation when ENCAnUDS1 and ENCAnUDS0 = {1, 1}.

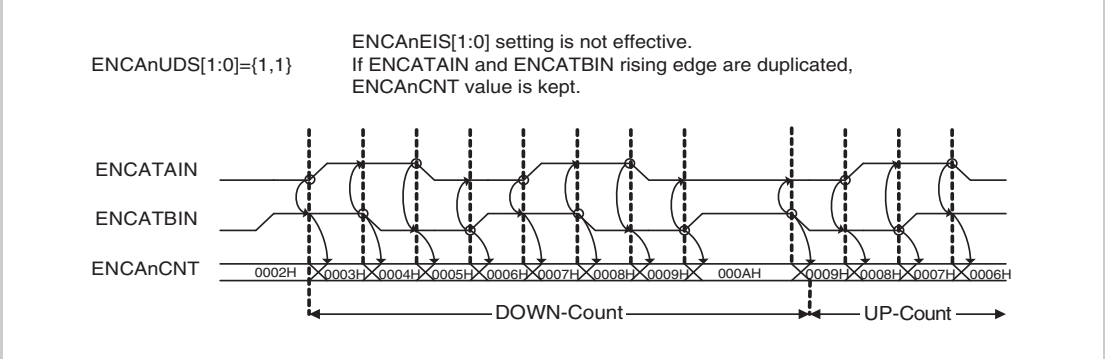


Figure 14-6 Count Operation when ENCAnUDS1 and ENCAnUDS0 = {1, 1}

14.4.3 Timer counter clear control by encoder input

The timer counter is cleared to 0000H by phase Z encoder input (ENCATZIN).

Two types of clearing methods can be selected by controlling the ENCAnSCE, ENCAnZCL, ENCAnBCL, ENCAnACL, ENCAnECS1, and ENCAnECS0 bits of the ENCAnIOC1 register.

Clearing Method	ENCAnSCE	ENCAnZCL	ENCAnBCL	ENCAnACL	ENCAnECS1, ENCAnECS0
(1)	0	Invalid	Invalid	Invalid	Valid
(2)	1	Valid	Valid	Valid	Invalid

(1) Clearing method when ENCAnSCE = 0

- Upon detection of the valid edge of ENCATZIN, the timer counter is cleared to 0000_H in synchronization with the operation clock.
- The valid edge of ENCATZIN is specified by the setting of the ENCAnECS1 and ENCAnECS0 bits.
- The settings of the ENCAnZCL, ENCAnBCL, and ENCAnACL bits are invalid.
- A clear interrupt signal (ENCATIEC) is output simultaneously with timer counter clearing.

(2) Clearing method when ENCAnSCE = 1

- The clear levels of the ENCATZIN, ENCATBIN, ENCATAIN inputs are detected according to the settings of the ENCAnZCL, ENCAnBCL, and ENCAnACL bits, and the timer counter is cleared to 0000_H in synchronization with the operating clock.
- The settings of the ENCAnECS1 and ENCAnECS0 bits are invalid.
- An encoder clear interrupt signal (ENCATIEC) is output simultaneously with timer counter clearing.

The clearing conditions of the timer counter according to the ENCAnZCL, ENCAnBCL, and ENCAnACL settings are listed in the table below.

Counter Clear Condition Setting			Encoder Input Level		
ENCAnZCL	ENCAnBCL	ENCAnACL	ENCATZIN	ENCATBIN	ENCATAIN
0	0	0	Low	Low	Low
0	0	1	Low	Low	High
0	1	0	Low	High	Low
0	1	1	Low	High	High
1	0	0	High	Low	Low
1	0	1	High	Low	High
1	1	0	High	High	Low
1	1	1	High	High	High

14.4.4 Functions of ENCAnCCR0

(1) Compare function

- When ENCAnCRM0 = 0, the ENCAnCCR0 register functions as a dedicated compare register.
- Upon compare match between the value of the timer counter and the ENCAnCCR0 setting value, a compare 0 match interrupt (ENCATINT0) is output.
- When ENCAnECM0 = 1, the timer counter is cleared to 0000_H in synchronization with the operating clock upon compare match if the next count operation is up-count.

ENCAnCCR0 function	Compare match clear control	Next count operation	Timer counter clearing upon compare match with ENCAnCCR0
ENCAnCRM0	ENCAnECM0		
0 (Compare)	0	Up-count	Does not clear (continues count operation).
		Down-count	
	1	Up-count	Clears timer counter to 0000_H.
		Down-count	Does not clear (continues count operation).

When ENCAnLD = 1

- Upon occurrence of an underflow, the setting value of the ENCAnCCR0 register is loaded to the timer counter.
- An underflow interrupt (ENCATIUD) is output.

(2) Capture function

- When ENCAnCRM0 = 1, the ENCAnCCR0 register functions as a dedicated capture register.
- Upon valid edge detection of the capture trigger input 0 (ENCATTIN0), the value of the timer counter is stored into ENCAnCCR0.
- A capture 0 interrupt (ENCATINT0) is output during capture operation.

14.4.5 Functions of ENCAnCCR1

(1) Compare function

- When ENCAnCRM1 = 0, the ENCAnCCR1 register functions as a dedicated compare register.
- Upon compare match between the value of the timer counter and the ENCAnCCR1 setting value, a compare 1 match interrupt (ENCATINT1) is output.
- When ENCAnECM1 = 1, the timer counter is cleared to 0000_H in synchronization with the operating clock upon compare match if the next count operation is down-count.

ENCAnCCR1 function	Compare match clear control	Next count operation	Timer counter clearing upon compare match with ENCAnCCR1
ENCAnCRM1	ENCAnECM1		
0 (Compare)	0	Up-count	Does not clear (continues count operation).
		Down-count	
	1	Up-count	Does not clear (continues count operation).
		Down-count	Clears timer counter to 0000_H.

Compare match interrupt detection mask function

- When ENCAnCME = 1, the compare 1 match interrupt detection mask function is enabled. In this state, the compare 1 match interrupt is output upon the first match of the value of the timer counter and the ENCAnCCR1 setting value, and interrupts are then masked for the second and subsequent compare matches.
- When ENCAnMCS = 0, the compare 1 match interrupt detection mask function is disabled by a write operation to the ENCAnCCR1 register.
- When ENCAnMCS = 1, the compare 1 match interrupt detection mask function is disabled by a timer counter clear operation accompanying phase Z or by a timer counter clear operation upon match between the ENCAnCCR0 register value and the timer counter value.
- When ENCAnMCS = 1 and ENCAnLDE = 1, the compare 1 match interrupt detection mask function is disabled by a loading operation of the ENCAnCCR0 register to the timer counter upon underflow detection.
- Setting ENCAnECM to “1” is prohibited when enabling the compare 1 match interrupt detection mask function.

ENCAnCCR1 function	Compare 1 match interrupt mask	Interrupt mask cancel trigger		Compare 1 match interrupt output upon compare match with ENCAnCCR1
ENCAnCRM1	ENCAnCME	ENCAnMCS	Underflow occurrence upon ENCAnLDE = 0	
0 (Compare)	0 (Mask function disabled)	– (Setting invalid)	-	Outputs compare 1 match interrupt upon each compare match.
	1 (Mask function enabled)	0 (Write operation to ENCAnCCR1)	Occurred (Loading of ENCAnCCR0 to timer counter)	Outputs compare 1 match interrupt once upon the first compare match. (Interrupts are masked for the second and subsequent matches until the cancel trigger occurs.)
		1 (Timer counter clear operation)		

(2) Capture function

- When ENCAnCRM1 = 1, the ENCAnCCR1 register functions as a dedicated capture register.

The operations for each of the ENCAnCTS settings are shown in the table below.

ENCAnCCR1 function	Capture trigger selection	Capture trigger signal	Timer counter clearing	Interrupt occurrence
ENCAnCRM1	ENCAnCTS			
1 (Capture)	0	Capture trigger 1 input (ENCATTIN1)	Does not clear timer counter.	(1) Capture 1 interrupt (ENCATINT1)
	1	Encoder clear input accompanying phase Z (ENCATZIN)	Clears timer counter.	(1) Capture 1 interrupt (ENCATINT1) (2) Encoder clear interrupt (ENCATIEC)

(3) Timer counter clearing upon compare register match

Timer counter clearing upon compare match between the value of the timer counter and the ENCAnCCR0/1 setting value, according to the settings of the ENCAnECM1 and ENCAnECM0 bits, is detailed in the following table.

ENCAnECM1 and ENCAnECM0	Next count operation	Timer counter clearing upon compare match with ENCAnCCR1	Timer counter clearing upon compare match with ENCAnCCR0
00	Up-count	Does not clear (continues count operation).	Does not clear (continues count operation).
	Down-count	Does not clear (continues count operation).	Does not clear (continues count operation).
01	Up-count	Does not clear (continues count operation).	Clears timer counter to 0000_H.
	Down-count	Does not clear (continues count operation).	Does not clear (continues count operation).
10	Up-count	Does not clear (continues count operation).	Does not clear (continues count operation).
	Down-count	Clears timer counter to 0000_H.	Does not clear (continues count operation).
11	Up-count	Does not clear (continues count operation).	Clears timer counter to 0000_H.
	Down-count	Clears timer counter to 0000_H.	Does not clear (continues count operation).

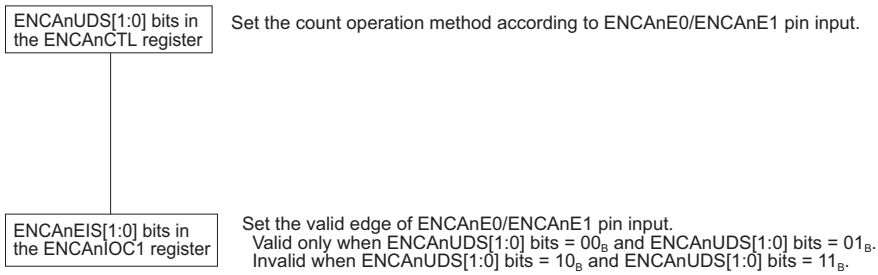
14.5 Setting Sequences

14.5.1 Encoder timer setting procedure

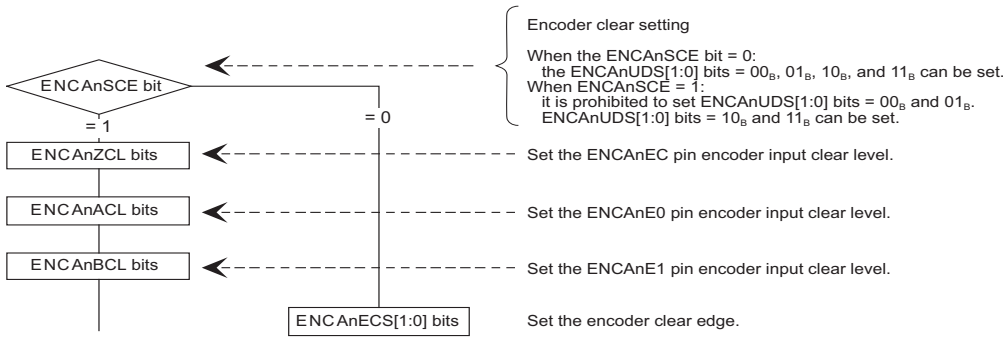
The encoder timer setting procedure is described below.

	Action	Setting status
Initial Setting		Power-off status (Writing to each register is disabled)
	Reset release	Power-on status, ENCAn operation stopped status. (Writing to each register is enabled)
ENCA Initial Settings	Perform the following initial settings. • Setting for counter • Setting for counter clear • Setting for ENCAnCCR0 register • Setting for ENCAnCCR1 register	This is the count operation stopped status. The value of the ENCAnTE bit indicating the operating status is 0.
	Perform the counter initial value settings. • Set any 16-bit value to ENCAnCNT register. (When, after setting this register, the ENCAnTS bit is set to “1”, the counter operation starts from the set count value.)	The set value is set as the initial value of the counter register.
Operation Start	Perform the counter operation start setting. • Set the ENCAnTS bit to “1”.	This is the counter operation starts status. The value of the ENCAnTE bit indicating the operating status is 1, and the count clock is supplied to the internal circuit.
Operating	Only those registers whose setting can be changed during operation can be rewritten. • ENCAnCCR0 register setting can be changed. • ENCAnCCR1 register setting can be changed. • ENCAnIOC0 register setting can be changed.	The count operation set with the initial setting is performed, and up/down counting is performed according to ENCATAIN and ENCATBIN.
Operation Stop	Perform the counter operation stop setting during operation. • Set the ENCAnTT bit to “1”.	This is the counter operation stopped status. The value of the ENCAnTE bit indicating the operating status is 0.
ENCA stop	Reset	This is the power-off status. The entire circuit and all the setting registers are initialized.

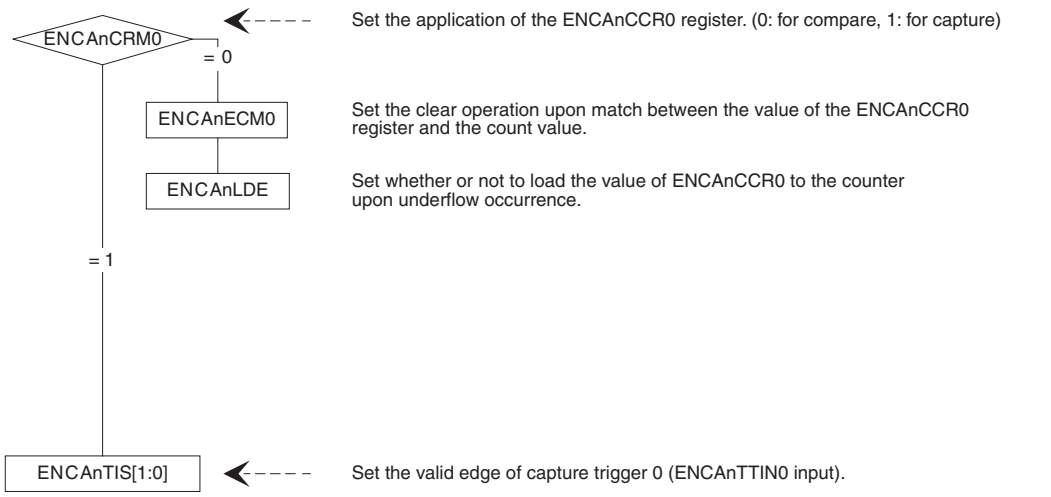
(1) Initial Setting Procedure for the Counter



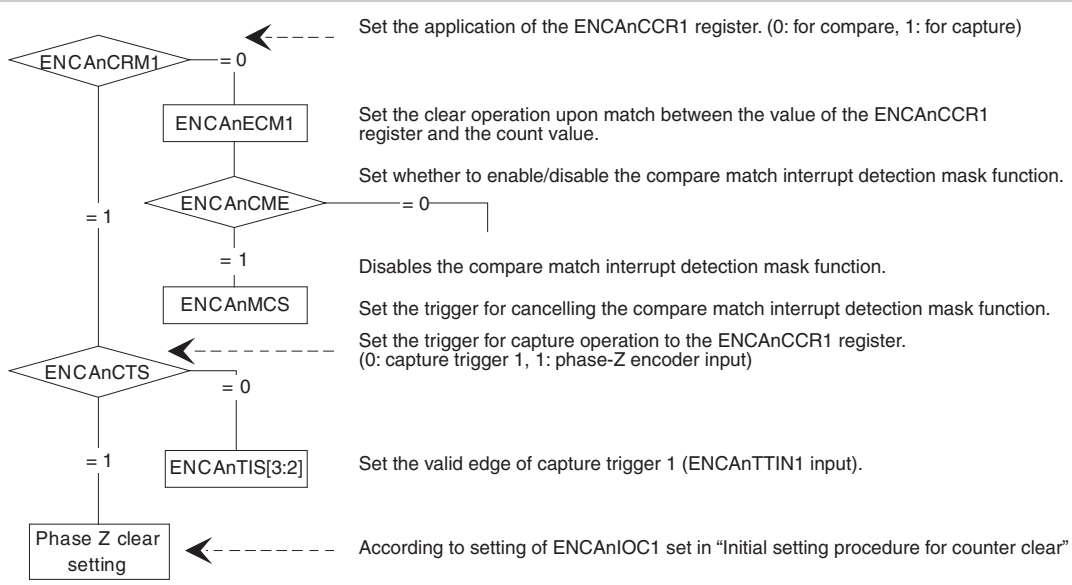
(2) Initial setting procedure for counter clear



(3) Setting procedure for ENCAnCCR0 register



(4) Setting procedure for ENCAnCCR1 register



14.6 Timing Chart

14.6.1 Overflow Occurrence and Overflow Flag Clear Operation

The operations of overflow occurrence is described below.

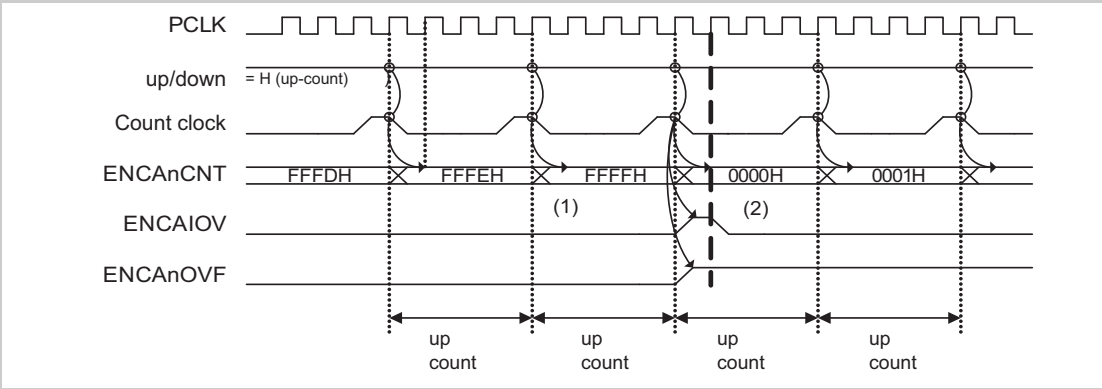


Figure 14-7 Overflow Occurrence

1. An overflow occurs when up-counting is performed when the counter value is $FFFF_{H.}$
2. When the count value changes from $FFFF_H$ to 0000_H, an overflow occurs. At the same time, an overflow interrupt is output and the overflow flag is set to 1.

The operations of overflow flag clearing is described below.

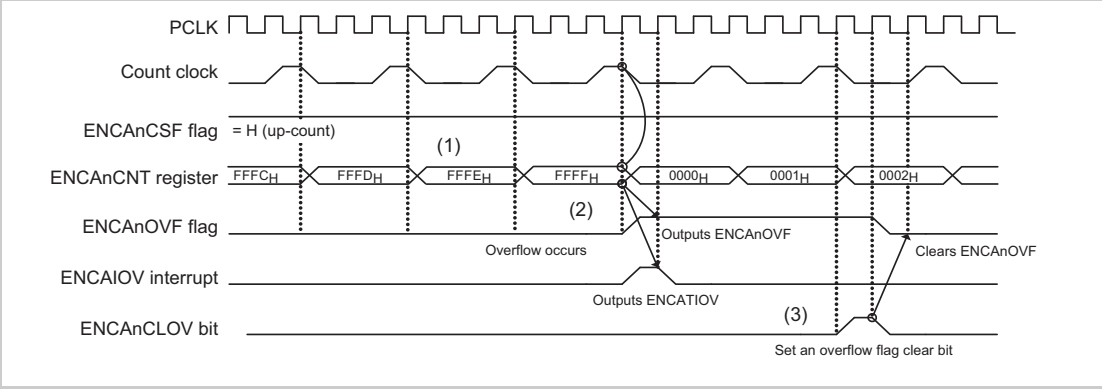


Figure 14-8 Overflow Flag Clear

1. The count value is counted up from $FFFE_H$ to $FFFF_H$.
2. When the count value changes from $FFFF_H$ to 0000_H, an overflow occurs. At the same time, an overflow interrupt is output and the overflow flag is set to 1.
3. By setting the ENCAnCLOV bit in the ENCAnFGC register to 1 by the overflow flag clearing method, the overflow flag is cleared to 0. The overflow flag is also cleared by setting the ENCAnTS bit in the ENCAnTS register to 1 when the ENCAnTE bit in the ENCAnTE register is 0, or setting the input signal of ENCAnTSST (simultaneous start trigger input) to “High”.

14.6.2 Underflow Occurrence and Underflow Flag Clear Operation

The operations of underflow occurrence is described below.

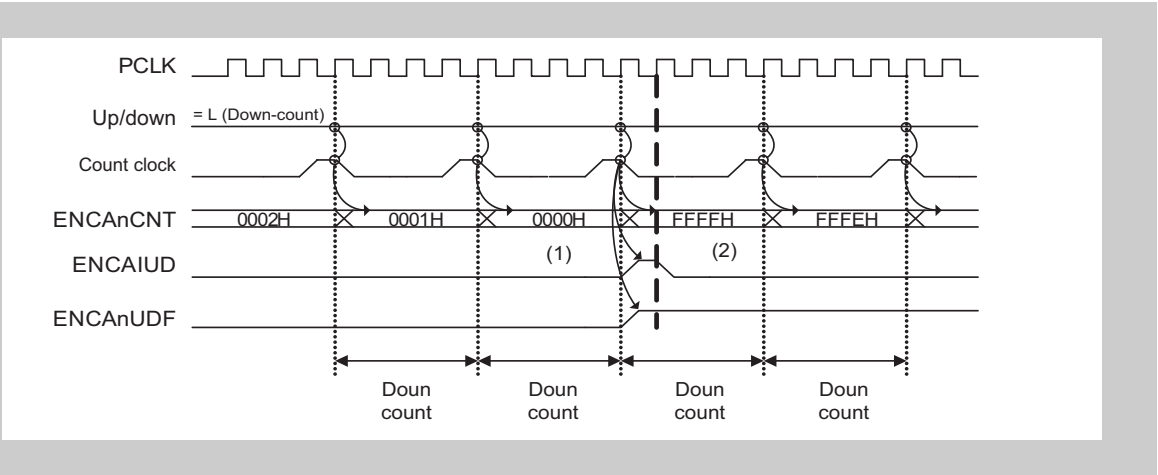


Figure 14-9 Underflow Occurrence

1. An underflow occurs when down-counting is performed when the counter value is 0000_H.
2. When the count value changes from 0000_H to $FFFF_H$, an underflow occurs. At the same time, an underflow interrupt is output and the underflow flag is set to 1.

The operations of underflow flag clearing is described below

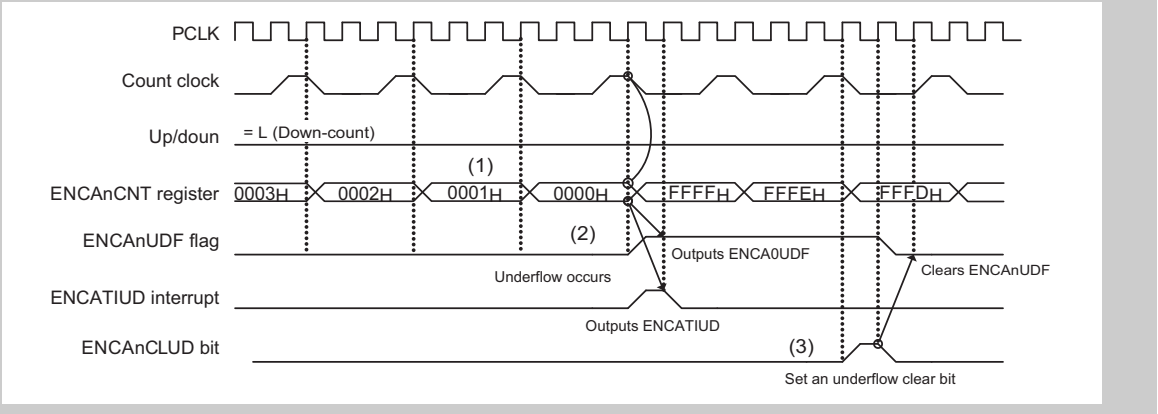


Figure 14-10 Underflow Flag Clear

1. The count value is counted down from 0001_H to 0000_H
2. When the count value changes from 0000_H to $FFFF_H$, an underflow occurs. At the same time, an underflow interrupt is output and the underflow flag is set to 1.
3. By setting the ENCAnCLUD bit in the ENCAnFGC register to 1 by the underflow flag clearing method, the underflow flag is cleared to 0. The underflow flag is also cleared by setting the ENCAnTS bit in the ENCAnTS register to 1 when the ENCAnTE bit in the ENCAnTE register is 0, or by setting the input signal of ENCAnTSST (simultaneous start trigger) to “High”.

14.6.3 Capture Operation by phase Z

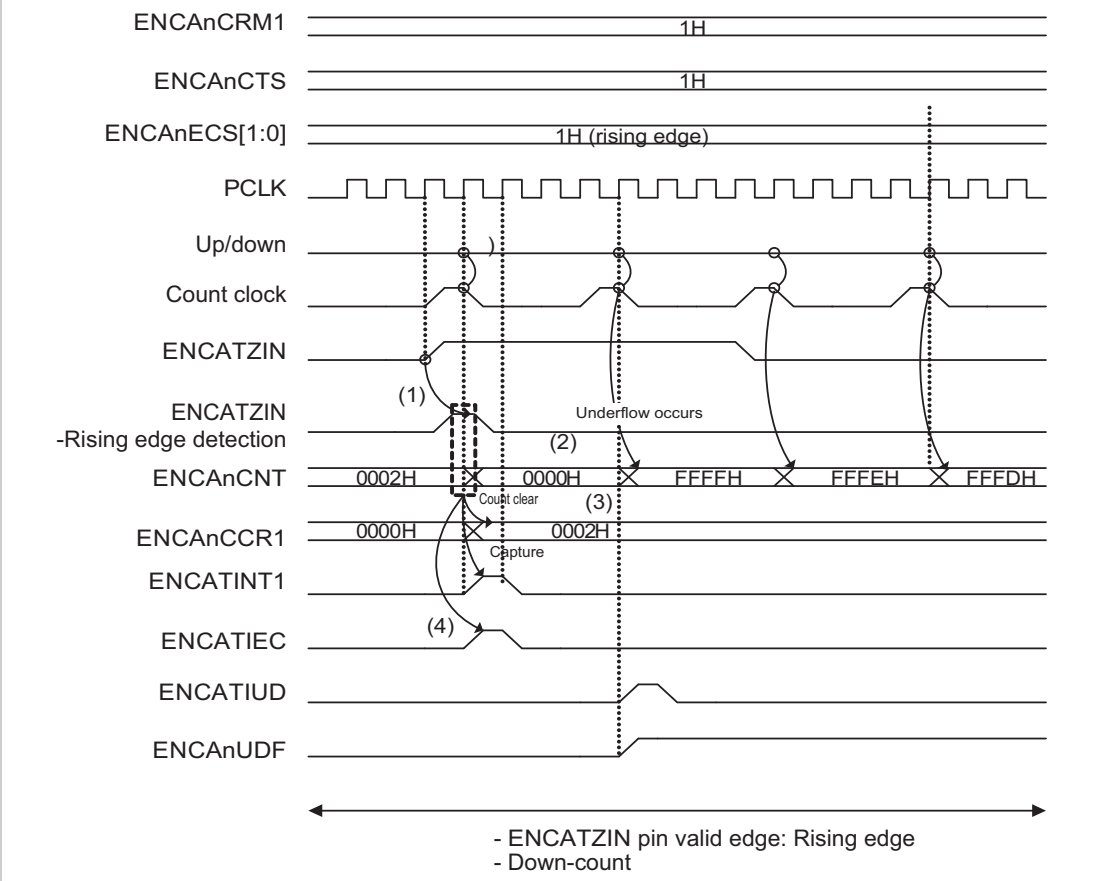


Figure 14-11 Timing Chart of Capture Operation by phase Z

1. Capture operation is performed by the rising edge of the ENCATZIN pin input trigger
   (The rising ede: ENCAnECS1 and ENCAnECS0 = “01”).
2. Clearing is performed by the phase Z and the count value is set to 0000_H.
3. The counter value (0002_H) is captured in the ENCAnCCR1 register by the rising edge of the ENCAZIN pin input
4. At the same time, a clear interrupt (ENCATIEC) and capture interrupt (ENCATINT1) due to the phase Z are output.

14.6.4 Conflict between Overflow Occurrence and Clear Operation by phase Z Input

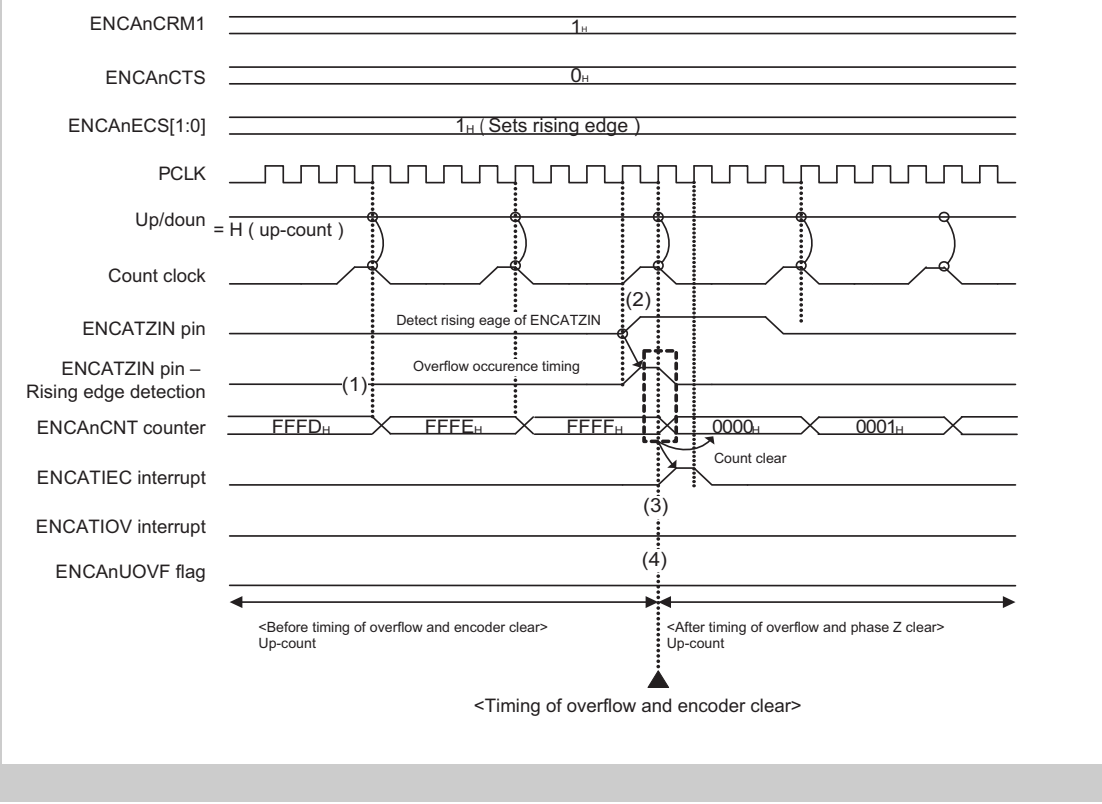


Figure 14-12 Conflict between Overflow Occurrence and Clear Operation by phase Z Input

1. An up-count from $FFFD_H$ is continuously performed.
2. When an overflow occurs if the count value is $FFFF_H$, and the rising edge of ENCATZIN is detected simultaneously, clear operation by the phase Z input is performed. The counter value is cleared to 0000_H.
3. When the counter value is cleared by the phase Z input, a clear interrupt (ENCATIEC) by phase Z input is output simultaneously. Because a clear operation by the phase Z input is performed simultaneously with the overflow occurrence, an overflow interrupt is not output (An overflow does not occur. Clear operation is performed by the phase Z input).
4. Because an overflow does not occur as is the case with step 3, the overflow flag is not set.

14.6.5 Conflict between Underflow Occurrence and Clear Operation by phase Z Input

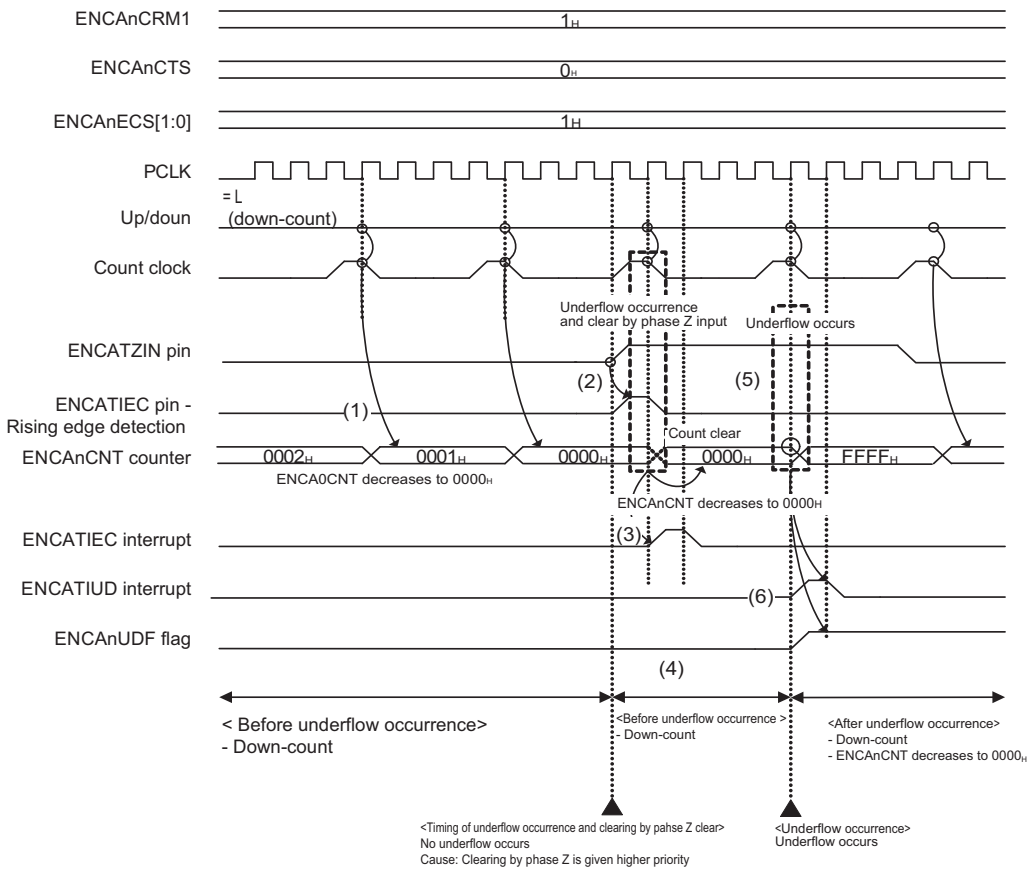


Figure 14-13 Conflict between Underflow Occurrence and Clear Operation by phase Z Input

1. A down-count from 0002_H is continuously performed.
2. When an underflow occurs if the count value is 0000_H, and the rising edge of ENCATZIN is detected simultaneously, clear operation by the phase Z input is performed. Even if the next clock signal is input during clear operation, the counter value remains at 0000_H.
3. When the counter value is cleared by the phase Z input, a phase Z clear interrupt (ENCATIEC) is output simultaneously. Because a clear operation by the phase Z input is performed simultaneously with the underflow occurrence, an underflow interrupt is not output (An underflow does not occur. Clear operation is performed by the phase Z input).
4. Because an underflow does not occur as is the case with step 3, the underflow flag is not set.
5. When a further down-count is performed after the counter value changes to 0000_H by clear operation by the phase Z input, the counter value changes from 0000_H to $FFFF_H$, and an underflow occurs.
6. When an underflow occurs, an underflow interrupt (ENCATIUD) is output, and the underflow flag (ENCAnUDF) is set.

14.6.6 Overflow Operation Immediately after Startup

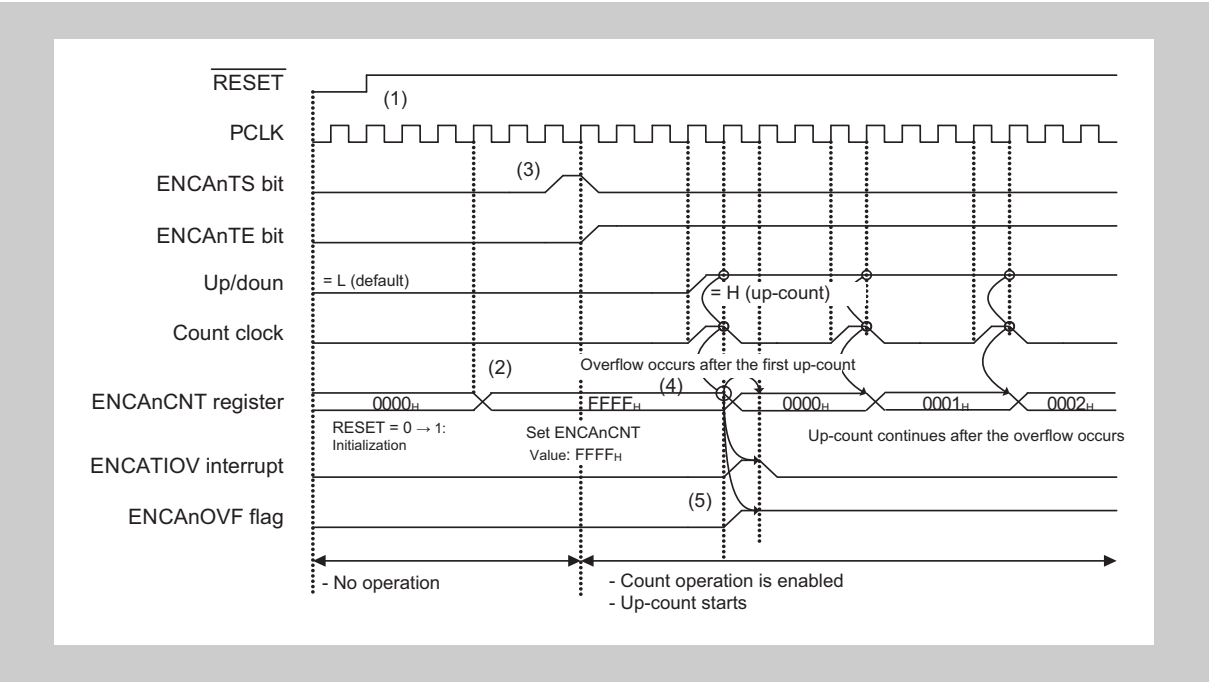


Figure 14-14 Overflow Operation Immediately after Startup

1. When the $\overline{\text{RESET}}$ value changes from “0” to “1”, the status is changes from “reset” to “reset release”.
2. The timer counter is set to $FFFF_H$ as the initial value.
3. ENCAnTS is set to “1”, and operation starts. ENCAnTE changes to “1”, which indicates that operation is enabled.
4. When an up-count is performed from $FFFF_H$ which is the initially set count value, the counter value changes from $FFFF_H$ to 0000_H, and an overflow occurs immediately after operation starts.
5. At the same time, by an overflow occurrence immediately after operation starts, an overflow interrupt (ENCATIOV) is output, and the overflow flag (ENCAnOVF) is set

14.6.7 Underflow Operation Immediately after Startup

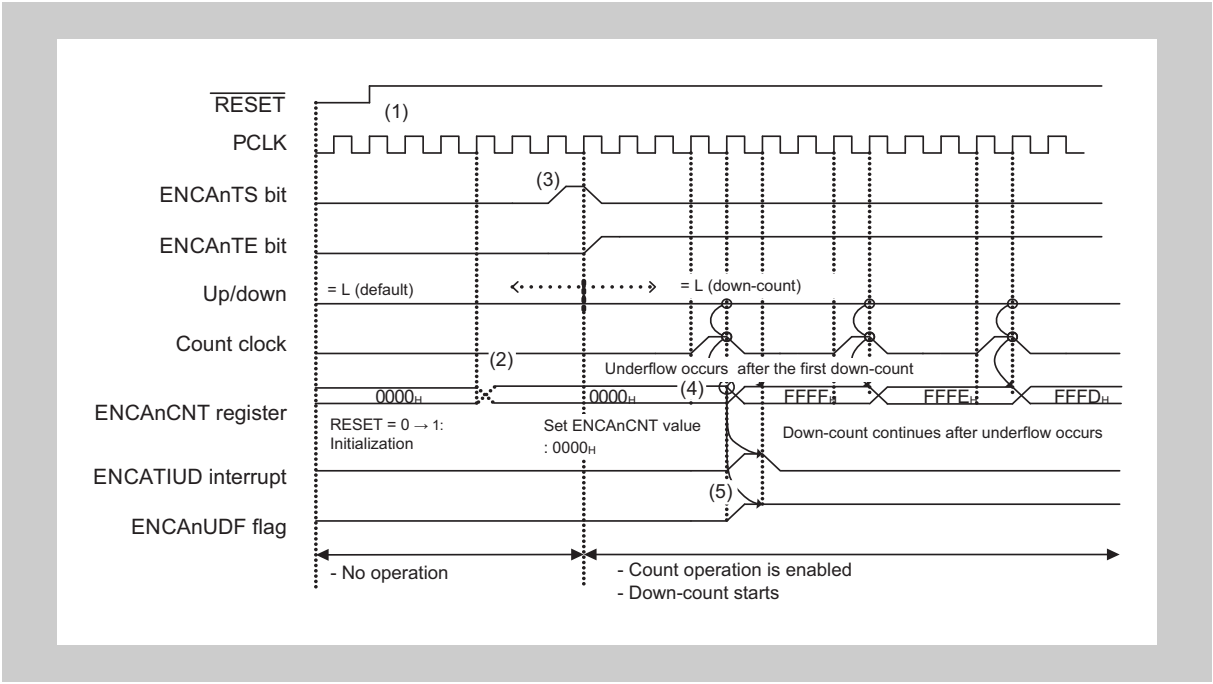


Figure 14-15 Underflow Operation Immediately after Startup

1. When the $\overline{\text{RESET}}$ value changes from “0” to “1”, the status is changes from “reset” to “reset release”.
2. The timer counter is set to 0000_H as the initial value.
3. ENCAnTS is set to “1”, and operation starts. ENCAnTE changes to “1”, which indicates that operation is enabled.
4. When a down-count is performed from 0000_H which is the initially set count value, the counter value changes from 0000_H to $FFFF_H$, and an underflow occurs immediately after operation starts.
5. At the same time, by an underflow occurrence immediately after operation starts, an underflow interrupt (ENCATIUD) is output, and the underflow flag (ENCAnUDF) is set.

14.6.8 Using the ENCAnLDE Function Immediately after Startup

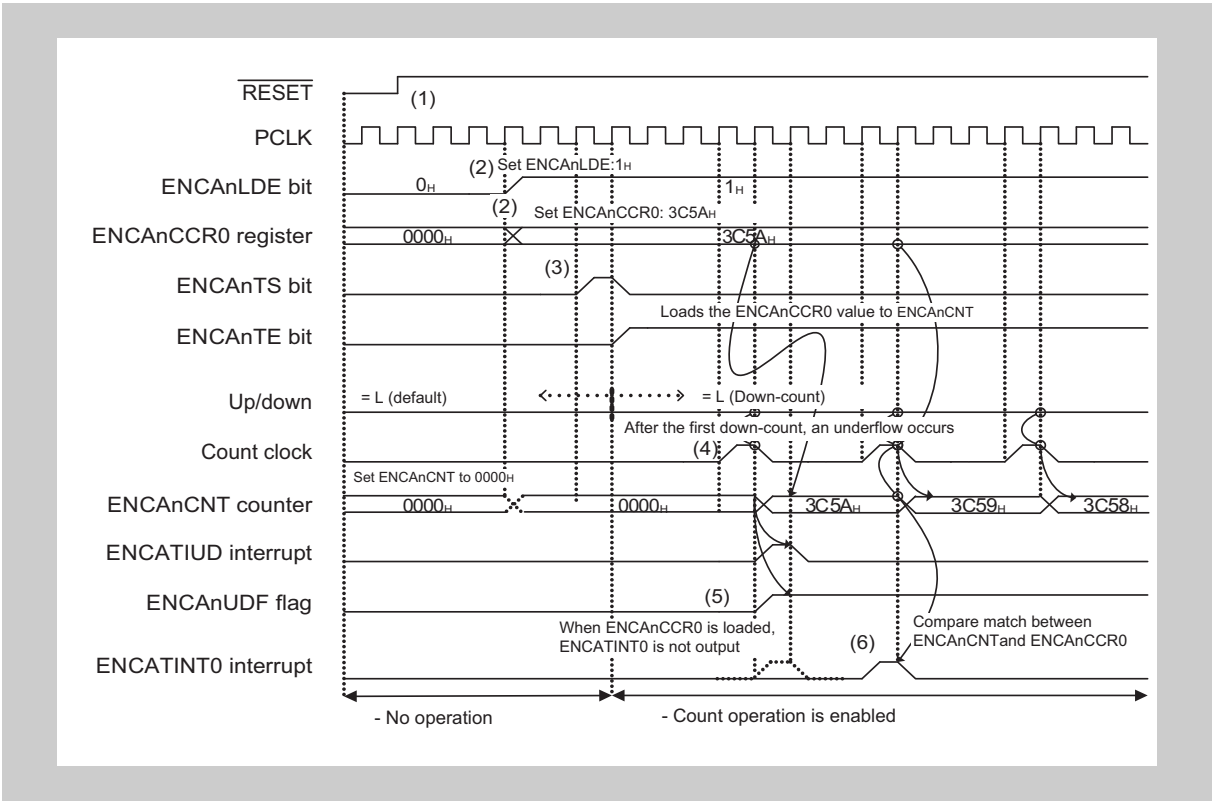


Figure 14-16 Using the ENCAnLDE Function Immediately after Startup

1. When the $\overline{\text{RESET}}$ value changes from “0” to “1”, the status is changes from “reset” to “reset release”.
2. The load enable bit (ENCAnLDE) is set to “1”, capture/compare register 0 (ENCAnCCR0) is set to $3C5A_H$, and the timer counter is set to the initial value 0000_H.
3. ENCAnTS is set to “1”, and operation starts. ENCAnTE changes to “1”, which indicates that operation is enabled.
4. When a down-count is performed from 0000_H which is the initially set count value, an underflow occurs immediately after operation starts. Because ENCAnLDE is set to “1”, the ENCAnCCR0 value, $3C5A_H$, is loaded to the timer counter (ENCATINT0 is not output during loading).
5. At the same time, by an underflow occurrence immediately after operation starts, an underflow interrupt (ENCATIUD) is output, and the underflow flag (ENCAnUDF) is set (After an underflow occurs, down-count operation from the loaded value ($3C5A_H$) continues).
6. After the ENCAnCCR0 value is loaded to ENCAnCNT, a match with ENCAnCCR0 is detected, and ENCATINT0 is output.

14.6.9 ENCAnLDE Function (Loading Count Value)

(1) <When ENCAnLDE = 0>

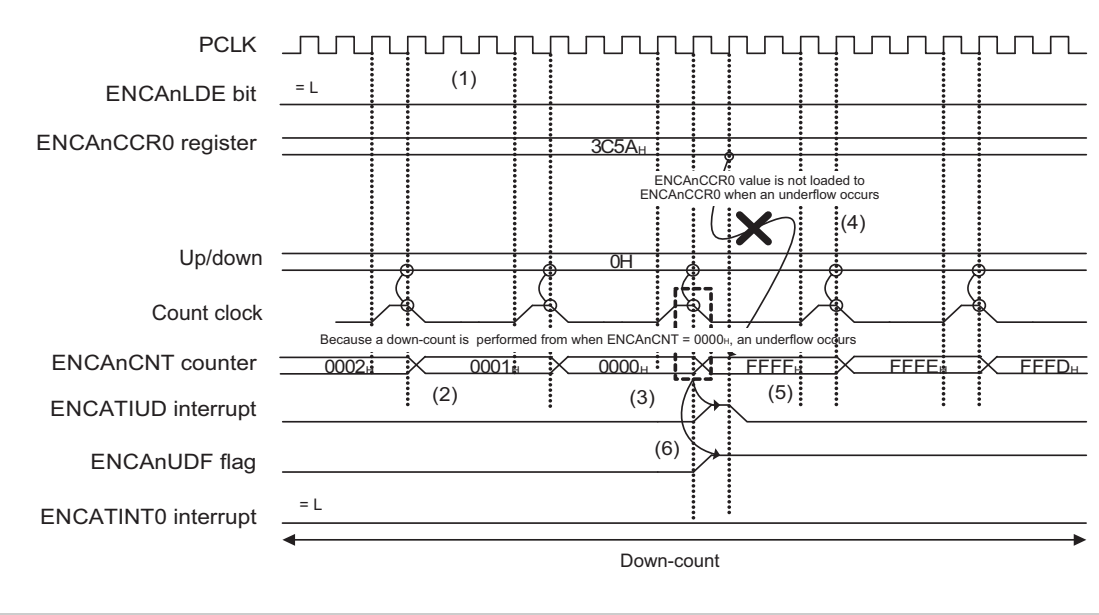


Figure 14-17 ENCAnLDE Function (when ENCAnLDE = 0)

1. ENCAnLDE is set to “0” (even if an underflow occurs, the ENCAnCCR0 value is not loaded).
2. A down-count is performed: $0002_H \rightarrow 0001_H \rightarrow 0000_H$
3. When a further down-count is performed after the counter value changes to 0000_H, an underflow occurs.
4. Because ENCAnLDE is set to “0”, the setting value of the ENCAnCCR0 register is not loaded to the counter even if an underflow occurs.
5. Operation changes to underflow operation (counter value: 0000_H ? $FFFF_H$).
6. An underflow interrupt (ENCATIUD) is output, and the underflow flag (ENCAnUDF) is set.

(2) <When ENCAnLDE = 1>

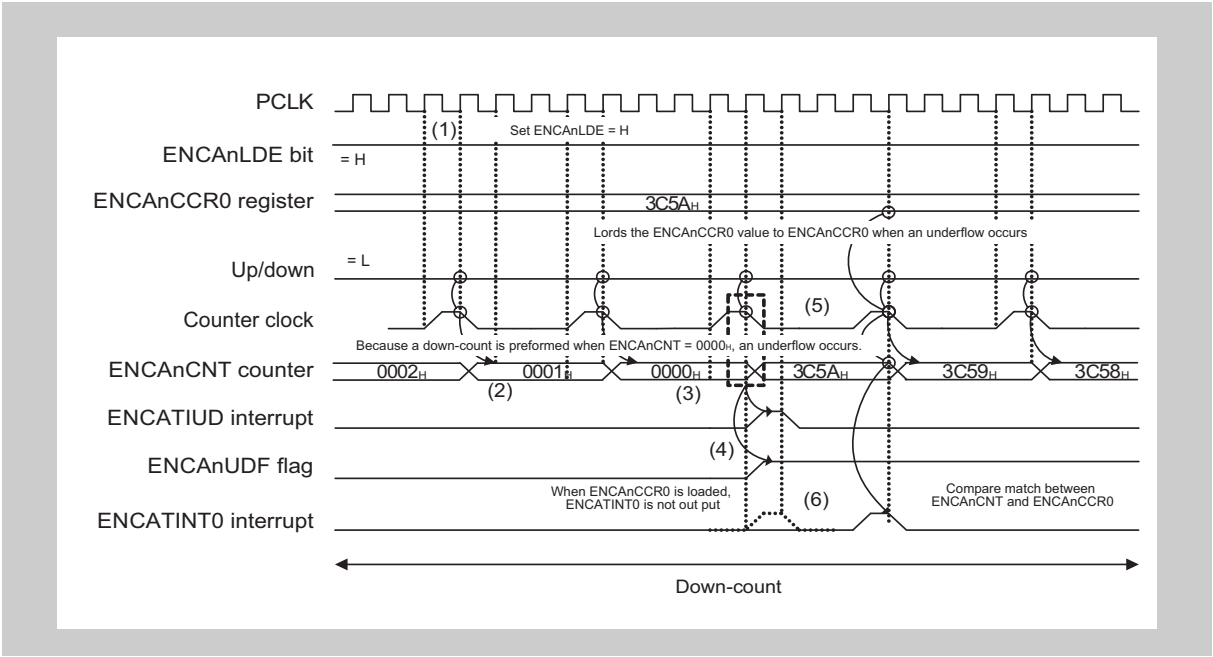


Figure 14-18 ENCAnLDE Function (when ENCAnLDE = 1)

1. ENCAnLDE is set to “1” (if an underflow occurs, the ENCAnCCR0 value is loaded to the counter).
2. A down-count is performed: $0002_H \rightarrow 0001_H \rightarrow 0000_H$.
3. When a further down-count is performed after the counter value changes to 0000_H, an underflow occurs.
4. An underflow interrupt is output, and the underflow flag is set.
5. Because ENCAnLDE is set to “1”, the setting value of the ENCAnCCR0 register is loaded to the counter if an underflow occurs. ENCAnCNT is set to $3C5A_H$.
6. After the ENCAnCCR0 value is set to ENCAnCNT, if the ENCAnCNT value matches the ENCAnCCR0 value on a count clock, a compare match interrupt (ENCATINT0) is output.

14.6.10 Conflict between ENCAnLDE Function (Loading Counter Value) and Rewrite of ENCAnCCR0 Register

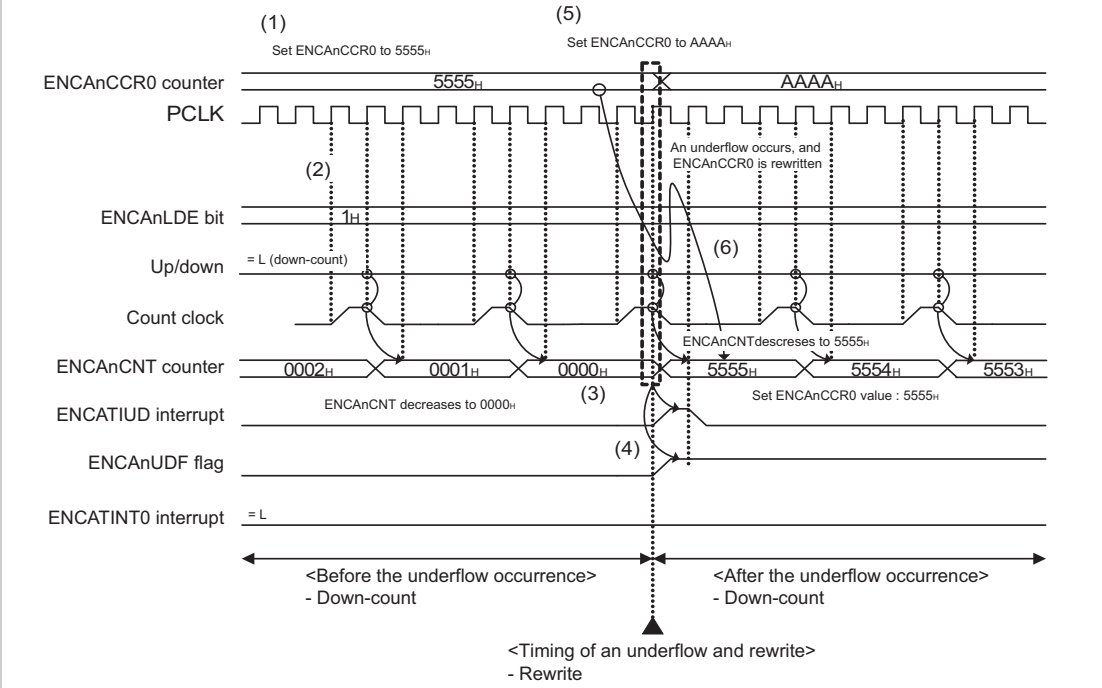


Figure 14-19 Conflict between ENCAnLDE Function and Rewrite Of ENCAnCCR0 Register

1. The ENCAnCCR0 register is currently set to 5555_H.
2. ENCAnLDE is currently set to “1”.
3. A down-count is performed ($0002_H \rightarrow 0001_H \rightarrow 0000_H$), and an underflow occurs.
4. An underflow interrupt (ENCATIUD) is output, and the underflow flag (ENCAnUDF) is set.
5. When an underflow occurs, the ENCAnCCR0 register value is changed from 5555_H to $AAAA_H$.
6. Additionally, when an underflow occurs, the ENCAnCCR0 value before the rewrite was performed (5555_H) is set in ENCAnCNT.

14.6.11 Conflict between ENCAnLDE Function (Loading Counter Value) and Clear Operation by Phase Z Input

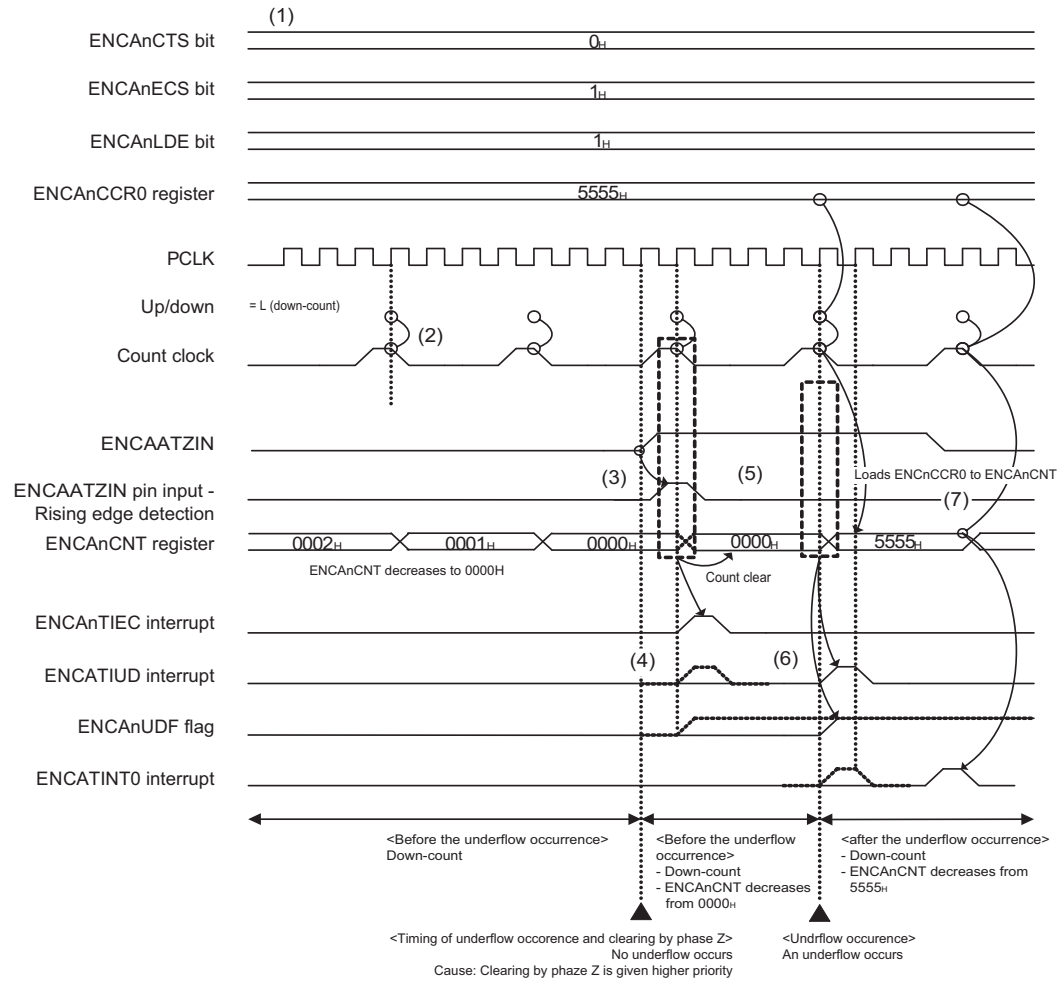


Figure 14-20 Conflict between ENCAnLDE Function (Loading Counter Value) and Clear Operation by Phase Z Input

1. The values are set as follows: ENCAnCTS = 0, ENCAnECS[1:0] = 01_B, ENCAnLDE = 1, and ENCAnCCR0 = 5555_H.
2. A down-count is performed: $0002_H \rightarrow 0001_H \rightarrow 0000_H$
3. When the count value becomes 0000_H, the rising edge of ENCAnEC pin is detected, and clear operation by the encoder clear input is performed.
4. Because a count clear is performed when the count value reaches 0000_H, a counter clear interrupt (ENCATIEC) by the phase Z input is output. An underflow does not occur because a down-count is not performed when the count value is 0000_H. Therefore, an underflow interrupt (ENCAIUD) is not output, and the underflow flag (ENCAnUDF) is not set.
5. After the count value is cleared to 0000_H by clear operation by the phase Z input, a down-count is performed and an underflow occurs.

6. An underflow interrupt (ENCATIUD) is output, and the underflow flag (ENCAnUDF) is set.
7. Because ENCAnLDE = “1”, if an underflow occurs, the ENCAnCCR0 value is loaded to ENCAnCNT.
8. After the ENCAnCCR0 value is set to ENCAnCNT, a compare match is detected according to the count clock. If the ENCAnCNT value matches the ENCAnCCR0 value, a compare match interrupt (ENCATINT0) is output.

14.6.12 Up-count after Conflict between ENCAnLDE Function (Loading Counter Value) and Clear Operation by the Phase Z Input

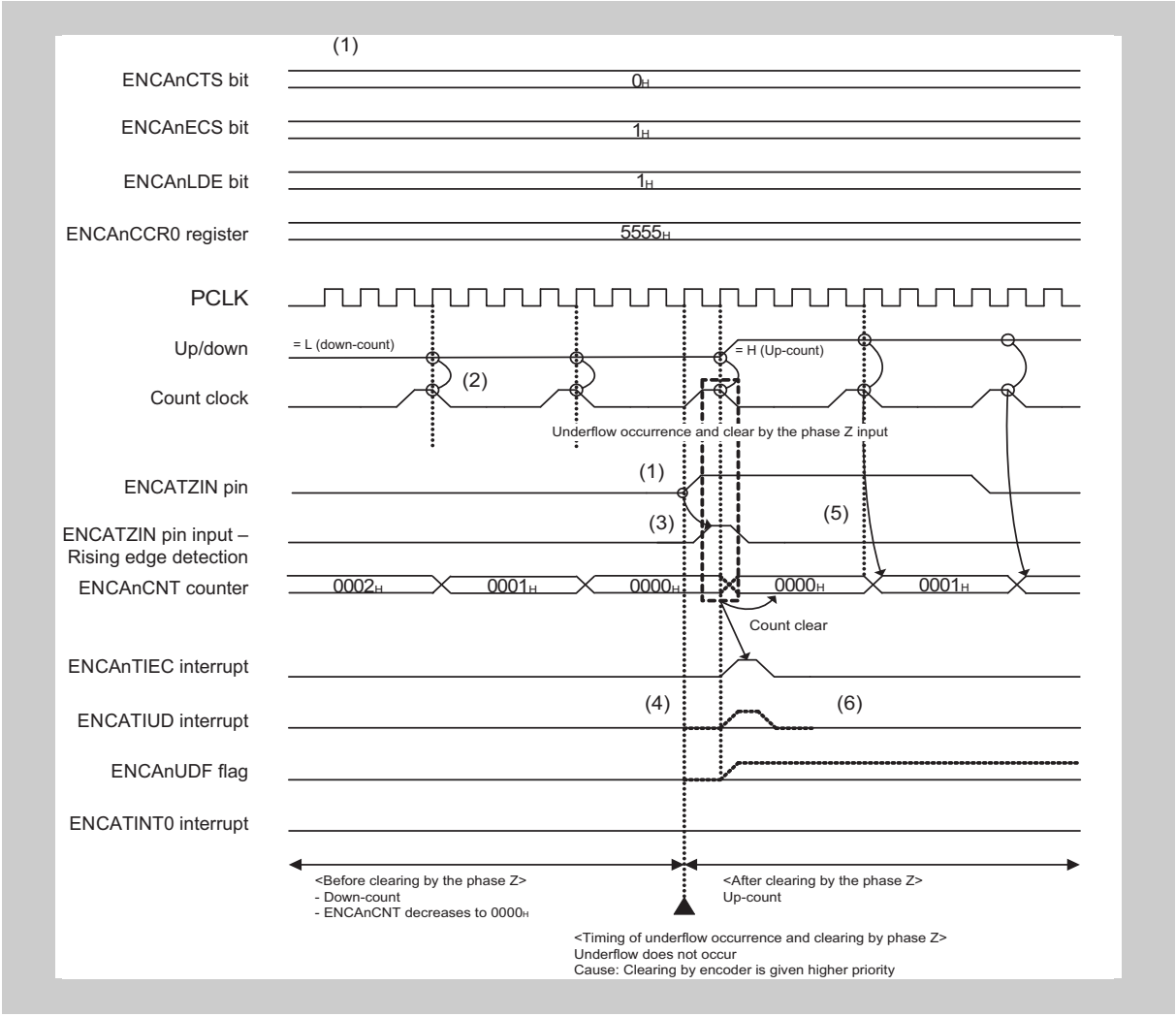


Figure 14-21 Up-count after Conflict between ENCAnLDE Function (Loading Counter Value) and Clear Operation by Encoder Clear

1. The values are set as follows: ENCAnCTS = 0, ENCAnECS[1:0] = 01B, ENCAnLDE = 1, and ENCAnCCR0 = 5555H.
2. A down-count is performed: $0002_H \rightarrow 0001_H \rightarrow 0000_H$
3. When the count value becomes 0000_H, the rising edge of ENCATZIN pin is detected, and clear operation by the phase Z input is performed.
4. Because a count clear is performed when the count value reaches 0000_H, a counter clear interrupt (ENCAnTIEC) by the phase Z input is output. An underflow does not occur because a down-count is not performed when the count value is 0000_H. Therefore, an underflow interrupt (ENCAIUD) is not output, and the underflow flag (ENCAnUDF) is not set.
5. After the count value is cleared to 0000_H by clear operation by the phase Z input, an up-count is performed
6. An underflow interrupt (ENCATIUD) is not output, and the underflow flag (ENCAnUDF) is not set.

14.6.13 Capture Operation between Count Clocks (ENCAnCCR1)

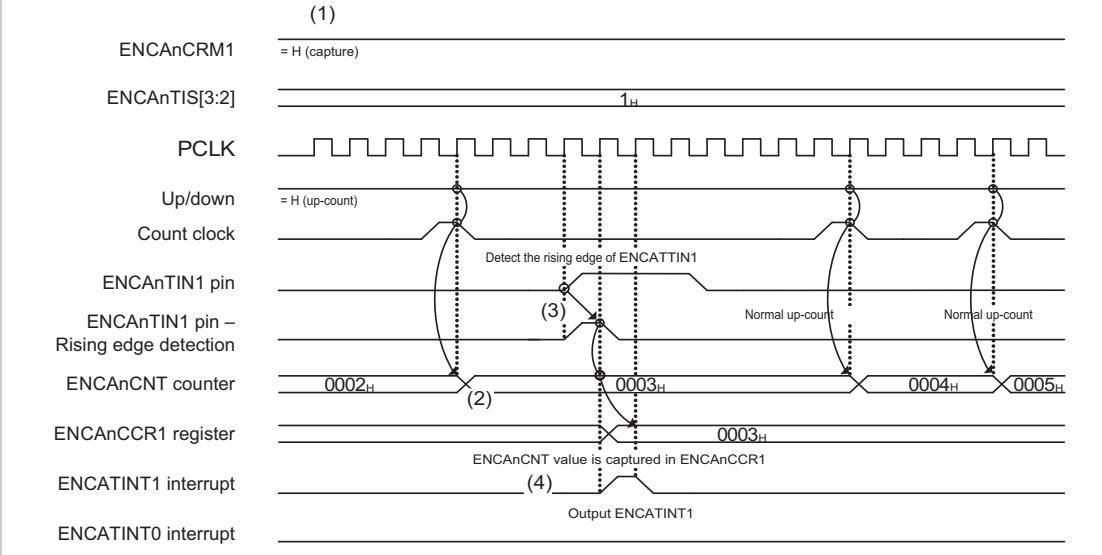


Figure 14-22 Capture Operation between Count Clocks (ENCAnCCR1)

1. The values are set as follows: ENCAnCRM1 = 1, and ENCAnTIS[3:2] = 01_B.
2. An up-count is performed.
3. The rising edge of the ENCATTIN1 input is detected, and the count value is captured in ENCAnCCR1.
4. An interrupt (ENCATINT1) corresponding to the capture to the ENCAnCCR1 register is output.

14.6.14 Capture Operation between Count Clocks (ENCAnCCR0)

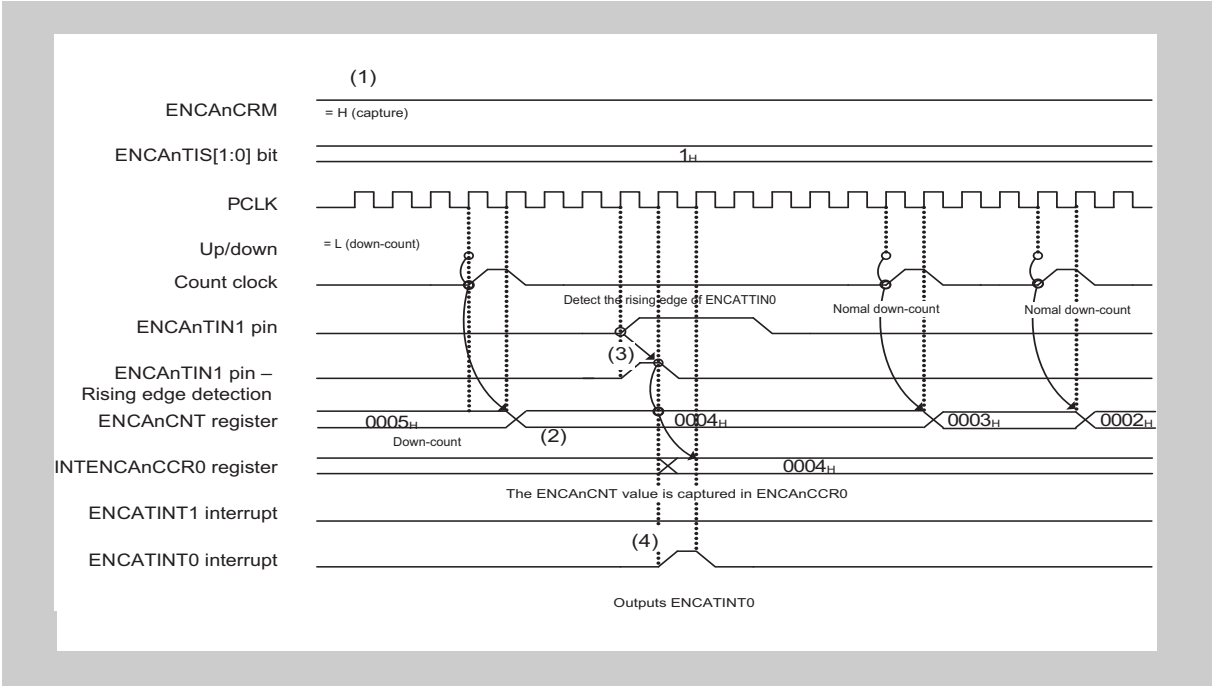


Figure 14-23 Capture Operation between Count Clocks (ENCAnCCR0)

1. The values are set as follows: ENCAnCRM0 = 1, and ENCAnTIS[1:0] = 01_B.
2. A down-count is performed.
3. The rising edge of the ENCATTIN0 input is detected, and the count value is captured in ENCAnCCR0.
4. An interrupt (ENCATINT0) corresponding to the capture to the ENCAnCCR0 register is output.

14.6.15 Encoder Operation when ENAnECM[1:0] =0, 1 and ENCAnCTS = 0

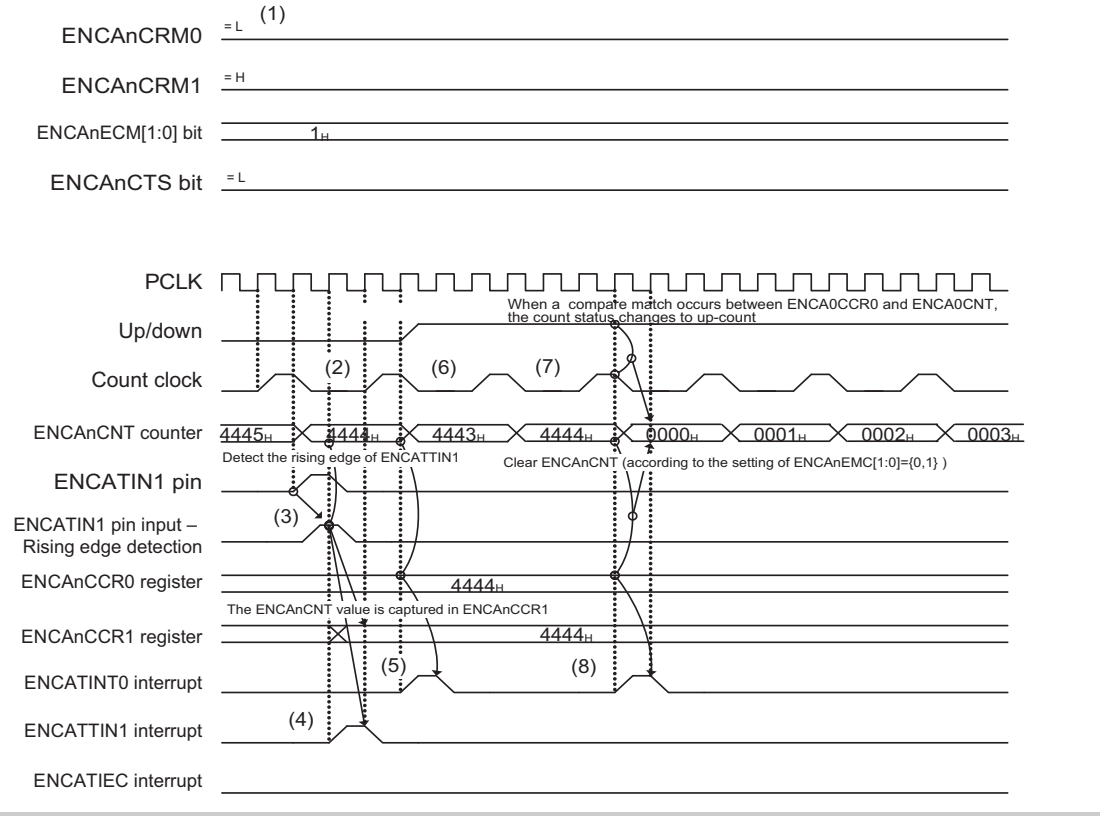


Figure 14-24 Encoder Operation when ENCAnECM[1:0] = 0,1 and ENCAnCTS = 0

1. The values are set as follows: ENCAnCCR0 = 4444_H, ENCAnCRM0 = 0, ENCAnCRM1 = 1, ENCAnECM[1:0] = 01B, and ENCAnCTS = 0.
2. A down-count is performed.
3. The rising edge of the ENCATTIN1 input is detected, and the ENCAnCNT value (4444_H) is captured in the ENCAnCCR1 register.
4. An interrupt signal (ENCATIEC) corresponding to the capture to the ENCAnCCR1 register is output.
5. When a compare match occurs between ENCAnCNT (counted down from 4445_H to 4444_H) and ENCAnCCR0 (4444_H), a compare match interrupt (ENCATINT0) with ENCAnCCR0 is output.
6. The count operation changes to up-count.
7. When ENCAnCNT is counted up from 4443_H to 4444_H, a compare match with ENCAnCCR0 occurs again. Because the count operation is up-count when the compare match occurs, the count value is cleared according to the setting of ENCAnECM1 and ENCAnECM0 (01_B), and the ENCAnCNT value changes to 0000_H.
8. When ENCAnCNT changes to 4444_H, a compare match interrupt (ENCATINT0) with ENCAnCCR0 is output.

14.6.16 Encoder Operation when ENCAnECM[1:0] =0, 1 and ENCAnCTS = 1

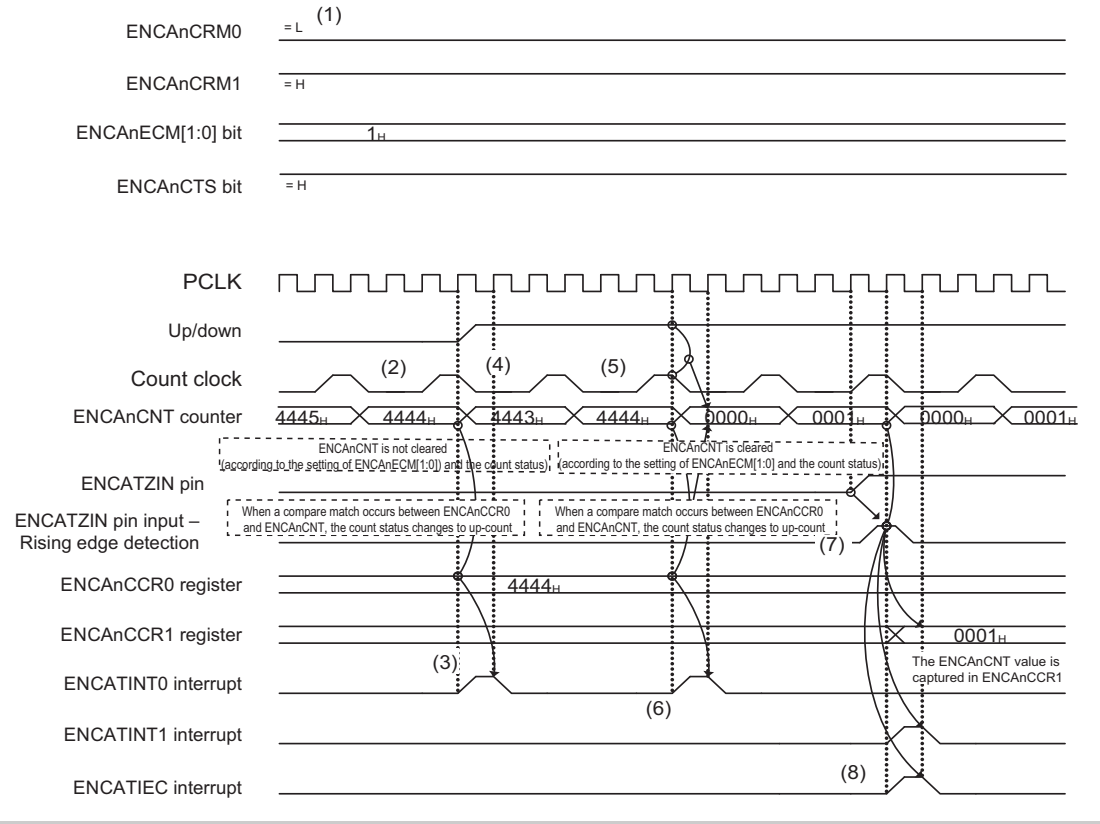


Figure 14-25 Encoder Operation when ENCAnECM[1:0] =0, 1 and ENCAnCTS = 1

1. The values are set as follows: ENCAnCCR0 = 4444H, ENCAnCRM0 = 0, ENCAnCRM1 = 1, ENCAnECM[1:0] = 01_B, and ENCAnCTS = 1.
2. A down-count is performed.
3. When a compare match occurs between ENCAnCNT (counted down from 4445_H to 4444_H) and ENCAnCCR0 (4444_H), a compare/capture interrupt (ENCATINT0) is output.
4. The count operation changes to up-count.
5. When ENCAnCNT is counted up from 4443_H to 4444_H, a compare match with ENCAnCCR0 occurs again. Because the count operation is up-count when the compare match occurs, the count value is cleared according to the setting of ENCAnECM1 and ENCAnECM0 (01_B), and the ENCAnCNT value changes to 0000_H.
6. When ENCAnCNT changes to 4444_H, a compare match interrupt (ENCATINT0) with ENCAnCCR0 is output.
7. After the count value is cleared, an up-count is performed, and the count value changes to 0001_H. At this point, the ENCAnCNT value (0001_H) is captured in ENCAnCCR1 by detecting the rising edge of the ENCAnEC signal, and the counter is cleared to 0000_H.

8. An interrupt (ENCATINT1) corresponding to the capture to the ENCAnCCR1 register and a clear interrupt (ENCATIEC) by ENCAnEC are output.

14.6.17 Encoder Operation when ENCAnECM1-0 = 0, 0

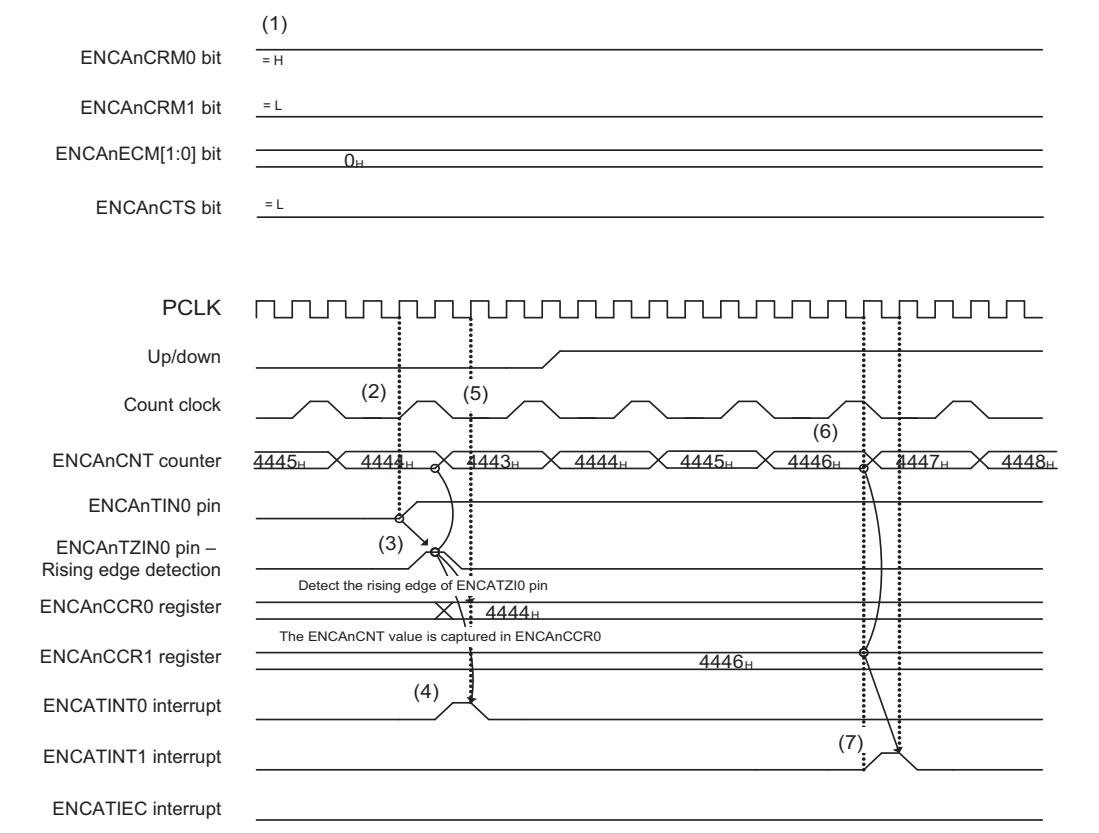


Figure 14-26 Encoder Operation when ENCAnECM1-0 = 0, 0

1. The values are set as follows: ENCAnCCR1 = 4446_H, ENCAnCRM0 = 1, ENCAnCRM1 = 0, ENCAnECM[1:0] = 00_B, and ENCAnCTS = 0.
2. A down-count is performed.
3. When the rising edge of ENCATTIN0 is detected, the ENCAnCNT value (4444_H) is captured in ENCAnCCR0.
4. An interrupt signal (ENCATINT0) corresponding to the capture to the ENCAnCCR0 register is output.
5. The count operation changes to up-count.
6. When ENCAnCNT changes to 4446_H, a compare match with ENCAnCCR1 is detected.
7. A compare match interrupt (ENCATINT1) with ENCAnCCR1 is output.

14.6.18 Capture Operation Performed upon Clearing by ENCATZIN when ENCAnSCE = 1

(1) Accompanying capture operation

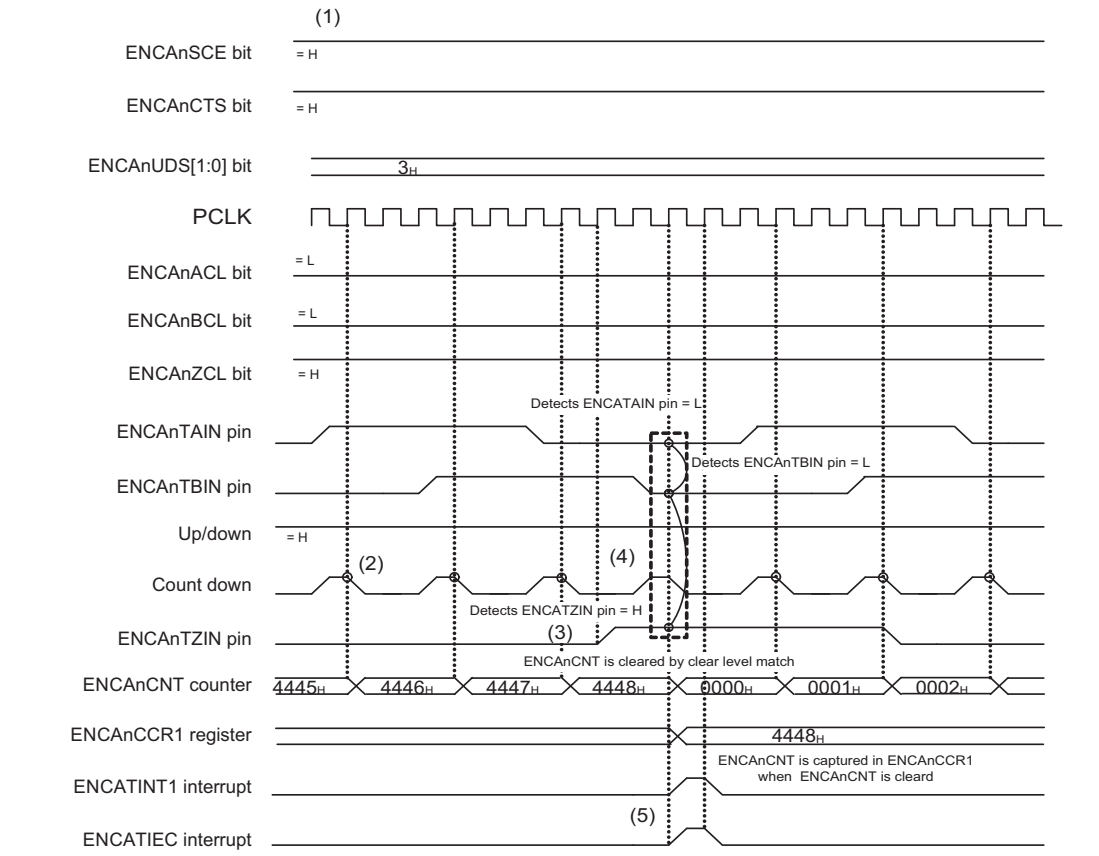


Figure 14-27 Capture Operation Performed upon Clearing by ENCATZIN when ENCAnSCE = 1

1. The values are set as follows: ENCAnSCE = 1, ENCAnCTS = 1, ENCAnUDS[1:0] =11_B, ENCAnACL = 0, ENCAnBCL = 0, and ENCAnZCL = 1.
2. An up-count is performed.
3. The count value is not cleared upon the rising edge of ENCAnEC.
4. When ENCATAIN, ENCATBIN and ENCATZIN reach the set clear level, the count value is cleared. The count value is captured in ENCAnCCR1 at the time of the clearing.
5. At the time of the clearing, an interrupt (ENCATINT1) corresponding to the capture to the ENCAnCCR1 register and a clear interrupt (ENCATIEC) by ENCATZIN are output.

(2) When the Timing of the ENCATZIN Input is Later than that of the ENCATBIN Input during Up-count (When ENCAnACL = 1, ENCAnBCL = 0, ENCAnZCL = 1, and ENCAnUDS[1:0] = 11_B)

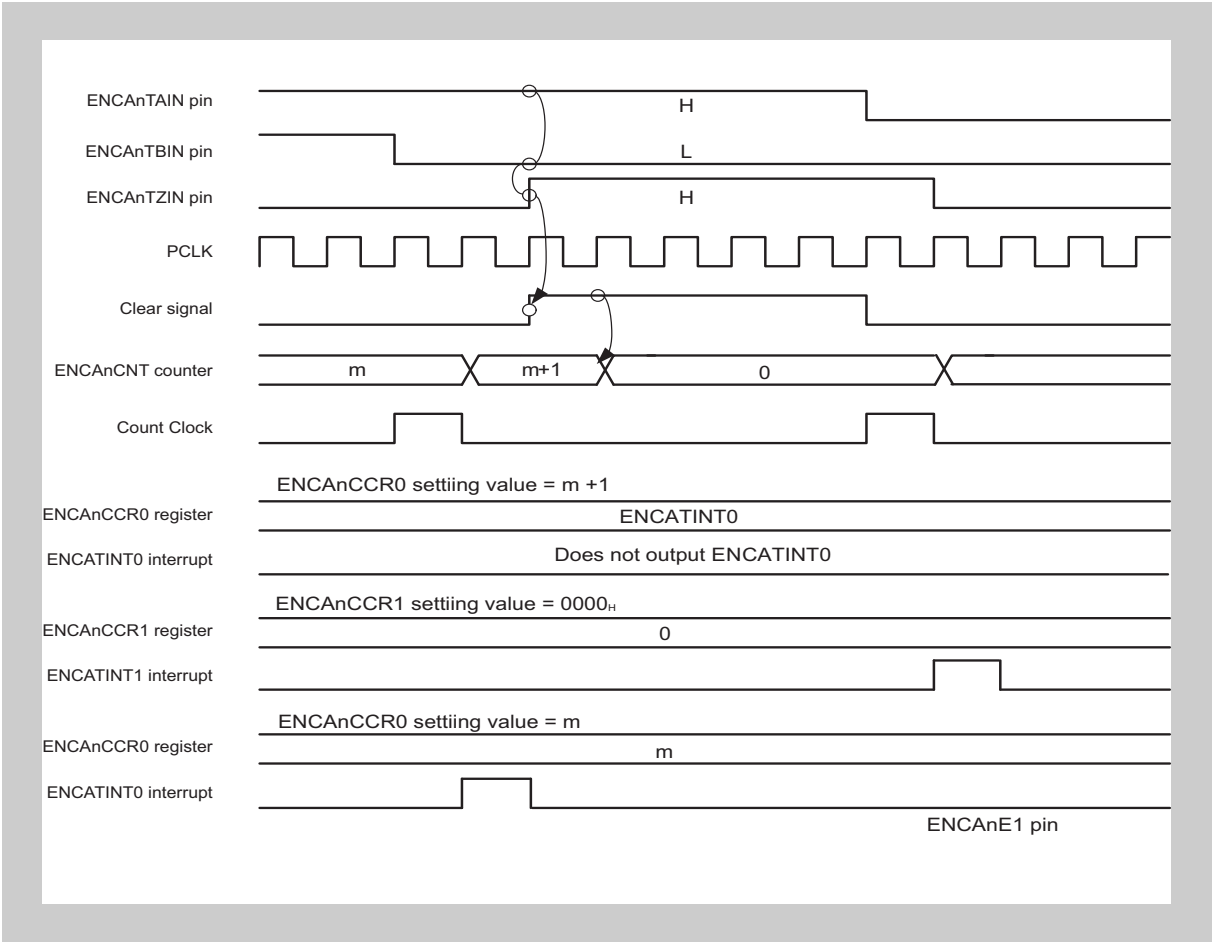


Figure 14-28 Clearing for when the Timing of the ENCATZIN Input is Later than that of the ENCATBIN Input during Up-count

(3) When the Timing of the ENCATZIN Input is the Same as that of the ENCATBIN Input during Up-count (When ENCAnACL = 1, ENCAnBCL = 0, ENCAnZCL = 1, and ENCAnUDS[1:0] = 11_B)

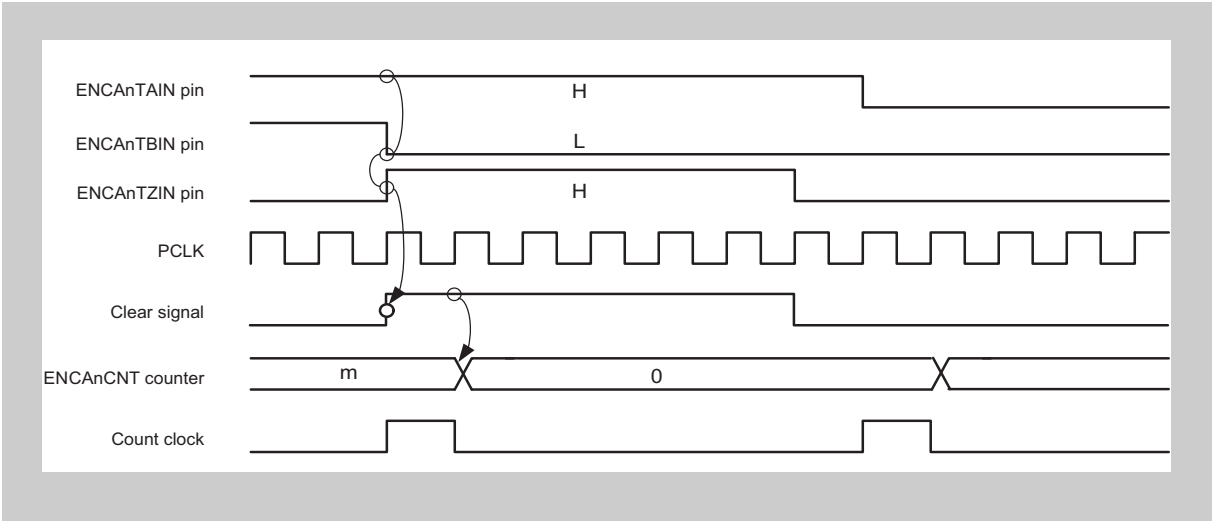


Figure 14-29 Clearing for when the Timing of the ENCATZIN Input is the Same as that of the ENCATBIN Input during Up-count

(4) When the Timing of the ENCATZIN Input is Earlier than that of the ENCATBIN Input during Up-count (When ENCAnACL = 1, ENCAnBCL = 0, ENCAnZCL = 1, and ENCAnUDS[1:0] = 11_B)

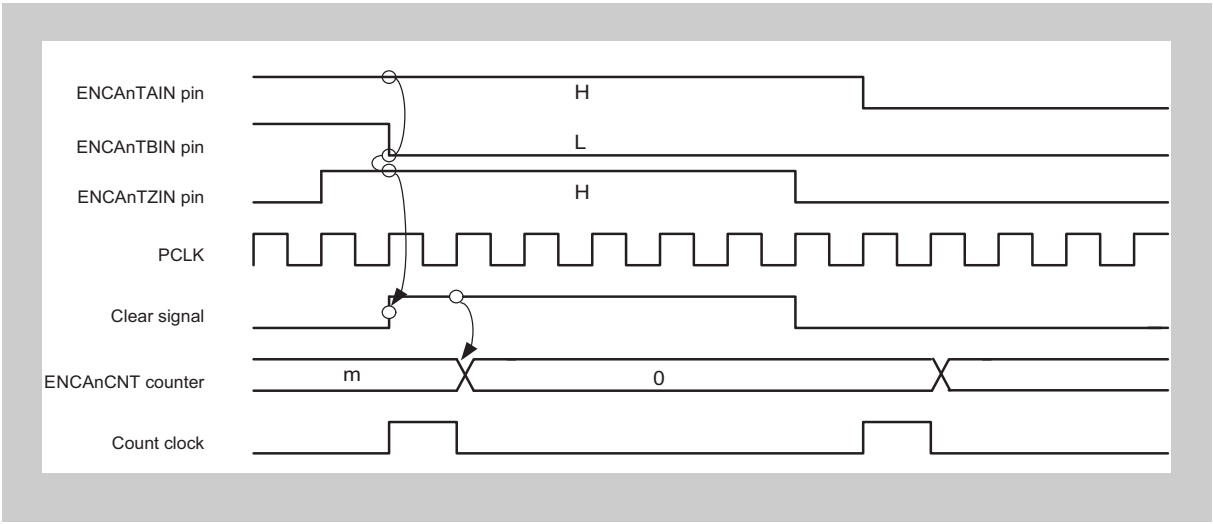


Figure 14-30 Clearing for when the Timing of the ENCATZIN Input is Earlier than that of the ENCATBIN Input during Up-count

(5) When the Timing of the ENCATZIN Input is Later than that of the ENCAnE1 Input during Down-count (When ENCATBIN = 1, ENCAnBCL = 0, ENCAnZCL = 1, and ENCAnUDS[1:0] = 11_B)

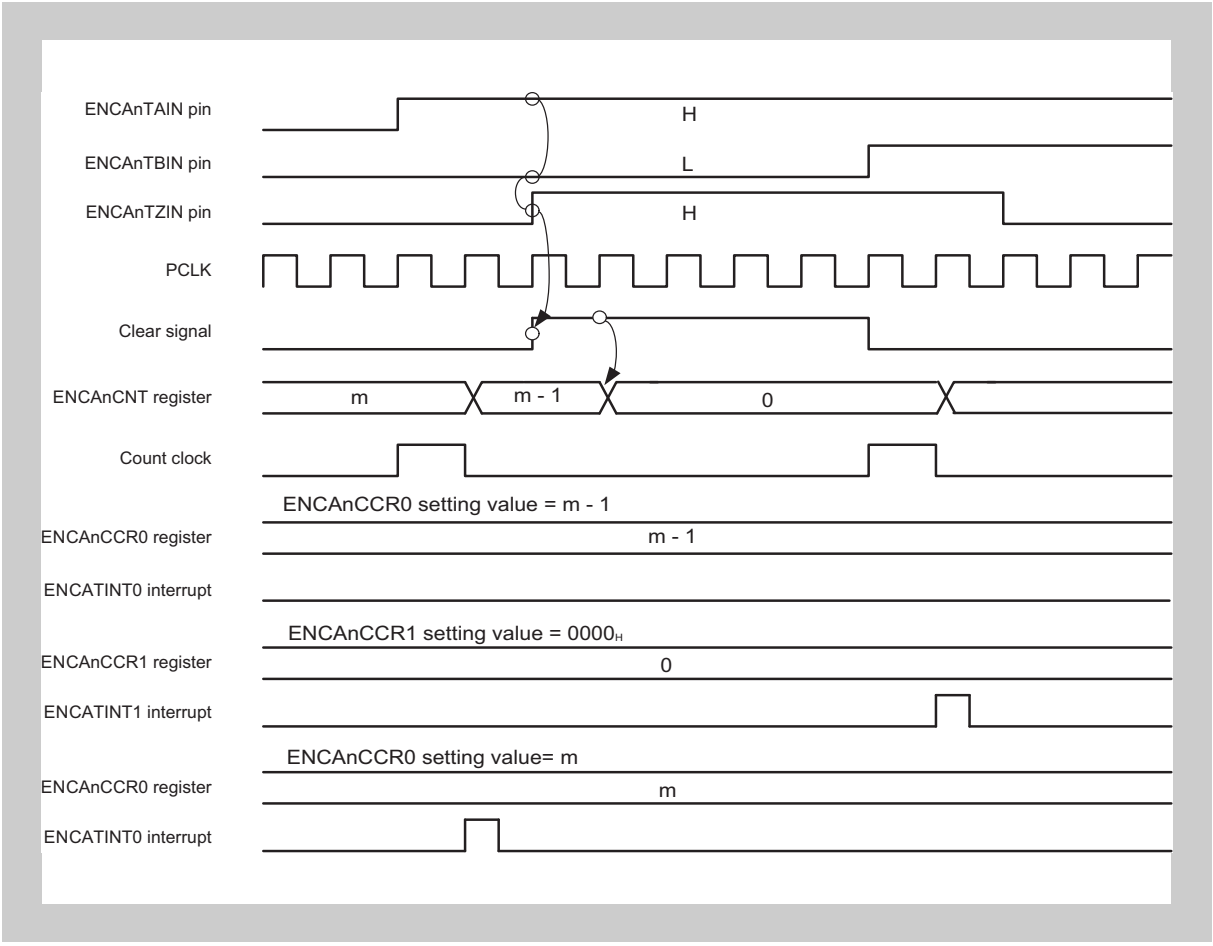


Figure 14-31 Clearing for when the Timing of the ENCATZIN Input is Later than that of the ENCATBIN Input during Down-count

14.6.19 Capture Operation Performed upon Clearing by ENCATZIN when ENCAnSCE = 0

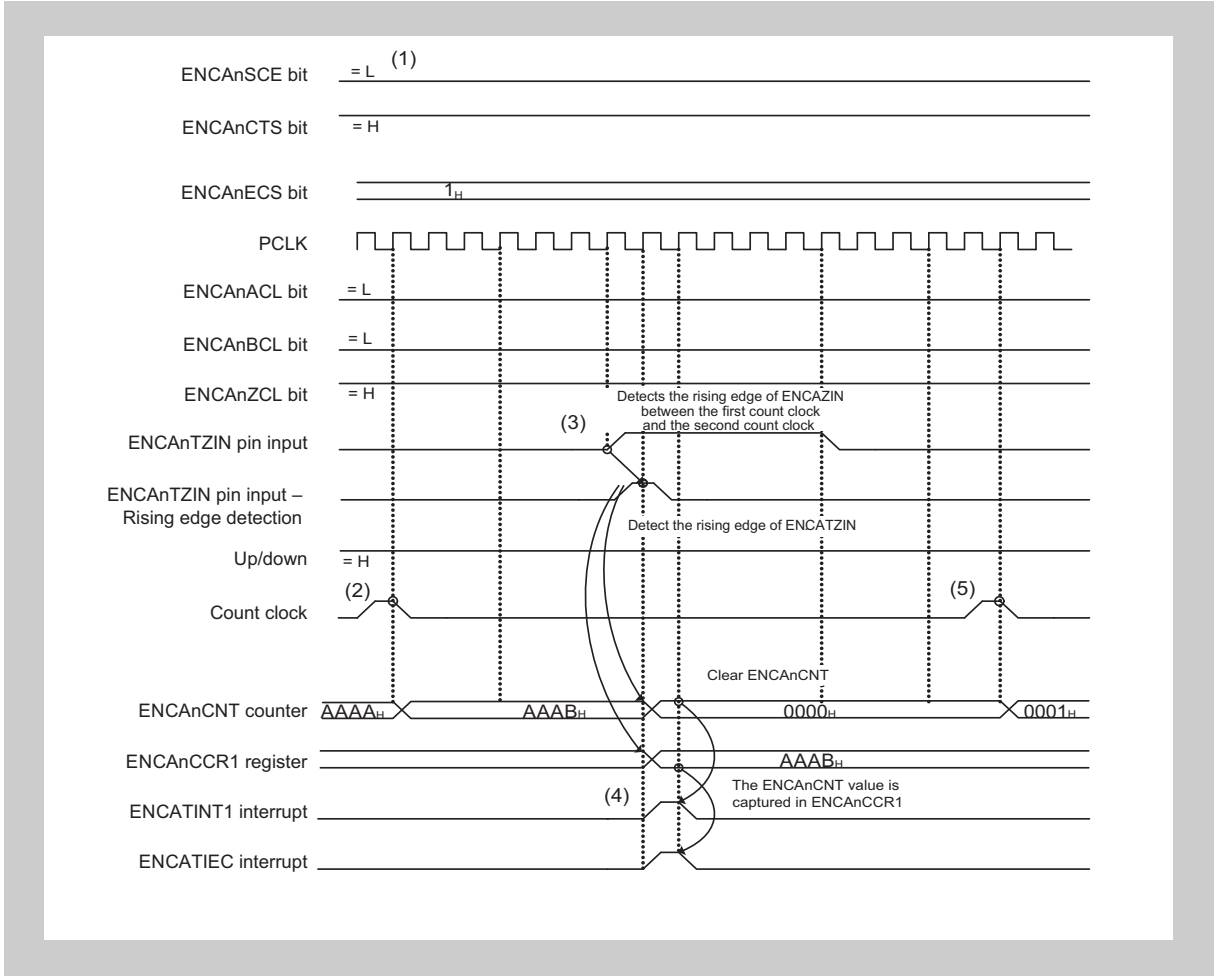


Figure 14-32 Capture Operation Performed upon Clearing by ENCATZIN when ENCAnSCE = 0

1. The values are set as follows: ENCAnSCE = 0, ENCAnCTS = 1, and ENCAnECS1 and ENCAnECS[1:0] = 01_B.
2. An up-count is performed.
3. The rising edge of the ENCATZIN input is detected, and the ENCAnCNT value ($AAAB_H$) is captured in the ENCAnCCR1 register. Concurrently, clear operation by ENCATZIN is performed, and ENCAnCNT is cleared to 0000_H.
4. A capture interrupt 1 (ENCATINT1) to the ENCAnCCR1 register and an encoder clear interrupt (ENCATIEC) by ENCATZIN are output.
5. After the count value is cleared, an up-count is performed, and the count value changes to 0001_H.

14.6.20 Match interrupt mask operation

(1) Mask operation and cancellation of masking

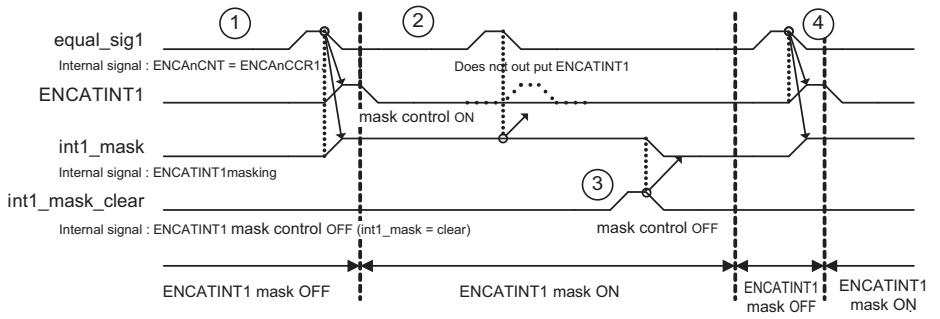


Figure 14-33 Match interrupt mask operation and cancellation of masking

Signals on the above timing chart are described below:

equal_sig1: Internal signal generated when ENCAnCNT matches with ENCAnCCR1

int1_mask: Interrupt detection mask signal. This signal is set when the first compare match occurs between ENCAnCNT and ENCAnCCR1.

int1_mask_clr: Interrupt detection mask cancellation signal. A trigger for mask cancellation differs depending on the ENCAnMCS setting.

- ENCAnMCS = 0: An interrupt detection mask is cancelled by writing to the ENCAnCCR1 register.
- ENCAnMCS = 1: An interrupt detection mask is cancelled by timer counter clearing operation when ENCAnCNT matches ENCAnCCR0, timer counter clearing operation by phase Z, or underflow detection when ENCAnLDE = 1.

1. The first compare match is detected when the match interrupt detection mask function is disabled.
   At this point, int1_mask, which masks detection of the second and subsequent compare match interrupts, is set to “1”, and the match interrupt detection mask function is enabled.
2. Because int1_mask is “High”, ENCATINT1 is not output even if equal_sig1 becomes “High”.
3. When a trigger which cancels a match interrupt detection mask occurs, int1_mask_clr is set to “1”, int1_mask is cleared to “L”, and the match interrupt detection mask function is disabled.
4. When the match interrupt detection mask function is disabled, the first compare match is detected, and concurrently, int1_mask is set to “1”.

(2) Mask operation and cancellation of masking (when ENCAnCNT matches with ENCAnCCR0)

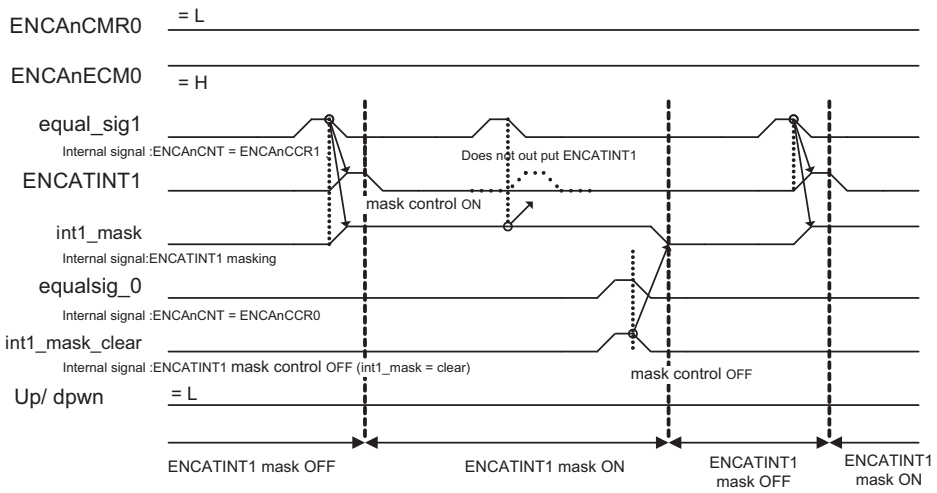


Figure 14-34 Match interrupt mask operation and cancellation of masking (when ENCAnCNT matches with ENCAnCCR0)

Signals on the above timing chart are described below:

equal_sig1:	Internal signal generated when ENCAnCNT matches with ENCAnCCR1
int1_mask:	Interrupt detection mask signal. This signal is set when the first compare match occurs between ENCAnCNT and ENCAnCCR1.
equal_sig0:	Internal signal generated when ENCAnCNT matches with ENCAnCCR0
int1_mask_clr:	Interrupt detection mask cancellation signal. A trigger for mask cancellation differs depending on the ENCAnMCS setting.
	ENCAnMCS = 0: An interrupt detection mask is cancelled by writing to the ENCAnCCR1 register.
	ENCAnMCS = 1: An interrupt detection mask is cancelled by timer counter clearing operation when ENCAnCNT matches ENCAnCCR0, timer counter clearing operation by phase Z, or underflow detection when ENCAnLDE = 1.
Up/down:	Internal signal whose count status is controlled by the UDS value of the phase-A input signal and phase-B input signal.

1. The first compare match is detected when the match interrupt detection mask function is disabled.
   At this point, int1_mask, which masks detection of the second and subsequent compare match interrupts, is set to “1”, and the match interrupt detection mask function is enabled.
2. Because int1_mask is “High”, ENCATINT1 is not output even if equal_sig1 becomes “High”.
3. By clear operation for when a compare match occurs between ENCAnCNT and ENCAnCCR0, int1_mask_clr is set to “1”, int1_mask is cleared to “L”, and the match interrupt detection mask function is disabled.
4. When the match interrupt detection mask function is disabled, the first compare match is detected, and concurrently, int1_mask is set to “1”.

Chapter 15 Timer Option Function (TAPA)

This chapter describes the Timer Option function (TAPA).

The first section describes all properties specific to the V850E2/MN4, such as instances, register base addresses, input/output signal names.
The subsequent sections describe the features that apply to all implementations.

15.1 Timer Option Function Features

Instances

This microcontroller has the following number of instances of TAPA.

Table 15-1 Instances of TAPA

	TAPA
Instance	4
Name	TAPA0 to TAPA3

Instances index n

Throughout this chapter, the individual instances of the TAPA are identified by the index “n” (n = 0 to 3), for example, TAPAnFLG for the TAPAn flag register.

Caution

The following bits of the registers are not supported in the V850E2/MN4.

- TAPA2CTL1.TAPA2ATS3 and TAPA2CTL1.TAPA2ATS2 bits
- TAPA3CTL1.TAPA3ATS3 and TAPA3CTL1.TAPA3ATS2 bits

Block diagram for motor control

The timer option function (TAPA), timer array unit A (TAUA), and/or peripheral interconnection (PIC) can be used as an inverter function that controls a motor.

When using this function, however, the available resources (channels) are limited as shown in *Figure 15-1 "Block diagram for motor control"* below.

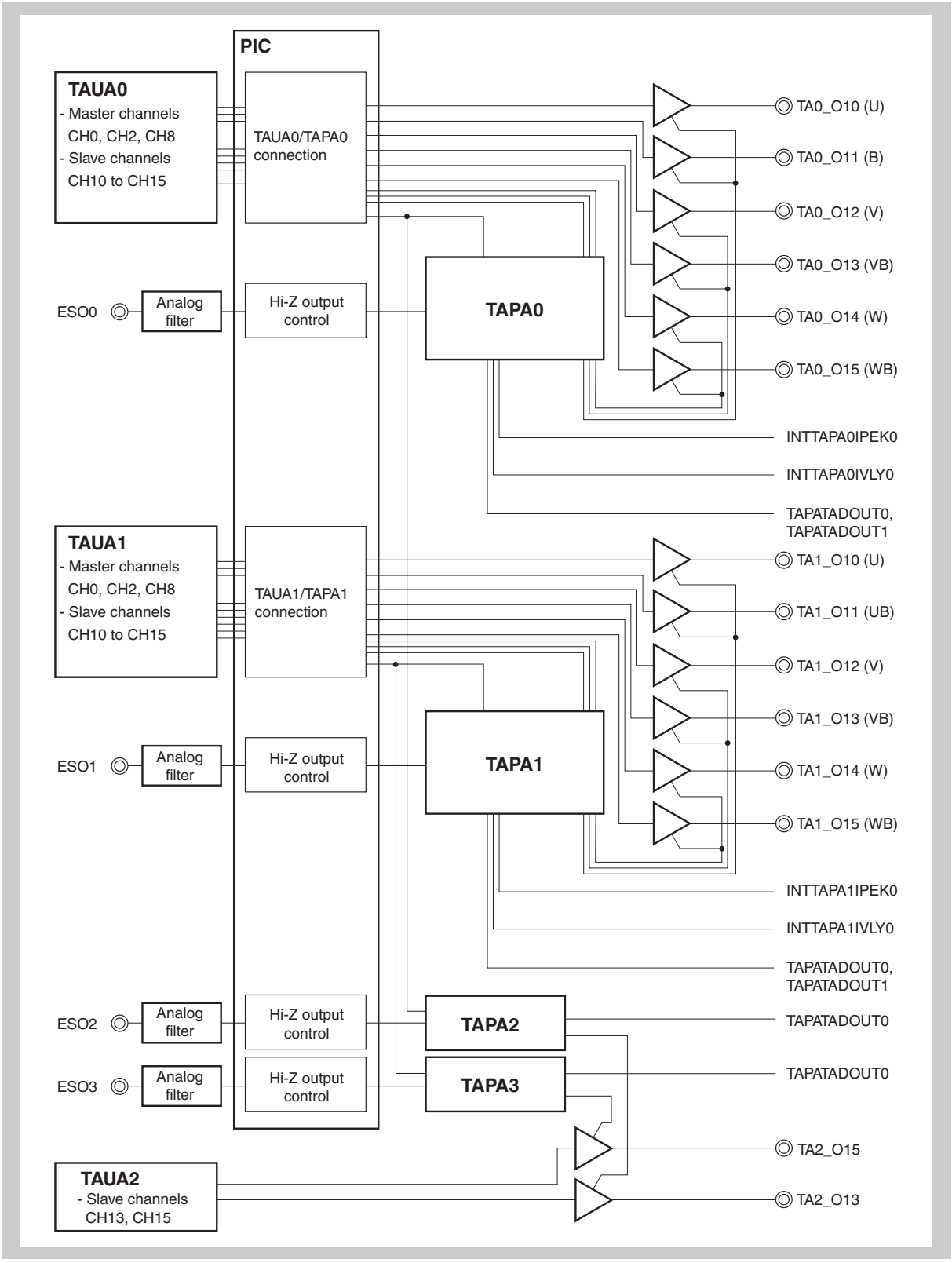


Figure 15-1 Block diagram for motor control

Register addresses

All TAPAn register addresses are given as address offsets from the individual base addresses <TAPAn_base_OS> or <TAPAn_base_USER>.
The register base addresses of each TAPAn are listed in the following table:

Table 15-2 Register base addresses <TAPAn_base>

TAPAn instance	Base address	Address
TAPA0	<TAPAn_base_OS>	FF81 5000$_H$
	<TAPAn_base_USER>	FFFF D400$_H$
TAPA1	<TAPAn_base_OS>	FF81 6000$_H$
	<TAPAn_base_USER>	FFFF D500$_H$
TAPA2	<TAPAn_base_OS>	FF81 7000$_H$
	<TAPAn_base_USER>	FFFF D600$_H$
TAPA3	<TAPAn_base_OS>	FF81 8000$_H$
	<TAPAn_base_USER>	FFFF D700$_H$

Clock supply

The following clock is supplied to TAPAn:

Table 15-3 TAPAn clock supply

TAPAn instance	Clock	Connected to:
TAPA0	PCLK	f_{PCLK}
TAPA1		
TAPA2		
TAPA3		

Interrupts and DMA/DTS

TAPA can generate the following interrupts and DMA/DTS requests:

Table 15-4 TAPAn interrupts and DMA/DTS requests

TAPAn signals	Function	Connected to:
TAPA0:		
TAPA0TIPEK0	TAPA0 peak interrupt	• Interrupt controller INTTAPA0IPEK0 • DTS controller trigger 98
TAPA0TIPEK1	TAPA0 peak interrupt	–
TAPA0TIVLY0	TAPA0 valley interrupt	• Interrupt controller INTTAPA0IVLY0 • DTS controller trigger 99
TAPA0TIVLY1	TAPA0 valley interrupt	–
TAPA1:		
TAPA1TIPEK0	TAPA1 peak interrupt	• Interrupt controller INTTAPA1IPEK0 • DTS controller trigger 103
TAPA1TIPEK1	TAPA1 peak interrupt	–
TAPA1TIVLY0	TAPA1 valley interrupt	• Interrupt controller INTTAPA1IVLY0 • DTS controller trigger 104
TAPA1TIVLY1	TAPA1 valley interrupt	–
TAPA2:		
TAPA2TIPEK0	TAPA2 peak interrupt	–
TAPA2TIPEK1	TAPA2 peak interrupt	–
TAPA2TIVLY0	TAPA2 valley interrupt	–
TAPA2TIVLY1	TAPA2 valley interrupt	–
TAPA3:		
TAPA3TIPEK0	TAPA3 peak interrupt	–
TAPA3TIPEK1	TAPA3 peak interrupt	–
TAPA3TIVLY0	TAPA3 valley interrupt	–
TAPA3TIVLY1	TAPA3 valley interrupt	–

Internal signals The internal signals of TAPAn are listed in the table below.

Table 15-5 TAPA internal signals (1/2)

TAPAn signals	Function	Connected to:
TAPA0:		
TAPATHASIN	Asynchronous Hi-Z input signal	PIC (TOP0TAPATHASIN)
TAPATSIM0	TAUA master 0 INT input	PIC (TOP0TAPATSIM0)
TAPATSIM1	TAUA master 1 INT input	–
TAPATCDENS0	TAUA slave 0 match detection input	PIC (ADOPA1ADCATTIN00)
TAPATCDENS1	TAUA slave 1 match detection input	PIC (ADOPA2ADCATTIN00)
TAPATADOUT0	A/D conversion trigger output 0	ADCA0 (ADCA0TTIN102)
TAPATADOUT1	A/D conversion trigger output 1	ADCA0 (ADCA0TTIN202)
TAPAUDCM0	TAUA master 0 up/down input	PIC (TOP0TAPATUDCM0)
TAPAUDCM1	TAUA master 1 up/down input	–
TAPATCDENM0	TAUA master 0 cycle detection input	–
TAPATCDENM1	TAUA master 1 cycle detection input	–
TAPATTOEM0	TAUA master 0 timer enable input	–
TAPATTOEM1	TAUA master 1 timer enable input	–
TAPA1:		
TAPATHASIN	Asynchronous Hi-Z input signal	PIC (TOP1TAPATHASIN)
TAPATSIM0	TAUA master 0 INT input	PIC (TOP1TAPATSIM0)
TAPATSIM1	TAUA master 1 INT input	–
TAPATCDENS0	TAUA slave 0 match detection input	PIC (ADOPA1ADCATTIN01)
TAPATCDENS1	TAUA slave 1 match detection input	PIC (ADOPA2ADCATTIN01)
TAPATADOUT0	A/D conversion trigger output 0	ADCA0 (ADCA0TTIN103)
TAPATADOUT1	A/D conversion trigger output 1	ADCA0 (ADCA0TTIN203)
TAPAUDCM0	TAUA master 0 up/down input	PIC (TOP1TAPATUDCM0)
TAPAUDCM1	TAUA master 1 up/down input	–
TAPATCDENM0	TAUA master 0 cycle detection input	–
TAPATCDENM1	TAUA master 1 cycle detection input	–
TAPATTOEM0	TAUA master 0 timer enable input	–
TAPATTOEM1	TAUA master 1 timer enable input	–

Table 15-5 TAPA internal signals (2/2)

TAPAn signals	Function	Connected to:
TAPA2:		
TAPATHASIN	Asynchronous Hi-Z input signal	PIC (TOP2TAPATHASIN)
TAPATSIM0	TAUA master 0 INT input	PIC (TOP0TAPATSIM0)
TAPATSIM1	TAUA master 1 INT input	–
TAPATCDENS0	TAUA slave 0 match detection input	PIC (ADOPA0ADCATTIN00)
TAPATCDENS1	TAUA slave 1 match detection input	–
TAPATADOUT0	A/D conversion trigger output 0	ADCA0 (ADCA0TTIN002)
TAPATADOUT1	A/D conversion trigger output 1	–
TAPAUDCM0	TAUA master 0 up/down input	PIC (TOP0TAPATUDCM0)
TAPAUDCM1	TAUA master 1 up/down input	–
TAPATCDENM0	TAUA master 0 cycle detection input	–
TAPATCDENM1	TAUA master 1 cycle detection input	–
TAPATTOEM0	TAUA master 0 timer enable input	–
TAPATTOEM1	TAUA master 1 timer enable input	–
TAPA3:		
TAPATHASIN	Asynchronous Hi-Z input signal	PIC (TOP3TAPATHASIN)
TAPATSIM0	TAUA master 0 INT input	PIC (TOP1TAPATSIM0)
TAPATSIM1	TAUA master 1 INT input	–
TAPATCDENS0	TAUA slave 0 match detection input	PIC (ADOPA0ADCATTIN01)
TAPATCDENS1	TAUA slave 1 match detection input	–
TAPATADOUT0	A/D conversion trigger output 0	ADCA0 (ADCA0TTIN003)
TAPATADOUT1	A/D conversion trigger output 1	–
TAPAUDCM0	TAUA master 0 up/down input	PIC (TOP1TAPATUDCM0)
TAPAUDCM1	TAUA master 1 up/down input	–
TAPATCDENM0	TAUA master 0 cycle detection input	–
TAPATCDENM1	TAUA master 1 cycle detection input	–
TAPATTOEM0	TAUA master 0 timer enable input	–
TAPATTOEM1	TAUA master 1 timer enable input	–

I/O signals The I/O signals of TAPAn are listed in the table below.

Table 15-6 TAPAn I/O signals

TAPAn signals	Function	Connected to:
TAPA0	Hi-Z control signal 0 (phase U)	Hi-Z control TAUA0 phase U output (TAUA0O10/TAUA0O11)
	Hi-Z control signal 1 (phase V)	Hi-Z control TAUA phase V output (TAUA0O12/TAUA0O13)
	Hi-Z control signal 2 (phase W)	Hi-Z control TAUA0 phase W output (TAUA0O14/TAUA0O15)
TAPA1	Hi-Z control signal 0 (phase U)	Hi-Z control TAUA1 phase U output (TAUA1O10/TAUA1O11)
	Hi-Z control signal 1 (phase V)	Hi-Z control TAUA1 phase V output (TAUA1O12/TAUA1O13)
	Hi-Z control signal 2 (phase W)	Hi-Z control TAUA1 phase W output (TAUA1O14/TAUA1O15)
TAPA2	Hi-Z control signal 0 (phase U)	Hi-Z control TAUA2 output (TAUA2O13)
	Hi-Z control signal 1 (phase V)	–
	Hi-Z control signal 2 (phase W)	–
TAPA3	Hi-Z control signal 0 (phase U)	Hi-Z control TAUA2 output (TAUA2O15)
	Hi-Z control signal 1 (phase V)	–
	Hi-Z control signal 2 (phase W)	–

15.1.1 Block diagram

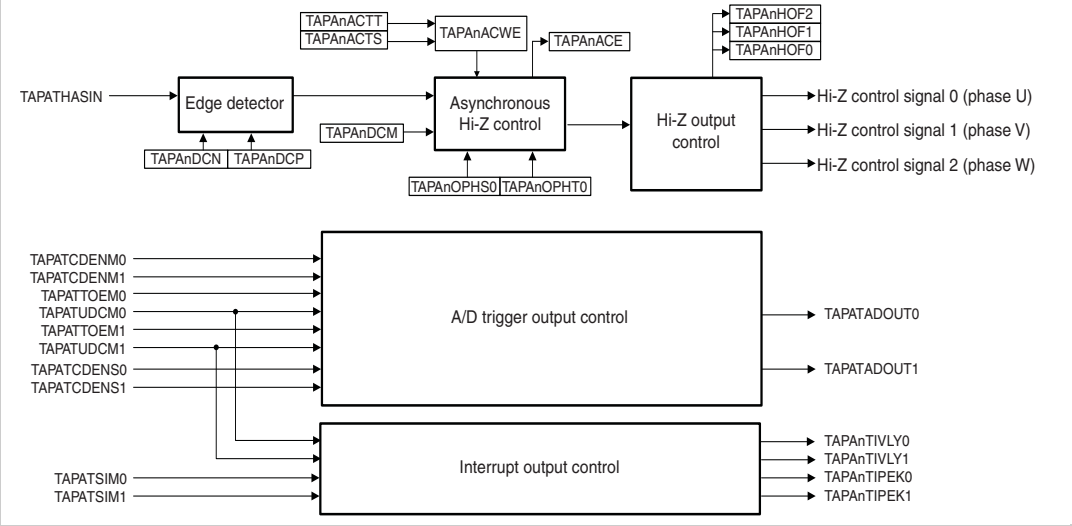


Figure 15-2 TAPA block diagram

15.2 Functional Overview

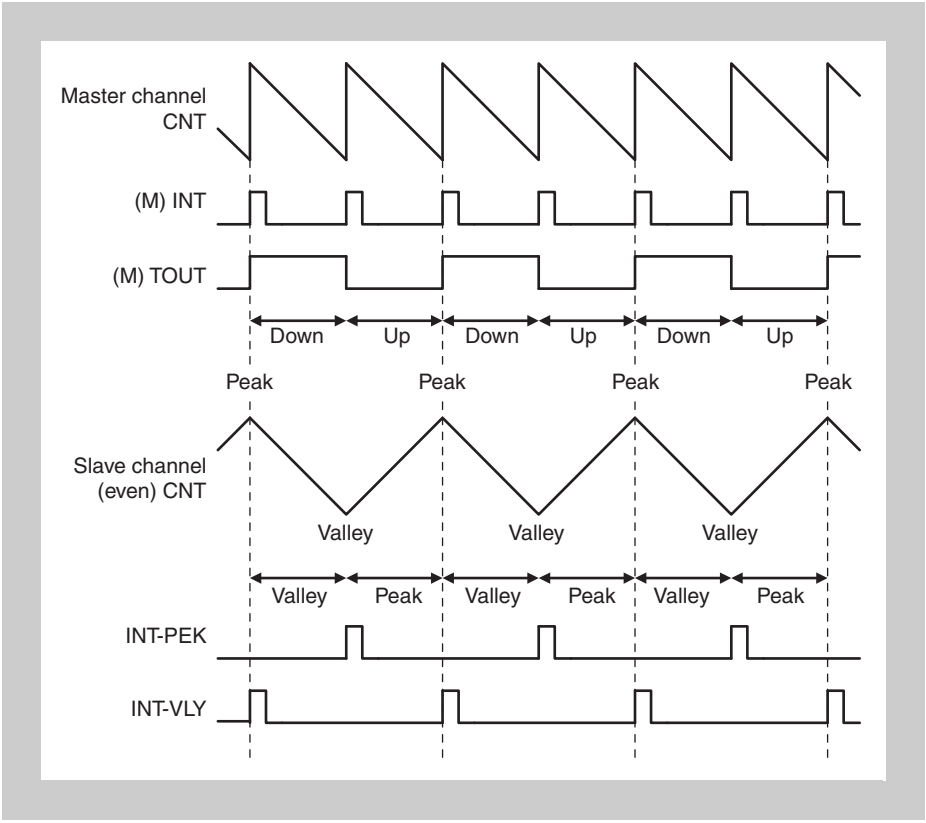
Function overview

The timer motor control function (TAPA) is to be connected to timer array unit A (TAUA) to implement motor systems.

- Asynchronous Hi-Z control functions
  Timer output for motor control can be forcibly stopped by asynchronously setting its level to high-impedance.
- INT signal output selection function
  Two types of interrupts, peak interrupts and valley interrupts, can be output based on the INTn signal output by the TAUA.
- A/D conversion trigger selection function
  Two A/D conversion trigger signals can be output based on the INTn signal output by the TAUA.

15.2.1 Peak and valley interrupts - Peak and valley of timer counter

In this document, the period from a TAUA up status (counting-up status) to generation of INT from the master channel is defined as a peak period, and this INT is defined as a peak interrupt. In contrast, the period from a TAUA down status (counting-down status) to generation of INT from the master channel is defined as a valley period, and this INT is defined as a valley interrupt.



15.3 Registers

The timer motor control function is controlled and operated by means of the following registers.

15.3.1 Registers overview

Table 15-7 Control registers overview

Register name	Shortcut	Address
Control register 0	TAPAnCTL0	<TAPAn_base_OS> + 20_H
Control register 1	TAPAnCTL1	<TAPAn_base_OS> + 24_H
Flag register	TAPAnFLG	<TAPAn_base_USER> + 00_H
Asynchronous Hi-Z control write enable register	TAPAnACWE	<TAPAn_base_USER> + 04_H
Asynchronous Hi-Z control start trigger register	TAPAnACTS	<TAPAn_base_USER> + 08_H
Asynchronous Hi-Z control stop trigger register	TAPAnACTT	<TAPAn_base_USER> + $0C_H$
Hi-Z start trigger register	TAPAnOPHS	<TAPAn_base_USER> + 14_H
Hi-Z stop trigger register	TAPAnOPHT	<TAPAn_base_USER> + 18_H

<TAPAn_base> The base addresses <TAPAn_base_OS> and <TAPAn_base_USER> of the TAPAn is defined in the first section of this chapter under the key word “Register addresses”.

15.3.2 Registers details

(1) TAPAnCTL0 - Control register 0

This register is used to set up the asynchronous Hi-Z control function.

The values of this register can be rewritten only when TAPAnFLG.TAPAnASCE is 0 and the TAUAnTE bit for the corresponding TAUA's master channel is 0.

Access This register can be read or written in 16-bit units.

Address <TAPAn_base_OS> + 20_H

Initial value 0000_H. This register is initialized by any reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	TAPAnDCM	TAPAnDCN	TAPAnDCP	0	0
R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R	R

Table 15-8 TAPAnCTL0 register contents

Bit position	Bit name	Function
4	TAPAnDCM	Clear condition configuration bit This control bit specifies the clear conditions for Hi-Z control output. 0: Enables manipulation of TAPAnOPHT0, regardless of the TAPATHASIN signal input level. 1: Disables manipulation of TAPAnOPHT0 if the TAPATHASIN signal input is active. Enables manipulation of TAPAnOPHT0 if the TAPATHASIN signal input is inactive.
3, 2	TAPAnDCN, TAPAnDCP	Hi-Z input edge selection bits These are control bits that specify the valid edge of TAPATHASIN.

TAPAnDCN	TAPAnDCP	Description
0	0	Does not detect valid edges.
0	1	Detects a rising edge as the valid edge (active level = high).
1	0	Detects a falling edge as the valid edge (active level = low).
1	1	Setting prohibited

(2) TAPAnCTL1 - Control register 1

This register is used to specify the A/D conversion trigger.

Access This register can be read or written in 8-bit units.

Address <TAPAn_base_OS> + 24_H

Initial value 00_H. This register is initialized by any reset.

7	6	5	4	3	2	1	0
0	0	0	0	TAPAn ATS3	TAPAn ATS2	TAPAn ATS1	TAPAn ATS0
R	R	R	R	R/W	R/W	R/W	R/W

Table 15-9 TAPAnCTL1 register contents

Bit position	Bit name	Function
3:2	TAPAn ATS[3:2]	A/D conversion trigger 1 selection bits These are control bits that specify the A/D conversion trigger output 1 (TAPATADOUT1).
1:0	TAPAn ATS[1:0]	A/D conversion trigger 0 selection bits These are control bits that specify the A/D conversion trigger output 0 (TAPATADOUT0).

TAPAnATS3	TAPAnATS2	Description
0	0	INT signal output while the triangle wave is falling (counting down)
0	1	INT signal output while the triangle wave is rising (counting up)
1	0	INT signal output while the triangle wave is rising (counting up) or falling (counting down)
1	1	INT signal and valley interrupt INTTAPAnIVLY1 output while the triangle wave is rising (counting up) or falling (counting down)

TAPAnATS1	TAPAnATS0	Description
0	0	INT signal output while the triangle wave is falling (counting down)
0	1	INT signal output while the triangle wave is rising (counting up)
1	0	INT signal output while the triangle wave is rising (counting up) or falling (counting down)
1	1	INT signal and valley interrupt INTTAPAnIVLY0 output while the triangle wave is rising (counting up) or falling (counting down)

(3) TAPAnFLG - Flag register

This flag register is for asynchronous Hi-Z control.

Access This register can be read in 16-bit units.

Address <TAPAn_base_USER> + 00_H

Initial value 0700_H. This register is initialized by any reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	TAPAnHOF2	TAPAnHOF1	TAPAnHOF0	0	0	0	0	0	0	0	TAPAnACE
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 15-10 TAPAnFLG register contents

Bit position	Bit name	Function
10:8	TAPAnHOFm	Hi-Z control monitor bits (m = 0, 1, 2) These bits are used to monitor the Hi-Z control status. 0: Hi-Z is not being controlled. 1: Hi-Z is being controlled.
0	TAPAnACE	Asynchronous Hi-Z control enable bit This bit indicates the status of the asynchronous Hi-Z control signal TAPATHASIN. 0: Indicates that the asynchronous Hi-Z control is stopped. 1: Indicates that the asynchronous Hi-Z control is enabled. The conditions for setting or clearing this bit are as follows: Clear condition: Writing 1 to TAPAnACTT while TAPAnACWE = 1 Set condition: Writing 1 to TAPAnACTS while TAPAnACWE = 1

(4) TAPAnACWE – Asynchronous Hi-Z control write enable register

This register is used to enable writing for asynchronous Hi-Z control.

Access This register can be read or written in 8-bit units.

Address <TAPAn_base_USER> + 04_H

Initial value 00_H. This register is initialized by any reset.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	TAPAnACWE
R	R	R	R	R	R	R	R/W

Table 15-11 TAPAnACWE register contents

Bit position	Bit name	Function
0	TAPAnACWE	Asynchronous control write enable bit This is a write enable bit for asynchronous Hi-Z control. After 1 is written, this bit is automatically cleared to 0 by writing 1 to TAPAnACTS and TAPAnACTT. 0: Disables writing to TAPAnACTS and TAPAnACTT. 1: Enables writing to TAPAnACTS and TAPAnACTT.

(5) TAPAnACTS - Asynchronous Hi-Z control start trigger register

This register is used to enable the start trigger for asynchronous Hi-Z control.

Access This register is write-only, in 8-bit units. This register is always read as 00_H.

Address <TAPAn_base_USER> + 08_H

Initial value 00_H. This register is initialized by any reset.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	TAPAn ACTS
R	R	R	R	R	R	R	W

Table 15-12 TAPAnACTS register contents

Bit position	Bit name	Function
0	TAPAnACTS	Asynchronous Hi-Z control start trigger bit This bit enables the start trigger for asynchronous Hi-Z control. The setting of this bit is valid only when TAPAnACWE = 1. 0: Writing 0 to this bit is ignored (no function). 1: Enables asynchronous Hi-Z control and then sets TAPAnACE to 1.

(6) TAPAnACTT - Asynchronous Hi-Z control stop trigger register

This bit enables the stop trigger for asynchronous Hi-Z control.

Access This register is write-only, in 8-bit units. This register is always read as 00_H.

Address <TAPAn_base_USER> + $0C_H$

Initial value 00_H. This register is initialized by any reset.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	TAPAn ACTT
R	R	R	R	R	R	R	W

Table 15-13 TAPAnACTT register contents

Bit position	Bit name	Function
0	TAPAnACTT	Asynchronous Hi-Z control stop trigger bit This bit enables the stop trigger for asynchronous Hi-Z control. The setting of this bit is valid only when TAPAnACWE = 1. 0: Writing 0 to this bit is ignored (no function). 1: Disables asynchronous Hi-Z control and then clears TAPAnACE to 0.

(7) TAPAnOPHS - Hi-Z start trigger register

This software trigger register is used to start Hi-Z control for timer output pins.

Access This register is write-only, in 8-bit units. This register is always read as 00_H.

Address <TAPAn_base_USER> + 14_H

Initial value 00_H. This register is initialized by any reset.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	TAPAn OPHS0
R	R	R	R	R	R	R	W

Table 15-14 TAPAnOPHS register contents

Bit position	Bit name	Function
0	TAPAnOPHS0	Hi-Z control start trigger bit This bit starts Hi-Z control for timer output pins. 0: Writing 0 to this bit is ignored (no function). 1: Starts Hi-Z control.

(8) TAPAnOPHT - Hi-Z stop trigger register

This software trigger register is used to stop Hi-Z control for timer output pins.

Access This register is write-only, in 8-bit units. This register is always read as 00_H.

Address <TAPAn_base_USER> + 18_H

Initial value 00_H. This register is initialized by any reset.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	TAPAn OPHT0
R	R	R	R	R	R	R	W

Table 15-15 TAPAnOPHT register contents

Bit position	Bit name	Function
0	TAPAnOPHT0	Hi-Z control stop trigger bit This bit stops Hi-Z control for timer output pins. 0: Writing 0 to this bit is ignored (no function). 1: Stops Hi-Z control. Whether the setting of this bit is valid or invalid depends on the setting of TAPAnCTL0.TAPAnDCM.

15.4 Basic Functions

15.4.1 Asynchronous Hi-Z control function

(1) Purpose

If the operation of the timer motor control function controlled by the CPU becomes abnormal, the rotation of the external motor also becomes abnormal. This function can forcibly set the motor control output to the Hi-Z state upon detection of abnormal motor operation, independent of the CPU control.

(2) Overview

This function forcibly stops TAPAn output through asynchronous Hi-Z control.

When the TAPATHASIN signal becomes active, the levels of the timer output pins are set to Hi-Z, and timer output is forcibly stopped.

If TAPAnCTL0.TAPAnDCM is 0, setting the TAPAnACTT.TAPAnOPHT0 bit stops Hi-Z output control regardless of the TAPATHASIN signal input level.

If TAPAnCTL0.TAPAnDCM is 1, setting the TAPAnACTT.TAPAnOPHT0 bit while the TAPATHASIN signal input level is inactive stops Hi-Z output control.

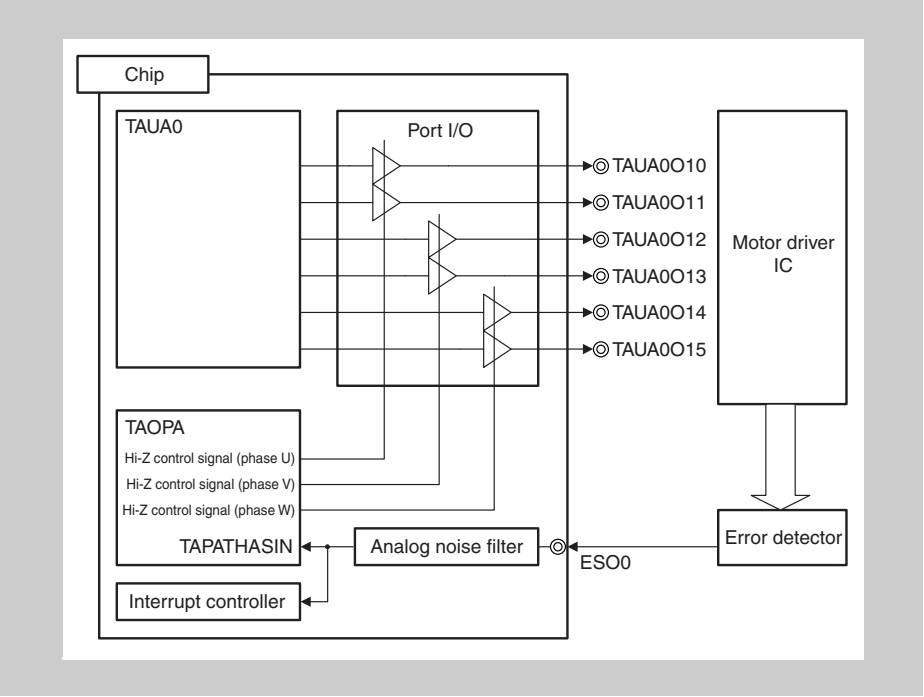
(3) System configuration example

A system configuration example is shown below, where an external error detection signal (the ESO0 pin) is used for the Hi-Z control of the TAUA0 outputs (the TAUA0O10 to TAUA0O15 pins).

When an external error detection signal is received, an interrupt is generated and the level of the motor control timer outputs (TAUA0 channels 10 to 15) is simultaneously set to Hi-Z.

Because the microcontroller might freeze when an error occurs, external error detection signals are continuously processed so that the motor control timer outputs can be set to Hi-Z even if no clock is supplied.

Note that an error is detected only when the valid edge of the error detection signal is detected. Therefore, no error is detected if the output level is fixed and the signal level does not change.


Chip
TAUA0
Port I/O
TAUA0O10
TAUA0O11
TAUA0O12
TAUA0O13
TAUA0O14
TAUA0O15
Motor driver IC
TAOPA
Hi-Z control signal (phase U)
Hi-Z control signal (phase V)
Hi-Z control signal (phase W)
TAPATHASIN
Analog noise filter
ESO0
Error detector
Interrupt controller

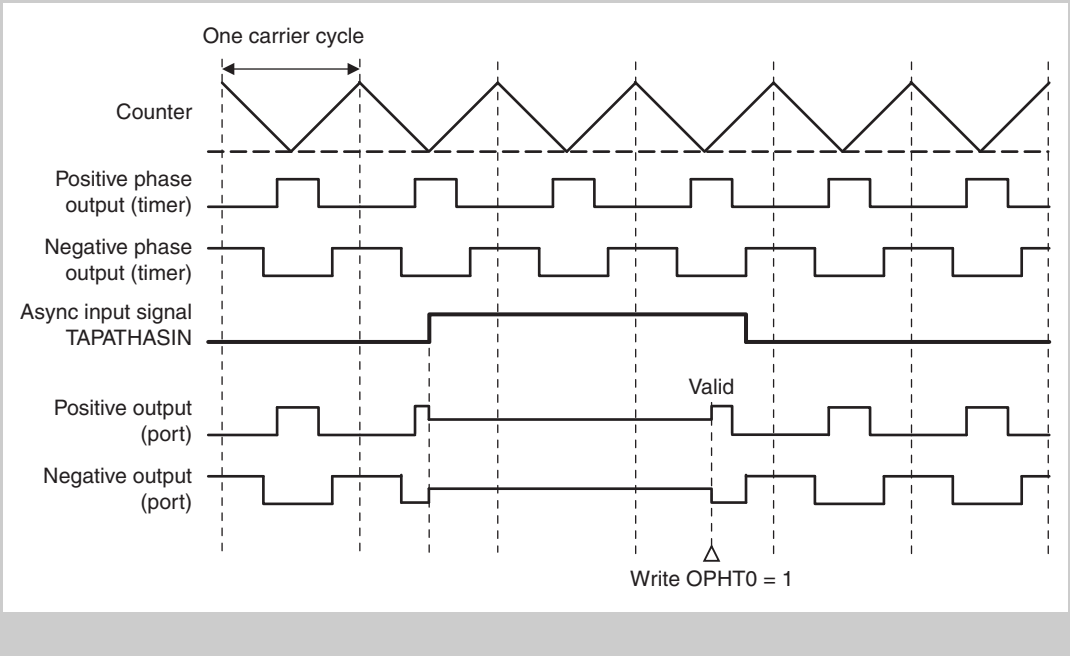
(4) Basic operation

Hi-Z control for timer output pins can be started as follows:

- Detecting the valid edge of TAPATHASIN
- Setting the start trigger bit TAPAnOPHS0 of the Hi-Z control signal

The levels of the timer output pins are set to Hi-Z until the stop trigger bit TAPAnOPHT0 of the Hi-Z control signal is set. Note that whether the setting of TAPAnOPTH0 is valid or invalid depends on the setting of TAPAnCTL0.TAPAnDCM.

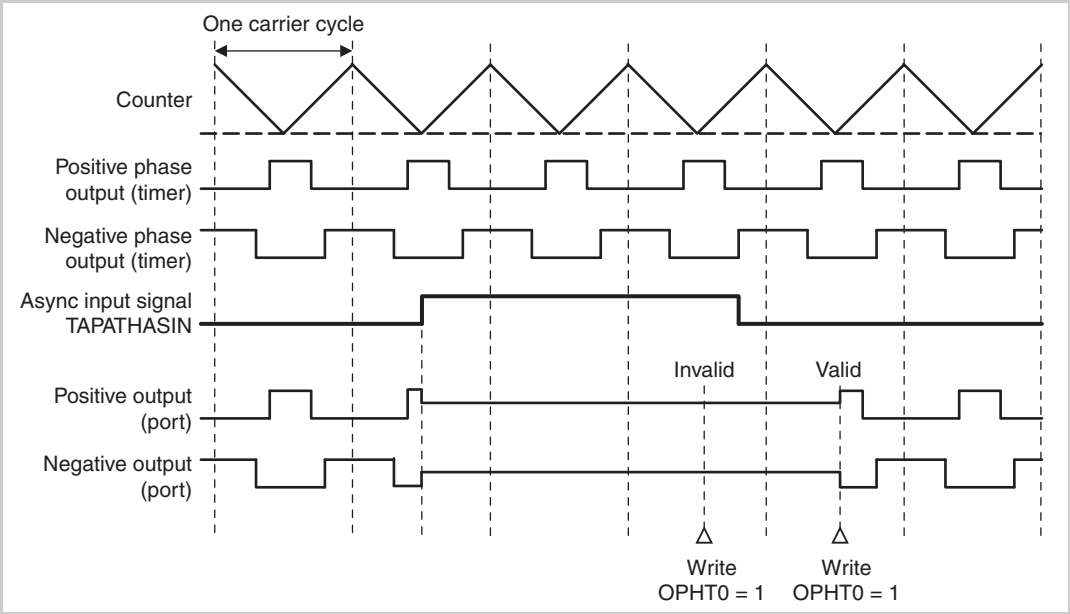
(a) Operation when TAPAnCTL0.TAPAnDCM = 0, TAPAnDCP = 0, and TAPAnDCN = 0



The timer outputs are forcibly stopped (Hi-Z output by port control) when the valid edge of the asynchronous input signal TAPATHASIN is detected.

The timer outputs restart when 1 is written to the Hi-Z stop trigger bit TAPAnOPHT0, regardless of the level of TAPATHASIN.

(b) Operation when TAPAnCTL0.TAPAnDCM = 1, TAPAnDCP = 1, and TAPAnDCN = 0



The timer outputs are forcibly stopped (Hi-Z output by port control) when the valid edge of asynchronous input signal TAPATHASIN is detected.

Writing 1 to the Hi-Z stop trigger bit TAPAnOPHT0 is ignored while the asynchronous input signal TAPATHASIN is active (high level because TAPAnDCP is 1).

The timer outputs restart when 1 is written to the Hi-Z stop trigger bit TAPAnOPHT0 after TAPATHASIN becomes inactive (low level because TAPAnDCP is 1).

(5) Asynchronous Hi-z control using software trigger

TAPA supports Hi-Z control using a software trigger.

Hi-Z control for timer outputs is possible by using the Hi-Z control start trigger bit TAPAnOPHS0 and Hi-Z control stop trigger bit TAPAnOPHT0.

(a) Function of Hi-Z control start trigger bit TAPAnOPHS0

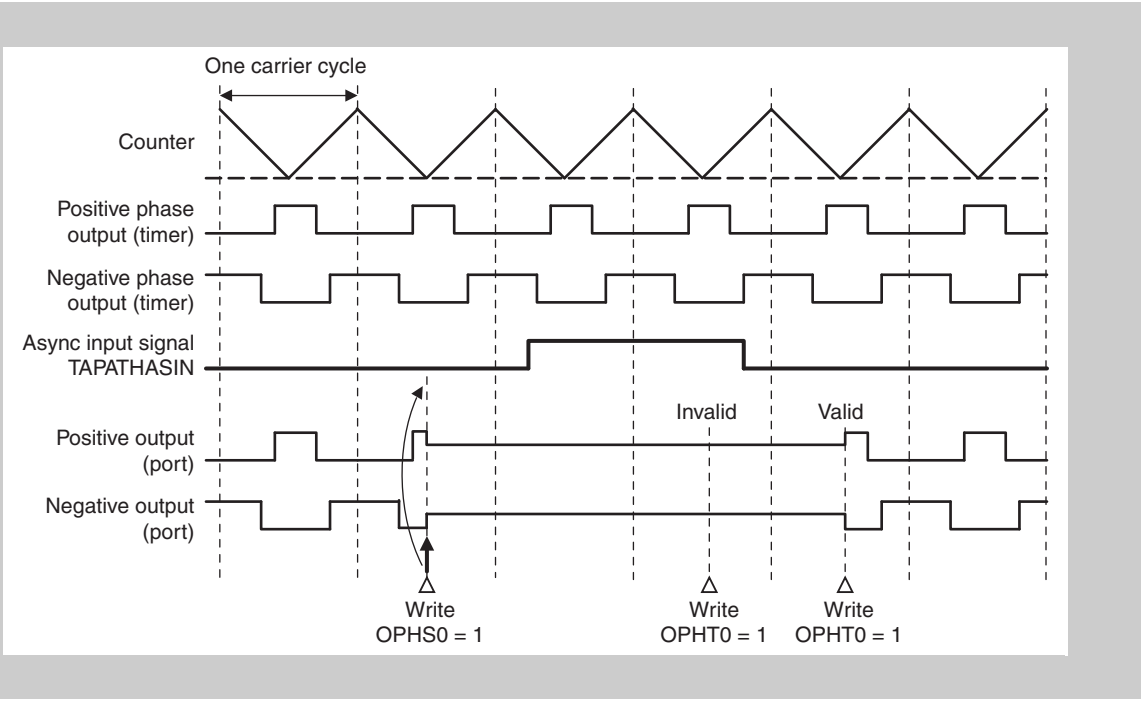
TAPAnDCM	Function
0/1	Writing 1 to TAPAnOPHS0 starts Hi-Z control and forcibly stops the timer outputs (Hi-Z output by port control).

(b) Function of Hi-Z control stop trigger bit TAPAnOPHT0

Whether the Hi-Z control stop trigger is valid or invalid depends on the conditions below:

TAPAnDCM	Function
0	Writing 1 to TAPAnOPHT0 stops Hi-Z control and restarts timer outputs.
1	If TAPATHASIN is inactive, writing 1 to TAPAnOPHT0 stops Hi-Z control and restarts timer outputs. If TAPATHASIN is active, writing 1 to TAPAnOPHT0 is ignored.

(c) Operation when TAPAnCTL0.TAPAnDCM = 1, TAPAnDCP = 1, and TAPAnDCN = 0



The timer outputs are forcibly stopped (Hi-Z output by port control) when 1 is written to TAPAnOPHS0.

After that, the levels of the timer outputs remain Hi-Z even if a rising edge of TAPATHASIN is detected.

Writing to TAPAnOPHT0 is ignored while TAPATHASIN is active (high level because TAPAnDCN is 0 and TAPAnDCP is 1).

After detection of a falling edge of TAPATHASIN, the timer outputs restart when 1 is written to TAPAnOPHT0 while TAPATHASIN is inactive (low level because TAPAnDCN is 0 and TAPAnDCP is 1).

(6) Operating procedure

The operating procedure for the asynchronous input Hi-Z control function is shown below:

	Operation	Status of TAPA
Initial setup	Set up the TAPAnCTL0 register. Specify TAPAnDCP and TAPAnDCN to select the input edge. Specify TAPAnDCM to select the clear mode.	Asynchronous Hi-Z control stopped (TAPAnFLG.TAPAnACE = 0)
Start operation	Set up the TAPAnACWE register. Set TAPAnACWE to 1. Set up the TAPAnACTS register. Set TAPAnACTS to 1.	Writing to TAPAnACTS is enabled. Asynchronous Hi-Z control enabled (TAPAnFLG.TAPAnACE = 1)
During operation	Hi-Z control for the timer function outputs can be started by controlling the following: • TAPAnOPHS register • Hi-Z control asynchronous input signal TAPATHASIN Hi-Z control for the timer function outputs can be stopped by controlling the following: • TAPAnOPHT register (If TAPAnDCM is 1, control by the TAPAnOPHT register is enabled only while TAPATHASIN is inactive.) The TAPA operating status can always be read using the TAPAnFLG register.	Hi-Z control for the timer output pins is started by detecting the valid edge of the Hi-Z control asynchronous input signal (TAPATHASIN) or by setting the start trigger bit TAPAnOPHS0. Hi-Z control for the timer output pins is stopped by setting the stop trigger bit TAPAnOPHT0 according to the operation mode specified by TAPAnCTL0.TAPAnDCM.
Stop operation	Set up the TAPAnACWE register. Set TAPAnACWE to 1. Set up the TAPAnACTT register. Set TAPAnACTT to 1.	Writing to TAPAnACTT is enabled. Asynchronous Hi-Z control stopped (TAPAnFLG.TAPAnACE = 0)

Restart

15.4.2 INT signal output selection function

(1) Configuration of the INT signal output selection function

This function generates the peak interrupt INTTAPAnIPEKm and valley interrupt INTTAPAnIVLYm by using the TAPATSIMm signal, which is connected to the INT signal on the TAUA's triangular carrier cycle generation channel (master).

For the connection destination of TAPATSIMm, see the table TAPAn internal signals in the first section of this chapter.

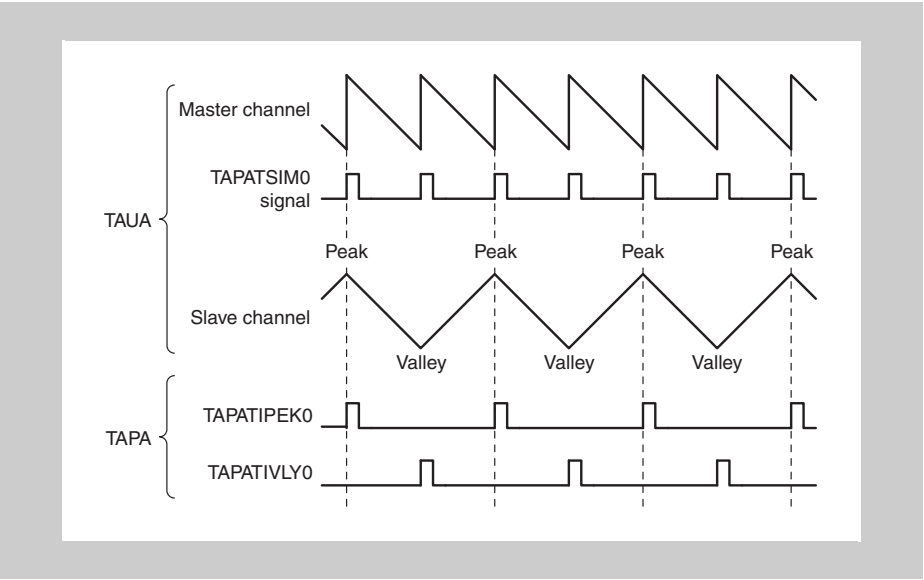


Figure 15-3 Basic timing chart of signals for the INT signal output selection function

Triangular carrier cycles are generated on the master channel.

An interrupt generated on the master channel in each half triangular carrier cycle is connected to TAPATSIMm, and then input to TAPAn. From TAPATSIMm, the peak interrupt INTTAPAnIPEKm is generated at the peak of the triangle wave, and the valley interrupt INTTAPAnIVLYm is generated at the valley of the triangle wave.

Caution INTTAPAnIPEKm and INTTAPAnIVLYm are generated regardless of the TAUA mode.

When not using these peak and valley interrupts, mask them by using the ICTAPAnIPEKm and ICTAPAnIVLYm registers, respectively.

15.4.3 A/D conversion trigger selection function

This function outputs the A/D conversion trigger signals TAPATADOUT0 and TAPATADOUT1 from the signals TAPATCDENS0 and TAPATCDENS1, which are connected to a compare match interrupt with TAUA's triangular carrier cycle.

(1) Configuration of A/D conversion trigger selection function

Table 15-16 Signals used for TAPATADOUT generation

Output signal	Slave match detection signal	Valley interrupt signal
TAPATADOUT0	TAPATCDENS0	TAPATIVLY0
TAPATADOUT1	TAPATCDENS1	TAPATIVLY0

Table 15-17 Operation of TAPATADOUT0 according to the setting of TAPAnCTL1.TAPAnATS[3:0]

TAPAn ATS3	TAPAn ATS2	Description
0	0	Outputs the INT signal while the triangle wave is falling (counting down).
0	1	Outputs the INT signal while the triangle wave is rising (counting up).
1	0	Outputs the INT signal while the triangle wave is rising (counting up) or falling (counting down).
1	1	Outputs the INT signal and valley interrupt INTTAPAnIVLY1 while the triangle wave is rising (counting up) or falling (counting down).

TAPAn ATS1	TAPAn ATS0	Description
0	0	Outputs the INT signal while the triangle wave is falling (counting down).
0	1	Outputs the INT signal while the triangle wave is rising (counting up).
1	0	Outputs the INT signal while the triangle wave is rising (counting up) or falling (counting down).
1	1	Outputs the INT signal and valley interrupt INTTAPAnIVLY0 while the triangle wave is rising (counting up) or falling (counting down).

(2) Waveforms of A/D conversion trigger output control operation in triangle PWM mode

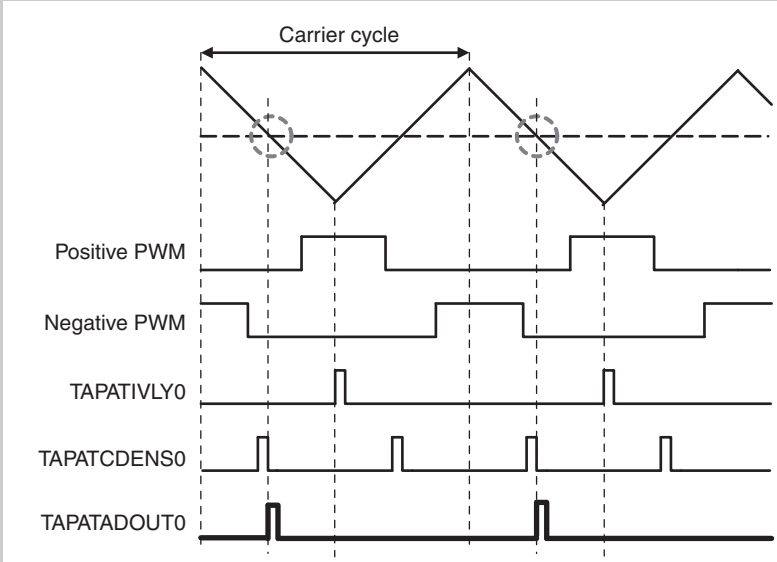


Figure 15-4 TAPAnATS[1:0] = 00B: Output of INT signal while the triangle wave is falling (counting down)

While the triangle wave is falling (counting down), the signals TAPATCDNS0 and TAPATCDNS1 are output as the A/D conversion trigger signals TAPATADOUT0 and TAPATADOUT1.

In this case, no A/D conversion trigger signal is output while the triangle wave is rising (counting up).

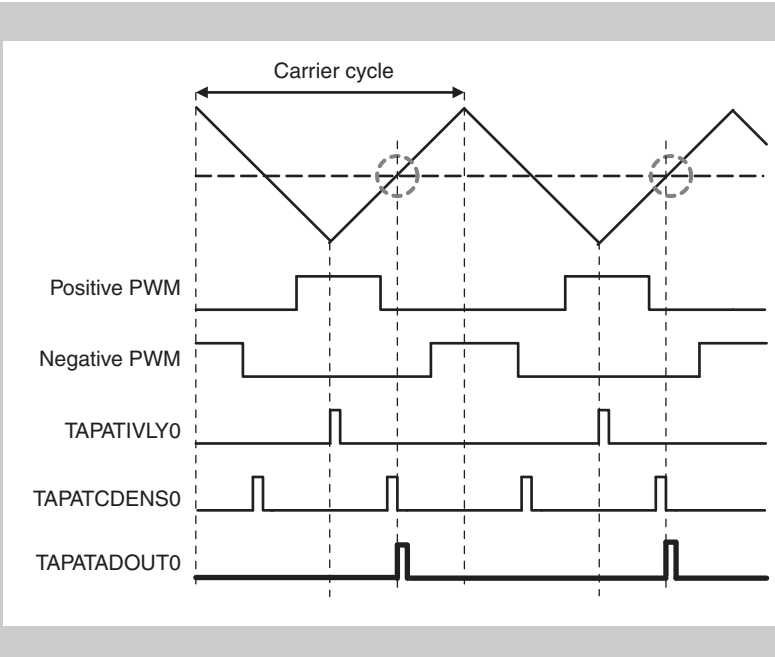


Figure 15-5 TAPAnATS[1:0] = 01B: Output of INT signal while the triangle wave is rising (counting up)

While the triangle wave is rising (counting up), the TAPATCDNS0 and TAPATCDNS1 signals are output as A/D conversion trigger signals TAPATADOUT0 and TAPATADOUT1.

In this case, no A/D conversion trigger signal is output while the triangle wave is falling (counting down).

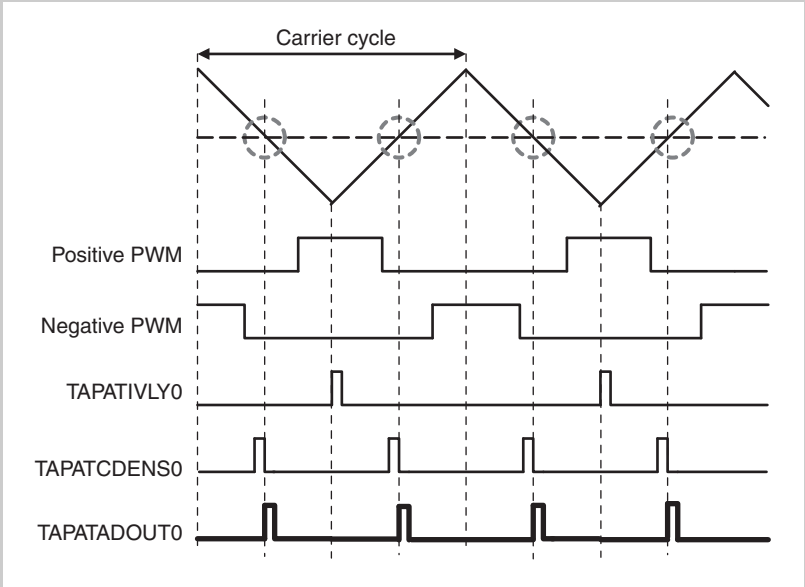


Figure 15-6 TAPAnATS[1:0] = 10B: Output of INT signal while the triangle wave is rising (counting up) or falling (counting down)

The signals TAPATCDNS0 and TAPATCDNS1 are output as the A/D conversion trigger signals TAPATADOUT0 and TAPATADOUT1.

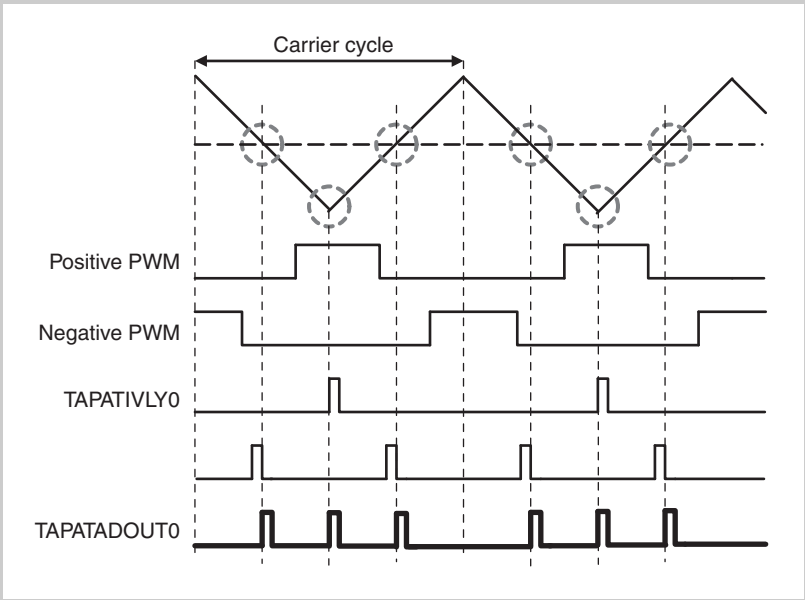


Figure 15-7 TAPAnATS[1:0] = 11B: Output of INT signal and valley interrupt while the triangle wave is rising (counting up) or falling (counting down)

The signals TAPATCDNS0 and TAPATCDNS1 and valley interrupt INTTAPAnIVLY0 are output as the A/D conversion trigger signals TAPATADOUT0 and TAPATADOUT1.

(3) Operating procedure for A/D conversion trigger selection function

The operating procedure for the A/D conversion trigger selection function is shown below.

	Operation	Status of TAUA and TAPA
Initial setup	Initialize TAUA. Specify the timer operation mode. Set up the TAPAnCTL1 register. Specify TAPAnATS[1:0] (TAPATADOUT0 setting). Specify TAPAnATS[3:2] (TAPATADOUT1 setting).	TAUA and TAPA stop the operation.
Start operation	Start the TAUA operation.	TAUA starts count operation.
During operation	TAUA operates according to the setting of each function.	The A/D conversion trigger selection function outputs either TAPATADOUT0 according to the setting of TAPAnATS[1:0] or TAPATADOUT1 according to the setting of TAPAnATS[3:2], based on the interrupt TAPATCDENS1 or TAPATCDENS0, which is input from TAUA, and the valley interrupt TAPATIVLY1 or TAPATIVLY0, which is generated by TAPA.
Stop operation	Stop the TAUA operation.	TAUA stops the count operation.

Restart

Chapter 16 Peripheral Interconnection (PIC)

The PIC function implements a timer simultaneous start function, motor control function, and A/D trigger selection function by connecting timers and A/D converters.

16.1 V850E2/MN4 PIC Features

Instances

This microcontroller has the following number of instances of the PIC.

Table 16-1 Instance

PIC	
Instances	1
Name	PIC0

Register addresses

All PIC0 register addresses are given as address offsets to the individual base address <PIC0_base_USER> or <PIC0_base_OS>.
The base addresses <PIC0_base_USER> and <PIC0_base_OS> of PIC0 are listed in the following table.

Table 16-2 Register base addresses

PIC0	Base address	Address
PIC0	<PIC0_base_USER>	FFFF DB00$_H$
	<PIC0_base_OS>	FF81 C000$_H$

Clock supply

PIC provides one clock input.

Table 16-3 PIC0 clock supply

PIC0	PIC0 clock	Connected to:
PIC0	PCLK	f_{PCLK}

I/O signals

The I/O signals connecting PIC with other functions are listed in the following table.

Table 16-4 I/O signals for PIC0 (1/2)

Signal name	I/O	Function	Connected to:
TOP0TAPATSIM0	O	TAUA master 0 INT input	TAPA0 TAPA2
TOP0TAPATUDCM0	O	TAUA master 0 up/down input	
TOP0TAPATHASIN	O	Asynchronous Hi-Z input signal	TAPA0
TOP1TAPATSIM0	O	TAUA master 1 INT input	TAPA1 TAPA3
TOP1TAPATUDCM0	O	TAUA master 1 up/down input	
TOP1TAPATHASIN	O	Asynchronous Hi-Z input signal	TAPA1
TOP2TAPATHASIN	O	Asynchronous Hi-Z input signal	TAPA2

Table 16-4 I/O signals for PIC0 (2/2)

Signal name	I/O	Function	Connected to:
TOP3TAPATHASIN	O	Asynchronous Hi-Z input signal	TAPA3
TAUA0TSST[15:0]	O	Simultaneous channel count start trigger input [15:0]	TAUA0
INTTAUA0I[15:0]	I	Cycle detection output	
TAUA0TTOUT[15:0]	I	Channel output	
TAUA0UDC[8,2,0]	I	Up/down output	TAUA0 (TAUA0TTOUT[8, 2, 0])
TAUA1TSST[15:0]	O	Simultaneous channel count start trigger input [15:0]	TAUA1
INTTAUA1I[15:0]	I	Cycle detection output	
TAUA1TTOUT[15:0]	I	Channel output	
TAUA1UDC[8, 2, 0]	I	Up/down output	TAUA1 (TAUA1TTOUT[8, 2, 0])
TAUJ0TSST[3:0]	O	Simultaneous channel count start trigger input [3:0]	TAUJ0
TAUJ1TSST0	O	Simultaneous channel count start trigger input 0	TAUA2 (TAUA2TSST12)
TAUJ1TSST1	O	Simultaneous channel count start trigger input 1	TAUA2 (TAUA2TSST13)
TAUJ1TSST2	O	Simultaneous channel count start trigger input 2	TAUA2 (TAUA2TSST14)
TAUJ1TSST3	O	Simultaneous channel count start trigger input 3	TAUA2 (TAUA2TSST15)
ADOPA0ADCATTIN00	O	A/D converter 0 hardware trigger input 0	ADCA0,TAPA2
ADOPA0ADCATTIN01	O	A/D converter 0 hardware trigger input 1	ADCA0,TAPA3
ADOPA1ADCATTIN00	O	A/D converter 1 hardware trigger input 0	ADCA0,TAPA0
ADOPA1ADCATTIN01	O	A/D converter 1 hardware trigger input 1	ADCA0,TAPA1
ADOPA2ADCATTIN00	O	A/D converter 2 hardware trigger input 0	ADCA0,TAPA0
ADOPA2ADCATTIN01	O	A/D converter 2 hardware trigger input 1	ADCA0,TAPA1
ANFESO0	I	ESO0 pin input	Pin
ANFESO1	I	ESO1 pin input	Pin
ANFESO2	I	ESO2 pin input	Pin
ANFESO3	I	ESO3 pin input	Pin
ENCA0TSST	O	Synchronous start trigger input	ENCA0
ENCA1TSST	O	Synchronous start trigger input	ENCA1
OST0TSST	O	Synchronous start trigger input	OSTM0
OST1TSST	O	Synchronous start trigger input	OSTM1

16.2 Functional Overview

Features summary

The peripheral interconnection (PIC) function connects the internal I/O signals of the timers and A/D converters in the V850E2/MN4 to multiple macros. PIC is provided with the following functions:

- Inter-timer connection
  - Timer synchronous operation
- Connection to timer motor control function
- Connection between A/D converters and timers
  - A/D trigger selection
- Connection between CAN controllers and timers
  - CAN time stamp function

16.2.1 Block diagrams

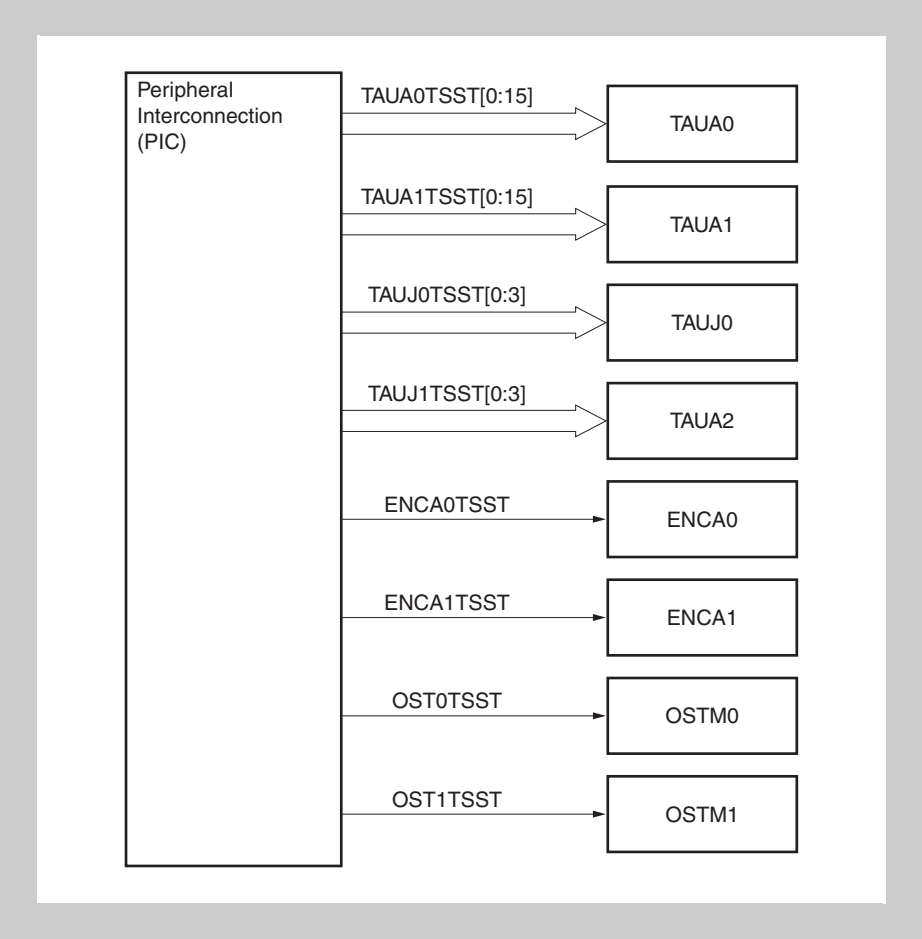


Figure 16-1 Block diagram of timer simultaneous start function

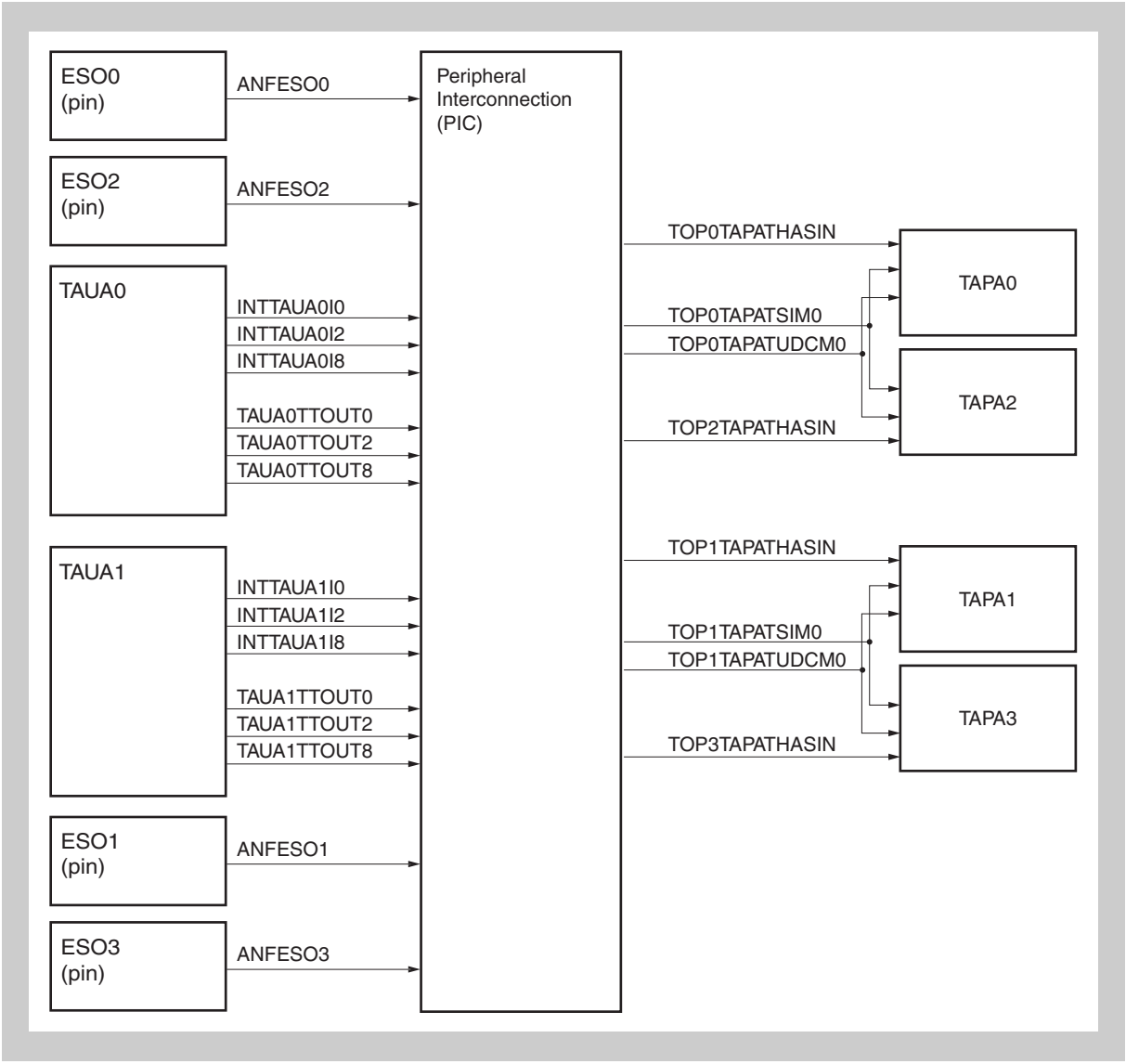


Figure 16-2 Block diagram of TAUA and TAPA connection

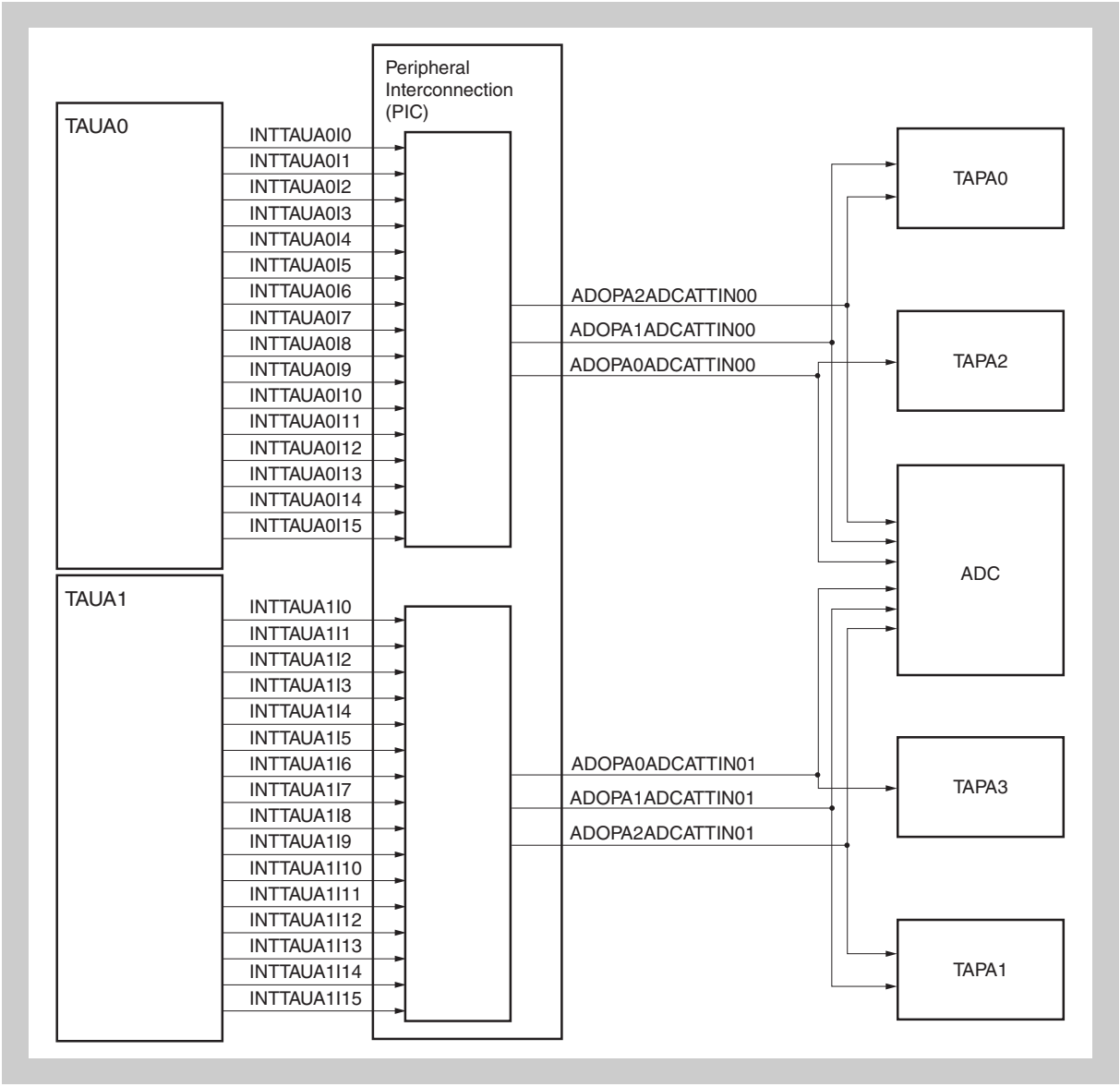


Figure 16-3 Block diagram of TAUA and A/D converter connection

16.3 Control Registers

The PIC is controlled and operated by means of the following registers:

Table 16-5 PIC registers

Register function	Name	Address
Simultaneous start trigger control register	PIC0SST	<PIC0_base_USER> + 04_H
Simultaneous start control register 0	PIC0SSER0	<PIC0_base_OS> + 80_H
Simultaneous start control register 1	PIC0SSER1	<PIC0_base_OS> + 84_H
Simultaneous start control register 2	PIC0SSER2	<PIC0_base_OS> + 88_H
PIC register 200	PIC0REG200	<PIC0_base_OS> + $8C_H$
PIC register 210	PIC0REG210	<PIC0_base_OS> + $9C_H$
Hi-Z output control register 0	PIC0HIZCEN0	<PIC0_base_OS> + $B4_H$
Hi-Z output control register 1	PIC0HIZCEN1	<PIC0_base_OS> + $B8_H$
Hi-Z output control register 2	PIC0HIZCEN2	<PIC0_base_OS> + DC_H
Hi-Z output control register 3	PIC0HIZCEN3	<PIC0_base_OS> + $E4_H$
A/D converter trigger output control register 400	PIC0ADTEN400	<PIC0_base_OS> + $C4_H$
A/D converter trigger output control register 401	PIC0ADTEN401	<PIC0_base_OS> + $C8_H$
A/D converter trigger output control register 402	PIC0ADTEN402	<PIC0_base_OS> + CC_H
A/D converter trigger output control register 410	PIC0ADTEN410	<PIC0_base_OS> + $D0_H$
A/D converter trigger output control register 411	PIC0ADTEN411	<PIC0_base_OS> + $D4_H$
A/D converter trigger output control register 412	PIC0ADTEN412	<PIC0_base_OS> + $D8_H$
PIC register 31	PIC0REG31	<PIC0_base_OS> + $C0_H$

16.3.1 Registers used to set up timer simultaneous start function

(1) PIC0SST - Simultaneous start trigger control register

Access This register can be read/written in 8 and 1-bit units.

Address <PIC0_base_USER> + 04_H

Default value 00_H

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	SYNCTRG
R	R	R	R	R	R	R	R/W

Table 16-6 PIC0SST register contents

Bit position	Bit name	Functions
0	SYNCTRG	Specify whether to generate a start trigger for the timer for which simultaneous start is enabled. 0: Disable 1: Generate a simultaneous start trigger (output of a 1-PCLK-width pulse)

Note The SYNCTRG bit is always read as 0.

(2) PIC0SSER0 - Simultaneous start control register 0

This register is used to enable or disable simultaneous start trigger output for each channel of TAUA0.

Access This register can be read or written in 16-bit units.

Address <PIC0_base_OS> + 80_H

Default value 0000_H

15	14	13	12	11	10	9	8
PIC0SSER015	PIC0SSER014	PIC0SSER013	PIC0SSER012	PIC0SSER011	PIC0SSER010	PIC0SSER009	PIC0SSER008
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7	6	5	4	3	2	1	0
PIC0SSER007	PIC0SSER006	PIC0SSER005	PIC0SSER004	PIC0SSER003	PIC0SSER002	PIC0SSER001	PIC0SSER000
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 16-7 PIC0SSER0 register contents

Bit position	Bit name	Function
m	PIC0SSER0m	Enable or disable simultaneous start trigger output for CHm of TAUA0. 0: Disable simultaneous start trigger output for CHm of TAUA0. 1: Enable simultaneous start trigger output for CHm of TAUA0.

(3) PIC0SSER1 - Simultaneous start control register 1

This register is used to enable or disable simultaneous start trigger output for each channel of TAUA1.

Access This register can be read or written in 16-bit units.

Address <PIC0_base_OS> + 84_H

Default value 0000_H

15	14	13	12	11	10	9	8
PIC0SSER115	PIC0SSER114	PIC0SSER113	PIC0SSER112	PIC0SSER111	PIC0SSER110	PIC0SSER109	PIC0SSER108
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7	6	5	4	3	2	1	0
PIC0SSER107	PIC0SSER106	PIC0SSER105	PIC0SSER104	PIC0SSER103	PIC0SSER102	PIC0SSER101	PIC0SSER100
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 16-8 PIC0SSER1 register contents

Bit position	Bit name	Function
m	PIC0SSER1m	Enable or disable simultaneous start trigger output for CHm of TAUA1. 0: Disable simultaneous start trigger output for CHm of TAUA1. 1: Enable simultaneous start trigger output for CHm of TAUA1.

(4) PIC0SSER 2 - Simultaneous start control register 2

This register is used to enable or disable simultaneous start trigger output for each channel of ENC, the OS timer, TAUJ1, and TAUJ0.

Access This register can be read or written in 16-bit units.

Address <PIC0_base_OS> + 88_H

Default value 0000_H

15	14	13	12	11	10	9	8
PIC0SSER115	PIC0SSER114	PIC0SSER113	PIC0SSER112	0	0	0	0
R/W	R/W	R/W	R/W	R	R	R	R

7	6	5	4	3	2	1	0
PIC0SSER107	PIC0SSER106	PIC0SSER105	PIC0SSER104	PIC0SSER103	PIC0SSER102	PIC0SSER101	PIC0SSER100
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 16-9 PIC0SSER2 register contents (1/2)

Bit position	Bit name	Function
15	PIC0SSER215	Enable or disable simultaneous start trigger output for ENCA1. 0: Disable simultaneous start trigger output for ENCA1. 1: Enable simultaneous start trigger output for ENCA1.
14	PIC0SSER214	Enable or disable simultaneous start trigger output for ENCA0. 0: Disable simultaneous start trigger output for ENCA0. 1: Enable simultaneous start trigger output for ENCA0.
13	PIC0SSER213	Enable or disable simultaneous start trigger output for OSTM1. 0: Disable simultaneous start trigger output for OSTM1. 1: Enable simultaneous start trigger output for OSTM1.
12	PIC0SSER212	Enable or disable simultaneous start trigger output for OSTM0. 0: Disable simultaneous start trigger output for OSTM0. 1: Enable simultaneous start trigger output for OSTM0.
7	PIC0SSER207	Enable or disable simultaneous start trigger output for CH15 of TAUA2. 0: Disable simultaneous start trigger output for CH15 of TAUA2. 1: Enable simultaneous start trigger output for CH15 of TAUA2.
6	PIC0SSER206	Enable or disable simultaneous start trigger output for CH14 of TAUA2. 0: Disable simultaneous start trigger output for CH14 of TAUA2. 1: Enable simultaneous start trigger for output CH14 of TAUA2.
5	PIC0SSER205	Enable or disable simultaneous start trigger output for CH13 of TAUA2. 0: Disable simultaneous start trigger output for CH13 of TAUA2. 1: Enable simultaneous start trigger output for CH13 of TAUA2.
4	PIC0SSER204	Enable or disable simultaneous start trigger output for CH12 of TAUA2. 0: Disable simultaneous start trigger output for CH12 of TAUA2. 1: Enable simultaneous start trigger output for CH12 of TAUA2.
3	PIC0SSER203	Enable or disable simultaneous start trigger output for CH3 of TAUJ0. 0: Disable simultaneous start trigger output for CH3 of TAUJ0. 1: Enable simultaneous start trigger output for CH3 of TAUJ0.

Table 16-9 PIC0SSER2 register contents (2/2)

Bit position	Bit name	Function
2	PIC0SSER202	Enable or disable simultaneous start trigger output for CH2 of TAUJ0. 0: Disable simultaneous start trigger output for CH2 of TAUJ0. 1: Enable simultaneous start trigger output for CH2 of TAUJ0.
1	PIC0SSER201	Enable or disable simultaneous start trigger output for CH1 of TAUJ0. 0: Disable simultaneous start trigger output for CH1 of TAUJ0. 1: Enable simultaneous start trigger output for CH1 of TAUJ0.
0	PIC0SSER200	Enable or disable simultaneous start trigger output for CH0 of TAUJ0. 0: Disable simultaneous start trigger output for CH0 of TAUJ0. 1: Enable simultaneous start trigger output for CH0 of TAUJ0.

16.3.2 Registers used to make connections between TAUA and TAPA

(1) PIC0REG2n0 - PIC register 2n0 (n = 0 or 1)

Access This register can be read or written in 32-bit units.

Address PIC0REG200: <PIC0_base_OS> + $8C_H$
PIC0REG210: <PIC0_base_OS> + $9C_H$

Default value $0000\ 0000_H$

31	30	29	28	27	26	25	24
0	0	0	0	0	0	PIC0REG 2n025	PIC0REG 2n024
R	R	R	R	R	R	R/W	R/W

23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R

Table 16-10 PIC0REG2n0 register contents

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>25:24</td><td>PIC0REG2n025
PIC0REG2n024</td><td>Select the signals to input to TOPnTAPATSIM0 and TAPnTAPATUDCM0.
<table>
<tr><th>PIC0REG 2n025</th><th>PIC0REG 2n024</th><th>Input signal</th></tr>
<tr><td>0</td><td>0</td><td>Low level</td></tr>
<tr><td>0</td><td>1</td><td>If CH0 of TAUA is specified as the master channel by the motor control function:
TOPnTAPATSIM0 = INTTAUAnI0
TAPnTAPATUDCM0 = TAUAnUDC0</td></tr>
<tr><td>1</td><td>0</td><td>If CH2 of TAUA is specified as the master channel by the motor control function:
TOPnTAPATSIM0 = INTTAUAnI2
TAPnTAPATUDCM0 = TAUAnUDC2</td></tr>
<tr><td>1</td><td>1</td><td>If CH8 of TAUA is specified as the master channel by the motor control function:
TOPnTAPATSIM0 = INTTAUAnI8
TAPnTAPATUDCM0 = TAUAnUDC8</td></tr>
</table>
</td></tr>
</table>

(2) PIC0HIZCENn - Hi-Z output control register n (n = 0 to 3)

This register is used to select the Hi-Z output control input signal of TAPAn.

Access This register can be read or written in 8-bit units.

Address PIC0HIZCEN0: <PIC0_base_OS> + $B4_H$
PIC0HIZCEN1: <PIC0_base_OS> + $B8_H$
PIC0HIZCEN2: <PIC0_base_OS> + DC_H
PIC0HIZCEN3: <PIC0_base_OS> + $E4_H$

Default value 00_H. This register is initialized by any reset.

Caution The settings of this register must be specified with “No edge detection” specified as the valid edge of the Hi-Z request signal of TAPAn.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	PIC0HIZCENn0
R	R	R	R	R	R	R	R/W

Table 16-11 PIC0HIZCENn register contents

Bit position	Bit name	Function
0	PIC0HIZCENn0	Select whether to enable or disable Hi-Z output control by the TAPAnESO pin input. 0: Disable 1: Enable

16.3.3 Registers used to make connections between TAUA and the A/D converter

(1) PIC0ADTEN40m - A/D converter trigger output control register 40m

This register is used to enable or disable selection of a signal from TAUA0 as the trigger source for channel group CGm (m = 0 to 2) of ADCA0.

Access This register can be read or written in 16-bit units.

Address PIC0ADTEN400: <PIC0_base_OS> + $C4_H$
PIC0ADTEN401: <PIC0_base_OS> + $C8_H$
PIC0ADTEN402: <PIC0_base_OS> + CC_H

Default value 0000_H

15	14	13	12	11	10	9	8
PIC0ADTEN 40m15	PIC0ADTEN 40m14	PIC0ADTEN 40m13	PIC0ADTEN 40m12	PIC0ADTEN 40m11	PIC0ADTEN 40m10	PIC0ADTEN 40m09	PIC0ADTEN 40m08
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7	6	5	4	3	2	1	0
PIC0ADTEN 40m07	PIC0ADTEN 40m06	PIC0ADTEN 40m05	PIC0ADTEN 40m04	PIC0ADTEN 40m03	PIC0ADTEN 40m02	PIC0ADTEN 40m01	PIC0ADTEN 40m00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 16-12 PIC0ADTEN40m register contents

Bit position	Bit name	Function
15:0	PIC0ADTEN40m [15:00]	Enable or disable a trigger from TAUA0 as the trigger source for channel group m of ADCA0. 0: Disable selection of the INTTAUA0Ix interrupt signal as a hardware trigger. 1: Enable selection of the INTTAUA0Ix interrupt signal as a hardware trigger.

(2) PIC0ADTEN41m - A/D converter trigger output control register 41m

This register is used to enable or disable selection of a signal from TAUA1 as the trigger source for channel group CGm (m = 0 to 2) of ADCA0.

Access This register can be read or written in 16-bit units.

Address PIC0ADTEN410: <PIC0_base_OS> + $D0_H$
PIC0ADTEN411: <PIC0_base_OS> + $D4_H$
PIC0ADTEN412: <PIC0_base_OS> + $D8_H$

Default value 0000_H

15	14	13	12	11	10	9	8
PIC0ADTEN 41m15	PIC0ADTEN 41m14	PIC0ADTEN 41m13	PIC0ADTEN 41m12	PIC0ADTEN 41m11	PIC0ADTEN 41m10	PIC0ADTEN 41m09	PIC0ADTEN 41m08
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

7	6	5	4	3	2	1	0
PIC0ADTEN 41m07	PIC0ADTEN 41m06	PIC0ADTEN 41m05	PIC0ADTEN 41m04	PIC0ADTEN 41m03	PIC0ADTEN 41m02	PIC0ADTEN 41m01	PIC0ADTEN 41m00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 16-13 PIC0ADTEN41m register contents

Bit position	Bit name	Function
15:0	PIC0ADTEN41m [15:00]	Enable or disable a trigger from TAUA1 as the trigger source for channel group m of ADCA0. 0: Disable selection of the INTTAUA1Ix interrupt signal as a hardware trigger. 1: Enable selection of the INTTAUA1Ix interrupt signal as a hardware trigger.

16.4 Functions

16.4.1 Timer simultaneous start function

This function generates a start trigger input signal, which is used to operate TAUA0, TAUA1, TAUA2, TAUJ0, ENCA0, ENCA1, OSTM0, and OSTM1 synchronously.

For details about the start trigger input, see the description about the trigger input for each macro.

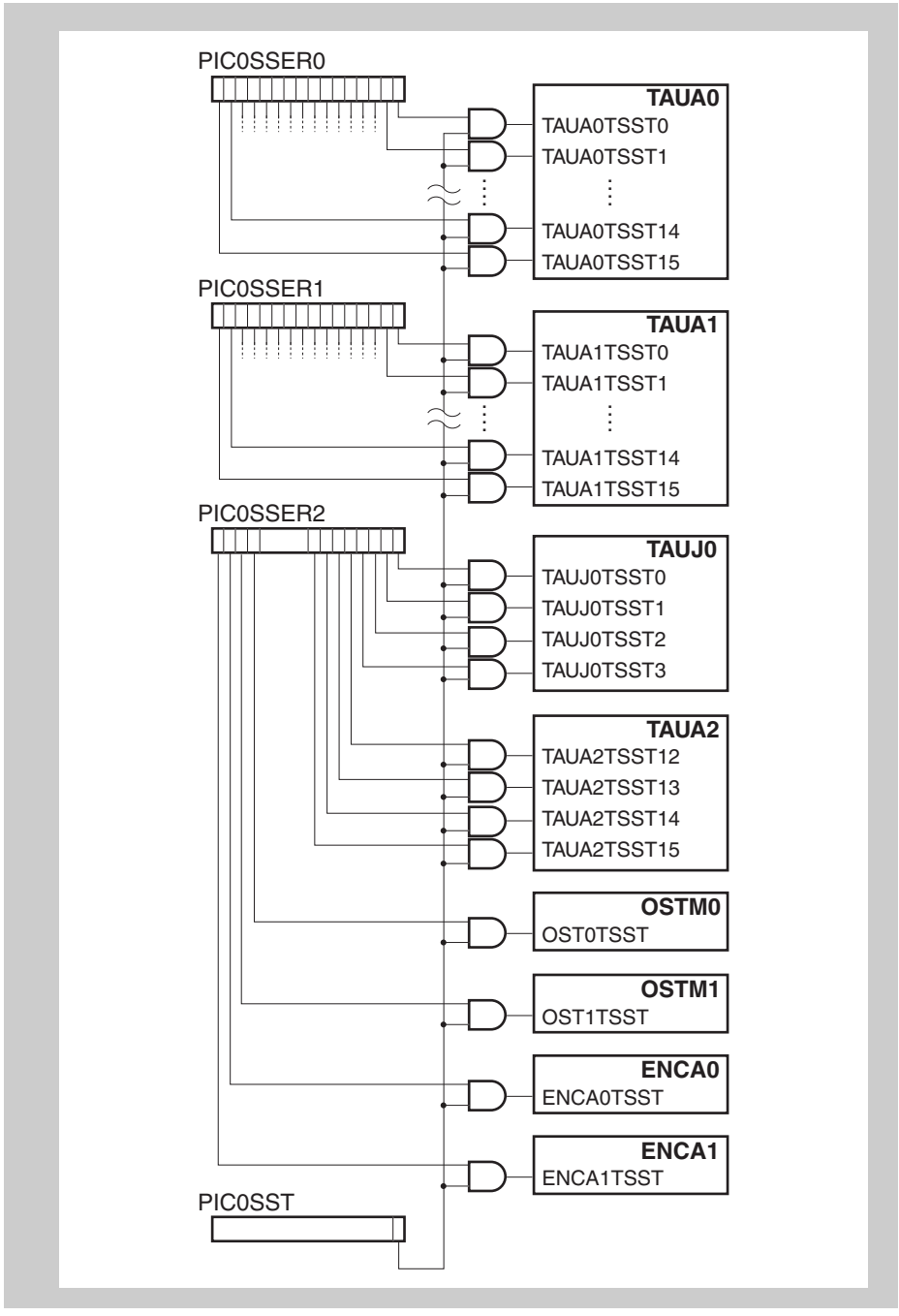


Figure 16-4 Block diagram of timer simultaneous start function

16.4.2 TAUA and TAPA connection function (including Hi-Z output control)

- Hi-Z output control

  This function controls the Hi-Z output control signal input to TAPATHASIN of each TAPA.

- TAUA0 and TAPA connection

  This function selects a pair of signals from among the signals output from TAUA0 (the INTTAUA0Im and TAUA0TTOUTm (m = 0, 2, or 8) signals) according to the setting of the PIC0REG200 register, and then outputs them to TAPA0 and TAPA2 as TOP0TAPATSIM0 and TAPA0TAPATUDCM0.

- TAUA1 and TAPA connection

  This function selects a pair of signals from among the signals output from TAUA1 (the INTTAUA1Im and TAUA1TTOUTm (m = 0, 2, or 8) signals) according to the setting of the PIC0REG200 register, and then outputs them to TAPA1 and TAPA3 as TOP1TAPATSIM0 and TAPA1TAPATUDCM0.

16.4.3 TAUA and A/D converter connection function

Each A/D converter has three channel groups, and A/D converter hardware triggers are provided for each channel group.

The TAUA and A/D converter connection function of PIC ORs the internal trigger signals output from TAUA0 and TAUA1, and then outputs the generated signal as the A/D converter hardware trigger signal for each channel group.

The internal trigger signals input from TAUA0 and TAUA1 are the cycle (borrow) detection signals specified by the CNTn register for each channel.

The A/D converter hardware trigger signals are also output to TAPA.

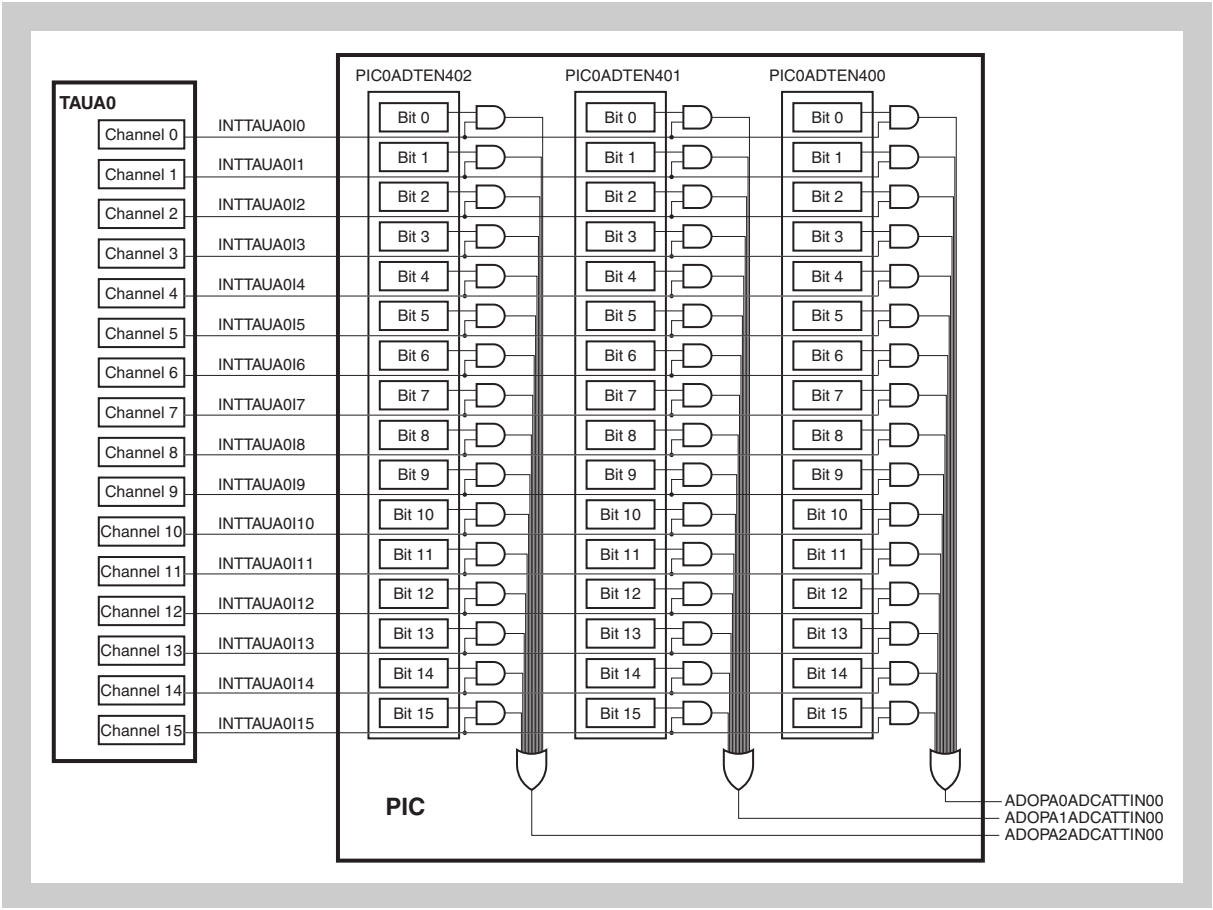


Figure 16-5 Block diagram of TAUA0 and A/D converter connection function

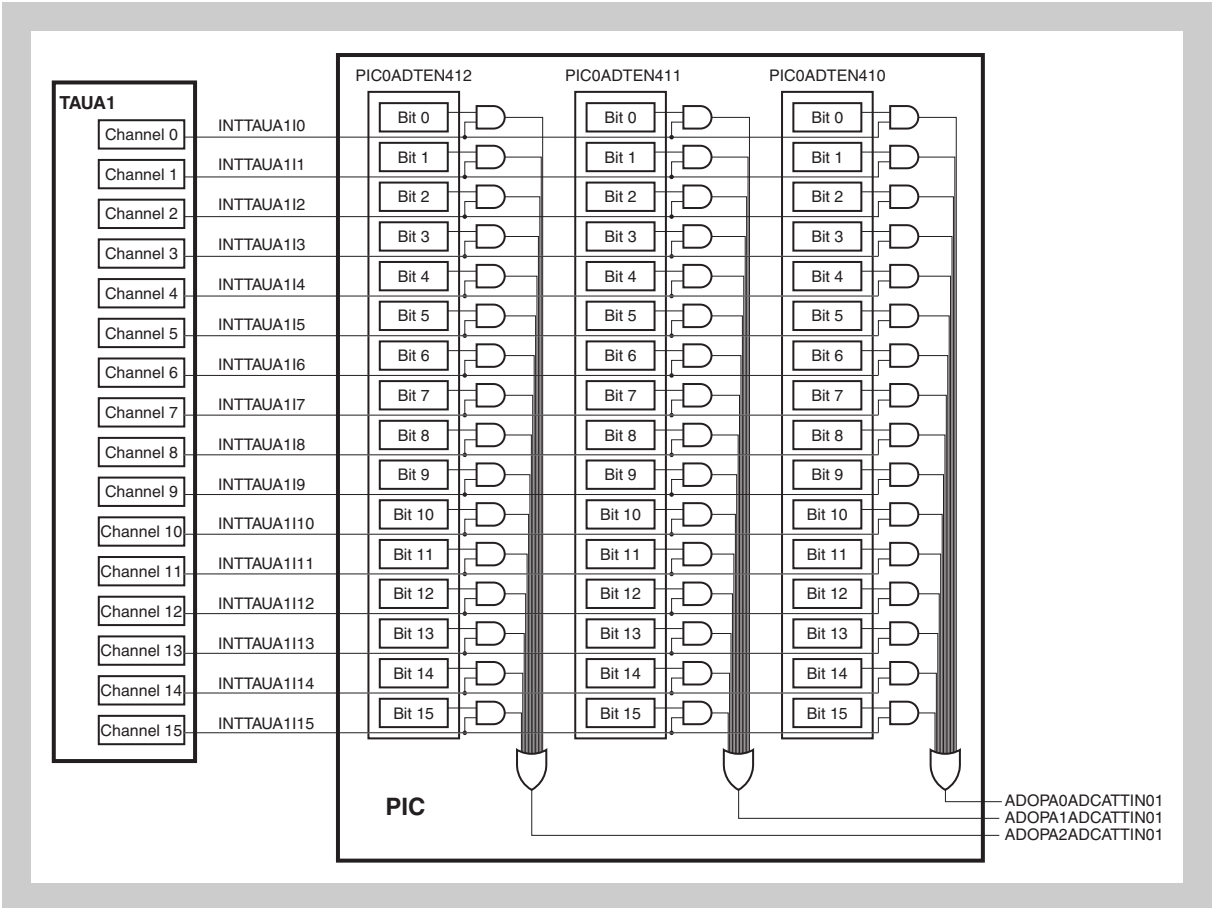


Figure 16-6 Block diagram of TAUA1 and A/D converter connection function

16.5 Connection between CAN controllers and timers (CAN time stamp function)

16.5.1 Overview

A time stamp can be specified for received messages by using the CAN controller and timer array unit together.

With this function, the timer array unit captures a timer value according to the TSOUT signal that is output from the CAN controller upon data frame reception. The CPU reads the captured value to obtain the time that the capture event occurred; in other words, the time stamp of the message received on the CAN bus.

FCN0 and FCN1 can be used together with CH1 of TAUJ0 and CH3 of TAUJ0, respectively.

16.5.2 Configuration

The instances and channels used for this function are as follows:

Table 16-14 Configuration of CAN time stamp function

CAN controller	Timer
FCN0	TAUJ0 CH1
FCN1	TAUJ0 CH3

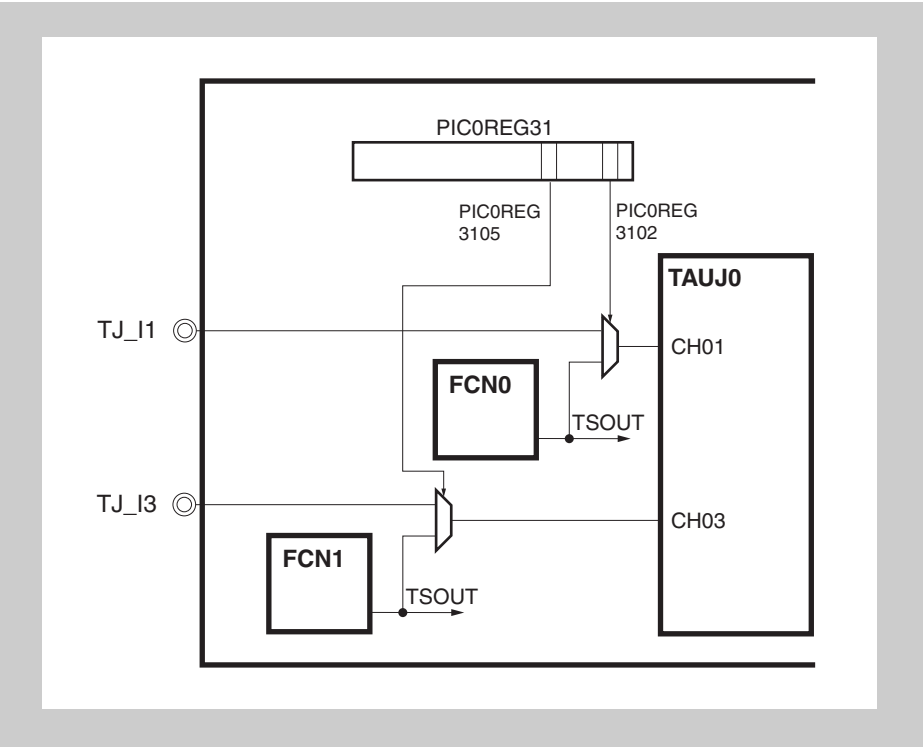


Figure 16-7 Block diagram of CAN time stamp function

16.5.3 Register

(1) PIC0REG31 - Timer I/O control register 31

This register is used to select the timer input signal.

Access This register can be read or written in 32-bit units.

Address <PIC0_base_OS> + $C0_H$

Default value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
7	6	5	4	3	2	1	0
0	0	PIC0REG3105	0	0	PIC0REG3102	0	0
R	R	R/W	R	R	R/W	R	R

Table 16-15 PIC0REG31 register contents

Bit position	Bit name	Function
5	PIC0REG3105	Select the input signal for CH3 of TAUJ0. 0: TJ_I3 1: CAN time stamp output from FCN1 (TSOUT signal)
2	PIC0REG3102	Select the input signal for CH1 of TAUJ0. 0: TJ_I1 1: CAN time stamp output from FCN0 (TSOUT signal)

Chapter 17 Window Watchdog Timer A (WDTA)

Caution Descriptions about the watchdog timer in this chapter are preliminary. The specifications are subject to change.

This chapter contains a generic description of window watchdog timer A.

17.1 "V850E2/MN4 WDTA Features" and *17.2 "WDTA Start-up Options"* describe all properties specific to the V850E2/MN4, such as instances, register base addresses, and input/output signal names.

The subsequent sections describe the features that apply to all implementations.

17.1 V850E2/MN4 WDTA Features

Instances This microcontroller has two instances of window watchdog timer A.

Table 17-1 Instances of WDTA

Window watchdog tmer A	μPD70F3510, 70F3512	μPD70F3514, 70F3515
Instances	1	2
Names	WDTA0	WDTA0, WDTA1

Instances index n Throughout this chapter, the individual instances of window watchdog timer A are identified by the index "n" (n = 0 to 1), for example, WDTAnWDTE for the WDTAn watchdog timer enable register.

Register addresses All WDTAn register addresses are given as addresses offset from the individual base address <WDTAn_base> of each WDTAn.

The <WDTAn_base> addresses of each WDTAn are listed in the following table:

Table 17-2 Register base addresses of WDTAn

WDTAn	<WDTAn_base> address
WDTA0	FFFF $C000_H$
WDTA1	FFFF $C100_H$

Clock supply The input clock for window watchdog timer A is WDTATCKI. WDTATCKI is connected to a clock generator.

Table 17-3 Clock of window watchdog timer A

WDTAn	Clock input signal	Connected to
WDTA0, WDTA1	WDTATCKI	f_{XW}

Interrupts and reset outputs

The interrupts and reset outputs of WDTAn are listed in the table below.

Table 17-4 WDTA interrupts and reset outputs

WDTAn signals	Function	Connected to
WDTA0		
WDTA0TRES	WDTA0 error reset	Reset Controller WDTA0RES
WDTA0TNMI	WDTA0 error NMI	Interrupt Controller WDTA0NMI
INTWDT0	WDTA0 75% interrupt	Interrupt Controller INTWDT0
WDTA1		
WDTA1TRES	WDTA1 error reset	Reset Controller WDTA1RES
WDTA1TNMI	WDTA1 error NMI	Interrupt Controller WDTA1NMI
INTWDT1	WDTA1 75% interrupt	Interrupt Controller INTWDT1

WDTATRTYP

The connections of the WDTA reset type input signals are listed in the table below.

Table 17-5 WDTATRTYP connections

WDTAn signals	Connected to
WDTA0TRTYP	0 (fixed)
WDTA1TRTYP	0 (fixed)

Restrictions

The V850E2/MN4 does not support the default start mode and variable activation code (VAC) for WDTA.

Accessing WDTAnEVAC and WDTAnREF is prohibited.

Register specific to the V850E2/MN4

The following product-specific register is provided for WDTA of the V850E2/MN4:

- WDTANMI monitoring register

For dual-core products, this register is used to identify the WDTAn that generated an NMI interrupt.

(1) WDTANMI monitoring register (WNMISTAT)

When WDTAn instances are used in NMI request mode, this register is used to identify the WDTAn in which an NMI interrupt (WDTAnNMI) was generated first due to an overflow error. The WNMI1 bit must be fixed to 0 for single-core products. After an NMI interrupt (WDTAnNMI) was generated, no additional NMI interrupt (WDTAnNMI) will be generated even if an overflow error occurs in the other WDTA.

Access This register is read-only, in 32-bit units.

Address FFFF DA00$_H$

Initial Values 0000 0000$_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
7	6	5	4	3	2	1	0
0	0	0	0	0	0	WNMI1	WNMI0
R	R	R	R	R	R	R	R

Table 17-6 WNMISTAT register contents

Bit position	Bit name	Function
1, 0	WNMI1, WNMI0	

WNMI1	WNMI0	Status
0	0	No NMI interrupt (WDTAnNMI) was generated in both WDTA0 and WDTA1.
0	1	An NMI interrupt (WDTA0NMI) was generated in WDTA0. After this, the WNMI1 bit will not be set even if an overflow error occurs in WDTA1.
1	0	An NMI interrupt (WDTA1NMI) was generated in WDTA1. After this, the WNMI0 bit will not be set even if an overflow error occurs in WDTA0.
1	1	An NMI interrupt (WDTA0NMI) was generated simultaneously in both WDTA0 and WDTA1.

17.2 WDTA Start-up Options

The start-up options determine the start-up configuration of WDTA after reset release. They are described in the following table.

Table 17-7 WDTA start-up options

Start-up option	Function	Description
OPWDEN	WDTA enabler/disabler	Enables/disables WDTA: 1: WDTA is fixed to be enabled
OPWDOVF[2:0]	Count clock setting	Specifies the reset value of the count clock control bits WDTAnMD.WDTAnOVF[2:0]. These bits are fixed to 000
OPWDTPR	Start mode signal selector	Specifies the signal that sets the start mode: 0: This bit fixed to be OPWDRUN start-up option
OPWDRUN	Automatic start enabler	Specifies the start mode: 0: Software trigger start mode is fixed
OPWDVAC	Variable Activation Code (VAC) enabler	Enables/disables the Variable Activation Code function (VAC) 0: VAC is fixed to be disabled
OPWDWS[1:0]	Initial open window size setting	Specifies the reset value of the open window size control bits WDTAnMD.WDTAnWS[1:0]. These bits are fixed to 111 The open window size control bits only apply after the first WDTA trigger and not after reset release. After reset release the open window size is 100%. Refer to *17.4.5 "Window function"* for details.
OPWDINT	INTWDTn (75% interrupt) request generation	Specifies the reset value of control bit WDTAnMD.WDTAnWIE. This bit enables/disables the output of the 75% interrupt request INTWDTn. This bit is fixed tobe 1 Refer to *17.4.4 "75% interrupt output"* for details.

17.3 Functional Overview

Features summary

WDTA has the following functions:

- Operation mode after reset selectable by using start-up option
- Fixed activation code and variable activation code (VAC) selectable
- Two start modes available:
  - Automatic start mode
  - Software trigger mode
- Operation upon error detection selectable:
  - Generation of NMI request WDTAnTNMI on error detection
  - Generation of reset WDTAnTRES on error detection
- Interrupt request generation at 75% of the counter overflow value
- Window function

The following figure shows the main components of WDTA:

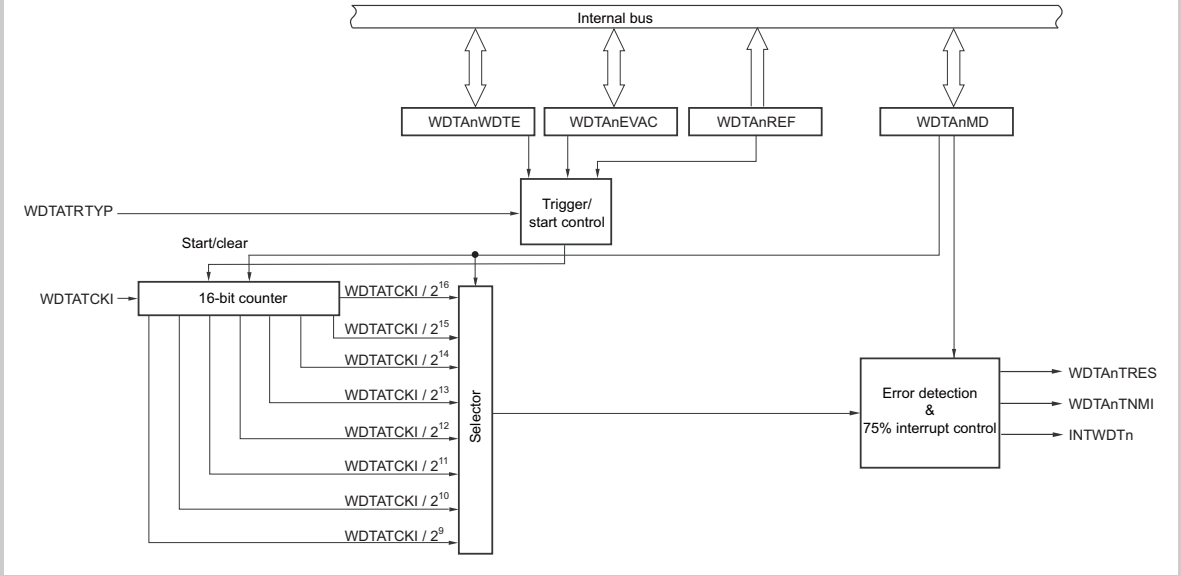


Figure 17-1 Block diagram of WDTA

17.4 Functional Description

WDTA generates a reset or a non-maskable interrupt if the 16-bit counter overflows or if any other error condition is fulfilled. For a description of all error conditions, refer to *17.4.3 “Error detection” on page 1125*.

The counter is cleared and restarted every time a WDTA trigger occurs in the open window period. Refer to *17.4.2 “WDTA trigger” on page 1124* and *17.4.5 “Window function” on page 1128* for details.

At 75% of the maximum counter value, WDTA can generate an interrupt request INTWDTn. Refer to *17.4.4 “75% interrupt output” on page 1126* for details.

The start-up option specifies the start mode and WDTA setting after reset release. The setting can be changed by writing to the watchdog timer mode register WDTAnMD. For details, see *17.4.1 “WDTA after reset release” on page 1121*.

17.4.1 WDTA after reset release

(1) Start modes

WDTA provides two modes for the counter start after reset release:

- Software trigger start mode

  The counter value remains 0000_H after reset release.

  The counter is started with the first WDTA trigger.

- Automatic start mode

  The counter starts automatically after reset release.

(2) Start mode selection

The start mode can be selected as follows:

- By start-up options
- By the WDTATRTYP input signal

  This signal indicates the reset type. Thus, the selected start mode after reset release depends on the reset type.

The start mode selection is listed in the following table.

Table 17-8 Start mode selection

Start-up options		Input signal	Reset type	Start mode
OPWDTPR	OPWDRUN	WDTATRTYP		
0	0	Ignored	Ignored	Software trigger
0	1	Ignored	Ignored	Automatic
1	Ignored	0	Any apart from automatic start reset source	Software trigger
1	Ignored	1	Automatic start reset source	Automatic

(3) WDTA settings after reset release

The WDTA settings are as follows between reset release and the first trigger:

Function	Setting	Remark
Start mode	Specified by start-up options	For a description of the start modes, refer to *17.4.1 "WDTA after reset release" on page 1121*.
Count clock		
75% interrupt mode		
Error mode	Reset mode	Any error condition before the first trigger generates a reset.
Open window size	100%	If automatic start mode is specified, the first trigger is valid any time before the counter overflows.

Change WDTA settings

After the first trigger, WDTA continues according to the settings of the watchdog timer mode register WDTAnMD.

To change the WDTA settings, WDTAnMD must be written *before* the first trigger. Changing the value of WDTAnMD *after* the first trigger leads to an error.

If WDTAnMD is not changed before the first trigger, the WDTA mode is specified by the initial value of WDTAnMD.

The new or initial value of WDTAnMD applies after the first trigger.

Automatic start mode timing

The automatic start mode timing and the changes to the WDTA settings are illustrated in the following figure.

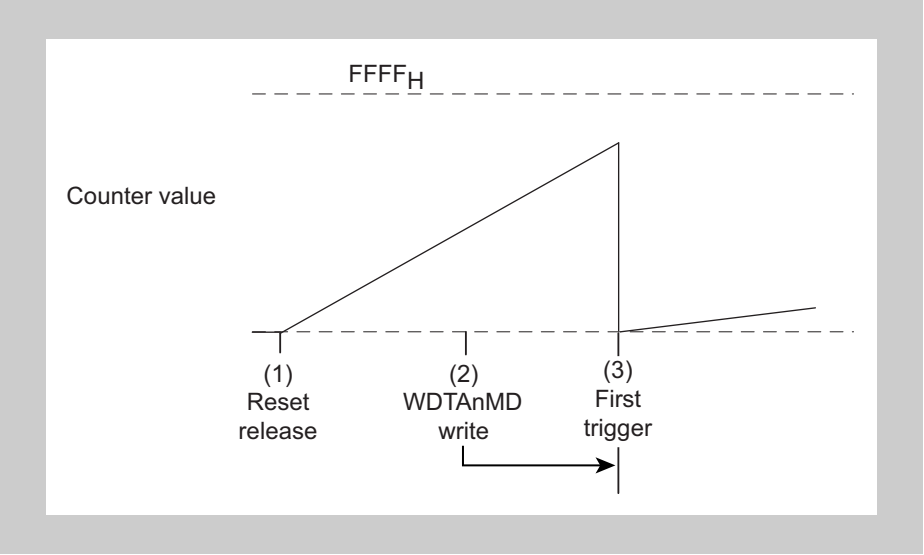


Figure 17-2 Timing diagram of WDTA start in automatic start mode

The timing diagram above shows the following:

1. After reset release, the counter starts immediately.
   The count clock is specified by the start-up options, for example:
   – Count clock after reset release = WDTATCKI / 2^{13}
   (OPWDOVF[2:0] = 100_B)
2. WDTAnMD is written before the first trigger. However, the settings are not applied immediately.
3. The first trigger must occur before the counter overflows.

   After the first trigger, the settings specified in WDTAnMD are applied, for example a new count clock.

Software trigger start mode timing

The software trigger start mode timing and the changes to the WDTA settings are illustrated in the following figure.

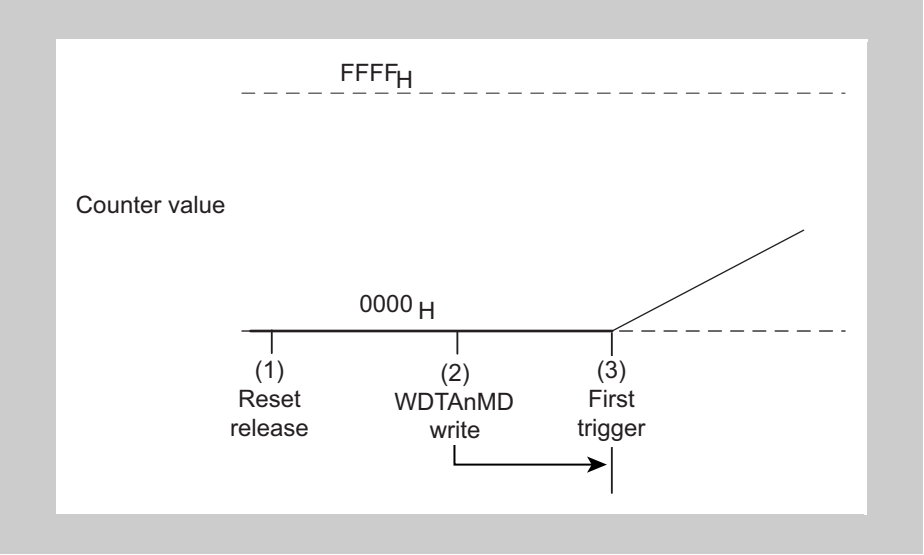


Figure 17-3 Timing diagram of WDTA start in software trigger start mode

The timing diagram above shows the following:

1. After reset release, the counter remains 0000_H until the first trigger.
   The count clock is specified by the start-up options, but it does not have any effect.
2. WDTAnMD is written before the first trigger. However, the settings are not applied immediately.
3. The counter starts at the first trigger.

   The count clock and other settings specified in WDTAnMD are applied.

17.4.2 WDTA trigger

The WDTA trigger has the following functions:

- Counter start trigger in software trigger start mode
- Counter restart trigger to avoid counter overflow

The trigger register used differs depending on whether the activation code is fixed or variable. The type of activation code and the associated trigger register are specified by using the start-up option OPWDVAC.

Table 17-9 Trigger register and activation code

Type of activation code	Trigger register	Activation code
Fixed	WDTAnWDTE	AC_H
Variable	WDTAnEVAC	Refer to *(1) "Variable activation code calculation" on page 1124* for details.

(1) Variable activation code calculation

The variable activation code (ExpectWDTE) is calculated using a reference value in register WDTAnREF. The reference value in WDTAnREF is updated each time the trigger register WDTAnEVAC is written.

Use the expression below to calculate the variable activation code (ExpectWDTE):

ExpectWDTE = AC_H – WDTAnREF (old)

Use the expression below to calculate how the WDTAnREF value is updated:

WDTAnREF (new) = rotate left 1 bit (ExpectWDTE)

The table below lists the variable activation codes according to the number of triggers.

Table 17-10 Variable activation code development

No[a]	WDTAnREF (old)		ExpectWDTE (AC_H - WDTAnREF)		WDTAnREF (new)	
0	0000 0000	00_H	1010 1100	AC_H	0101 1001	59_H
1	0101 1001	59_H	0101 0011	53_H	1010 0110	$A6_H$
2	1010 0110	$A6_H$	0000 0110	06_H	0000 1100	$0C_H$
…	…	…	…	…	…	…

a) Number of triggers after reset

Note Bit 7 of the WDTAnEVAC register (WDTAnEVAC7) cannot be cleared to 0 after WDTA has been started. Thus even if bit 7 of the activation code is 0, WDTA will not stop.

17.4.3 Error detection

The conditions for error detection are:

- Overflow interval time is exceeded (counter overflow)
- Wrong activation code is written to the trigger register
- Writing to the trigger register outside the open window.
- Illegal update of watchdog timer mode register WDTAnMD:
  - Writing a *new* value to WDTAnMD after the first trigger leads to an error detection.
  - Writing the same value to WDTAnMD after the first trigger does *not* lead to an error detection.

Error mode

When an error is detected, either an NMI request (WDTAnTNMI) or a reset (WDTAnTRES) is generated.

WDTAnMD.WDTAnERM selects the error mode:

- WDTAnMD.WDTAnERM = 0: NMI mode
- WDTAnMD.WDTAnERM = 1: reset mode

The following figure shows the reset or NMI request generation when the counter overflows and automatic start mode is selected.

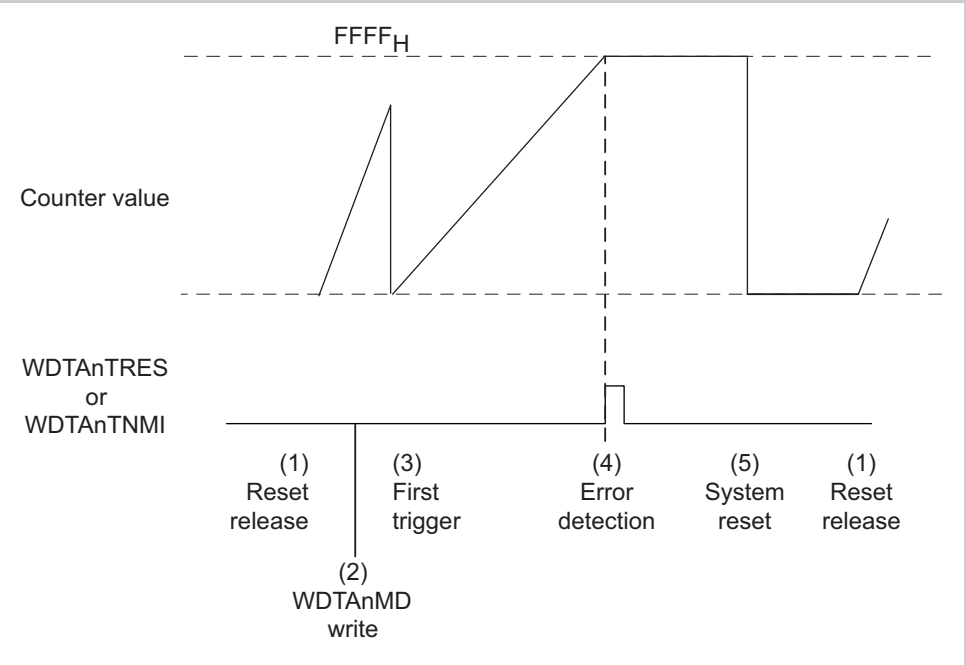


Figure 17-4 Timing diagram of WDTA NMI request or reset generation

The timing diagram above shows the following:

1. After reset release, the counter starts (automatic start mode is selected).
2. WDTAnMD is written before the first trigger. However, the settings are not applied immediately.
3. The counter is cleared at the first trigger and the new WDTA settings are applied.
4. When the counter overflows, an error is detected. Depending on the error mode, either interrupt request WDTAnTNMI or reset WDTAnTRES is generated.

   The counter value remains until the system reset is performed.
5. When the system is reset, the counter is cleared and stopped until reset release.

17.4.4 75% interrupt output

When the counter reaches 75% of the maximum counter value, the interrupt request INTWDTn is generated.

This function can be automatically enabled with the start-up option OPWDINT = 1.

This function can be enabled or disabled by specifying a setting in the WDTAnMD.WDTAnWIE register.

The following figure shows 75% interrupt request generation under the following conditions:

- Automatic start mode selected
- Count clock changes after first trigger

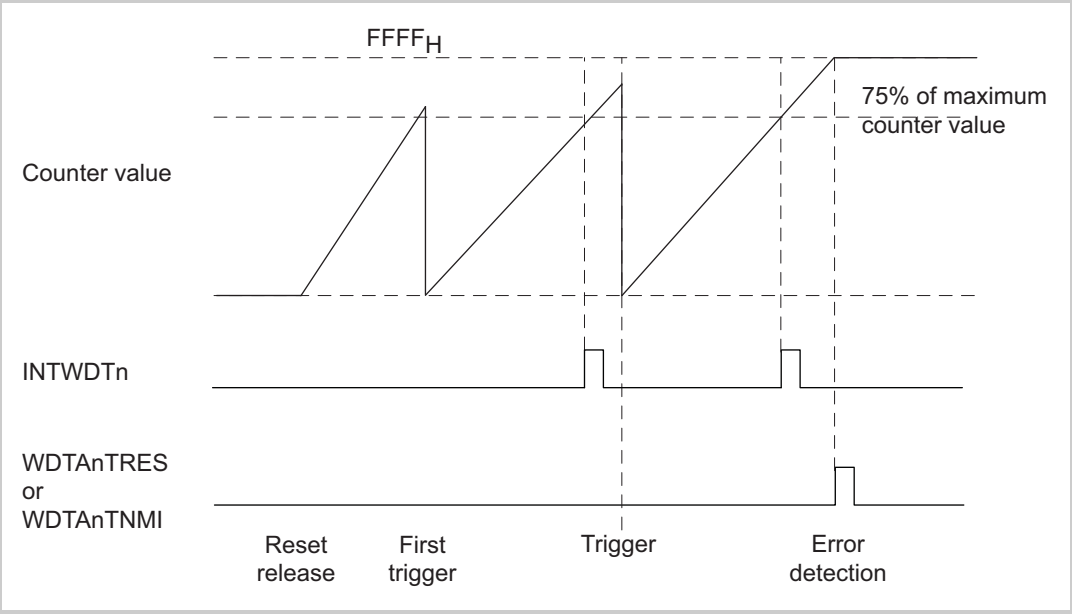


Figure 17-5 Timing diagram of WDTA 75% interrupt output

17.4.5 Window function

When the open window size is set to less than 100%, an error is detected if the trigger occurs outside the open window.

The definition of the open window size differs before and after the first trigger:

- After reset release, the open window size is 100%.

  OPWDWS[1:0] and bits WDTAnMD.WDTAnWS[1:0] are not applied.

- After the first trigger, the open window size is specified by bits WDTAnMD.WDTAnWS[1:0].

The following figure shows WDTA operation with an open window size of 25% and with automatic start mode selected.

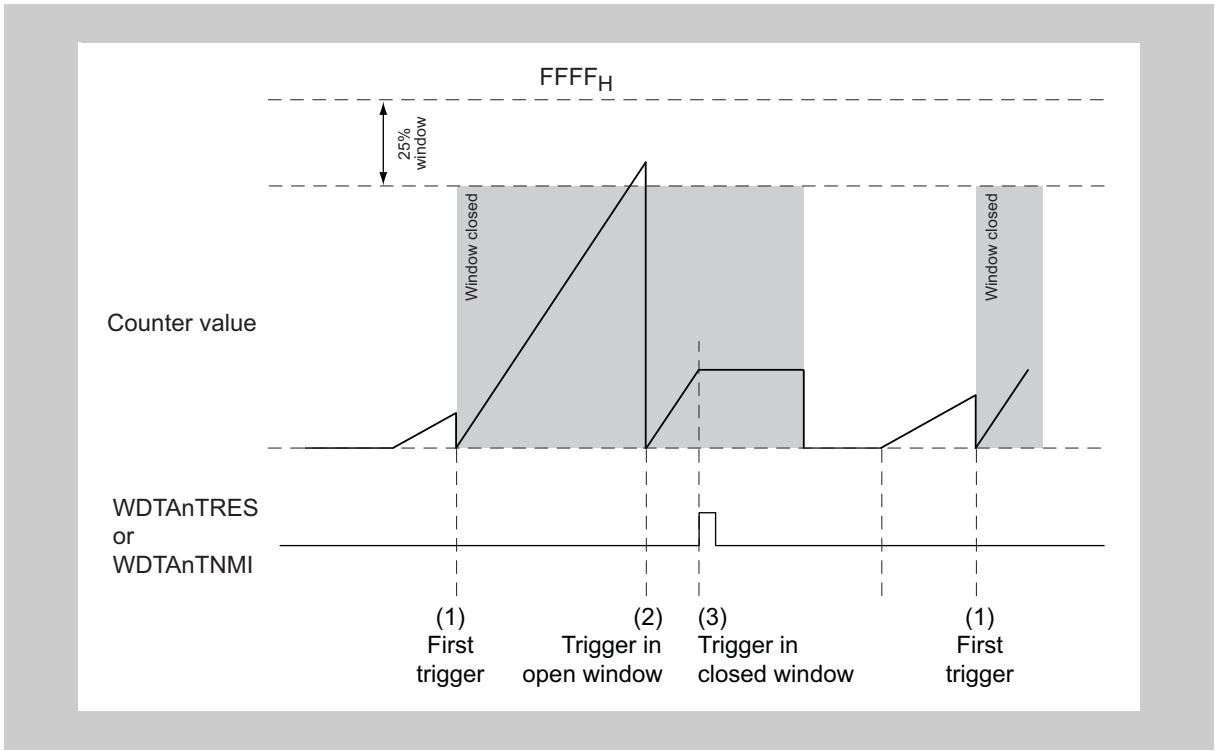


Figure 17-6 Timing diagram of WDTA window function

The timing diagram above shows the following:

1. The open window size is fixed to 100% for the first trigger.
2. A trigger that occurs in the open window does not generate an error.
3. A trigger that occurs in the closed window generates a WDTAnTNMI request or a WDTAnTRES reset, depending on the selected error mode.

17.5 Registers

This section contains a description of all registers of WDTA.

17.5.1 WDTA register overview

WDTA is controlled and operated by the following registers:

Table 17-11 WDTA register overview

Register name	Shortcut	Address
WDTA enable register	WDTAnWDTE	<WDTAn_base> + 0000_H
WDTA enable register for VAC	WDTAnEVAC	<WDTAn_base> + 0004_H
WDTA reference value register	WDTAnREF	<WDTAn_base> + 0008_H
WDTA mode register	WDTAnMD	<WDTAn_base> + $000C_H$

17.5.2 WDTA registers details

(1) WDTAnWDTE – WDTA enable register

This register is the WDTA start control and trigger register if the VAC function is not used (start-up option OPWDVAC = 0).

WDTA trigger Writing AC_H to this register restarts the counter. Refer to *17.4.2 "WDTA trigger" on page 1124* for details.

The behaviour of this register depends on activation of the VAC function, refer to *Table 17-14 "WDTAnWDTE behaviour"*.

Access This register can be read/written in 8-bit units.

Address <WDTAn_base> + 0000_H

Initial Value Depends on start-up options OPWDEN, OPWDTPR, WDTATRTYP, OPWDRUN and OPWDVAC. Refer to *Table 17-13 "WDTAnRUN initial value"*.

Reset input initializes this register to 0000H.

7	6	5	4	3	2	1	0
WDTAnRUN	0	1	0	1	1	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 17-12 WDTAnWDTE register contents

Bit position	Bit name	Function
7	WDTAnRUN	Enables/disables WDTAn: 0: WDTAn disabled 1: WDTAn enabled Since WDTA cannot be stopped once it was started, this bit can only be cleared by a reset.

Initial value This bit is only valid if WDTA is enabled (OPWDEN = 1) and VAC is disabled (OPWDVAC = 0). In this case, the initial value of bit WDTAnRUN depending on other start-up options is listed below:

Table 17-13 WDTAnRUN initial value

Start-up options		Input signal	Start mode	Initial value of WDTAnRUN
OPWD TPR	OPWD RUN	WDTATRTYP		
0	0	Ignored	Software trigger	0
0	1	Ignored	Automatic	1
1	0	Ignored	Software trigger	0
1	Ignored	0	Software trigger	0
1	Ignored	1	Automatic	1

The behaviour of WDTAnWDTE during read/write accesses depends on activation of the VAC mode, as shown in the table below.

Table 17-14 WDTAnWDTE behaviour

OPWDVAC	WDTAnWDTE		Remark
	Read	Write	
0	AC_H	Ignored	VAC disabled: WDTAnWDTE enabled
1	$2C_H$	Ignored WDTA trigger AC_H[a]	VAC enabled: WDTAnWDTE disabled

a) Any other write value will lead to an error detection.

(2) WDTAnEVAC – WDTA enable VAC register

This register is the start control and trigger register if the VAC function is used (start-up option OPWDVAC = 1).

WDTA trigger Writing the correct activation code to this register restarts the counter. Refer to *17.4.2 "WDTA trigger" on page 1124*.

The behaviour of this register depends on activation of the VAC function, refer to *Table 17-17 "WDTAnEVAC behaviour"*.

Access This register can be read/written in 8-bit units.

Address <WDTAn_base> + 0004_H

Initial Value Depends on start-up options OPWDEN, OPWDTPR, WDTATRTYP, OPWDRUN and OPWDVAC. Refer to *Table 17-15 "WDTAnEVAC register contents"*.

Reset input initializes this register to 0000_H.

7	6	5	4	3	2	1	0
WDTAnEVAC7	0	1	0	1	1	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 17-15 WDTAnEVAC register contents

Bit position	Bit name	Function
7	WDTAnEVAC7	Starts or stops WDTAn. 0: Stop WDTAn. 1: Start WDTAn. Since WDTA cannot be stopped once it was started, this bit can only be cleared by a reset. Thus even if bit 7 of the activation code is 0, WDTA will not stop.

Initial value This bit is only valid if WDTA is enabled (OPWDEN = 1) and VAC is enabled (OPWDVAC = 1). In this case, the initial value of bit WDTAnEVAC7 depending on other start-up options is listed below:

Table 17-16 WDTAnEVAC7 initial value

Start-up options		Input signal	Start mode	Initial value of WDTAnEVAC7
OPWDTPR	OPWDRUN	WDTATRTYP		
0	0	Ignored	Software trigger	0
0	1	Ignored	Automatic	1
1	Ignored	0	Software trigger	0
1	Ignored	1	Automatic	1

The behaviour of WDTAnEVAC during read/write accesses depends on activation of the VAC mode, as shown in the table below.

Table 17-17 WDTAnEVAC behaviour

OPWDVAC	WDTAnEVAC		Remark
	Read	Write	
0	$2C_H$	ignored	VAC disabled: WDTAnEVAC disabled
1	last written VAC	WDTA trigger VAC[a]	VAC enabled: WDTAnEVAC enabled

a) Any other write value will lead to an error detection.

(3) WDTAnREF – WDTA reference value register

This register contains the reference value for calculating the activation code of the VAC function. It is automatically updated after every trigger operation. Refer to *17.4.2 "WDTA trigger" on page 1124*.

If VAC is disabled (OPWDVAC = 0), reading this register returns 00_H.

Access This register can be read in 8-bit units.

Address <WDTAn_base> + 0008_H

Initial Value Reset input initializes this register to 00H.

7	6	5	4	3	2	1	0
WDTAnREF[7:0]							
R	R	R	R	R	R	R	R

Table 17-18 WDTAnREF register contents

Bit position	Bit name	Function
7 to 0	WDTAnREF[7:0]	Reference value for activation code calculation.

(4) WDTAnMD – WDTA mode register

This register specifies the overflow interval time, the 75% interrupt output mode, the error mode, and the open window size.
It can be updated only once after reset release and before the first trigger. The updated value is effective from the next WDTA trigger.
Updating this register after WDTA has been started leads to error detection, but the read value of this register can be written without generating an error.

Access This register can be read/written in 8-bit units.

Address <WDTAn_base> + $000C_H$

Initial Value Depends on start-up options OPWDOVF[2:0], OPWDINT and OPWDWS[1:0]. Refer to *17.2 "WDTA Start-up Options"*.
Reset input initializes this register to 0000H.

7	6	5	4	3	2	1	0
0	WDTAnOVF[2:0]			WDTAnWIE	WDTAnERM	WDTAnWS[1:0]	
R/W[a]	R/W	R/W	R/W	R	R/W	R/W	R/W

a) Writing to this bit is ignored, reading returns 0.

Table 17-19 WDTAnMD register contents

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>6 to 4</td><td>WDTAnOVF[2:0]</td><td>Selects the overflow interval time:
<table>
<tr><th>WDTAnOVF2</th><th>WDTAnOVF1</th><th>WDTAnOVF0</th><th>Overflow interval time</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>2^9/WDTATCKI</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>2^{10}/WDTATCKI</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>2^{11}/WDTATCKI</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>2^{12}/WDTATCKI</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>2^{13}/WDTATCKI</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>2^{14}/WDTATCKI</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>2^{15}/WDTATCKI</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>2^{16}/WDTATCKI</td></tr>
</table>
The reset values of WDTAnOVF[2:0] depend on start-up option OPWDOVF[2:0].</td></tr>
<tr><td>3</td><td>WDTAnWIE</td><td>Enables/disables the 75% interrupt request INTWDTn:
0: INTWDTn disabled
1: INTWDTn enabled
The reset value of WDTAnWIE depends on start-up option OPWDINT.</td></tr>
<tr><td>2</td><td>WDTAnERM</td><td>Specifies the error mode:
0: NMI request mode
1: Reset mode</td></tr>
<tr><td>1, 0</td><td>WDTAnWS[1:0]</td><td>Selects the open window size:
<table>
<tr><th>WDTAnWS1</th><th>WDTAnWS0</th><th>Open window size</th></tr>
<tr><td>0</td><td>0</td><td>25%</td></tr>
<tr><td>0</td><td>1</td><td>50%</td></tr>
<tr><td>1</td><td>0</td><td>75%</td></tr>
<tr><td>1</td><td>1</td><td>100%</td></tr>
</table>
The reset values of WDTAnWS[1:0] depend on start-up option OPWDWS[1:0].</td></tr>
</table>

Chapter 18 OS Timer (OSTM)

This chapter contains a generic description of the OS timer.

The first section describes all properties specific to the V850E2/MN4, such as instances, register base addresses, and input/output signal names.
The subsequent sections describe the features that apply to all implementations.

18.1 V850E2/MN4 OSTM Features

Instances

This microcontroller has the following number of instances of the OS timer.

Table 18-1 Instances of OS Timer

OS timer	μPD70F3510, 70F3512	μPD70F3514, 70F3515
Instances	1	2
Names	OSTM0	OSTM0, OSTM1

Instances index n

Throughout this chapter, the individual instances of OSTM is identified by the index “n” (n = 0 or n = 0, 1), for example, OSTMnCTL for the OSTM control register.

Register addresses

All OS timer register addresses are given as addresses offset from the individual base addresses <OSTMn_base>.
The <OSTMn_base> addresses of each OSTMn are listed in the following table:

Table 18-2 Register base addresses <OSTMn_base>

OSTMn instance	<OSTMn_base> address
OSTM0	$FF80\ 0000_H$
OSTM1	$FF80\ 1000_H$

Clock supply

The following clock is supplied to the OS timer:

Table 18-3 OSTM clock supply

OSTMn instance	OSTMn clock	Connected to
OSTM0	PCLK	f_{PCLK}
OSTM1	PCLK	f_{PCLK}

I/O signals

The I/O signals of the OS timer are listed in the table below.

Table 18-4 OSTMn I/O signals

OSTMn signals	Function	Connected to
OSTMnTCKE	Count clock enable	Fixed to 1.
OSTMnTSST	Count start	PIC
OSTMnTOUT	OS Timer output	No connection

Interrupts

The OS Timer can generate the following interrupt requests:

Table 18-5 OSTMn interrupt requests

OSTMn signals	Function	Connected to
OSTM0		
OSTM0TINT	OSTM0 interrupt	Interrupt Controller INTOSTM0
OSTM1		
OSTM1TINT	OSTM1 interrupt	Interrupt Controller INTOSTM1

18.2 Functional Overview

Features summary

The OS timer has the following features:

- Two operation modes
  - Interval timer mode
  - Free-run compare mode
- Two output modes (if OSTMnTTOUT signal is externally output)
  - Software control mode
  - Timer output toggle mode
- Synchronization with other peripheral functions (if a signal is input to OSTMnTSST)
- OSTMnTINT interrupt

The following block diagram shows the main components of the OS timer.

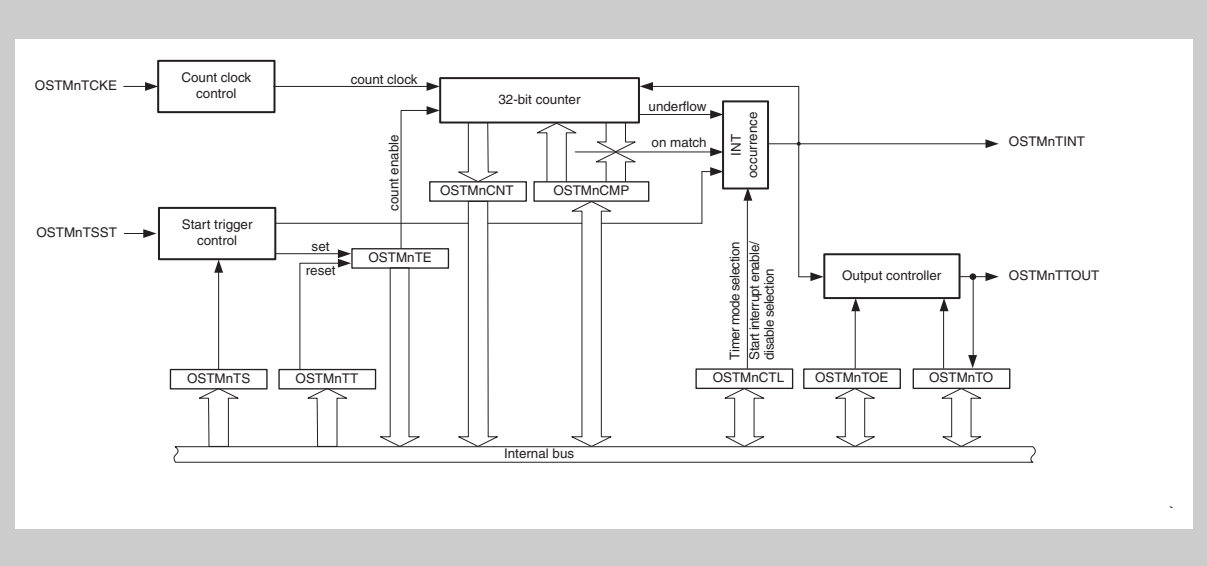


Figure 18-1 Block diagram of the OS timer

18.3 Functional Description

The OS timer is a 32-bit timer/counter.

It can be used as an interval timer or in free-run compare mode. The selected operation mode specifies the count direction (up/down) and controls the interrupt request generation.

The OS timer can be synchronized with other peripheral functions by using the count clock enable signal OSTMnTCKE and the count start signal OSTMnTSST. (See *18.3.1 "Count clock" on page 1138* and *18.3.4 "Starting and stopping the timer" on page 1140* for details.)

18.3.1 Count clock

The count clock of the OS timer is defined by PCLK and the OSTMnTCKE input:

- To use PCLK as the count clock, OSTMnTCKE must be fixed to 1.
- If the OSTMnTCKE signal is input, the count operation is based on this signal.

This is illustrated in the following figures.

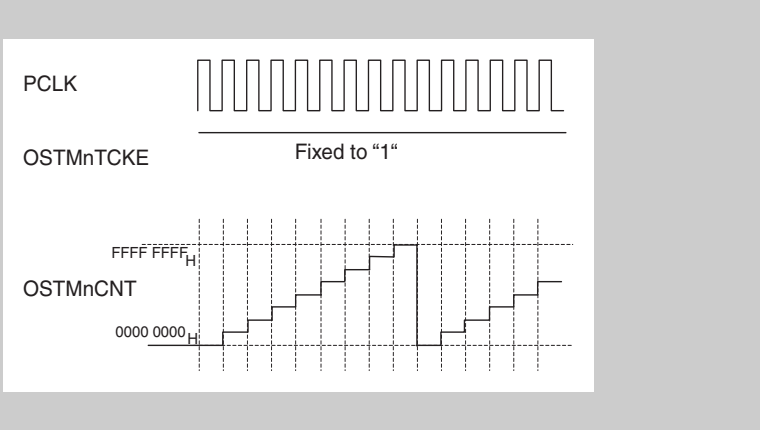


Figure 18-2 Count operation with OSTMnTCKE fixed to 1

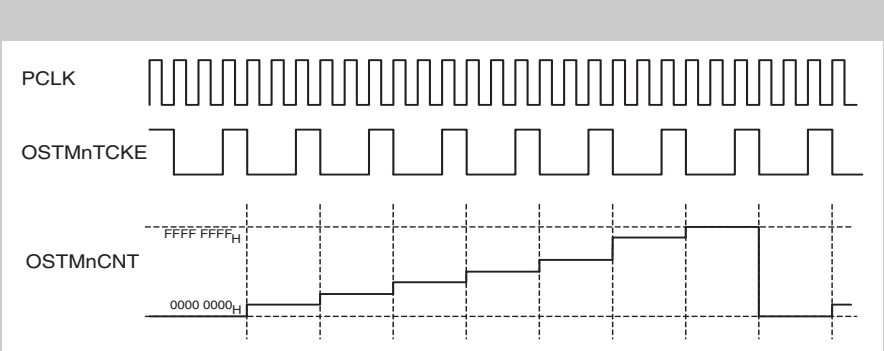


Figure 18-3 Count operation based on the OSTMnTCKE signal

18.3.2 Output modes

The OS timer has the following output modes:

- Software control mode:

  The level set to OSTMnTO.OSTMnTO is output to OSTMnTTOUT.

- Timer output toggle mode:

  OSTMnTTOUT toggles when the OSTMnTINT request is generated.

The output mode is selected by bit OSTMnTOE.OSTMnTOE.

Both output modes are illustrated in the following figure.

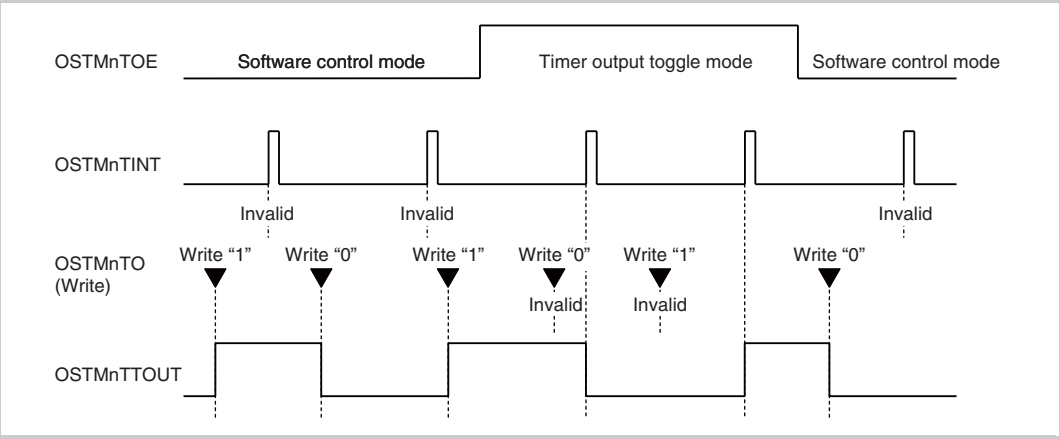


Figure 18-4 Timing diagram of output modes

The timing diagram above shows the following:

- In software control mode, OSTMnTTOUT changes to the value that was set in OSTMnTO.OSTMnTO.
- In timer output toggle mode, OSTMnTO.OSTMnTO and OSTMnTTOUT toggle when the OSTMnTINT interrupt request is generated.

18.3.3 Interrupt request generation

Interrupt request OSTMnTINT is generated on counter underflow (interval timer mode) or when the counter matches the compare value (free-run compare mode).

Additionally, an interrupt request can be generated at counter start or counter restart. This is controlled by bit OSTMnCTL.OSTMnMD0.

Since OSTMnTINT triggers toggling of OSTMnTTOUT in timer output toggle mode (OSTMnTOE.OSTMnTOE = 1), the setting of the OSTMnCTL.OSTMnMD0 bit also affects the output of OSTMnTTOUT.

This is illustrated in the following figure.

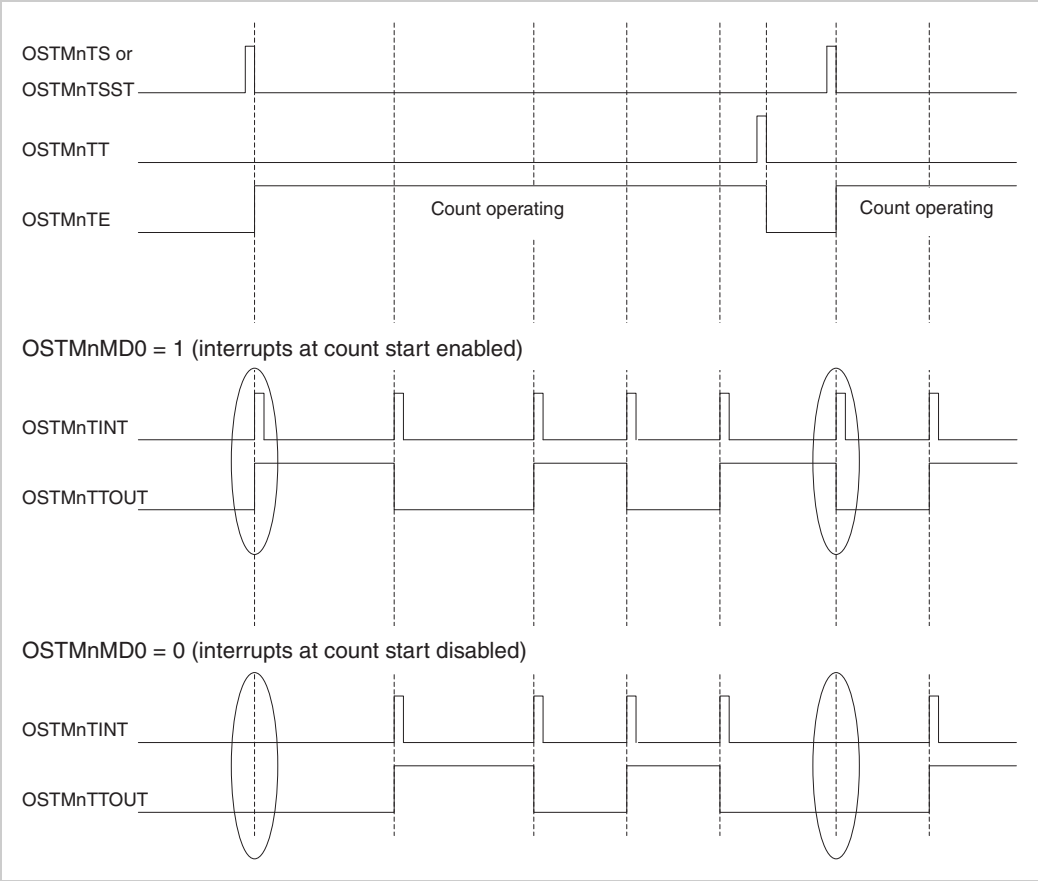


Figure 18-5 Interrupt generation at count start

18.3.4 Starting and stopping the timer

The OS timer is started and stopped as follows:

Start The timer is started

- by setting OSTMnTS.OSTMnTS, or
- when the OSTMnTSST signal changes from 0 to 1.

The status bit OSTMnTE.OSTMnTE is set to 1.

Depending on the operation mode, the counter starts to count down or to count up. Refer to *18.3.5 “Interval timer mode” on page 1141* and *18.3.6 “Free-run compare mode” on page 1144* for details.

Stop The timer is stopped by setting the bit OSTMnTT.OSTMnTT = 1.

Status bit OSTMnTE.OSTMnTE is cleared.

When the counter is stopped, the registers OSTMnTO and OSTMnCNT hold their current values until a new count operation starts.

Simultaneous start If the OSTMnTSST signal is connected to another internal peripheral function, the OS timer can be started in synchronization with the peripheral function.

18.3.5 Interval timer mode

In interval timer mode, the OSTM can be used as a reference timer generating interrupt requests at fixed intervals.

(1) Basic operation in interval timer mode

In interval timer mode, the timer counts down, starting from the value specified in the OSTMnCMP register. When the counter underflows (0000 0000_H is reached), an interrupt request OSTMnTINT is generated.

The interval timer mode is set by OSTMnCTL.OSTMnMD1 = 0.

The OSTMnCMP register can be rewritten at any time. If it is rewritten during count operation, the counter loads the new OSTMnCMP value when the next 0000 0000_H is reached. Then the counter continues with the new value.

OSTMnTINT and OSTMnTTOUT periods

The periods of OSTMnTINT and OSTMnTTOUT are:

- OSTMnTINT occurrence period = count clock period * (OSTMnCMP + 1)
- OSTMnTTOUT period = OSTMnTINT occurrence period * 2

The following figure shows the basic operation of the OS timer in interval timer mode with counter start interrupt enabled (OSTMnCTL.OSTMnMD0 = 1) and OSTMnTTOUT in timer output toggle mode (OSTMnTOE.OSTMnTOE = 1):

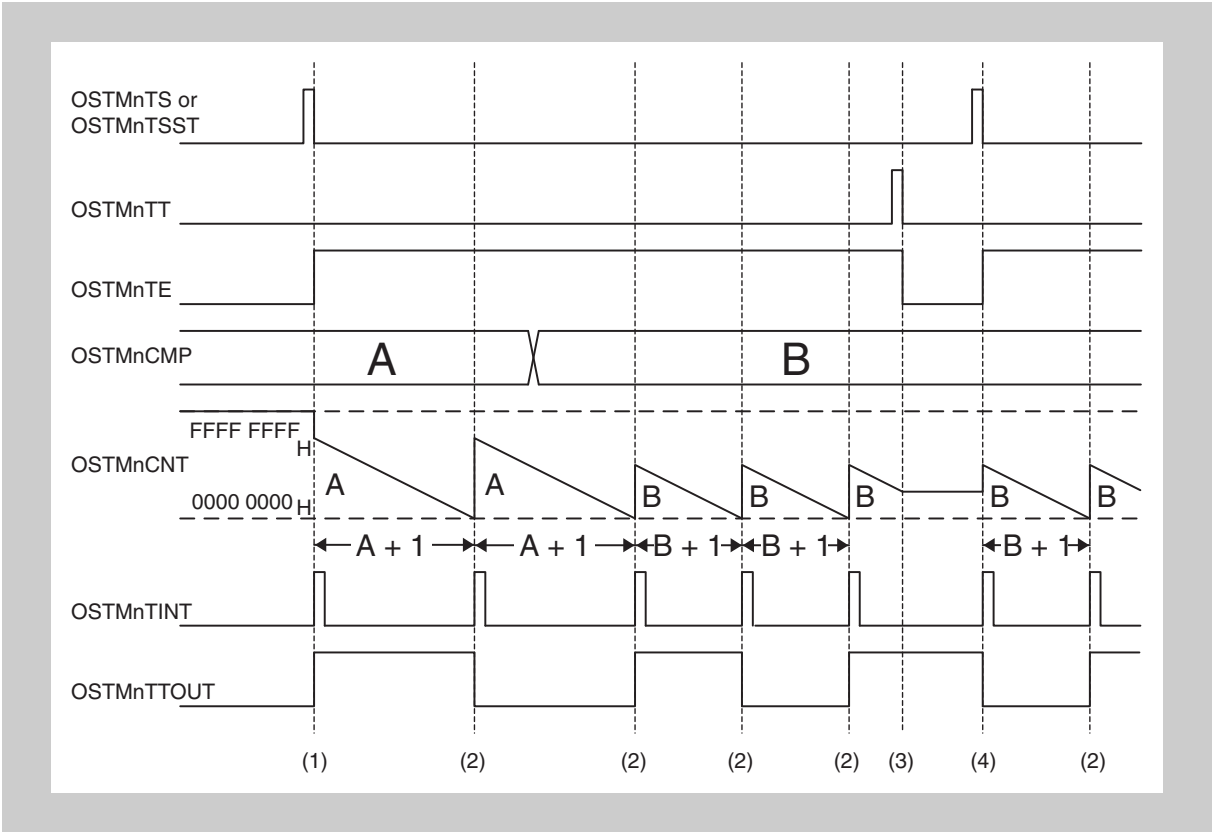


Figure 18-6 Timing diagram of OSTM in interval timer mode

The timing diagram above shows the following:

1. The counter starts counting when OSTMnTS.OSTMnTS = 1 or OSTMnTSST = 1. The OSTMnTE.OSTMnTE bit is set, indicating that a count operation is in progress.
   The counter starts counting down from the value of OSTMnCMP.
   If OSTMnCTL.OSTMnMD0 = 1, the interrupt request OSTMnTINT is generated when counting starts, and the OSTMnTTOUT output toggles.
   The counter value is indicated by the OSTMnCNT register.
2. When the counter reaches 0000 0000_H, the interrupt request OSTMnTINT is output and the OSTMnTTOUT output toggles. The counter loads the new start value from OSTMnCMP and continues to count down.
3. If OSTMnTT.OSTMnTT is set and the counter stops, the OSTMnTE.OSTMnTE bit is cleared, indicating that the counter is stopped.

   The counter holds the current value until it restarts counting.
4. If OSTMnTS.OSTMnTS or OSTMnTSST is set to restart counting, the counter loads the value from OSTMnCMP and starts counting down.

Forced restart

A forced restart of the counter is performed by setting OSTMnTS.OSTMnTS = 1 or generating a high to low transition of the OSTMnTSST signal during the count operation.

The counter loads the start value from the OSTMnCMP register and continues to count down.

The following figure shows the forced restart of the OS timer in interval timer mode, with counter start interrupt enabled (OSTMnCTL.OSTMnMD0 = 1) and OSTMnTTOUT in timer output toggle mode (OSTMnTOE.OSTMnTOE = 1):

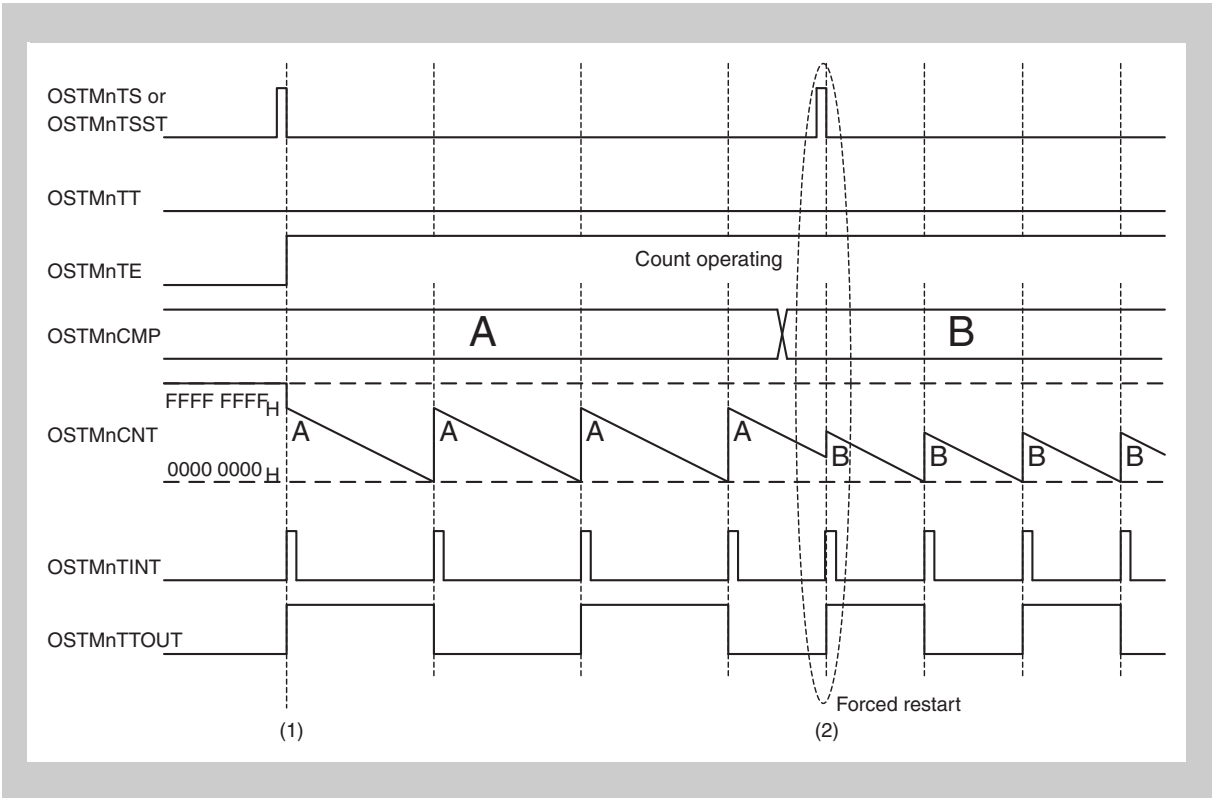


Figure 18-7 Timing diagram of forced restart in interval timer mode

The timing diagram above shows the following:

1. The counter is started as shown and explained in *Figure 18-6 "Timing diagram of OSTM in interval timer mode" on page 1141*.
2. If OSTMnTS.OSTMnTS or OSTMnTSST is set while the counter is operating (OSTMnTE.OSTMnTE = 1), the counter restarts counting.

   The counter immediately restarts counting down, starting with the current value of OSTMnCMP.
   If OSTMnCTL.OSTMnMD0 = 1, the interrupt request OSTMnTINT is generated when counting starts, and the OSTMnTTOUT output toggles.

(2) Operation when OSTMnCMP = 0000 0000_H

When OSTMnCMP = 0000 0000_H the OS timer behaves as follows:

- While the counter is enabled, the OSTMnTINT interrupt request is always 1.
- If OSTMnTTOUT output is in timer output toggle mode, OSTMnTTOUT toggles at each count clock cycle.

The following figure shows the operation of the OS timer, when OSTMnCMP = 0000 0000_H, counter start interrupt is enabled (OSTMnCTL.OSTMnMD0 = 1) and OSTMnTTOUT is in timer output toggle mode (OSTMnTOE.OSTMnTOE = 1).

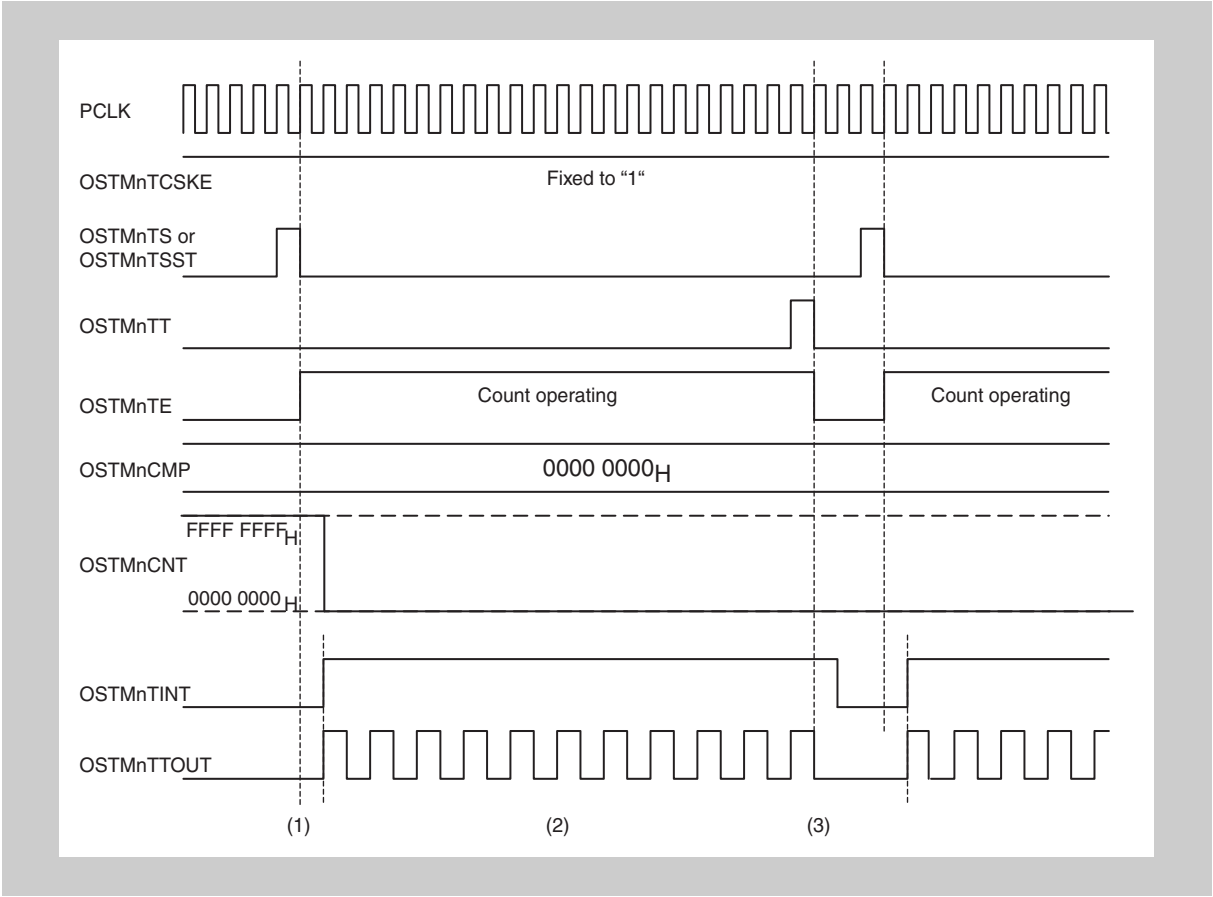


Figure 18-8 Timing diagram when OSTMnCMP = 0000 0000_H in interval timer mode

The timing diagram above shows the following:

1. After counter start, the counter starts counting, but will be loaded with the OSTMnCMP value, and thus remains 0000 0000_H.
2. The interrupt request OSTMnTINT is continuously asserted and OSTMnTTOUT starts toggling at each count clock cycle.
3. After counter stop, the interrupt request OSTMnTINT is deasserted and OSTMnTTOUT stops toggling.

(3) Initialization for interval timer mode

The setting procedure in interval timer mode after a reset release is described below:

Initialization

1. Set the start value of the down-counter in the OSTMnCMP register.
2. To use the OSTMnTTOUT output pin:
   – In software control mode, initialize OSTMnTO
   – Select the output mode (OSTMnTOE.OSTMnTOE = 1).
3. Select the interval timer mode by clearing bit OSTMnCTL.OSTMnMD1.
4. Select the interrupt mode at counter start (OSTMnCTL.OSTMnMD0 = 1).

18.3.6 Free-run compare mode

(1) Basic operation in free-run compare mode

In free-run compare mode, the counter counts up from 0000 0000_H to FFFF $FFFF_H$. When the value of the OSTMnCMP register matches the current count value, the OSTMnTINT interrupt request is output.

The free-run compare mode is selected by setting OSTMnCTL.OSTMnMD1 = 1.

The OSTMnCMP register can be rewritten at any time.

The following figure shows the basic operation of the OS timer in free-run compare mode with counter start enabled (OSTMnCTL.OSTMnMD0 = 1) and OSTMnTTOUT in timer output toggle mode (OSTMnTOE.OSTMnTOE = 1):

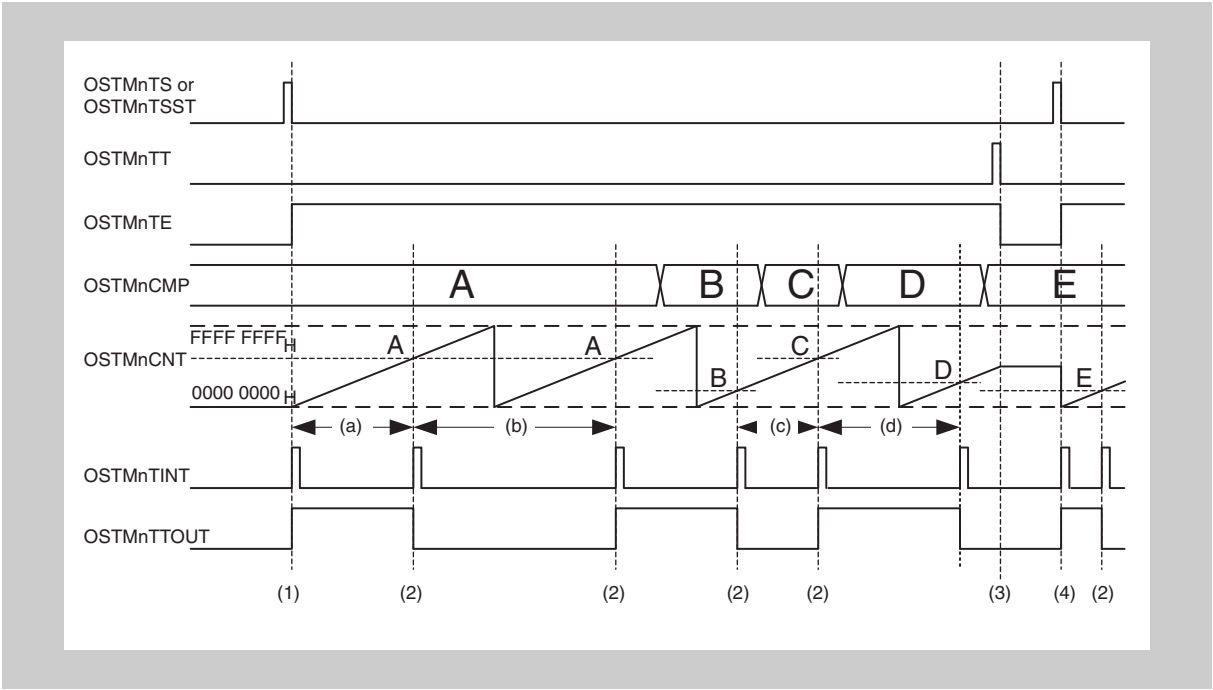


Figure 18-9 Timing diagram of OSTM in free-run compare mode

The timing diagram above shows the following:

1. The counter starts counting when OSTMnTS.OSTMnTS = 1 or OSTMnTSST = 1. The OSTMnTE.OSTMnTE bit is set, indicating that the counter is operating.
   The counter counts up from 0000 0000_H to FFFF $FFFF_H$. The counter value is indicated by OSTMnCNT register.
2. When the value of the OSTMnCMP register matches the current counter value, the interrupt request OSTMnTINT is output and the OSTMnTTOUT output toggles.
3. At counter stop (OSTMnTT.OSTMnTT = 1), the OSTMnTE.OSTMnTE bit is cleared, indicating that the counter is disabled.

   The counter holds the current value until it restarts counting.
4. If OSTMnTS.OSTMnTS or OSTMnTSST is set to restart counting, the counter starts counting from 0000 0000_H.

The OSTMnTINT occurrence period is different at count start and depends on the old and new compare value if OSTMnCMP is rewritten during operation:

Table 18-6 OSTMnTINT occurrence timing

Old compare	New compare	Counter value at time of rewrite	OSTMnTINT occurrence period	Label in timing diagram
Counter start			(A + 1) × count clock period	(a)
A	A	No rewrite	(FFFF FFFF$_H$ + 1) × count clock period	(b)
B	C > B	B < counter value < C	(C - B) * count clock period	(c)
C	D < C	Counter value > D, C	(FFFF FFFF$_H$ – C + D + 1) × count clock period	(d)

Forced restart A forced restart operation is not performed even if the bit OSTMnTS.OSTMnTS is set or OSTMnTSST = 1 during the count operation. The counter ignores this setting and continues counting.

(2) Operation when OSTMnCMP = 0000 0000$_H$

The following figure shows the operation of the OS timer when OSTMnCMP = 0000 0000$_H$, counter start interrupt is enabled (OSTMnCTL.OSTMnMD0 = 1) and OSTMnTTOUT is in timer output toggle mode (OSTMnTOE.OSTMnTOE = 1).

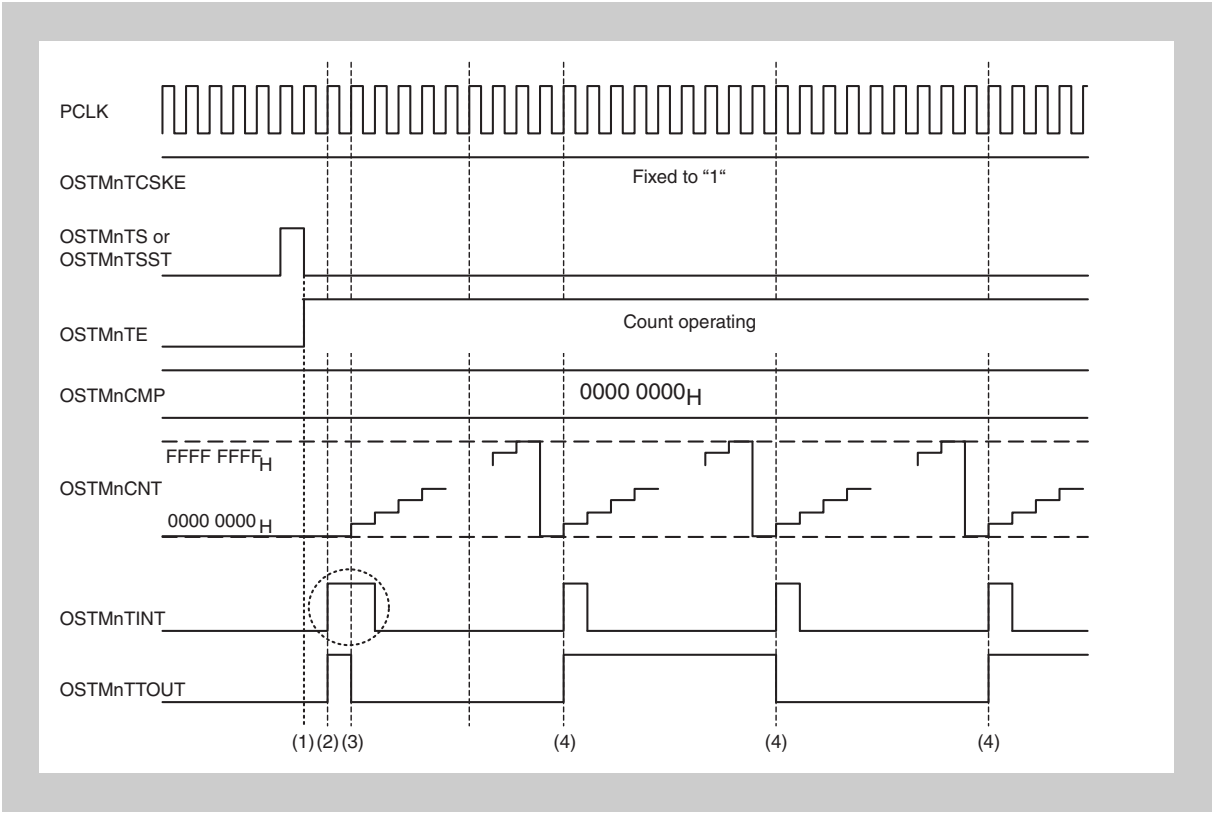


Figure 18-10 Timing diagram when OSTMnCMP = 0000 0000$_H$ in free-run compare mode

The timing diagram above shows the following:

1. After counter start, the counter starts counting up from 0000 0000_H to FFFF $FFFF_H$.
2. The interrupt request OSTMnTINT at counter start is generated.
3. If the current count value matches OSTMnCMP, the compare interrupt is generated. In the above case with OSTMnCMP = 0000 0000_H, OSTMnTINT stays active for 2 PCLK periods.
4. Every FFFF $FFFF_H$ clock cycles the interrupt request OSTMnTINT is output and OSTMnTTOUT toggles.

(3) Initialization for free-run compare mode

The setting procedure in free-run compare mode after a reset release is described below:

Initialization

1. Set the compare value in the OSTMnCMP register.
2. To use the OSTMnTTOUT output pin:
   - In software control mode, initialize OSTMnTO.
   - Select the output mode (OSTMnTOE.OSTMnTOE = 1).
3. Select the free-run compare mode by setting the bit OSTMnCTL.OSTMnMD1.
4. Select the interrupt mode at counter start by the bit OSTMnCTL.OSTMnMD0.

18.4 Registers

This section contains a description of all registers of the OS timer.

18.4.1 OS timer register overview

The OS timer is controlled and operated by the following registers:

Table 18-7 OS timer register overview

Register name	Shortcut	Address
OSTM compare register	OSTMnCMP	<OSTMn_base>
OSTM counter register	OSTMnCNT	<OSTMn_base> + 04_H
OSTM output register	OSTMnTO	<OSTMn_base> + 08_H
OSTM output enable register	OSTMnTOE	<OSTMn_base> + $0C_H$
OSTM count enable status register	OSTMnTE	<OSTMn_base> + 10_H
OSTM count start trigger register	OSTMnTS	<OSTMn_base> + 14_H
OSTM count stop trigger register	OSTMnTT	<OSTMn_base> + 18_H
OSTM control register	OSTMnCTL	<OSTMn_base> + 20_H

18.4.2 OS timer register details

(1) OSTMnCMP - OSTM compare register

This register stores the start value of the down-counter or the value with which the counter is compared, depending on the operation mode.

Access This register can be read/written in 32-bit units.

Address <OSTMn_base>

Initial Value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OSTMnCMP[31:16]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OSTMnCMP[15:0]															
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

Table 18-8 OSTMnCMP register contents

Bit position	Bit name	Function
31 to 0	OSTMnCMP[31:0]	• In interval timer mode: start value of the down-counter • In free-run compare mode: compare value

(2) OSTMnCNT - OSTM counter register

This register indicates the count value of the timer.

Access This register is read-only, in 32-bit units.

Address <OSTMn_base> + 4_H

Initial Value The initial value depends on the operation mode of the OS timer, see *Table 18-10 "Correlation between operation mode, counting direction and initial value" on page 1149*. This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
OSTMnCNT[31:16]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OSTMnCNT[15:0]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 18-9 OSTMnCNT register contents

Bit position	Bit name	Function
31 to 0	OSTMnCNT[31:0]	32-bit counter value

The following table shows the correlation between operation mode, counting direction and initial value. The initial value is the value that is read after the operating mode has been changed.

Table 18-10 Correlation between operation mode, counting direction and initial value

Timer operation mode	OSTMnCTL.OSTMnMD1	Counting direction	Initial value
Interval timer mode	0[a]	Decrement	FFFF FFFF$_H$
Free-run compare mode	1	Increment	0000 0000$_H$

a) Value after reset.

(3) OSTMnTO - OSTM output register

This register specifies and reads the level of OSTMnTTOUT.

Access This register can be read/written in 8-bit units. It can only be written when the software control mode is enabled (OSTMnTOE.OSTMnTOE = 0).

Address <OSTMn_base> + 8_H

Initial Value 00_H. This register is initialized by any reset.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	OSTMnTO
R	R	R	R	R	R	R	R/W

Table 18-11 OSTMnTO register contents

Bit position	Bit name	Function
0	OSTMnTO	Specifies/reads the level of OSTMnTTOUT: 0: Low level 1: High level

(4) OSTMnTOE - OSTM output enable register

This register specifies OSTMnTTOUT output mode.

Access This register can be read/written in 8-bit units.

Address <OSTMn_base> +C_H

Initial Value 00_H. This register is initialized by any reset.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	OSTMn TOE
R	R	R	R	R	R	R	R/W

Table 18-12 OSTMnTOE register contents

Bit position	Bit name	Function
0	OSTMnTOE	Specifies the OSTMnTTOUT output mode: 0: Software control mode: The level set to OSTMnTO.OSTMnTO is output to OSTMnTTOUT. 1: Timer output toggle mode: OSTMnTTOUT toggles when the interrupt request OSTMnTINT is generated.

(5) OSTMnTE - OSTM count enable status register

This register indicates whether the counter is enabled or disabled.

Access This register is read-only, in 8-bit units.

Address <OSTMn_base> + 10_H

Initial Value 00_H. This register is initialized by any reset.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	OSTMnTE
R	R	R	R	R	R	R	R

Table 18-13 OSTMnTE register contents

Bit position	Bit name	Function
0	OSTMnTE	Indicates, whether the counter is enabled or disabled: 0: Counter disabled 1: Counter enabled This bit is set if OSTMnTS.OSTMnTS is set or OSTMnTSST becomes 1. This bit is cleared if OSTMnTT.OSTMnTT is set.

Note If the counter is disabled, the counter value OSTMnCNT remains its value.

If the counter is restarted again, it

- restarts from the value specified by OSTMnCMP in interval timer mode.
- restarts with count value 0000 0000_H in free-run compare mode.

(6) OSTMnTS - OSTM count start trigger register

This register starts the counter.

Access This register can be written in 8-bit units. It is always read as 00_H.

Address <OSTMn_base> + 14_H

Initial Value 00_H. This register is initialized by any reset.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	OSTMnTS
R	R	R	R	R	R	R	W

Table 18-14 OSTMnTS register contents

Bit position	Bit name	Function
0	OSTMnTS	Starts the counter: 0: Ignored 1: Starts the counter and sets OSTMnTE.OSTMnTE = 1. • In interval timer mode, forced restart is executed when this bit is set while OSTMnTE.OSTMnTE = 1. • In free-run compare mode, setting this bit is ignored while OSTMnTE.OSTMnTE = 1.

(7) OSTMnTT - OSTM count stop trigger register

This register stops the counter.

Access This register can be written in 8-bit units. It is always read as 00_H.

Address <OSTMn_base> + 18_H

Initial Value 00_H. This register is initialized by any reset.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	OSTMnTT
R	R	R	R	R	R	R	W

Table 18-15 OSTMnTT register contents

Bit position	Bit name	Function
0	OSTMnTT	Stops the counter: 0: Ignored 1: Stops the counter and clears the bit OSTMnTE.OSTMnTE.

(8) OSTMnCTL - OSTM control register

This register specifies the counter operation mode and controls the generation of the interrupt request OSTMnTINT at counter start.

Access This register can be read/written in 8-bit units. It can only be written when the counter is disabled (OSTMnTE.OSTMnTE = 0).

Address <OSTMn_base> + 20_H

Initial Value 00_H. This register is initialized by any reset.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	OSTMn MD1	OSTMn MD0
R	R	R	R	R	R	R/W	R/W

Table 18-16 OSTMnCTL register contents

Bit position	Bit name	Function
1	OSTMnMD1	Specifies the counter operation mode: 0: Interval timer mode 1: Free-run compare mode
0	OSTMnMD0	Controls the OSTMnTINT interrupt request at counter start: 0: Disables interrupt at counter start 1: Enables interrupt at counter start

Chapter 19 Asynchronous Serial Interface E (UARTEn)

This chapter contains a generic description of asynchronous serial interface E.

The first section describes all properties specific to the V850E2/MN4, such as instances, register base addresses, and input/output signal names.
The subsequent sections describe the features that apply to all implementations.

19.1 V850E2/MN4 UARTEn Features

Instances

This microcontroller has six instances of asynchronous serial interface E (UARTEn).

Table 19-1 Instances of UARTEn

Asynchronous serial interface E	
Instance	6
Name	UARTE0 to UARTE5

Instances index n

Throughout this chapter, the instance of UARTE is identified by the index "n" (n = 0 to 5), for example, URTEnCTL0 for UARTEn control register 0.

Caution UARTE of the V850E2/MN4 does not support the LIN function.

Register addresses All UARTEn register addresses are given as addresses offset from the individual base address <URTEn_base_OS> or <URTEn_base_USER> for UARTEn. The <URTEn_base_OS> and <URTEn_base_USER> addresses of each UARTEn are listed in the following table:

Table 19-2 Register base addresses <URTEn_base>

UARTEn	Base address	Address
UARTE0	<URTEn_base_OS>	FF60 0000$_H$
	<URTEn_base_USER>	FFFF EE00$_H$
UARTE1	<URTEn_base_OS>	FF61 0000$_H$
	<URTEn_base_USER>	FFFF EF00$_H$
UARTE2	<URTEn_base_OS>	FF62 0000$_H$
	<URTEn_base_USER>	FFFF F000$_H$
UARTE3	<URTEn_base_OS>	FF63 0000$_H$
	<URTEn_base_USER>	FFFF F100$_H$
UARTE4	<URTEn_base_OS>	FF64 0000$_H$
	<URTEn_base_USER>	FFFF F200$_H$
UARTE5	<URTEn_base_OS>	FF65 0000$_H$
	<URTEn_base_USER>	FFFF F300$_H$

Clock supply All UARTEn provide one clock input. It is connected to the P bus clock f_{PCLK}.

Table 19-3 UARTEn clock supply

UARTEn	Clock supply	Connected to
UARTE0 to UARTE5	PCLK	f_{PCLK}

I/O signals The I/O signals of UARTEn are listed in the table below.

Table 19-4 UARTE I/O signals

UARTEn signal	Function	Connected to
URTEnTTXD	Transmit data output	Port TXDn
URTEnTRXD	Receive data input	Port RXDn

Interrupts The interrupts of UARTEn are listed in the table below.

Table 19-5 UARTE interrupts

UARTEn signal	Function	Connected to
UARTE0		
INTUAE0TIT	Transmission interrupt	• Interrupt controller 161 (INTCSIG0IC) • DMA controller trigger 117
INTUAE0TIR	Reception interrupt	• Interrupt controller 160 (INTCSIG0IR) • DMA controller trigger 116
INTUAE0TIS	Status interrupt	• Interrupt controller 159 (INTCSIG0IRE)
INTUAE0TRA	Receive/status interrupt	• Not used.
UARTE1		
INTUAE1TIT	Transmission interrupt	• Interrupt controller 164 (INTCSIG1IC) • DMA controller trigger 119
INTUAE1TIR	Reception interrupt	• Interrupt controller 163 (INTCSIG1IR) • DMA controller trigger 118
INTUAE1TIS	Status interrupt	• Interrupt controller 162 (INTCSIG1IRE)
INTUAE1TRA	Receive/status interrupt	• Not used.
UARTE2		
INTUAE2TIT	Transmission interrupt	• Interrupt controller 167 (INTCSIG2IC) • DMA controller trigger 121
INTUAE2TIR	Reception interrupt	• Interrupt controller 166 (INTCSIG2IR) • DMA controller trigger 120
INTUAE2TIS	Status interrupt	• Interrupt controller 165 (INTCSIG2IRE)
INTUAE2TRA	Receive/status interrupt	• Not used.
UARTE3		
INTUAE3TIT	Transmission interrupt	Interrupt controller 170 (INTCSIG3IC) DMA controller trigger 123
INTUAE3TIR	Reception interrupt	Interrupt controller 169 (INTCSIG3IR) DMA controller trigger 122
INTUAE3TIS	Status interrupt	Interrupt controller 168 (INTCSIG3IRE)
INTUAE3TRA	Receive/status interrupt	Not used.
UARTE4		
INTUAE4TIT	Transmission interrupt	Interrupt controller 173 (INTCSIG4IC) DMA controller trigger 125
INTUAE4TIR	Reception interrupt	Interrupt controller 172 (INTCSIG4IR) DMA controller trigger 124
INTUAE4TIS	Status interrupt	Interrupt controller 171 (INTCSIG4IRE)
INTUAE4TRA	Receive/status interrupt	Not used.
UARTE5		
INTUAE5TIT	Transmission interrupt	Interrupt controller 176 (INTCSIG5IC) DMA controller trigger 127
INTUAE5TIR	Reception interrupt	Interrupt controller 175 (INTCSIG5IR) DMA controller trigger 126
INTUAE5TIS	Status interrupt	Interrupt controller 174 (INTCSIG5IRE)
INTUAE5TRA	Receive/status interrupt	Not used.

19.2 Features

- Full-duplex communication:
  - Internal UARTEn receive data register n (URTEnRX)
  - Internal UARTEn transmit data register n (URTEnTX)
- 2-pin configuration:
  - URTEnTTXD: Transmit data output pin
  - URTEnTRXD: Receive data input pin
- Reception error and status output function
  - Parity error
  - Framing error
  - Overrun error
  - Data consistency error
- Interrupt requests: 4
  - Transmission interrupt INTUAEnTIT
  - Reception interrupt INTUAEnTIR
  - Status interrupt INTUAEnTIS
  - Receive/status interrupt INTUAEnTRA
- Character length: 7, 8 bits
- Parity function: odd, even, 0, none
- Transmission stop bit: 1, 2 bits
- MSB-/LSB-first transfer selectable
- Transmit/receive data inverted input/output possible
- 13 to 20 bits selectable for the BF (Break Field) in the LIN (Local Interconnect Network) communication format
  - Recognition of 11 bits or more possible for BF reception in LIN communication format
  - BF reception flag provided
- BF reception can be detected during data communication
- Bus monitor function to keep data consistency of the transmit data

19.3 Configuration

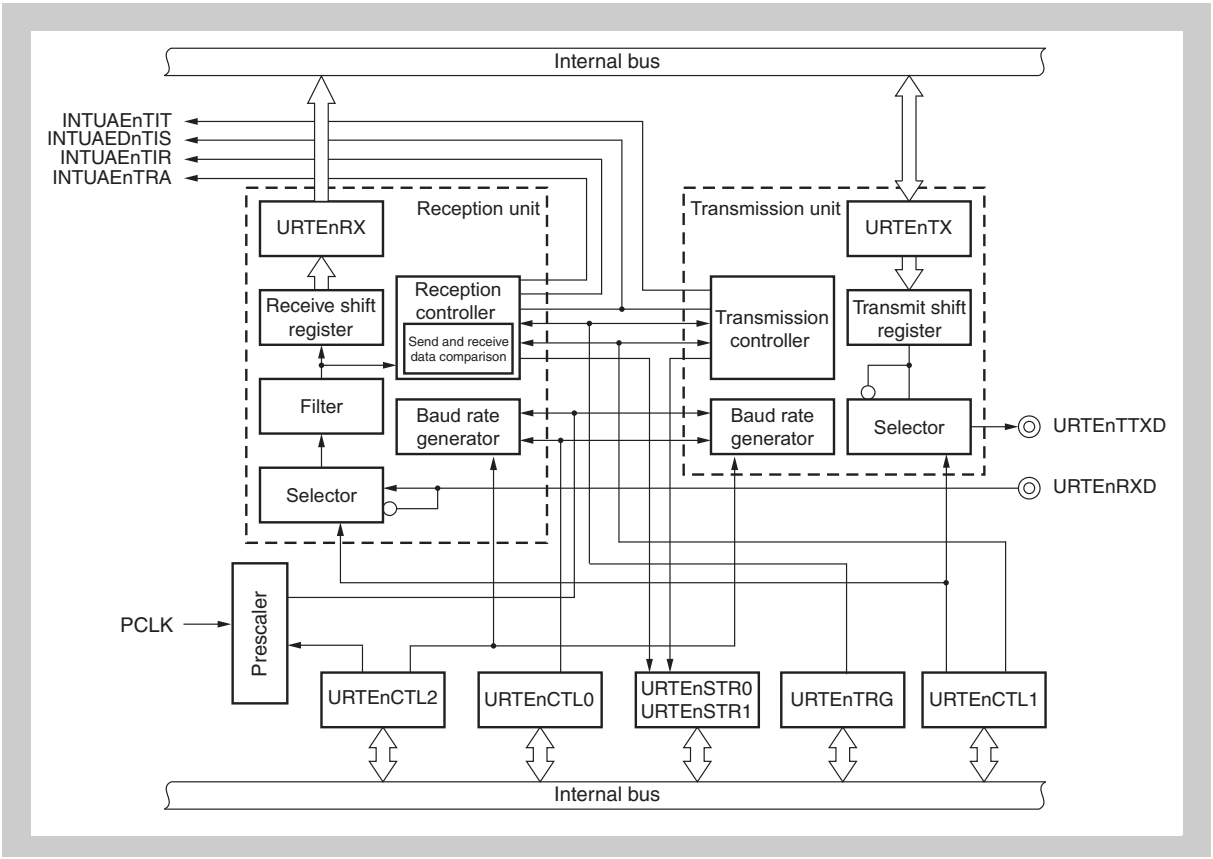


Figure 19-1 Block diagram of asynchronous serial interface UARTEn

19.4 UARTEn Registers

UARTEn is controlled and operated by means of the following registers:

Table 19-6 UARTEn registers

Register name	Symbol	Address
Control register 0	URTEnCTL0	<URTEn_base_USER> + 00_H
Control register 1	URTEnCTL1	<URTEn_base_OS> + 20_H
Control register 2	URTEnCTL2	<URTEn_base_OS> + 24_H
Trigger register	URTEnTRG	<URTEn_base_USER> + 04_H
Status register 0	URTEnSTR0	<URTEn_base_USER> + 08_H
Status register 1	URTEnSTR1	<URTEn_base_USER> + $0C_H$
Status clear register	URTEnSTC	<URTEn_base_USER> + 10_H
Receive data register	URTEnRX	<URTEn_base_USER> + 14_H
Transmit data register	URTEnTX	<URTEn_base_USER> + 18_H

Base address The base addressees <URTEn_base_USER> and <URTEn_base_OS> of UARTEn are defined in the first section of this chapter under the key word “Register addresses”.

(1) URTEnCTL0 - UARTEn control register 0

This register controls the serial transfer operation of UARTEn.

Access This register can be read/written in 8-bit and 1-bit units.

Address <URTEn_base_USER> + 00_H

Initial Value 00_H. This register is initialized by any reset.

7	6	5	4	3	2	1	0
URTEn PW	URTEn TXE	URTEn RXE	0	0	0	0	URTEn SLDC
R/W	R/W	R/W	R	R	R	R	R/W

Table 19-7 URTEnCTL0 register contents

Bit position	Bit name	Function
7	URTEnPW	UARTEn enable 0: Stop UARTEn operation 1: Enable UARTEn operation Changing this bit initializes all transmission and reception units.
6	URTEnTXE	Transmission operation enable 0: Disable transmission operation 1: Enable transmission operation • To start transmission, set URTEnPW, and then set URTEnTXE. To stop transmission, clear URTEnTXE, and then clear URTEnPW (they can be cleared at the same time). • To initialize the transmission unit, clear URTEnTXE, wait for two prescaler clock cycles, and then set URTEnTXE again. For details about the prescaler clock, see *(3) "URTEnCTL2 - UARTEn control register 2" on page 1164*.
5	URTEnRXE	Reception operation enable 0: Disable reception operation 1: Enable reception operation • To enable reception, set URTEnPW, and then set URTEnRXE. To stop reception, clear URTEnRXE, and then clear URTEnPW (they can be cleared at the same time). • To initialize the reception unit, clear URTEnRXE, wait for two prescaler clock cycles, and then set URTEnRXE again. Reception is enabled when the time of two prescaler clock cycles has elapsed since URTEnRXE is set. The rising edge detection of the URTEnTRXD signal is enabled when four prescaler clock cycles has elapsed after URTEnRXE is set. For details about the prescaler clock, see *(3) "URTEnCTL2 - UARTEn control register 2" on page 1164*.
0	URTEn SLDC	Data consistency check enable 0: Disable consistency check 1: Enable consistency check This bit selects the handling of data consistency error checks when transmitting data. When this bit is set to 1, the transmit data and receive data are compared, and if a mismatch is detected, URTEnSTR1.URTEnDCE is set to 1 and a status interrupt request INTUAEnTIS is issued. This bit is referenced only when starting transmission. Consequently, if this bit value is changed later on during transmission processing, the transmission processing continues, using the value set at the start of transmission.

Cautions

1. Disable transmission if UARTEn meets all the conditions below:
   - Transmission and reception are enabled (URTEnCTL0.URTEnPW = URTEnRXE = URTEnTXE = 1).
   - Data consistency check is enabled (URTEnCTL0.URTEnSLDC = 1).
   - Data is being transmitted or has been transmitted.

   Use the following procedure to keep reception enabled:
   - Check that no data is pending for transmission (URTEnSTR0.URTEnSSBT = URTEnSST = 0).
   - Check that no data is pending for reception (URTEnSTR0.URTEnSSBR = URTEnSSR = 0).
   - Disable transmission by clearing URTEnCTL0.URTEnTXE.

   The reason why this procedure is required is that the data consistency error flag URTEnSTR1.URTEnDCE is cleared if URTEnCTL0.URTEnTXE is cleared.
   Thus a potential data consistency error would not occur if transmission is disabled during a data transfer or after its completion.
2. Disable reception if UARTEn meets all the conditions below:
   - Transmission and reception are enabled (URTEnCTL0.URTEnPW = URTEnRXE = URTEnTXE = 1).
   - Data consistency check is enabled (URTEnCTL0.URTEnSLDC = 1).
   - Data is being transmitted or has been transmitted.

   Use the following procedure to keep transmission enabled:
   - Check that no data is pending for transmission (URTEnSTR0.URTEnSSBT = URTEnSST = 0).
   - Check that no data is pending for reception (URTEnSTR0.URTEnSSBR = URTEnSSR = 0).
   - Disable reception by clearing URTEnCTL0.URTEnRXE.

   The reason why this procedure is required is that the data consistency error flag URTEnSTR1.URTEnDCE is cleared and invalid if URTEnCTL0.URTEnTXE is cleared.
   Thus a potential data consistency error of already transmitted data would not occur.
3. Do not start data transmission if all the conditions below are met:
   - Data consistency check is enabled (URTEnCTL0.URTEnSLDC = 1).
   - BF reception is enabled (URTEnSTR0.URTEnSSBR = 1).
   - BF detection during reception is disabled (URTEnCTL1.URTEnSLBM = 0).

   A data consistency error will occur under above conditions when BF reception is completed. The status interrupt INTUAEnTIS will be asserted and the reception interrupt request INTUAEnTIR will not be generated (URTEnSTR1.URTEnBSF remains 0). Consequently BF reception completion will not be recognized.

(2) URTEnCTL1 - UARTEn control register 1

This register defines the data frame properties of UARTEn serial data transfers.

Access This register can be read/written in 16-bit units.

Address <URTEn_base_OS> + 20_H

Initial Value 5002_H. This register is initialized by any reset.

15	14	13	12	11	10	9	8
URTEn SLBM	URTEnBLG[2:0]			0	0	0	URTEn CLG
R/W	R/W	R/W	R/W	R	R	R	R/W

7	6	5	4	3	2	1	0
URTEnSLP[1:0]		URTEn TDL	URTEn RDL	0	URTEn SLG	URTEn SLD	URTEn SLIT
R/W	R/W	R/W	R/W	R	R/W	R/W	R/W

Table 19-8 URTEnCTL1 register contents (1/3)

Bit position	Bit name	Function
15	URTEnSLBM	BF receive mode selection 0: BF reception during data reception disabled. 1: BF reception during data reception enabled. • Changing this bit is only allowed if reception is disabled (URTEnCTL0.URTEnPW = 0 or URTEnCTL0.URTEnRXE = 0).
14 to 12	URTEn BLG[2:0]	BF bit length during transmission URTEnBLG2 / URTEnBLG1 / URTEnBLG0 / BF length: 1 0 1 — 13 bits 1 1 0 — 14 bits 1 1 1 — 15 bits 0 0 0 — 16 bits 0 0 1 — 17 bits 0 1 0 — 18 bits 0 1 1 — 19 bits 1 0 0 — 20 bits Changing these bits is only allowed if transmission is disabled (URTEnCTL0.URTEnPW = 0 or URTEnCTL0.URTEnTXE = 0).
8	URTEnCLG	Receive/transmit data bit length 0: 7 bits 1: 8 bits • When the transmission/reception is performed in the LIN format, set URTEnCLG to 1. • Changing this bit is only allowed if reception and transmission is disabled (URTEnCTL0.URTEnPW = 0 or URTEnCTL0.URTEnRXE = 0 and URTEnCTL0.URTEnTXE = 0).

Table 19-8 URTEnCTL1 register contents (2/3)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>7, 6</td><td>URTEn SLP[1:0]</td><td>Parity bit selection
<table>
<tr><th rowspan="2">URTEnSLP1</th><th rowspan="2">URTEnSLP0</th><th colspan="2">Operation</th></tr>
<tr><th>Transmission</th><th>Reception</th></tr>
<tr><td>0</td><td>0</td><td>Output without parity bit</td><td>Received with no parity</td></tr>
<tr><td>0</td><td>1</td><td>Output 0 parity (0-fixed)</td><td>No parity judgment</td></tr>
<tr><td>1</td><td>0</td><td>Output odd parity</td><td>Judged as odd parity</td></tr>
<tr><td>1</td><td>1</td><td>Output even parity</td><td>Judged as even parity</td></tr>
</table>
• If “Reception with no parity judgment” is selected during reception, a parity check is not performed. Therefore, since the URTEnSTR1.UTREnPE bit is not set, no error interrupt is output.
• When transmission/reception is performed in the LIN format, the URTEnPE bit is not set and no error interrupt is output upon detection of a parity error.
• Changing these bits is only allowed if reception and transmission is disabled (URTEnCTL0.URTEnPW = 0 or URTEnCTL0.URTEnRXE = URTEnCTL0.URTEnTXE = 0).</td></tr>
<tr><td>5</td><td>URTEnTDL</td><td>Transmission data level control
0: No inverted output of transmit data
1: Inverted output of transmit data
• The output level of the URTEnTTXD pin can be inverted using this bit. It inverts the URTEnTTXD output level immediately, regardless of the values of URTEnCTL0.URTEnPW and URTEnCTL0.URTEnTXE. Therefore, if URTEnTDL is set to 1 while the operation is disabled, the URTEnTTXD outputs low level.
• Changing this bit is only allowed if transmission is disabled (URTEnCTL0.URTEnPW = 0 or URTEnCTL0.URTEnTXE = 0).</td></tr>
<tr><td>4</td><td>URTEnRDL</td><td>Reception data level control
0: No inverted output of receive data
1: Inverted output of receive data
• The output level of the URTEnTRXD pin can be inverted using this bit. It inverts the URTEnTRXD input level immediately, regardless of the values of URTEnCTL0.URTEnPW and URTEnCTL0.URTEnRXE. Therefore, if URTEnRDL is set to 1 while the operation is disabled, the URTEnTRXD inputs low level.
• Changing this bit is only allowed if reception is disabled (URTEnCTL0.URTEnPW = 0 or URTEnCTL0.URTEnRXE = 0).</td></tr>
</table>

Table 19-8 URTEnCTL1 register contents (3/3)

Bit position	Bit name	Function
2	URTEnSLG	Stop bit number selection for transmission data 0: 1 bit 1: 2 bits • The stop bit length during data or BF reception is always handled as "1". • Changing this bit is only allowed if transmission is disabled (URTEnCTL0.URTEnPW = 0 or URTEnCTL0.URTEnTXE = 0).
1	URTEnSLD	Transfer direction selection 0: MSB-first transfer 1: LSB-first transfer • When the transmission/reception is performed in the LIN format, set URTEnSLD to 1. • Changing this bit is only allowed if reception and transmission is disabled (URTEnCTL0.URTEnPW = 0 or URTEnCTL0.URTEnRXE = URTEnCTL0.URTEnTXE = 0).
0	URTEnSLIT	Transmission interrupt request (INTUAEnTIT) timing selection 0: INTUAEnTIT generated at the start of transmission, i.e. when the transmit data is stored to the transmission shift register 1: INTUAEnTIT generated at transmission completion • Changing this bit is only allowed if transmission is disabled (URTEnCTL0.URTEnPW = 0 or URTEnCTL0.URTEnTXE = 0).

(3) URTEnCTL2 - UARTEn control register 2

This register defines the baud rates of UARTEn serial data transfers.

Access This register can be read/written in 16-bit units.

Address <URTEn_base_OS> + 24_H

Initial Value $EFFF_H$. This register is initialized by any reset.

15	14	13	12	11	10	9	8
URTEnPRS[2:0]			0	URTEnBRS[11:8]			
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W

7	6	5	4	3	2	1	0
URTEnBRS[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 19-9 URTEnCTL2 register contents

Bit position	Bit name	Function
15 to 13	URTEn PRS[2:0]	Prescaler clock (PRSCLK) division value 0: PRSCLK = PCLK / 2^0 1: PRSCLK = PCLK / 2^1 2: PRSCLK = PCLK / 2^2 3: PRSCLK = PCLK / 2^3 4: PRSCLK = PCLK / 2^4 5: PRSCLK = PCLK / 2^5 6: PRSCLK = PCLK / 2^6 7: PRSCLK = PCLK / 2^7
11 to 0	URTEn BRS[11:0]	Baud rate clock (BRCLK) division value

URTEn BRS[11:0]	Transmit/receive BRCLK	BF receive clock
000_H	PCLK / (2 x 4)	PCLK / 4
001_H		
002_H		
003_H		
004_H		
005_H	PCLK / (2 x 5)	PCLK / 5
...	PCLK / (2 x URTEnBRS[11:0])	PCLK / URTEnBRS[11:0]
FFE_H	PCLK / (2 x 4094)	PCLK / 4094
FFF_H	PCLK / (2 x 4095)	PCLK / 4095

Caution Writing to this register is only allowed if the UARTEn operation is disabled (URTEnCTL0.URTEnPW = 0).

PCLK The value of the UARTEn input clock is defined in the first section of this chapter under the key word “Clock supply”.

(4) URTEnTRG - UARTEn trigger register

This register controls the UARTEn transmission/reception trigger of BF.

Access This register can be read/written in 8-bit and 1-bit units.

Address <URTEn_base_USER> + 04_H

Initial Value 00H. This register is initialized by any reset.

7	6	5	4	3	2	1	0
0	URTEnBRT	URTEnBTT	0	0	0	0	0
R	R/W	R/W	R	R	R	R	R

Table 19-10 URTEnTRG register contents

Bit position	Bit name	Function
6	URTEnBRT	BF reception trigger 0: Read value is always 0, writing 0 is ignored 1: Enable BF reception • When reception is enabled, writing 1 to this bit enables BF reception (URTEnSTR0.URTEnSSBR = 1) and BF reception processing begins when the falling edge of the receive serial signal is detected. • If 1 is written to this bit during reception processing, the current reception processing is terminated. Consequently, the received data is not stored, the framing, parity and overflow error bits are not updated based on the data that was being received and no interrupts are generated. Meanwhile, the BF counter value is continuously being used. • After BF reception, the reception status is set according to the URTEnCTL1.URTEnSLBM setting. • Setting this bit to 1 is only allowed if reception is enabled (URTEnCTL0.URTEnPW = URTEnCTL0.URTEnRXE = 1). After URTEnBRT is set to 1, completion of BF reception is reported by either of the following two methods, based on the URTEnCTL1.URTEnSLBM setting: • if URTEnCTL1.URTEnSLBM = 0: When BF reception is complete, a reception interrupt request INTUAEnTIR is generated. • if URTEnCTL1.URTEnSLBM = 1: When BF reception is complete, URTEnSTR1.URTEnBSF is set to 1 and the status interrupt request INTUAEnTIS is generated.
5	URTEnBTT	BF transmission trigger 0: Read value is always 0, writing 0 is ignored 1: Enable BF transmission • When this bit is set while URTEnSTR0.URTEnSSBT = 0 and transmission is enabled (URTEnDCE = 0), a BF transmit request is set, and URTEnSSBT is set to 1. • When this bit is set during data transmission, a BF is transmitted after the current transmission processing is completed. Even if this bit is set before the BF transmission is completed, a BF is transmitted only once. • When transmission is enabled (URTEnPW = URTEnTXE = 1), setting this bit clears all previously set data transmit requests (which have not been transmitted), leaving only BF transmit requests. If the URTEnTX7 to URTEnTX0 bits are written after setting this bit, data is transmitted after the BF is transmitted. • If both a BF transmit request and a data transmit request have been set when a transmission starts, the BF transmission takes priority. • When URTEnDCE = 1, writing 1 to this bit is ignored. • Setting this bit to 1 is only allowed if transmission is enabled (URTEnCTL0.URTEnPW = URTEnCTL0.URTEnTXE = 1).

(5) URTEnSTR0 - UARTEn status register 0

This register indicates the current status of serial data transmissions.

Access This register can be read in 8-bit or 1-bit units.
Writing to this register is only allowed if UARTEn operation is disabled (URTEnCTL0.URTEnPW = 0). If UARTEn operation is enabled (URTEnCTL0.URTEnPW = 1), any written values are disregarded and the initial values are restored.

Address <URTEn_base_USER> + 08_H

Initial Value 00_H. This register is initialized by any reset and when URTEnCTL0.URTEnPW is set or cleared.

7	6	5	4	3	2	1	0
0	URTEn SSBR[a]	URTEn SSBT[b]	0	0	0	URTEn SSR[a]	URTEn SST[b]
R	R	R	R	R	R	R	R

a) These bits are also initialized if reception is disabled by URTEnCTL0.URTEnRXE = 0.

b) These bits are also initialized if transmission is disabled by URTEnCTL0.URTEnTXE = 0.

Table 19-11 URTEnSTR0 register contents

Bit position	Bit name	Function
6	URTEnSSBR	BF reception enable status 0: BF reception is disabled 1: BF reception is enabled by setting URTEnTRG.URTEnBRT to 1 (BF reception standby mode or BF reception busy).
5	URTEnSSBT	BF transmission enable status 0: BF transmission is disabled 1: BF transmission is enabled by setting URTEnTRG.URTEnBTT to 1 (BF transmission standby mode or BF transmission busy).
1	URTEnSSR	Data reception status 0: No data reception ongoing 1: Data reception ongoing (data reception busy)
0	URTEnSST	Data transmission status 0: No transmission pending or ongoing 1: Data in URTEnTX[7:0] pending to be transmitted or transmission ongoing

(6) URTEnSTR1 - UARTEn status register 1

This register indicates results of serial data transmissions.

Access This register can be read in 8-bit and 1-bit units.
Writing to this register is only allowed if UARTEn operation is disabled (URTEnCTL0.URTEnPW = 0). If UARTEn operation is enabled (URTEnCTL0.URTEnPW = 1), any written values are disregarded and the initial values are restored.

Address <URTEn_base_USER> + $0C_H$

Initial Value 00_H. This register is initialized by any reset and when URTEnCTL0.URTEnPW is set or cleared.

7	6	5	4	3	2	1	0
0	0	0	URTEn BSF[a]	URTEn DCE[b]	URTEn PE[a]	URTEn FE[a]	URTEn OVE[a]
R	R	R	R	R	R	R	R

a) These bits are also initialized if reception is disabled by URTEnCTL0.URTEnRXE = 0.

b) This bit is also initialized if transmission is disabled by URTEnCTL0.URTEnTXE = 0.

Table 19-12 URTEnSTR1 register contents (1/2)

Bit position	Bit name	Function
4	URTEnBSF	BF reception successful flag 0: BF transmission is disabled by clearing URTEnTRG.URTEnBTT. 1: BF transmission is enabled by setting URTEnTRG.URTEnBTT (BF transmission standby mode or BF transmission busy). The URTEnBSF bit is cleared by the following: - URTEnCTL0.URTEnPW = 1 - URTEnCTL0.URTEnRXE = 0 - URTEnSTC.URTEnCLBS = 1
3	URTEnDCE	Data consistency error flag 0: Transmit/receive data (transmit/receive BF) mismatch was not detected. 1: Transmit/receive data (transmit/receive BF) mismatch was detected. When the BF receive mode selection bit is set during LIN communication, it is necessary to read this bit by using status interrupt processing and to confirm the beginning of a new frame slot. The URTEnDCE bit is cleared by the following: - URTEnCTL0.URTEnPW = 0 - URTEnCTL0.URTEnTXE = 0 - URTEnSTC.URTEnCLDC = 1
2	URTEnPE	Parity error flag 0: No parity error was detected in the received data. 1: A parity error was detected in the received data. The operation of URTEnPE is controlled by the settings of URTEn.URTEnSLP[1:0]. The URTEnPE bit is cleared by the following: - URTEnCTL0.URTEnPW = 0 - URTEnCTL0.URTEnRXE = 0 - URTEnSTC.URTEnCLP = 1
1	URTEnFE	Framing error flag 0: No framing error was detected in the received data. 1: A framing error was detected in the received data. The URTEnFE bit is cleared by the following: - URTEnCTL0.URTEnPW = 0 - URTEnCTL0.URTEnRXE = 0 - URTEnSTC.URTEnCLF = 1

Table 19-12 URTEnSTR1 register contents (2/2)

Bit position	Bit name	Function
0	URTEnOVE	Overrun error flag 0: No overrun error was detected in the received data. 1: An overrun error was detected in the received data. If an overrun error occurs, the data is discarded, and the next data received will not written to the receive data register URTEnRX. The URTEnOVE bit is cleared by the following: - URTEnCTL0.URTEnPW = 0 - URTEnCTL0.URTEnRXE = 0 - URTEnSTC.URTEnCLOV = 1

Note If the bits of these registers are set (1) and cleared (0) at the same time, setting takes priority over clearing.

For further information concerning error detections, refer to *19.6.5 "Transmission data consistency check"* and *19.6.9 "Reception errors"*.

Caution In case reception and transmission is enabled and a consistency check error occurs (URTEnSTR1.URTEnDCE = 1), follow the procedure below prior next data transmission:

- disable transmission by URTEnCTL0.URTEnTXE = 0
- enable transmission by URTEnCTL0.URTEnTXE = 1
- initiate a transmission by URTEnTRG.URTEnBTT = 1 (BT transmission trigger) or writing any data to URTEnTX

Afterwards new transmissions can be started.

(7) URTEnSTC - UARTEn status clear register

This register is used to clear the status bits of the status register 1 URTEnSTR1.

Access This register can be read/written in 8-bit and 1-bit units.
Reading this register returns always 00_H.

Address <URTEn_base_USER> $+10_H$

Initial Value 00_H. This register is initialized by any reset.

7	6	5	4	3	2	1	0
0	0	0	URTEn CLBS	URTEn CLDC	URTEn CLP	URTEn CLF	URTEn CLOV
R	R	R	R/W	R/W	R/W	R/W	R/W

Table 19-13 URTEnSTC register contents

Bit position	Bit name	Function
4	URTEnCLBS	Clear BF reception successful flag 0: Writing 0 is ignored 1: Writing 1 clears URTEnSTR1.URTEnBSF
3	URTEnCLDC	Clear data consistency error flag 0: Writing 0 is ignored 1: Writing 1 clears URTEnSTR1.URTEnDCE If URTEnDCE is cleared by setting URTEnCLDC, any pending data or BF transmit requests will be ignored.
2	URTEnCLP	Clear parity error flag 0: Writing 0 is ignored 1: Writing 1 clears URTEnSTR1.URTEnPE
1	URTEnCLF	Clear framing error flag 0: Writing 0 is ignored 1: Writing 1 clears URTEnSTR1.URTEnFE
0	URTEnCLOV	Clear overrun error flag 0: Writing 0 is ignored 1: Writing 1 clears URTEnSTR1.URTEnOVE

(8) URTEnRX - UARTEn receive data register

This register stores received data.

The data stored in the receive shift register is transferred to URTEnRX upon completion of reception of 1 byte of data.

7-bit transfers If the data length has been specified as 7 bits (URTEnCTL1.URTEnCLG = 0) and

- reception is LSB-first (URTEnCTL1.URTEnSLD = 1),:
  The receive data is transferred to URTEnRX[6:0] and the MSB URTEnRX[7] always becomes 0.
- reception is MSB-first (URTEnCTL1.URTEnSLD = 0),:
  The receive data is transferred to URTEnRX[7:1] and the LSB URTEnRX[0] always becomes 0.

For further information on data formats, refer to *19.6.1 "Data formats"*.

Overrun error When an overrun error (URTEnSTR1.URTEnOVE = 1) occurs, the receive data at this time is not transferred to URTEnRX and is discarded.

When reception processing ends and data reception is confirmed without any overrun errors, the received data is stored to URTEnRX according to the specified storage format.
Access

This register can be read in 8-bit units.
Writing to this register is only allowed if UARTEn operation is disabled (URTEnCTL0.URTEnPW = 0). If UARTEn operation is enabled (URTEnCTL0.URTEnPW = 1), any written values are disregarded and the initial values are restored.

Access This register can be read in 8-bit units.

Address <URTEn_base_USER> + 14_H

Initial Value FF_H. This register is initialized by any reset and when UARTEn operation is enabled by setting URTEnCTL0.URTEnPW.

7	6	5	4	3	2	1	0
URTEnRX[7:0]							
R	R	R	R	R	R	R	R

Table 19-14 URTEnRx register contents

Bit position	Bit name	Function
7 to 0	URTEn RX[7:0]	URTEn receive data

(9) URTEnTX - UARTEn transmit data register

This register is used to stores data to be transmitted.

Transmit data in URTEnTX is stored to the transmission shift register according to the specified transmit data format.

7-bit transfers If the data length has been specified as 7 bits (URTEnCTL1.URTEnCLG = 0) and

- transmission is LSB-first (URTEnCTL1.URTEnSLD = 1),: URTEnTX[6:0] is transferred to the shift register.
- transmission is MSB-first (URTEnCTL1.URTEnSLD = 0),: The LSB URTEnTX[0] is always set to “0” and URTEnTX[7:1] is transferred to the shift register.

For further information on data formats, refer to *19.6.1 "Data formats"*.

Access This register can be read/written in 8-bit units.

Address <URTEn_base_USER> + 18_H

Initial Value FF_H. This register is initialized by any reset.

7	6	5	4	3	2	1	0
URTEnTX[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 19-15 URTEnTx register contents

Bit position	Bit name	Function
7 to 0	URTEn TX[7:0]	URTEn transmit data

When transmission is enabled (URTEnCTL0 bits URTEnPW = URTETXE = 1), a write to URTEnTX starts transmission.

Note If the next data is written to URTEnTX before the current transmission is completed, continuous transmission can be enabled by waiting for the current data transmission to end before transmitting the next data.

19.5 Interrupt Request Signals

The following four interrupt request signals are generated by UARTEn.

- Transmission interrupt request INTUAEnTIT
- Reception interrupt request INTUAEnTIR
- Status interrupt request INTUAEnTIS
- Receive/status interrupt request INTUAEnTRA

19.5.1 Transmission interrupt request INTUAEnTIT

If transmit data is transferred from the URTEnTX register to transmit shift register with transmission enabled, the transmission interrupt request INTUAEnTIT is generated.

The condition for generation of a transmit interrupt request depends on the setting of the URTEnCTL1.URTEnSLIT:

- at start of transmission process: URTEnCTL1.URTEnSLIT = 0

  A transmission interrupt request is issued when starting transmission of the first bit (this is the start bit for data transmission or the first BF bit for BF transmission).

- at end of transmission process: URTEnCTL1.URTEnSLIT = 1

  A transmission interrupt request is issued after completing transmission of the last bit (the first bit of the stop bit when the stop bit length is 1, or the second bit of the stop bit when the stop bit length is 2).

Note If a data consistency error is detecting, this interrupt is not generated.

The following diagrams show the timing of the transmission interrupt request for both cases.

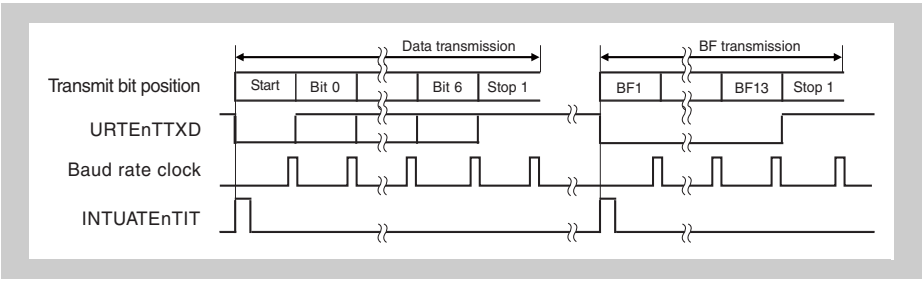


Figure 19-2 Transmission interrupt request timing for URTEnCTL1.URTEnSLIT = 0

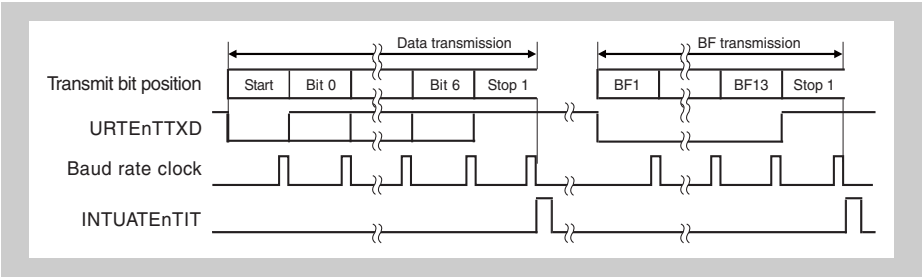


Figure 19-3 Transmission interrupt request timing for URTEnCTL1.URTEnSLIT = 1

19.5.2 Reception interrupt request INTUAEnTIR

A reception interrupt request is generated when the first bit of the stop bit is sampled.

In case of erroneous reception, the status interrupt INTUAEnTIS is generated instead of INTUAEnTIR.

The reception interrupt request INTUAEnTIR is not generated in the reception disabled status.

The following diagrams show the timing of the reception interrupt request during data/BF reception.

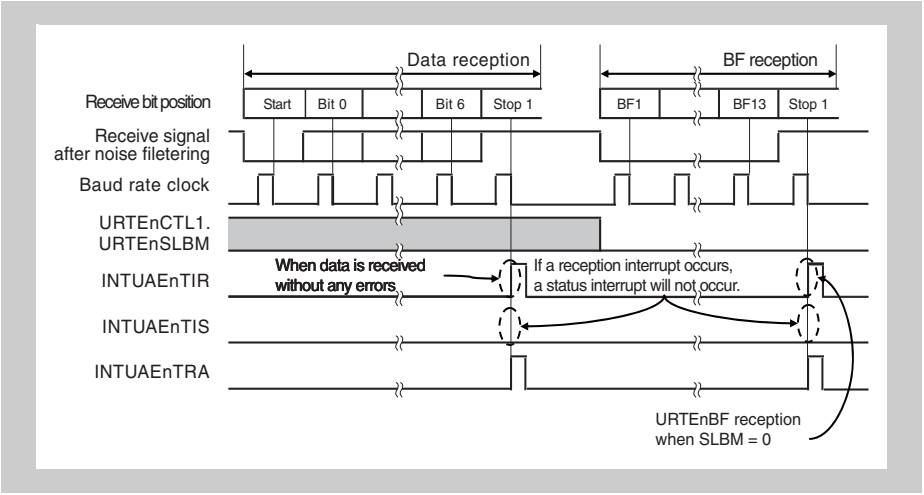


Figure 19-4 Reception interrupt request timing

19.5.3 Status interrupt request INTUAEnTIS

A status interrupt request is generated if an error condition occurred during reception or transmission, as reflected in the status register 1 URTEnSTR1.

If BF reception is enabled (URTEnCTL1.URTEnSLBM = 1) during LIN communication, the status interrupt request signal is generated when a consecutive low level (BF) of 11 bits or more is received.

19.5.4 Receive/status interrupt request INTUAEnTRA

The reception/status interrupt request is asserted, if a reception or status interrupt request is generated. That means:

INTUAEnTRA = INTUAEnTIR or INTUAEnTIS

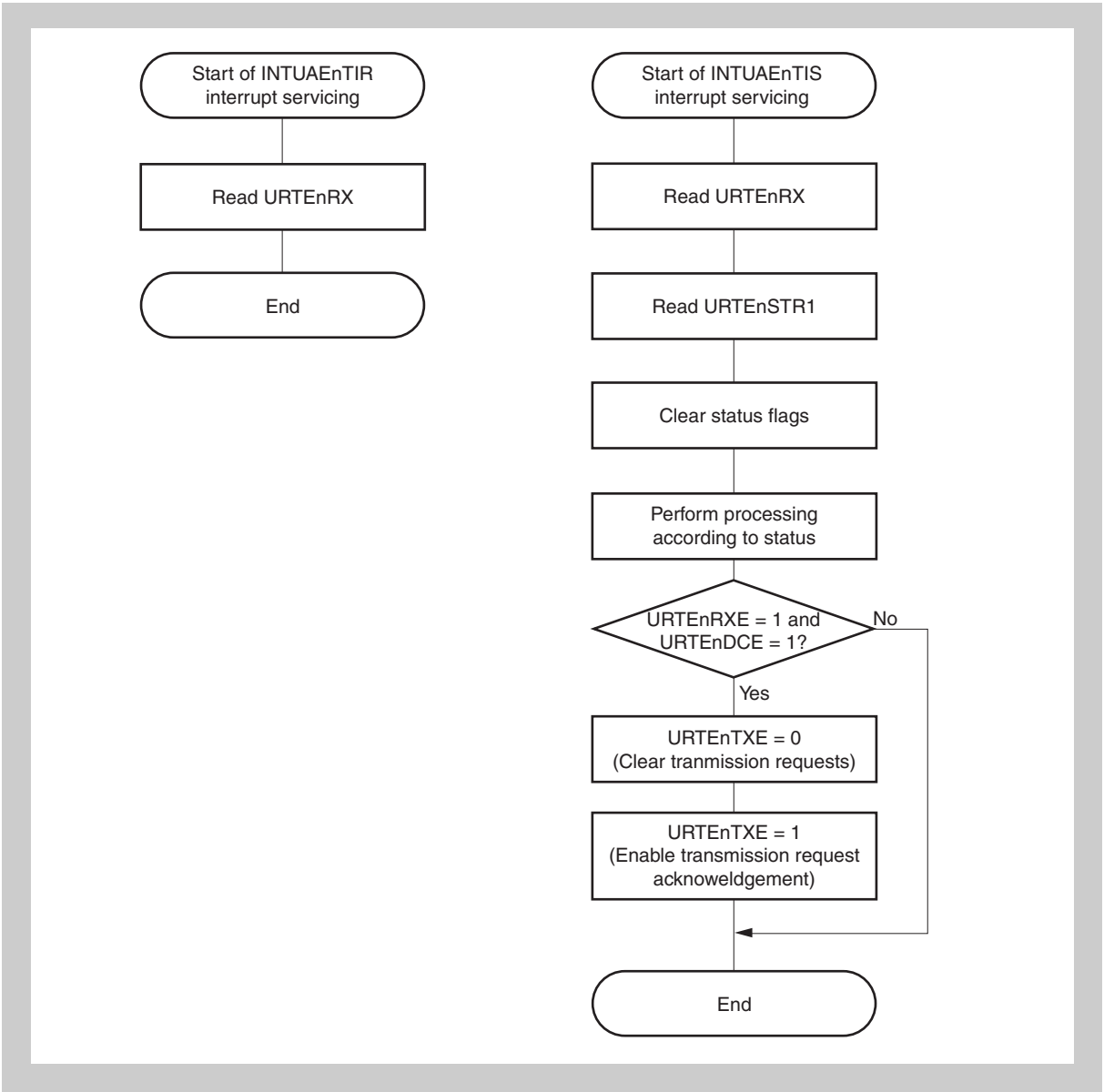


Figure 19-5 Processing flow after interrupt generation

19.6 Operation

19.6.1 Data formats

Full-duplex serial data reception and transmission is performed.

As shown in the figures below, one data frame of transmit/receive data consists of a start bit, character bits, parity bit, and stop bit(s).

Several properties of a transmit/receive data frame can be specified by control bits of the URTEnCTL1 register:

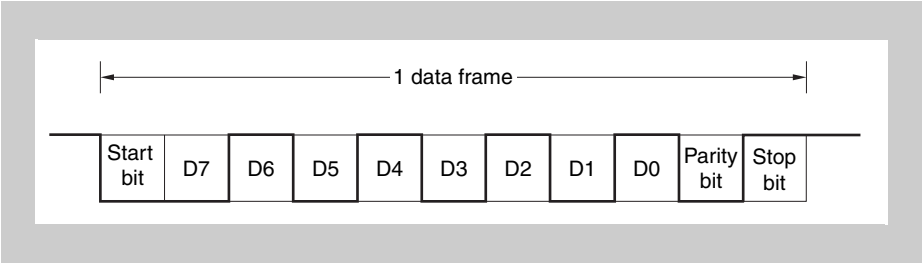
Table 19-16 Data format specification

Item	Options	Control bits
Start bit	1 bit	Fixed
Character bits	7 bits / 8 bits	URTEnCTL1.URTEnCLG
Parity	Even parity/odd parity/ 0 parity/no parity	URTEnCTL1.URTEnSLP[1:0]
Stop bit	1 bit / 2 bits	URTEnCTL1.URTEnSLG
Data order	MSB first / LSB first	URTEnCTL1.URTEnSLD
Tx data level	inverted / not inverted	URTEnCTL1.URTEnTDL
Rx data level	inverted / not inverted	URTEnCTL1.URTEnRDL

(1) UARTEn transmit/receive data format

(a) 8-bit data length, LSB first, even parity, 1 stop bit, transfer data: 55_H

(b) 8-bit data length, MSB first, even parity, 1 stop bit, transfer data: 55_H



(c) 8-bit data length, MSB first, even parity, 1 stop bit, transfer data: 55_H, URTEnTTXD inversion

(d) 7-bit data length, LSB first, odd parity, 2 stop bits, transfer data: 36_H

(e) 8-bit data length, LSB first, no parity, 1 stop bit, transfer data: 87_H

19.6.2 BF transmission/reception format

The UARTEn has an BF (Break Field) transmission/reception control function to enable use of the LIN function.

About LIN

LIN stands for Local Interconnect Network and is a low-speed (1 to 20 kbps) serial communication protocol intended to aid the cost reduction of an automotive network.

LIN communication is single-master communication, and up to 15 slaves can be connected to one master.

The LIN slaves are used to control the switches, actuators, and sensors, and these are connected to the LIN master via the LIN network.

Normally, the LIN master is connected to a network such as CAN (Controller Area Network).

In addition, the LIN bus uses a single-wire method and is connected to the nodes via a transceiver that complies with ISO9141.

In the LIN protocol, the master transmits a frame with baud rate information and the slave receives it and corrects the baud rate error. Therefore, communication is possible when the baud rate error in the slave is ±14% or less.

Figure 19-6 "LIN transmission outline" and *Figure 19-7 "LIN reception outline"* outline the transmission and reception manipulations of LIN.

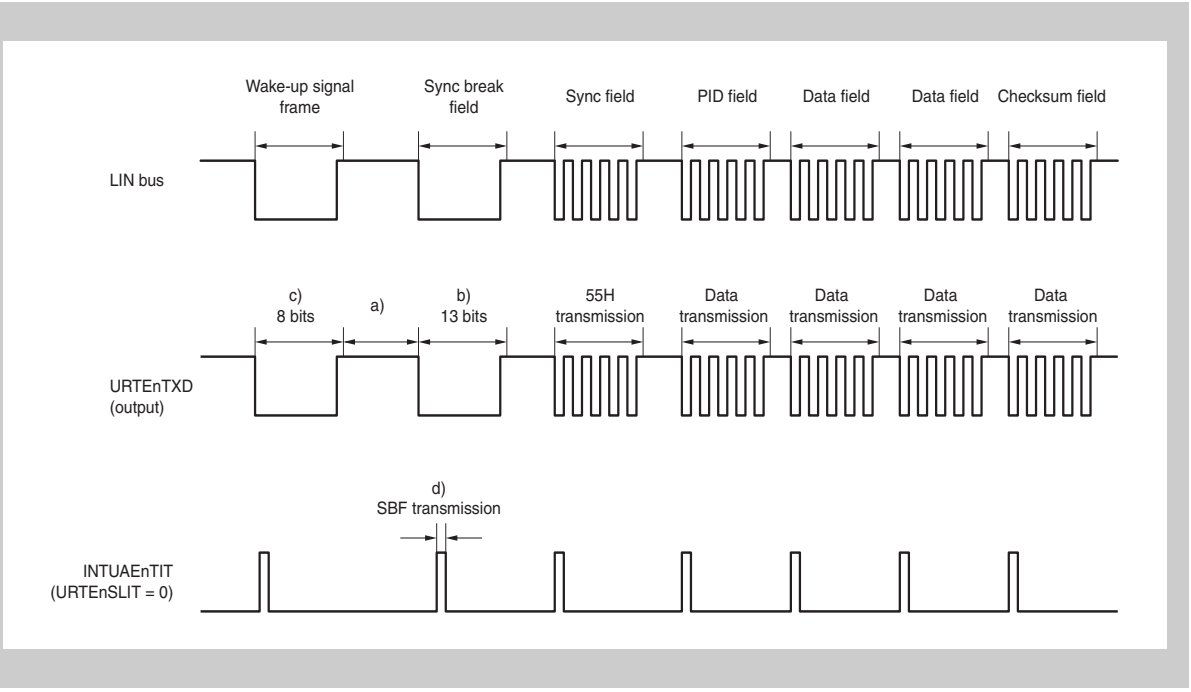


Figure 19-6 LIN transmission outline

a) The interval between fields is controlled by software.
b) BF output is performed by hardware. The output width is the bit length set by URTEnCTL1.URTEnBLG[2:0].
If even finer output width adjustments are required, such adjustments can be performed using URTEnCTLn.URTEnBRS[11:0].
c) 80_H transfer in the 8-bit mode is substituted for the wakeup signal frame.
d) A transmission enable interrupt INTUAEnTIT is generated at the start of each transmission. INTUAEnTIT is also generated at the start of each BF transmission.

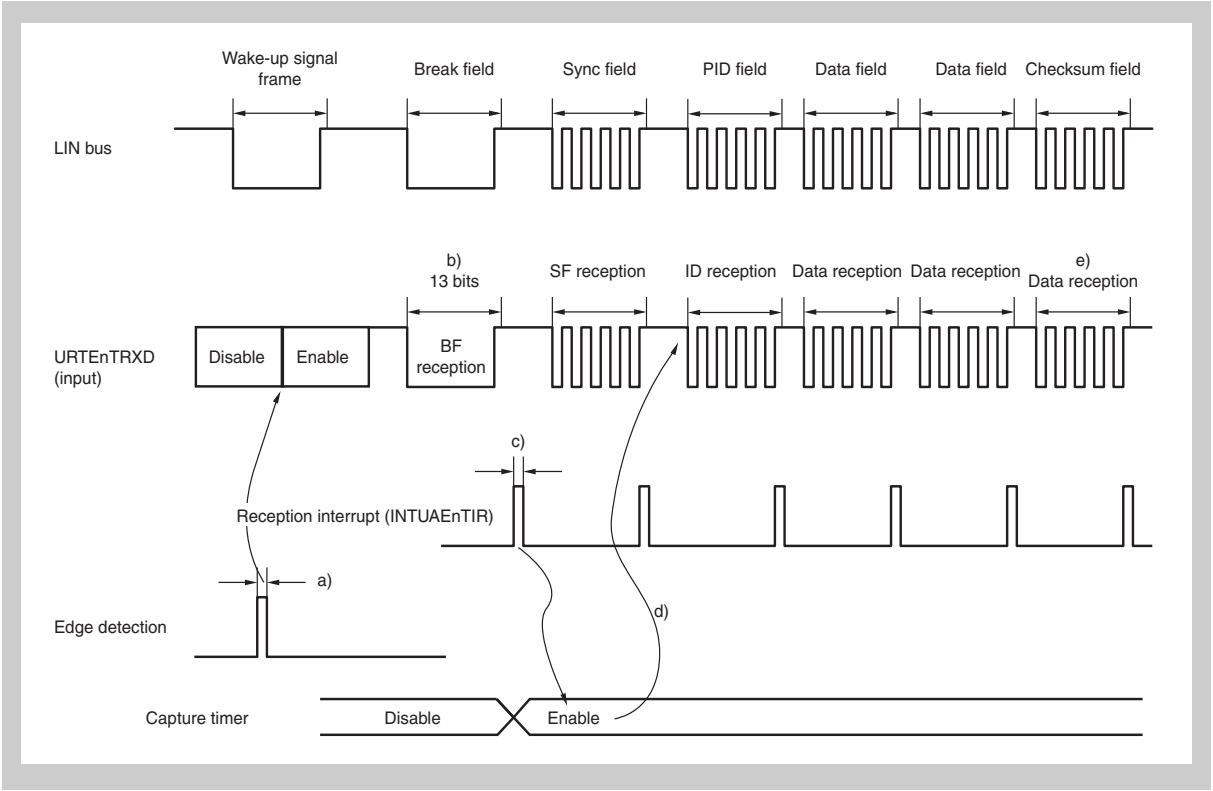


Figure 19-7 LIN reception outline

a) The wakeup signal sent by the pin edge detector enables UARTEn, and sets the BF reception mode.

b) BF reception is judged to end normally when a BF of 11 or more bits is received. An interrupt is generated as shown in the table below, according to the setting of the BF reception mode selection bit URTEnCTL1.URTEnSLBM and the value of the URTEnSTR0.URTEnSSBR bit.

URTEnSLBM	URTEnSSBR	Interrupts
1	x	INTUAEnTIS
0	1	INTUAEnTIR
0	0	A framing error has occurred, so and INTUAEnTIS is generated.

c) When BF reception ends normally, an interrupt is generated as follows according to the setting of the BF reception mode selection bit URTEnCTL1.URTEnSLBM:
- When URTEnDTL1.URTEnSLBM is 0, the reception interrupt INTUAEnTIR is generated.
- When URTEnDTL1.URTEnSLBM is 1, the status interrupt INTUAEnTIS is generated and the BF reception success flag URTEnSTR1.URTEnBSF is set.

If the BF reception trigger bit URTEnTRG.URTEnBRT is 1, error detection for overrun, parity, and framing errors is not performed during BF reception. Also, data transfer from the receive shift register to the receive data register URTEnRX is not performed. URTEnRX holds the previous value at this time.

d) In order to adjust the baud rate clock properly, the URTEnTRXD signal must be connected to the timer capture input. The transfer rate and the baud rate error can be calculated by measuring the time between URTEnTRXD edges, and the baud rate can be adjusted by specifying a value for the baud rate setting bits URTEnCTL2.URTEnBRS[11:0].

e) A checksum field is identified by software. When a checksum field is received, UARTEn is initialized and set to the BF reception mode by software. But if URTEnCTL1.URTEnSLBM is 1 at this time, UARTEn automatically starts BF reception without entering the BF reception mode.

19.6.3 BF transmission

When the URTEnCTL0 bits URTEnPW = URTEnTXE = 1, the transmission enabled status is entered, and BF transmission is started by setting the BF transmission trigger URTEnTRG.URTEnBTT = 1.

Thereafter, URTEnSTR0.URTEnSSBT is set to “1” and a low level width of 13 to 20 bits, as specified by URTEnCTL1.URTEnBLG[2:0], is output.
A transmission interrupt INTUAEnTIT) is generated upon BF

- transmission start, if URTEnCTL1.URTEnSLIT = 0
- transmission end, if URTEnCTL1.URTEnSLIT = 1.

Following the end of BF transmission, URTEnSTR0.URTEnSSBT is automatically cleared. Thereafter, the UARTEn transmission mode is restored.

Transmission is suspended until the data to transmit next is written to the URTEnTX register and URTEnSTR0.URTEnSST is set, or until the BF transmission trigger URTEnTRG.URTEnBTT is set and URTEnSTR0.URTEnSSBT changes to 1.

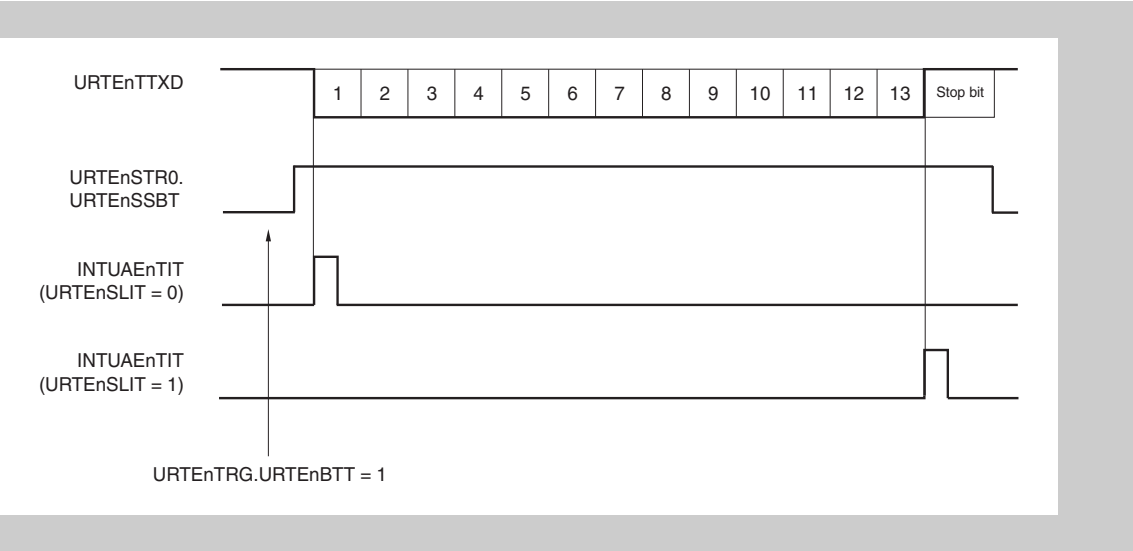
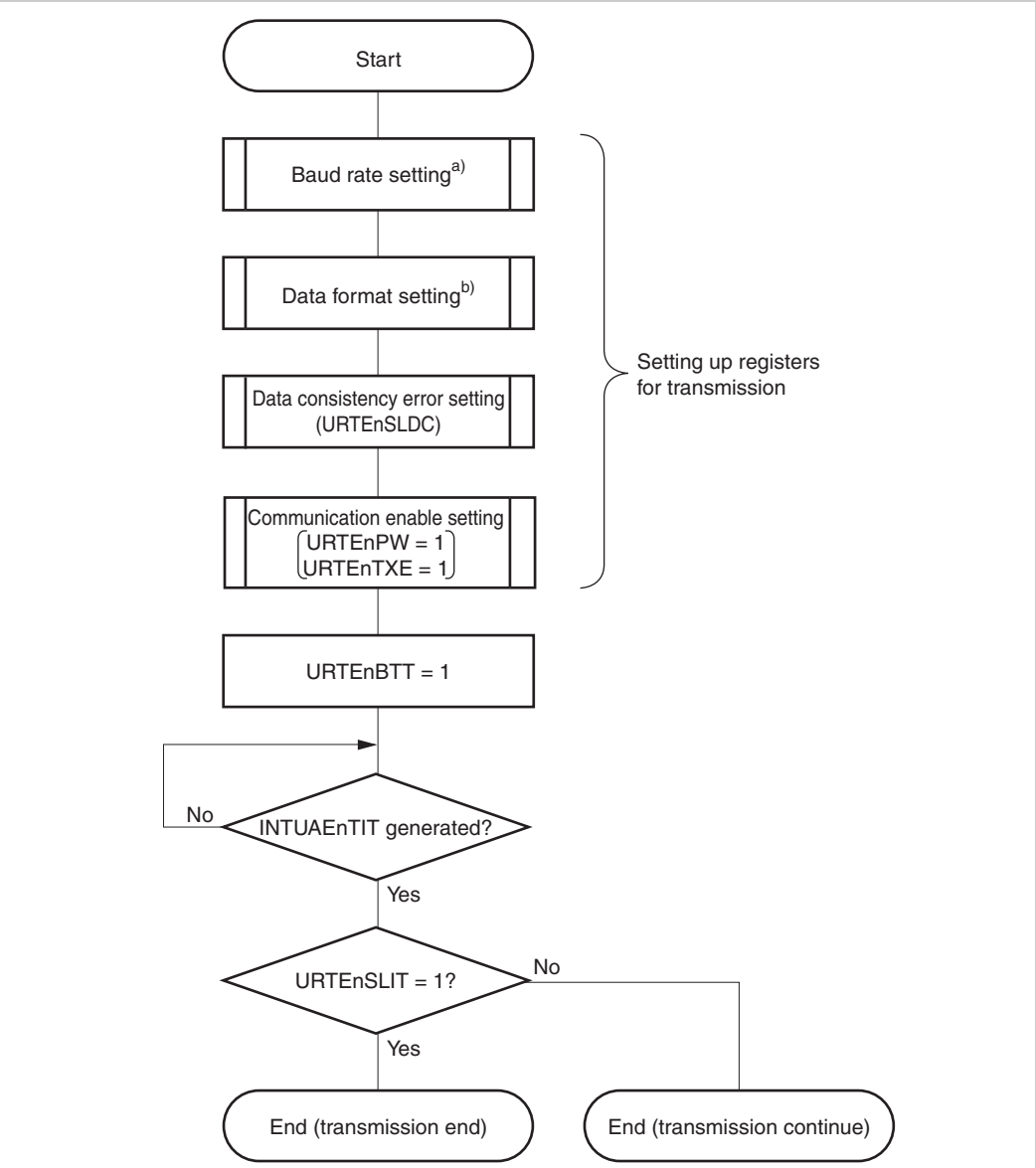


Figure 19-8 BF transmission



a) Set by using the URTEnPRS[2:0] and URTEnBRS[11:0] bits.

b) Set by using the URTEnBLG[2:0], URTEnCLG, URTEnSLP[1:0], URTEnTDL, URTEnSLG, URTEnSLD, and URTEnSLIT bits.

Figure 19-9 Flowchart of BF transmission

19.6.4 BF reception

The reception enabled status is achieved by setting the URTEnCTL0.URTEnPW bit to 1 and then setting the URTEnCTL0.URTEnRXE bit to 1.

The BF reception wait status is set by setting the BF reception trigger URTEnTRG.URTEnBRT = 1.

In the BF reception wait status, similarly to the UARTEn reception wait status, the URTEnTRXD pin is monitored and start bit detection is performed.

Following detection of the low level, reception is started and the internal counter counts up according to the set baud rate.

When a high level is received and if the BF width is 11 or more bits, while the BF receiving mode selection bit

- URTEnCTL1.URTEnSLBM = 0,
  the reception interrupt INTUAEnTIR is generated.
- URTEnCTL1.URTEnSLBM = 1,
  the status interrupt INTUAEnTIS is generated and BF reception success flag URTEnSTR1.URTEnBSF is set at the same time.

The URTEnSTR0.URTEnSSBR bit is automatically cleared and BF reception ends.

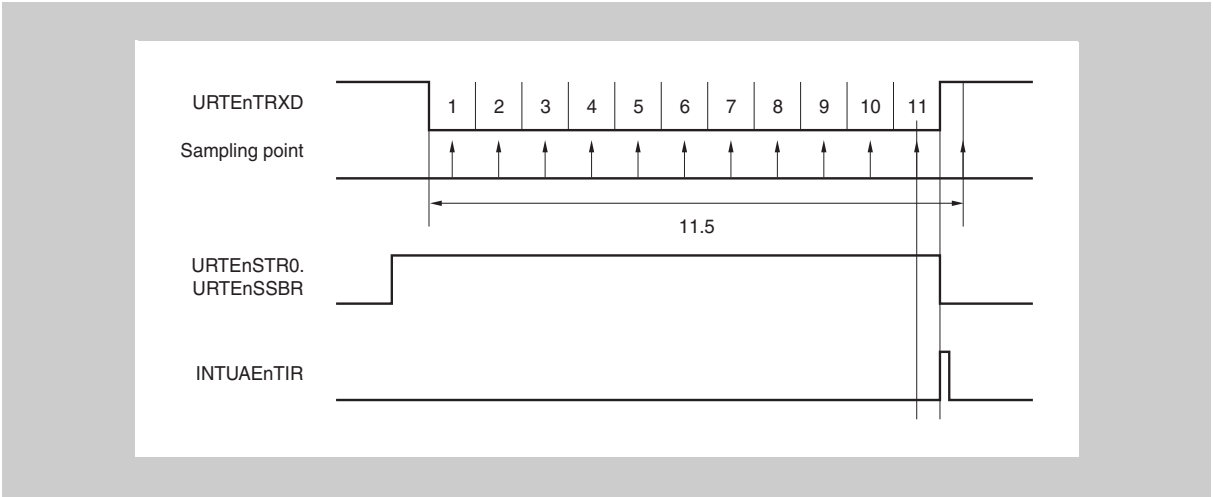


Figure 19-10 Normal BF reception (stop bit after more than 10.5 “L” bits)

Error detection for the URTEnSTR1 error flags URTEnOVE, URTEnPE, and URTEnFE is suppressed and UARTEn communication error detection processing is not performed.
Moreover, the erroneous data is not stored in URTEnRX, but the initial value FFH is held.

If the BF width is 10 or fewer bits, reception is terminated as error processing without generating an interrupt, and the BF reception mode is returned to. URTEnSTR0.URTEnSSBR is not cleared at this time.

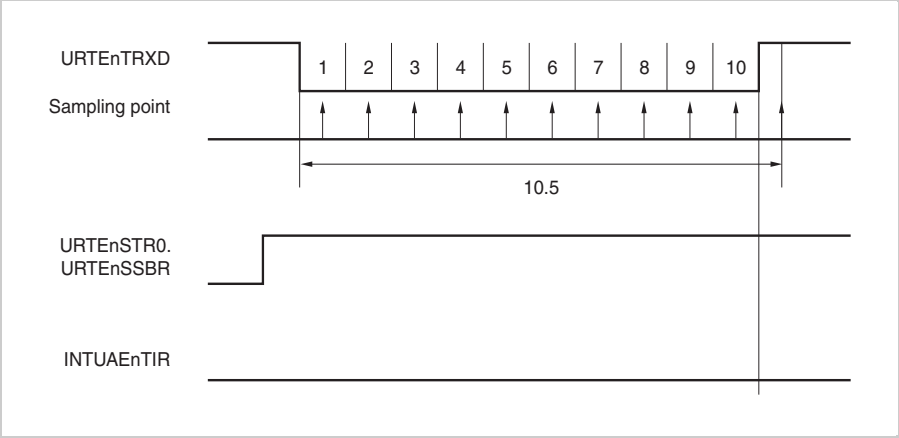


Figure 19-11 BF reception error (stop bit within 10.5 “L” bits)

The BF mode can be selected between a single BF receive mode and an any time BF receive mode in by URTEnCTL1.URTEnSLBM. The status of a successful reception of the BF is indicated by URTEnSTR1.URTEnBSF.

Note URTEnSTR0.URTEnSSBR is set to “1” when

- URTEnTRG.URTEnBRT is set to “1”, or
- the error is cleared by normal BF reception.

19.6.5 Transmission data consistency check

The UARTEn incorporates a data consistency check function to detect a mismatch between the transmit data written to transmit register URTEnTX and the data on the bus when the device operates in master mode.

Data consistency check is enabled by URTEnCTL0.URTEnSLDC = 1.

The data consistency is checked by comparing the transmit data in the transmit register URTEnTX and the receive data in the receive register URTEnRX. In case of a mismatch the data consistency error flag URTEnSTR1.URTEnDCE is set and a status interrupt request INTUAEnTIS occurs.

The consistency check of data is not done in reception mode.

The consistency check of the send data and the input data terminal level is done even if the reception is disabled during sending. In that case also the reception completion interrupt request signal INTUAEnTIR, the URTEnSTR1 status bits URTEnBSF, URTEnFE, URTEnOVE and the status interrupt request signal INTUAEnTIS will not be generated as well. Receive data does not need to be read.

Refer to *(6) "URTEnSTR1 - UARTEn status register 1" on page 1167* for details.

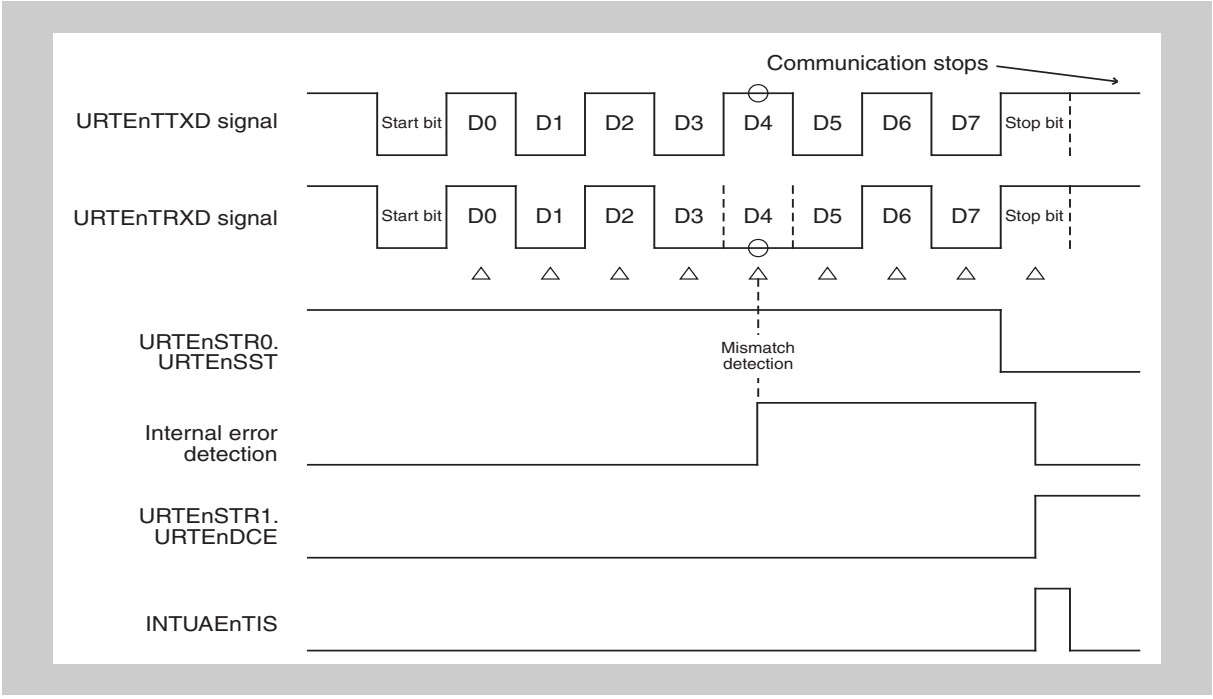


Figure 19-12 Timing example of data consistency error (no BF reception active, i.e. URTEnSTR0.URTEnSSBR = 0)

19.6.6 UARTEn transmission

Transmission start

Set the transmission enabled status by performing the following procedures.

- Specify the baud rate by UARTEn control register 2 (URTEnCTL2).
- Specify the transmit parity, data character length, stop bit length, transmit data order, transmission interrupt request timing and output logic level by UARTEn control register 1 (URTEnCTL1).
- Enable UARTEn operation and transmission by URTEnCTL0.URTEnPW = URTEnCTL0.URTEnTXE = 1)

Write of the transmit data to the transmission buffer register URTEnTX starts transmission. The data which is saved in the URTEnTX register is transferred to the transmit shift register URTEnTXS. Then, the start, parity and stop bits are added and the data frame is output serially via URTEnTTXD.

Transmission stop

When URTEnCTL0.URTEnPW or URTEnCTL0.URTEnTXE is set to 0, transmission operations are stopped immediately, even during transmission processing.

Concurrent BF and data transmission

When a BF transmit request and a data transmit request have both been set, BF transmission takes priority.

Data consistency check

If a data consistency error is detected, the subsequent data is not transmitted until URTEnCLDC is set to 1, URTEnPW is set to 0, or URTEnTXE is set to 0.

INTUAEnTIT timing

The time required to generate the transmission interrupt INTUAEnTIT depends on the setting of the URTEnCTL1.URTEnSLIT bit:

- URTEnCTL1.URTEnSLIT = 0
  INTUAEnTIT is generated at the start of transmission, i.e. when the data from the data register URTEnTX is transferred to the transmit shift register and transmission is started.
- URTEnCTL1.URTEnSLIT = 1
  INTUAEnTIT is generated when the entire data transmission process is completed, i.e. when the last bit of the data frame has been transmitted.

Once INTUAEnTIT is generated, the next data can be written to URTEnTX.

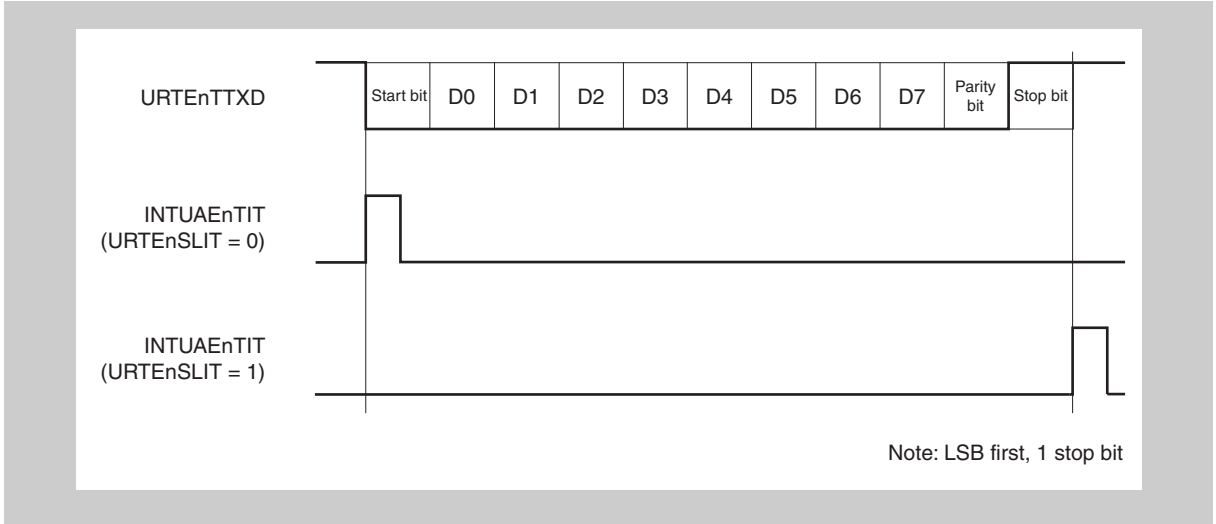
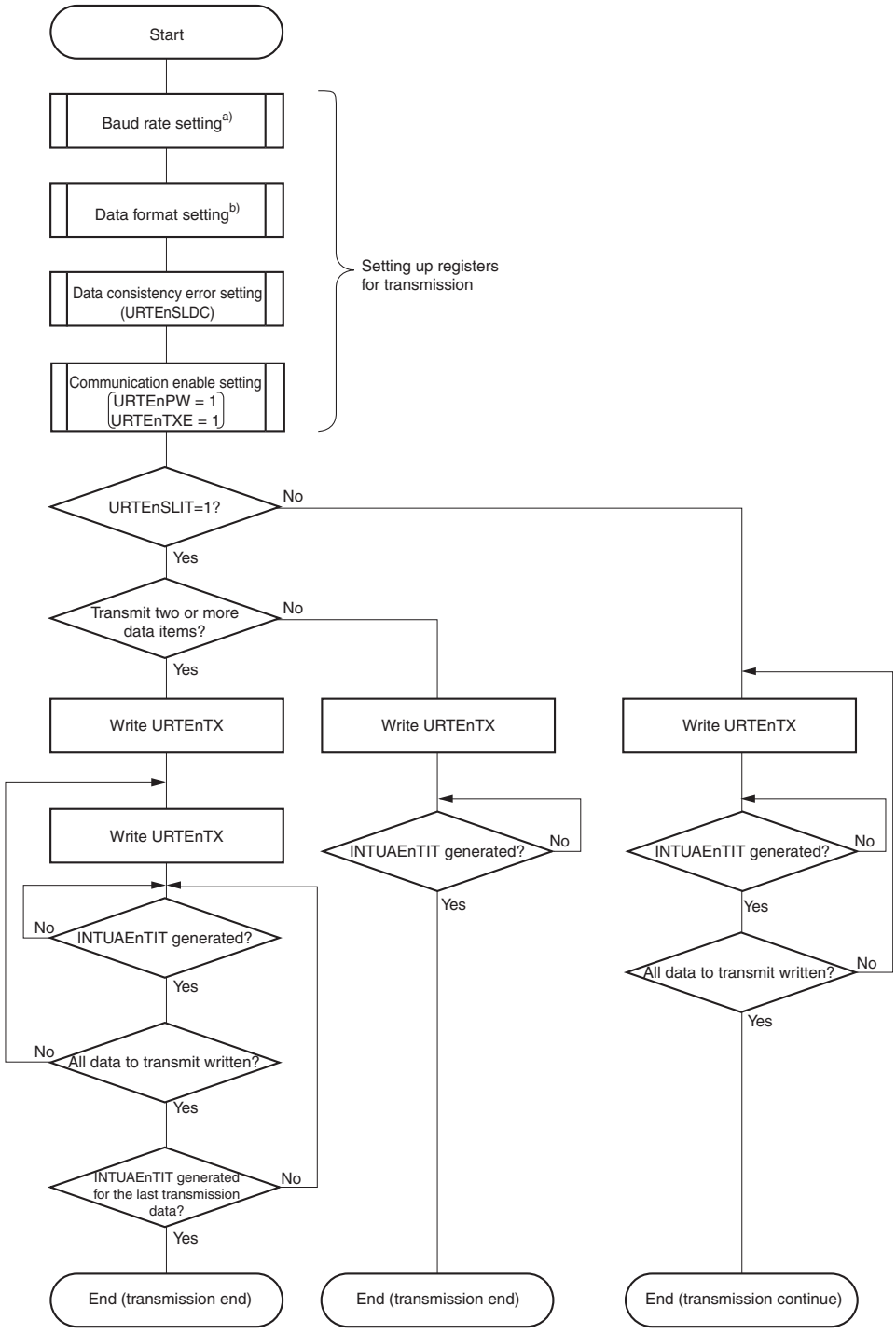


Figure 19-13 Transmission interrupt timing



a) Set by using the URTEnPRS[2:0] and URTEnBRS[11:0] bits.

b) Set by using the URTEnCLG, URTEnSLP[1:0], URTEnTDL, URTEnSLG, URTEnSLD, and URTEnSLIT bits.

Figure 19-14 Flowchart of data transmission

19.6.7 Continuous transmission procedure

Continuous transmission is achieved by writing the next transmit data to the transmit data register, while shifting out of the previous data from the transmission shift register URTEnTXS is ongoing.

Note In order to maintain correct write timing, the transmission interrupt INTUAEnTIT must be generated at the start of each transmission (URTEnCTL1.URTEnSLIT = 0).

Caution If the value is written to the URTEnTX register before the INTUAEnTIT is generated, the transmit data set before is overwritten by the new transmit data.

To initialize the transmission unit, confirm that no transmission is ongoing (URTEnSTR0 bits URTEnSSBT = URTEnSST = 0). If the initialization is performed during an ongoing transmission, the transmission is aborted.

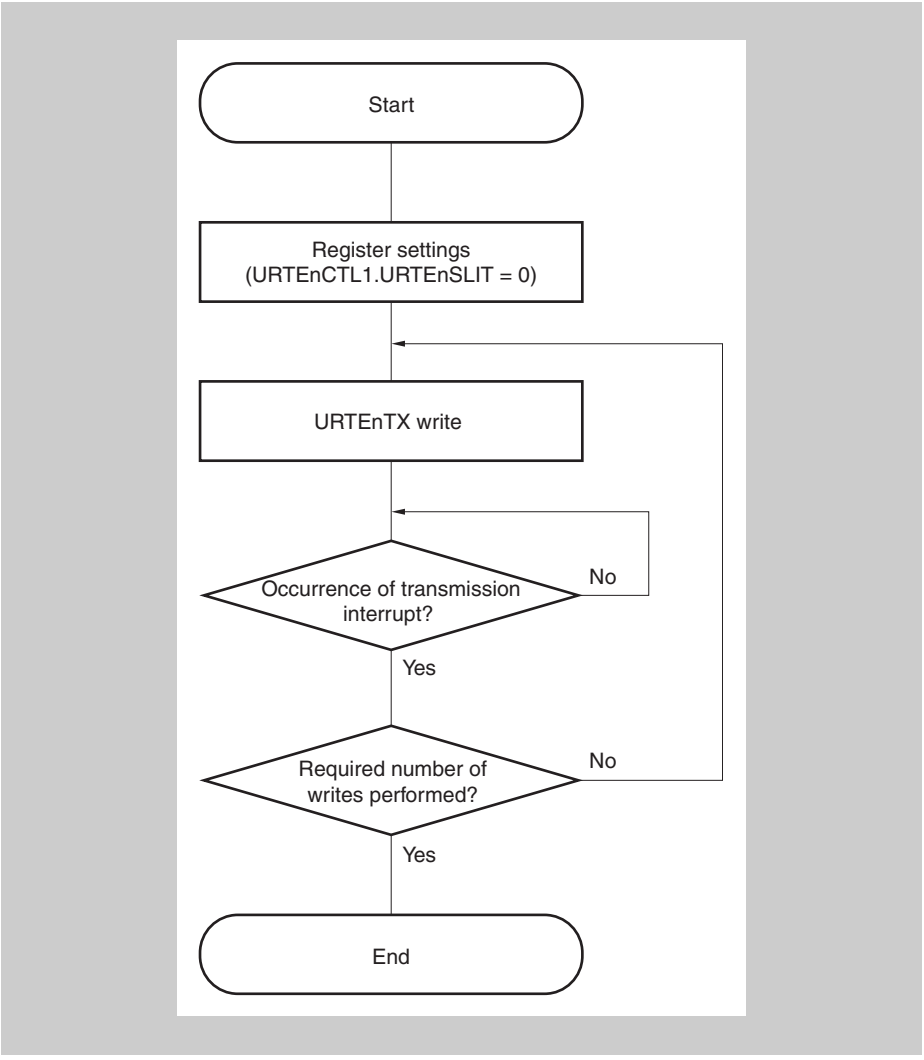


Figure 19-15 Continuous transmission processing flow

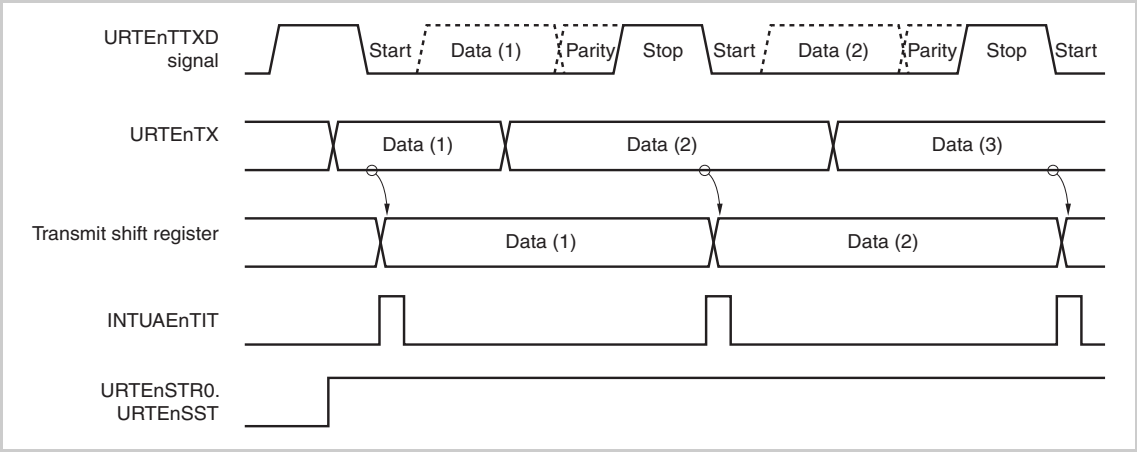


Figure 19-16 Continuous transmission operation timing - transmission start

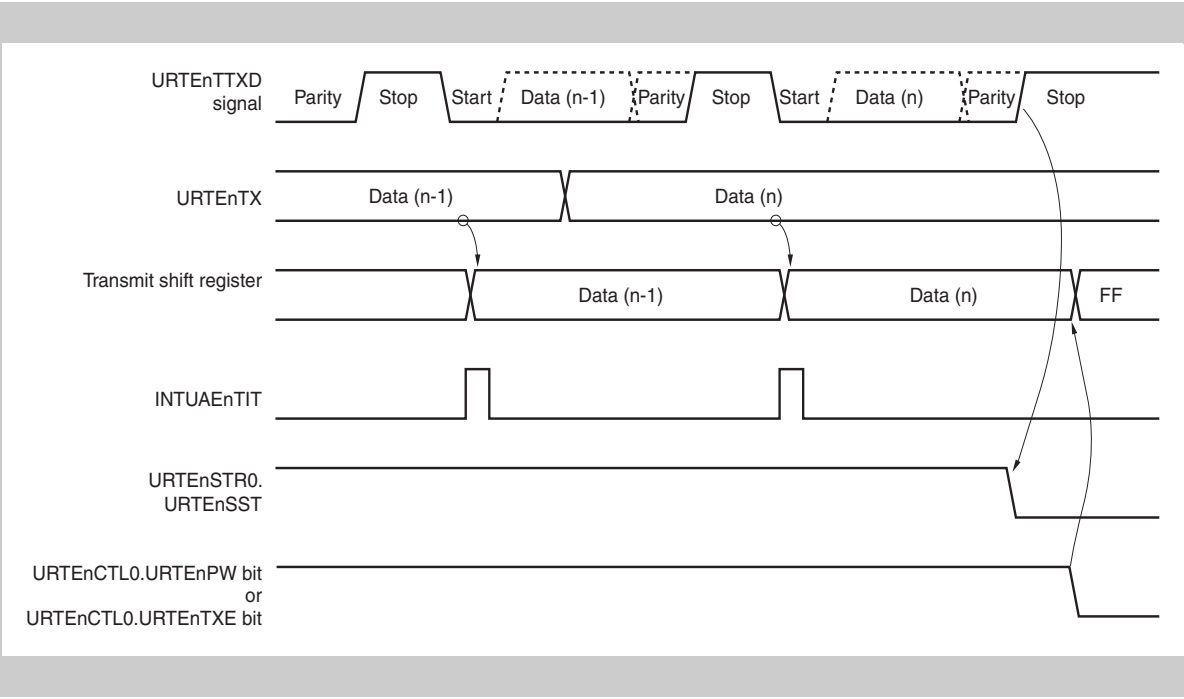


Figure 19-17 Continuous transmission operation timing - transmission end

19.6.8 UARTEn reception

Reception start

Set the reception enabled status by using the following procedure:

- Specify the baud rate by URTEnCTL2.
- Specify the receive parity, data character length, stop bit length, receive data order and output logic level by URTEnCTL1.
- Enable UARTEn operation and reception by setting URTEnCTL0.URTEnPW and URTEnCTL0.URTEnRXE.

When the falling edge of the URTEnTRXD pin input level is detected, data sampling of the URTEnTRXD input is started. The start bit is recognized if the URTEnTRXD pin is low level after the time of a half bit is passed after the detection of the falling edge (shown in the figure below). After a start bit has been recognized, the receive operation starts, and serial data is stored in the receive shift register according to the set baud rate. When the reception interrupt INTUAEnTIR is asserted upon reception of the stop bit, the data stored in the receive shift register is written to the receive data register URTEnRX.

Reception stop

When URTEnCTL0.URTEnPW or URTEnCTL0.URTEnRXE is set to 0, reception operations are stopped immediately, even during reception processing.

Reception errors

If an overrun error occurs (URTEnSTR1.URTEnOVE = 1), the receive data at this time is not transferred to the URTEnRX register and is discarded.
Even if a parity error (URTEnSTR1.URTEnPE = 1) or a framing error (URTEnSTR1.URTEnFE = 1) occurs during reception, reception continues until the reception position of the first stop bit, and the reception data is transferred to the URTEnRX.
When a reception error occurs, the status interrupt INTUAEnTIS –and therefore also the receive/status interrupt INTUAEnTRA– is generated, but not the reception interrupt INTUAEnTIR.

To change the receive data order, parity, data character length, or the stop bit length, clear the power bit (URTEnCTL0.URTEnPW = 0) first.

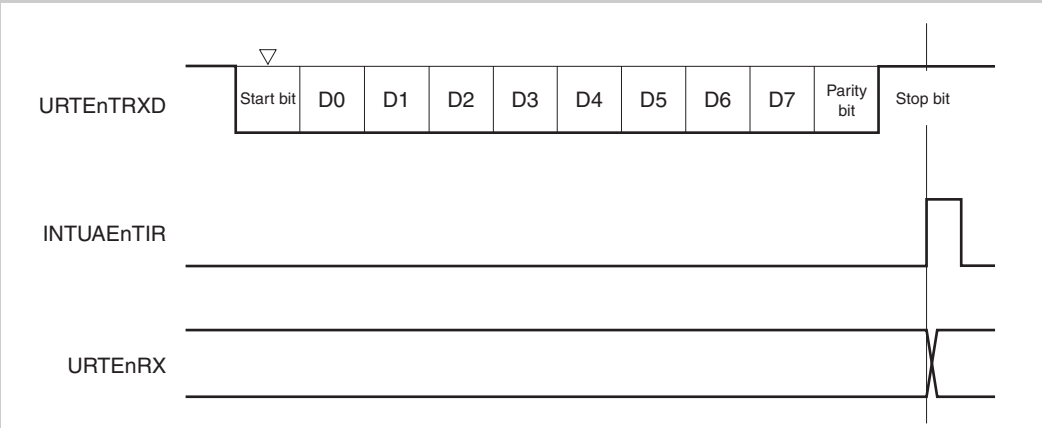


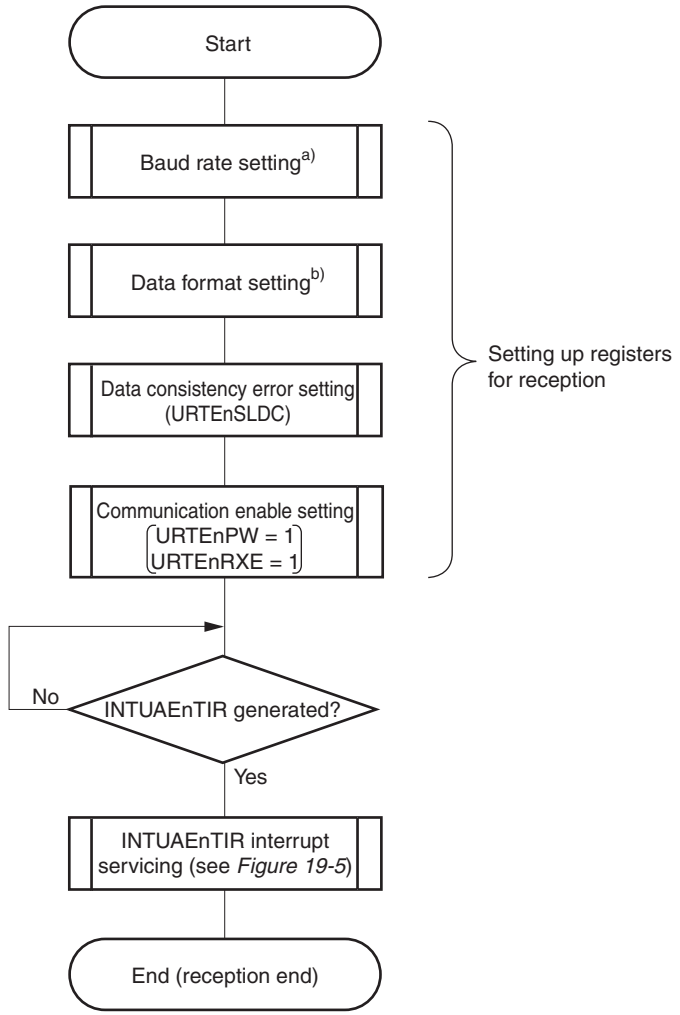
Figure 19-18 UARTEn reception

Cautions

1. Be sure to read the URTEnRX register even when a reception error occurs. If the URTEnRX register is not read, an overrun error occurs during reception of the next data.
2. The operation during reception is performed assuming that there is only one stop bit. A second stop bit is ignored.
3. When reception is completed, read the URTEnRX register after the reception interrupt INTUAEnTIR has been generated, and clear the URTEnCTL0.URTEnPW or URTEnCTL0.URTEnRXE bit to 0.
   If the URTEnCTL0.URTEnPW or URTEnCTL0.URTEnRXE bit is cleared to 0 before the INTUAEnTIR is generated, the read value of the URTEnRX register cannot be guaranteed.
4. If receive completion processing (INTUAEnTIR interrupt generation) and the URTEnCTL0.URTEnPW bit = 0 or URTEnCTL0.URTEnRXE bit = 0 conflict, INTUAEnTIR may be generated in spite of these being no data stored in the URTEnRX register.

Notes

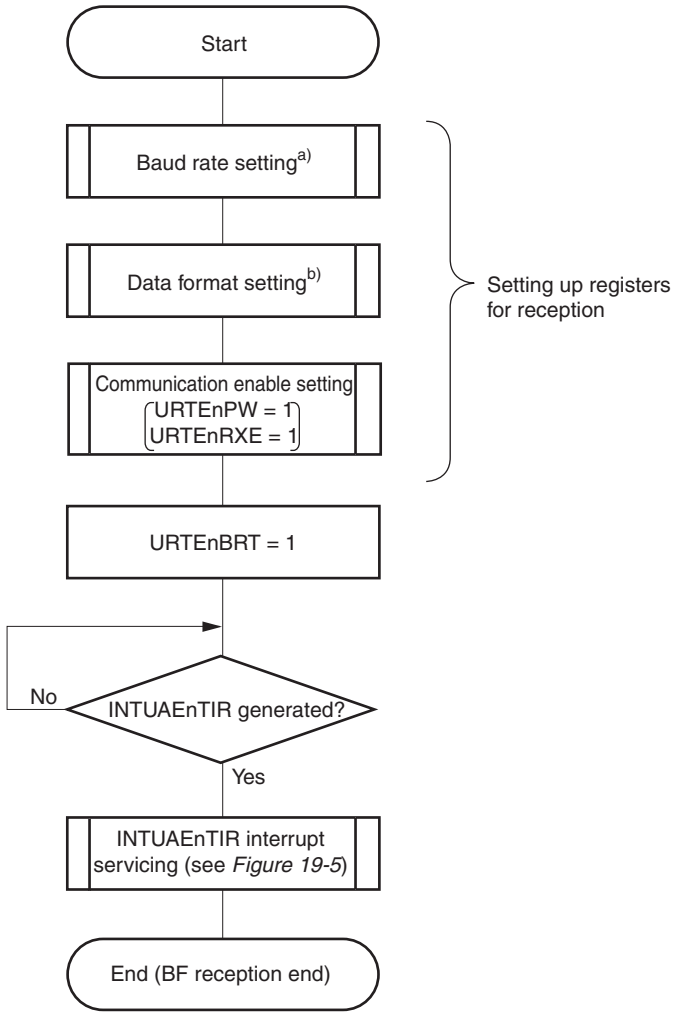
1. If low level is always input to the URTEnTRXD pin, it is not judged as the start bit.
2. In continuous reception, immediately after the stop bit is detected at the first reception bit (when the reception interrupt is generated), the next start bit can be detected.



a) Set by using the URTEnPRS[2:0] and URTEnBRS[11:0] bits.

b) Set by using the URTEnSLBM, URTEnCLG, URTEnSLP[1:0], URTEnRDL, and URTEnSLD bits.

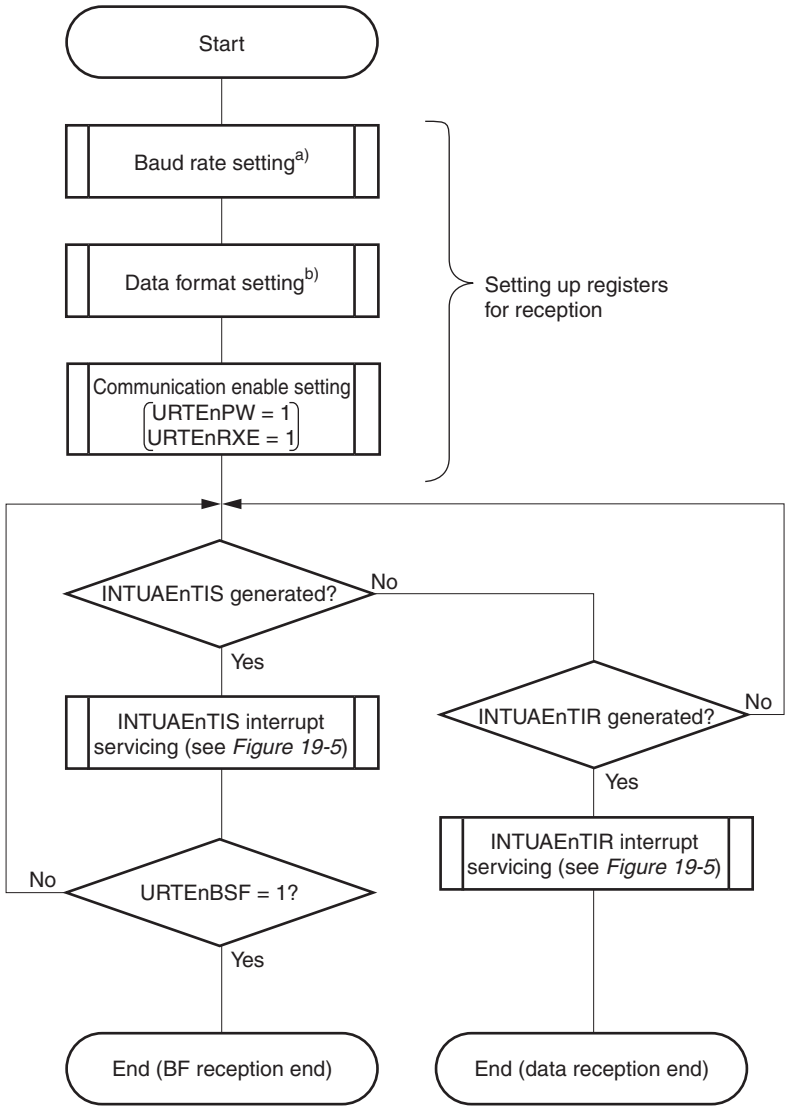
Figure 19-19 Flowchart of data reception when URTEnSLBM = 0, URTEnSSBR = 0



a) Set by using the URTEnPRS[2:0] and URTEnBRS[11:0] bits.

b) Set by using the URTEnSLBM, URTEnCLG, URTEnSLP[1:0], URTEnRDL, and URTEnSLD bits.

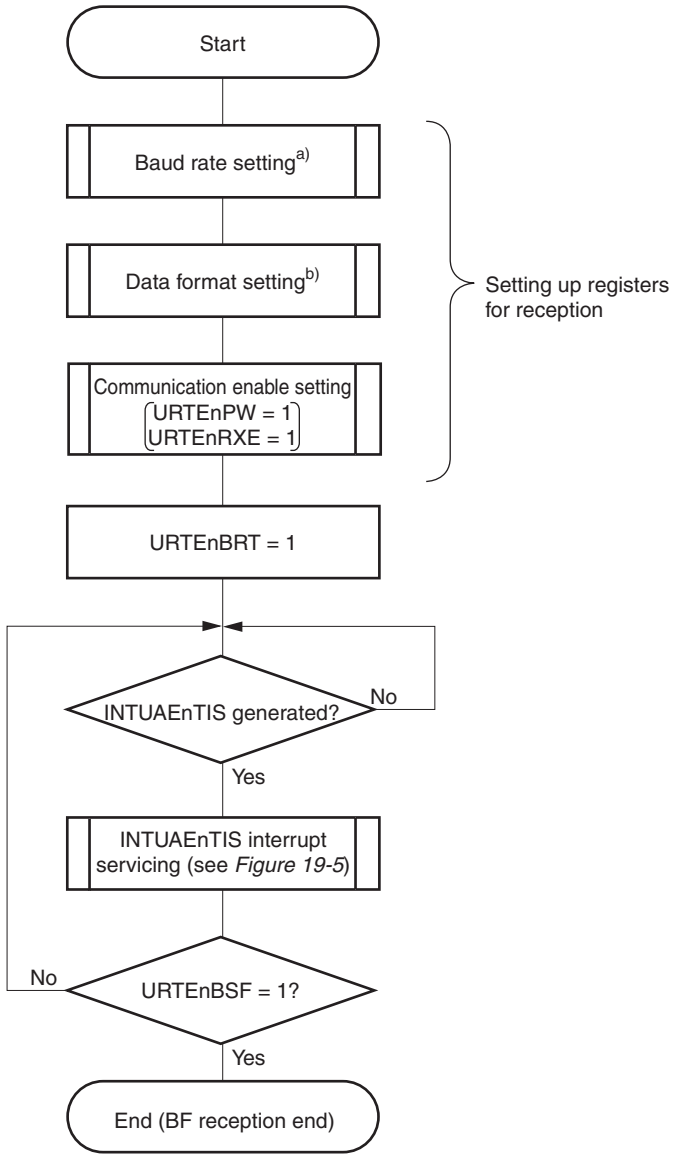
Figure 19-20 Flowchart of data reception when URTEnSLBM = 0, URTEnSSBR = 1



a) Set by using the URTEnPRS[2:0] and URTEnBRS[11:0] bits.

b) Set by using the URTEnSLBM, URTEnCLG, URTEnSLP[1:0], URTEnRDL, and URTEnSLD bits.

Figure 19-21 Flowchart of data reception when URTEnSLBM = 0, URTEnSSBR = 0



a) Set by using the URTEnPRS[2:0] and URTEnBRS[11:0] bits.

b) Set by using the URTEnSLBM, URTEnCLG, URTEnSLP[1:0], URTEnRDL, and URTEnSLD bits.

Figure 19-22 Flowchart of data reception when URTEnSLBM = 1, URTEnSSBR = 1

19.6.9 Reception errors

Errors during a receive operation are of three types: parity errors, framing errors, and overrun errors. Data reception result error flags are set in the URTEnSTR1 register and a status interrupt request signal INTUAEnTIS is generated when an error occurs.

It is possible to ascertain which error occurred during reception by reading the contents of the URTEnSTR1 register.

Clear a reception error flag by writing 1 to its associated bit in the status clear register URTEnSTC.

Table 19-17 Reception error causes

Error flag in URTEnSTR1	Reception error	Cause
URTEnPE	Parity error	Received parity bit does not match the setting
URTEnFE	Framing error	Stop bit not detected
URTEnOVE	Overrun error	Reception of next data completed before data was read from receive buffer

Note Even in case of a parity or framing error, data is transferred from the receive shift register to the receive data register URTEnRX. Consequently the data from URTEnRX must be read. Otherwise an overrun error URTEnSTR1.URTEnOVE will occur at reception of the next data.

In case of an overrun error, the receive shift register data is not transferred to URTEnRX, thus the previous data is not overwritten.

19.6.10 Parity types and operations

Caution When using the LIN function, fix the URTEnCTL1.URTEnSLP[1:0] to 00_B.

The parity bit is used to detect bit errors in the communication data. Normally the same parity is used on the transmission side and the reception side.

In the case of even parity and odd parity, it is possible to detect odd-count bit errors. In the case of 0 parity and no parity, errors cannot be detected.

(1) Even parity

- During transmission
  The number of bits whose value is “1” among the transmit data, including the parity bit, is controlled so as to be an even number. The parity bit values are as follows:
  - Odd number of bits whose value is “1” among transmit data:1
  - Even number of bits whose value is “1” among transmit data:0
- During reception
  The number of bits whose value is “1” among the reception data, including the parity bit, is counted, and if it is an odd number, a parity error is output.

(2) Odd parity

- During transmission
  Opposite to even parity, the number of bits whose value is “1” among the transmit data, including the parity bit, is controlled so that it is an odd number. The parity bit values are as follows.
  - Odd number of bits whose value is “1” among transmit data: 0
  - Even number of bits whose value is “1” among transmit data: 1
- During reception
  The number of bits whose value is “1” among the receive data, including the parity bit, is counted, and if it is an even number, a parity error is output.

(3) 0 parity

During transmission, the parity bit is always made 0, regardless of the transmit data.

During reception, parity bit check is not performed. Therefore, no parity error occurs, regardless of whether the parity bit is 0 or 1.

(4) No parity

No parity bit is added to the transmit data.

Reception is performed assuming that there is no parity bit. No parity error occurs since there is no parity bit.

19.6.11 Digital receive data noise filter

The receive data signal input URTEnTRXD is equipped with a digital noise filter to eliminate noise and spikes.

This filter samples the URTEnTRXD pin using the prescaler output clock PRSCLK.

When the same sampling value is read twice, the URTEnTRXD signal is validated as the input data.
Therefore, data not exceeding the width of 1 prescaler output clock cycle is judged to be noise and thus eliminated.

The noise filter causes a delay of 4 prescaler output clock PRSCLK cycles when capturing the serial data URTEnTRXD, until it is forwarded as valid.

19.7 Baud Rate Generator

The transmission and reception baud rate BRCLK are derived from the APB bus clock PCLK by use of a prescaler and a baud rate generator, as shown in the figure below.

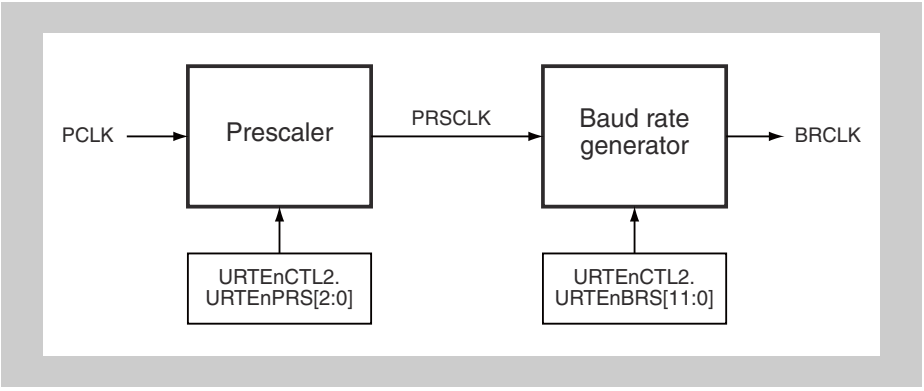


Figure 19-23 Configuration of baud rate generator

The prescaler output clock PRSCLK is a fraction of PCLK, the divisor is set up the value URTEnCTL2.URTEnPRS[2:0]:

$$PRSCLK = PCLK / 2^{URTEnPRS[2:0]}$$

PRSCLK is further divided by the baud rate generator by a value, determined by URTEnCTL2.URTEnBRS[11:0].
The baud rate generator distinguishes between the baud rate for data frames and BF receptions, as listed in the table below. The BF reception clock is the double of the baud rate clock BRCLK.

Table 19-18 Baud rate generator clocks output

URTEnCTL2. URTEnBRS[11:0]	Transmit/receive BRCLK	BF receive clock
000_H	PCLK / (2 x 4)	PCLK / 4
001_H		
002_H		
003_H		
004_H		
005_H	PCLK / (2 x 5)	PCLK / 5
...	PCLK / (2 x URTEnBRS[11:0])	PCLK / URTEnBRS[11:0]
FFE_H	PCLK / (2 x 4094)	PCLK / 4094
FFF_H	PCLK / (2 x 4095)	PCLK / 4095

Table 19-19 Example of baud rate generator settings

Baud rate	f_{XX} = 200 MHz			f_{XX} = 192 MHz			f_{XX} = 160 MHz			f_{XX} = 144 MHz		
	f_{PCLK} = 66.667 MHz			f_{PCLK} = 64.000 MHz			f_{PCLK} = 53.333 MHz			f_{PCLK} = 48.000 MHz		
	URTEnPRS [2:0]	URTEnBRS [11:0]	ERR [%]	URTEnPRS [2:0]	URTEnBRS [11:0]	ERR [%]	URTEnPRS [2:0]	URTEnBRS [11:0]	ERR [%]	URTEnPRS [2:0]	URTEnBRS [11:0]	ERR [%]
300	5	3472	0.01	5	3333	0.01	5	2778	–0.01	5	2500	0.00
600	4	3472	0.01	4	3333	0.01	4	2778	–0.01	4	2500	0.00
1200	3	3472	0.01	3	3333	0.01	3	2778	–0.01	3	2500	0.00
2400	2	3472	0.01	2	3333	0.01	2	2778	–0.01	2	2500	0.00
4800	1	3472	0.01	1	3333	0.01	1	2778	–0.01	1	2500	0.00
9600	0	3472	0.01	0	3333	0.01	0	2778	–0.01	0	2500	0.00
14400	0	2315	–0.01	0	2222	0.01	0	1852	–0.01	0	1667	–0.02
19200	0	1736	0.01	0	1667	–0.02	0	1389	–0.01	0	1250	0.00
31250	0	1067	–0.03	0	1024	0.00	0	853	0.04	0	768	0.00
38400	0	868	0.01	0	833	0.04	0	694	0.06	0	625	0.00
57600	0	579	–0.05	0	556	–0.08	0	463	–0.01	0	417	–0.08
76800	0	434	0.01	0	417	–0.08	0	347	0.06	0	313	–0.16
115200	0	289	0.12	0	278	–0.08	0	231	0.21	0	208	0.16
153600	0	217	0.01	0	208	0.16	0	174	–0.22	0	156	0.16
312500	0	107	–0.31	0	102	0.39	0	85	0.39	0	77	–0.26
1000000	0	33	1.01	0	32	0.00	0	27	–1.23	0	24	0.00
2000000	0	17	–1.96	0	16	0.00	0	13	2.56	0	12	0.00
3000000	0	11	1.01	0	11	–3.03	0	9	–1.23	0	8	0.00
4000000	–	–	–	0	8	0.00	–	–	–	0	6	0.00
6000000	–	–	–	–	–	–	–	–	–	0	4	0.00
8000000	–	–	–	0	4	0.00	–	–	–	–	–	–
8333333	0	4	0.00	–	–	–	–	–	–	–	–	–

Chapter 20 Asynchronous Serial Interface J (UARTJn)

This chapter contains a generic description of asynchronous serial interface J.

The first section describes all properties specific to the V850E2/MN4, such as instances, register base addresses, and input/output signal names.
The subsequent sections describe the features that apply to all implementations.

20.1 V850E2/MN4 UARTJn Features

Instances

This microcontroller has four instances of asynchronous serial interface J UARTJn.

Table 20-1 Instances of UARTJn

Asynchronous serial interface J	
Instance	4
Name	UARTJ0 to UARTJ3

Instances index n

Throughout this chapter, the instance of UARTJ is identified by the index "n" (n = 0 to 3), for example, UARTJnCTL0 for UARTJn control register 0.

Caution UARTJ of the V850E2/MN4 does not support the LIN function.

Register addresses

All UARTJn register addresses are given as addresses offset from the individual base address <URTJn_base_OS> or <URTJn_base_USER>. The <URTJn_base_OS> and <URTJn_base_USER> addresses of each UARTJn are listed in the following table:

Table 20-2 Register base addresses <URTJn_base>

UARTJn instance	Base address	Address
UARTJ0	<URTJn_base_OS>	FF5C 0000$_H$
	<URTJn_base_USER>	FFFF EA00$_H$
UARTJ1	<URTJn_base_OS>	FF5D 0000$_H$
	<URTJn_base_USER>	FFFF EB00$_H$
UARTJ2	<URTJn_base_OS>	FF5E 0000$_H$
	<URTJn_base_USER>	FFFF EC00$_H$
UARTJ3	<URTJn_base_OS>	FF5F 0000$_H$
	<URTJn_base_USER>	FFFF ED00$_H$

Clock supply All UARTJn provide one clock input. It is connected to the P bus clock f_{PCLK}.

Table 20-3 UARTJn clock supply

UARTJn instance	UARTJn clock	Connected to
UARTJ0 to UARTJ3	PCLK	f_{PCLK}

I/O signals The I/O signals of the asynchronous serial interface J are listed in the table below.

Table 20-4 UARTJn I/O signals

UARTJn signals	Function	Connected to
URTJnTTXD	Transmit data output	Port TXDnF
URTJnTRXD	Receive data input	Port RXDnF

Interrupts The interrupts of the asynchronous serial interface J are listed in the table below.

Table 20-5 UARTJn interrupts

UARTJn signals	Function	Connected to
UARTJ0		
INTUAJ0TIT	Transmission interrupt	• Interrupt controller 145 (INTCSIH0IC) • DTS controller trigger 109
INTUAJ0TIR	Reception interrupt	• Interrupt controller 144 (INTCSIH0IR) • DTS controller trigger 108
INTUAJ0TIS	Status interrupt	• Interrupt controller 143 (INTCSIH0IRE)
UARTJ1		
INTUAJ1TIT	Transmission interrupt	• Interrupt controller 149 (INTCSIH1IC) • DTS controller trigger 112
INTUAJ1TIR	Reception interrupt	• Interrupt controller 148 (INTCSIH1IR) • DTS controller trigger 111
INTUAJ1TIS	Status interrupt	• Interrupt controller 147 (INTCSIH1IRE)
UARTJ2		
INTUAJ2TIT	Transmission interrupt	• Interrupt controller 153 (INTCSIH2IC) • DTS controller trigger 115
INTUAJ2TIR	Reception interrupt	• Interrupt controller 152 (INTCSIH2IR) • DTS controller trigger 114
INTUAJ2TIS	Status interrupt	• Interrupt controller 151(INTCSIH2IRE)
UARTJ3		
INTUAJ3TIT	Transmission interrupt	• Interrupt controller 157 (INTCSIH3IC) • DTS controller trigger 118
INTUAJ3TIR	Reception interrupt	• Interrupt controller 156 (INTCSIH3IR) • DTS controller trigger 117
INTUAJ3TIS	Status interrupt	• Interrupt controller 155 (INTCSIH3IRE)

20.2 Functional Overview

- Full-duplex communication via built-in receive and transmit FIFOs:
  - Internal UARTJn 10 bit x 16 receive data FIFO (URTJnFRX)
  - Internal UARTJn 8 bit x 16 transmit data FIFO (URTJnFTX)
- 2-pin configuration:
  - URTJnTTXD: Transmit data output pin
  - URTJnTRXD: Receive data input pin
- Various error detection functions
  - Rx parity error
  - Rx framing error
  - Tx data consistency error
- Tx FIFO overflow error
  - Rx FIFO overrun error
  - Rx timeout error
  - Rx BF receive error
- Various FIFO status information
  - Rx FIFO full/empty status
  - Tx FIFO empty/empty status
  - Rx FIFO fill level
  - Tx FIFO fill level
- Interrupt requests: 3
  - Transmission interrupt INTUAJnTIT
  - Reception interrupt INTUAJnTIR
  - Status interrupt INTUAJnTIS
- Character length: 7, 8 bits
- Parity function: odd, even, 0, none
- Transmission stop bit: 1, 2 bits
- MSB-/LSB-first transfer selectable
- Transmit/receive data inverted input/output possible
- 13 to 20 bits selectable for the BF (Break Field) in the LIN (Local Interconnect Network) communication format
  - Recognition of 11 bits or more possible for BF reception in LIN communication format
  - BF reception flag provided
- BF reception can be detected during data communication
- Bus monitor function to keep data consistency of the transmit data

20.3 Configuration

The block diagram shows the main components of UARTJn.

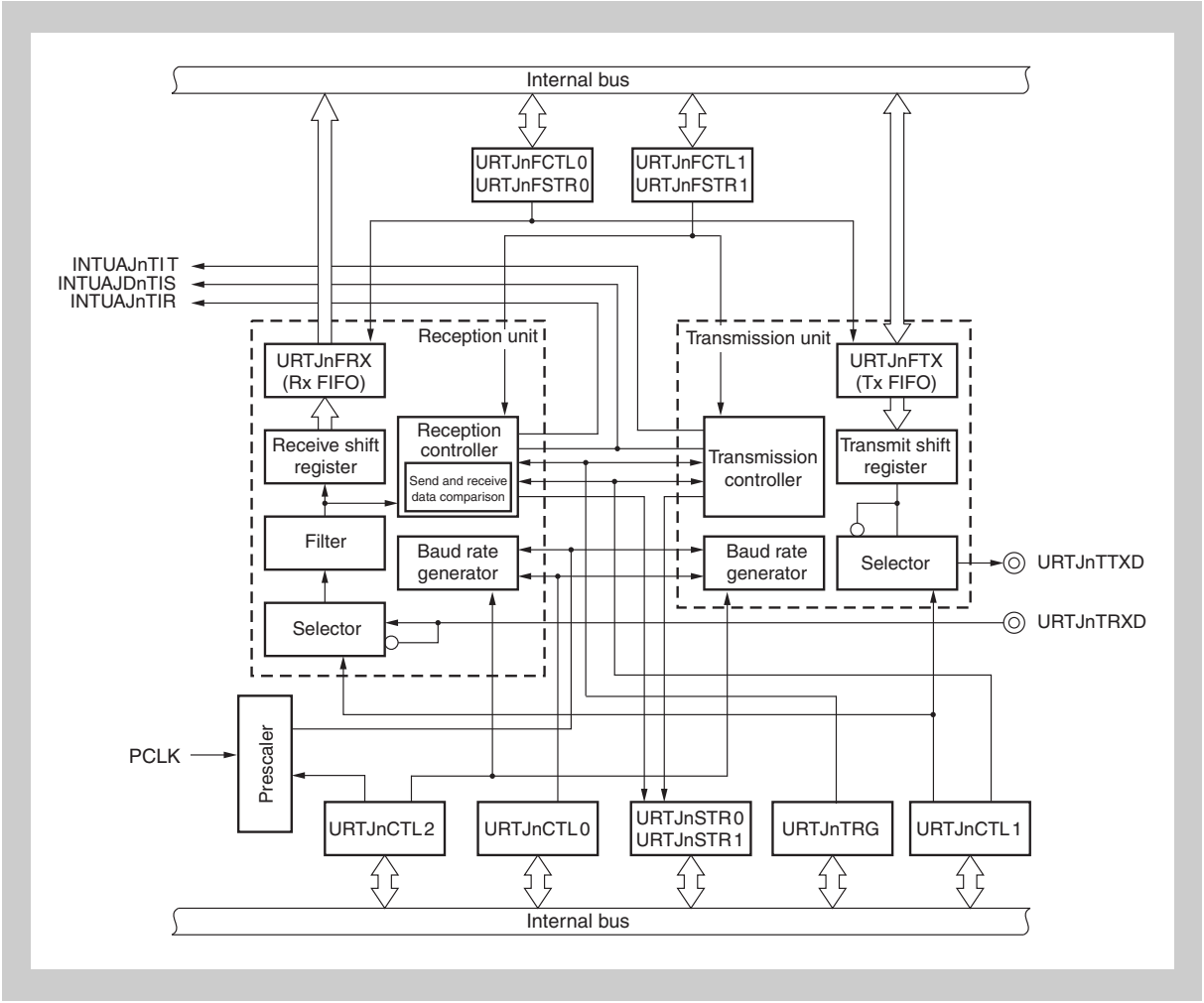


Figure 20-1 Block diagram of asynchronous serial interface UARTJn

20.4 UARTJn Registers

UARTJn is controlled and operated by means of the following registers:

Table 20-6 UARTJn registers

Register name	Symbol	Address
Control register 0	URTJnCTL0	<URTJn_base_USER> + 00_H
Control register 1	URTJnCTL1	<URTJn_base_OS> + 20_H
Control register 2	URTJnCTL2	<URTJn_base_OS> + 24_H
Trigger register	URTJnTRG	<URTJn_base_USER> + 04_H
Status register 0	URTJnSTR0	<URTJn_base_USER> + 08_H
Status register 1	URTJnSTR1	<URTJn_base_USER> + $0C_H$
Status clear register	URTJnSTC	<URTJn_base_USER> + 10_H
FIFO control register 0	URTJnFCTL0	<URTJn_base_USER> + 80_H
FIFO control register 1	URTJnFCTL1	<URTJn_base_OS> + 1020_H
FIFO status register 0	URTJnFSTR0	<URTJn_base_USER> + 84_H
FIFO status register 1	URTJnFSTR1	<URTJn_base_USER> + 88_H
FIFO status clear register	URTJnFSTC	<URTJn_base_USER> + $8C_H$
FIFO receive data register	URTJnFRX	<URTJn_base_USER> + 90_H
FIFO transmit data register	URTJnFTX	<URTJn_base_USER> + 94_H

Base address The register base addresses of UARTJn are defined in the first section of this chapter under the key word “Register addresses”.

(1) URTJnCTL0 - UARTJn control register 0

This register controls UARTJn the basic serial transfer operation.

Access This register can be read/written in 8-bit and 1-bit units.

Address <URTJn_base_USER> + 00_H

Initial Value 00_H. This register is initialized by any reset.

7	6	5	4	3	2	1	0
URTJnPW	URTJn TXE	URTJn RXE	0	0	0	0	URTJn SLDC
R/W	R/W	R/W	R	R	R	R	R/W

Table 20-7 URTJnCTL0 register contents

Bit position	Bit name	Function
7	URTJnPW	UARTJn enable 0: Stop UARTJn operation 1: Enable UARTJn operation Changing this bit initializes all transmission and reception units.
6	URTJnTXE	Transmission operation enable 0: Disable transmission operation 1: Enable transmission operation • To start transmission, set URTJnPW and then set URTJnTXE. To stop transmission, clear URTJnTXE, and then clear URTJnPW (they can be cleared at the same time). • To initialize the transmission unit, clear URTJnTXE, wait for two prescaler clock cycles, and then set URTJnTXE again. For details about the prescaler clock, see *(3) "URTJnCTL2 - UARTJn control register 2" on page 1209*.
5	URTJnRXE	Reception enable 0: Disable reception operation 1: Enable reception operation • To enable reception, set URTJnPW, and then set URTJnRXE. To stop reception, clear URTJnRXE, and then clear URTJnPW (they can be cleared at the same time). • To initialize the reception unit, clear URTJnRXE, wait for two prescaler clock cycles, and then set URTJnRXE again. Reception is enabled when the time of two prescaler clock cycles has elapsed since URTJnRXE is set. The rising edge detection of the URTJnTRXD signal is enabled when four prescaler clock cycles has elapsed after URTJnRXE is set. For details about the prescaler clock, see *(3) "URTJnCTL2 - UARTJn control register 2" on page 1209*.
0	URTJnSLDC	Data consistency check enable 0: Disable consistency check 1: Enable consistency check This bit selects the handling of data consistency error checks when transmitting data. When this bit is set to 1, the transmit data and receive data are compared, and if a mismatch is detected, URTJnSTR1.URTJnDCE is set to 1 and a status interrupt request INTUAJnTIS is issued. This bit is referenced only when starting transmission. Consequently, if this bit value is changed later on during transmission processing, the transmission processing continues, using the value set at the start of transmission.

Cautions

1. Disable transmission if UARTJn meets all the conditions below:
   - Transmission and reception are enabled (URTJnCTL0.URTJnPW = URTJnRXE = URTJnTXE = 1).
   - Data consistency check is enabled (URTJnCTL0.URTJnSLDC = 1).
   - Data is being transmitted or has been transmitted.

   Use the following procedure to keep reception enabled:
   - Check that no data is pending for transmission (URTJnSTR0.URTJnSSBT = URTJnSST = 0).
   - Check that no data is pending for reception (URTJnSTR0.URTJnSSBR = URTJnSSR = 0).
   - Disable transmission by clearing URTJnCTL0.URTJnTXE.

   The reason why this procedure is required is that the data consistency error flag URTJnSTR1.URTJnDCE is cleared if URTJnCTL0.URTJnTXE is cleared.
   Thus a potential data consistency error would not occur if transmission is disabled during a data transfer or after its completion.
2. Disable reception if UARTJn meets all the conditions below:
   - Transmission and reception are enabled (URTJnCTL0.URTJnPW = URTJnRXE = URTJnTXE = 1).
   - Data consistency check is enabled (URTJnCTL0.URTJnSLDC = 1).
   - Data is being transmitted or has been transmitted.

   Use the following procedure to keep transmission enabled:
   - Check that no data is pending for transmission (URTJnSTR0.URTJnSSBT = URTJnSST = 0).
   - Check that no data is pending for reception (URTJnSTR0.URTJnSSBR = URTJnSSR = 0).
   - Disable reception by clearing URTJnCTL0.URTJnRXE.

   The reason why this procedure is required is that the data consistency error flag URTJnSTR1.URTJnDCE is cleared and invalid if URTJnCTL0.URTJnTXE is cleared. Thus a potential data consistency error of already transmitted data would not occur.
3. Do not start data transmission if all the conditions below are met:
   - Data consistency check is enabled (URTJnCTL0.URTJnSLDC = 1).
   - BF reception is enabled (URTJnSTR0.URTJnSSBR = 1).
   - BF detection during reception is disabled (URTJnCTL1.URTJnSLBM = 0).

   A data consistency error will occur under above conditions when BF reception is completed. The status interrupt INTUAJnTIS will be asserted and the reception interrupt request INTUAJnTIR will not be generated (URTJnSTR1.URTJnBSF remains 0). Consequently BF reception completion will not be recognized.

(2) URTJnCTL1 - UARTJn control register 1

This register defines the data frame properties of UARTJn serial data transfers.

Access This register can be read/written in 16-bit units.

Address <URTJn_base_OS> + 20_H

Initial Value 5002_H. This register is initialized by any reset.

15	14	13	12	11	10	9	8
URTJn SLBM	URTJnBLG[2:0]			0	0	0	URTJn CLG
R/W	R/W	R/W	R/W	R	R	R	R/W

7	6	5	4	3	2	1	0
URTJnSLP[1:0]		URTJn TDL	URTJn RDL	0	URTJn SLG	URTJn SLD	URTJn SLIT
R/W	R/W	R/W	R/W	R	R/W	R/W	R/W

Table 20-8 URTJnCTL1 register contents (1/3)

Bit position	Bit name	Function
15	URTJnSLBM	BF receive mode selection 0: BF reception during data reception disabled. 1: BF reception during data reception enabled. • Changing this bit is only allowed if reception is disabled (URTJnCTL0.URTJnPW = 0 or URTJnCTL0.URTJnRXE = 0).
14 to 12	URTJn BLG[2:0]	BF bit length during transmission URTJnBLG2 / URTJnBLG1 / URTJnBLG0 / BF length 1 / 0 / 1 / 13 bits 1 / 1 / 0 / 14 bits 1 / 1 / 1 / 15 bits 0 / 0 / 0 / 16 bits 0 / 0 / 1 / 17 bits 0 / 1 / 0 / 18 bits 0 / 1 / 1 / 19 bits 1 / 0 / 0 / 20 bits Changing these bits is only allowed if transmission is disabled (URTJnCTL0.URTJnPW = 0 or URTJnCTL0.URTJnTXE = 0).
8	URTJnCLG	Receive/transmit data bit length 0: 7 bits 1: 8 bits • When the transmission/reception is performed in the LIN format, set URTJnCLG to 1. • Changing this bit is only allowed if reception and transmission is disabled (URTJnCTL0.URTJnPW = 0 or URTJnCTL0.URTJnRXE = URTJnCTL0.URTJnTXE = 0).

Table 20-8 URTJnCTL1 register contents (2/3)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>7, 6</td><td>URTJn
SLP[1:0]</td><td>Parity bit selection
<table>
<tr><th rowspan="2">URTJnSLP1</th><th rowspan="2">URTJnSLP0</th><th colspan="2">Operation</th></tr>
<tr><th>Transmission</th><th>Reception</th></tr>
<tr><td>0</td><td>0</td><td>Output without parity bit</td><td>Received with no parity</td></tr>
<tr><td>0</td><td>1</td><td>Output 0 parity (0-fixed)</td><td>No parity judgment</td></tr>
<tr><td>1</td><td>0</td><td>Output odd parity</td><td>Judged as odd parity</td></tr>
<tr><td>1</td><td>1</td><td>Output even parity</td><td>Judged as even parity</td></tr>
</table>
• If “Reception with no parity judgment” is selected during reception, a parity check is not performed. Therefore, since the URTJnSTR1.URTJnPE bit is not set, no error interrupt is output.
• When transmission/reception is performed in the LIN format, set URTJnSLP[1:0] to 00_B.
• Changing these bits is only allowed if reception and transmission is disabled (URTJnCTL0.URTJnPW = 0 or URTJnCTL0.URTJnRXE = URTJnCTL0.URTJnTXE = 0).</td></tr>
<tr><td>5</td><td>URTJnTDL</td><td>Transmission data level control
0: No inverted output of transmit data
1: Inverted output of transmit data
• The output level of the URTJnTTXD pin can be inverted using this bit. It inverts the URTJnTTXD output level immediately, regardless of the values of URTJnCTL0.URTJnPW and URTJnCTL0.URTJnTXE. Therefore, if URTJnTDL is set to 1 while the operation is disabled, the URTJnTTXD outputs low level.
• Changing this bit is only allowed if transmission is disabled (URTJnCTL0.URTJnPW = 0 or URTJnCTL0.URTJnTXE = 0).</td></tr>
<tr><td>4</td><td>URTJnRDL</td><td>Reception data level control
0: No inverted output of receive data
1: Inverted output of receive data
• The output level of the URTJnTRXD pin can be inverted using this bit. It inverts the URTJnTRXD input level immediately, regardless of the values of URTJnCTL0.URTJnPW and URTJnCTL0.URTJnRXE. Therefore, if URTJnRDL is set to 1 while the operation is disabled, the URTJnTRXD inputs low level.
• Changing this bit is only allowed if reception is disabled (URTJnCTL0.URTJnPW = 0 or URTJnCTL0.URTJnRXE = 0).</td></tr>
</table>

Table 20-8 URTJnCTL1 register contents (3/3)

Bit position	Bit name	Function
2	URTJnSLG	Stop bit number selection for transmission data 0: 1 bit 1: 2 bits • The stop bit length during data or BF reception is always handled as "1". • Changing this bit is only allowed if transmission is disabled (URTJnCTL0.URTJnPW = 0 or URTJnCTL0.URTJnTXE = 0).
1	URTJnSLD	Transfer direction selection 0: MSB-first transfer 1: LSB-first transfer • When the transmission/reception is performed in the LIN format, set URTJnSLD to 1. • Changing this bit is only allowed if reception and transmission is disabled (URTJnCTL0.URTJnPW = 0 or URTJnCTL0.URTJnRXE = URTJnCTL0.URTJnTXE = 0).
0	URTJnSLIT	Transmission interrupt request (INTUAJnTIT) timing selection 0: INTUAJnTIT generated at the start of transmission, i.e. when the transmit data is stored to the transmission shift register 1: INTUAJnTIT generated at transmission completion • Changing this bit is only allowed if transmission is disabled (URTJnCTL0.URTJnPW = 0 or URTJnCTL0.URTJnTXE = 0).

(3) URTJnCTL2 - UARTJn control register 2

This register defines the baud rates of UARTJn serial data transfers.

Access This register can be read/written in 16-bit units.

Address <URTJn_base_OS> + 24_H

Initial Value $EFFF_H$. This register is initialized by any reset.

15	14	13	12	11	10	9	8
URTJnPRS[2:0]			0	URTJnBRS[11:8]			
R/W	R/W	R/W	R	R/W	R/W	R/W	R/W

7	6	5	4	3	2	1	0
URTJnBRS[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20-9 URTJnCTL2 register contents

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>15 to 13</td><td>URTJn
PRS[2:0]</td><td>Prescaler clock (PRSCLK) division value
0: PRSCLK = PCLK / 2^0
1: PRSCLK = PCLK / 2^1
2: PRSCLK = PCLK / 2^2
3: PRSCLK = PCLK / 2^3
4: PRSCLK = PCLK / 2^4
5: PRSCLK = PCLK / 2^5
6: PRSCLK = PCLK / 2^6
7: PRSCLK = PCLK / 2^7</td></tr>
<tr><td>11 to 0</td><td>URTJn
BRS[11:0]</td><td>Baud rate clock (BRCLK) division value
<table>
<tr><th>URTJn BRS[11:0]</th><th>Transmit/receive BRCLK</th><th>BF receive clock</th></tr>
<tr><td>000_H</td><td rowspan="5">PCLK / (2 x 4)</td><td rowspan="5">PCLK / 4</td></tr>
<tr><td>001_H</td></tr>
<tr><td>002_H</td></tr>
<tr><td>003_H</td></tr>
<tr><td>004_H</td></tr>
<tr><td>005_H</td><td>PCLK / (2 x 5)</td><td>PCLK / 5</td></tr>
<tr><td>...</td><td>PCLK /
(2 x URTJnBRS[11:0])</td><td>PCLK / URTJnBRS[11:0]</td></tr>
<tr><td>FFE_H</td><td>PCLK / (2 x 4094)</td><td>PCLK / 4094</td></tr>
<tr><td>FFF_H</td><td>PCLK / (2 x 4095)</td><td>PCLK / 4095</td></tr>
</table></td></tr>
</table>

Caution Writing to this register is only allowed if the UARTJn operation is disabled (URTJnCTL0.URTJnPW = 0).

PCLK The value of the UARTJn input clock is defined in the first section of this chapter under the key word “Clock supply”.

(4) URTJnTRG - UARTJn trigger register

This register controls the UARTJn transmission/reception trigger of BF.

Access This register can be read/written in 8-bit and 1-bit units.

Address <URTJn_base_USER> + 04_H

Initial Value 00_H. This register is initialized by any reset.

7	6	5	4	3	2	1	0
0	URTJn BRT	URTJn BTT	0	0	0	0	0
R	R/W	R/W	R	R	R	R	R

Table 20-10 URTJnTRG register contents

Bit position	Bit name	Function
6	URTJnBRT	BF reception trigger 0: Read value is always 0, writing 0 is ignored 1: Enable BF reception • When reception is enabled, writing 1 to this bit enables BF reception (URTJnSTR0.URTJnSSBR = 1) and BF reception processing begins when the falling edge of the receive serial signal is detected. • If 1 is written to this bit during reception processing, the current reception processing is terminated. Consequently, the received data is not stored, the framing, parity and overflow error bits are not updated based on the data that was being received and no interrupts are generated. Meanwhile, the BF counter value is continuously being used. • After BF reception, the reception status is set according to the URTJnCTL1.URTJnSLBM setting. • Setting this bit to 1 is only allowed if reception is enabled (URTJnCTL0.URTJnPW = URTJnCTL0.URTJnRXE = 1). After URTJnBRT is set to 1, completion of BF reception is reported by either of the following two methods, based on the URTJnCTL1.URTJnSLBM setting: • if URTJnCTL1.URTJnSLBM = 0 When BF reception is complete, the reception interrupt request INTUAJnTIR is output. • if URTJnCTL1.URTJnSLBM = 1 When BF reception is complete, URTJnSTR1.URTJnBSF is set to 1 and a status interrupt request INTUAJnTIS is output.
5	URTJnBTT	BF transmission trigger 0: Read value is always 0, writing 0 is ignored 1: Enable BF transmission • When this bit is set while URTJnSTR0.URTJnSSBT = 0 and transmission is enabled (URTJnDCE = 0), a BF transmit request is set, and URTJnSSBT is set to 1. • When this bit is set during data transmission, a BF is transmitted after the current transmission processing is completed. Even if this bit is set before the BF transmission is completed, a BF is transmitted only once. • When transmission is enabled (URTJnPW = URTJnTXE = 1), setting this bit clears all previously set data transmit requests (which have not been transmitted), leaving only BF transmit requests. If the URTJnTX7 to URTJnTX0 bits are written after writing 1 to this bit, data is transmitted after the BF is transmitted. • If both a BF transmit request and a data transmit request have been set when transmission starts, the BF transmission takes priority. • When URTJnDCE = 1, writing 1 to this bit is ignored. • Setting this bit to 1 is only allowed if transmission is enabled (URTJnCTL0.URTJnPW = URTJnCTL0.URTJnTXE = 1).

(5) URTJnSTR0 - UARTJn status register 0

This register indicates the current status of serial data transmissions.

Access This register can be read in 8-bit and 1-bit units.
Writing to this register is only allowed if UARTJn operation is disabled (URTJnCTL0.URTJnPW = 0). If UARTJn operation is enabled (URTJnCTL0.URTJnPW = 1), any written values are disregarded and the initial values are restored.

Address <URTJn_base_USER> + 08_H

Initial Value 00_H. This register is initialized by any reset and when URTJnCTL0.URTJnPW is set or cleared.

7	6	5	4	3	2	1	0
0	URTJn SSBR[a]	URTJn SSBT[b]	0	0	0	URTJn SSR[a]	URTJn SST[b]
R	R	R	R	R	R	R	R

a) These bits are also initialized if reception is disabled by URTJnCTL0.URTJnRXE = 0.

b) These bits are also initialized if transmission is disabled by URTJnCTL0.URTJnTXE = 0.

Table 20-11 URTJnSTR0 register contents

Bit position	Bit name	Function
6	URTJnSSBR	BF reception enable status 0: BF reception is disabled 1: BF reception is enabled by setting URTJnTRG.URTJnBRT to 1 (BF reception standby mode or BF reception busy).
5	URTJnSSBT	BF transmission enable status 0: BF transmission is disabled 1: BF transmission is enabled by setting URTJnTRG.URTJnBTT to 1 (BF transmission standby mode or BF transmission busy).
1	URTJnSSR	Data reception status 0: No data reception ongoing 1: Data reception ongoing (data reception busy)
0	URTJnSST	Data transmission status 0: No transmission pending or ongoing 1: Data in URTJnTX[7:0] pending to be transmitted or transmission ongoing

(6) URTJnSTR1 - UARTJn status register 1

This register indicates results of serial data transmissions.

Access This register can be read in 8-bit and 1-bit units.
Writing to this register is only allowed if UARTJn operation is disabled (URTJnCTL0.URTJnPW = 0). If UARTJn operation is enabled (URTJnCTL0.URTJnPW = 1), any written values are disregarded and the initial values are restored.

Address <URTJn_base_USER> + $0C_H$

Initial Value 00_H. This register is initialized by any reset and when URTJnCTL0.URTJnPW is set or cleared.

7	6	5	4	3	2	1	0
0	0	0	URTJn BSF[a]	URTJn DCE[b]	URTJn PE[a]	URTJn FE[a]	0
R	R	R	R	R	R	R	R

a) These bits are also initialized if reception is disabled by URTJnCTL0.URTJnRXE = 0.

b) This bit is also initialized if transmission is disabled by URTJnCTL0.URTJnTXE = 0.

Table 20-12 URTJnSTR1 register contents (1/2)

Bit position	Bit name	Function
4	URTJnBSF	BF reception successful flag 0: BF transmission is disabled by clearing URTJnTRG.URTJnBTT. 1: BF transmission is enabled by setting URTJnTRG.URTJnBTT (BF transmission standby mode or BF transmission busy). The URTJnBSF bit is cleared by the following: • URTJnCTL0.URTJnPW = 0 • URTJnCTL0.URTJnRXE = 0 • URTJnSTC.URTJnCLBS = 1
3	URTJnDCE	Data consistency error flag 0: Transmit/receive data (transmit/receive BF) mismatch was not detected. 1: Transmit/receive data (transmit/receive BF) mismatch was detected. When the BF receive mode selection bit is set during LIN communication, it is necessary to read this bit by using status interrupt processing and to confirm the beginning of a new frame slot. The URTJnDCE bit is cleared by the following: • URTJnCTL0.URTJnPW = 0 • URTJnCTL0.URTJnTXE = 0 • URTJnSTC.URTJnCLDC = 1
2	URTJnPE	Parity error flag 0: No parity error was detected in the received data. 1: A parity error was detected in the received data. The operation of URTJnPE is controlled by the settings of URTJn.URTJnSLP[1:0]. The URTJnPE bit is cleared by the following: • URTJnCTL0.URTJnPW = 0 • URTJnCTL0.URTJnRXE = 0 • URTJnSTC.URTJnCLP = 1

Table 20-12 URTJnSTR1 register contents (2/2)

Bit position	Bit name	Function
1	URTJnFE	Framing error flag 0: No framing error was detected in the received data. 1: A framing error was detected in the received data. The URTJnFE bit is cleared by the following: • URTJnCTL0.URTJnPW = 0 • URTJnCTL0.URTJnRXE = 0 • URTJnSTC.URTJnCLF = 1

Note If the bits of these registers are set (1) and cleared (0) at the same time, setting takes priority over clearing.

For further information concerning error detections, refer to *20.6.5 "UARTJn transmission"* and *20.6.7 "Reception errors"*.

Caution In case reception and transmission is enabled and a consistency check error occurs (URTJnSTR1.URTJnDCE = 1), follow the procedure below prior next data transmission:

- disable transmission by URTJnCTL0.URTJnTXE = 0
- initiate transmission by URTJnTRG.URTJnBTT = 1 (BT transmission trigger) or writing any data to URTJnFTX
- enable transmission by URTJnCTL0.URTJnTXE = 1

Afterwards new transmissions can be started.

(7) URTJnSTC - UARTJn status clear register

This register is used to clear the status bits of the status register 1 URTJnSTR1.

Access This register can be read/written in 8-bit and 1-bit units.
Reading this register returns always 00_H.

Address <URTJn_base_USER> $+10_H$

Initial Value 00_H. This register is initialized by any reset.

7	6	5	4	3	2	1	0
0	0	0	URTJn CLBS	URTJn CLDC	URTJn CLP	URTJn CLF	0
R	R	R	R/W	R/W	R/W	R/W	R/W

Table 20-13 URTJnSTC register contents

Bit position	Bit name	Function
4	URTJnCLBS	Clear BF reception successful flag 0: writing 0 is ignored 1: writing 1 clears URTJnSTR1.URTJnBSF
3	URTJnCLDC	Clear data consistency error flag 0: writing 0 is ignored 1: writing 1 clears URTJnSTR1.URTJnDCE If URTJnDCE is cleared by setting this bit, any pending data or BF transmit requests will be ignored.
2	URTJnCLP	Clear parity error flag 0: writing 0 is ignored 1: writing 1 clears URTJnSTR1.URTJnPE
1	URTJnCLF	Clear framing error flag 0: writing 0 is ignored 1: writing 1 clears URTJnSTR1.URTJnFE

(8) URTJnFCTL0 - FIFO control register 0

This register defines the fill stage of the Rx and Tx FIFOs, at which the reception (INTUAJnTIR) and transmission (INTUAJnTIT) interrupt requests are generated.

Access This register can be read/written in 16-bit units.

Address <URTJn_base_USER> + 80_H

Initial Value $0F00_H$. This register is initialized by any reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	URTJnSLRP[3:0]				0	0	0	0	URTJnSLTP[3:0]			
R	R	R	R	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

Table 20-14 URTJnFCTL0 register contents

Bit position	Bit name	Function
11 to 8	URTJn SLRP[3:0]	Rx FIFO level interrupt setting URTJnSLRP[3:0] defines the Rx FIFO pointer status, at which the reception interrupt request INTUAJnTIR is generated. INTUAJnTIR is generated if URTJnFSTR0.URTJnSSRW[4:0] = (10_H - URTJnSLRP[3:0]), in other words, readable data of (10_H - URTJnSLRP[3:0]) words still remains in the Rx FIFO.
3 to 0	URTJn SLTP[3:0]	Tx FIFO level interrupt setting URTJnSLTP[3:0] defines the Tx FIFO pointer status, at which the transmission interrupt request INTUAJnTIT is generated. INTUAJnTIT is generated if URTJnFSTR0.URTJnSSTW[4:0] = (10_H - URTJnSLTP[3:0]), in other words, writable space of (10_H - URTJnSLTP[3:0]) words still remains in the Tx FIFO.

(9) URTJnFCTL1 - FIFO control register 1

This register controls the Rx time-out detection.

Access This register can be read/written in 8-bit and 1-bit units.

Address <URTJn_base_OS> + 1020_H

Initial Value $3F_H$. This register is initialized by any reset.

7	6	5	4	3	2	1	0
0	0	URTJnSLRT[5:0]					
R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 20-15 URTJnFCTL1 register contents

Bit position	Bit name	Function
5 to 0	URTJn SLRT[5:0]	Rx time-out control: 00_H:Time-out detection disabled 01_H to $3F_H$:Time-out time = (URTJnSLRT[5:0] x 8) cycles of baud rate clock BRCLK

(10) URTJnFSTR0 - FIFO status register 0

This register informs about the fill status of the Rx and Tx FIFOs.

Access This register can be read in 16-bits units, and can only be written if URTJnPW = 0.

Address <URTJn_base_USER> +84_H

Initial Value 0010_H. This register is initialized by any reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	URTJnSSRW[4:0]					0	0	0	URTJnSSTW[4:0]				
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 20-16 URTJnFSTR0 register contents

Bit position	Bit name	Function
12 to 8	URTJn SSRW[4:0]	Rx FIFO status URTJnSSRW[4:0] indicates the number of unread received data words in the Rx FIFO. These bits can be written only if URTJnPW = 0, in a range from 00_H to 10_H. If these bits are read, a value of (10_H - value written to URTJnSSRW[4:0]) is read. If 01_H is written, for example, the value read will be $0F_H$.
4 to 0	URTJn SSTW[4:0]	Tx FIFO status URTJnSSTW[4:0] indicates the amount of writable space (in words) in the Tx FIFO. These bits can be written only if URTJnPW = 0, in a range from 00_H to 10_H. If these bits are read, a value of (10_H - value written to URTJnSSTW[4:0]) is read. If 01_H is written, for example, the value read will be $0F_H$.

(11) URTJnFSTR1 - FIFO status register 1

This register controls the Rx time-out detection.

Access This register can be read in 8-bits units, and can be only written if URTJnPW = 0.

Address <URTJn_base_USER> + 88_H

Initial Value 05_H. This register is initialized by any reset.

7	6	5	4	3	2	1	0
URTJn TMOE	URTJn TOFE	URTJn ROVE	0	URTJn SSTF	URTJn SSTE	URTJn SSRF	URTJn SSRE
R	R	R	R	R	R	R	R

Table 20-17 URTJnFSTR1 register contents

Bit position	Bit name	Function
7	URTJnTMOE	Time-out error detection 0: No time-out error was detected 1: A time-out error was detected To clear this bit after a time-out error has been detected, set URTJnFSTC.URTJnCLTM to 1. If a timeout error is detected at the same time as this bit is cleared, this bit remains set. URTJnTMOE can be written only if URTJnPW = 0. If 1 is written, the value read will be 1.
6	URTJnTOFE	Tx FIFO overflow error detection 0: No Tx FIFO overflow error was detected 1: A Tx FIFO overflow error was detected To clear this bit after a Tx FIFO overflow error has been detected, set URTJnFSTC.URTJnCLTO to 1. If a timeout error is detected at the same time as this bit is cleared, this bit remains set. URTJnTOFE can be written only if URTJnPW = 0. If 1 is written, the value read will be 1.
5	URTJnROVE	Rx FIFO overrun error detection 0: No Rx FIFO overrun error was detected 1: A Rx FIFO overrun error was detected To clear this bit after a Rx FIFO overrun error has been detected, set URTJnFSTC.URTJnCLRO to 1. If a timeout error is detected at the same time as this bit is cleared, this bit remains set. URTJnROVE can be written only if URTJnPW = 0. If 1 is written, the value read will be 1.
3	URTJnSSTF	Tx FIFO full status 0: Tx FIFO is not full 1: Tx FIFO is full
2	URTJnSSTE	Tx FIFO empty status 0: Tx FIFO is not empty 1: Tx FIFO is empty
1	URTJnSSRF	Rx FIFO full status 0: Rx FIFO is not full 1: Rx FIFO is full
0	URTJnSSRE	Rx FIFO empty status 0: Rx FIFO is not empty 1: Rx FIFO is empty

(12) URTJnFSTC - FIFO status clear register

By means of this register the error flags of URTJnFSTR1 can be cleared. Further on the pointers of the Rx and Tx FIFOs can be cleared, thus indicating empty status for both FIFOs.

Access This register can be read/written in 8-bit and 1-bit units.
Reading this register returns always 00_H.

Address <URTJn_base_USER> + $8C_H$

Initial Value 00_H. This register is initialized by any reset.

7	6	5	4	3	2	1	0
URTJn CLTM	URTJn CLTO	URTJn CLRO	0	0	0	URTJn CLTP	URTJn CLRP
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20-18 URTJnFSTC register contents

Bit position	Bit name	Function
7	URTJnCLTM	Time-out error flag clear 0: Read value is always 0, writing 0 is ignored 1: Writing 1 sets URTJnFSTR1.URTJnTMOE = 0
6	URTJnCLTO	Tx FIFO overflow error flag clear 0: Read value is always 0, writing 0 is ignored 1: Writing 1 sets URTJnFSTR1.URTJnTOFE = 0
5	URTJnCLRO	Rx FIFO overrun error flag clear 0: Read value is always 0, writing 0 is ignored 1: Writing 1 sets URTJnFSTR1.URTJnROVE = 0
1	URTJnCLTP	Tx FIFO pointer clear 0: Read value is always 0, writing 0 is ignored 1: Writing 1 sets the Tx FIFO pointer to 00_H, thus - URTJnFSTR0.URTJnSSTW[4:0] = 00_H (Tx FIFO pointer) - URTJnFSTR1.URTJnTOFE = 0 (no Tx FIFO overflow error status) - URTJnFSTR1.URTJnSSTF = 0 (Tx FIFO not full status) - URTJnFSTR1.URTJnSSTE = 1 (Tx FIFO empty status)
0	URTJnCLRP	Rx FIFO pointer clear 0: Read value is always 0, writing 0 is ignored 1: Writing 1 sets the Rx FIFO pointer to 00_H, thus - URTJnFSTR0.URTJnSSRW[4:0] = 00_H (Rx FIFO pointer) - URTJnFSTR1.URTJnROVE = 0 (no Rx FIFO overrun error status) - URTJnFSTR1.URTJnSSRF = 0 (Rx FIFO not full status) - URTJnFSTR1.URTJnSSRE = 1 (Rx FIFO empty status)

(13) URTJnFRX - FIFO receive data register

By this register the reception data is read from the Rx FIFO.
Each reception data is accompanied by flags, which indicate a parity and framing error during reception.

7-bit transfers

If the data length has been specified as 7 bits (URTJnCTL1.URTJnCLG = 0) and

- reception is LSB-first (URTJnCTL1.URTJnSLD = 1),:
  The receive data is transferred to the Rx FIFO URTJnFRX.URTJnRX[6:0] and the data MSB URTJnFRX.URTJnRX[7] always becomes 0.
- reception is MSB-first (URTJnCTL1.URTJnSLD = 0),:
  The receive data is transferred to the Rx FIFO URTJnFRX.URTJnRX[7:1] and the data LSB URTJnFRX.URTJnRX[0] always becomes 0.

For further information on data formats, refer to *20.6.1 "Data formats"*.

Access This register can be read in 16-bit units.

Address <URTJn_base_USER> + 90_H

Initial Value $00FF_H$. This register is initialized by any reset.

15	14	13	12	11	10	9	8
URTJnPE	URTJnFE		0	0	0	0	0
R	R	R	R	R	R	R	R

7	6	5	4	3	2	1	0
URTJnRX[7:0]							
R	R	R	R	R	R	R	R

Table 20-19 URTJnFRX register contents

Bit position	Bit name	Function
15	URTJnPE	Parity error flag 0: No parity error was detected during URTJnRX[7:0] reception 1: A parity error was detected during URTJnRX[7:0] reception
14	URTJnFE	Framing error flag 0: No framing error was detected during URTJFnRX[7:0] reception 1: A framing error was detected during URTJnRX[7:0] reception
7 to 0	URTJn RX[7:0]	Reception data

Pointer modification Each read from URTJnFRX decreases the amount unread data in the Rx FIFO and thus decreases URTJnFSTR0.URTJnSSRW[4:0].

(14) URTJnFTX - FIFO transmit data register

By this register the transmission data is written to the Tx FIFO.

7-bit transfers If the data length has been specified as 7 bits (URTJnCTL1.URTJnCLG = 0) and

- transmission is LSB-first (URTJnCTL1.URTJnSLD = 1),:
  The URTJnFTX.URTJnTX[6:0] value is transferred to the shift register as the Tx FIFO data.
- transmission is MSB-first (URTJnCTL1.URTJnSLD = 0),:
  The URTJnFTX.URTJnTX[7:1] value is transferred to the shift register as the Tx FIFO data.

For further information on data formats, refer to *20.6.1 "Data formats"*.

Access This register can be read/written in 8-bit units.

Address <URTJn_base_USER> + 94_H

Initial Value FF_H. This register is initialized by any reset.

7	6	5	4	3	2	1	0
URTJnTX[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 20-20 URTJnFTX register contents

Bit position	Bit name	Function
7 to 0	URTJnTX[7:0]	Transmission data

Read access Reading URTJnFTX returns the most recent data, that was written to the Tx FIFO.

Pointer modification Each write to URTJnFTX decreases the amount of writable space (words) in the Tx FIFO and thus decreases URTJnFSTR0.URTJnSSTW[4:0].

Overflow error If URTJnFTX is written while the Tx FIFO is full (URTJnFSTR1.URTJnSSTF = 1, the written data is discarded, an overflow error is detected (URTJnFSTR1.URTJnROVE = 1) and the status interrupt INTUAJnTIS is generated.

20.5 Interrupt request signals

The following three interrupt request signals are generated by UARTJn.

- Transmission interrupt request INTUAJnTIT
- Reception interrupt request INTUAJnTIR
- Status interrupt request INTUAJnTIS

20.5.1 Transmission interrupt request INTUAJnTIT

A transmit interrupt request INTUAJnTIT can be configured to be generated upon a certain fill level of the Tx FIFO.

The Tx FIFO fill level for the transmit interrupt request can be set by URTJnFCTL0.URTJnSLTP[3:0], whereas the interrupt is generated if

URTJnFSTR0.URTJnSSTW[4:0] = 10H - URTJnSLTP[3:0]

The amount of writable space in the Tx FIFO, at which INTUAJnTIT is generated, depends on the selected timing of the transmission interrupt request:

- if URTJnCTL1.URTJnSLIT = 0: 10_H - URTJnSLTP[3:0]
- if URTJnCTL1.URTJnSLIT = 1: $0F_H$ - URTJnSLTP[3:0]

Writable space of the number of words shown above remained in the Tx FIFO when the interrupt was generated.

INTUAJnTIT timing

The time to generate the transmission interrupt INTUAJnTIT, and thus indicating the specified amount of writable space in the Tx FIFO, depends on the setting of the URTJnCTL1.URTJnSLIT bit:

- URTJnCTL1.URTJnSLIT = 0: at start of transmission
  The transmission interrupt request INTUAJnTIT is issued when transmission of the first bit is starting.
  In case of data transmission, this indicates the transmission start of the FIFO data of fill level URTJnFCTL0.URTJnSLTP[3:0].
  In case of BF transmission every transmission start of a BF generates INTUAJnTIT.
- URTJnCTL1.URTJnSLIT = 1: at end of transmission
  INTUAJnTIT is generated when the entire data transmission process is completed, i.e. when the last bit of the transmit data has been transmitted.
  The transmission interrupt request INTUAJnTIT is issued when transmission of the last bit is completed.
  In case of data transmission, this indicates the transmission end of the FIFO data of fill level URTJnFCTL0.URTJnSLTP[3:0].
  In case of BF transmission every transmission completion of a BF generates INTUAJnTIT.

The following diagram show the timing of the transmission interrupt request INTUAJnTIT during data transmission for both cases.

INTUAJnTIT at transmission errors

If an error is detected during data consistency check, the interrupt INTUAJnTIT is not generated.

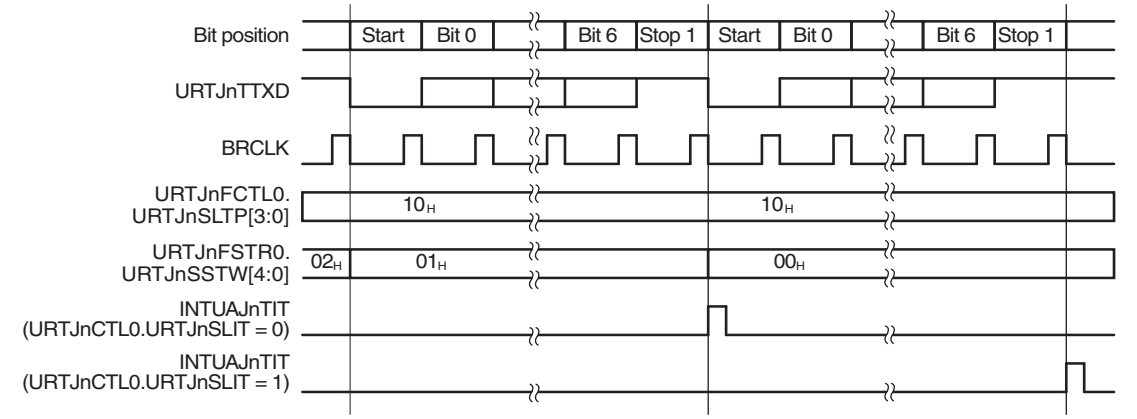


Figure 20-2 Transmission interrupt request timing

Caution After a transmission interrupt is generated, an additional transmission interrupt might be generated in a system that keeps transmission waiting for at least one frame when the FIFO is empty. Therefore, you must clear the interrupt request flag (EICn.EIRFn) in the interrupt routine to 0.

20.5.2 Reception interrupt request INTUAJnTIR

A reception interrupt request INTUAJnTIR can be configured to be generated upon a certain fill level of the Rx FIFO.

The Rx FIFO fill level for the reception interrupt request can be set by URTJnFCTL0.URTJnSLRP[3:0], whereas the interrupt is generated if

URTJnFSTR0.URTJnSSRW[4:0] = 10H - URTJnSLRP[3:0]

INTUAJnTIR at reception errors

INTUAJnTIR is also generated if a parity or framing error was detected, provided the above Rx FIFO fill condition is met.
In case of a Rx FIFO overrun error, no data is stored to the Rx FIFO, thus INTUAJnTIR is not generated.

Caution

After a reception interrupt is generated, an additional reception interrupt might be generated in a system that keeps reception waiting for at least one frame when the FIFO is full. Therefore, you must clear the interrupt request flag (EICn.EIRFn) in the interrupt routine to 0.

The following diagram shows the timing of the reception interrupt request during data reception.

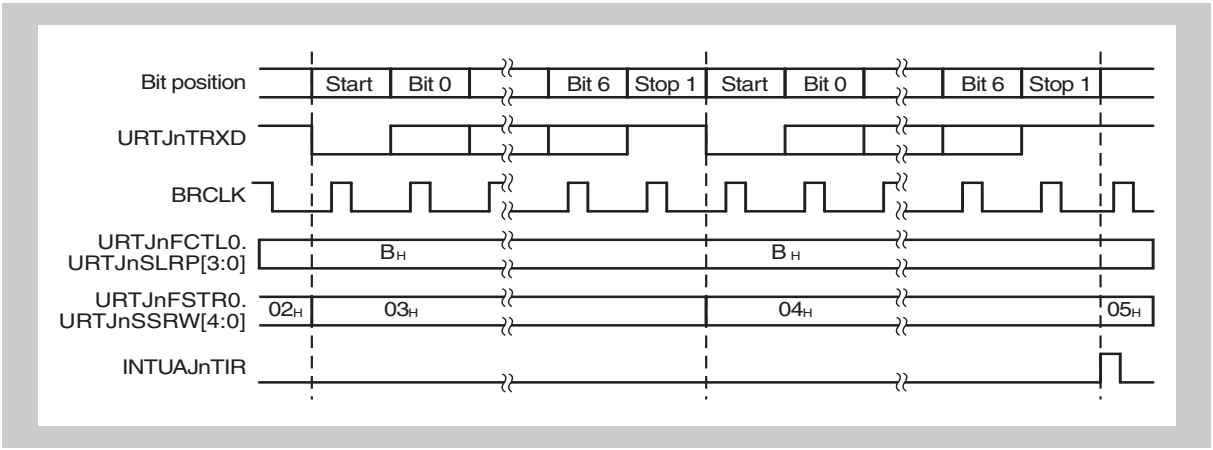


Figure 20-3 Reception interrupt request timing

BF reception

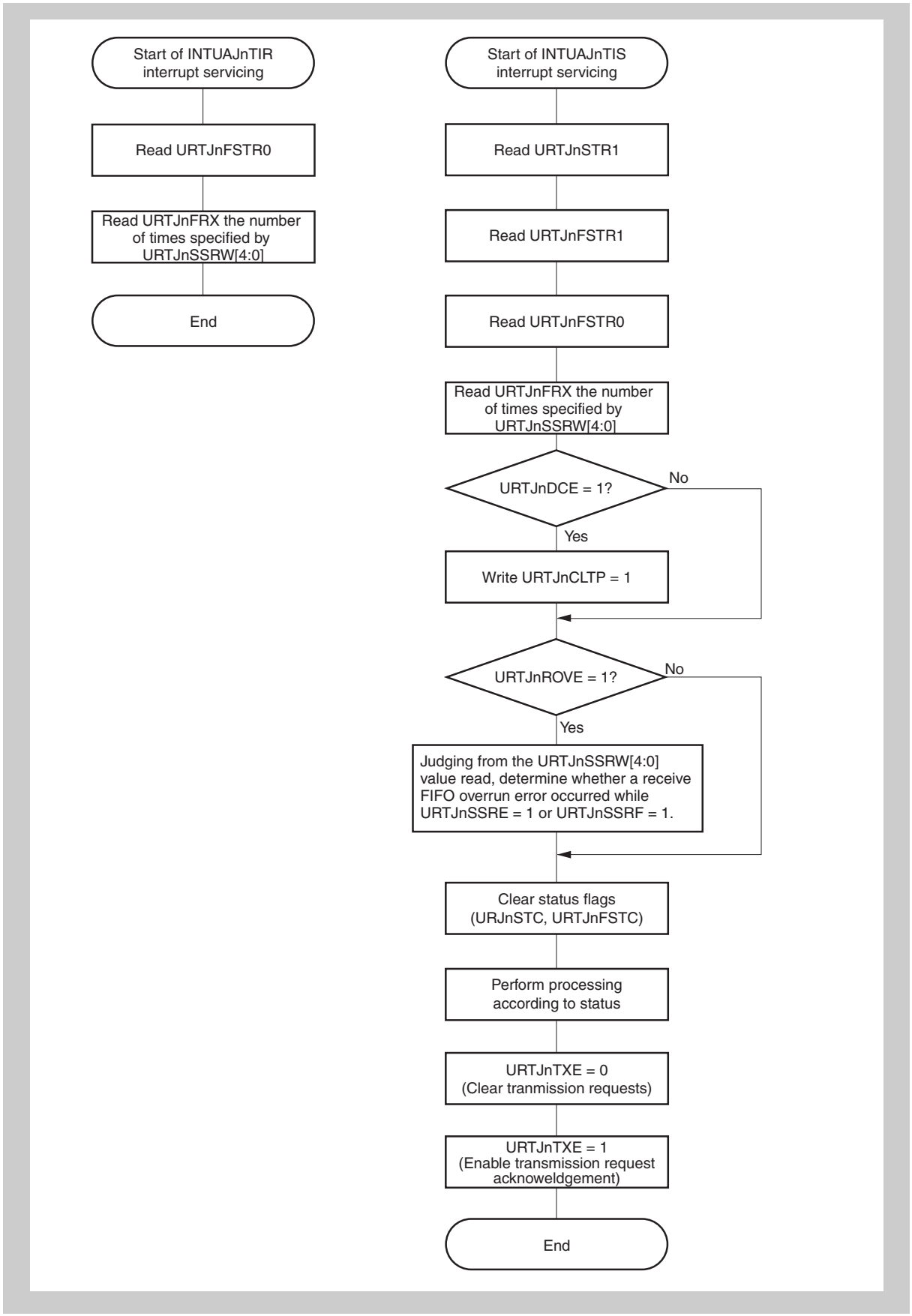
In case of BF reception, INTUAJnTIR is always generated upon completion of the BF reception.

20.5.3 Status interrupt request INTUAJnTIS

A status interrupt request is generated if an error condition occurred during reception or transmission:

- transmission data consistency check error (URTJnSTR1.URTJnDCE = 1)
- reception data parity error (URTJnSTR1.URTJnPE = 1)
- reception data framing error (URTJnSTR1.URTJnFE = 1)
- time-out error (URTJnFSTR1.URTJnTMOE = 1)
- Tx FIFO overflow error (URTJnFSTR1.URTJnTOFE = 1)
- Rx FIFO overrun error (URTJnFSTR1.URTJnROVE = 1)

if BF length of more than 10 bits is detected while BF reception is enabled during data reception (URTJnCTL1.URTJnSLBM = 1)



<R> **Figure 20-4 Processing flow after interrupt generation**

20.6 Operation

20.6.1 Data formats

Full-duplex serial data reception and transmission is performed.

As shown in the figures below, one data frame of transmit/receive data consists of a start bit, character bits, parity bit, and stop bit(s).

Several properties of a transmit/receive data frame can be specified by control bits of the URTJnCTL1 register:

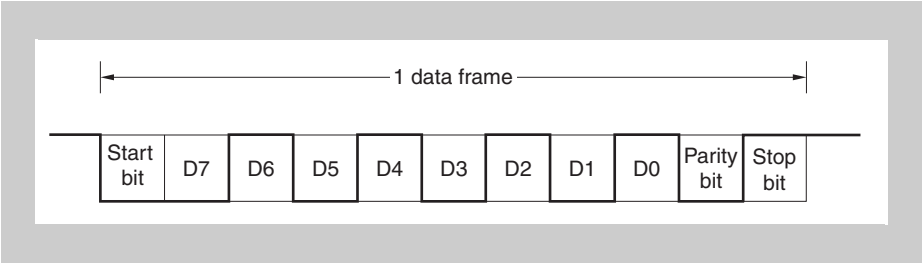
Table 20-21 Data format specification

Item	Options	Control bits
Start bit	1 bit	Fixed
Character bits	7 bits / 8 bits	URTJnCTL1.URTJnCLG
Parity	Even parity/odd parity/ 0 parity no parity	URTJnCTL1.URTJnSLP[1:0]
Stop bit	1 bit / 2 bits	URTJnCTL1.URTJnSLG
Data order	MSB first / LSB first	URTJnCTL1.URTJnSLD
Tx data level	inverted / not inverted	URTJnCTL1.URTJnTDL
Rx data level	inverted / not inverted	URTJnCTL1.URTJnRDL

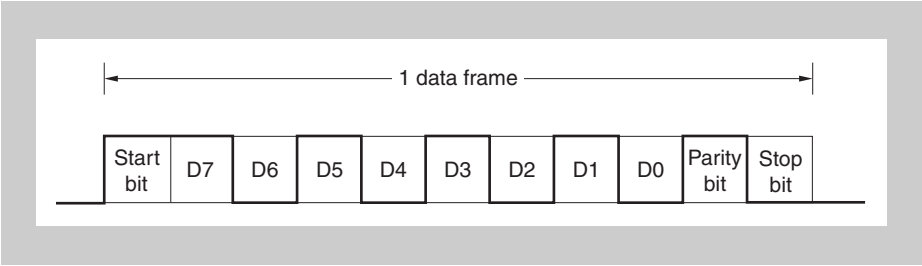
(1) UARTJn transmit/receive data format

(a) 8-bit data length, LSB first, even parity, 1 stop bit, transfer data: 55_H

(b) 8-bit data length, MSB first, even parity, 1 stop bit, transfer data: 55_H

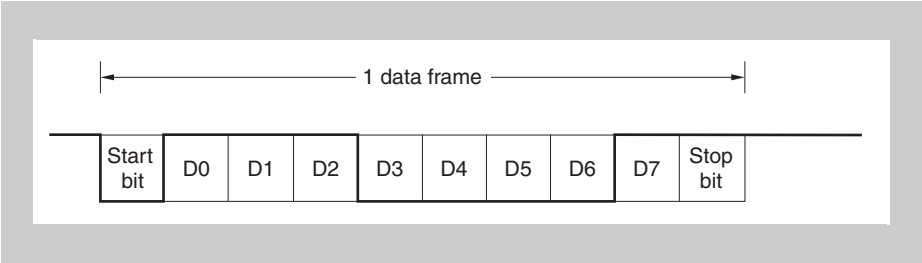


(c) 8-bit data length, MSB first, even parity, 1 stop bit, transfer data: 55_H, URTJnTTXD inversion



(d) 7-bit data length, LSB first, odd parity, 2 stop bits, transfer data: 36_H

(e) 8-bit data length, LSB first, no parity, 1 stop bit, transfer data: 87_H



20.6.2 BF transmission/reception format

The UARTJn has a BF (Break Field) transmission/reception control function to enable use of the LIN functions.

About LIN

LIN stands for Local Interconnect Network and is a low-speed (1 to 20 kbps) serial communication protocol intended to aid the cost reduction of an automotive network.

LIN communication is single-master communication, and up to 15 slaves can be connected to one master.

The LIN slaves are used to control the switches, actuators, and sensors, and these are connected to the LIN master via the LIN network.

Normally, the LIN master is connected to a network such as CAN (Controller Area Network).

In addition, the LIN bus uses a single-wire method and is connected to the nodes via a transceiver that complies with ISO9141.

In the LIN protocol, the master transmits a frame with baud rate information and the slave receives it and corrects the baud rate error. Therefore, communication is possible when the baud rate error in the slave is ±14% or less.

Figure 20-5 "LIN transmission outline" and *Figure 20-6 "LIN reception outline"* outline the transmission and reception manipulations of LIN.

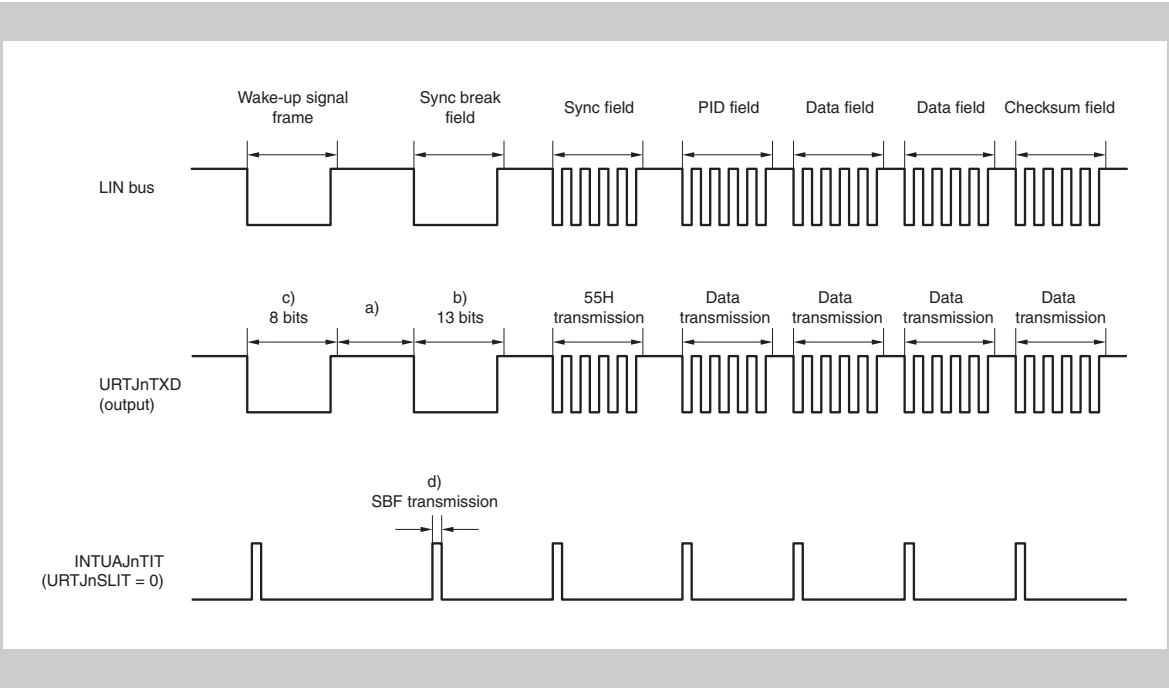


Figure 20-5 LIN transmission outline

a) The interval between fields is controlled by software.
b) BF output is performed by hardware. The output width is the bit length set by URTJnCTL1.URTJnBLG[2:0].
If even finer output width adjustments are required, such adjustments can be performed using URTJnCTLn.URTJnBRS[11:0].
c) 80_H transfer in the 8-bit mode is substituted for the wakeup signal frame.
d) A transmission enable interrupt INTUAJnTIT is generated at the start of each transmission. INTUAJnTIT is also generated at the start of each BF transmission.

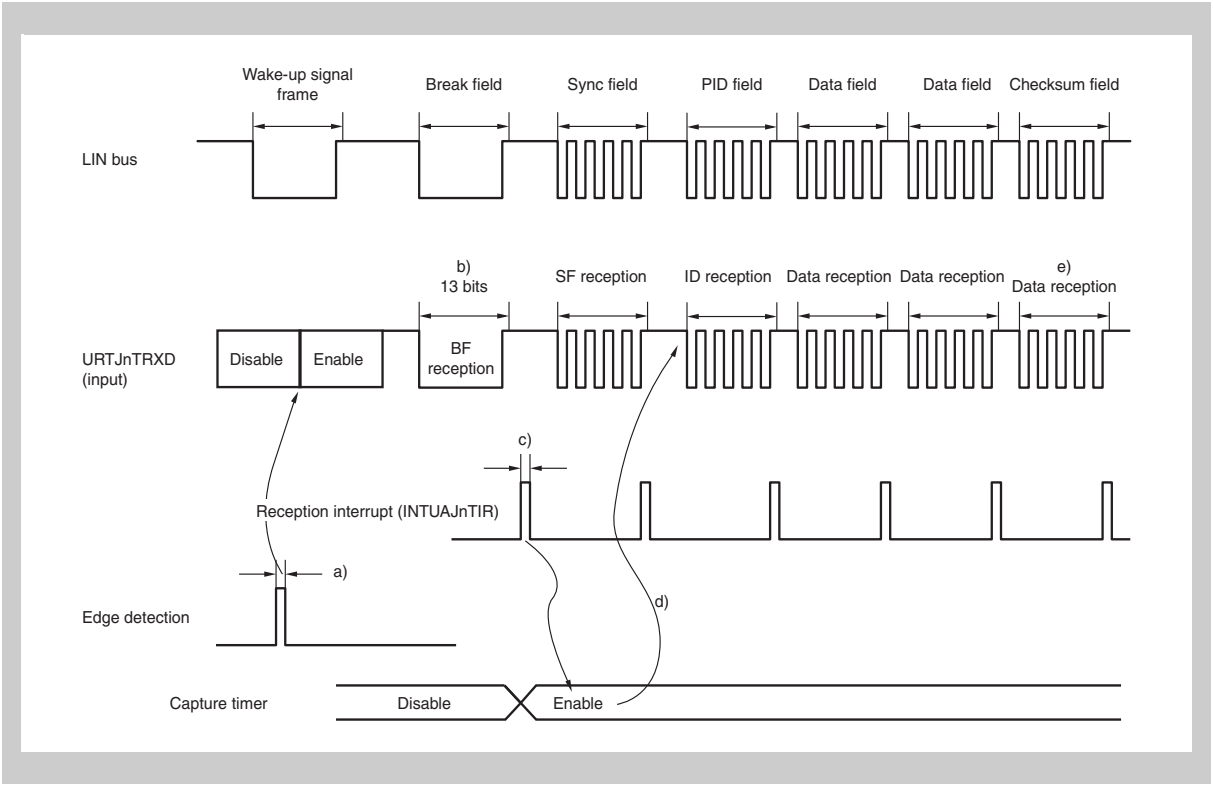


Figure 20-6 LIN reception outline

a) The wakeup signal sent by the pin edge detector enables UARTJn, and sets the BF reception mode.

b) BF reception is judged to end normally when a BF of 11 or more bits is received. An interrupt is generated as shown in the table below, according to the setting of the BF reception mode selection bit URTJnCTL1.URTJnSLBM and the value of the URTJnSTR0.URTJnSSBR bit.

URTJnSLBM	URTJnSSBR	Interrupts
1	x	INTUAJnTIS
0	1	INTUAJnTIR
0	0	A framing error has occurred, so INTUAJnTIS is generated.

c) When BF reception ends normally, an interrupt is generated as follows according to the setting of the BF reception mode selection bit URTJnCTL1.URTJnSLBM:
- When URTJnDTL1.URTJnSLBM is 0, the reception interrupt INTUAJnTIR is generated.
- When URTJnDTL1.URTJnSLBM is 1, the status interrupt INTUAJnTIS is generated and the BF reception success flag URTJnSTR1.URTJnBSF is set.

If the BF reception trigger bit URTJnTRG.URTJnBRT is 1, error detection for the overrun, parity, and framing errors is not performed during BF reception. Also, data transfer from the receive shift register to the receive data register URTJnRX is not performed. URTJnRX holds the previous value at this time.

d) In order to adjust the baud rate clock properly, the URTJnTRXD signal must be connected to the timer capture input. The transfer rate and the baud rate error can be calculated by measuring the time between URTJnTRXD edges, and the baud rate can be adjusted by specifying a value for the baud rate setting bits URTJnCTL2.URTJnBRS[11:0].

e) A checksum field is identified by software. When a checksum field is received, UARTJn is initialized and set to the BF reception mode by software. But if URTJnCTL1.URTJnSLBM is 1 at this time, UARTJn automatically starts BF reception without entering the BF reception mode.

20.6.3 BF transmission

When the URTJnCTL0 bits URTJnPW = URTJnTXE = 1, the transmission enabled status is entered, and BF transmission is started by setting the BF transmission trigger URTJnTRG.URTJnBTT = 1.

Thereafter, URTJnSTR0.URTJnSSBT is set to “1” and a low level width of 13 to 20 bits, as specified by URTJnCTL1.URTJnBLG[2:0], is output.
A transmission interrupt INTUAJnTIT) is generated upon BF

- transmission start, if URTJnCTL1.URTJnSLIT = 0
- transmission end, if URTJnCTL1.URTJnSLIT = 1.

Following the end of BF transmission, URTJnSTR0.URTJnSSBT is automatically cleared. Thereafter, the UARTJn transmission mode is restored.

Transmission is suspended until the data to transmit next is written to the URTJnTX register and URTJnSTR0.URTJnSST is set, or until the BF transmission trigger URTJnTRG.URTJnBTT is set and URTJnSTR0.URTJnSSBT changes to 1.

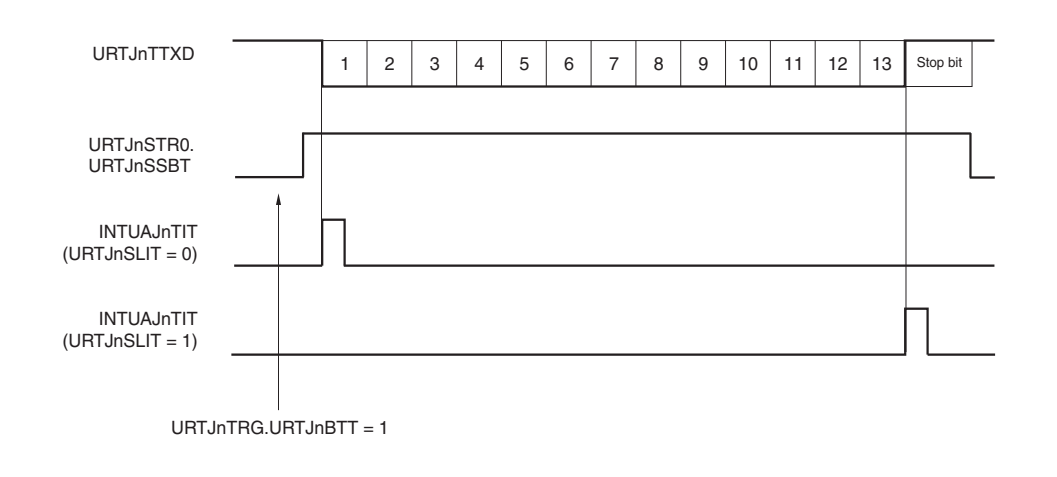
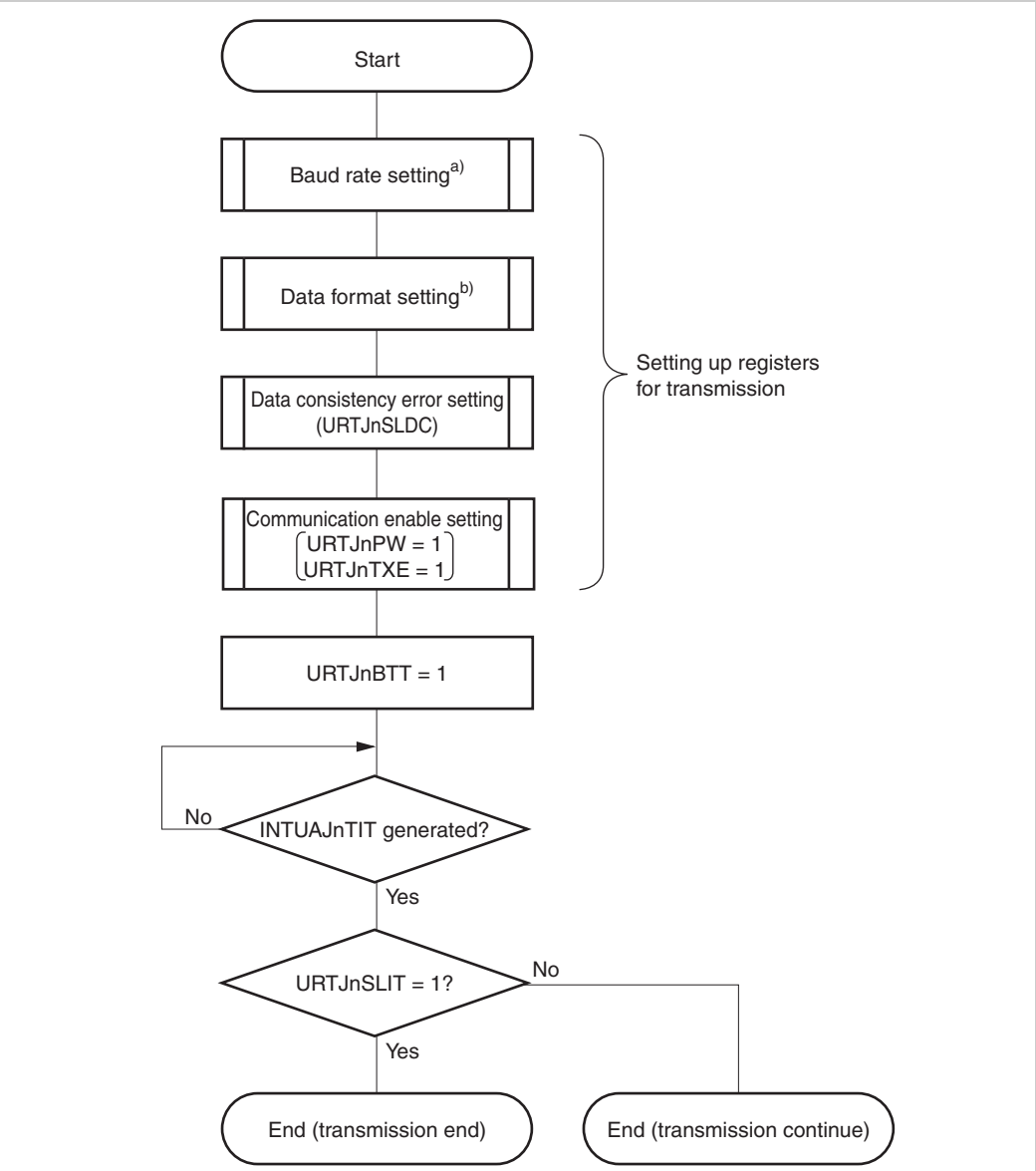


Figure 20-7 BF transmission



a) Set by using the URTJnPRS[2:0] and URTJnBRS[11:0] bits.

b) Set by using the URTJnBLG[2:0], URTJnCLG, URTJnSLP[1,0], URTJnTDL, URTJnSLG, URTJnSLD, and URTJnSLIT bits.

Figure 20-8 Flowchart of BF transmission

20.6.4 BF reception

The reception enabled status is achieved by setting the URTJnCTL0.URTJnPW bit to 1 and then setting the URTJnCTL0.URTJnRXE bit to 1.

The BF reception wait status is set by setting the BF reception trigger URTJnTRG.URTJnBRT = 1.

In the BF reception wait status, the URTJnTRXD pin is monitored and start bit detection is performed.

Following detection of the low level, reception is started and the internal counter counts up according to the set baud rate.

When a high level is received and if the BF width is 11 or more bits, while the BF receiving mode selection bit

- URTJnCTL1.URTJnSLBM = 0,
  the reception interrupt INTUAJnTIR is generated.
- URTJnCTL1.URTJnSLBM = 1,
  the status interrupt INTUAJnTIS is generated and BF reception success flag URTJnSTR1.URTJnBSF is set at the same time.

The URTJnSTR0.URTJnSSBR bit is automatically cleared and BF reception ends.

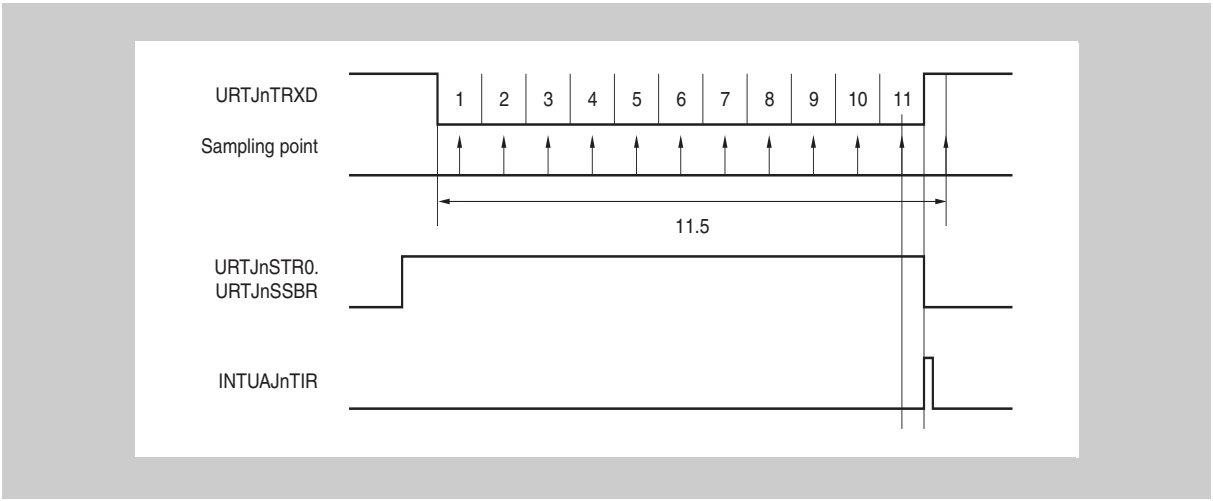


Figure 20-9 Normal BF reception (stop bit after more than 10.5 “L” bits)

Error detection for the URTJnSTR1 error flags URTJnOVE, URTJnPE, and URTJnFE is suppressed and UARTJn communication error detection processing is not performed.
Moreover, the erroneous data is not stored in URTJnRX, but the initial value FFH is held.

If the BF width is 10 or fewer bits, reception is terminated as error processing without generating an interrupt, and the BF reception mode is returned to. URTJnSTR0.URTJnSSBR is not cleared at this time.

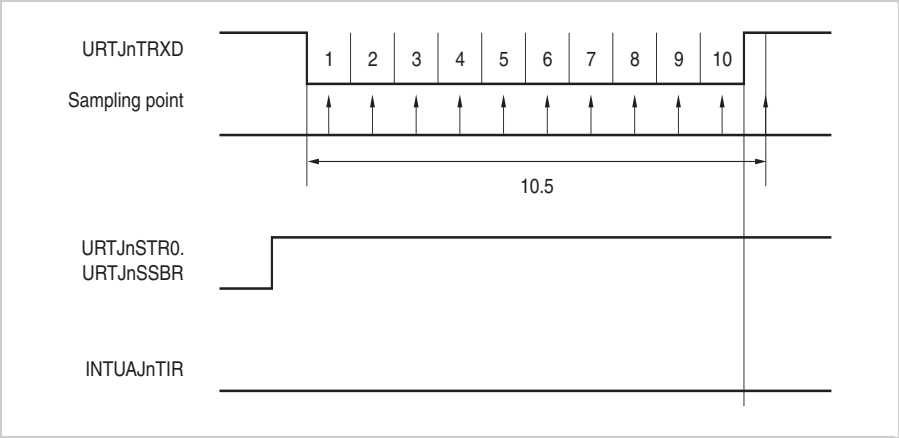


Figure 20-10 BF reception error (stop bit within 10.5 “L” bits)

The BF mode can be selected between a single BF receive mode and an any time BF receive mode in by URTJnCTL1.URTJnSLBM. The status of a successful reception of the BF is indicated by URTJnSTR1.URTJnBSF.

Note URTJnSTR0.URTJnSSBR is set to “1” when

- URTJnTRG.URTJnBRT is set to “1”, or
- the error is cleared by normal BF reception.

20.6.5 UARTJn transmission

(1) Transmission FIFO

The Tx FIFO comprises 8 bit x 16 levels to hold the 8-bit data to be transmitted consecutively.

The Tx FIFO is filled by writing to the URTJnFTX register.

Tx FIFO status

Various status information are provided to check the fill level of the Tx FIFO:

- The amount of writable space in the Tx FIFO can be checked by reading the Tx FIFO pointer URTJnFSTR0.URTJnSSTW[4:0]
- FIFO full/not full status is indicated by URTJnSTR1.URTJnSSTF (= 1: full)
- FIFO empty/not empty status is indicated by URTJnSTR1.URTJnSSTE (= 1: empty)

Pointer modification

Each write to URTJnFTX decreases the amount of writable space in the Tx FIFO and thus decreases URTJnFSTR0.URTJnSSTW[4:0].

Overflow error

If URTJnFTX is written while the Tx FIFO is full (URTJnFSTR1.URTJnSSTF = 1), the written data is discarded, an overflow error is detected (URTJnFSTR1.URTJnTOFE = 1) and the status interrupt INTUAJnTIS is generated.

URTJnFTX read

Reading URTJnFTX returns the most recent data, that was written to the Tx FIFO.

(2) Transmission start and stop

Transmission start

Set the transmission enabled status by performing the following procedures.

- Specify the baud rate by URTJnCTL2.
- Specify the transmit parity, data character length, stop bit length, transmit data order, transmission interrupt request timing and output logic level by URTJnCTL1.
- Enable UARTJn operation and transmission by URTJnCTL0.URTJnPW = URTJnCTL0.URTJnTXE = 1)

Write of the transmit data to the Tx FIFO via URTJnFTX starts transmission. The data which is saved in the Tx FIFO is transferred to the transmit shift register. Then, the start, parity and stop bits are added and the data frame is output serially via URTJnTTXD.

Transmission stop

When URTJnCTL0.URTJnPW or URTJnCTL0.URTJnTXE is set to 0, transmission operations are stopped immediately, even during transmission processing.

Concurrent BF and data transmission

When a BF transmit request and a data transmit request have both been set, BF transmission takes priority.

(3) Transmission data consistency check

The UARTJn incorporates a data consistency check function to detect a mismatch between the transmit data output via the signal URTJnTTXD and the data received from the URTJnTRXD signal, when UARTJn operates in transmission mode.

Note For performing the data consistency check the URTJnTTXD signal must be fed back to URTJnTRXD externally.

Data consistency check is enabled by URTJnCTL0.URTJnSLDC = 1.

In case of a mismatch between the URTJnTTXD and URTJnTRXD signals the data consistency error flag URTJnSTR1.URTJnDCE is set and a status interrupt request INTUAJnTIS occurs.

Data consistency check can be performed with reception enabled or disabled.

If reception is disabled (URTJnCTL0.URTJnRXE = 0) the reception completion interrupt request INTUAJnTIR, the URTJnSTR1 status bits URTJnBSF, URTJnFE, URTJnPE and the status interrupt request signal INTUAJnTIS will not be generated. Receive data is not stored to the Rx FIFO.

If reception is enabled (URTJnCTL0.URTJnRXE = 1) the receive data is treated as in normal reception mode, i.e. all status bits and interrupts are handled, and the data is stored to the Rx FIFO.

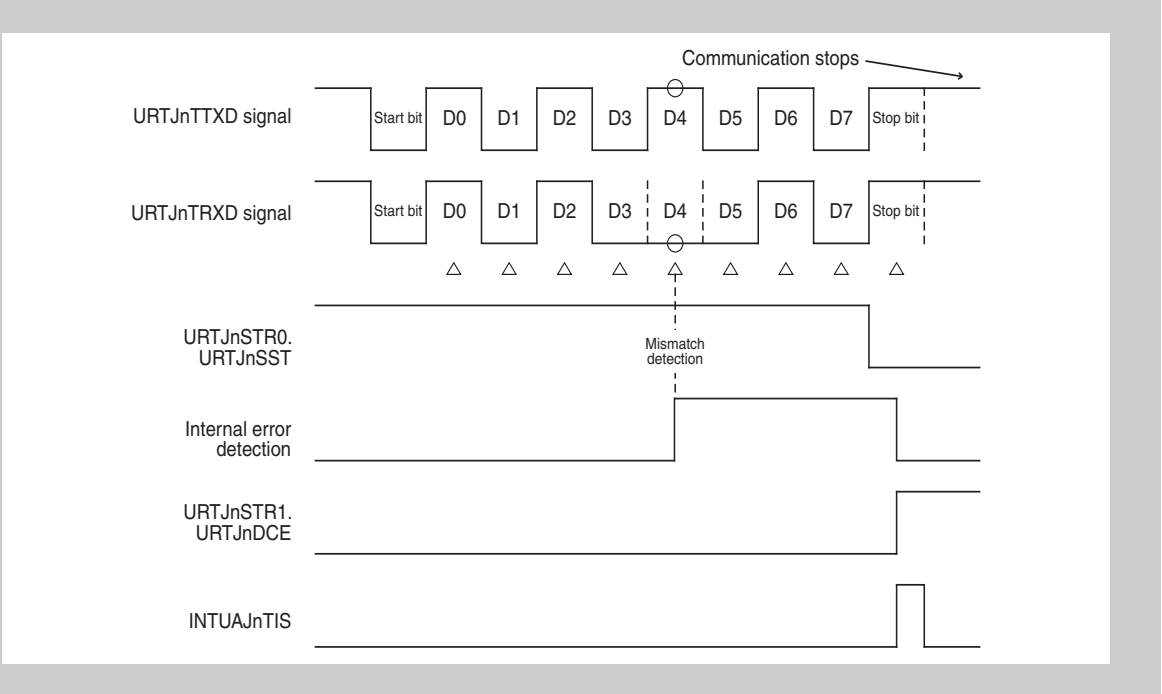


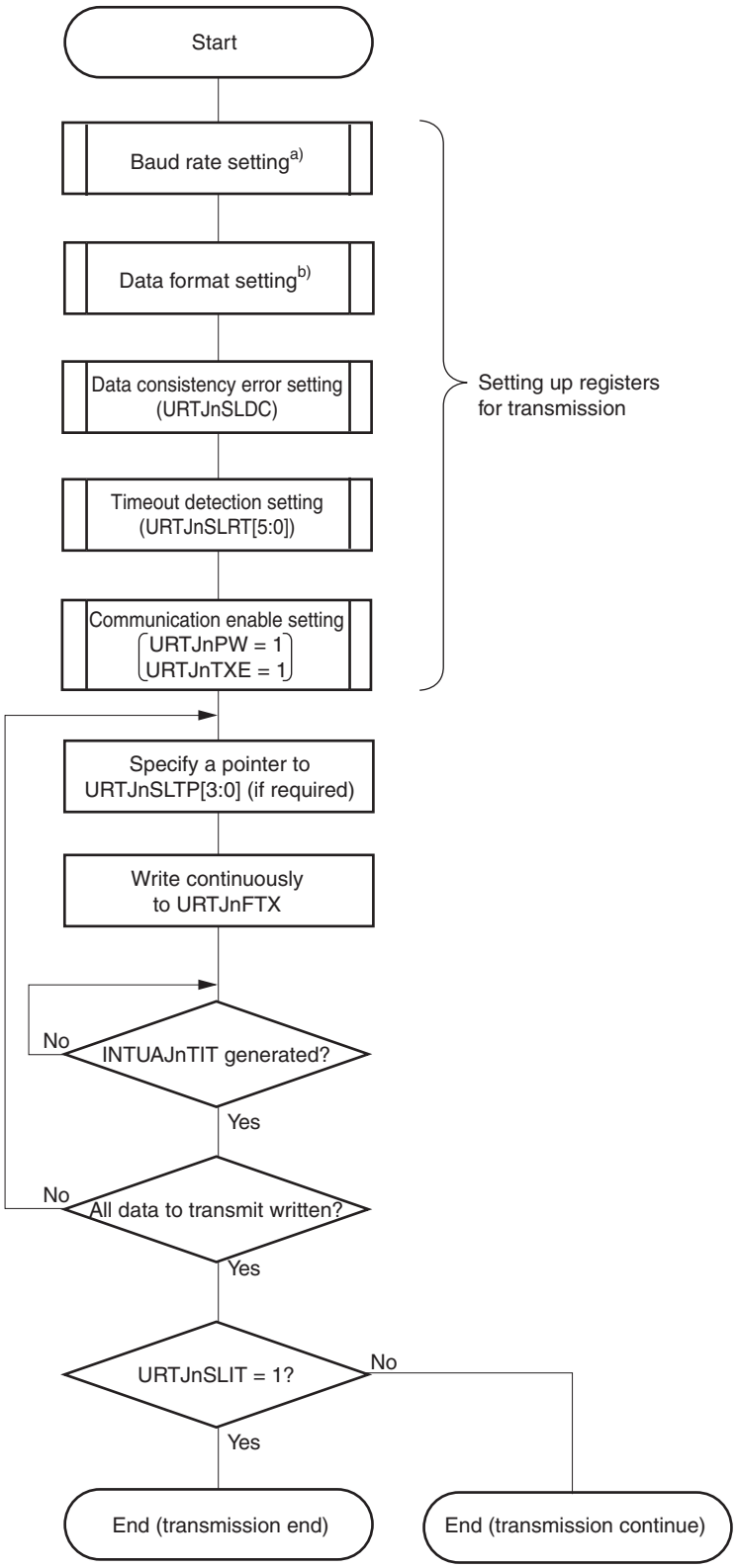
Figure 20-11 Timing example of data consistency error (no BF reception active, i.e. URTJnSTR0.URTJnSSBR = 0)

If a data consistency error was detected (URTJnSTR1.URTJnDCE = 1), the subsequent data is not transmitted until the data consistency error flag is cleared (URTJnSTC.URTJnCLDC = 1) or transmission is disabled (URTJnCTL0.URTJnPW = 0, or URTJnCTL0.URTJnTXE = 0).

(4) Continuous transmission procedure

Continuous transmission is achieved by maintaining a certain fill level of the Tx FIFO.

This means in particular to set the generation of the transmission interrupt INTUAJnTIT, that indicates the Tx FIFO fill level, appropriately via the Tx FIFO level interrupt setting URTJnSLTP[3:0].



a) Set by using the URTJnPRS[2:0] and URTJnBRS[11:0] bits.

b) Set by using the URTJnCLG, URTJnSLP[1,0], URTJnTDL, URTJnSLG, URTJnSLD, and URTJnSLIT bits.

Figure 20-12 Flowchart of data transmission

20.6.6 UARTJn reception

(1) Reception FIFO

The Rx FIFO comprises 10 bit x 16 levels to store the 8-bit data that has been received and additionally two error flags, indicating parity and framing errors.

The Rx FIFO is emptied by reading from the URTJnFRX register.

Rx FIFO status

Various status information are provided to check the fill level of the Rx FIFO:

- The number of received words in the Rx FIFO can be checked by reading the Rx FIFO pointer URTJnFSTR0.URTJnSSRW[4:0]
- FIFO full/not full status is indicated by URTJnFSTR1.URTJnSSRF (= 1: full)
- FIFO empty/not empty status is indicated by URTJnFSTR1.URTJnSSRE (= 1: empty)

Pointer modification

Each reception increases the number of data words in receiption FIFO and thus increases URTJnSTR0.URTJnSSRW[4:0].

Overrun error

If new data is received while the Rx FIFO is full (URTJnFSTR1.URTJnSSRF = 1), the received data is discarded, an overrun error is detected (URTJnFSTR1.URTJnROVE = 1) and the status interrupt INTUAJnTIS is generated.

(2) Reception start and stop

Reception start

Set the reception enabled status by the following procedure:

- Specify the baud rate by URTJnCTL2.
- Specify the receive parity, data character length, stop bit length, receive data order and output logic level by URTJnCTL1.
- Enable UARTJn operation and reception by URTJnCTL0.URTJnPW = URTJnCTL0.URTJnRXE = 1).

When the sampling of the input level of the URTJnTRXD pin is performed and the falling edge is detected, the data sampling of the URTJnTRXD input is started. The start bit is recognized if the URTJnTRXD pin is low level after the time of a half bit is passed after the detection of the falling edge (shown in the figure below). After a start bit has been recognized, the receive operation starts, and serial data is stored in the receive shift register according to the set baud rate. When the reception interrupt INTUAJnTIR is asserted upon reception of the stop bit, the data stored in the receive shift register is written to the Rx FIFO.

Reception stop

When URTJnCTL0.URTJnPW or URTJnCTL0.URTJnRXE is set to 0, reception operations are stopped immediately, even during reception processing.

Rx format change When the receive data order, parity, data character length, and the stop bit length are changed, clear the power bit (URTJnCTL0.URTJnPW = 0) or clear both the transmission enabled bit and the reception enabled bit (URTJnTXE = 0, URTJnRXE = 0), and then change the setting.

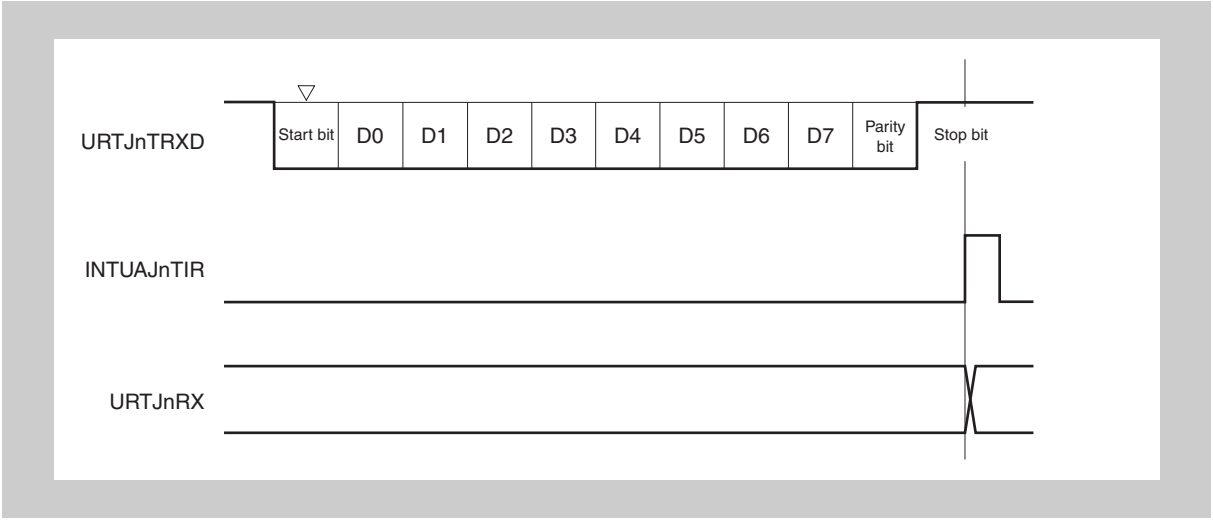
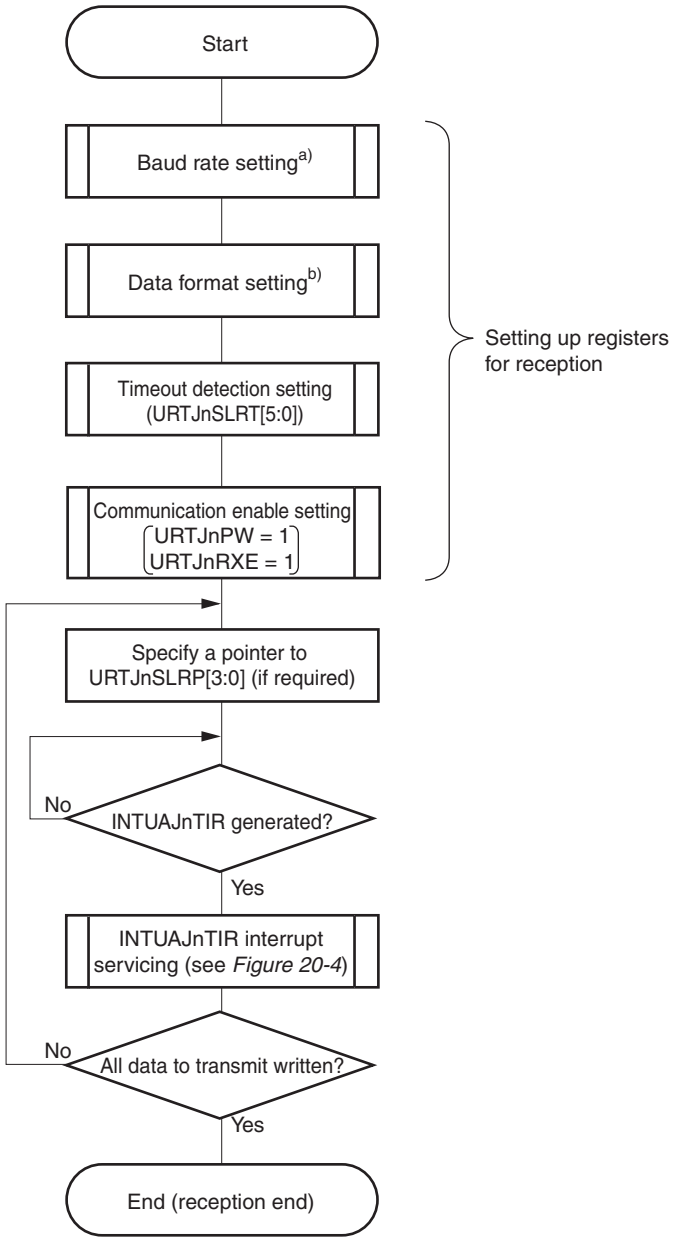


Figure 20-13 UARTJn reception

Caution The operation during reception is performed assuming that there is only one stop bit. A second stop bit is ignored.

Notes

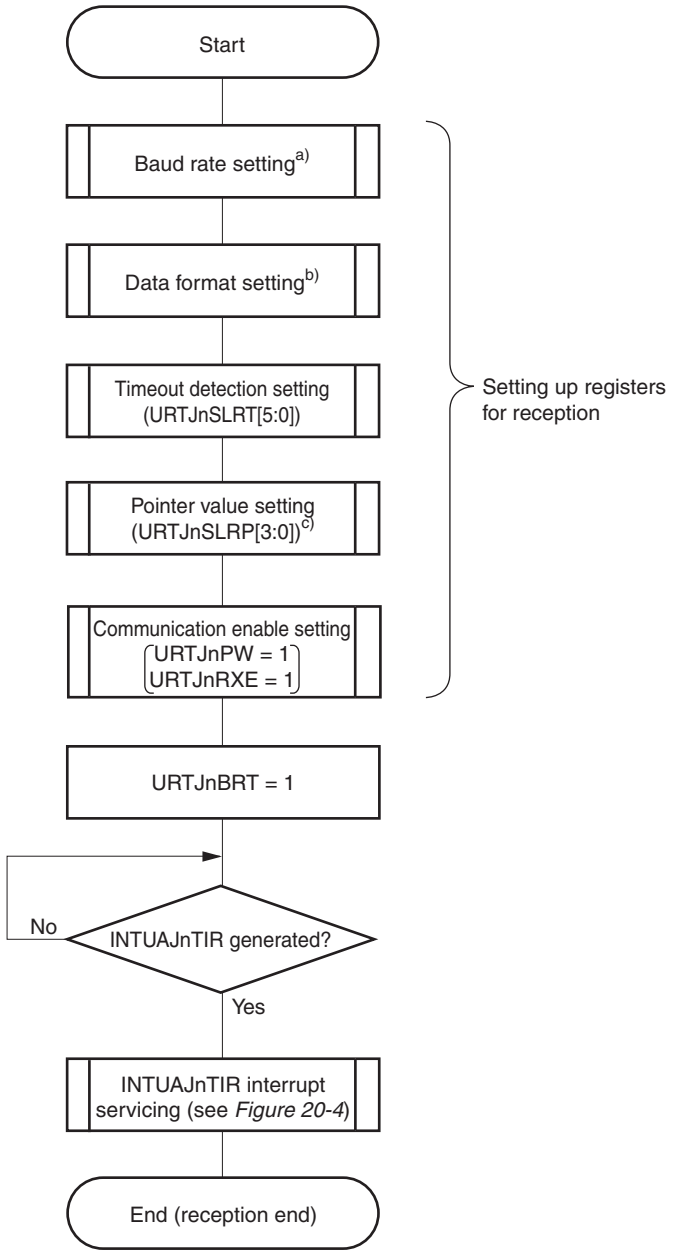
1. If low level is always input to the URTJnTRXD pin, it is not judged as the start bit.
2. In continuous reception, immediately after the stop bit is detected at the first reception bit (when the reception interrupt is generated), the next start bit can be detected.



a) Set by using the URTJnPRS[2:0] and URTJnBRS[11:0] bits.

b) Set by using the URTJnSLBM, URTJnCLG, URTJnSLP[1:0], URTJnRDL, and URTJnSLD bits.

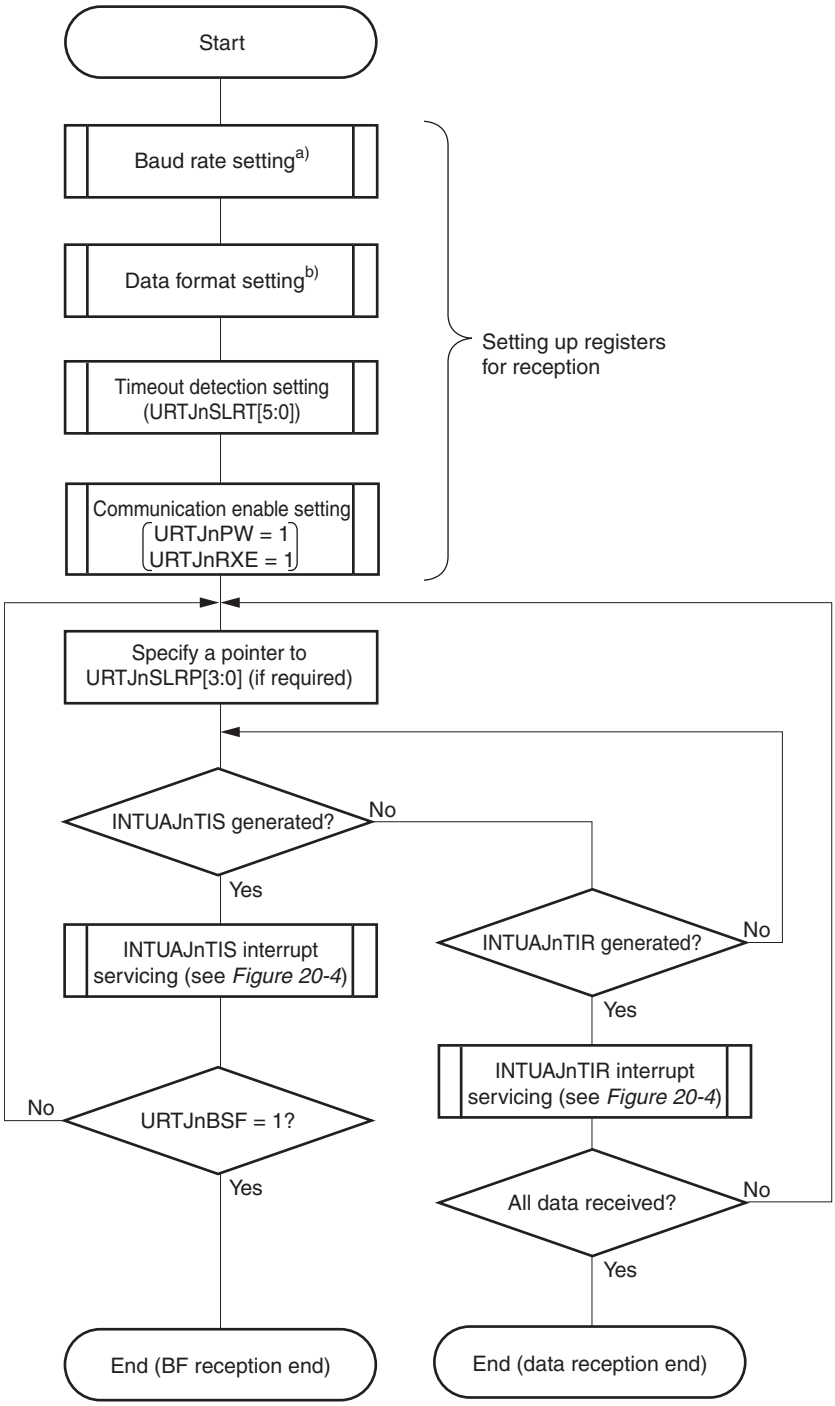
Figure 20-14 Flowchart of data reception when URTJnSLBM = 0, URTJnSSBR = 0



a) Set by using the URTJnPRS[2:0] and URTJnBRS[11:0] bits.

b) Set by using the URTJnSLBM, URTJnCLG, URTJnSLP[1:0], URTJnRDL, and URTJnSLD bits.

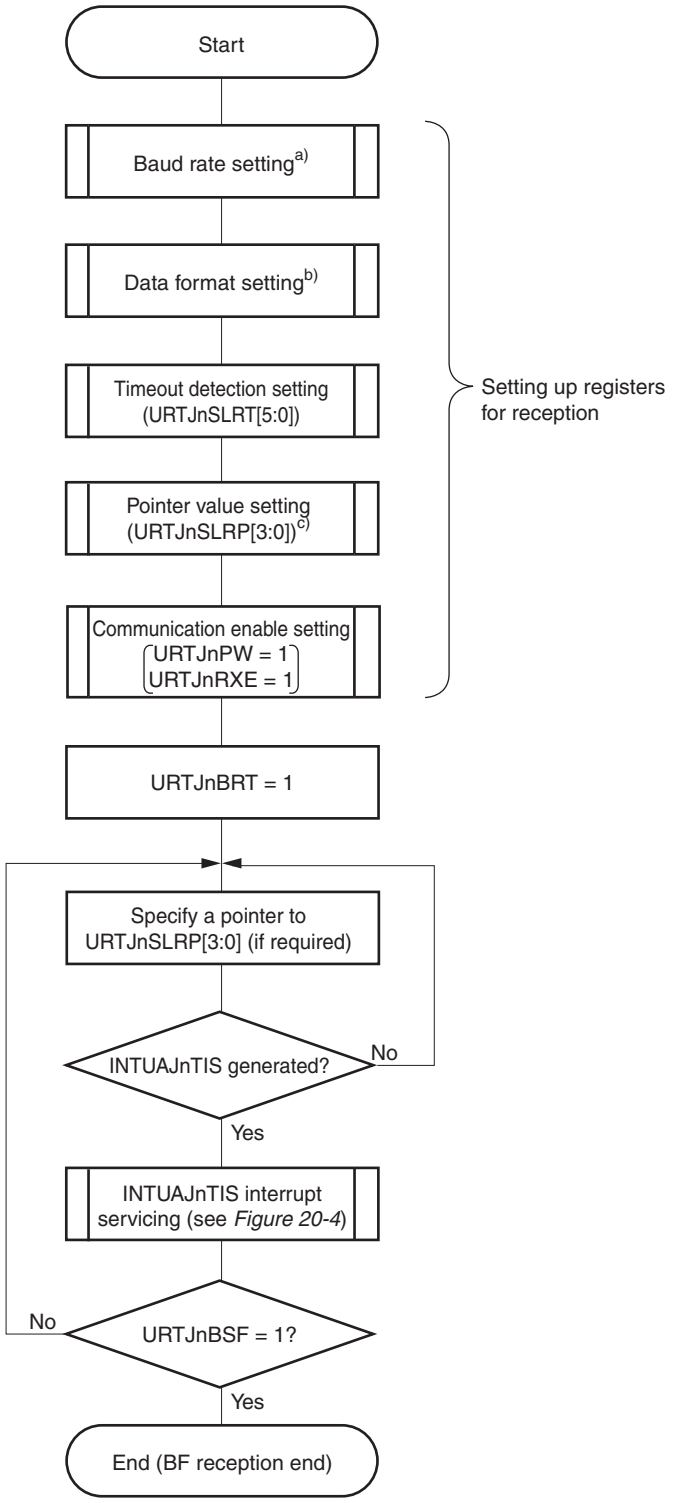
Figure 20-15 Flowchart of data reception when URTJnSLBM = 0, URTJnSSBR = 1



a) Set by using the URTJnPRS[2:0] and URTJnBRS[11:0] bits.

b) Set by using the URTJnSLBM, URTJnCLG, URTJnSLP[1:0], URTJnRDL, and URTJnSLD bits.

Figure 20-16 Flowchart of data reception when URTJnSLBM = 0, URTJnSSBR = 0



a) Set by using the URTJnPRS[2:0] and URTJnBRS[11:0] bits.

b) Set by using the URTJnSLBM, URTJnCLG, URTJnSLP[1:0], URTJnRDL, and URTJnSLD bits.

Figure 20-17 Flowchart of data reception when URTJnSLBM = 1, URTJnSSBR = 1

20.6.7 Reception errors

Errors during a receive operation are of four types:

- parity errors
- framing errors
- overrun errors
- timeout errors

Various data reception result error flags are provided to identify the error cause and the status interrupt request signal INTUAJnTIS is generated when an error occurs.

Table 20-22 Reception error causes and indicators

Reception error	Error flags	Cause
Parity error	URTJnSTR1.URTJnPE = 1 upon first parity error URTJnFRX.URTJnPE = 1 for each data in Rx FIFO	Received parity bit does not match the setting
Framing error	URTJnSTR1.URTJnFE = 1 upon first framing error URTJnFRX.URTJnFE = 1 for each data in Rx FIFO	Stop bit not detected
Overrun error	URTJnFSTR1.URTJnROVE = 1	Reception of next data completed while Rx FIFO is full
Timeout error	URTJnFSTR1.URTJnTMOE = 1	No Rx FIFO access within a certain time period

Overrun error

An overrun error occurs (URTJnFSTR1.URTJnROVE = 1), when data has been received while the Rx FIFO is full. The received data is not transferred to the Rx FIFO, but is discarded.

Parity and framing error

If a parity error or a framing error occurs during reception

- the associated error bit is set:
  - for a parity error: URTJnSTR1.URTJnPE = 1
  - for a framing error: URTJnSTR1.URTJnFE = 1,
- reception continues until the reception position of the first stop bit,
- the reception data and the error flag URTJnFRX.URTJnPE respectively URTJnFRX.URTJnFE is transferred to the Rx FIFO,
- the status interrupt INTUAJnTIS is generated,
- in case the Rx FIFO fill level reaches the predefined level URTJnFSTR0.URTJnSSRW[4:0], the reception interrupt INTUAJnTIR is generated.

Note The error flags URTJnFRX.URTJnPE and URTJnFRX.URTJnFE are set upon detection of the first parity respectively framing error and stay at 1 until they are cleared by URTJnSTC.URTJnCLP = 1 respectively URTJnSTC.URTJnCLF = 1.

Time-out error A time-out error occurs under following conditions:

- The Rx FIFO is not empty.
- Neither received data has been stored to nor data has been read from the Rx FIFO within a certain time period.

The time period is programmable by setting URTJnFCTL1.URTJnSLRT[5:0]. This value specifies the time period as multiple of the baud rate clock BRCLK period.

If a time-out error occurs, the flag URTJnFSTR1.URTJnTMOE is set to 1 and the status interrupt request INTUAJnTIS is asserted.

20.6.8 Parity types and operations

Caution When using the LIN function, fix the URTJnCTL1.URTJnSLP[1:0] to 00_B.

The parity bit is used to detect bit errors in the communication data. Normally the same parity is used on the transmission side and the reception side.

In the case of even parity and odd parity, it is possible to detect odd-count bit errors. In the case of 0 parity and no parity, errors cannot be detected.

(1) Even parity

- During transmission
  The number of bits whose value is “1” among the transmit data, including the parity bit, is controlled so as to be an even number. The parity bit values are as follows:
  - Odd number of bits whose value is “1” among transmit data:1
  - Even number of bits whose value is “1” among transmit data:0
- During reception
  The number of bits whose value is “1” among the reception data, including the parity bit, is counted, and if it is an odd number, a parity error is output.

(2) Odd parity

- During transmission
  Opposite to even parity, the number of bits whose value is “1” among the transmit data, including the parity bit, is controlled so that it is an odd number. The parity bit values are as follows.
  - Odd number of bits whose value is “1” among transmit data: 0
  - Even number of bits whose value is “1” among transmit data: 1
- During reception
  The number of bits whose value is “1” among the receive data, including the parity bit, is counted, and if it is an even number, a parity error is output.

(3) 0 parity

During transmission, the parity bit is always made 0, regardless of the transmit data.

During reception, parity bit check is not performed. Therefore, no parity error occurs, regardless of whether the parity bit is 0 or 1.

(4) No parity

No parity bit is added to the transmit data.

Reception is performed assuming that there is no parity bit. No parity error occurs since there is no parity bit.

20.6.9 Digital receive data noise filter

The receive data signal input URTJnTRXD is equipped with a digital noise filter to eliminate noise and spikes.

This filter samples the URTJnTRXD pin using the prescaler output clock PRSCLK.

When the same sampling value is read twice, the URTJnTRXD signal is validated as the input data.
Therefore, URTJnTRXD signal data not exceeding the width of 1 prescaler output clock cycle is judged to be noise and thus eliminated.

The noise filter causes a delay of 4 prescaler output clock PRSCLK cycles when capturing the serial data URTJnTRXD, until it is forwarded as valid.

20.7 Baud Rate Generator

The transmission and reception baud rate BRCLK are derived from the APB bus clock PCLK by use of a prescaler and a baud rate generator, as shown in the figure below.

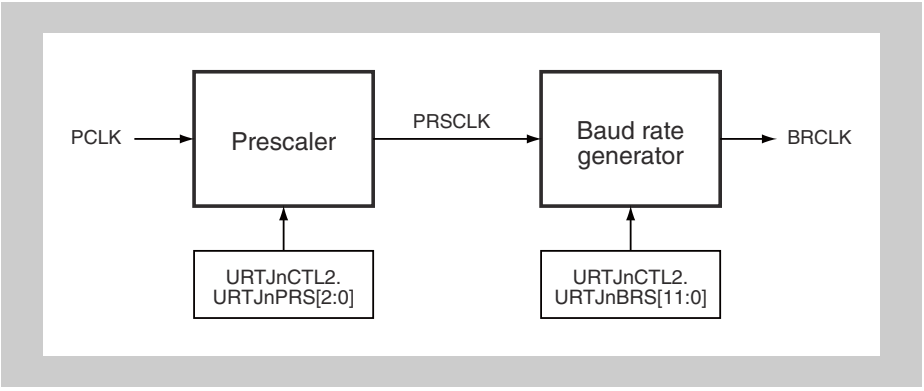


Figure 20-18 Configuration of baud rate generator

The prescaler output clock PRSCLK is a fraction of PCLK, the divisor is set up the value URTEnCTL2.URTJnPRS[2:0]:

$$PRSCLK = PCLK / 2^{URTJnPRS[2:0]}$$

PRSCLK is further divided by the baud rate generator by a value, determined by URTJnCTL2.URTJnBRS[11:0].
The baud rate generator distinguishes between the baud rate for data frames and BF receptions, as listed in the table below. The BF reception clock is the double of the baud rate clock BRCLK.

Table 20-23 Baud rate generator clocks output

URTJnCTL2. URTJnBRS[11:0]	Transmit/receive BRCLK	BF receive clock
000_H	PCLK / (2 x 4)	PCLK / 4
001_H		
002_H		
003_H		
004_H		
005_H	PCLK / (2 x 5)	PCLK / 5
...	PCLK / (2 x URTJnBRS[11:0])	PCLK / URTJnBRS[11:0]
FFE_H	PCLK / (2 x 4094)	PCLK / 4094
FFF_H	PCLK / (2 x 4095)	PCLK / 4095

Table 20-24 Example of baud rate generator settings

Baud rate	f_{XX} = 200 MHz			f_{XX} = 192 MHz			f_{XX} = 160 MHz			f_{XX} = 144 MHz		
	f_{PCLK} = 66.667 MHz			f_{PCLK} = 64.000 MHz			f_{PCLK} = 53.333 MHz			f_{PCLK} = 48.000 MHz		
	URTJnPRS [2:0]	URTJnBRS [11:0]	ERR [%]	URTJnPRS [2:0]	URTJnBRS [11:0]	ERR [%]	URTJnPRS [2:0]	URTJnBRS [11:0]	ERR [%]	URTJnPRS [2:0]	URTJnBRS [11:0]	ERR [%]
300	5	3472	0.01	5	3333	0.01	5	2778	–0.01	5	2500	0.00
600	4	3472	0.01	4	3333	0.01	4	2778	–0.01	4	2500	0.00
1200	3	3472	0.01	3	3333	0.01	3	2778	–0.01	3	2500	0.00
2400	2	3472	0.01	2	3333	0.01	2	2778	–0.01	2	2500	0.00
4800	1	3472	0.01	1	3333	0.01	1	2778	–0.01	1	2500	0.00
9600	0	3472	0.01	0	3333	0.01	0	2778	–0.01	0	2500	0.00
14400	0	2315	–0.01	0	2222	0.01	0	1852	–0.01	0	1667	–0.02
19200	0	1736	0.01	0	1667	–0.02	0	1389	–0.01	0	1250	0.00
31250	0	1067	–0.03	0	1024	0.00	0	853	0.04	0	768	0.00
38400	0	868	0.01	0	833	0.04	0	694	0.06	0	625	0.00
57600	0	579	–0.05	0	556	–0.08	0	463	–0.01	0	417	–0.08
76800	0	434	0.01	0	417	–0.08	0	347	0.06	0	313	–0.16
115200	0	289	0.12	0	278	–0.08	0	231	0.21	0	208	0.16
153600	0	217	0.01	0	208	0.16	0	174	–0.22	0	156	0.16
312500	0	107	–0.31	0	102	0.39	0	85	0.39	0	77	–0.26
1000000	0	33	1.01	0	32	0.00	0	27	–1.23	0	24	0.00
2000000	0	17	–1.96	0	16	0.00	0	13	2.56	0	12	0.00
3000000	0	11	1.01	0	11	–3.03	0	9	–1.23	0	8	0.00
4000000	–	–	–	0	8	0.00	–	–	–	0	6	0.00
6000000	–	–	–	–	–	–	–	–	–	0	4	0.00
8000000	–	–	–	0	4	0.00	–	–	–	–	–	–
8333333	0	4	0.00	–	–	–	–	–	–	–	–	–

Chapter 21 Clocked Serial Interface G (CSIG)

This chapter contains a generic description of clocked serial interface G (CSIG).

The first section describes all V850E2/MN4 specific properties, such as instances, register base addresses, and input/output signal names.
The subsequent sections describe the features that apply to all implementations.

21.1 V850E2/MN4 CSIG Features

Instances

This microcontroller has the following number of instances of CSIG.

Table 21-1 Instances of CSIG

Clocked serial interface G	
Instance	6
Name	CSIG0 to CSIG5

Instances index n

Throughout this chapter, the individual instances of CSIG is identified by the index "n" (n = 0 to 5), for example, CSIGnCTL0 for the CSIGn control register.

Register addresses

All CSIGn register addresses are given as addresses offset from the individual base address <CSIGn_base_USER> or <CSIGn_base_OS>.
The base addresses <CSIGn_base_USER> and <CSIGn_base_OS> of each CSIGn are listed in the following table:

Table 21-2 Register base addresses

CSIGn instance	Base address	<CSIGn_base> address
CSIG0	<CSIGn_base_USER>	FFFF E400$_H$
	<CSIGn_base_OS>	FF70 0000$_H$
CSIG1	<CSIGn_base_USER>	FFFF E500$_H$
	<CSIGn_base_OS>	FF71 0000$_H$
CSIG2	<CSIGn_base_USER>	FFFF E600$_H$
	<CSIGn_base_OS>	FF72 0000$_H$
CSIG3	<CSIGn_base_USER>	FFFF E700$_H$
	<CSIGn_base_OS>	FF73 0000$_H$
CSIG4	<CSIGn_base_USER>	FFFF E800$_H$
	<CSIGn_base_OS>	FF74 0000$_H$
CSIG5	<CSIGn_base_USER>	FFFF E900$_H$
	<CSIGn_base_OS>	FF75 0000$_H$

Clock supply All CSIG provide the following clock inputs:

Table 21-3 CSIGn clock supply

CSIGn instance	CSIGn clock	Connected to
CSIG0 to CSIG5	PCLK	f_{PCLK}

Baud rate For CSIG, communication is possible at the maximum transfer speeds (baud rates) listed in the following table.

Table 21-4 Maximum CSIGn transfer speeds (baud rates)

Mode	Maximum transfer speed (baud rate)
Master mode	8.33 Mbps (target)
Slave mode	8.33 Mbps (target)

Interrupts and DMA/DTS CSIG can generate the following interrupt and DMA/DTS requests:

Table 21-5 CSIGn interrupt and DMA/DTS requests (1/2)

CSIGn signals	Function	Connected to
CSIG0		
CSIG0TIC	Communication interrupt	• Interrupt controller161 (INTCSIG0IC) • DMA controller trigger 117
CSIG0TIR	Reception interrupt	• Interrupt controller 160 (INTCSIG0IR) • DMA controller trigger 116
CSIG0TIRE	Communication error interrupt	• Interrupt controller 159 (INTCSIG0IRE)
CSIG1		
CSIG1TIC	Communication interrupt	• Interrupt controller 164 (INTCSIG1IC) • DMA controller trigger 119
CSIG1TIR	Reception interrupt	• Interrupt controller 163 (INTCSIG1IR) • DMA controller trigger 118
CSIG1TIRE	Communication error interrupt	• Interrupt controller 162 (INTCSIG1IRE)
CSIG2		
CSIG2TIC	Communication interrupt	• Interrupt controller 167 (INTCSIG2IC) • DMA controller trigger 121
CSIG2TIR	Reception interrupt	• Interrupt controller 166 (INTCSIG2IR) • DMA controller trigger 120
CSIG2TIRE	Communication error interrupt	• Interrupt controller 165 (INTCSIG2IRE)
CSIG3		
CSIG3TIC	Communication interrupt	• Interrupt controller 170 (INTCSIG3IC) • DMA controller trigger 123
CSIG3TIR	Reception interrupt	• Interrupt controller 169 (INTCSIG3IR) • DMA controller trigger 122
CSIG3TIRE	Communication error interrupt	• Interrupt controller 168 (INTCSIG3IRE)

Table 21-5 CSIGn interrupt and DMA/DTS requests (2/2)

CSIGn signals	Function	Connected to
CSIG4		
CSIG4TIC	Communication interrupt	• Interrupt controller 173 (INTCSIG4IC) • DMA controller trigger 125
CSIG4TIR	Reception interrupt	• Interrupt controller 172 (INTCSIG4IR) • DMA controller trigger 124
CSIG4TIRE	Communication error interrupt	• Interrupt controller 171 (INTCSIG4IRE)
CSIG5		
CSIG5TIC	Communication interrupt	• Interrupt controller 176 (INTCSIG5IC) • DMA controller trigger 127
CSIG5TIR	Reception interrupt	• Interrupt controller 175 (INTCSIG5IR) • DMA controller trigger 126
CSIG5TIRE	Communication error interrupt	• Interrupt controller 174 (INTCSIG5IRE)

I/O signals The I/O signals of the clocked serial interface G are listed in the following table.

Table 21-6 CSIGn I/O signals (1/2)

CSIGn signal	Function	Connected to
CSIG0		
CSIG0TSCK	Serial clock signal	Port SCK0
CSIG0TSI	Serial data input signal	Port SI0
CSIG0TSO	Serial data output signal	Port SO0
$\overline{\text{CSIG0TSSI}}$	Slave select input signal	Port CSI0_SSI
CSIG0TRYO	Ready/busy output signal	Port CSI0_RYO
CSIG0TRYI	Ready/busy input signal	Port CSI0_RYI
CSIG1		
CSIG1TSCK	Serial clock signal	Port SCK1
CSIG1TSI	Serial data input signal	Port SI1
CSIG1TSO	Serial data output signal	Port SO1
$\overline{\text{CSIG1TSSI}}$	Slave select input signal	Port CSI1_SSI
CSIG1TRYO	Ready/busy output signal	Port CSI1_RYO
CSIG1TRYI	Ready/busy input signal	Port CSI1_RYI
CSIG2		
CSIG2TSCK	Serial clock signal	Port SCK2
CSIG2TSI	Serial data input signal	Port SI2
CSIG2TSO	Serial data output signal	Port SO2
$\overline{\text{CSIG2TSSI}}$	Slave select input signal	Port CSI2_SSI
CSIG2TRYO	Ready/busy output signal	Port CSI2_RYO
CSIG2TRYI	Ready/busy input signal	Port CSI2_RYI

Table 21-6 CSIGn I/O signals (2/2)

CSIGn signal	Function	Connected to
CSIG3		
CSIG3TSCK	Serial clock signal	Port SCK3
CSIG3TSI	Serial data input signal	Port SI3
CSIG3TSO	Serial data output signal	Port SO3
$\overline{\text{CSIG3TSSI}}$	Slave select input signal	Port CSI3_SSI
CSIG3TRYO	Ready/busy output signal	Port CSI3_RYO
CSIG3TRYI	Ready/busy input signal	Port CSI3_RYI
CSIG4		
CSIG4TSCK	Serial clock signal	Port SCK4
CSIG4TSI	Serial data input signal	Port SI4
CSIG4TSO	Serial data output signal	Port SO4
$\overline{\text{CSIG4TSSI}}$	Slave select input signal	Port CSI4_SSI
CSIG4TRYO	Ready/busy output signal	Port CSI4_RYO
CSIG4TRYI	Ready/busy input signal	Port CSI4_RYI
CSIG5		
CSIG5TSCK	Serial clock signal	Port SCK5
CSIG5TSI	Serial data input signal	Port SI5
CSIG5TSO	Serial data output signal	Port SO5
$\overline{\text{CSIG5TSSI}}$	Slave select input signal	Port CSI5_SSI
CSIG5TRYO	Ready/busy output signal	Port CSI5_RYO
CSIG5TRYI	Ready/busy input signal	Port CSI5_RYI

21.2 Functional Overview

Features summary

- Three-wire serial synchronous data transfer
- Master mode and slave mode selectable
- Slave select input signal ($\overline{\text{CSIGnTSSI}}$)
- Built-in baud rate generator
- Adjustable baud rate; in slave mode it is determined by the input clock
- Maximum transmission speed:
  - in master mode: PCLK/4
  - in slave mode: PCLK/6

Caution There might be restrictions on the maximum baud rate that can actually be used depending on the product. Specify the baud rate so as not to exceed the maximum rate for each product.

- Phase of clock and data selectable
- Data transfer with MSB or LSB first selectable
- Transfer data length selectable from 7 to 16 bits in 1-bit units
- Extended data length (EDL) function for transferring more than 16 bits of data
- Three selectable transfer modes:
  - Transmission mode
  - Reception mode
  - Transmission/reception mode
- Built-in handshake function
- Separate transmit and receive buffers (two 16-bit registers)
- Error detection (data consistency check, parity, overrun)
- Three different interrupt request signals (CSIGnTIC, CSIGnTIR, CSIGnTIRE)
- Loop back mode (LBM) function for self test

The block diagram shows the main components of CSIG.

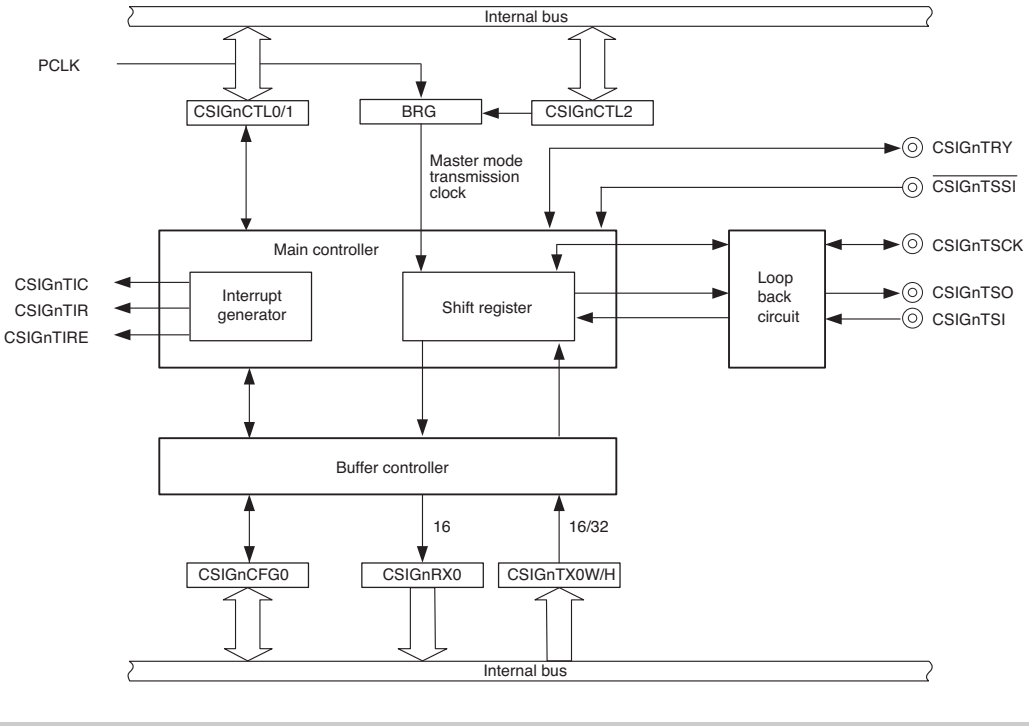


Figure 21-1 CSIG block diagram

In master mode, the serial clock CSIGnTSCK is generated by the built-in baud rate generator (BRG). In slave mode, the serial clock is supplied from an external source.

21.3 Functional Description

The Clocked Serial Interface G uses three signals for communication:

- Serial clock CSIGnTSCK (output for the master mode or input for the slave mode)
- Data output signal CSIGnTSO
- Data input signal CSIGnTSI

Additional signals are available for external control and monitoring.

- $\overline{\text{CSIGnTSSI}}$: Slave select input signal
- CSIGnTRY: Handshake signal (input for the master mode or output for the slave mode)

Whether CSIG operates in the master or slave mode can be specified by using the CSIGCTL2 register.

Data transmission is bit-wise and serial and synchronous to the serial clock.

The most important registers for setting up CSIG are:

Register	Function
CSIGnCTL0	Enables/disables the operation clock (PCLK), data transmission, and data reception
CSIGnCTL1	Controls options like interrupt timing, extended data length, data consistency check, loop-back mode, handshake, etc.
CSIGnCTL2	Selects master/slave mode and – effective in master mode – the baud rate of the internal baud rate generator (BRG)
CSIGnCFG0	Configures the communication protocol

21.3.1 Operating modes (master/slave)

Master/slave selection is performed by using the CSIGnCTL2.CSIGnPRS[2:0] bits, and, when the master is selected, the source clock of the serial clock must also be selected.

(1) Master mode

In the master mode, the serial clock is generated by the internal baud rate generator (BRG) and provided to the slave(s) by signal CSIGnTSCK.

Master mode is enabled by setting CSIGnCTL2.CSIGnPRS[2:0] to anything but 111_B. In the master mode, the BRG frequency can be specified by specifying values for the CSIGnCTL2.CSIGnPRS[2:0] and CSIGnCTL2.CSIGnBRS[11:0] bits in combination.

Clock defaults The default level of CSIGnTSCK depends on the clock phase selection bit: It is high when CSIGnCTL1.CSIGnCKR = 0, and is low when CSIGnCTL1.CSIGnCKR = 1.

The example below shows the communication in master mode for eight data bits when CSIGnCTL1.CSIGnCKR = 0, CSIGnCFG0.CSIGnDAP = 0, and MSB first:

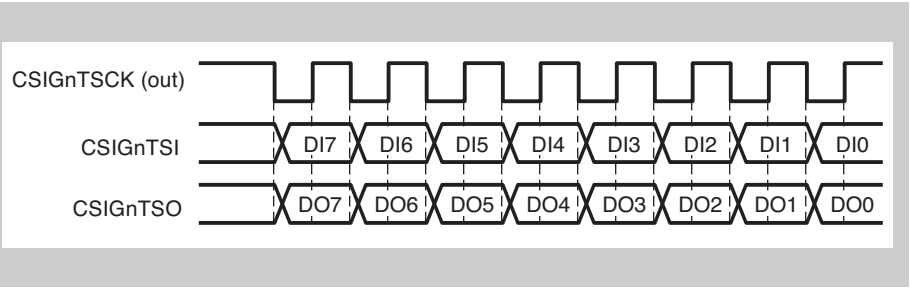


Figure 21-2 Transmission/reception in master mode

(2) Slave mode

In the slave mode, another device is the communication master. The external clock is supplied through the CSIGnTSCK signal. When the serial clock signal is detected, a transmission or reception operation immediately starts.

Slave mode is selected by setting CSIGnCTL2.CSIGnPRS[2:0] to 111_B.

Note When using the slave mode, the baud rate generator (BRG) can be disabled by clearing the CSIGnCTL2.CSIGnBRS[11:0] bits, reducing power consumption.

The example below shows the communication in the save mode for eight data bits when CSIGnCTL1.CSIGnCKR = 0, CSIGnCFG0.CSIGnDAP = 0, and the MSB is first.

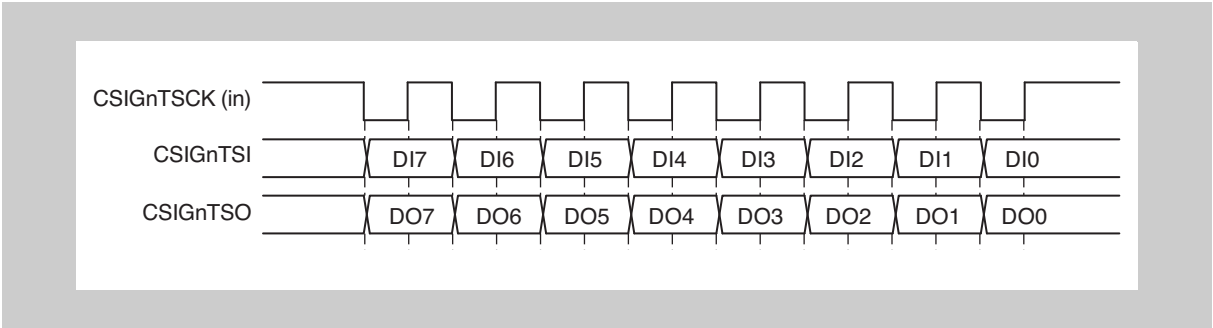


Figure 21-3 Transmission/reception in slave mode

21.3.2 Master/slave connections

(1) One master and one slave

The following figure illustrates the connections between one master and one slave.

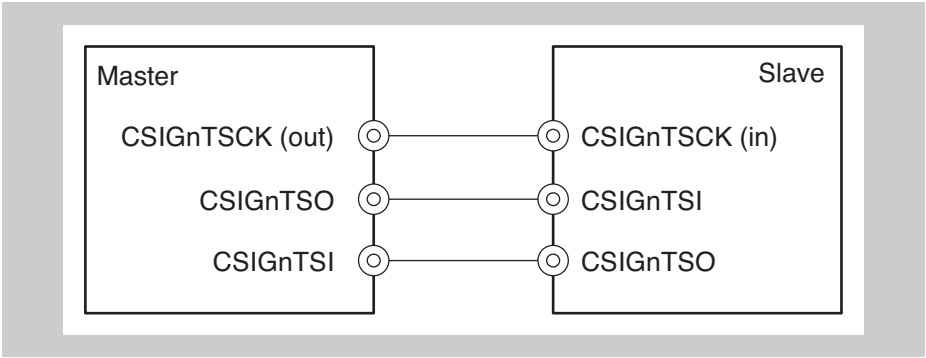


Figure 21-4 Simple master/slave connection

(2) One master and multiple slaves

The following figure illustrates the connections between one master and multiple slaves. In this example, a configuration in which the master supplies one slave select (SS) signal to each slave is possible. This signal is connected to the slave select input $\overline{\text{CSIGnTSSI}}$ of the slave.

The $\overline{\text{CSIGnTSSI}}$ signal recognition function can be enabled/disabled by bit CSIGnCTL1.CSIGnSSE.

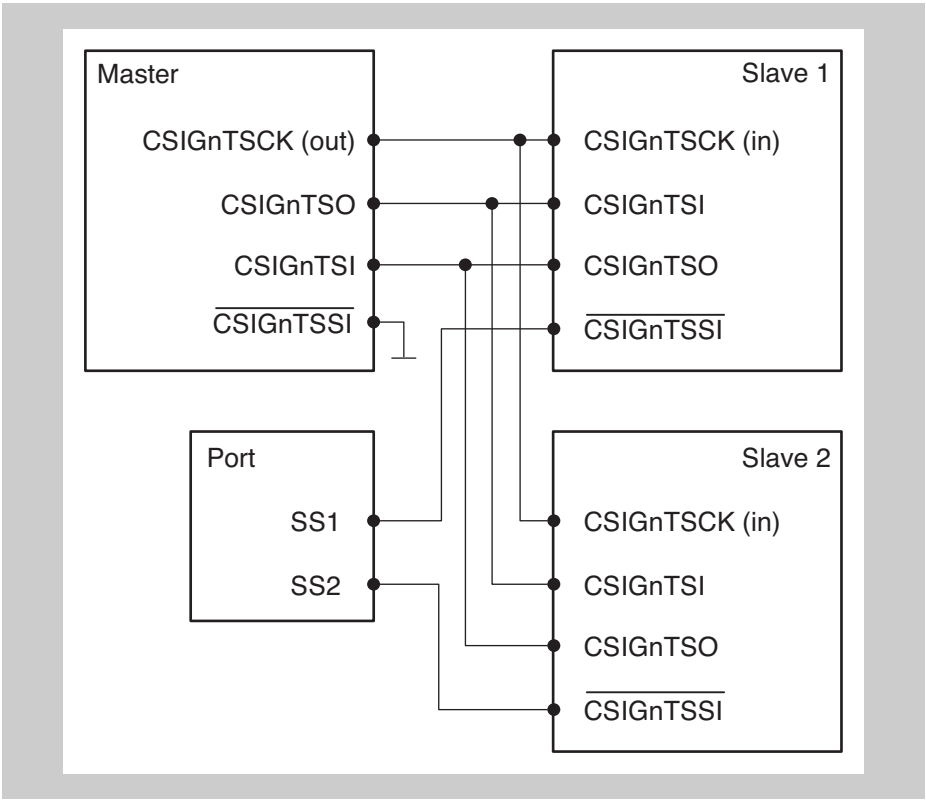


Figure 21-5 Master to multiple slaves connection

A slave is selected (enabled) when its $\overline{\text{CSIGnTSSI}}$ signal is low.

If a slave is not selected, it will neither receive nor transmit data. In addition, its output CSIGnTSO is set to input mode in order to avoid interference with the output of another slave that was selected.

(3) CSIGnTSO output control

CSIG can output CSIGnTSO when all of the following conditions are satisfied:

- The CSIG is enabled (CSIGnCTL0.CSIGnPWR = 1).
- The CSIG is operated in transmission or transmission/reception mode (CSIGnCTL0.CSIGnTXE = 1).
- The CSIG is operated with slave select enabled (CSIGnCTL1.CSIGnSSE = 1).
- The slave mode selection signal $\overline{\text{CSIGnTSSI}}$ is inactive, i.e. on high level.

By this signal congestions on the external CSIGnTSO signal line are avoided.

21.3.3 Serial clock selection

In master mode, the transmission baud rate is selectable using the CSIGnPRS[2:0] and CSIGnBRS[11:0] bits in the CSIGnCTL2 register.

The following figure shows a block diagram of the BRG.

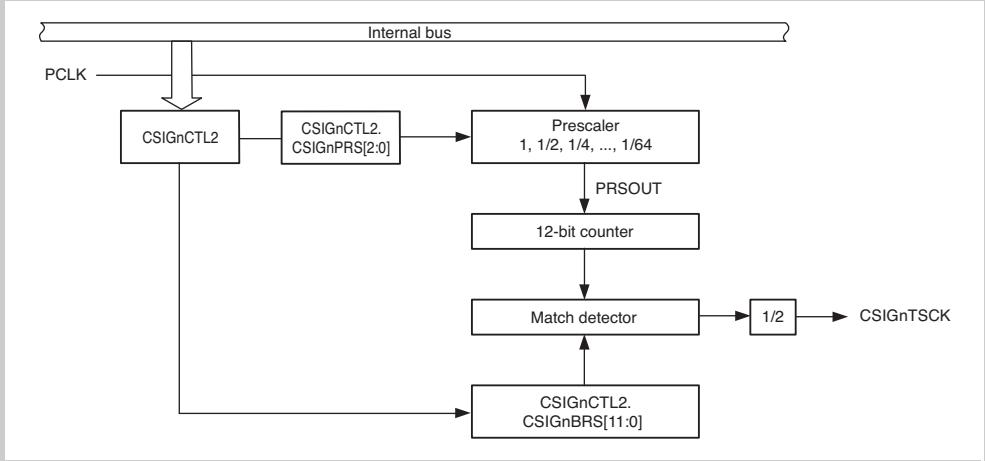


Figure 21-6 BRG block diagram

Clearing CSIGnCTL2.CSIGnBRS[11:0] disables the BRG.

Baud rate calculation

The baud rate is calculated as: PCLK / (2^m × k × 2), where

m = CSIGnPRS[2:0] = 0 to 6

k = CSIGnBRS[11:0] = 1 to 4095

Baud rate limits

When setting the baud rate, please note:

- Maximum acceptable baud rate in master mode is PCLK / 4.
- Maximum acceptable baud rate in slave mode is PCLK / 6.
- Minimum baud rate in both modes is PCLK / 524160.

Caution There might be restrictions on the maximum baud rate that can actually be used depending on the product. Specify the baud rate so as not to exceed the maximum rate for each product.

Example

If PCLK = 66.7 MHz, the maximum baud rate is

- 16.667 Mbps (PCLK / 4) in master mode
- 11.111 Mbps (PCLK / 6) in slave mode

The slowest baud rate is 127.188 bps (PCLK / 524160).

21.3.4 Data transfer modes

(1) Transmission mode

Setting CSIGnCTL0.CSIGnTXE = 1 and CSIGnCTL0.CSIGnRXE = 0 puts CSIG in transmission mode. Transmission starts when transmission data is written in the CSIGnTX0W or CSIGnTX0H register.

(2) Reception mode

Setting CSIGnCTL0.CSIGnTXE = 0 and CSIGnCTL0.CSIGnRXE = 1 puts CSIG in reception mode.

In the master mode, reception starts when dummy data is read from the CSIGnRX0 register.
All following receptions are triggered by a read from the reception data register CSIGnRX0, as long as CSIGnBCTL0.CSIGnSCE = 1.

In the slave mode, reception starts when the serial clock CSIGnTSCK is supplied from the master.

Note In reception mode, any previously received data must be read from the reception data register CSIGnRX0 in order to avoid any overwrite situation.

Moreover the communication start bit CSIGnBCTL0.CSIGnSCE has to be set to 1 and has to set back to 0 before reading the last received data from CSIGnRX0.

The recommended procedure is:

1. Set CSIGnBCTL0.CSIGnSCE = 1.
2. Read CSIGnRX0 (dummy data).
3. Wait for the reception interrupt CSIGnTIR.
4. Read CSIGnRX0 (received data).
   In case of further data receptions continue at step 3, repeat a read operation until all data has been received.
   Before reading the last received data from CSIGnRX0, clear CSIGnBCTL0.CSIGnSCE.

(3) Transmission/reception mode

Setting CSIGnCTL0.CSIGnTXE = 1 and CSIGnCTL0.CSIGnRXE = 1 puts CSIG in transmission/reception mode.

Communication starts when transmission data is written to the CSIGnTX0W or CSIGnTX0H register.

21.3.5 Data length selection

(1) Data length between 7 and 16 bits

Transmission data length is selectable from 7 to 16 bits using the CSIGnCFG0.CSIGnDLS[3:0] bits. The examples below show the communication with MSB first (CSIGnCFG0.CSIGnDIR = 0):

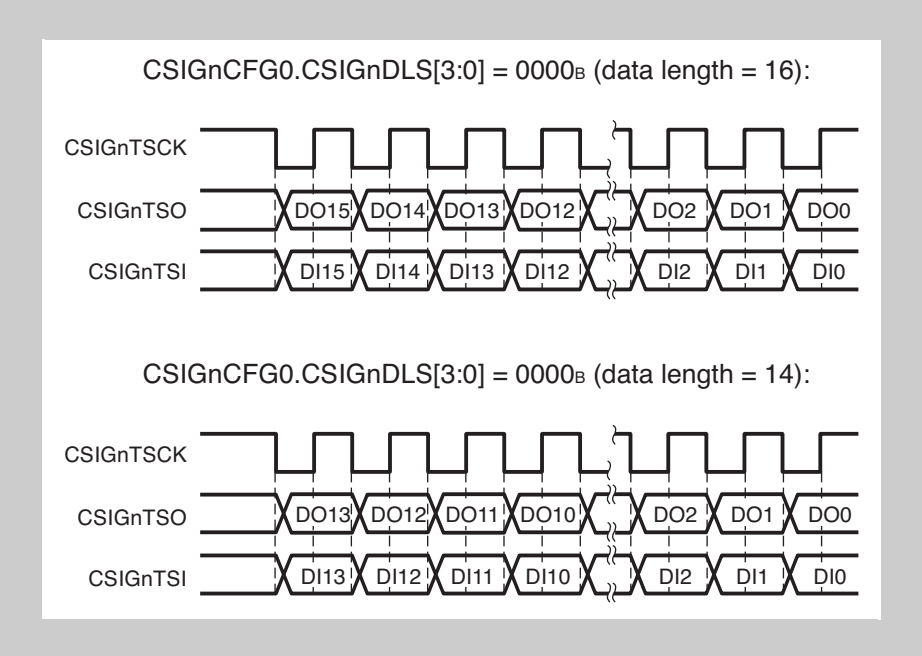


Figure 21-7 Data length select function

(2) Data length greater than 16 bits

If the length of the data to be sent/received exceeds 16 bits, the extended data length (EDL) feature can be used.

The EDL function is enabled by setting CSIGnCTL1.CSIGnEDLE.

The EDL function works as follows:

- Data is divided into 16-bit blocks and a remainder. For example, a 42-bit character string is divided into two 16-bit blocks and a 10-bit remainder.
- For the remainder, the data length is specified for the CSIGnCFG0.CSIGnDLS[3:0] bits.
- When transmitting 16-bit blocks, set the CSIGnTX0W.CSIGnEDL bit. In this case, the data written to CSIGnTX0W is sent as a 16-bit data length regardless of the CSIGnCFG0.CSIGnDLS[3:0] bits.
- When the specified length of data (the remainder when CSIGnTX0W.CSIGnEDL = 0) is transmitted, the transfer ends.

Example Example of transmitting the 40-bit data $123456789A_H$

The 40-bit data is divided into two 16-bit blocks of data and one 8-bit block of data.

- Initialize CSIGnCFG0.CSIGnDLS[3:0] = 8_D.
- To send the string $123456789A_H$ with MSB first, write the following sequence to CSIGnTX0W:
  - $2000\ 1234_H$ (CSIGnTX0W.CSIGnEDL = 1)
  - $2000\ 5678_H$ (CSIGnTX0W.CSIGnEDL = 1)
  - $0000\ 009A_H$ (CSIGnTX0W.CSIGnEDL = 0)

The following figure illustrates the timing.

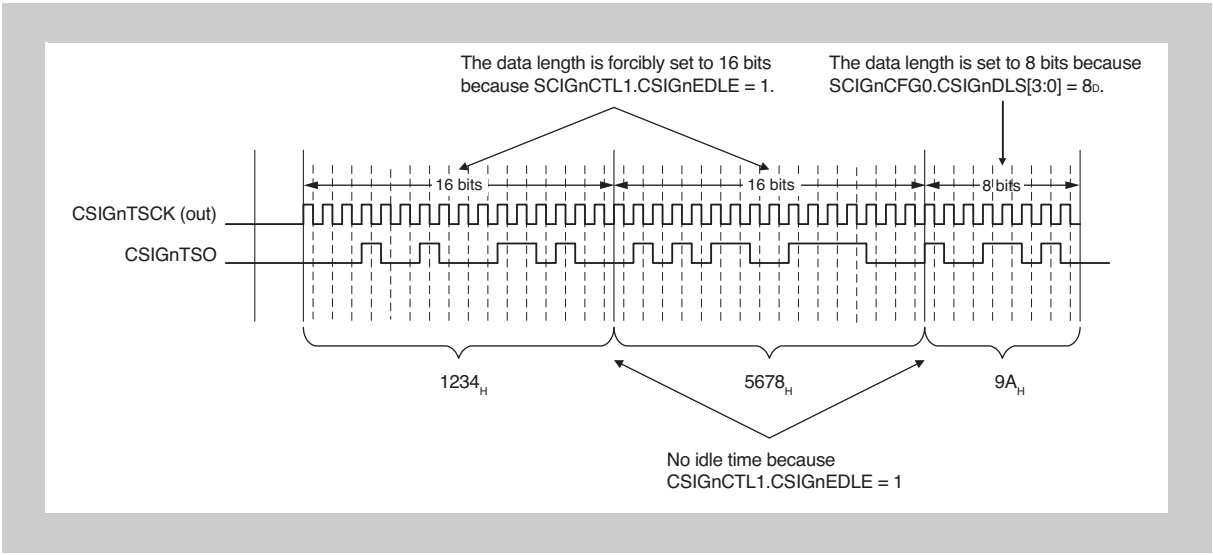


Figure 21-8 EDL timing chart

Notes

1. A data length less than 7 bits can be specified only when the EDL function is used.
2. It is not possible to send two consecutive data with a data length of less than 7 bits.
3. If parity is enabled, the parity bit is added after the last bit.
4. The following describes an example where the transmitted data is 123456_H.
   - CSIGnCFG0.CSIGnDIR is cleared to 0 (MSB first).
     $2000\ 1234_H$ is written to CSIGnTX0W (CSIGnTX0W.CSIGnEDL = 1).
     $0000\ 0056_H$ is written to CSIGnTX0W (CSIGnTX0W.CSIGnEDL = 0).
   - CSIGnCFG0.CSIGnDIR is set to 1 (LSB first).
     $2000\ 3456_H$ is written to CSIGnTX0W (CSIGnTX0W.CSIGnEDL = 1).
     $0000\ 0012_H$ is written to CSIGnTX0W (CSIGnTX0W.CSIGnEDL = 0).
5. The EDL function cannot be used in the slave mode (CSIGnCTL1.CSIGnPRS[2:0] = 1, 1, 1) and reception mode (CSIGnCTL0.CSIGnTXE = 0, CSIGnCTL0.CSIGnRXE = 1).

21.3.6 Serial data direction selection

The serial data direction is selectable using the CSIGnDIR bit in the CSIGnCFG0 register.

The examples below show the communication for a data length of 8 bit (CSIGnCFG0.CSIGnDLS[3:0] = 1000_B):

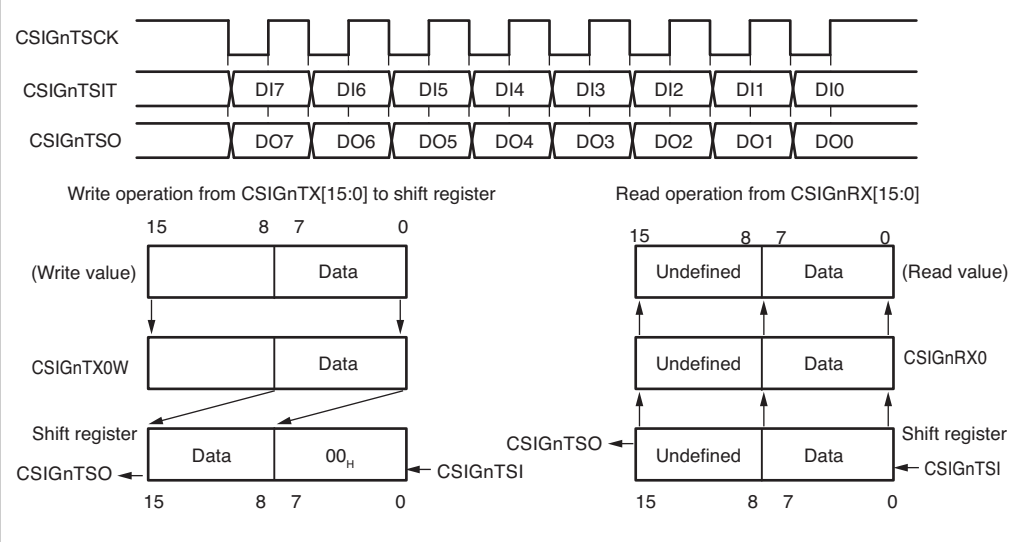


Figure 21-9 Serial data direction select function - MSB first (CSIGnDIR = 0)

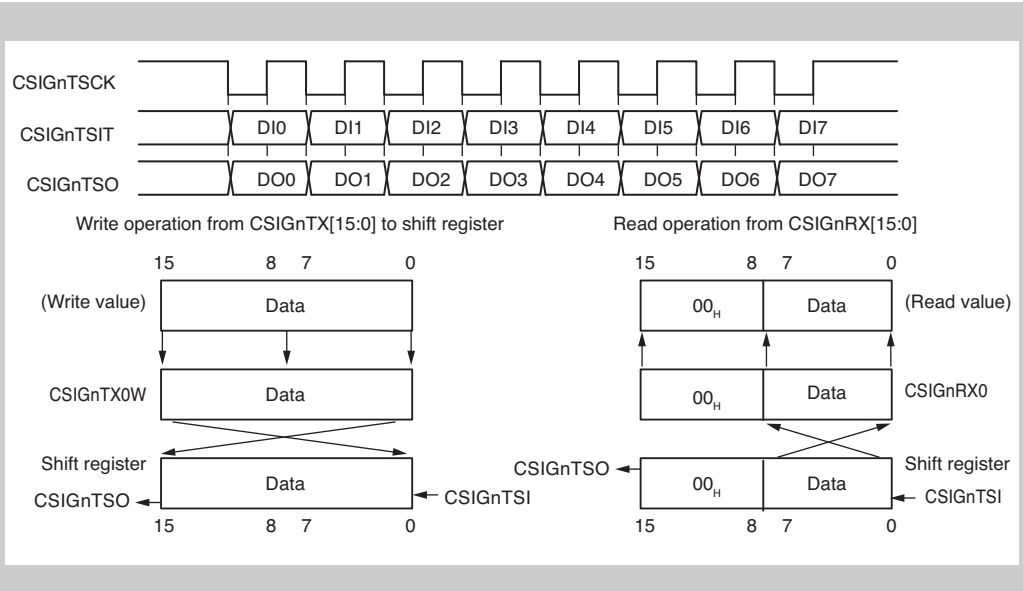


Figure 21-10 Serial data direction select function - LSB first (CSIGnDIR = 1)

21.3.7 Communication in slave mode

The following figure illustrates the communication signals and timings in slave mode.

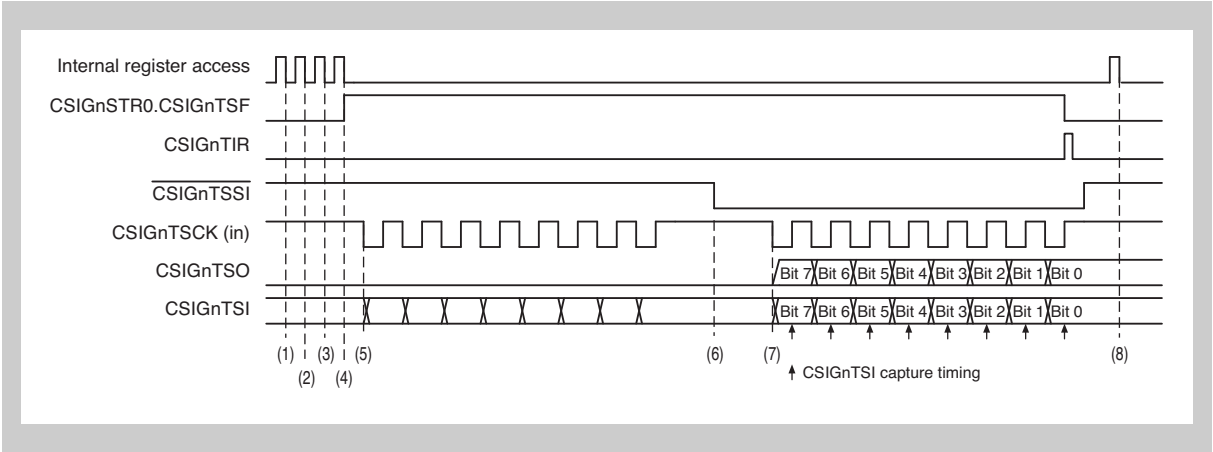


Figure 21-11 Transmit/receive communication timing in slave mode

1. CSIG is placed in the slave mode by setting CSIGnCTL2.CSIGnPRS[2:0] to 111_B. The $\overline{\text{CSIGnTSSI}}$ signal is enabled (CSIGnCTL1.CSIGnSSE = 1).
2. CSIGnCTL1.CSIGnCKR and CSIGnCFG0.CSIGnDAP0 are 0, the data length is 8 bits (CSIGnCFG0.CSIGnDLS0[3:0] = 1000_B), and the data direction is MSB first (CSIGnCFG0.CSIGnDIR0 = 0).
3. CSIG is set to the transmission/reception mode (CSIGnCTL0.CSIGnPWR = 1, CSIGnCTL0.CSIGnTXE = 1, and CSIGnCTL0.CSIGnRXE = 1). Communication is enabled to start.
4. If transfer data is written to the transmission data register CSIGnTX0W or CSIGnTX0H, the transfer status flag CSIGnSTR0.CSIGnTSF is automatically set, and the system waits for the $\overline{\text{CSIGnTSSI}}$ signal to become low.
5. Data transmission/reception does not start while the $\overline{\text{CSIGnTSSI}}$ signal is high, even if the external serial clock CSIGnTSCK is input. CSIGnTSO retains the value, and input to CSIGnTSI is ignored.
6. When the $\overline{\text{CSIGnTSSI}}$ signal becomes low, it indicates that CSIGnTSO is enabled and data transmission is possible.
7. If a serial clock is input while $\overline{\text{CSIGnTSSI}}$ is low, the transfer data is transmitted from CSIGnTSO in synchronization with the serial clock, and data is received from CSIGnTSI at the same time.
8. The CSIGnRX0 register is read.

21.3.8 CSIG interrupts

CSIG can generate the following interrupts:

- CSIGnTIC (communication interrupt)
- CSIGnTIR (reception interrupt)
- CSIGnTIRE (error interrupt)

(1) CSIGnTIC (communication interrupt)

This interrupt is normally generated after every data transfer. It can be used to trigger a DMA for writing new transmission data to register CSIGnTX0W or CSIGnTX0H.

The following example assumes:

- Master mode
- CSIGnCTL1.CSIGnSIT = 0 (no interrupt delay)
- CSIGnCTL1.CSIGnCKR = 0, CSIGnCFG0.CSIGnDAP = 0 (normal clock and data phase)
- CSIGnCFG0.CSIGnDLS[3:0] = 1000_B (data length 8 bits)
- CSIGnCTL1.CSIGnSLIT = 0 (normal interrupt timing)

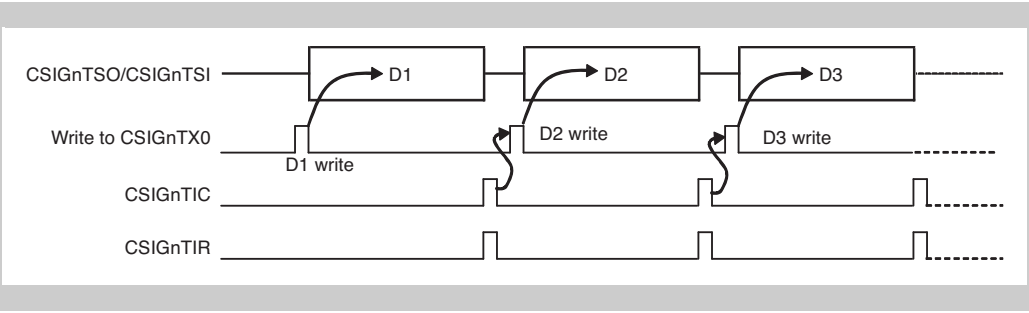


Figure 21-12 Generation of CSIGnTIC after communication (CSIGnCTL1.CSIGnSLIT = 0)

However, CSIGnTIC can also be set up to occur when the CSIGnTX0W or CSIGnTX0H register is free for the next data. This is specified by setting CSIGnCTL1.CSIGnSLIT = 1.

This mode allows more efficient data transfers.

The effect is illustrated in the figure below.

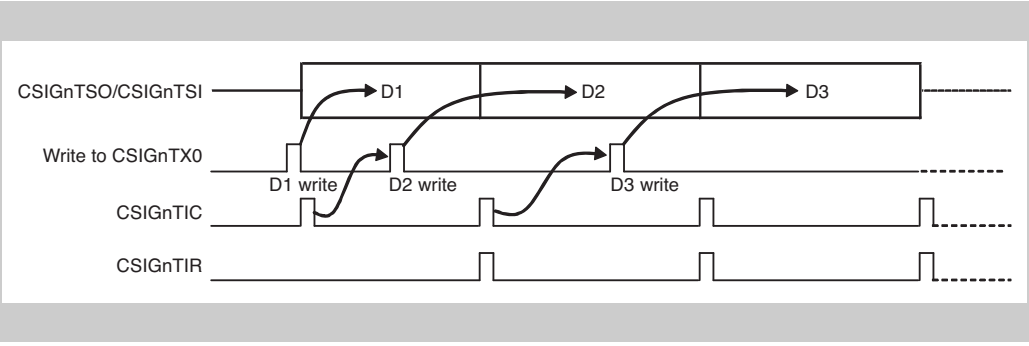


Figure 21-13 Generation of CSIGnTIC at the beginning of communication

(2) CSIGnTIR (reception interrupt)

This interrupt is generated in reception and transmission/reception mode after data has been received and is available in the reception data register.

The following example assumes:

- Master mode
- CSIGnCTL1.CSIGnSIT = 0 (no interrupt delay)
- CSIGnCTL1.CSIGnCKR = 0, CSIGnCFG0.CSIGnDAP = 0 (normal clock and data phase)
- CSIGnCFG0.CSIGnDLS[3:0] = 1000_B (data length 8 bits).

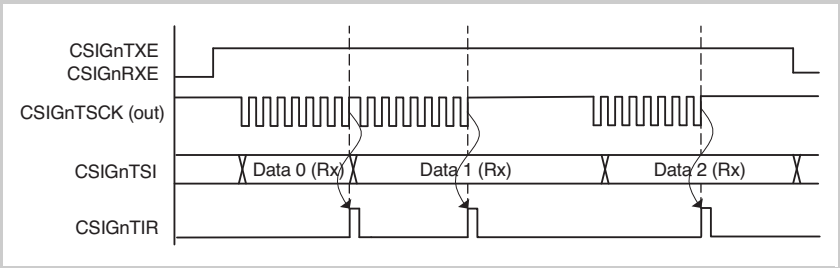


Figure 21-14 Generation of CSIGnTIR

(3) CSIGnTIRE (reception error interrupt)

This interrupt is generated whenever an error is detected.

Table 21-7 Data error types

Error type	Communication status after error interrupt
Parity error	Interrupt is generated and communication continues
Data consistency error	Interrupt is generated and communication continues
Overrun error	If CSIGnCTL1.CSIGnHSE = 0 (no handshake), communication continues after the interrupt is generated. (Communication does not stop.)
	If CSIGnCTL1.CSIGnHSE = 1 (handshake exists), communication is stopped according to the handshake. No interrupt is generated, and no overrun error occurs.

The type of error that caused the generation of CSIGnTIRE is indicated in register CSIGnSTR0.

For details about the various error types, refer to *21.3.10 “Error detection” on page 1270*.

(4) Interrupt delay

In the master mode, all interrupts generated by the master can be delayed one half cycle of the serial clock CSIGnTSCK. This function cannot be used in the slave mode.

To specify this delay, set the CSIGnCTL1.CSIGnSIT bit to 1.

The figure below shows an example of using the interrupt delay function with the following settings: CSIGnCTL1.CSIGnSIT = 1 (interrupt delay enabled), CSIGnCTL1.CSIGnCKR = 0, CSIGnCFG0.CSIGnDAP = 0 (normal clock phase and data phase), and CSIGnCFG0.CSIGnDLS[3:0] = 1000_B (8-bit data length).

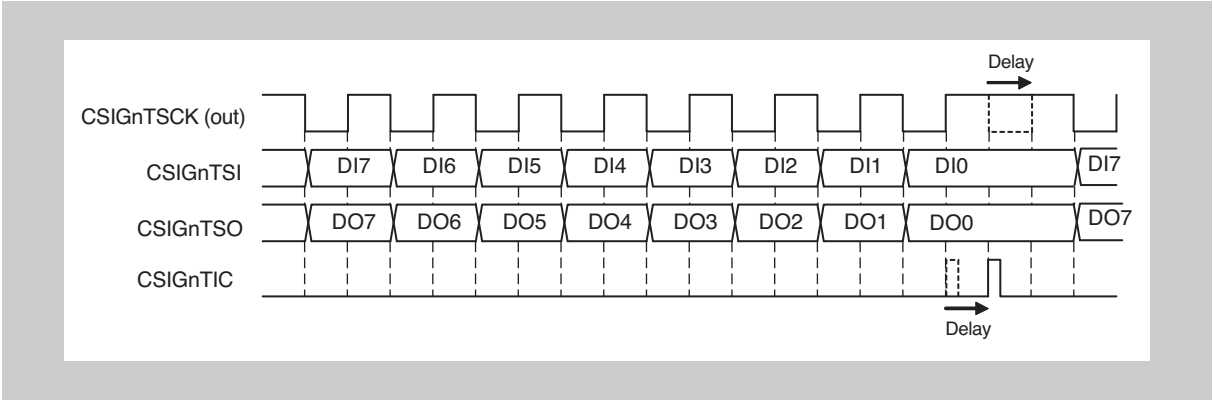


Figure 21-15 Interrupt delay function (CSIGnCTL1.CSIGnSIT = 1)

21.3.9 Handshake function

CSIG features a handshake function to synchronize the master and the slave devices. This function can be enabled/disabled by bit CSIGnCTL1.CSIGnHSE. For handshake, the signal CSIGnTRY is used. In addition, when the handshake function is disabled (CSIGnCTL1.CSIGnHSE = 0), the CSIGnTRY signal is output at the low level.

The timing depends on the data phase selection bit CSIGnCFG0.CSIGnDAP.

(1) Slave mode

When CSIGnCTL1.CSIGnHSE = 1 and the slave is busy, the CSIGnTRY signal is output at the low level. This happens when previous receive data is still in the CSIGnRX0 register, and new data cannot be copied from the shift register to CSIGnRX0 (CSIGnRX0 full condition).

The following examples assume a data length of 8 bits.

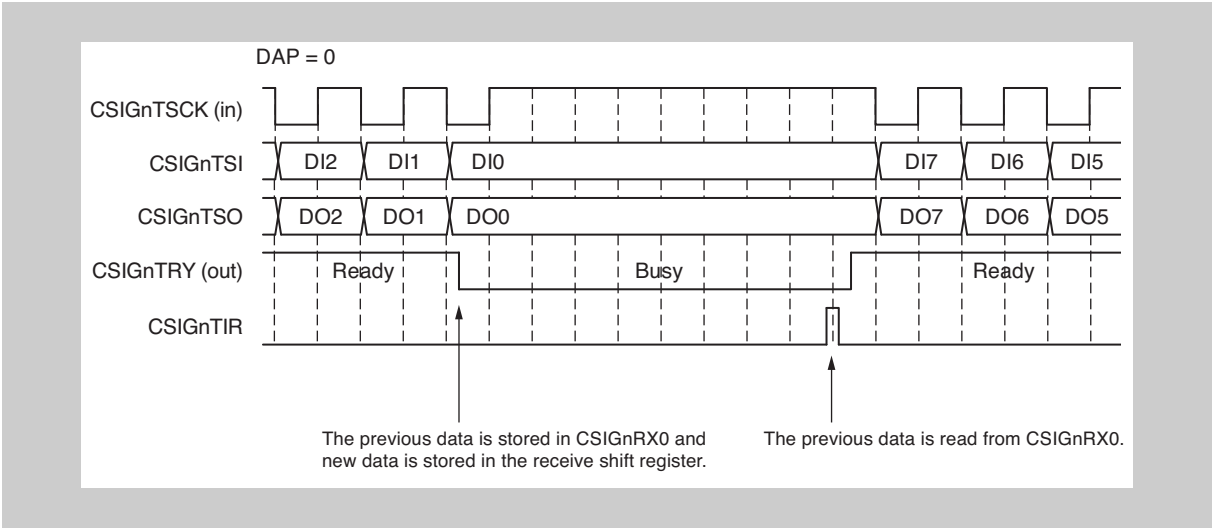


Figure 21-16 Ready/busy signal from slave (CSIGnCFG0.CSIGnDAP = 0)

As long as the slave is busy, the master has to wait (i.e. suspend the serial clock). The slave sets CSIGnTRY to high (“ready”) as soon as the reception data register CSIGnRX0 has been read.

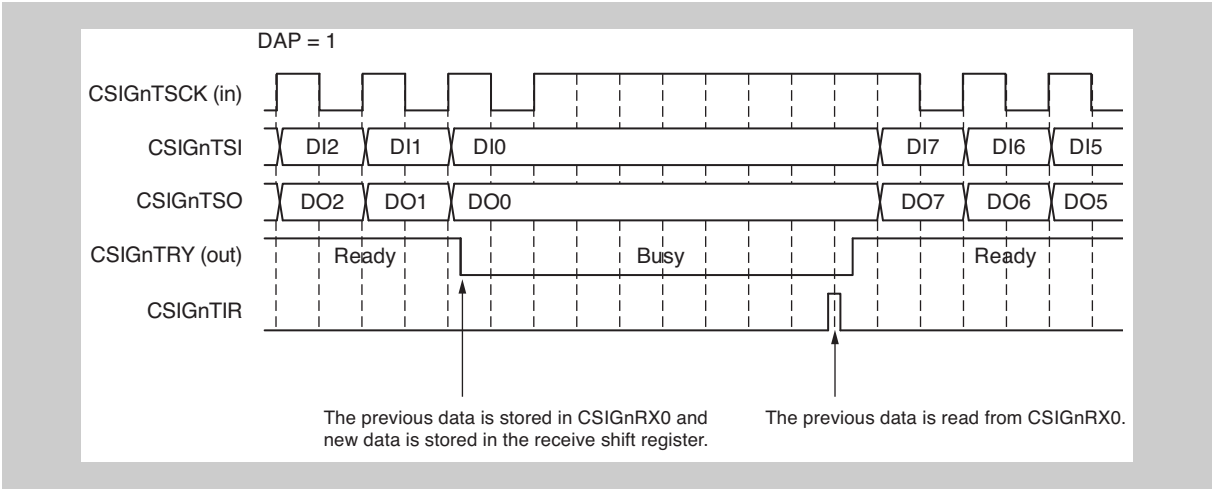


Figure 21-17 Ready/busy signal from slave (CSIGnCFG0.CSIGnDAP = 1)

(2) Master mode

When the master detects the CSIGnTRY is at the low level, the following transfer is put on hold, and the master goes into wait status. It suspends the clock CSIGnTSCK.

The CSIGnTRY level is checked at each half clock cycle of CSIGnTSCK.

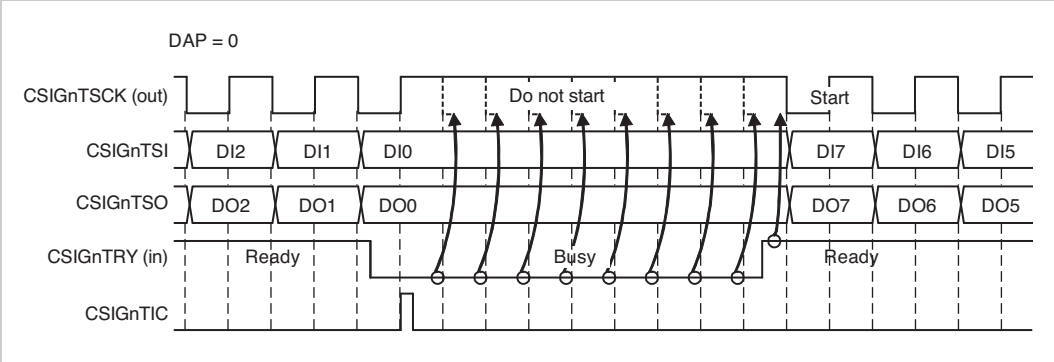


Figure 21-18 Master's reaction to CSIGnTRY (CSIGnCFG0.CSIGnDAP = 0)

If the CSIGnTRY low signal comes from the slave while data transfer is in progress, the serial clock is suspended after the transfer is complete.

The master resumes the communication as soon as CSIGnTRY becomes high (slave is "ready").

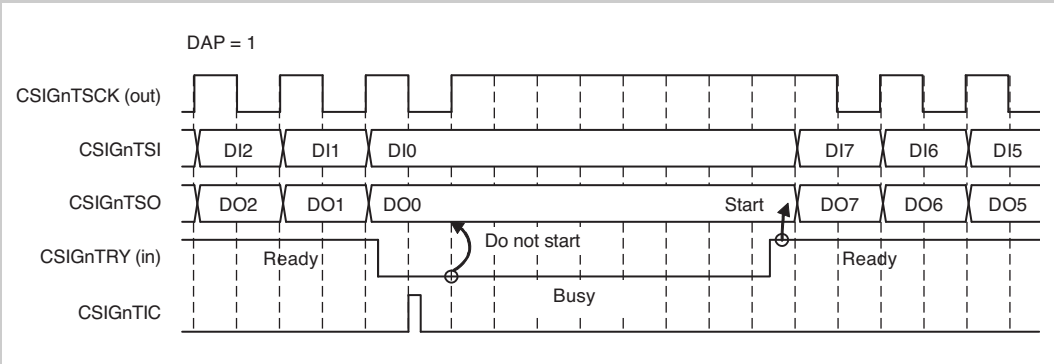


Figure 21-19 Master's reaction to CSIGnTRY (CSIGnCFG0.CSIGnDAP = 1)

Caution If multiple slaves are connected, the master must only detect the CSIGnTRY signal of the slave it has selected for communication.

CSIGnTRY must be pulled down by the external slave before the next transfer starts. Even if the signal is pulled down by the slave during the transfer, the transfer continues.

21.3.10 Error detection

CSIG can detect three error types:

- Data consistency error (transmission data)
- Parity error (reception data)
- Overrun error

Check for data consistency and parity errors can be enabled/disabled individually.

If one of these errors is detected, the interrupt CSIGnTIRE is generated and the corresponding flag is set.

(1) Data consistency check

The purpose of the data consistency check is to ensure that the data physically sent to the output signal is identical with the original data that was copied to the shift register.

The data consistency check can be enabled/disabled by bit CSIGnCTL1.CSIGnDCS. It is not active if data transmission is disabled (CSIGnCTL0.CSIGnTXE = 0).

When the data consistency check is active, the data transferred from CSIGnTX0W or CSIGnTX0H to the shift register is copied to a separate register. In addition, the physical levels at output signal CSIGnTSO are captured, and their logical interpretation is fed into an own shift register.

After completion of the transmission, the data sent is compared with the original transmission data.

Mismatch is considered as a data consistency error.

When a data consistency error occurs:

- Interrupt CSIGnTIRE is generated.
- Bit CSIGnSTR0.CSIGnDCE is set.

The function is illustrated in the following block diagram.

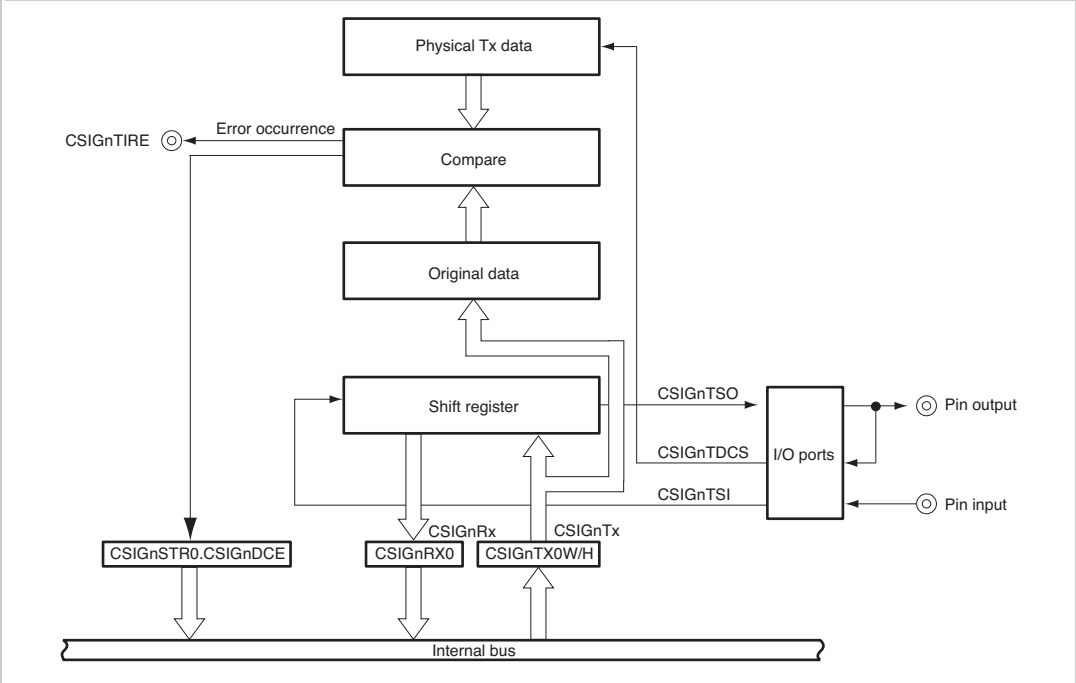


Figure 21-20 Functional block diagram of the data consistency check

(2) Parity check

Parity is a common mean to detect a single bit failure during data transmission. CSIG can append a parity bit to the last data bit (even if extended data length is used).

The use and type of parity is specified in CSIGnCFG0.CSIGnPS[1:0].

Parity check is enabled if CSIGnCFG0.CSIGnPS[1] = 1.

The parity bit is checked after reception is complete.

When a parity error occurs:

- Interrupt CSIGnTIRE is generated
- Bit CSIGnSTR0.CSIGnPE is set

The following figure shows an example.

- Data length is 8 bits.
- The data transmitted is 05_H and 35_H.
- Data direction is LSB first.
- Parity type is odd.

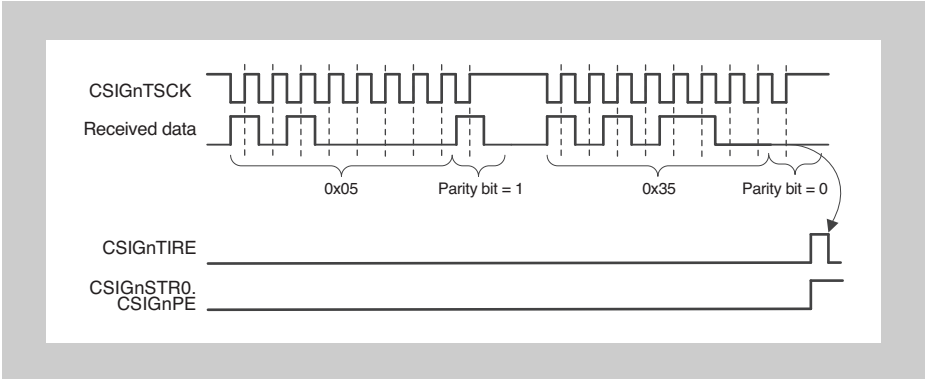


Figure 21-21 Parity check example

For the first 8 bits, the parity bit is 1. There is no parity error, because the total number of ones (including the parity bit) is odd.

For the second 8 bits, the parity bit is 0. This is detected as a parity error, because the total number of ones (including the parity bit) is even.

If using the extended data length (EDL) function, the parity bit is added after the last data bit.

(3) Overrun error

This error occurs when newly received data cannot be transferred from the shift register to the reception data register CSIGnRX0. This happens when CSIGnRX0 was not read and therefore contains previously received data.

In the master mode, because the serial clock is stopped until the CPU reads reception data, overrun errors do not occur. In the slave mode, the handshake function can be used to avoid this error.

When an overrun error occurs:

- The interrupt CSIGnTIRE is generated.
- CSIGnSTR0.CSIGnOVE is set.
- The CSIGnRX0 register is written with the reception data.
- Communication continues.

The following figure illustrates the function.

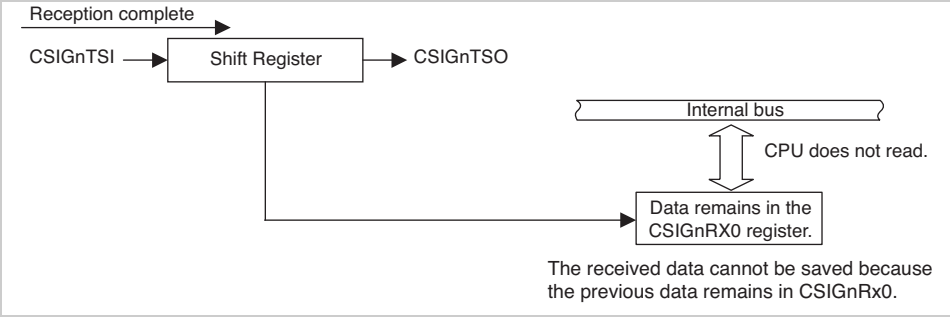


Figure 21-22 Overrun error detection

The following figure illustrates an example where:

- Rx data 3 was not read
- Rx data 4 was received, but cannot be stored

Thus an overrun error occurs.

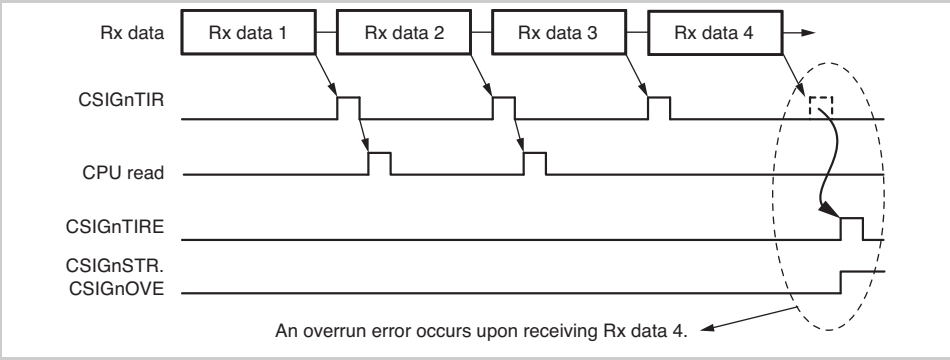


Figure 21-23 Overrun error detection - example

Note An overrun error can be avoided in slave mode by using the handshake.

When handshake is used in slave mode, the receiver (slave) signals to the transmitter (master) that it is busy. The transmitter then waits until the receiver has read its reception data register and is ready again.

For details, see *21.3.9 “Handshake function” on page 1268*.

21.3.11 Loop-back mode

Loop-back mode is a special mode for self-test. This feature is only available in master mode.

When this mode is active, the transmission and reception signals are internally connected, as shown in the figures below. The signals CSIGnTSCK, CSIGnTSO, and CSIGnTSI are disconnected from the ports. In addition, the CSIGnTSO output level is fixed to low, and CSIGnTSCK becomes inactive. The handshake function cannot be used at this time. The rest of CSIG works as in normal operation.

To perform a self-test of the CSIG, CSIGnCTL1.CSIGnLBM is set to 1, and a normal transfer operation is executed. Next, whether the reception data and transmission data are the same is checked.

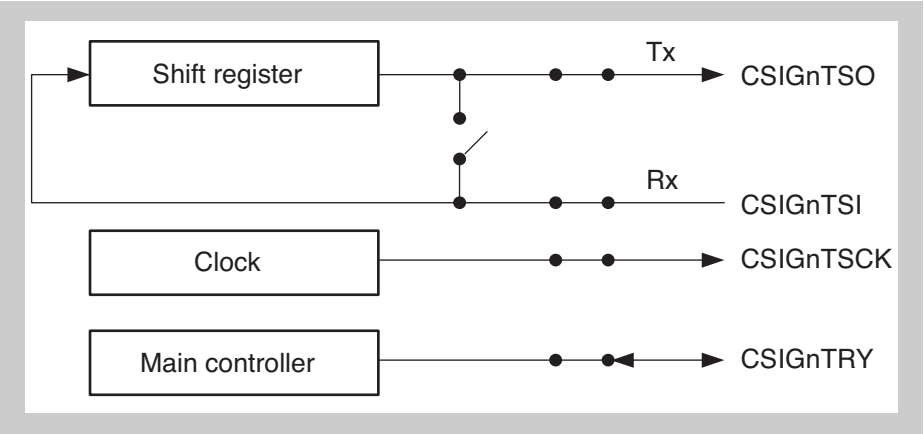


Figure 21-24 Normal operation (CSIGnLBM = 0)

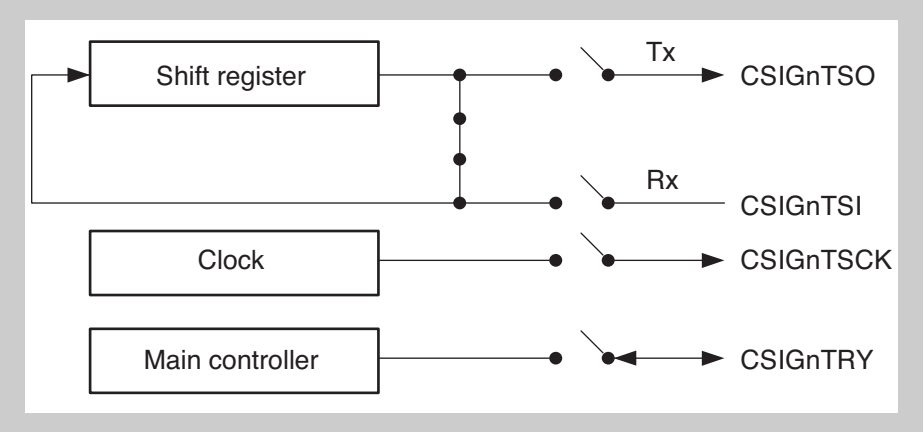


Figure 21-25 Operation in loop-back mode (CSIGnLBM = 1)

21.4 CSIG Control Registers

CSIGn is controlled and operated by means of the following registers:

Table 21-8 CSIGn register overview

Register name	Shortcut	Address
Control register 0	CSIGnCTL0	<CSIGn_base_USER> + 0000_H
Control register 1	CSIGnCTL1	<CSIGn_base_OS> + 0010_H
Control register 2	CSIGnCTL2	<CSIGn_base_OS> + 0014_H
Status register 0	CSIGnSTR0	<CSIGn_base_USER> + 0004_H
Status clear register 0	CSIGnSTCR0	<CSIGn_base_USER> + 0008_H
Reception mode control register 0	CSIGnBCTL0	<CSIGn_base_USER> + 0080_H
Configuration register 0	CSIGnCFG0	<CSIGn_base_OS> + 1010_H
Transmission data register 0 for word access	CSIGnTX0W	<CSIGn_base_USER> + 0084_H
Transmission data register 0 for half word access	CSIGnTX0H	<CSIGn_base_USER> + 0088_H
Reception data register 0	CSIGnRX0	<CSIGn_base_USER> + $008C_H$

<CSIGn_base> The base addresses <CSIGn_base> of CSIGn is defined in the first section of this chapter under the key word “Register addresses”.

(1) CSIGnCTL0 - CSIG control register 0

This register controls the operation clock and enables/disables transmission/ reception.

Access This register can be read/written in 8-bit or 1-bit units.

Address <CSIGn_base_USER> + 0000_H

Initial Value 00_H. This register is initialized by any reset.

Caution Be sure to set “0” to bits 4 to 1 and “1” to bit 0.

7	6	5	4	3	2	1	0
CSIGn PWR	CSIGn TXE	CSIGn RXE	0	0	0	0	a
R/W	R/W	R/W	R	R	R	R	R/W

a) Be sure to set “1” to bit 0 though the initial value is 0.

Table 21-9 CSIGnCTL0 register contents

Bit position	Bit name	Function
7	CSIGnPWR	Controls operation clock. 0: Stop operation clock 1: Provide operation clock Clearing CSIGnPWR to 0 resets the internal circuits, stops operation, and sets CSIG to standby state. No clock is provided to internal circuits. If CSIGnPWR is cleared during communication, ongoing communication is immediately aborted. A restart of the communication is then required.
6	CSIGnTXE	Enables/disables transmission. 0: Transmission disabled 1: Transmission enabled
5	CSIGnRXE	Enables/disables reception. 0: Reception disabled 1: Reception enabled

Cautions

1. When CSIGnPWR = 0, do not change bits 6 (CSIGnTXE), 5 (CSIGnRXE), and 0.
   However, these bits can be changed at the same time that the CSIGnPWR bit changes from 0 to 1.
2. Do not modify CSIGnTXE or CSIGnRXE while a data transmission is pending or ongoing, i.e. if CSIGnSTR0.CSIGnTSF = 1.

(2) CSIGnCTL1 - CSIG control register 1

This register specifies the interrupt timing and the interrupt delay mode. It enables/disables extended data length control, data consistency check, loop-back mode, handshake function, and slave select function.

Access This register can be read/written in 32-bit units.

Address <CSIGn_base_OS> + 0010_H

Initial Value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	CSIGnCKR	CSIGnSLIT
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	CSIGnEDLE	0	CSIGnDCS	0	CSIGnLBM	CSIGnSIT	CSIGnHSE	CSIGnSSE
R	R	R	R	R	R	R	R	R/W	R	R/W	R	R/W	R/W	R/W	R/W

Caution Changing the contents of this register is only permitted when CSIGnCTL0.CSIGnPWR = 0.

Table 21-10 CSIGnCTL1 register contents (1/2)

Bit position	Bit name	Function
17	CSIGnCKR	Selects the CSIGnTSCK clock phase. 0: The CSIGnTSCK default level is high. 1: The CSIGnTSCK default level is low. This bit is used in combination with the CSIGnCFG0.CSIGnDAP bit. For details, refer to *(7) "CSIGnCFG0 - CSIG configuration register 0" on page 1284*.
16	CSIGnSLIT	Selects the timing of interrupt CSIGnTIC. 0: Normal interrupt timing (interrupt is generated after the transfer) 1: Interrupt generation when CSIGnTX0W or CSIGnTX0H is free for next data. For details, refer to *(1) "CSIGnTIC (communication interrupt)" on page 1265*.
7	CSIGnEDLE	Enables/disables extended data length (EDL) mode. 0: Extended data length mode disabled 1: Extended data length mode enabled For details, refer to *(2) "Data length greater than 16 bits" on page 1261*.
5	CSIGnDCS	Enables/disables data consistency check. 0: Data consistency check disabled 1: Data consistency check enabled For details, refer to *(1) "Data consistency check" on page 1270*.
3	CSIGnLBM	Controls loop-back mode (LBM). 0: Loop-back mode deactivated 1: Loop-back mode activated For details, refer to *21.3.10 "Error detection" on page 1270*. This bit cannot be changed in the slave mode.

Table 21-10 CSIGnCTL1 register contents (2/2)

Bit position	Bit name	Function
2	CSIGnSIT	Selects interrupt delay mode. 0: No delay 1: Half clock delay for all interrupts This bit is only valid in master mode. In slave mode, no delay is generated. For details, refer to *21.3.8 "CSIG interrupts" on page 1265*.
1	CSIGnHSE	Enables/disables handshake mode. 0: Handshake function disabled 1: Handshake function enabled For details, refer to *21.3.9 "Handshake function" on page 1268*.
0	CSIGnSSE	Enables/disables slave select function (SS). 0: Input signal $\overline{\text{CSIGnTSSI}}$ is ignored 1: Input signal $\overline{\text{CSIGnTSSI}}$ is enabled If the slave select function is not used, this bit must be set to 0 (see also *21.3.2 "Master/slave connections" on page 1257*).

Details about CSIGnCTL1.CSIGnSSE:

Table 21-11 Operation of the slave select function during reception

CSIGnCTL0. CSIGnRXE	CSIGnCTL1. CSIGnSSE	$\overline{\text{CSIGnTSSI}}$	Reception operation
0	–	–	Reception disabled
1	0	–	Possible
1	1	0	Possible
1	1	1	Impossible

Table 21-12 Operation of the slave select function during transmission

CSIGnCTL0. CSIGnTXE	CSIGnCTL1. CSIGnSSE	$\overline{\text{CSIGnTSSI}}$	Transmission operation
0	–	–	Transmission disabled
1	0	–	Possible
1	1	0	Possible
1	1	1	Impossible

(3) CSIGnCTL2 - CSIG control register 2

This register selects the operating mode, prescaler, and baud rate.

Access This register can be read/written in 16-bit units.

Address <CSIGn_base_OS> + 0014_H

Initial Value $E000_H$. This register is initialized by any reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSIGnPRS[2:0]			0	CSIGnBRS[11:0]											
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

Caution Changing the contents of this register is only permitted when CSIGnCTL0.CSIGnPWR = 0.

Table 21-13 CSIGnCTL2 register contents

Bit position	Bit name	Function
15 to 13	CSIGnPRS [2:0]	Selects the operating mode and the value of the prescaler.
11 to 0	CSIGnBRS [11:0]	Selects the baud rate. The setting of these bits is valid only in the master mode, and is ignored in the slave mode.

CSIGn PRS2	CSIGn PRS1	CSIGn PRS0	Base clock (PRSOUT) selection
0	0	0	PCLK (master mode)
0	0	1	PCLK / 2 (master mode)
0	1	0	PCLK / 4 (master mode)
0	1	1	PCLK / 8 (master mode)
1	0	0	PCLK / 16 (master mode)
1	0	1	PCLK / 32 (master mode)
1	1	0	PCLK / 64 (master mode)
1	1	1	External clock via CSIGnTSCK (in) (slave mode)

CSIGnBRS[11:0]	Baud rate at CSIGnTSCK
0	BRG is stopped
1	PCLK / (2^m × 1 × 2)
2	PCLK / (2^m × 2 × 2)
3	PCLK / (2^m × 3 × 2)
4	PCLK / (2^m × 4 × 2)
...	...
4095	PCLK / (2^m × 4095 × 2)

Note
m = 0 to 6: Value specified for CSIGnPRS[2:0]

(4) CSIGnSTR0 - CSIG status register 0

This register indicates the status of CSIG.

Access This register can be read in 32-bit units.

Address <CSIGn_base_USER> + 0004_H

Initial Value $0000\ 0010_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	CSIGn TSF	0	0	1	CSIGn DCE	0	CSIGn PE	CSIGn OVE
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21-14 CSIGnSTR0 register contents (1/2)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>7</td><td>CSIGnTSF</td><td>Transfer status flag
0: Idle state
1: Transmission is in progress or being prepared
Setting and clearing of this bit is as follows:
<table>
<tr><th>Master mode</th><th>Set condition</th><th>Clear condition</th></tr>
<tr><td>Transmission mode</td><td rowspan="2">Writing to transmission data register</td><td rowspan="2">Within 0.5 clock cycles from the last CSIGnTSCK edge</td></tr>
<tr><td>Transmission/ reception mode</td></tr>
<tr><td>Reception mode</td><td>Reading from reception data register</td><td></td></tr>
</table>
<table>
<tr><th>Slave mode</th><th>Set condition</th><th>Clear condition</th></tr>
<tr><td>Transmission mode</td><td rowspan="2">Writing to transmission data register</td><td rowspan="2">Within 0.5 clock cycles from the last CSIGnTSCK edge</td></tr>
<tr><td>Transmission/ reception mode</td></tr>
<tr><td>Reception mode</td><td>CSIGnTSCK input</td><td></td></tr>
</table></td></tr>
</table>

Table 21-14 CSIGnSTR0 register contents (2/2)

Bit position	Bit name	Function
3	CSIGnDCE	Data consistency error flag 0: No data consistency error detected 1: Data consistency error detected This bit is cleared by setting CSIGnSTCR0.CSIGnDCEC. This bit can be written to when CSIGnCTL0.CSIGnPWR = 0. This bit is initialized when CSIGnCTL0.CSIGnPWR changes from 0 to 1 or from 1 to 0. If this bit is set due to a data consistency error being detected and cleared by CSIGnSTCR0.CSIGnDCEC at the same time, setting the bit is prioritized.
1	CSIGnPE	Parity error flag 0: No parity error detected 1: Parity error detected This bit is cleared by setting CSIGnSTCR0.CSIGnPEC. This bit can be written to when CSIGnCTL0.CSIGnPWR = 0. This bit is initialized when CSIGnCTL0.CSIGnPWR changes from 0 to 1 or from 1 to 0. If this bit is set due to a parity error being detected and cleared by CSIGnSTCR0.CSIGnPEC at the same time, setting the bit is prioritized.
0	CSIGnOVE	Overrun error flag 0: No overrun error detected 1: Overrun error detected This bit is cleared by setting CSIGnSTCR0.CSIGnOVEC. This bit can be written to when CSIGnCTL0.CSIGnPWR = 0. This bit is initialized when CSIGnCTL0.CSIGnPWR changes from 0 to 1 or from 1 to 0. This bit is fixed to 0 in the dual buffer mode. If this bit is set due to an overrun error being detected and cleared by CSIGnSTCR0.CSIGnOVEC at the same time, setting the bit is prioritized.

(5) CSIGnSTCR0 - CSIG status clear register 0

This register clears the status flags of the CSIGnSTR0 status register.

Access This register can be written in 16-bit units.

When read, the value 0000_H is always returned.

Address <CSIGn_base_USER> + 0008_H

Initial Value 0000_H. This register is initialized by any reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	CSIGnDCEC	0	CSIGnPEC	CSIGnOVEC
R	R	R	R	R	R	R	R	R	R	R	R	W	R	W	W

Table 21-15 CSIGnSTCR0 register contents

Bit position	Bit name	Function
3	CSIGnDCEC	Controls the data consistency error flag clear command. 0: No operation. Read value is always 0. 1: Clear data consistency error flag (CSIGnSTR0.CSIGnDCE)
1	CSIGnPEC	Controls the parity error flag clear command. 0: No operation. Read value is always 0. 1: Clear parity error flag (CSIGnSTR0.CSIGnPE)
0	CSIGnOVEC	Controls the overrun error flag clear command. 0: No operation. Read value is always 0. 1: Clear overrun error flag (CSIGnSTR0.CSIGnOVE)

(6) CSIGnBCTL0 - CSIG reception mode control register 0

This register enables/disables the data transfer in reception mode.

Access This register can be read/written in 8-bit or 1-bit units.

Address <CSIGn_base_USER> + 0080_H

Initial Value 01_H. This register is initialized by any reset.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	CSIGnSCE
R	R	R	R	R	R	R	R/W

Table 21-16 CSIGnBCTL0 register contents

Bit position	Bit name	Function
0	CSIGnSCE	Disables/enables next data reception start by reading CSIGnRX0. 0: Next reception disabled 1: Next reception enabled For details, refer to *(2) "Reception mode" on page 1260* and *21.3.7 "Communication in slave mode" on page 1264*.

Cautions

1. Writing the CSIGnSCE bit must be executed one clock before a CSIGnTIR interrupt is generated.
2. The CSIGnSCE bits must be fixed to 1 when the operating mode is transmission or transmission/reception mode.

(7) CSIGnCFG0 - CSIG configuration register 0

This register configures the communication protocol – data length, parity, transfer direction, clock phase, and data phase.

Access This register can be read/written in 32-bit units.

Address <CSIGn_base_OS> + 1010_H

Initial Value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	CSIGn PS[1:0]		CSIGnDLS[3:0]				0	0	0	0	0	CSIGn DIR	0	CSIGn DAP
R	R	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R/W	R	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Cautions

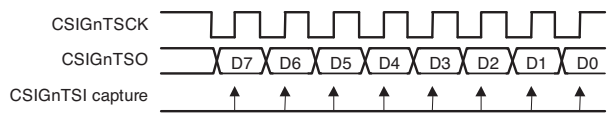
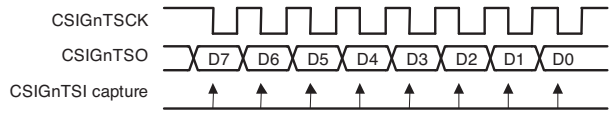
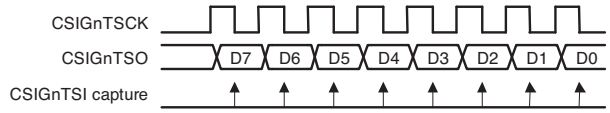
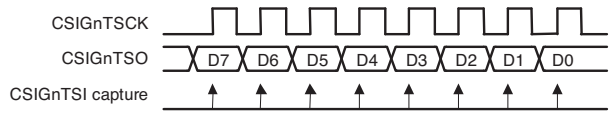
1. Writing is only possible when CSIGnCTL0.CSIGnPWR = 0 (except when writing the same value, which is possible even when CSIGnCTL0.CSIGnPWR = 1).
2. Be sure to clear bits 31, 30, 23 to 19, 17, and 15 to 0.

Table 21-17 CSIGnCFG0 register contents (1/2)

Bit position	Bit name	Function
29 and 28	CSIGnPS [1:0]	Specifies the parity.

CSIGn PS1	CSIGn PS0	Transmission	Reception
0	0	No parity transmitted	Parity reception is not expected.
0	1	Add parity bit fixed at 0	Parity bit reception is expected, but parity judgment is not performed.
1	0	Add odd parity	Odd parity bit reception is expected.
1	1	Add even parity	Even parity bit reception is expected.

Table 21-17 CSIGnCFG0 register contents (2/2)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>27 to 24</td><td>CSIGnDLS [3:0]</td><td>Selects the data length.
<table>
<tr><th>CSIGnDLS[3:0]</th><th>Data length</th></tr>
<tr><td>0000_B</td><td>16 bits</td></tr>
<tr><td>0001_B</td><td>1 bit</td></tr>
<tr><td>0010_B</td><td>2 bits</td></tr>
<tr><td>...</td><td>...</td></tr>
<tr><td>1111_B</td><td>15 bits</td></tr>
</table>
Note
Data length between 1 bit and 6 bits requires that EDL function is used (see also (2) “Data length greater than 16 bits”). In addition, it is forbidden to transmit two consecutive data with a length of less than 7 bits.
The setting of these bits is valid when CSIGnTX0W.CSIGnEDL[3:0] = 0.</td></tr>
<tr><td>18</td><td>CSIGnDIR</td><td>Selects the serial data direction.
0: Data is sent/received with MSB first
1: Data is sent/received with LSB first</td></tr>
<tr><td>16</td><td>CSIGnDAP</td><td>CKR: Clock phase selection bit
DAP: Data phase selection bit
<table>
<tr><th>CSIGn CKR</th><th>CSIGn DAP</th><th>Clock and data phase selection</th></tr>
<tr><td>0</td><td>0</td><td> 
</td></tr>
<tr><td>0</td><td>1</td><td> 
</td></tr>
<tr><td>1</td><td>0</td><td> 
</td></tr>
<tr><td>1</td><td>1</td><td> 
</td></tr>
</table>
For details about the CSIGnCKR bit, see (2) “CSIGnCTL1 - CSIG control register 1” on page 1277.</td></tr>
</table>

(8) CSIGnTX0W - Transmission data register 0 for word access

This register stores the transmission data. It has to be used when the extended data length function is enabled (CSIGnCTL1.SCIGnEDLE = 1).

Access This register can be read/written in 32-bit units.

Address <CSIGn_base_USER> + 0084_H

Initial Value Undefined

Caution Writing to this register is prohibited when CSIGnCTL0.CSIGnTXE and CSIGnCTL0.CSIGnRXE = 0.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	CSIGnEDL	0	0	0	0	0	0	0	0	0	0	0	0	0
R	R	R/W	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSIGnTX[15:0]															
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

Table 21-18 CSIGnTX0W register contents

Bit position	Bit name	Function
29	CSIGnEDL	Specifies whether the associated data requires the extended data length (EDL) option. 0: Normal operation 1: Extended data length activated The associated data is transmitted as of 16 bits. The inter-data delay time and idle time are not inserted after data transmission. When CSIGnCTL1.CSIGnEDLE = 1 and CSIGnTX0W.CSIGnEDL = 1, the same slave must also be selected for the second data. If the slave for the second data is changed, the correct operation is not guaranteed. **Caution** This bit can only be used when CSIGnCTL1.CSIGnEDLE = 1.
15 to 0	CSIGn TX[15:0]	Stores the transmission data.

(9) CSIGnTX0H - Transmission data register 0 for half word access

This register stores the transmission data. It can be used when the Extended Data Length function is disabled (CSIGnCTL1.CSIGnEDLE = 0).

Access This register can be read/written in 16-bit units.

Address <CSIGn_base_USER> + 0088_H

Initial Value 0000_H. This register is initialized by any reset.

Caution Writing to this register is prohibited when CSIGnCTL0.CSIGnTXE and CSIGnCTL0.CSIGnRXE = 0.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSIGnTX[15:0]															
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

Table 21-19 CSIGnTX0H register contents

Bit position	Bit name	Function
15 to 0	CSIGn TX[15:0]	Stores the transmission data.

(10) CSIGnRX0 - Reception data register 0

This register stores the received data.

Access This register can be read in 16-bit units.

Address <CSIGn_base_USER> + $008C_H$

Initial Value 0000_H. This register is initialized by any reset.

Cautions

1. This register can be read when CSIGnCTL0.CSIGnPWR = 1, and can be written to when CSIGnCTL0.CSIGnPWR = 0.
2. This register is initialized when the value of CSIGnCTL0.CSIGnPWR changes.
3. Reading this register is prohibited when CSIGnCTL0.CSIGnTXE and CSIGnCTL0.CSIGnRXE = 0.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSIGnRX[15:0]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 21-20 CSIGnRX0 register contents

Bit position	Bit name	Function
15 to 0	CSIGn RX[15:0]	Stores the reception data.

21.5 Operating Procedure Example

(1) For transmission/reception in the master mode

The following conditions are assumed for the procedure shown here:

- Transmission data length: 8 bits (CSIGnCFG0.CSIGnDLS[3:0] = 1000_B)
- Transmission direction: MSB first (CSIGnCFG0.CSIGnDIR = 0)
- Normal clock phase and data phase (CSIGnCTL1.CSIGnCKR = 0, CSIGnCFG0.CSIGnDAP = 0)
- No delay for any interrupt (CSIGnCTL1.CSIGnSIT = 0)
- A CSIGnTIC interrupt is generated when transferring starts. (CSIGnCTL1.CSIGnCLIT = 1)

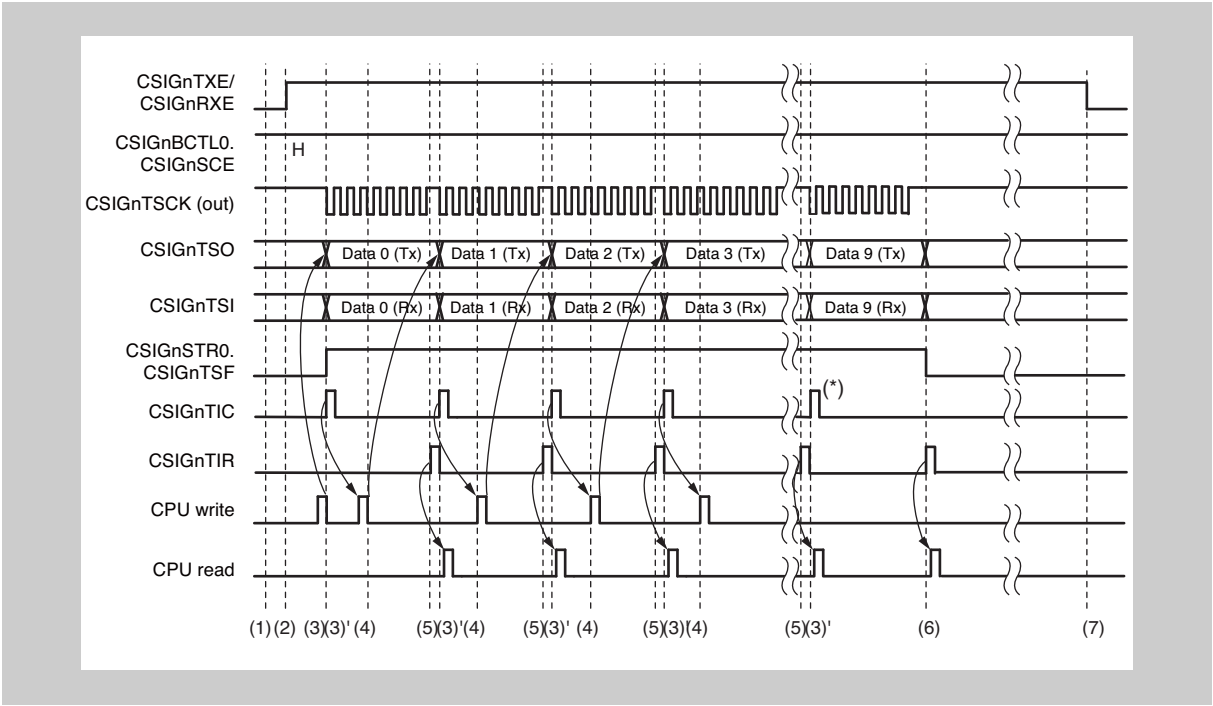


Figure 21-26 For transmission/reception in the master mode

Procedure:

1. Set up the following registers before setting CSIGnCTL0.CSIGnPWR to 1: CSIGnCTL1, CSIGnCTL2, CISGnBCTL0, CSIGnCFG0
2. CSIGnCTL0.CSIGnPWR = 1 (clock enabled)
   CSIGnCTL0.CSIGnTXE = 1 (transmission enabled)
   CSIGnCTL0.CSIGnRXE = 1 (reception enabled)
   Bit 0 of CSIGnCTL0 = 1
3. Write the first data to the transmission data register CSIGnTX0W. Transmission automatically starts.

3'. When CSIGnCTL1.CSIGnSLIT is set to 1, CSIGnTIC is generated at the start edge of CSIGnTSCK. CSIGnTIC indicates that the second data can be written to CSIGnTX0W.

4. Write the second data to CSIGnTX0W. By writing the second data immediately after writing the first data, the unnecessary inter-data delay can be avoided.

5. Each time data is received, a CSIGnTIR interrupt is generated.
   – CSIGnTIR indicates that the reception data register CSIGnRX0 must be read.
6. If the CSIGnTIC interrupt indicated by “*” in the figure is the last one, it is not necessary to write to the transmission data register CSIGnTX0W based on the corresponding CSIGnTIC interrupt.
7. Finally, clear CSIGnCTL0.CSIGnTXE and CSIGnCTL0.CSIGnRXE to disable transmission/reception operations. In addition, clear CSIGnCTL0.CSIGnPWR to reduce the power consumption of CSIG.

(2) For reception in the master mode

The following conditions are assumed for the procedure shown here:

- Transmission data length: 8 bits (CSIGnCFG0.CSIGnDLS[3:0] = 1000_B)
- Transmission direction: MSB first (CSIGnCFG0.CSIGnDIR = 0)
- Normal clock phase and data phase (CSIGnCTL1.CSIGnCKR = 0, CSIGnCFG0.CSIGnDAP = 0)
- No delay for any interrupt (CSIGnCTL1.CSIGnSIT = 0)
- A CSIGnTIC interrupt is generated when transferring starts. (CSIGnCTL1.CSIGnCLIT = 1)

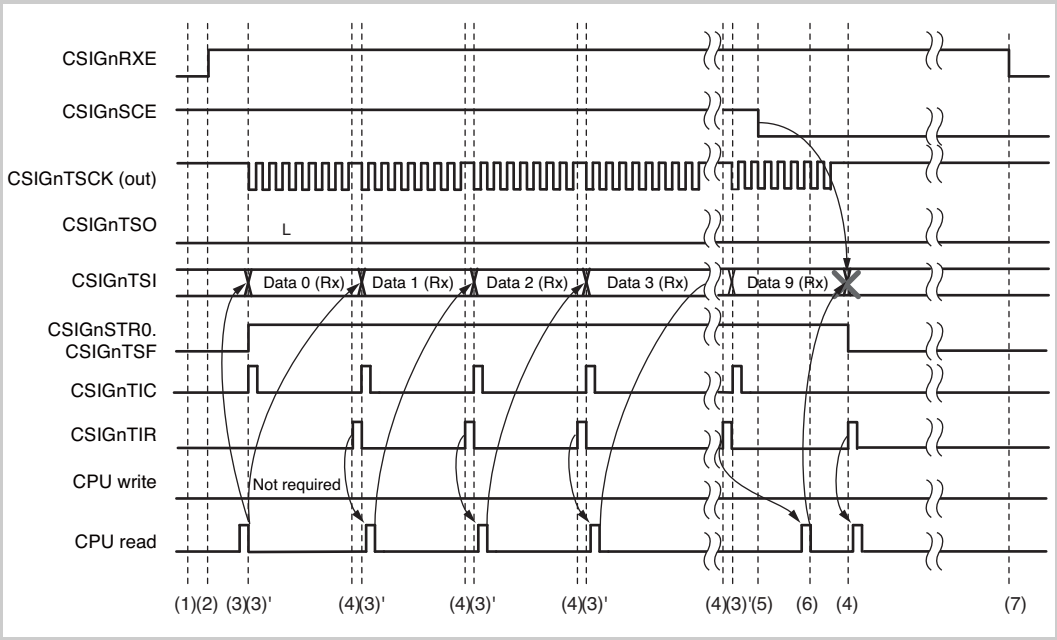


Figure 21-27 For reception in the master mode

Procedure:

1. Set up the following registers before setting CSIGnCTL0.CSIGnPWR to 1: CSIGnCTL1, CSIGnCTL2, CISGnBCTL0, CSIGnCFG0
2. CSIGnCTL0.CSIGnPWR = 1 (clock enabled)
   CSIGnCTL0.CSIGnTXE = 0 (transmission disabled)
   CSIGnCTL0.CSIGnRXE = 1 (reception enabled)
   Bit 0 of CSIGnCTL0 = 1
3. Read the dummy data from the reception data register CSIGnRX0. Reception automatically starts.

3'. When CSIGnCTL1.CSIGnSLIT is set to 1, CSIGnTIC is generated at the start edge of CSIGnTSCK.

4. Each time data is received, a CSIGnTIR interrupt is generated.
   – CSIGnTIR indicates that the reception data register CSIGnRX0 must be read.
5. To finish the consecutive reception with the data currently received, clear the CSIGnBCTL0.CSIGnSCE bit.

6. Reception does not start even if reception data is read.
7. Finally, clear CSIGnCTL0.CSIGnRXE to disable reception operations. In addition, clear CSIGnCTL0.CSIGnPWR to reduce the power consumption of CSIG.

(3) For transmission/reception in the slave mode

The following conditions are assumed for the procedure shown here:

- Transmission data length: 8 bits (CSIGnCFG0.CSIGnDLS[3:0] = 1000_B)
- Transmission direction: MSB first (CSIGnCFG0.CSIGnDIR = 0)
- Normal clock phase and data phase (CSIGnCTL1.CSIGnCKR = 0, CSIGnCFG0.CSIGnDAP = 0)
- No delay for any interrupt (CSIGnCTL1.CSIGnSIT = 0)
- A CSIGnTIC interrupt is generated when transferring starts. (CSIGnCTL1.CSIGnCLIT = 1)

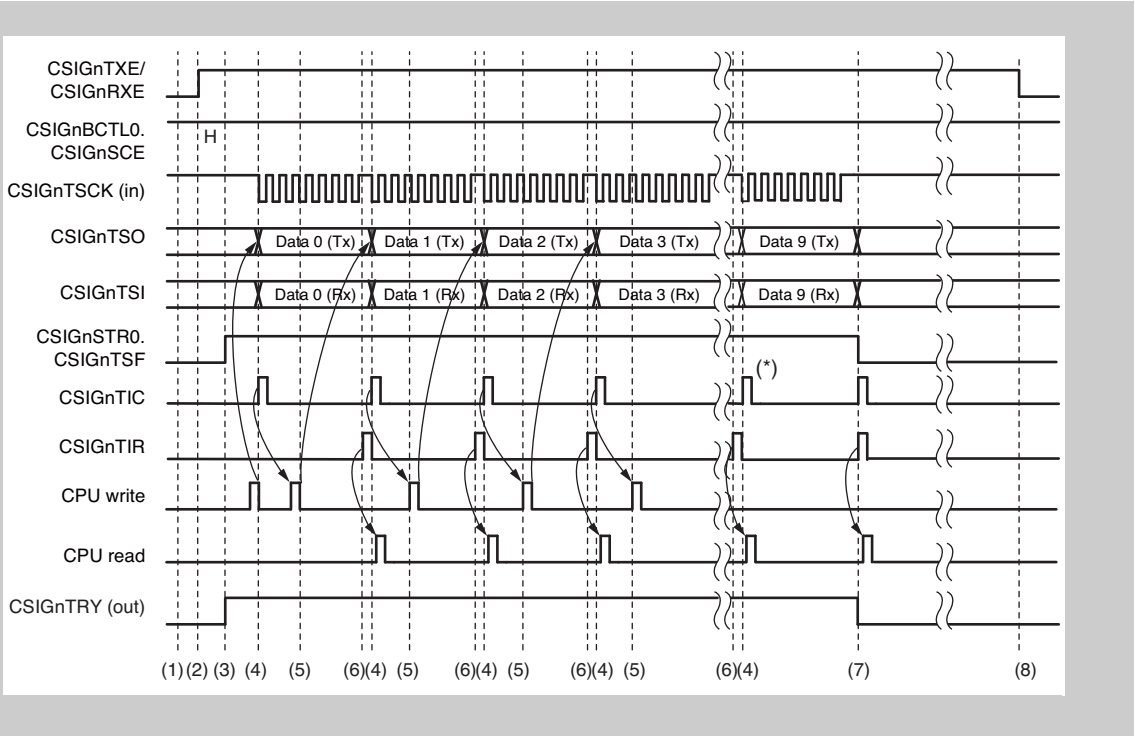


Figure 21-28 For transmission/reception in the slave mode

Procedure:

1. Set up the following registers before setting CSIGnCTL0.CSIGnPWR to 1: CSIGnCTL1, CSIGnCTL2, CISGnBCTL0, CSIGnCFG0
2. CSIGnCTL0.CSIGnPWR = 1 (clock enabled)
   CSIGnCTL0.CSIGnTXE = 1 (transmission enabled)
   CSIGnCTL0.CSIGnRXE = 1 (reception enabled)
   Bit 0 of CSIGnCTL0 = 1
3. Write the first data to the transmission data register CSIGnTX0W.
4. When a serial clock is supplied from the master, transmission automatically starts.
   When CSIGnCTL1.CSIGnSLIT is set to 1, CSIGnTIC is generated at the start edge of CSIGnTSCK. CSIGnTIC indicates that the second data can be written to CSIGnTX0W.
5. Write the second data to CSIGnTX0W. By writing the second data immediately after writing the first data, the unnecessary inter-data delay can be avoided.

6. Each time data is received, a CSIGnTIR interrupt is generated.
   – CSIGnTIR indicates that the reception data register CSIGnRX0 must be read.
7. If the CSIGnTIC interrupt indicated by “*” in the figure is the last one, it is not necessary to write to the transmission data register CSIGnTX0W based on the corresponding CSIGnTIC interrupt.
8. Finally, clear CSIGnCTL0.CSIGnTXE and CSIGnCTL0.CSIGnRXE to disable transmission/reception operations. In addition, clear CSIGnCTL0.CSIGnPWR to reduce the power consumption of CSIG.

(4) For reception in the slave mode

The following conditions are assumed for the procedure shown here:

- Transmission data length: 8 bits (CSIGnCFG0.CSIGnDLS[3:0] = 1000_B)
- Transmission direction: MSB first (CSIGnCFG0.CSIGnDIR = 0)
- Normal clock phase and data phase (CSIGnCTL1.CSIGnCKR = 0, CSIGnCFG0.CSIGnDAP = 0)
- No delay for any interrupt (CSIGnCTL1.CSIGnSIT = 0)
- A CSIGnTIC interrupt is generated when transferring starts. (CSIGnCTL1.CSIGnCLIT = 1)

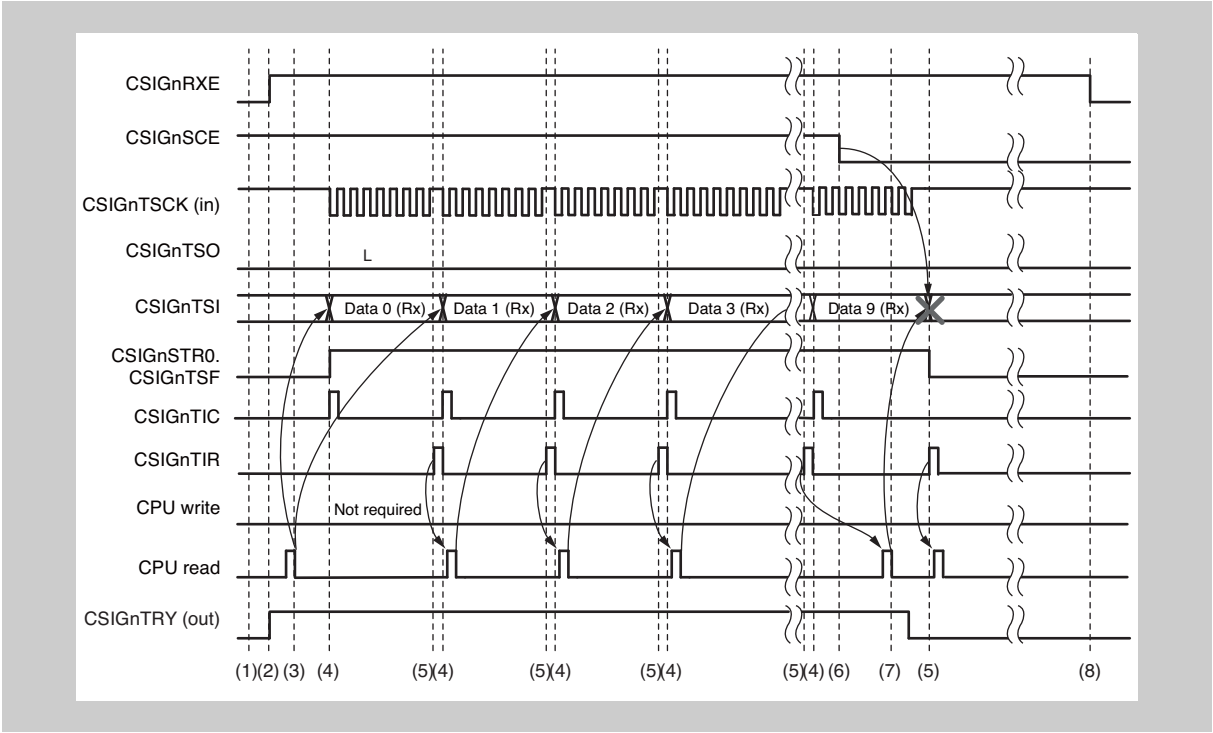


Figure 21-29 For reception in the slave mode

Procedure:

1. Set up the following registers before setting CSIGnCTL0.CSIGnPWR to 1: CSIGnCTL1, CSIGnCTL2, CISGnBCTL0, CSIGnCFG0
2. CSIGnCTL0.CSIGnPWR = 1 (clock enabled)
   CSIGnCTL0.CSIGnTXE = 0 (transmission disabled)
   CSIGnCTL0.CSIGnRXE = 1 (reception enabled)
   Bit 0 of CSIGnCTL0 = 1
3. Read the dummy data from the reception data register CSIGnRX0. Reception automatically starts.
4. When a serial clock is supplied from the master, transmission automatically starts.
   When CSIGnCTL1.CSIGnSLIT is set to 1, CSIGnTIC is generated at the start edge of CSIGnTSCK.
5. Each time data is received, a CSIGnTIR interrupt is generated.
   – CSIGnTIR indicates that the reception data register CSIGnRX0 must be read.

6. To finish the consecutive reception with the data currently received, clear the CSIGnBCTL0.CSIGnSCE bit.
7. Reception does not start even if reception data is read.
8. Finally, clear CSIGnCTL0.CSIGnRXE to disable reception operations. In addition, clear CSIGnCTL0.CSIGnPWR to reduce the power consumption of CSIG.

Chapter 22 Clocked Serial Interface H (CSIH)

This chapter contains a generic description of clocked serial interface H (CSIH).

The first section describes all V850E2/MN4 specific properties, such as instances, register base addresses, and input/output signal names.
The subsequent sections describe the features that apply to all implementations.

22.1 V850E2/MN4 CSIH Features

Instances

This microcontroller has following number of instances of CSIH.

Table 22-1 Instances of CSIH

Clocked serial interface H	
Instance	4
Name	CSIH0 to CSIH3

Instances index n

Throughout this chapter, the individual instances of CSIH is identified by the index "n" (n = 0 to 3), for example, CSIHnCTL0 for CSIHn control register 0.

Chip select index x

CSIH has up to 8 chip select signals. Throughout this chapter, the individual chip select signals are identified by the index “x” (x = 0 to 7), thus a certain chip select signal is denoted as CSx.
The number of chip selects for each instance of CSIH is given in the following table:

Table 22-2 Number of chip select signals of CSIH

CSIHn instance	Chip select signal numbers
CSIH0	CS0 to CS7
CSIH1	CS0 to CS7
CSIH2	CS0 to CS7
CSIH3	CS0 to CS7

Register addresses

All CSIHn register addresses are given as addresses offset from the individual base address <CSIHn_base_USER> or <CSIHn_base_OS>.
The base addresses <CSIHn_base_USER> and <CSIHn_base_OS> of each CSIHn are listed in the following table:

Table 22-3 Register base addresses

CSIHn instance	Base address	<CSIHn_base> address
CSIH0	<CSIHn_base_USER>	FFFF $E000_H$
	<CSIHn_base_OS>	FF6C 0000_H
CSIH1	<CSIHn_base_USER>	FFFF $E100_H$
	<CSIHn_base_OS>	FF6D 0000_H
CSIH2	<CSIHn_base_USER>	FFFF $E200_H$
	<CSIHn_base_OS>	FF6E 0000_H
CSIH3	<CSIHn_base_USER>	FFFF $E300_H$
	<CSIHn_base_OS>	FF6F 0000_H

Clock supply

All CSIH provide one clock input:

Table 22-4 CSIHn clock supply

CSIHn instance	CSIHn clock	Connected to
CSIH0 to CSIH3	PCLK	f_{PCLK}

Maximum transfer speed (baud rate)

For CSIH, communication is possible at the maximum transfer speeds (baud rates) listed in the following table.

Table 22-5 Maximum CSIHn transfer speeds (baud rates)

Mode	Maximum transfer speed (baud rate)
Master mode	5.625 Mbps (Max.)
Slave mode	5.625 Mbps (Max.)

Interrupts and DMA/ DTS CSIH can generate the following interrupt and DMA/DTS requests:

Table 22-6 CSIHn interrupt and DMA/DTS requests

CSIHn signals	Function	Connected to
CSIH0		
CSIH0TIC	Communication interrupt	• Interrupt controller 145 (INTCSIH0IC) • DTS controller trigger 109
CSIH0TIR	Reception interrupt	• Interrupt controller 144 (INTCSIH0IR) • DTS controller trigger 108
CSIH0TIRE	Reception error interrupt	• Interrupt controller 143 (INTCSIH0IRE)
CSIH0TIJC	Job completion interrupt	• Interrupt controller 146 (INTCSIH0IJC) • DTS controller trigger 110
CSIH1		
CSIH1TIC	Communication interrupt	• Interrupt controller 149 (INTCSIH1IC) • DTS controller trigger 112
CSIH1TIR	Reception interrupt	• Interrupt controller 148 (INTCSIH1IR) • DTS controller trigger 111
CSIH1TIRE	Reception error interrupt	• Interrupt controller 147 (INTCSIH1IRE)
CSIH1TIJC	Job completion interrupt	• Interrupt controller 150 (INTCSIH1IJC) • DTS controller trigger 113
CSIH2		
CSIH2TIC	Communication interrupt	• Interrupt controller 153 (INTCSIH2IC) • DTS controller trigger 115
CSIH2TIR	Reception interrupt	• Interrupt controller 152 (INTCSIH2IR) • DTS controller trigger 114
CSIH2TIRE	Reception error interrupt	• Interrupt controller 151 (INTCSIH2IRE)
CSIH2TIJC	Job completion interrupt	• Interrupt controller 154 (INTCSIH2IJC) • DTS controller trigger 116
CSIH3		
CSIH3TIC	Communication interrupt	• Interrupt controller 157 (INTCSIH3IC) • DTS controller trigger 118
CSIH3TIR	Reception interrupt	• Interrupt controller 156 (INTCSIH3IR) • DTS controller trigger 117
CSIH3TIRE	Reception error interrupt	• Interrupt controller 155 (INTCSIH3IRE)
CSIH3TIJC	Job completion interrupt	• Interrupt controller 158 (INTCSIH3IJC) • DTS controller trigger 119

I/O signals The I/O signals of CSIH are listed in the following table.

Table 22-7 CSIHn I/O signals

CSIHn signal	Function	Connected to
CSIH0		
CSIH0TSCK	Serial clock signal	Port SCK0F
CSIH0TSI	Serial data input signal	Port SI0F
CSIH0TSO	Serial data output signal	Port SO0F
$\overline{\text{CSIH0TSSI}}$	Slave select input signal	Port CSI0F_SSI
CSIH0TRYO	Ready/busy output signal	Port CSI0F_RYO
CSIH0TRYI	Ready/busy input signal	Port CSI0F_RYI
CSIH0TCSSn[7:0]	Chip select signals	Port CSI0F_CS[7:0]
CSIH1		
CSIH1TSCK	Serial clock signal	Port SCK1F
CSIH1TSI	Serial data input signal	Port SI1F
CSIH1TSO	Serial data output signal	Port SO1F
$\overline{\text{CSIH1TSSI}}$	Slave select input signal	Port CSI1F_SSI
CSIH1TRYO	Ready/busy output signal	Port CSI1F_RYO
CSIH1TRYI	Ready/busy input signal	Port CSI1F_RYI
CSIH1TCSSn[7:0]	Chip select signals	Port CSI1F_CS[7:0]
CSIH2		
CSIH2TSCK	Serial clock signal	Port SCK2F
CSIH2TSI	Serial data input signal	Port SI2F
CSIH2TSO	Serial data output signal	Port SO2F
$\overline{\text{CSIH2TSSI}}$	Slave select input signal	Port CSI2F_SSI
CSIH2TRYO	Ready/busy output signal	Port CSI2F_RYO
CSIH2TRYI	Ready/busy input signal	Port CSI2F_RYI
CSIH2TCSSn[7:0]	Chip select signals	Port CSI2F_CS[7:0]
CSIH3		
CSIH3TSCK	Serial clock signal	Port SCK3F
CSIH3TSI	Serial data input signal	Port SI3F
CSIH3TSO	Serial data output signal	Port SO3F
$\overline{\text{CSIH3TSSI}}$	Slave select input signal	Port CSI3F_SSI
CSIH3TRYO	Ready/busy output signal	Port CSI3F_RYO
CSIH3TRYI	Ready/busy input signal	Port CSI3F_RYI
CSIH3TCSSn[7:0]	Chip select signals	Port CSI3F_CS[7:0]

22.2 Functional Overview

Features summary

- Three-wire serial synchronous data transfer
- Master mode and slave mode selectable
- Multiple slaves configuration plus RCB (Recessive Configuration for Broadcasting) thanks to eight configurable chip select output signals
- Slave select input signal ($\overline{\text{CSIHnTSSI}}$)
- Built-in baud rate generator
- Baud rate adjustable; in slave mode determined by input clock
- Maximum transmission speed:
  - in master mode: PCLK/4
  - in slave mode: PCLK/6

Caution There might be restrictions on the maximum baud rate that can actually be used depending on the product. Specify the baud rate so as not to exceed the maximum rate for each product.

- Phase of clock and data selectable
- Data transfer with MSB or LSB first selectable
- Transfer data length selectable from 7 to 16 bits in 1-bit units
- Extended data length (EDL) function for transferring data with more than 16 bits
- Three selectable transfer modes:
  - transmit-only mode
  - receive-only mode
  - transmit/receive mode
- Built-in handshake function
- Error detection (data consistency check, parity, timeout, overflow, overrun)
- Full support of job concept
- 128 words I/O buffer memory
- Memory mode selectable (FIFO, dual buffer, Tx-only buffer, direct access)
- Four different interrupt request signals (CSIHnTIC, CSIHnTIR, CSIHnTIRE, CSIHnTIJC)
- Loop back mode (LBM) function for self test

The block diagram shows the main components of the CSIH.

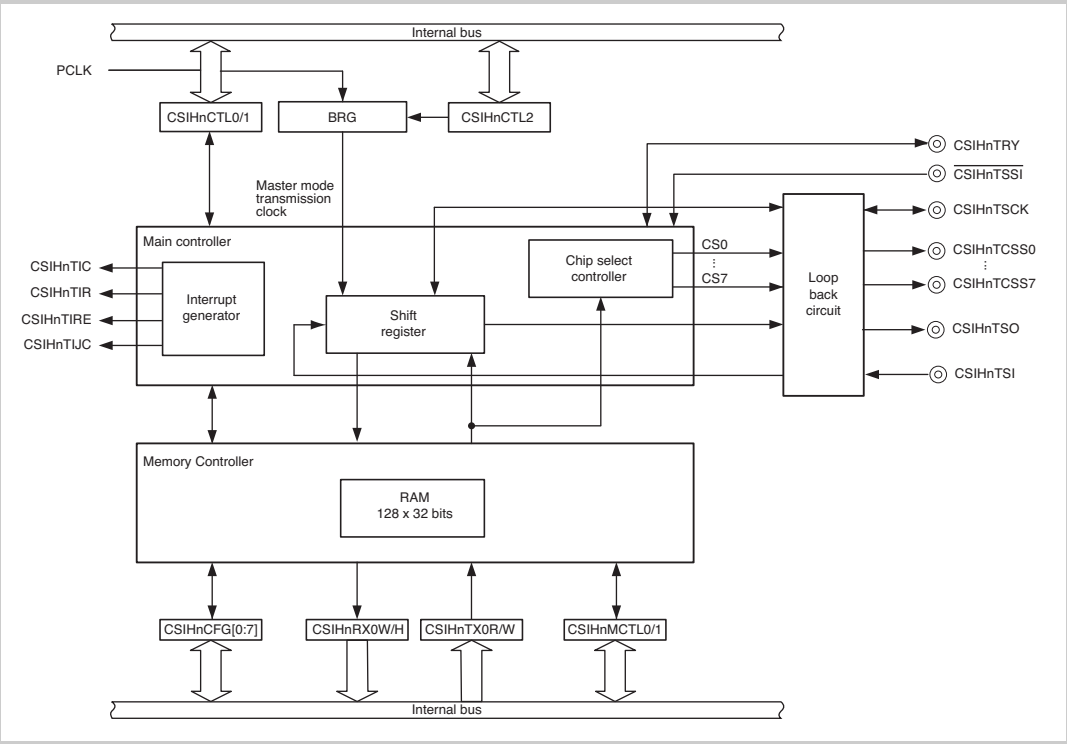


Figure 22-1 CSIH block diagram

In the master mode, the serial clock CSIHnTSCK is generated by the internal baud rate generator (BRG). In the slave mode, the serial clock is supplied from an external source.

The built-in memory can be configured as FIFO, dual buffer (separate transmit and receive buffers), or transmit-only buffer. It can also be bypassed for data transmission and reception without buffering.

The loop back circuit disconnects the CSIH completely from the ports and supports internal self test.

Note This chapter describes the following modes:

- The “operating mode” separates between master and slave mode. In this context, only a master can control and communicate with several slaves (for details, see *22.3.1 “Operating modes (master/slave)” on page 1304*).
- The “job mode” is related to the Autosar job concept (for details, see *22.3.4 “Chip select timing details” on page 1310*).
- The “memory mode” takes the various configurations of the associated buffer memory into account (for details, see *22.3.7 “CSIH buffer memory” on page 1315*).
- The “data transfer mode” specifies the kind of the communication – transmit-only, receive only, or both (for details, see *22.3.8 “Data transfer modes” on page 1317*).

22.3 Functional Description

The Clocked Serial Interface H uses three signals for communication:

- Serial clock CSIHnTSCK (output for the master mode or input for the slave mode)
- Data output signal CSIHnTSO
- Data input signal CSIHnTSI

Additional signals are available for external control and monitoring.

- $\overline{\text{CSIHnTSSI}}$: Slave select input signal
- CSIHnTRY: Handshake signal (input for the master mode or output for the slave mode)
- CSIHnTCSS[7:0]: Chip select signal

Data transmission is bit-wise and serial and synchronous to the serial clock.

The most important registers for setting up the CSIH are:

Register	Function
CSIHnCTL0	Enables or disables the operation clock (PCLK) and enables or disables data transmission and reception. Defines end-of-job behavior and enables/disables buffering (bypass of the buffer).
CSIHnCTL1	Controls options like interrupt timing, extended data length, job feature, data consistency check, loop-back mode, handshake, etc.
CSIHnCTL2	Selects master/slave mode and – effective in master mode – the baud rate of the Internal baud rate generator (BRG)
CSIHnMCTL0	Selects memory mode and specifies timeout
CSIHnMCTL1	Controls the memory in FIFO mode
CSIHnMCTL2	Controls the memory in the dual buffer mode or transmit-only buffer mode
CSIHnCFGx	Registers to configure the communication protocol for each chip select signal

22.3.1 Operating modes (master/slave)

Master/slave selection is performed by using the CSIHnCTL2.CSIHnPRS[2:0] bits, and, when the master is selected, the source clock of the transmission clock must also be selected.

(1) Master mode

In the master mode, the serial clock is generated by the internal baud rate generator (BRG) and provided to the slave(s) by signal CSIHnTSCK.

Master mode is enabled by setting CSIHnCTL2.CSIHnPRS[2:0] to anything but 111_B. In the master mode, the BRG frequency can be specified by specifying values for the CSIHnCTL2.CSIHnPRS[2:0] and CSIHnCTL2.CSIHnBRS[11:0] bits in combination.

Chip select signals

In master mode, one or several chip select signals can be used. If several slaves are connected to the master, the chip select signals can be used to address one or several of the slaves. Only a selected slave is then enabled for communication.

The communication protocol as well as additional parameters are stored separately for each chip select signal. This makes it possible to adapt the data transfer individually to the requirements of each slave. For details, see *22.3.3 “Chip selection (CS) features” on page 1308*.

Clock defaults

The default level of CSIHnTSCK depends on the clock phase selection bit: It is high when CSIHnCFGx.CSIHnCKPx = 0, and is low when CSIHnCFGx.CSIHnCKPx = 1.

The example below shows the communication in master mode for 8 data bits, CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0, and MSB first:

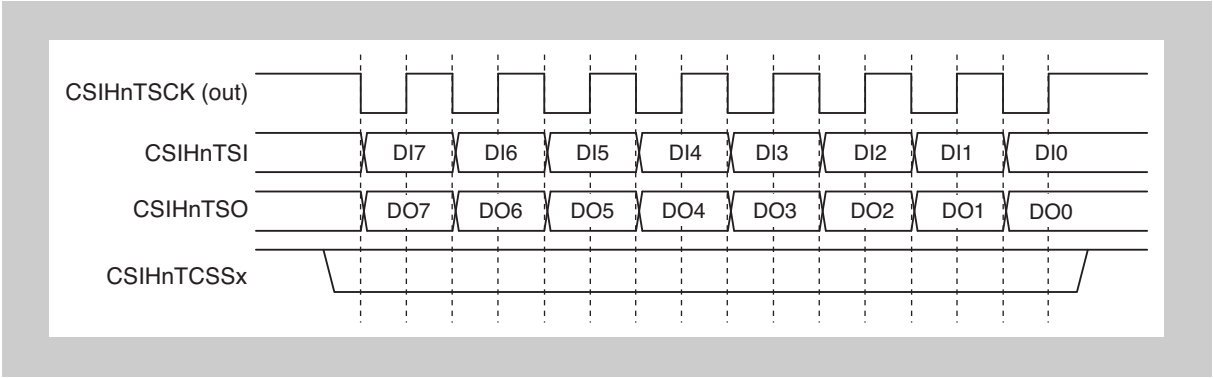


Figure 22-2 Transmit/receive in master mode

(2) Slave mode

In the slave mode, another device is the communication master. The serial clock is supplied through the CSIHnTSCK signal. When the serial clock signal is detected, a transmission or reception operation immediately starts.

Slave mode is selected by setting CSIHnCTL2.CSIHnPRS[2:0] to 111_B.

In slave mode, the transmission protocol setting of the CSIHnCFG0 register are relevant (The CSIHnCFG1 to CSIHnCFG7 register settings become invalid.):

- CSIHnPS0: Parity usage
- CSIHnDLS0: Data length selection
- CSIHnDIR0: Data direction
- CSIHnCKP0, CSIHnDAP0: Clock phase and data phase

Note When using the slave mode, the baud rate generator (BRG) can be disabled by clearing the CSIHnCTL2.CSIHnBRS[11:0] bits, reducing power consumption. However, when using the timeout error function, the BRG must be set to a value other than 0.

The example below shows the communication in the save mode for eight data bits when CSIHnCTL1.CSIHnCKR = 0, CSIHnCFGx.CSIHnDAPx = 0, and the MSB is first.

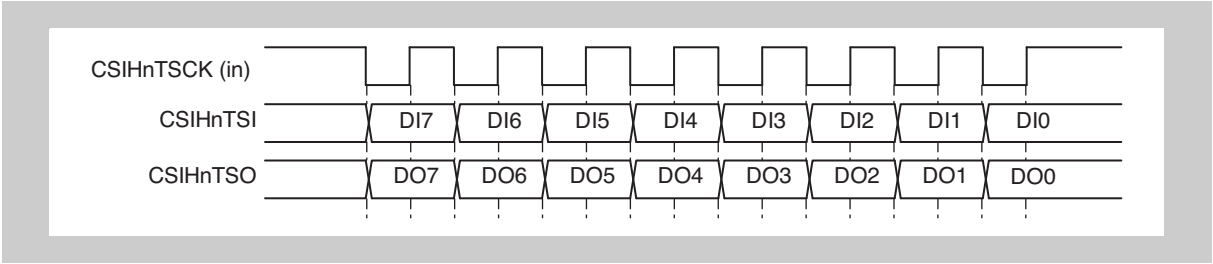


Figure 22-3 Transmission/reception in the slave mode

22.3.2 Master/slave connections

(1) One master and one slave

The following figure illustrates the connections between one master and one slave.

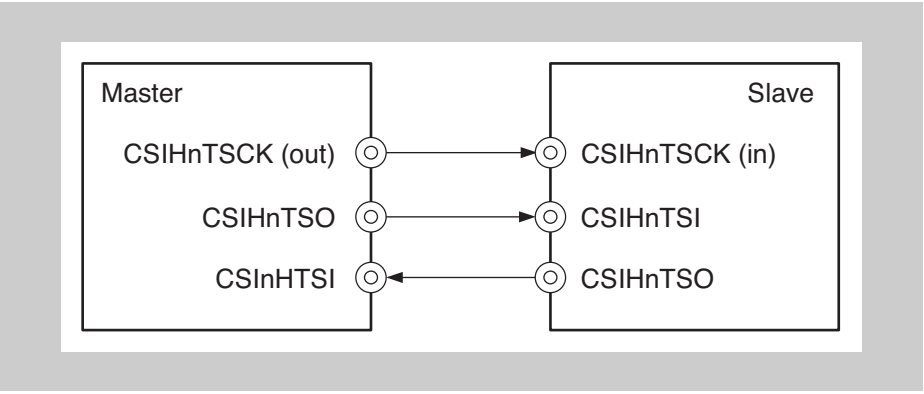


Figure 22-4 Direct master/slave connection

(2) One master and multiple slaves

The following figure illustrates the connections between one master and multiple slaves. In this example, a configuration in which the master supplies one chip select (CS) signal to each slave is possible. This signal is connected to the slave select input $\overline{\text{CSIHnTSSI}}$ of the slave.

The $\overline{\text{CSIHnTSSI}}$ signal recognition function can be enabled/disabled by bit CSIHnCTL1.CSIHnSSE.

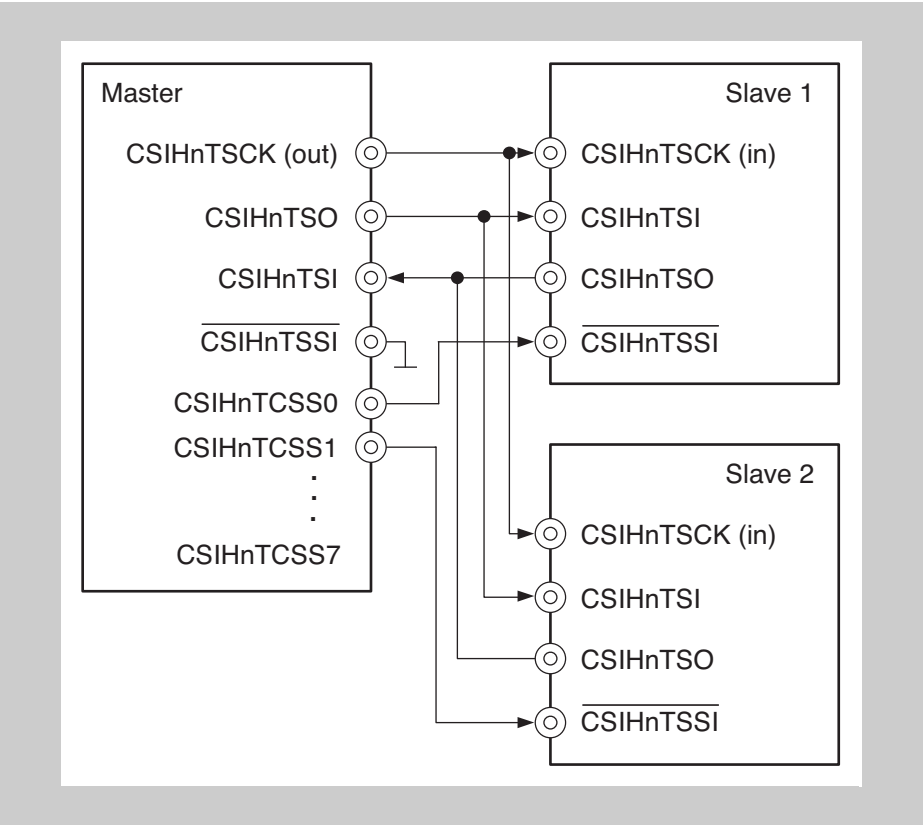


Figure 22-5 Master to multiple slaves connection

The default chip select level is active low. In other words, when the slave select input signal ($\overline{\text{CSIHnTSSI}}$) of a slave is at the low level, that slave is selected as a CSIH slave (and enabled). However, to use a chip select signal (CS) for another device, programming that sets the chip select signal output level to active high is possible.

If a slave is not selected, it will neither receive nor transmit data. In addition, its output CSIHnTSO is set to input mode in order to avoid interference with the output of another slave that was selected.

(3) CSIHnTSO output control

The CSIH can output CSIHnTSO when all of the following conditions are satisfied:

- The CSIH is enabled (CSIHnCTL0.CSIHnPWR = 1).
- The CSIH is operated in transmit-only or transmit/receive mode (CSIHnCTL0.CSIHnTXE = 1).
- The CSIH is operated with slave select enabled (CSIHnCTL1.CSIHnSSE = 1).
- The slave mode selection signal $\overline{\text{CSIHnTSSI}}$ is inactive, i.e. on high level.

By this signal congestions on the external CSIHnTSO signal line are avoided.

22.3.3 Chip selection (CS) features

The chip select signals CSIHnTCSSx can be used by the master to select one or several slaves for communication.

(1) Configuration registers

The parameters for each chip select signal CSIHnTCSSx are defined in the corresponding configuration register CSIHnCFGx. The parameters include the communication protocol and additional CS parameters.

The communication protocol specifies:

- Data length: The number of bits to be sent or received.
- Transfer direction: MSB or LSB first.
- Parity usage: Odd, even, 0 parity, or none.
- Clock phase and data phase.

Additional parameters for each chip select and only available in master mode are:

- Prescaler selection of the baudrate generator separately for each chip select
- Chip select priority: Separates between “dominant” and “recessive” chip select signals. The priority applies if two or more chip selects with different configurations are simultaneously activated for message broadcasting. In this case, the configuration that is set as dominant is used.

  The principle is also called “Recessive Configuration for Broadcasting” (RCB).

Caution It is forbidden to specify several chip select signals as dominant with different configurations unless all dominant chip selects have the same configuration.

- Chip select timing:
  - Setup time T_{setup}: The time from setting the CS signal active to starting data output.
  - Inter-data time T_{inter}: The time between data while the same CS signal is active.
  - Hold time T_{hold}: Hold time of CS active level before changing the CS.
  - Idle time T_{idle}: Inactive time after terminating a CS signal or after every data transfer to the same CSx.

The figure below shows the timing of the chip select (CSx) signal setup time, inter-data time, hold time, and idle time. No matter which CSIHnCFGx.CSIHnIDLx bit is set (to 1), idle time is added to all CS segments.

Figure 22-6 “Chip select timings” shows an example in which the default active low setting is specified for the CS1 and CS2 signals (CSIHnCTL1.CSIHnCSL1 = 0, CSIHnCTL1.CSIHnCSL2 = 0). The active level can be separately specified for each CS.

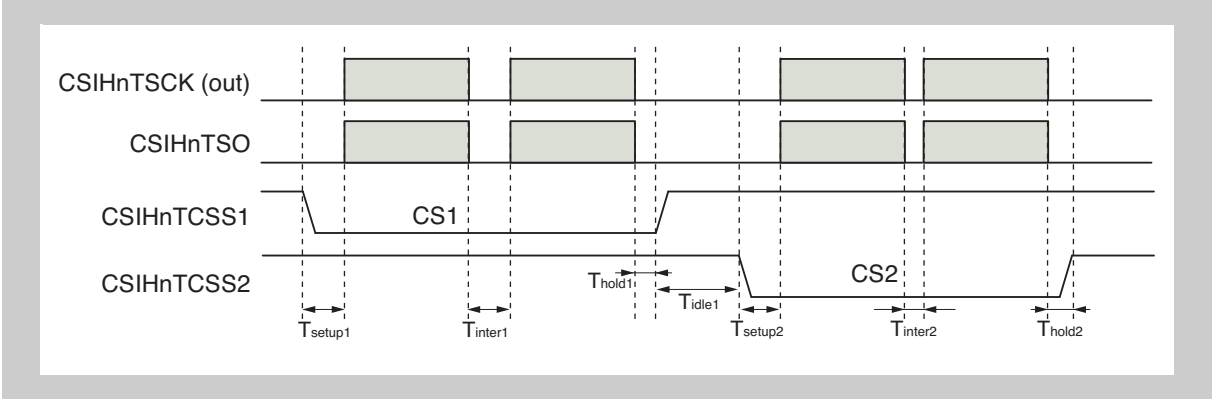


Figure 22-6 Chip select timings

Note that each CS can have a different value for setup, inter-data gap, hold and idle periods

A particular chip select signal is activated by setting the appropriate bit in the transmission data register CSIHnTX0W.CSIHnCS[7:0].

The reception data register indicates in CSIHnRX0W.CSIHnCS[7:0] the chip select signal associated with the received data.

(2) CS example

The following figure shows an example of two consecutive transmissions.

The first communication uses CS0 to address one single slave. The second (for which communication is performed using the dominant-side communication settings) enables CS0 and CS1 to broadcast a message to two slaves. The priority of CS0 is set to “recessive: low priority”, the priority of CS1 to “dominant: high priority”.

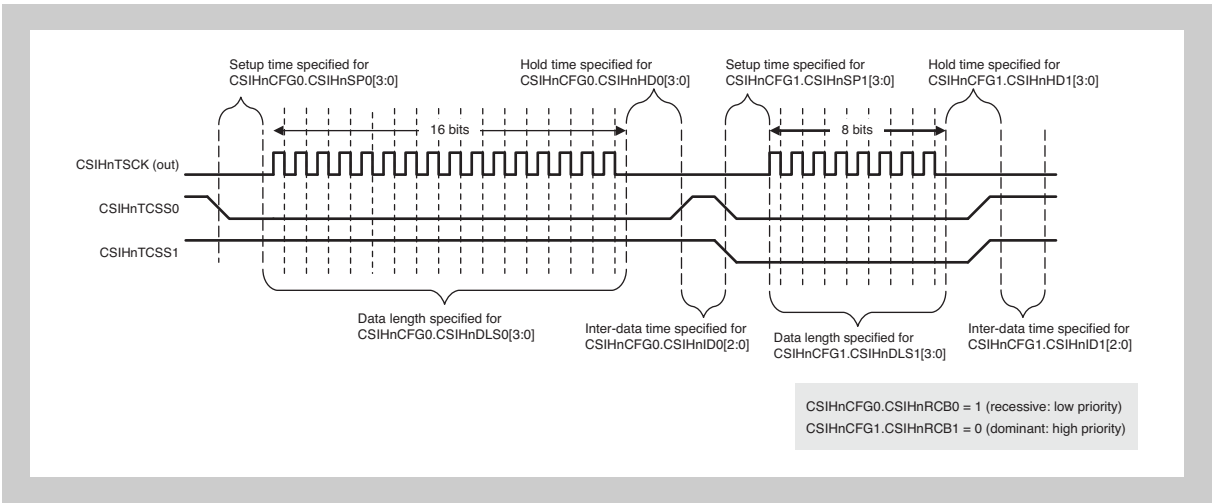


Figure 22-7 Chip select and RCB example

22.3.4 Chip select timing details

(1) Changing the clock phase

The serial clock level is specified for each chip select according to CSIHnCFGx.CSIHnCKPx. The chip select or serial clock level is switched during the idle time. The minimum idle time is 1/2 of a serial clock (CSIHnTSCK) cycle (0.5 SCK).

If the idle time is set to 0.5 transmission clock cycles (in CSIHnCFGx.CSIHnIDx[2:0]) and two consecutive data are sent with different CSIHnCFGx.CSIHnCKPx configuration, the idle time is automatically extended to one cycle of CSIHnTSCK.

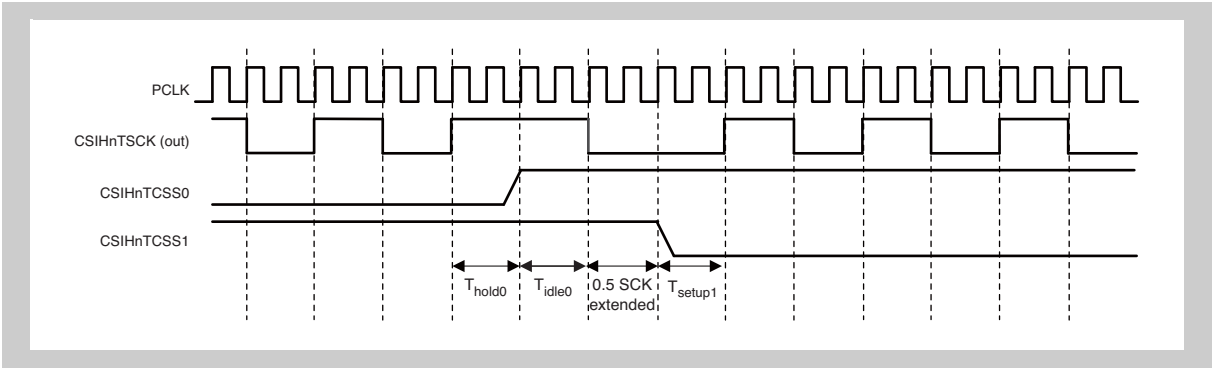


Figure 22-8 Clock phase timing (in the case of PCLK/4, $T_{hold0} = T_{setup1}$ = 0.5 SCK, T_{idle0} = 0.5 SCK, CKP0 = 0 (CSIHnTCSS0) → CKP1 = 1 (CSIHnTCSS1))

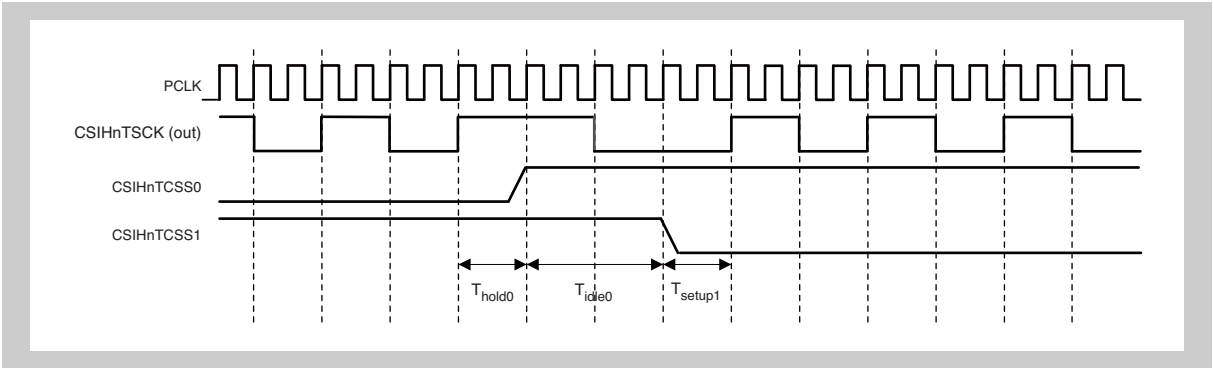


Figure 22-9 Clock phase timing (in the case of PCLK/4, $T_{hold0} = T_{setup1}$ = 0.5 SCK, T_{idle0} = 1.0 SCK, CKP0 = 0 (CSIHnTCSS0) → CKP1 = 1 (CSIHnTCSS1))

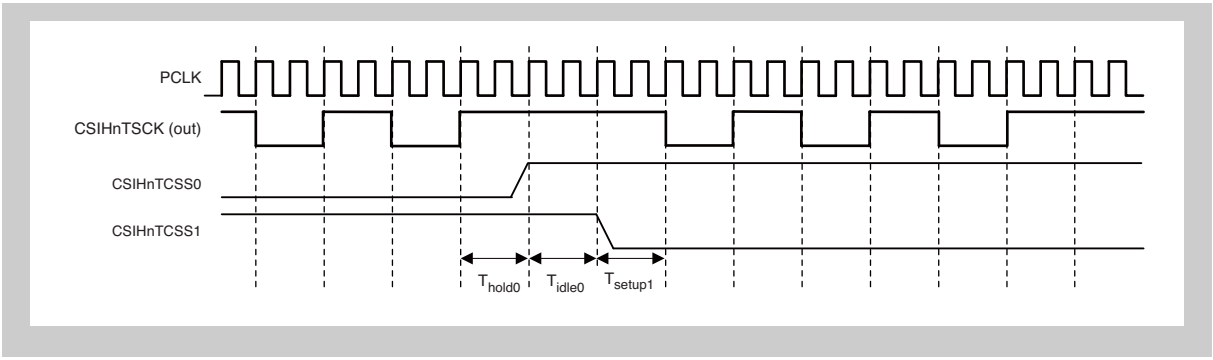


Figure 22-10 Clock phase timing (in the case of PCLK/4, $T_{hold0} = T_{setup1}$ = 0.5 SCK, T_{idle0} = 0.5 SCK, CKP0 = 0 (CSIHnTCSS0) → CKP1 = 0 (CSIHnTCSS1))

(2) Changing the data phase

The bit CSIHnCFGx.CSIHnDAPx defines the phase of the data bits relative to the clock.

If CSIHnCFGx.CSIHnDAPx = 0, the transmission clock CSIHnTSCK holds its level after the last bit of a data is transferred.

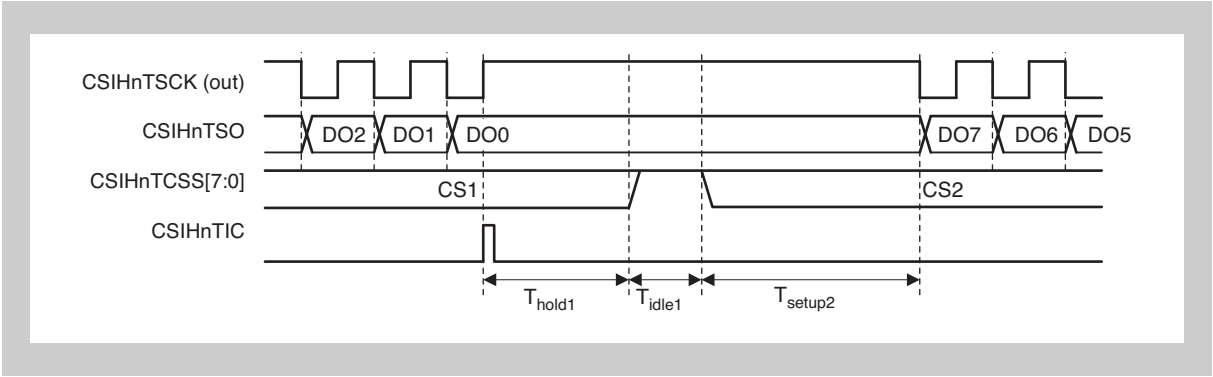


Figure 22-11 Data phase timing with CSIHnCFG1.CSIHnCKP1 = 0, CSIHnCFG1.CSIHnDAP1 = 0 and CSIHnCFG2.CSIHnCKP2 = 0, CSIHnCFG2.CSIHnDAP2 = 0

If the default clock phase changes between two consecutive chip selects, the transmission clock CSIHnTSCK changes its level after the last bit of the first data is transferred:

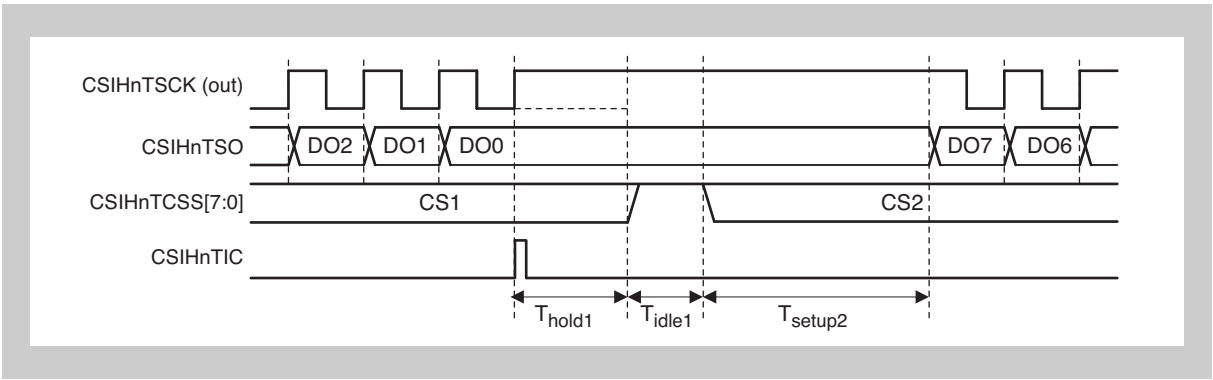


Figure 22-12 Data phase timing with CSIHnCFG1.CSIHnCKP1 = 1, CSIHnCFG1.CSIHnDAP1 = 0 and CSIHnCFG2.CSIHnCKP2 = 0, CSIHnCFG2.CSIHnDAP2 = 1

Note that the minimum idle time of one CSIHnTSCK cycle is automatically inserted, if CSIHnCFGx.CSIHnIDx[2:0] = 0 (T_{idle1} = 0.5 CSIHnTSCK cycles).

22.3.5 The job concept

In terms of CSIH, a job consists of a number of data that are transferred.

Note The job mode can only be used in master mode (CSIHnCTL2.CSIHnPRS[2:0] = 111_B).

Job mode enabled The job mode is enabled and disabled by CSIHnCTL1.CSIHnJE, while the CSIH is disabled by CSIHnCTL0.CSIHnPWR = 0.

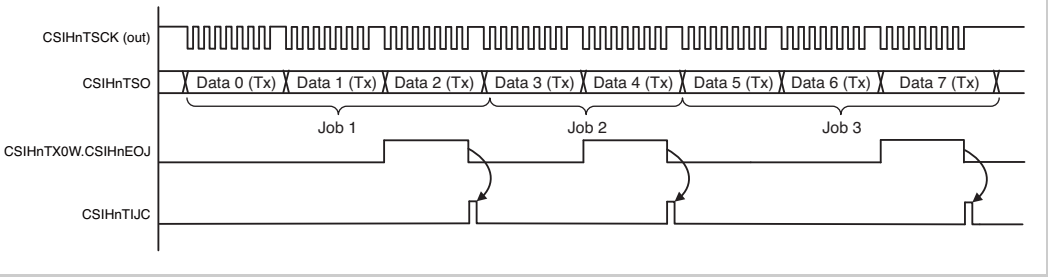


Figure 22-13 Job examples

A job ends when a data with the end-of-job bit set, i.e. with CSIHnTX0W.CSIHnEOJ = 1.

A communication stop can be specified to occur after a job has finished. This is done by setting CSIHnCTL0.CSIHnJOBE. When CSIHnJOBE is set, the communication continues until a data is sent, for which the CSIHnEOJ bit was set. After this data is sent, the communication is stopped and the end-of job-interrupt CSIHnTIJC is generated.

22.3.6 Serial clock selection

In master mode, the transmission baudrate is selectable using

- CSIHnCTL2.CSIHnPRS[2:0]
- CSIHnCTL2.CSIHnBRS[11:0]
- CSIHnCFGx.CSIHn.CSIHnPSCLx

While the settings in the CSIHnCTL2 register determine the transmission base clock CSIHnBPCLK, a chip select dedicated prescaler, controlled by CSIHnCFGx.CSIHn.CSIHnPSCLx, allows to generated different baudrates for different chip selects.

The following figure shows a block diagram of the baudrate generator.

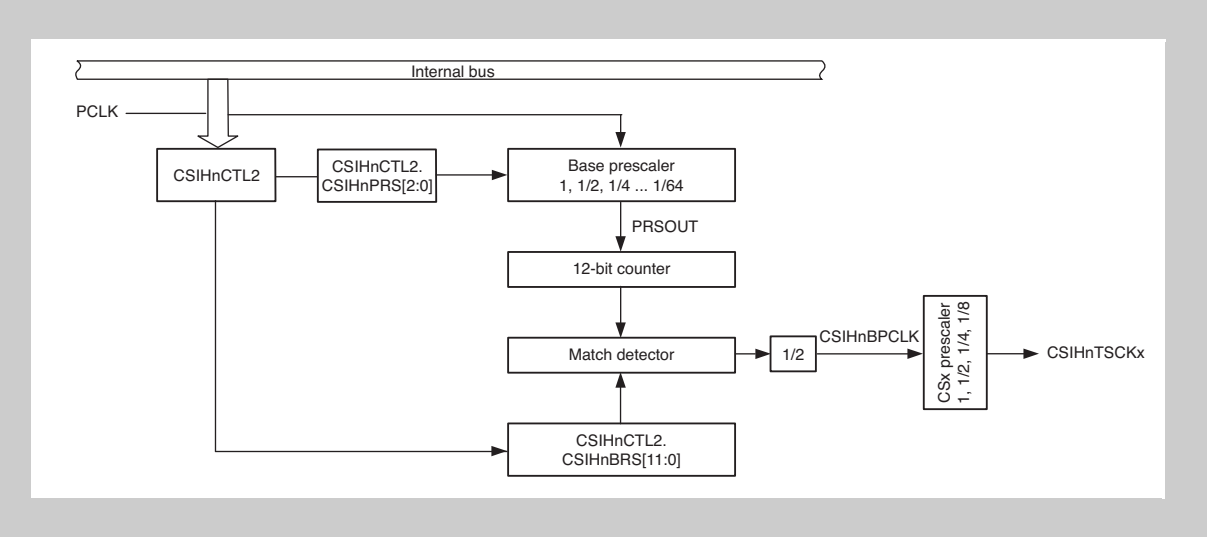


Figure 22-14 Baud rate generator block diagram

Clearing CSIHnCTL2.CSIHnBRS[11:0] disables the baudrate generator, and thus all CSIHnTSCKx are stopped.

Baud rate calculation

The baudrate is calculated as:

$$CSIHnTSCKx = PCLK / (2^m \times k \times 2 \times 2^j)$$

where

m = CSIHnCTL2.CSIHnPRS[2:0] = 0 to 6

k = CSIHnCTL2.CSIHnBRS[11:0] = 1 to 4095

j = CSIHnCFGx.CSIHnPSCLx[1:0]

Baud rate limits

When setting the baud rate, please note:

- Maximum acceptable baud rate in master mode is PCLK / 4.
- Maximum acceptable baud rate in slave mode is PCLK / 6 (must be ensured by the external master).
- Minimum baud rate in both modes is PCLK / 524160.

Caution There might be restrictions on the maximum baud rate that can actually be used depending on the product. Specify the baud rate so as not to exceed the maximum rate for each product.

Example If PCLK = 80 MHz, the maximum baud rate is

- 20.0 Mbps (PCLK / 4) in master mode
- 13.3 Mbps (PCLK / 6) in slave mode

The slowest baud rate is 190.8 bps (PCLK / 524160).

22.3.7 CSIH buffer memory

The CSIH has a configurable RAM that can be used for buffered I/O. The size is 128 words. One word consists of 32 bits of data.

The following configurations are available:

Mode	CSIHnCTL0. CSIHnMBS	CSIHnMCTL0. CSIHnMMS[1:0]
FIFO mode	0	00_B
Dual buffer mode		01_B
Transmit-only buffer mode		10_B
Direct access mode	1	X

(1) FIFO mode

In FIFO mode, data can be written to the CSIHnTX0W register without waiting for completion of the transmission, and data can be received without reading the CSIHnRX0W register immediately. provided the FIFO is not full.

Data to be transmitted is stored to the FIFO memory. Transmission and reception occur simultaneously – one bit is sent, one bit is received. That means, received data overwrites the transmitted data in the FIFO.

CSIH automatically updates the respective FIFO memory pointers when a data package is committed, sent or received:

Pointer description	Control bits	Range
Number of words that have not been transmitted	CSIHnSTR0.CSIHnSPF[7:0]	0 to 128
Number of words stored in the reception FIFO buffer	CSIHnSTR0.CSIHnSRP[7:0]	0 to 128
Address of data to be sent	CSIHnMRWP0.CSIHnTRWA[6:0]	0000_H to $01FC_H$
Address of received data	CSIHnMRWP0.CSIHnRRA[6:0]	0000_H to $01FC_H$

The CSIH status register contains also two FIFO status flags:

- CSIHnSTR0.CSIHnFLF: FIFO full
- CSIHnSTR0.CSIHnEMF: FIFO empty

When this mode is started, bit CSIHnSTCR0.CSIHnPCT must be set. This resets all FIFO pointers and flags.

(2) Dual buffer mode

In this mode, the memory is divided into two parts of equal size – this means 64 words for transmit data and 64 words for received data. In dual buffer mode, the respective buffer pointers indicate:

Pointer description	Pointers[a]	Range
Destination address for data written to or read from CSIHnTX0W/H	CSIHnMRWP0.CSIHnTRWA[6:0]	0000_H to $00FC_H$
Address of data read from CSIHnRX0W/H	CSIHnMRWP0.CSIHnRRA[6:0]	0000_H to $00FC_H$
Transmission pointer	CSIHnMCTL2.CSIHnSOP[6:0]	0000_H to $00FC_H$

a) Each pointer is automatically incremented after each read or write.

(3) Transmit-only buffer mode

In this mode the entire memory is used to save transmission data.

Received data must be read directly from CSIHnRX0W/H.

In transmit-only buffer mode, the respective buffer pointer is:

Pointer description	Pointer[a]	Range
Destination address for data written to or read from CSIHnTX0W/H	CSIHnMRWP0.CSIHnTRWA[6:0]	0000_H to $01FC_H$
Transmission pointer	CSIHnMCTL2.CSIHnSOP[6:0]	0000_H to $01FC_H$

a) Each pointer is automatically incremented after each read or write.

(4) Direct access mode

In direct access mode, the CSIH memory is completely bypassed:

- Transmission data provided by the CPU to the transmission data register CSIHnTX0W or CSIHnTX0H is directly copied to the shift register.
- Reception data is directly copied from the shift register to the reception data register CSIHnRX0W or CSIHnRX0H.

22.3.8 Data transfer modes

(1) Transmit-only mode

Setting CSIHnCTL0.CSIHnTXE = 1 and CSIHnCTL0.CSIHnRXE = 0 puts the CSIH in transmit-only mode. Start of transmission depends on the memory mode:

- In case of FIFO or direct access mode, transmission starts when transmit data is written to the CSIHnTX0W or CSIHnTX0H register.
- In case of dual buffer or transmit-only buffer mode, transmission starts when bit CSIHnMCTL2.CSIHnBTST is set.

(2) Receive-only mode

Setting CSIHnCTL0.CSIHnTXE = 0 and CSIHnCTL0.CSIHnRXE = 1 puts the CSIH in receive-only mode.

In master mode, the start of reception depends on the memory mode:

- In case of FIFO, transmit-only buffer or direct access mode, reception starts when dummy data is written in the CSIHnTX0W or CSIHnTX0H register.
- In the dual buffer or transmit-only buffer mode, transmission starts when the CSIHnMCTL2.CSIHnBTST bit is set.

In slave mode, reception starts as soon as the transmission clock CSIHnTSCK from the master is received. It is not necessary to write data to the CSIHnTX0W or CSIHnTX0H register of the slave.

(3) Transmit & receive mode

Setting CSIHnCTL0.CSIHnTXE = 1 and CSIHnCTL0.CSIHnRXE = 1 puts the CSIH in transmit/receive mode.

The start of the communication (transmission and reception) depends on the memory mode:

- In case of FIFO or direct access mode, communication starts when transmit data is written to the CSIHnTX0W or CSIHnTX0H register.
- In case of dual buffer or transmit-only buffer mode, communication starts when bit CSIHnMCTL2.CSIHnBTST is set.

(4) Summary

The following table provides a summary. It shows how the data transfer is started in the various memory, operating, and transfer modes.

Table 22-8 Start of data transfer

Memory modes	Transfer modes	Operating modes	Condition for starting a data transfer
FIFO mode, direct access mode	Transmission mode Transmission/reception mode	Master, slave	When transmission data is written to the CSIHnTX0W or CSIHnTX0H register
	Reception mode	Master	When dummy data is written to the CSIHnTX0W or CSIHnTX0H register
		Slave	When the serial clock CSIHnTSCK is received from the master
Transmit-only buffer mode, dual buffer mode	Transmission mode Transmission/reception mode Reception mode	Master, slave	When 1 is written to CSIHnMCTL2.CSIHnBTST

22.3.9 Data length selection

(1) Data length between 7 and 16 bits

CSIHnCFGx.CSIHnDLSx[3:0] can be used to select the data packet length for each chip select signal in the range from seven to 16 bits. The examples below show the communication with MSB first (CSIHnCFGx.CSIHnDIRx = 0).

Data length = 16 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 0000_B):

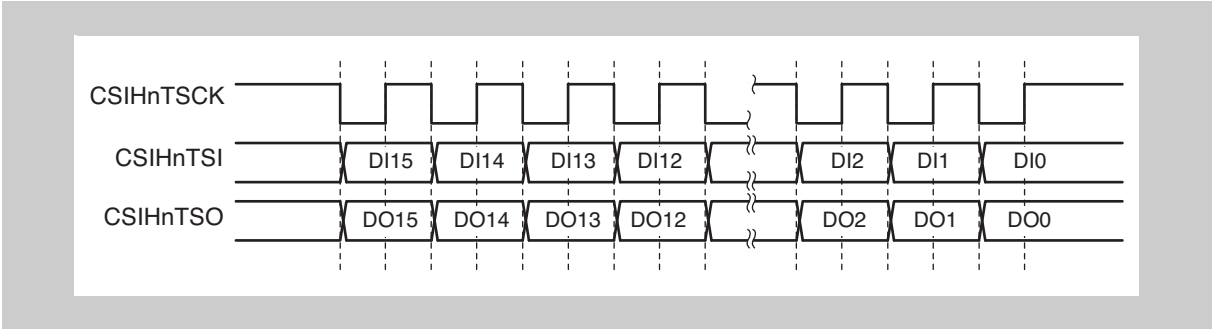


Figure 22-15 16-bit data, MSB first

Data length = 14 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1110_B):

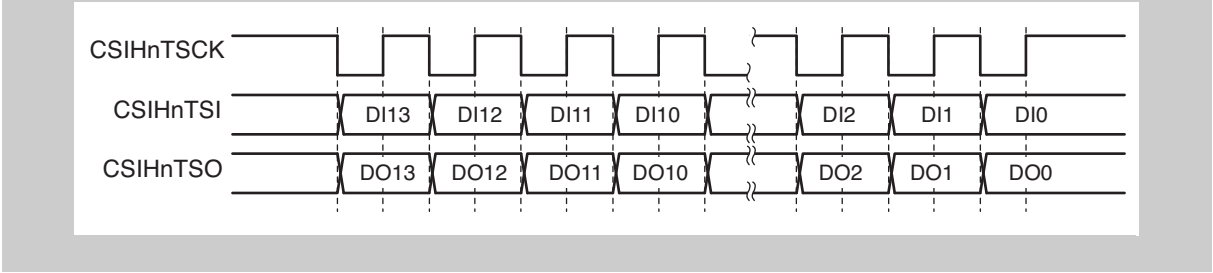


Figure 22-16 14 bit-data, MSB first

(2) Data length greater than 16 bits

If the length of the data to be sent/received exceeds 16 bits, the extended data length (EDL) feature can be used.

The EDL function is enabled by setting CSIHnCTL1.CSIHnEDLE.

The operation and setup procedure of the EDL function are described below:

- Data is divided into 16-bit blocks and a remainder. For example, a 42-bit character string is divided into two 16-bit blocks and a 10-bit remainder.
- For the remainder, the data length is specified for the CSIHnCFGx.CSIHnDLSx[3:0] bits.
- When transmitting 16-bit blocks, set the CSIHnTX0W.CSIHnEDL bit. In this case, the data written to CSIHnTX0W is sent as a 16-bit data length regardless of the CSIHnCFGx.CSIHnDLSx[3:0] bits.
- When the specified length of data (the remainder when CSIHnTX0W.CSIHnEDL = 0) is transmitted, the transfer ends.

Example Example of transmitting the 40-bit data $123456789A_H$ to CS0

The 40-bit data is divided into two 16-bit blocks of data and one 8-bit block of data.

- Initialize CSIHnCFGx.CSIHnDLSx[3:0] = 8_D.
- To send the string $123456789A_H$ with MSB first, write the following sequence to CSIHnTX0W:
  - $2000\ 1234_H$ (CSIHnTX0W.CSIHnEDL = 1)
  - $2000\ 5678_H$ (CSIHnTX0W.CSIHnEDL = 1)
  - $0000\ 009A_H$ (CSIHnTX0W.CSIHnEDL = 0)

The following figure illustrates the timing.

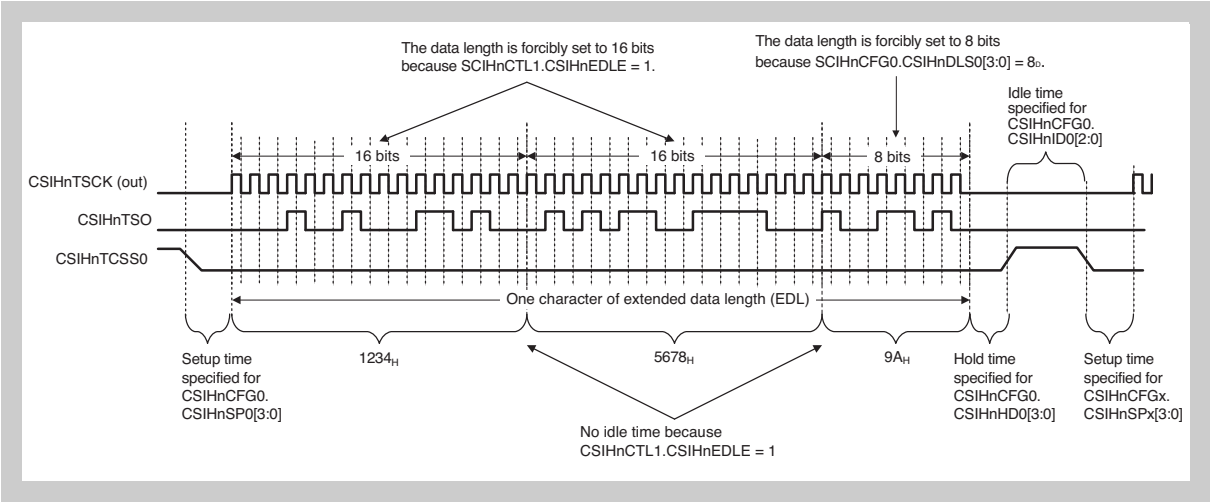


Figure 22-17 EDL timing chart

Notes

1. A data length less than 7 bits can be specified only when the EDL function is used.
2. It is not possible to send two consecutive data with a data length of less than 7 bits.
3. If parity is enabled, the parity bit is added after the last bit.
4. The following describes an example where the transmitted data is 123456_H.
   - CSIHnCFGx.CSIHnDIR is cleared to 0 (MSB first).
     $2000\ 1234_H$ is written to CSIHnTX0W (CSIHnTX0W.CSIHnEDL = 1).
     $0000\ 0056_H$ is written to CSIHnTX0W (CSIHnTX0W.CSIHnEDL = 0).
   - CSIHnCFGx.CSIHnDIR is set to 1 (LSB first).
     $2000\ 3456_H$ is written to CSIHnTX0W (CSIHnTX0W.CSIHnEDL = 1).
     $0000\ 0012_H$ is written to CSIHnTX0W (CSIHnTX0W.CSIHnEDL = 0).
5. The EDL function cannot be used in the slave mode (CSIHnCTL1.CSIHnPRS[2:0] = 1, 1, 1) and reception mode (CSIHnCTL0.CSIHnTXE = 0, CSIHnCTL0.CSIHnRXE = 1).

22.3.10 Serial data direction selection

The serial data direction is selectable for each chip select signal using the CSIHnDIRx bit in the CSIHnCFGx register.

The examples below show the communication for a data length of 8 bit (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B):

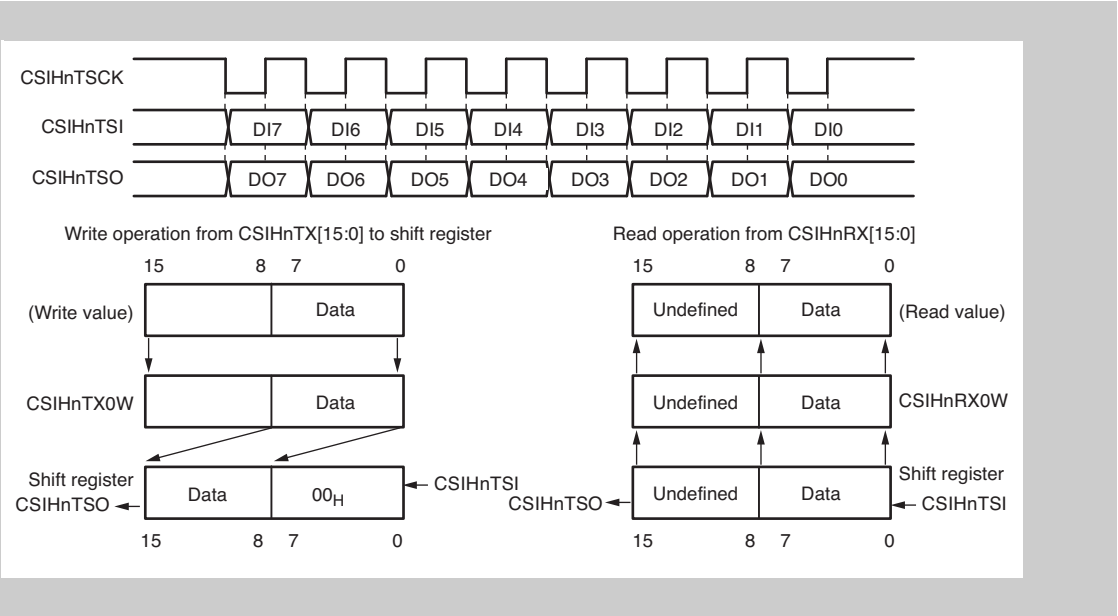


Figure 22-18 Serial data direction select function - MSB first (CSIHnDIR = 0)

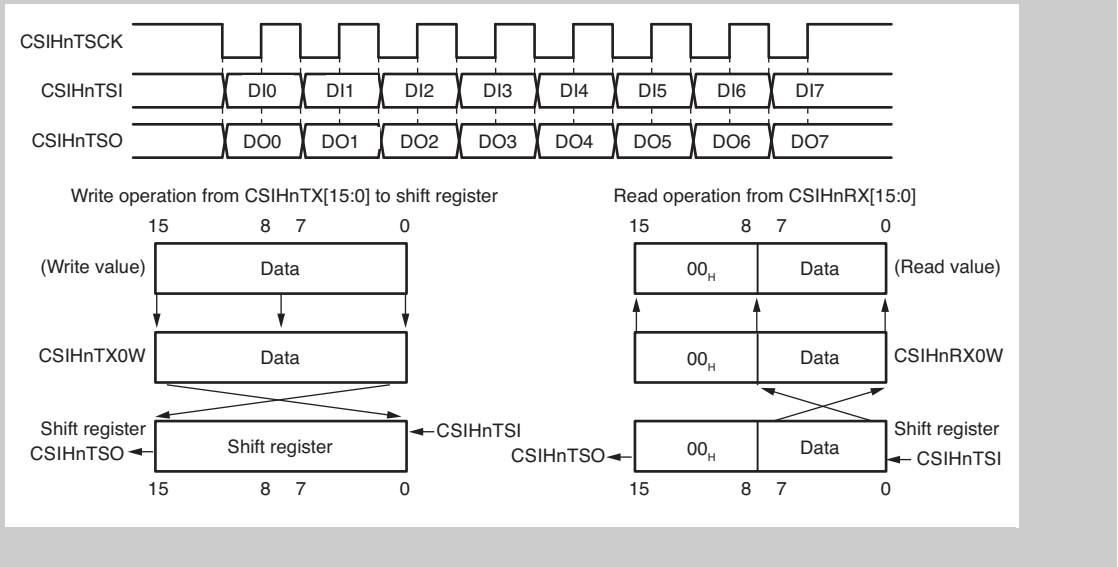


Figure 22-19 Serial data direction select function - LSB first (CSIHnDIR = 1)

22.3.11 Communication in slave mode

The following figure illustrates the communication signals and timings in slave mode.

In slave mode, the data transfer configuration is determined by the CSIHnCFG0 register.

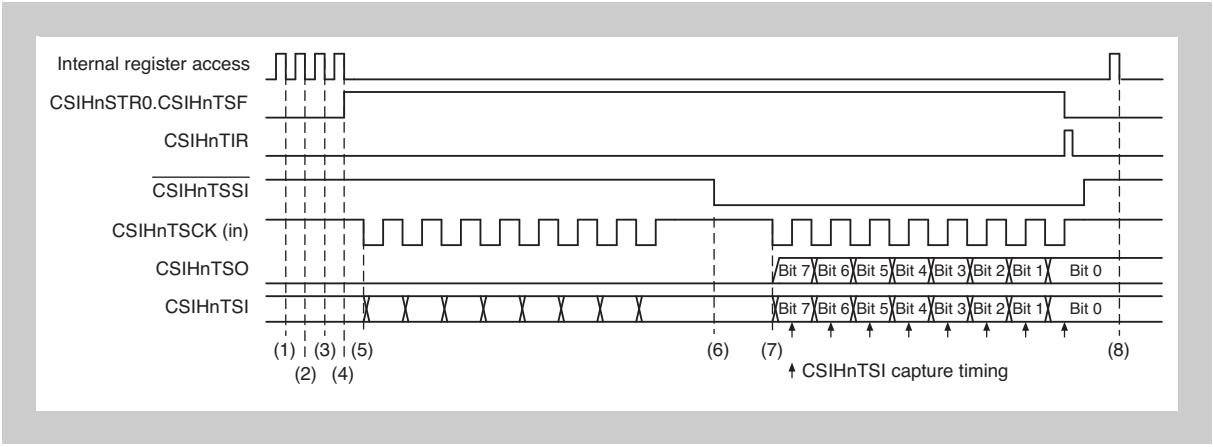


Figure 22-20 Transmit/receive communication timing in slave mode

1. CSIH is placed in the slave mode by setting CSIHnCTL2.CSIHnPRS[2:0] to 111_B. The $\overline{\text{CSIHnTSSI}}$ signal is enabled (CSIHnCTL1.CSIHnSSE = 1).
2. CSIHnCTL1.CSIHnCKR and CSIHnCFG0.CSIHnDAP0 are 0, the data length is 8 bits (CSIHnCFG0.CSIHnDLS0[3:0] = 1000_B), and the data direction is MSB first (CSIHnCFG0.CSIHnDIR0 = 0).
3. CSIH is set to the transmission/reception mode (CSIHnCTL0.CSIHnPWR = 1, CSIHnCTL0.CSIHnTXE = 1, and CSIHnCTL0.CSIHnRXE = 1). Communication is enabled to start.
4. If transfer data is written to the transmission data register CSIHnTX0W or CSIHnTX0H, the transfer status flag CSIHnSTR0.CSIHnTSF is automatically set, and the system waits for the $\overline{\text{CSIHnTSSI}}$ signal to become low.
5. Data transmission/reception does not start while the $\overline{\text{CSIHnTSSI}}$ signal is high, even if the external serial clock CSIHnTSCK is input. CSIHnTSO retains the value, and input to CSIHnTSI is ignored.
6. When the $\overline{\text{CSIHnTSSI}}$ signal becomes low, it indicates that CSIHnTSO is enabled and data transmission is possible.
7. If a serial clock is input while $\overline{\text{CSIHnTSSI}}$ is low, the transfer data is transmitted from CSIHnTSO in synchronization with the serial clock, and data is received from CSIHnTSI at the same time.
8. The CSIHnRX0W or CSIHnRX0H register is read.

Note For details about the operating procedure in the slave mode for each operation mode, see *22.5 "Operating Procedures"*.

22.3.12 CSIH interrupt requests

CSIH can generate the following interrupt requests:

- CSIHnTIC (communication interrupt)
- CSIHnTIR (reception interrupt)
- CSIHnTIRE (error interrupt)
- CSIHnTIJC (job completion interrupt)

(1) CSIHnTIC (communication interrupt)

The conditions for generating CSIHnTIC differ depending on the memory mode and whether the job mode is enabled.

The following table gives an overview.

Table 22-9 CSIHnTIC generation

Memory modes	Interrupt source	
	Job mode disabled CSIHnCTL1.CSIHnJE = 0	**Job mode enabled CSIHnCTL1.CSIHnJE = 1**
FIFO mode	CSIHnTIC is generated immediately before the transmission data in the FIFO buffer disappears to inform the application that new data must be added. CSIHnTIC is generated when the number of transmission data items remaining in the FIFO buffer, CSIHnSTR0.CSIHnSPF[7:0], becomes equal to CSIHnMCTL1.CSIHnFES[6:0].	
	However, CSIHnTIC is not generated if the job is interrupted[a].	–
Transmit-only buffer mode, dual buffer mode	CSIHnTIC is generated when communication ends (as specified by the CSIHnMCTL2.CSIHnND[7:0] bits).	CSIHnTIC is generated when data is transmitted while CSIHnTX0W.CSIHnCIRE is “1”. However, when the data and a job interrupt request[a] are transmitted while CSIHnTX0W.CSIHnCIRE is “1”, CSIHnTIJC is generated instead of CSIHnTIC.
Direct access mode	CSIHnTIC is generated each time a data transfer is performed. However, CSIHnTIC is not generated if the job is interrupted[a].	Except when communication is interrupted, CSIHnTIC is generated each time a data transfer is performed. However, when the data and a job interrupt request[a] are transmitted while CSIHnTX0W.CSIHnCIRE is “1”, CSIHnTIJC is generated instead of CSIHnTIC.

a) Job abortion condition: CSIHnTX0W.CSIHnEOJ = 1 *and* CSIHnCTL0.CSIHnJOBE = 1

CSIHnTIC in direct access mode

The following example shows the CSIHnTIC behavior in direct access mode.

The following example assumes:

- Master mode
- Direct access memory mode
- No delay for any interrupt (CSIHnCTL1.CSIHnSIT = 0)
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- Data length 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B)
- Normal CSIHnTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)

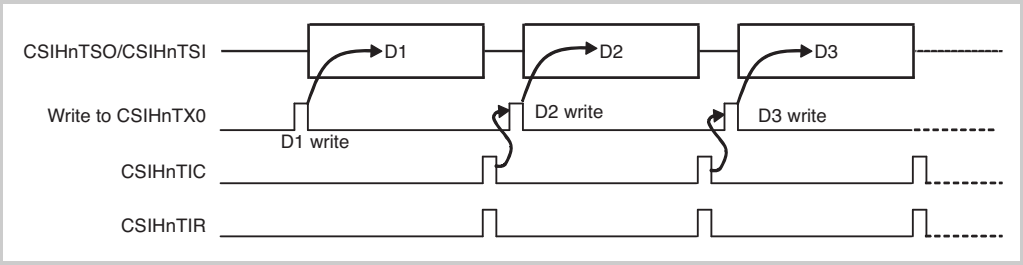


Figure 22-21 Generation of CSIHnTIC after transfer (CSIHnCTL1.CSIHnSLIT = 0)

If job mode is enabled (CSIHnCTL1.CSIHnJE = 1) and a job ends because data is sent with CSIHnTX0W.CSIHnEOJ = 1 and communication stop is requested (CSIHnCTL0.CSIHnJOBE = 1), then CSIHnTIC is replaced by the job completion interrupt CSIHnTIJC.

CSIHnTIC can also be set up to occur as soon as the CSIHnTX0 register is free for the next data. This is specified by setting CSIHnCTL1.CSIHnSLIT = 1.

Note This mode allows faster data transfer but is only available in direct access memory mode.

The effect is illustrated in the figure below.

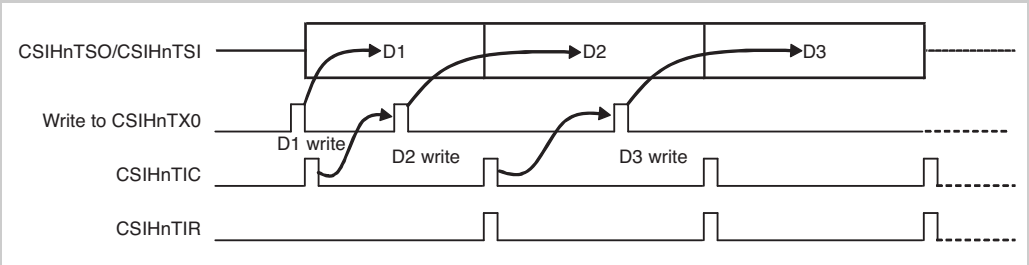


Figure 22-22 Immediate generation of CSIHnTIC (CSIHnCTL1.CSIHnSLIT = 1)

Thus, the new data can be written in advance.

CSIHnTIC in FIFO mode

The following example shows the CSIHnTIC behavior in FIFO mode.

The following example assumes:

- Master mode
- FIFO memory mode
- No delay for any interrupt (CSIHnCTL1.CSIHnSIT = 0)
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- Data length 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B)

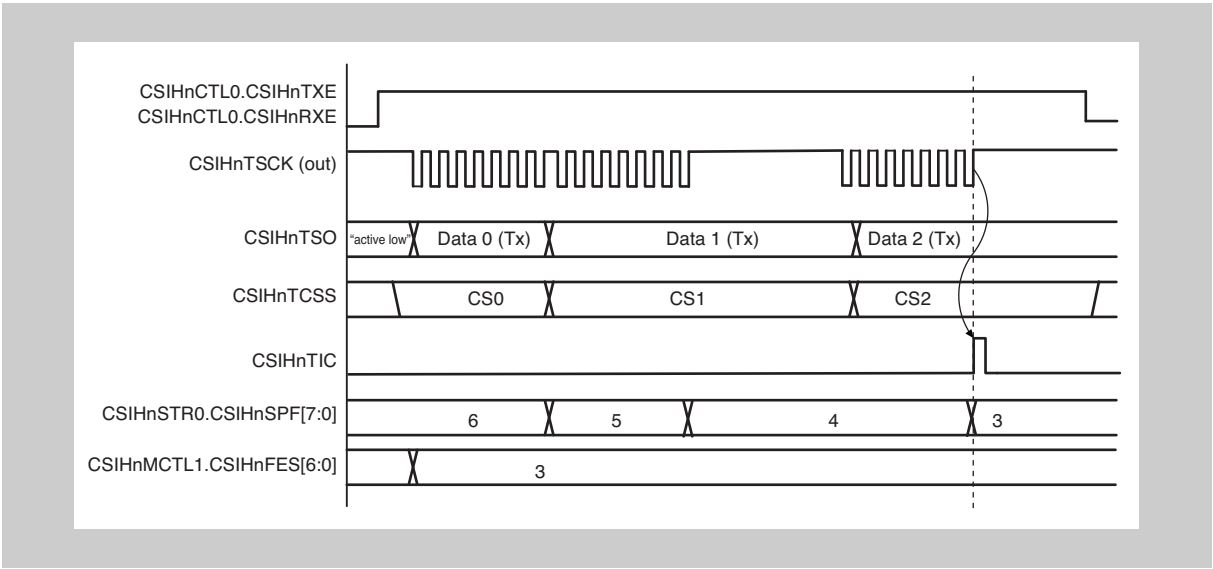


Figure 22-23 Generation of CSIHnTIC in FIFO memory mode

The condition for generating CSIHnTIC in the FIFO mode (an empty reception buffer) is specified for CSIHnMCTL1.CSIHnFES[6:0]. For the example in the above figure, three data items are specified as the condition. The CSIHnSTR0.CSIHnSPF[7:0] bits indicate the number of data items that remain in the FIFO buffer and have not been transmitted. When the number of remaining items matches the condition, the interrupt CSIHnTIC is generated.

CSIHnTIC in job mode

The following example shows the CSIHnTIC behavior in job mode.

The following example assumes:

- Master mode
- Job mode enabled (CSIHnCTL1.CSIHnJE = 1)
- No delay for any interrupt (CSIHnCTL1.CSIHnSIT = 0)
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- Data length 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B)
- Normal CSIHnTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)
- Dual buffer mode (CSIHnCTL0.CSIHnMBS = 0, CSIHnMCTL0.CSIHnMMS[1:0] = 01H)

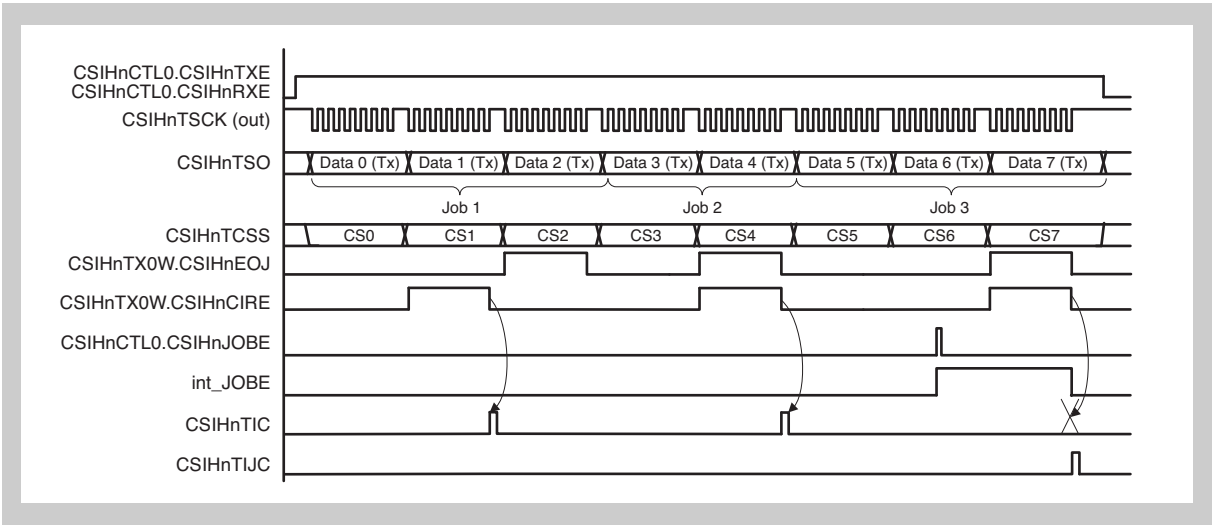


Figure 22-24 Generation of CSIHnTIC in job mode

Note The int_JOBE signal in the above timing chart is the internal signal of the CSIHnJOBE bit.

The rules for generating CSIHnTIC in job mode are:

Table 22-10 Generation of CSIHnTIC in job mode

Memory modes	CSIHnTX0W. CSIHnCIRE	CSIHnTX0W. CSIHnEOJ	CSIHnTIC
FIFO mode (CSIHnCTL1. CSIHnJE = 1)	0 (FIFO empty[a])	0	Generated
		1	CSIHnCTL0.CSIHnJOBE = 0: Generated CSIHnCTL0.CSIHnJOBE = 1: CSIHnTIJC is generated instead of CSIHnTIC.
	1 (FIFO empty[a])	0	Generated
		1	CSIHnCTL0.CSIHnJOBE = 0: Generated CSIHnCTL0.CSIHnJOBE = 1: CSIHnTIJC is generated instead of CSIHnTIC.
	0 (Data in FIFO)	0	Not generated
		1	Not generated CSIHnCTL0.CSIHnJOBE = 1: CSIHnTIJC is generated instead of CSIHnTIC.
	1 (Data in FIFO)	0	Not generated CSIHnTIJC is generated instead of CSIHnTIC.
		1	
Dual buffer mode, transmit-only buffer mode (CSIHnCTL1. CSIHnJE = 1)	0	0	Not generated
		1	Not generated CSIHnCTL0.CSIHnJOBE = 1: CSIHnTIJC is generated instead of CSIHnTIC.
	1	0	Generated
		1	CSIHnCTL0.CSIHnJOBE = 0: Generated CSIHnCTL0.CSIHnJOBE = 1: CSIHnTIJC is generated instead of CSIHnTIC.
Direct access mode (CSIHnCTL1. CSIHnJE = 1)	–	0	Generated
		1	
	–	0	Generated
		1	CSIHnCTL0.CSIHnJOBE = 1: CSIHnTIJC is generated instead of CSIHnTIC.

a) The value of CSIHnSTR0.CSIHnSPF7-0 is the same as that of CSIHnMCTL1.CSIHnFE6-0.

(2) CSIHnTIR reception interrupt

Depending on the memory mode and the job mode, this interrupt is generated according to the following conditions:

Table 22-11 CSIHnTIR interrupt generation

Memory mode	Master and slave	
	Job mode disabled CSIHnCTL1.CSIHnJE = 0	Job mode enabled CSIHnCTL1.CSIHnJE = 1
FIFO	This interrupt occurs when the FIFO buffer is almost full with received data, indicating to the application that the FIFO must be emptied. CSIHnTIR is generated, if the number of received data in the FIFO CSIHnSTR0.CSIHnSRP[7:0] equals CSIHnMCTL1.CSIHnFFS[6:0].	
Dual buffer	The interrupt is generated when communication ends (as specified by the CSIHnMCTL2.CSIHnND[7:0] bits) and CSIHnCTL0.CSIHnRXE = 1.	The interrupt is generated each time data is received if CSIHnCTL0.CSIHnRXE = 1.
Transmit-only buffer, direct access	The interrupt is generated each time data is received if CSIHnCTL0.CSIHnRXE = 1.	

In transmit-only or dual buffer mode, this interrupt is generated in receive-only and transmit/receive mode after each data has been received.

CSIHnTIR in direct access mode

The following example shows the CSIHnTIR behavior in direct access mode.

The following example assumes:

- Master mode
- Direct access mode
- No delay for any interrupt (CSIHnCTL1.CSIHnSIT = 0)
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- Data length 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B)

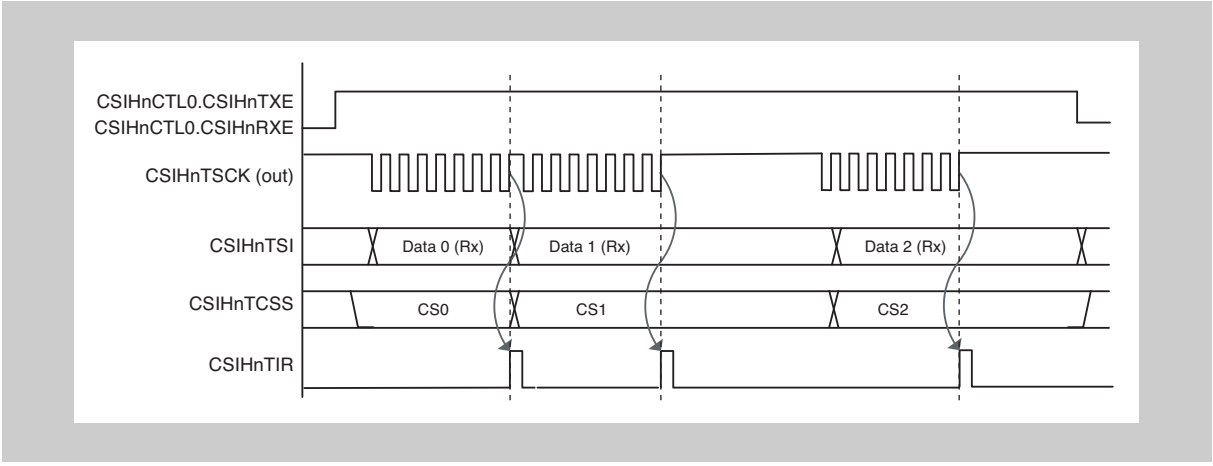


Figure 22-25 Generation of CSIHnTIR in direct access memory mode

CSIHnTIR in the dual buffer mode

The following example shows the CSIHnTIR behavior in buffer mode.

The following example assumes:

- Master mode
- Transmit-only or dual buffer mode
- No delay for any interrupt (CSIHnCTL1.CSIHnSIT = 0)
- Default clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- 8-bit data length (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B)
- Three data items transmitted (CSIHnMCTL2.CSIHnND[7:0] = 03_H)
- Job mode disabled (CSIHnCTL1.CSIHnJE = 0)

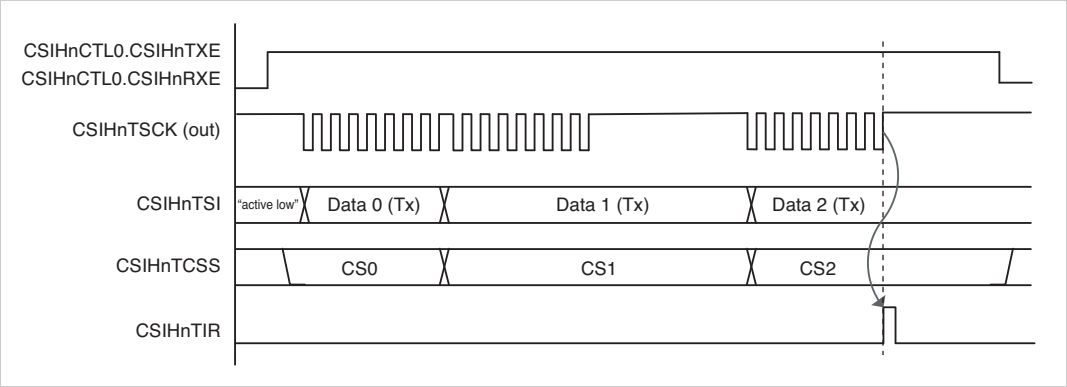


Figure 22-26 CSIHnTIR generation in the dual buffer mode

(3) CSIHnTIRE reception error interrupt

This interrupt is generated whenever an error is detected.

Table 22-12 Data error types

Error type	Communication status after error interrupt	Comment
FIFO overflow error	Interrupt is generated and communication continues	The data written to the FIFO is lost, but previously started communications are continued.
Parity error	Interrupt is generated and communication continues	–
Data consistency error	Interrupt is generated and communication continues	–
Timeout error	Interrupt is generated and communication continues	–
Overrun error	If CSIHnCTL1.CSIHnHSE = 0 (no handshake), communication continues after the interrupt is generated. (Communication does not stop.)	This error occurs (but only for the FIFO mode) if the CPU reads reception data after the number of reception data items reaches 0.
	If CSIHnCTL1.CSIHnHSE = 1 (handshake exists), communication is stopped according to the handshake. No interrupt is generated, and no overrun error occurs.	This error occurs when data reception finishes in one of the following statuses: • FIFO mode: The FIFO is full. (Overwriting is performed but the pointer is not incremented.) • Transmit-only buffer mode or direct access mode: Reception data remains in the CSIHnRX0 register.

The type of error that caused the generation of CSIHnTIRE is flagged in register CSIHnSTR0.

Additionally a parity and data consistency error flag is attached to the reception data in CSIHnRX0W.

For details about the various error types, refer to *22.3.14 “Error detection” on page 1336*.

(4) CSIHnTIJC job completion interrupt

This interrupt supports the handling of jobs – refer to *22.3.4 "Chip select timing details" on page 1310*. This interrupt is only available in master mode.

Job mode is enabled by setting CSIHnCTL1.CSIHnJE = 1. When CSIHnCTL1.CSIHnJE = 0, CSIHnTIJC is not generated.

Depending on the memory mode, this interrupt is generated according to the following conditions:

Table 22-13 CSIHnTIJC interrupt generation

Memory mode	Interrupt source	
	Job mode disabled CSIHnCTL1.CSIHnJE = 0	Job mode enabled CSIHnCTL1.CSIHnJE = 1
FIFO mode	Not applicable	After job abortion[a] is triggered, communication stops on job completion.
Transmit-only buffer mode		
Dual buffer mode		
Direct access mode		

a) Job abortion condition: CSIHnTX0W.CSIHnEOJ = 1 and CSIHnCTL0.CSIHnJOBE = 1

(5) Delay for all interrupts

In the master mode, all interrupts generated by the master can be delayed one half cycle of the serial clock CSIHnTSCK. This function cannot be used in the slave mode.

To specify this delay, set the CSIHnCTL1.CSIHnSIT bit to 1.

The figure below shows an example of using the interrupt delay function with the following settings: CSIHnCTL1.CSIHnSIT = 1 (interrupt delay enabled), CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0 (normal clock phase and data phase), and CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B (8-bit data length).

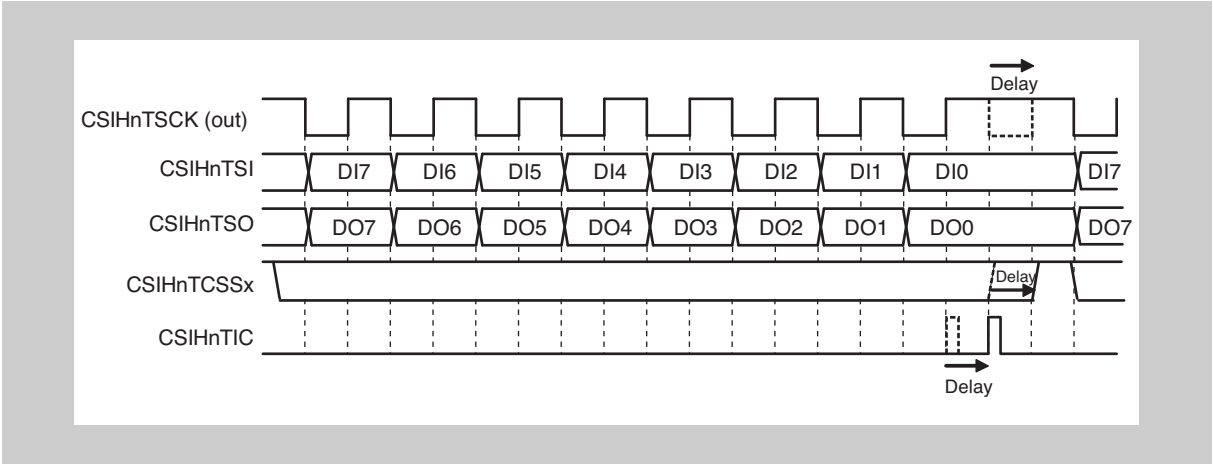


Figure 22-27 Interrupt delay function (CSIHnCTL1.CSIHnSIT = 1)

When CSIHnCTL1.CSIHnSIT is set to 1, a delay of half a serial clock cycle is added. This also delays the end of the current chip select signal (CSIHnTCSSx).

22.3.13 Handshake function

CSIH features a handshake function to synchronize the master and the slave devices. This function can be enabled/disabled by bit CSIHnCTL1.CSIHnHSE. For handshake, the CSIHnTRY signal is used. In addition, when the handshake function is disabled (CSIHnCTL1.CSIHnHSE = 0), the CSIHnTRY signal is output at the low level.

The timing depends on the data phase selection bit CSIHnCFGx.CSIHnDAPx.

(1) Slave mode

When CSIHnCTL1.CSIHnHSE = 1 and the slave is busy, the CSIHnTRY signal is output at the low level. The busy state refers to the following two cases:

- When the slave is not ready for transmission
- When the transmission data cannot be transferred from the shift register to the CSIHnRXW/H register because received data remains in the CSIHnRXW/H register

Also, the state differs according to the memory mode. *Table 22-14 "When the slave is not read for transmission"* and *Table 22-15 "When the transmission data cannot be transferred from the shift register"* describe the state for each memory mode.

Table 22-14 When the slave is not read for transmission

Memory mode	State
	CSIHnCTL0.CSIHnRXE = 1
FIFO mode	The next transmission data buffer is empty. (CSIHnSTR0.CSIHnEMF = 1)
Direct access mode	The next transmission data buffer is empty.
Dual buffer mode	CSIHnMCTL2.CSIHnBTST is not set to 1.
Transmit-only buffer mode	

Table 22-15 When the transmission data cannot be transferred from the shift register

Memory mode	State
	CSIHnCTL0.CSIHnRXE = 1
FIFO mode	The FIFO buffer is full. (CSIHnSTR0.CSIHnFLF = 1)
Direct access mode	The CSIHnRX0W/H register is full.
Dual buffer mode	–
Transmit-only buffer mode	The CSIHnRX0W/H register is full.

Note When the master mode is specified and the CSIHnTRY (in) signal is detected to be at the low level, the following transmission is put on hold and the system waits. While the system waits, the CSIHnTRY (in) signal level is checked every one-half clock cycle of CSIHnTCSS0. However, the chip select signal CSIHnTCSSx is output before the system starts waiting, and is held during the wait.

Cautions

1. For the master, specify settings so that only the CSIHnTRY pin is detected from the slave selected by CSIHnTCSSx.
2. Even if an active level signal from the slave to the master is detected during transmission, the next transmission is not performed until the current transmission operation ends.

Two descriptions are provided below, one for the FIFO mode and one for the direct access mode.

When the memory mode is FIFO mode

The slave is in transmit-only or transmit/receive mode but has no transmission data in its buffer. This status is indicated by the flag CSIHnSTR0.CSIHnEMF.

The following examples assume a data length of 8 bits.

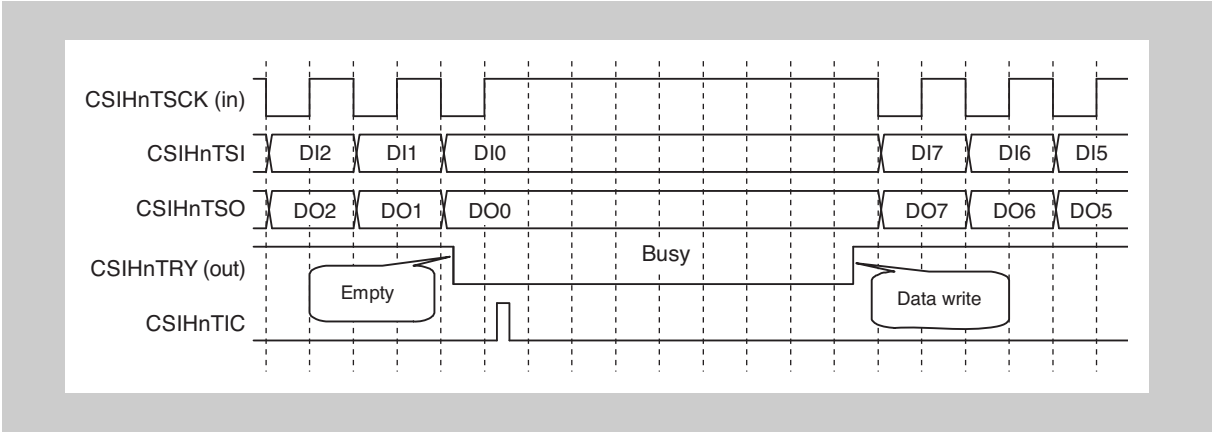


Figure 22-28 Busy signal from the slave (FIFO mode, CSIHnCFGx.CSIHnDAPx = 0)

The slave sets CSIHnTRY to high (“ready”) as soon as new transmission data is written to the FIFO.

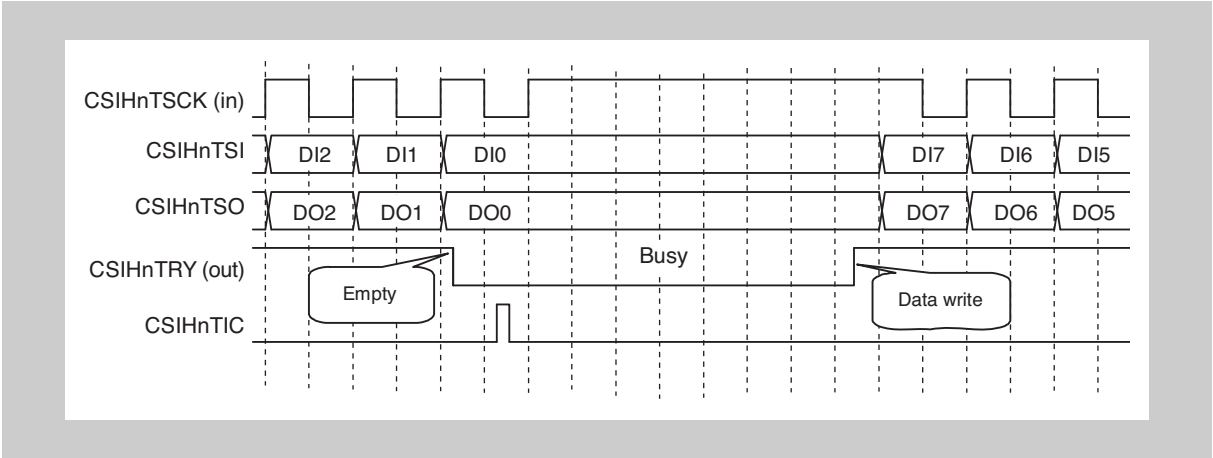


Figure 22-29 Busy signal from the slave (FIFO mode, CSIHnCFGx.CSIHnDAPx = 1)

When the memory mode is direct access mode

The slave is in receive-only or transmit/receive mode but previously received data is still in the CSIHnRX0 register, and new data cannot be copied from the shift register to CSIHnRX0 (CSIHnRX0 full condition).

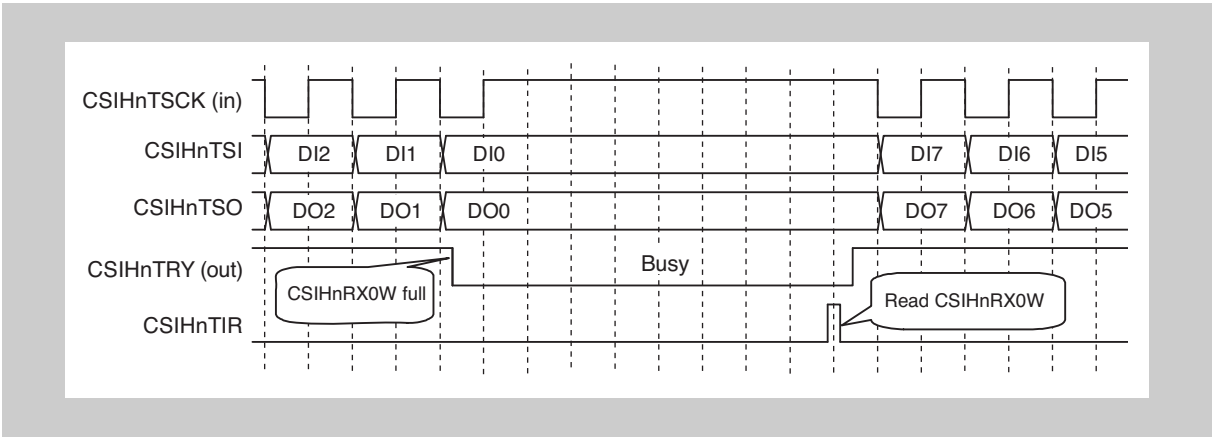


Figure 22-30 Busy signal from the slave (direct access mode, CSIHnCFGx.CSIHnDAPx = 0)

The slave sets CSIHnTRY to high (“ready”) as soon as the reception data register CSIHnRX0 has been read.

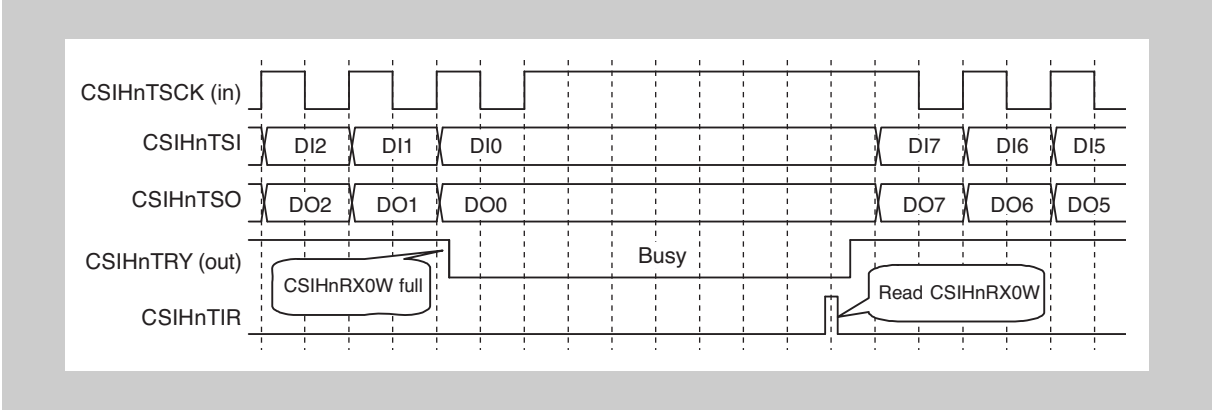


Figure 22-31 Busy signal from the slave (direct access mode, CSIHnCFGx.CSIHnDAPx = 1)

(2) Master mode

When the master detects that CSIHnTRY is at the low level, the following transfer is put on hold, and the master goes on stand by. It suspends the clock CSIHnTSCK.

The CSIHnTRY level is checked at each half clock cycle of CSIHnTSCK.

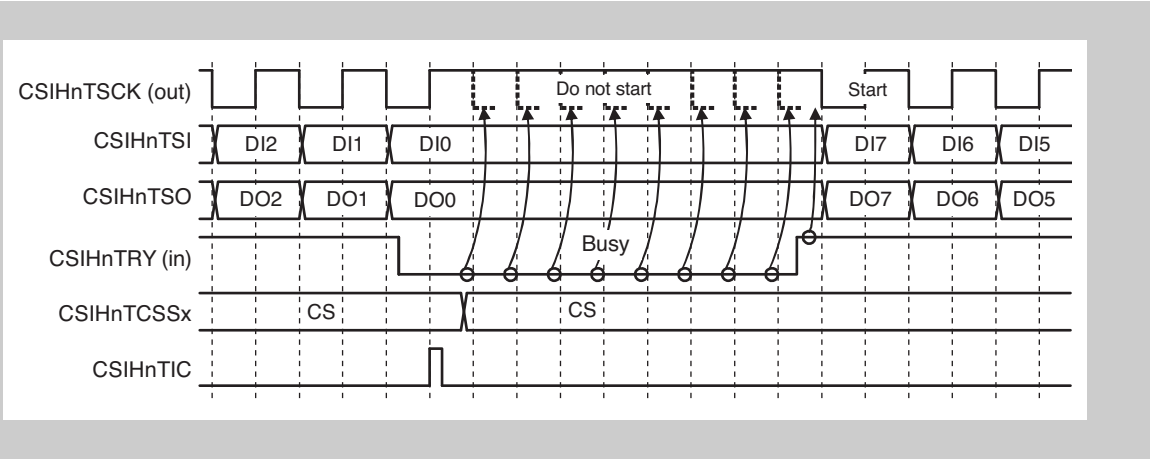


Figure 22-32 Master's reaction on CSIHnTRY (CSIHnCFGx.CSIHnDAPx = 0)

If the CSIHnTRY signal is pulled down while a data transfer is in progress, the serial clock is suspended after the transfer is complete.

The master resumes the communication as soon as CSIHnTRY becomes high (slave is "ready").

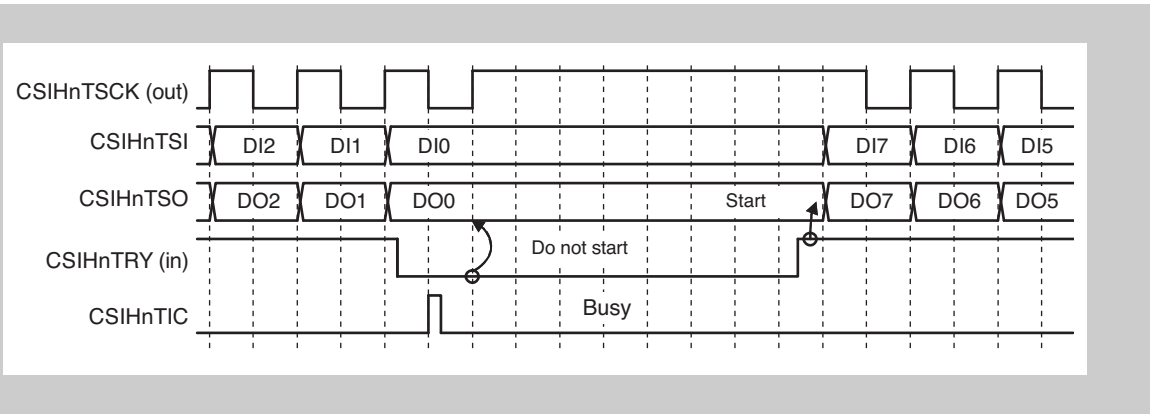


Figure 22-33 Master's reaction on CSIHnTRY (CSIHnCFGx.CSIHnDAPx = 1)

Caution If multiple slaves are connected, the master must only detect the CSIHnTRY signal of the slave it has selected for communication.

The slave must set the CSIHnTRY signal to low before the next data transfer starts. If the CSIHnTRY signal is set to low during a data transfer, the data transfer continues to the end.

22.3.14 Error detection

CSIH can detect five error types:

- Data consistency error (transmission data)
- Parity error (received data)
- Overrun error (received data)
- Timeout error (in FIFO mode)
- Overflow error (in FIFO mode)

Check for parity, data consistency and timeout errors can be enabled/disabled individually.

If one of these errors is detected, the interrupt request CSIHnTIRE is generated and the corresponding flag is set.

(1) Data consistency check

The purpose of the data consistency check is to ensure that the data physically sent as output signal is identical with the original data that was copied to the shift register.

The data consistency check can be enabled/disabled by bit CSIHnCTL1.CSIHnDCS. It is not active if data transmission is disabled (CSIHnCTL0.CSIHnTXE = 0).

When the data consistency check is active, the data transferred from CSIHnTX0W or CSIHnTX0H to the shift register is copied to a separate register. In addition, the physical levels at CSIHnTSO are read back via the CSIHnTDCS signal into an own shift register.

After completion of the transmission, the sent data is compared with the original transmission data.

Mismatch is considered as a data consistency error.

When a data consistency error occurs:

- Interrupt CSIHnTIRE is generated.
- Bit CSIHnSTR0.CSIHnDCE is set.

Additionally, CISHnRX0W.CSIHnTDCE is set with the corresponding data.

The function is illustrated in the following block diagram.

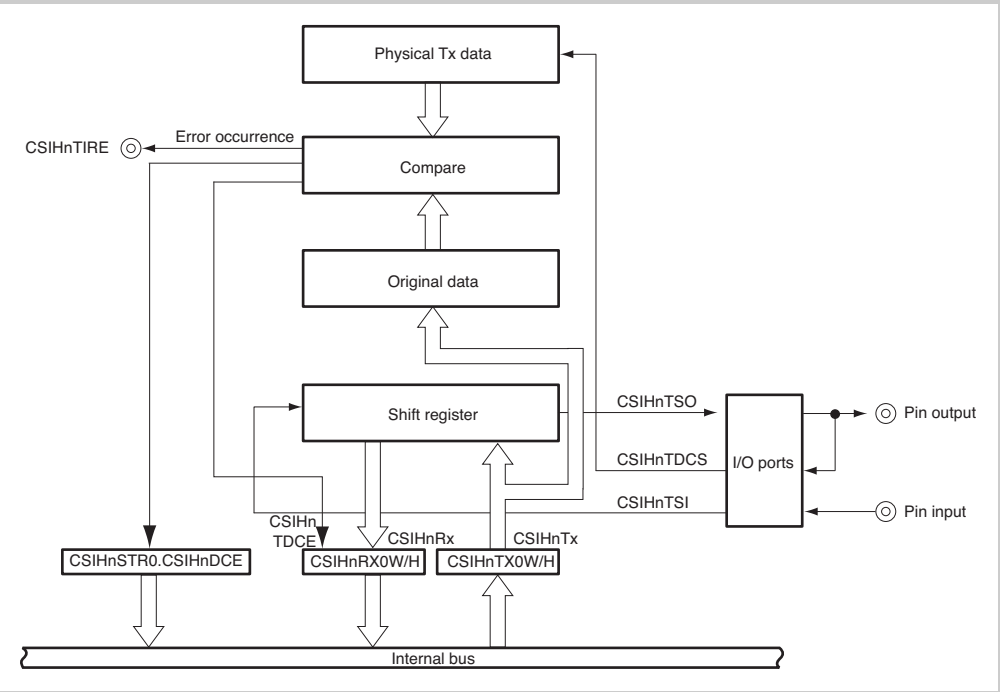


Figure 22-34 Data consistency check functional block diagram

(2) Parity check

Parity checks are often used to detect single bit errors during data transmission. CSIH can append a parity bit to the last data bit (even if extended data length is used).

The use and type of parity is specified in CSIHnCFGx.CSIHnPSx[1:0].

Parity check is enabled if CSIHnCFGx.CSIHnPSx[1] = 1.

The parity bit is checked after a reception is complete.

When a parity error occurs:

- Interrupt CSIHnTIRE is generated.
- Bit CSIHnSTR0.CSIHnPE is set.

Additionally, CISHnRX0W.CSIHnRPE is set with the corresponding data.

The following figure shows an example.

- Data length is 8 bits.
- The data transmitted is 05_H and 35_H.
- Data direction is LSB first.
- Parity type is odd.

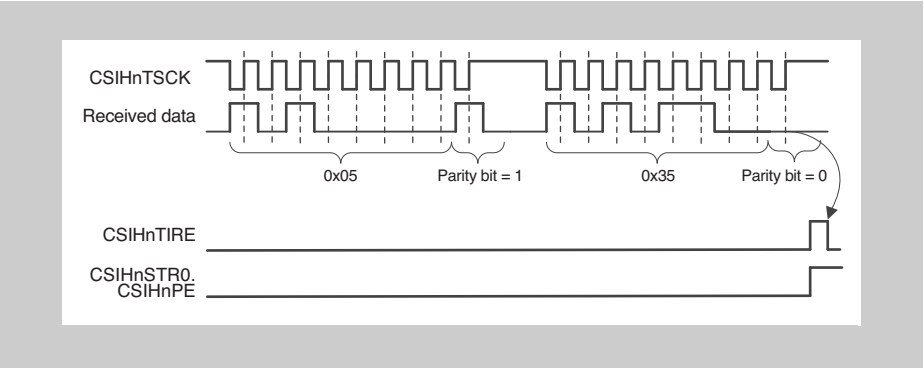


Figure 22-35 Parity check example

The parity bit of the first data is 1. There is no parity error, because the total number of ones (including the parity bit) is odd.

The parity bit of the second data is 0. This is detected as a parity error, because the total number of ones (including the parity bit) is even.

If using the extended data length (EDL) function, the parity bit is added after the last data bit.

(3) Timeout error

Timeout error checks are only possible in slave FIFO mode.

A timeout error occurs if neither of the following occurs within a specific time:

- Reading reception data in the FIFO buffer
- Reception of data by the FIFO buffer from CSIHnTSI

The time is defined in CSIHnMCTL0.CSIHnTO[4:0] in multiples of 8 times the transmission clock CSIHnSCK. Timeout error occurs when the specified time is exceeded (When CSIHnMCTL0.CSIHnTO[4:0] is cleared to 00000_B, the timeout time is not detected.).

A dedicated timeout counter measures the time between the last and the next read operation.

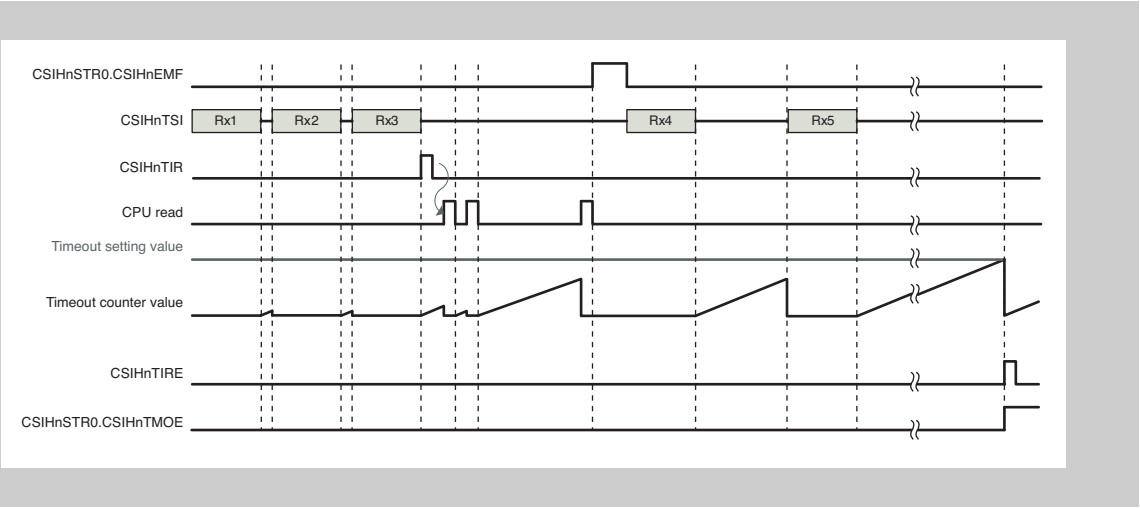


Figure 22-36 Timeout error check functional timing chart

The timeout counter starts when:

- Reception ends.
- Reading data from the CPU ends. (If the buffer is empty, the counter does not start.)
- A timeout error is detected.

After a timeout error is detected, if the system is left as is, the timeout counter restarts.

If the time specified for the CSIHnMCTL0.CSIHnTO[4:0] bits is reached again, another CSIHnTIR interrupt is output.

The timeout counter continues counting as long as reception data is not read. To stop the timeout counter, read all the reception data, or set CSIHnSTCR0.CSIHnPCT (to 1). However, the pointer is cleared in this case.

The timeout counter is reset when:

- Data is read.
- One new data item is received.
- A timeout error is detected.
- The CSIHnSTCR0.CSIHnPCT bit is set.

When a timeout error occurs:

- Interrupt CSIHnTIRE is generated.
- Bit CSIHnSTR0.CSIHnTMOE is set.

(4) Overflow error

Overflow errors can occur in the FIFO mode. An overflow error occurs when transmission data is written to the CSIHnTX0W or CSIHnTX0H register while the FIFO buffer is full of transmission data and reception data.

Example 100 data have been transmitted. That means, the FIFO contains 100 received data. The application starts to read the received data.

While the read operation is in progress, the application begins to write another set of 50 transmission data to the FIFO. However, only 10 received data have been read up to now, 90 are still in the FIFO.

In this case, only 38 cells are available for new transmission data. When the CPU tries to write the 39th data, an overflow error happens.

This is illustrated in the following figure:

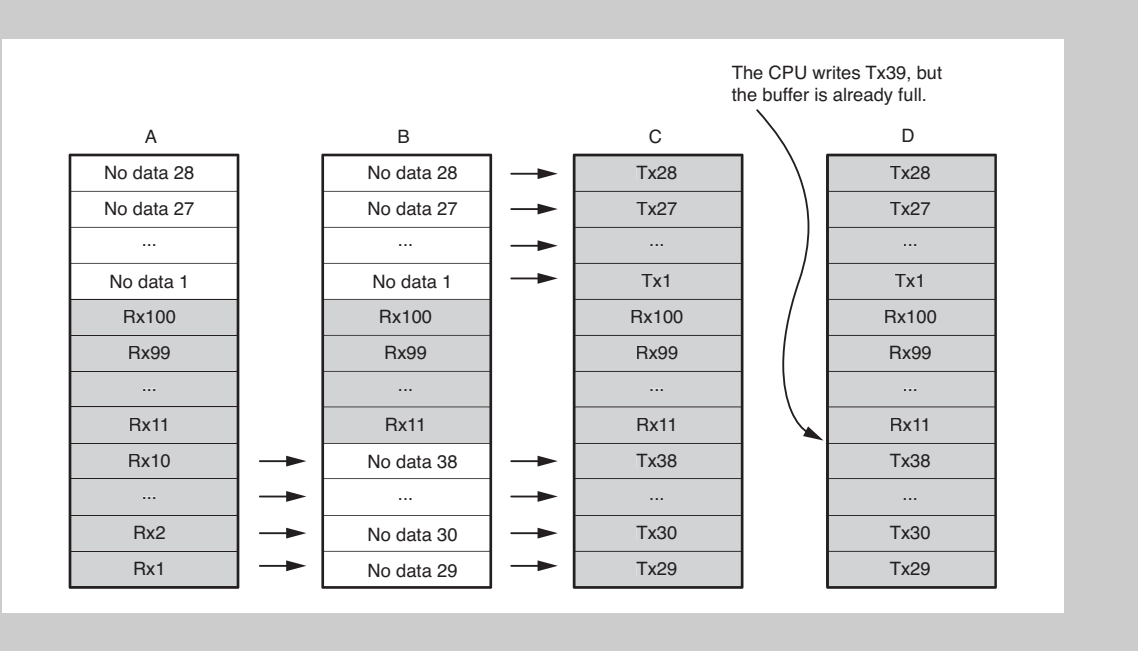


Figure 22-37 FIFO overflow

The data after 39 are discarded. The following figure shows the associated timing.

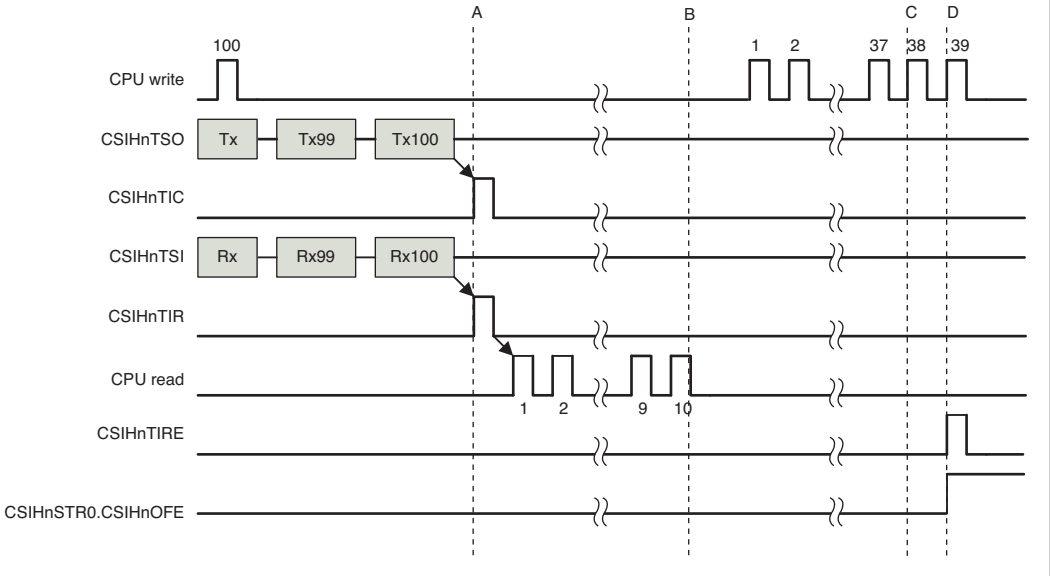


Figure 22-38 FIFO overflow timing

When an overflow error occurs:

- Interrupt CSIHnTIRE is generated.
- Bit CSIHnSTR0.CSIHnOFE is set.

(5) Overrun error

Overrun errors can occur in the direct access mode, transmit-only buffer mode, and FIFO mode. They cannot occur in the dual buffer mode.

Direct access/ transmit-only buffer

In direct access and transmit-only buffer mode, this error occurs when newly received data cannot be transferred from the shift register to the reception data register CSIHnRX0. This happens when CSIHnRX0 was not read and therefore contains previous reception data.

In the master mode, because the serial clock is stopped until the CPU reads reception data, overrun errors do not occur. In the slave mode, the handshake function can be used to avoid this problem.

The following figure illustrates the function.

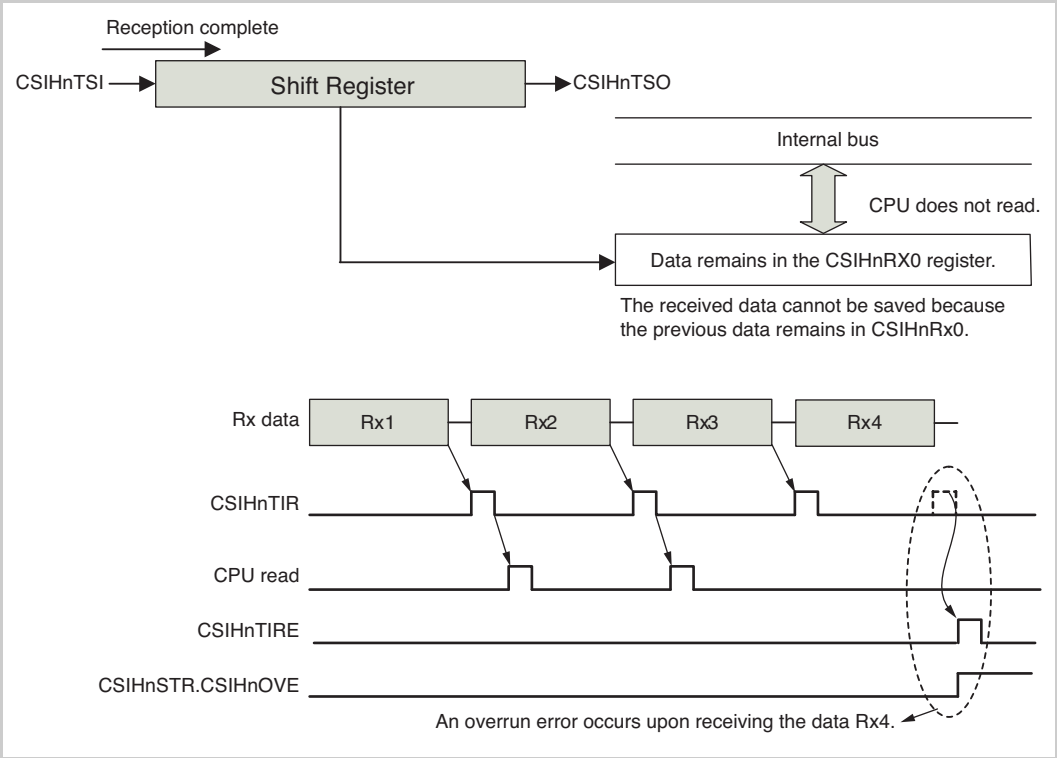


Figure 22-39 Overrun error detection in direct access and transmit-only buffer mode

FIFO mode In FIFO mode, an overrun error occurs if:

1. FIFO buffer full
   Because the FIFO buffer is full, new received data cannot be transferred from the shift register to the FIFO buffer.
2. No data
   The CPU attempts to read reception data that does not exist.

Note If the CPU attempts to read reception data that does not exist in the FIFO mode, an overrun error occurs even if data reception is disabled (CSIHnCTL0.CSIHnRXE = 0).

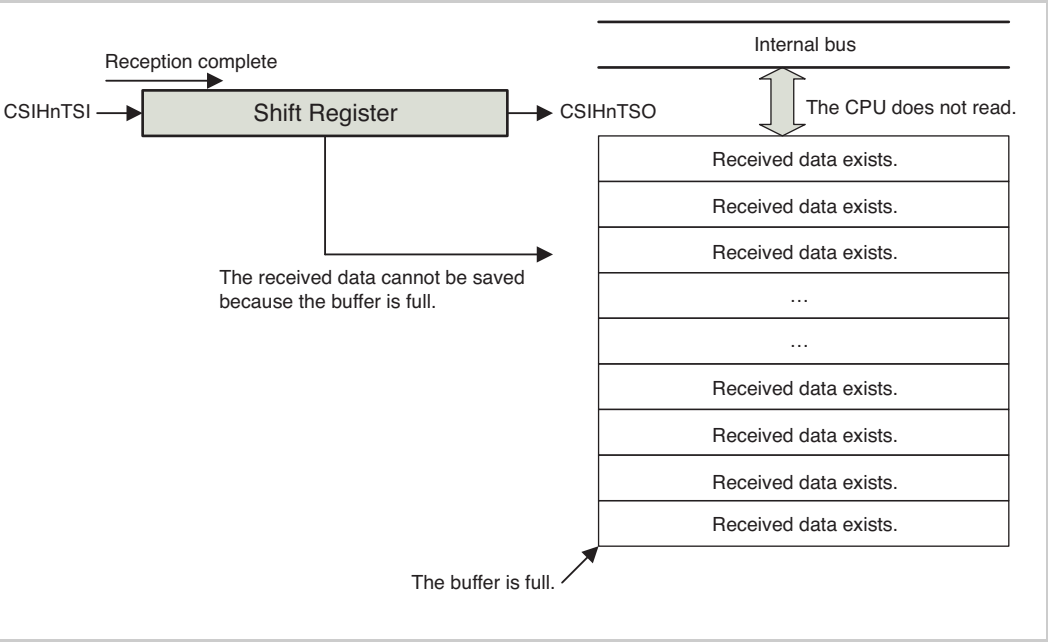


Figure 22-40 Overrun error detection in FIFO mode (FIFO full)

3. The CPU attempts to read non existing reception data

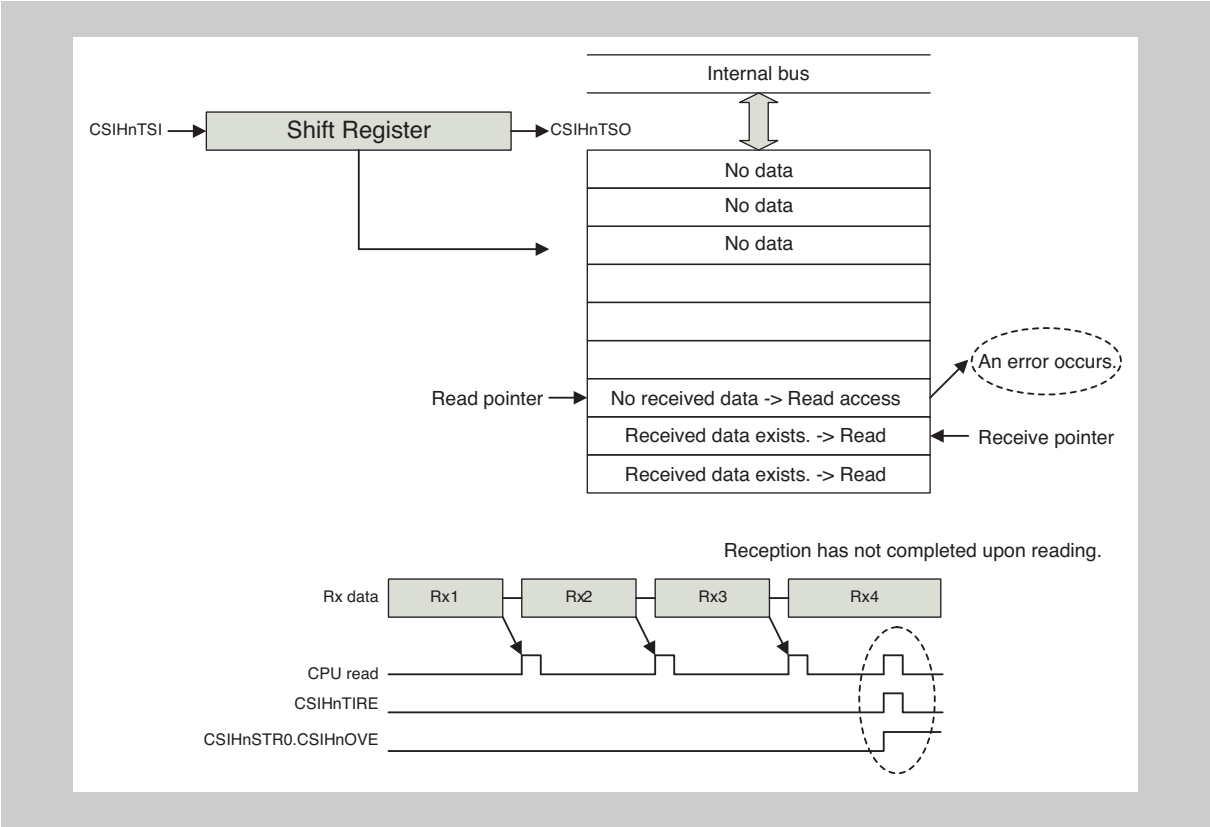


Figure 22-41 Overrun error detection in FIFO mode (no data)

When an overrun error occurs:

- Interrupt CSIHnTIRE is generated.
- Bit CSIHnSTR0.CSIHnOVE is set.
- Communication is stopped (except if the CPU tried to read non existing data).
- Communication continues (unless the CPU attempted to read data that did not exist).

Note An overrun error can be avoided in slave mode by using the handshake. When handshake is used in slave mode, the receiver (slave) signals to the transmitter (master) that it is busy. The transmitter then waits until the receiver has read its reception data register and is ready again.

For details, see *22.3.13 “Handshake function” on page 1332*.

22.3.15 Loop-back mode

Loop-back mode is a special mode for self-test. This feature is only available in master mode.

When this mode is active, the transmit and receive signals are internally connected, as shown in the figures below. The signals CSIHnTSCK, CSIHnTSO, and CSIHnTSI are disconnected from the ports. In addition, the CSIHnTSO output level is fixed to low, and CSIHnTSCK becomes inactive according to the setting of CSIHnCFGx.CSIHnCKPx. The handshake function cannot be used at this time. The rest of CSIH works as in normal operation.

The CSIHnTSCK, CSIHnTSO, CSIHnTSI, and CSIHnTCSSn[7:0] signals are disconnected from ports. The CSIHnTSO signal is fixed to the low output level, and the CSIHnTSCK and CSIHnTCSSn[7:0] signals are set to the inactive level (the level specified for the CSIHnCFGx.CSIHnCKPx bit in the case of the CSIHnTSCK signal, and the level specified for the CSIHnCTL1.CSIHnCLS[7:0] bits in the case of the CSIHnTCSSn[7:0] signal).

To perform a self-test of the CSIH, CSIHnCTL1.CSIHnLBM is set to 1, and a normal transfer operation is executed. Next, whether the reception data and transmission data are the same is checked.

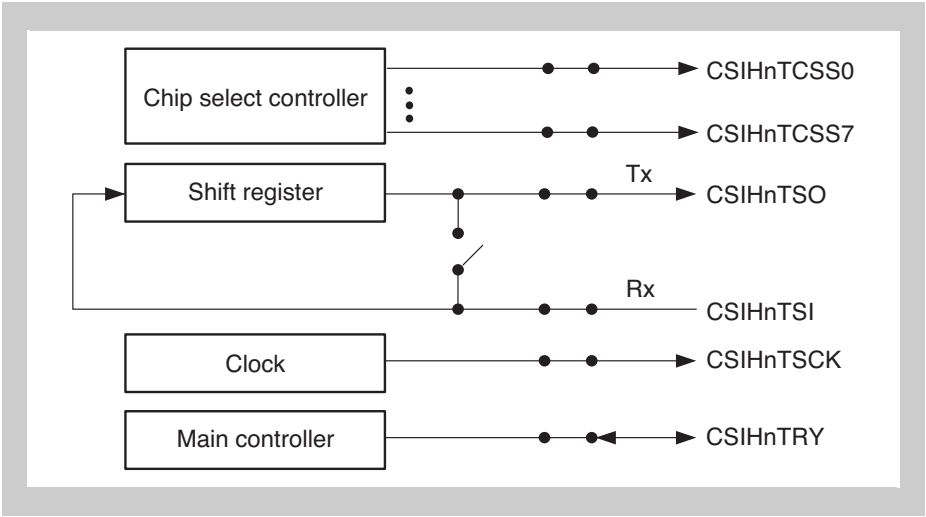


Figure 22-42 Normal operation

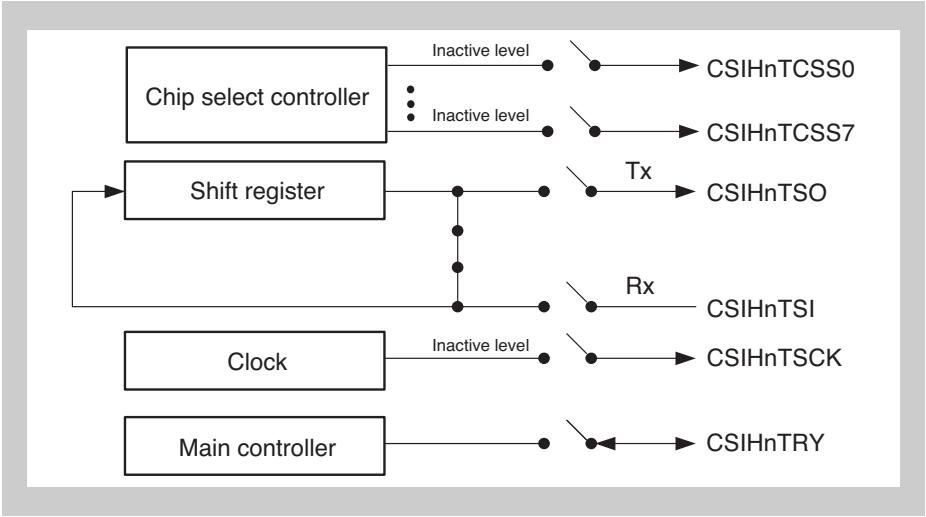


Figure 22-43 Loop-back operation

22.4 CSIH Control Registers

CSIHn is controlled and operated by means of the following registers:

Table 22-16 CSIH register overview

Register name	Shortcut	Address
Control register 0	CSIHnCTL0	<CSIHn_base_USER> + 0000_H
Control register 1	CSIHnCTL1	<CSIHn_base_OS> + 0010_H
Control register 2	CSIHnCTL2	<CSIHn_base_OS> + 0014_H
Status register 0	CSIHnSTR0	<CSIHn_base_USER> + 0004_H
Status clear register 0	CSIHnSTCR0	<CSIHn_base_USER> + 0008_H
Memory control register 0	CSIHnMCTL0	<CSIHn_base_OS> + 1040_H
Memory control register 1	CSIHnMCTL1	<CSIHn_base_USER> + 0080_H
Memory control register 2	CSIHnMCTL2	<CSIHn_base_USER> + 0084_H
Transmit data register 0 for word access	CSIHnTX0W	<CSIHn_base_USER> + 0088_H
Transmit data register 0 for half word access	CSIHnTX0H	<CSIHn_base_USER> + $008C_H$
Receive data register 0 for word access	CSIHnRX0W	<CSIHn_base_USER> + 0090_H
Receive data register 0 for half word access	CSIHnRX0H	<CSIHn_base_USER> + 0094_H
Memory read/write pointer register 0	CSIHnMRWP0	<CSIHn_base_USER> + 0098_H
Configuration register 0	CSIHnCFG0	<CSIHn_base_OS> + 1044_H
Configuration register 1	CSIHnCFG1	<CSIHn_base_OS> + 1048_H
Configuration register 2	CSIHnCFG2	<CSIHn_base_OS> + $104C_H$
Configuration register 3	CSIHnCFG3	<CSIHn_base_OS> + 1050_H
Configuration register 4	CSIHnCFG4	<CSIHn_base_OS> + 1054_H
Configuration register 5	CSIHnCFG5	<CSIHn_base_OS> + 1058_H
Configuration register 6	CSIHnCFG6	<CSIHn_base_OS> + $105C_H$
Configuration register 7	CSIHnCFG7	<CSIHn_base_OS> + 1060_H

<CSIHn_base> The base addresses <CSIHn_base> of the CSIHn is defined in the first section of this chapter under the key word “Register addresses”.

22.4.1 CSIH register details

(1) CSIHnCTL0 - CSIH control register 0

This register controls CSIHn. It mainly enables or disables the operation clock, transmission/reception, and the memory assigned to transmission/reception. It forces the stop of communication at the end of the current job.

Access This register can be read/written in 8-bit or 1-bit units.

Address <CSIHn_base_USER> + 0000_H

Initial Value 00_H. This register is initialized by any reset.

7	6	5	4	3	2	1	0
CSIHn PWR	CSIHn TXE	CSIHn RXE	0	0	0	CSIHn JOBE	CSIHn MBS
R/W	R/W	R/W	R	R	R	R/W	R/W

Table 22-17 CSIHnCTL0 register contents

Bit position	Bit name	Function
7	CSIHnPWR	Controls the operation clock. 0: Stops operation clock 1: Provides operation clock Clearing CSIHnPWR to 0 resets the internal circuits, stops operation, and sets the CSIH to standby state. No clock is provided to internal circuits, thus the power consumption of the CSIHn is minimized. If CSIHnPWR is cleared during communication, ongoing communication is immediately aborted. In this case, it is necessary to restart communication from the beginning.
6	CSIHnTXE	Enables/disables transmission. 0: Transmission disabled 1: Transmission enabled
5	CSIHnRXE	Enables/disables reception. 0: Receive disabled 1: Receive enabled
1	CSIHnJOBE	Stops the communication at the end of the current job (Communication ends when data is written to the transmission buffer while CSIHnTX0W.CSIHnEOJ = 1 (indicating that the job has ended).). 0: Communication stop is not required 1: Communication stop This bit can be used to abort an ongoing job. It is automatically cleared. Even if this bit is set, 0 is always returned when it is read. In FIFO mode, the next communication should then be started after clearing the pointers by setting CSIHnSTCR0.CSIHnPCT = 1. **Caution CSIHnJOBE is only valid when CSIHnCTL1.CSIHnJE = 1. Setting this bit is prohibited in the slave mode. When this bit is read, 0 is always returned.**
0	CSIHnMBS	Bypasses the memory for transmission and/or reception data. 0: Memory mode CSIH memory is used for transmission and/or reception data 1: Direct access mode CSIH memory is bypassed **Caution** In the slave mode, perform rewriting at the same time that CSIHnCTL0.CSIHnPWR changes from 0 to 1.

Cautions

1. When CSIHnPWR = 0, do not change the CSIHnTXE, CSIHnRXE, CSIHnJOBE, or CSIHnMBS bit. However, the CSIHnTXE, CSIHnRXE, or CSIHnMBS bit can be changed at the same time that the CSIHnPWR bit changes from 0 to 1.
2. Do not modify CSIHnTXE or CSIHnRXE or CSIHnMBS while a data transmission is pending or going on, i.e. if CSIHnSTR0.CSIHnTSF = 1.

(2) CSIHnCTL1 - CSIH control register 1

This register controls CSIHn. It mainly specifies the clock phase, interrupt timing, and interrupt delay mode, controls the extended data length, and enables or disables the data consistency check, loopback mode, handshake function, and job mode. This register also selects the active output level of each chip select signal and the chip select signal operation to perform after the last data is transferred.

Access This register can be read/written in 32-bit units.

Address <CSIHn_base_OS> + 0010_H

Initial Value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	0	0	0	0	0	CSIHn CKR	CSIHn SLIT
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/R	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSIHnCSL7[7:0]								CSIHn EDLE	CSIHn JE	CSIHn DCS	CSIHn CSRI	CSIHn LBM	CSIHn SIT	CSIHn HSE	CSIHn SSE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Caution Changing the contents of this register is only permitted when CSIHnCTL0.CSIHnPWR = 0.

Table 22-18 CSIHnCTL1 register contents (1/2)

Bit position	Bit name	Function
17	CSIHnCKR	Selects the CSIHnTSCK clock phase. 0: The default CSIHnTSCK level is the high level. 1: The default CSIHnTSCK level is the low level. **Caution** When using this bit without using the chip select function, clear CSIHnCFGx.CSIHnCKPx to 0.
16	CSIHnSLIT	Selects the timing of interrupt CSIHnTIC. 0: Normal interrupt timing (interrupt is generated after the transfer) 1: When the contents of the CSIHnTX0W or CSIHnTX0H register are transferred to the shift register, an interrupt is immediately generated. (This only functions in the direct access mode.) For details, refer to *"CSIHnTIC in direct access mode" on page 1324*.
15 to 8	CSIHnCSLx	Selects the active output level of chip select signal x (CSIHnTCSSx). 0: Chip select is active low 1: Chip select is active high For details, refer to *22.3.3 "Chip selection (CS) features" on page 1308*.
7	CSIHnEDLE	Enables/disables extended data length (EDL) mode. 0: Extended data length mode disabled 1: Extended data length mode enabled For details, refer to *2 "Data length greater than 16 bits" on page 1319*.

Table 22-18 CSIHnCTL1 register contents (2/2)

Bit position	Bit name	Function
6	CSIHnJE	Enables/disables job mode. 0: Job mode disabled 1: Job mode enabled For details, refer to *22.3.4 "Chip select timing details" on page 1310*. The CSIHnCTL0.CSIHnJOBE, CSIHnTX0W.CSIHnEOJ, and CSIHnTX0W.CSIHnCIRE bits are only valid when this bit is 1. Setting this bit is prohibited in the slave mode.
5	CSIHnDCS	Enables/disables data consistency check. 0: Data consistency check disabled 1: Data consistency check enabled For details, refer to *1 "Data consistency check" on page 1336*.
4	CSIHnCSRI	Defines chip select behavior after last data transfer. 0: Chip select holds active level 1: Chip select returns to inactive level The last data is identified at the interrupt timing while in the direct access mode or FIFO mode. The direct access mode is used while CSIHnCTL1.CSIHnSLIT = 1.
3	CSIHnLBM	Controls loop-back mode (LBM). 0: Loop-back mode deactivated 1: Loop-back mode activated For details, refer to *22.3.15 "Loop-back mode" on page 1346*. Setting this bit is prohibited in the slave mode.
2	CSIHnSIT	Selects interrupt delay mode. 0: No delay 1: Half clock delay for all interrupts This bit is only valid in master mode. In slave mode, no delay is generated. For details, refer to *"CSIHnTIC in direct access mode" on page 1324*.
1	CSIHnHSE	Enables/disables handshake mode. 0: Handshake function disabled 1: Handshake function enabled For details, refer to *22.3.13 "Handshake function" on page 1332*.
0	CSIHnSSE	Enables/disables slave select function. 0: Input signal $\overline{\text{CSIHnTSSI}}$ is ignored 1: Input signal $\overline{\text{CSIHnTSSI}}$ is recognized If the slave select function is not used, this bit must be set to 0 (see also *22.3.2 "Master/slave connections" on page 1306*).

Details about CSIHnCTL1.CSIHnSSE:

Table 22-19 Operation of the slave select function during reception

CSIHnCTL0. CSIHnRXE	CSIHnCTL1. CSIHnSSE	$\overline{\text{CSIHnTSSI}}$	Receive operation
0	–	–	Reception disabled
1	0	–	Possible
1	1	0	Possible
1	1	1	Disabled

Table 22-20 Operation of the slave select function during transmission

CSIHnCTL0. CSIHnTXE	CSIHnCTL1. CSIHnSSE	$\overline{\text{CSIHnTSSI}}$	Transmit operation
0	–	–	Transmission disabled
1	0	–	Possible
1	1	0	Possible
1	1	1	Disabled

(3) CSIHnCTL2 - CSIH control register 2

This register controls CSIHn. It selects the operating mode, prescaler, and baud rate.

For details, refer to *22.3.6 "Serial clock selection" on page 1313*.

Access This register can be read/written in 16-bit units.

Address <CSIHn_base_OS> + 0014_H

Initial Value $E000_H$. This register is initialized by any reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSIHnPRS[2:0]			0	CSIHnBRS[11:0]											
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

Caution Changing the contents of this register is only permitted when CSIHnCTL0.CSIHnPWR = 0.

Table 22-21 CSIHnCTL2 register contents

Bit position	Bit name	Function
15 to 13	CSIHnPRS [2:0]	Selects the operating mode and the value of the prescaler.
11 to 0	CSIHnBRS [11:0]	Selects the baud rate.

CSIHn PRS2	CSIHn PRS1	CSIHn PRS0	Base clock (PRSOUT) selection
0	0	0	PCLK (master mode)
0	0	1	PCLK/2 (master mode)
0	1	0	PCLK/4 (master mode)
0	1	1	PCLK/8 (master mode)
1	0	0	PCLK/16 (master mode)
1	0	1	PCLK/32 (master mode)
1	1	0	PCLK/64 (master mode)
1	1	1	External clock by way of CSIHnTSCK (in) (slave mode)

CSIHnBRS[11:0]	Baud rate at CSIHnTBLK
0	BRG is stopped
1	PCLK/(2^m × 1 × 2)
2	PCLK/(2^m × 2 × 2)
3	PCLK/(2^m × 3 × 2)
4	PCLK/(2^m × 4 × 2)
...	...
4095	PCLK/(2^m × 4095 × 2)

Note
m = 0 to 6: Value specified for CSIHnPRS[2:0]

(4) CSIHnSTR0 - CSIH status register 0

This register indicates the status of the CSIH.

Access This register can be read in 32-bit units.

Address <CSIHn_base_USER> + 0004_H

Initial Value $8000\ 0010_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CSIHnSRP[7:0]								CSIHnSPF[7:0]							
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSIHn TMOE	CSIHn OFE	0	0	0	0	0	0	CSIHn TSF	0	CSIHn FLF	CSIHn EMF	CSIHn DCE	0	CSIHn PE	CSIHn OVE
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 22-22 CSIHnSTR0 register contents (1/3)

Bit position	Bit name	Function
31 to 24	CSIHnSRP[7:0]	Indicates the number of received words in FIFO mode. **CSIHnSRP[7:0]** / **Description** 00_H / Number of received words (0 to 128_D) … / … 80_H / … Other than the above / Setting prohibited These bits are cleared by CSIHnSTCR0.CSIHnPCT. In the dual buffer mode or transmit-only buffer mode, because the number of data items is managed according to CSIHnMCTL2.CSIHnND[7:0], these bits are fixed to 00_H. They are also fixed to 00_H in the direct access mode because there is no pointer.
23 to 16	CSIHnSPF[7:0]	Indicates the number of unsent data in FIFO mode. **CSIHnSPF[7:0]** / **Description** 00_H / Number of unsent data (0 to 128_D) … / … 80_H / … Other than the above / Setting prohibited These bits are cleared by CSIHnSTCR0.CSIHnPCT. In the dual buffer mode or transmit-only buffer mode, because the number of data items is managed according to CSIHnMCTL2.CSIHnND[7:0], these bits are fixed to 00_H. They are also fixed to 00_H in the direct access mode because there is no pointer.

Table 22-22 CSIHnSTR0 register contents (2/3)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>15</td><td>CSIHnTMOE</td><td>Timeout error flag for the FIFO mode
Indicates whether a timeout error was detected in the FIFO mode.
0: No timeout error was detected in the FIFO mode.
1: A timeout error was detected in the FIFO mode.
For details, see 3 “Timeout error” on page 1339.
This bit is cleared by CSIHnSTCR0.CSIHnTMOEC.
This bit can be written to when CSIHnSTCR0.CSIHnPWR = 0.
This bit is initialized when CSIHnCTL0.CSIHnPWR changes from 0 to 1 or from 1 to 0.
If this bit is set due to a timeout error being detected and cleared by CSIHnSTCR0.CSIHnTMOEC at the same time, setting the bit is prioritized.</td></tr>
<tr><td>14</td><td>CSIHnOFE</td><td>Overflow error flag for the FIFO mode
Indicates whether an overflow error was detected in the FIFO mode.
0: No overflow error was detected in the FIFO mode.
1: An overflow error was detected in the FIFO mode.
For details, see 4 “Overflow error” on page 1341.
This bit is cleared by CSIHnSTCR0.CSIHnOFEC.
This bit can be written to when CSIHnSTCR0.CSIHnPWR = 0.
This bit is initialized when CSIHnCTL0.CSIHnPWR changes from 0 to 1 or from 1 to 0.
If 129 transmission data items are written to the CSIHnTX0W or CSIHnTX0H register when CSIHnCTL0.CSIHnPWR = 0, an overflow error occurs.
If this bit is set due to an overflow error being detected and cleared by CSIHnSTCR0.CSIHnOFEC at the same time, setting the bit is prioritized.</td></tr>
<tr><td>7</td><td>CSIHnTSF</td><td>Transfer status flag
0: Idle state
1: Transmission is in progress or being prepared
Setting and clearing of this bit is as follows:
<table>
<tr><th rowspan="2">Master mode</th><th colspan="2">Set condition</th><th rowspan="2">Clear condition</th></tr>
<tr><th>Direct access mode, FIFO mode</th><th>Dual buffer mode, transmit-only buffer mode</th></tr>
<tr><td>Transmission mode</td><td rowspan="3">Writing to transmit data register</td><td rowspan="3">Setting CSIHnMCTL2.CSIHnBTST</td><td rowspan="3">Within 0.5 clock cycles from the last CSIHnTSCK edge</td></tr>
<tr><td>Transmission/reception mode</td></tr>
<tr><td>Reception mode</td></tr>
</table>
<table>
<tr><th rowspan="2">Slave mode</th><th colspan="2">Set condition</th><th rowspan="2">Clear condition</th></tr>
<tr><th>Direct access mode, FIFO mode</th><th>Dual buffer mode, transmit-only buffer mode</th></tr>
<tr><td>Transmission mode</td><td rowspan="2">Writing to transmit data register</td><td rowspan="3">Setting CSIHnMCTL2.CSIHnBTST</td><td rowspan="3">Within 0.5 clock cycles from the last CSIHnTSCK edge</td></tr>
<tr><td>Transmission/reception mode</td></tr>
<tr><td>Reception mode</td><td>CSIHnTSCK input timing</td></tr>
</table></td></tr>
</table>

Table 22-22 CSIHnSTR0 register contents (3/3)

Bit position	Bit name	Function
5	CSIHnFLF	Indicates whether the buffer is full while in the FIFO mode. 0: FIFO buffer is not full 1: FIFO buffer is full This bit is set when the sum of the CSIHnSTR0.CSIHnSRP[7:0] bit value and CSIHnSTR0.CSIHnSPF[7:0] bit value matches 80_H, and is cleared when this sum does not match 80_H. This bit is cleared by CSIHnSTCR0.CSIHnPCT. The FIFO buffer sometimes fills up with data that has not been transmitted and reception data.
4	CSIHnEMF	Indicates whether the buffer is empty while in the FIFO mode. 0: FIFO buffer is not empty 1: FIFO buffer is empty This bit is set by SCIHnSTCR0.CSIFnPCT. This bit is set when the sum of the CSIHnSTR0.CSIHnSRP[7:0] bit value and CSIHnSTR0.CSIHnSPF[7:0] bit value matches 00_H, and is cleared when this sum does not match 00_H. The FIFO buffer sometimes does not contain any data that has not been transmitted or reception data.
3	CSIHnDCE	Data consistency error flag 0: No data consistency error detected 1: Data consistency error detected This bit is cleared by setting CSIHnSTCR0.CSIHnDCEC. This bit can be written to when CSIHnCTL0.CSIHnPWR = 0. This bit is initialized when CSIHnCTL0.CSIHnPWR changes from 0 to 1 or from 1 to 0. If this bit is set due to a data consistency error being detected and cleared by CSIHnSTCR0.CSIHnDCEC at the same time, setting the bit is prioritized.
1	CSIHnPE	Parity error flag 0: No parity error detected 1: Parity error detected This bit is cleared by setting CSIHnSTCR0.CSIHnPEC. This bit can be written to when CSIHnCTL0.CSIHnPWR = 0. This bit is initialized when CSIHnCTL0.CSIHnPWR changes from 0 to 1 or from 1 to 0. If this bit is set due to a parity error being detected and cleared by CSIHnSTCR0.CSIHnPEC at the same time, setting the bit is prioritized.
0	CSIHnOVE	Overrun error flag 0: No overrun error detected 1: Overrun error detected This bit is cleared by setting CSIHnSTCR0.CSIHnOVEC. This bit can be written to when CSIHnCTL0.CSIHnPWR = 0. This bit is initialized when CSIHnCTL0.CSIHnPWR changes from 0 to 1 or from 1 to 0. This bit is fixed to 0 in the dual buffer mode. If this bit is set due to an overrun error being detected and cleared by CSIHnSTCR0.CSIHnOVEC at the same time, setting the bit is prioritized.

Table 22-23 Memory mode operation

Bit name	Bit position	Direct access mode	FIFO mode	Transmit-only buffer mode	Dual buffer mode
CSIHnSRP[7:0]	31 to 24	Fixed to 0	Number of received data items	Fixed to 0	Fixed to 0
CSIHnSPF[7:0]	23 to 16	Fixed to 0	Number of data items that have not been transmitted	Fixed to 0	Fixed to 0
CSIHnTMOE	15	Fixed to 0	0: No error has been detected. 1: An error has been detected.	Fixed to 0	Fixed to 0
CSIHnOFE	14	Fixed to 0	0: No error has been detected. 1: An error has been detected.	Fixed to 0	Fixed to 0
CSIHnTSF	7	0: Idle state 1: Transmitting or preparing to transmit			
CSIHnFLF	5	Fixed to 0	0: Not full 1: Full	Fixed to 0	Fixed to 0
CSIHnEMF	4	Fixed to 1	0: Not empty 1: Empty	Fixed to 1	Fixed to 1
CSIHnDCE	3	0: No error has been detected. 1: An error has been detected.			
CSIHnPE	1	0: No error has been detected. 1: An error has been detected.			
CSIHnOVE	0	0: No error has been detected. 1: An error has been detected.			Fixed to 0

(5) CSIHnSTCR0 - CSIH status clear register 0

This register clears the status flags of the CSIHnSTR0 status register.

Access This register can be written in 16-bit units.

When read, the value 0000_H is always returned.

Address <CSIHn_base_USER> + 0008_H

Initial Value 0000_H. This register is initialized by any reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSIHn TMOEC	CSIHn OFEC	0	0	0	0	0	CSIHn PCT	0	0	0	0	CSIHn DCEC	0	CSIHn PEC	CSIHn OVEC
W	W	R	R	R	R	R	W	R	R	R	R	W	R	W	W

Table 22-24 CSIHnSTCR0 register contents

Bit position	Bit name	Function
15	CSIHnTMOEC	Timeout error flag clear command 0: No operation. Read value is always 0. 1: Clear time out error flag (CSIHnSTR0.CSIHnTMOE)
14	CSIHnOFEC	Overflow error flag clear command 0: No operation. Read value is always 0. 1: Clear overflow error flag (CSIHnSTR0.CSIHnOFE)
8	CSIHnPCT	Controls the FIFO buffer pointers. 0: No operation. Read value is always 0. 1: In the dual buffer mode, transmit-only buffer mode, or FIFO mode, clear all the following FIFO buffer pointers: - CSIHnMRWP0.CSIHnTRWA[6:0] - CSIHnMRWP0.CSIHnRRA[6:0] - CSIHnMCTL2.CSIHnSOP[6:0] In only the FIFO mode, also clear all the following status bits: - CSIHnSTR0.CSIHnSPF[7:0] - CSIHnSTR0.CSIHnSRP[7:0] - CSIHnSTR0.CSIHnFLF - CSIHnSTR0.CSIHnTSF Note that CSIHnSTR0.CSIHnEMF is set (indicating an empty FIFO buffer). **Caution** When this bit is set during communication, the communication stops.
3	CSIHnDCEC	Data consistency error flag clear command 0: No operation. Read value is always 0. 1: Clear data consistency error flag (CSIHnSTR0.CSIHnDCE)
1	CSIHnPEC	Parity error flag clear command 0: No operation. Read value is always 0. 1: Clear parity error flag (CSIHnSTR0.CSIHnPE)
0	CSIHnOVEC	Overrun error flag clear command 0: No operation. Read value is always 0. 1: Clear overrun error flag (CSIHnSTR0.CSIHnOVE)

(6) CSIHnMCTL0 - CSIH Memory control register 0

This register selects the memory mode and the timeout setting.

Access This register can be read/written in 16-bit units.

Address <CSIHn_base_OS> + 1040_H

Initial Value $001F_H$. This register is initialized by any reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	CSIHn MMS[1:0]		0	0	0	CSIHnTO[4:0]				
R	R	R	R	R	R	R/W	R/W	R	R	R	R/W	R/W	R/W	R/W	R/W

Table 22-25 CSIHnMCTL0 register contents

Bit position	Bit name	Function
9 to 8	CSIHn MMS[1:0]	Selects the memory mode. **CSIHn MMS1 / CSIHn MMS0 / Description** 0 / 0 / FIFO mode 0 / 1 / Dual buffer mode 1 / 0 / Transmit-only buffer mode 1 / 1 / Prohibited After changing the memory mode, set the CSIHnSTCR0.CSIHnPCT bit and clear the individual buffer pointers and other data. **Caution** The memory mode can only be changed when CSIHnCTL0.CSIHnPWR and CSIHnCTL0.CSIHnMBS = 0.
4 to 0	CSIHn TO[4:0]	Select the FIFO mode timeout setting. **CSIHnTO[4:0] / Description** 00000_B / No timeout detection 00001_B / Timeout is (1 x 8 x BRG output clocks) 00010_B / Timeout is (2 x 8 x BRG output clocks) ... / ... 11111_B / Timeout is (31 x 8 x BRG output clocks) **Caution** The timeout setting can only be changed when CSIHnCTL0.CSIHnPWR = 0. Clear these bits to 00000_B when in the master mode or a memory mode other than the FIFO mode (the direct access mode, dual buffer mode, or transmission mode). For details about timeout detection, see also *3 "Timeout error" on page 1339*.

(7) CSIHnMCTL1 - CSIH Memory control register 1

This register selects the conditions to generate the interrupt requests CSIHnTIC and CSIHnTIR in FIFO mode.

Access This register can be read/written in 32-bit units.

Address <CSIHn_base_USER> + 0080_H

Initial Value 0000 $007F_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	CSIHnFES[6:0]						
R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	CSIHnFFS[6:0]						
R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note This register can be written to during communication.

Table 22-26 CSIHnMCTL1 register contents

Bit position	Bit name	Function
22 to 16	CSIHnFES[6:0]	Select the condition for generating the CSIHnTIC interrupt (no transmission data). When the number of transmission data items in the FIFO buffer that have not been transmitted (checked by using the CSIHnSTR0.CSIHnSPF[7:0] bits) matches CSIHnMCTL1.CSIHnFES[6:0], a CSIHnTIC interrupt request is generated.
6 to 0	CSIHnFFS[6:0]	Select the condition for generating the CSIHnTIR interrupt (a full reception buffer). When the number of received data items in the FIFO buffer (checked by using the CSIHnSTR0.CSIHnSRP[7:0] bits) matches (128 – CSIHnMCTL1.CSIHnFFS[6:0]), a CSIHnTIR interrupt request is generated.

(8) CSIHnMCTL2 - CSIH Memory control register 2

This register controls memory operations while in the dual buffer mode or transmit-only buffer mode and generates triggers to start communication.

Access This register can be read/written in 32-bit units.

Address <CSIHn_base_USER> + 0084_H

Initial Value $0000\ 0000_H$. This register is initialized by any reset.

Cautions

1. Writing to this register is prohibited when CSIHnSTR0.CSIHnTSF = 1 (during a transfer).
2. Writing to the CSIHnMCTL2 register is prohibited in the following cases:
   - When CSIHnCTL0.CSIHnPWR = 0
   - When CSIHnCTL0.CSIHnTXE = CSIHnCTL0.CSIHnRXE = 0
   - When in the direct access mode or FIFO mode

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CSIHnBTST	0	0	0	0	0	0	0	CSIHnND[7:0]							
R/W	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	CSIHnSOP[6:0]						
R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 22-27 CSIHnMCTL2 register contents

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>31</td><td>CSIHnBTST</td><td>Provides a start trigger for buffer transfer.
0: No operation
1: Start transfer command
The read value is always 0.
Caution
This bit can only be used in dual buffer mode and transmit-only buffer mode.
In the direct access mode and FIFO mode, this bit is disabled.</td></tr>
<tr><td>23 to 16</td><td>CSIHnND[7:0]</td><td>Specify the number of data items.
When read, these bits indicate the number of remaining communication data items.
<table>
<tr><th>CSIHn ND[7:0]</th><th>Dual buffer mode</th><th>Transmit-only buffer mode</th><th>FIFO mode</th><th>Direct access mode</th></tr>
<tr><td>00_H</td><td>Transmit 0 data items.</td><td>Transmit 0 data items.</td><td>No effect</td><td>No effect</td></tr>
<tr><td>01_H</td><td>Transmit 1 data item.</td><td>Transmit 1 data item.</td><td>No effect</td><td>No effect</td></tr>
<tr><td>...</td><td>...</td><td>...</td><td>No effect</td><td>No effect</td></tr>
<tr><td>$3F_H$</td><td>Transmit 63 data items.</td><td>Transmit 63 data items.</td><td>No effect</td><td>No effect</td></tr>
<tr><td>40_H</td><td>Transmit 64 data items.</td><td>Transmit 64 data items.</td><td>No effect</td><td>No effect</td></tr>
<tr><td>...</td><td>Prohibited</td><td>...</td><td>No effect</td><td>No effect</td></tr>
<tr><td>$7F_H$</td><td>Prohibited</td><td>Transmit 127 data items.</td><td>No effect</td><td>No effect</td></tr>
<tr><td>80_H</td><td>Prohibited</td><td>Transmit 128 data items.</td><td>No effect</td><td>No effect</td></tr>
<tr><td>Other than the above</td><td colspan="4">Setting prohibited</td></tr>
</table>
The value of these bits is automatically decremented after transferring the data.
During a transfer, the number of remaining data items can be read from these bits.
The value of these bits is not decremented while in the direct access mode.</td></tr>
<tr><td>6 to 0</td><td>CSIHn SOP[6:0]</td><td>Select the transmission data pointer.
<table>
<tr><th>CSIHn SOP[6:0]</th><th>Dual buffer mode</th><th>Transmit-only buffer mode</th><th>FIFO mode</th><th>Direct access mode</th></tr>
<tr><td>00_H</td><td>0000_H</td><td>0000_H</td><td>0000_H</td><td>No effect</td></tr>
<tr><td>01_H</td><td>0004_H</td><td>0004_H</td><td>0004_H</td><td>No effect</td></tr>
<tr><td>...</td><td>...</td><td>...</td><td>...</td><td>No effect</td></tr>
<tr><td>$3F_H$</td><td>$00FC_H$</td><td>$00FC_H$</td><td>$00FC_H$</td><td>No effect</td></tr>
<tr><td>40_H</td><td>Prohibited</td><td>0100_H</td><td>0100_H</td><td>No effect</td></tr>
<tr><td>...</td><td>Prohibited</td><td>...</td><td>...</td><td>No effect</td></tr>
<tr><td>$7F_H$</td><td>Prohibited</td><td>$01FC_H$</td><td>$01FC_H$</td><td>No effect</td></tr>
</table>
When CSIHnCTL0.CSIHnPWR = 0 or CSIHnSTR0.CSIHnPCT is set to forcibly stop communication, these bits are cleared by the hardware.
Note
In the FIFO mode, these bits indicate the transmission address. The value of these bits is not decremented while in the direct access mode.</td></tr>
</table>

(9) CSIHnMRWP0 - CSIH memory read/write pointer register 0

This register sets the pointers for reading from and writing to the dual or transmit-only buffer.

Access This register can be read/written in 32-bit units.

Address <CSIHn_base_USER> + 0098_H

Initial Value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	0	CSIHnRRA[6:0]						
R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	CSIHnTRWA[6:0]						
R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Caution This register can be written to during communication.
Writing to this register in the direct access mode or FIFO mode is prohibited.

Table 22-28 CSIHnMRWP0 register contents (1/2)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>22 to 16</td><td>CSIHn RRA[6:0]</td><td>Selects the read pointer of the Rx buffer.
<table>
<tr><th>CSIHn RRA[6:0]</th><th>Dual buffer mode</th><th>Transmit-only buffer mode</th><th>FIFO mode</th><th>Direct access mode</th></tr>
<tr><td>00_H</td><td>0000_H</td><td>0000_H</td><td>0000_H</td><td>No effect</td></tr>
<tr><td>01_H</td><td>0004_H</td><td>0004_H</td><td>0004_H</td><td>No effect</td></tr>
<tr><td>...</td><td>...</td><td>...</td><td>...</td><td>No effect</td></tr>
<tr><td>$3F_H$</td><td>$00FC_H$</td><td>$00FC_H$</td><td>$00FC_H$</td><td>No effect</td></tr>
<tr><td>40_H</td><td>Prohibited</td><td>0100_H</td><td>0100_H</td><td>No effect</td></tr>
<tr><td>...</td><td>Prohibited</td><td>...</td><td>...</td><td>No effect</td></tr>
<tr><td>$7F_H$</td><td>Prohibited</td><td>$01FC_H$</td><td>$01FC_H$</td><td>No effect</td></tr>
</table>
These bits are automatically incremented when reception data is read.
If an overrun error occurs while reading the CSIHnRX0W or CSIHnRX0H register (when the CPU reads the CSIHnRX0W or CSIHnRX0H register while there is no data), the read pointer is not incremented.
These bits are cleared when CSIHnSTCR0.CSIHnPCT is set.
These bits are not incremented in the direct access mode or transmit-only buffer mode.
When writing in the transmit-only buffer mode, clear these bits to 0000_H.
In the FIFO mode, these bits indicate the read address of the reception data.</td></tr>
</table>

Table 22-28 CSIHnMRWP0 register contents (2/2)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>6 to 0</td><td>CSIHn TRWA[6:0]</td><td>Selects the read/write pointer of the Tx buffer.
<table>
<tr><th>CSIHn TRWA[6:0]</th><th>Dual buffer mode</th><th>Transmit-only buffer mode</th><th>FIFO mode</th><th>Direct access mode</th></tr>
<tr><td>00_H</td><td>0000_H</td><td>0000_H</td><td>0000_H</td><td>No effect</td></tr>
<tr><td>01_H</td><td>0004_H</td><td>0004_H</td><td>0004_H</td><td>No effect</td></tr>
<tr><td>...</td><td>...</td><td>...</td><td>...</td><td>No effect</td></tr>
<tr><td>$3F_H$</td><td>$00FC_H$</td><td>$00FC_H$</td><td>$00FC_H$</td><td>No effect</td></tr>
<tr><td>40_H</td><td>Prohibited</td><td>0100_H</td><td>0100_H</td><td>No effect</td></tr>
<tr><td>...</td><td>Prohibited</td><td>...</td><td>...</td><td>No effect</td></tr>
<tr><td>$7F_H$</td><td>Prohibited</td><td>$01FC_H$</td><td>$01FC_H$</td><td>No effect</td></tr>
</table>
When transmission data is read or written from the CPU, these bits are automatically incremented.
These bits are cleared when CSIHnSTCR0.CSIHnPCT is set.
In the direct access mode, these bits are not incremented.
In the FIFO mode, these bits indicate the read/write address of the transmission data.</td></tr>
</table>

(10) CSIHnCFGx - CSIH configuration register x

These eight registers specify for each chip select signal CSIHnTCSSx prescaler, parity, data length, recessive configuration for broadcasting, serial data direction, clock phase and data phase, idle enforcement configuration, idle timing, hold timing, inter-data time, and setup timing.

Slave mode In slave mode the transmission protocol setting of the CSIHnCFG0 register are relevant:

- CSIHnPS0: Parity usage
- CSIHnDLS0: Data length selection
- CSIHnDIR0: Data direction
- CSIHnCKP0, CSIHnDAP0: Clock phase and data phase

In the slave mode, clear the CSIHnCFG0 register bits other than the above and the CSIHnCFG1 to CSIHnCFG7 registers to 0.

Access This register can be read/written in 32-bit units.

Address CSIHnCFG0: <CSIHn_base_OS> + 1044_H
CSIHnCFG1: <CSIHn_base_OS> + 1048_H
CSIHnCFG2: <CSIHn_base_OS> + $104C_H$
CSIHnCFG3: <CSIHn_base_OS> + 1050_H
CSIHnCFG4: <CSIHn_base_OS> + 1054_H
CSIHnCFG5: <CSIHn_base_OS> + 1058_H
CSIHnCFG6: <CSIHn_base_OS> + $105C_H$
CSIHnCFG7: <CSIHn_base_OS> + 1060_H

Initial Value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CSIHn PSCLx[1:0]		CSIHn PSx[1:0]		CSIHnDLSx[3:0]				0	0	0	0	CSIHn RCBx	CSIHn DIRx	CSIHn CKPx	CSIHn DAPx
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSIHn IDLx	CSIHnIDx[2:0]			CSIHnHDx[3:0]				CSIHnINx[3:0]				CSIHnSPx[3:0]			
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

Caution Writing is only possible when CSIHnCTL0.CSIHnPWR = 0 (except when writing the same value, which is possible even when CSIHnCTL0.CSIHnPWR = 1).

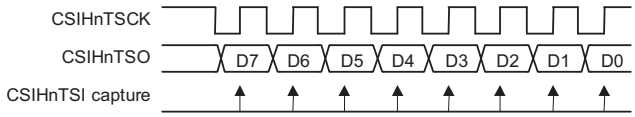
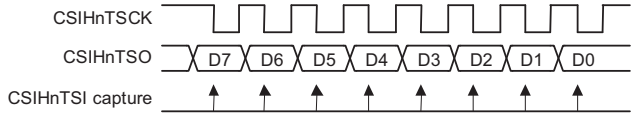
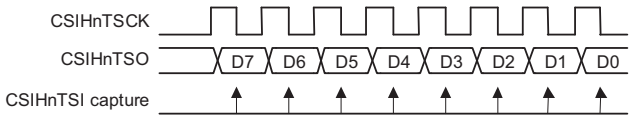
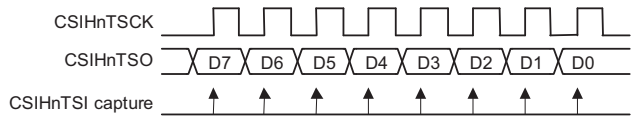
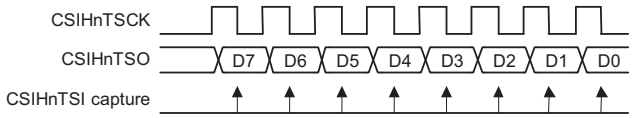
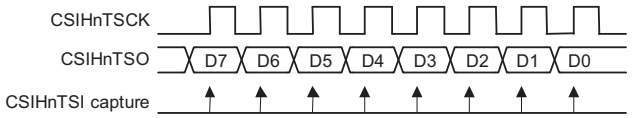
Table 22-29 CSIHnCFGx register contents (1/4)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>31 and 30</td><td>CSIHn PSCLx[1:0]</td><td>Selects the prescaler for chip select x.
<table>
<tr><th>CSIHn PSCLx1</th><th>CSIHn PSCLx0</th><th>Prescaler output</th></tr>
<tr><td>0</td><td>0</td><td>CSIHnBPCLK</td></tr>
<tr><td>0</td><td>1</td><td>CSIHnBPCLK / 2</td></tr>
<tr><td>1</td><td>0</td><td>CSIHnBPCLK / 4</td></tr>
<tr><td>1</td><td>1</td><td>CSIHnBPCLK / 8</td></tr>
</table>
These bits are only available in master mode.
For details about CSIHnBPCLK, see 22.3.6 “Serial clock selection” on page 1313.</td></tr>
<tr><td>29 and 28</td><td>CSIHn PSx[1:0]</td><td>Selects the parity for chip select x for transmission and reception.
<table>
<tr><th>CSIHn PSx1</th><th>CSIHn PSx0</th><th>Transmission</th><th>Reception</th></tr>
<tr><td>0</td><td>0</td><td>No parity transmitted</td><td>Parity reception is not expected.</td></tr>
<tr><td>0</td><td>1</td><td>Add parity bit fixed at 0</td><td>Parity bit reception is expected, but parity judgment is not performed.</td></tr>
<tr><td>1</td><td>0</td><td>Add odd parity</td><td>Odd parity bit reception is expected.</td></tr>
<tr><td>1</td><td>1</td><td>Add even parity</td><td>Even parity bit reception is expected.</td></tr>
</table></td></tr>
<tr><td>27 to 24</td><td>CSIHn DLSx[3:0]</td><td>Selects the data length for chip select x.
<table>
<tr><th>CSIHn DLSx[3:0]</th><th>Data length</th></tr>
<tr><td>0000_B</td><td>16 bits</td></tr>
<tr><td>0001_B</td><td>1 bit</td></tr>
<tr><td>0010_B</td><td>2 bits</td></tr>
<tr><td>...</td><td>...</td></tr>
<tr><td>1111_B</td><td>15 bits</td></tr>
</table>
Note
For details about the CSIHnDLSx[3:0] bit setting, see 22.3.9 “Data length selection”.
For the CSIHnDLSx[3:0] bits, 0001_B (1 bit) to 0110_B (6 bits) can be specified only when the data length is 16 bits or more.</td></tr>
<tr><td>19</td><td>CSIHn RCBx</td><td>Selects the recessive configuration for broadcasting for chip select x.
0: Dominant (higher priority)
1: Recessive (lower priority)
For details, see 1 “Configuration registers” on page 1308.</td></tr>
<tr><td>18</td><td>CSIHnDIRx</td><td>Selects the serial data direction for chip select x.
0: Data is sent/received with MSB first
1: Data is sent/received with LSB first
For details, see 22.3.10 “Serial data direction selection” on page 1321</td></tr>
</table>

Table 22-29 CSIHnCFGx register contents (2/4)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>17 and 16</td><td>CSIHnCKPx

CSIHnDAPx</td><td>CKP: Clock phase select bit
DAP: Data phase select bit

CSIHnCTL1.CSIHnCKR = 0
<table>
<tr><th>CSIHn CKPx</th><th>CSIHn DAPx</th><th>Clock phase and data phase selection</th></tr>
<tr><td>0</td><td>0</td><td> 
</td></tr>
<tr><td>0</td><td>1</td><td> 
</td></tr>
<tr><td>1</td><td>0</td><td> 
</td></tr>
<tr><td>1</td><td>1</td><td> 
</td></tr>
</table>
CSIHnCTL1.CSIHnCKR = 1
<table>
<tr><th>CSIHn CKPx</th><th>CSIHn DAPx</th><th>Clock phase and data phase selection</th></tr>
<tr><td>0</td><td>0</td><td> 
</td></tr>
<tr><td>0</td><td>1</td><td> 
</td></tr>
<tr><td>1</td><td>x</td><td>Setting prohibited</td></tr>
</table>
Caution
When not using the chip select function, fix the CSIHnCKPx bit to 0, and use the CSIHnCTL1.CSIHnCKR bit to specify the clock phase.</td></tr>
<tr><td>15</td><td>CSIHnIDLx</td><td>Selects the idle enforcement configuration for chip select x.
0: If the chip select value did not change, the chip select signal stays active. If a different chip select value is defined, chip select signal x becomes idle.
1: An idle state is inserted after every transfer to chip select x.

This bit is only available in master mode.
If CSIHnCTL1.CSIHnJE = 1 and CSIHnTX0W.CSIHnEOJ = 1, chip select signal x definitely becomes idle even if CSIHnCFG0-7.CSIHnIDLn is cleared to 0.
For details about the idle state, see Figure 22-6 “Chip select timings” on page 1309.</td></tr>
</table>

Table 22-29 CSIHnCFGx register contents (3/4)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>14 to 12</td><td>CSIHn IDx[2:0]</td><td>Selects the idle time for chip select x.
<table>
<tr><th>CSIHn DLSx[3:0]</th><th>Idle timing</th></tr>
<tr><td>000_B</td><td>0.5 transmission clock cycles</td></tr>
<tr><td>001_B</td><td>1.0 transmission clock cycle</td></tr>
<tr><td>010_B</td><td>1.5 transmission clock cycles</td></tr>
<tr><td>...</td><td>... (2.5, 3.5, 4.5, 6.5)</td></tr>
<tr><td>111_B</td><td>8.5 transmission clock cycles</td></tr>
</table>
These bits are only available in master mode.</td></tr>
<tr><td>11 to 8</td><td>CSIHn HDx[3:0]</td><td>Selects the hold time for chip select x in transmission clock cycles.
<table>
<tr><th>CSIHn INx[3:0]</th><th>Hold timing with CSIHnCTL1.CSIHnSIT = 0</th><th>Hold timing with CSIHnCTL1.CSIHnSIT = 1</th></tr>
<tr><td>0000_B</td><td>0.5 serial clock cycles</td><td>1.0 serial clock cycles</td></tr>
<tr><td>0001_B</td><td>1.0 serial clock cycle</td><td>1.5 serial clock cycles</td></tr>
<tr><td>0010_B</td><td>1.5 serial clock cycles</td><td>2.0 serial clock cycles</td></tr>
<tr><td>...</td><td>... (2.5, 3.5, 4.5, 6.5, 8.5, 9.5 10.5, 11.5, 12.5, 14.5, 16.5, 18.5)</td><td>... (3.0, 4.0, 5.0, 7.0, 9.0, 10.0, 11.0, 12.0, 13.0, 15.0, 17.0, 19.0)</td></tr>
<tr><td>1111_B</td><td>20.5 serial clock cycles</td><td>21.0 serial clock cycles</td></tr>
</table>
These bits are only available in master mode.</td></tr>
</table>

Table 22-29 CSIHnCFGx register contents (4/4)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>7 to 4</td><td>CSIHn INx[3:0]</td><td>Selects the inter-data time for chip select x in transmission clock cycles.
<table>
<tr><th>CSIHn INx[3:0]</th><th>Inter-data time when CSIHnCTL1.CSIHnSIT = 0</th><th>Inter-data time when CSIHnCTL1.CSIHnSIT = 1</th></tr>
<tr><td>0000_B</td><td>0.0 serial clock cycles</td><td>0.5 serial clock cycles</td></tr>
<tr><td>0001_B</td><td>0.5 serial clock cycles</td><td>1.0 serial clock cycles</td></tr>
<tr><td>0010_B</td><td>1.0 serial clock cycles</td><td>1.5 serial clock cycles</td></tr>
<tr><td>0011_B</td><td>2.0 serial clock cycles</td><td>2.5 serial clock cycles</td></tr>
<tr><td>...</td><td>... (3.0, 4.0, 6.0, 8.0, 9.0, 10.0, 11.0, 12.0, 14.0, 16.0, 18.0)</td><td>... (3.5, 4.5, 6.5, 8.5, 9.5, 10.5, 11.5 12.5, 14.5, 16.5, 18.5)</td></tr>
<tr><td>1111_B</td><td>20.0 serial clock cycles</td><td>20.5 serial clock cycles</td></tr>
</table>
These bits are only available in master mode.</td></tr>
<tr><td>3 to 0</td><td>CSIHn SPx[3:0]</td><td>Selects the setup time for chip select x in transmission clock cycles.
<table>
<tr><th>CSIHn SPx[3:0]</th><th>Setup delay</th></tr>
<tr><td>0000_B</td><td>0.5 serial clock cycles</td></tr>
<tr><td>0001_B</td><td>1.0 serial clock cycles</td></tr>
<tr><td>0010_B</td><td>1.5 serial clock cycles</td></tr>
<tr><td>...</td><td>... (2.5, 3.5, 4.5, 6.5, 8.5, 9.5 10.5, 11.5, 12.5, 14.5, 16.5, 18.5)</td></tr>
<tr><td>1111_B</td><td>20.5 serial clock cycles</td></tr>
</table>
These bits are only available in master mode.</td></tr>
</table>

(11) CSIHnTX0W - CSIH transmit data register 0 for word access

This register stores the transmission data. In addition, it specifies the communication interrupt request, the end-of-job, the extended data length, and the chip select activation.

Access This register can be read/written in 32-bit units.

Address <CSIHn_base_USER> + 0088_H

Initial Value Undefined

Cautions

1. Reading this register is prohibited during communication in the FIFO mode.
2. Reading from and writing to this register are prohibited in the FIFO mode when CSIHnCTL0.CSIHnPWR = 0.
3. Writing to this register is prohibited in the direct access mode when CSIHnCTL0.CSIHnTXE = CSIHnCTL0.CSIHnRXE = 0.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
CSIHnCIRE	CSIHnEOJ	CSIHnEDL	0	0	0	0	0	CSIHnCS[7:0]							
R/W	R/W	R/W	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSIHnTX[15:0]															
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

Table 22-30 CSIHnTX0W register contents (1/2)

Bit position	Bit name	Function
31	CSIHnCIRE	Enables the communication interrupt request CSIHnTIC in dual or transmit-only buffer mode or the job completion interrupt CSIHnTJC request in FIFO mode. 0: No interrupt requested 1: Interrupt requested. Generates interrupt CSIHnTIC or CSIHnTJC after transmission. For details, see *"CSIHnTIC in direct access mode" on page 1324* and *4 "CSIHnTIJC job completion interrupt" on page 1331*. **Caution** This bit is only valid when the job mode is enabled (CSIHnCTL1.CSIHnJE = 1).
30	CSIHnEOJ	Specifies the end of a job. 0: No end-of-job data 1: End-of-job data **Caution** This bit is only valid when the job mode is enabled (CSIHnCTL1.CSIHnJE = 1). For use in the slave mode, clear this bit.

Table 22-30 CSIHnTX0W register contents (2/2)

Bit position	Bit name	Function
29	CSIHnEDL	Specifies whether the associated data requires the extended data length (EDL) option. 0: Normal operation 1: Extended data length activated The associated data is transmitted as of 16 bits. The inter-data delay time and idle time are not inserted after data transmission. When CSIHnCTL1.CSIHnEDLE = 1 and CSIHnTX0W.CSIHnEDL = 1, the same CS must also be selected for the second data. If the CS for the second data is changed, the correct operation is not guaranteed. **Caution** This bit can only be used when CSIHnCTL1.CSIHnEDLE = 1.
23 to 16	CSIHnCSx	Activates one or several chip select signals. 0: Chip select x is activated for the associated transmission 1: Chip select x is deactivated for the associated transmission Setting CSIHnTX0W.CSIHnCS[7:0] to FF_H is prohibited. **Caution** If several chip select signals are enabled for broadcasting, the configuration of one with CSIHnCFGx.CSIHnRCBx = 0 (dominant) is used. In this case, all dominant chip selects must be set to precisely the same configuration. For use in the slave mode, set the CSIHnCS[7:0] bits to FE_H.
15 to 0	CSIHnTX[15:0]	Stores the transmission data.

(12) CSIHnTX0H - CSIH transmit data register 0 for half word access

This register stores the transmission data. This register is the same as bits 15 to 0 of register CSIHnTX0W.

Access This register can be read/written in 16-bit units.

Address <CSIHn_base_USER> + $008C_H$

Initial Value Undefined

Cautions

1. Reading this register is prohibited during communication in the FIFO mode.
2. Reading from and writing to this register are prohibited in the FIFO mode when CSIHnCTL0.CSIHnPWR = 0.
3. Writing to this register is prohibited in the direct access mode when CSIHnCTL0.CSIHnTXE = CSIHnCTL0.CSIHnRXE = 0.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSIHnTX[15:0]															
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

Table 22-31 CSIHnTX0H register contents

Bit position	Bit name	Function
15 to 0	CSIHnTX[15:0]	Stores the transmission data.

(13) CSIHnRX0W - CSIH receive data register 0 for word access

This register stores the received data.

Access This register is read-only, in 32-bit units.

Address <CSIHn_base_USER> + 0090_H

Initial Value Undefined

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	CSIHnRPE	CSIHnTDCE	CSIHnCS[7:0]							
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSIHnRX[15:0]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Cautions

1. This register can be read when CSIHnCTL0.CSIHnPWR = 1, and can be written to when CSIHnCTL0.CSIHnPWR = 0.
2. This register is Initialized when CSIHnCTL0.CSIHnPWR changes from 0 to 1 or from 1 to 0.
3. Reading from and writing to this register are prohibited in the FIFO mode when CSIHnCTL0.CSIHnPWR = 0.

Table 22-32 CSIHnRX0W register contents

Bit position	Bit name	Function
25	CSIHnRPE	Indicates whether a reception data parity error was detected. 0: No parity error has been detected in the received data. 1: A parity error has been detected in the received data.
24	CSIHnTDCE	Indicates whether a transmission data consistency error was detected. A data consistency check is performed on transmission data. The result of the check performed on the data transmitted at the same time as saving received data to CSIHnRX0W.CSIHnRX[15:0] is applied to this bit. 0: No data consistency error has been detected in the transmitted data. 1: A data consistency error has been detected in the transmitted data.
23 to 16	CSIHnCSx	Indicate whether the chip select signal is active. When in the master mode, the status of the chip select signal upon receiving the data saved to CSIHnRX0W.CSIHnRX[15:0] (that is, which CS to perform communication for) is stored in these bits. 0: Chip select signal x was active upon receiving the data. 1: Chip select signal x was inactive upon receiving the data. When in the slave mode, because it is necessary to specify CS0 (CSIHnTX0W.CSIHnCS[7:0] = FE_H) as the communication partner when transmission is enabled, FE_H is saved in the transmission mode or transmission/reception mode. The value is always 00_H when in the reception mode.
15 to 0	CSIHnRX[15:0]	Store the reception data. Read the value of the CSIHnRX0W or CSIHnRX0H register at least one serial clock cycle before the interrupt is generated.

(14) CSIHnRX0H - CSIH receive data register 0 for half word access

This register stores the reception data. This register is the same as bits 15 to 0 of register CSIHnTX0W.

Access This register is read-only, in 16-bit units.

Address <CSIHn_base_USER> + 0094_H

Initial Value Undefined

Cautions

1. This register can be read when CSIHnCTL0.CSIHnPWR = 1, and can be written to when CSIHnCTL0.CSIHnPWR = 0.
2. This register is initialized when CSIHnCTL0.CSIHnPWR is changed from 0 to 1 or 1 to 0.
3. Reading from and writing to this register are prohibited in the FIFO mode when CSIHnCTL0.CSIHnPWR = 0.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSIHnRX[15:0]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 22-33 CSIHnRX0H register contents

Bit position	Bit name	Function
15 to 0	CSIHnRX[15:0]	Stores the reception data.

22.5 Operating Procedures

The following examples and instructions are sorted according to the memory mode:

- Direct access
- Transmit-only buffer
- Dual buffer
- FIFO

22.5.1 Procedures in direct access mode

(1) For transmission/reception in the master mode, and when the job mode is disabled

The following conditions are assumed for the procedure shown here:

- Transmission data length: 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B)
- Transmission direction: MSB first (CSIHnCFGx.CSIHnDIRx = 0)
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No delay for any interrupt (CSIHnCTL1.CSIHnSIT = 0)
- Job mode disabled (CSIHnCTL1.CSIHnJE = 0)
- A CSIHnTIC interrupt is generated when transferring starts. (CSIHnCTL1.CSIHnCLIT = 1)
- Direct access mode (CSIHnCTL0.CSIHnMBS = 0)

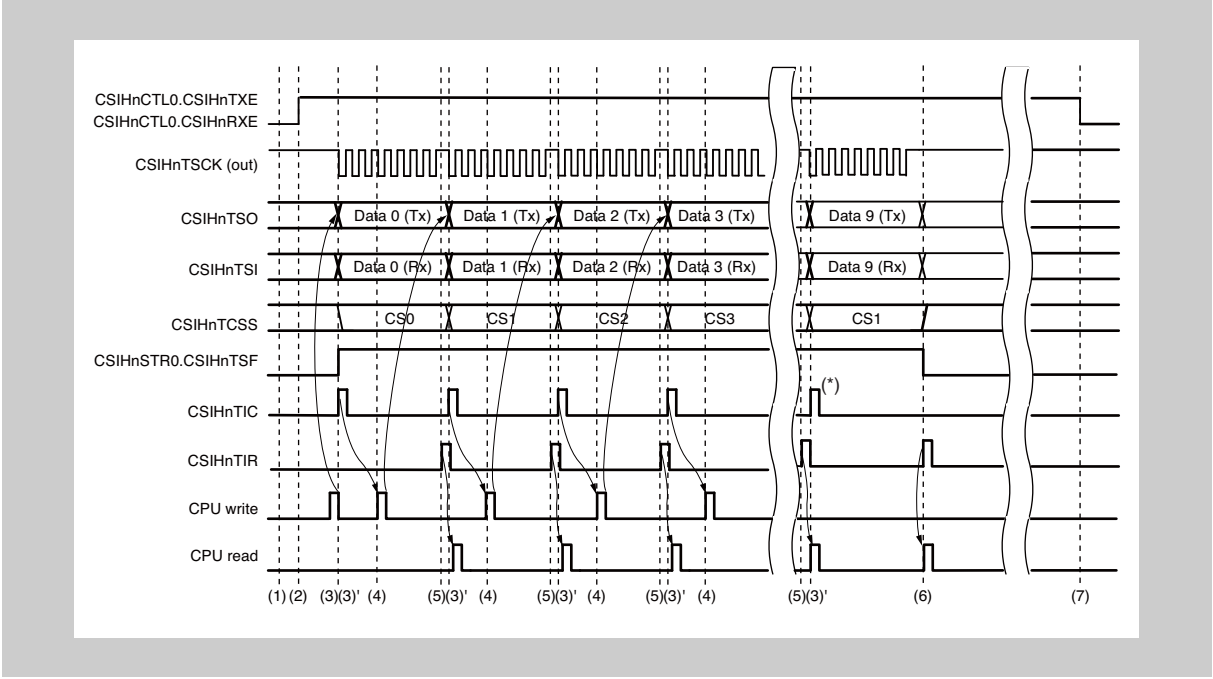


Figure 22-44 Direct access mode (for transmission/reception in the master mode, and when the job mode is disabled)

Procedure:

1. Set up the following registers before setting CSIHnCTL0.CSIHnPWR to 1:
   CSIHnCTL1, CSIHnCTL2 (transfer mode, operating mode)
   CSIHnCFGx (communication protocol)
   (For this example, the chip select signals CS0 to CS3 are used.)
2. CSIHnCTL0.CSIHnPWR = 1 (clock enabled)
   CSIHnCTL0.CSIHnTXE = 1 (transmission enabled)
   CSIHnCTL0.CSIHnRXE = 1 (reception enabled)
   CSIHnCTL0.CSIHnMBS = 1 (direct access mode selected)
3. Write the first data to the transmission data register CSIHnTX0W. This write operation activates CS0, and transmission automatically starts.

3'. When CSIHnCTL1.CSIHnSLIT is set to 1, CSIHnTIC is generated at the start edge of CSIHnTSCK. CSIHnTIC indicates that the second data can be written to CSIHnTX0W.

4. Write the second data to CSIHnTX0W. If necessary, it is possible to change the CS and make a different device the communication partner. By writing the second data immediately after writing the first data, the unnecessary inter-data delay can be avoided.
5. Each time data is received, a CSIHnTIR interrupt is generated.
   – CSIHnTIR indicates that the reception data register CSIHnRX0 must be read.
6. If the CSIHnTIC interrupt indicated by “*” in the figure is the last one, it is not necessary to write to the transmission data register CSIHnTX0W based on the corresponding CSIHnTIC interrupt.
7. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to disable transmission/reception operations. In addition, clear CSIHnCTL0.CSIHnPWR to reduce the power consumption of the CSIH.

(2) For reception in the master mode, and when the job mode is disabled

The following conditions are assumed for the procedure shown here:

- Transmission data length: 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B)
- Transmission direction: MSB first (CSIHnCFGx.CSIHnDIRx = 0)
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No delay for any interrupt (CSIHnCTL1.CSIHnSIT = 0)
- Job mode enabled (CSIHnCTL1.CSIHnJE = 1)
- A CSIHnTIC interrupt is generated when transferring starts. (CSHICTL1.CSIHnSLIT = 1)
- Direct access mode (CSHICTL0.CSIHnMBS = 1)

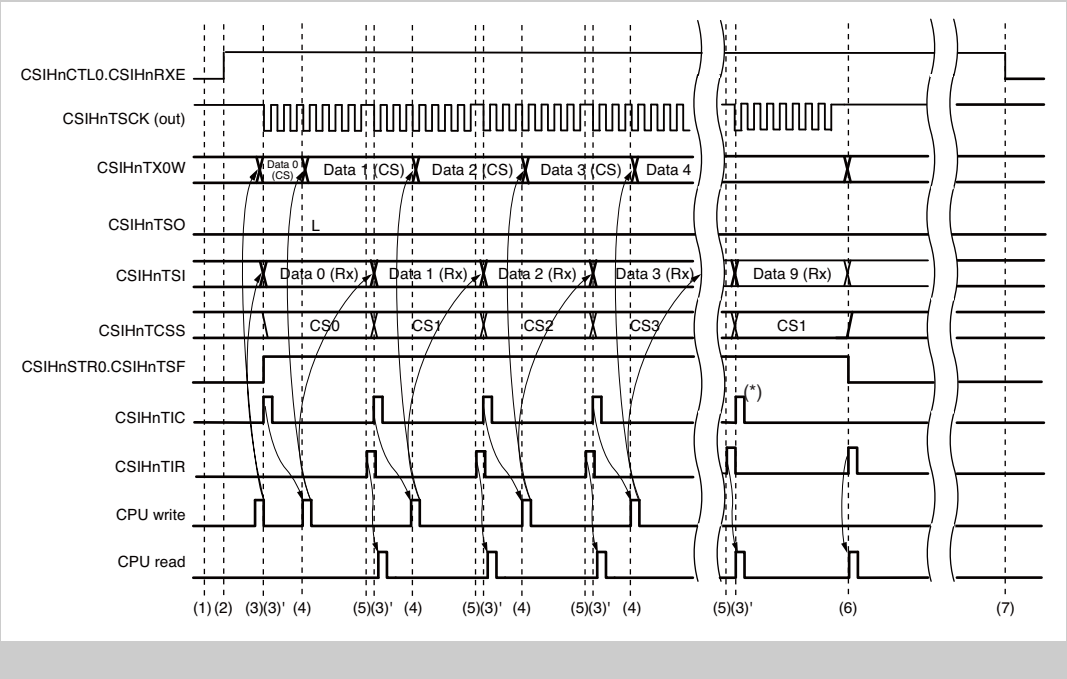


Figure 22-45 Direct access mode (for reception in the master mode, and when the job mode is disabled)

Procedure:

1. Set up the following registers before setting CSIHnCTL0.CSIHnPWR to 1:
   CSIHnCTL1, CSIHnCTL2 (transfer mode, operating mode)
   CSIHnCFGx (communication protocol)
   (For this example, the chip select signals CS0 to CS3 are used.)
2. CSIHnCTL0.CSIHnPWR = 1 (clock enabled)
   CSIHnCTL0.CSIHnTXE = 0 (transmission disabled)
   CSIHnCTL0.CSIHnRXE = 1 (reception enabled)
   CSIHnCTL0.CSIHnMBS = 1 (direct access mode selected)
3. Write the transmission data[a] to the transmission data register CSIHnTX0W for the CS data. This write operation activates CS0, and reception automatically starts.

a) The transmission data is not used, but the chip select signal is enabled.

3'. When CSIHnCTL1.CSIHnSLIT is set to 1, CSIHnTIC is generated at the start edge of CSIHnTSCK. CSIHnTIC indicates that the second data can be written to CSIHnTX0W.

4. Write the second data to CSIHnTX0W. If necessary, it is possible to change the CS and make a different device the communication partner. By writing the second data immediately after writing the first data, the unnecessary inter-data delay can be avoided.

5. Each time data is received, a CSIHnTIR interrupt is generated.

   – CSIHnTIR indicates that the reception data register CSIHnRX0W must be read.

6. If the CSIHnTIC interrupt indicated by “*” in the figure is the last one, it is not necessary to write to the transmission data register CSIHnTX0W based on the corresponding CSIHnTIC interrupt.

7. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to disable transmission/reception operations. In addition, clear CSIHnCTL0.CSIHnPWR to reduce the power consumption of the CSIH.

(3) For transmission/reception in the slave mode, and when the job mode is disabled

The following conditions are assumed for the procedure shown here:

- Transmission data length: 8 bits (CSIHnCFG0.CSIHnDLS0[3:0] = 1000_B)
- Transmission direction: MSB first (CSIHnCFG0.CSIHnDIR0 = 0)
- Normal clock phase and data phase (CSIHnCFG0.CSIHnCKP0 = 0, CSIHnCFG0.CSIHnDAP0 = 0)
- Job mode disabled (CSIHnCTL1.CSIHnJE = 0)
- Handshake enabled (CSIHnCTL1.CSIHnHSE = 1)
- CSIHnTIC interrupt generated at the transfer start timing (CSIHnCTL1.CSIHnSLIT = 1)
- Direct access mode (CSIHnCTL0.CSIHnMBS = 1)

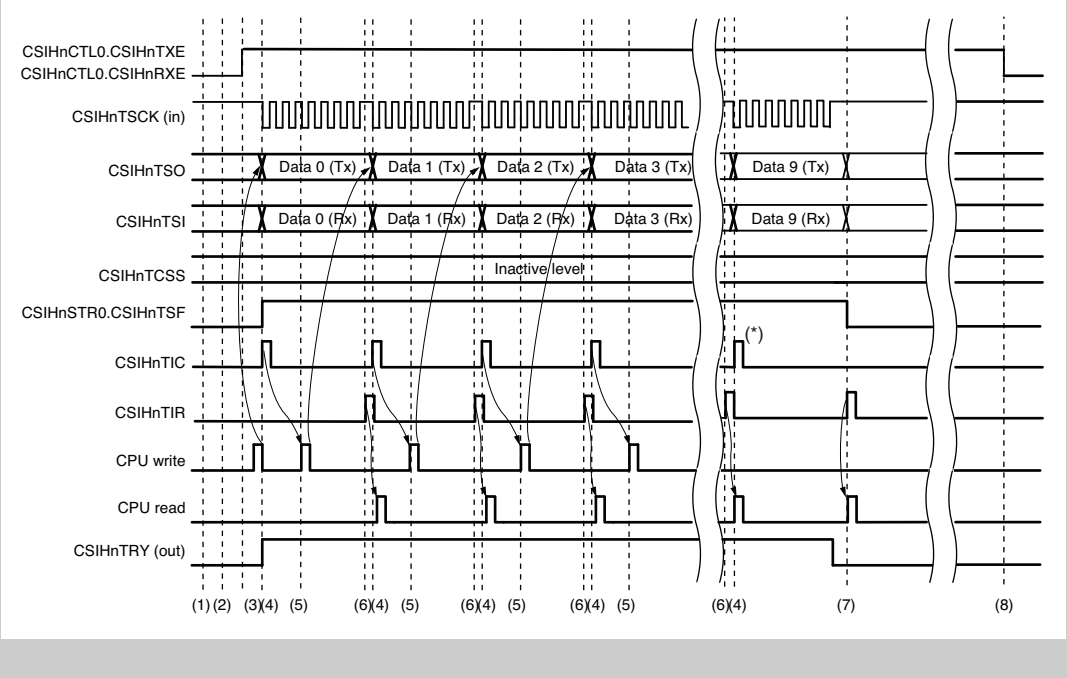


Figure 22-46 Direct access mode (for transmission/reception in the slave mode, and when the job mode is disabled)

Procedure:

1. Set up the following registers before setting CSIHnCTL0.CSIHnPWR to 1:
   CSIHnCTL1, CSIHnCTL2 (transfer mode, operating mode)
   CSIHnCFG0 (communication protocol)
2. CSIHnCTL0.CSIHnPWR = 1 (clock enabled)
   CSIHnCTL0.CSIHnTXE = 1 (transmission enabled)
   CSIHnCTL0.CSIHnRXE = 1 (reception enabled)
   CSIHnCTL0.CSIHnMBS = 1 (direct access mode selected)
3. Write the first data to the transmission data register CSIHnTX0W. The CSIHnTRY signal is switched from BUSY (low level) to READY (high level) by writing data. When a serial clock is supplied from the master, communication automatically starts.
4. When CSIHnCTL1.CSIHnSLIT is set to 1, CSIHnTIC is generated at the start edge of CSIHnTSCK. CSIHnTIC indicates that the second data can be written to CSIHnTX0W.

5. Write the second data to CSIHnTX0W. By writing the second data immediately after writing the first data, the unnecessary inter-data delay can be avoided.
6. Each time data is received, a CSIHnTIR interrupt is generated.
   – CSIHnTIR indicates that the reception data register CSIHnRX0W must be read.
7. If the CSIHnTIC interrupt indicated by "*" in the figure is the last one, it is not necessary to write to the transmission data register CSIHnTX0W based on the corresponding CSIHnTIC interrupt.
8. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to disable transmission/reception operations. In addition, clear CSIHnCTL0.CSIHnPWR to reduce the power consumption of the CSIH.

(4) For reception in the slave mode, and when the job mode is disabled

The following conditions are assumed for the procedure shown here:

- Transmission data length: 8 bits (CSIHnCFG0.CSIHnDLS0[3:0] = 1000_B)
- Transmission direction: MSB first (CSIHnCFG0.CSIHnDIR0 = 0)
- Normal clock phase and data phase (CSIHnCFG0.CSIHnCKP0 = 0, CSIHnCFG0.CSIHnDAP0 = 0)
- Job mode disabled (CSIHnCTL1.CSIHnJE = 0)
- Handshake enabled (CSIHnCTL1.CSIHnHSE = 1)
- CSIHnTIC interrupt generated at the transfer start timing (CSIHnCTL1.CSIHnSLIT = 1)
- Direct access mode (CSIHnCTL0.CSIHnMBS = 1)

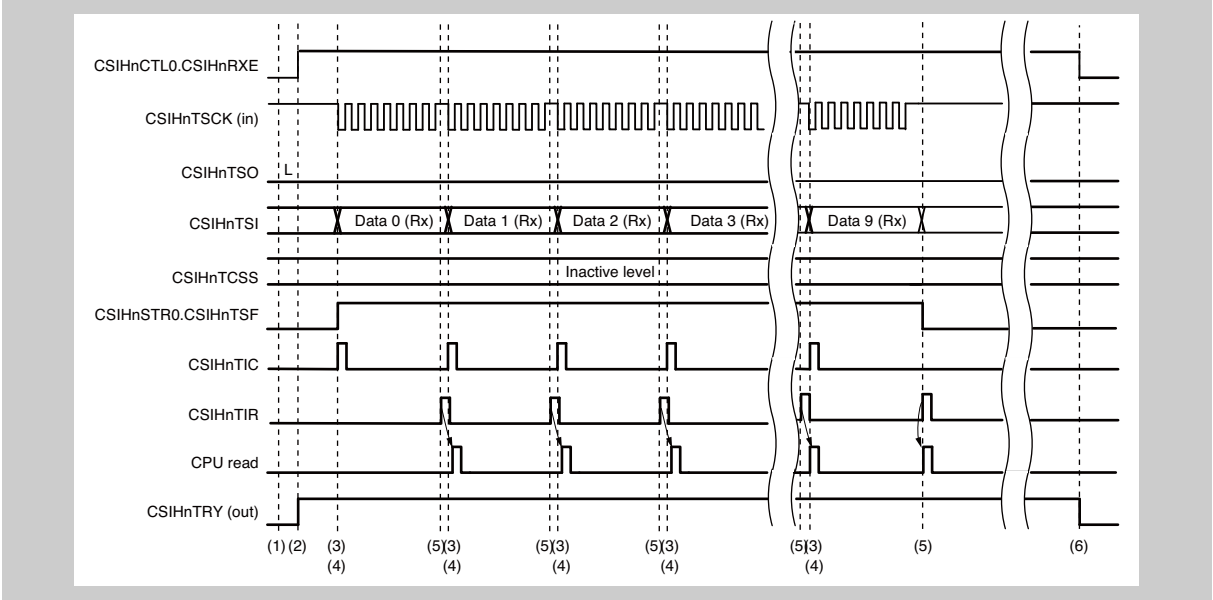


Figure 22-47 Direct access mode (for reception in the slave mode, and when the job mode is disabled)

Procedure:

1. Set up the following registers before setting CSIHnCTL0.CSIHnPWR to 1:
   CSIHnCTL1, CSIHnCTL2 (transfer mode, operating mode)
   CSIHnCFG0 (communication protocol)
2. CSIHnCTL0.CSIHnPWR = 1 (clock enabled)
   CSIHnCTL0.CSIHnTXE = 0 (transmission disabled)
   CSIHnCTL0.CSIHnRXE = 1 (reception enabled)
   CSIHnCTL0.CSIHnMBS = 1 (direct access mode selected)
3. When a serial clock is supplied from the master, reception automatically starts.
4. When CSIHnCTL1.CSIHnSLIT is set to 1, CSIHnTIC is generated at the start edge of CSIHnTSCK.
5. Each time data is received, a CSIHnTIR interrupt is generated.
   – CSIHnTIR indicates that the reception data register CSIHnRX0W must be read.
6. Finally, clear CSIHnCTL0.CSIHnRXE to disable reception operations. In addition, clear CSIHnCTL0.CSIHnPWR to reduce the power consumption of the CSIH.

(5) For transmission/reception in the master mode, and when the job mode is enabled

The following conditions are assumed for the procedure shown here:

- Transmission data length: 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B)
- Transmission direction: MSB first (CSIHnCFGx.CSIHnDIRx = 0)
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No delay for any interrupt (CSIHnCTL1.CSIHnSIT = 0)
- Job mode enabled (CSIHnCTL1.CSIHnJE = 1)
- CSIHnTIC interrupt generated at the transfer start timing (CSIHnCTL1.CSIHnSLIT = 1)
- Direct access mode (CSIHnCTL0.CSIHnMBS = 1)
- Two jobs that each transmit three data packets

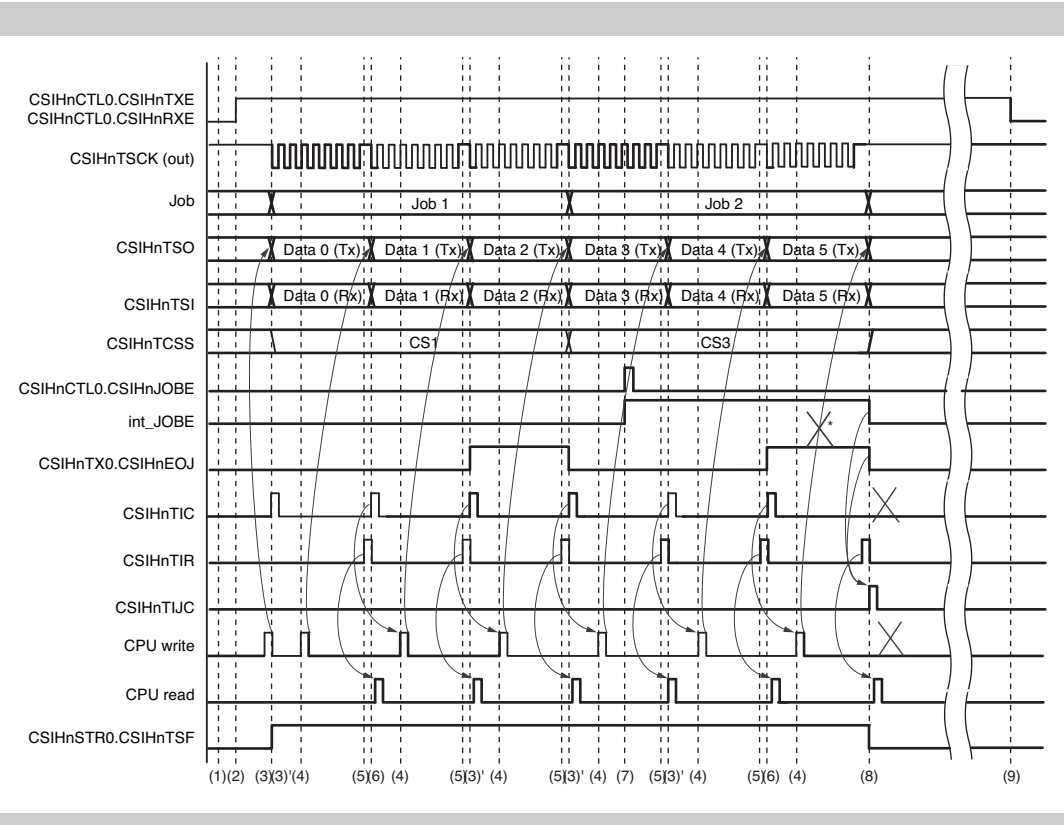


Figure 22-48 Direct access mode (for transmission/reception in the master mode, and when the job mode is enabled)

Note The int_JOBE signal in the above timing chart is the internal signal of the CSIHnJOBE bit.

Procedure:

1. Set up the following registers before setting CSIHnCTL0.CSIHnPWR to 1:
   CSIHnCTL1, CSIHnCTL2 (transfer mode, operating mode)
   CSIHnCFGx (communication protocol)
   (For this example, the chip select signals CS1 and CS3 are used.)
2. CSIHnCTL0.CSIHnPWR = 1 (clock enabled)
   CSIHnCTL0.CSIHnTXE = 1 (transmission enabled)
   CSIHnCTL0.CSIHnRXE = 1 (reception enabled)
   CSIHnCTL0.CSIHnMBS = 1 (direct access mode selected)
3. Write the first transmission data packet to the transmission data register CSIHnTX0W. Transmission automatically starts when the first data can be used.

   The CSIHnSTR0.CSIHnTSF flag indicates that communication is being performed.

3'. When CSIHnCTL1.CSIHnSLIT is set to 1, CSIHnTIC is generated at the start edge of CSIHnTSCK. CSIHnTIC indicates that the second data can be written to CSIHnTX0W.

4. Write the second data to CSIHnTX0W. By writing the second data immediately after writing the first data, the unnecessary inter-data delay can be avoided.
5. Each time data is received, a CSIHnTIR interrupt request is generated.
   – CSIHnTIR indicates that the reception data register CSIHnRX0W must be read.
6. If the CSIHnTX0W register transfer data is the last job data, CSIHnTX0W.CSIHnEOJ is set to 1.
7. By setting CSIHnCTL0.CSIHnJOBE to 1, communication is forcibly stopped when the current job (job 2) ends.
8. After communication is forcibly stopped, the interrupt request CSIHnTIC is replaced with CSIHnTIJC. CSIHnTIR is generated as usual.

   The interrupt request CSIHnTIJC indicates that communication was forcibly stopped when the current job ended.
   The interrupt request CSIHnTIC is not generated. Note that the usable transmission data in the CSIHnTX0 register (indicated by “*” in the figure) is not transmitted.
9. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to disable transmission/reception operations. In addition, clear CSIHnCTL0.CSIHnPWR to reduce the power consumption of CSIH.

(6) For reception in the master mode, and when the job mode is enabled

The following conditions are assumed for the procedure shown here:

- Transmission data length: 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B)
- Transmission direction: MSB first (CSIHnCFGx.CSIHnDIRx = 0)
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No delay for any interrupt (CSIHnCTL1.CSIHnSIT = 0)
- Job mode enabled (CSIHnCTL1.CSIHnJE = 1)
- CSIHnTIC interrupt generated at the transfer start timing (CSIHnCTL1.CSIHnSLIT = 1)
- Direct access mode (CSIHnCTL0.CSIHnMBS = 1)
- Two jobs that each transmit three data packets

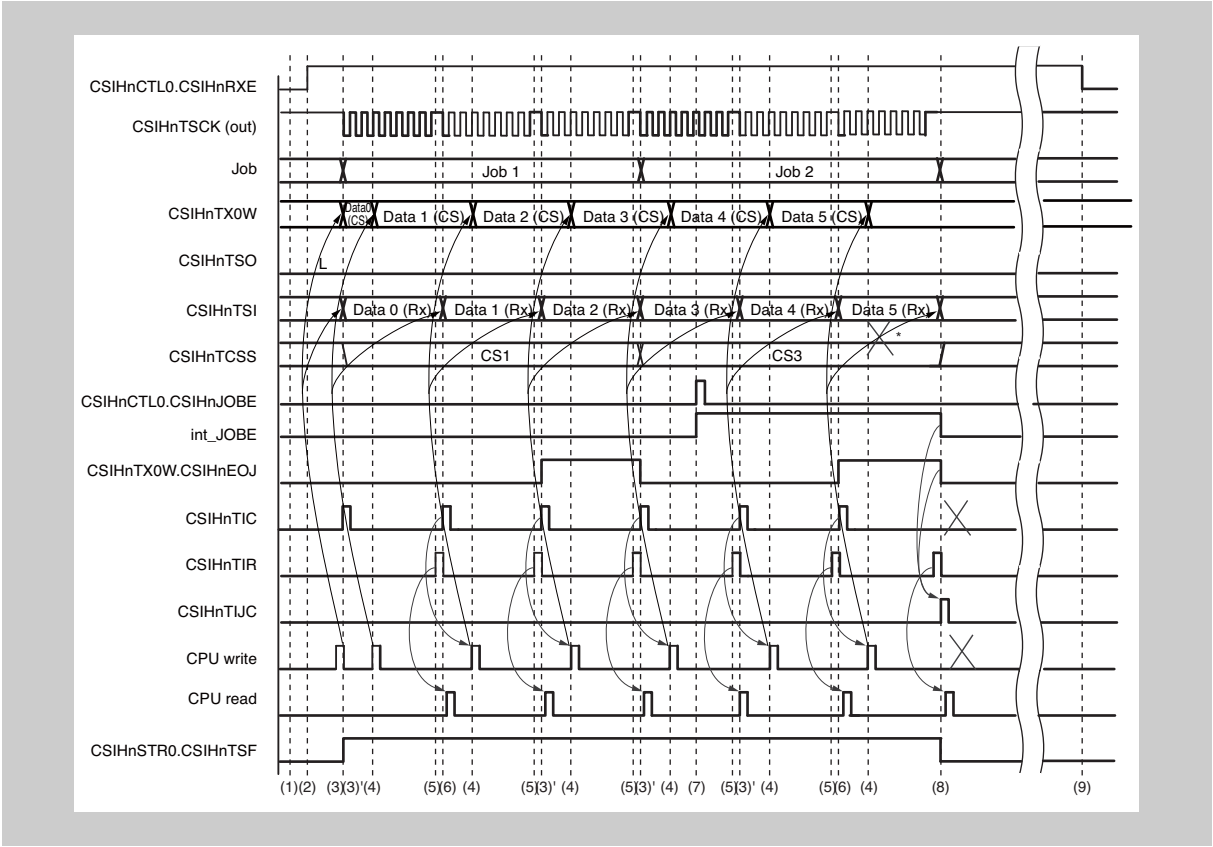


Figure 22-49 Direct access mode (for reception in the master mode, and when the job mode is enabled)

Note The int_JOBE signal in the above timing chart is the internal signal of the CSIHnJOBE bit.

Procedure:

1. Set up the following registers before setting CSIHnCTL0.CSIHnPWR to 1:
   CSIHnCTL1, CSIHnCTL2 (transfer mode, operating mode)
   CSIHnCFGx (communication protocol)
   (For this example, the chip select signals CS1 and CS3 are used.)
2. CSIHnCTL0.CSIHnPWR = 1 (clock enabled)
   CSIHnCTL0.CSIHnTXE = 0 (transmission disabled)
   CSIHnCTL0.CSIHnRXE = 1 (reception enabled)
   CSIHnMBS = 1 (direct access mode selected)
3. Write the transmission data to the transmission data register CSIHnTX0W for reception.
   Reception automatically starts. In addition, the CSIHnSTR0.CSIHnTSF bit is set.

3'. When CSIHnCTL1.CSIHnSLIT is set to 1, CSIHnTIC is generated at the start edge of CSIHnTSCK. CSIHnTIC indicates that the second data can be written to CSIHnTX0W.

4. Write the second data to CSIHnTX0H. By writing the second data immediately after writing the first data, the unnecessary inter-data delay can be avoided.
5. Each time data is received, a CSIHnTIR interrupt request is generated.
   – CSIHnTIR indicates that the reception data register CSIHnRX0 must be read.
6. If the CSIHnTX0W register transfer data is the last job data, CSIHnTX0W.CSIHnEOJ is set to 1.
7. By setting CSIHnCTL0.CSIHnJOBE to 1, communication is forcibly stopped when the current job (job 2) ends.
8. When int_JOBE is set and the last job 2 data is received, the interrupt request CSIHnTIJC is generated instead of CSIHnTIC.

   CSIHnTIR is generated as usual.
   The interrupt request CSIHnTIJC indicates that reception was forcibly stopped when the current job ended.
   The interrupt request CSIHnTIC is not generated. Note that the data indicated by “*” in the figure is not transferred.
9. Finally, clear CSIHnCTL0.CSIHnRXE to disable reception operations. In addition, clear CSIHnCTL0.CSIHnPWR to reduce the power consumption of CSIH.

22.5.2 Procedures in transmit-only buffer mode

This section provides examples where the job mode is enabled or disabled.

(1) For transmission/reception in the master mode, and when the job mode is disabled

The following conditions are assumed for the procedure shown here:

- Transmission data length: 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B)
- Transmission direction: MSB first (CSIHnCFGx.CSIHnDIRx = 0)
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No delay for any interrupt (CSIHnCTL1.CSIHnSIT = 0)
- Job mode disabled (CSIHnCTL1.CSIHnJE = 0)
- Transmit-only buffer mode (CSIHnCTL0.CSIHnMBS = 0, CSIHnMCTL0.CSIHnMMS[1:0] = 10_B)
- Number of transmitted data items: 9 (CSIHnMCTL2.CSIHnND[7:0] = 09_H)
- Transfer start address: 10_H (CSIHnMCTL2.CSIHnSOP[6:0] = 10_H)
- Normal CSIHnTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)

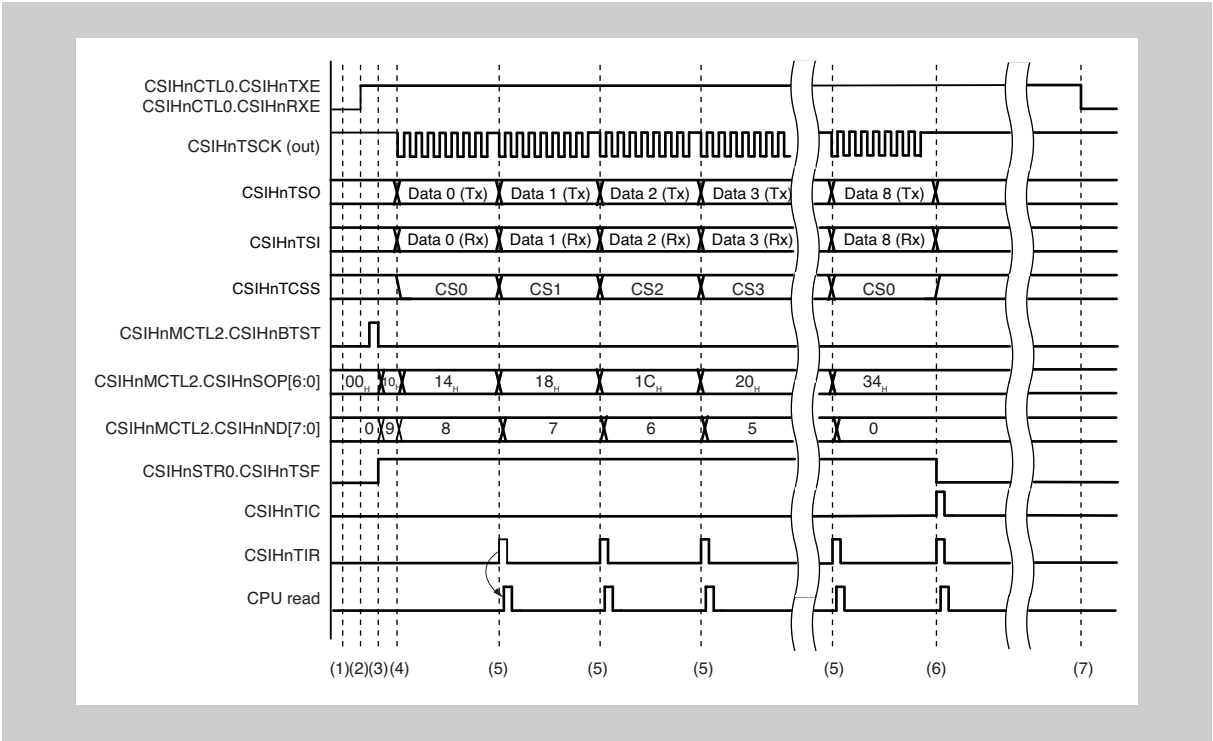


Figure 22-50 Transmit-only buffer mode (for transmission/reception in the master mode, and when the job mode is disabled)

Note The procedure for writing data to the buffer is not described here. The first data address is specified for CSIHnMRWP0.CSIHnTRWA[6:0], and the transfer data is written to CSIHnTX0W. Each time transfer data is written, the value of CSIHnMRWP0.CSIHnTRWA[6:0] is incremented.

Procedure:

1. Set up the following registers before setting CSIHnCTL0.CSIHnPWR to 1:
   CSIHnCTL1, CSIHnCTL2 (transfer mode, operating mode)
   CSIHnMCTL0.CSIHnMMS[1:0] = 10_B (memory mode)
   CSIHnCFGx (communication protocol)
   (For this example, the chip select signals CS0 to CS3 are used.)
2. CSIHnCTL0.CSIHnPWR = 1 (clock enabled)
   CSIHnCTL0.CSIHnTXE = 1 (transmission enabled)
   CSIHnCTL0.CSIHnRXE = 1 (reception enabled)
   CSIHnCTL0.CSIHnMBS = 0 (memory mode)
3. The transmission pointer and number of data items are specified using the CSIHnMCTL2.CSIHnSOP[6:0] and CSIHnMCTL2.CSIHnND[7:0] bits. Communication is started by setting CSIHnMCTL2.CSIHnBTST.
4. Transmission/reception starts. The CSIHnMCTL2.CSIHnSOP[6:0] bits are automatically incremented, and the CSIHnMCTL2.CSIHnND[7:0] bits are decremented each time a data item is transmitted.
5. When all the data is received, CSIHnTIR is generated. The CSIHnTIR interrupt indicates that the reception data register CSIHnRX0W must be read.
6. When all the data is transmitted, a CSIHnTIC interrupt request is generated.
7. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to disable transmission/reception operations. In addition, clear CSIHnCTL0.CSIHnPWR to reduce the power consumption while not using CSIH.

(2) For reception in the master mode, and when the job mode is disabled

The following conditions are assumed for the procedure shown here:

- Transmission data length: 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B)
- Transmission direction: MSB first (CSIHnCFGx.CSIHnDIRx = 0)
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No delay for any interrupt (CSIHnCTL1.CSIHnSIT = 0)
- Job mode enabled (CSIHnCTL1.CSIHnJE = 1)
- Transmit-only buffer mode (CSIHnCTL0.CSIHnMBS = 0, CSIHnMCTL0.CSIHnMMS[1:0] = 10_B)
- Number of transmitted data items: 9 (CSIHnMCTL2.CSIHnND[7:0] = 09_H)
- Transfer start address: 10_H (CSIHnMCTL2.CSIHnSOP[6:0] = 10_H)
- Normal CSIHnTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)

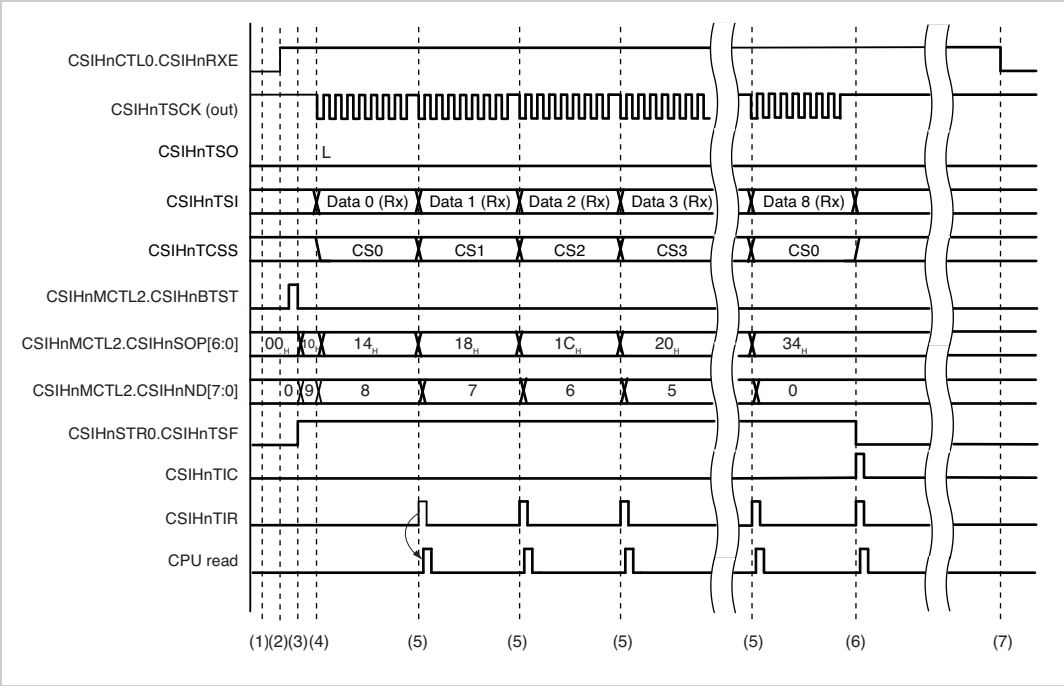


Figure 22-51 Transmit-only buffer mode (for reception in the master mode, and when the job mode is disabled)

Note The procedure for writing data to the buffer is not described here. The first data address is specified for CSIHnMRWP0.CSIHnTRWA[6:0], and the transfer data is written to CSIHnTX0W. Each time transfer data is written, the value of CSIHnMRWP0.CSIHnTRWA[6:0] is incremented.

Procedure:

1. Set up the following registers before setting CSIHnCTL0.CSIHnPWR to 1:
   CSIHnCTL1, CSIHnCTL2 (transfer mode, operating mode)
   CSIHnMCTL0.CSIHnMMS[1:0] = 10_B (memory mode)
   CSIHnCFGx (communication protocol)
   (For this example, the chip select signals CS0 to CS3 are used.)
2. CSIHnCTL0.CSIHnPWR = 1 (clock enabled)
   CSIHnCTL0.CSIHnTXE = 0 (transmission disabled)
   CSIHnCTL0.CSIHnRXE = 1 (reception enabled)
   CSIHnCTL0.CSIHnMBS = 0 (memory mode)
3. The transmission pointer and number of data items are specified using the CSIHnMCTL2.CSIHnSOP[6:0] and CSIHnMCTL2.CSIHnND[7:0] bits. Communication is started by setting CSIHnMCTL2.CSIHnBTST.
4. Reception starts. The CSIHnMCTL2.CSIHnSOP[6:0] bits are automatically incremented, and the CSIHnMCTL2.CSIHnND[7:0] bits are decremented each time a data packet is transmitted.
5. When all the data is received, CSIHnTIR is generated. The CSIHnTIR interrupt indicates that the reception data register CSIHnRX0W must be read.
6. When all the data is received, a CSIHnTIC interrupt request is generated.
7. Finally, clear CSIHnCTL0.CSIHnRXE to disable reception operations. In addition, clear CSIHnCTL0.CSIHnPWR to reduce the power consumption while not using CSIH.

(3) For transmission/reception in the slave mode, and when the job mode is disabled

The following conditions are assumed for the procedure shown here:

- Transmission data length: 8 bits (CSIHnCFG0.CSIHnDLS0[3:0] = 1000_B)
- Transmission direction: MSB first (CSIHnCFG0.CSIHnDIR0 = 0)
- Normal clock phase and data phase (CSIHnCFG0.CSIHnCKP0 = 0, CSIHnCFG0.CSIHnDAP0 = 0)
- Job mode disabled (CSIHnCTL1.CSIHnJE = 0)
- Handshake enabled (CSIHnCTL1.CSIHnHSE = 1)
- Transmit-only buffer mode (CSIHnCTL0.CSIHnMBS = 0, CSIHnMCTL0.CSIHnMMS[1:0] = 10_B)
- Number of transmitted data items: 9 (CSIHnMCTL2.CSIHnND[7:0] = 09_H)
- Transfer start address: 10_H (CSIHnMCTL2.CSIHnSOP[6:0] = 10_H)
- Normal CSIHnTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)

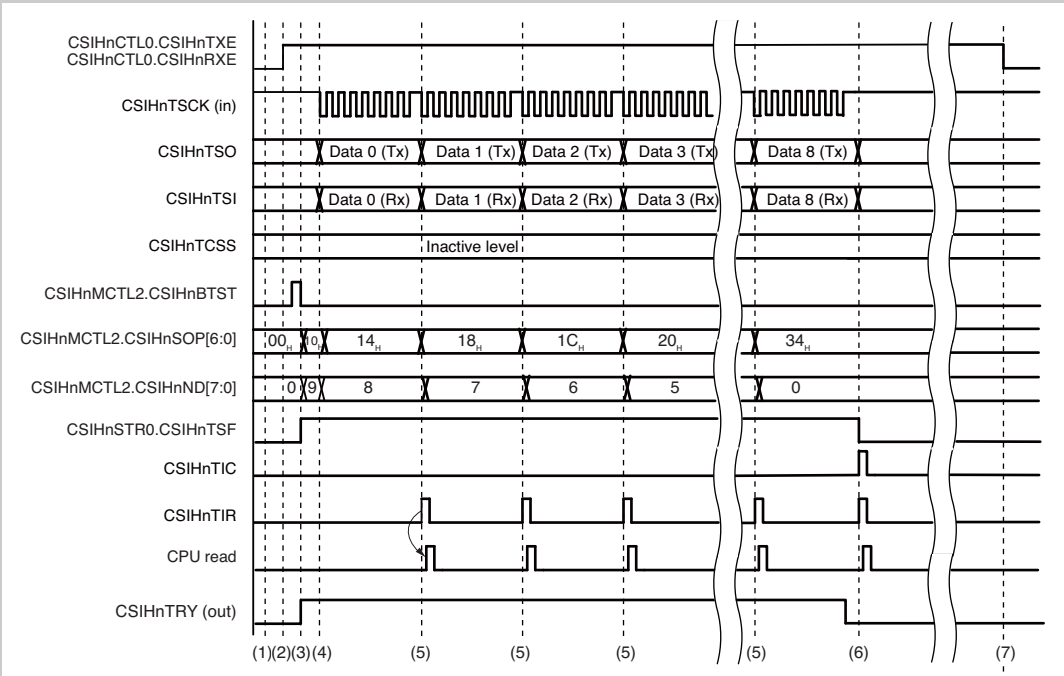


Figure 22-52 Transmit-only buffer mode (for transmission/reception in the slave mode, and when the job mode is disabled)

Note The procedure for writing data to the buffer is not described here. The first data address is specified for CSIHnMRWP0.CSIHnTRWA[6:0], and the transfer data is written to CSIHnTX0W. Each time transfer data is written, the value of CSIHnMRWP0.CSIHnTRWA[6:0] is incremented.

Procedure:

1. Set up the following registers before setting CSIHnCTL0.CSIHnPWR to 1:
   CSIHnCTL1, CSIHnCTL2 (transfer mode, operating mode)
   CSIHnMCTL0.CSIHnMMS[1:0] = 10_B (memory mode)
   CSIHnCFG0 (communication protocol)
2. CSIHnCTL0.CSIHnPWR = 1 (clock enabled)
   CSIHnCTL0.CSIHnTXE = 1 (transmission enabled)
   CSIHnCTL0.CSIHnRXE = 1 (reception enabled)
   CSIHnCTL0.CSIHnMBS = 0 (memory mode)
3. The transmission pointer and number of data items are specified using the CSIHnMCTL2.CSIHnSOP[6:0] and CSIHnMCTL2.CSIHnND[7:0] bits. Communication is started by setting CSIHnMCTL2.CSIHnBTST.
4. When a serial clock is supplied from the master, communication starts. The CSIHnMCTL2.CSIHnSOP[6:0] bits are automatically incremented, and the CSIHnMCTL2.CSIHnND[7:0] bits are decremented each time a data packet is transmitted.
5. Each time data is received, CSIHnTIR is generated. The CSIHnTIR interrupt indicates that the reception data register CSIHnRX0W must be read.
6. When all the data is received, a CSIHnTIC interrupt request is generated.
7. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to disable transmission/reception operations. In addition, clear CSIHnCTL0.CSIHnPWR to reduce the power consumption while not using CSIH.

(4) For reception in the slave mode, and when the job mode is disabled

The following conditions are assumed for the procedure shown here:

- Transmission data length: 8 bits (CSIHnCFG0.CSIHnDLS0[3:0] = 1000_B)
- Transmission direction: MSB first (CSIHnCFG0.CSIHnDIR0 = 0)
- Normal clock phase and data phase (CSIHnCFG0.CSIHnCKP0 = 0, CSIHnCFG0.CSIHnDAP0 = 0)
- Job mode disabled (CSIHnCTL1.CSIHnJE = 0)
- Handshake enabled (CSIHnCTL1.CSIHnHSE = 1)
- Transmit-only buffer mode (CSIHnCTL0.CSIHnMBS = 0, CSIHnMCTL0.CSIHnMMS[1:0] = 10_B)
- Number of transmitted data items: 9 (CSIHnMCTL2.CSIHnND[7:0] = 09_H)
- Transfer start address: 10_H (CSIHnMCTL2.CSIHnSOP[6:0] = 10_H)
- Normal CSIHnTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)

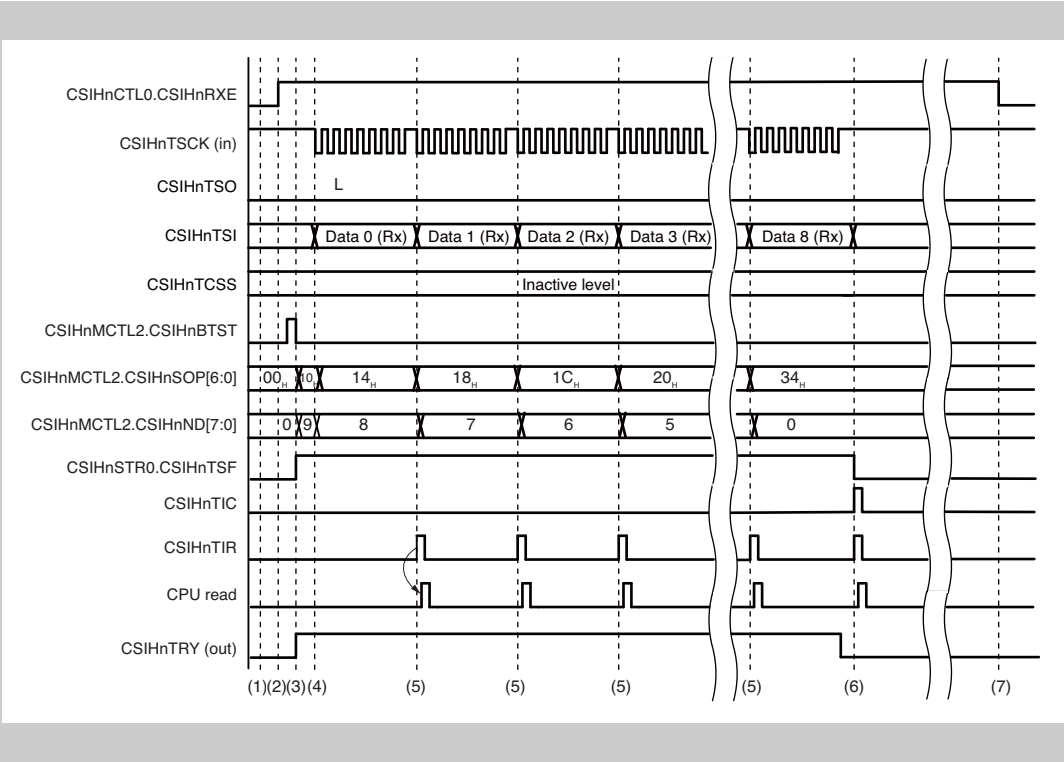


Figure 22-53 Transmit-only buffer mode (for reception in the slave mode, and when the job mode is disabled)

Procedure:

1. Set up the following registers before setting CSIHnCTL0.CSIHnPWR to 1:
   CSIHnCTL1, CSIHnCTL2 (transfer mode, operating mode)
   CSIHnMCTL0.CSIHnMMS[1:0] = 10_B (memory mode)
   CSIHnCFG0 (communication protocol)
2. CSIHnCTL0.CSIHnPWR = 1 (clock enabled)
   CSIHnCTL0.CSIHnTXE = 0 (transmission disabled)
   CSIHnCTL0.CSIHnRXE = 1 (reception enabled)
   CSIHnCTL0.CSIHnMBS = 0 (memory mode)
3. The transmission pointer and number of data items are specified using the CSIHnMCTL2.CSIHnSOP[6:0] and CSIHnMCTL2.CSIHnND[7:0] bits. Reception is started by setting CSIHnMCTL2.CSIHnBTST.
4. When a serial clock is supplied from the master, reception starts. The CSIHnMCTL2.CSIHnSOP[6:0] bits are automatically incremented, and the CSIHnMCTL2.CSIHnND[7:0] bits are decremented each time a data packet is transmitted.
5. Each time data is received, CSIHnTIR is generated. The CSIHnTIR interrupt indicates that the reception data register CSIHnRX0W must be read.
6. When all the data is received, a CSIHnTIC interrupt request is generated.
7. Finally, clear CSIHnCTL0.CSIHnRXE to disable reception operations. In addition, clear CSIHnCTL0.CSIHnPWR to reduce the power consumption while not using CSIH.

(5) For transmission/reception in the master mode, and when the job mode is enabled

The following conditions are assumed for the procedure shown here:

- Transmission data length: 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B)
- Transmission direction: MSB first (CSIHnCFGx.CSIHnDIRx = 0)
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No delay for any interrupt (CSIHnCTL1.CSIHnSIT = 0)
- Job mode enabled (CSIHnCTL1.CSIHnJE = 1)
- Transmit-only buffer mode (CSIHnCTL0.CSIHnMBS = 0, CSIHnMCTL0.CSIHnMMS[1:0] = 10_B)
- Number of transmitted data items: 9 (CSIHnMCTL2.CSIHnND[7:0] = 09_H)
- Transfer start address: 10_H (CSIHnMCTL2.CSIHnSOP[6:0] = 10_H)
- Normal CSIHnTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)

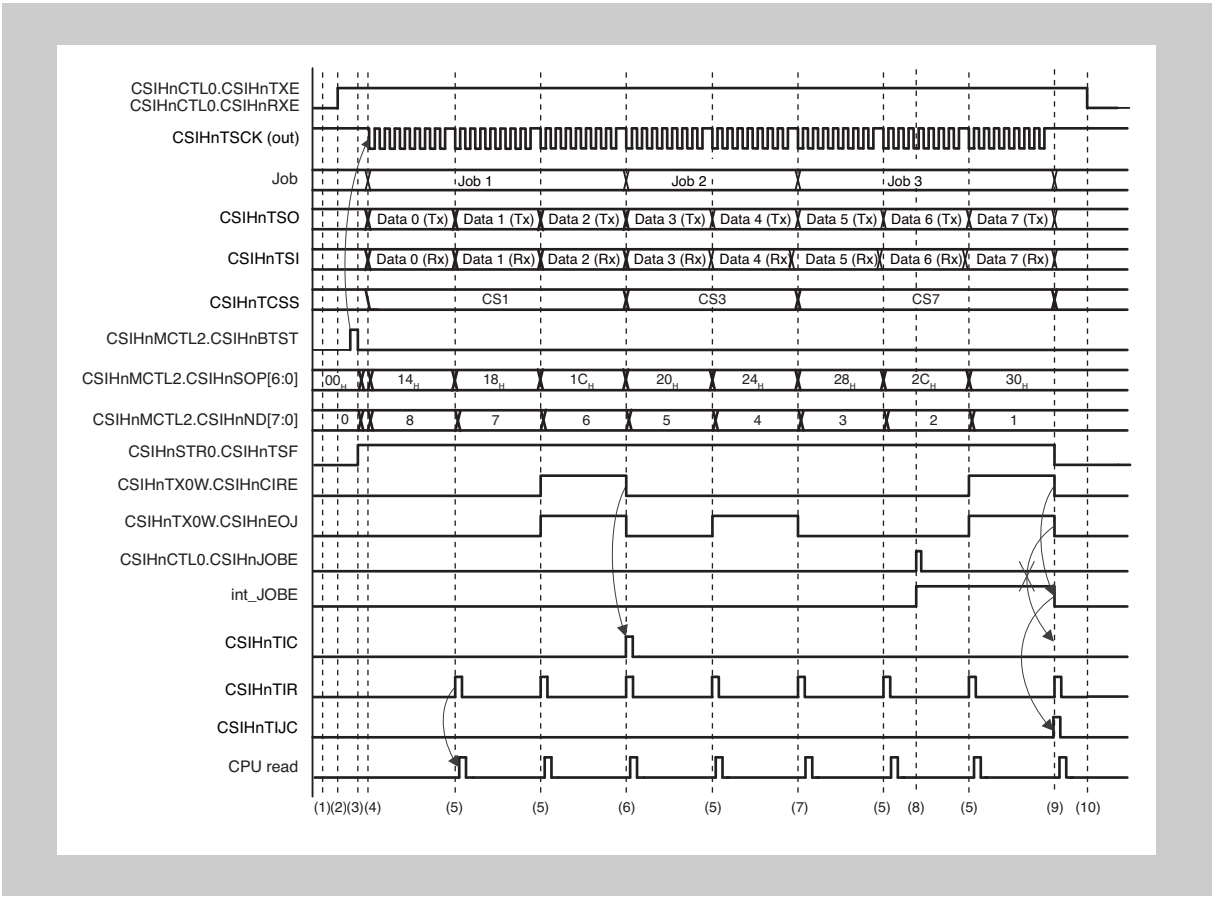


Figure 22-54 Transmit-only buffer mode (for transmission/reception in the master mode, and when the job mode is enabled)

Notes

1. The procedure for writing data to the buffer is not described here. The first data address is specified for CSIHnMRWP0.CSIHnTRWA[6:0], and the transfer data is written to CSIHnTX0W. Each time transfer data is written, the value of CSIHnMRWP0.CSIHnTRWA[6:0] is incremented.
2. The int_JOBE signal in the above timing chart is the internal signal of the CSIHnJOBE bit.

Procedure:

1. Set up the following registers before setting CSIHnCTL0.CSIHnPWR to 1:
   CSIHnCTL1, CSIHnCTL2 (transfer mode, operating mode)
   CSIHnMCTL0.CSIHnMMS[1:0] = 10_B (memory mode)
   CSIHnCFGx (communication protocol)
   (For this example, the chip select signals CS1, CS3, and CS7 are used.)
2. CSIHnCTL0.CSIHnPWR = 1 (clock enabled)
   CSIHnCTL0.CSIHnTXE = 1 (transmission enabled)
   CSIHnCTL0.CSIHnRXE = 1 (reception enabled)
   CSIHnCTL0.CSIHnMBS = 0 (memory mode)
3. The transmission pointer and number of data items are specified using the CSIHnMCTL2.CSIHnSOP[6:0] and CSIHnMCTL2.CSIHnND[7:0] bits.
   Communication is started by setting CSIHnMCTL2.CSIHnBTST.
4. Transmission starts. The CSIHnMCTL2.CSIHnSOP[6:0] bits are automatically incremented, and the CSIHnMCTL2.CSIHnND[7:0] bits are decremented each time a data item is transmitted.
5. Each time a data item is received, a CSIHnTIR interrupt request is generated.
   CSIHnTIR indicates that the reception data register CSIHnRX0W must be read.
6. CSIHnTIC is generated by setting CSIHnTX0W.CSIHnCIRE to 1.
   CSIHnTIC indicates that the last data (CSIHnTX0W.CSIHnEOJ = 1) of the current job was transmitted.
7. Because the last data (CSIHnTX0W.CSIHnEOJ = 1) of the current job was transmitted by clearing CSIHnTX0W.CSIHnCIRE, the interrupt request CSIHnTIC is not generated.
8. By setting CSIHnCTL0.CSIHnJOBE, communication is forcibly stopped when job 3 ends.
9. After communication is forcibly stopped, the interrupt requests CSIHnTIJC and CSIHnTIR are generated when job 3 ends.

   The interrupt request CSIHnTIJC indicates that communication was forcibly stopped when the current job ended.
   Because the interrupt request CSIHnTIJC is generated instead of the interrupt request CSIHnTIC, the interrupt request CSIHnTIC is not generated.
10. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to disable transmission/reception operations. In addition, clear CSIHnCTL0.CSIHnPWR to reduce the power consumption while not using the CSIH.

(6) For reception in the master mode, and when the job mode is enabled

The following conditions are assumed for the procedure shown here:

- Transmission data length: 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B)
- Transmission direction: MSB first (CSIHnCFGx.CSIHnDIRx = 0)
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No delay for any interrupt (CSIHnCTL1.CSIHnSIT = 0)
- Job mode enabled (CSIHnCTL1.CSIHnJE = 1)
- Transmit-only buffer mode (CSIHnCTL0.CSIHnMBS = 0, CSIHnMCTL0.CSIHnMMS[1:0] = 10_B)
- Number of transmitted data items: 9 (CSIHnMCTL2.CSIHnND[7:0] = 09_H)
- Transfer start address: 10_H (CSIHnMCTL2.CSIHnSOP[6:0] = 10_H)
- Normal CSIHnTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)

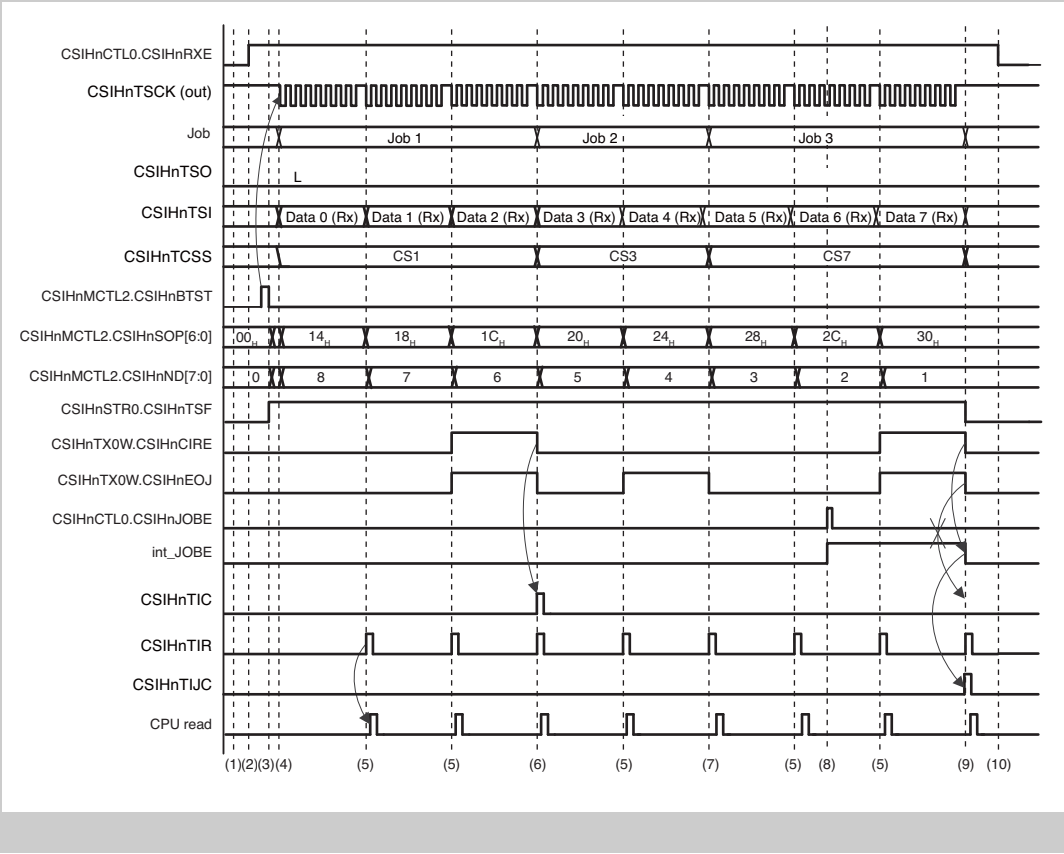


Figure 22-55 Transmit-only buffer mode (for reception in the master mode, and when the job mode is enabled)

Notes

1. The procedure for writing data to the buffer is not described here. The first data address is specified for CSIHnMRWP0.CSIHnTRWA[6:0], and the transfer data is written to CSIHnTX0W. Each time transfer data is written, the value of CSIHnMRWP0.CSIHnTRWA[6:0] is incremented.
2. The int_JOBE signal in the above timing chart is the internal signal of the CSIHnJOBE bit.

Procedure:

1. Set up the following registers before setting CSIHnCTL0.CSIHnPWR to 1:
   CSIHnCTL1, CSIHnCTL2 (transfer mode, operating mode)
   CSIHnMCTL0.CSIHnMMS[1:0] = 10_B (memory mode)
   CSIHnCFGx (communication protocol)
   (For this example, the chip select signals CS1, CS3, and CS7 are used.)
2. CSIHnCTL0.CSIHnPWR = 1 (clock enabled)
   CSIHnCTL0.CSIHnTXE = 0 (transmission disabled)
   CSIHnCTL0.CSIHnRXE = 1 (reception enabled)
   CSIHnCTL0.CSIHnMBS = 0 (memory mode)
3. The transmission pointer and number of data items are specified using the CSIHnMCTL2.CSIHnSOP[6:0] and CSIHnMCTL2.CSIHnND[7:0] bits.
   Communication is started by setting CSIHnMCTL2.CSIHnBTST.
4. Reception starts. The CSIHnMCTL2.CSIHnSOP[6:0] bits are automatically incremented, and the CSIHnMCTL2.CSIHnND[7:0] bits are decremented each time a data item is transmitted.
5. Each time a data item is received, a CSIHnTIR interrupt request is generated.
   CSIHnTIR indicates that the reception data register CSIHnRX0W must be read.
6. CSIHnTIC is generated by setting CSIHnTX0W.CSIHnCIRE to 1.
   CSIHnTIC indicates that the last data (CSIHnTX0W.CSIHnEOJ = 1) of the current job was transmitted.
7. Because the last data (CSIHnTX0W.CSIHnEOJ = 1) of the current job was transmitted by clearing CSIHnTX0W.CSIHnCIRE, the interrupt request CSIHnTIC is not generated.
8. By setting CSIHnCTL0.CSIHnJOBE, reception is forcibly stopped when job 3 ends.
9. After communication is forcibly stopped, the interrupt requests CSIHnTIJC and CSIHnTIR are generated when job 3 ends.

   The interrupt request CSIHnTIJC indicates that reception was forcibly stopped when the current job ended.
   Because the interrupt request CSIHnTIJC is generated instead of the interrupt request CSIHnTIC, the interrupt request CSIHnTIC is not generated.
10. Finally, clear CSIHnCTL0.CSIHnRXE to disable reception operations.
    In addition, clear CSIHnCTL0.CSIHnPWR to reduce the power consumption while not using the CSIH.

22.5.3 Procedures in dual buffer mode

(1) For transmission/reception in the master mode, and when the job mode is disabled

The following conditions are assumed for the procedure shown here:

- Transmission data length: 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B)
- Transmission direction: MSB first (CSIHnCFGx.CSIHnDIRx = 0)
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No delay for any interrupt (CSIHnCTL1.CSIHnSIT = 0)
- Job mode disabled (CSIHnCTL1.CSIHnJE = 0)
- Dual buffer mode (CSIHnCTL0.CSIHnMBS = 0, CSIHnMCTL0.CSIHnMMS[1:0] = 01_B)
- Number of data packets: 9 (CSIHnMCTL2.CSIHnND[7:0] = 09_H)
- Transfer start address: 10_H (CSIHnMCTL2.CSIHnSOP[6:0] = 10_H)
- Normal CSIHnTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)

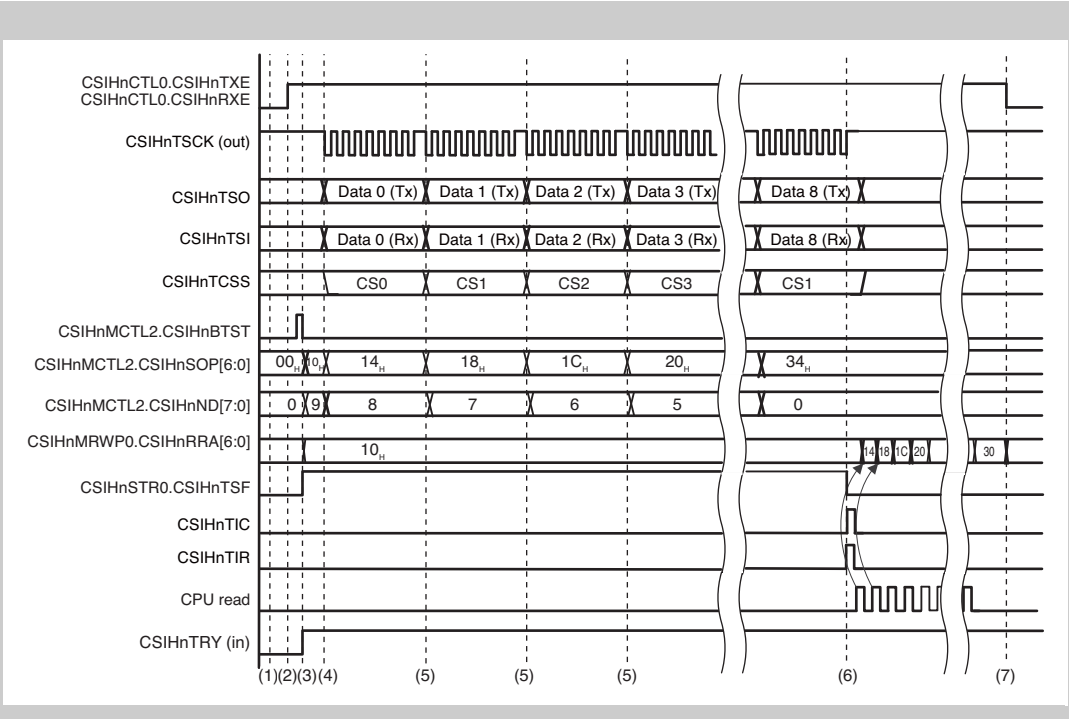


Figure 22-56 Dual buffer mode (for transmission/reception in the master mode, and when the job mode is disabled)

Note The procedure for writing data to the buffer is not described here. The first data address is specified for CSIHnMRWP0.CSIHnTRWA[6:0], and the transfer data is written to CSIHnTX0W. Each time transfer data is written, the value of CSIHnMRWP0.CSIHnTRWA[6:0] is incremented.

Procedure:

1. Set up the following registers before setting CSIHnCTL0.CSIHnPWR to 1:
   CSIHnCTL1, CSIHnCTL2 (transfer mode, operating mode)
   CSIHnMCTL0.CSIHnMMS[1:0] = 01_B (memory mode)
   CSIHnCFGx (communication protocol)
   (For this example, the chip select signals CS0 to CS3 are used.)
   CSIHnSTCR0.CSIHnPCT = 1 (buffer pointers cleared)
2. CSIHnCTL0.CSIHnPWR = 1 (clock enabled)
   CSIHnCTL0.CSIHnTXE = 1 (transmission enabled)
   CSIHnCTL0.CSIHnRXE = 1 (reception enabled)
   CSIHnCTL0.CSIHnMBS = 0 (memory mode)
3. The transmission pointer and number of data items are specified using the CSIHnMCTL2.CSIHnSOP[6:0] and CSIHnMCTL2.CSIHnND[7:0] bits. Communication is started by setting CSIHnMCTL2.CSIHnBTST.
4. Transmission starts. Each time a data item is transmitted, the CSIHnMCTL2.CSIHnSOP[6:0] bits are automatically incremented, and the CSIHnMCTL2.CSIHnND[7:0] bits are decremented.
5. (4) is repeated until the last data is transmitted/received.
   The interrupt requests CSIHnTIC and CSIHnTIR are not generated.
6. When all the communication ends, the interrupt requests CSIHnTIC and CSIHnTIR are generated.
   The CPU starts reading the received data from the reception buffer. The read start address is specified by the CSIHnMRWP0.CSIHnRRA[6:0] bits. The CSIHnMRWP0.CSIHnRRA[6:0] bits are incremented each time a data item is read.
7. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to disable transmission/reception operations. In addition, clear CSIHnCTL0.CSIHnPWR to reduce the power consumption while not using the CSIH.

(2) For reception in the master mode, and when the job mode is disabled

The following conditions are assumed for the procedure shown here:

- Transmission data length: 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B)
- Transmission direction: MSB first (CSIHnCFGx.CSIHnDIRx = 0)
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No delay for any interrupt (CSIHnCTL1.CSIHnSIT = 0)
- Job mode disabled (CSIHnCTL1.CSIHnJE = 0)
- Dual buffer mode (CSIHnCTL0.CSIHnMBS = 0, CSIHnMCTL0.CSIHnMMS[1:0] = 01_B)
- Number of data packets: 9 (CSIHnMCTL2.CSIHnND[7:0] = 09_H)
- Transfer start address: 10_H (CSIHnMCTL2.CSIHnSOP[6:0] = 10_H)
- Normal CSIHnTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)

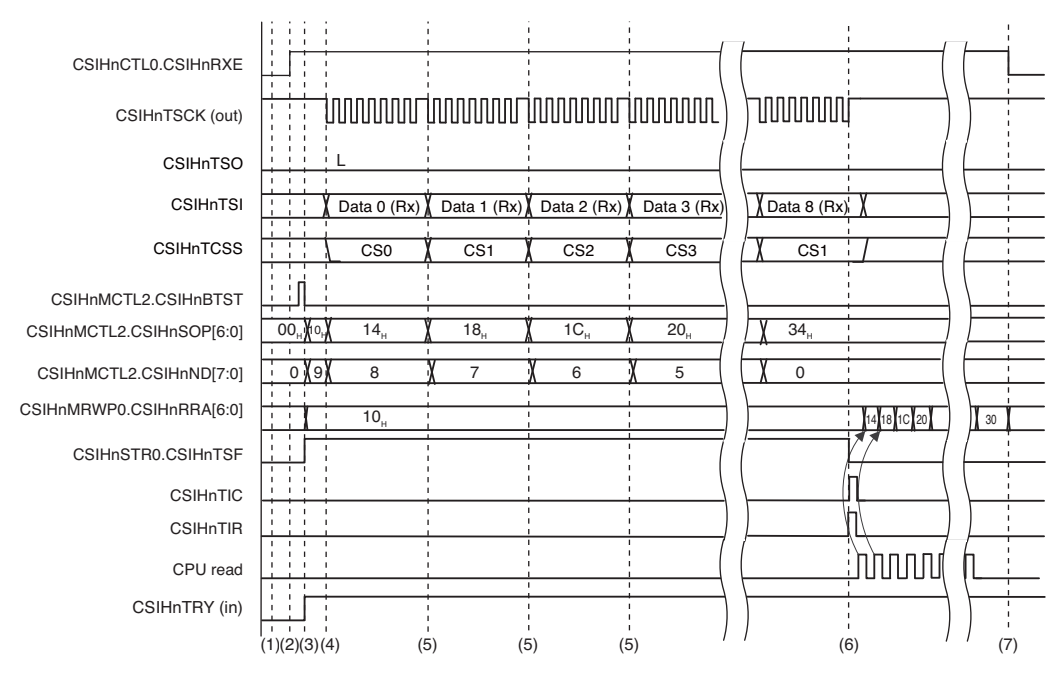


Figure 22-57 Dual buffer mode (for reception in the master mode, and when the job mode is disabled)

Note The procedure for writing data to the buffer is not described here. The first data address is specified for CSIHnMRWP0.CSIHnTRWA[6:0], and the transfer data is written to CSIHnTX0W. Each time transfer data is written, the value of CSIHnMRWP0.CSIHnTRWA[6:0] is incremented.

Procedure:

1. Set up the following registers before setting CSIHnCTL0.CSIHnPWR to 1:
   CSIHnCTL1, CSIHnCTL2 (transfer mode, operating mode)
   CSIHnMCTL0.CSIHnMMS[1:0] = 01_B (memory mode)
   CSIHnCFGx (communication protocol)
   (For this example, the chip select signals CS0 to CS3 are used.)
   CSIHnSTCR0.CSIHnPCT = 1 (buffer pointers cleared)
2. CSIHnCTL0.CSIHnPWR = 1 (clock enabled)
   CSIHnCTL0.CSIHnTXE = 0 (transmission disabled)
   CSIHnCTL0.CSIHnRXE = 1 (reception enabled)
   CSIHnCTL0.CSIHnMBS = 0 (memory mode)
3. The transmission pointer and number of data items are specified using the CSIHnMCTL2.CSIHnSOP[6:0] and CSIHnMCTL2.CSIHnND[7:0] bits.
   Reception is started by setting CSIHnMCTL2.CSIHnBTST.
4. Reception starts. Each time a data item is received, the CSIHnMCTL2.CSIHnSOP[6:0] bits are automatically incremented, and the CSIHnMCTL2.CSIHnND[7:0] bits are decremented.
5. (4) is repeated until the last data is received.
   The interrupt requests CSIHnTIC and CSIHnTIR are not generated.
6. When all the reception ends, the interrupt requests CSIHnTIC and CSIHnTIR are generated.
   The CPU starts reading the received data from the reception buffer. The read start address is specified by the CSIHnMRWP0.CSIHnRRA[6:0] bits. The CSIHnMRWP0.CSIHnRRA[6:0] bits are incremented each time a data item is read.
7. Finally, clear CSIHnCTL0.CSIHnRXE to disable reception operations.
   In addition, clear CSIHnCTL0.CSIHnPWR to reduce the power consumption while not using the CSIH.

(3) For transmission/reception in the slave mode, and when the job mode is disabled

The following conditions are assumed for the procedure shown here:

- Transmission data length: 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B)
- Transmission direction: MSB first (CSIHnCFGx.CSIHnDIRx = 0)
- Normal clock phase and data phase (CSIHnCFG0.CSIHnCKP0 = 0, CSIHnCFG0.CSIHnDAP0 = 0)
- No delay for any interrupt (CSIHnCTL1.CSIHnSIT = 0)
- Job mode disabled (CSIHnCTL1.CSIHnJE = 0)
- Handshake enabled (CSIHnCTL1.CSIHnHSE = 1)
- Dual buffer mode (CSIHnCTL0.CSIHnMBS = 0, CSIHnMCTL0.CSIHnMMS[1:0] = 01_B)
- Number of data packets: 9 (CSIHnMCTL2.CSIHnND[7:0] = 09_H)
- Transfer start address: 10_H (CSIHnMCTL2.CSIHnSOP[6:0] = 10_H)
- Normal CSIHnTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)

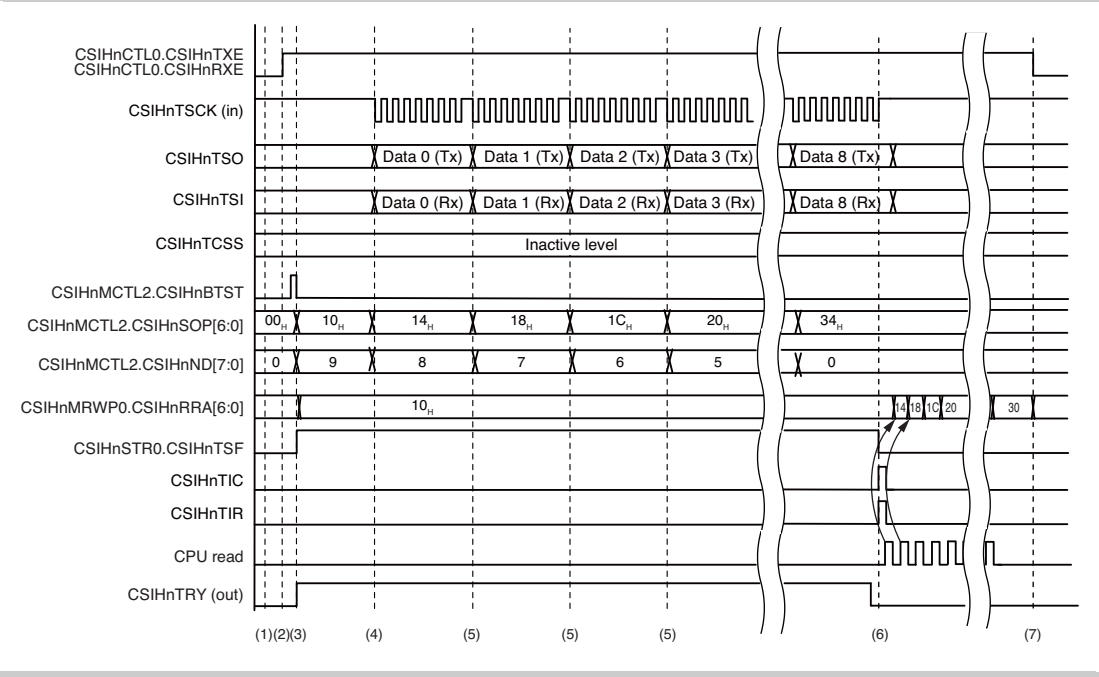


Figure 22-58 Dual buffer mode (for transmission/reception in the slave mode, and when the job mode is disabled)

Note The procedure for writing data to the buffer is not described here. The first data address is specified for CSIHnMRWP0.CSIHnTRWA[6:0], and the transfer data is written to CSIHnTX0W. Each time transfer data is written, the value of CSIHnMRWP0.CSIHnTRWA[6:0] is incremented.

Procedure:

1. Set up the following registers before setting CSIHnCTL0.CSIHnPWR to 1:
   CSIHnCTL1, CSIHnCTL2 (transfer mode, operating mode)
   CSIHnMCTL0.CSIHnMMS[1:0] = 01_B (memory mode)
   CSIHnCFG0 (communication protocol)
   CSIHnSTCR0.CSIHnPCT = 1 (buffer pointers cleared)
2. CSIHnCTL0.CSIHnPWR = 1 (clock enabled)
   CSIHnCTL0.CSIHnTXE = 1 (transmission enabled)
   CSIHnCTL0.CSIHnRXE = 1 (reception enabled)
   CSIHnCTL0.CSIHnMBS = 0 (memory mode)
3. The transmission pointer and number of data items are specified using the CSIHnMCTL2.CSIHnSOP[6:0] and CSIHnMCTL2.CSIHnND[7:0] bits.
   Communication is started by setting CSIHnMCTL2.CSIHnBTST.
4. Transmission starts. Each time a data item is transmitted, the CSIHnMCTL2.CSIHnSOP[6:0] bits are automatically incremented, and the CSIHnMCTL2.CSIHnND[7:0] bits are decremented.
5. (4) is repeated until the last data is transmitted/received.
   The interrupt requests CSIHnTIC and CSIHnTIR are not generated.
6. When all the communication ends, the interrupt requests CSIHnTIC and CSIHnTIR are generated.
   The CPU starts reading the received data from the reception buffer. The read start address is specified by the CSIHnMRWP0.CSIHnRRA[6:0] bits. The CSIHnMRWP0.CSIHnRRA[6:0] bits are incremented each time a data item is read.
7. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to disable transmission/reception operations. In addition, clear CSIHnCTL0.CSIHnPWR to reduce the power consumption while not using the CSIH.

(4) For reception in the slave mode, and when the job mode is disabled

The following conditions are assumed for the procedure shown here:

- Transmission data length: 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B)
- Transmission direction: MSB first (CSIHnCFGx.CSIHnDIRx = 0)
- Normal clock phase and data phase (CSIHnCFG0.CSIHnCKP0 = 0, CSIHnCFG0.CSIHnDAP0 = 0)
- No delay for any interrupt (CSIHnCTL1.CSIHnSIT = 0)
- Job mode disabled (CSIHnCTL1.CSIHnJE = 0)
- Handshake enabled (CSIHnCTL1.CSIHnHSE = 1)
- Dual buffer mode (CSIHnCTL0.CSIHnMBS = 0, CSIHnMCTL0.CSIHnMMS[1:0] = 01_B)
- Number of data packets: 9 (CSIHnMCTL2.CSIHnND[7:0] = 09_H)
- Transfer start address: 10_H (CSIHnMCTL2.CSIHnSOP[6:0] = 10_H)
- Normal CSIHnTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)

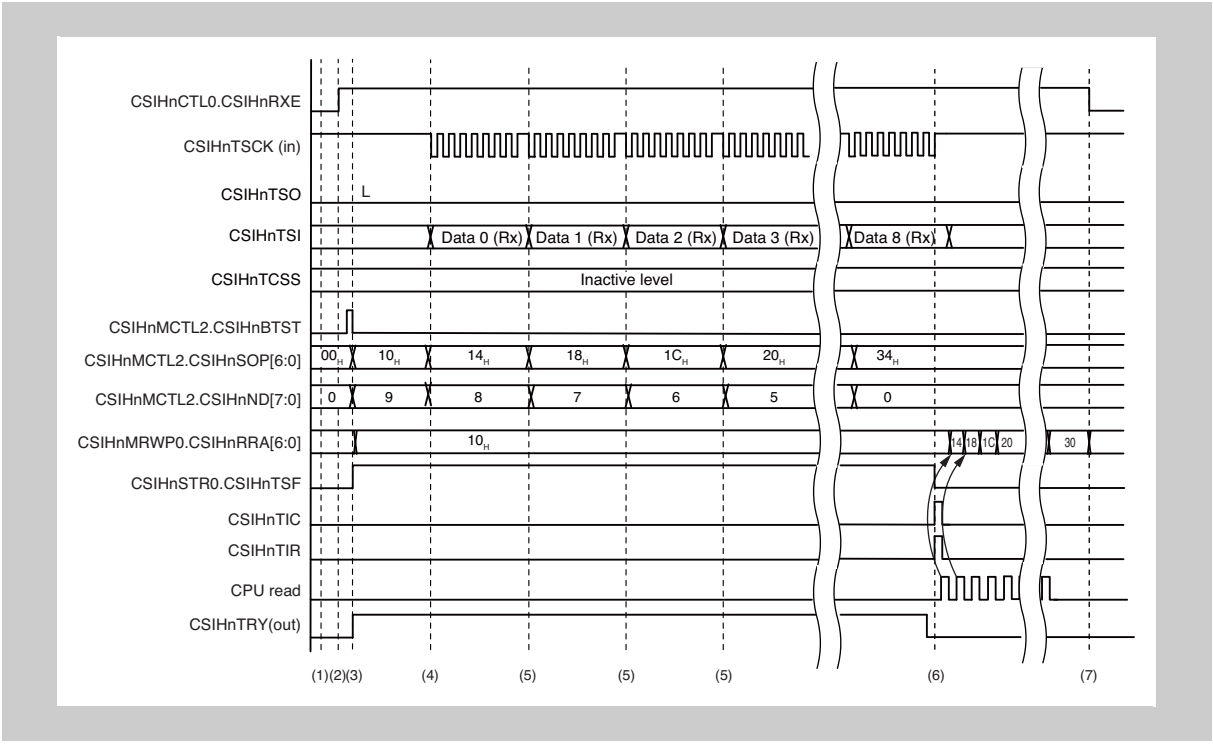


Figure 22-59 Dual buffer mode (for reception in the slave mode, and when the job mode is disabled)

Note The procedure for writing data to the buffer is not described here. The first data address is specified for CSIHnMRWP0.CSIHnTRWA[6:0], and the transfer data is written to CSIHnTX0W. Each time transfer data is written, the value of CSIHnMRWP0.CSIHnTRWA[6:0] is incremented.

Procedure:

1. Set up the following registers before setting CSIHnCTL0.CSIHnPWR to 1:
   CSIHnCTL1, CSIHnCTL2 (transfer mode, operating mode)
   CSIHnMCTL0.CSIHnMMS[1:0] = 01_B (memory mode)
   CSIHnCFG0 (communication protocol)
   CSIHnSTCR0.CSIHnPCT = 1 (buffer pointers cleared)
2. CSIHnCTL0.CSIHnPWR = 1 (clock enabled)
   CSIHnCTL0.CSIHnTXE = 0 (transmission disabled)
   CSIHnCTL0.CSIHnRXE = 1 (reception enabled)
   CSIHnCTL0.CSIHnMBS = 0 (memory mode)
3. The transmission pointer and number of data items are specified using the CSIHnMCTL2.CSIHnSOP[6:0] and CSIHnMCTL2.CSIHnND[7:0] bits.
   Reception is started by setting CSIHnMCTL2.CSIHnBTST.
4. Reception starts. Each time a data item is received, the CSIHnMCTL2.CSIHnSOP[6:0] bits are automatically incremented, and the CSIHnMCTL2.CSIHnND[7:0] bits are decremented.
5. (4) is repeated until the last data is received.
   The interrupt requests CSIHnTIC and CSIHnTIR are not generated.
6. When all the reception ends, the interrupt requests CSIHnTIC and CSIHnTIR are generated.
   The CPU starts reading the received data from the reception buffer. The read start address is specified by the CSIHnMRWP0.CSIHnRRA[6:0] bits. The CSIHnMRWP0.CSIHnRRA[6:0] bits are incremented each time a data item is read.
7. Finally, clear CSIHnCTL0.CSIHnRXE to disable reception operations.
   In addition, clear CSIHnCTL0.CSIHnPWR to reduce the power consumption while not using the CSIH.

(5) For transmission/reception in the master mode, and when the job mode is enabled

The following conditions are assumed for the procedure shown here:

- Transmission data length: 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B)
- Transmission direction: MSB first (CSIHnCFGx.CSIHnDIRx = 0)
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No delay for any interrupt (CSIHnCTL1.CSIHnSIT = 0)
- Job mode enabled (CSIHnCTL1.CSIHnJE = 1)
- Dual buffer mode (CSIHnCTL0.CSIHnMBS = 0, CSIHnMCTL0.CSIHnMMS[1:0] = 01_B)
- Number of data packets: 12 (CSIHnMCTL2.CSIHnND[7:0] = 12_H)
- Transfer start address: 00_H (CSIHnMCTL2.CSIHnSOP[6:0] = 00_H)
- Normal CSIHnTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)

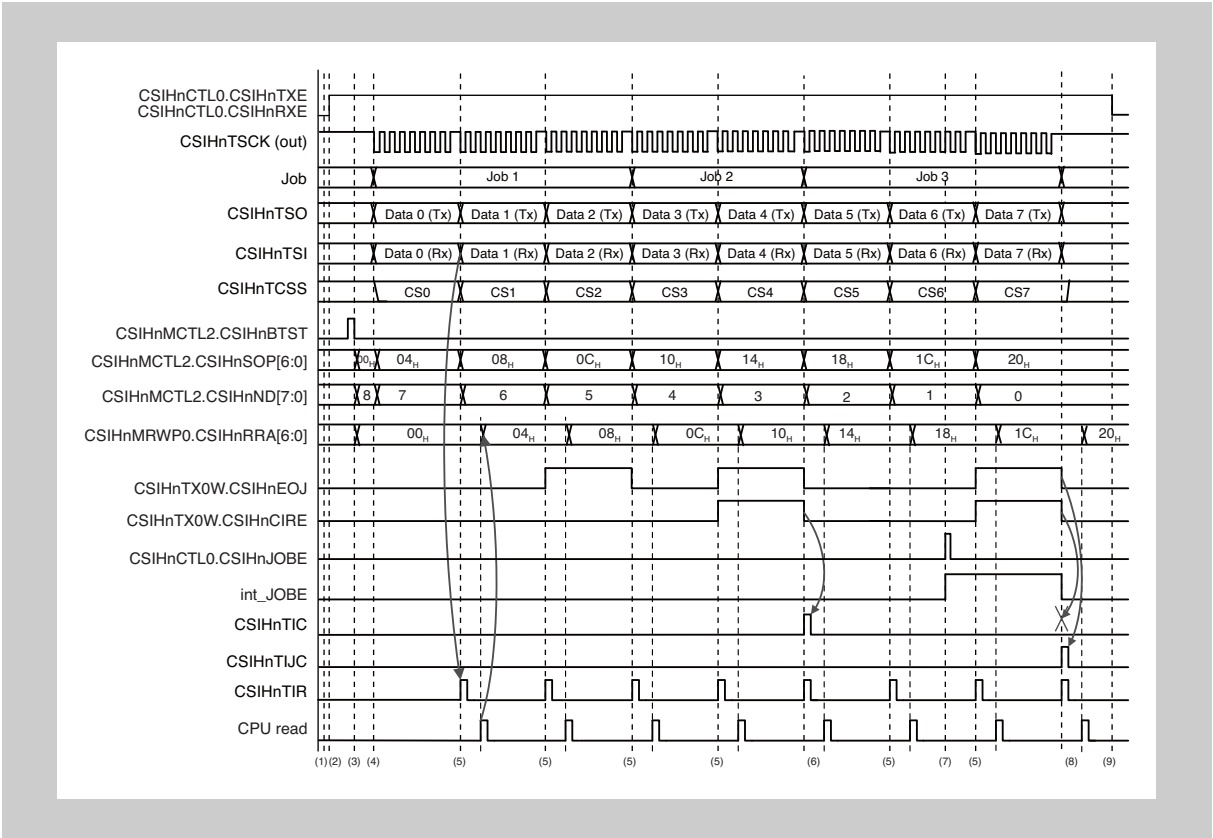


Figure 22-60 Dual buffer mode (for transmission/reception in the master mode, and when the job mode is enabled)

Notes

1. The procedure for writing data to the buffer is not described here. The first data address is specified for CSIHnMRWP0.CSIHnTRWA[6:0], and the transfer data is written to CSIHnTX0W. Each time transfer data is written, the value of CSIHnMRWP0.CSIHnTRWA[6:0] is incremented.
2. The int_JOBE signal in the above timing chart is the internal signal of the CSIHnCTL0.CSIHnJOBE bit.

Procedure:

1. Set up the following registers before setting CSIHnCTL0.CSIHnPWR to 1:
   CSIHnCTL1, CSIHnCTL2 (transfer mode, operating mode)
   CSIHnMCTL0.CSIHnMMS[1:0] = 01_B (memory mode)
   CSIHnCFGx (communication protocol)
   (For this example, the chip select signals CS0 to CS7 are used.)
   CSIHnSTCR0.CSIHnPCT = 1 (buffer pointers cleared)
2. CSIHnCTL0.CSIHnPWR = 1 (clock enabled)
   CSIHnCTL0.CSIHnTXE = 1 (transmission enabled)
   CSIHnCTL0.CSIHnRXE = 1 (reception enabled)
   CSIHnCTL0.CSIHnMBS = 0 (memory mode)
3. The transmission pointer and number of data items are specified using the CSIHnMCTL2.CSIHnSOP[6:0] and CSIHnMCTL2.CSIHnND[7:0] bits. Communication is started by setting CSIHnMCTL2.CSIHnBTST.
4. Transmission starts. Each time a data item is transmitted, the CSIHnMCTL2.CSIHnSOP[6:0] bits are automatically incremented, and the CSIHnMCTL2.CSIHnND[7:0] bits are decremented.
5. When all the data is received, CSIHnTIR is generated. The CSIHnTIR interrupt indicates that the reception data register CSIHnRX0W must be read.
6. CSIHnTIC is generated by setting CSIHnTX0W.CSIHnCIRE to 1. CSIHnTIC indicates that the last data (CSIHnTX0W.CSIHnEOJ = 1) of the current job was transmitted.
7. By setting CSIHnCTL0.CSIHnJOBE to 1, communication is forcibly stopped when job 3 ends.
8. After communication is forcibly stopped, the interrupt requests CSIHnTIJC and CSIHnTIR are generated when job 3 ends.
   The interrupt request CSIHnTIJC indicates that communication was forcibly stopped when the current job ended.
   Because the interrupt request CSIHnTIJC is generated instead of the interrupt request CSIHnTIC, the interrupt request CSIHnTIC is not generated. Note that transfer data is not transmitted by the CSIHnTX0W register.
9. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to disable transmission/reception operations. In addition, clear CSIHnCTL0.CSIHnPWR to reduce the power consumption while not using the CSIH.

(6) For reception in the master mode, and when the job mode is enabled

The following conditions are assumed for the procedure shown here:

- Transmission data length: 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B)
- Transmission direction: MSB first (CSIHnCFGx.CSIHnDIRx = 0)
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No delay for any interrupt (CSIHnCTL1.CSIHnSIT = 0)
- Job mode enabled (CSIHnCTL1.CSIHnJE = 1)
- Dual buffer mode (CSIHnCTL0.CSIHnMBS = 0, CSIHnMCTL0.CSIHnMMS[1:0] = 01_B)
- Number of data packets: 12 (CSIHnMCTL2.CSIHnND[7:0] = 12_H)
- Transfer start address: 00_H (CSIHnMCTL2.CSIHnSOP[6:0] = 00_H)
- Normal CSIHnTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)

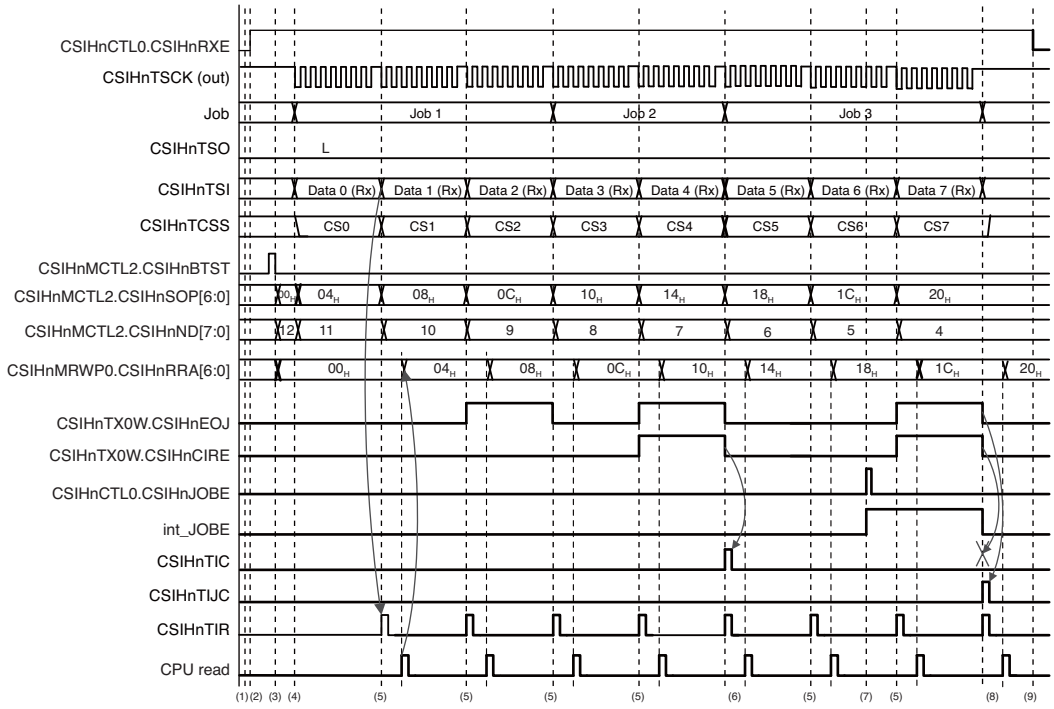


Figure 22-61 Dual buffer mode (for reception in the master mode, and when the job mode is enabled)

Notes

1. The procedure for writing data to the buffer is not described here. The first data address is specified for CSIHnMRWP0.CSIHnTRWA[6:0], and the transfer data is written to CSIHnTX0W. Each time transfer data is written, the value of CSIHnMRWP0.CSIHnTRWA[6:0] is incremented.
2. The int_JOBE signal in the above timing chart is the internal signal of the CSIHnCTL0.CSIHnJOBE bit.

Procedure:

1. Set up the following registers before setting CSIHnCTL0.CSIHnPWR to 1:
   CSIHnCTL1, CSIHnCTL2 (transfer mode, operating mode)
   CSIHnMCTL0.CSIHnMMS[1:0] = 01_B (memory mode)
   CSIHnCFGx (communication protocol)
   (For this example, the chip select signals CS0 to CS7 are used.)
   CSIHnSTCR0.CSIHnPCT = 1 (buffer pointers cleared)
2. CSIHnCTL0.CSIHnPWR = 1 (clock enabled)
   CSIHnCTL0.CSIHnTXE = 0 (transmission disabled)
   CSIHnCTL0.CSIHnRXE = 1 (reception enabled)
   CSIHnCTL0.CSIHnMBS = 0 (memory mode)
3. The transmission pointer and number of data items are specified using the CSIHnMCTL2.CSIHnSOP[6:0] and CSIHnMCTL2.CSIHnND[7:0] bits. Communication is started by setting CSIHnMCTL2.CSIHnBTST.
4. Reception starts. Each time a data item is received, the CSIHnMCTL2.CSIHnSOP[6:0] bits are automatically incremented, and the CSIHnMCTL2.CSIHnND[7:0] bits are decremented.
5. Each time data is received, CSIHnTIR is generated. The CSIHnTIR interrupt indicates that the reception data register CSIHnRX0W must be read.
6. CSIHnTIC is generated by setting CSIHnTX0W.CSIHnCIRE to 1. CSIHnTIC indicates that the last data (CSIHnTX0W.CSIHnEOJ = 1) of the current job was transmitted.
7. By setting CSIHnCTL0.CSIHnJOBE to 1, reception is forcibly stopped when job 3 ends.
8. After reception is forcibly stopped, the interrupt requests CSIHnTIJC and CSIHnTIR are generated when job 3 ends.
   The interrupt request CSIHnTIJC indicates that reception was forcibly stopped when the current job ended.
   Because the interrupt request CSIHnTIJC is generated instead of the interrupt request CSIHnTIC, the interrupt request CSIHnTIC is not generated. Note that transfer data is not transmitted by the CSIHnTX0W register.
9. Finally, clear CSIHnCTL0.CSIHnRXE to disable reception operations. In addition, clear CSIHnCTL0.CSIHnPWR to reduce the power consumption while not using the CSIH.

22.5.4 Procedures in FIFO mode

(1) For transmission/reception in the master mode, and when the job mode is disabled

The following conditions are assumed for the procedure shown here:

- Transmission data length: 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B)
- Transmission direction: MSB first (CSIHnCFGx.CSIHnDIRx = 0)
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No delay for any interrupt (CSIHnCTL1.CSIHnSIT = 0)
- Job mode disabled (CSIHnCTL1.CSIHnJE = 0)
- FIFO mode (CSIHnCTL0.CSIHnMBS = 0, CSIHnMCTL0.CSIHnMMS[1:0] = 00_B)
- Normal CSIHnTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)

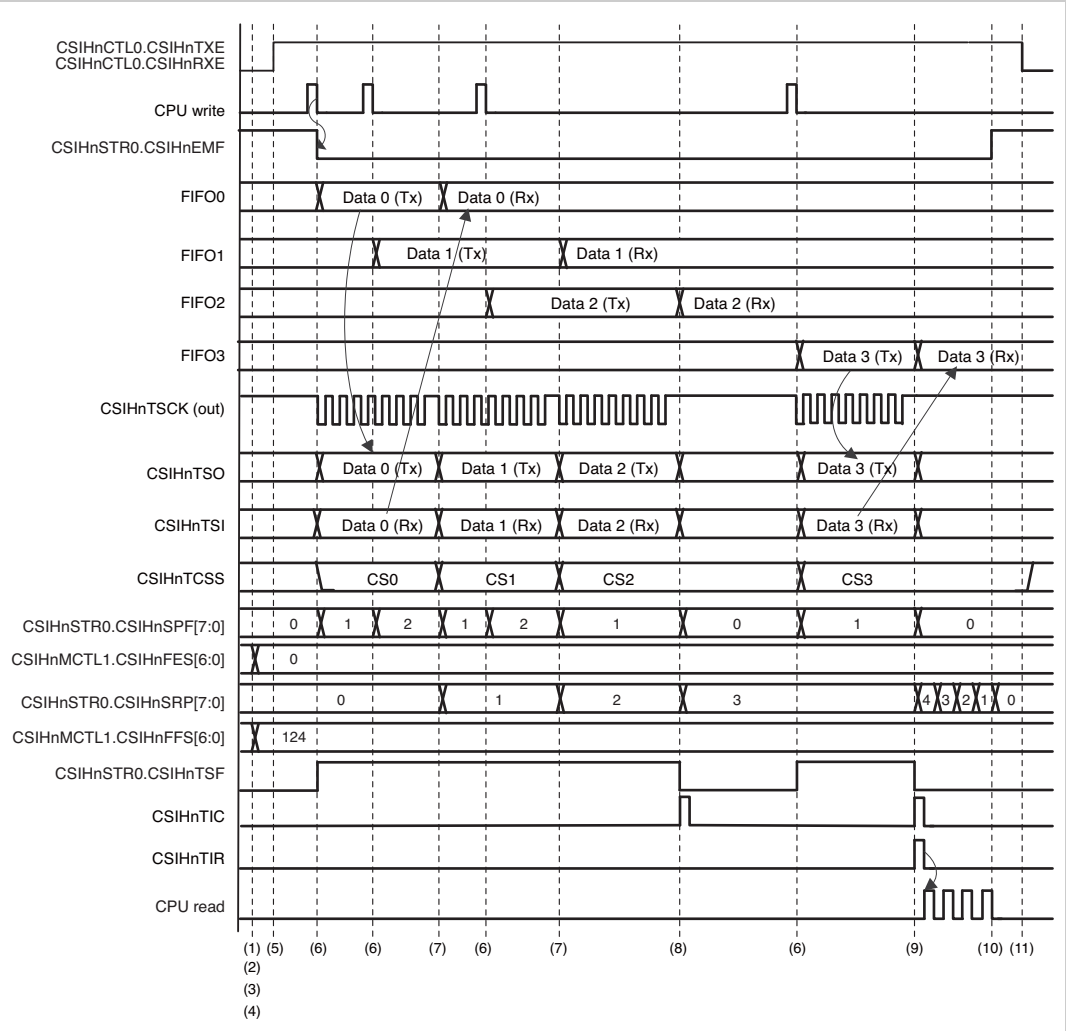


Figure 22-62 FIFO mode (for transmission/reception in the master mode, and when the job mode is disabled)

Procedure:

1. Set up the following registers before setting CSIHnCTL0.CSIHnPWR to 1:
   CSIHnCTL1, CSIHnCTL2 (transfer mode, operating mode)
   CSIHnMCTL0.CSIHnMMS[1:0] = 00_B (memory mode)
   CSIHnCFGx (communication protocol)
   (For this example, the chip select signals CS0 to CS3 are used.)
2. Set CSIHnSTCR0.CSIHnPCT to 1 to clear all the buffer pointers.
3. Make sure that CSIHnSTR0.CSIHnFLF = 0, CSIHnSTR0.CSIHnEMF = 1, and CSIHnSTR0.CSIHnSPF[7:0] = 00_H.
4. Specify the CSIHnTIC interrupt condition for CSIHnMCTL1.CSIHnFES[6:0].
   Specify the CSIHnTIR interrupt condition for CSIHnMCTL1.CSIHnFFS[6:0].
5. CSIHnCTL0.CSIHnPWR = 1 (clock enabled)
   CSIHnCTL0.CSIHnTXE = 1 (transmission enabled)
   CSIHnCTL0.CSIHnRXE = 1 (reception enabled)
   CSIHnCTL0.CSIHnMBS = 0 (memory mode)
6. When transmission data is written to the transmission data register CSIHnTX0W, communication starts.
7. Some of the communication finishes, but CSIHnTIC is not generated because the values of CSIHnSTR0.CSIHnSPF[7:0] and CSIHnMCTL1.CSIHnFES[6:0] do not match.
8. CSIHnTIC is generated because the values of CSIHnSTR0.CSIHnSPF[7:0] and CSIHnMCTL1.CSIHnFES[6:0] match.
9. The interrupt request CSIHnTIR is generated because the values of CSIHnMCTL1.CSIHnFFS[6:0] and (128 – CSIHnSTR0.CSIHnSRP[7:0]) match.
   The interrupt request CSIHnTIC is generated because the values of CSIHnSTR0.CSIHnSPF[7:0] and CSIHnMCTL1.CSIHnFES[6:0] match.
   The CPU starts reading the received data stored in the reception buffer.
10. The CPU finishes reading the received data. The CSIHnSTR0.CSIHnEMF bit is set because the FIFO buffer is empty.
11. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to disable transmission/reception operations. In addition, clear CSIHnCTL0.CSIHnPWR to reduce the power consumption while not using the CSIH.

(2) For reception in the master mode, and when the job mode is disabled

The following conditions are assumed for the procedure shown here:

- Transmission data length: 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B)
- Transmission direction: MSB first (CSIHnCFGx.CSIHnDIRx = 0)
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No delay for any interrupt (CSIHnCTL1.CSIHnSIT = 0)
- Job mode disabled (CSIHnCTL1.CSIHnJE = 0)
- FIFO mode (CSIHnCTL0.CSIHnMBS = 0, CSIHnMCTL0.CSIHnMMS[1:0] = 00_B)
- Normal CSIHnTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)

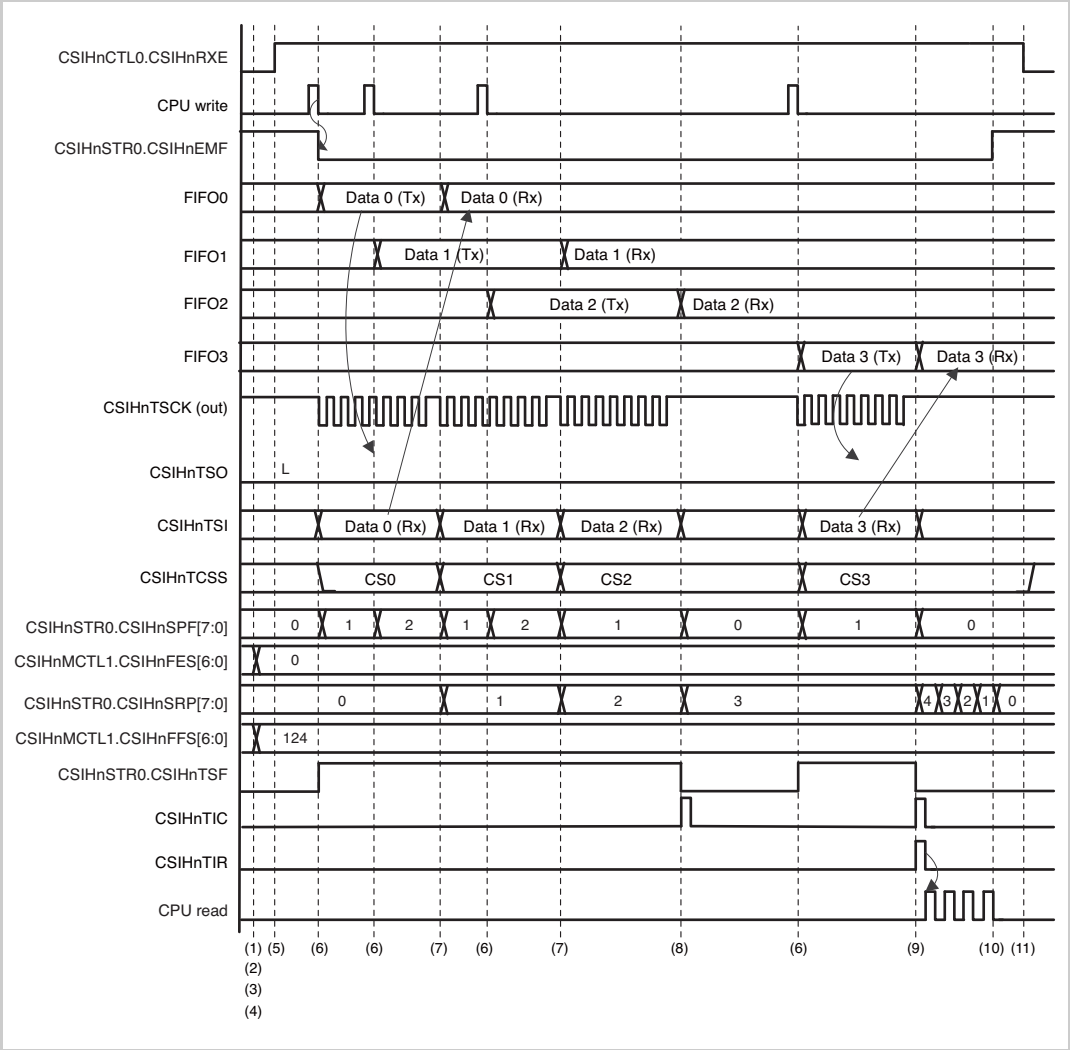


Figure 22-63 FIFO mode (for reception in the master mode, and when the job mode is disabled)

Procedure:

1. Set up the following registers before setting CSIHnCTL0.CSIHnPWR to 1:
   CSIHnCTL1, CSIHnCTL2 (transfer mode, operating mode)
   CSIHnMCTL0.CSIHnMMS[1:0] = 00_B (memory mode)
   CSIHnCFGx (communication protocol)
   (For this example, the chip select signals CS0 to CS3 are used.)
2. Set CSIHnSTCR0.CSIHnPCT to 1 to clear all the buffer pointers.
3. Make sure that CSIHnSTR0.CSIHnFLF = 0, CSIHnSTR0.CSIHnEMF = 1, and CSIHnSTR0.CSIHnSPF[7:0] = 00_H.
4. Specify the CSIHnTIC interrupt condition for CSIHnMCTL1.CSIHnFES[6:0].
   Specify the CSIHnTIR interrupt condition for CSIHnMCTL1.CSIHnFFS[6:0].
5. CSIHnCTL0.CSIHnPWR = 1 (clock enabled)
   CSIHnCTL0.CSIHnTXE = 0 (transmission disabled)
   CSIHnCTL0.CSIHnRXE = 1 (reception enabled)
   CSIHnCTL0.CSIHnMBS = 0 (memory mode)
6. When transmission data is written to the transmission data register CSIHnTX0W, communication starts. (The transmission data is not used, but the chip select signal is enabled.)
7. Some of the communication finishes, but CSIHnTIC is not generated because the values of CSIHnSTR0.CSIHnSPF[7:0] and CSIHnMCTL1.CSIHnFES[6:0] do not match.
8. CSIHnTIC is generated because the values of CSIHnSTR0.CSIHnSPF[7:0] and CSIHnMCTL1.CSIHnFES[6:0] match.
9. The interrupt request CSIHnTIR is generated because the values of CSIHnMCTL1.CSIHnFFS[6:0] and (128 – CSIHnSTR0.CSIHnSRP[7:0]) match.
   The interrupt request CSIHnTIC is generated because the values of CSIHnSTR0.CSIHnSPF[7:0] and CSIHnMCTL1.CSIHnFES[6:0] match.
   The CPU starts reading the received data stored in the reception buffer.
10. The CPU finishes reading the received data. The CSIHnSTR0.CSIHnEMF bit is set because the FIFO buffer is empty.
11. Finally, clear CSIHnCTL0.CSIHnRXE to disable reception operations.
    In addition, clear CSIHnCTL0.CSIHnPWR to reduce the power consumption while not using the CSIH.

(3) For transmission/reception in the slave mode, and when the job mode is disabled

The following conditions are assumed for the procedure shown here:

- Transmission data length: 8 bits (CSIHnCFG0.CSIHnDLS0[3:0] = 1000_B)
- Transmission direction: MSB first (CSIHnCFG0.CSIHnDIR0 = 0)
- Normal clock phase and data phase (CSIHnCFG0.CSIHnCKP0 = 0, CSIHnCFG0.CSIHnDAP0 = 0)
- No delay for any interrupt (CSIHnCTL1.CSIHnSIT = 0)
- Job mode disabled (CSIHnCTL1.CSIHnJE = 0)
- Handshake enabled (CSIHnCTL1.CSIHnHSE = 1)
- FIFO mode (CSIHnCTL0.CSIHnMBS = 0, CSIHnMCTL0.CSIHnMMS[1:0] = 00_B)
- Normal CSIHnTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)

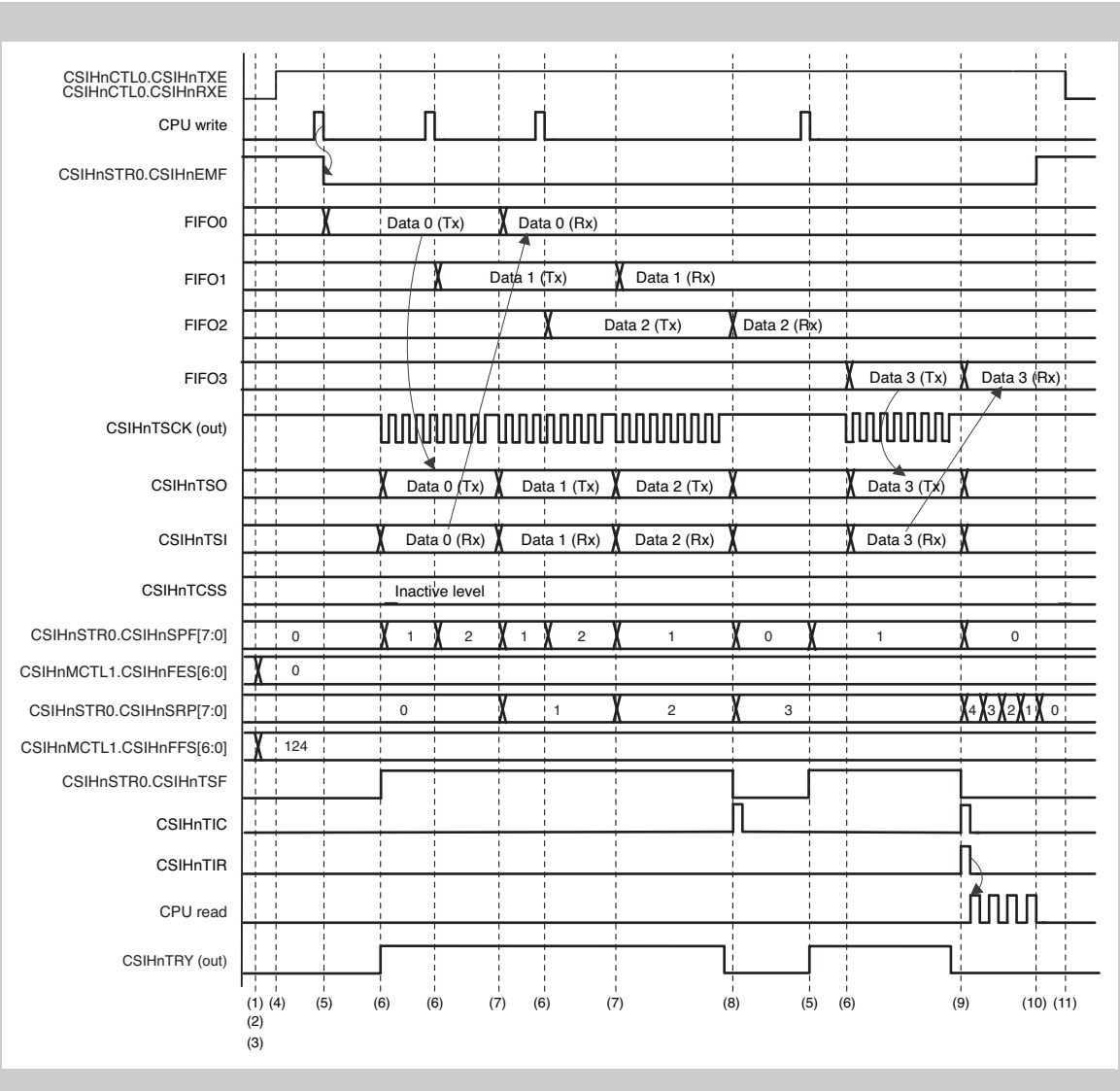


Figure 22-64 FIFO mode (for transmission/reception in the slave mode, and when the job mode is disabled)

Procedure:

1. Set up the following registers before setting CSIHnCTL0.CSIHnPWR to 1:
   CSIHnCTL1, CSIHnCTL2 (transfer mode, operating mode)
   CSIHnMCTL0.CSIHnMMS[1:0] = 00_B (memory mode)
   CSIHnCFG0 (communication protocol)
2. Set CSIHnSTCR0.CSIHnPCT to 1 to clear all the buffer pointers.
   Make sure that CSIHnSTR0.CSIHnFLF = 0, CSIHnSTR0.CSIHnEMF = 1, and CSIHnSTR0.CSIHnSPF[7:0] = 00_H.
3. Specify the CSIHnTIC interrupt condition for CSIHnMCTL1.CSIHnFES[6:0].
   Specify the CSIHnTIR interrupt condition for CSIHnMCTL1.CSIHnFFS[6:0].
4. CSIHnCTL0.CSIHnPWR = 1 (clock enabled)
   CSIHnCTL0.CSIHnTXE = 1 (transmission enabled)
   CSIHnCTL0.CSIHnRXE = 1 (reception enabled)
   CSIHnCTL0.CSIHnMBS = 0 (memory mode)
5. Write the transfer data to the transmission data register CSIHnTX0W. The CSIHnTRY signal is switched from BUSY (low level) to READY (high level) when data is written.
6. When a serial clock is supplied from the master, communication automatically starts.
7. Some of the communication finishes, but CSIHnTIC is not generated because the values of CSIHnSTR0.CSIHnSPF[7:0] and CSIHnMCTL1.CSIHnFES[6:0] do not match.
8. CSIHnTIC is generated because the values of CSIHnSTR0.CSIHnSPF[7:0] and CSIHnMCTL1.CSIHnFES[6:0] match.
9. The interrupt request CSIHnTIR is generated because the values of CSIHnMCTL1.CSIHnFFS[6:0] and (128 – CSIHnSTR0.CSIHnSRP[7:0]) match.
   The interrupt request CSIHnTIC is generated because the values of CSIHnSTR0.CSIHnSPF[7:0] and CSIHnMCTL1.CSIHnFES[6:0] match.
   The CPU starts reading the received data stored in the reception buffer.
10. The CPU finishes reading the received data. The CSIHnSTR0.CSIHnEMF bit is set because the FIFO buffer is empty.
11. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to disable transmission/reception operations. In addition, clear CSIHnCTL0.CSIHnPWR to reduce the power consumption while not using the CSIH.

(4) For reception in the slave mode, and when the job mode is disabled

The following conditions are assumed for the procedure shown here:

- Transmission data length: 8 bits (CSIHnCFG0.CSIHnDLS0[3:0] = 1000_B)
- Transmission direction: MSB first (CSIHnCFG0.CSIHnDIR0 = 0)
- Normal clock phase and data phase (CSIHnCFG0.CSIHnCKP0 = 0, CSIHnCFG0.CSIHnDAP0 = 0)
- No delay for any interrupt (CSIHnCTL1.CSIHnSIT = 0)
- Job mode disabled (CSIHnCTL1.CSIHnJE = 0)
- Handshake enabled (CSIHnCTL1.CSIHnHSE = 1)
- FIFO mode (CSIHnCTL0.CSIHnMBS = 0, CSIHnMCTL0.CSIHnMMS[1:0] = 00_B)
- Normal CSIHnTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)

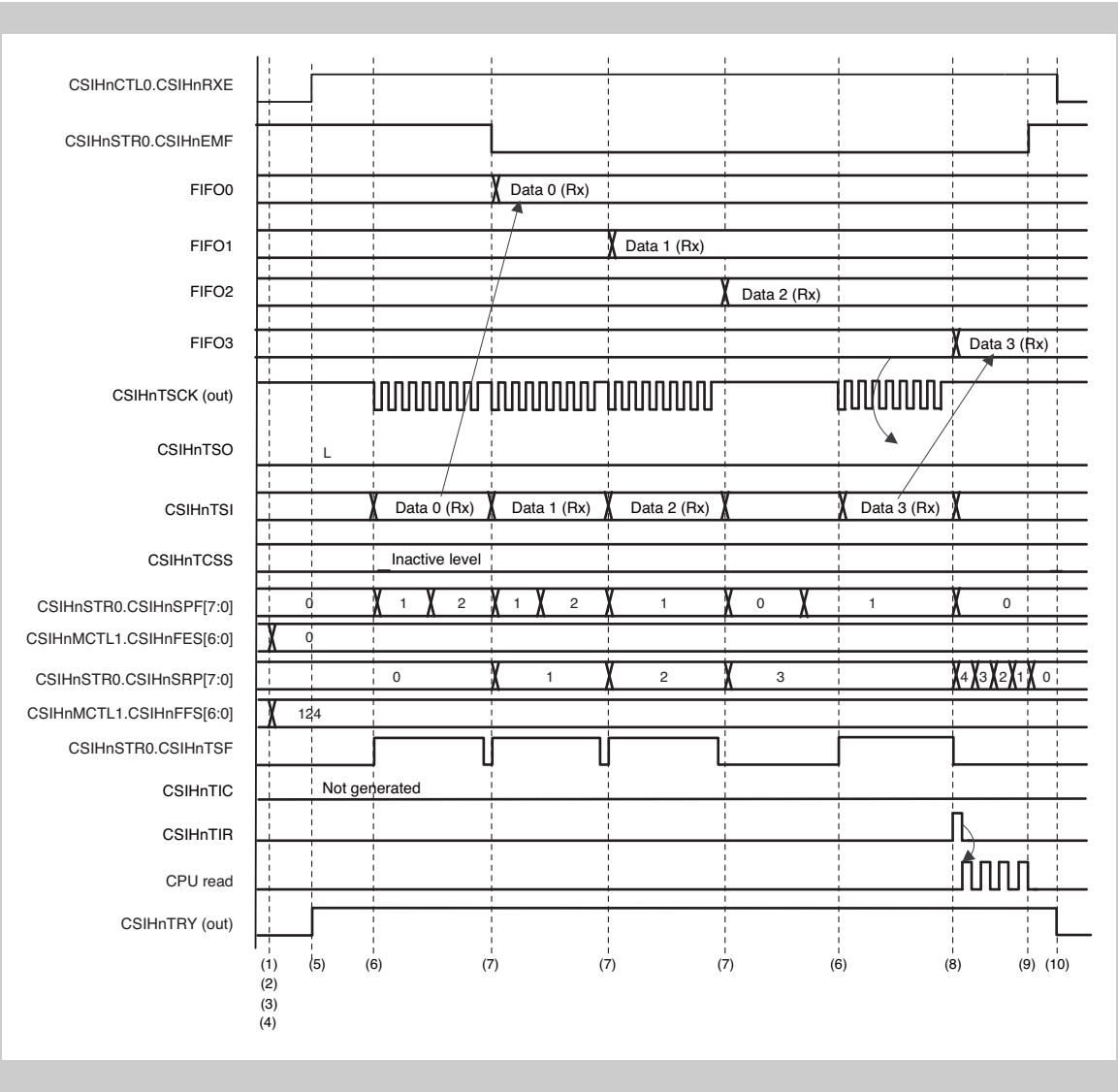


Figure 22-65 FIFO mode (for reception in the slave mode, and when the job mode is disabled)

Procedure:

1. Set up the following registers before setting CSIHnCTL0.CSIHnPWR to 1:
   CSIHnCTL1, CSIHnCTL2 (transfer mode, operating mode)
   CSIHnMCTL0.CSIHnMMS[1:0] = 00_B (memory mode)
   CSIHnCFG0 (communication protocol)
2. Set CSIHnSTCR0.CSIHnPCT to 1 to clear all the buffer pointers.
3. Make sure that CSIHnSTR0.CSIHnFLF = 0, CSIHnSTR0.CSIHnEMF = 1, and CSIHnSTR0.CSIHnSPF[7:0] = 00_H.
4. Specify the CSIHnTIR interrupt condition for CSIHnMCTL1.CSIHnFFS[6:0].
5. CSIHnCTL0.CSIHnPWR = 1 (clock enabled)
   CSIHnCTL0.CSIHnTXE = 0 (transmission disabled)
   CSIHnCTL0.CSIHnRXE = 1 (reception enabled)
   CSIHnCTL0.CSIHnMBS = 0 (memory mode)
   The CSIHnTRY signal is switched from BUSY (low level) to READY (high level) by clearing CSIHnCTL0.CSIHnPWR to 0 and setting CSIHnCTL0.CSIHnRXE to 1.
6. When a serial clock is supplied from the master, reception automatically starts.
7. Some of the communication finishes, but CSIHnTIC is not generated because the system is in the reception mode.
8. The interrupt request CSIHnTIR is generated because the values of CSIHnMCTL1.CSIHnFFS[6:0] and (128 – CSIHnSTR0.CSIHnSRP[7:0]) match.
   The CPU starts reading the received data stored in the reception buffer.
9. The CPU finishes reading the received data. The CSIHnSTR0.CSIHnEMF bit is set because the FIFO buffer is empty.
10. Finally, clear CSIHnCTL0.CSIHnRXE to disable reception operations.
    In addition, clear CSIHnCTL0.CSIHnPWR to reduce the power consumption while not using the CSIH.

(5) For transmission/reception in the master mode, and when the job mode is enabled

The following conditions are assumed for the procedure shown here:

- Transmission data length: 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B)
- Transmission direction: MSB first (CSIHnCFGx.CSIHnDIRx = 0)
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No delay for any interrupt (CSIHnCTL1.CSIHnSIT = 0)
- Job mode is enabled (CSIHnCTL1.CSIHnJE = 1)
- FIFO mode (CSIHnCTL0.CSIHnMBS = 0, CSIHnMCTL0.CSIHnMMS[1:0] = 00_B)
- Normal CSIHnTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)
- Job 1 = four data items, job 2 = three data items, and job 3 = five data items

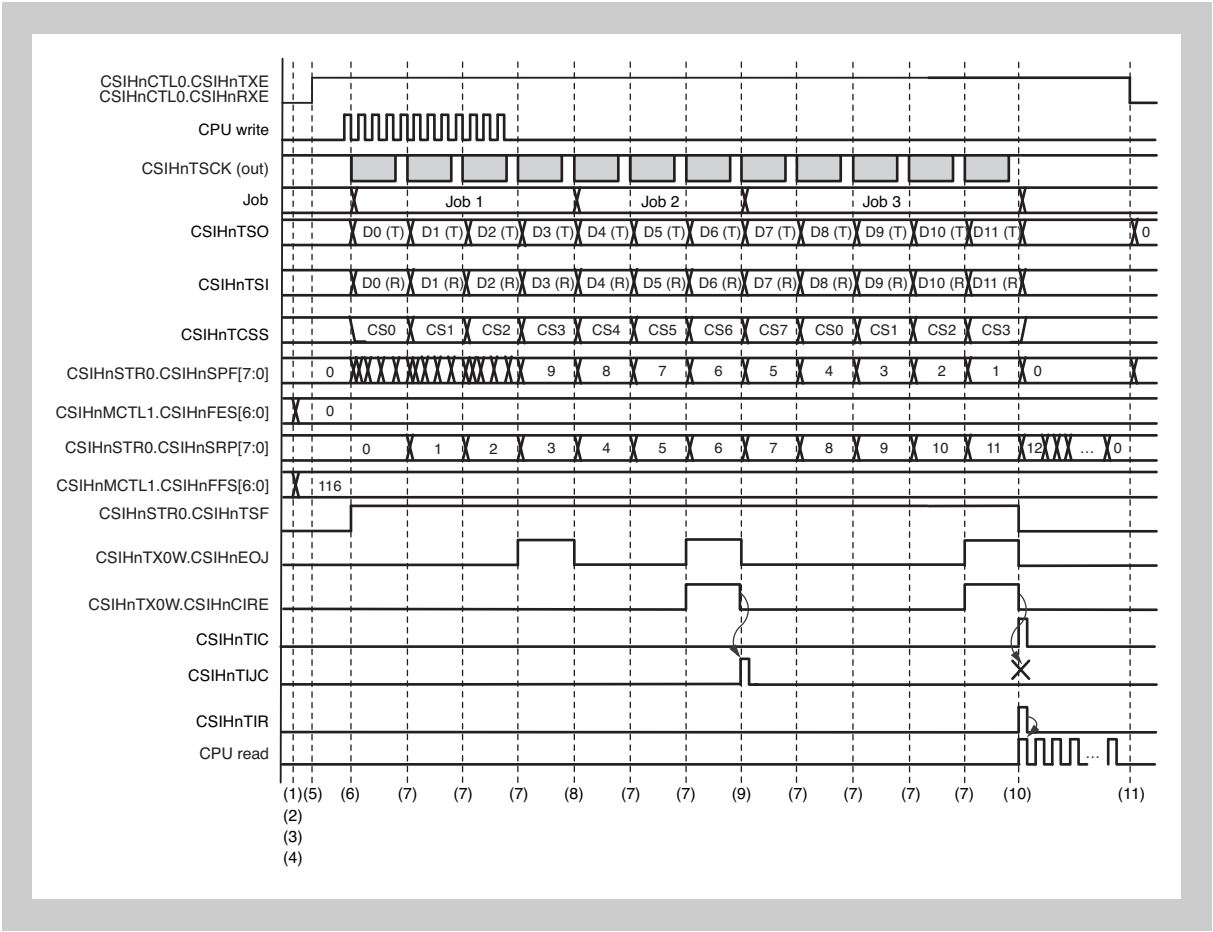


Figure 22-66 FIFO mode (for transmission/reception in the master mode, and when the job mode is enabled)

Note The int_JOBE signal in the above timing chart is the internal signal of the CSIHnCTL0.CSIHnJOBE bit.

Procedure:

1. Set up the following registers before setting CSIHnCTL0.CSIHnPWR to 1:
   CSIHnCTL1, CSIHnCTL2 (transfer mode, operating mode)
   CSIHnMCTL0.CSIHnMMS[1:0] = 00_B (memory mode)
   CSIHnCFGx (communication protocol)
   (For this example, the chip select signals CS0 to CS7 are used.)
2. Set CSIHnSTCR0.CSIHnPCT to 1 to clear all the buffer pointers.
3. Make sure that CSIHnSTR0.CSIHnFLF = 0, CSIHnSTR0.CSIHnEMF = 1, and CSIHnSTR0.CSIHnSPF[7:0] = 00_H.
4. Specify the CSIHnTIC interrupt condition for CSIHnMCTL1.CSIHnFES[6:0].
   Specify the CSIHnTIR interrupt condition for CSIHnMCTL1.CSIHnFFS[6:0].
5. CSIHnCTL0.CSIHnPWR = 1 (clock enabled)
   CSIHnCTL0.CSIHnTXE = 1 (transmission enabled)
   CSIHnCTL0.CSIHnRXE = 1 (reception enabled)
   CSIHnCTL0.CSIHnMBS = 0 (memory mode)
6. When transmission data is written to the transmission data register CSIHnTX0W, communication starts.
7. Some of the communication finishes, but CSIHnTIC is not generated because the values of CSIHnSTR0.CSIHnSPF[7:0] and CSIHnMCTL1.CSIHnFES[6:0] do not match.
8. Because the last data (CSIHnTX0W.CSIHnEOJ = 1) of the current job was transmitted by clearing CSIHnTX0W.CSIHnCIRE, the interrupt request CSIHnTIC is not generated.
9. Because the last data (CSIHnTX0W.CSIHnEOJ = 1) of the current job was transmitted by setting CSIHnTX0W.CSIHnCIRE, the interrupt request CSIHnTIC is generated.
10. CSIHnTIC is generated because the values of CSIHnSTR0.CSIHnSPF[7:0] and CSIHnMCTL1.CSIHnFES[6:0] match.
    Because CSIHnTIC was generated, CSIHnTIJC is not generated.
    The interrupt request CSIHnTIR is generated because the values of CSIHnMCTL1.CSIHnFFS[6:0] and (128 – CSIHnSTR0.CSIHnSRP[7:0]) match.
    The CPU starts reading the received data stored in the reception buffer.
11. Finally, clear CSIHnCTL0.CSIHnTXE and CSIHnCTL0.CSIHnRXE to disable transmission/reception operations. In addition, clear CSIHnCTL0.CSIHnPWR to reduce the power consumption while not using the CSIH.

(6) For reception in the master mode, and when the job mode is enabled

The following conditions are assumed for the procedure shown here:

- Transmission data length: 8 bits (CSIHnCFGx.CSIHnDLSx[3:0] = 1000_B)
- Transmission direction: MSB first (CSIHnCFGx.CSIHnDIRx = 0)
- Normal clock phase and data phase (CSIHnCFGx.CSIHnCKPx = 0, CSIHnCFGx.CSIHnDAPx = 0)
- No delay for any interrupt (CSIHnCTL1.CSIHnSIT = 0)
- Job mode enabled (CSIHnCTL1.CSIHnJE = 1)
- FIFO mode (CSIHnCTL0.CSIHnMBS = 0, CSIHnMCTL0.CSIHnMMS[1:0] = 00_B)
- Normal CSIHnTIC interrupt timing (CSIHnCTL1.CSIHnSLIT = 0)
- Job 1 = four data items, job 2 = three data items, and job 3 = five data items

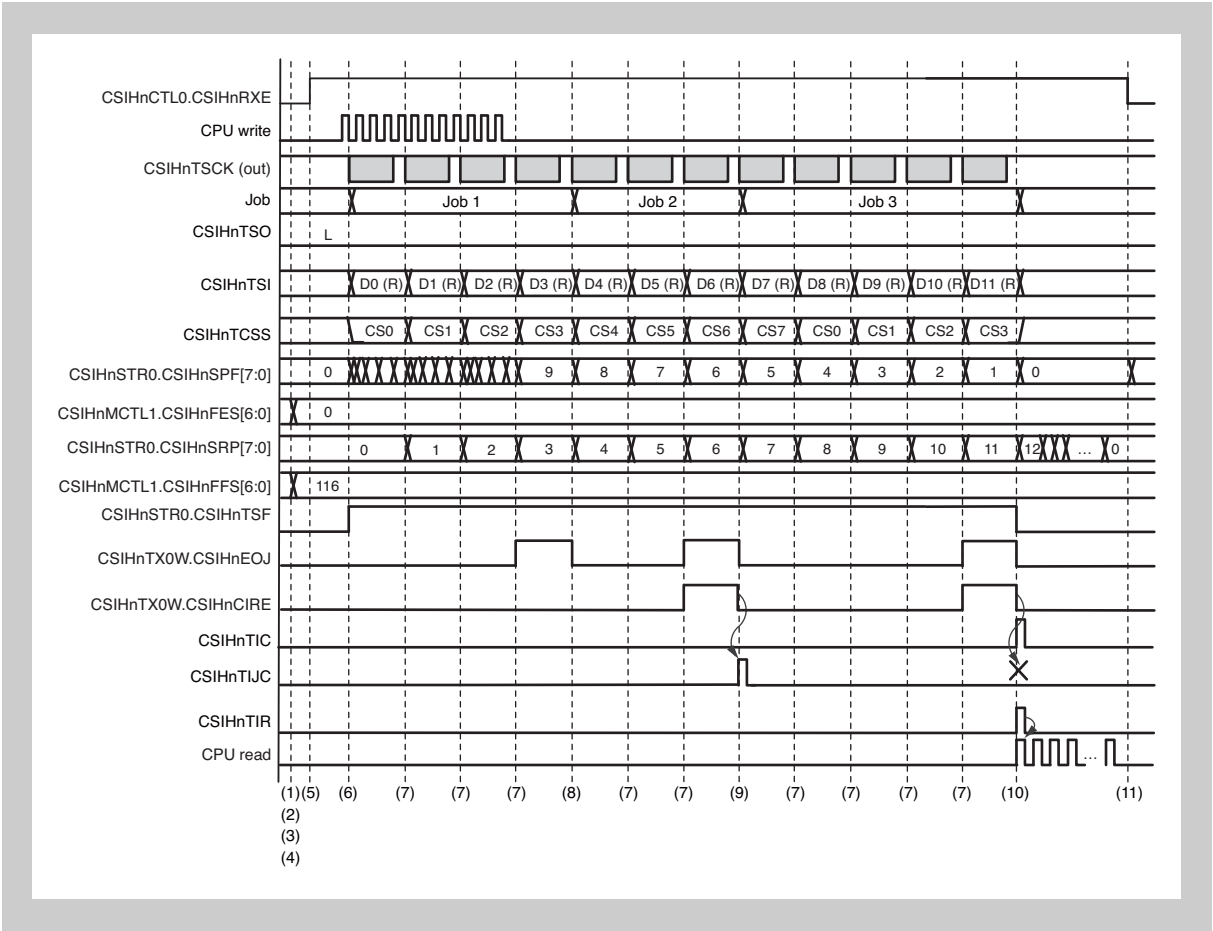


Figure 22-67 FIFO mode (for reception in the master mode, and when the job mode is enabled)

Note The int_JOBE signal in the above timing chart is the internal signal of the CSIHnCTL0.CSIHnJOBE bit.

Procedure:

1. Set up the following registers before setting CSIHnCTL0.CSIHnPWR to 1:
   CSIHnCTL1, CSIHnCTL2 (transfer mode, operating mode)
   CSIHnMCTL0.CSIHnMMS[1:0] = 00_B (memory mode)
   CSIHnCFGx (communication protocol)
   (For this example, the chip select signals CS0 to CS7 are used.)
2. Set CSIHnSTCR0.CSIHnPCT to 1 to clear all the buffer pointers.
3. Make sure that CSIHnSTR0.CSIHnFLF = 0, CSIHnSTR0.CSIHnEMF = 1, and CSIHnSTR0.CSIHnSPF[7:0] = 00_H.
4. Specify the CSIHnTIC interrupt condition for CSIHnMCTL1.CSIHnFES[6:0].
   Specify the CSIHnTIR interrupt condition for CSIHnMCTL1.CSIHnFFS[6:0].
5. CSIHnCTL0.CSIHnPWR = 1 (clock enabled)
   CSIHnCTL0.CSIHnTXE = 1 (transmission enabled)
   CSIHnCTL0.CSIHnRXE = 1 (reception enabled)
   CSIHnCTL0.CSIHnMBS = 0 (memory mode)
6. When transmission data is written to the transmission data register CSIHnTX0W, communication starts. (The transmission data is not used, but the chip select signal is enabled.)
7. Some of the reception finishes, but CSIHnTIC is not generated because the values of CSIHnSTR0.CSIHnSPF[7:0] and CSIHnMCTL1.CSIHnFES[6:0] do not match.
8. Because the last data (CSIHnTX0W.CSIHnEOJ = 1) of the current job was transmitted by clearing CSIHnTX0W.CSIHnCIRE, the interrupt request CSIHnTIC is not generated.
9. Because the last data (CSIHnTX0W.CSIHnEOJ = 1) of the current job was transmitted by setting CSIHnTX0W.CSIHnCIRE, the interrupt request CSIHnTIC is generated.
10. CSIHnTIC is generated because the values of CSIHnSTR0.CSIHnSPF[7:0] and CSIHnMCTL1.CSIHnFES[6:0] match.
    Because CSIHnTIC was generated, CSIHnTIJC is not generated.
    The interrupt request CSIHnTIR is generated because the values of CSIHnMCTL1.CSIHnFFS[6:0] and (128 – CSIHnSTR0.CSIHnSRP[7:0]) match.
    The CPU starts reading the received data stored in the reception buffer.
11. Finally, clear CSIHnCTL0.CSIHnRXE to disable reception operations.
    In addition, clear CSIHnCTL0.CSIHnPWR to reduce the power consumption while not using the CSIH.

Chapter 23 I2C BUS (IICB)

Caution To use the IIC bus function, use the SCLn and SDAn pins, and set them to N-ch open drain.

This chapter describes the I2C bus (IICB).

The first section describes all properties specific to the V850E2/MN4, such as instances, register base addresses, input/output signal names.
The subsequent sections describe the features that apply to all implementations.

23.1 V850E2/MN4 IICB Features

Instances This microcontroller has the following number of instances of the IICB.

Table 23-1 Instances of IICB

IICB	
Instances	6
Names	IICB to IICB5

Instances index n Throughout this chapter, the individual instances of the IICB are identified by the index “n” (n = 5), for example, IICBnDAT for the IICBn data register.

Register addresses All IICBn register addresses are given as address offsets from the individual base addresses <IICBn_base>.
The base address <IICBn_base> of each IICBn is listed in the following table:

Table 23-2 Register base addresses <IICBn_base>

IICBn	IICBn instance	<IICBn_base>
IICB0	<IICBn_base_OS>	FF82 0000H
	<IICBn_base_USER>	FFFF F400H
IICB1	<IICBn_base_OS>	FF82 1000H
	<IICBn_base_USER>	FFFF F500H
IICB2	<IICBn_base_OS>	FF82 2000H
	<IICBn_base_USER>	FFFF F600H
IICB3	<IICBn_base_OS>	FF82 3000H
	<IICBn_base_USER>	FFFF F700H
IICB4	<IICBn_base_OS>	FF82 4000H
	<IICBn_base_USER>	FFFF F800H
IICB5	<IICBn_base_OS>	FF82 5000H
	<IICBn_base_USER>	FFFF F900H

Clock supply

The IICBn uses PCLK as the clock input. PCLK is connected to the clock generator.

Table 23-3 IICBn clock supply

IICBn instance	Clock	Connected to:
IICB0 to IICB5	PCLK	f_{PCLK}

Interrupt request signal

The IICBn uses a data transmit/receive interrupt request signal (IICBTIAn) and a status interrupt request signal (IICBTISn).

Table 23-4 IICBn interrupt request signals

Interrupt request signal	Function	Connected to
IICB0		
IICBTIA	Data transmit/receive interrupt request signal	• Interrupt controller 144 (INTCSIH0IR) • DMA controller trigger 108 • DTS controller trigger 108
IICBTIS	Status interrupt request signal	• Interrupt controller 143 (INTCSIH0IRE)
IICB1		
IICBTIA	Data transmit/receive interrupt request signal	• Interrupt controller 148 (INTCSIH1IR) • DMA controller trigger 110 • DTS controller trigger 111
IICBTIS	Status interrupt request signal	• Interrupt controller 147 (INTCSIH1IRE)
IICB2		
IICBTIA	Data transmit/receive interrupt request signal	• Interrupt controller 152 (INTCSIH2IR) • DMA controller trigger 112 • DTS controller trigger 114
IICBTIS	Status interrupt request signal	• Interrupt controller 151 (INTCSIH2IRE)
IICB3		
IICBTIA	Data transmit/receive interrupt request signal	• Interrupt controller 156 (INTCSIH3IR) • DMA controller trigger 114 • DTS controller trigger 117
IICBTIS	Status interrupt request signal	• Interrupt controller 155 (INTCSIH3IRE)
IICB4		
IICBTIA	Data transmit/receive interrupt request signal	• Interrupt controller 172 (INTCSIG4IR) • DMA controller trigger 124
IICBTIS	Status interrupt request signal	• Interrupt controller 171 (INTCSIG4IRE)
IICB5		
IICBTIA	Data transmit/receive interrupt request signal	• Interrupt controller 175 (INTCSIG5IR) • DMA controller trigger 126
IICBTIS	Status interrupt request signal	• Interrupt controller 174 (INTCSIG5IRE)

23.2 Functional Overview

Operating mode	Standard mode (SCL clock frequency: 100 kHz max.) Fast mode (SCL clock frequency: 400 kHz max.)
Transfer mode	Single transfer mode Continuous transfer mode
Pin configuration	SCLn: Serial clock pin SDAn: Serial transmit/receive data pin
Interrupt request signal	Data transmit/receive interrupt request signal (IICBTIAn) Status interrupt request signal (IICBTISn)
Communication data length	8 bits
Multimaster support	Multiple masters can control the bus simultaneously.
SCLn level width	The high-level width and low-level width of the serial clock signal (SCLn) can be changed.
Automatic detection	The start and stop conditions can be detected automatically.

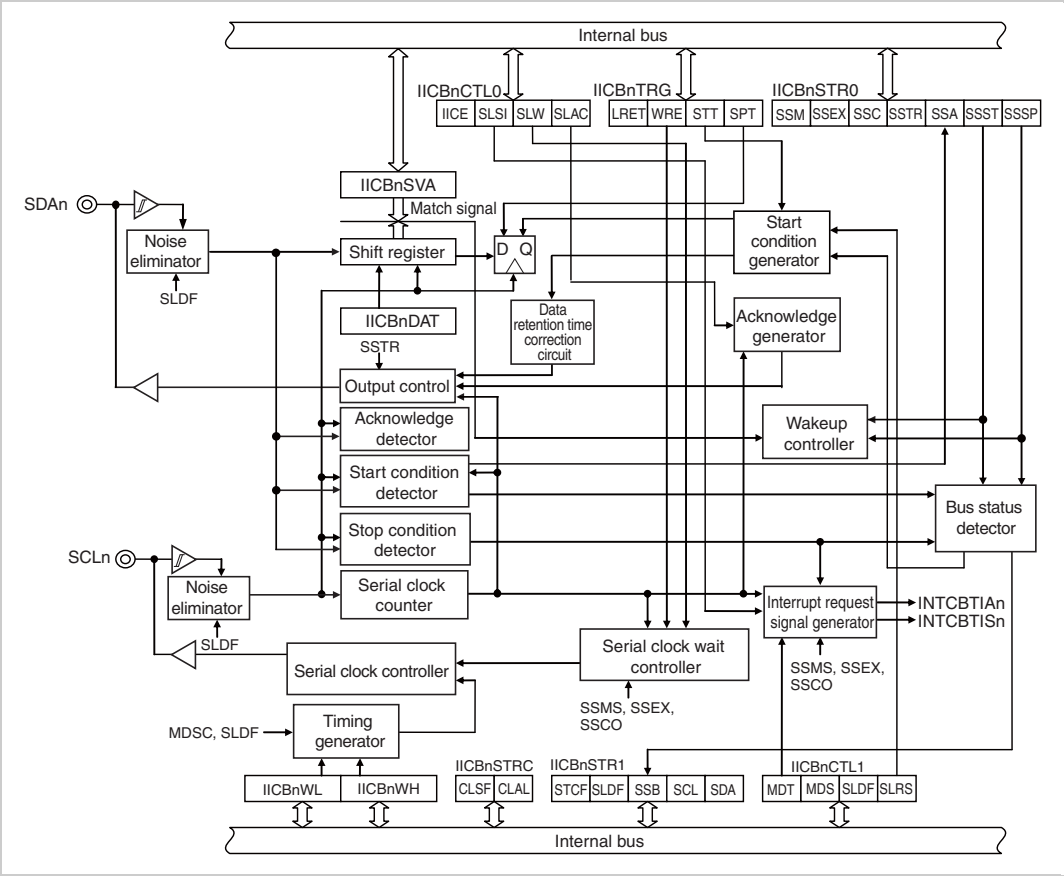


Figure 23-1 Block diagram of IICBn

23.3 IIC Bus Mode Functions

23.3.1 Pin configuration

The serial clock pin (SCLn) and serial data bus pin (SDAn) are configured as follows.

SCLn... This pin is used for serial clock input and output.
This pin is an N-ch open-drain output for both master and slave devices. Input is Schmitt input.

SDAn... This pin is used for serial data input and output.
This pin is an N-ch open-drain output for both master and slave devices. Input is Schmitt input.

Because the outputs of the serial clock line and serial data bus line are N-ch open-drain outputs, an external pull-up resistor must be connected to these lines.

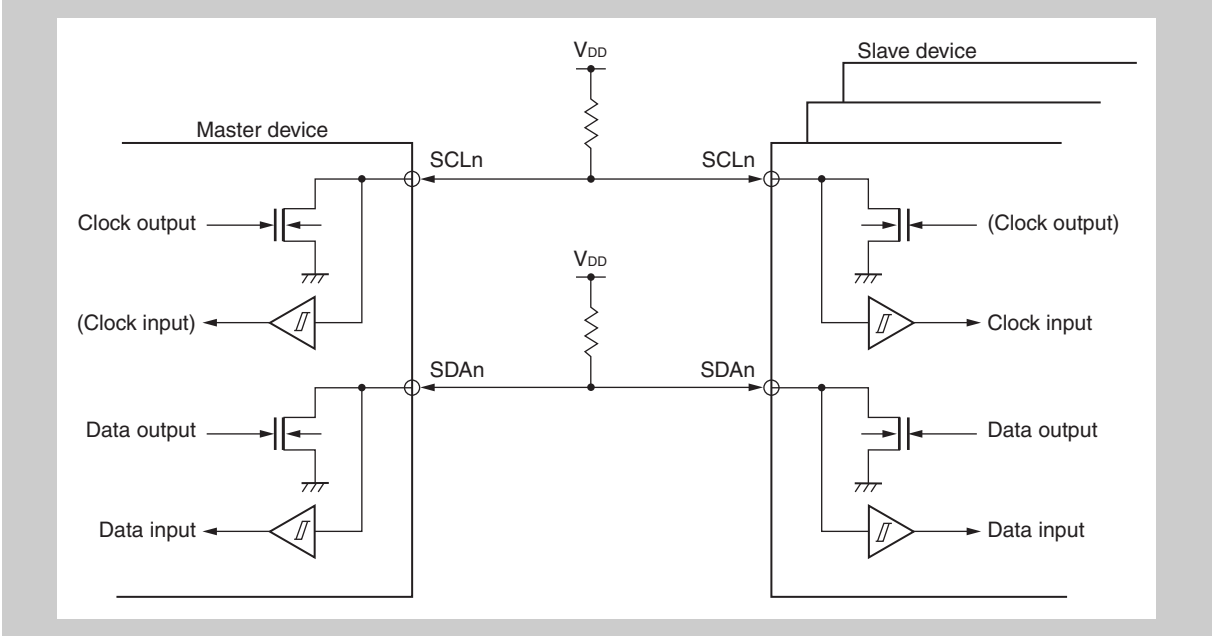


Figure 23-2 Pin configuration diagram

23.4 IIC Bus Definition

This section describes the IIC bus's serial data communication format and the signals used by the IIC bus.

Figure 23-3 "IIC bus serial data transfer timing" shows the transfer timing for the "start condition", "address", "transfer direction specification", "data", and "stop condition", which are output onto the IIC bus's serial data bus.

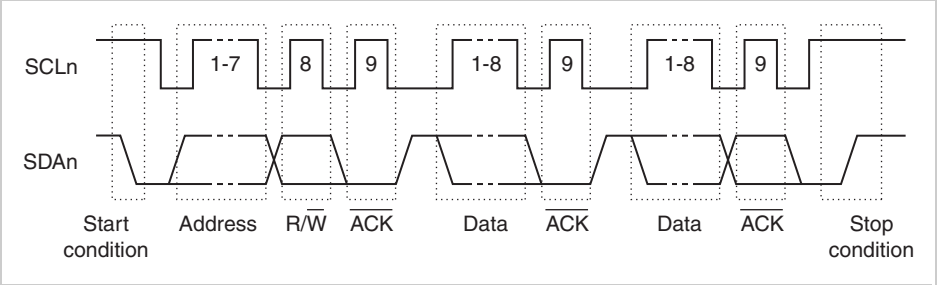


Figure 23-3 IIC bus serial data transfer timing

The start condition, slave address, and stop condition are output by the master device.

$\overline{\text{ACK}}$ can be output by either the master or slave device. (Normally, it is output by the device that receives 8-bit data.)

The serial clock signal (SCLn) is continuously output by the master device. In the slave device, the low-level period of the SCLn signal can be extended to insert a wait.

23.4.1 Start Condition

The start condition is met if the SDAn signal level changes from high to low while the SCLn signal is high. The start condition is output when the master device starts serial data transfer to a slave device. When the IICBn is in the slave mode, it detects the start condition.

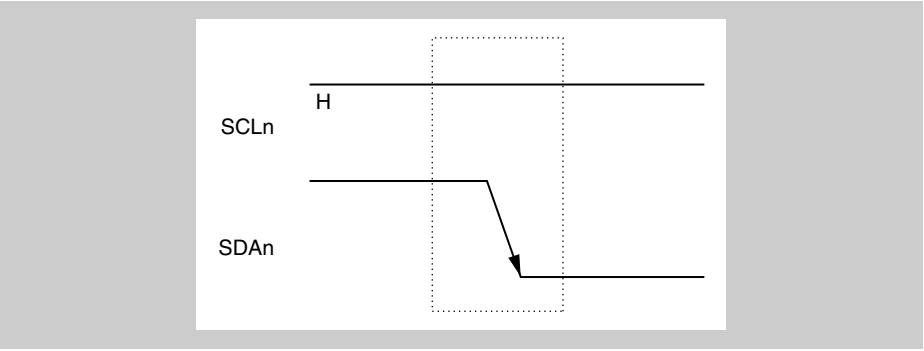


Figure 23-4 Start condition

23.4.2 Addresses

The 7 bits of data following the start condition are defined as an address.

An address is a 7-bit data segment that is output in order to select one of the slave devices that are connected to the master device via the bus lines. Therefore, each slave device connected via the bus lines must have a unique address.

The slave device checks whether the 7-bit data matches its own address. If they match, that slave device is selected as the communication destination and communicates with the master device until the master device outputs another start condition or a stop condition.

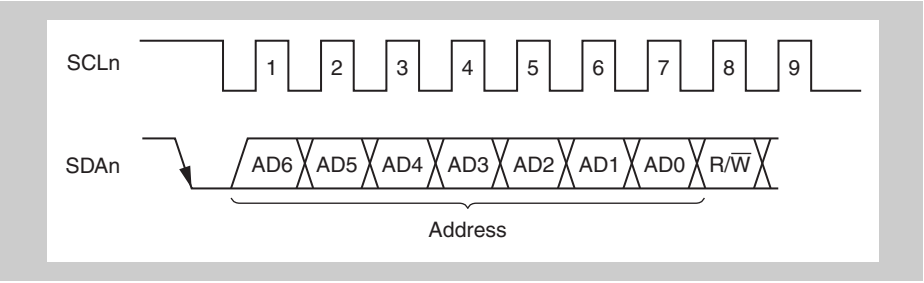


Figure 23-5 Addresses

23.4.3 Extension Code

When the higher 4 bits of the address are 0000 or 1111, these bits are called extension code. *Table 23-5 "Extension code bit definitions"* lists the bit definitions of extension code.

Table 23-5 Extension code bit definitions

Slave address	R/W bit	Description
0000 000	0	General call address
0000 000	1	Start byte
0000 001	×	CBUS address
0000 010	×	Address reserved for different bus format
0000 011	×	Reserved for future use
0000 1xx	×	HS mode master code[a]
1111 0xx	×	10-bit slave address specification
1111 1xx	×	Reserved for future use

a) The HS mode cannot be used for IICB.

23.4.4 Transfer direction specification

After the 7-bit address data, the master device transmits 1 bit that specifies the transfer direction.

If this transfer direction specification bit is 0, it indicates that the master device transmits data to a slave device. If this bit is 1, it indicates that the master device receives data from a slave device.

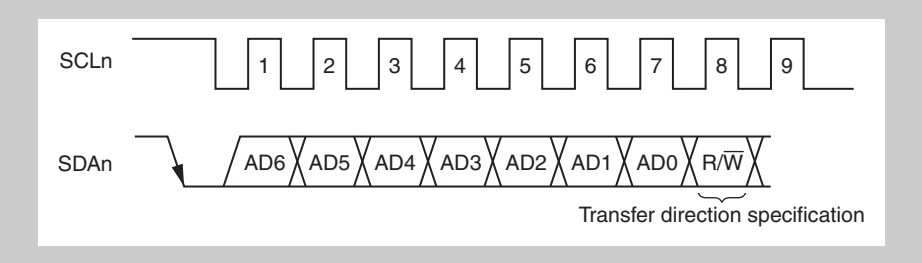


Figure 23-6 Transfer direction specification

23.4.5 Acknowledge ($\overline{ACK}$)

The 1-bit data after the transfer direction bit (R/$\overline{W}$) and the 1-bit data after the 8-bit data during address transfer are defined as an acknowledge signal ($\overline{ACK}$). $\overline{ACK}$ is used to check the serial data status of the transmitting and receiving devices.

The receiving device returns $\overline{ACK}$ after receiving 8-bit data.

The transmitting device normally receives $\overline{ACK}$ after transmitting 8-bit data. If the transmitting device receives $\overline{ACK}$ from the receiving device, it continues processing assuming that the transmitted data is normally received.

If the master device is the receiving device and receives the final data, it does not return $\overline{ACK}$ and outputs a stop condition. If the slave device is the receiving device and does not return $\overline{ACK}$, the master device outputs either a stop condition or a restart condition and then stops the current transmission. Failure to return $\overline{ACK}$ may be caused by the following factors.

(1) The transmitted data has not been received normally.

(2) The final data has been received.

(3) The receiving device (slave) does not exist for the specified address.

$\overline{ACK}$ is output when the SDAn line of the receiving device changes to low level at the 9th clock (normal reception).

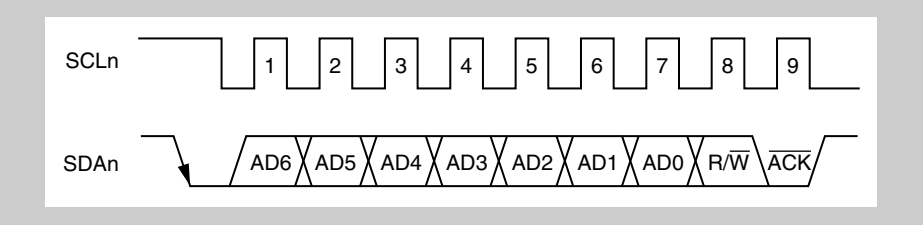


Figure 23-7 Acknowledge ($\overline{ACK}$)

23.4.6 Data

The bits other than the nine bits following the start condition (seven address bits, an R/$\overline{W}$ bit, and an acknowledge ($\overline{ACK}$) bit) and the acknowledge bits are defined as data.

If a 10-bit address is specified using an extension code, the 8-bit data that is transferred after the address is used as the second address.

23.4.7 Stop condition

A stop condition is met if the SDAn signal level changes from low to high while the SCLn signal is high.
The stop condition is output when serial data transfer from the master device to the slave device has been completed.

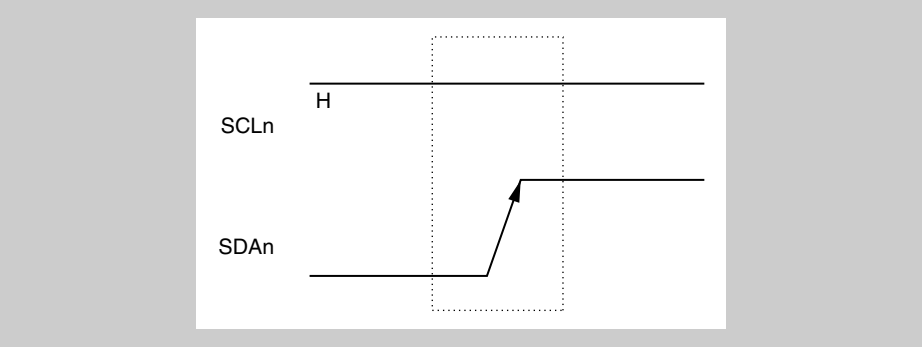


Figure 23-8 Stop condition

23.4.8 Wait state

A wait state is used to report to the communication destination that the IICBn (master or slave) is preparing to transmit or receive data.

The wait state is reported to the communication destination by setting the SCLn signal to low. The next data transfer cannot start until both the master and slave devices exit the wait state.

(a) When a wait at the 9th clock is specified for the master and a wait at the 8th clock for the slave (master: transmission, slave: reception)

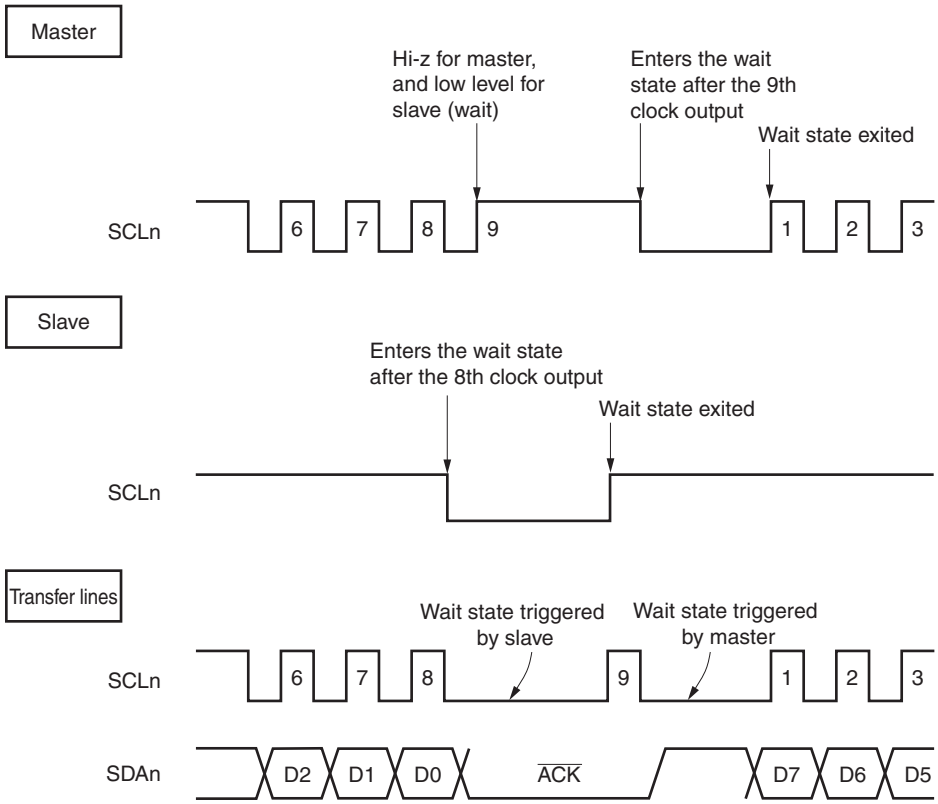


Figure 23-9 Wait state (1/2)

(b) When a wait at the 9th clock is specified for both the master and slave
(master: transmission, slave: reception)

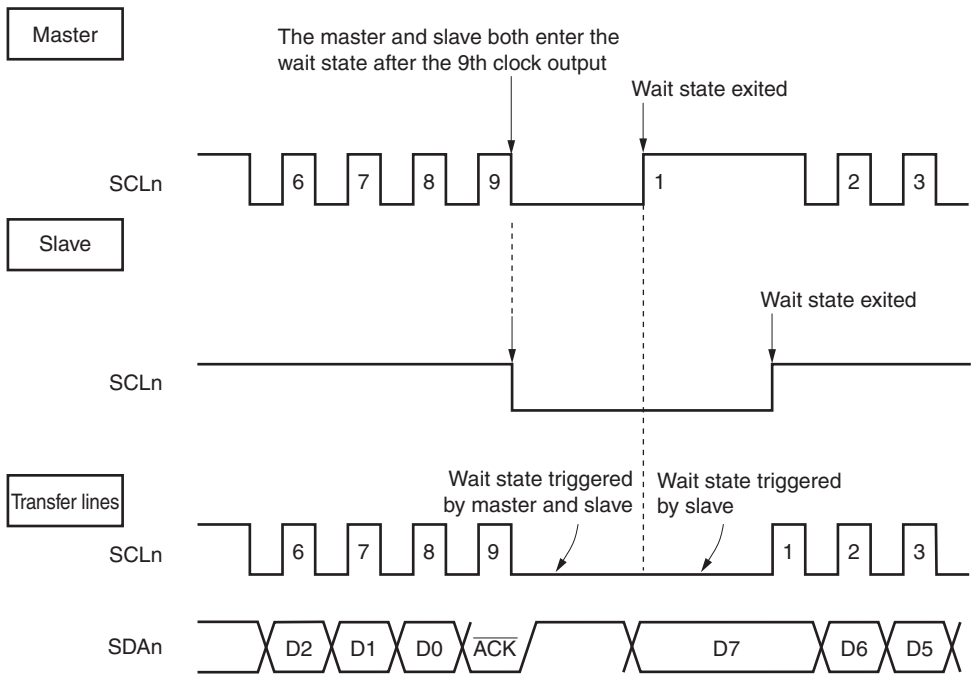


Figure 23-9 Wait state (2/2)

23.4.9 Arbitration

When several master devices simultaneously output a start condition, communication with the master devices continues until the data differs, while adjusting the clocks. An example where two masters simultaneously output a start condition and arbitration is conducted is described below.

This example assumes that one master outputs the SDAn line high (master 1) and the other master outputs the SDAn line low (master 2) while the SCLn line is low.
In this case, the communication with master 2 is prioritized, and communication is not authorized for master 1.

This kind of operation is called arbitration, and the state in which communication is not authorized is called arbitration loss. The master that lost arbitration releases the bus by setting both the SCLn and SDAn line to high impedance.

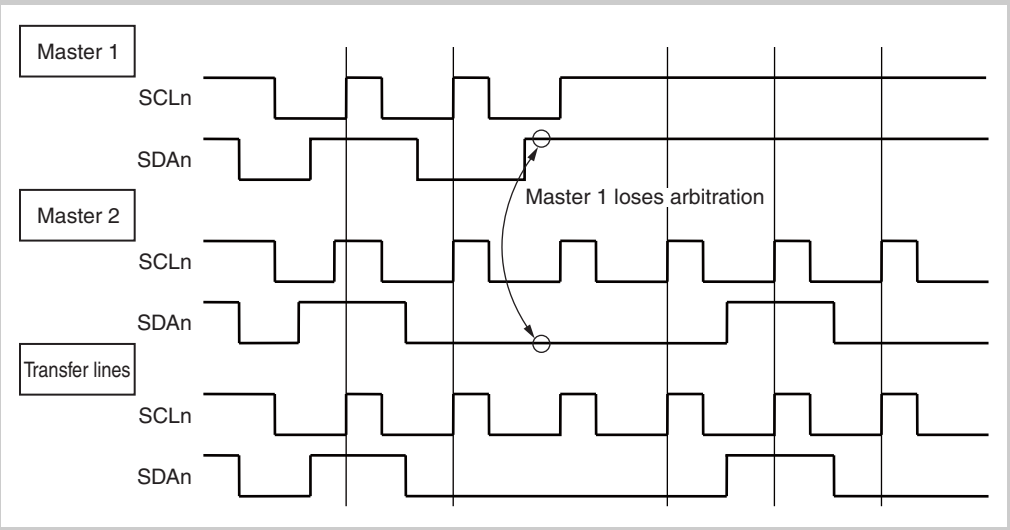


Figure 23-10 Arbitration timing example

23.5 Registers

Caution In this section, the operation when an extension code is received is omitted. For details about the extension code, refer to *23.6.5 "Extension code"*.

(1) IICBnDAT – IICBn data register

This register is used to transmit and receive transfer data.

Access This register can be read/written in 8-bit units.

Address <IICBn_base_USER> + 0000_H

Initial Value 00_H. This register is initialized by any reset. This register is also initialized by changing the value of the IICBnCTL0.IICBnIICE bit from 1 to 0 or from 0 to 1.

Cautions

1. When the IICBn becomes a master in the single transfer mode or continuous transfer mode, after the IICBnTRG.IICBnSTT bit has been set to 1, writing to the IICBnDAT register is allowed only once to transfer the address and communication direction.
2. When transferring data in the single transfer mode, writing to the IICBnDAT register in communication state other than the wait state is prohibited.
3. When transferring data in the continuous transfer mode, writing to the IICBnDAT register in response to an INTIICBTIAn interrupt request signal is only allowed once.
4. When executing transmission operations in the continuous transfer mode, do not read the IICBnDAT register.
   Similarly, when performing reception operations in the continuous transfer mode, do not write to the IICBnDAT register.

7	6	5	4	3	2	1	0
IICBnDAT7	IICBnDAT6	IICBnDAT5	IICBnDAT4	IICBnDAT3	IICBnDAT2	IICBnDAT1	IICBnDAT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23-6 IICBnDAT register contents

Bit position	Bit name	Function
7 to 0	IICBnDAT[7:0]	During reception, these bits hold the received data. During transmission, these bits write the transmit data. The prescribed procedure must be followed during access (read, write) to the IICBnDAT register. For the setting sequence, refer to *23.9 "Setting Sequence"*. The IICBn exits the wait state by performing access to the IICBnDAT register. • In single transfer mode - When write access to the IICBnDAT register is performed • In continuous transfer mode - When write access to the IICBnDAT register is performed - When read access to the IICBnDAT register is performed during a wait state for data transfer that is not triggered by NACK signal reception

(2) IICBnSVA – IICBn slave address register

This register stores the slave address of the IICBn bus.

Access This register can be read/written in 8-bit units.

Address <IICBn_base_USER> + 0004_H

Initial Value 00_H. This register is initialized by any reset.

Caution Write access to the IICBnSVA register is prohibited when the value of the IICBnCTL0.IICBnIICE bit is 1.

7	6	5	4	3	2	1	0
IICBnSVA7	IICBnSVA6	IICBnSVA5	IICBnSVA4	IICBnSVA3	IICBnSVA2	IICBnSVA1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Table 23-7 IICBnSVA register contents

Bit position	Bit name	Function
7 to 1	IICBnSVA[7:1]	Store the slave address of the IICBn bus. Address match/address mismatch is judged by comparing the received address and the IICBnSVA register. If the received address matches the IICBnSVA register, the IICBnSTR0.IICBnSSCO bit is set to 1.

(3) IICBnCTL0 – IICBn control register 0

This register is used to control the operations of the IICBn.

Access This register can be read/written in 8- or 1-bit units.

Address <IICBn_base_USER> + 0008_H

Initial Value 00_H. This register is initialized by any reset.

7	6	5	4	3	2	1	0
IICBn IICE	0	0	IICBn MDTX1	IICBn MDTX0	IICBn SLSI	IICBn SLWT	IICBn SLAC
R/W	R	R	R/W	R/W	R/W	R/W	R/W

Table 23-8 IICBnCTL0 register contents (1/3)

Bit position	Bit name	Function
7	IICBnIICE	Enables/disables operation of the IICBn. 0: Enables operation of IICBn. 1: Disables operation of IICBn. Synchronous reset of the following registers is executed when the value of the IICBnCTL0.IICBnIICE bit changes from 1 to 0, or the value of the IICBnCTL0.IICBnIICE bit changes from 0 to 1. • IICBnDAT and IICBnSTR0 registers When IICBnCTL0.IICBnIICE is 0, the SCLn and SDAn pins go into the high impedance state.
4	IICBnMDTX1	Specifies the transfer mode upon detection of expansion code in the slave. 0: Single transfer mode 1: Continuous transfer mode • Single transfer mode The IICBn enters a wait state after each transfer according to the setting of the IICBnCTL0.IICBnSLWT bit. • Continuous transfer mode The IICBn performs continuous communication without entering a wait state when the IICBnDAT register is read or written upon the output of the data transmit/receive interrupt request signal (INTIICBTIAn). For the operation in each mode, refer to *23.6 "Operation"*. **Caution** Rewrite is allowed only when IICBnCTL0.IICBnIICE is 0.
3	IICBnMDTX0	Specifies the transfer mode when the address matches between the master and slave. 0: Single transfer mode 1: Continuous transfer mode • Single transfer mode The IICBn enters a wait state after each transfer according to the setting of the IICBnCTL0.IICBnSLWT bit. • Continuous transfer mode The IICBn performs continuous communication without entering a wait state when the IICBnDAT register is read or written upon the output of the data transmit/receive interrupt request signal (INTIICBTIAn). For the operation in each mode, refer to *23.6 "Operation"*. **Caution** Rewrite is allowed only when IICBnCTL0.IICBnIICE is 0.

Table 23-8 IICBnCTL0 register contents (2/3)

Bit position	Bit name	Function
2	IICBnSLSI	Enables/disables status interrupt request signal (INTIICBTISn) output when a stop condition is detected. 0: Disables INTIICBTISn signal output when stop condition is detected. 1: Enables INTIICBTISn signal output when stop condition is detected. Set this bit to 1 when performing the following types of communication. - When the IICBn performs communication as a master while the communication reserve function is enabled - When the IICBn participates in communications as a slave - When the IICBn may lose in arbitration (when making the IICBn operate as a master in a multi-master environment))
1	IICBnSLWT	Controls a wait and interrupt request output timing. 0: The IICBn enters the wait state and an interrupt request is output at the falling edge of the 8th clock during single transfer. 1: The IICBn enters the wait state and an interrupt request is output at the falling edge of the 9th clock during single transfer. The IICBnCTL0.IICBnSLWT bit controls wait state transition and interrupt request output at the following timing. - 8th and 9th clocks during data transfer For the conditions for transition to the wait state, refer to *23.6.4 "Entering and exiting wait state"*. During address transfer, the conditions for transiting to the wait state and for interrupt request output are as follows, regardless of the setting of the IICBnCTL0.IICBnSLWT bit. • In single transfer mode - Master: A data transmit/receive interrupt request signal (INTIICBTIAn) is output and the IICBn enters the wait state upon detection of the falling edge of the 9th clock. - Slave: During an address match, the INTIICBTIAn signal is output and the IICBn enters the wait state upon detection of the falling edge of the 9th clock. During address mis-match, the INTIICBTIAn signal is not output and the IICBn does not enter the wait state. • In continuous transfer mode In the continuous transfer mode, transition to the wait state is not affected by the setting of the IICBnCTL0.IICBnSLWT bit. - Reception: The IICBn enters the wait state at the falling edge of the 8th clock. - Transmission: The IICBn enters the wait state at the falling edge of the 9th clock. **Caution** During single transfer mode, rewriting this bit is allowed only when IICBnCTL0.IICBnIICE is 0 or while in a wait period.

Table 23-8 IICBnCTL0 register contents (3/3)

Bit position	Bit name	Function
0	IICBnSLAC	Controls acknowledge signal output. 0: Disables acknowledge signal output. Master:The acknowledge signal is not output during data reception (SDAn = “H”). Slave:The acknowledge signal is not output during data transfer when an address match occurs (SDAn = “H”). 1: Enables acknowledge signal output. Master:The acknowledge signal is output during data reception (SDAn = “L”). Slave:The acknowledge signal is output during data transfer when an address match occurs (SDAn =“L”). When the IICBn is operating as a slave, in the case of an address match, no acknowledge signal is output during address transfer regardless of the value of the IICBnCTL0.IICBnSLAC bit (SDAn = “L”). also, no acknowledge signal is output (SDAn = “H”) while the IICBn is transmitting data or when it does not participate in communications.

(4) IICBnCTL1 – IICBn control register 1

This register controls the operation of the IICBn.

Access This register can be read/written in 8-bit units.

Address <IICBn_base_OS> + 0020_H

Initial Value 00_H. This register is initialized by any reset.

Caution Write access to the IICBnCTL1 register is prohibited when the value of the IICBnCTL0.IICBnIICE is 1.

7	6	5	4	3	2	1	0
IICBn MDSC	IICBn LGDF2	IICBn LGDF1	IICBn LGDF0	IICBn MDLB	0	IICBn SLSE	IICBn SLRS
R/W	R/W	R/W	R/W	R/W	R	R/W	R/W

Table 23-9 IICBnCTL1 register contents (1/2)

Bit position	Bit name	Function
7	IICBnMDSC	Specifies the operation mode for the IICBn. 0: Standard mode (SCL clock frequency: 100 kHz max.) 1: Fast mode (SCL clock frequency: 400 kHz max.)
6 to 4	IICBnLGDF[2:0]	Specify the digital filter sampling frequency. Note that the digital filter can be used only in the fast mode. 000:Does not use digital filter. SCLn and SDAn are used without passing through the digital filter in the IICBn. The digital filter circuit operations are stopped. Other than above:Uses digital filter. SCLn and SDAn are used passing through the digital filter in the IICBn. When using a digital filter, set bits IICBnCTL1.IICBnLGDF[2:0] as follows.

IICBnCTL1.IICBnLGDF[2:0] bits	Frequency
001	Minimum frequency [a] ≤ PCLK ≤ 20 MHz
010	20 MHz < PCLK ≤ 40 MHz
011	40 MHz < PCLK ≤ 60 MHz
100	60 MHz < PCLK ≤ 80 MHz
101	80 MHz < PCLK ≤ 100 MHz
110, 111	Setting prohibited

a) A list of the minimum frequencies by setting is shown below.

Operation mode (IICBnCTL1.IICBnMDSC)	When digital filter used (IICBnCTL1.IICBnLGDF[2:0] bits are 000)	When digital filter used (IICBnCTL1.IICBnLGDF[2:0] bits are not 000)
Standard mode (0)	1.0 MHz	Use prohibited
Fast mode (1)	3.5 MHz	4.0 MHz

Table 23-9 IICBnCTL1 register contents (2/2)

Bit position	Bit name	Function
3	IICBnMDLB	Specifies the loop back mode. 0: Do not loop back. 1: Loop back. By setting the IICBnCTL1.IICBnMDLB bit, the output serial clock signal (SCLn) and serial transmit/receive data signal (SDAn) are looped back and used as the input serial clock signal (SCLn) and input serial transmit/receive data signal (SDAn). The output SCLn and SDAn immediately before output will be looped back. Note that both SCLn and SDAn are high level if the IICBnCTL1.IICBnMDLB bit is “1”.
1	IICBnSLSE	Enables/disables start condition output in the initial communication state. 0: Disables start condition output in the initial communication state. 1: Enables start condition output in the initial communication state. If the IICBnCTL1.IICBnSLSE bit is set to 1, a start condition can be output by setting the IICBnTRG.IICBnSTT bit to 1 in the initial communication state (from when the IICBnCTL0.IICBnIICE bit is set to 1 until detection of a stop condition). The IICBnCTL1.IICBnSLSE bit is automatically cleared to 0 upon detection of a start condition (even without a 0 write operation). **Caution** Clear the IICBnCTL1.IICBnSLSE bit to 0 when participating in communications after other communications have started. When other communications are being performed, if the IICBnTRG.IICBnSTT bit has been set to 1 with the IICBnCTL1.IICBnSLSE bit set to 1, the other communications may be damaged.
0	IICBnSLRS	Enables/disables the communication reserve function. 0: Enables communication reserve function. 1: Disables communication reserve function. Communication reserve function enabled state: If the IICBnCTL1.IICBnSLRS bit is cleared to 0 while the IICBn is not operating as a master, the communication reserve state can be set by setting the IICBnTRG.IICBnSTT bit to 1 while the bus is being used. Whether the communication reserve state is set can be confirmed by checking the IICBnSTR0.IICBnSSRS bit. Communication reserve function disabled state: If the IICBnTRG.IICBnSTT bit is set to 1 while the IICBn is not participating in communications as a master and the bus is being used, the value of the IICBnSTR0.IICBnSTCF becomes 1 and communication reservation is not done.

(5) IICBnWL – IICBn low level width setting register

This register is used to set the low level width of the serial clock register (SCLn).

Access This register can be read/written in 16-bit units.

Address <IICBn_base_OS> + 0024_H

Initial Value $03FF_H$. This register is initialized by any reset.

Caution Write access to the IICBnWL register is prohibited when the value of the IICBnCTL0.IICBnIICE bit is 1.

15	14	13	12	11	10	9	8
0	0	0	0	0	0	IICBnWL9	IICBnWL8
R	R	R	R	R	R	R/W	R/W

7	6	5	4	3	2	1	0
IICBnWL7	IICBnWL6	IICBnWL5	IICBnWL4	IICBnWL3	IICBnWL2	IICBnWL1	IICBnWL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23-10 IICBnWL register contents

Bit position	Bit name	Function
9 to 0	IICBnWL[9:0]	Specify the t_{LOW} period (low level width of the SCLn clock) of the I^2C bus specification. The value of the IICBnWL register is used to determine the serial output timing of other I^2C bus specifications. For the serial output timing setting conditions, refer to *Table 23-12 "Conditions for generating serial output timing" on page 1445* .

(6) IICBnWH – IICBn high-level width setting register

This register is used to set the high level width of the serial clock signal (SCLn).

Access This register can be read/written in 16-bit units.

Address <IICBn_base_OS> + 0028_H

Initial Value $03FF_H$. This register is initialized by any reset.

Caution Write access to the IICBnWH register is prohibited when the value of the IICBnCTL0.IICBnIICE bit is 1.

15	14	13	12	11	10	9	8
0	0	0	0	0	0	IICBnWH9	IICBnWH8
R	R	R	R	R	R	R/W	R/W

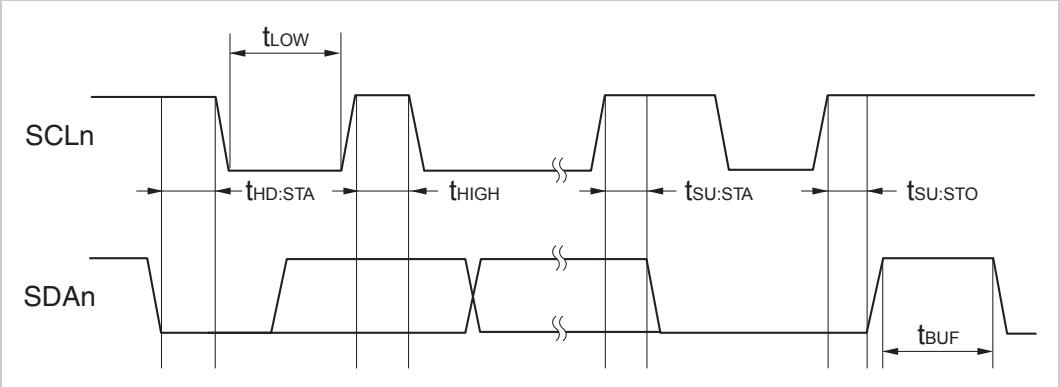
7	6	5	4	3	2	1	0
IICBnWH7	IICBnWH6	IICBnWH5	IICBnWH4	IICBnWH3	IICBnWH2	IICBnWH1	IICBnWH0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 23-11 IICBnWH register contents

Bit position	Bit name	Function
9 to 0	IICBnWH[9:0]	Specify the t_{HIGH} period (high level width of the SCLn clock) of the I^2C bus specification. The value of the IICBnWH register is used to determine the serial output timing of other I^2C bus specifications. For the serial output timing setting conditions, refer to *Table 23-12 "Conditions for generating serial output timing" on page 1445* .

Table 23-12 Conditions for generating serial output timing

Symbol	Description	Standard mode	Fast mode
$t_{HD:STA}$	Start condition hold time	$\frac{IICB0WH}{PCLK}$	$\frac{IICB0WH}{PCLK}$
t_{LOW}	SCL low-level width period	$\frac{IICB0WL}{PCLK}$	$\frac{IICB0WL}{PCLK}$
t_{HIGH}	SCL high-level width period	$\frac{IICB0WH}{PCLK}$	$\frac{IICB0WH}{PCLK}$
$t_{SU:STA}$	Start condition setup time	$\frac{IICB0WL}{PCLK}$	$\frac{IICB0WL}{PCLK}$
$t_{SU:STO}$	Stop condition setup time	$\frac{IICB0WH}{PCLK}$	$\frac{IICB0WH}{PCLK}$
t_{BUF}	Bus free time (interval between stop condition and start condition)	$\frac{IICB0WL}{PCLK}$	$\frac{IICB0WL}{PCLK}$
$t_{HD:DAT}$	Data hold time	$\frac{IICB0WL[9:2]}{PCLK}$	$\frac{IICB0WL[9:2]}{PCLK}$



(a) Setting transfer clock by using IICBnWL and IICBnWH registers

The various timings in compliance with the I^2C bus specifications can be set by setting the IICBnWL register and IICBnWH register.

- Setting transfer clock on master side

$$\text{Transfer clock (Hz)} = \frac{\text{PCLK}}{(\text{IICBnWL} + \text{IICBnWH} + \text{PCLK}\ (t_R + t_F)}$$

At this time, the optimal setting values of IICBnWL and IICBnWH are as follows.
(The fractional parts of all setting values are rounded up.)

– In the fast mode

$$\text{IICBnWL} = \frac{0.52}{\text{Transfer clock}} \times \text{PCLK}$$

$$\text{IICBnWH} = \left(\frac{0.48}{\text{Transfer clock}} - t_R - t_F\right) \times \text{PCLK}$$

– In the standard mode

$$\text{IICBnWL} = \frac{0.47}{\text{Transfer clock}} \times \text{PCLK}$$

$$\text{IICBnWH} = \left(\frac{0.53}{\text{Transfer clock}} - t_R - t_F\right) \times \text{PCLK}$$

Caution The data hold time must be 0.9 μs or less in the fast mode and 3.45 μs or less in the standard mode.

Note The data hold time is determined by the IICBWL register setting as follows:

Data hold time = IICBnWL.IICBnWL[9:2] / PCLK

- Setting IICBnWL and IICBnWH on slave side
  (The fractional parts of all setting values are rounded up.)
  - In the fast mode

    IICBnWL = 1.3 μs × PCLK

    IICBnWH = (1.2 μs − t_R − t_F) × PCLK
  - In the standard mode

    IICBnWL = 4.7 μs × PCLK

    IICBnWH = (5.3 μs − t_R − t_F) × PCLK

Note IICBnWL: IICBn low-level width setting register
IICBnWH: IICBn high-level width setting register
t_F: SDAn and SCLn signal falling times
t_R: SDAn and SCLn signal rising times
PCLK: Frequency of the clock supplied to the IICBn
f_{CLK}: SCL clock frequency

(7) IICBnTRG – IICBn trigger register

This register is used to set the IICBn trigger.

Access This register can be read/written in 8- or 1-bit units.

Address <IICBn_base_USER> + $000C_H$

Initial Value 00_H. This register is initialized by any reset.

7	6	5	4	3	2	1	0
0	0	0	0	IICBn LRET	IICBn WRET	IICBn STT	IICBn SPT
R	R	R	R	R/W	R/W	R/W	R/W

Table 23-13 IICBnTRG register contents (1/4)

Bit position	Bit name	Function
3	IICBnLRET	Communication exit trigger bit 0: The read value is always 0, and writing 0 is ignored. 1: The IICBn exits the current communication and enters the wait state. This bit is automatically cleared to 0 following execution. The following occurs when IICBnTRG.IICBnLRET is 1. - SCLn and SDAn each go into high impedance (communication wait state). - Bits IICBnSSMS, IICBnSSDR, IICBnSSWT, IICBnSSEX, IICBnSSC0, IICBnSSTR, IICBnSSAC, IICBnSSRS, and IICBnSSST of the IICBnSTR0 register are cleared to 0. - When IICBnTRG.IICBnSTT = 1 (start condition output preparation) or IICBnTRG.IICBnSPT =1 (stop condition output preparation) has been set, output of a start condition or stop condition is stopped. The communication reserved state is released if the IICBn exits the communication in the communication reserved state. If it is necessary for the IICBn to operate a master again after this, the IICBnTRG.IICBnSTT bit must be set to 1 again. **Caution** If IICBnTRG.IICBnLRET is set to 1 during master operation (IICBnSTR0.IICBnSSMS = 1), the bus is released. Since serial clock output stops, problems occur during communication on the slave side.
2	IICBnWRET	This is the trigger bit for exiting the wait state. 0: Does not exit the wait state. 1: Exits the wait state and resumes communication. This bit is automatically cleared following execution. If the IICBn have exited the wait state by setting the IICBnTRG.IICBnWRET bit to 1 during the wait state triggered by the falling edge of the 9th clock, the IICBnSTR0.IICBnSSTR bit is cleared to 0 and SDAn goes into high impedance (this enables the external master to output a stop condition or start condition.) If the IICBn is not in the wait state (IICBnSTR0.IICBnSSWT = 0), setting this bit to 1 has no meaning. There are other conditions for exiting the wait state in addition to the setting of this bit. For details, refer to *23.6.4 "Entering and exiting wait state"*.

Table 23-13 IICBnTRG register contents (2/4)

Bit position	Bit name	Function
1	IICBnSTT	Start condition trigger bit 0: Does not output a start condition. 1: Outputs a start condition (This bit is automatically cleared to 0 after it has been set to 1. The IICBnTRG.IICBnSTT bit can be set to 1 under the following conditions: [1] IICBnSTR0.IICBnSSMS bit = Master state (1) • Single transfer mode - During wait state triggered by the falling edge of the 9th clock (both address transfer and data transfer) - During data reception, only after clearing the IICBnCTL0.IICBnSLAC bit to 0 to report the end of reception to the slave • Continuous transfer mode - During wait state triggered by the falling edge of the 9th clock of address transfer - During data transfer - During data reception, only after clearing the IICBnCTL0.IICBnSLAC bit to 0 to report the end of reception to the slave In the case of the wait period during the 9th clock, following wait cancellation, and in all other cases, upon detecting the falling edge of the 9th clock, SDAn and SCLn are set to the high level after the low-level width period of the SCLn clock, and then, when SDAn is set to the low level after waiting for the start condition setup time to elapse, a start condition is output. Next, SCLn is set to the low level after the start condition hold time has elapsed. For each time, see *Table 23-12 "Conditions for generating serial output timing" on page 1445* . [2] Slave state or communication wait state (IICBnSTR0.IICBnSSMS = 0) • IICBnSTR0.IICBnSSBS bit = 0 (bus release state) After the bus free time elapses, a start condition is output when SDAn is changed from the high level to the low level while SCLn is high level. (At this time, SCLn outputs a high level signal.) Next, SCLn is set to the low level after the start condition hold time has elapsed. For each time, see *Table 23-12 "Conditions for generating serial output timing" on page 1445* . • IICBnSTR0.IICBnSSBS bit = 1 (bus communication state) This status indicates that communication is performed on the bus while the IICBn is not operating as a master. - When communication reserve function is enabled (IICBnCTL1.IICBnSLRS bit = 0): A start condition is output after the bus has been released (the stop condition has been detected) and the bus free time has elapsed. However, even if the bus free time has not elapsed, upon detecting a start condition, SDAn is immediately set to the low level without waiting for the bus free time to elapse. For each time, see *Table 23-12 "Conditions for generating serial output timing" on page 1445* . - When communication reserve function is disabled (IICBnCTL1.IICBnSLRS bit = 1): The IICBnSTR0.IICBnSTCF bit is set to 1 and a start condition is not output.

Table 23-13 IICBnTRG register contents (3/4)

Bit position	Bit name	Function
1	IICBnSTT	**Caution** [2] shows the operations according to the value of the IICBnSTR0.IICBnSSBS bit when the IICBnTRG.IICBnSTT bit is 0. Even if the IICBnTRG.IICBnSTT bit is set to 1 after checking the value of the IICBnSTR0.IICBnSSBS bit through register read, the value of IICBnSTR0.IICBnSSBS may differ from its value when it was checked. The output processing of the start condition is started by setting the IICBnTRG.IICBnSTT bit to 1, but upon detection of the following states, output processing of the start condition is stopped and the start condition is not output. - When 0 is written to the IICBnCTL0.IICBnIICE bit - When 1 is written to the IICBnTRG.IICBnLRET bit - Upon detection of arbitration loss - When 1 is written to the IICBnTRG.IICBnSPT bit after 1 is written to the IICBnTRG.IICBnSTT bit while the IICBn is operating as a master in the continuous transfer mode - When 1 is written to the IICBnTRG.IICBnSTT and IICBnTRG.IICBnSPT bits during the same data transfer period while the IICBn is operating as a master in the continuous transfer mode (In this case, writing 1 to the IICBnTRG.IICBnSTT bit is enabled.) **Cautions** 1. When start in the initial communication state is enabled (IICBnCTL1.IICBnSLSE bit = 1), the start condition is output regardless of the bus status when the IICBnTRG.IICBnSTT bit is set to 1. If other communications are performed at that time, they may be damaged. 2. Setting the IICBnTRG.IICBnSTT bit at the same time as the IICBnTRG.IICBnSPT bit is prohibited.
0	IICBnSPT	Stop condition trigger bit 0: Does not output a stop condition. 1: Outputs a stop condition (This bit is automatically cleared after it has been set to 1). The IICBnTRG.IICBnSPT bit can be set to 1 under the following conditions while the IICBn is performing communication as a master. • Single transfer mode - Wait state triggered by the falling edge of the 9th clock (both address transfer and data transfer) - During data reception, only after clearing the IICBnCTL0.IICBnSLAC bit to 0 to report the end of reception to the slave • Continuous transfer mode The IICBnTRG.IICBnSPT bit can be set to 1 in the following states. - During the wait state triggered by the falling edge of the 9th clock of address transfer - During data transfer - Detection of a NACK signal (IICBnSTR0.IICBnSSAC bit = 0) during the wait state triggered by the falling edge of the 9th clock for during data reception A stop condition can be output with the following procedure. (If the IICBn is in the wait state, after exiting the wait state) SCLn is released when SDAn has output a low level, and SCLn = high level, SDAn is low level are waited for. Then, following the lapse of the $t_{SU:STO}$ time, a stop condition is output by setting SDAn to high level.

Table 23-13 IICBnTRG register contents (4/4)

Bit position	Bit name	Function
0	IICBnSPT	The output processing of the stop condition is started by setting the IICBnTRG.IICBnSPT bit to 1, but upon detection of the following states, output processing of the stop condition is stopped and the stop condition is not output. - When 0 is written to the IICBnCTL0.IICBnIICE bit - When 1 is written to the IICBnTRG.IICBnLRET bit - Upon detection of a stop condition - Upon detection of arbitration loss - When 1 is written to the IICBnTRG.IICBnSTT bit after IICBnTRG.IICBnSPT has been set to 1 while the IICBn is operating as a master in the continuous transfer mode **Cautions 1.** Setting the IICBnTRG.IICBnSPT bit to 1 is prohibited during slave operation (IICBnSTR0.IICBnSSMS bit = 0) **2.** Setting the IICBnTRG.IICBnSPT bit to 1 at the same time as the IICBnTRG.IICBnSTT bit is prohibited.

(8) IICBnSTR0 – IICBn status register 0

This register indicates the statuses of the IICBn and the bus.

Access This register can be read only in 16-bit units. However, when IICBnCTL0.IICBnIICE is 0, this register can also be write accessed.

Address <IICBn_base_USER> + 0010_H

Initial Value 0000_H. This register is initialized by any reset. This register is also initialized by changing the value of the IICBnCTL0.IICBnIICE bit from 1 to 0 or from 0 to 1.

15	14	13	12	11	10	9	8
IICBn SSMS	0	IICBn SSDR	IICBn SSWT	IICBn SSEX	IICBn SSCO	IICBn SSTR	IICBn SSAC
R	R	R	R	R	R	R	R

7	6	5	4	3	2	1	0
IICBn SSRS	IICBn SSBS	IICBn SSST	IICBn SSSP	0	0	IICBn STCF	IICBn ALDF
R	R	R	R	R	R	R	R

Table 23-14 IICBnSTR0 register contents (1/6)

Bit position	Bit name	Function
15	IICBnSSMS	Master state check flag: Indicates that the IICBn is operating as a master. 1: Indicates that the IICBn is operating as a master. Setting condition: Upon detection of a start condition after 1 is written to the IICBnTRG.IICBnSTT bit Clearing conditions: • When 1 is written to the IICBnTRG.IICBnLRET bit • Upon detection of a stop condition • Upon detection of arbitration loss If a setting condition coincides with a clearing condition, the clearing condition takes priority.
13	IICBnSSDR	IICBnDAT register status flag 1: Indicates that data in the IICBnDAT register remains unprocessed. During reception operation:Received data remains unread in the IICBnDAT register. During transmission operation:Data written to the IICBnDAT register has not been transferred to the shift register. Setting condition: • When the IICBnDAT register is written during address transfer and data transfer while the IICBnSTR0.IICBnSSWT bit is 0 (Note that, even if the IICBnSTR0.IICBnSSWT bit is 0, the IICBnSSDR bit is not set to 1 if address data is written to the IICBnDAT register while the IICBn is operating as a master, because the address data is directly transferred to the shift register in this case.) • At the falling edge of the 9th clock after an address match with a slave • While IICBnCTL0.IICBnSLWT = 0 and single mode reception is being performed, at the falling edge of the 8th clock during data reception

Table 23-14 IICBnSTR0 register contents (2/6)

Bit position	Bit name	Function
13	IICBnSSDR	• At the falling edge of the 8th clock while in the continuous transfer mode (reception), regardless of the IICBnCTL0.IICBnSLWT bit value • While IICBnCTL0.IICBnSLWT = 1, at the falling edge of the 9th clock during data reception Clearing conditions: • Clearing conditions given priority over setting conditions - When 1 is written to the IICBnTRG.IICBnLRET bit - Upon detection of arbitration loss - At the falling edge of the 9th clock during address transfer while the IICBn is operating as a master - At the falling edge of the 8th clock during data transmission while IICBnCTL0.IICBnSLWT = 0 and continuous transmission is being performed - At the falling edge of the 9th clock during data transmission while IICBnCTL0.IICBnSLWT = 1 and continuous transmission is being performed • Clearing condition for which setting conditions are given priority (while in the continuous transfer mode (transmission) - When the IICBnDAT register is read while the shift register does not have any received data that must be transferred to the IICBnDAT register
12	IICBnSSWT	Wait state flag 1: Indicates that the IICBn is in the wait state. Setting condition: ■ In single transfer mode <Common to master/slave> • During data transfer, upon detection of the falling edge of the 8th clock with IICBnCTL0.IICBnSLWT = 0 • During data transfer, upon detection of the falling edge of the 9th clock with IICBnCTL0.IICBnSLWT = 1 <Master> • When the IICBn becomes a master (IICBnSTR0.IICBnSSMS = 1) after 1 is written to the IICBnTRG.IICBnSTT bit, and the falling edge of the first SCLn is detected without the IICBnDAT register being written • Upon detection of the falling edge of the 9th clock during address transfer <Slave> • Upon detection of the falling edge of the 9th clock during address transfer when an address match occurred ■ In continuous transfer mode <During data transfer period, common to master/slave> • During data transmission, when the data to be transmitted next has not been written - When IICBnCTL0.IICBnSLWT = 0, at the falling edge of the 8th clock during data transmission with IICBnSTR0.IICBnSSDR = 0 - When IICBnCTL0.IICBnSLWT = 1, at the falling edge of the 9th clock during data transmission with IICBnSTR0.IICBnSSDR = 0 • During data reception, when the previous received data has not been read - When IICBnCTL0.IICBnSLWT = 0, at the falling edge of the 8th clock during data reception with IICBnSTR0.IICBnSSDR = 1 - When IICBnCTL0.IICBnSLWT = 1, at the falling edge of the 9th clock during data reception with IICBnSTR0.IICBnSSDR = 1 - Upon NACK detection (However, only if 1 has not been written to IICBnTRG.IICBnSTT or IICBnTRG.IICBnSPT while the IICBn is operating as a master)

Table 23-14 IICBnSTR0 register contents (3/6)

Bit position	Bit name	Function
12	IICBnSSWT	During address transfer period, operating as master> - When the IICBn becomes a master (IICBnSTR0.IICBnSSMS = 1) after 1 is written to the IICBnTRG.IICBnSTT bit, and the first falling edge of SCLn is detected without the IICBnDAT register being written - Upon NACK detection (However, only if 1 has not been written to IICBnTRG.IICBnSTT or IICBnTRG.IICBnSPT) <During address transfer period, operating as slave> - Upon detection of the falling edge of the 9th clock while IICBnSTR0.IICBnSSTR bit is 0 during address transfer when an address match occurred - Upon NACK detection Clearing conditions: • Clearing conditions given priority over setting conditions - When 1 is written to the IICBnTRG.IICBnLRET bit - When 1 is written to the IICBnTRG.IICBnSTT bit while the IICBn is operating as a master in the continuous transfer mode - When 1 is written to the IICBnTRG.IICBnSPT bit while the IICBn is operating as a master in the continuous transfer mode - When the IICBnDAT register is written while the IICBn is performing transmission in the continuous transfer mode - During the wait state triggered by the falling edge of the 8th clock, when the IICBnDAT register is read while reception is performed in the continuous transfer mode - During the wait state triggered by the falling edge of the 9th clock, when the IICBnDAT register is read while the IICBn is performing reception in the continuous transfer mode and an acknowledge signal (ACK) has been received • Clearing condition for which setting conditions are given priority - When 1 is written to the IICBnTRG.IICBnWRET bit - When 1 is written to the IICBnTRG.IICBnSTT bit while the IICBn is operating as a master in the single transfer mode - When 1 is written to the IICBnTRG.IICBnSPT bit while the IICBn is operating as a master in the single transfer mode - When the IICBnDAT register is written while the IICBn is performing reception in the single transfer mode **Caution** If the IICBn exits the wait state that was triggered by the falling edge of the 9th clock by writing 1 to the IICBnTRG.IICBnWRET bit, the IICBnSTR0.IICBnSSTR bit is cleared to 0 and the bus is released (both SCLn and SDAn go into high impedance).
11	IICBnSSEX	Expansion code reception detection flag 1: Indicates that an expansion code has been received. Setting condition: Upon detection of the falling edge of the 8th clock while transferring received address data whose higher 4 bits are either 0000 or 1111 Clearing conditions: • When 1 is written to the IICBnTRG.IICBnLRET bit • Upon detection of a stop condition • Upon detection of a start condition **Caution** When the expansion codes match, the processing after the interrupt differs according to the ensuing data, and therefore is dependent on software processing.

Table 23-14 IICBnSTR0 register contents (4/6)

Bit position	Bit name	Function
10	IICBnSSCO	Address match detection flag 1: Indicates that an address that matches the IICBnSVA register has been detected. Setting condition: Upon detection of the falling edge of the 8th clock while transferring a received address that matches the IICBnSVA register Clearing conditions: • When 1 is written to the IICBnTRG.IICBnLRET bit • Upon detection of a stop condition • Upon detection of a start condition
9	IICBnSSTR	Transmission status detection flag 1: Indicates that data is being transmitted to the serial data bus. Setting condition: <Master> - Upon detection of a start condition after 1 is written to the IICBnTRG.IICBnSTT bit <Slave> - Upon detection of the falling edge of the 8th clock following reception of 1 to R/$\overline{W}$ bit during address transfer when an address match occurred Clearing conditions: <Common to master/slave> - When 1 is written to the IICBnTRG.IICBnLRET bit - Upon detection of a stop condition - When 1 is written to the IICBnTRG.IICBnWRET bit during the wait state triggered by the falling edge of the 9th clock <Master> - Upon detection of the falling edge of the 8th clock following reception of 1 to R/$\overline{W}$ bit during address transfer - Upon detection of arbitration loss <Slave> - Upon detection of a start condition
8	IICBnSSAC	Acknowledge ($\overline{ACK}$) detection flag 1:Indicates that an acknowledge signal has been detected. Setting condition: Upon detection of the falling edge of SCLn when a low level has been received at the $\overline{ACK}$ bit during participation in communications Clearing conditions: • When 1 is written to the IICBnTRG.IICBnLRET bit • Upon detection of the rising edge of SCLn **Caution** The value of the IICBnSTR0.IICBnSSAC bit changes regardless of whether or not an interrupt has occurred.
7	IICBnSSRS	Communication reserve state flag 0: Not communication reserve state 1: Communication reserve state Setting condition: When 1 is written to the IICBnTRG.IICBnSTT bit during bus communication while the IICBn is not operating as a master, in the communication reserve function enabled state (IICBnCTL1.IICBnSLRS = 0) Clearing conditions: • When 1 is written to the IICBnTRG.IICBnLRET bit • When IICBnSTR0.IICBnSSMS = 1

Table 23-14 IICBnSTR0 register contents (5/6)

Bit position	Bit name	Function
6	IICBnSSBS	IICBn bus status flag 0: Bus released state (initial communication state when IICBnCTL1.IICBnSLSE = 1) 1: Bus communication state (initial communication state when IICBnCTL1.IICBnSLSE = 0) Setting condition: • Upon detection of a start condition • When 1 is written to the IICBnCTL0.IICBnIICE bit when IICBnCTL1.IICBnSLSE = 0 Clearing conditions: Upon detection of a stop condition **Note** The IICBnSTR0.IICBnSSBS bit operates whether or not the IICBn is participating in communications.
5	IICBnSSST	Start condition detection flag 1: Indicates that a start condition has been detected. Setting condition: Upon detection of a start condition Clearing conditions: • When 1 is written to the IICBnTRG.IICBnLRET bit • Upon detection of a stop condition • Upon detection of the rising edge of SCLn following the end of address transfer **Note** The IICBnSTR0.IICBnSSST bit operates whether or not the IICBn is participating in communications.
4	IICBnSSSP	Stop condition detection flag 1: Indicates that a stop condition has been detected. Setting condition: Upon detection of a stop condition Clearing conditions: Upon detection of the falling edge of the first SCLn following start condition detection **Note** The IICBnSTR0.IICBnSSSP bit operates whether or not the IICBn is participating in communications.
1	IICBnSTCF	IICBnTRG.IICBnSTT bit clear flag 1: Indicates that the IICBnTRG.IICBnSTT bit has been cleared because start condition output failed. Setting condition: When 1 is written to the IICBnTRG.IICBnSTT bit during bus communication when the IICBn is not operating as a master, in the communication reserve function disabled state (IICBnCTL1.IICBnSLRS = 1) **Caution** Even if the bus is released in the external bus state, this bit is set to 1 when 1 is written to the IICBnTRG.IICBnSTT bit if the communication reserve function is disabled, unless the IICBn recognizes the bus release state (IICBnSTR0.IICBnSSBS = 1). Clearing condition: When 1 is written to the IICBnSTRC.IICBnCLSF bit

Table 23-14 IICBnSTR0 register contents (6/6)

Bit position	Bit name	Function
0	IICBnALDF	Arbitration loss detection flag 1: Indicates that an arbitration loss has been detected. Setting condition: Upon detection of arbitration loss Clearing condition: When 1 is written to the IICBnSTRC.IICBnCLAF bit If a setting condition coincides with a clearing condition, the setting condition takes priority. Upon detection of arbitration loss, the IICBnSTR0.IICBnSSMS and IICBnSTR0.IICBnSSTR bits are cleared to 0. (SCLn and SDAn become high level and the bus is released.) **Caution** When the IICBnSTR0.IICBnALDF bit is set to 1 due to arbitration loss, the INTIICBTIA or INTIICBTISn interrupt request signal is output. After confirming that the IICBnSTR0.IICBnALDF bit has been set to 1 with an interrupt request signal, clear the IICBnSTR0.IICBnALDF bit with the IICBnSTRC.IICBnCLAF bit. If the value of the IICBnSTR0.IICBnALDF is not cleared and remains 1, the INTIICBTISn interrupt request signal will be output at the interrupt timing, even during unrelated communication.

(9) IICBnSTR1 – IICBn status register 1

This register indicates the status of the serial bus.

Access This register is read-only, in 8-bit units.

Address <IICBn_base_USER> + 0014_H

Initial Value 00_H. This register is initialized by any reset.

Caution The serial clock (SCLn) and serial transmit/receive data (SDAn) are also read from an external source in the loopback mode (IICBnCTL1.IICBnMDLB = 1).

7	6	5	4	3	2	1	0
0	0	0	0	0	0	IICBn SSCL	IICBn SSDA
R	R	R	R	R	R	R	R

Table 23-15 IICBnSTR1 register contents

Bit position	Bit name	Function
1	IICBnSSCL	Indicates the level of the SCLn pin (input). 0: Low level 1: High level
0	IICBnSSDA	Indicates the level of the SDAn pin (input). 0: Low level 1: High level

(10) IICBnSTRC – IICBn status clear register

This register clears the IICBnSTCF and IICBnALDF bits of the IICBnSTR0 register.

Access This register can be read/written in 8-bit units.

Address <IICBn_base_USER> + 0018_H

Initial Value 00_H. This register is initialized by any reset.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	IICBn CLSF	IICBn CLAF
R	R	R	R	R	R	R/W	R/W

Table 23-16 IICBnSTRC register contents

Bit position	Bit name	Function
1	IICBnCLSF	Clears the IICBnSTR0.IICBnSTCF bit. 1: Clears the IICBnSTR0.IICBnSTCF bit. **Note** If the IICBnSTRC.IICBnCLSF bit is read after setting data, 0 is returned.
0	IICBnCLAF	Clears the IICBnSTR0.IICBnALDF bit. 1: Clears the IICBnSTR0.IICBnALDF bit. **Caution** If writing 1 to the IICBnSTRC.IICBnCLAF bit and the setting condition of the IICBnSTR0.IICBnALDF bit occur at the same time, the setting condition of the IICBnSTR0.IICBnALDF takes priority. **Note** If the IICBnSTRC.IICBnCLAF bit is read after data setting, 0 is returned.

23.6 Operation

The IICBn supports two transfer modes, single transfer mode and continuous transfer mode.

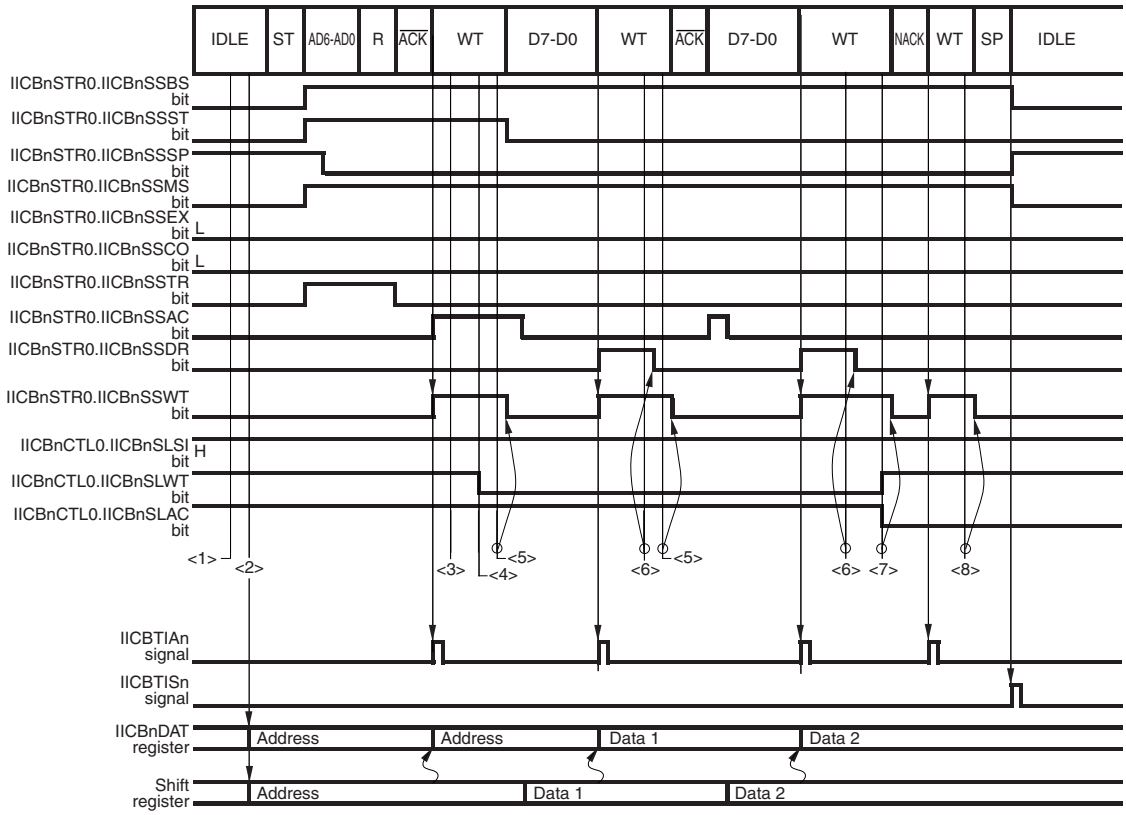
The transfer mode when the addresses of the master device and the slave device match is selected with the IICBnCTL0.IICBnMDTX0 bit, and the transfer mode when the extension code is detected by the slave device is selected with the IICBnCTL0.IICBnMDTX1 bit.

23.6.1 Single transfer mode

In the single transfer mode, a data transmit/receive interrupt request signal is output at the timing specified using the IICBnCTL0.IICBnSLW bit to make the IICBn enter the wait state, and transmit/receive data processing is performed during this wait state.

The various processing operations are described below.

(1) Example of communication in single transfer mode (master reception)



<1> Start condition output
Set the IICBnTRG.IICBnSTT bit (to 1).

<2> Address and transfer direction specification output
Set the address of the slave device and the transfer direction as 8 bits into the IICBnDAT register.

<3> Acknowledge result check
Check the acknowledge result by reading the IICBnSTR0.IICBnSSAC bit using the IICBTIAn interrupt.

<4> Wait timing setting
During data reception, clear the IICBnCTL0.IICBnSLWT bit (to 0) so that the IICBn enters the wait state at the falling edge of the 8th clock.

<5> Data reception
Exit the wait state by setting the IICBnTRG.IICBnWRET bit (to 1) to start reception.

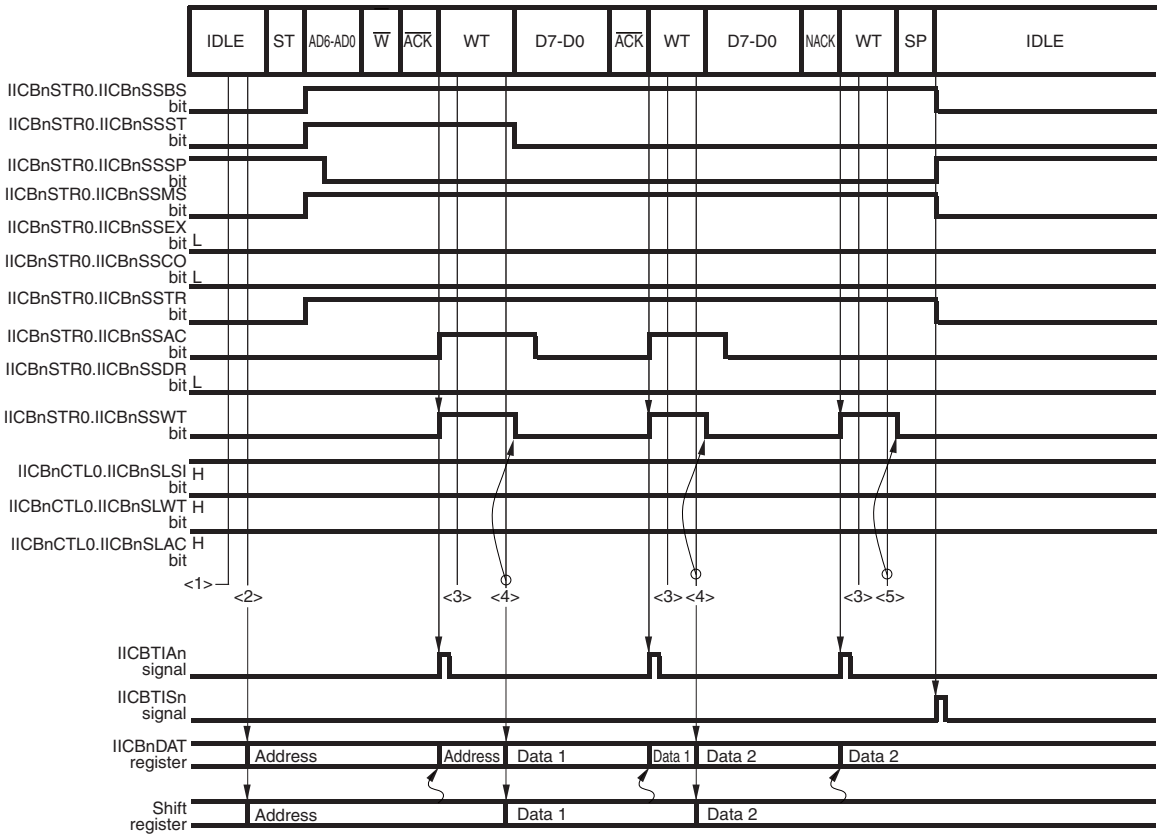
<6> Receive data load
Read the receive data from the IICBnDAT register using the IICBTIAn interrupt.

<7> Data reception completion processing

- Set the IICBnCTL0.IICBnSLWT bit to 1 and the IICBnCTL0.IICBnSLAC bit to 0.
- Next, exit the wait state by setting the IICBnTRG.IICBnWRET bit (to 1). The end of the data is notified to the transmitting device without outputting $\overline{\text{ACK}}$.

<8> Stop condition output
Set the IICBnTRG.IICBnSPT bit (to 1).

(2) Example of communication in single transfer mode (master transmission)



Note During data transmission, set the IICBnCTL0.IICBnSLWT bit (to 1) so that the IICBn enters the wait state at the falling edge of the 9th clock.

<1> Start condition output
Set the IICBnTRG.IICBnSTT bit (to 1).

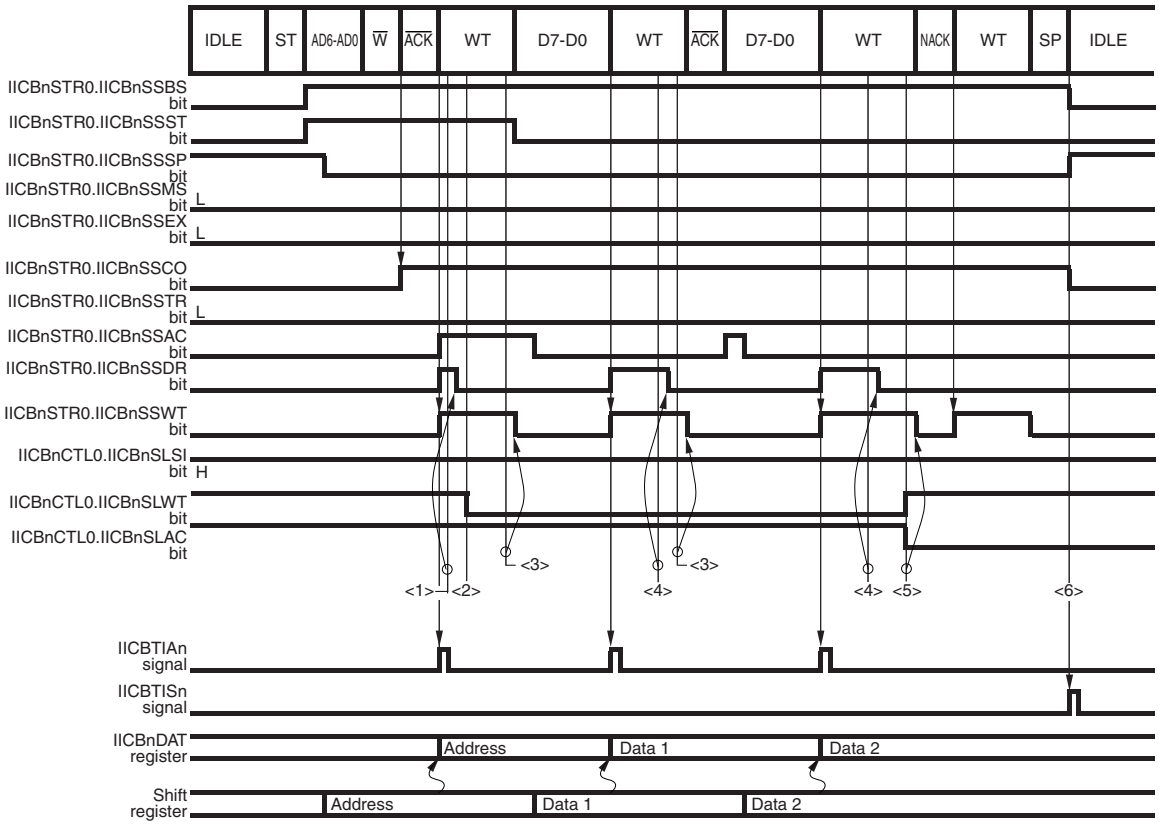
<2> Address and transfer direction specification output
Set the address of the slave device and the transfer direction as 8 bits into the IICBnDAT register.

<3> Acknowledge result check
Check the acknowledge result by reading the IICBnSTR0.IICBnSSAC bit using the IICBTIAn interrupt.

<4> Data transmission
Exit the wait state by setting the transmit data into the IICBnDAT register to start transmission.

<5> Stop condition output
Set the IICBnTRG.IICBnSPT bit (to 1).

(3) Example of communication in single transfer mode (slave reception)



<1> Operation mode check in slave mode

- Check the operation mode using the IICBTIAn interrupt.
- Check the address transfer, address match, and reception operation with the IICBnSTR0.IICBnSSST, IICBnSTR0.IICBnSSCO, and IICBnSTR0.IICBnSSTR bits.
- Read the IICBnDAT register (empty read).

<2> Wait timing setting
During data reception, clear the IICBnCTL0.IICBnSLWT bit (to 0) so that the IICBn enters the wait state at the falling edge of the 8th clock.

<3> Data reception
Exit the wait state by setting the IICBnTRG.IICBnWRET bit (to 1) to start reception.

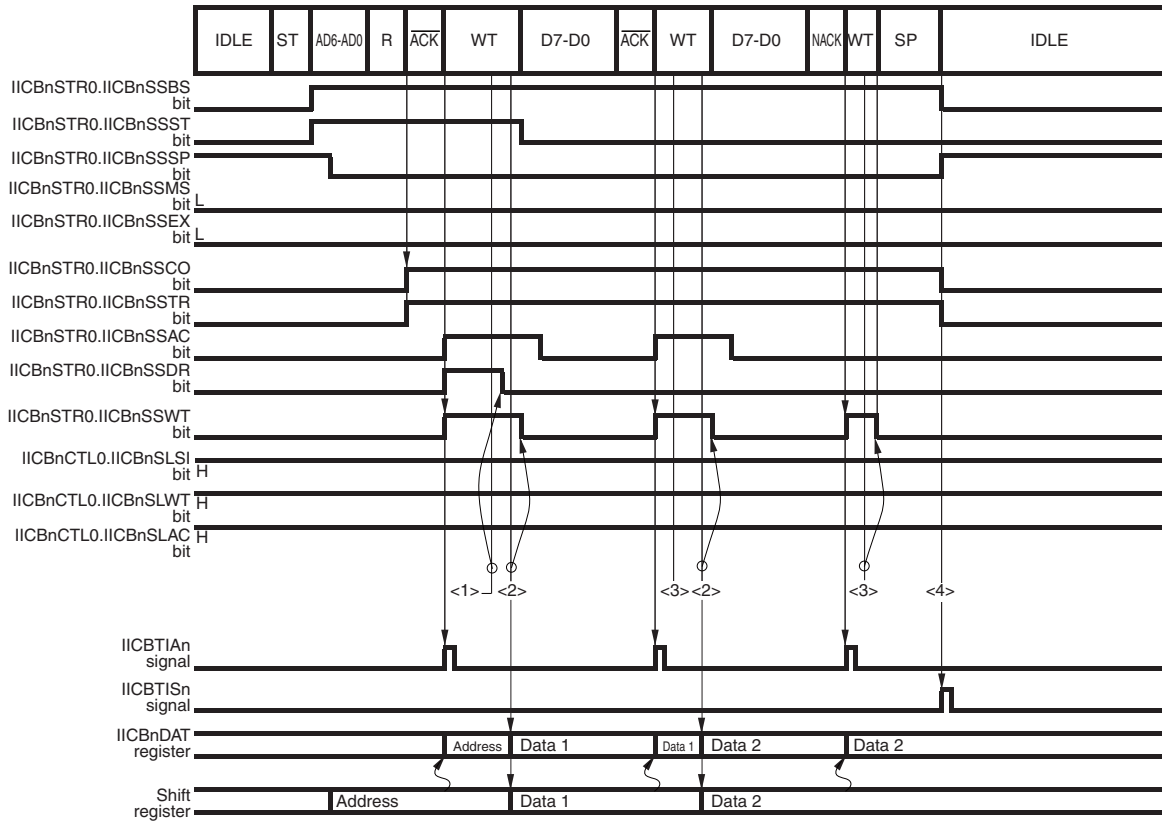
<4> Receive data load
Read the receive data from the IICBnDAT register using the IICBTIAn interrupt.

<5> Data reception completion processing

- Set the IICBnCTL0.IICBnSLWT bit to 1 and the IICBnCTL0.IICBnSLAC bit to 0.
- Next, exit the wait state by setting the IICBnTRG.IICBnWRET bit (to 1). The end of the data is notified to the transmitting device without outputting $\overline{\text{ACK}}$.

<6> Stop condition detection
Detect the stop condition using the IICBTISn interrupt.

(4) Example of communication in single transfer mode (slave transmission)



Note During data transmission, set the IICBnCTL0.IICBnSLWT bit (to 1) so that the IICBn enters the wait state at the falling edge of the 9th clock.

<1> Operation mode check in slave mode

- Check the operation mode using the IICBTIAn interrupt.
- Check the address transfer, address match, and reception operation with the IICBnSTR0.IICBnSSST, IICBnSTR0.IICBnSSCO, and IICBnSTR0.IICBnSSTR bits.
- Read the IICBnDAT register (empty read).

<2> Data transmission
Exit the wait state by setting the transmit data into the IICBnDAT register to start transmission.

<3> Acknowledge result check
Check the acknowledge result by reading the IICBnSTR0.IICBnSSAC bit using the IICBTIAn interrupt.
If $\overline{\text{ACK}}$ is not output, the transmission is judged to have been completed, and the IICBn exits the wait state by setting the IICBnTRG.IICBnWRET bit (to 1).

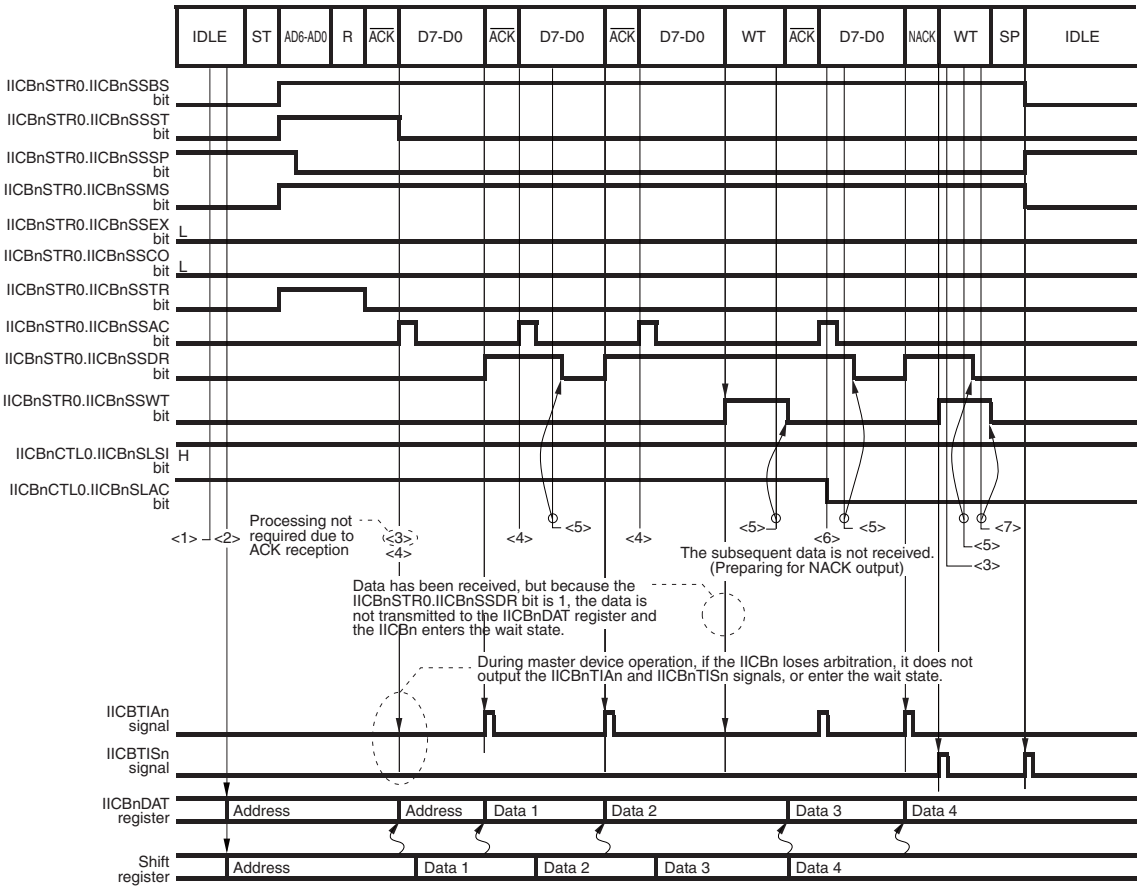
<4> Stop condition detection
Detect the stop condition using the IICBTISn interrupt.

23.6.2 Continuous transfer mode

The continuous transfer mode allows continuous communication without entering the wait state by reading/writing data to/from the IICBnDAT register each time the data transmit/receive interrupt request signal (IICBTIAn) is output.

The processing operations are described below.

(1) Example of communication in continuous transfer mode (master reception)



<1> Start condition output
Set the IICBnTRG.IICBnSTT bit (to 1).

<2> Address and transfer direction specification output
Set the address of the slave device and the transfer direction as 8 bits into the IICBnDAT register.

<3> Acknowledge result check
The IICBTISn interrupt occurs only if the slave device does not return $\overline{ACK}$. Check the acknowledge result by reading the IICBnSTR0.IICBnSSAC bit.

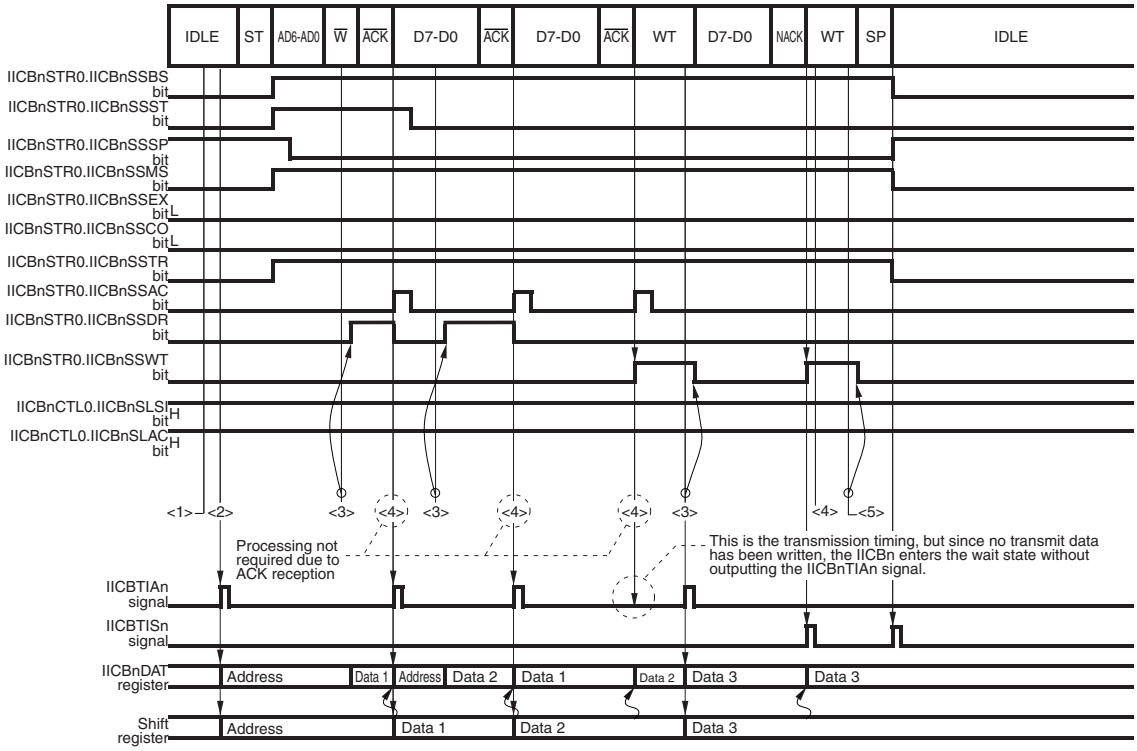
<4> Acknowledge result check
If there is no unread data in the IICBnDAT register by the time reception starts, the IICBn starts reception without entering the wait state.

<5> Receive data load
Read the receive data from the IICBnDAT register using the IICBTIAn interrupt.

<6> Data reception completion processing
By clearing the IICBnCTL0.IICBnSLAC bit (to 0) before reading the receive data immediately preceding the final receive data, the next $\overline{ACK}$ is not output and the end of the data is notified to the transmitting device.

<7> Stop condition output
Set the IICBnTRG.IICBnSPT bit (to 1).

(2) Example of communication in continuous transfer mode (master transmission)



<1> Start condition output
Set the IICBnTRG.IICBnSTT bit (to 1).

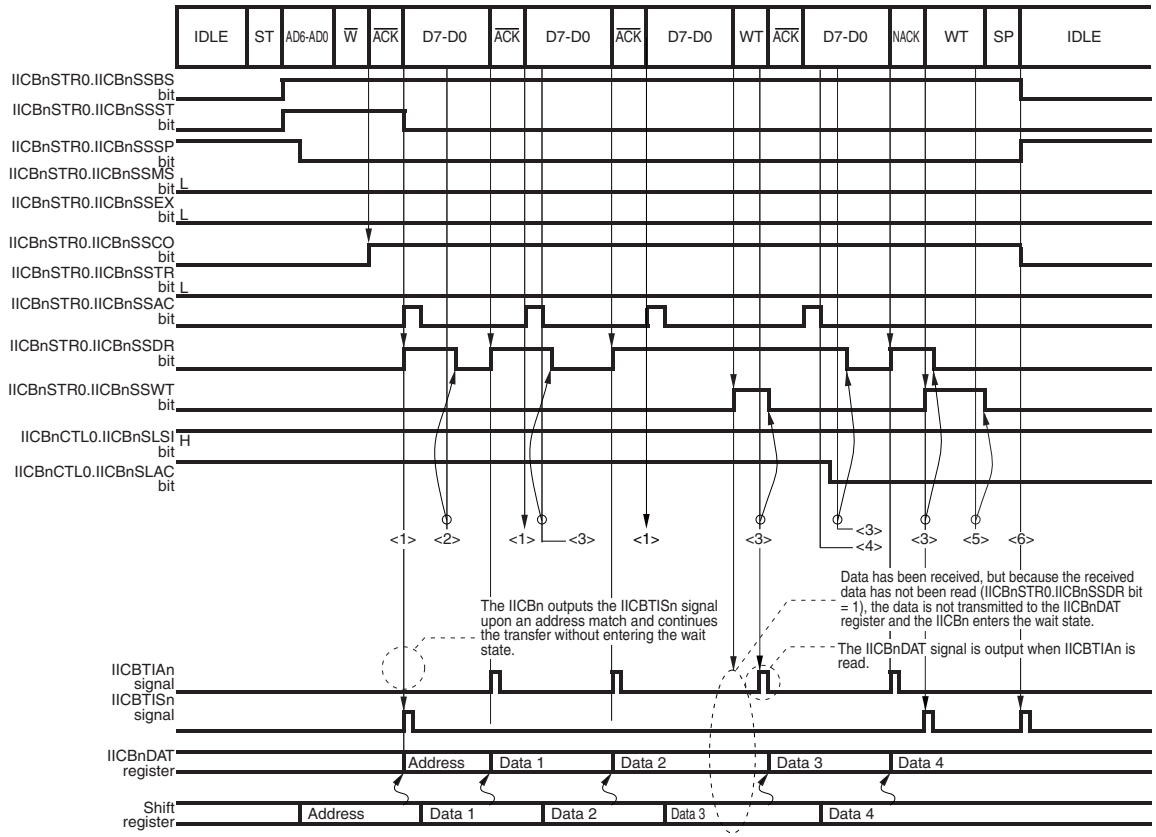
<2> Address and transfer direction specification output
Set the address of the slave device and the transfer direction as 8 bits into the IICBnDAT register.

<3> Data transmission
Set the transmit data to the IICBnDAT register using the IICBTIAn interrupt.

<4> Acknowledge result check
The IICBTISn interrupt occurs only if the slave device does not return $\overline{\text{ACK}}$. Check the acknowledge result by reading the IICBnSTR0.IICBnSSAC bit.

<5> Stop condition output
Set the IICBnTRG.IICBnSPT bit (to 1).

(3) Example of communication in continuous transfer mode (slave reception)



<1> Data reception
If there is no unread data in the IICBnDAT register by the time reception starts, the IICBn starts reception without entering the wait state.

<2> Operation mode check in slave mode

- Check the operation mode using the IICBTISn interrupt.
- Check the address transfer, address match, and reception operation with the IICBnSTR0.IICBnSSST, IICBnSTR0.IICBnSSCO, and IICBnSTR0.IICBnSSTR bits.
- Read the IICBnDAT register (empty read).

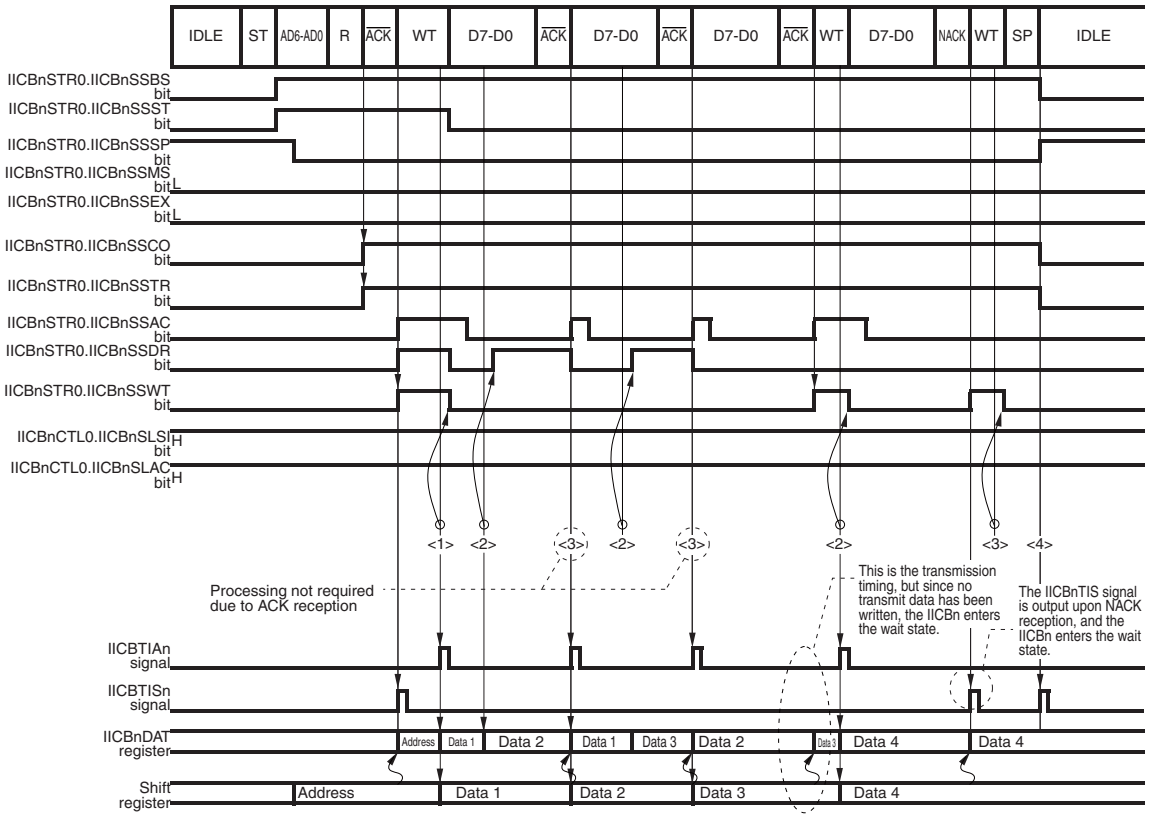
<3> Receive data load
Read the receive data from the IICBnDAT register using the IICBTIAn interrupt.

<4> Data reception completion processing <1>
By clearing the IICBnCTL0.IICBnSLAC bit (to 0) before reading the receive data immediately preceding the final receive data, the next $\overline{ACK}$ is not output and the end of the data is notified to the transmitting device.

<5> Data reception completion processing <2>
The IICBTISn interrupt occurs only if the slave device does not return $\overline{ACK}$.
Exit the wait state by setting the IICBnTRG.IICBnWRET bit (to 1).

<6> Stop condition detection
Detect the stop condition using the IICBTISn interrupt.

(4) Slave transmission in continuous transfer mode



<1> Operation mode check in slave mode

- Check the operation mode using the IICBTISn interrupt.
- Check the address transfer, address match, and reception operation with the IICBnSTR0.IICBnSSST, IICBnSTR0.IICBnSSCO, and IICBnSTR0.IICBnSSTR bits.
- After reading (empty read) the IICBnDAT register, set the first transmit data to the IICBnDAT register.

<2> Data transmission
Set the transmit data to the IICBnDAT register using the IICBTIAn interrupt.

<3> Acknowledge result check
The IICBTISn interrupt occurs only if the slave device does not return $\overline{\text{ACK}}$. Check the acknowledge result by reading the IICBnSTR0.IICBnSSAC bit.
If $\overline{\text{ACK}}$ is not output, the transmission is judged to have been completed, and the IICBn exits the wait state by setting the IICBnTRG.IICBnWRET bit (to 1).

<4> Stop condition detection
Detect the stop condition using the IICBTISn interrupt.

23.6.3 Arbitration

When the IICBn operates as the master device and loses arbitration, it enters the slave standby state by setting both SCLn and SDAn to high level upon detection of the arbitration loss, and then the IICBnSTR0.IICBnALDF bit is set (to 1) each time the status interrupt request signal (IICBTISn) is output.

(1) Status upon occurrence of arbitration

The statuses upon occurrence of arbitration during master device operation (IICBnSTR0.IICBnSSMS bit = 1) are listed below.

[1] Address transmission

[2] R/$\overline{W}$ bit transmission of address transfer

[3] Extension code transmission

[4] R/$\overline{W}$ bit transmission of extension code transfer

[5] Data transmission

[6] $\overline{ACK}$ bit transmission after data reception

[7] Start condition detection during address transfer or data transfer

[8] Stop condition detection during address transfer or data transfer

[9] The SDAn signal is low when the IICBn is attempting to output a restart condition

[10] The SDAn signal is low when the IICBn is attempting to output a stop condition

[11] The falling edge of the SCLn signal is detected when the IICBn is attempting to output a restart condition

23.6.4 Entering and exiting wait state

The IICBn enters the wait state at the following timings.

Table 23-17 Wait state transit timings

Timing	Description	Refer to:
Δ0	Upon detection of the first falling edge of the SCLn, following detection of start condition as the master device	*1 "Wait state at falling edge of first SCLn after IICBn became master" on page 1472*
Δ1	Upon detection of the falling edge of the 9th SCLn during address transfer after the start condition	*2 "Wait state upon detection of the falling edge of the 9th SCLn during address transfer after the start condition" on page 1473*
Δ2	Upon detection of the falling edge of the 8th SCLn during data transfer	*3 "Wait state upon detection of the falling edge of the 8th SCLn during data transfer" on page 1474*
Δ3	Upon detection of the falling edge of the 9th SCLn during data transfer	*4 "Wait state upon detection of the falling edge of the 9th SCLn during data transfer" on page 1474*

Note ST: Start condition
AD6 to AD0: Address
R/$\overline{W}$: Transfer direction specification
$\overline{ACK}$: Acknowledge
D7 to D0: Data
SP: Stop condition

The method to exit the wait state differs according to the wait state.

Exit the wait state by applying the appropriate method for each of the four wait states as described below.

(1) Wait state at falling edge of first SCLn after IICBn became master

Δ0 indicates the wait state when the data to be transferred has not been written (to the IICBnDAT register) when the falling edge of the first SCLn after the IICBn became the master is detected, after 1 was written to the IICBnTRG.IICBnSTT bit.

(a) Wait state transit condition

The IICBn enters the wait state if data is not written to the IICBnDAT register in the period from when the IICBnTRG.IICBnSTT bit becomes 1 until the Δ0 timing, upon detection of the first falling edge of SCLn after the IICBn became master, after 1 was written to the IICBnTRG.IICBnSTT bit.

However, the valid times to write data to the IICBnDAT register (without entering the wait state) after 1 was written to the IICBnTRG.IICBnSTT bit differ depending on whether the communication reservation function is enabled. The valid times to write to the IICBnDAT register for each of these cases are shown in *Figure 23-11 "Valid times to write to IICBnDAT register"*.

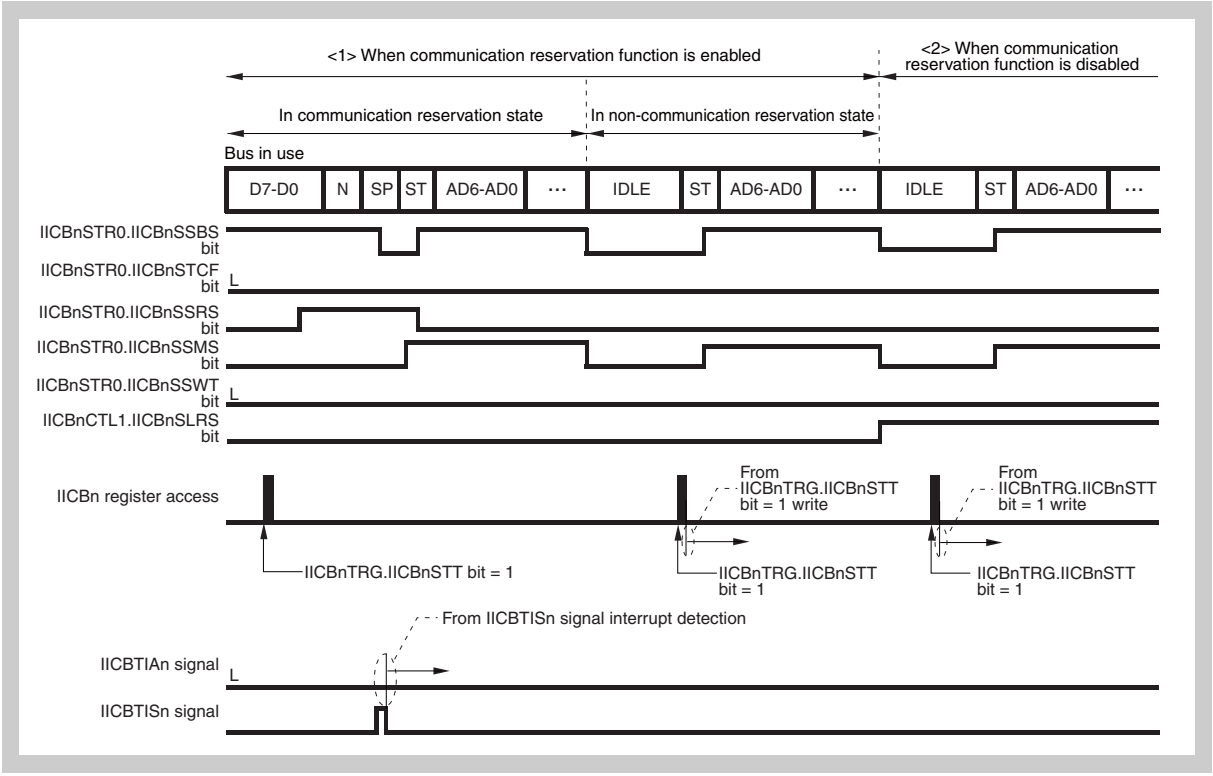


Figure 23-11 Valid times to write to IICBnDAT register

Caution The communication reservation function is disabled (<2> in the above figure) while the IICBnSTR0.IICBnSTCF bit is 0.
When the IICBnSTR0.IICBnSTCF bit becomes 1, setting from IICBnSTR0.IICBnSTCF bit = 1 write is required again.

(b) Wait state exit conditions

Exit the wait state by writing to the IICBnDAT register.

(2) Wait state upon detection of the falling edge of the 9th SCLn during address transfer after the start condition

Δ1 indicates the wait state entered upon completion of address transfer.

(a) Wait state transit condition

<Single transfer mode>

In the single transfer mode, the IICBn always enters the wait state while it operates as the master.

While the IICBn operates as a slave, it enters the wait state upon an address match, or upon extension code detection while the IICBnCTL0.IICBnSLWT bit is 1.

<Continuous transfer mode>

In the continuous transfer mode, the IICBn enters the wait state in the following cases.

- Upon detection of NACK
- When the IICBn operates as the master and transmits data, if the data to be transferred next has not been written
- When the IICBn operates as a slave, if the received data has not been read, or during transmission

(b) Wait state exit conditions

<Single transfer mode>

Exit the wait state by writing to the IICBnDAT register during transmission, or by writing 1 to the IICBnTRG.IICBnWRET bit during reception. When the IICBn operates as the master and the IICBnSTR0.IICBnSSAC bit is 0 or the IICBn is at the transmission side, the wait state can be exited by writing 1 to the IICBnTRG.IICBnSTT or the IICBnTRG.IICBnSPT bit.

<Continuous transfer mode>

Exit the wait state by writing to the IICBnDAT register during transmission, or by reading the IICBnDAT register during reception. When the IICBn operates as the master and the IICBnSTR0.IICBnSSAC bit is 0, the wait state can be exited by writing 1 to the IICBnTRG.IICBnSTT or IICBnTRG.IICBnSPT bit.

(3) Wait state upon detection of the falling edge of the 8th SCLn during data transfer

Δ2 indicates the wait state entered upon detection of the falling edge of the 8th SCLn during data transfer.

(a) Wait state transit condition

<Single transfer mode>

When the IICBn participates in communications and the IICBnCTL0.IICBnSLWT bit is 0, the IICBn enters the wait state if the falling edge of the 8th SCLn is detected.

<Continuous transfer mode>

When the IICBn participates in communications and the IICBnSTR0.IICBnSSTR bit is 0, the IICBn enters the wait state if processing of the previous data (read from the IICBnDAT register) has not completed and 1 has not been written to the IICBnTRG.IICBnSTT and IICBnTRG.IICBnSPT bits before the falling edge of the 8th SCLn.

(b) Wait state exit conditions

<Single transfer mode>

Exit the wait state by writing to the IICBnDAT register during transmission, or by writing 1 to the IICBnTRG.IICBnWRET bit during reception.

<Continuous transfer mode>

Exit the wait state by reading the IICBnDAT register.

(4) Wait state upon detection of the falling edge of the 9th SCLn during data transfer

Δ3 indicates the wait state entered upon detection of the falling edge of the 9th SCLn during data transfer.

During the continuous transfer mode, the IICBn enters the wait state upon NACK reception.

(a) Wait state transit condition

<Single transfer mode>

When the IICBn participates in communications and the IICBnCTL0.IICBnSLWT bit is 1, the IICBn enters the wait state if the falling edge of the 9th SCLn is detected.

<Continuous transfer mode>

When the IICBn participates in communications, it enters the wait state in the following three cases during data transmission:

- Upon reception of NACK by ACK bit while the IICBnCTL0.IICBnSLWT bit is 1
- When transmit data is not written to the data register
- When the previous received data is not read

(b) Wait state exit conditions

The wait state exit conditions are listed for each transfer mode in *Table 23-18 "Wait state exit conditions"*.

Table 23-18 Wait state exit conditions

Master/ slave	Transfer mode	Transfer direction	IICBnSTR0. IICBnSSAC bit	Exit conditions
Master	Single transfer mode	Reception	0	IICBnTRG.IICBnSTT bit = 1 or IICBnTRG.IICBnSPT bit = 1
			1	IICBnTRG.IICBnWRET bit = 1
		Transmission	0	IICBnTRG.IICBnSTT bit = 1 or IICBnTRG.IICBnSPT bit = 1
			1	Write to IICBnDAT register or IICBnTRG.IICBnSTT bit = 1 or IICBnTRG.IICBnSPT bit = 1
	Continuous transfer mode	Reception	0	IICBnTRG.IICBnSTT bit = 1 or IICBnTRG.IICBnSPT bit = 1
			1	Read from IICBnDAT register[a]
		Transmission	0	IICBnTRG.IICBnSTT bit = 1 or IICBnTRG.IICBnSPT bit = 1
			1	Write to IICBnDAT register[b]
Slave	Single transfer mode	Reception	-	IICBnTRG.IICBnWRET bit = 1
		Transmission	0	IICBnTRG.IICBnWRET bit = 1
			1	Write to IICBnDAT register[a]
	Continuous transfer mode	Reception	0	IICBnTRG.IICBnWRET bit = 1
		Transmission	0	IICBnTRG.IICBnWRET bit = 1
			1	Write to IICBnDAT register

a) Condition for exiting the wait state that was entered when no transmit data has been written to the data register
b) Condition for exiting the wait state that was entered when the received data has not been read

23.6.5 Extension code

The processing when the extension code is received differs according to the data after the extension code and thus must be executed through the user's software.

Therefore, the operation differs from that during normal slave address reception. These differences are described below.

(1) When the upper 4 bits of the received address are 0000 or 1111, the extension code reception flag (IICBnSTR0.IICBnSSEX bit) is set to 1 to indicate that an extension code has been received. The status interrupt request signal (IICBTISn) is output at the falling edge of the 8th clock, and the IICBn enters the wait state (IICBnTRG.IICBnSSWT = 1). The IICBnSTR0.IICBnSSDR bit is then set (to 1).

(2) During address transfer, the acknowledge output can be controlled by setting the IICBnCTL0.IICBnSLAC bit. (Note that an acknowledge is always output upon an address match, regardless of the setting of this bit, during address transfer for normal slave address reception.)

(3) The method for exiting the wait state entered upon extension code detection depends on the setting of the IICBnCTL0.IICBnMDTX1 bit as follows.

<When IICBnCTL0.IICBnMDTX1 bit is 0>

During transmission while the IICBnCTL0.IICBnSLWT bit is 0, exit the wait state by writing to the IICBnDAT register. During transmission while the IICBnCTL0.IICBnSLWT bit is 1, or during reception, exit the wait state by writing 1 to the IICBnTRG.IICBnWRET bit.

<When IICBnCTL0.IICBnMDTX1 bit is 1>

During transmission, exit the wait state by writing to the IICBnDAT register, and, during reception, exit the wait state by reading from the IICBnDAT register.

(4) At the falling edge of the 9th clock, if the IICBnCTL0.IICBnSLWT bit is 1, the interrupt request signal (IICBTIAn) is output and the IICBn enters the wait state (IICBnTRG.IICBnSSWT = 1). If the IICBnCTL0.IICBnSLWT bit is 0, the interrupt request signal (IICBTIAn) is not output and the IICBn does not enter the wait state.

(5) If the IICBn receives an extension code, it participates in communications even if the addresses do not match.

For example, to avoid operating the IICBn as a slave device after receiving an extension code, set the IICBnTRG.IICBnLRET bit to 1. The IICBn enters the standby state for the next communication.

23.7 Interrupt Request Signals

Caution In this section, the operation when an extension code is received is omitted. For details about the extension code, refer to *23.6.5 "Extension code"*.

The IICBn has two interrupt request signals, the data transmit/receive interrupt request signal (IICBTIAn) and the status interrupt request signal (IICBTISn). Both signals are pulses of one PCLK clock width. The interrupt request signal output timing differs according to the transfer mode set using the IICBnCTL0.IICBnMDTX1 and IICBnCTL0.IICBnMDTX0 bits. The interrupt request signals are explained below for each transfer mode.

To perform transfer with an address match between the master device and the slave device, select the single transfer mode or continuous transfer mode with the IICBnCTL0.IICBnMDTX0 bit, and to perform transfer with extension code detection by the slave, select the single transfer mode or continuous transfer mode using the IICBnCTL0.IICBnMDTX1 bit.

23.7.1 Single transfer mode

The interrupt request signal timing in the single transfer mode is described in *Table 23-19 “Interrupt request signal output timing (single transfer mode)”* below.
During the single transfer mode, for the IICBTIAn and IICBTISn interrupt request signals, whether to output an interrupt is judged based on the IICBn state when the falling edge of SCLn is detected during the bus cycle. Note, however, that whether to output an interrupt is judged based on the IICBn state when a stop condition is detected at the Δ4 timing.

Table 23-19 Interrupt request signal output timing (single transfer mode)

Output timing	Description	Refer to:
Δ1	Upon detection of the falling edge of the 9th SCLn during address transfer	*1 “Interrupt request signal output conditions and output interrupt request signals during address transfer” on page 1479*
Δ2	Upon detection of the falling edge of the 8th SCLn during data transfer	*2 “Interrupt request signal output conditions and interrupt request signals output during data transfer” on page 1480*
Δ3	Upon detection of the falling edge of the 9th SCLn during data transfer	*2 “Interrupt request signal output conditions and interrupt request signals output during data transfer” on page 1480*
Δ4	Upon detection of a stop condition	*3 “Interrupt request signal output upon stop condition detection” on page 1480*

Note
ST: Start condition
AD6 to AD0: Address
R/$\overline{W}$: Transfer direction specification
$\overline{ACK}$: Acknowledge
D7 to D0: Data
SP: Stop condition

(1) Interrupt request signal output conditions and output interrupt request signals during address transfer

Δ1 in *Table 23-19 "Interrupt request signal output timing (single transfer mode)"* indicates the interrupt request signal output timing during an address transfer. *Table 23-20 "Interrupt request signal output conditions and interrupt request signals output during address transfer (single transfer mode)"* indicates the interrupt request signal output condition and the interrupt request signal that is output (IICBTIAn or IICBTISn) at the timing of Δ1.

Table 23-20 Interrupt request signal output conditions and interrupt request signals output during address transfer (single transfer mode)

IICBn SSMS	IICBn ALDF	IICBn SLWT	IICBn SSCO	Δ1		Remark
				Interrupt	Wait	
1	0	×	×	IICBTIAn	Wait	-
1	1	×	×	This state does not exist.		-
0	0	×	0	IICBTISn[a]	-	After restart, non-participation in communications
0	0	×	1	IICBTIAn	Wait	-
0	1	×	0	IICBTISn	-	After arbitration loss, non-participation in communications
0	1	×	1	IICBTIAn	Wait	-

a) In case of an address match or extension code detection, before the restart condition

Note ×: don't care

(2) Interrupt request signal output conditions and interrupt request signals output during data transfer

Δ2 and Δ3 in *Table 23-19 "Interrupt request signal output timing (single transfer mode)"* indicate the interrupt request signal output timing during a data transfer. The interrupt request signal output timing of Δ2 or Δ3 is determined according to the setting of the IICBnCTL0.IICBnSLWT bit. *Table 23-21 "Interrupt request signal output conditions and interrupt request signals output during address transfer (single transfer mode)"* indicates the interrupt request signal output condition and the interrupt request signal that is output (IICBTIAn or IICBTISn) at the timing of Δ2 and Δ3.

Table 23-21 Interrupt request signal output conditions and interrupt request signals output during address transfer (single transfer mode)

IICBn SSMS	IICBn ALDF	IICBn SLWT	IICBn SSCO	Δ2		Δ3		Remark
				Interrupt	Wait	Interrupt	Wait	
1	0	0	×	IICBTIAn	Wait	-	-	-
1	0	1	×	-	-	IICBTIAn	Wait	-
1	1	×	×	This state does not exist.				-
0	0	×	0	-	-	-	-	Non-participation in communications
0	0	0	1	IICBTIAn	Wait	-	-	-
0	0	1	1	-	-	IICBTIAn	Wait	-
0	1	0	0	IICBTISn	-	-	-	Non-participation in communications after arbitration loss
0	1	1	0	-	-	IICBTISn	-	Non-participation in communications after arbitration loss
0	1	0	1	IICBTIAn	Wait	-	-	-
0	1	1	1	-	-	IICBTIAn	Wait	-

Note ×: don't care

(3) Interrupt request signal output upon stop condition detection

Δ4 in *Table 23-19 "Interrupt request signal output timing (single transfer mode)"* indicates the interrupt request signal output timing upon detection of a stop condition.

Interrupt request signal output is controlled according to the IICBnCTL0.IICBnSLSI bit. If a stop condition is detected while the IICBnCTL0.IICBnSLSI bit is 1, the status interrupt request signal (IICBTISn) is output.

23.7.2 Continuous transfer mode

(1) Data transmit/receive interrupt request signal (IICBTIAn)

The conditions for outputting an IICBTIAn signal in the continuous transfer mode are described below.

- Interrupt request signal output condition during reception

  When receive data is saved from the shift register to the IICBnDAT register (timing <1> in *Figure 23-12 "IICBTIAn signal output timing (reception in continuous transfer mode)"*)

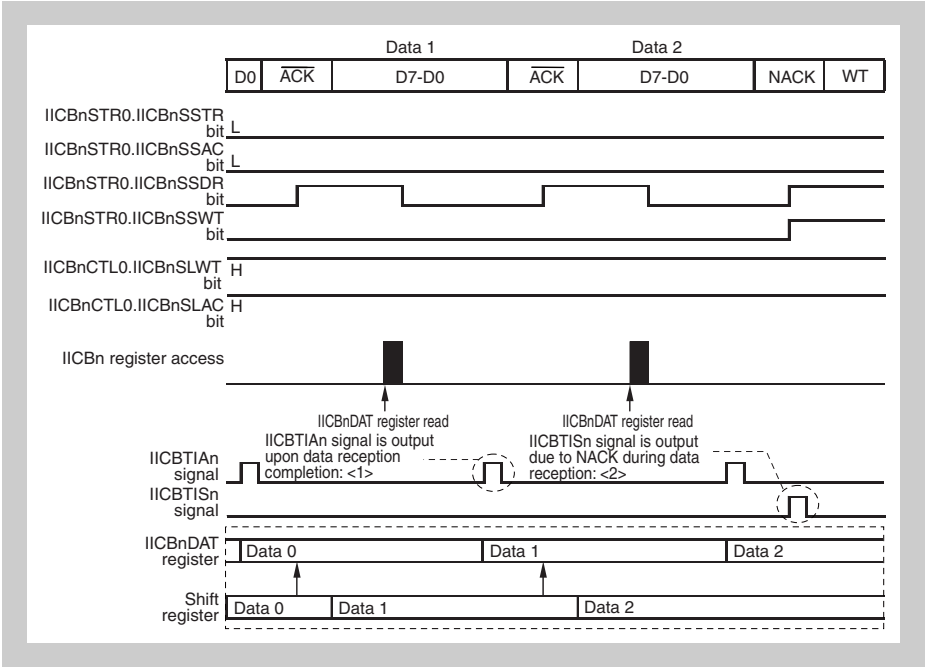


Figure 23-12 IICBTIAn signal output timing (reception in continuous transfer mode)

- Interrupt request signal output condition during transmission

  When data is written to the IICBnDAT register while there is no transmit data in the shift register and IICBnDAT register (timing <2> in *Figure 23-13 "IICBTIAn signal output timing (transmission in continuous transfer mode)"*).

  When data is saved from the IICBnDAT register to the shift register (timing <1> in *Figure 23-13 "IICBTIAn signal output timing (transmission in continuous transfer mode)"*).

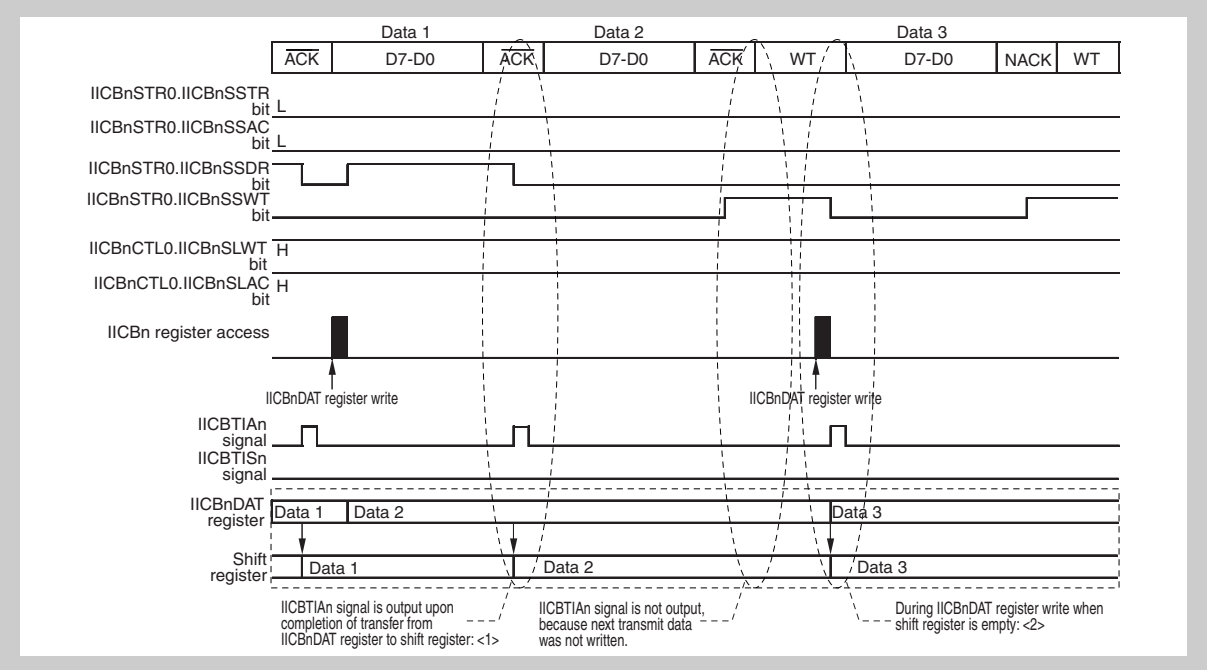


Figure 23-13 IICBTIAn signal output timing (transmission in continuous transfer mode)

(2) Status interrupt request signal (IICBTISn)

The IICBTISn signal output timing in the continuous transfer mode is the same as that in the single transfer mode.

Table 23-22 IICBTISn signal output timing

Output timing	Description	Refer to:
Δ1	Upon detection of the falling edge of the 9th SCLn during address transfer after the start condition	*a "IICBTISn signal output conditions during address transfer" on page 1484*
Δ2	Upon detection of the falling edge of the 8th SCLn during data transfer	*b "IICBTISn signal output conditions during data transfer" on page 1485*
Δ3	Upon detection of the falling edge of the 9th SCLn during data transfer	*b "IICBTISn signal output conditions during data transfer" on page 1485*
Δ4	Upon detection of a stop condition	*c "IICBTISn signal output upon detection of stop condition" on page 1486*

Note
ST: Start condition
AD6 to AD0: Address
R/$\overline{W}$: Transfer direction specification
$\overline{ACK}$: Acknowledge
D7 to D0: Data
SP: Stop condition

(a) IICBTISn signal output conditions during address transfer

Δ1 in *Table 23-22 "IICBTISn signal output timing"* indicates the IICBTISn signal output timing during address transfer. *Table 23-23 "IICBTISn signal output conditions during address transfer (continuous transfer mode)"* indicates the IICBTISn signal output conditions at the Δ1 timing.

Table 23-23 IICBTISn signal output conditions during address transfer (continuous transfer mode)

IICBn SSMS	IICBn SSCO	IICBn ALDF	Transfer direction	IICBn SSDR	IICBn SSAC	Δ1	
						Interrupt	Wait
1	×	0	Transmission	0	1	-	Wait
1	×	0	Transmission	0	0	IICBTISn	Wait
1	×	0	Transmission	1	1	-	-
1	×	0	Transmission	1	0	IICBTISn	Wait
1	×	0	Reception	0	1	-	-
1	×	0	Reception	0	0	IICBTISn	Wait
1	×	0	Reception	1	1	IICBTISn during IICBnDAT read[a]	Wait
1	×	0	Reception	1	0	IICBTISn during IICBnDAT read	Wait
1	×	1	×	×	×	This state does not exist.	
0	0	0	×	×	×	IICBTISn[b]	-
0	0	1	×	×	×	IICBTISn	-
0	1	×	Transmission	×	1	IICBTISn	Wait
0	1	×	Reception	0	1	IICBTISn	-
0	1	×	Reception	1	1	IICBTISn during IICBnDAT read	Wait

a) Upon restarting without reading IICBnDAT after the reception ends
b) Upon an address match before restart condition

Caution For Δ1, the IICBnSTR0.IICBnSSAC bit is always 0.

Note ×: don't care

(b) IICBTISn signal output conditions during data transfer

Δ2 and Δ3 in *Table 23-22 "IICBTISn signal output timing"* indicate the IICBTISn signal output timings during data transfer. *Table 23-24 "IICBTISn signal output conditions during data transfer (continuous transfer mode)"* indicates the IICBTISn signal output conditions at the Δ2 and Δ3 timings.

Table 23-24 IICBTISn signal output conditions during data transfer (continuous transfer mode)

IICBn SSMS	IICBn SSCO	IICBn SLWT	IICBn ALDF	Transfer direction	IICBn SSDR	IICBn SSAC	IICBnSTT or IICBnSPT	Δ2		Δ3	
								Interrupt	Wait	Interrupt	Wait
1	×	0	×	Transmission	0	1	a	-	-	-	Wait
1	×	0	×	Transmission	0	0	a	-	-	IICBTISn	Wait
1	×	0	×	Transmission	1	1	a	-	-	-	-
1	×	0	×	Transmission	1	0	a	-	-	IICBTISn	Wait
1	×	0	×	Reception	0	1	a	-	-	-	-
1	×	0	×	Reception	0	0	a	-	-	IICBTISn	Wait
1	×	0	×	Reception	1	1	a	-	-	-	-
1	×	0	×	Reception	1	0	a	-	-	IICBnDAT after IICBTISn read	Wait
1	×	×	×	×	×	0	b		-	IICBTISn	-
1	×	×	×	×	×	1	b	-	-	-	-
0	0	×	0	×	×	×	×	-	-	-	-
0	0	0	1	Reception	×	×	×	IICBTIS	-	-	-
0	0	1	1	Transmission	×	×	×	-	-	IICBTISn	-
0	1	0	×	Transmission	0	1	a	-	-	-	Wait
0	1	0	×	Transmission	0	0	a	-	-	IICBTISn	Wait
0	1	0	×	Transmission	1	1	a	-	-	-	-
0	1	0	×	Transmission	1	0	a	-	-	IICBTISn	Wait
0	1	0	×	Reception	0	1	a	-	-	-	-
0	1	0	×	Reception	0	0	a	-	-	IICBTISn	Wait
0	1	0	×	Reception	1	1	a	-	-	-	-
0	1	0	×	Reception	1	0	a	-	-	IICBnDAT during IICBTISn read	Wait

a) When 1 has not been written to the IICBnTRG.IICBnSTT or IICBnTRG.IICBnSPT bit
b) When 1 has been written to the IICBnTRG.IICBnSTT or IICBnTRG.IICBnSPT bit

Note ×: don't care

(c) IICBTISn signal output upon detection of stop condition

∆4 in *Table 23-22 "IICBTISn signal output timing"* indicates the IICBTISn signal output timing upon detection of a stop condition.

IICBTISn signal output is controlled according to the IICBnCTL0.IICBnSLSI bit. If a stop condition is detected while the IICBnCTL0.IICBnSLSI bit is 1, the IICBTISn signal is output.

23.8 Interrupt Outputs and Statuses

This section describes the statuses of the IICBnSTR0 register during interrupt output by communication flow.

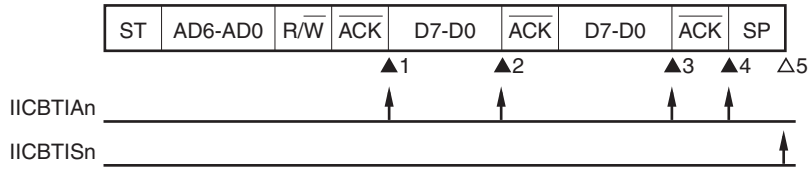
The meanings of the symbols used in the figures are as follows.

ST:	Start condition
AD6-AD0:	Address
R, $\overline{W}$, R/$\overline{W}$:	Transfer direction specification
$\overline{ACK}$:	Acknowledge
NACK:	Not acknowledge
D7-D0:	Data
SP:	Stop condition

23.8.1 Single transfer mode (master device operation)

(1) Start ~ Address ~ Data ~ Data ~ Stop (normal transmission/reception)

<1> When IICBnCTL0.IICBnSLWT bit is 0



▲1: IICBnSTR0 register = 1-0100X1 0110--00B

▲2: IICBnSTR0 register = 1-0100X0 0100--00B

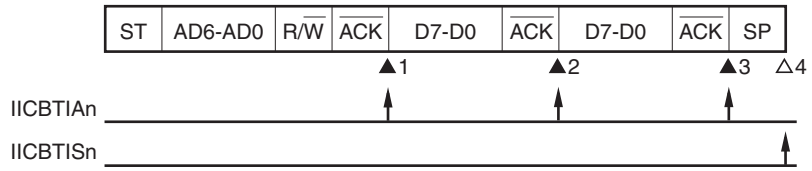
▲3: IICBnSTR0 register = 1-0100X0 0100--00B (IICBnCTL0.IICBnSLWT bit = 1)

▲4: IICBnSTR0 register = 1-0100XX 0100--00B (IICBnTRG.IICBnSPT bit = 1)

△5: IICBnSTR0 register = 0-000000 0001--00B

Note ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
\- Undefined
X don't care

<2> When IICBnCTL0.IICBnSLWT bit is 1



▲1: IICBnSTR0 register = 1-0100X1 0110--00B

▲2: IICBnSTR0 register = 1-0100X0 0100--00B

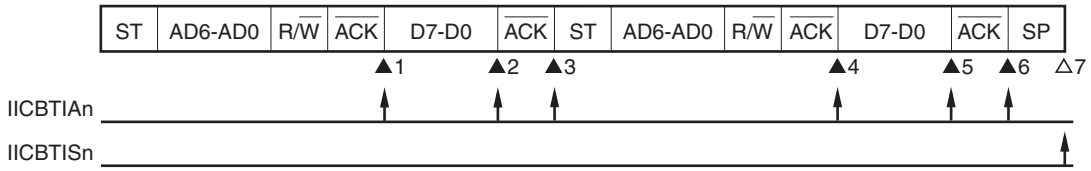
▲3: IICBnSTR0 register = 1-0100XX 0100--00B (IICBnTRG.IICBnSPT bit = 1)

△4: IICBnSTR0 register = 0-000000 0001--00B

Note ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
\- Undefined
X don't care

(2) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop (restart)

<1> When IICBnCTL0.IICBnSLWT bit is 0



▲1: IICBnSTR0 register = 1-0100X1 0110--00B

▲2: IICBnSTR0 register = 1-0100X0 0100--00B (IICBnCTL0.IICBnSLWT bit = 1)

▲3: IICBnSTR0 register = 1-0100XX 0100--00B (IICBnTRG.IICBnSTT bit = 1, IICBnCTL0.IICBnSLWT bit = 0)

▲4: IICBnSTR0 register = 1-0100X1 0110--00B

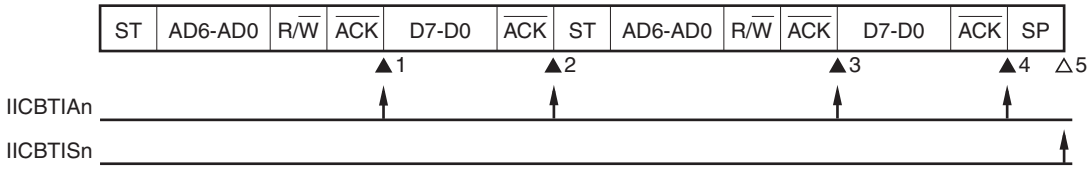
▲5: IICBnSTR0 register = 1-0100X0 0100--00B (IICBnCTL0.IICBnSLWT bit = 1)

▲6: IICBnSTR0 register = 1-0100XX 0100--00B (IICBnTRG.IICBnSPT bit = 1)

△7: IICBnSTR0 register = 0-000000 0001--00B

Note ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
- Undefined
X don't care

<2> When IICBnCTL0.IICBnSLWT bit is 1



▲1: IICBnSTR0 register = 1-0100X1 0110--00B

▲2: IICBnSTR0 register = 1-0100XX 0100--00B (IICBnTRG.IICBnSTT bit = 1)

▲3: IICBnSTR0 register = 1-0100X1 0110--00B

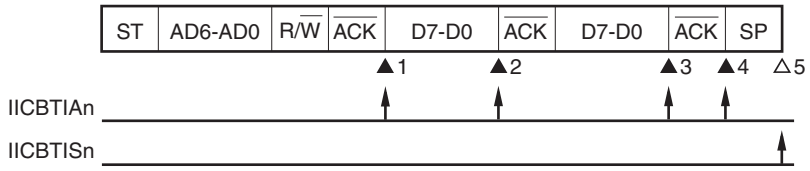
▲4: IICBnSTR0 register = 1-0100XX 0100--00B (IICBnTRG.IICBnSPT bit = 1)

△5: IICBnSTR0 register = 0-000000 0001--00B

Note ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
- Undefined
X don't care

(3) Start ~ Code ~ Data ~ Data ~ Stop (extension code transmission)

<1> When IICBnCTL0.IICBnSLWT bit is 0



▲1: IICBnSTR0 register = 1-0110X1 0110--00B

▲2: IICBnSTR0 register = 1-0110X0 0100--00B

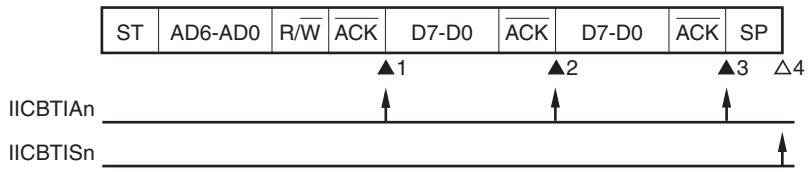
▲3: IICBnSTR0 register = 1-0110X0 0100--00B (IICBnCTL0.IICBnSLWT bit = 1)

▲4: IICBnSTR0 register = 1-0110XX 0100--00B (IICBnTRG.IICBnSPT bit = 1)

△5: IICBnSTR0 register = 0-000000 0001--00B

Note ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
\- Undefined
X don't care

<2> When IICBnCTL0.IICBnSLWT bit is 1



▲1: IICBnSTR0 register = 1-0110X1 0110--00B

▲2: IICBnSTR0 register = 1-0110X1 0100--00B

▲3: IICBnSTR0 register = 1-0110XX 0100--00B (IICBnTRG.IICBnSPT bit = 1)

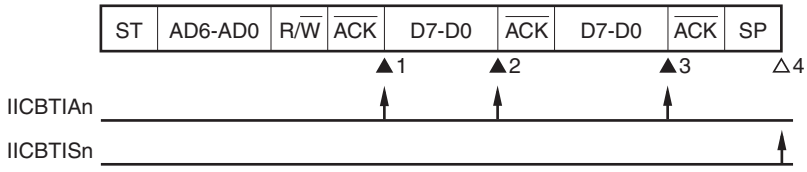
△4: IICBnSTR0 register = 0-000000 0001--00B

Note ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
\- Undefined
X don't care

23.8.2 Single transfer mode (slave device operation: during slave address reception (IICBnSTR0.IICBnSSC0 bit = 1))

(1) Start ~ Address ~ Data ~ Data ~ Stop

<1> When IICBnCTL0.IICBnSLWT bit is 0



▲1: IICBnSTR0 register = 0-0101X1 0110--00B

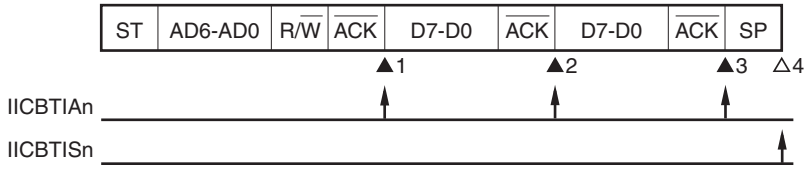
▲2: IICBnSTR0 register = 0-0101X0 0100--00B

▲3: IICBnSTR0 register = 0-0101X0 0100--00B

△4: IICBnSTR0 register = 0-000000 0001--00B

Note ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
- Undefined
X don't care

<2> When IICBnCTL0.IICBnSLWT bit is 1



▲1: IICBnSTR0 register = 0-0101X1 0110--00B

▲2: IICBnSTR0 register = 0-0101X1 0100--00B

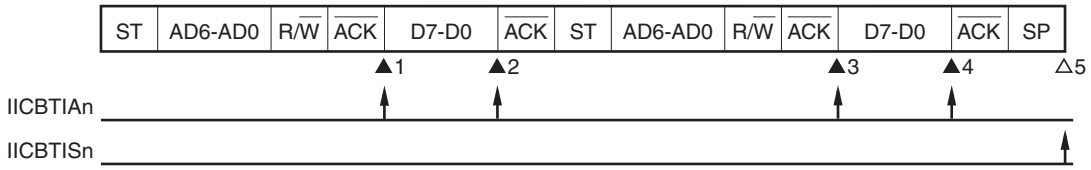
▲3: IICBnSTR0 register = 0-0101XX 0100--00B

△4: IICBnSTR0 register = 0-000000 0001--00B

Note ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
- Undefined
X don't care

(2) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

<1> When IICBnCTL0.IICBnSLWT bit is 0 (after restart, address match)



▲1: IICBnSTR0 register = 0-0101X1 0110--00B

▲2: IICBnSTR0 register = 0-0101X0 0100--00B

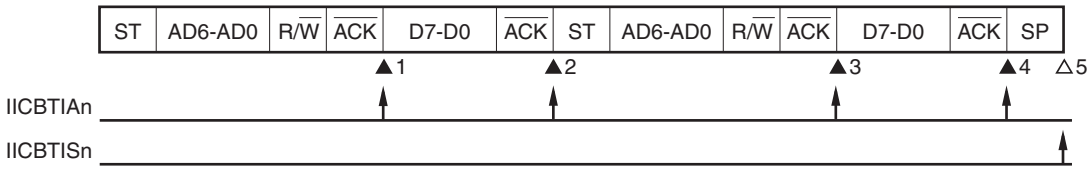
▲3: IICBnSTR0 register = 0-0101X1 0110--00B

▲4: IICBnSTR0 register = 0-0101X0 0100--00B

△5: IICBnSTR0 register = 0-000000 0001--00B

Note ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
- Undefined
X don't care

<2> When IICBnCTL0.IICBnSLWT bit is 1 (after restart, address match)



▲1: IICBnSTR0 register = 0-0101X1 0110--00B

▲2: IICBnSTR0 register = 0-0101XX 0100--00B

▲3: IICBnSTR0 register = 0-0101X1 0110--00B

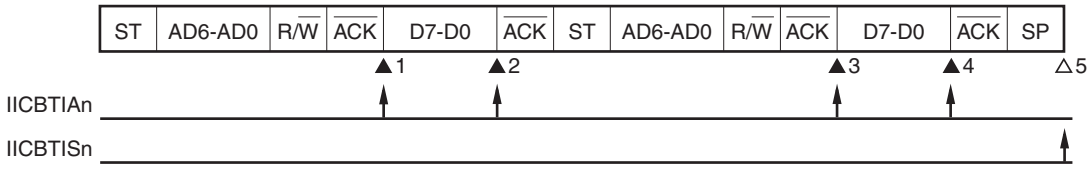
▲4: IICBnSTR0 register = 0-0101XX 0100--00B

△5: IICBnSTR0 register = 0-000000 0001--00B

Note ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
- Undefined
X don't care

(3) Start ~ Address ~ Data ~ Start ~ Code ~ Data ~ Stop

<1> When IICBnCTL0.IICBnSLWT bit is 0 (after restart, extension code reception)



▲1: IICBnSTR0 register = 0-0101X1 0110--00B

▲2: IICBnSTR0 register = 0-0101X0 0100--00B

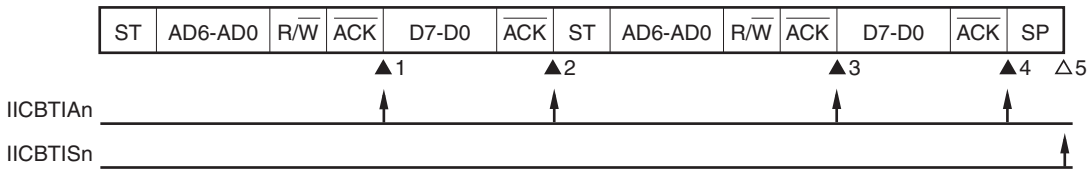
▲3: IICBnSTR0 register = 0-0110X1 0110--00B

▲4: IICBnSTR0 register = 0-0110X0 0100--00B

△5: IICBnSTR0 register = 0-000000 0001--00B

Note ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
\- Undefined
X don't care

<2> When IICBnCTL0.IICBnSLWT bit is 1 (after restart, extension code reception)



▲1: IICBnSTR0 register = 0-0101X1 0110--00B

▲2: IICBnSTR0 register = 0-0101XX 0100--00B

▲3: IICBnSTR0 register = 0-0110X1 0110--00B

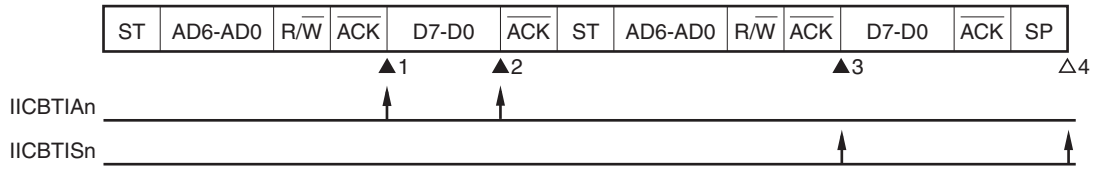
▲4: IICBnSTR0 register = 0-0110XX 0100--00B

△5: IICBnSTR0 register = 0-000000 0001--00B

Note ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
\- Undefined
X don't care

(4) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

<1> When IICBnCTL0.IICBnSLWT bit is 0 (after restart, address mismatch (extension code mismatch))



▲1: IICBnSTR0 register = 0-0101X1 0110--00B

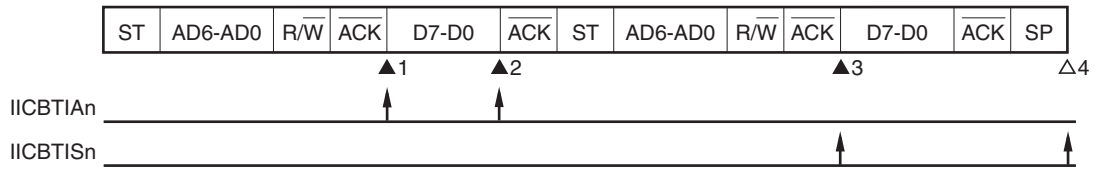
▲2: IICBnSTR0 register = 0-0101X0 0100--00B

▲3: IICBnSTR0 register = 0-0000X0 0110--00B

△4: IICBnSTR0 register = 0-000000 0001--00B

Note ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
- Undefined
X don't care

<2> When IICBnCTL0.IICBnSLWT bit is 1 (after restart, address mismatch (extension code mismatch))



▲1: IICBnSTR0 register = 0-0101X1 0110--00B

▲2: IICBnSTR0 register = 0-0101X0 0100--00B

▲3: IICBnSTR0 register = 0-0000X0 0110--00B

△4: IICBnSTR0 register = 0-000000 0001--00B

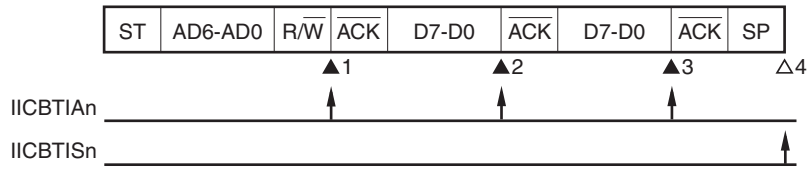
Note ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
- Undefined
X don't care

23.8.3 Single transfer mode (slave device operation: during extension code reception (IICBnSTR0.IICBnSSEX bit = 1))

The IICBn always participates in communications when it receives an extension code.

(1) Start ~ Code ~ Data ~ Data ~ Stop

<1> When IICBnCTL0.IICBnSLWT bit is 0



▲1: IICBnSTR0 register = 0-0110X0 0110--00B

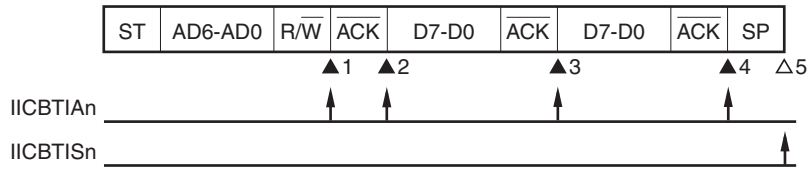
▲2: IICBnSTR0 register = 0-0110X0 0100--00B

▲3: IICBnSTR0 register = 0-0110X0 0100--00B

△4: IICBnSTR0 register = 0-000000 0001--00B

Note ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
\- Undefined
X don't care

<2> When IICBnCTL0.IICBnSLWT bit is 1



▲1: IICBnSTR0 register = 0-0110X0 0110--00B

▲2: IICBnSTR0 register = 0-0110X1 0110--00B

▲3: IICBnSTR0 register = 0-0110X0 0100--00B

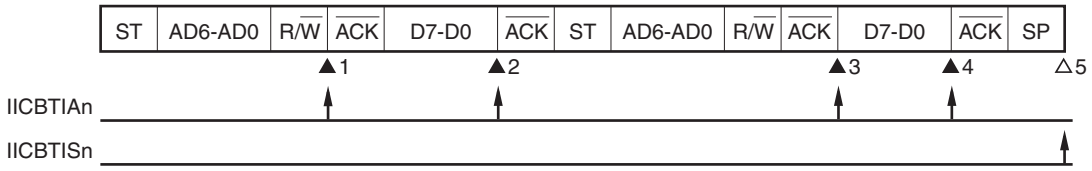
▲4: IICBnSTR0 register = 0-0110XX 0100--00B

△5: IICBnSTR0 register = 0-000000 0001--00B

Note ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
\- Undefined
X don't care

(2) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

<1> When IICBnCTL0.IICBnSLWT bit is 0 (after restart, address match)



▲1: IICBnSTR0 register = 0-0110X0 0110--00B

▲2: IICBnSTR0 register = 0-0110X0 0100--00B

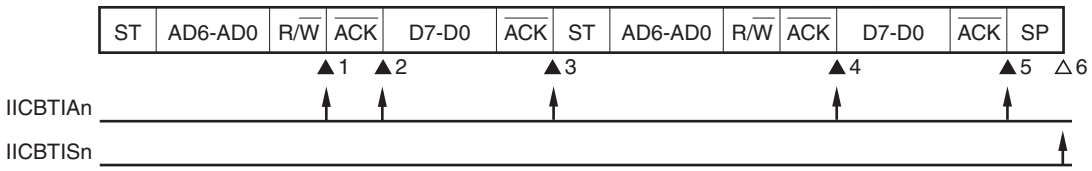
▲3: IICBnSTR0 register = 0-0101X1 0110--00B

▲4: IICBnSTR0 register = 0-0101X0 0100--00B

△5: IICBnSTR0 register = 0-000000 0001--00B

Note ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
- Undefined
X don't care

<2> When IICBnCTL0.IICBnSLWT bit is 1 (after restart, address match)



▲1: IICBnSTR0 register = 0-0110X0 0110--00B

▲2: IICBnSTR0 register = 0-0110X1 0110--00B

▲3: IICBnSTR0 register = 0-0110X0 0100--00B

▲4: IICBnSTR0 register = 0-0101X1 0110--00B

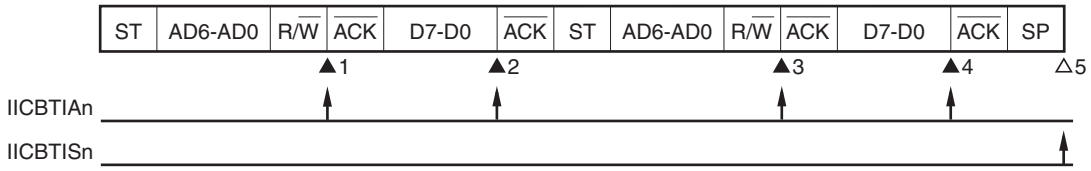
▲5: IICBnSTR0 register = 0-0101XX 0100--00B

△6: IICBnSTR0 register = 0-000000 0001--00B

Note ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
- Undefined
X don't care

(3) Start ~ Code ~ Data ~ Start ~ Code ~ Data ~ Stop

<1> When IICBnCTL0.IICBnSLWT bit is 0 (after restart, extension code reception)



▲1: IICBnSTR0 register = 0-0110X0 0110--00B

▲2: IICBnSTR0 register = 0-0110X0 0100--00B

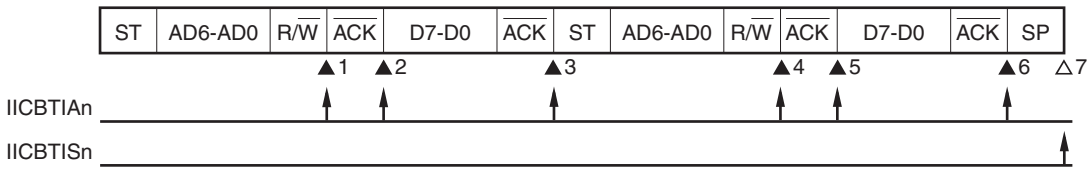
▲3: IICBnSTR0 register = 0-0110X0 0110--00B

▲4: IICBnSTR0 register = 0-0110X0 0100--00B

△5: IICBnSTR0 register = 0-000000 0001--00B

Note ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
- Undefined
X don't care

<2> When IICBnCTL0.IICBnSLWT bit is 1 (after restart, extension code reception)



▲1: IICBnSTR0 register = 0-0110X0 0110--00B

▲2: IICBnSTR0 register = 0-0110X1 0110--00B

▲3: IICBnSTR0 register = 0-0110XX 0100--00B

▲4: IICBnSTR0 register = 0-0110X0 0110--00B

▲5: IICBnSTR0 register = 0-0110X1 0110--00B

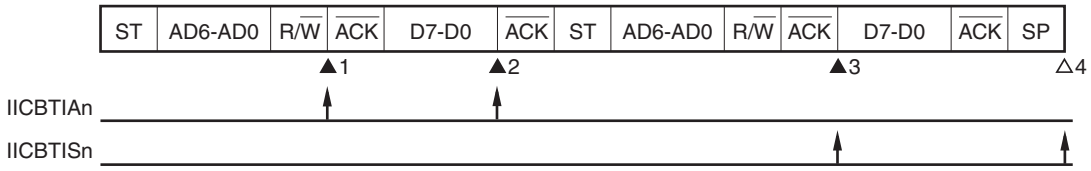
▲6: IICBnSTR0 register = 0-0110XX 0100--00B

△7: IICBnSTR0 register = 0-000000 0001--00B

Note ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
- Undefined
X don't care

(4) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

<1> When IICBnCTL0.IICBnSLWT bit is 0 (after restart, address mismatch (extension code mismatch))



▲1: IICBnSTR0 register = 0-0110X0 0110--00B

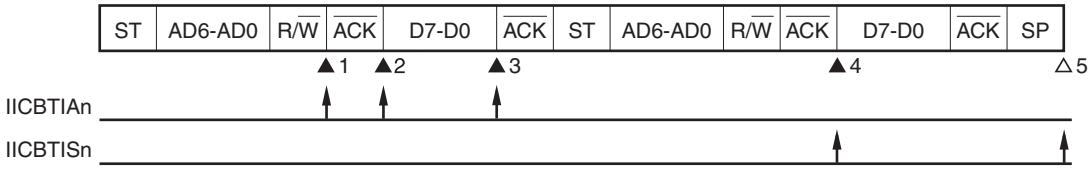
▲2: IICBnSTR0 register = 0-0110X0 0100--00B

▲3: IICBnSTR0 register = 0-0000X0 0110--00B

△4: IICBnSTR0 register = 0-000000 0001--00B

Note ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
\- Undefined
X don't care

<2> When IICBnCTL0.IICBnSLWT bit is 1 (after restart, address mismatch (extension code mismatch))



▲1: IICBnSTR0 register = 0-0110X0 0110--00B

▲2: IICBnSTR0 register = 0-0110X1 0110--00B

▲3: IICBnSTR0 register = 0-0000X0 0100--00B

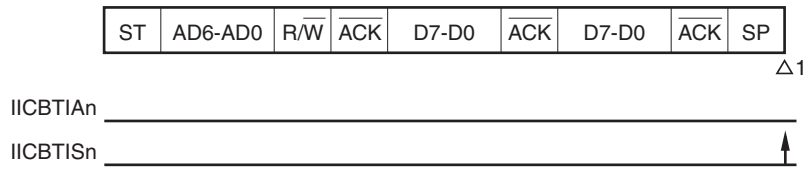
▲4: IICBnSTR0 register = 0-0000X0 0110--00B

△5: IICBnSTR0 register = 0-000000 0001--00B

Note ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
\- Undefined
X don't care

23.8.4 Single transfer mode (non-participation in communications)

(1) Start ~ Code ~ Data ~ Data ~ Stop



△1: IICBnSTR0 register = 0-000000 0001--00B

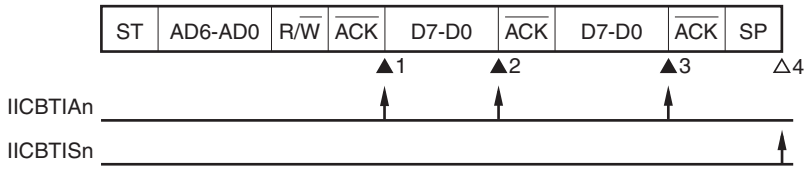
Note △ Output only when IICBnCTL0.IICBnSLSI = 1
\- Undefined

23.8.5 Single transfer mode (arbitration loss operation (IICBnSTR0.IICBnALDF bit = 1): operation as slave after arbitration loss)

When using IICBn as the master in a multi-master system, read the IICBnSTR0.IICBnALDF bit for each IICBTISn interrupt occurrence to confirm the arbitration result.

(1) Address match after arbitration loss

<1> When IICBnCTL0.IICBnSLWT bit is 0



▲1: IICBnSTR0 register = 0-0101X1 0110--01B (IICBnSTRC.IICBnCLAF bit = 1)

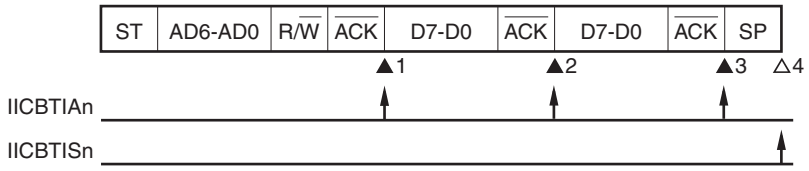
▲2: IICBnSTR0 register = 0-0101X0 0100--00B

▲3: IICBnSTR0 register = 0-0101X0 0100--00B

△4: IICBnSTR0 register = 0-000000 0001--00B

Note ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
\- Undefined
X don’t care

<2> When IICBnCTL0.IICBnSLWT bit is 1



▲1: IICBnSTR0 register = 0-0101X1 0110--01B (IICBnSTRC.IICBnCLAF bit = 1)

▲2: IICBnSTR0 register = 0-0101X1 0100--00B

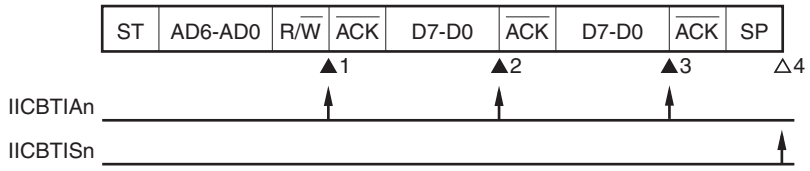
▲3: IICBnSTR0 register = 0-0101XX 0100--00B

△4: IICBnSTR0 register = 0-000000 0001--00B

Note ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
\- Undefined
X don’t care

(2) Upon extension code detection after arbitration loss

<1> When IICBnCTL0.IICBnSLWT bit is 0



▲1: IICBnSTR0 register = 0-0110X0 0110--01B (IICBnSTRC.IICBnCLAF bit = 1)

▲2: IICBnSTR0 register = 0-0110X0 0100--00B

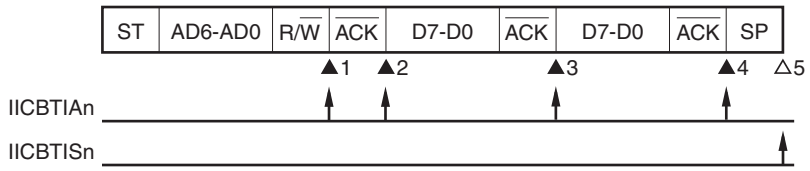
▲3: IICBnSTR0 register = 0-0110X0 0100--00B

△4: IICBnSTR0 register = 0-000000 0001--00B

Notes **1.** ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
\- Undefined
X don't care

2.n = 0 to 5

<2> When IICBnCTL0.IICBnSLWT bit is 1



▲1: IICBnSTR0 register = 0-0110X0 0110--01B (IICBnSTRC.IICBnCLAF bit = 1)

▲2: IICBnSTR0 register = 0-0110X1 0110--00B

▲3: IICBnSTR0 register = 0-0110X0 0100--00B

▲4: IICBnSTR0 register = 0-0110XX 0100--00B

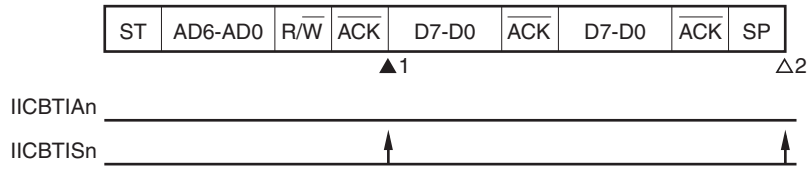
△5: IICBnSTR0 register = 0-000000 0001--00B

Note ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
\- Undefined
X don't care

23.8.6 Single transfer mode (arbitration loss operation (IICBnSTR0.IICBnALDF bit = 1): non-participation in communications after arbitration loss)

When using IICBn as the master in a multi-master system, read the IICBnSTR0.IICBnALDF bit for each IICBTISn interrupt occurrence to confirm the arbitration result.

(1) Arbitration loss during transmission of slave address



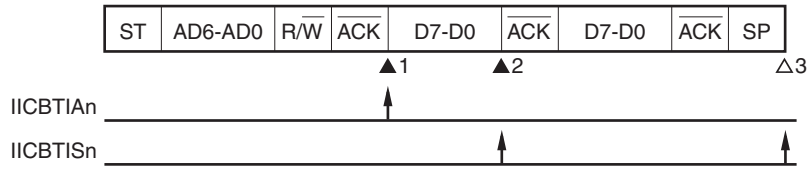
▲1: IICBnSTR0 register = 0-0000X1 0110--01B (IICBnSTRC.IICBnCLAF bit = 1)

△2: IICBnSTR0 register = 0-000000 0001--00B

Note ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
\- Undefined
X don't care

(2) Arbitration loss during data transfer

<1> When IICBnCTL0.IICBnSLWT bit is 0



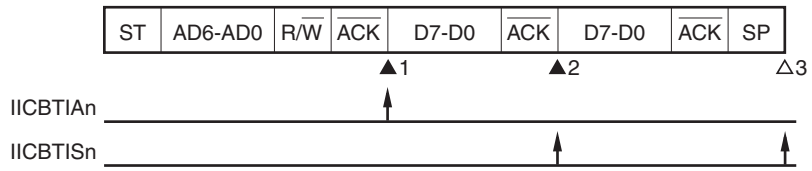
▲1: IICBnSTR0 register = 1-1000X1 0110--00B

▲2: IICBnSTR0 register = 0-0000X0 0100--01B (IICBnSTRC.IICBnCLAF bit = 1)

△3: IICBnSTR0 register = 0-000000 0001--00B

Note ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
\- Undefined
X don't care

<2> When IICBnCTL0.IICBnSLWT bit is 1



▲1: IICBnSTR0 register = 1-1000X1 0110--00B

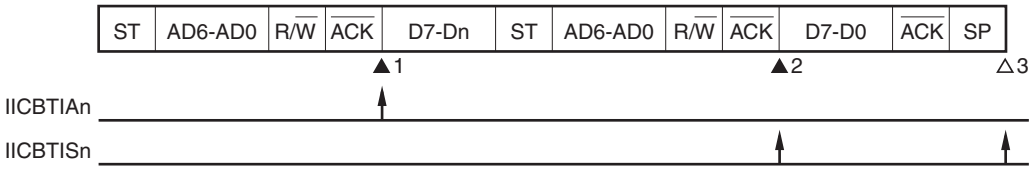
▲2: IICBnSTR0 register = 0-0000X0 0100--01B (IICBnSTRC.IICBnCLAF bit = 1)

△3: IICBnSTR0 register = 0-000000 0001--00B

Note ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
\- Undefined
X don't care

(3) Arbitration loss for the restart condition during data transfer

<1> When IICBnCTL0.IICBnSLWT bit is 1 (extension code mismatch, address mismatch)



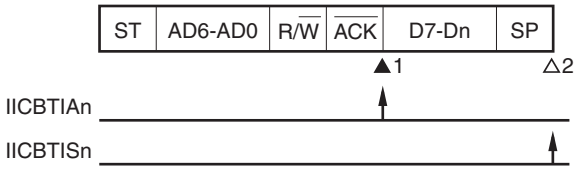
▲1: IICBnSTR0 register = 1-1000X1 0110--00B

▲2: IICBnSTR0 register = 0-0000X0 0100--01B (IICBnSTRC.IICBnCLAF bit = 1)

△3: IICBnSTR0 register = 0-000000 0001--00B

Note ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
\- Undefined
X don't care

(4) Arbitration loss for the stop condition during data transfer



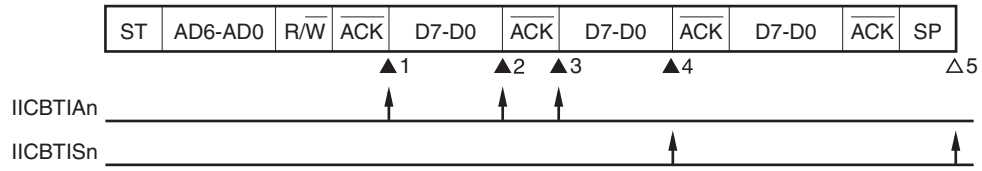
▲1: IICBnSTR0 register = 1-1000X1 0110--00B

△2: IICBnSTR0 register = 0-000000 0001--00B

Note ▲ Always output
△ Output regardless of the setting of IICBnCTL0.IICBnSLSI bit
\- Undefined
X don't care

(5) Arbitration loss because the SDAn signal is low level when attempting to output restart condition

<1> When IICBnCTL0.IICBnSLWT bit is 0



▲1: IICBnSTR0 register = 1-1000X1 0110--00B

▲2: IICBnSTR0 register = 1-1000X0 0100--00B (IICBnCTL0.IICBnSLWT bit = 1)

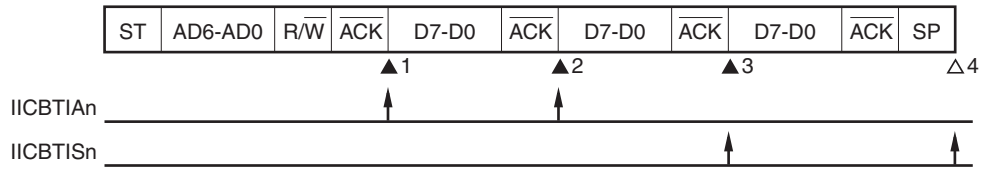
▲3: IICBnSTR0 register = 1-1000XX 0100--00B (IICBnCTL0.IICBnSLWT bit = 0, IICBnTRG.IICBnSTT bit = 1)

▲4: IICBnSTR0 register = 0-0000X0 0100--01B (IICBnSTRC.IICBnCLAF bit = 1)

△5: IICBnSTR0 register = 0-000000 0001--00B

Note ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
\- Undefined
X don't care

<2> When IICBnCTL0.IICBnSLWT bit is 1



▲1: IICBnSTR0 register = 1-1000X1 0110--00B

▲2: IICBnSTR0 register = 1-1000XX 0100--00B (IICBnCTL0.IICBnSLWT bit = 0, IICBnTRG.IICBnSTT bit = 1)

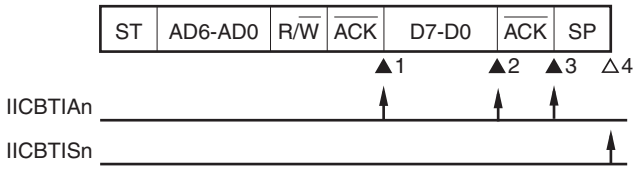
▲3: IICBnSTR0 register = 0-0000X0 0100--01B (IICBnSTRC.IICBnCLAF bit = 1)

△4: IICBnSTR0 register = 0-000000 0001--00B

Note ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
\- Undefined

(6) Arbitration loss for the stop condition when attempting to output restart condition

<1> When IICBnCTL0.IICBnSLWT bit is 0



▲1: IICBnSTR0 register = 1-1000X1 0110--00B

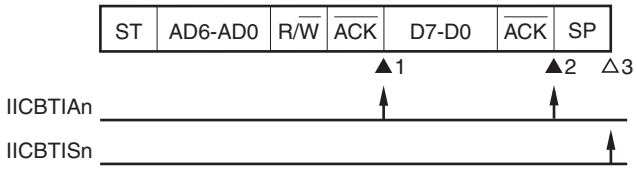
▲2: IICBnSTR0 register = 1-1000X0 0100--00B (IICBnCTL0.IICBnSLWT bit = 0)

▲3: IICBnSTR0 register = 1-0000XX 0100--00B (IICBnTRG.IICBnSTT bit = 1)

△4: IICBnSTR0 register = 0-000000 0001--01B

Note ▲ Always output
△ Output regardless of the setting of IICBnCTL0.IICBnSLSI bit
- Undefined
X don't care

<2> When IICBnCTL0.IICBnSLWT bit is 1



▲1: IICBnSTR0 register = 1-1000X1 0110--00B

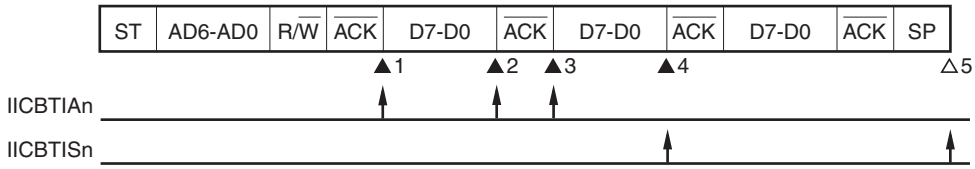
▲2: IICBnSTR0 register = 1-0000XX 0100--00B (IICBnTRG.IICBnSTT bit = 1)

△3: IICBnSTR0 register = 0-000000 0001--01B

Note ▲ Always output
△ Output regardless of the setting of IICBnCTL0.IICBnSLSI bit
- Undefined
X don't care

(7) Arbitration loss because the SDAn signal is low level when attempting to output stop condition

<1> When IICBnCTL0.IICBnSLWT bit is 0



▲1: IICBnSTR0 register = 1-1000X1 0110--00B

▲2: IICBnSTR0 register = 1-1000X0 0100--00B (IICBnCTL0.IICBnSLWT bit = 1)

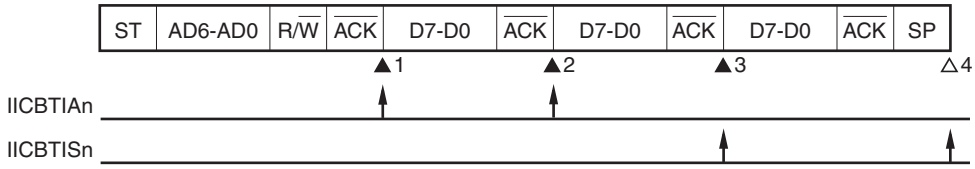
▲3: IICBnSTR0 register = 1-1000XX 0100--00B (IICBnCTL0.IICBnSLWT bit = 0, ICBnTRG.IICBnSPT bit = 1)

▲4: IICBnSTR0 register = 0-0000XX 0100--01B (IICBnSTRC.IICBnCLAF bit = 1)

△5: IICBnSTR0 register = 0-000000 0001--01B

Note ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
\- Undefined
X don't care

<2> When IICBnCTL0.IICBnSLWT bit is 1



▲1: IICBnSTR0 register = 1-1000X1 0110--00B

▲2: IICBnSTR0 register = 1-1000XX 0100--00B (IICBnTRG.IICBnSPT bit = 1)

▲3: IICBnSTR0 register = 0-0000XX 0100--01B (IICBnSTRC.IICBnCLAF bit = 1)

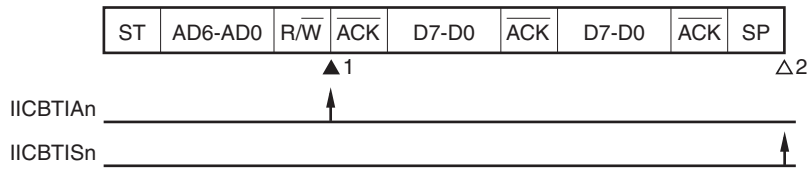
△4: IICBnSTR0 register = 0-000000 0001--01B

Note ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
\- Undefined
X don't care

23.8.7 Single transfer mode (arbitration loss operation (IICBnSTR0.IICBnALDF bit = 1): non-participation in communications after arbitration loss (during extension code transfer))

When using IICBn as the master in a multi-master system, read the IICBnSTR0.IICBnALDF bit for each IICBTISn interrupt occurrence to confirm the arbitration result.

(1) Arbitration loss during extension code transfer

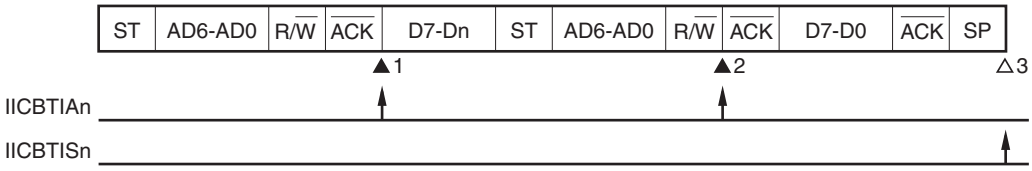


▲1: IICBnSTR0 register = 0-1100X0 0110--01B (IICBnSTRC.IICBnCLAF bit = 1, IICBnTRG.IICBnLRET bit = 1)

△2: IICBnSTR0 register = 0-000000 0001--01B

Note ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
\- Undefined
X don't care

(2) Arbitration loss for the restart condition during data transfer (extension code match)



▲1: IICBnSTR0 register = 1-1000X1 0110--00B

▲2: IICBnSTR0 register = 0-1100X0 0100--01B (IICBnSTRC.IICBnCLAF bit = 1, IICBnTRG.IICBnLRET bit = 1)

△3: IICBnSTR0 register = 0-000000 0001--01B

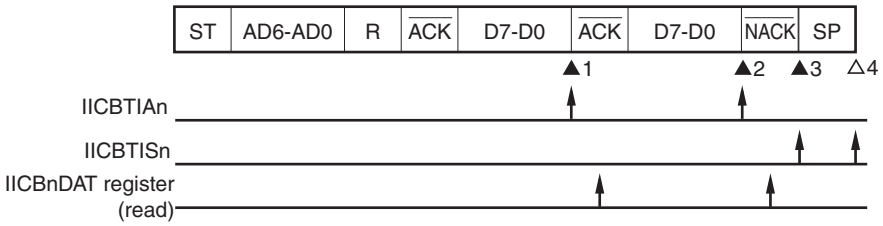
Note ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
\- Undefined

23.8.8 Continuous transfer mode (master device operation (reception))

Note The interrupts enclosed in brackets ([ ]) do not make the IICBn enter the wait state. Note that these interrupts are not output when a stop condition is detected.

(1) Start ~ Address ~ Data ~ Data ~ Stop

<1> When IICBnCTL0.IICBnSLWT bit is 0



[▲1: IICBnSTR0 register = 1-100000 0100--00B]

IICBnCTL0.IICBnSLAC bit = 0

IICBnDAT register read

[▲2: IICBnSTR0 register = 1-100000 0100--00B]

IICBnDAT register read

→ IICBnSTR0 register = 1-000000 0100--00B

▲3: IICBnSTR0 register = 1-010000 0100--00B

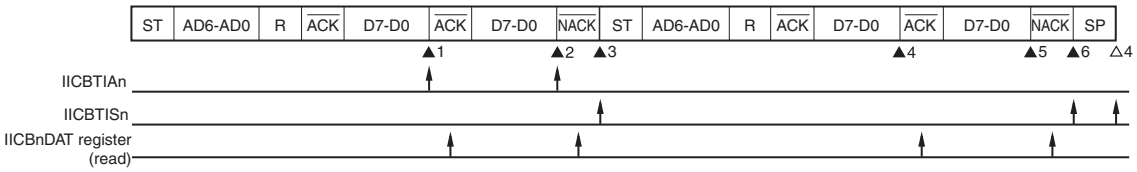
→ IICBnTRG.IICBnSPT bit = 1

△4: IICBnSTR0 register = 0-000000 0001--00B

Note ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
- Undefined

(2) Start ~ Address ~ Data × 2 ~ Start ~ Address ~ Data × 2 ~ Stop

<1> When IICBnCTL0.IICBnSLWT bit is 0



[▲1: IICBnSTR0 register = 1-100001 0100--00B]

IICBnCTL0.IICBnSLAC bit = 0

IICBnDAT register read

[▲2: IICBnSTR0 register = 1-100000 0100--00B]

IICBnCTL0.IICBnSLAC bit = 0

IICBnDAT register read

→ IICBnSTR0 register = 1-010000 0100--00B

▲3: IICBnSTR0 register = 1-010000 0100--00B

→ IICBnTRG.IICBnSTT bit = 1

[▲4: IICBnSTR0 register = 1-100000 0100--00B]

IICBnDAT register read

[▲5: IICBnSTR0 register = 1-100000 0100--00B]

IICBnCTL0.IICBnSLAC bit = 0

IICBnDAT register read

→ IICBnSTR0 register = 1-000000 0100--00B

▲6: IICBnSTR0 register = 1-010000 0100--00B

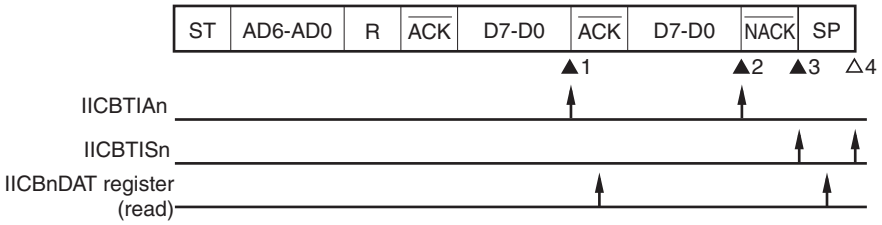
→ IICBnTRG.IICBnSTT bit = 1

△7: IICBnSTR0 register = 0-000000 0001--00B

Note ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
- Undefined

(3) Start ~ Code ~ Data ~ Data ~ Stop

<1> When IICBnCTL0.IICBnSLWT bit is 0



[▲1: IICBnSTR0 register = 1-101001 0100--00B]

IICBnDAT register read

→ IICBnSTR0 register = 1-0010001 0100--00B

[▲2: IICBnSTR0 register = 1-101000 0100--00B]

IICBnCTL0.IICBnSLAC bit = 0

IICBnDAT register read

→ IICBnSTR0 register = 1-011000 0100--00B

▲3: IICBnSTR0 register = 1-01000 0100--00B

→ IICBnTRG.IICBnSPT bit = 1

△4: IICBnTRG.IICBnSTR0 register = 0-000000 0001--00B

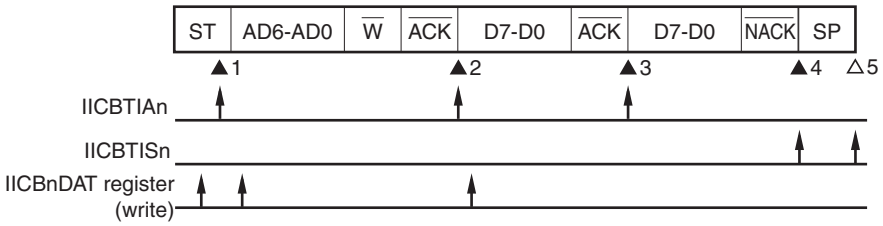
Note ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
\- Undefined

23.8.9 Continuous transfer mode (master device operation (transmission))

Note The interrupts enclosed in brackets ([ ]) do not make the IICBn enter the wait state. Note that these interrupts are not output when a stop condition is detected.

(1) Start ~ Address ~ Data ~ Data ~ Stop

<1> When IICBnCTL0.IICBnSLWT bit is 1



IICBnDAT register write (address)

[▲1: IICBnSTR0 register = X-0000X0 0X0X--00B]

IICBnDAT register write

[▲2: IICBnSTR0 register = 1-000011 0110--00B]

IICBnDAT register write

[▲3: IICBnSTR0 register = 1-000011 0100--00B]

▲4: IICBnSTR0 register = 1-010010 0100--00B

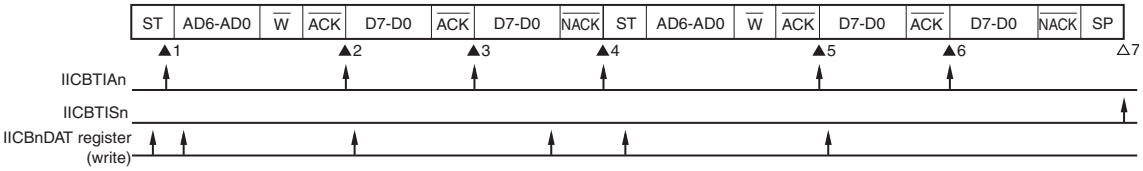
IICBnTRG.IICBnSPT bit = 1

△5: IICBnSTR0 register = 0-000000 0001--00B

Note ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
- Undefined
X don't care

(2) Start ~ Address ~ Data × 2 ~ Start ~ Address ~ Data × 2 ~ Stop

<1> When IICBnCTL0.IICBnSLWT bit is 1



IICBnDAT register write (address)

[▲1: IICBnSTR0 register = X-0000X0 0X0X--00B]

IICBnDAT register write

[▲2: IICBnSTR0 register = 1-000011 0110--00B]

IICBnDAT register write

[▲3: IICBnSTR0 register = 1-000011 0100--00B]

IICBnTRG.IICBnSTT bit = 1

IICBnDAT register write (address)

[▲4: IICBnSTR0 register = 1-000010 010X--00B]

IICBnDAT register write

[▲5: IICBnSTR0 register = 1-000011 0110--00B]

IICBnDAT register write

[▲6: IICBnSTR0 register = 1-000011 0110--00B]

IICBnTRG.IICBnSPT bit = 1

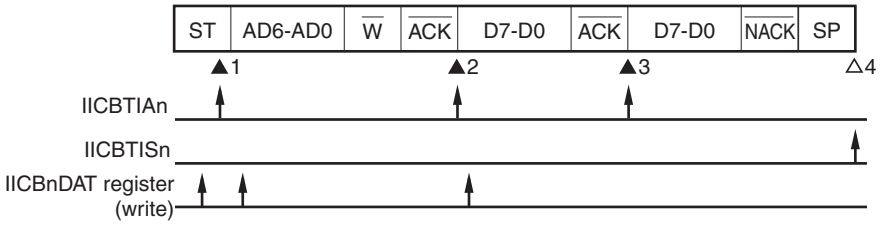
IICBnDAT register write

△7: IICBnSTR0 register = 0-000000 0001--00B

Note ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
- Undefined
X don't care

(3) Start ~ Code ~ Data ~ Data ~ Stop

<1> When IICBnCTL0.IICBnSLWT bit is 1



IICBnDAT register write (address)

[▲1: IICBnSTR0 register = X-0000X0 0X0X--00B]

IICBnDAT register write

[▲2: IICBnSTR0 register = 1-000011 0110--00B]

IICBnDAT register write

[▲3: IICBnSTR0 register = 1-000011 0100--00B]

IICBnTRG.IICBnSPT bit = 1

△4: IICBnSTR0 register = 0-000000 0001--00B

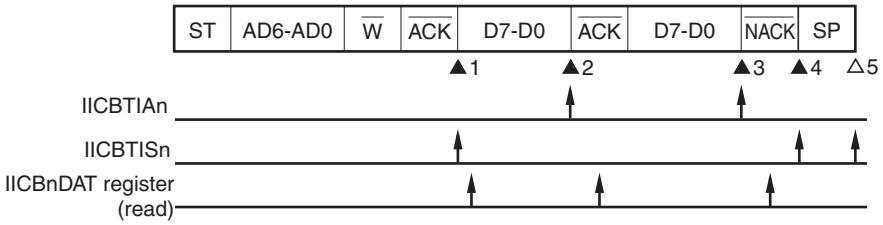
Note ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
\- Undefined
X don't care

23.8.10 Continuous transfer mode (slave device operation (reception): during slave address reception (IICBnSTR0.IICBnSSC0 bit = 1))

Note The interrupts enclosed in brackets ([ ]) do not make the IICBn enter the wait state. Note that these interrupts are not output when a stop condition is detected.

(1) Start ~ Address ~ Data ~ Data ~ Stop

<1> When IICBnCTL0.IICBnSLWT bit is 0



[▲1: IICBnSTR0 register = 0-100101 0110--00B]

IICBnDAT register read

[▲2: IICBnSTR0 register = 0-100100 0100--00B]

IICBnDAT register read

→ IICBnSTR0 register = 0-000100 0100--00B

[▲3: IICBnSTR0 register = 0-100100 0100--00B]

IICBnCTL0.IICBnSLAC bit = 0

IICBnDAT register read

→ IICBnSTR0 register = 0-000100 0100-00B

▲4: IICBnSTR0 register = 0-010100 0100-00B

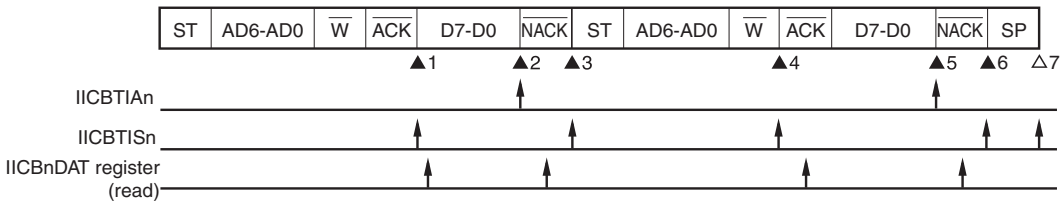
IICBnTRG.IICBnWRET bit = 1

△5: IICBnSTR0 register = 0-000000 0001--00B

Note ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
- Undefined

(2) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

<1> When IICBnCTL0.IICBnSLWT bit is 0 (after restart, address match)



[▲1: IICBnSTR0 register = 0-110101 0110--00B]

IICBnDAT register read

[▲2: IICBnSTR0 register = 0-100101 0100--00B]

IICBnCTL0.IICBnSLAC bit = 0

→ IICBnDAT register read

▲3: IICBnSTR0 register = 0-110101 0110--00B

IICBnTRG.IICBnWRET bit = 1

[▲4: IICBnSTR0 register = 0-100100 0110--00B]

IICBnCTL0.IICBnSLAC bit = 0

IICBnDAT register read

→ IICBnSTR0 register = 0-000100 0110--00B

[▲5: IICBnSTR0 register = 0-100100 0100--00B]

▲6: IICBnSTR0 register = 0-010100 0100--00B

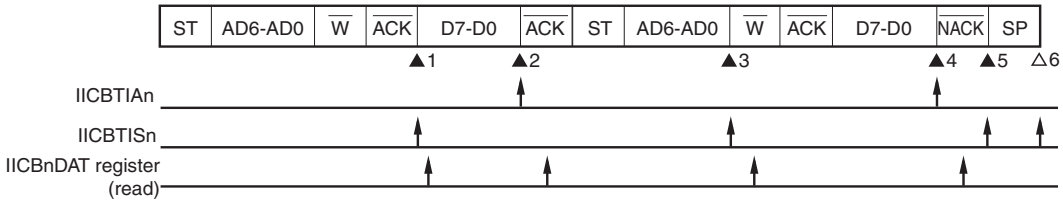
IICBnTRG.IICBnWRET bit = 1

△7: IICBnSTR0 register = 0-000000 0001--00B

Note ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
- Undefined

(3) Start ~ Address ~ Data ~ Start ~ Code ~ Data ~ Stop

<1> When IICBnCTL0.IICBnSLWT bit is 0 (after restart, extension code reception)



[▲1: IICBnSTR0 register = 0-100101 0110--00B]

IICBnDAT register read

[▲2: IICBnSTR0 register = 0-100100 0100--00B]

IICBnDAT register read

[▲3: IICBnSTR0 register = 0-100100 0110--00B]

IICBnCTL0.IICBnSLAC bit = 0

IICBnDAT register read

[▲4: IICBnSTR0 register = 0-100100 0110--00B]

IICBnDAT register read

▲5: IICBnSTR0 register = 0-111000 0100--00B

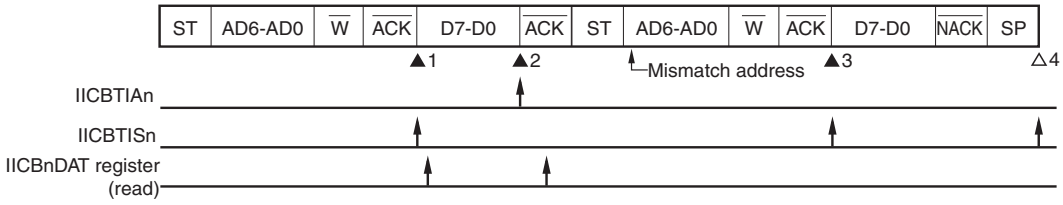
IICBnTRG.IICBnWRET bit = 1

△6: IICBnSTR0 register = 0-000000 0001--00B

Note ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
\- Undefined

(4) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

<1> When IICBnCTL0.IICBnSLWT bit is 0 (after restart, address mismatch (extension code mismatch))



[▲1: IICBnSTR0 register = 0-000101 0110--00B]

IICBnDAT register read

[▲2: IICBnSTR0 register = 0-100100 0100--00B]

IICBnDAT register read

[▲3: IICBnSTR0 register = 0-000000 0110--00B]

△4: IICBnSTR0 register = 0-000000 0001--00B

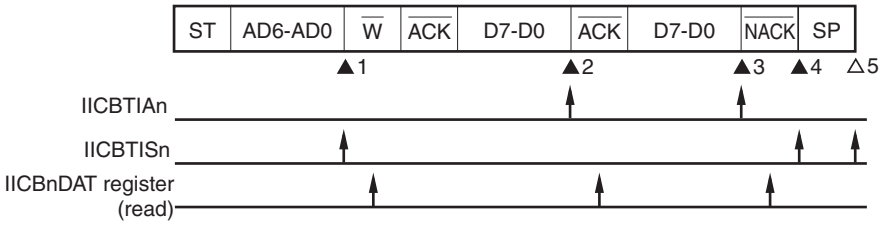
Note ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
\- Undefined
X don't care

23.8.11 Continuous transfer mode (slave device operation (reception): during extension code reception (IICBnSTR0.IICBnSSEX bit = 1))

Note The interrupts enclosed in brackets ([ ]) do not make the IICBn enter the wait state. Note that these interrupts are not output when a stop condition is detected.

(1) Start ~ Code ~ Data ~ Data ~ Stop

<1> When IICBnCTL0.IICBnSLWT bit is 0



[▲1: IICBnSTR0 register = 0-101000 0110--00B]

IICBnDAT register read

[▲2: IICBnSTR0 register = 0-101001 0110--00B]

IICBnCTL0.IICBnSLAC bit = 0

IICBnDAT register read

[▲3: IICBnSTR0 register = 0-10001 0100--00B]

IICBnDAT register read

▲4: IICBnSTR0 register = 0-111000 0100--00B

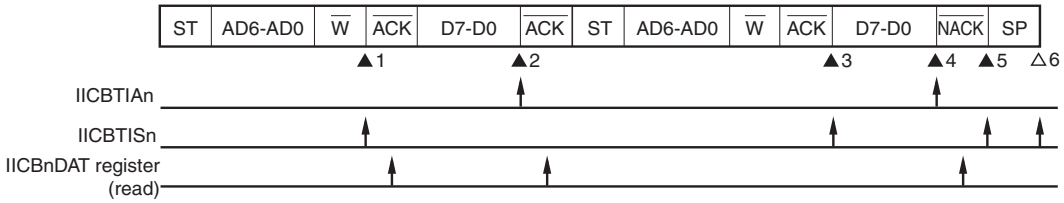
IICBnTRG.IICBnWRET bit = 1

△5: IICBnSTR0 register = 0-000000 0001--00B

Note ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
\- Undefined

(2) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

<1> When IICBnCTL0.IICBnSLWT bit is 0 (after restart, address match)



[▲1: IICBnSTR0 register = 0-101000 0110--00B]

IICBnDAT register read

[▲2: IICBnSTR0 register = 0-011000 0110--00B]

IICBnDAT register read

[▲3: IICBnSTR0 register = 0-111001 0100--00B]

IICBnCTL0.IICBnSLAC bit = 0

IICBnDAT register read

[▲4: IICBnSTR0 register = 0-010100 0110--00B]

IICBnDAT register read

▲5: IICBnSTR0 register = 0-110100 0100--00B

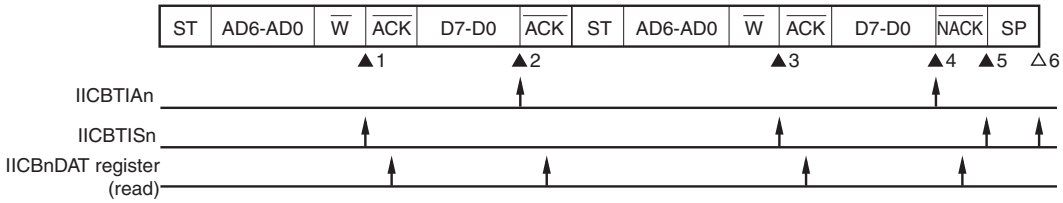
IICBnTRG.IICBnWRET bit = 1

△6: IICBnSTR0 register = 0-000000 0001--00B

Note ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
\- Undefined

(3) Start ~ Code ~ Data ~ Start ~ Code ~ Data ~ Stop

<1> When IICBnCTL0.IICBnSLWT bit is 0 (after restart, extension code reception)



[▲1: IICBnSTR0 register = 0-101000 0110--00B]

IICBnDAT register read

[▲2: IICBnSTR0 register = 0-011001 0110--00B]

IICBnDAT register read

[▲3: IICBnSTR0 register = 0-101000 0110--00B]

IICBnCTL0.IICBnSLAC bit = 0

IICBnDAT register read

[▲4: IICBnSTR0 register = 0-101001 0110--00B]

IICBnDAT register read

▲5: IICBnSTR0 register = 0-011000 0100--00B

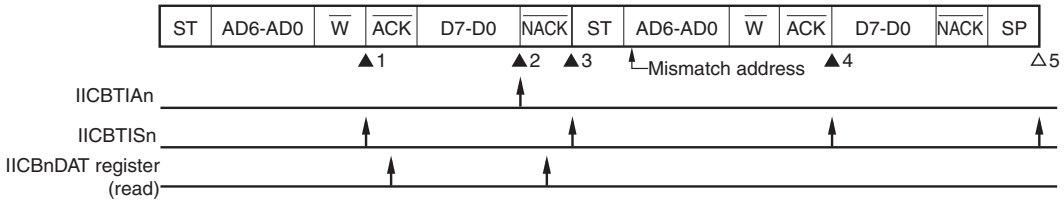
IICBnTRG.IICBnWRET bit = 1

△6: IICBnSTR0 register = 0-000000 0001--00B

Note ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
\- Undefined

(4) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

<1> When IICBnCTL0.IICBnSLWT bit is 0 (after restart, address mismatch (extension code mismatch))



[▲1: IICBnSTR0 register = 0-101000 0110--00B]

IICBnCTL0.IICBnSLAC bit = 0

IICBnDAT register read

[▲2: IICBnSTR0 register = 0-101001 0110--00B]

IICBnCTL0.IICBnSLAC bit = 0

▲3: IICBnSTR0 register = 0-010000 0100--00B

IICBnTRG.IICBnWRET bit = 1

[▲4: IICBnSTR0 register = 0-000000 0110--00B]

△5: IICBnSTR0 register = 0-000000 0001--00B

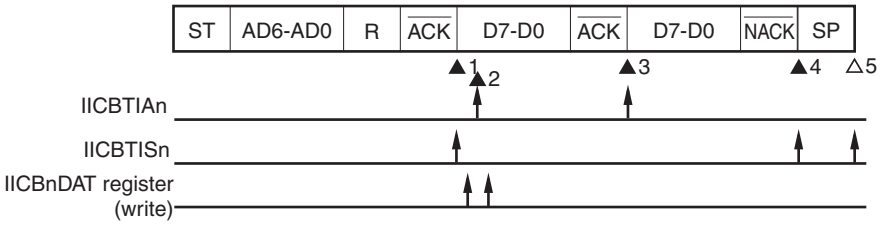
Note ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
\- Undefined
X don't care

23.8.12 Continuous transfer mode (slave device operation (transmission): during slave address reception (IICBnSTR0.IICBnSSC0 bit = 1))

Note The interrupts enclosed in brackets ([ ]) do not make the IICBn enter the wait state. Note that these interrupts are not output when a stop condition is detected.

(1) Start ~ Address ~ Data ~ Data ~ Stop

<1> When IICBnCTL0.IICBnSLWT bit is 1



▲1: IICBnSTR0 register = 0-110111 0110--00B

IICBnDAT register write

[▲2: IICBnSTR0 register = 0-00011X 0100--00B]

IICBnDAT register write

→ IICBnSTR0 register = 0-100011X 0100--00B

▲3: IICBnSTR0 register = 0-000111 0100--00B

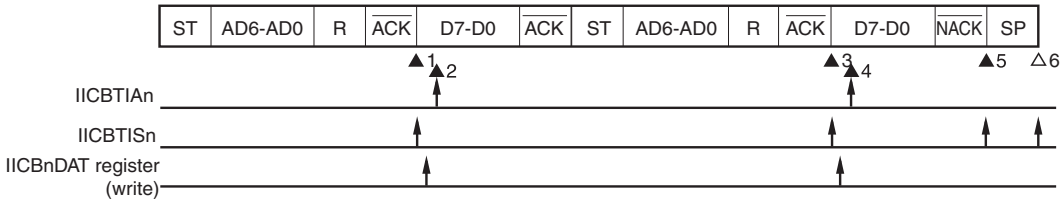
▲4: IICBnSTR0 register = 0-010110 0100--00B

△5: IICBnSTR0 register = 0-000000 0001--00B

Note
- ▲ Always output
- △ Output only when IICBnCTL0.IICBnSLSI = 1
- \- Undefined
- X don't care

(2) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

<1> When IICBnCTL0.IICBnSLWT bit is 1 (after restart, address match)



▲1: IICBnSTR0 register = 0-010111 0110--00B

IICBnDAT register write

[▲2: IICBnSTR0 register = 0-00111X 01X0--00B]

▲3: IICBnSTR0 register = 0-010111 0110--00B

IICBnDAT register write

[▲4: IICBnSTR0 register = 0-100101 01X0--00B]

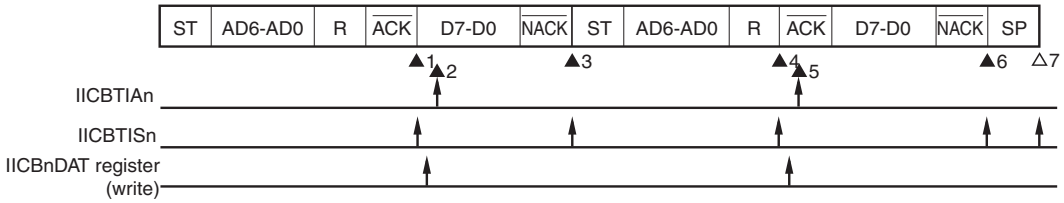
▲5: IICBnSTR0 register = 0-110100 0100--00B

△6: IICBnSTR0 register = 0-000000 0001--00B

Note ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
\- Undefined
X don't care

(3) Start ~ Address ~ Data ~ Start ~ Code ~ Data ~ Stop

<1> When IICBnCTL0.IICBnSLWT bit is 1 (after restart, extension code reception)



▲1: IICBnSTR0 register = 0-110111 0110--00B

IICBnDAT register write

[▲2: IICBnSTR0 register = 0-100111 0100--00B]

▲3: IICBnSTR0 register = 0-111010 0110--00B

▲4: IICBnSTR0 register = 0-111010 0110--00B

IICBnDAT register write

[▲5: IICBnSTR0 register = 0-111011 0110--00B]

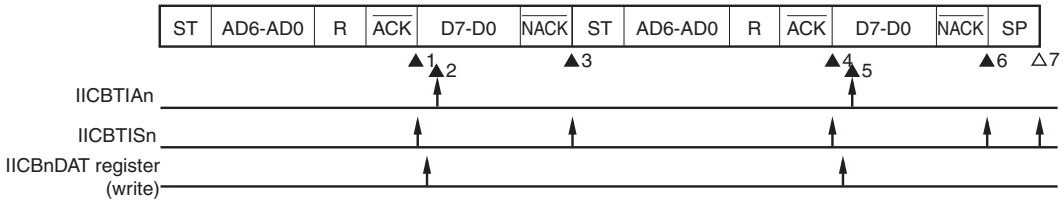
▲6: IICBnSTR0 register = 0-111010 0100--00B

△7: IICBnSTR0 register = 0-000000 0001--00B

Note ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
\- Undefined

(4) Start ~ Address ~ Data ~ Start ~ Address ~ Data ~ Stop

<1> When IICBnCTL0.IICBnSLWT bit is 1 (after restart, address mismatch (extension code mismatch))



▲1: IICBnSTR0 register = 0-110111 0110--00B

IICBnDAT register write

[▲2: IICBnSTR0 register = 0-100111 0100--00B]

▲3: IICBnSTR0 register = 0-000010 0100--00B

▲4: IICBnSTR0 register = 0-000011 0110--00B

IICBnDAT register write

[▲5: IICBnSTR0 register = 0-00001X 0100--00B]

▲6: IICBnSTR0 register = 0-000010 0100--00B

△7: IICBnSTR0 register = 0-000000 0001--00B

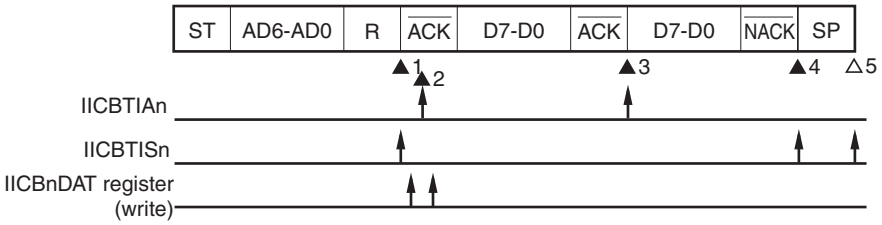
Note ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
\- Undefined
X don't care

23.8.13 Continuous transfer mode (slave device operation (transmission): during extension code reception (IICBnSTR0.IICBnSSEX bit = 1))

Note The interrupts enclosed in brackets ([ ]) do not make the IICBn enter the wait state. Note that these interrupts are not output when a stop condition is detected.

(1) Start ~ Code ~ Data ~ Data ~ Stop

<1> When IICBnCTL0.IICBnSLWT bit is 1



▲1: IICBnSTR0 register = 0-011010 0110--00B

IICBnDAT register write

[▲2: IICBnSTR0 register = 0-011011 0110--00B]

IICBnDAT register write

[▲3: IICBnSTR0 register = 0-011011 0100--00B]

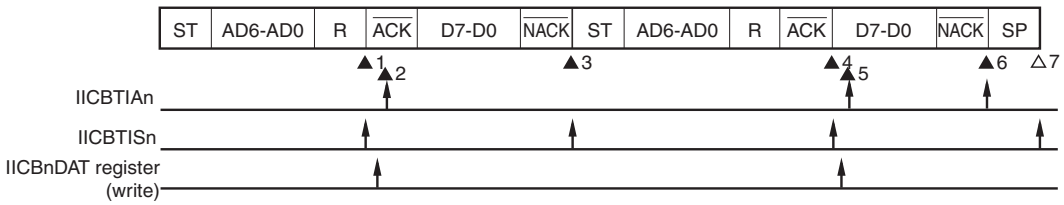
▲4: IICBnSTR0 register = 0-111010 0100--00B

△5: IICBnSTR0 register = 0-000010 0001--00B

Note ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
\- Undefined

(2) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

<1> When IICBnCTL0.IICBnSLWT bit is 1 (after restart, address match)



▲1: IICBnSTR0 register = 0-011000 0110--00B

IICBnDAT register write

[▲2: IICBnSTR0 register = 0-011001 0110--00B]

▲3: IICBnSTR0 register = 0-011000 0100--00B

▲4: IICBnSTR0 register = 0-010101 0110--00B

IICBnDAT register write

[▲5: IICBnSTR0 register = 0-010101 0110--00B]

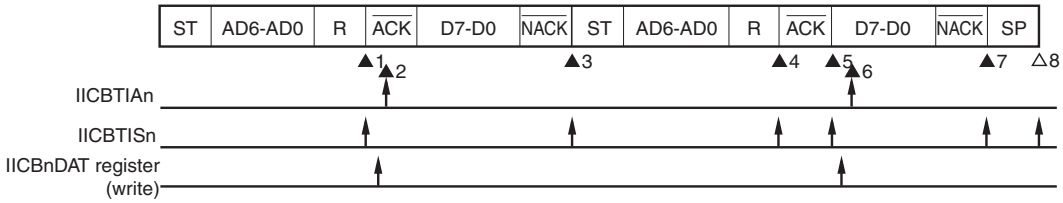
▲6: IICBnSTR0 register = 0-010100 0100--00B

△7: IICBnSTR0 register = 0-000000 0001--00B

Note ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
\- Undefined

(3) Start ~ Code ~ Data ~ Start ~ Code ~ Data ~ Stop

<1> When IICBnCTL0.IICBnSLWT bit is 1 (after restart, extension code reception)



▲1: IICBnSTR0 register = 0-011000 0110--00B

IICBnDAT register write

[▲2: IICBnSTR0 register = 0-011001 0110--00B]

▲3: IICBnSTR0 register = 0-011000 0100--00B

▲4: IICBnSTR0 register = 0-011000 0110--00B

▲5: IICBnSTR0 register = 0-011001 0110--00B

IICBnDAT register write

[▲6: IICBnSTR0 register = 0-011001 0110--00B]

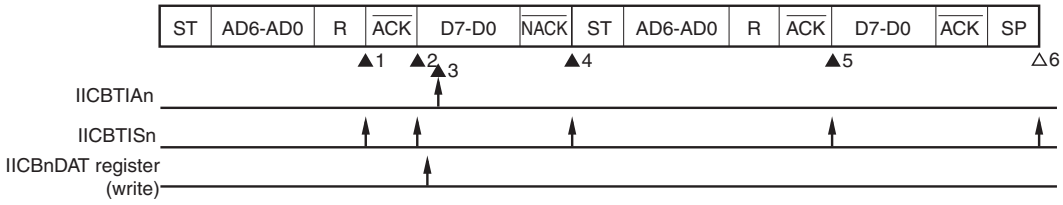
▲7: IICBnSTR0 register = 0-011000 0100--00B

△8: IICBnSTR0 register = 0-000000 0001--00B

Note ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
\- Undefined

(4) Start ~ Code ~ Data ~ Start ~ Address ~ Data ~ Stop

<1> When IICBnCTL0.IICBnSLWT bit is 1 (after restart, address mismatch (extension code mismatch))



▲1: IICBnSTR0 register = 0-011000 0110--00B

▲2: IICBnSTR0 register = 0-011001 0110--00B

IICBnDAT register write

[▲3: IICBnSTR0 register = 0-011010 0100--00B]

▲4: IICBnSTR0 register = 0-000000 0100--00B

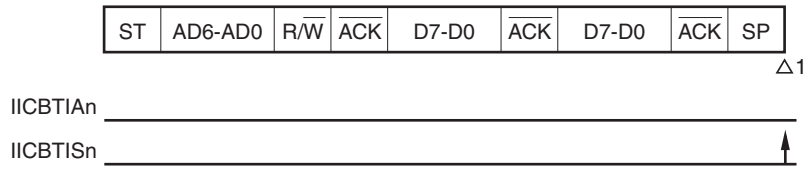
▲5: IICBnSTR0 register = 0-000000 0110--00B

△6: IICBnSTR0 register = 0-000000 0001--00B

Note ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
\- Undefined

23.8.14 Continuous transfer mode (non-participation in communications)

(1) Start ~ Code ~ Data ~ Data ~ Stop



△1: IICBnSTR0 register = 0-0000X0 0001--00B

Note △ Output only when IICBnCTL0.IICBnSLSI = 1
\- Undefined

23.8.15 Continuous transfer mode (arbitration loss operation (IICBnSTR0.IICBnALDF bit = 1) (when address was transferred during reception): operation as slave after arbitration loss)

When using IICBn as the master in a multi-master system, read the IICBnSTR0.IICBnALDF bit for each IICBTISn interrupt occurrence to confirm the arbitration result.

(1) Address match after arbitration loss

<1> During reception, when IICBnCTL0.IICBnSLWT bit is 0

[▲1: IICBnSTR0 register = 0-100101 0110--01B]

IICBnSTRC.IICBnCLAF bit = 1

IICBnDAT register read

[▲2: IICBnSTR0 register = 0-100101 0100--00B]

IICBnCTL0.IICBnSLAC bit = 0

IICBnDAT register read

[▲3: IICBnSTR0 register = 0-100100 0100--00B]

IICBnDAT register read

▲4: IICBnSTR0 register = 0-010100 0100--00B

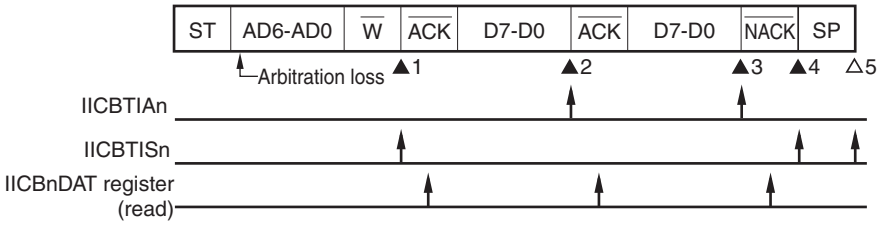
IICBnTRG.IICBnWRET bit = 1

△5: IICBnSTR0 register = 0-000000 0001--00B

Note ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
\- Undefined

(2) Upon extension code detection after arbitration loss

<1> During reception, when IICBnCTL0.IICBnSLWT bit is 0



[▲1: IICBnSTR0 register = 0-101000 0110--01B]

IICBnSTRC.IICBnCLAF bit = 1

IICBnDAT register read

[▲2: IICBnSTR0 register = 0-101000 0110--00B]

IICBnCTL0.IICBnSLAC bit = 0

IICBnDAT register read

[▲3: IICBnSTR0 register = 0-101000 0100--00B]

IICBnDAT register read

▲4: IICBnSTR0 register = 0-011000 0100--00B]

IICBnTRG.IICBnWRET bit = 1

△5: IICBnSTR0 register = 0-000000 0001--00B

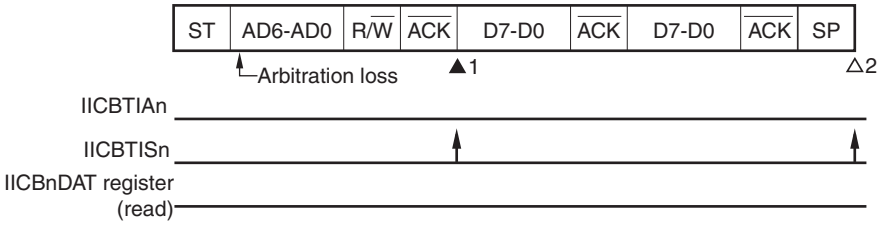
Note ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
\- Undefined

23.8.16 Continuous transfer mode (arbitration loss operation (IICBnSTR0.IICBnALDF bit = 1) (when address was transferred during reception): non-participation in communications after arbitration loss)

When using IICBn as the master in a multi-master system, read the IICBnSTR0.IICBnALDF bit for each IICBTISn interrupt occurrence to confirm the arbitration result.

(1) Arbitration loss during slave address transmission

<1> During reception, when IICBnCTL0.IICBnSLWT bit is 0



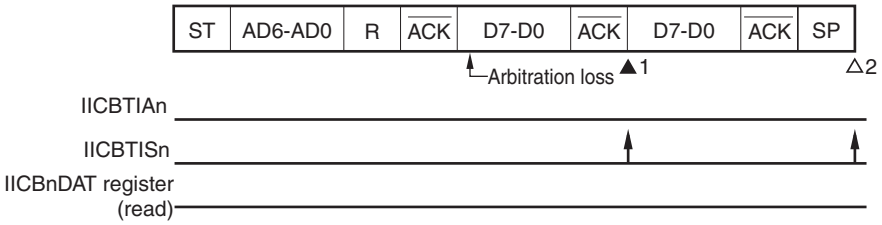
▲1: IICBnSTR0 register = 0-000001 0110--01B (IICBnSTRC.IICBnCLAF bit = 1)

△2: IICBnSTR0 register = 0-000000 0001--00B

Note ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
\- Undefined

(2) Arbitration loss during data transfer

<1> During reception, when IICBnCTL0.IICBnSLWT bit is 1



[▲1: IICBnSTR0 register = 0-000000 0100--01B]

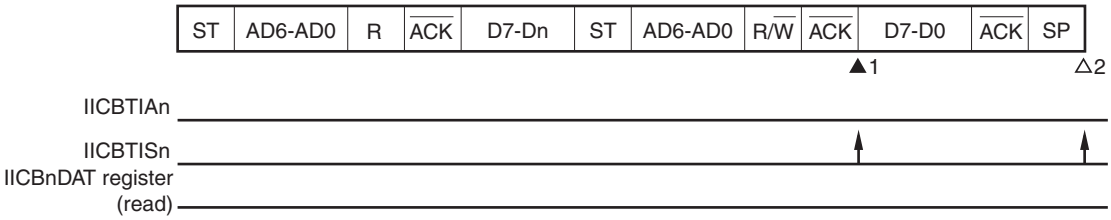
IICBnSTRC.IICBnCLAF bit = 1

△2: IICBnSTR0 register = 0-000000 0001--00B

Note ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
\- Undefined

(3) Arbitration loss for the restart condition during data transfer

<1> During reception, when IICBnCTL0.IICBnSLWT bit is 1 (extension code mismatch, address mismatch)



[▲1: IICBnSTR0 register = 0-000001 0100--01B]

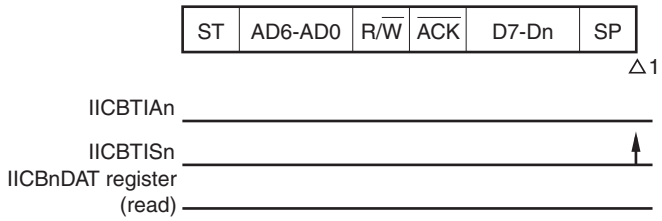
IICBnSTRC.IICBnCLAF bit = 1

△2: IICBnSTR0 register = 0-000000 0001--00B

Note ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
- Undefined

(4) Arbitration loss for the stop condition during data transfer

<1> During reception, when IICBnCTL0.IICBnSLWT bit is 1



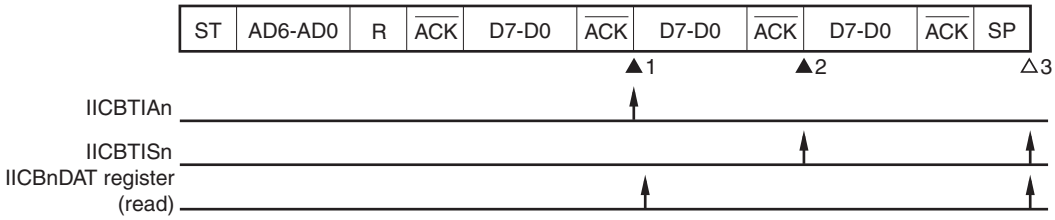
△1: IICBnSTR0 register = 0-000000 0001--01B

IICBnSTRC.IICBnCLAF bit = 1

Note △ Output only when IICBnCTL0.IICBnSLSI = 1
- Undefined

(5) Arbitration loss because the SDAn signal is low level when attempting to output restart condition

<1> When IICBnCTL0.IICBnSLWT bit is 1



[▲1: IICBnSTR0 register = 1-1000XX 0100--00B]

IICBnDAT register read

IICBnTRG.IICBnSTT bit = 1

▲2: IICBnSTR0 register = 0-000000 0100--01B

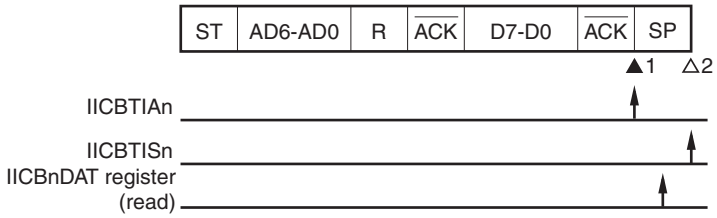
IICBnSTRC.IICBnCLAF bit = 1

△3: IICBnSTR0 register = 0-000000 0001--00B

Note ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
\- Undefined
X don't care

(6) Arbitration loss for the stop condition when attempting to output restart condition

<1> When IICBnCTL0.IICBnSLWT bit is 1



[▲1: IICBnSTR0 register = 1-000001 0100--00B]

IICBnDAT register read

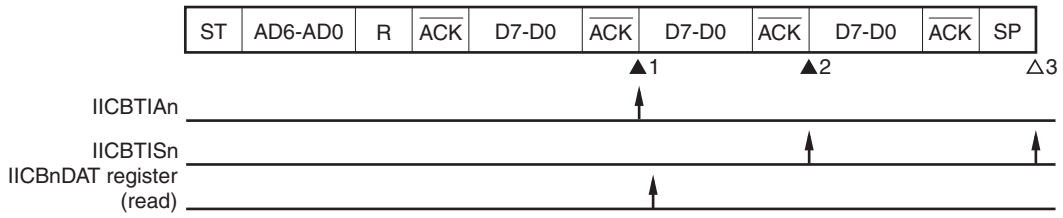
IICBnTRG.IICBnSTT bit = 1

△2: IICBnSTR0 register = 0-000000 0001--01B

Note ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
- Undefined

(7) Arbitration loss because the SDAn signal is low level when attempting to output stop condition

<1> When IICBnCTL0.IICBnSLWT bit is 1



[▲1: IICBnSTR0 register = 1-1000XX 0100--00B]

IICBnDAT register read

IICBnTRG.IICBnSPT bit = 1

[▲2: IICBnSTR0 register = 0-0000XX 0100--01B (IICBnSTRC.IICBnCLAF bit = 1)

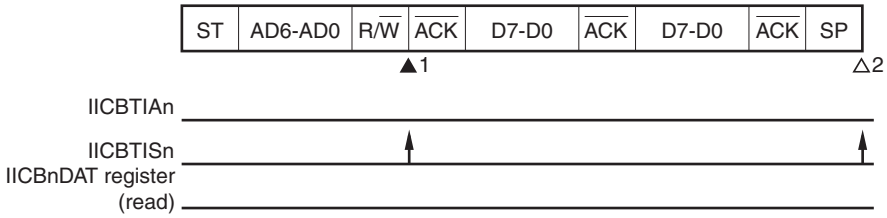
△3: IICBnSTR0 register = 0-000000 0001--01B

Note ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
- Undefined
X don't care

23.8.17 Continuous transfer mode (arbitration loss operation (IICBnSTR0.IICBnALDF bit = 1) (when address was transferred during reception): non-participation in communications after arbitration loss (during extension code transfer))

When using IICBn as the master in a multi-master system, read the IICBnSTR0.IICBnALDF bit for each IICBTISn interrupt occurrence to confirm the arbitration result.

(1) Arbitration loss during extension code transfer



[▲1: IICBnSTR0 register = 0-1000X0 0110--01B]

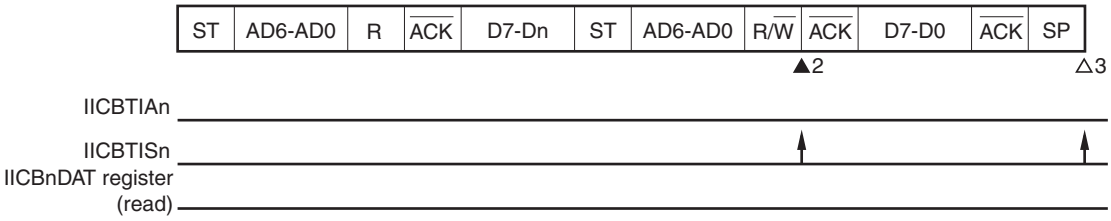
IICBnSTRC.IICBnCLAF bit = 1

IICBnTRG.IICBnLRET bit = 1

△2: IICBnSTR0 register = 0-000000 0001--01B

Note ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
\- Undefined
X don’t care

(2) Arbitration loss for the restart condition during data transfer (extension code match)



▲1: IICBnSTR0 register = 1-0000X1 0110--00B

▲2: IICBnSTR0 register = 0-0100X0 0100--01B (IICBnSTRC.IICBnCLAF bit = 1, IICBnTRG.IICBnLRET bit = 1)

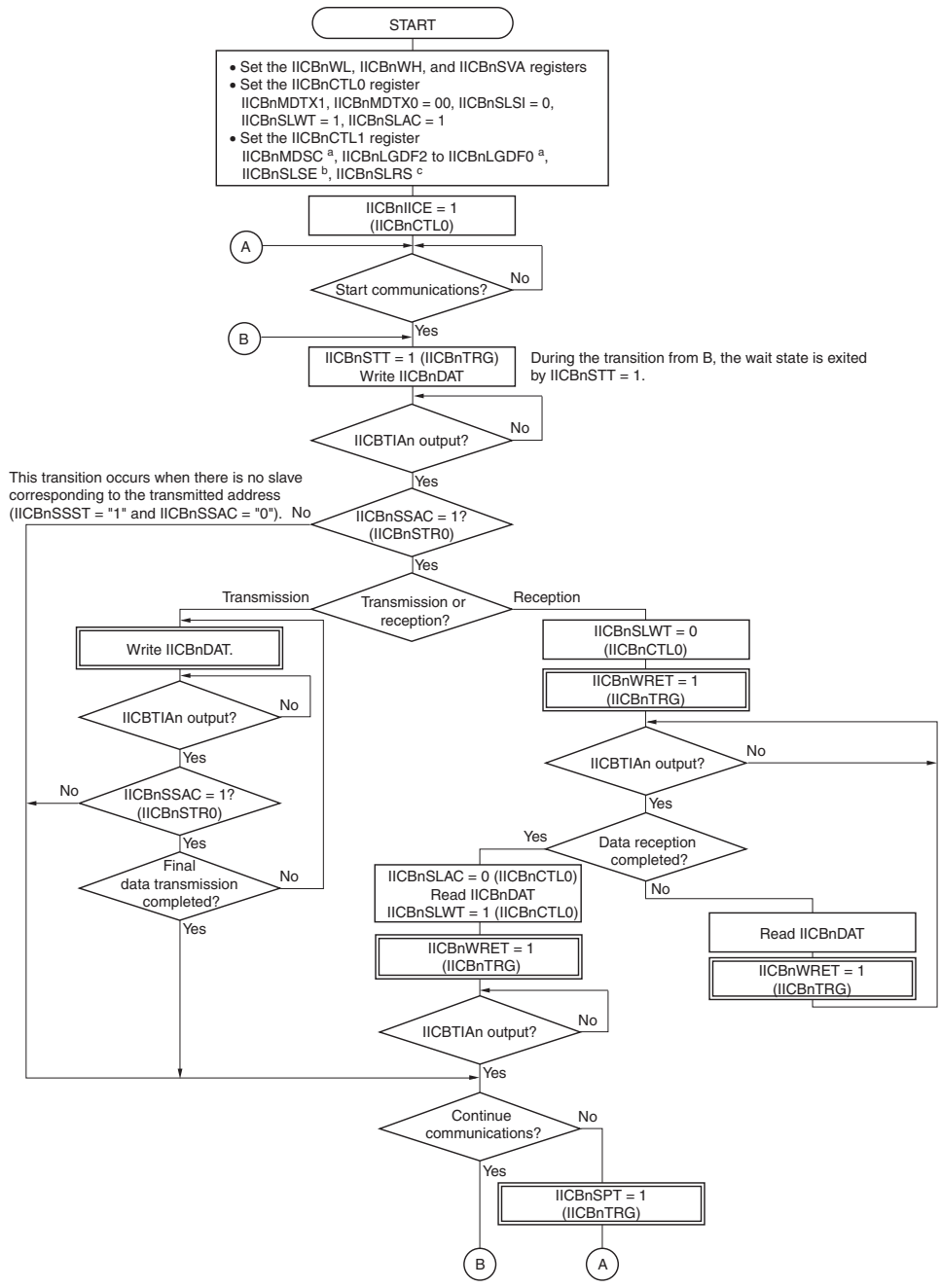
△3: IICBnSTR0 register = 0-000000 0001--01B

Note ▲ Always output
△ Output only when IICBnCTL0.IICBnSLSI = 1
\- Undefined

23.9 Setting Sequence

23.9.1 Single master environment

(1) Master operate setting sequence during single transfer mode

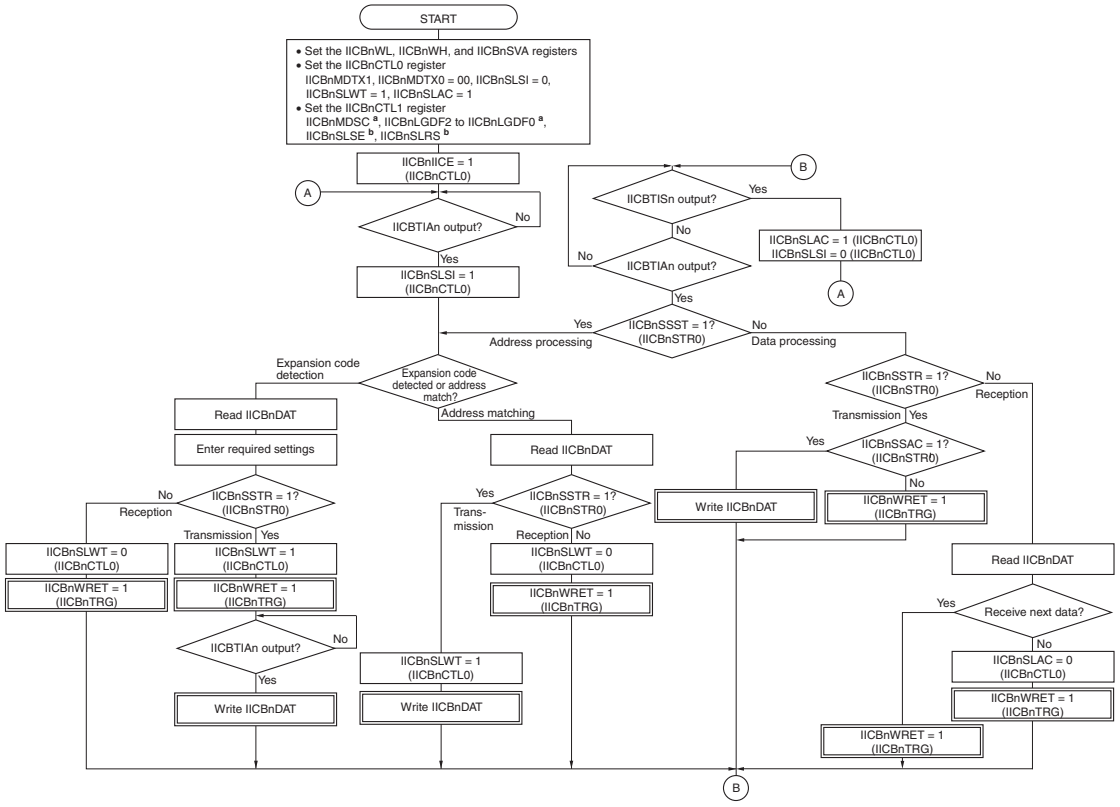


a) Enter settings according to the environment.
b) Set 1 to enable output of a start condition in the initial communication state. Otherwise, set 0.
c) Values are not referenced in this environment.

Note The double-boxed items indicate processing that exits the wait state.

Figure 23-14 Master operate setting sequence during single transfer mode (single master environment)

(2) Slave operate setting sequence during single transfer mode

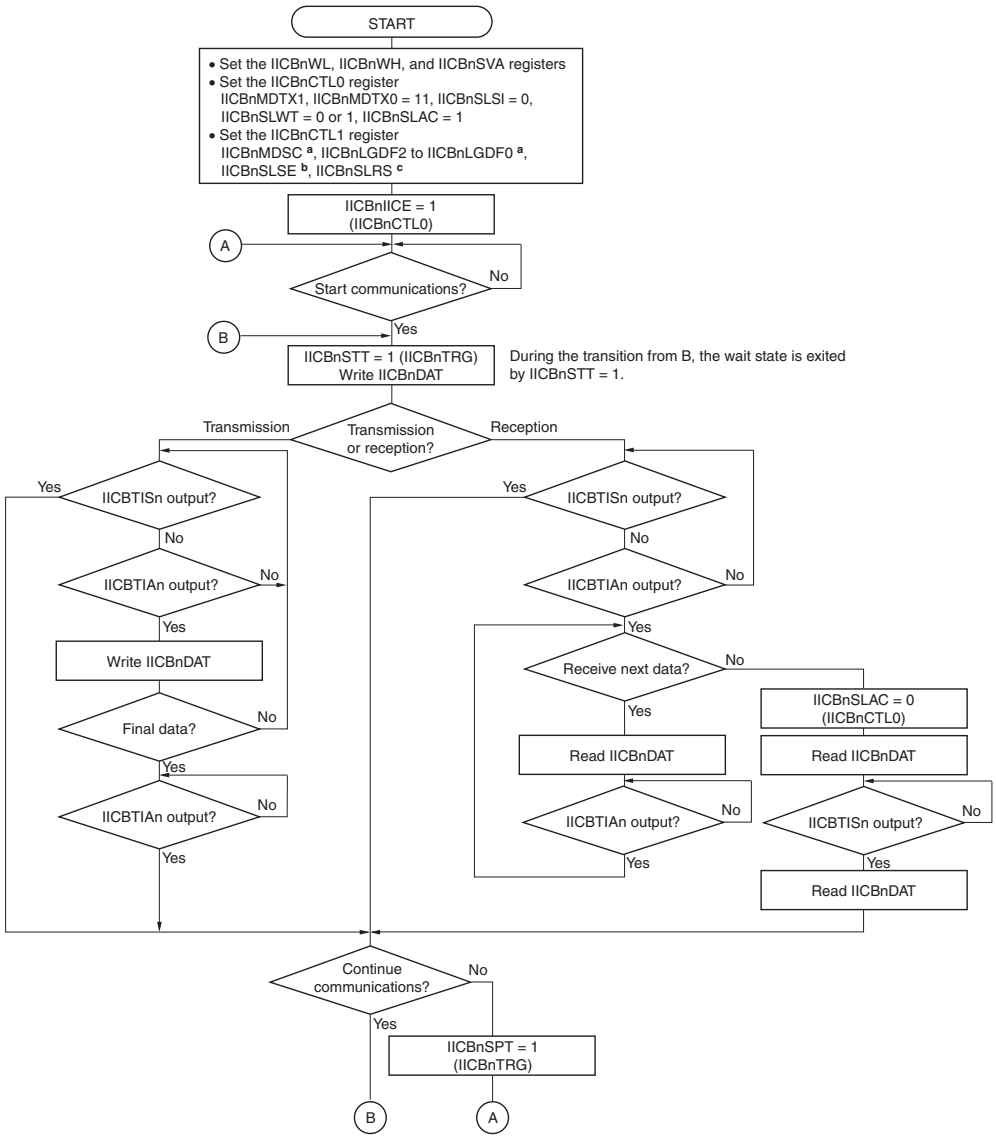


a) Enter settings according to the environment.
b) Values are not referenced in this environment.

Note The double-boxed items indicate processing that exits the wait state.

Figure 23-15 Slave operate setting sequence during single transfer mode (single master environment)

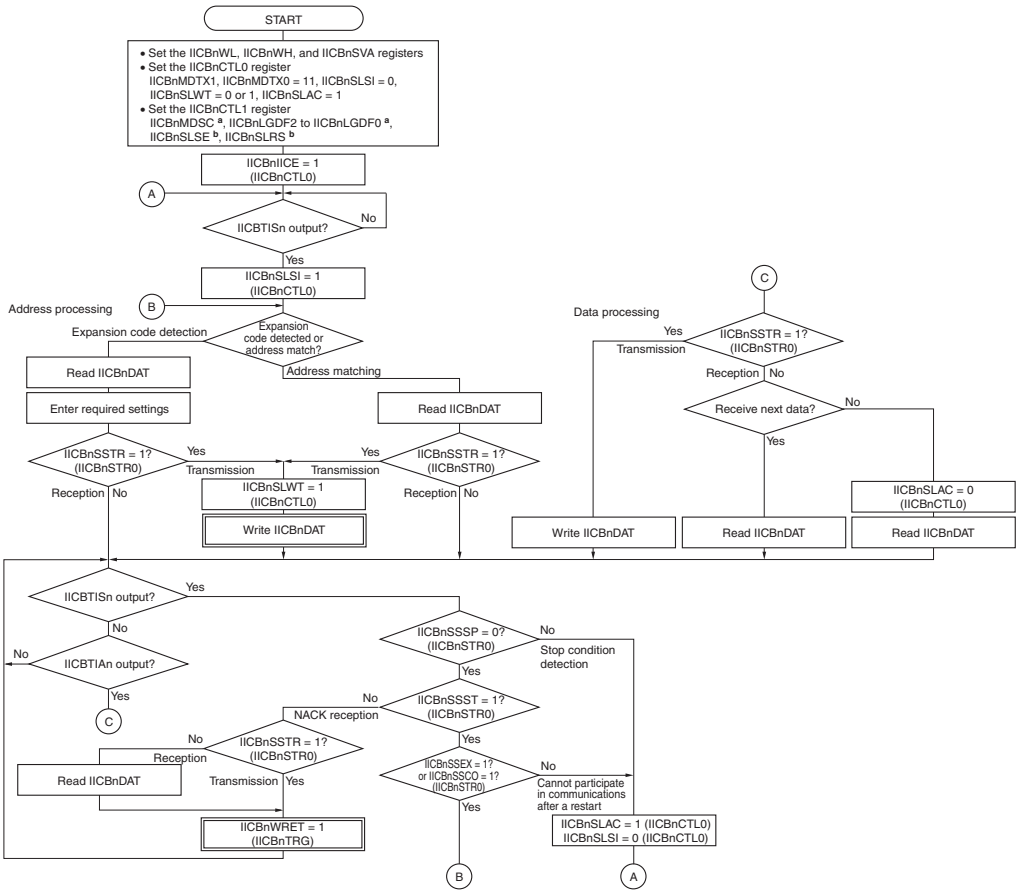
(3) Master operate setting sequence during continuous transfer mode



a) Enter settings according to the environment.
b) Set 1 to enable output of a start condition in the initial communication state. Otherwise, set 0.
c) Values are not referenced in this environment.

Figure 23-16 Master operate setting sequence during continuous transfer mode (single master environment)

(4) Slave operate setting sequence during continuous transfer mode



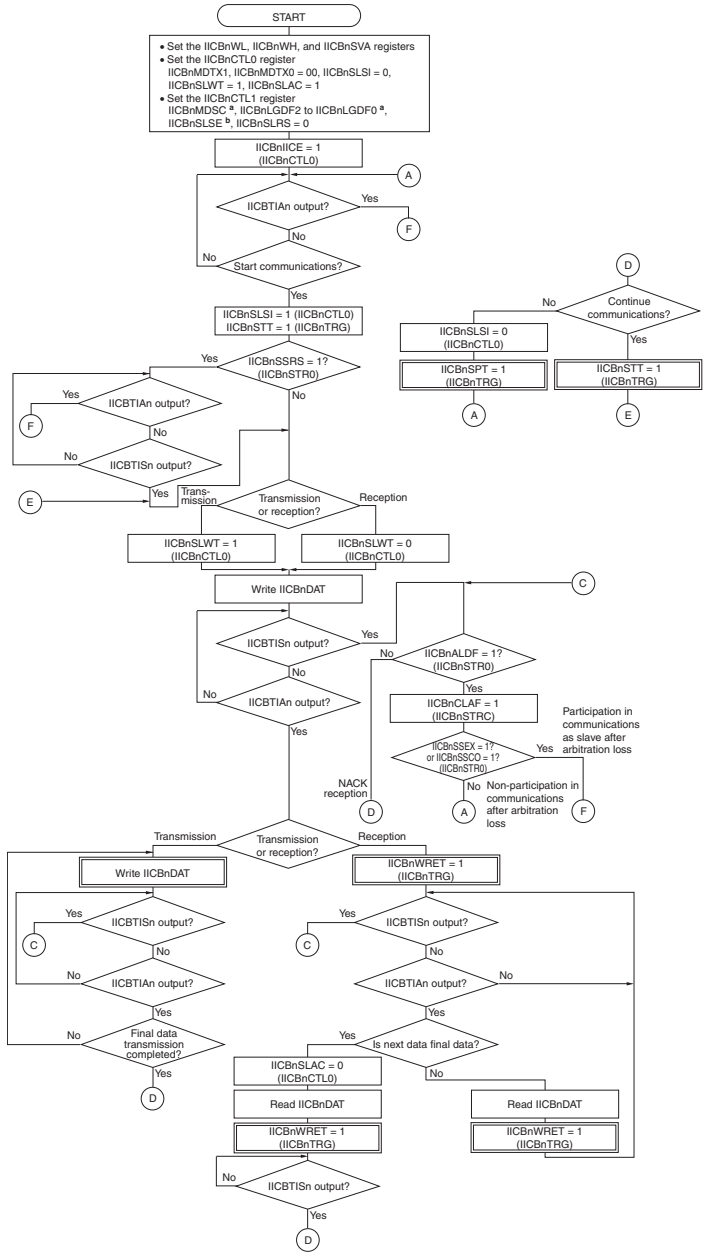
a) Enter settings according to the environment.
b) Values are not referenced in this environment.

Note The double-boxed items indicate processing that exits the wait state.

Figure 23-17 Slave operate setting sequence during continuous transfer mode (single master environment)

23.9.2 Multi-master environment

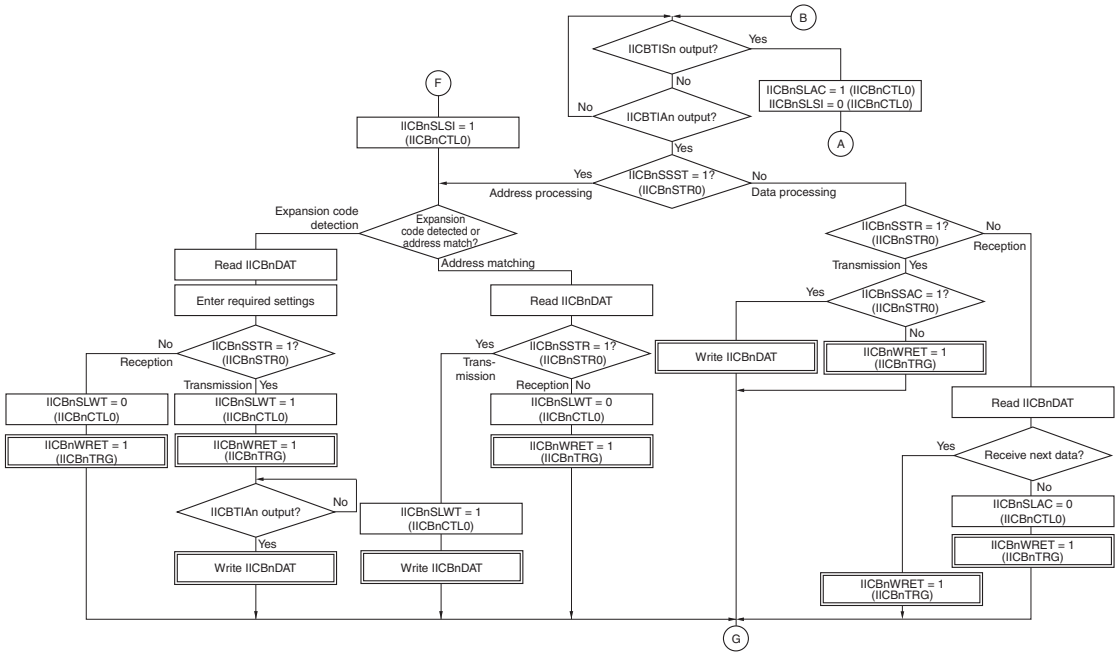
(1) Single transfer mode setting sequence when communication reserve function is enabled (IICBnCTL1.IICBnSLRS bit = 0)



a) Enter settings according to the environment.
b) Set 1 to enable output of a start condition in the initial communication state. Otherwise, set 0.

Note The double-boxed items indicate processing that exits the wait state.

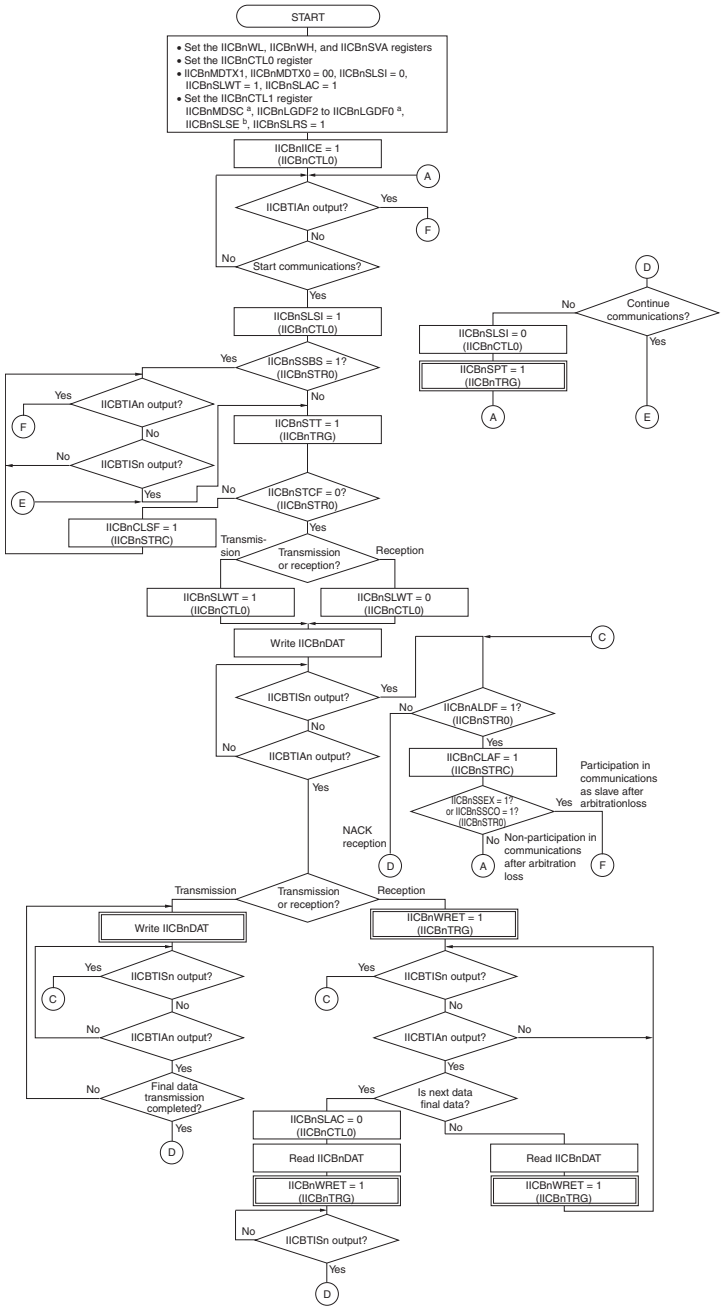
Figure 23-18 Single transfer mode setting sequence when communication reserve function is enabled (IICBnCTL1.IICBnSLRS bit = 0) (multi-master environment) (1/2)



Note The double-boxed items indicate processing that exiting the wait state.

Figure 23-18 Single transfer mode setting sequence when communication reserve function is enabled (IICBnCTL1.IICBnSLRS bit = 0) (multi-master environment) (2/2)

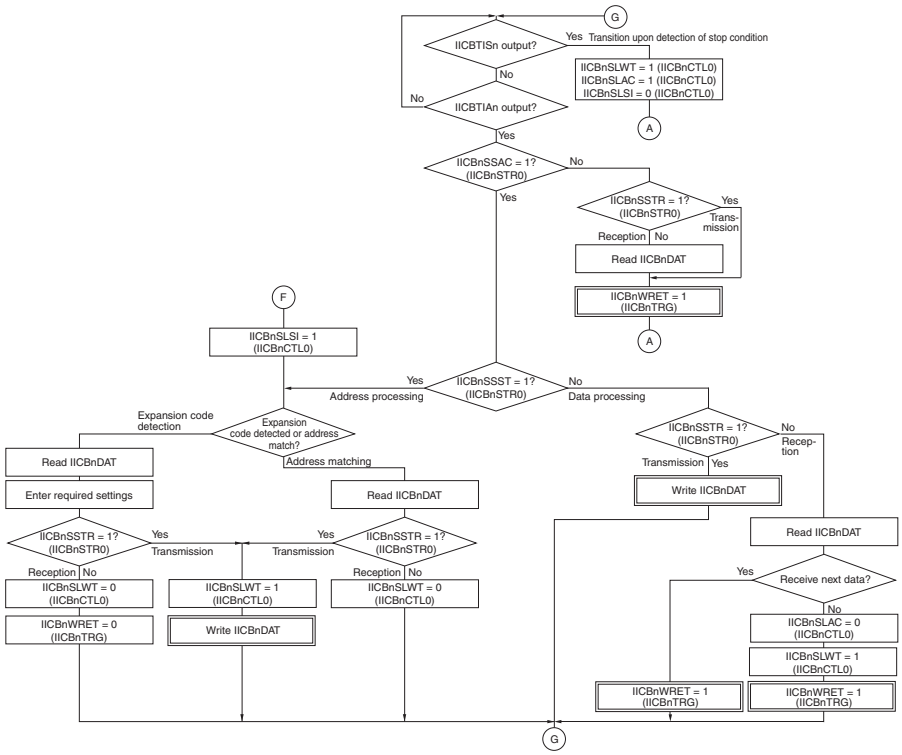
(2) Single transfer mode setting sequence when communication reserve function is disabled (IICBnCTL1.IICBnSLRS bit = 1)



a) Enter settings according to the environment.
b) Set 1 to enable output of a start condition in the initial communication state. Otherwise, set 0.

Note The double-boxed items indicate processing that exits the wait state.

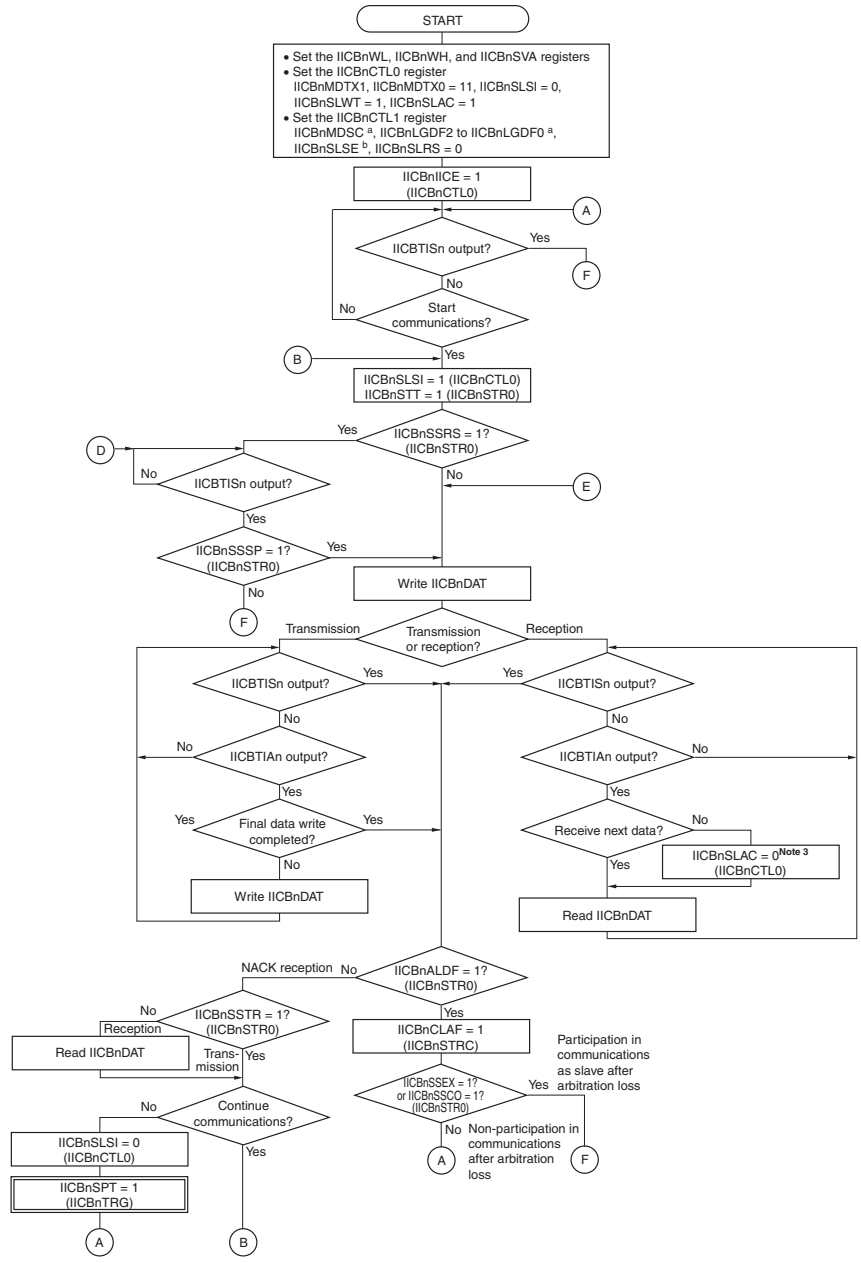
Figure 23-19 Single transfer mode setting sequence when communication reserve function is disabled (IICBnCTL1.IICBnSLRS bit = 1) (multi-master environment) (1/2)



Note The double-boxed items indicate processing that exits the wait state.

Figure 23-19 Single transfer mode setting sequence when communication reserve function is disabled (IICBnCTL1.IICBnSLRS bit = 1) (multi-master environment) (2/2)

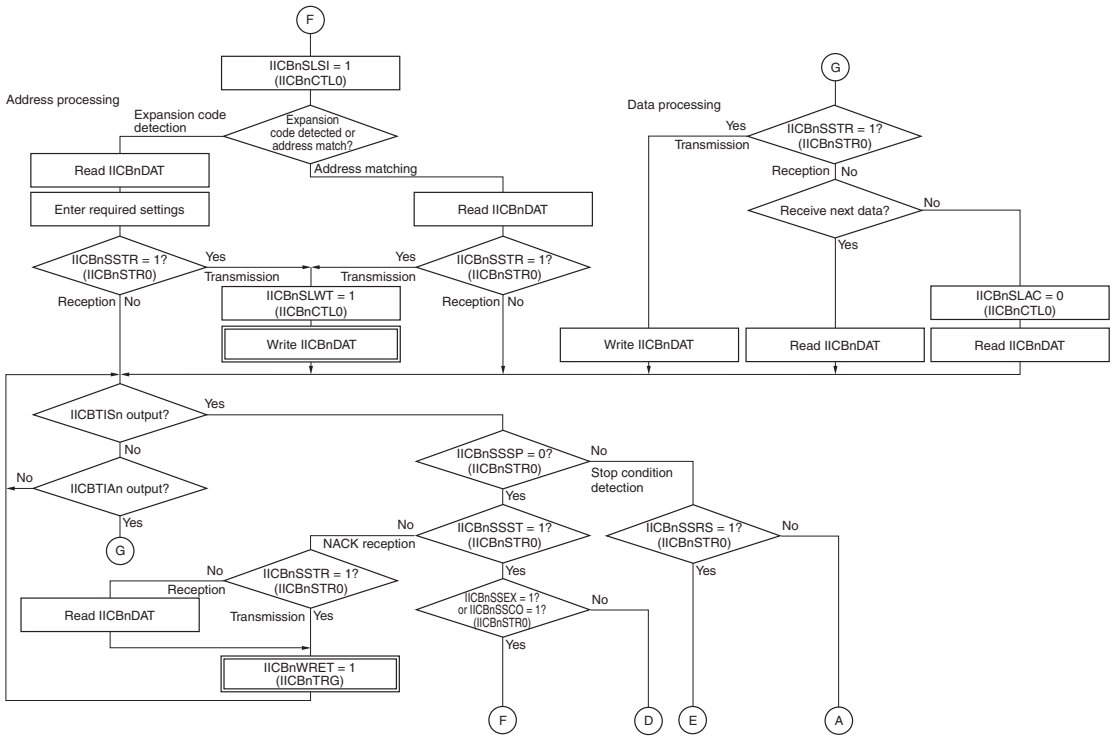
(3) Continuous transfer mode setting sequence when communication reserve function is enabled (IICBnCTL1.IICBnSLRS bit = 0)



a) Enter settings according to the environment.
b) Set 1 to enable output of a start condition in the initial communication state. Otherwise, set 0.
c) The IICBnCTL0.IICBnSLAC bit must be cleared to 0 before the 8th SCLn clock.

Note The double-boxed items indicate processing that exits the wait state.

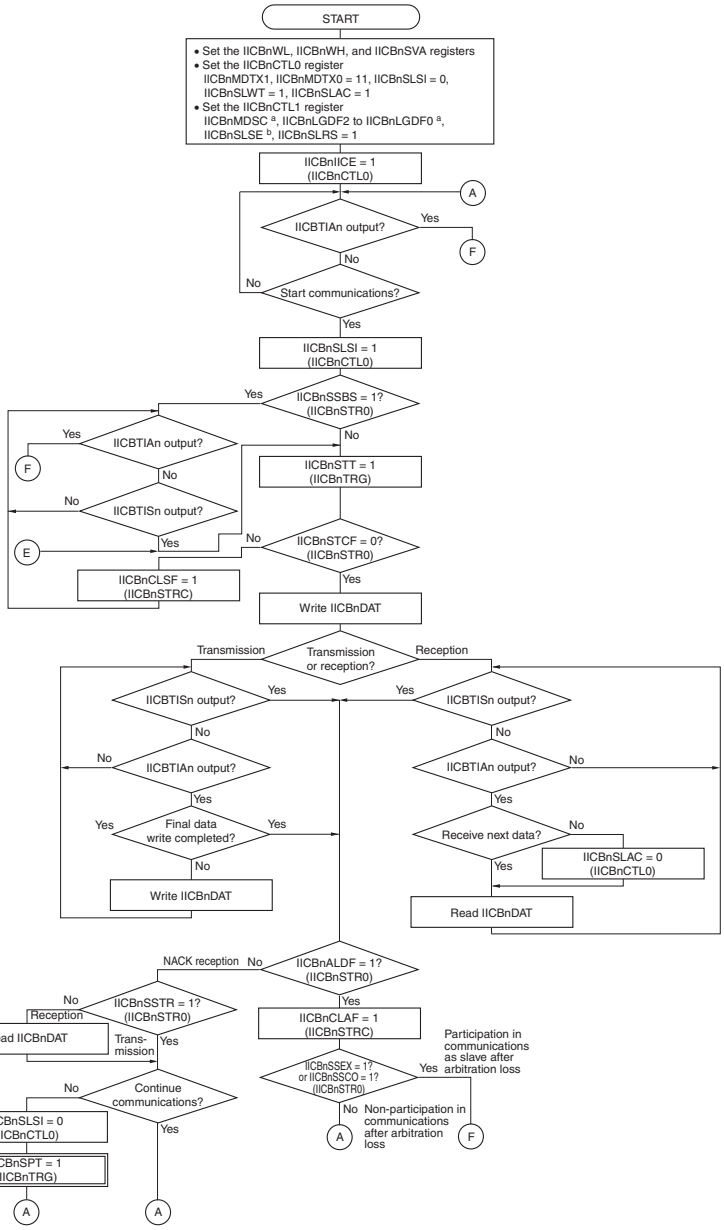
Figure 23-20 Continuous transfer mode setting sequence when communication reserve function is enabled (IICBnCTL1.IICBnSLRS bit = 0) (multi-master environment) (1/2)



Note The double-boxed items indicate processing that exits the wait state.

Figure 23-20 Continuous transfer mode setting sequence when communication reserve function is enabled (IICBnCTL1.IICBnSLRS bit = 0) (multi-master environment) (2/2)

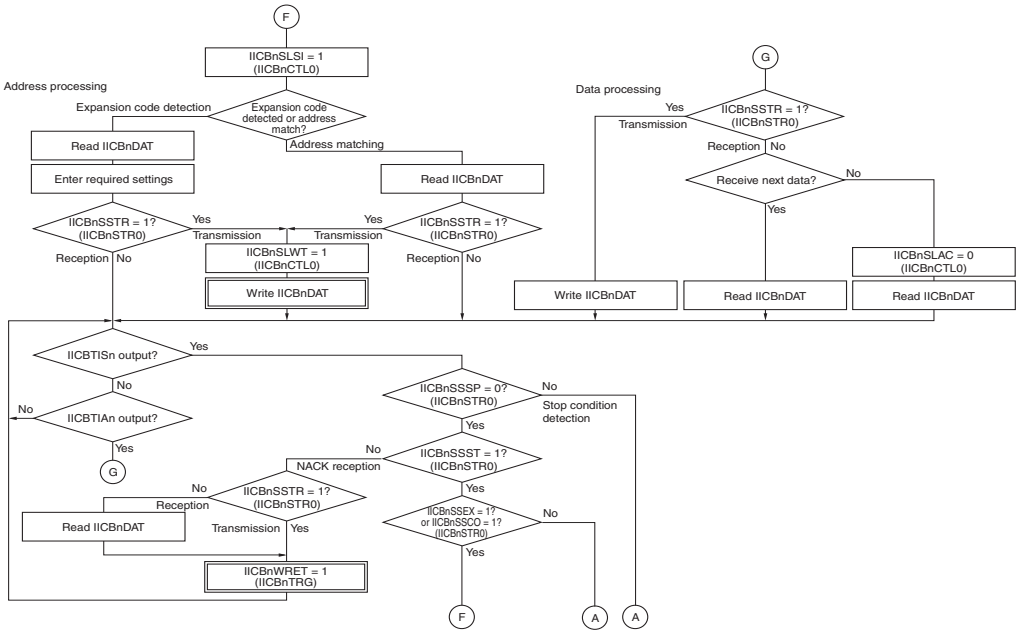
(4) Continuous transfer mode setting sequence when communication reserve function is disabled (IICBnCTL1.IICBnSLRS bit = 1)



a) Enter settings according to the environment.
b) Set 1 to enable output of a start condition in the initial communication state. Otherwise, set 0.

Note The double-boxed items indicate processing that exits the wait state.

Figure 23-21 Continuous transfer mode setting sequence when communication reserve function is disabled (IICBnCTL1.IICBnSLRS bit = 1) (multi-master environment) (1/2)



Note The double-boxed items indicate processing that exits the wait state.

Figure 23-21 Continuous transfer mode setting sequence when communication reserve function is disabled (IICBnCTL1.IICBnSLRS bit = 1) (multi-master environment) (2/2)

Chapter 24 CAN Controller (FCN)

The product features on-chip CAN (Controller Area Network) controllers that comply with the CAN protocol as standardized in ISO 11898.

This chapter contains a generic description of the CAN Controller (FCN).

The first section describes all V850E2/MN4 devices specific properties, such as instances, register base addresses, input/output signal names, etc. The subsequent sections describe the features that apply to all implementations.

24.1 FCN Features of V850E2/MN4

Instances This microcontroller has three instances of the CAN Controller.

Table 24-1 Instances of FCN

CAN Controller	μPD70F3510	μPD70F3512, 70F3514, 70F3515
Instances	None	2
Name	–	FCN0, FCN1

Instances index n Throughout this chapter, the instance of a CAN Controller is identified by the index “n” (n = 0, 1), for example, FCNnGMCLCTL for the FCNn control register.

Table 24-2 Message buffers of FCNn

FCNn instance	Number m of message buffers
FCN0	64
FCN1	64

Caution This products do not include the FCN0DNBMRX2 and the FCN0DNBMRX3 register.

Message buffers index m Throughout this chapter, the FCN message buffer registers are identified by “m” (m = 000 to 063), for example FCNnMmDAT4B for FCN instance n, message data byte 4 of message buffer register m.

Register addresses All CAN Controller register addresses are given as address offsets to the individual base address <FCNn_base>.
Table 24-3 “Register base addresses <FCNn_base>” shows the <FCNn_base> addresses of FCNn.

Table 24-3 Register base addresses <FCNn_base>

FCNn instance	<FCNn_base> address
FCN0	FF48 0000_H
FCN1	FF4A 0000_H

Clock supply All CAN Controllers provide one clock input. The CAN Controllers are connected to P bus clock f_{PCLK}.

Table 24-4 CAN Controller clock supply

FCNn instance	Clock	Connected to:
FCN0, FCN1	CANCLK (f_{CAN})	f_{PCLK}

Interrupts *Table 24-5 "FCNn interrupt requests"* shows the interrupts of the CAN Controllers.

Table 24-5 FCNn interrupt requests

FCNn Interrupt	Function	Connected to
FCN0		
INTC0ERR	FCN0 error detected	• Interrupt controller INTCSIH1IRE
INTC0REC	FCN0 reception completion	• Interrupt controller INTCSIH1IR • DTS controller trigger 111
INTC0TRX	FCN0 transmission completion	• Interrupt controller INTCSIH1IC • DTS controller trigger 112
INTC0WUP	FCN0 sleep wake-up / transmission abortion	• Interrupt controller INTFCNWUP
FCN1		
INTC1ERR	FCN1 error detected	• Interrupt controller INTCSIG5IRE
INTC1REC	FCN1 reception completion	• Interrupt controller INTCSIG5IR
INTC1TRX	FCN1 transmission completion	• Interrupt controller NTCSIG5IC
INTC1WUP	FCN1 sleep wake-up / transmission abortion	• Interrupt controller INTFCNWUP

I/O signals The I/O signals of the CAN Controllers are listed in the table below.

Table 24-6 CAN Controllers I/O signals

FCNn signals	Function	Connected to
FCN0		
CRXD0	FCN0 CAN bus receive input	CAN0RXD
CTXD0	FCN0 CAN bus transmit output	CAN0TXD
FCN1		
CRXD1	FCN1 CAN bus receive input	CAN1RXD
CTXD1	FCN1 CAN bus transmit output	CAN1TXD

24.2 Features

- Compliant with ISO 11898
- Standard frame and extended frame transmission/reception enabled
- Transfer rate: 1 Mbps max. (If FCN clock input is ≥ 16 MHz)
- 64 message buffers per channel
- Receive/transmit history list function (can be set individually for each message buffer)
- Automatic block transmission function
- Multi-buffer receive block function
- Mask setting of 8 patterns is possible for each channel, applicable for data and remote frames
- Data bit time, communication baud rate and sample point can be controlled by FCN module bit-rate prescaler register (FCNnCMBRPRS) and bit rate register (FCNnCMBTCTL)
  - As an example the following sample-point configurations can be configured:
    66.7%, 70.0%, 75.0%, 80.0%, 81.3%, 85.0%, 87.5%
  - Baud rates in the range of 10 kbps up to 1 Mbps can be configured
- Enhanced features:
  - Each message buffer can be configured to operate as a transmit or a receive message buffer
  - A transmission request can be aborted by clearing the Transmit-Request flag of the concerned message buffer. Supported by Transmission Abort Interrupt, on successful abortion.
  - Automatic block transmission operation mode (ABT)
  - Time stamp function for FCN channels 0 to 2 in collaboration with timers capture channels
  - A centralized global data update bit monitor register makes it possible to check all data update bits from one location

24.2.1 Overview of functions

Table 24-7 "Overview of functions" presents an overview of the CAN Controller functions.

Table 24-7 Overview of functions

Function	Details
Protocol	CAN protocol ISO 11898 (standard and extended frame transmission/reception)
Baud rate	Maximum 1 Mbps (minimum FCN clock input = 16 MHz)
Data storage	Storing messages in the FCN RAM
Number of messages	• 64/128 message buffers per channel • Each message buffer can be set to be either a transmit message buffer or a receive message buffer.
Message reception	• Unique ID can be set to each message buffer. • Mask setting of 8 patterns is possible for each channel, applicable for data and remote frames • A receive completion interrupt is generated each time a message is received and stored in a message buffer (receive completion interrupts can be enabled/disabled for each message buffer) • Two or more receive message buffers can be used as a FIFO receive buffer (multi-buffer receive block function). • Receive history list function (can be set individually for each message buffer) • Centralized global data update bit monitor register
Message transmission	• Unique ID can be set to each message buffer. • Receive completion interrupts can be enabled/disabled for each message buffer • Transmit Abort interrupt and Transmit Completion flag for each message buffer (only one transmission of any buffer can be aborted at a time) • Message buffer numbers 0 to 15/31 specified as the transmit message buffers can be used for automatic block transfer. The message transmission interval is programmable (using the automatic block transmission ("ABT") function). • Transmission history list function (can be set individually for each message buffer)
Remote frame processing	• Remote frame processing by transmit message buffer • Remote frame processing by receive message buffer, when applying one of the 8 masks
Time stamp function	• The time stamp function can be set for a message reception when a 32-bit timer is used in combination. • Time stamp capture trigger can be selected (SOF or EOF in a CAN message frame can be detected.).
Diagnostic function	• Readable error counters • "Valid protocol operation flag" for verification of bus connections • Receive-only mode • Single-shot mode • CAN protocol error identification • Self-test mode
Release from bus-off state	• Forced release from bus-off possible by software. • No automatic release from bus-off (software must send recovery request).
Power save mode	• CAN sleep mode (can be woken up by CAN bus) • CAN stop mode (cannot be woken up by CAN bus)

24.2.2 Configuration

The CAN Controller is composed of the following four blocks.

- Internal bus interface
  This functional block provides an internal bus interface and a means of transmitting and receiving messages between the FCN module and the host CPU.
- MCM (Message Control Module)
  This functional block controls access to the CAN protocol layer and to the FCN RAM within the FCN module.
- CAN protocol layer
  This functional block is involved in the operation of the CAN protocol and its related settings.
- CAN RAM
  This is the CAN memory functional block, which is used to store message IDs, message data, etc.

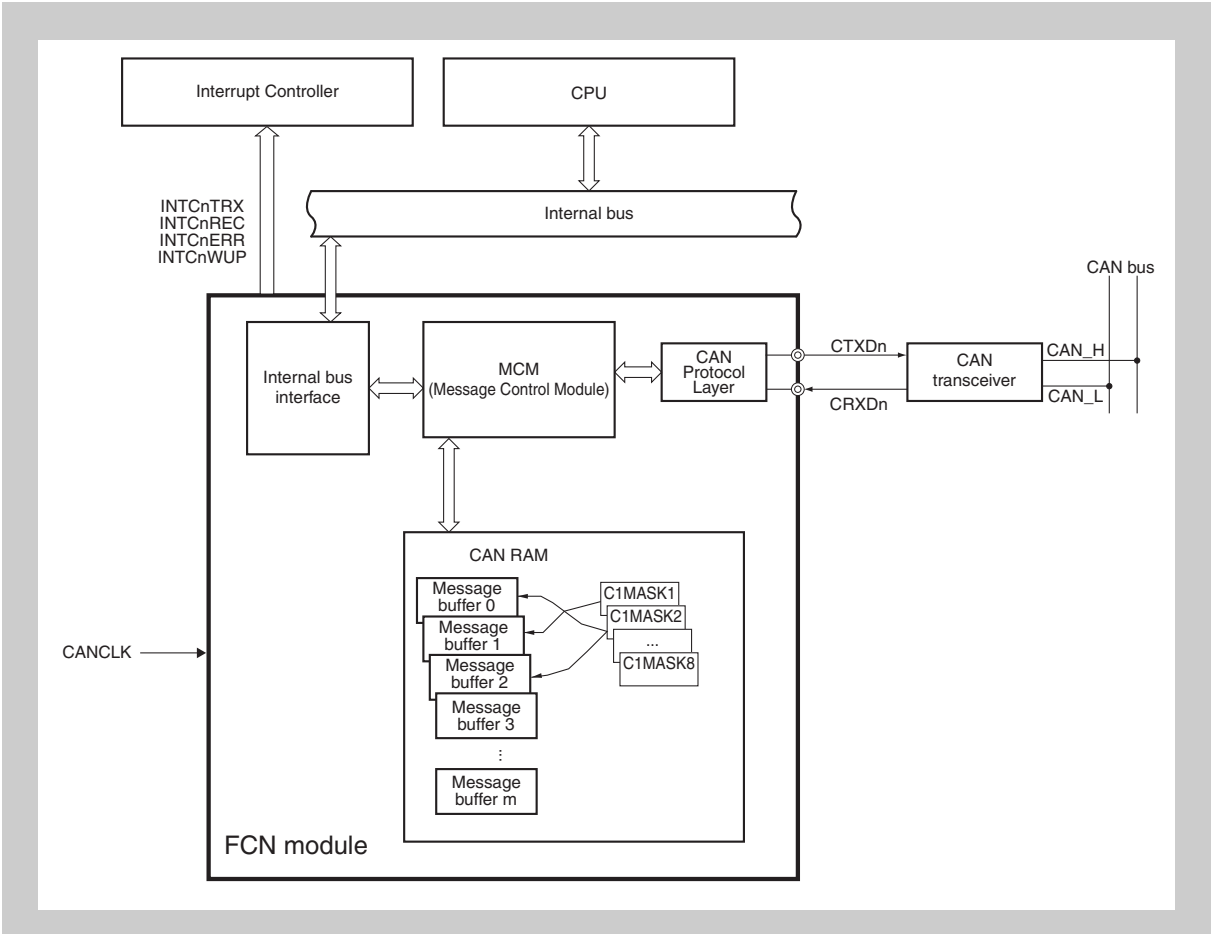


Figure 24-1 Block diagram of the CAN Controller

Caution The CAN RAM includes a memory reference module. If a CAN RAM error occurs during a software reset, the message buffer RAM read error detection bit (FCNnGMCLCTL.FCNnGMCLECCF) will be set. If this bit is set, be sure to check the FCN function for errors.

24.3 Internal Registers of FCN

24.3.1 CAN Controller configuration

Table 24-8 List of FCN registers (1/2)

Item	Register Name
FCNn global registers	FCNn global control register (FCNnGMCLCTL)
	FCNn global clock selection register (FCNnGMCSPRE)
	FCNn global automatic block transmission control register (FCNnGMABCTL)
	FCNn global automatic block transmission delay setting register (FCNnGMADCTL)
	FCNn global Data New bit monitor registers (FCNnDNBMRX0 - FCNnDNBMRX3)
FCNn module registers	FCNn module mask 1 registers (FCNnCMMKCTL01H, FCNnCMMKCTL02H, FCNnCMMKCTL01W)
	FCNn module mask 2 registers (FCNnCMMKCTL03H, FCNnCMMKCTL04H, FCNnCMMKCTL03W)
	FCNn module mask3 registers (FCNnCMMKCTL05H, FCNnCMMKCTL06H, FCNnCMMKCTL05W)
	FCNn module mask 4 registers (FCNnCMMKCTL07H, FCNnCMMKCTL08H, FCNnCMMKCTL07W)
	FCNn module mask 5 registers (FCNnCMMKCTL09H, FCNnCMMKCTL10H, FCNnCMMKCTL09W)
	FCNn module mask 6 registers (FCNnCMMKCTL11H, FCNnCMMKCTL12H, FCNnCMMKCTL11W)
	FCNn module mask 7 registers (FCNnCMMKCTL13H, FCNnCMMKCTL14H, FCNnCMMKCTL13W)
	FCNn module mask 8 registers (FCNnCMMKCTL15H, FCNnCMMKCTL16H, FCNnCMMKCTL15W)
	FCNn module control register (FCNnCMCLCTL)
	FCNn module last error information register (FCNnCMLCSTR)
	FCNn module information register (FCNnCMINCSTR)
	FCNn module error counter register (FCNnCMERCNT)
	FCNn module interrupt enable register (FCNnCMIECTL)
	FCNn module interrupt status register (FCNnCMISCTL)
	FCNn module bit rate prescaler and FCN clock selector register (FCNnCMBRPRS)
	FCNn module bit rate register (FCNnCMBTCTL)
	FCNn module last in-pointer register (FCNnCMLISTR)
	FCNn module receive history list register (FCNnCMRGRX)
	FCNn module last out-pointer register (FCNnCMLOSTR)
	FCNn module transmit history list register (FCNnCMTGTX)
	FCNn module time stamp register (FCNnCMTSCTL)

Table 24-8 List of FCN registers (2/2)

Item	Register Name
FCNn message buffer registers	FCNn message data byte 0 to 3 registers m (FCNnMmDAT0W, FCNnMmDAT0H, FCNnMmDAT2H, FCNnMmDAT0B, FCNnMmDAT1B, FCNnMmDAT2B, FCNnMmDAT3B)
	FCNn message data byte 4 to 7 registers m (FCNnMmDAT4W, FCNnMmDAT4H, FCNnMmDAT6H, FCNnMmDAT4B, FCNnMmDAT5B, FCNnMmDAT6B, FCNnMmDAT7B)
	FCNn message data length register m (FCNnMmDTLGB)
	FCNn message configuration register m (FCNnMmSTRB)
	FCNn message ID registers m (FCNnMmMID0H, FCNnMmMID1H, FCNnMmMID0W)
	FCNn message control register m (FCNnMmCTL)

24.3.2 CAN Controller Registers Overview

Address offset All register addresses are given as offsets to the base address <FCNn_base>. The <FCNn_base> addresses of the registers are defined in the first section of this chapter under the keyword “Register addresses”.

(1) FCNn global and module registers

Table 24-9 FCNn global and module registers (1/2)

Address offset	Register name	Symbol	R/W	Access bit	After reset
0 0008$_H$	FCNn global clock selection register	FCNnGMCSPRE	R/W	8	0F$_H$
0 0020$_H$	FCNn global automatic block transmission delay setting register	FCNnGMADCTL	R/W	8	00$_H$
0 8000$_H$	FCNn global control register	FCNnGMCLCTL	R/W	16	00x0$_H$[a]
0 8018$_H$	FCNn global automatic block transmission control register	FCNnGMABCTL	R/W	16	0000$_H$
1 00C0$_H$	FCNn global data update bit monitor register 0	FCNnDNBMRX0	R	32	[b]
1 00D0$_H$	FCNn global data update bit monitor register 1	FCNnDNBMRX1	R	32	[b]
1 00E0$_H$	FCNn global data update bit monitor register 2	FCNnDNBMRX2	R	32	[b]
1 00F0$_H$	FCNn global data update bit monitor register 3	FCNnDNBMRX3	R	32	[b]
0 8300$_H$	FCNn module mask 1 register	FCNnCMMKCTL01H	R/W	16	[b]
0 8308$_H$		FCNnCMMKCTL02H			
1 0300$_H$		FCNnCMMKCTL01W		32	
0 8310$_H$	FCNn module mask 2 register	FCNnCMMKCTL03H	R/W	16	[b]
0 8318$_H$		FCNnCMMKCTL04H			
1 0310$_H$		FCNnCMMKCTL03W		32	
0 8320$_H$	FCNn module mask 3 register	FCNnCMMKCTL05H	R/W	16	[b]
0 8328$_H$		FCNnCMMKCTL06H			
1 0320$_H$		FCNnCMMKCTL05W		32	
0 8330$_H$	FCNn module mask 4 register	FCNnCMMKCTL07H	R/W	16	[b]
0 8338$_H$		FCNnCMMKCTL08H			
1 0330$_H$		FCNnCMMKCTL07W		32	
0 8340$_H$	FCNn module mask 5 register	FCNnCMMKCTL09H	R/W	16	[b]
0 8348$_H$		FCNnCMMKCTL10H			
1 0340$_H$		FCNnCMMKCTL09W		32	
0 8350$_H$	FCNn module mask 6 register	FCNnCMMKCTL11H	R/W	16	[b]
0 8358$_H$		FCNnCMMKCTL12H			
1 0350$_H$		FCNnCMMKCTL11W		32	
0 8360$_H$	FCNn module mask 7 register	FCNnCMMKCTL13H	R/W	16	[b]
0 8368$_H$		FCNnCMMKCTL14H			
1 0360$_H$		FCNnCMMKCTL13W		32	
0 8370$_H$	FCNn module mask 8 register	FCNnCMMKCTL15H	R/W	16	[b]
0 8378$_H$		FCNnCMMKCTL16H			
1 0370$_H$		FCNnCMMKCTL15W		32	
0 0248$_H$	FCNn module last error information register	FCNnCMLCSTR	R/W	8	00$_H$

Table 24-9 FCNn global and module registers (2/2)

Address offset	Register name	Symbol	R/W	Access bit	After reset
0 024C$_H$	FCNn module information register	FCNnCMINSTR	R	8	00$_H$
0 0268$_H$	FCNn module bit-rate prescaler and clock selector register	FCNnCMBRPRS	R/W	8	FF$_H$
0 0278$_H$	FCNn module last receive pointer register	FCNnCMLISTR	R	8	Undefined
0 0288$_H$	FCNn module last transmit pointer register	FCNnCMLOSTR	R	8	Undefined
0 8240$_H$	FCNn module control register	FCNnCMCLCTL	R/W	16	0000$_H$
0 8250$_H$	FCNn module error counter register	FCNnCMERCNT	R	16	0000$_H$
0 8258$_H$	FCNn module interrupt enable register	FCNnCMIECTL	R/W	16	0000$_H$
0 8260$_H$	FCNn module interrupt status register	FCNnCMISCTL	R/W	16	0000$_H$
0 8270$_H$	FCNn module bit-rate register	FCNnCMBTCTL	R/W	16	370F$_H$
0 8280$_H$	FCNn module receive history list register	FCNnCMRGRX	R/W	16	xx02$_H$
0 8290$_H$	FCNn module transmit history list register	FCNnCMTGTX	R/W	16	xx02$_H$
0 8298$_H$	FCNn module time stamp register	FCNnCMTSCTL	R/W	16	0000$_H$

a) Initial value depends on FCNnGMCLCTL.FCNnGMCLECCF, which indicates error detections when reading from message buffer RAM. Refer to the detailed description of the FCNnGMCLCTL register.

b) After resetting, the value will be 0000$_H$ or 00000000$_H$.

24.3.3 Register bit configuration

Table 24-10 FCN global register bit configuration

Address offset	Symbol	Bit 7/ 15/31/ 23	Bit 6/ 14/30/ 22	Bit 5/ 13/29/ 21	Bit 4/ 12/28/ 20	Bit 3/ 11/27/ 19	Bit 2/ 10/26/ 18	Bit 1/9/ 25/17	Bit 0/8/ 24/16
0 8000$_H$	FCNnGMCLCTL (W)	0	0	FCNnGMCLCLMB		0	0	0	FCNnGMCLCLOM
		0	0	0	FCNnGMCLSESR	0	0	FCNnGMCLSEDE	FCNnGMCLSEOM
	FCNnGMCLCTL (R)	0	0	FCNnGMCLECCF	FCNnGMCLSORF	0	0	FCNnGMCLESDE	FCNnGMCLPWOM
		FCNnGMCLSSMO	0	0	0	0	0	0	0
0 0008$_H$	FCNnGMCSPRE	0	0	0	0	FCNnGMCSPRSC[3:0]			
0 8018$_H$	FCNnGMABCTL (W)	0	0	0	0	0	0	0	FCNnGMABCLAT
		0	0	0	0	0	0	FCNnGMABSEAC	FCNnGMABSEAT
	FCNnGMABCTL (R)	0	0	0	0	0	0	FCNnGMABCLRF	FCNnGMABABTT
		0	0	0	0	0	0	0	0
0 0020$_H$	FCNnGMADCTL	0	0	0	0	FCNnGMADSSAD[3:0]			
1 00C0$_H$	FCNnDNBMRX0 (R)	FCNnDNBMSSDN[7:0]							
		FCNnDNBMSSDN[15:8]							
		FCNnDNBMSSDN[23:16]							
		FCNnDNBMSSDN[31:24]							
1 00D0$_H$	FCNnDNBMRX1 (R)	FCNnDNBMSSDN[39:32]							
		FCNnDNBMSSDN[47:40]							
		FCNnDNBMSSDN[55:48]							
		FCNnDNBMSSDN[63:56]							
1 00E0$_H$	FCNnDNBMRX2 (R)[a]	FCNnDNBMSSDN[71:64]							
		FCNnDNBMSSDN[79:72]							
		FCNnDNBMSSDN[87:80]							
		FCNnDNBMSSDN[95:88]							
1 00F0$_H$	FCNnDNBMRX3 (R)[a]	FCNnDNBMSSDN[103:96]							
		FCNnDNBMSSDN[111:104]							
		FCNnDNBMSSDN[119:112]							
		FCNnDNBMSSDN[127:120]							

a) Only available with 128 message buffers (m = 0 to 127)

Table 24-11 FCN module mask control 16-bit registers bit configuration

Address offset	Symbol	Bit 15	Bit 14	Bit 13	Bit 12 to 0
0 8300$_H$	FCNnCMMKCTL01H	FCNnCMMKSSID[15:0]			
0 8308$_H$	FCNnCMMKCTL02H	0	0	0	FCNnCMMKSSID[28:16]
0 8310$_H$	FCNnCMMKCTL03H	FCNnCMMKSSID[15:0]			
0 8318$_H$	FCNnCMMKCTL04H	0	0	0	FCNnCMMKSSID[28:16]
0 8320$_H$	FCNnCMMKCTL05H	FCNnCMMKSSID[15:0]			
0 8328$_H$	FCNnCMMKCTL06H	0	0	0	FCNnCMMKSSID[28:16]
0 8330$_H$	FCNnCMMKCTL07H	FCNnCMMKSSID[15:0]			
0 8338$_H$	FCNnCMMKCTL08H	0	0	0	FCNnCMMKSSID[28:16]
0 8340$_H$	FCNnCMMKCTL09H	FCNnCMMKSSID[15:0]			
0 8348$_H$	FCNnCMMKCTL10H	0	0	0	FCNnCMMKSSID[28:16]
0 8350$_H$	FCNnCMMKCTL11H	FCNnCMMKSSID[15:0]			
0 8358$_H$	FCNnCMMKCTL12H	0	0	0	FCNnCMMKSSID[28:16]
0 8360$_H$	FCNnCMMKCTL13H	FCNnCMMKSSID[15:0]			
0 8368$_H$	FCNnCMMKCTL14H	0	0	0	FCNnCMMKSSID[28:16]
0 8370$_H$	FCNnCMMKCTL15H	FCNnCMMKSSID[15:0]			
0 8378$_H$	FCNnCMMKCTL16H	0	0	0	FCNnCMMKSSID[28:16]

Table 24-12 FCN module mask control 32-bit registers bit configuration

Address offset	Symbol	Bit 31	Bit 30	Bit 29	Bit 28 to 0
$1\ 0300_H$	FCNnCMMKCTL01W	0	0	0	FCNnCMMKSSID[28:0]
$1\ 0310_H$	FCNnCMMKCTL03W	0	0	0	FCNnCMMKSSID[28:0]
$1\ 0320_H$	FCNnCMMKCTL05W	0	0	0	FCNnCMMKSSID[28:0]
$1\ 0330_H$	FCNnCMMKCTL07W	0	0	0	FCNnCMMKSSID[28:0]
$1\ 0340_H$	FCNnCMMKCTL09W	0	0	0	FCNnCMMKSSID[28:0]
$1\ 0350_H$	FCNnCMMKCTL11W	0	0	0	FCNnCMMKSSID[28:0]
$1\ 0360_H$	FCNnCMMKCTL13W	0	0	0	FCNnCMMKSSID[28:0]
$1\ 0370_H$	FCNnCMMKCTL15W	0	0	0	FCNnCMMKSSID[28:0]

Table 24-13 FCN module register bit configuration (1/2)

Address offset	Symbol	Bit 7/15	Bit 6/14	Bit 5/13	Bit 4/12	Bit 3/11	Bit 2/10	Bit 1/9	Bit 0/8
0 8240$_H$	FCNnCM CLCTL (W)	0	FCNnCM CLCLAL	FCNnCM CLCLVL	FCNnCMCLCLPS[1:0]		FCNnCMCLCLOP[2:0]		
		FCNnCM CLSERC	FCNnCM CLCSEAL	0	FCNnCMCLSEPS[1:0]		FCNnCMCSELOP[2:0]		
	FCNnCM CLCTL (R)	FCNnCM CLERCF	FCNnCM CLALBF	FCNnCM CLVALF	FCNnCMCLMDPF[1:0]		FCNnCMCLMDOF[2:0]		
		0	0	0	0	0	0	FCNnCMC LSSRS	FCNnCMC LSSTS
0 00248$_H$	FCNnCM LCSTR (W)	0	0	0	0	0	0	0	0
	FCNnCM LCSTR (R)	0	0	0	0	0	FCNnCMLCSSLC[2:0]		
0 024CH	FCNnCM INSTR	0	0	0	FCNnCMI NBOFF	FCNnCMINSSTE[1:0]		FCNnCMINSSRE[1:0]	
0 8250$_H$	FCNnCM ERCNT	FCNnCMERTECF[7:0]							
		FCNnCM ERRPSF	FCNnCMERRECF[6:0]						
0 8258$_H$	FCNnCM IECTL (W)	0	FCNnCMIECLIE[6:0]						
		0	FCNnCMIESEIE[6:0]						
	FCNnCM IECTL (R)	0	FCNnCMIEINTF[6:0]						
		0	0	0	0	0	0	0	0
0 8260$_H$	FCNnCM ISCTL (W)	0	FCNnCMISCLTS[6:0]						
		0	0	0	0	0	0	0	0
	FCNnCM ISCTL (R)	0	FCNnCMISITSF[6:0]						
		0	0	0	0	0	0	0	0
0 0268$_H$	FCNnCM BRPRS	FCNnCMBRPRS[7:0]							
0 8270$_H$	FCNnCM BTCTL	0	0	0	0	FCNnCMBTS1LG[3:0]			
		0	0	FCNnCMBTJWLG[1:0]		0	FCNnCMBTS2LG[2:0]		
0 0278$_H$	FCNnCM LISTR	FCNnCMLISSLR[7:0]							
0 8280$_H$	FCNnCM RGRX (W)	0	0	0	0	0	0	0	FCNnCMR GCLRV
		0	0	0	0	0	0	0	0
	FCNnCM RGRX (R)	0	0	0	0	0	0	FCNnCMR GSSPM	FCNnCMR GRVFF
		FCNnCMRDSSPT[7:0]							
0 0288$_H$	FCNnCM LOSTR	FCNnCMLOSSLT[7:0]							
0 8290$_H$	FCNnCM TGTX (W)	0	0	0	0	0	0	0	FCNnCMT GCLTV
		0	0	0	0	0	0	0	0
	FCNnCM TGTX (R)	0	0	0	0	0	0	FCNnCMT GSSPM	FCNnCMT GTVFF
		FCNnCMTGSSPT[7:0]							

Table 24-13 FCN module register bit configuration (2/2)

Address offset	Symbol	Bit 7/15	Bit 6/14	Bit 5/13	Bit 4/12	Bit 3/11	Bit 2/10	Bit 1/9	Bit 0/8
0 8298$_H$	FCNnCM TSCTL (W)	0	0	0	0	0	FCNnCMT SCLLK	FCNnCMT SCLSL	FCNnCMT SCLTS
		0	0	0	0	0	FCNnCMT SSELK	FCNnCMT SSESL	FCNnCMT SSETS
	FCNnCM TSCTL (R)	0	0	0	0	0	FCNnCMT SLOKE	FCNnCMT SSELE	FCNnCMT STSGE
		0	0	0	0	0	0	0	0

Table 24-14 FCN message buffer register bit configuration (1/2)

Address offset	Symbol	Bit 7/15/ 31/23	Bit 6/14/ 30/22	Bit 5/13/ 29/21	Bit 4/12/ 28/20	Bit 3/11/ 27/19	Bit 2/10/ 26/18	Bit 1/9/ 25/17	Bit 0/8/ 24/16
$1\ 1000_H$ + m x 40_H	FCNnMm DAT0W	FCNnMmSSD[07:00]							
		FCNnMmSSD[17:10]							
		FCNnMmSSD[27:00]							
		FCNnMmSSD[37:30]							
$0\ 9000_H$ + m x 40_H	FCNnMm DAT0H	FCNnMmSSD[07:00]							
		FCNnMmSSD[17:10]							
$0\ 1000_H$ + m x 40_H	FCNnMm DAT0B	FCNnMmSSD[07:00]							
$0\ 1004_H$ + m x 40_H	FCNnMm DAT1B	FCNnMmSSD[17:10]							
$0\ 9008_H$ + m x 40_H	FCNnMm DAT2H	FCNnMmSSD[27:20]							
		FCNnMmSSD[37:30]							
$0\ 1008_H$ + m x 40_H	FCNnMm DAT2B	FCNnMmSSD[27:20]							
$0\ 100C_H$ + m x 40_H	FCNnMm DAT3B	FCNnMmSSD[37:30]							
$1\ 1010_H$ + m x 40_H	FCNnMm DAT4W	FCNnMmSSD[47:40]							
		FCNnMmSSD[57:50]							
		FCNnMmSSD[67:60]							
		FCNnMmSSD[77:70]							
$0\ 9010_H$ + m x 40_H	FCNnMm DAT4H	FCNnMmSSD[47:40]							
		FCNnMmSSD[57:50]							
$0\ 1010_H$ + m x 40_H	FCNnMm DAT4B	FCNnMmSSD[47:40]							
$0\ 1014_H$ + m x 40_H	FCNnMm DAT5B	FCNnMmSSD[57:50]							
$0\ 9018_H$ + m x 40_H	FCNnMm DAT6H	FCNnMmSSD[67:60]							
		FCNnMmSSD[77:70]							
$0\ 1018_H$ + m x 40_H	FCNnMm DAT6B	FCNnMmSSD[67:60]							
$0\ 101C_H$ + m x 40_H	FCNnMm DAT7B	FCNnMmSSD[77:70]							
$0\ 1020_H$ + m x 40_H	FCNnMm DTLGB	0				FCNnMmDTLG[3:0]			
$0\ 1024_H$ + m x 40_H	FCNnMm STRB	FCNnMm SSOW	FCNnMmSSMT[3:0]				FCNnMm SSRT	0	FCNnMm SSAM
$0\ 9028_H$ + m x 40_H	FCNnMm MID0H	FCNnMmSSID[7:0]							
		FCNnMmSSID[15:8]							
$0\ 9030_H$ + m x 40_H	FCNnMm MID1H	FCNnMmSSID[23:16]							
		FCNnMm SSIE	0	0	FCNnMmSSID[28:24]				

Table 24-14 FCN message buffer register bit configuration (2/2)

Address offset	Symbol	Bit 7/15/31/23	Bit 6/14/30/22	Bit 5/13/29/21	Bit 4/12/28/20	Bit 3/11/27/19	Bit 2/10/26/18	Bit 1/9/25/17	Bit 0/8/24/16
$1\ 1028_H$ + m x 40_H	FCNnMm MID0W	FCNnMmSSID[7:0]							
		FCNnMmSSID[15:8]							
		FCNnMmSSID[23:16]							
		FCNnMm SSIE	0	0	FCNnMmSSID[28:24]				
$0\ 9038_H$ + m x 40_H	FCNnMmCTL (W)	0	FCNnMm CLNH	0	FCNnMm CLMW	FCNnMm CLIE	FCNnMm CLDN	FCNnMm CLTR	FCNnMm CLRY
		0	FCNnMm SENH	0	0	FCNnMm SEIE	0	FCNnMm SETR	FCNnMm SERY
	FCNnMmCTL (R)	0	FCNnMm NHMF	0	FCNnMm MOWF	FCNnMm IENF	FCNnMm DTNF	FCNnMm TRQF	FCNnMm RDYF
		0	0	FCNnMm MUCF	0	0	0	FCNnMm TCPF	0

24.4 Bit Set/Clear Function

The FCN control registers include registers whose bits can be set or cleared via the CPU and via the CAN Controller. These register bits can not be changed directly by the CPU by any bit manipulation instructions, such as SET1, CLR1, and NOT1. Instead a special bit-set/bit-clear mechanism is used.

All registers where bit manlpulation operations are prohibited are organised in such a way that all bits allowed for changing by the CPU are located in the lower byte (RWx in the register layout below), while in the upper byte either no or read-only information is located (ROx in the register layout below).

The registers can be read in the usual way getting all 16 data bits in their current setting and as described in the register description.
For setting or clearing any of the lower 8 bits the following mechanism is implemented:

When writing 16-bit data to the register address

Bit clear

- each of the lower 8 data bits (CLx in the register layout below) indicates whether the corresponding register bit RWx should be
  - cleared, i.e. set to 0: if CLx = 1, the corresponding RWx is cleared to 0
  - remain unchanged: if CLx = 0, the corresponding RWx does not change

Bit set

- each of the upper 8 data bits (SEx in the register layout below) indicate whether the corresponding register bit should be
  - set, i.e. set to 1: if SEx = 1, the corresponding RWx is set to 1
  - remain unchanged: if SEx = 0, the corresponding RWx does not change

Register layout for read access:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RO7	RO6	RO5	RO4	RO3	RO2	RO1	RO0	RW7	RW6	RW5	RW4	RW3	RW2	RW1	RW0
changing by the CPU not possible								bits for CPU manipulation via SE7-SE0 and CL7-CL0							

Register layout for write access:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SE7	SE6	SE5	SE4	SE3	SE2	SE1	SE0	CL7	CL6	CL5	CL4	CL3	CL2	CL1	CL0
SEx = 1 sets the corresponding RW7-RW0								CLx = 1 clears the corresponding RW7-RW0							

The following table denotes the operations applied to the RWx bits:

Table 24-15 Bit set/clear operation

CLx	SEx	Operation on RWx
0	0	Not changed
0	1	Set
1	0	Cleared
1	1	Not changed

Example The following shows an example.

Changing the register with the content 1883_H as follows:

- Bit 3 shall be set: SE3 = 1
- Bit 1 shall be cleared: CL1 = 1

Register read before bit manipulations:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	1	1	0	0	0	1	0	0	0	**0**	0	**1**	1
may hold any value, here 18_H								RW7-RW0: 83_H							

Register write access:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	**1**	0	0	0	0	0	0	0	0	0	**1**	0
SE3 = 1: 08_H								CL1 = 1: 02_H							

Register read after bit manipulations:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	1	1	0	0	0	1	0	0	0	**1**	0	**0**	1
may hold any value, here 18_H								RW7-RW0: 89_H							

24.5 Control Registers

24.5.1 FCN global registers

(1) FCNnGMCLCTL - FCNn global control register

This register is used to control the operation of the FCN module.

Access This register can be read/written in 16-bit units.

Address <FCNn_base> + $0\ 8000_H$

Initial Value $00x0_H$.[a] The register is initialized by any reset.

a) Soft reset starts automatically after hard reset.
So initial value is:
--- If error is not detected after soft reset, then it is 0000_H.
--- If error is not detected on soft reset, then it is 0010_H.
--- If error is detected after soft reset, then it is 0020_H.
--- If error is detected on soft reset, then it is 0030_H.

(a) FCNnGMCLCTL read

15	14	13	12	11	10	9	8
FCNnGMCLSSMO	0	0	0	0	0	0	0
7	6	5	4	3	2	1	0
0	0	FCNnGMCLECCF	FCNnGMCLSORF	0	0	FCNnGMCLESDE	FCNnGMCLPWOM

FCNnGMCLSSMO	Bit enabling access to message buffer register, transmit/receive history registers
0	Write access and read access to the message buffer register and the transmit/receive history list registers is disabled.
1	Write access and read access to the message buffer register and the transmit/receive history list registers is enabled.

Cautions

1. While the FCNnGMCLCTL.FCNnGMCLSSMO is cleared (to 0), software access to FCN message buffer registers (i.e. all FCNnMm registers) and registers related to transmit history or receive history (FCNnCMLOSTR, FCNnCMTGTX, FCNnCMLISTR, and FCNnCMRGRX) is disabled.
2. FCNnGMCLCTL.FCNnGMCLSSMO is read-only. Even if 1 is written while it is 0, its value does not change, and access to the message buffer registers, or registers related to transmit history or receive history remains disabled.

Note FCNnGMCLCTL.FCNnGMCLSSMO is cleared to (0) when the FCN module enters FCN sleep mode/FCN stop mode, or when the FCNnGMCLCTL.FCNnGMCLPWOM is cleared to (0).
FCNnGMCLSSMO is set to (1) when the FCN sleep mode/FCN stop mode is released, or when the FCNnGMCLCTL.FCNnGMCLPWOM is set to (1).

FCNnGMCLECCF	Message buffer RAM read error detect bit
0	Not detect error for reading from message buffer RAM.
1	Detect error for reading from message buffer RAM.

Notes

1. FCNnGMCLCTL.FCNnGMCLECCF is set (1) in case of detecting a memory error when reading from the message buffer RAM during the soft reset process. Once FCNnGMCLECCF is set (1), it keeps the level until it is cleared (0).
2. Only use this bit to check for memory errors after executing a soft reset.
3. It is impossible to clear FCNnGMCLECCF (0) during FCNnGMCLCTL.FCNnGMCLSORF is set (1) (soft reset is ongoing).

FCNnGMCLSORF	Soft reset execution status bit
0	No soft reset
1	Soft reset is ongoing

Notes

1. While a soft reset is ongoing (FCNnGMCLCTL.FCNnGMCLSORF is set (1)), it is impossible to set FCNnGMCLCTL.FCNnGMCLPWOM and FCNnGMCLCTL.EFSD.
   It is possible to set start a software reset by FCNnGMCLCTL.FCNnGMCLSESR = 1 during FCNnGMCLCTL.FCNnGMCLPWOM bit is clear (0).
2. When FCNnGMCLCTL.FCNnGMCLSORF is set (1), the initialization of message buffer RAM starts. It is possible to detect error during initializing message buffer RAM, if FCNnGMCLCTL.FCNnGMCLECCF is cleared before setting FCNnGMCLSORF.
3. When FCNnGMCLCTL.FCNnGMCLSORF is set (1) again in the condition that is is already set (1), the soft reset procedure does not restart, but continues.
4. After releasing hardware reset FCNnGMCLCTL.FCNnGMCLSORF is set (1) automatically and initialization of message buffer RAM starts.
5. It is impossible that clearing FCNnGMCLCTL.FCNnGMCLPWOM (0) and setting FCNnGMCLCTL.FCNnGMCLSORF (1) are done at the same time.
6. If a hardware RESET occurs during FCNnGMCLCTL.FCNnGMCLSORF = 1, then the soft reset procedure is stopped (aborted), and the hardware RESET starts.

FCNnGMCLESDE	Bit enabling forced shut down
0	Forced shutdown by setting FCNnGMCLCTL.FCNnGMCLPWOM = 0 is disabled.
1	Forced shutdown by setting FCNnGMCLCTL.FCNnGMCLPWOM = 0 is enabled.

Caution To request a forced shut down, FCNnGMCLCTL.FCNnGMCLPWOM must be cleared to 0 in a subsequent, immediately following access after FCNnGMCLCTL.FCNnGMCLESDE has been set to 1. If any access to another register (including reading the FCNnGMCLCTL register) is executed without clearing FCNnGMCLPWOM immediately after FCNnGMCLESDE has been set to 1, FCNnGMCLESDE is forcibly cleared to 0, and the forced shut down request is invalid.

FCNnGMCLPWOM	Global operation mode bit
0	FCN module is disabled.
1	FCN module is enabled to operate.

Caution FCNnGMCLCTL.FCNnGMCLPWOM can be cleared only in the initialization mode or immediately after FCNnGMCLCTL.FCNnGMCLESDE is set (forced shutdown).

(b) FCNnGMCLCTL write

15	14	13	12	11	10	9	8
0	0	0	FCNnGMCLSESR	0	0	FCNnGMCLSESD	FCNnGMCLSEOM
7	6	5	4	3	2	1	0
0	0	FCNnGMCLCLMB	0	0	0	0	FCNnGMCLCLOM

FCNnGMCLSESR	Software reset start
0	No changes.
1	Start soft reset.

FCNnGMCLSESD	FCNnGMCLESDE bit setting
0	No change in FCNnGMCLESDE bit.
1	FCNnGMCLESDE bit set to 1.

FCNnGMCLSEOM	FCNnGMCLCLOM	FCNnGMCLPWOM bit setting
0	1	FCNnGMCLCTL.FCNnGMCLPWOM bit cleared to 0.
1	0	FCNnGMCLCTL.FCNnGMCLPWOM bit set to 1.
Other than above		No change of FCNnGMCLCTL.FCNnGMCLPWOM bit.

Caution Set FCNnGMCLCTL.FCNnGMCLPWOM and FCNnGMCLCTL.FCNnGMCLESDE bit always separately.

FCNnGMCLCLMB	FCNnGMCLCTL.FCNnGMCLECCF bit clear
0	No change in FCNnGMCLCTL.FCNnGMCLECCF bit.
1	FCNnGMCLCTL.FCNnGMCLECCF bit cleared to 0.

(2) FCNnGMCSPRE - FCNn global clock selection register

This register is used to select the FCN module system clock.

Access This register can be read/written in 8-bit units.

Address <FCNn_base> + 0008_H

Initial Value $0F_H$. The register is initialized by any reset.

7	6	5	4	3	2	1	0
0	0	0	0	FCNnGMCSPRSC[3:0]			

FCNnGMCSPRSC[3:0]	Pre-CAN protocol layer basic system clock (f_{CANPRE})
0000_B	$f_{CAN}/1$
0001_B	$f_{CAN}/2$
0010_B	$f_{CAN}/3$
0011_B	$f_{CAN}/4$
0100_B	$f_{CAN}/5$
0101_B	$f_{CAN}/6$
0110_B	$f_{CAN}/7$
0111_B	$f_{CAN}/8$
1000_B	$f_{CAN}/9$
1001_B	$f_{CAN}/10$
1010_B	$f_{CAN}/11$
1011_B	$f_{CAN}/12$
1100_B	$f_{CAN}/13$
1101_B	$f_{CAN}/14$
1110_B	$f_{CAN}/15$
1111_B	$f_{CAN}/16$ (default value)

Note f_{CAN} = clock supplied to FCN on system level (clock generation, distribution and selection).

(3) FCNnGMABCTL - FCNn global automatic block transmission control register

This register is used to control the automatic block transmission (ABT) operation.

Access This register can be read/written in 16-bit units.

Address <FCNn_base> + $0\ 8018_H$

Initial Value 0000_H. The register is initialized by any reset.

(a) FCNnGMABCTL read

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
7	6	5	4	3	2	1	0
0	0	0	0	0	0	FCNnGMABCLRF	FCNnGMABABTT

FCNnGMABCLRF	Automatic block transmission engine clear status bit
0	Clearing the automatic transmission engine is completed.
1	The automatic transmission engine is being cleared.

Note Start automatic transmission engine clearance by FCNnGMABCTL.FCNnGMABCLAT = 1 while FCNnGMABCTL.FCNnGMABCLRF = 0.
The operation is not guaranteed if FCNnGMABCLRF is set to 1 while FCNnGMABCLRF = 1.

FCNnGMABABTT	Automatic block transmission status bit
0	Automatic block transmission is stopped.
1	Automatic block transmission is under execution.

(b) FCNnGMABCTL write

15	14	13	12	11	10	9	8
0	0	0	0	0	0	FCNnGMABSEAC	FCNnGMABSEAT
7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	FCNnGMABCLAT

Note When the automatic block transmission engine is cleared by setting FCNnGMABCTL.FCNnGMABSEAC to 1, FCNnGMABCLRF is automatically set, and cleared to 0 as soon as the requested clearing processing is completed.

Cautions

1. Before changing the normal operation mode with ABT to the initialization mode, be sure to set the FCNnGMABCTL register to the default value (0000_H) and confirm the FCNnGMABCTL register is surely initialized to the default value (0000_H).
2. Do not start automatic block transmission in the initialization mode. If automatic block transmission is started in the initialization mode, the operation is not guaranteed after the CAN Controller has entered the normal operation mode with ABT.
3. Do not start automatic block transmission while FCNnCMCLCTL.FCNnCMCLSSTS is set to 1 (transmission in progress). Confirm FCNnCMCLSSTS = 0 directly in advance before starting automatic block transmission.

FCNnGMABSEAC	Automatic block transmission engine clear request bit
0	The automatic block transmission engine is in idle status or under operation.
1	Request to clear the automatic block transmission engine. After the automatic block transmission engine has been cleared, automatic block transmission is started from message buffer 0 by setting the FCNnGMABCTL.FCNnGMABABTT = 1.

FCNnGMABSEAT	FCNnGMABCLAT	Automatic block transmission start bit
0	1	Request to stop automatic block transmission.
1	0	Request to start automatic block transmission.
Other than above		No change of FCNnGMABCTL.FCNnGMABABTT.

(4) FCNnGMADCTL - FCNn global automatic block transmission delay register

This register is used to set the interval at which the data of the message buffer assigned to ABT is to be transmitted in the normal operation mode with ABT.

Access This register can be read/written in 8-bit units.

Address <FCNn_base> + 0020_H

Initial Value 00_H. The register is initialized by any reset.

7	6	5	4	3	2	1	0
0	0	0	0	FCNnGMADSSAD[3:0]			

FCNnGMADSSAD[3:0]	Data frame interval during automatic block transmission in DBT[a]
0000_B	0 DBT (default value)
0001_B	2^5 DBT
0010_B	2^6 DBT
0011_B	2^7 DBT
0100_B	2^8 DBT
0101_B	2^9 DBT
0110_B	2^{10} DBT
0111_B	2^{11} DBT
1000_B	2^{12} DBT
Other than above	Setting prohibited

a) Unit: Data bit time (DBT)

Cautions

1. Do not change the contents of the FCNnGMADCTL register while FCNnGMABCTL.FCNnGMABCLRF = 1 (clearing of ABT in progress).
2. The timing at which the ABT message is actually transmitted onto the CAN bus differs depending on the status of transmission from the other station or how a request to transmit a message other than an ABT message is made.

(5) FCNnDNBMRXk – FCNn global data update bit monitor register (k = 0 to 3)

These registers are used to read the data update bits of several message buffers at a time, globally.

Access These registers can be read in 32-bit units.

Address FCNnDNBMRX0: <FCNn_base> + 1 $00C0_H$
FCNnDNBMRX1: <FCNn_base> + 1 $00D0_H$
Following registers are available only with m = 128 message buffers:
FCNnDNBMRX2: <FCNn_base> + 1 $00E0_H$
FCNnDNBMRX3: <FCNn_base> + 1 $00F0_H$

Initial Value 0000_H. This register is initialized by any reset.

31	30	29	28	27	26	25	24
FCNnDNBMSSDN[31:24]							

23	22	21	20	19	18	17	16
FCNnDNBMSSDN[23:16]							

15	14	13	12	11	10	9	8
FCNnDNBMSSDN[15:8]							

7	6	5	4	3	2	1	0
FCNnDNBMSSDN[7:0]							

FCNnDNBMSSDN[31:0]	Message buffer data update bit
0	No remote or data frame has been stored into the message buffer.
1	A remote or data frame has been stored into the message buffer.

24.5.2 FCN module registers

(1) FCNnCMMKCTLaH - FCNn module mask control register

These registers are used to extend the number of receivable messages into the same message buffer by masking part of the identifier (ID) comparison of a message and invalidating the ID of the masked part.

Two 16-bit registers FCNnCMMKCTLaH (a = 01 to 16) can also be accessed via a single 32-bit access to the registers FCNnCMMKCTLaW (a = 01, 03, 05, 07, 09, 11, 13, 15).

Access The FCNnCMMKCTLaH registers can be read/written in 16-bit units.
The FCNnCMMKCTLaW registers can be read/written in 32-bit units.

Address

FCNnCMMKCTL01H: <FCNn_base> + $0\ 8300_H$
FCNnCMMKCTL02H: <FCNn_base> + $0\ 8308_H$
FCNnCMMKCTL03H: <FCNn_base> + $0\ 8310_H$
FCNnCMMKCTL04H: <FCNn_base> + $0\ 8318_H$
FCNnCMMKCTL05H: <FCNn_base> + $0\ 8320_H$
FCNnCMMKCTL06H: <FCNn_base> + $0\ 8328_H$
FCNnCMMKCTL07H: <FCNn_base> + $0\ 8330_H$
FCNnCMMKCTL08H: <FCNn_base> + $0\ 8338_H$
FCNnCMMKCTL09H: <FCNn_base> + $0\ 8340_H$
FCNnCMMKCTL10H: <FCNn_base> + $0\ 8348_H$
FCNnCMMKCTL11H: <FCNn_base> + $0\ 8350_H$
FCNnCMMKCTL12H: <FCNn_base> + $0\ 8358_H$
FCNnCMMKCTL13H: <FCNn_base> + $0\ 8360_H$
FCNnCMMKCTL14H: <FCNn_base> + $0\ 8368_H$
FCNnCMMKCTL15H: <FCNn_base> + $0\ 8370_H$
FCNnCMMKCTL16H: <FCNn_base> + $0\ 8378_H$

FCNnCMMKCTL01W: <FCNn_base> + $1\ 0300_H$
FCNnCMMKCTL03W: <FCNn_base> + $1\ 0310_H$
FCNnCMMKCTL05W: <FCNn_base> + $1\ 0320_H$
FCNnCMMKCTL07W: <FCNn_base> + $1\ 0330_H$
FCNnCMMKCTL09W: <FCNn_base> + $1\ 0340_H$
FCNnCMMKCTL11W: <FCNn_base> + $1\ 0350_H$
FCNnCMMKCTL13W: <FCNn_base> + $1\ 0360_H$
FCNnCMMKCTL15W: <FCNn_base> + $1\ 0370_H$

Initial Value 0000_H for FCNnCMMKCTLaH. This register is initialized by any reset.
$0000\ 0000_H$ for FCNnCMMKCTLaW. This register is initialized by any reset.

(a) FCNnCMMKCTLaH (a = 01, 03, 05, 07, 09, 11, 13, 15)

15	14	13	12	11	10	9	8
FCNnCMMKSSID[15:8]							

7	6	5	4	3	2	1	0
FCNnCMMKSSID[7:0]							

(b) FCNnCMMKCTLaH (a = 02, 04, 06, 08, 10, 12, 14, 16)

15	14	13	12	11	10	9	8
0	0	0	FCNnCMMKSSID[28:24]				

7	6	5	4	3	2	1	0
FCNnCMMKSSID[23:16]							

(c) FCNnCMMKCTLaW (a = 01, 03, 05, 07, 09, 11, 13, 15)

31	30	29	28	27	26	25	24
			FCNnCMMKSSID[28:24]				

23	22	21	20	19	18	17	16
FCNnCMMKSSID[23:16]							

15	14	13	12	11	10	9	8
FCNnCMMKSSID[15:8]							

7	6	5	4	3	2	1	0
FCNnCMMKSSID[7:0]							

FCNnCMMKSSID[i] [a]	Mask pattern setting of ID bit
0	The ID bit **i** of the message buffer m set by FCNnMmSSID[**i**] are compared with the ID bits of the received message frame.
1	The ID bit **i** of the message buffer m set by FCNnMmSSID[**i**] are not compared with the ID bits of the received message frame (they are masked).

a) i = [28:0]

Note Masking is always defined by an ID length of 29 bits. If a mask is assigned to a message with a standard ID, FCNnCMMKSSID[17:0] are ignored. Therefore, only FCNnCMMKSSID[28:18] of the received ID are masked. The same mask can be used for both the standard and extended IDs.

(2) FCNnCMCLCTL - FCNn module control register

This register is used to control the operation mode of the FCN module.

Access This register can be read/written in 16-bit units.

Address <FCNn_base> + 0 8240_H

Initial Value 0000_H. The register is initialized by any reset.

(a) FCNnCMCLCTL read

15	14	13	12	11	10	9	8
0	0	0	0	0	0	FCNnCMCLSSRS	FCNnCMCLSSTS

7	6	5	4	3	2	1	0
FCNnCMCLERCF	FCNnCMCLALBF	FCNnCMCLVALF	FCNnCMCLMDPF[1:0]		FCNnCMCLMDOF[2:0]		

FCNnCMCLSSRS	Reception status bit
0	Reception is stopped.
1	Reception is in progress.

Notes

1. FCNnCMCLSSRS is set to 1 under the following conditions (timing)
   - The SOF bit of a receive frame is detected
   - On occurrence of arbitration loss during a transmit frame
2. FCNnCMCLSSRS is cleared to 0 under the following conditions (timing)
   - When a recessive level is detected at the second bit of the interframe space
   - On transition to the initialization mode at the first bit of the interframe space

FCNnCMCLSSTS	Transmission status bit
0	Transmission is stopped.
1	Transmission is in progress.

Notes

1. FCNnCMCLSSTS is set to 1 under the following conditions (timing)
   - The SOF bit of a transmit frame is detected
2. FCNnCMCLSSTS is cleared to 0 under the following conditions (timing)
   - During transition to bus-off state
   - On occurrence of arbitration loss in transmit frame
   - On detection of recessive level at the second bit of the interframe space
   - On transition to the initialization mode at the first bit of the interframe space

FCNnCMCLERCF	Error counter clear bit
0	The FCNnCMERCNT and FCNnCMINSTR registers are not cleared in the initialization mode.
1	The FCNnCMERCNT and FCNnCMINSTR registers are cleared in the initialization mode.

Caution FCNnCMCLERCF is used to clear the error counter FCNnCMERCNT and information register FCNnCMINSTR for re-initialization or forced recovery from the bus-off state. The error counter and the information register can be cleared under the following conditions (by setting FCNnCMCLERCF):

- In the initialization mode during the bus-off period
- In the initialization mode after the FCN module starts up (by changing FCNnGMCLPWOM from 0 to 1)
- In the initialization mode entered after all the transmission requests have been cleared in accordance with the transmission abort processing shown in *Table 24-24 "Transmission abort processing (except normal operation mode with ABT)"* in an operation mode. (In normal operation mode with ABT, clear all the transmission requests in accordance with the transmission abort processing shown in *Table 24-25 "Transmission abort processing (normal operation mode with ABT) - Repeat option for aborted message"*.)

Notes

1. When the FCNnCMERCNT and FCNnCMINSTR registers have been cleared, FCNnCMCLERCF is also cleared to 0 automatically.
2. FCNnCMCLERCF can be set to 1 at the same time as a request to change the initialization mode to an operation mode is made.
3. FCNnCMCLERCF is read-only in the FCN sleep mode or FCN stop mode.
4. The error counter can also be cleared by a normal shutdown or forced shutdown of the CAN controller.

FCNnCMCLALBF	Bit to set operation in case of arbitration loss
0	Re-transmission is not executed in case of an arbitration loss in the single-shot mode.
1	Re-transmission is executed in case of an arbitration loss in the single-shot mode.

Note FCNnCMCLALBF is valid only in the single-shot mode.

FCNnCMCLVALF	Valid receive message frame detection bit
0	A valid message frame has not been received since FCNnCMCLVALF was last cleared to 0.
1	A valid message frame has been received since FCNnCMCLVALF was last cleared to 0.

Notes

1. Detection of a valid receive message frame is not dependent upon storage in the receive message buffer (data frame/remote frame) or transmit message buffer (remote frame).
2. If only two CAN nodes are connected to the CAN bus with one transmitting a message frame in the normal mode and the other in the receive-only mode, FCNnCMCLVALF is not set to 1 before the transmitting node enters the error passive state, because in receive-only mode no acknowledge is generated.
3. To clear FCNnCMCLVALF, set FCNnCMCLCLVL to 1 first and confirm that FCNnCMCLVALF is cleared. If it is not cleared, perform clearing processing again.

FCNnCMCLMDPF[1:0]	Power save mode
00_B	No power save mode is selected.
01_B	FCN sleep mode
10_B	Setting prohibited
11_B	FCN stop mode

Cautions

1. Transition to and from the FCN stop mode must be made via FCN sleep mode. A request for direct transition to and from the FCN stop mode is ignored.
2. The FCNnGMCLSSMO flag of FCNnGMCLCTL must be checked after releasing a power save mode, prior to access the message buffers again.
3. FCN sleep mode requests are kept pending, until cancelled by software or entered on appropriate bus condition (bus idle). Software can check the actual status by reading FCNnCMCLMDPF[1:0].
4. Power save mode cannot be set in combination with the change of operation mode. Be sure to perform these operations in different steps.

Note When the system transitions from initialization mode to a communication mode, the FCN module participates in communication after first confirming the CAN bus idle period. Although it is possible to transition to sleep mode before the idle period has been confirmed, in this case, the wakeup condition will always change from recessive level to dominant level.

FCNnCMCLMDOF[2:0]	Operation mode
000_B	No operation mode is selected (FCN module is in the initialization mode).
001_B	Normal operation mode
010_B	Normal operation mode with automatic block transmission function (normal operation mode with ABT)
011_B	Receive-only mode
100_B	Single-shot mode
101_B	Self-test mode
Other than above	Setting prohibited

Cautions

1. Transit to initialization mode or power saving modes may take some time. Be sure to verify the success of mode change by reading the values, before proceeding.
2. If initialization mode is set while receiving data in operation mode, data in the message buffer that sets the FCNnMmDTNF flag might be received last. However, the receive history list is cleared upon transition to operation mode. It is therefore necessary to confirm that initialization mode was set by reading the operation mode. Before restarting operation mode, make sure to clear all FCNnMmDTNF flags in every valid reception message buffer.

Note FCNnCM.FCNnCMCLMDOF[2:0] are read-only in the FCN sleep mode or FCN stop mode.

(b) FCNnCMCLCTL write

15	14	13	12	11	10	9	8
FCNnCM CLSERC	FCNnCM CLSEAL	0	FCNnCM CLSEPS[1:0]		FCNnCM CLSEOP[2:0]		
7	6	5	4	3	2	1	0
0	FCNnCM CLCLAL	FCNnCM CLCLVL	FCNnCM CLCLPS[1:0]		FCNnCM CLCLOP[2:0]		

FCNnCMCLSERC	Setting of FCNnCMCLERCF bit
1	FCNnCMCLERCF is set to 1.
Other than above	FCNnCMCLERCF is not changed.

FCNnCMCLSEAL	FCNnCMCLCLAL	Setting of FCNnCMCLALBF bit
0	1	FCNnCMCLALBF is cleared to 0.
1	0	FCNnCMCLALBF is set to 1.
Other than above		FCNnCMCLALBF is not changed.

FCNnCMCLCLVL	Setting of FCNnCMCLVALF bit
0	FCNnCMCLVALF is not changed.
1	FCNnCMCLVALF is cleared to 0.

FCNnCMSESEPS0	FCNnCMCLCLPS0	Setting of FCNnCMCLMDPF0 bit
0	1	FCNnCMCLMDPF0 is cleared to 0.
1	0	FCNnCMCLMDPF0 is set to 1.
Other than above		FCNnCMCLMDPF0 is not changed.

FCNnCMSESEPS1	FCNnCMCLCLPS1	Setting of FCNnCMCLMDPF1 bit
0	1	FCNnCMCLMDPF1 is cleared to 0.
1	0	FCNnCMCLMDPF1 is set to 1.
Other than above		FCNnCMCLMDPF1 is not changed.

FCNnCMCLSEOP0	FCNnCMCLCLOP0	Setting of FCNnCMCLMDOF0 bit
0	1	FCNnCMCLMDOF0 is cleared to 0.
1	0	FCNnCMCLMDOF0 is set to 1.
Other than above		FCNnCMCLMDOF0 is not changed.

FCNnCMCLSEOP1	FCNnCMCLCLOP1	Setting of FCNnCMCLMDOF1 bit
0	1	FCNnCMCLMDOF1 is cleared to 0.
1	0	FCNnCMCLMDOF1 is set to 1.
Other than above		FCNnCMCLMDOF1 is not changed.

FCNnCMCLSEOP2	FCNnCMCLCLOP2	Setting of FCNnCMCLMDOF2 bit
0	1	FCNnCMCLMDOF2 is cleared to 0.
1	0	FCNnCMCLMDOF2 is set to 1.
Other than above		FCNnCMCLMDOF2 is not changed.

(3) FCNnCMLCSTR - FCNn module last error information register

This register provides the error information of the CAN protocol.

Access This register can be read/written in 8-bit units.

Address <FCNn_base> + 0248_H

Initial Value 00_H. The register is initialized by any reset.

7	6	5	4	3	2	1	0
0	0	0	0	0	FCNnCMLCSSLC[2:0]		

Notes

1. The contents of the FCNnCMLCSTR register are not cleared when the FCN module changes from an operation mode to the initialization mode.
2. If an attempt is made to write a value other than 00_H to the FCNnCMLCSTR register by software, the access is ignored.

FCNnCMLCSSLC[2:0]	Last CAN protocol error information
000_B	No error
001_B	Stuff error
010_B	Form error
011_B	ACK error
100_B	Bit error. (The FCN module tried to transmit a recessive-level bit as part of a transmit message (except the arbitration field), but the value on the CAN bus is a dominant-level bit.)
101_B	Bit error. (The FCN module tried to transmit a dominant-level bit as part of a transmit message, ACK bit, error frame, or overload frame, but the value on the CAN bus is a recessive-level bit.)
110_B	CRC error
111_B	Undefined

(4) FCNnCMINSTR - FCNn module information register

This register indicates the status of the FCN module.

Access This register is read-only in 8-bit units.

Address <FCNn_base> + 0 $024C_H$

Initial Value 00_H. The register is initialized by any reset.

7	6	5	4	3	2	1	0
0	0	0	FCNnCMINBOFF	FCNnCMINSSTE[1:0]		FCNnCMINSSRE[1:0]	

FCNnCMINBOFF	Bus-off state bit
0	Not bus-off state (transmit error counter ≤ 255). (The value of the transmit counter is less than 256.)
1	Bus-off state (transmit error counter > 255). (The value of the transmit counter is 256 or more.)

FCNnCMINSSTE[1:0]	Transmission error counter status bit
00_B	The value of the transmission error counter is less than that of the warning level (< 96).
01_B	The value of the transmission error counter is in the range of the warning level (96 to 127).
10_B	Undefined
11_B	The value of the transmission error counter is in the range of the error passive or bus-off status (≥ 128).

FCNnCMINSSRE[1:0]	Reception error counter status bit
00_B	The value of the reception error counter is less than that of the warning level (< 96).
01_B	The value of the reception error counter is in the range of the warning level (96 to 127).
10_B	Undefined
11_B	The value of the reception error counter is in the error passive range (≥ 128).

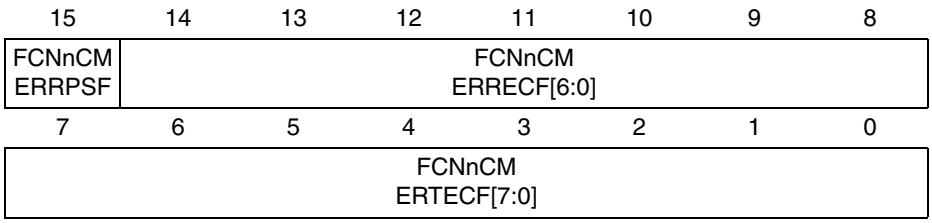
(5) FCNnCMERCNT - FCNn module error counter register

This register indicates the count value of the transmission/reception error counter.

Access This register is read-only in 16-bit units.

Address <FCNn_base> + $0\ 8250_H$

Initial Value 0000_H. The register is initialized by any reset.



FCNnCMERRPSF	Reception error passive status bit
0	The reception error counter is not in the error passive range (< 128)
1	The reception error counter is in the error passive range (≥ 128)

FCNnCMERRECF[6:0]	Reception error counter bit
0 to 127	The reception error count. These bits reflect the status of the reception error counter. The error count is defined by the CAN protocol.

Note FCNnCMERRECF[6:0] are invalid in the reception error passive state (FCNnCMINSTR.FCNnCMINSSRE[1:0] = 11_B).

FCNnCMERTECF[7:0]	Transmission error counter bit
0 to 255	Number of transmission errors. These bits reflect the status of the transmission error counter. The number of errors is defined by the CAN protocol.

Note FCNnCMERTECF[7:0] are invalid in the bus-off state (FCNnCMINSTR.FCNnCMINBOFF = 1).

(6) FCNnCMIECTL - FCNn module interrupt enable register

This register is used to enable or disable the interrupts of the FCN module.

Access This register can be read/written in 16-bit units.

Address <FCNn_base> + $0\ 8258_H$

Initial Value 0000_H. The register is initialized by any reset.

(a) FCNnCMIECTL read

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
7	6	5	4	3	2	1	0
0	FCNnCMIEINTF[6:0]						

FCNnCMIEINTF[6:0]	FCN module interrupt enable bit
0	Output of the interrupt corresponding to interrupt status register FCNnCMISCTL is disabled.
1	Output of the interrupt corresponding to interrupt status register FCNnCMISCTL is enabled.

(b) FCNnCMIECTL write

15	14	13	12	11	10	9	8
0	FCNnCMIESEIE[6:0]						
7	6	5	4	3	2	1	0
0	FCNnCMIECLIE[6:0]						

FCNnCMIESEIE[6:0]	FCNnCMIECLIE[6:0]	Setting of FCNnCMIEINTF[6:0] bit
0	1	FCNnCMIEINTF[6:0] bit is cleared to 0.
1	0	FCNnCMIEINTF[6:0] bit is set to 1.
Other than above		FCNnCMIEINTF[6:0] bit is not changed.

(7) FCNnCMISCTL - FCNn module interrupt status register

This register indicates the interrupt status of the FCN module.

Access This register can be read/written in 16-bit units.

Address <FCNn_base> + $0\ 8260_H$

Initial Value 0000_H. The register is initialized by any reset.

(a) FCNnCMISCTL read

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
7	6	5	4	3	2	1	0
0	FCNnCMISITSF[6:0]						

FCNnCMISITSF[6:0]	FCN interrupt status bit
0	No related interrupt source event is pending
1	A related interrupt source event is pending

Interrupt status bit	Related interrupt source event
FCNnCMISITSF6	FCN module transmission abort interrupt status bit
FCNnCMISITSF5	Wakeup interrupt from FCN sleep mode [a]
FCNnCMISITSF4	Arbitration loss interrupt
FCNnCMISITSF3	CAN protocol error interrupt
FCNnCMISITSF2	CAN error status interrupt
FCNnCMISITSF1	Interrupt on completion of reception of valid message frame to message buffer m
FCNnCMISITSF0	Interrupt on normal completion of transmission of message frame from message buffer m

a) FCNnCMISITSF5 is set only when the FCN module is woken up from the FCN sleep mode by a CAN bus operation. It is not set when the FCN sleep mode has been released by software.

(b) FCNnCMISCTL write

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
7	6	5	4	3	2	1	0
0	FCNnCMISITSF[6:0]						

FCNnCMISCLTS[6:0]	Clearing of FCNnCMISITSF[6:0]
0	FCNnCMISITSF[6:0] bits are not changed.
1	FCNnCMISITSF[6:0] bits are cleared to 0.

Caution Clear the status bit of this register by software, when the confirmation of each status is necessary in the interrupt processing, because these bits are not cleared automatically.

(8) FCNnCMBRPRS - FCNn module bit rate prescaler register

This register is used to select the CAN protocol layer basic system clock (f_{TQ}). The communication baud rate is set to the FCNnCMBTCTL register.

Access This register can be read/written in 8-bit units.

Address <FCNn_base> + $0\ 0268_H$

Initial Value FF_H. The register is initialized by any reset.

7	6	5	4	3	2	1	0
FCNnCMBRPRS[7:0]							

FCNnCMBRPRS[7:0]	CAN protocol layer basic system clock (f_{TQ})
0	$f_{CANPRE}/1$
1	$f_{CANPRE}/2$
n	$f_{CANPRE}/(n+1)$
:	:
255	$f_{CANPRE}/256$ (default value)

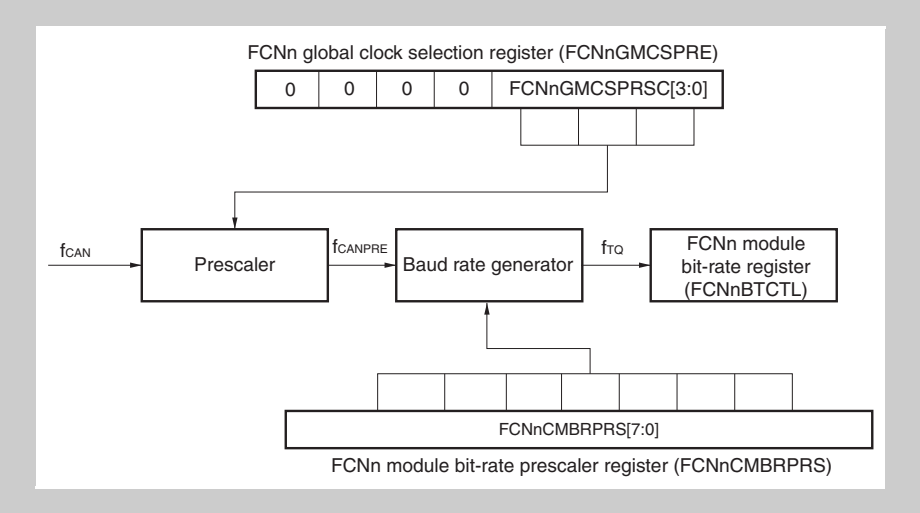


Figure 24-2 FCN module clock

Note f_{CAN}: Clock supplied to FCN
f_{CANPRE}: PRE-CAN protocol layer basic system clock
f_{TQ}: CAN protocol layer basic system clock

Caution FCNnCMBRPRS can be write-accessed only in the initialization mode.

(9) FCNnCMBTCTL - FCNn module bit rate register

This register is used to control the data bit time of the communication baud rate.

Access This register can be read/written in 16-bit units.

Address <FCNn_base> + 0 8270_H

Initial Value $370F_H$. The register is initialized by any reset.

15	14	13	12	11	10	9	8
0	0	FCNnCM BTJWLG[1:0]		0	FCNnCM BTS2LG[2:0]		
7	6	5	4	3	2	1	0
0	0	0	0	FCNnCMBTS1LG[3:0]			

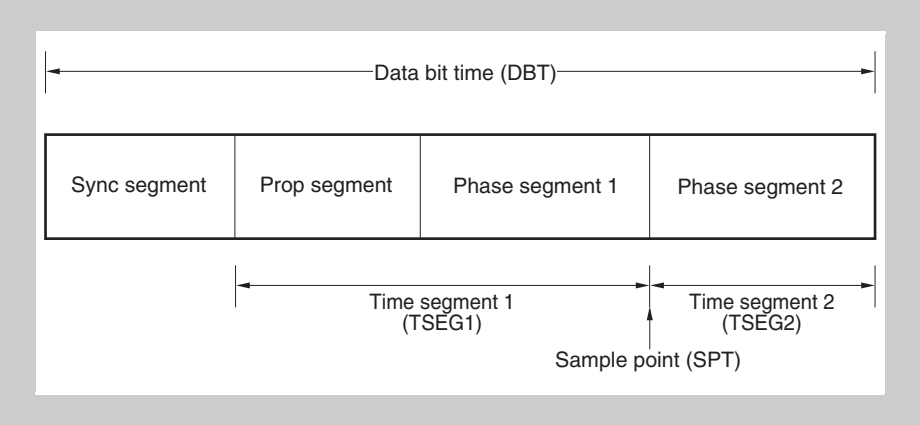


Figure 24-3 Data bit time

FCNnCMBTJWLG[1:0]	Length of synchronization jump width (SJW)
00_B	$1T_Q$
01_B	$2T_Q$
10_B	$3T_Q$
11_B	$4T_Q$ (default value)

FCNnCMBTS2LG[2:0]	Length of time segment 2 (TSEG2)
000_B	$1T_Q$
001_B	$2T_Q$
010_B	$3T_Q$
011_B	$4T_Q$
100_B	$5T_Q$
101_B	$6T_Q$
110_B	$7T_Q$
111_B	$8T_Q$ (default value)

FCNnCMBTS1LG[3:0]	Length of time segment 1 (TSEG1)
0000_B	Setting prohibited
0001_B	$2T_Q$[a]
0010_B	$3T_Q$[a]
0011_B	$4T_Q$
0100_B	$5T_Q$
0101_B	$6T_Q$
0110_B	$7T_Q$
0111_B	$8T_Q$
1000_B	$9T_Q$
1001_B	$10T_Q$
1010_B	$11T_Q$
1011_B	$12T_Q$
1100_B	$13T_Q$
1101_B	$14T_Q$
1110_B	$15T_Q$
1111_B	$16T_Q$ (default value)

a) This setting must not be made when the FCNnCMBRPRS register = 00_H

Note $T_Q = 1/f_{TQ}$ (f_{TQ}: CAN protocol layer basic system clock)

(10) FCNnCMLISTR - FCNn module last in-pointer register

This register indicates the number of the message buffer in which a data frame or a remote frame was last stored.

Access This register is read-only in 8-bit units.

Address <FCNn_base> + 0278_H

Initial Value Undefined.

7	6	5	4	3	2	1	0
FCNnCMLISSLR[7:0]							

FCNnCMLISSLR[7:0]	Last in-pointer register of receive history list
0 to 63 [a] 0 to 127 [b]	Reading the FCNnCMLISTR register obtains the number of the message buffer storing the last data frame or remote frame to be received.

a) On 64 message buffer FCN.
b) On 128 message buffer FCN.

Note The read value of FCNnCMLISTR is undefined if a data frame or a remote frame has never been received and stored in the message buffer. If FCNnCMRGRX.FCNnCMRGSSPM is set to 1 after the FCN module has changed from the initialization mode to an operation mode, therefore, the read value of FCNnCMLISTR is undefined.

(11) FCNnCMRGRX - FCNn module receive history list register

This register is used to read the receive history list (RHL).

Access This register can be read/written in 16-bit units.

Address <FCNn_base> + 0 8280_H

Initial Value $xx02_H$. The register is initialized by any reset.

(a) FCNnCMRGRX read

15	14	13	12	11	10	9	8
FCNnCMRGSSPT[7:0]							
7	6	5	4	3	2	1	0
0	0	0	0	0	0	FCNnCMRGSSPM	FCNnCMRGRVFF

FCNnCMRGSSPT[7:0]	Receive history list read pointer
0 to 63 [a] 0 to 127 [b]	When FCNnCMRGRX is read, the contents of the element indexed by the receive history list get pointer (FCNnCMRGRX.FCNnCMRGSSPT) of the receive history list are read. These contents indicate the number of the message buffer in which a data frame or a remote frame has been stored.

a) On 64 message buffer FCN.
b) On 128 message buffer FCN.

FCNnCMRGSSPM[a]	Receive history list pointer match
0	The receive history list has at least one message buffer number that has not been read.
1	The receive history list has no message buffer numbers that have not been read.

a) The read value of FCNnCMTGSSPT[7:0] is invalid when FCNnCMTGSSPM = 1.

FCNnCMRGRVFF[a]	Receive history list overflow bit [b]
0	All the message buffer numbers that have not been read are preserved. All the numbers of the message buffers in which a new data frame or remote frame has been received and stored are recorded to the receive history list (the receive history list has a vacant element).
1	At least **(i)** entries have been stored since the host processor has serviced the RHL last time (i.e. read FCNnCMRGRX). The first **(i-1)** entries are sequentially stored while the last entry can have been overwritten whenever newly received message is stored, because all buffer numbers are stored at position **(i)**, when FCNnCMRGRVFF is set. Thus the sequence of receptions can not be recovered completely now.

a) If all the receive history entries have been read by the FCNnCMRGRX register while FCNnCMRGRVFF is set (1), FCNnCMRGSSPM is not cleared even if a new message is received.
b) i = 47 on 64 message buffer FCN;
i = 95 on 128 message buffer FCN.

(b) FCNnCMRGRX write

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	FCNnCMRGCLRV

FCNnCMRGCLRV	Clearing of FCNnCMRGRVFF bit
0	FCNnCMRGRVFF bit is not changed.
1	FCNnCMRGRVFF bit is cleared to 0.

(12) FCNnCMLOSTR - FCNn module last out-pointer register

This register indicates the number of the message buffer, from which a data frame or a remote frame was transmitted last.

Access This register is read-only in 8-bit units.

Address <FCNn_base> + 0288_H

Initial Value Undefined

7	6	5	4	3	2	1	0
FCNnCMLOSSLT[7:0]							

FCNnCMLOSSLT[7:0]	Last out-pointer of transmit history list
0 to 63 [a] 0 to 127 [b]	When the FCNnCMLOSTR register is read, the contents of the element indexed by the last out-pointer (FCNnCMLOSTR[7:0]) of the receive history list are read. These contents indicate the number of the message buffer to which a data frame or a remote frame was transmitted last.

a) On 64 message buffer FCN.
b) On 128 message buffer FCN.

Note The value read from the FCNnCMLOSTR register is undefined if a data frame or remote frame has never been transmitted from a message buffer.

(13) FCNnCMTGTX - FCNn module transmit history list register

This register is used to read the transmit history list (THL).

Access This register can be read/written in 16-bit units.

Address <FCNn_base> + 0 8290_H

Initial Value $xx02_H$. The register is initialized by any reset.

(a) FCNnCMTGTX read

15	14	13	12	11	10	9	8
FCNnCMTGSSPT[7:0]							
7	6	5	4	3	2	1	0
0	0	0	0	0	0	FCNnCMTGSSPM	FCNnCMTGTVFF

FCNnCMTGSSPT[7:0]	Transmit history list read pointer
0 to 63 [a] 0 to 127 [b]	When the FCNnCMTGTX register is read, the contents of the element indexed by the read pointer (FCNnCMTGSSPT[7:0]) of the transmit history list are read. These contents indicate the number of the message buffer to which a data frame or a remote frame was transmitted last.

a) On 64 message buffer FCN.
b) On 128 message buffer FCN.

FCNnCMTGSSPM[a]	Transmit history pointer match
0	The transmit history list has at least one message buffer number that has not been read.
1	The transmit history list has no message buffer numbers that have not been read.

a) The read value of FCNnCMTGSSPT[7:0] is invalid when the FCNnCMTGSSPM = 1.

FCNnCMTGTVFF[a]	Transmit history list overflow bit [b]
0	All the message buffer numbers that have not been read are preserved. All the numbers of the message buffers to which a new data frame or remote frame has been transmitted are recorded to the transmit history list (the transmit history list has a vacant element).
1	At least **(i)** entries have been stored since the host processor has serviced the THL last time (i.e. read FCNnCMTGTX). The first **(i-1)** entries are sequentially stored while the last entry can have been overwritten whenever newly received message is stored, because all buffer numbers are stored at position **(i)**, when FCNnCMTGTVFF is set. Thus the sequence of receptions can not be recovered completely now.

a) If FCNnCMTGTVFF is set, FCNnCMTGSSPM is no longer cleared on message transmission, but FCNnCMTGSSPM is still set, if all entries of FCNnCMTGTX are read by software.
b) i = 15 on 64 message buffer FCN;
i = 31 on 128 message buffer FCN.

Note Transmission from message buffers ...

- 0 to 16 (for 64 message buffer FCN)
- 0 to 32 (for 128 message buffer FCN)

... is not recorded to the transmit history list in the normal operation mode with ABT.

(b) FCNnCMTGTX write

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	FCNnCMTGCLTV

FCNnCMTGCLTV	Setting of FCNnCMTGTVFF bit
0	FCNnCMTGTVFF bit is not changed
1	FCNnCMTGTVFF bit is cleared to 0

(14) FCNnCMTSCTL - FCNn module time stamp register

This register is used to control the time stamp function.

Access This register can be read/written in 16-bit units.

Address <FCNn_base> + 0 8298_H

Initial Value 0000_H. The register is initialized by any reset.

(a) FCNnCMTSCTL read

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
7	6	5	4	3	2	1	0
0	0	0	0	0	FCNnCMTSLOKE	FCNnCMTSSELE	FCNnCMTSTSGE

Note The lock function of the time stamp function must not be used when the FCN module is in the normal operation mode with ABT.

FCNnCMTSLOKE	Time stamp lock function enable bit
0	Time stamp lock function stopped. The TSOUT signal is toggled each time the selected time stamp capture event occurs.
1	Time stamp lock function enabled. The TSOUT signal is toggled each time the selected time stamp capture event occurs. However, the TSOUT output signal is locked when a data frame has been correctly received to message buffer 0 [a].

a) FCNnCMTSTSGE is automatically cleared to 0.

FCNnCMTSSELE	Time stamp capture event selection bit
0	The timestamp capture event is SOF.
1	The time stamp capture event is the last bit of EOF.

FCNnCMTSTSGE	TSOUT operation setting bit
0	TSOUT toggle operation is disabled.
1	TSOUT toggle operation is enabled.

(b) FCNnCMTSCTL write

15	14	13	12	11	10	9	8
0	0	0	0	0	FCNnCMTSSELK	FCNnCMTSSES	FCNnCMTSSETS
7	6	5	4	3	2	1	0
0	0	0	0	0	FCNnCMTSCLLK	FCNnCMTSCLSL	FCNnCMTSCLTS

FCNnCMTSSELK	FCNnCMTSCLLK	Setting of FCNnCMTSLOKE bit
0	1	FCNnCMTSLOKE is cleared to 0.
1	0	FCNnCMTSLOKE is set to 1.
Other than above		FCNnCMTSLOKE is not changed.

FCNnCMTSSESL	FCNnCMTSCLSL	Setting of FCNnCMTSSELE bit
0	1	FCNnCMTSSELE is cleared to 0.
1	0	FCNnCMTSSELE is set to 1.
Other than above		FCNnCMTSSELE is not changed.

FCNnCMTSSETS	FCNnCMTSCLTS	Setting of FCNnCMTSTSGE bit
0	1	FCNnCMTSTSGE is cleared to 0.
1	0	FCNnCMTSTSGE is set to 1.
Other than above		FCNnCMTSTSGE is not changed.

24.5.3 FCN message buffer registers

(1) FCNnMmDATxB/H/W, FCNn message data byte registers

These registers are used to store the data of a transmit/receive message.

Access The FCNnMmDATxW registers can be read/written in 32-bit units.
The FCNnMmDATxH registers can be read/written in 16-bit units.
The FCNnMmDATxB registers can be read/written in 8-bit units.

Address FCNnMmDAT0B: <FCNn_base> + 0 1000_H + m x 40_H
FCNnMmDAT1B: <FCNn_base> + 0 1004_H + m x 40_H
FCNnMmDAT2B: <FCNn_base> + 0 1008_H + m x 40_H
FCNnMmDAT3B: <FCNn_base> + 0 $100C_H$ + m x 40_H
FCNnMmDAT4B: <FCNn_base> + 0 1010_H + m x 40_H
FCNnMmDAT5B: <FCNn_base> + 0 1014_H + m x 40_H
FCNnMmDAT6B: <FCNn_base> + 0 1018_H + m x 40_H
FCNnMmDAT7B: <FCNn_base> + 0 $101C_H$ + m x 40_H

FCNnMmDAT0H: <FCNn_base> + 0 9000_H + m x 40_H
FCNnMmDAT2H: <FCNn_base> + 0 9008_H + m x 40_H
FCNnMmDAT4H: <FCNn_base> + 0 9010_H + m x 40_H
FCNnMmDAT6H: <FCNn_base> + 0 9018_H + m x 40_H

FCNnMmDAT0W: <FCNn_base> + 1 1000_H + m x 40_H
FCNnMmDAT4W: <FCNn_base> + 1 1010_H + m x 40_H

Initial Value $0000\ 0000_H$ for FCNnMmDATxW. This register is initialized by any reset.
0000_H for FCNnMmDATxH. This register is initialized by any reset.
00_H for FCNnMmDATxB. This register is initialized by any reset.

(a) FCNnCMmDATxB (x = 0 to 7)

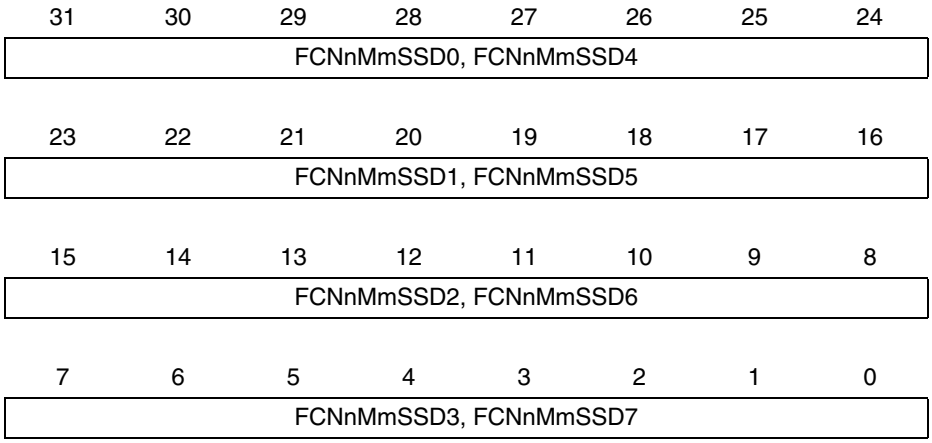
7	6	5	4	3	2	1	0
FCNnMmSSD0, FCNnMmSSD1, FCNnMmSSD2, FCNnMmSSD3, FCNnMmSSD4, FCNnMmSSD5, FCNnMmSSD6, FCNnMmSSD7							

(b) FCNnCMmDATxH (x = 0, 2, 4, 6)

15	14	13	12	11	10	9	8
FCNnMmSSD0, FCNnMmSSD2, FCNnMmSSD4, FCNnMmSSD6							

7	6	5	4	3	2	1	0
FCNnMmSSD1, FCNnMmSSD3, FCNnMmSSD5, FCNnMmSSD7							

(c) FCNnCMmDATxW (x = 0, 4)



(2) FCNnMmDTLGB - FCNn message data length register m

This register is used to set the number of bytes of the data field of a message buffer (DLC).

Access This register can be read/written in 8-bit units.

Address <FCNn_base> + 0 1020_H + m x 40_H

Initial Value 00_H. This register is initialized by any reset.

7	6	5	4	3	2	1	0
0	0	0	0	FCNnMmDTLG[3:0]			

FCNnMmDTLG[3:0]	Data length of transmit/receive message
0000_B	0 bytes
0001_B	1 byte
0010_B	2 bytes
0011_B	3 bytes
0100_B	4 bytes
0101_B	5 bytes
0110_B	6 bytes
0111_B	7 bytes
1000_B	8 bytes
1001_B	Setting prohibited (If these bits are set during transmission, 8-byte data is transmitted regardless of the set FCNnMmDTLG[3:0] value when a data frame is transmitted. However, the DLC actually transmitted to the CAN bus is the DLC value set to this register.)Note
1010_B	
1011_B	
1100_B	
1101_B	
1110_B	
1111_B	

Note The data and DLC value actually transmitted to CAN bus are as follows.

Type of transmit frame	Length of transmit data	DLC transmitted
Data frame	Number of bits specified by FCNnMmDTLG[3:0] (However, 8 bytes if value ≥ 8)	FCNnMmDTLGB.FCNnMmDTLG[3:0] bits
Remote frame	0 bytes	

Cautions

1. Be sure to set bits 7 to 4 to 0000_B.
2. Receive data is stored in as many FCNnMmDATxB register as the number of bytes (however, the upper limit is 8) corresponding to DLC of the received frame. The FCNnMmDATxB register in which no data is stored is undefined.
3. On reception, FCNnMmDTLGB is updated according to the received frame.

(3) FCNnMmSTRB - FCNn message configuration register m

This register is used to specify the type of the message buffer and to set a mask.

Access This register can be read/written in 8-bit units.

Address <FCNn_base> + 0 1024_H + m x 40_H

Initial Value 00_H. This register is initialized by any reset.

7	6	5	4	3	2	1	0
FCNnMm SSOW	FCNnMm SSMT[3:0]				FCNnMm SSRT	0	FCNnMm SSAM

FCNnMmSSOW	Overwrite control bit
0	The message buffer that has already received a data frame[a] is not overwritten by a newly received data frame. The newly received data frame is discarded.
1	The message buffer that has already received a data frame[a] is overwritten by a newly received data frame.

a) The “message buffer that has already received a data frame” is a receive message buffer whose FCNnMmCTL.FCNnMmDTNF bit has been set to 1.

Note A remote frame is received and stored, regardless of the setting of FCNnMmCTL.FCNnMmSSOW and FCNnMmCTL.FCNnMmDTNF. A remote frame that satisfies the other conditions is always received and stored in the corresponding message buffer (interrupt generated, FCNnMmDTNF flag set, FCNnMmDTLGB.FCNnMmDTLG[3:0] updated, and recorded to the receive history list).

FCNnMmSSRT	Remote frame request bit
0	Transmit / receive a data frame.
1	Transmit / receive a remote frame.

FCNnMmSTRB.FCNnMmSSRT specifies the type of message frame that is transmitted or received from/to a message buffer.

Notes

1. If the message buffer is defined as a transmit message buffer, and a remote frame shall be received into it, the FCNnMmSSRT bit must be cleared.
2. Even if a valid remote frame has been received in a transmit message buffer, the FCNnMmSSRT bit of the transmit message buffer that has received the frame remains cleared to 0.
3. Even if a remote frame whose ID matches has been received from the CAN bus, if the FCNnMmSSRT bit of a transmit message buffer is set to 1 (to transmit a remote frame), that remote frame is not stored in this transmit message buffer.
4. If the message buffer is defined as a receive message buffer, the FCNnMmSSRT bit must be set, in order to receive remote frames instead of data frames.

FCNnMmSSMT[3:0]	Message buffer type setting bit
0000_B	Transmit message buffer
0001_B	Receive message buffer (no mask setting)
0010_B	Receive message buffer (mask 1 set)
0011_B	Receive message buffer (mask 2 set)
0100_B	Receive message buffer (mask 3 set)
0101_B	Receive message buffer (mask 4 set)
0110_B	Receive message buffer (mask 5 set)
0111_B	Receive message buffer (mask 6 set)
1000_B	Receive message buffer (mask 7 set)
1001_B	Receive message buffer (mask 8 set)
Other than above	Setting prohibited

Note The setting of FCNnMmSSMT is also valid to select masks in conjunction with remote frame reception. To receive remote frames in receive message buffers, the flag FCNnMmSSRT of the message buffer must be set.

FCNnMmSSAM	Message buffer assignment bit
0	Message buffer not used.
1	Message buffer used.

Caution Be sure to write 0 to bit 1.

(4) FCNnMmMID0H, FCNnMmMID1H, FCNnMmMID0W - FCNn message ID register m

These registers are used to set an identifier (ID).

Access FCNnMmMID0H, FCNnMmMID1H can be read/written in 16-bit units.
FCNnMmMID0W can be read/written in 32-bit units.

Address FCNnMmMID0H: <FCNn_base> + 0 9028_H + m x 40_H
FCNnMmMID1H: <FCNn_base> + 0 9030_H + m x 40_H

FCNnMmMID0W: <FCNn_base> + 1 1028_H + m x 40_H

Initial Value 0000_H for FCNnMmMID0H and FCNnMmMID1H. This register is initialized by any reset.
0000 0000_H for FCNnMmMID0W. This register is initialized by any reset.

(a) FCNnMmMID0H

15	14	13	12	11	10	9	8
FCNnMmSSID[15:8]							

7	6	5	4	3	2	1	0
FCNnMmSSID[7:0]							

(b) FCNnMmMID1H

15	14	13	12	11	10	9	8
FCNnMmSSIE	0	0	FCNnMmSSID[28:24]				

7	6	5	4	3	2	1	0
FCNnMmSSID[23:16]							

(c) FCNnCMmMID0W

31	30	29	28	27	26	25	24
FCNnMmSSIE	0	0	FCNnMmSSID[28:24]				

23	22	21	20	19	18	17	16
FCNnMmSSID[23:16]							

15	14	13	12	11	10	9	8
FCNnMmSSID[15:8]							

7	6	5	4	3	2	1	0
FCNnMmSSID[7:0]							

FCNnMmSSIE	Format mode specification bit
0	Standard format mode (FCNnMmSSID[28:18]: 11 bits, FCNnMmSSID[17:0] are not used)
1	Extended format mode (FCNnMmSSID[28:0]: 29 bits)

FCNnMmSSID[28:0]	Message ID
FCNnMmSSID[28:18]	Standard ID value of 11 bits (when FCNnMmSSIE = 0)
FCNnMmSSID[28:0]	Extended ID value of 29 bits (when FCNnMmSSIE = 1)

Cautions

1. Be sure to write 0 to bits 14 and 13 of FCNnMmMID1H, respectively bits 30 and 29 of FCNnMmMID0W register.
2. Be sure to align the ID value according to the given bit positions into this registers. Note that for standard ID, the ID value must be shifted to fit into FCNnMmSSID[28:18] bit positions.

(5) FCNnMmCTL - FCNn message control register m

This register is used to control the operation of the message buffer.

Access This register can be read/written in 16-bit units.

Address <FCNn_base> + 0 9038_H + m x 40_H

Initial Value 0000_H. This register is initialized by any reset.

(a) FCNnMmCTL read

15	14	13	12	11	10	9	8
0	0	FCNnMm MUCF	0	0	0	FCNnMm TCPF	0
7	6	5	4	3	2	1	0
0	FCNnMm NHMF	0	FCNnMm MOWF	FCNnMm IENF	FCNnMm DTNF	FCNnMm TRQF	FCNnMm RDYF

FCNnMmMUCF	Bit indicating that message buffer data is being updated
0	The FCN module is not updating the message buffer (no data is being received and stored).
1	The FCN module is updating the message buffer (data is being received and stored).

FCNnMmTCPF[a]	Transmission complete flag
0	Transmission failed.[b]
1	Transmission is complete.

a) FCNnMmTCPF is cleared if FCNnMmRDYF is changed or FCNnMmTRQF is set.

b) If transmission abort was requested by clearing the FCNnMmTRQF flag by the application, FCNnMmTCPF = 0 indicates a successful transmission abort.

FCNnMmNHMF	History mask flag [a]
0	Updating of the receive history list register FCNnCMRGRX and transmit history list register FCNnCMTGTX is not masked.
1	Updating of the receive history list register FCNnCMRGRX and transmit history list register FCNnCMTGTX is masked.

a) If updating is masked, the transmit and receive history lists are not updated even when reception or transmission on the corresponding message buffer finishes.

FCNnMmMOWF	Message buffer overwrite status bit
0	The message buffer is not overwritten by a newly received data or remote frame.
1	The message buffer is overwritten by a newly received data or remote frame.

Note This bit will not be set (1) if a remote frame is received and stored in a transmit message buffer with FCNnMmDTNF = 1.

FCNnMmIENF	Message buffer interrupt request enable bit
0	Receive message buffer: Valid message reception completion interrupt disabled. Transmit message buffer: Normal message transmission completion interrupt and transmit abort interrupt disabled.
1	Receive message buffer: Valid message reception completion interrupt enabled. Transmit message buffer: Normal message transmission completion interrupt enabled.

Caution Set FCNnMmIENF and FCNnMmRDYF always separately.

FCNnMmDTNF	Message buffer data update bit
0	A new data frame or remote frame has been stored in the message buffer.
1	No new data frame or remote frame has been stored in the message buffer.

Caution Do not set FCNnMmDTNF to 1 by software. Be sure to write 0 to bit 10.

FCNnMmTRQF	Message buffer transmission request bit
0	No message frame transmitting request that is pending or being transmitted is in the message buffer.
1	The message buffer is holding transmission of a message frame pending or is transmitting a message frame.

Cautions
1. Do not set FCNnMmTRQF and FCNnMmRDYF to 1 at the same time. Set FCNnMmRDYF = 1 before setting FCNnMmTRQF = 1.
2. Only set FCNnMmTRQF to 1 for transmit message buffers (not to buffers for which FCNnMmSSMT[3:0] ≠ 4'b0000 or FCNnMmSSAM = 0).

FCNnMmRDYF	Message buffer ready bit
0	The message buffer can be written by software. The FCN module cannot write to the message buffer.
1	Writing the message buffer by software is ignored (except a write access to the FCNnMmRDYF, FCNnMmTRQF, FCNnMmDTNF, and FCNnMmMOWF). The FCN module can write to the message buffer.

Cautions

1. Set FCNnMmIENF and FCNnMmRDYF always separately.
2. Do not set FCNnMmTRQF and FCNnMmRDYF to 1 at the same time. Set FCNnMmRDYF = 1 before setting FCNnMmTRQF = 1.
3. Do not clear FCNnMmRDYF to “0” during message transmission. Follow the transmission abort process about clearing FCNnMmRDYF for redefinition of the message buffer.
4. Clearing of FCNnMmRDYF may take some time, depending on activity of the CAN Controller. Repeat the clearing access, until reading of FCNnMmRDYF confirms that the bit is cleared.
5. Be sure that FCNnMmRDYF is cleared before writing to the other message buffer registers, by checking the status of FCNnMmRDYF.

(b) FCNnMmCTL write

15	14	13	12	11	10	9	8
0	FCNnMm SENH	0	0	FCNnMm SEIE	0	FCNnMm SETR	FCNnMm SERY
7	6	5	4	3	2	1	0
0	FCNnMm CLNH	0	FCNnMm CLMW	FCNnMm CLIE	FCNnMm CLDN	FCNnMm CLTR	FCNnMm CLRY

FCNnMmSENH	FCNnMmCLNH	Setting of FCNnMmNHMF bit
0	1	FCNnMmNHMF is cleared.
1	0	FCNnMmNHMF is set.
Other than above		FCNnMmNHMF is not changed.

FCNnMmCLMW	Setting of FCNnMmMOWF bit
0	FCNnMmMOWF is not changed.
1	FCNnMmMOWF is cleared.

FCNnMmSEIE	FCNnMmCLIE	Setting of FCNnMmIENF bit
0	1	FCNnMmIENF is cleared.
1	0	FCNnMmIENF is set.
Other than above		FCNnMmIENF is not changed.

FCNnMmCLDN	Setting of FCNnMmDTNF bit
1	FCNnMmDTNF is cleared.
0	FCNnMmDTNF is set.

Note If FCNnMmDTNF is cleared at the end of ID field reception, the frames being received will be saved into the corresponding message buffer.

FCNnMmSETR	FCNnMmCLTR	Setting of FCNnMmTRQF bit
0	1	FCNnMmTRQF is cleared.
1	0	FCNnMmTRQF is set.
Other than above		FCNnMmTRQF is not changed.

FCNnMmSERY	FCNnMmCLRY	Setting of FCNnMmRDYF bit
0	1	FCNnMmRDYF is cleared.
1	0	FCNnMmRDYF is set.
Other than above		FCNnMmRDYF is not changed.

24.6 CAN Controller Initialization

24.6.1 Initialization of FCN module

Before FCN module operation is enabled, the FCN module system clock needs to be determined by setting FCNnGMCSPRE.FCNnGMCSPRSC[3:0] by software. Do not change the setting of the FCN module system clock after FCN module operation is enabled.

The FCN module is enabled by setting FCNnGMCLCTL.FCNnGMCLPWOM.

For the procedure of initializing the FCN module, refer to *24.14 “Operation of the CAN Controller” on page 1654* .

24.6.2 Initialization of message buffer

After the FCN module is enabled, the message buffers might contain an undefined value (except after a software reset). A minimum initialization for all the message buffers, even for those not used in the application, is necessary before switching the FCN module from the initialization mode to one of the operation modes.

- Clear FCNnMmRDYF, FCNnMmTRQF, and FCNnMmDTNF of the FCNnMmCTL registers to 0.
- Clear all FCNnMmSTRB.FCNnMmSSAM to 0.

24.6.3 Redefinition of message buffer

Redefining a message buffer means changing the ID and control information of the message buffer while a message is being received or transmitted, without affecting other transmission/reception operations.

(1) To redefine message buffer in initialization mode

Place the FCN module in the initialization mode once and then change the ID and control information of the message buffer in the initialization mode. After changing the ID and control information, set the FCN module to an operation mode.

(2) To redefine message buffer during reception

Perform redefinition as shown in *Figure 24-17 “Message buffer redefinition during reception” on page 1658* .

(3) To redefine message buffer during transmission

To rewrite the contents of a transmit message buffer to which a transmission request has been set, perform transmission abort processing (see *(1) “Transmission abort process except for in normal operation mode with automatic block transmission (ABT)”* and *(2) “Transmission abort process for ABT transmission in normal operation mode with automatic block transmission (ABT)” on page 1631* for details). Confirm that transmission has been aborted or completed, and then redefine the message buffer. After redefining the transmit message buffer, set a transmission request using the procedure described below. When setting a transmission request to a message buffer that

has been redefined without aborting the transmission in progress, however, the 1-bit wait time is not necessary.

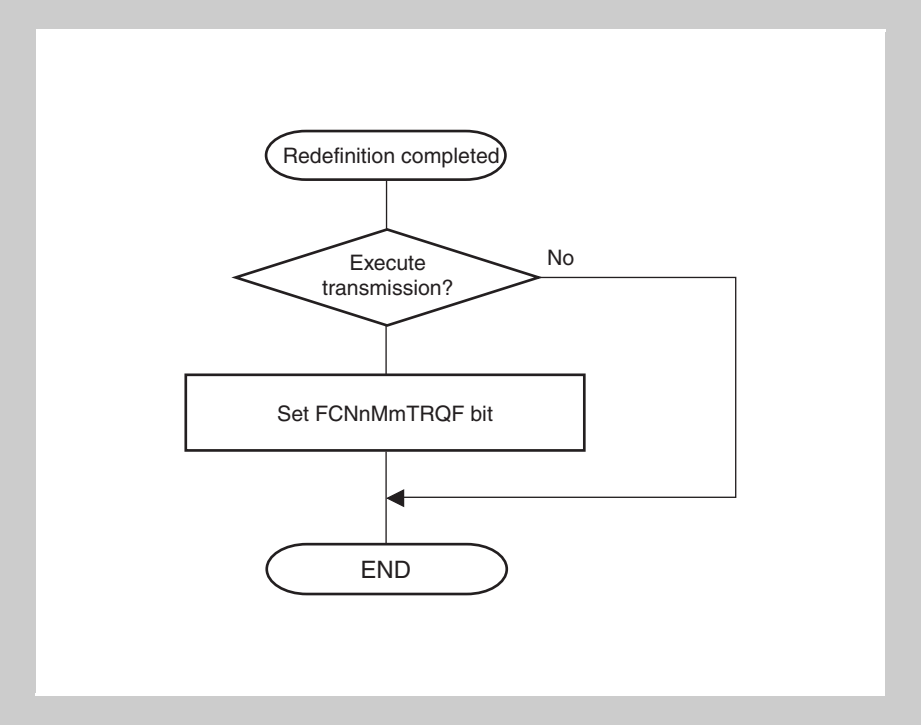


Figure 24-4 Setting transmission request (FCNnMmCTL.FCNnMmTRQF) to transmit message buffer after redefinition

Cautions

1. When a message is received, reception filtering is performed in accordance with the ID and mask set to each receive message buffer. If the procedure in *Figure 24-17 “Message buffer redefinition during reception” on page 1658* is not observed, the contents of the message buffer after it has been redefined may contradict the result of reception (result of reception filtering). If this happens, check that the ID and IDE received first and stored in the message buffer following redefinition are those stored after the message buffer has been redefined. If no ID and IDE are stored after redefinition, redefine the message buffer again.
2. When a message is transmitted, the transmission priority is checked in accordance with the ID, IDE, and FCNnMmSTRB.FCNnMmSSRT set to each transmit message buffer to which a transmission request was set. The transmit message buffer having the highest priority is selected for transmission. If the procedure in *Figure 24-4 “Setting transmission request (FCNnMmCTL.FCNnMmTRQF) to transmit message buffer after redefinition” on page 1614* is not observed, a message with an ID not having the highest priority may be transmitted after redefinition.

24.6.4 Transition from initialization mode to operation mode

The FCN module can be switched to the following operation modes.

- Normal operation mode
- Normal operation mode with ABT
- Receive-only mode
- Single-shot mode
- Self-test mode

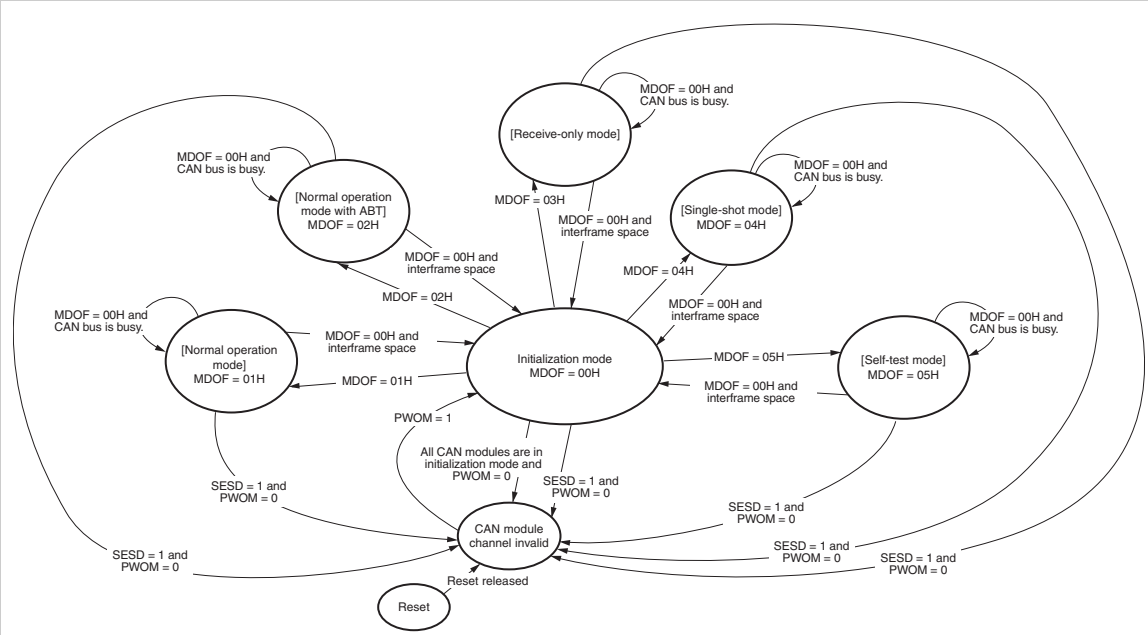


Figure 24-5 Transition to operation modes

Note In the figure above following abbreviations are used:

- MDOF = FCNnCMCLCTL.FCNnCMCLMDOF[2:0]
- PWOM = FCNnGMCLCTL.FCNnGMCLPWOM
- SESD = FCNnGMCLCTL.FCNnGMCLSESD

The transition from the initialization mode to an operation mode is controlled by the bit string FCNnCM.FCNnCMCLMDOF[2:0].

Changing from one operation mode into another requires shifting to the initialization mode in between. Do not change one operation mode to another directly; otherwise the operation will not be guaranteed.

Requests for transition from an operation mode to the initialization mode are held pending when the CAN bus is not in the interframe space (i.e., frame reception or transmission is in progress), and the FCN module enters the initialization mode at the first bit in the interframe space (the values of the FCNnCMCLCTL.FCNnCMCLMDOF[2:0] are changed to $000_{B)}$. After issuing a request to change the mode to the initialization mode, read FCNnCMCLCTL.FCNnCMCLMDOF[2:0] until their value becomes 000_B to confirm that the module has entered the initialization mode (see *Figure 24-14 “Re-initialization without using the software reset function” on page 1655*).

24.7 Message Reception

24.7.1 Message reception

In all the operation modes, the complete message buffer area is analyzed to find a suitable buffer to store a newly received message. All message buffers satisfying the following conditions are included in that evaluation (RX-search process).

- Used as a message buffer
  (FCNnMmSTRB.FCNnMmSSAM = 1.)
- Set as a receive message buffer
  (FCNnMmSTRB.FCNnMmSSMT[3:0] = 0001_B to 1001_B)
- Ready for reception
  (FCNnMmCTL.FCNnMmRDYF = 1.)

When two or more message buffers of the FCN module are found to be able to receive a message, the message is stored according to the priority explained below. The message is always stored in the message buffer with the highest priority, not in a message buffer with a low priority. For example, when an unmasked receive message buffer and a receive message buffer linked to mask 1 have the same ID, the received message is not stored in the message buffer linked to mask 1, even if that message buffer has not received a message and a message has already been received in the unmasked receive message buffer. In other words, when a condition has been set in two or more message buffers with different priorities, the message buffer with the highest priority always stores the message; the message is not stored in message buffers with a lower priority. This also applies when the message buffer with the highest priority is unable to store a message (i.e., when FCNnMmCTL.FCNnMmDTNF = 1 indicating that a message has already been received, but rewriting is disabled because FCNnMmSTRB.FCNnMmSSOW = 0). In this case, the message is not actually stored in the candidate message buffer with the highest priority, but neither is it stored in a message buffer with a lower priority.

Table 24-16 MBRB priorities

Priority	Storing condition if same ID is set	
1 (high)	Unmasked message buffer	FCNnMmDTNF = 0
		FCNnMmDTNF = 1 and FCNnMmSSOW = 1
2	Message buffer linked to mask 1	FCNnMmDTNF = 0
		FCNnMmDTNF = 1 and FCNnMmSSOW = 1
3	Message buffer linked to mask 2	FCNnMmDTNF = 0
		FCNnMmDTNF = 1 and FCNnMmSSOW = 1
...	...	
9 (low)	Message buffer linked to mask 8	FCNnMmDTNF = 0
		FCNnMmDTNF = 1 and FCNnMmSSOWt = 1

24.7.2 Receive data read

To keep data consistency when reading FCN message buffers, perform the data reading according to *Figure 24-31 “Reception via interrupt (using FCNnCMLISTR register)” on page 1673* to *Figure 24-34 “Reception via software polling” on page 1677* .

During message reception, the FCN module sets FCNnMmCTL.FCNnMmDTNF two times: at the beginning of the storage process of data to the message buffer, and again at the end of this storage process. During this storage process, FCNnMmCTL.FCNnMmMUCF of the message buffer is set (refer to *Figure 24-6 “Reception timing”*).

The receive history list is also updated just before the storage process. In addition, during storage process (FCNnMmCTL.FCNnMmMUCF = 1), FCNnMmCTL.FCNnMmRDYF of the message buffer is locked to avoid any coincidental data write by CPU. Note that the storage process may be disturbed (delayed) when the CPU accesses the message buffer.

Caution To reliably store a message in a message buffer, the DN bit for that buffer must be cleared before message search processing starts (after a frame ID is output on the bus). This might occur as early as the 15th CAN bit after EOF of the previous frame. To reliably receive successively sent CAN frames on the bus, using multiple message buffers for frame reception is recommended.

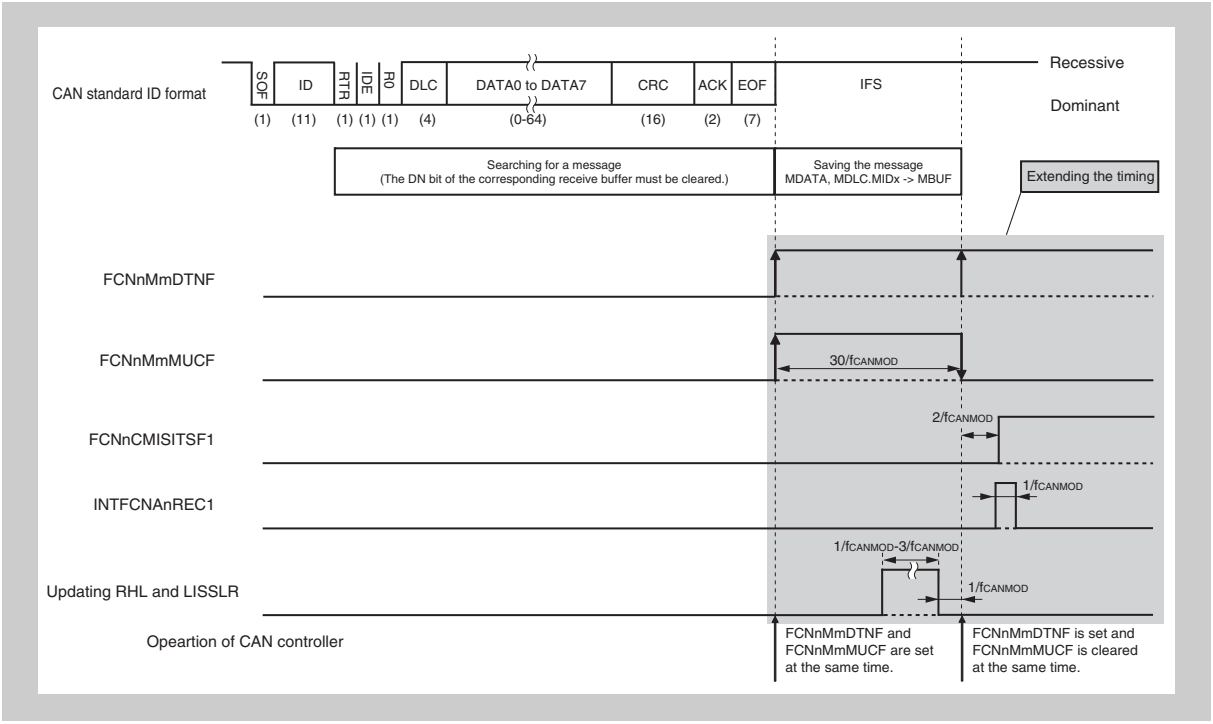


Figure 24-6 Reception timing

24.7.3 Receive history list function

The receive history list (RHL) function records in the receive history list the number of the receive message buffer in which each data frame or remote frame was received and stored. The RHL consists of storage elements equivalent to up to 47 messages (on 64 message buffer FCN) or up to 95 messages (on 128 message buffer FCN), the last in-message pointer FCNnCMLISSLR[7:0] with the corresponding FCNnCMLISTR register and the receive history list get pointer FCNnCMRGSSPT with the corresponding FCNnCMRGRX register.

The RHL is undefined immediately after the transition of the FCN module from the initialization mode to one of the operation modes.

The FCNnCMLISTR register holds the contents of the RHL element indicated by the value of the FCNnCMLISTR.FCNnCMLISSLR[7:0] pointer minus 1. (See *Figure 24-7 "Receive history list"*) It is consequently possible to check the number of the message buffer that received and stored the last data frame or remote frame by reading the FCNnCMLISTR register. The FCNnCMLISSLR[7:0] pointer is utilized as a write pointer that indicates to what part of the RHL a message buffer number is recorded. Any time a data frame or remote frame is received and stored, the corresponding message buffer number is recorded to the RHL element indicated by the FCNnCMLISSLR[7:0] pointer. Each time recording to the RHL has been completed, the FCNnCMLISSLR[7:0] pointer is automatically incremented. In this way, the number of the message buffer that has received and stored a frame will be recorded chronologically.

For message buffers, where the flag FCNnMmCTL.FCNnMmNHMF is set, no entry in the history lists is recorded.

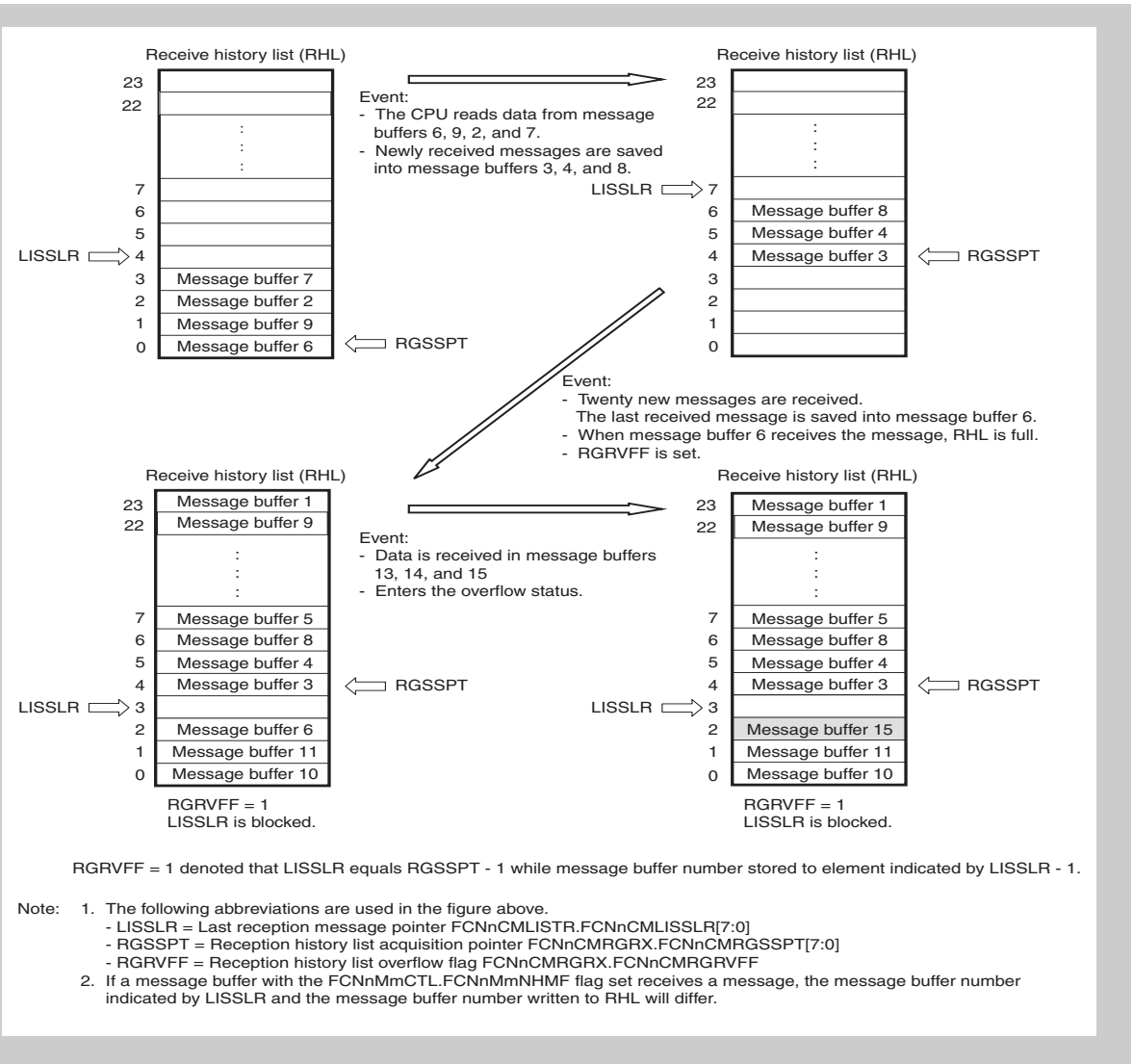
The FCNnCMRGRX.FCNnCMRGSSPT pointer is utilized as a read pointer that reads a recorded message buffer number from the RHL. This pointer indicates the first RHL element that the CPU has not read yet. By reading the FCNnCMRGRX register by software, the number of a message buffer that has received and stored a data frame or remote frame can be read. Each time a message buffer number is read from the FCNnCMRGRX register, the FCNnCMRGSSPT pointer is automatically incremented.

If the value of the FCNnCMRGRX.FCNnCMRGSSPT pointer matches the value of the FCNnCMLISTR.FCNnCMLISSLR[7:0] pointer, FCNnCMRGRX.FCNnCMRGSSPM (receive history list pointer match) is set to 1. This indicates that no message buffer number that has not been read remains in the RHL. If a new message buffer number is recorded, the FCNnCMLISSLR[7:0] pointer is incremented and because its value no longer matches the value of the FCNnCMRGSSPT pointer, FCNnCMRGSSPM is cleared. In other words, the numbers of the unread message buffers exist in the RHL.

If the FCNnCMLISTR.FCNnCMLISSLR[7:0] pointer is incremented and matches the value of the FCNnCMRGRX.FCNnCMRGSSPT pointer minus 1, FCNnCMRGRX.FCNnCMRGRVFF (receive history list overflow) is set to 1. This indicates that the RHL is full of numbers of message buffers that have not been read. When further message reception and storing occur, the last recorded message buffer number is overwritten by the number of the message buffer that received and stored the newly received message. In this case, after FCNnCMRGRVFF has been set (1), the recorded message buffer numbers in the RHL do not completely reflect the chronological order. However messages itself are not lost and can be located by CPU search in message buffer memory with the help of FCNnMmCTL.FCNnMmDTNF, or by reading the global registers FCNnDNBMRX[3:0].

Caution If the history list is in the overflow condition (FCNnCMRGRX.FCNnCMRGRVFF is set), reading the history list contents is still possible, until the history list is empty (indicated by FCNnCMRGRX.FCNnCMRGSSPM flag set). Nevertheless, the history list remains in the overflow condition, until FCNnCMRGRVFF is cleared by software. If FCNnCMRGRVFF is not cleared, the FCNnCMRGSSPM flag will also not be updated (cleared) upon a message storage of newly received frame. This may lead to the situation, that FCNnCMRGSSPM indicates an empty history list, although a reception has taken place, while the history list is in the overflow state (FCNnCMRGRVFF and FCNnCMRGSSPM are set).

As long as the RHL still has free entries, the sequence of occurrence is maintained. If more receptions occur without reading the RHL by the host processor, complete sequence of receptions can not be recovered.



RGRVFF = 1 denoted that LISSLR equals RGSSPT - 1 while message buffer number stored to element indicated by LISSLR - 1.

Note: 1. The following abbreviations are used in the figure above.
- LISSLR = Last reception message pointer FCNnCMLISTR.FCNnCMLISSLR[7:0]
- RGSSPT = Reception history list acquisition pointer FCNnCMRGRX.FCNnCMRGSSPT[7:0]
- RGRVFF = Reception history list overflow flag FCNnCMRGRX.FCNnCMRGRVFF

2. If a message buffer with the FCNnMmCTL.FCNnMmNHMF flag set receives a message, the message buffer number indicated by LISSLR and the message buffer number written to RHL will differ.

Figure 24-7 Receive history list

24.7.4 Mask function

For any message buffer, which is used for reception, the assignment to one of eight global reception masks (or no mask) can be selected.

By using the mask function, the message ID comparison can be reduced by masked bits, herewith allowing the reception of several different IDs into one buffer.

While the mask function is in effect, an identifier bit that is defined to be 1 by a mask in the received message is not compared with the corresponding identifier bit in the message buffer.

However, this comparison is performed for any bit whose value is defined as 0 by the mask.

For example, let us assume that all messages that have a standard-format ID, in which bits ID27 to ID25 are 0 and bits ID24 and ID22 are 1, are to be stored in message buffer 14. The procedure for this example is shown below.

(1) Identifier to be stored in message buffer

ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21	ID20	ID19	ID18
x	0	0	0	1	x	1	x	x	x	x

(2) Identifier to be configured in message buffer 14 (example) (using FCN1M014MID0W register)

ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21	ID20	ID19	ID18
x	0	0	0	1	x	1	x	x	x	x
ID17	ID16	ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8	ID7
x	x	x	x	x	x	x	x	x	x	x
ID6	ID5	ID4	ID3	ID2	ID1	ID0				
x	x	x	x	x	x	x				

Notes
1. ID with the ID27 to ID25 bits cleared to 0 and the ID24 and ID22 bits set to 1 is registered (initialized) to message buffer 14.
2. Message buffer 14 is set as a standard format identifier that is linked to mask 1 (FCN1M014STRB.FCN1M014SSMT[3:0] = 0010_B).

(3) Mask setting for FCN module 1 (mask 1) (example)
(using FCAN1 module mask 1 register FCN1CMMKCTL01W)

FNCnCMMKSSID[..]

ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21	ID20	ID19	ID18
x	0	0	0	1	x	1	x	x	x	x

ID17	ID16	ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8	ID7
x	x	x	x	x	x	x	x	x	x	x

ID6	ID5	ID4	ID3	ID2	ID1	ID0
x	x	x	x	x	x	x

1: Not compared (masked)
0: Compared

FCN1CMMKSSID[27:24] and FCN1CMMKSSID[21] are cleared to 0, and FCN1CMMKSSID[28], FCN1CMMKSSID[23], and FCN1CMMKSSID[21:0] are set to 1.

24.7.5 Multi buffer receive block function

The multi buffer receive block (MBRB) function is used to store a block of data in two or more message buffers sequentially with no CPU interaction, by setting the same ID to two or more message buffers with the same message buffer type. These message buffers can be allocated anywhere in the message buffer memory, they do not even have to follow each other adjacently.

Suppose, for example, the same message buffer type is set to 10 message buffers, message buffers 10 to 19, and the same ID is set to each message buffer. If the first message whose ID matches an ID of the message buffers is received, it is stored in message buffer 10. At this point, FCNnMmCTL.FCNnMmDTNF of message buffer 10 is set, prohibiting overwriting the message buffer when subsequent messages are received.

When the next message with a matching ID is received, it is received and stored in message buffer 11. Each time a message with a matching ID is received, it is sequentially (in the ascending order) stored in message buffers 12, 13, and so on. Even when a data block consisting of multiple messages is received, the messages can be stored and received without overwriting the previously received matching-ID data.

Whether a data block has been received and stored can be checked by setting FCNnMmCTL.FCNnMmIENF of each message buffer. For example, if a data block consists of k messages, k message buffers are initialized for reception of the data block. FCNnMmIENF in message buffers 0 to (k-2) is cleared to 0 (interrupts disabled), and FCNnMmIENF in message buffer k-1 is set to 1 (interrupts enabled). In this case, a reception completion interrupt occurs when a message has been received and stored in message buffer k-1, indicating that MBRB has become full. Alternatively, by clearing FCNnMmIENF of message buffers 0 to (k-3) and setting FCNnMmIENF of message buffer k-2, a warning that MBRB is about to overflow can be issued.

The basic conditions of storing receive data in each message buffer for the MBRB are the same as the conditions of storing data in a single message buffer.

Cautions

1. MBRB can be configured for each of the same message buffer types. Therefore, even if a message buffer of another MBRB whose ID matches but whose message buffer type is different has a vacancy, the received message is not stored in that message buffer, but instead discarded.
2. MBRB does not have a ring buffer structure. Therefore, after a message is stored in the message buffer having the highest number in the MBRB configuration, a newly received message will not be stored in the message buffer having the lowest message buffer number.
3. MBRB operates based on the reception and storage conditions; there are no settings dedicated to MBRB, such as function enable bits. By setting the same message buffer type and ID to two or more message buffers, MBRB is automatically configured.
4. With MBRB, “matching ID” means “matching ID after mask”. Even if the ID set to each message buffer is not the same, if the ID that is masked by the mask register matches, it is considered a matching ID and the buffer that has this ID is treated as the storage destination of a message.
5. The priority between MBRBs is mentioned in the table *Table 24-16 “MBRB priorities”*.

24.7.6 Remote frame reception

In all the operation modes, when a remote frame is received, the message buffer that is to store the remote frame is searched from all the message buffers satisfying the following conditions (1 and 2, condition 1 has priority on reception acceptance). If condition 1 is not fulfilled, the remaining message buffers are scanned, whether condition 2 could be fulfilled.

- Condition 1:
  Set as a transmit message buffer
  (FCNnMmSTRB.FCNnMmSSMT[3:0] = 0000_B)
  - Used as a message buffer
    (FCNnMmSTRB.FCNnMmSSAM = 1.)
  - Ready for reception
    (FCNnMmCTL.FCNnMmRDYF = 1.)
  - Set to data frame message type
    (FCNnMmSTRB.FCNnMmSSRT = 0.)
  - Transmission request is not set.
    (FCNnMmCTL.FCNnMmTRQF = 0.)
- Condition 2:
  Set as a receive message buffer
  (FCNnMmSTRB.FCNnMmSSMT[3:0] = 0001_B ... 1001_B)
  - Used as a message buffer
    (FCNnMmSTRB.FCNnMmSSAM = 1.)
  - Ready for reception
    (FCNnMmCTL.FCNnMmRDYF = 1.)
  - Set to remote frame message type
    (FCNnMmSTRB.FCNnMmSSRT = 1.)
  - Buffer is ready to store a message
    (FCNnMmCTL.FCNnMmDTNF = 0, or FCNnMmSTRB.FCNnMmSSOW = 1 with FCNnMmCTL.FCNnMmDTNF = 1).

Upon acceptance of a remote frame, the following actions are executed if the ID of the received remote frame matches the ID of a message buffer that satisfies the above conditions.

- The FCNnMmDTLG[3:0] bit string in the FCNnMmDTLGB register store the received DLC value.
- When received in a transmit message buffer, registers FCNnMmDAT0B to FCNnMmDAT7B in the data area will not be updated (the data from before reception is stored).
- FCNnMmCTL.FCNnMmDTNF is set to 1.
- FCNnCMISCTL.FCNnCMISITSF1 is set to 1 (if FCNnMmCTL.FCNnMmIENF of the message buffer that receives and stores the frame is set to 1).
- The receive completion interrupt (INTCnREC) is output (if FCNnMmCTL. FCNnMmIENF of the message buffer that receives and stores the frame is set to 1 and if FCNnCMIECTL.FCNnCMIEINTF1 is set to 1).
- The message buffer number is recorded in the receive history list, if the flag FCNnMmCTL.FCNnMmNHMF is not set.

Caution When a transmit message buffer is found for receiving and storing a remote frame, overwrite control by FCNnMmSTRB.FCNnMmSSOW of the message buffer and FCNnMmCTL.FCNnMmDTNF are not checked. The setting of FCNnMmSSOW is ignored, and FCNnMmDTNF is set in any case.

Notes

1. If more than one transmit message buffer has the same ID and the ID of the received remote frame matches that ID, the remote frame is stored in the transmit message buffer with the lowest message buffer number.
2. If transmit and receive message buffers are found, which could receive a remote frame matching with its ID, either masked or unmasked, the remote frame is stored in the transmit message buffer.
3. If several receive message buffers would match for reception for a remote frame, the reception priority is identical as for a data frame.
4. If a receive message buffer is found to match for a remote frame reception, and selected for storage, but this receive message buffer does not allow the storage, because FCNnMmDTNF is set, and FCNnMmSSOW is not set, the remote frame is not stored at all.

24.8 Message Transmission

24.8.1 Message transmission

A message buffer with its FCNnMmCTL.FCNnMmTRQF bit set to 1 participates in the search for the most high-prioritized message when the following conditions are fulfilled. This behavior is valid for all operational modes.

- Used as a message buffer (FCNnMmSTRB.FCNnMmSSAM = 1)
- Set as a transmit message buffer (FCNnMmSTRB.FCNnMmSSMT[3:0] = 0000_B)
- Ready for transmission (FCNnMmCTL.FCNnMmRDYF = 1)

The CAN system is a multi-master communication system. In a system like this, the priority of message transmission is determined based on message identifiers (IDs). To facilitate transmission processing by software when there are several messages awaiting transmission, the FCN module uses hardware to check the ID of the message with the highest priority and automatically identifies that message. This eliminates the need for software-based priority control.

Transmission priority is controlled by the identifier (ID).

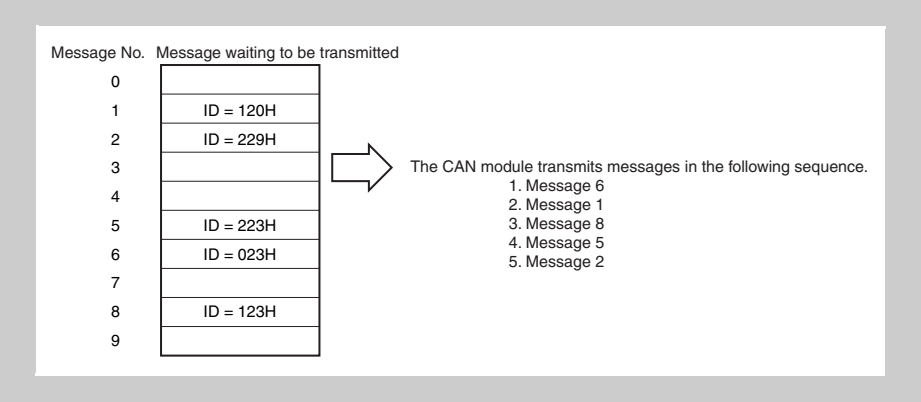


Figure 24-8 Message processing example

After the transmit message search, the transmit message with the highest priority of the transmit message buffers that have a pending transmission request (message buffers with the FCNnMmCTL.FCNnMmTRQF bit set to 1 in advance) is transmitted.

If a new transmission request is set, the transmit message buffer with the new transmission request is compared with the transmit message buffer with a pending transmission request. If the new transmission request has a higher priority, it is transmitted, unless transmission of a message with a low priority has already started. If transmission of a message with a low priority has already started, however, the new transmission request is transmitted later. To solve this priority inversion effect, the software can perform a transmission abort request for the lower priority message. The highest priority is determined according to the following rules.

Priority	Conditions	Description
1 (high)	Value of first 11 bits of ID [ID28 to ID18]:	The message frame with the lowest value represented by the first 11 bits of the ID is transmitted first. If the value of an 11-bit standard ID is equal to or smaller than the first 11 bits of a 29-bit extended ID, the 11-bit standard ID has a higher priority than a message frame with a 29-bit extended ID.
2	Frame type	A data frame with an 11-bit standard ID (FCNnMmSTRB.FCNnMmSSRT cleared to 0) has a higher priority than a remote frame with a standard ID and a message frame with an extended ID.
3	ID type	A message frame with a standard ID (message buffer identifier register FCNnMmMID... bit FCNnMmSSIE is cleared to 0) has a higher priority than a message frame with an extended ID.
4	Value of lower 18 bits of ID [ID17 to ID0]:	If one or more transmission-pending extended ID message frame has equal values in the first 11 bits of the ID and the same frame type (equal FCNnMmSTRB.FCNnMmSSRT bit values), the message frame with the lowest value in the lower 18 bits of its extended ID is transmitted first.
5 (low)	Message buffer number	If two or more message buffers request transmission of message frames with the same ID, the message from the message buffer with the lowest message buffer number is transmitted first.

Notes 1. If the automatic block transmission request bit FCNnGMABCTL.FCNnGMABABTT is set to 1 in the normal operation mode with ABT, FCNnMmCTL.FCNnMmTRQF is set to 1 only for one message buffer in the ABT message buffer group.

If ABT mode is triggered by setting FCNnGMABCTL.FCNnGMABSEAT = 1, then one of the FCNnMmCTL.FCNnMmTRQF in the ABT area (64 message buffers FCN:0 to 15, and 128 message buffers FCN:0 to 31) will be set to 1. Beyond this transmit request, the application can request transmissions (set FCNnMmTRQF to 1) for other TX-message buffers that do not belong to the ABT area. In that case an interval arbitration process (TX-search) evaluates all TX-message buffers with FCNnMmTRQF set to 1 and chooses the message buffer that contains the highest prioritized identifier for the next transmission. If there are 2 or more identifiers that have the highest priority (i.e. identical identifiers), the message located at the lowest message buffer number is transmitted at first.

Upon successful transmission of a message frame, the following operations are performed.

- The FCNnMmCTL.FCNnMmTRQF flag of the corresponding transmit message buffer is automatically cleared to 0.
- The transmission completion status bit FCNnCMISCTL.FCNnCMISITSF0 is set to 1 (if the interrupt enable bit FCNnMmIENF of the corresponding transmit message buffer is set to 1).
- An interrupt request signal INTCnTRX is output (if FCNnCMIECTL.FCNnCMIEINTF0 is set to 1 and if the interrupt enable bit FCNnMmIENF of the corresponding transmit message buffer is set to 1).

2. When changing the contents of a transmit buffer, the FCNnMmCTL.FCNnMmRDYF flag of this buffer must be cleared before updating the buffer contents. As during internal transfer actions, the FCNnMmRDYF flag may be locked temporarily, the status of FCNnMmRDYF must be checked by software, after changing it.

24.8.2 Transmit history list function

The transmit history list (THL) function records in the transmit history list the number of the transmit message buffer from which data or remote frames have been were sent. The THL consists of storage elements equivalent to up to 15 messages (on 64 message buffer FCN) or up to 31 messages (on 128 message buffer FCN), the last out-message pointer FCNnCMLOSTR[7:0] with the corresponding FCNnCMLOSTR register, and the transmit history list get pointer FCNnCMTGSSPT[7:0] with the corresponding FCNnCMTGTX register.

The THL is undefined immediately after the transition of the FCN module from the initialization mode to one of the operation modes.

The FCNnCMLOSTR register holds the contents of the THL element indicated by the value of the FCNnCMLOSTR.FCNnCMLOSTR[7:0] pointer minus 1. By reading the FCNnCMLOSTR register, therefore, the number of the message buffer that transmitted a data frame or remote frame first can be checked. The FCNnCMLOSTR[7:0] pointer is utilized as a write pointer that indicates to what part of the THL a message buffer number is recorded. Any time a data frame or remote frame is transmitted, the corresponding message buffer number is recorded to the THL element indicated by the FCNnCMLOSTR[7:0] pointer. Each time recording to the THL has been completed, the FCNnCMLOSTR[7:0] pointer is automatically incremented. In this way, the number of the message buffer that has received and stored a frame will be recorded chronologically.

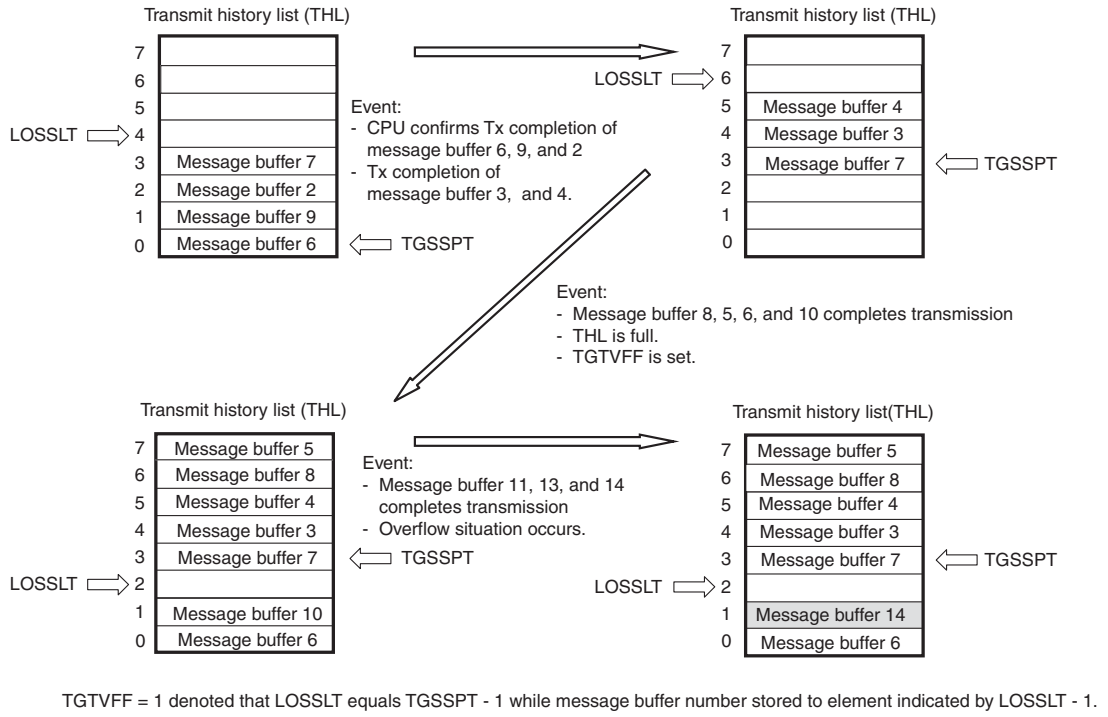
For message buffers, where the flag FCNnMmCTL.FCNnMmNHMF is set, no entry in the history lists is recorded.

The FCNnCMTGTX.FCNnCMTGSSPT[7:0] pointer is utilized as a read pointer that reads a recorded message buffer number from the THL. This pointer indicates the first THL element that the CPU has not yet read. By reading the FCNnCMTGTX register by software, the number of a message buffer that has completed transmission can be read. Each time a message buffer number is read from the FCNnCMTGTX register, the FCNnCMTGSSPT[7:0] pointer is automatically incremented.

If the value of the FCNnCMTGTX.FCNnCMTGSSPT[7:0] pointer matches the value of the FCNnCMLOSTR.FCNnCMLOSTR[7:0] pointer, FCNnCMTGTX.FCNnCMTGSSPM (transmit history list pointer match) is set to 1. This indicates that no message buffer numbers that have not been read remain in the THL. If a new message buffer number is recorded, the FCNnCMLOSTR[7:0] pointer is incremented and because its value no longer matches the value of the FCNnCMTGSSPT[7:0] pointer, FCNnCMTGSSPM is cleared. In other words, the numbers of the unread message buffers exist in the THL.

If the FCNnCMLOSTR.FCNnCMLOSTR[7:0] pointer is incremented and matches the value of the FCNnCMTGTX.FCNnCMTGSSPT[7:0] pointer minus 1, FCNnCMTGTX.FCNnCMTGTVFF (transmit history list overflow) is set to 1. This indicates that the THL is full of message buffer numbers that have not been read. If a new message is received and stored, the message buffer number recorded last is overwritten by the message buffer number that transmitted its message afterwards. In this case, after FCNnCMTGTVFF has been set (1), therefore, the recorded message buffer numbers in the THL do not completely reflect the chronological order. Even in this case, however, the CPU can identify the number of the message buffer that completed reception by searching all reception buffers (the CPU does this before resetting transmission). Regardless of the FCNnCMTGTX.FCNnCMTVFF setting, 14 (64 message buffers) or 30 (128 message buffer) transmit message buffer numbers are stored in THL.

Caution If the history list is in the overflow condition (FCNnCMTGTX.FCNnCMTGTVFF is set), reading the history list contents is still possible, until the history list is empty (indicated by FCNnCMTGTX.FCNnCMTGSSPM flag set). Nevertheless, the history list remains in the overflow condition, until FCNnCMTGTVFF is cleared by software. If FCNnCMTGTVFF is not cleared, the FCNnCMTGTX.FCNnCMTGSSPM flag will also not be updated (cleared) upon successful transmission of a new message. This may lead to the situation, that FCNnCMTGSSPM indicates an empty history list, although a successful transmission has taken place, while the history list is in the overflow state (FCNnCMTGTVFF and FCNnCMTGSSPM are set).



Note: 1. The following abbreviations are used in the figure above.
- LOSSLT = Last transmission message pointer FCNnCMLOSTR.FCNnCMLOSSLT[7:0]
- TGSSPT = Transmission history list acquisition pointer FCNnCMTGTX.FCNnCMTGSSPT[7:0]
- TGTVFF = Transmission history list overflow flag FCNnCMTGTX.FCNnCMTGTVFF

2. If a message buffer with the FCNnMmCTL.FCNnMmNHMF flag set transmits a message, the message buffer number indicated by LOSSLT and the message buffer number written to THL will differ.

Figure 24-9 Transmit history list

24.8.3 Automatic block transmission (ABT)

The automatic block transmission (ABT) function is used to transmit two or more data frames successively with no CPU interaction. The maximum number of transmit message buffers assigned to the ABT function is 16 (for 64 message buffer FCN) or 32 (for 128 message buffer FCN), always located in the lowest message buffers.

By setting FCNnCMCLCTL.FCNnCMCLMDOF[2:0] to 010_B, “normal operation mode with automatic block transmission function” (hereafter referred to as ABT mode) can be selected.

To issue an ABT transmission request, define the message buffers by software first. Set FCNnMmSTRB.FCNnMmSSAM = 1 in all the message buffers used for ABT, and define all the buffers as transmit message buffers by setting the FCNnMmSTRB.FCNnMmSSMT[3:0] bits to 0000_B. Be sure to set the same ID for the message buffers for ABT even when that ID is being used for all the message buffers. To use two or more IDs, set the ID of each message buffer by using the FCNnMmMID0H and FCNnMmMID1H or FCNnMmMID0W registers. Set the FCN message data bytes registers before issuing a transmission request for the ABT function.

After initialization of message buffers for ABT is finished, FCNnMmCTL.FCNnMmRDYF needs to be set to 1. In the ABT mode, FCNnMmCTL.FCNnMmTRQF does not have to be manipulated by software.

After the data for the ABT message buffers has been prepared, set FCNnGMABCTL.FCNnGMABSEAT = 1. Automatic block transmission is then started. When ABT is started, FCNnMmCTL.FCNnMmTRQF in the first message buffer (message buffer 0) is automatically set to 1. After transmission of the data of message buffer 0 is finished, the FCNnMmTRQF of the next message buffer, message buffer 1, is set automatically. In this way, transmission is executed successively.

A delay time can be inserted by program in the interval in which the transmission request FCNnMmCTL.FCNnMmTRQF is automatically set while successive transmission is being executed. The delay time to be inserted is defined by the FCNnGMADCTL register. The unit of the delay time is DBT (data bit time). DBT depends on the setting of the FCNnCMBRPRS and FCNnCMBTCTL registers.

Among transmit objects within the ABT-area, the priority of the transmission ID is not evaluated. Messages are sent by order of message number, starting with message buffer 0. When the transmission of the data frame from the last message buffer is complete, FCNnGMABCTL.FCNnGMABABTT is automatically cleared to 0, and ABT operation completes.

If there is an ABT message buffer for which FCNnMmCTL.FCNnMmRDYF is cleared during ABT, no data frame is transmitted from that buffer, ABT is stopped, and FCNnGMABCTL.FCNnGMABABTT is cleared. After that, transmission can be resumed from the message buffer where ABT stopped, by setting FCNnMmRDYF and FCNnGMABABTT to 1 by software. To not resume transmission from the message buffer where ABT stopped, the internal ABT engine can be reset by setting the FCNnGMABCTL.FCNnGMABCLRF bit to 1 while ABT mode is stopped and FCNnGMABABTT is cleared to 0. In this case, transmission is started from message buffer 0 if FCNnGMABCTL.FCNnGMABSEAC is cleared to 0 and then FFCNnGMABABTT is set to 1.

An interrupt can be used to check if data frames have been transmitted from all the message buffers for ABT. To do so, FCNnMmCTL.FCNnMmIENF of each message buffer except the last message buffer needs to be cleared (0).

If a transmit message buffer other than those used by the ABT function is assigned to a transmit message buffer, the message to be transmitted next is determined by the priority of the transmission ID of the ABT message buffer whose transmission is currently held pending and the transmission ID of the message buffers other than those used by the ABT function.

Transmission of a data frame from an ABT message buffer is not recorded in the transmit history list (THL).

Cautions

1. Set FCNnGMABCTL.FCNnGMABSEAC = 1 while FCNnGMABCTL.FCNnGMABABTT is cleared to 0 in order to resume ABT operation at buffer No. 0. If FCNnGMABSEAC is set to 1 while FCNnGMABABTT is set to 1, the subsequent operation is not guaranteed.
2. If the automatic block transmission engine is cleared by setting FCNnGMABCTL.FCNnGMABSEAC = 1, FCNnGMABSEAC is automatically cleared immediately after the processing of the clearing request is completed.
3. Do not trigger automatic block transmission in the initialization mode. If FCNnGMABCTL.FCNnGMABSEAT is set in the initialization mode, the proper operation is not guaranteed after the mode is changed from the initialization mode to the ABT mode.
4. Do not set FCNnMmCTL.FCNnMmTRQF of the ABT message buffers to 1 by software in the normal operation mode with ABT. Otherwise, the operation is not guaranteed.
5. The FCNnGMADCTL register is used to set the delay time that is inserted in the period from completion of the preceding ABT message to setting of FCNnMmCTL.FCNnMmTRQF for the next ABT message when the transmission requests are set in the order of message numbers for each message for ABT that is successively transmitted in the ABT mode. The timing at which the messages are actually transmitted onto the CAN bus varies depending on the status of transmission from other stations and the status of the setting of the transmission request for messages other than the ABT messages.
6. If a transmission request is made for a message other than an ABT message and if no delay time is inserted in the interval in which transmission requests for ABT are automatically set (FCNnGMADCTL = 00_H), messages other than ABT messages may be transmitted not depending on their priority compared to the priority of the ABT message.
7. Do not clear FCNnMmCTL.FCNnMmRDYF to 0 when FCNnGMABCTL.FCNnGMABABTT = 1.

24.8.4 Transmission abort process

(1) Transmission abort process except for in normal operation mode with automatic block transmission (ABT)

The user can clear FCNnMmCTL.FCNnMmTRQF to 0 to abort a transmission request. FCNnMmTRQF will be cleared immediately if the abort was successful. Whether the transmission was successfully aborted or not can be checked using FCNnCMCLCTL.FCNnCMCLSSTS and the FCNnCMTGTX register, or the FCNnMmCTL.FCNnMmTCPF flag, which indicate the transmission status on the CAN bus (for details, refer to the processing in *Figure 24-24 "Transmission abort processing (except normal operation mode with ABT)" on page 1666*).

(2) Transmission abort process for ABT transmission in normal operation mode with automatic block transmission (ABT)

To abort ABT that is already started, clear FCNnGMABCTL.FCNnGMABABTT to 0. In this case, FCNnGMABCTL.FCNnGMABABTT remains 1 if an ABT message is currently being transmitted and until the transmission is completed (successfully or not), and is cleared to 0 as soon as transmission is finished. This aborts ABT.

If the last transmission (before ABT) was successful, the normal operation mode with ABT is left with the internal ABT pointer pointing to the next message buffer to be transmitted.

In the case of an erroneous transmission, the position of the internal ABT pointer depends on the status of FCNnMmCTL.FCNnMmTRQF in the last transmitted message buffer. If FCNnMmTRQF is cleared to 0 when clearing FCNnGMABCTL.FCNnGMABABTT is requested, the internal ABT pointer is incremented (+1) and points to the next message buffer in the ABT area (for details, refer to the process in *Figure 24-26 "ABT transmission request abort processing (in normal operation mode with ABT) (1)" on page 1668*).

Caution Be sure to abort ABT by clearing FCNnGMABCTL.FCNnGMABABTT to 0. The operation is not guaranteed if aborting transmission is requested by clearing FCNnMmCTL.FCNnMmRDYF.

When the normal operation mode with ABT is resumed after ABT has been aborted and FCNnGMABCTL.FCNnGMABSEAT is set to 1, the next ABT message buffer to be transmitted can be determined from the following table.

Status of FCNnMmCTL. FCNnMmTRQF of ABT message buffer	Abort after successful transmission	Abort after erroneous transmission
Set (1)	Next message buffer in the ABT area[a]	Same message buffer in the ABT area
Cleared (0)	Next message buffer in the ABT area[a]	Next message buffer in the ABT area[a]

a) The above resumption operation can be performed only if a message buffer ready for ABT exists in the ABT area. For example, an abort request that is issued while ABT of highest ABT message buffer is in progress is regarded as completion of ABT, rather than abort, if transmission of this message buffer has been successfully completed, even if FCNnGMABCTL.FCNnGMABABTT is cleared to 0. If FCNnMmCTL.FCNnMmRDYF in the next message buffer in the ABT area is cleared to 0, the internal ABT pointer is retained, but the resumption operation is not performed even if FCNnGMABABTT is set to 1, and ABT ends immediately.

24.8.5 Remote frame transmission

Remote frames can be transmitted only from transmit message buffers. Set whether a data frame or remote frame is transmitted via FCNnMmSTRB.FCNnMmSSRT. Setting FCNnMmSSRT = 1 sets remote frame transmission.

24.9 Power Saving Modes

24.9.1 FCN sleep mode

The FCN sleep mode can be used to set the CAN Controller to stand-by mode in order to reduce power consumption. The FCN module can enter the FCN sleep mode from all operation modes. Release of the FCN sleep mode returns the FCN module to exactly the same operation mode from which the FCN sleep mode was entered.

In the FCN sleep mode, the FCN module does not transmit messages, even when transmission requests are issued or pending.

(1) Entering FCN sleep mode

The CPU issues a FCN sleep mode transition request by setting FCNnCMCLCTL.FCNnCMCLMDPF[1:0] = 01_B.

This transition request is acknowledged only under the following conditions.

1. The FCN module is already in one of the following operation modes
   – Normal operation mode
   – Normal operation mode with ABT
   – Receive-only mode
   – Single-shot mode
   – Self-test mode
   – FCN stop mode in all the above operation modes
2. The CAN bus state is bus idle (the 4th bit in the interframe space is recessive).
   If the CAN bus is fixed to dominant, the request for transition to the FCN sleep mode is held pending. Also the transition from FCN stop mode to FCN sleep mode is independent of the CAN bus state.
3. No transmission request is pending.
4. Power save mode cannot be set in combination with the change of operation mode. Be sure to perform these operations in different steps.

Note If a sleep mode request is pending, and at the same time a message is received in a message box, the sleep mode request is not cancelled, but is executed right after message storage has been finished. This may result in FCN being in sleep mode, while the CPU would execute the RX interrupt routine. Therefore, the interrupt routine must check the access to the message buffers as well as reception history list registers by using the FCNnGMCLSSMO flag, if sleep mode is used.

If any one of the conditions mentioned above is not met, the FCN module will operate as follows.

- If the FCN sleep mode is requested from the initialization mode, the FCN sleep mode transition request is ignored and the FCN module remains in the initialization mode.
- If the CAN bus state is not bus idle (i.e., the CAN bus state is either transmitting or receiving) when the FCN sleep mode is requested in one of the operation modes, immediate transition to the FCN sleep mode is not possible. In this case, the FCN sleep mode transition request is held pending until the CAN bus state becomes bus idle (the 4th bit in the interframe space is recessive). In the time from the FCN sleep mode request to successful transition, FCNnCMCLCTL.FCNnCMCLMDPF[1:0]

remain 00_B. When the module has entered the FCN sleep mode, the FCNnCMCLMDPF[1:0] bits are set to 01_B.

- If a request for transition to the initialization mode and a request for transition to the FCN sleep mode are made at the same time while the FCN module is in one of the operation modes, the request for the initialization mode is enabled. The FCN module enters the initialization mode at a predetermined timing. At this time, the FCN sleep mode request is not held pending and is ignored.
- Even when initialization mode and sleep mode are not requested simultaneously (i.e the first request has not been granted while the second request is made), the request for initialization has priority over the sleep mode request. The sleep mode request is cancelled when the initialization mode is requested. When a pending request for initialization mode is present, a subsequent request for Sleep mode request is cancelled right at the point in time where it was submitted.

(2) Status in FCN sleep mode

The FCN module is in the following state after it enters the FCN sleep mode:

- The internal operating clock is stopped and the power consumption is minimized.
- The function to detect the falling edge of the FCN reception pin (CRXDn) remains in effect to wake up the FCN module from the CAN bus.
- To wake up the FCN module from the CPU, data can be set to FCNnCMCLCTL.FCNnCMCLMDPF[1:0], but nothing can be written to other FCN module registers or bits.
- The FCN module registers can be read, except for the FCNnCMLISTR, FCNnCMRGRX, FCNnCMLOSTR, and FCNnCMTGTX registers.
- The FCN message buffer registers cannot be written or read.
- FCNnGMCLCTL.FCNnGMCLSSMO is cleared.
- The registers FCNnDNBMRX cannot be read.
- A request for transition to the initialization mode is not acknowledged and is ignored.

(3) Releasing FCN sleep mode

The FCN sleep mode is released by the following events:

- When the CPU sets FCNnCMCLCTL.FCNnCMCLMDPF[1:0] to 00_B
- A falling edge at the FCN reception pin CRXDn (i.e. the CAN bus level shifts from recessive to dominant)

Caution Even if the falling edge belongs to the SOF of a receive message, this message will not be received and stored. If the CPU has turned off the clock supply to the FCN module while the FCN module was in sleep mode, even subsequently the FCN sleep mode will not be released and FCNnCMCLMDPF[1:0] will remain 01_B unless the clock to the FCN module is supplied again. In addition to this, the receive message will not be received after that.

After releasing the sleep mode, the FCN module returns to the operation mode from which the FCN sleep mode was requested and FCNnCMCLMDPF must be reset by software to 00_B. If the FCN sleep mode is released by a change in the CAN bus state, FCNnCMISCTL.FCNnCMISITSF5 is set to 1, regardless of FCNnCMIECTL.FCNnCMIEINTF[6:0]. After the FCN module is released from the FCN sleep mode, it participates in the CAN bus again by automatically detecting 11 consecutive recessive-level bits on the CAN bus. The user application has to wait until FCNnGMCLCTL.FCNnGMCLSSMO = 1, before accessing message buffers again.

When a request for transition to the initialization mode is made while the FCN module is in the FCN sleep mode, that request is ignored; the FCN module has to be released from sleep mode by software first before entering the initialization mode.

Cautions

1. Be aware that the release of FCN sleep mode by CAN bus event, and thus the wake up interrupt may happen at any time, even right after requesting sleep mode, if a CAN bus event occurs.
2. Always reset the FCNnCMCLCTL.FCNnCMCLMDPF[1:0] bits to 00_B, when waking up from FCN sleep mode, before accessing any other registers of the FCN module.
3. Always clear the interrupt flag FCNnCMISCTL.FCNnCMISITSF5, when waking up from FCN sleep mode.

24.9.2 FCN stop mode

The FCN stop mode can be used to set the CAN controller to stand-by mode to reduce power consumption. The FCN module can enter the FCN stop mode only from the FCN sleep mode. Release of the FCN stop mode puts the FCN module in the FCN sleep mode.

The FCN stop mode can only be released (entering FCN sleep mode) by setting 01_B to FCNnCMCLCTL.FCNnCMCLMDPF[1:0] and not by a change in the CAN bus state. No message is transmitted even when transmission requests are issued or pending.

(1) Entering FCN stop mode

A FCN stop mode transition request is issued by setting 11_B to FCNnCMCLCTL.FCNnCMCLMDPF[1:0].

A FCN stop mode request is only acknowledged when the FCN module is in the FCN sleep mode. In all other modes, the request is ignored.

Caution To set the FCN module to the FCN stop mode, the module must be in the FCN sleep mode. To confirm that the module is in the sleep mode, check that the FCNnCMCLCTL.FCNnCMCLMDPF[1:0] = 01_B, and then request the FCN stop mode. If a bus change occurs at the FCN reception pin CRXDn while this process is being performed, the FCN sleep mode is automatically released. In this case, the FCN stop mode transition request cannot be acknowledged.

(2) Status in FCN stop mode

The FCN module is in the following state after it enters the FCN stop mode.

- The internal operating clock is stopped and the power consumption is minimized.
- To wake up the FCN module from the CPU, data can be set to FCNnCMCLCTL.FCNnCMCLMDPF[1:0], but nothing can be written to other FCN module registers or bits.
- The FCN module registers can be read, except for the FCNnCMLISTR, FCNnCMRGRX, FCNnCMLOSTR, and FCNnCMTGTX registers.
- The FCN message buffer registers cannot be written or read.
- FCNnGMCLCTL.FCNnGMCLSSMO is cleared.
- The registers FCNnDNBMRX cannot be read.
- An initialization mode transition request is not acknowledged and is ignored.

(3) Releasing FCN stop mode

The FCN stop mode can only be released by writing 01_B to FCNnCMCLCTL.FCNnCMCLMDPF[1:0]. After releasing the FCN stop mode, the FCN module enters the FCN sleep mode.

When the initialization mode is requested while the FCN module is in the FCN stop mode, that request is ignored; the CPU has to release the stop mode and subsequently FCN sleep mode before entering the initialization mode. It is impossible to enter the other operation mode directly from the FCN stop mode not entering the FCN sleep mode, that request is ignored.

24.9.3 Example of using power saving modes

In some application systems, it may be necessary to place the CPU in a power saving mode to reduce the power consumption. By using the power saving mode specific to the FCN module and the power saving mode specific to the CPU in combination, the CPU can be woken up from the power saving status by the CAN bus.

Here is an example for using the power saving modes.

- First, put the FCN module in the FCN sleep mode (FCNnCMCLCTL.FCNnCMCLMDPF[1:0] = 01_B).
  After successfully confirming this state by reading back the sleep mode status, put the CPU in the power saving mode. Disable interrupts for the CPU, while processing additional tasks after the FCN module is in sleep mode, to avoid that the FCN wakeup interrupt is acknowledged.
  If a rising edge from recessive to dominant is detected on the CRXDn FCN reception pin in this state, FCNnCMISCTL.FCNnCMISITSF5 in the FCN module will be set to 1. If FCNnCNIECTL.FCNnCMIEINT5 is set to 1, a wakeup interrupt (INTCnWUP) is generated.
  The FCN module is automatically released from FCN sleep mode (FCNnCMCLMDPF[1:0] = 00_B) and returns to normal operation mode.
- The CPU, in response to INTCnWUP, can release its own power saving mode and return to normal operation mode.

  To further reduce the power consumption of the CPU, the internal clock - including that of the FCN module - may be stopped. In this case, the operating clock supplied to the FCN module is stopped after the FCN module has been put in FCN sleep mode. Then the CPU enters a power saving mode in which the clock supplied to the CPU is stopped.
- If an edge transition from recessive to dominant is detected at the FCN reception pin CRXDn in this status, the FCN module can set FCNnCMISCTL.FCNnCMISITSF5 to 1 and generate the wakeup interrupt INTCnWUP even if it is not supplied with the clock.
- The other functions, however, do not operate, because clock supply to the FCN module is stopped, and the module remains in FCN sleep mode.
- The CPU, in response to INTCnWUP,
  - releases its power saving mode,
  - resumes supply of the internal clocks - including the clock to the FCN module - after the oscillation stabilization time has elapsed, and
  - starts instruction execution.
- The FCN module is immediately released from the FCN sleep mode when clock supply is resumed, and returns to the normal operation mode (FCNnCMCLCTL.FCNnCMCLMDPF[1:0] = 00_B).

24.10 Interrupt Function

The FCN module provides 6 different interrupt sources.

The occurrence of these interrupt sources is stored in interrupt status registers. Four separate interrupt request signals are generated from the six interrupt sources. When an interrupt request signal that corresponds to two or more interrupt sources is generated, the interrupt sources can be identified by using an interrupt status register. After an interrupt source has occurred, the corresponding interrupt status bit must be cleared to 0 by software.

Table 24-17 List of FCN module interrupt sources

No.	Interrupt status bit FCNnCMISCTL.	Interrupt enable bit FCNnCMIECTL. [a]	Interrupt request signal	Interrupt source description
1	FCNnCMISITSF0	FCNnCMIESEIE0	INTCnTRX	Message frame successfully transmitted from message buffer m
2	FCNnCMISITSF1	FCNnCMIESEIE1	INTCnREC	Valid message frame reception in message buffer m
3	FCNnCMISITSF2	FCNnCMIESEIE2	INTCnERR	FCN module error state interrupt • This interrupt is generated when the transmission/reception error counter is at the warning level, or in the error passive or bus-off state.
4	FCNnCMISITSF3	FCNnCMIESEIE3		FCN module protocol error interrupt • This interrupt is generated when a stuff error, form error, ACK error, bit error, or CRC error occurs.
5	FCNnCMISITSF4	FCNnCMIESEIE4		FCN module arbitration loss interrupt
6	FCNnCMISITSF5	FCNnCMIESEIE5	INTCnWUP	FCN module wakeup interrupt from FCN sleep mode • This interrupt is generated when the FCN module wakes up from FCN sleep mode, due to detection of a rise on the FCN reception pin (change of CAN bus from recessive to dominant).
7	FCNnCMISITSF6	FCNnCMIESEIE6		FCN module transmit abort interrupt status • This interrupt is generated when the abortion of a transmission was successful (aborted message was not sent).

a) The message buffer interrupt enable bit FCNnMmCTL.FCNnMmIENF of the corresponding message buffer has to be set to 1 for that message buffer to participate in the interrupt generation process.

24.11 Diagnosis Functions and Special Operational Modes

The FCN module provides a receive-only mode, single-shot mode, and self-test mode to support CAN bus diagnosis functions or the operation of special CAN communication methods.

24.11.1 Receive-only mode

The receive-only mode is used to monitor receive messages without causing any interference on the CAN bus and can be used for CAN bus analysis nodes.

For example, this mode can be used for automatic baud-rate detection. The FCN module can change its baud rate during reception until “valid reception” is detected, so that the baud rate matches the baud rate on the transmission side (“valid reception” means a message frame has been received in the CAN protocol layer without occurrence of an error and with an appropriate ACK between nodes connected to the CAN bus). A valid reception does not require message frames to be stored in a receive message buffer (data frames) or transmit message buffer (remote frames). The event of valid reception is indicated by setting FCNnCMCLCTL.FCNnCMCLVALF = 1.

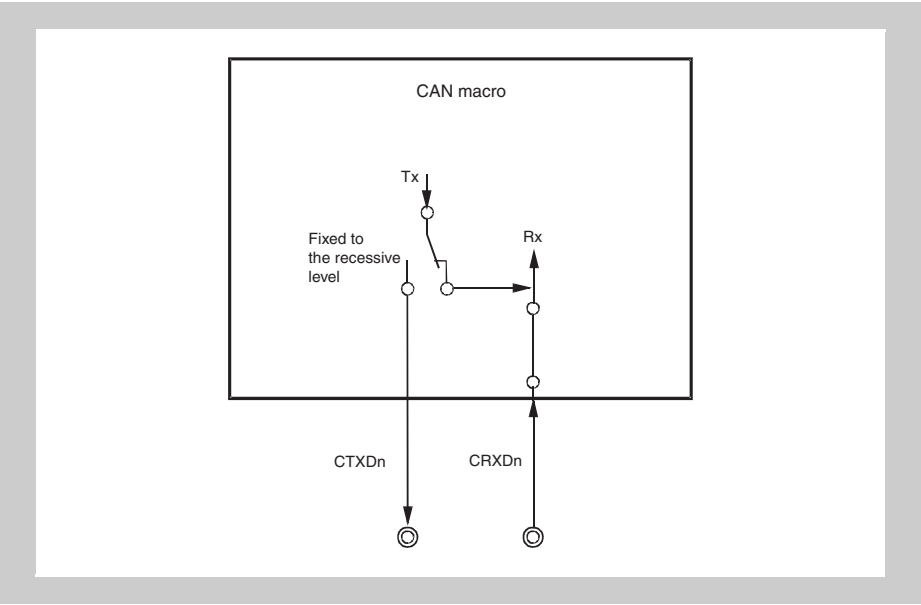


Figure 24-10 FCN module terminal connection in receive-only mode

In the receive-only mode, no message frames can be transmitted from the FCN module to the CAN bus. Transmit requests issued for message buffers defined as transmit message buffers are held pending.

In the receive-only mode, the FCN transmission pin CTXDn in the FCN module is fixed to the recessive level. Therefore, no active error flag can be transmitted from the FCN module to the CAN bus even when a CAN bus error is detected while receiving a message frame. Since no transmission can be issued from the FCN module, the transmission error counter the FCNnCMERCNT.FCNnCMERTECF[7:0] bits are never updated. Therefore, a FCN module in the receive-only mode does not enter the bus-off state.

Furthermore, in the receive-only mode ACK is not returned to the CAN bus in this mode upon the valid reception of a message frame. Internally, the local node recognizes that it has transmitted ACK. An overload frame cannot be transmitted to the CAN bus.

Caution If only two CAN nodes are connected to the CAN bus and one of them is operating in the receive-only mode, there is no ACK on the CAN bus. Due to the missing ACK, the transmitting node will transmit an active error flag, and repeat transmitting a message frame. The transmitting node becomes error passive after transmitting the message frame 16 times (assuming that the error counter was 0 in the beginning and no other errors have occurred). After the message frame for the 17th time is transmitted, the transmitting node generates a passive error flag. The receiving node in the receive-only mode detects the first valid message frame at this point, and the FCNnCMCLCTL.FCNnCMCLVALF bit is set to 1 for the first time.

24.11.2 Single-shot mode

In the single-shot mode, automatic re-transmission as defined in the CAN protocol is switched off. (According to the CAN protocol, a message frame transmission that has been aborted by either arbitration loss or error occurrence has to be repeated without control by software.) All other behavior of single shot mode is identical to normal operation mode. Features of single shot mode can not be used in combination with normal mode with ABT.

The single-shot mode disables the re-transmission of an aborted message frame transmission according to the setting of FCNnCMCLCTL.FCNnCMCLALBF. When FCNnCMCLALBF is cleared to 0, re-transmission upon arbitration loss and upon error occurrence is disabled. If FCNnCMCLALBF is set to 1, re-transmission upon error occurrence is disabled, but re-transmission upon arbitration loss is enabled. As a consequence, FCNnMmCTL.FCNnMmTRQF in a message buffer defined as a transmit message buffer is cleared to 0 by the following events:

- Successful transmission of the message frame
- Arbitration loss while sending the message frame
- Error occurrence while sending the message frame

The events arbitration loss and error occurrence can be distinguished by checking FCNnCMISCTL.FCNnCMISITSF4 and FCNnCMISCTL.FCNnCMISITSF3 respectively, and the type of the error can be identified by reading FCNnCMLCSTR.FCNnCMLCSSLC[2:0].

Upon successful transmission of the message frame, the transmit completion interrupt bit FCNnCMISCTL.FCNnCMISITSF0 is set to 1. If FCNnCMIECTL.FCNnCMIEINTF0 is set to 1 at this time, an interrupt request signal is output.

The single-shot mode can be used when emulating time-triggered communication methods (e.g., TTCAN level 1).

Caution FCNnCMCLCTL.FCNnCMCLALBF is only valid in single-shot mode. It does not influence the operation of re-transmission upon arbitration loss in the other operation modes.

24.11.3 Self-test mode

In the self-test mode, message frame transmission and message frame reception can be tested without connecting the CAN node to the CAN bus or without affecting the CAN bus.

In the self-test mode, the FCN module is completely disconnected from the CAN bus, but transmission and reception are internally looped back. The FCN transmission pin CTXDn is fixed to the recessive level.

If the falling edge on the FCN reception pin CRXDn is detected after the FCN module has entered the FCN sleep mode from the self-test mode, however, the module is released from the FCN sleep mode in the same manner as the other operation modes. Use the CRXDn FCN reception pin as a port pin in order to keep the module in FCN sleep mode.

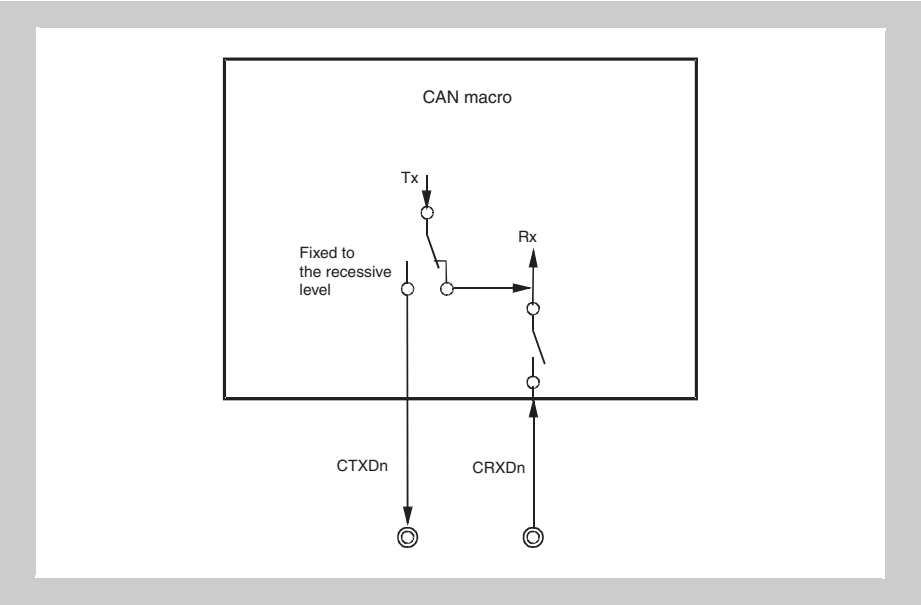


Figure 24-11 FCN module terminal connection in self-test mode

24.11.4 Receive/transmit operation in each operation mode

The following table shows outline of the receive/transmit operation in each operation mode.

Table 24-18 Outline of the receive/transmit in each operation mode

Operation mode	Transmission of data/ remote frame	Transmission of ACK	Transmission of error/ overload frame	Transmission retry	Automatic block transmission (ABT)	Set of FCNnCMCLVALF bit	Store data to message buffer
Initialization mode	No	No	No	No	No	No	No
Normal operation mode	Yes	Yes	Yes	Yes	No	Yes	Yes
Normal operation mode with ABT	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Receive only mode	No	No	No	No	No	Yes	Yes
Single-shot mode	Yes	Yes	Yes	No[a]	No	Yes	Yes
Self-test mode	Yes[b]	Yes[b]	Yes[b]	Yes[b]	No	Yes[b]	Yes[b]

a) When the arbitration lost occurs, control of re-transmission is possible by FCNnCMCLCTL.FCNnCMCLALBF.
b) Each signals are not generated to outside, but generated into the FCN module.

24.12 Time Stamp Function

CAN is an asynchronous serial communication protocol. All nodes connected to the CAN bus have a local, autonomous clock. As a consequence, the clocks of the nodes have no relation (i.e., the clocks are asynchronous and may have different frequencies).

In some applications, however, a common time base over the network (= global time base) is needed. In order to build up a global time base, a time stamp function is used. The essential mechanism of a time stamp function is the capture of timer values triggered by signals on the CAN bus.

24.12.1 Time stamp function

The CAN Controller supports the capturing of timer values triggered by a specific frame. An on-chip 16-bit capture timer unit in a microcontroller system is used in addition to the CAN Controller. The 16-bit capture timer unit captures the timer value according to a trigger signal (TSOUT) for capturing that is output when a data frame is received from the CAN Controller. The CPU can retrieve the time of occurrence of the capture event, i.e., the time stamp of the message received from the CAN bus, by reading the captured value. The TSOUT signal can be selected from the following two event sources and is specified by FCNnCMTSCTL.FCNnCMTSSELE.

- SOF event (start of frame)
  (FCNnCMTSCTL.FCNnCMTSSELE = 0)
- EOF event (last bit of end of frame)
  (FCNnCMTSCTL.FCNnCMTSSELE = 1)

The TSOUT signal is enabled by setting FCNnCMTSCTL.FCNnCMTSTSGE = 1.

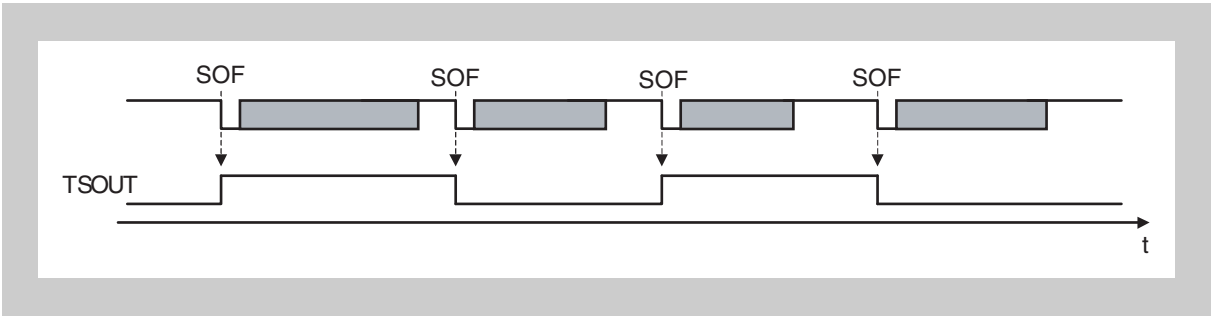


Figure 24-12 Timing diagram of capture signal TSOUT

The TSOUT signal toggles its level upon occurrence of the selected event during data frame reception (in *Figure 24-12 "Timing diagram of capture signal TSOUT"*, the SOF is used as the trigger event source). To capture a timer value by using the TSOUT signal, the capture timer unit must detect the capture signal at both the rising edge and falling edge.

This time stamp function is controlled by the FCNnCMTSLOKE bit of the FCNnCMTSCTL register. When FCNnCMTSLOKE is cleared to 0, the TSOUT signal toggles upon occurrence of the selected event. If FCNnCMTSLOKE is set to 1, the TSOUT signal toggles upon occurrence of the selected event, but the toggle is stopped as FCNnCMTSCTL.FCNnCMTSTSGE is automatically cleared to 0 as soon as the message storing to the message buffer 0 starts. This suppresses the subsequent toggle occurrence by the TSOUT signal, so

that the time stamp value toggled last (= captured last) can be saved as the time stamp value of the time at which the data frame was received in message buffer 0.

Caution The time stamp function using the FCNnCMTSLOKE bit stops toggle of the TSOUT signal by receiving a data frame in message buffer 0. Toggle of the TSOUT signal does not stop when a data frame is received in a message buffer other than message buffer 0.
A data frame cannot be received in message buffer 0 when the FCN module is in the normal operation mode with ABT, because message buffer 0 must be set as a transmit message buffer.
In this operation mode, therefore, the function to stop toggle of the TSOUT signal by the FCNnCMTSLOKE bit cannot be used.

24.13 Baud Rate Settings

24.13.1 Baud rate setting conditions

Make sure that the settings are within the range of limit values for ensuring correct operation of the CAN Controller, as follows.

- 5 TQ ≤ SPT (sampling point) ≤ 17 TQ

  SPT = TSEG1 + 1
- 8 TQ ≤ DBT (data bit time) ≤ 25 TQ

  DBT = TSEG1 + TSEG2 + 1 TQ = TSEG2 + SPT
- 1 TQ ≤ SJW (synchronization jump width) ≤ 4 TQ

  SJW ≤ DBT – SPT
- <R> 4TQ ≤ TSEG1 ≤ 16TQ [3 ≤ FCNnCMBTS1LG[3:0] ≤ 15]
- 1 ≤ TSEG2 ≤ 8

Notes

1. TQ = 1/f_{TQ} (f_{TQ}: CAN protocol layer basic system clock)
2. The values of TSEG1, TSEG2, and SJW are defined by the bits of the FCNnCMBTCTL register as follows:
   TSEG1: FCNnCMBTCTL.FCNnCMBTS1LG[3:0] + 1
   TSEG2: FCNnCMBTCTL.FCNnCMBTS2LG[2:0] + 1
   SJW: FCNnCMBTCTL.FCNnCMBTJWLG[1:0] + 1

Table 24-19 "Settable bit rate combinations" shows the combinations of bit rates that satisfy the above conditions.

Table 24-19 Settable bit rate combinations (1/3)

Valid bit rate setting					FCNnCMBTCTL register setting value		Sampling point (unit: %)
DBT length	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT1	PHASE SEGMENT2	FCNnCMBTS1LG[3:0]	FCNnCMBTS2LG[2:0]	
25	1	8	8	8	1111	111	68.0
24	1	7	8	8	1110	111	66.7
24	1	9	7	7	1111	110	70.8
23	1	6	8	8	1101	111	65.2
23	1	8	7	7	1110	110	69.6
23	1	10	6	6	1111	101	73.9
22	1	5	8	8	1100	111	63.6
22	1	7	7	7	1101	110	68.2
22	1	9	6	6	1110	101	72.7
22	1	11	5	5	1111	100	77.3
21	1	4	8	8	1011	111	61.9
21	1	6	7	7	1100	110	66.7
21	1	8	6	6	1101	101	71.4
21	1	10	5	5	1110	100	76.2
21	1	12	4	4	1111	011	81.0
20	1	3	8	8	1010	111	60.0

Table 24-19 Settable bit rate combinations (2/3)

Valid bit rate setting					FCNnCMBTCTL register setting value		Sampling point (unit: %)
DBT length	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT1	PHASE SEGMENT2	FCNnCMBTS1LG[3:0]	FCNnCMBTS2LG[2:0]	
20	1	5	7	7	1011	110	65.0
20	1	7	6	6	1100	101	70.0
20	1	9	5	5	1101	100	75.0
20	1	11	4	4	1110	011	80.0
20	1	13	3	3	1111	010	85.0
19	1	2	8	8	1001	111	57.9
19	1	4	7	7	1010	110	63.2
19	1	6	6	6	1011	101	68.4
19	1	8	5	5	1100	100	73.7
19	1	10	4	4	1101	011	78.9
19	1	12	3	3	1110	010	84.2
19	1	14	2	2	1111	001	89.5
18	1	1	8	8	1000	111	55.6
18	1	3	7	7	1001	110	61.1
18	1	5	6	6	1010	101	66.7
18	1	7	5	5	1011	100	72.2
18	1	9	4	4	1100	011	77.8
18	1	11	3	3	1101	010	83.3
18	1	13	2	2	1110	001	88.9
18	1	15	1	1	1111	000	94.4
17	1	2	7	7	1000	110	58.8
17	1	4	6	6	1001	101	64.7
17	1	6	5	5	1010	100	70.6
17	1	8	4	4	1011	011	76.5
17	1	10	3	3	1100	010	82.4
17	1	12	2	2	1101	001	88.2
17	1	14	1	1	1110	000	94.1
16	1	1	7	7	0111	110	56.3
16	1	3	6	6	1000	101	62.5
16	1	5	5	5	1001	100	68.8
16	1	7	4	4	1010	011	75.0
16	1	9	3	3	1011	010	81.3
16	1	11	2	2	1100	001	87.5
16	1	13	1	1	1101	000	93.8
15	1	2	6	6	0111	101	60.0
15	1	4	5	5	1000	100	66.7
15	1	6	4	4	1001	011	73.3
15	1	8	3	3	1010	010	80.0
15	1	10	2	2	1011	001	86.7

Table 24-19 Settable bit rate combinations (2/3)

Valid bit rate setting					FCNnCMBTCTL register setting value		Sampling point (unit: %)
DBT length	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT1	PHASE SEGMENT2	FCNnCMBTS1LG[3:0]	FCNnCMBTS2LG[2:0]	
20	1	5	7	7	1011	110	65.0
20	1	7	6	6	1100	101	70.0
20	1	9	5	5	1101	100	75.0
20	1	11	4	4	1110	011	80.0
20	1	13	3	3	1111	010	85.0
19	1	2	8	8	1001	111	57.9
19	1	4	7	7	1010	110	63.2
19	1	6	6	6	1011	101	68.4
19	1	8	5	5	1100	100	73.7
19	1	10	4	4	1101	011	78.9
19	1	12	3	3	1110	010	84.2
19	1	14	2	2	1111	001	89.5
18	1	1	8	8	1000	111	55.6
18	1	3	7	7	1001	110	61.1
18	1	5	6	6	1010	101	66.7
18	1	7	5	5	1011	100	72.2
18	1	9	4	4	1100	011	77.8
18	1	11	3	3	1101	010	83.3
18	1	13	2	2	1110	001	88.9
18	1	15	1	1	1111	000	94.4
17	1	2	7	7	1000	110	58.8
17	1	4	6	6	1001	101	64.7
17	1	6	5	5	1010	100	70.6
17	1	8	4	4	1011	011	76.5
17	1	10	3	3	1100	010	82.4
17	1	12	2	2	1101	001	88.2
17	1	14	1	1	1110	000	94.1
16	1	1	7	7	0111	110	56.3
16	1	3	6	6	1000	101	62.5
16	1	5	5	5	1001	100	68.8
16	1	7	4	4	1010	011	75.0
16	1	9	3	3	1011	010	81.3
16	1	11	2	2	1100	001	87.5
16	1	13	1	1	1101	000	93.8
15	1	2	6	6	0111	101	60.0
15	1	4	5	5	1000	100	66.7
15	1	6	4	4	1001	011	73.3
15	1	8	3	3	1010	010	80.0
15	1	10	2	2	1011	001	86.7

Table 24-19 Settable bit rate combinations (3/3)

Valid bit rate setting					FCNnCMBTCTL register setting value		Sampling point (unit: %)
DBT length	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT1	PHASE SEGMENT2	FCNnCMBTS1LG[3:0]	FCNnCMBTS2LG[2:0]	
15	1	12	1	1	1100	000	93.3
14	1	1	6	6	0110	101	57.1
14	1	3	5	5	0111	100	64.3
14	1	5	4	4	1000	011	71.4
14	1	7	3	3	1001	010	78.6
14	1	9	2	2	1010	001	85.7
14	1	11	1	1	1011	000	92.9
13	1	2	5	5	0110	100	61.5
13	1	4	4	4	0111	011	69.2
13	1	6	3	3	1000	010	76.9
13	1	8	2	2	1001	001	84.6
13	1	10	1	1	1010	000	92.3
12	1	1	5	5	0101	100	58.3
12	1	3	4	4	0110	011	66.7
12	1	5	3	3	0111	010	75.0
12	1	7	2	2	1000	001	83.3
12	1	9	1	1	1001	000	91.7
11	1	2	4	4	0101	011	63.6
11	1	4	3	3	0110	010	72.7
11	1	6	2	2	0111	001	81.8
11	1	8	1	1	1000	000	90.9
10	1	1	4	4	0100	011	60.0
10	1	3	3	3	0101	010	70.0
10	1	5	2	2	0110	001	80.0
10	1	7	1	1	0111	000	90.0
9	1	2	3	3	0100	010	66.7
9	1	4	2	2	0101	001	77.8
9	1	6	1	1	0110	000	88.9
8	1	1	3	3	0011	010	62.5
8	1	3	2	2	0100	001	75.0
8	1	5	1	1	0101	000	87.5
7[a]	1	2	2	2	0011	001	71.4
7[a]	1	4	1	1	0100	000	85.7
6[a]	1	1	2	2	0010	001	66.7
6[a]	1	3	1	1	0011	000	83.3
5[a]	1	2	1	1	0010	000	80.0
4[a]	1	1	1	1	0001	000	75.0

a) Setting with a DBT value of 7 or less is valid only when the value of the FCNnCMBRPRS register is other than 00_H.

Caution The values in *Table 24-19 "Settable bit rate combinations"* do not guarantee the operation of the network system. Thoroughly check the effect on the network system, taking into consideration oscillation errors and delays of the CAN bus and CAN transceiver.

24.13.2 Representative examples of baud rate settings

Table 24-20 "Representative examples of baud rate settings (f_{CANPRE} = 8 MHz)" and *Table 24-21 "Representative examples of baud rate settings (f_{CANPRE} = 16 MHz)"* show representative examples of baud rate settings.

Table 24-20 Representative examples of baud rate settings (f_{CANPRE} = 8 MHz) (1/2)

Set baud rate value (unit: kbps)	Division ratio of FCNnCMBRPRS register	FCNnCMBRPRS register set value	Valid bit rate setting (unit: TQ)					FCNnCMBTCTL register setting value		Sampling point (unit: %)
			Length of DBT	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT1	PHASE SEGMENT2	FCNnCMBTS1LG[3:0]	FCNnCMBTS2LG[2:0]	
500	1	00000000	16	1	1	7	7	0111	110	56.3
500	1	00000000	16	1	3	6	6	1000	101	62.5
500	1	00000000	16	1	5	5	5	1001	100	68.8
500	1	00000000	16	1	7	4	4	1010	011	75.0
500	1	00000000	16	1	9	3	3	1011	010	81.3
500	1	00000000	16	1	11	2	2	1100	001	87.5
500	1	00000000	16	1	13	1	1	1101	000	93.8
500	2	00000001	8	1	1	3	3	0011	010	62.5
500	2	00000001	8	1	3	2	2	0100	001	75.0
500	2	00000001	8	1	5	1	1	0101	000	87.5
250	2	00000001	16	1	1	7	7	0111	110	56.3
250	2	00000001	16	1	3	6	6	1000	101	62.5
250	2	00000001	16	1	5	5	5	1001	100	68.8
250	2	00000001	16	1	7	4	4	1010	011	75.0
250	2	00000001	16	1	9	3	3	1011	010	81.3
250	2	00000001	16	1	11	2	2	1100	001	87.5
250	2	00000001	16	1	13	1	1	1101	000	93.8
250	4	00000011	8	1	3	2	2	0100	001	75.0
250	4	00000011	8	1	5	1	1	0101	000	87.5
125	4	00000011	16	1	1	7	7	0111	110	56.3
125	4	00000011	16	1	3	6	6	1000	101	62.5
125	4	00000011	16	1	5	5	5	1001	100	68.8
125	4	00000011	16	1	7	4	4	1010	011	75.0
125	4	00000011	16	1	9	3	3	1011	010	81.3
125	4	00000011	16	1	11	2	2	1100	001	87.5

Table 24-20 Representative examples of baud rate settings (f_{CANPRE} = 8 MHz) (2/2)

Set baud rate value (unit: kbps)	Division ratio of FCNnCMBRPRS register	FCNnCMBRPRS register set value	Valid bit rate setting (unit: TQ)					FCNnCMBTCTL register setting value		Sampling point (unit: %)
			Length of DBT	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT1	PHASE SEGMENT2	FCNnCMBTS1LG[3:0]	FCNnCMBTS2LG[2:0]	
125	4	00000011	16	1	13	1	1	1101	000	93.8
125	8	00000111	8	1	3	2	2	0100	001	75.0
125	8	00000111	8	1	5	1	1	0101	000	87.5
100	4	00000011	20	1	7	6	6	1100	101	70.0
100	4	00000011	20	1	9	5	5	1101	100	75.0
100	5	00000100	16	1	7	4	4	1010	011	75.0
100	5	00000100	16	1	9	3	3	1011	010	81.3
100	8	00000111	10	1	3	3	3	0101	010	70.0
100	8	00000111	10	1	5	2	2	0110	001	80.0
100	10	00001001	8	1	3	2	2	0100	001	75.0
100	10	00001001	8	1	5	1	1	0101	000	87.5
83.3	4	00000011	24	1	7	8	8	1110	111	66.7
83.3	4	00000011	24	1	9	7	7	1111	110	70.8
83.3	6	00000101	16	1	5	5	5	1001	100	68.8
83.3	6	00000101	16	1	7	4	4	1010	011	75.0
83.3	6	00000101	16	1	9	3	3	1011	010	81.3
83.3	6	00000101	16	1	11	2	2	1100	001	87.5
83.3	8	00000111	12	1	5	3	3	0111	010	75.0
83.3	8	00000111	12	1	7	2	2	1000	001	83.3
83.3	12	00001011	8	1	3	2	2	0100	001	75.0
83.3	12	00001011	8	1	5	1	1	0101	000	87.5
33.3	10	00001001	24	1	7	8	8	1110	111	66.7
33.3	10	00001001	24	1	9	7	7	1111	110	70.8
33.3	12	00001011	20	1	7	6	6	1100	101	70.0
33.3	12	00001011	20	1	9	5	5	1101	100	75.0
33.3	15	00001110	16	1	7	4	4	1010	011	75.0
33.3	15	00001110	16	1	9	3	3	1011	010	81.3
33.3	16	00001111	15	1	6	4	4	1001	011	73.3
33.3	16	00001111	15	1	8	3	3	1010	010	80.0
33.3	20	00010011	12	1	5	3	3	0111	010	75.0
33.3	20	00010011	12	1	7	2	2	1000	001	83.3
33.3	24	00010111	10	1	3	3	3	0101	010	70.0
33.3	24	00010111	10	1	5	2	2	0110	001	80.0
33.3	30	00011101	8	1	3	2	2	0100	001	75.0
33.3	30	00011101	8	1	5	1	1	0101	000	87.5

Cautions

1. The values in *Table 24-20 "Representative examples of baud rate settings (f_{CANPRE} = 8 MHz)"* do not guarantee the operation of the network system. Thoroughly check the effect on the network system, taking into consideration oscillation errors and delays of the CAN bus and CAN transceiver.
2. Baud rates higher than 500 kbit/s are not allowed with $f_{CANPRE} \leq 8$ MHz.

Table 24-21 Representative examples of baud rate settings (f_{CANPRE} = 16 MHz) (1/2)

Set baud rate value (unit: kbps)	Division ratio of FCNnCMBRPRS register	FCNnCMBRPRS register set value	Valid bit rate setting (unit: TQ)					FCNnCMBTCTL register setting value		Sampling point (unit: %)
			Length of DBT	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT1	PHASE SEGMENT2	FCNnCMBTS1LG[3:0]	FCNnCMBTS2LG[2:0]	
1000	1	00000000	16	1	1	7	7	0111	110	56.3
1000	1	00000000	16	1	3	6	6	1000	101	62.5
1000	1	00000000	16	1	5	5	5	1001	100	68.8
1000	1	00000000	16	1	7	4	4	1010	011	75.0
1000	1	00000000	16	1	9	3	3	1011	010	81.3
1000	1	00000000	16	1	11	2	2	1100	001	87.5
1000	1	00000000	16	1	13	1	1	1101	000	93.8
1000	2	00000001	8	1	3	2	2	0100	001	75.0
1000	2	00000001	8	1	5	1	1	0101	000	87.5
500	2	00000001	16	1	1	7	7	0111	110	56.3
500	2	00000001	16	1	3	6	6	1000	101	62.5
500	2	00000001	16	1	5	5	5	1001	100	68.8
500	2	00000001	16	1	7	4	4	1010	011	75.0
500	2	00000001	16	1	9	3	3	1011	010	81.3
500	2	00000001	16	1	11	2	2	1100	001	87.5
500	2	00000001	16	1	13	1	1	1101	000	93.8
500	4	00000011	8	1	3	2	2	0100	001	75.0
500	4	00000011	8	1	5	1	1	0101	000	87.5
250	4	00000011	16	1	3	6	6	1000	101	62.5
250	4	00000011	16	1	5	5	5	1001	100	68.8
250	4	00000011	16	1	7	4	4	1010	011	75.0
250	4	00000011	16	1	9	3	3	1011	010	81.3
250	4	00000011	16	1	11	2	2	1100	001	87.5
250	8	00000111	8	1	3	2	2	0100	001	75.0
250	8	00000111	8	1	5	1	1	0101	000	87.5
125	8	00000111	16	1	3	6	6	1000	101	62.5
125	8	00000111	16	1	7	4	4	1010	011	75.0
125	8	00000111	16	1	9	3	3	1011	010	81.3
125	8	00000111	16	1	11	2	2	1100	001	87.5

Table 24-21 Representative examples of baud rate settings (f_{CANPRE} = 16 MHz) (2/2)

Set baud rate value (unit: kbps)	Division ratio of FCNnCMBRPRS register	FCNnCMBRPRS register set value	Valid bit rate setting (unit: TQ)					FCNnCMBTCTL register setting value		Sampling point (unit: %)
			Length of DBT	SYNC SEGMENT	PROP SEGMENT	PHASE SEGMENT1	PHASE SEGMENT2	FCNnCMBTS1LG[3:0]	FCNnCMBTS2LG[2:0]	
125	16	00001111	8	1	3	2	2	0100	001	75.0
125	16	00001111	8	1	5	1	1	0101	000	87.5
100	8	00000111	20	1	9	5	5	1101	100	75.0
100	8	00000111	20	1	11	4	4	1110	011	80.0
100	10	00001001	16	1	7	4	4	1010	011	75.0
100	10	00001001	16	1	9	3	3	1011	010	81.3
100	16	00001111	10	1	3	3	3	0101	010	70.0
100	16	00001111	10	1	5	2	2	0110	001	80.0
100	20	00010011	8	1	3	2	2	0100	001	75.0
83.3	8	00000111	24	1	7	8	8	1110	111	66.7
83.3	8	00000111	24	1	9	7	7	1111	110	70.8
83.3	12	00001011	16	1	7	4	4	1010	011	75.0
83.3	12	00001011	16	1	9	3	3	1011	010	81.3
83.3	12	00001011	16	1	11	2	2	1100	001	87.5
83.3	16	00001111	12	1	5	3	3	0111	010	75.0
83.3	16	00001111	12	1	7	2	2	1000	001	83.3
83.3	24	00010111	8	1	3	2	2	0100	001	75.0
83.3	24	00010111	8	1	5	1	1	0101	000	87.5
33.3	30	00011101	24	1	7	8	8	1110	111	66.7
33.3	30	00011101	24	1	9	7	7	1111	110	70.8
33.3	24	00010111	20	1	9	5	5	1101	100	75.0
33.3	24	00010111	20	1	11	4	4	1110	011	80.0
33.3	30	00011101	16	1	7	4	4	1010	011	75.0
33.3	30	00011101	16	1	9	3	3	1011	010	81.3
33.3	32	00011111	15	1	8	3	3	1010	010	80.0
33.3	32	00011111	15	1	10	2	2	1011	001	86.7
33.3	37	00100100	13	1	6	3	3	1000	010	76.9
33.3	37	00100100	13	1	8	2	2	1001	001	84.6
33.3	40	00100111	12	1	5	3	3	0111	010	75.0
33.3	40	00100111	12	1	7	2	2	1000	001	83.3
33.3	48	00101111	10	1	3	3	3	0101	010	70.0
33.3	48	00101111	10	1	5	2	2	0110	001	80.0
33.3	60	00111011	8	1	3	2	2	0100	001	75.0
33.3	60	00111011	8	1	5	1	1	0101	000	87.5

Caution The values in *Table 24-21 “Representative examples of baud rate settings (f_{CANPRE} = 16 MHz)”* do not guarantee the operation of the network system. Thoroughly check the effect on the network system, taking into consideration oscillation errors and delays of the CAN bus and CAN transceiver.

24.14 Operation of the CAN Controller

The processing procedure for showing in this chapter is recommended processing procedure to operate FCN.

Develop the program referring to recommended processing procedure in this chapter.

24.14.1 Initialization

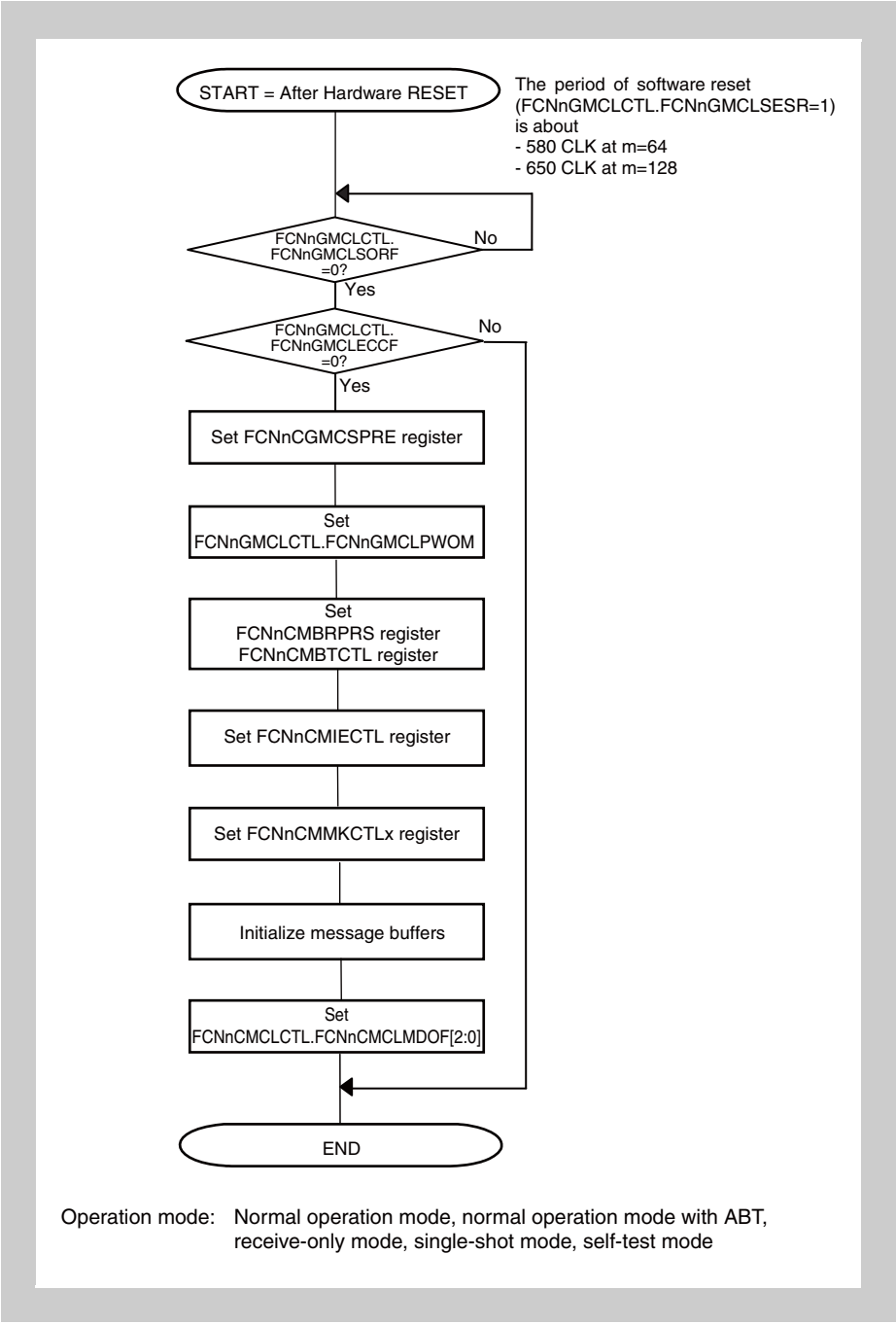


Figure 24-13 Initialization

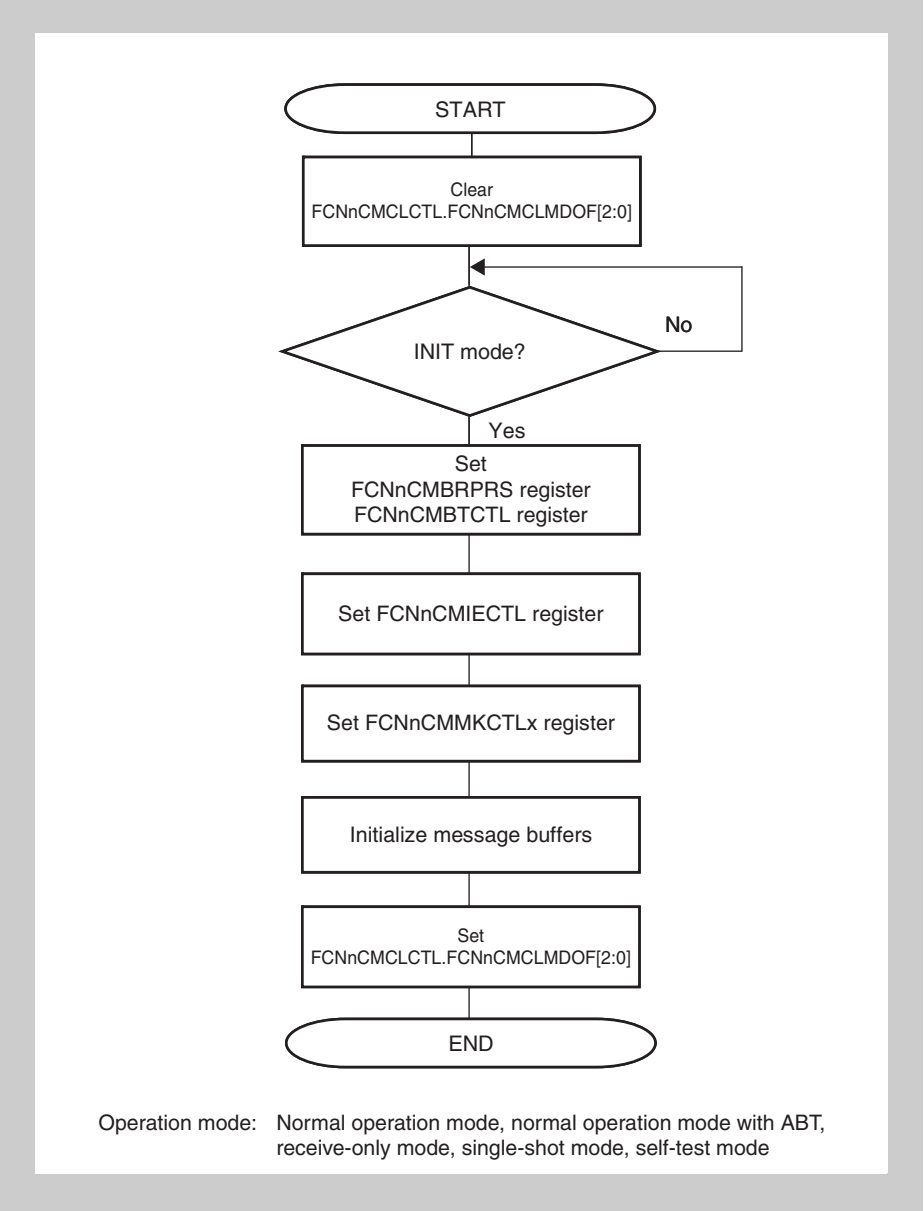


Figure 24-14 Re-initialization without using the software reset function

Caution To clear the error counter (by setting FCNnCMCLERCF) during re-initialization, do so in one of the following states.

- In the initialization mode state following FCN module start (by setting FCNnGMCLPWOM while FCNnGMCLPWOM = 0)
- In the initialization mode state following clearing of all transmission requests according to the transmission abort processing described in *Figure 24-24 "Transmission abort processing (except normal operation mode with ABT)"* during the operation mode (Clear all the transmission requests according to the transmission abort processing described in *Figure 24-25 "Transmission abort processing (normal operation mode with ABT) - Repeat option for aborted message"* in the normal operation mode with ABT.)

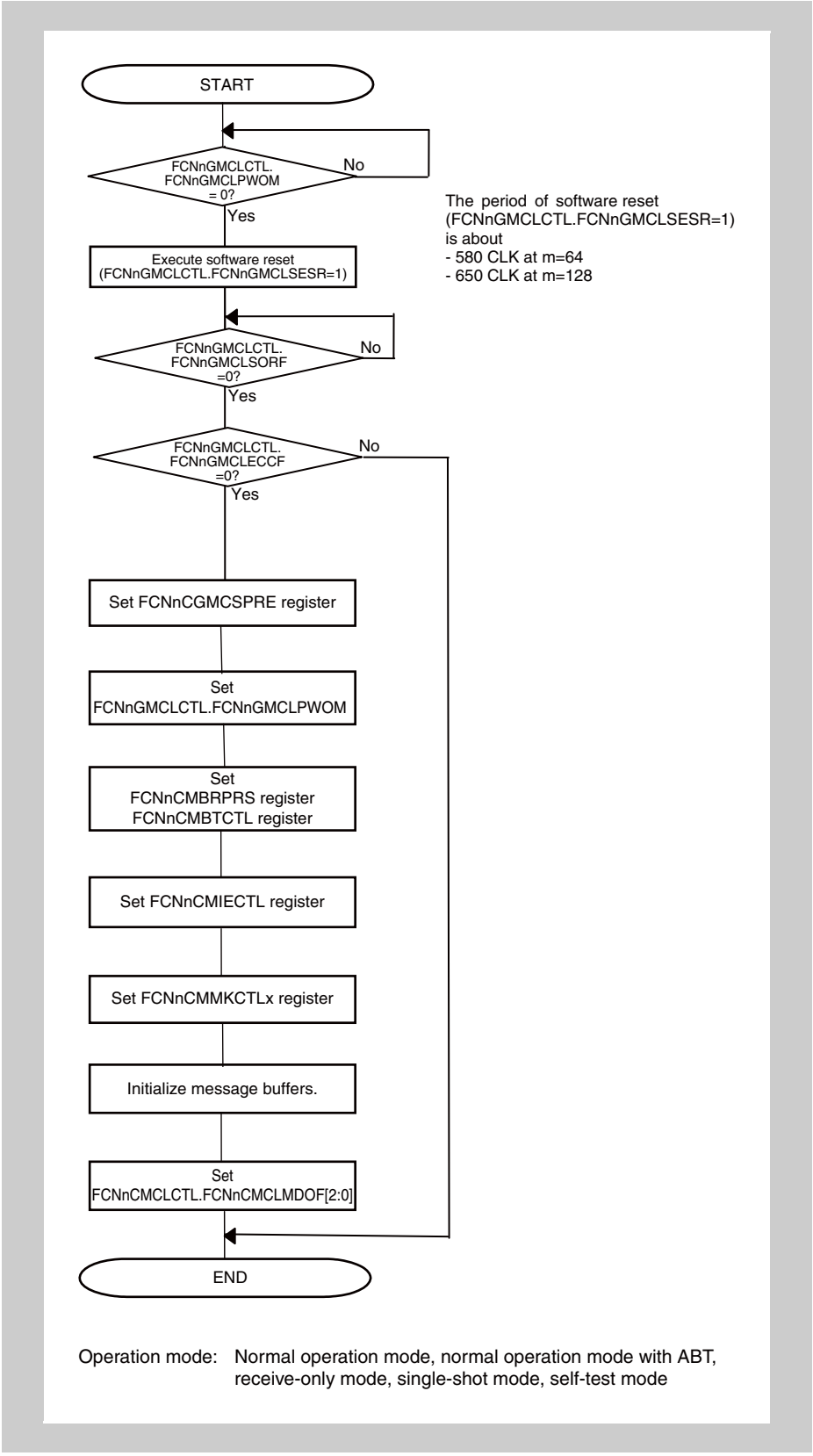


Figure 24-15 Re-initialization with Software Reset function

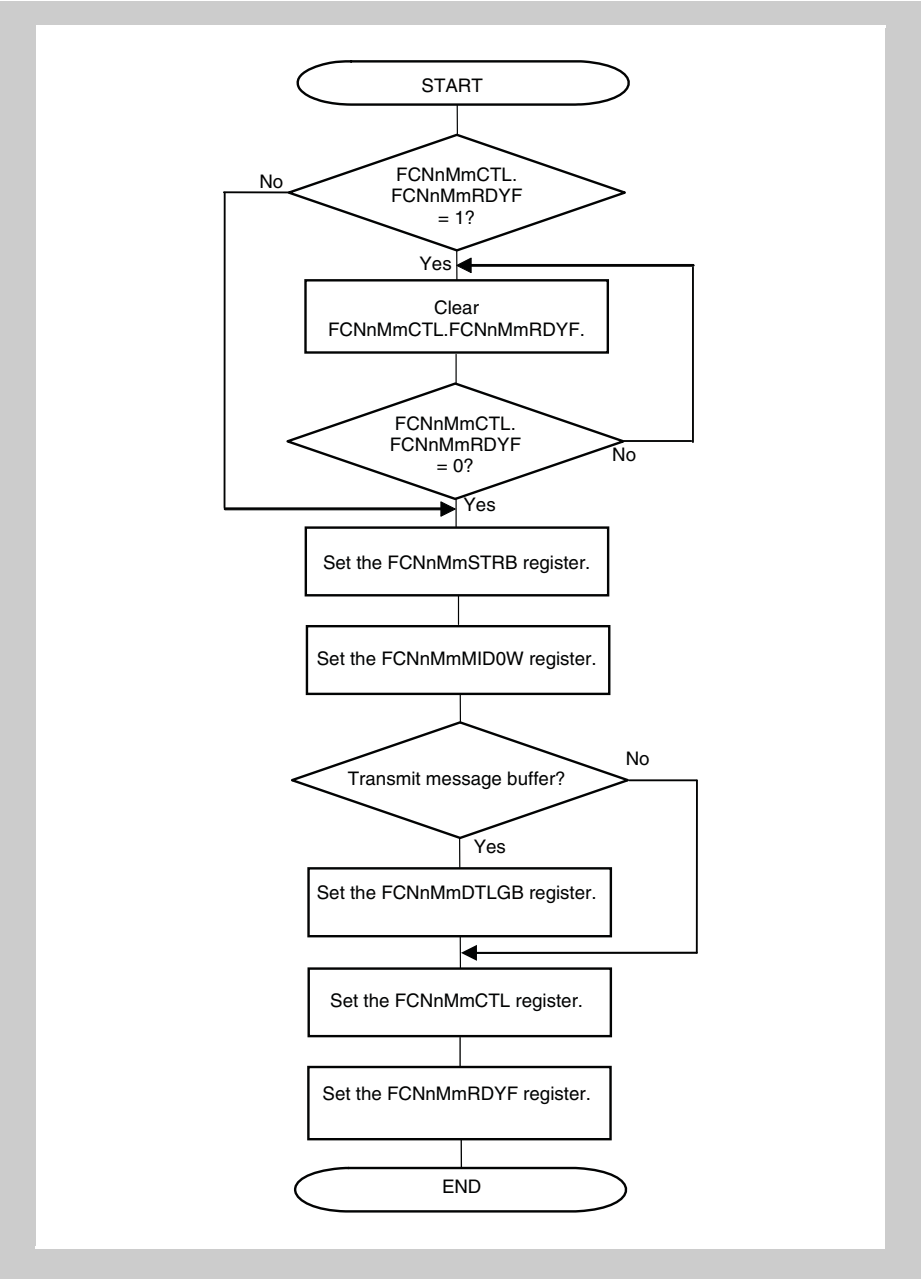
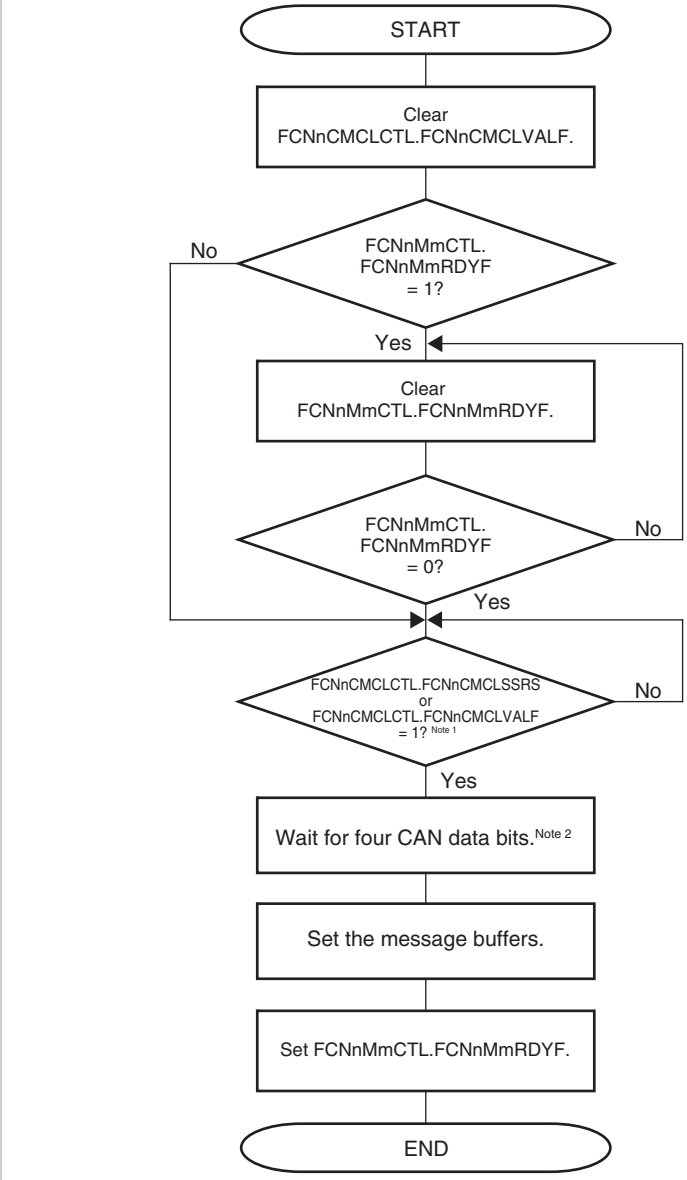


Figure 24-16 Message buffer initialization

Cautions

1. Before a message buffer is initialized, FCNnMmCTL.FCNnMmRDYF must be cleared.
2. Make the following settings for message buffers not used by the application.
   - Clear FCNnMmRDYF, FCNnMmTRQF, and FCNnMmDTNF bits of the FCNnMmCTL register to 0.
   - Clear FCNnMmSTRB.FCNnMmSSAM to 0.

Figure 24-17 "Message buffer redefinition during reception" shows the processing for a receive message buffer (FCNnMmSTRB.FCNnMmSSMT[3:0] = 0001_B to 1000_B).



Notes 1. Confirm that a message is being received. It is necessary to set FCNmMmCTL.FCNnMmRDFY after a message is completely received.

2. Wait for four CAN data bits so that message buffers are not redefined while saving the received messages.

Figure 24-17 Message buffer redefinition during reception

Figure 24-18 "Message buffer redefinition during transmission" shows the processing for a transmit message buffer during transmission (FCNnMmSTRB.FCNnMmSSMT[3:0] = 0000_B).

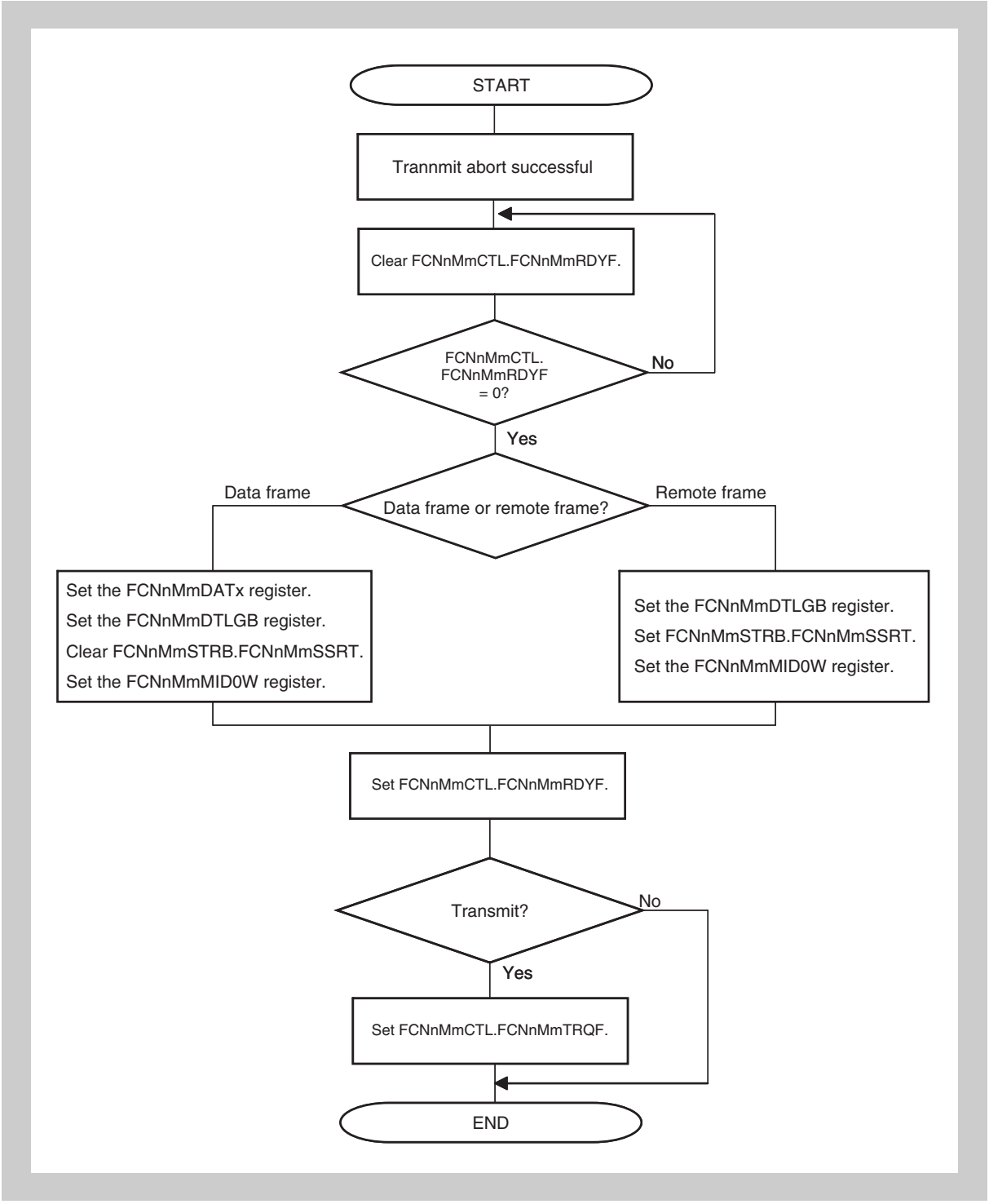


Figure 24-18 Message buffer redefinition during transmission

24.14.2 Message transmission

Figure 24-19 "Message transmit processing" shows the processing for a transmit message buffer (FCNnMmSTRB.FCNnMmSSMT[3:0] = 0000_B).

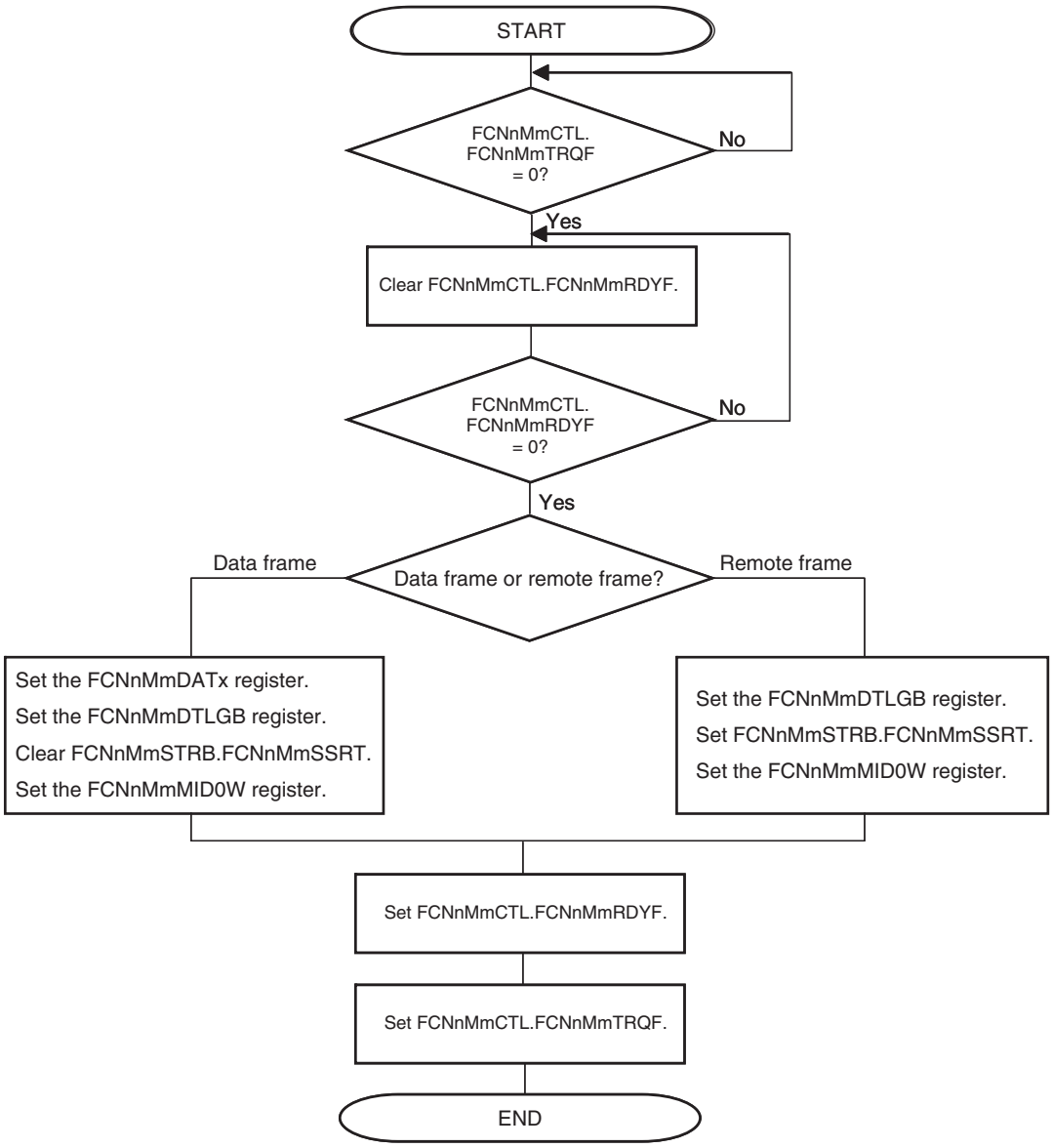


Figure 24-19 Message transmit processing

Cautions

1. FCNnMmCTL.FCNnMmTRQF should be set after FCNnMmCTL.FCNnMmRDYF is set.
2. FCNnMmCTL.FCNnMmRDYF and FCNnMmCTL.FCNnMmTRQF should not be set at the same time.

Figure 24-20 "ABT message transmit processing" shows the processing for a transmit message buffer (FCNnMmSTRB.FCNnMmSSMT[3:0] = 0000_B)

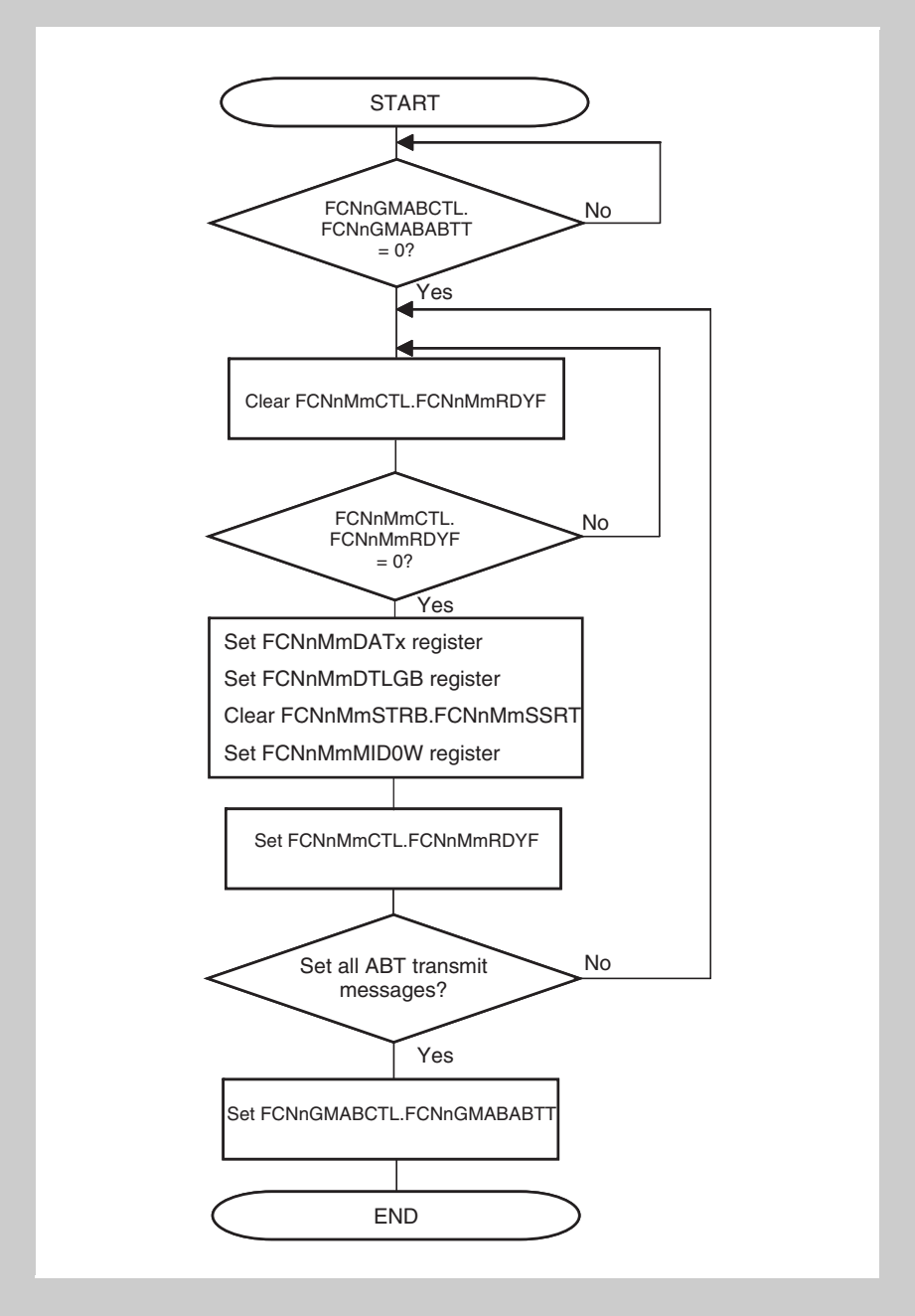


Figure 24-20 ABT message transmit processing

Note This processing (normal operation mode with ABT) can only be applied to message buffers usable with ABT mode. For message buffers other than the ABT message buffers, see *Figure 24-19 "Message transmit processing" on page 1660* .

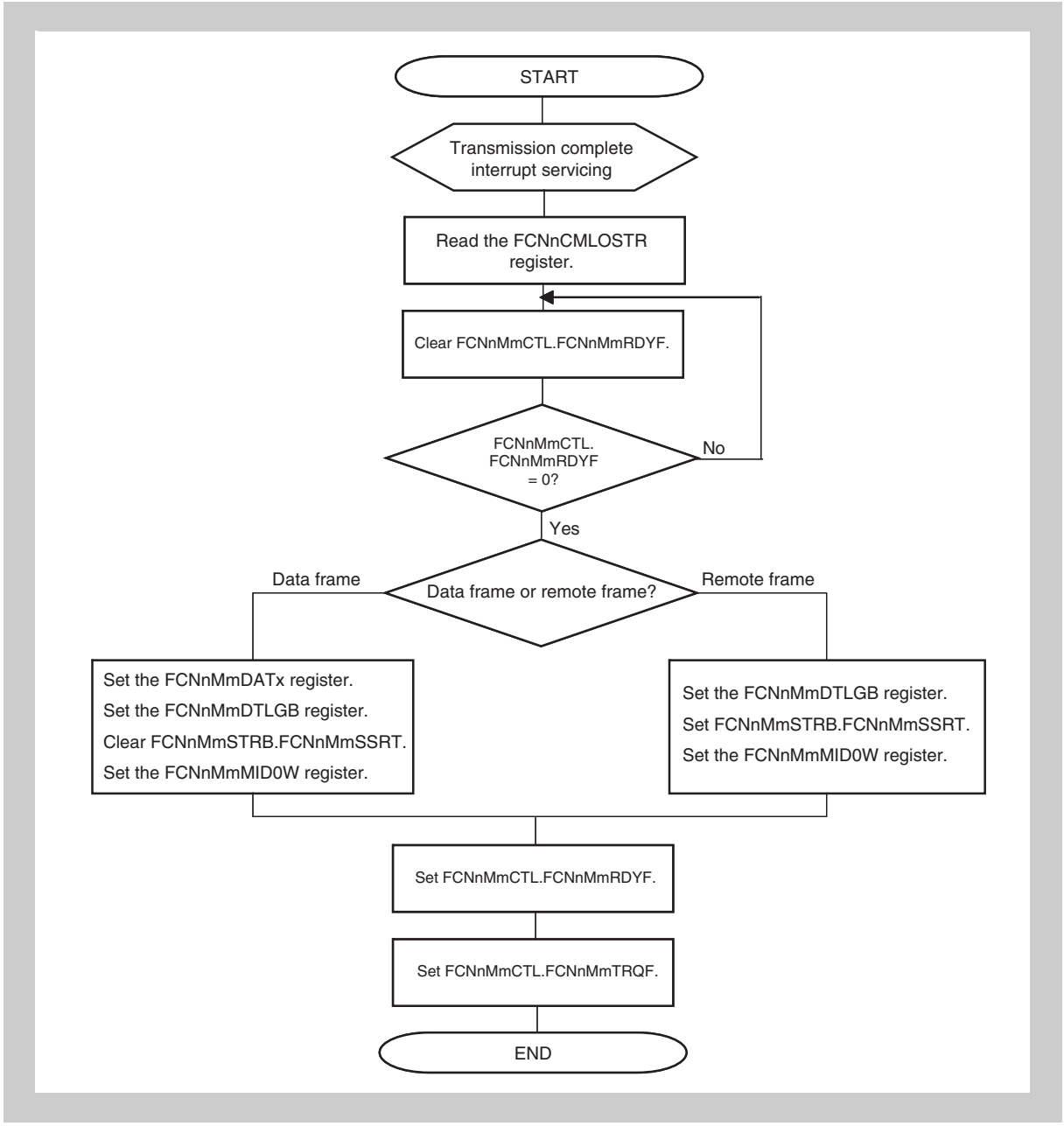


Figure 24-21 Transmission via interrupt (using FCNnCMLOSTR register)

Cautions

1. FCNnMmCTL.FCNnMmTRQF should be set after FCNnMmCTL.FCNnMmRDYF is set.
2. FCNnMmCTL.FCNnMmRDYF and FCNnMmCTL.FCNnMmTRQF should not be set at the same time.

Note Also check the FCNnGMCLSSMO flag at the beginning and at the end of the interrupt routine, in order to check the access to the message buffers as well as TX history list registers, in case a pending sleep mode had been executed. If FCNnGMCLSSMO is detected to be cleared at any check, the actions and

results of the processing have to be discarded and processed again, after FCNnGMCLSSMO is set again.
It is recommended to cancel any sleep mode requests, before processing TX interrupts.

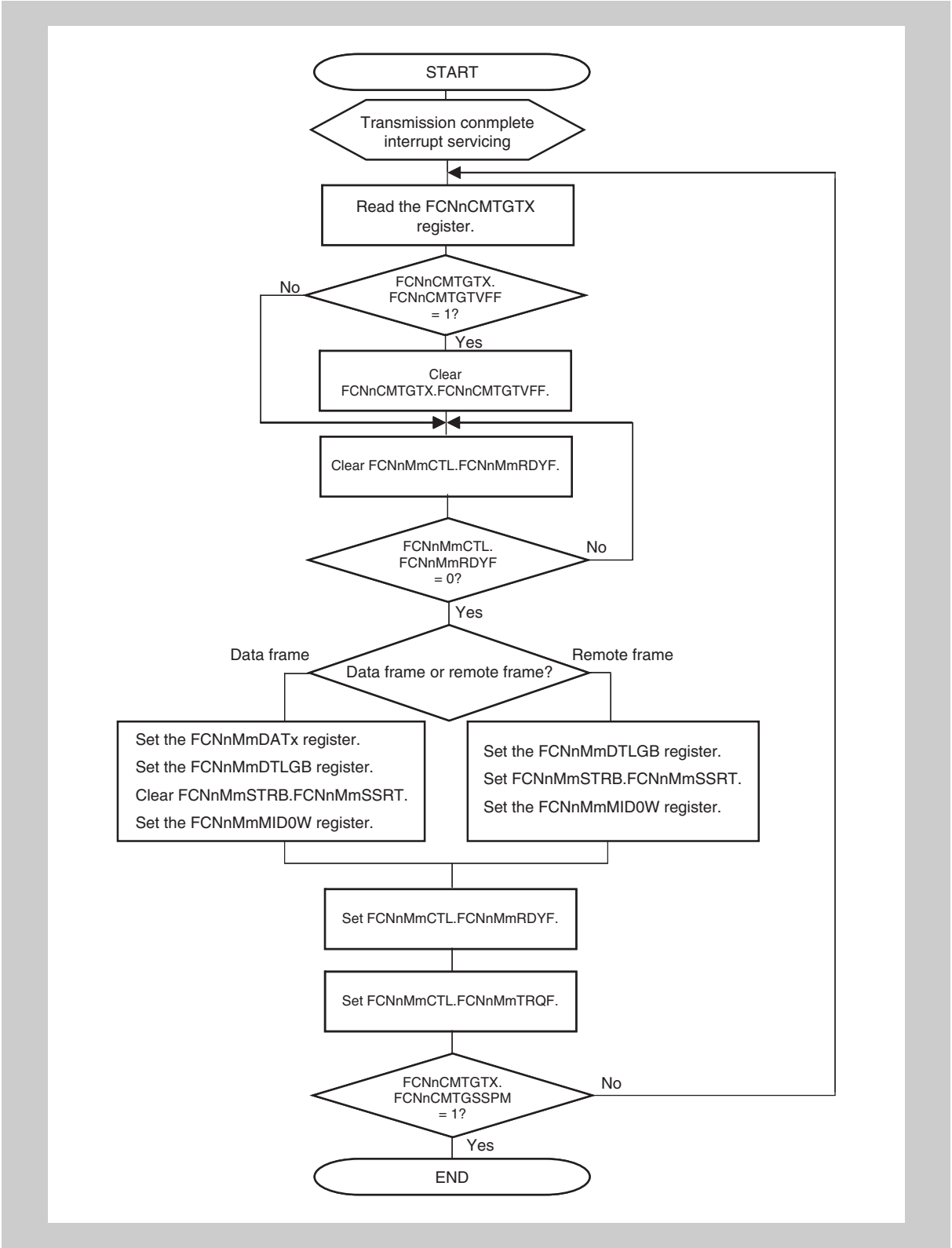


Figure 24-22 Transmission via interrupt (using FCNnCMTGTX register)

Cautions

1. FCNnMmCTL.FCNnMmTRQF should be set after FCNnMmCTL.FCNnMmRDYF is set.
2. FCNnMmCTL.FCNnMmRDYF and FCNnMmCTL.FCNnMmTRQF should not be set at the same time.

Notes

1. Also check the FCNnGMCLSSMO flag at the beginning and at the end of the interrupt routine, in order to check the access to the message buffers as well as TX history list registers, in case a pending sleep mode had been executed. If FCNnGMCLSSMO is detected to be cleared at any check, the actions and results of the processing have to be discarded and processed again, after FCNnGMCLSSMO is set again.
   It is recommended to cancel any sleep mode requests, before processing TX interrupts.
2. If FCNnCMTGTX.FCNnCMTGTVFF was set once, the transmit history list is inconsistent. Consider to scan all configured transmit buffers for completed transmissions.

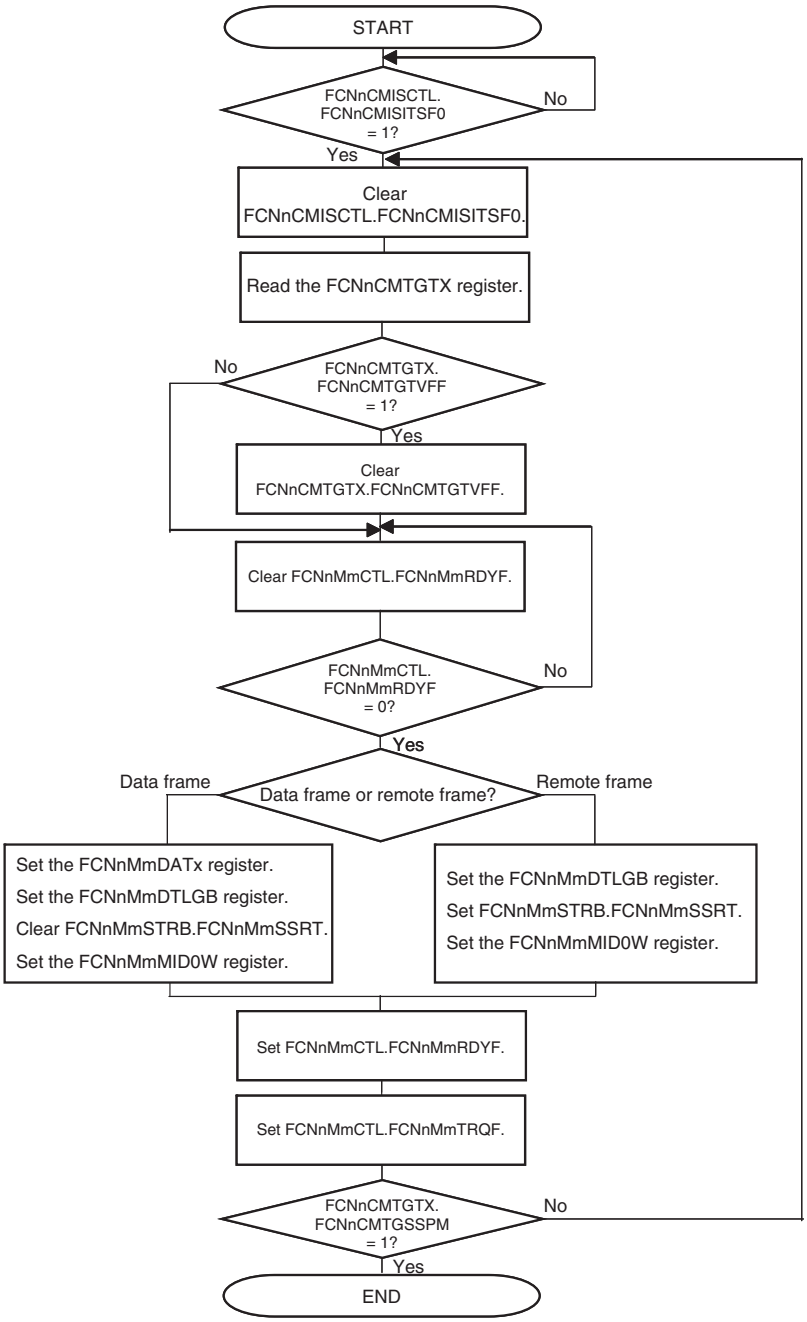


Figure 24-23 Transmission via software polling

Cautions

1. FCNnMmCTL.FCNnMmTRQF should be set after FCNnMmCTL.FCNnMmRDYF is set.
2. FCNnMmCTL.FCNnMmRDYF and FCNnMmCTL.FCNnMmTRQF should not be set at the same time.

Notes

1. Also check the FCNnGMCLSSMO flag at the beginning and at the end of the polling routine, in order to check the access to the message buffers as well as TX history list registers, in case a pending sleep mode had been executed. If FCNnGMCLSSMO is detected to be cleared at any check, the actions and results of the processing have to be discarded and processed again, after FCNnGMCLSSMO is set again.
2. If FCNnCMTGTX.FCNnCMTGTVFF was set once, the transmit history list is inconsistent. Consider to scan all configured transmit buffers for completed transmissions.

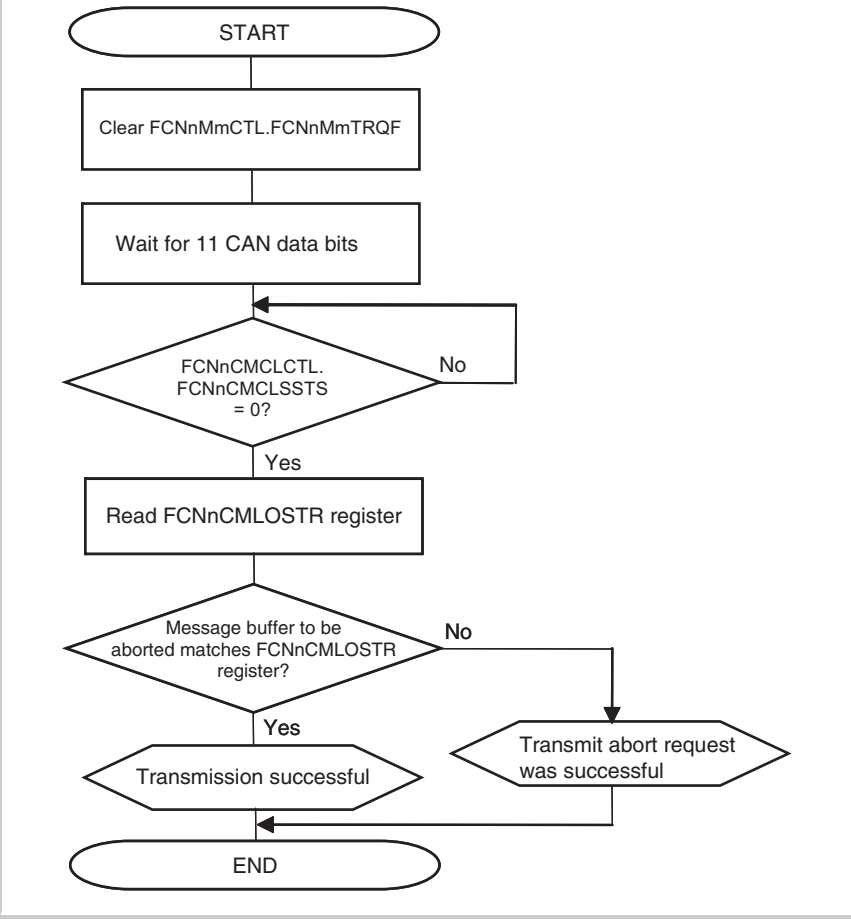


Figure 24-24 Transmission abort processing (except normal operation mode with ABT)

Cautions

1. Clear FCNnMmCTL.FCNnMmTRQF for aborting transmission request, not FCNnMmCTL.FCNnMmRDYF.
2. Before making a sleep mode transition request, confirm that there is no transmission request left using this processing.
3. FCNnCMCLCTL.FCNnCMCLSSTS can be periodically checked by a user application or can be checked after the transmit completion interrupt.
4. Do not execute any new transmission request including in the other message buffers while transmission abort processing is in progress.

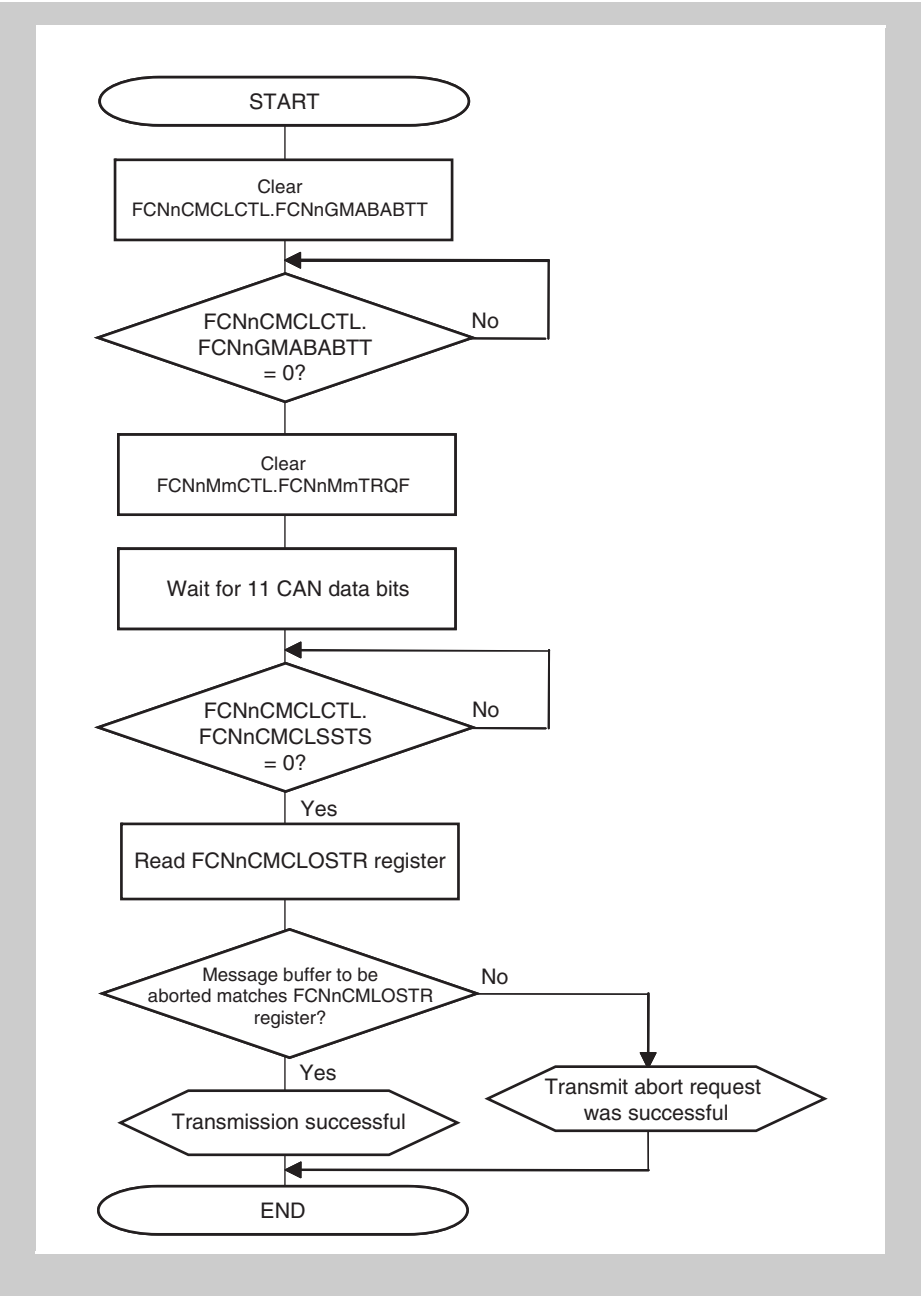


Figure 24-25 Transmission abort processing (normal operation mode with ABT) - Repeat option for aborted message

Cautions

1. Clear FCNnMmCTL.FCNnMmTRQF for aborting transmission request, not FCNnMmCTL.FCNnMmRDYF.
2. Before making a sleep mode transition request, confirm that there is no transmission request left using this processing.
3. FCNnCMCLCTL.FCNnCMCLSSTS can be periodically checked by a user application or can be checked after the transmit completion interrupt.
4. Do not execute any new transmission request including in the other message buffers while transmission abort processing is in progress.

Figure 24-26 "ABT transmission request abort processing (in normal operation mode with ABT) (1)" shows the processing to not skip resumption of transmitting a message that was stopped when transmission of an ABT message buffer was aborted.

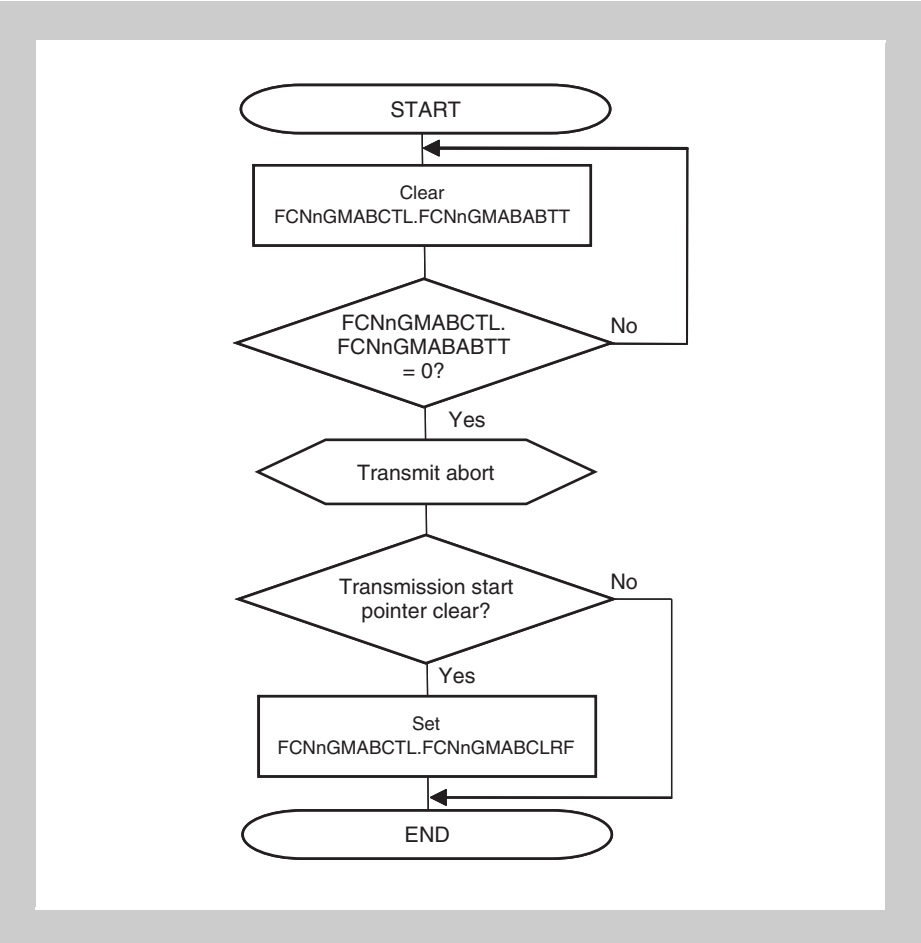


Figure 24-26 ABT transmission request abort processing (in normal operation mode with ABT) (1)

Cautions

1. Do not set any transmission requests while ABT transmission abort processing is in progress.
2. Make a FCN sleep mode/FCN stop mode request after FCNnGMABCTL.FCNnGMABABTT is cleared (after ABT mode is stopped) following the procedure shown in *Figure 24-26 "ABT transmission request abort processing (in normal operation mode with ABT) (1)"*. When clearing a transmission request in an area other than the ABT area, follow the procedure shown in *Figure 24-24 "Transmission abort processing (except normal operation mode with ABT)" on page 1666* .

Figure 24-27 "ABT transmission request abort processing (in normal operation mode with ABT) (2)" shows the processing to not skip resumption of transmitting a message that was stopped when transmission of an ABT message buffer was aborted.

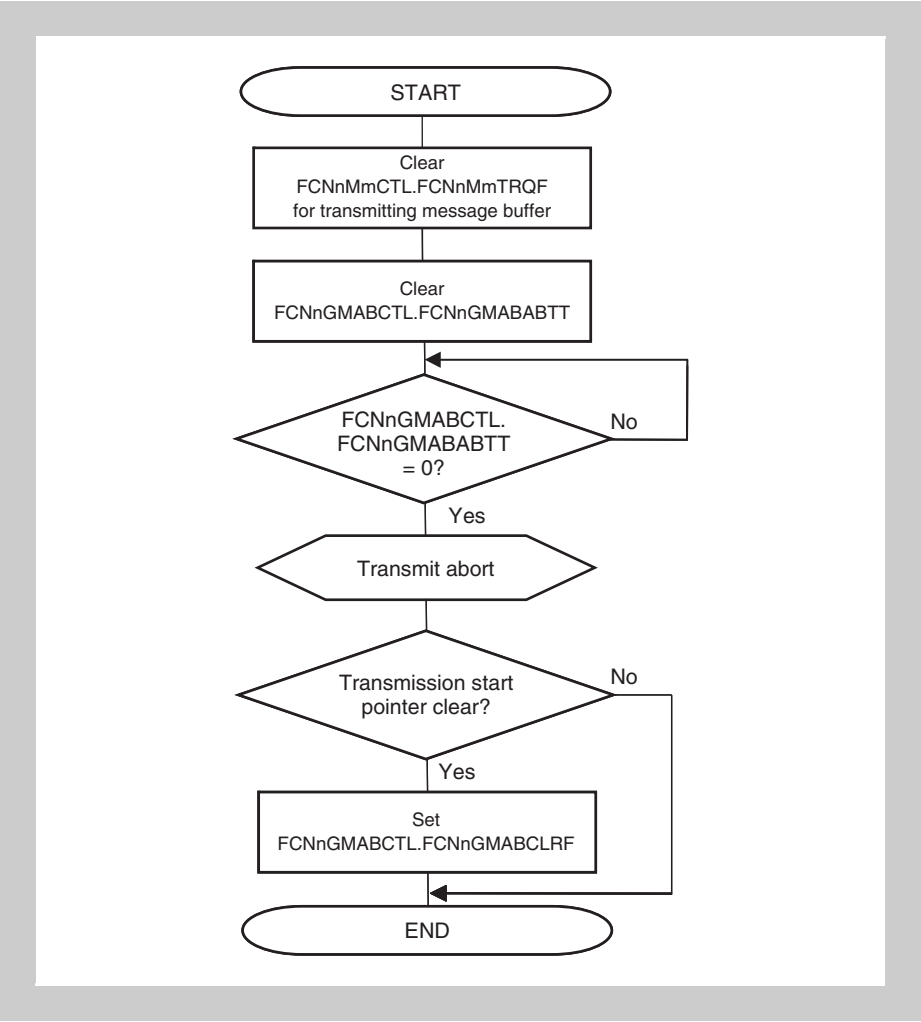


Figure 24-27 ABT transmission request abort processing (in normal operation mode with ABT) (2)

Cautions

1. Do not set any transmission requests while ABT transmission abort processing is in progress.
2. Make a FCN sleep mode/FCN stop mode request after FCNnGMABCTL.FCNnGMABABTT is cleared (after ABT mode is stopped) following the procedure shown in *Figure 24-26 "ABT transmission request abort processing (in normal operation mode with ABT) (1)"*. When clearing a transmission request in an area other than the ABT area, follow the procedure shown in *Figure 24-24 "Transmission abort processing (except normal operation mode with ABT)" on page 1666* .

Figure 24-28 “ABT transmission request abort processing (normal operation mode with ABT) with transmission completely finished flag” shows the processing on ABT mode, when using the Transmit Abort functionality (transmit flag). The box “Transmission abort success” represents the checking of the transmission abort success by checking the FCNnMmTCPF flag within the ABT message buffers.

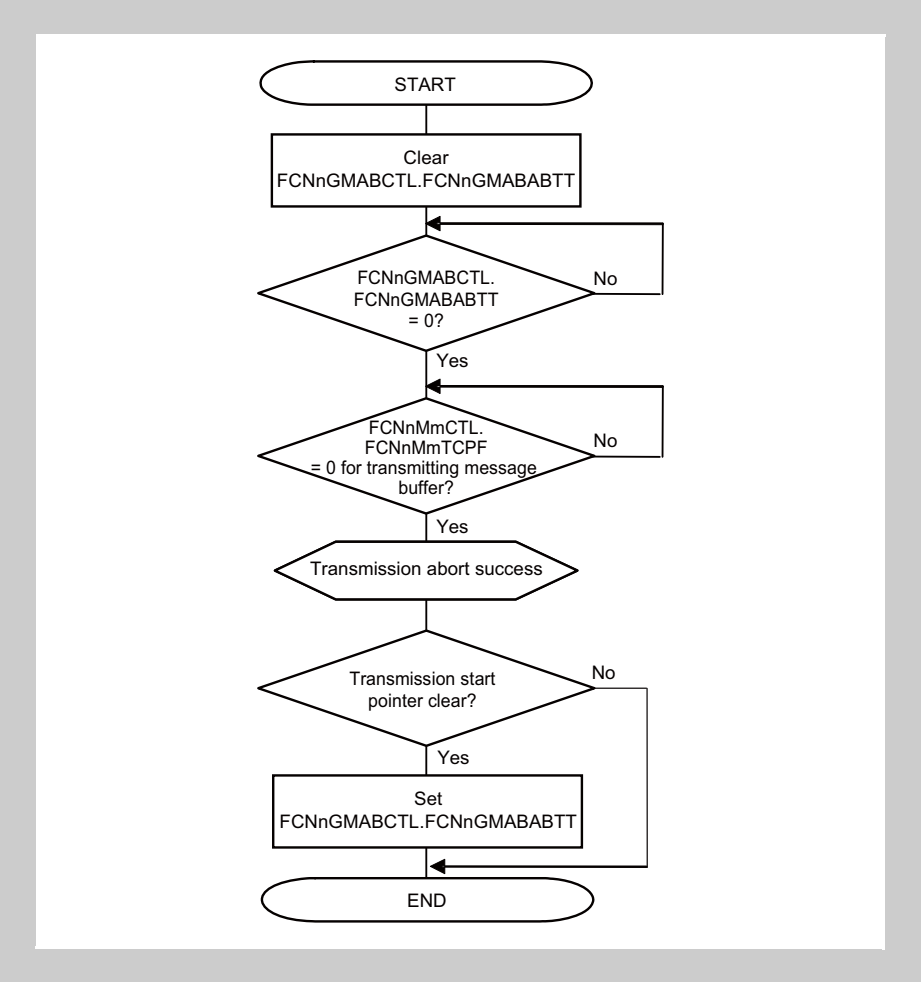


Figure 24-28 ABT transmission request abort processing (normal operation mode with ABT) with transmission completely finished flag

Cautions

1. Do not set any transmission requests while ABT transmission abort processing is in progress.
2. Make a FCN sleep mode/FCN stop mode request after FCNnGMABCTL.FCNnGMABABTT is cleared (after ABT mode is stopped) following the procedure shown in *Figure 24-26 “ABT transmission request abort processing (in normal operation mode with ABT) (1)”*. When clearing a transmission request in an area other than the ABT area, follow the procedure shown in *Figure 24-24 “Transmission abort processing (except normal operation mode with ABT)” on page 1666* .

Note There is the case that all ABT is transmitted completely even if ABT transmission abort processing is performed successfully. Then it is possible to know which message is finished transmission.

Figure 24-29 "Transmission request abort processing with transmit abort interrupt and transmission completely finished flag" shows the processing when using the Transmit Abort functionality (Transmit Abort Interrupt).

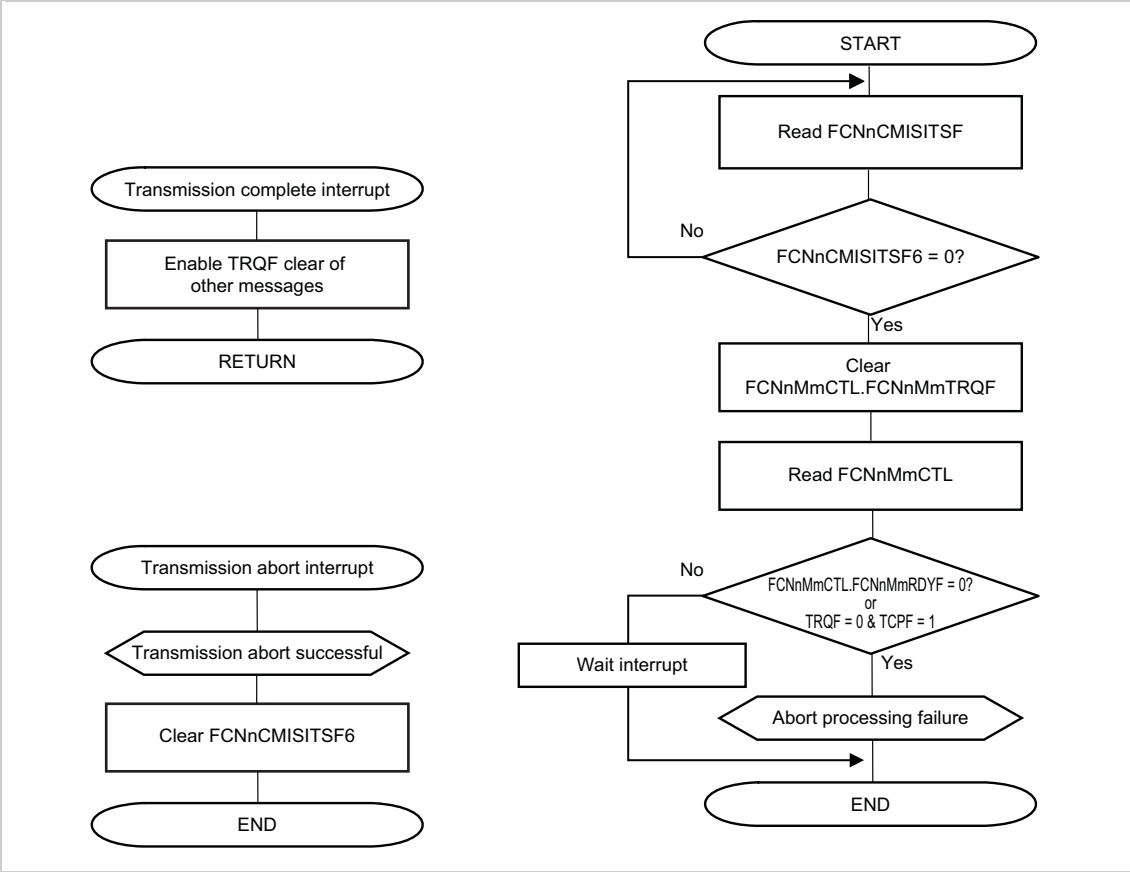


Figure 24-29 Transmission request abort processing with transmit abort interrupt and transmission completely finished flag

Note The determination that FCNnMmRDYF=0 is made with consideration for the case when FCNnMmRDYF is cleared during transmission completion processing due to an interrupt.

Cautions

1. Abort transmissions by clearing FCNnMmTRQF rather than by clearing FCNnMmRDYF.
2. Make sure that the transmission request made via this flow sequence has completely ended before sending a sleep request.
3. Do not update messages subject to transmission abort processing (FCNnMmRDYF or FCNnMmTRQF is set), for example via transmission complete interrupt processing.
4. Do not clear FCNnMmTRQF in other message buffers while performing transmission abort processing.
5. If you change the ID to one with lower priority during transmission abort processing, then wait for at least one frame after clearing FCNnMmTRQF before sending a transmission request.
6. Do not enable wake-up interrupts on RXONLY-CH if you use transmit abort interrupts. (DIAG version only)
7. Always read FCNnMmTRQF and FCNnMmTCPF at least once.

Figure 24-30 "Transmission request abort processing with transmission completely finished flag" shows the processing when using the Transmit Abort functionality (Transmission Completely Finished Flag FCNnMmTCPF).

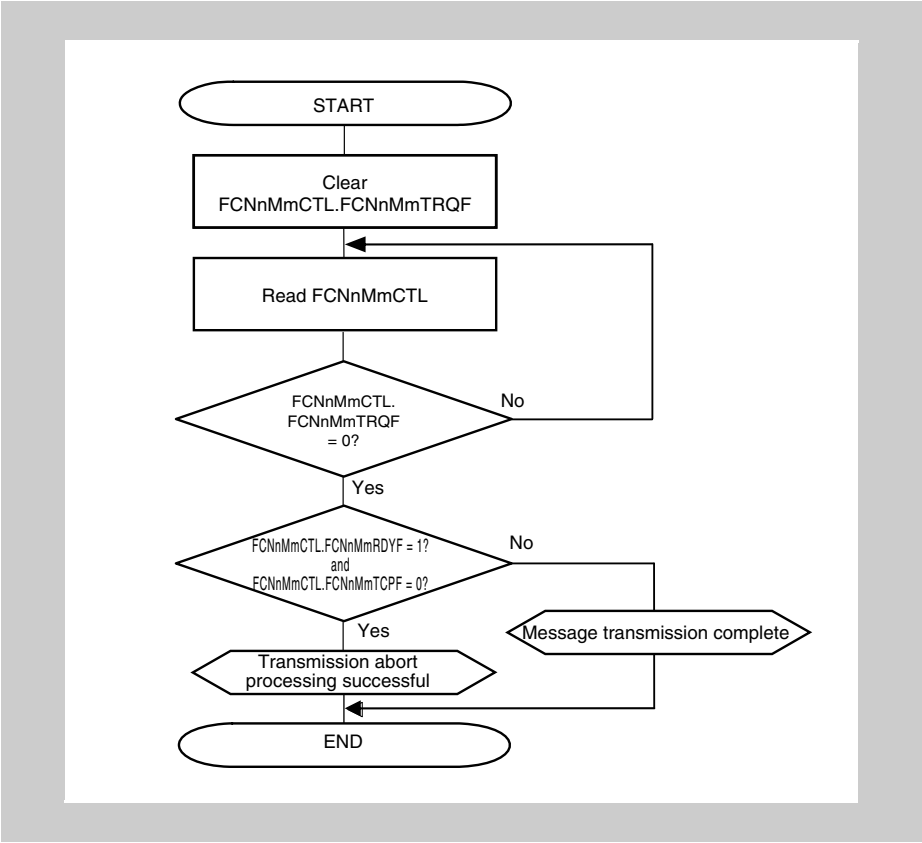


Figure 24-30 Transmission request abort processing with transmission completely finished flag

Note The determination that FCNnMmRDYF=0 is made with consideration for the case when FCNnMmRDYF is cleared during transmission completion processing due to an interrupt.

Cautions

1. Abort transmissions by clearing FCNnMmTRQF rather than by clearing FCNnMmRDYF.
2. Make sure that the transmission request made via this flow sequence has completely ended before sending a sleep request.
3. Do not update messages subject to transmission abort processing (FCNnMmRDYF or FCNnMmTRQF is set), for example via transmission complete interrupt processing.
4. If you change the ID to one with lower priority during transmission abort processing, then wait for at least one frame after clearing FCNnMmTRQF before sending a transmission request.
5. Always read FCNnMmTRQF and FCNnMmTCPF at least once.

24.14.3 Message reception

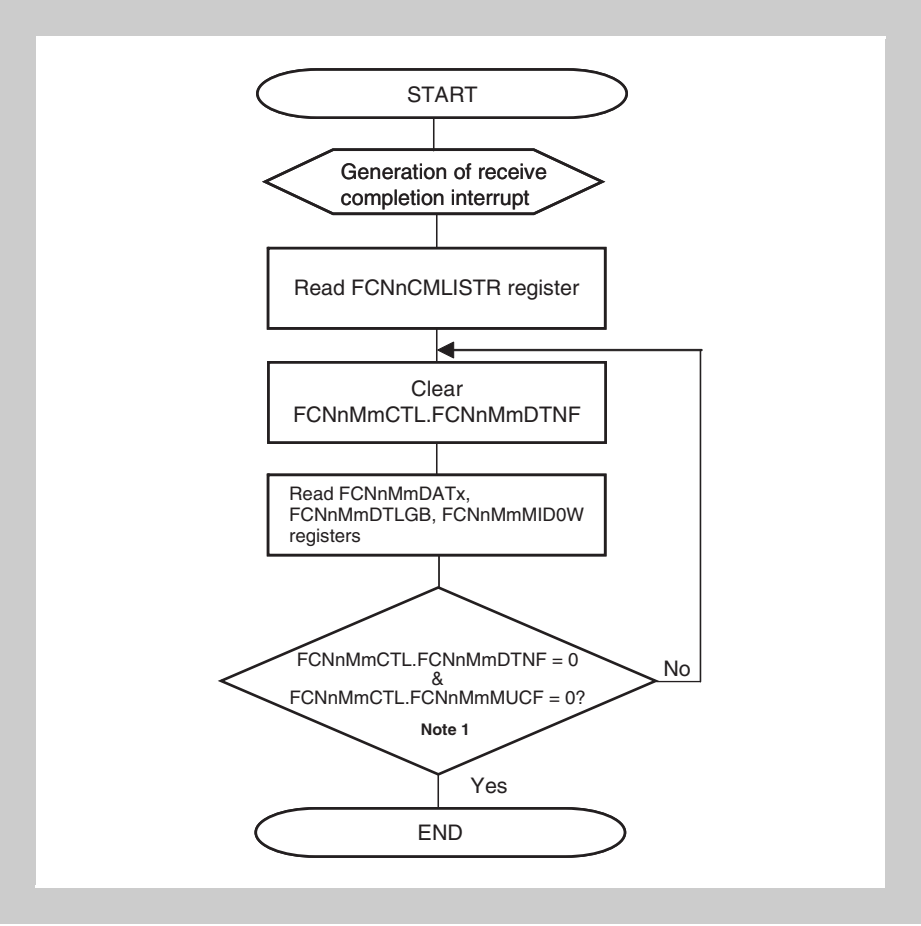


Figure 24-31 Reception via interrupt (using FCNnCMLISTR register)

Notes

1. Check FCNnMmCTL.FCNnMmMUCF and FCNnMmCTL.FCNnMmDTNF bits using one read access.
2. Also check the FCNnGMCLSSMO flag at the beginning and at the end of the interrupt routine, in order to check the access to the message buffers as well as reception history list registers, in case a pending sleep mode had been executed. If FCNnGMCLSSMO is detected to be cleared at any check, the actions and results of the processing have to be discarded and processed again, after FCNnGMCLSSMO is set again.
   It is recommended to cancel any sleep mode requests, before processing RX interrupts.

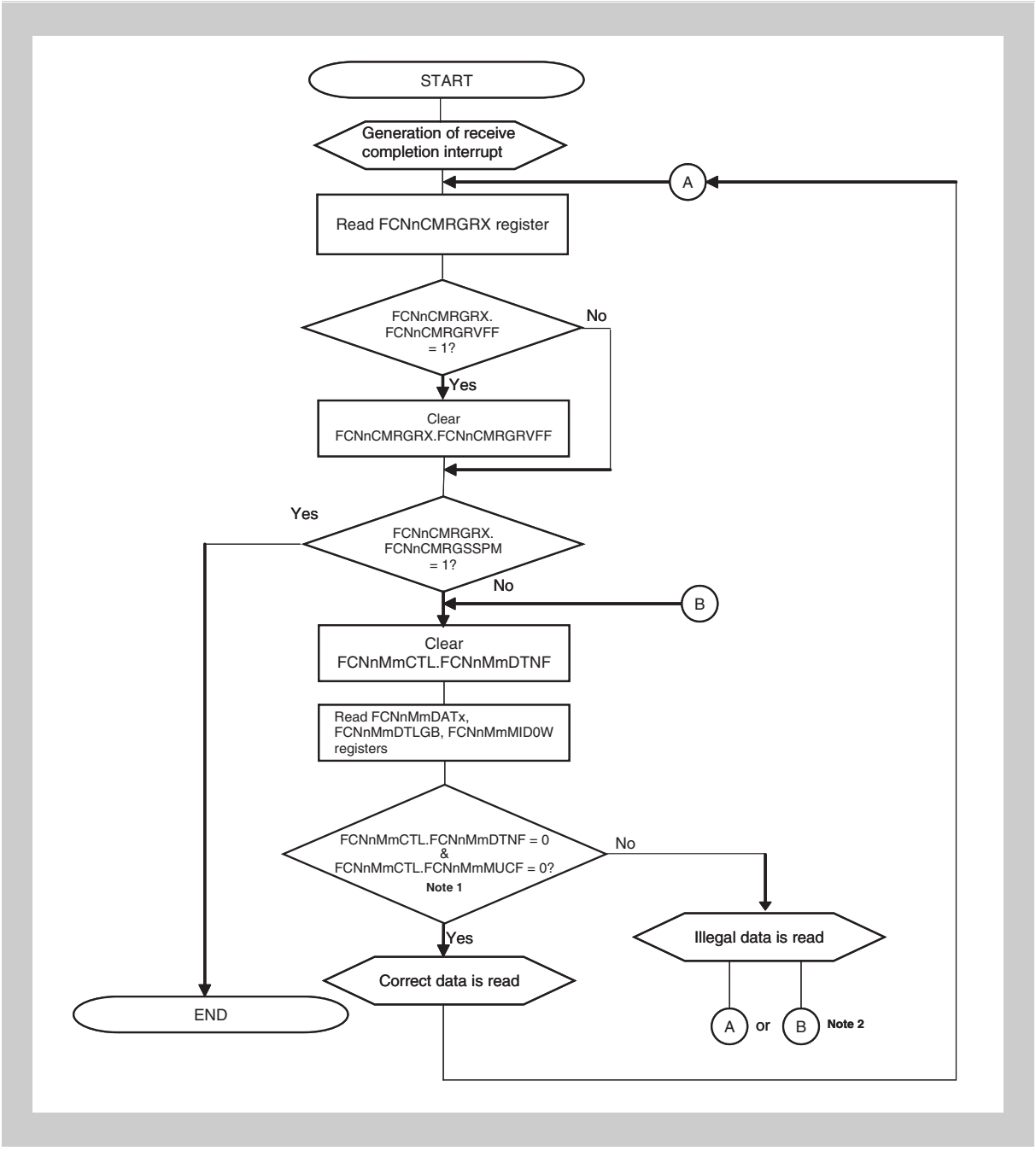


Figure 24-32 Reception via interrupt (using FCNnCMRGRX register)

Notes

1. Check FCNnMmCTL.FCNnMmMUCF and FCNnMmCTL.FCNnMmDTNF bits using one read access.
2. Depending of the processing target of the application, two ways are possible:
   –Way A: The message is not processed within this pass, but with the next pass, depending on the timing this can happen latest with the next Receive Interrupt.
   Other messages will be processed earlier.
   – Way B: The message is processed within this pass, the loop waits on this message.
   Other messages will be processed later.
3. Also check the FCNnGMCLSSMO flag at the beginning and at the end of the interrupt routine, in order to check the access to the message buffers as well as reception history list registers, in case a pending sleep mode had been executed. If FCNnGMCLSSMO is detected to be cleared at any check, the actions and results of the processing have to be discarded and processed again, after FCNnGMCLSSMO is set again.
   It is recommended to cancel any sleep mode requests, before processing RX interrupts.
4. If FCNnCMRGRX.FCNnCMRGRVFF was set once, the receive history list is inconsistent. Consider to scan all configured receive buffers for receptions.
5. For the processing shown in *Figure 24-32 “Reception via interrupt (using FCNnCMRGRX register)”*, the processing shown in *Figure 24-33 “Reception via interrupt (using FCNnCMRGRX register), alternative way”* can be used alternatively.

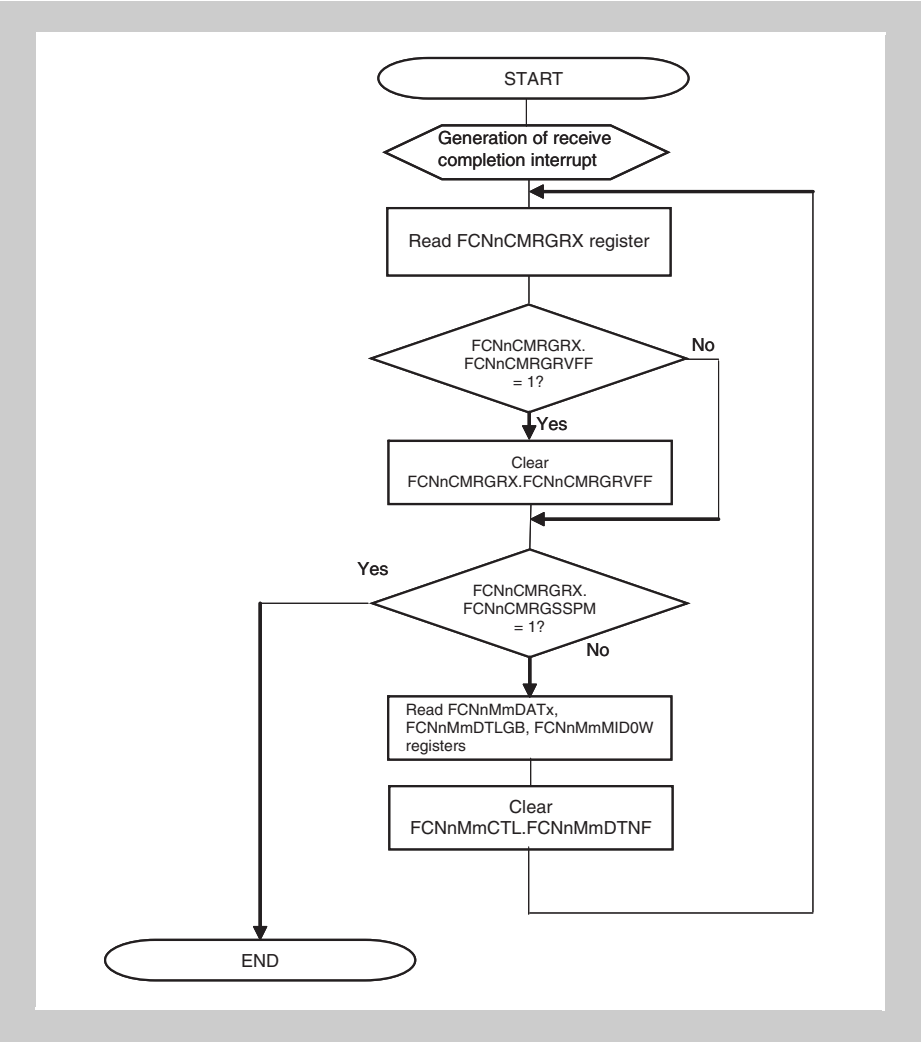


Figure 24-33 Reception via interrupt (using FCNnCMRGRX register), alternative way

Notes

1. Also check the FCNnGMCLSSMO flag at the beginning and at the end of the interrupt routine, in order to check the access to the message buffers as well as reception history list registers, in case a pending sleep mode had been executed. If FCNnGMCLSSMO is detected to be cleared at any check, the actions and results of the processing have to be discarded and processed again, after FCNnGMCLSSMO is set again.
   It is recommended to cancel any sleep mode requests, before processing RX interrupts.
2. If FCNnCMRGRX.FCNnCMRGRVFF was set once, the receive history list is inconsistent. Consider to scan all configured receive buffers for receptions.
3. This flow will not provide most recently received data for the application. However, due to less effort on processing, it reduces interrupt load.
4. The overwrite function (FCNnMmSTRB.FCNnMmSSOW=1) must not be used with this flow - data inconsistency could occur.
5.

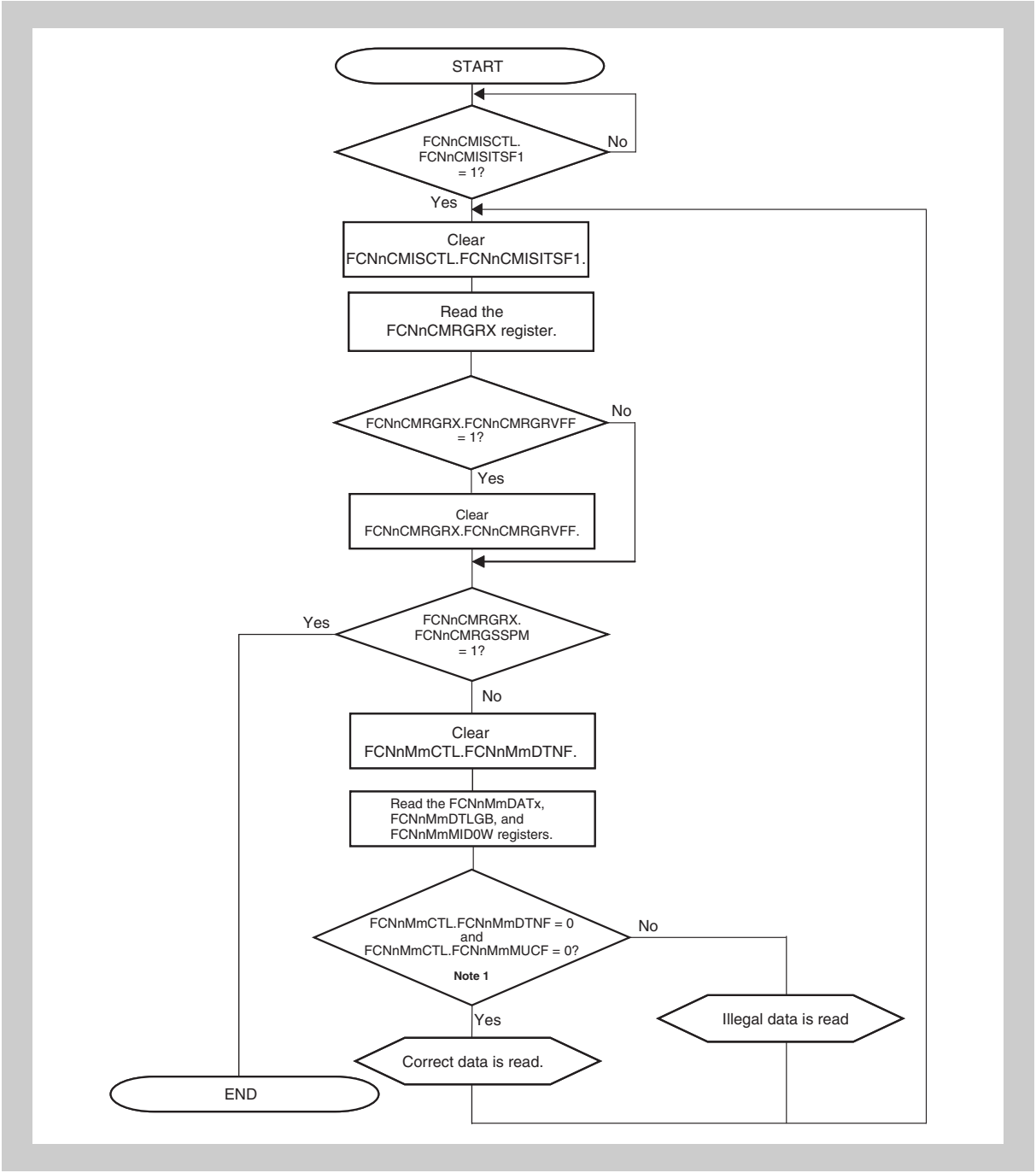


Figure 24-34 Reception via software polling

Notes

1. Check FCNnMmCTL.FCNnMmMUCF and FCNnMmCTL.FCNnMmDTNF bits using one read access.
2. Also check the FCNnGMCLSSMO flag at the beginning and at the end of the polling routine, in order to check the access to the message buffers as well as reception history list registers, in case a pending sleep mode had been executed. If FCNnGMCLSSMO is detected to be cleared at any check, the actions and results of the processing have to be discarded and processed again, after FCNnGMCLSSMO is set again.
3. If FCNnCMRGRX.FCNnCMRGRVFF was set once, the receive history list is inconsistent. Consider to scan all configured receive buffers for receptions.

24.14.4 Power save modes

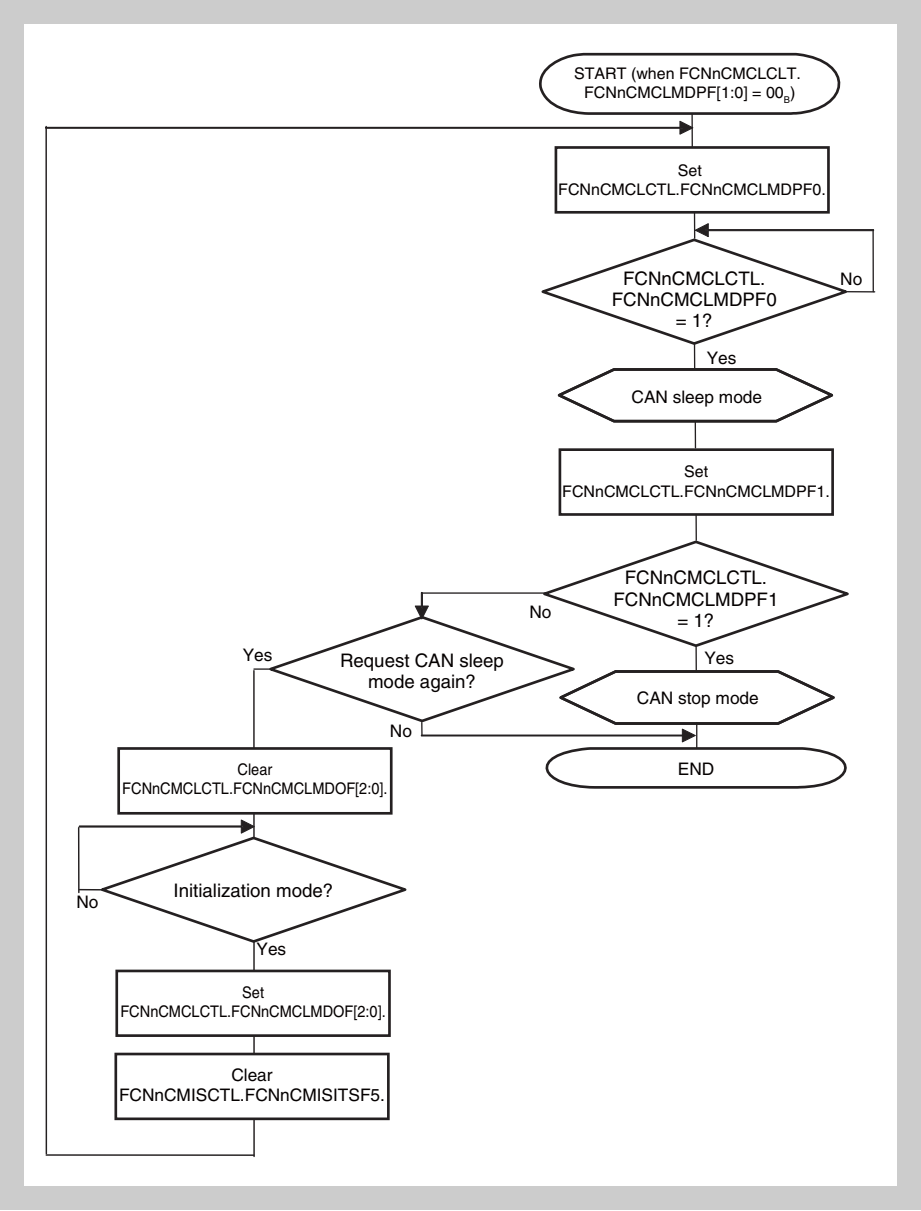


Figure 24-35 Setting FCN sleep mode/stop mode

Caution To abort transmission before making a request for the FCN sleep mode, perform processing according to previously given flowcharts.

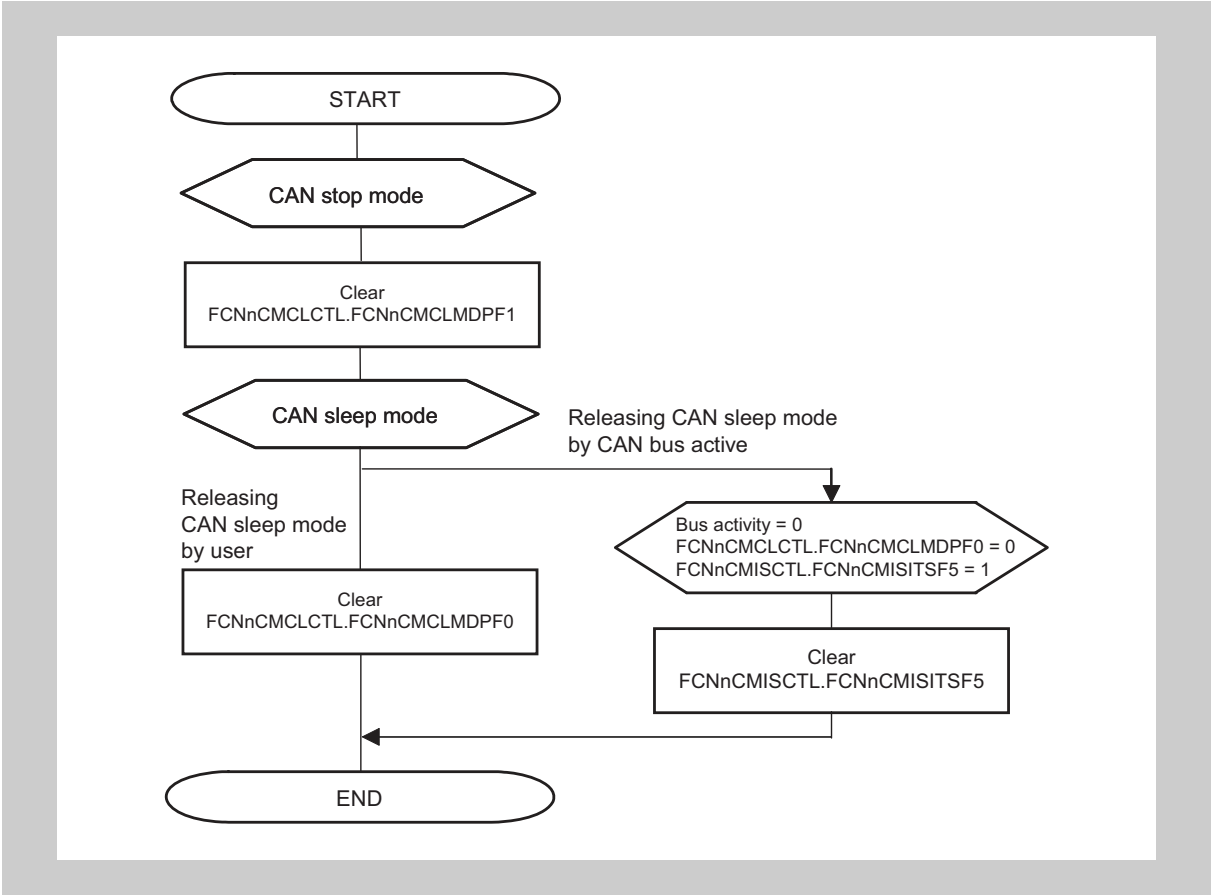


Figure 24-36 Clear FCN sleep/stop mode

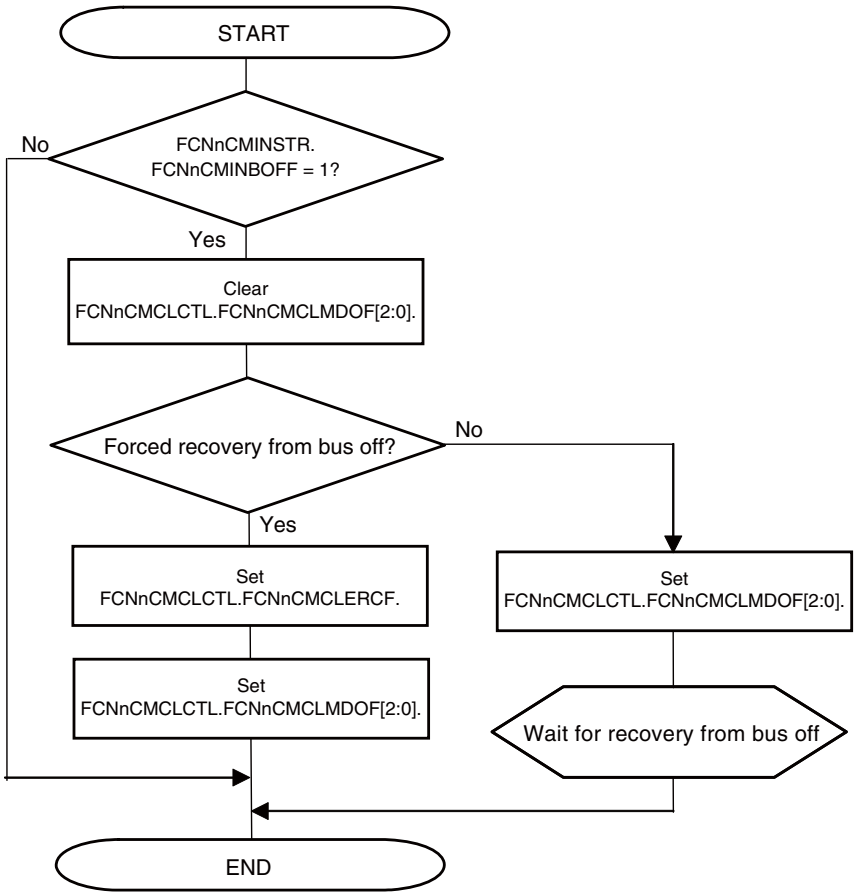


Figure 24-37 Bus-off recovery

Caution When the transmission from the initialization mode to any operation modes is requested to execute bus-off recovery sequence again in the bus-off recovery sequence, reception error counter is cleared.
Therefore it is necessary to detect 11 consecutive recessive-level bits 128 times on the bus again.

Note Operation mode: Normal operation mode, normal operation mode with ABT, receive-only mode, singleshot mode, self-test mode.

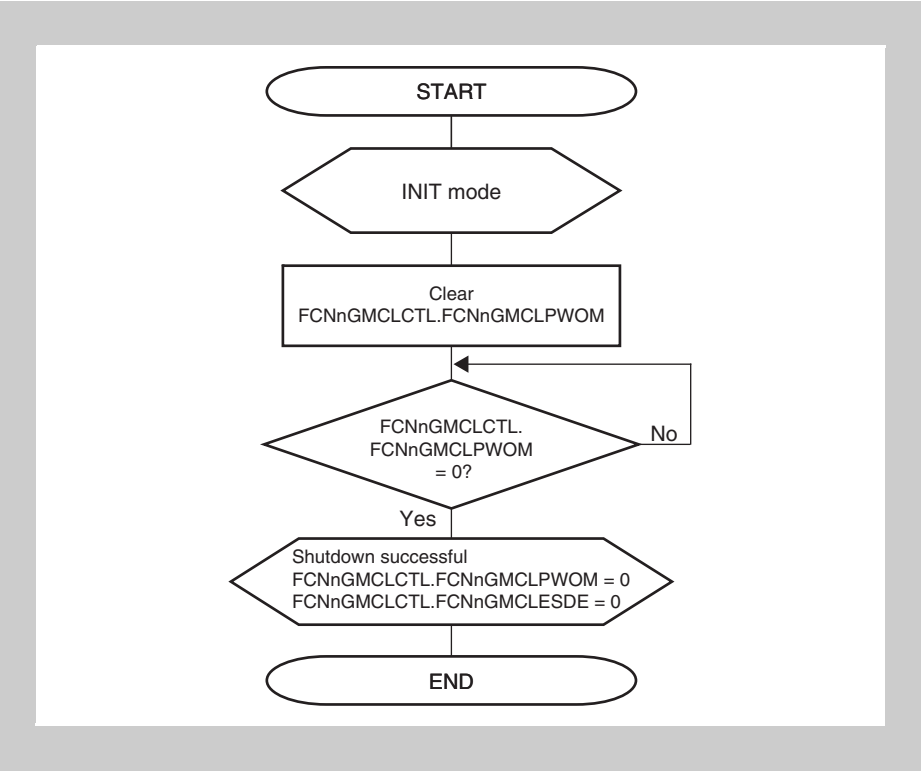


Figure 24-38 Normal shutdown process

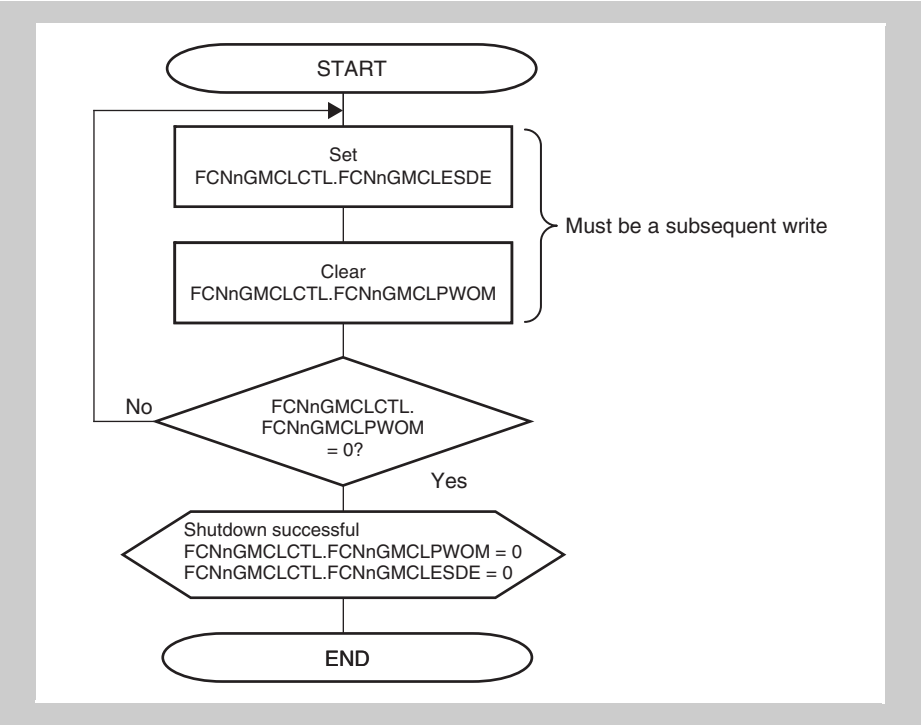


Figure 24-39 Forced shutdown process

Caution Do not read- or write-access any registers by software between setting the FCNnGMCLESDE bit and clearing the FCNnGMCLPWOM bit.

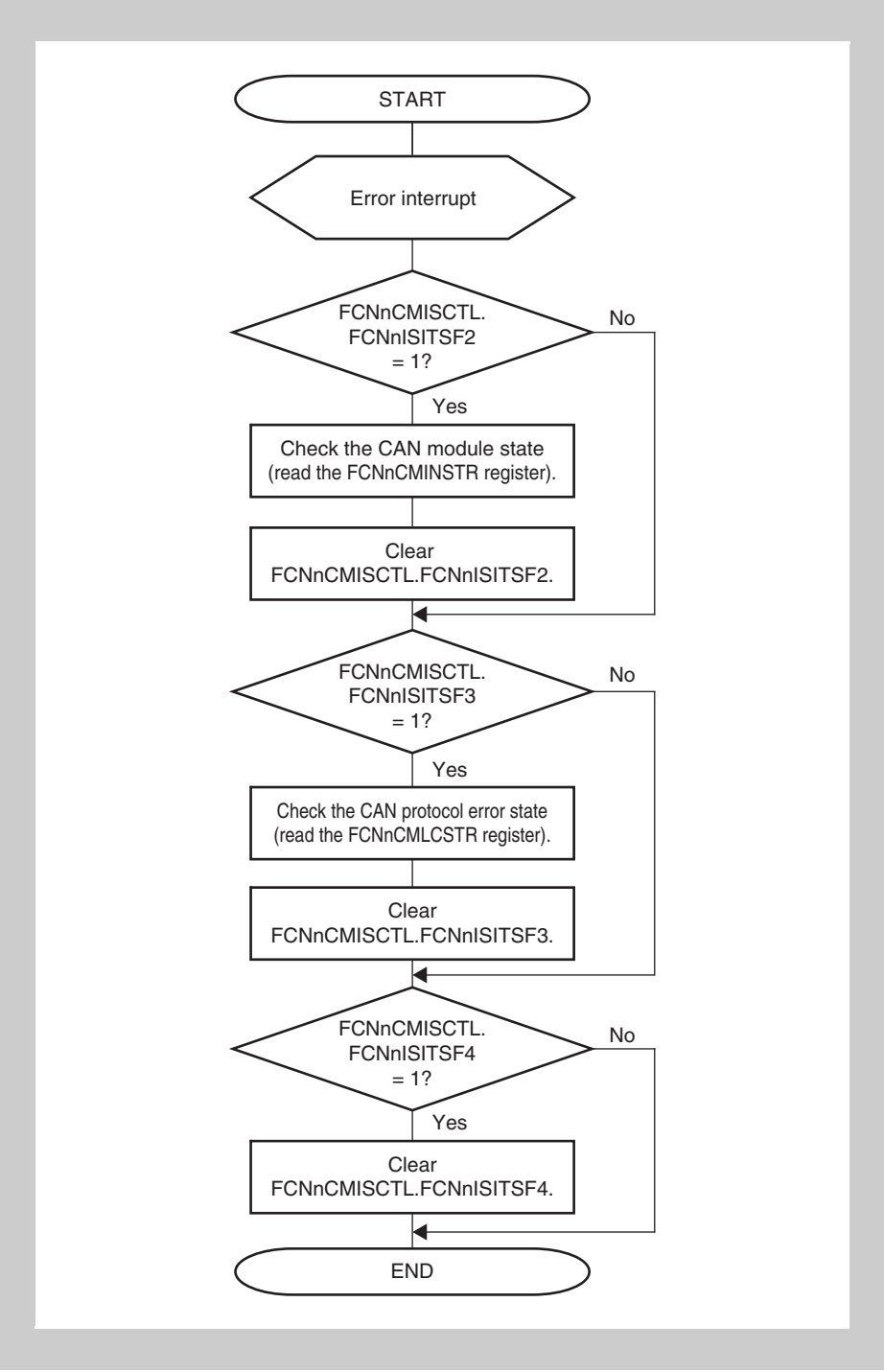


Figure 24-40 Error handling

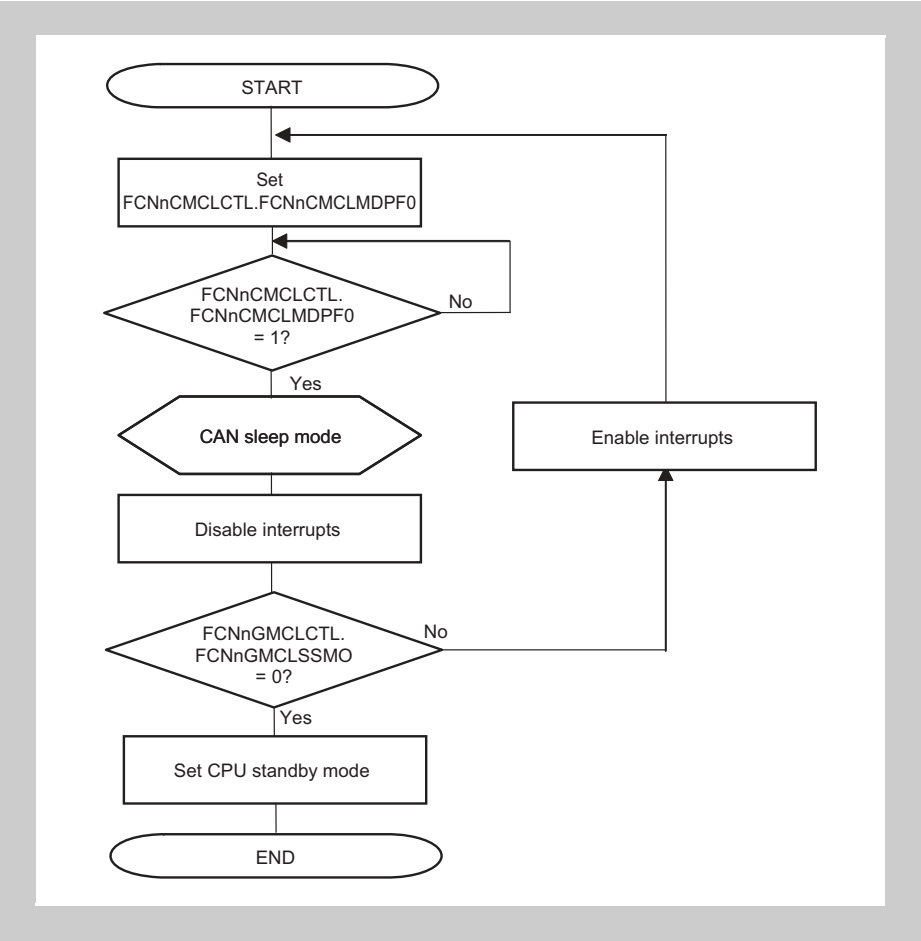


Figure 24-41 Setting CPU stand-by (from FCN sleep mode)

Notes

1. Before the CPU is set in the CPU standby mode, please check if the FCN sleep mode has been reached.
   However, after check of the FCN sleep mode, until the CPU is set in the CPU standby mode, the FCN sleep mode may be cancelled by wakeup from CAN bus.
2. There is a possibility, that between the check of FCNnGMCLSSMO = 0 and setting of the CPU standby mode a wake up condition on the CAN bus occurs. In that case the CAN module releases the SLEEP mode, the FCNnCMISITSF5 bit is set and if enabled the wake up interrupt will be generated.

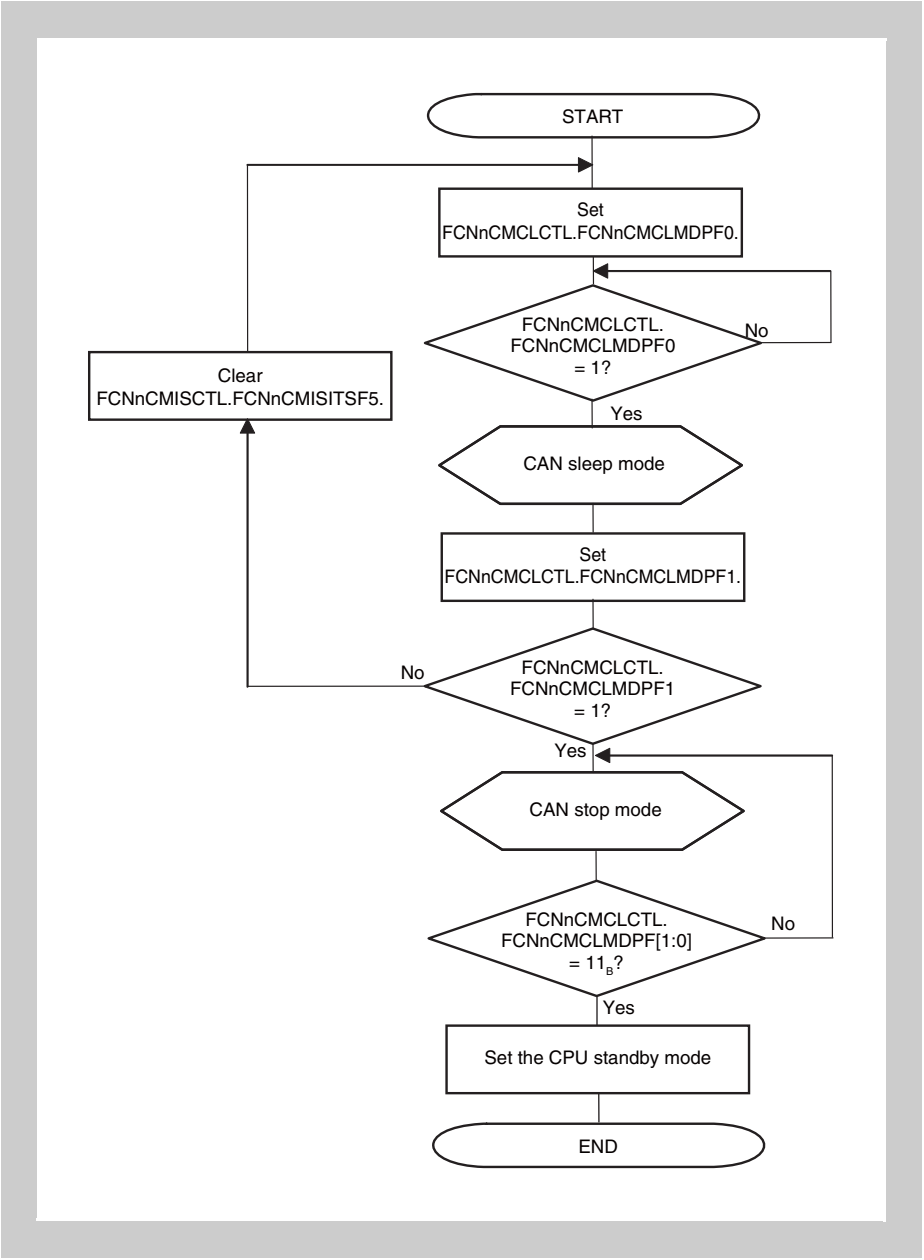


Figure 24-42 Setting CPU stand-by (from FCN stop mode)

Caution The FCN stop mode can only be released by setting FCNnCMCLCTL.FCNnCMCLMDPF[1:0] to 01_B and not by a change in the FCN bus state.

Chapter 25 A/D Converter (ADCA)

This chapter provides a general description of the A/D converter (ADCA).

25.1 "V850E2/MN4 Features" describes the features specific to the V850E2/MN4, such as the number of channels, base addresses of registers, and I/O signal names.

25.2 "Function Overview" and the sections that follow describe the features common to all products that have ADCA.

25.1 V850E2/MN4 Features

The V850E2/MN4 incorporates an A/D converter (ADCA).

The number of channels and other specifications vary depending on the product. The following table shows the specifications of the V850E2/MN4:

ADC \ Product		V850E2/MN4
ADC-equipped unit (indicated by n)		1 (n = 0)
Analog input pins (indicated by m)		12 (m = 0 to 11)
Channel groups (indicated by i)		3 (i = 0 to 2)
Register base address	<ADCAn_base_OS>[a]	FF81 D000$_H$
	<ADCAn_base_USER>[a]	FFFF DC00$_H$
Resolution		10/12[b] bits
Discharge function		None
	ADCAnCTL1.ADCAnDISC	Not supported. Be sure to write “0” to this bit.
Hardware trigger expansion function		Available
Channel S/H functions (indicated by x)[c]		x = 0 to 5
Self-diagnosis functions		None
	ADCAnCG0.ADCAnDIAG	Not supported. Be sure to write “0” to this bit.
	ADCAnIOC0.ADCAnIDG	
	ADCAnDGCTL0 ADCAnDGCTL1 ADCAnPDCTL0 ADCAnDGCR	Access prohibited
Supply clock (PCLK)		f_{PCLK}
ADCAnCNVi signals		Available (i = 0 to 2)
Stand-by mode of product		Not supported.
ADCHALT modes		Not supported.
	ADCAnSTR2.ADCAnRQ3	Fixed to 0.
	ADCAnSTR2.ADCAnST3	Fixed to 0.
	ADCAnTRG3	Not supported.
	ADCAnTRG7	Not supported.
Sampling and conversion times/sampling mode		A (see *Table 25-3 “Sampling and Conversion Times”*)
ADCAnCTL1 register ADCAnTRMi		Available
Stabilization time setting (ADCAnCNT)		1 (μs) (target)

a) All register addresses related to the A/D converter described in this chapter are defined as offsets from the above base addresses.

b) To use 12-bit resolution for the V850E/MN4, supply 5.0 V to the A/D converter power.

c) Make sure to turn on the amplifier power when using the channel S/H circuit.

The following table shows what each signal is connected to:

Table 25-1 ADCA Interrupts and DMA Requests

Signal Name	Function	Connected to
INTADCA0T0	CG0 A/D conversion end interrupt	Interrupt controller 47 (INTADCA0I0) DMA controller trigger 28 DTS controller trigger 28
INTADCA0T1	CG1 A/D conversion end interrupt	Interrupt controller 48 (INTADCA0I1) DMA controller trigger 29 DTS controller trigger 29
INTADCA0T2	CG2 A/D conversion end interrupt	Interrupt controller 49 (INTADCA0I2) DMA controller trigger 30 DTS controller trigger 30
INTADCA0TLLT	Latest conversion latch timing signal	Not supported
INTADCA0TERR	Error interrupt	Interrupt controller 46 (INTADCA0ERR)

This microcontroller has a hardware trigger function. The following table shows what the CGi hardware trigger ADCA0TRGi is connected to:

Table 25-2 What the Hardware Trigger Is Connected to (1/2)

ADCAn Channel Group	Start Trigger Bit (ADCAnTSELi Register)	Trigger Input Signal			ADCAn Trigger Signal
		Name	Connected to		
			Unit	Signal	
CG0	ADCA0T0SEL00	ADCA0TTIN000	PIC	ADOPA0ADCATTIN00	ADCA0TTRG0
	ADCA0T0SEL01	ADCA0TTIN001	PIC	ADOPA0ADCATTIN01	
	ADCA0T0SEL02	ADCA0TTIN002	TAPA2	TAPATADOUT0	
	ADCA0T0SEL03	ADCA0TTIN003	TAPA3	TAPATADOUT0	
	ADCA0T0SEL04	ADCA0TTIN004	ENCA0	ENCATINT1	
	ADCA0T0SEL05	ADCA0TTIN005	ENCA1	ENCATINT1	
	ADCA0T0SEL06	ADCA0TTIN006	Pin	ADTRG00	
	ADCA0T0SEL07	ADCA0TTIN007	Pin	ADTRG01	
	ADCA0T0SEL08	ADCA0TTIN008	Not connected	–	
	ADCA0T0SEL09	ADCA0TTIN009	Not connected	–	
	ADCA0T0SEL10	ADCA0TTIN010	Not connected	–	
	ADCA0T0SEL11	ADCA0TTIN011	Not connected	–	
	ADCA0T0SEL12	ADCA0TTIN012	Not connected	–	
	ADCA0T0SEL13	ADCA0TTIN013	Not connected	–	
	ADCA0T0SEL14	ADCA0TTIN014	Not connected	–	
	ADCA0T0SEL15	ADCA0TTIN015	Not connected	–	
CG1	ADCA0T1SEL00	ADCA0TTIN100	PIC	ADOPA1ADCATTIN00	ADCA0TTRG1
	ADCA0T1SEL01	ADCA0TTIN101	PIC	ADOPA1ADCATTIN01	
	ADCA0T1SEL02	ADCA0TTIN102	TAPA0	TAPATADOUT0	
	ADCA0T1SEL03	ADCA0TTIN103	TAPA1	TAPATADOUT0	
	ADCA0T1SEL04	ADCA0TTIN104	ENCA0	ENCATINT1	
	ADCA0T1SEL05	ADCA0TTIN105	ENCA1	ENCATINT1	
	ADCA0T1SEL06	ADCA0TTIN106	Pin	ADTRG00	
	ADCA0T1SEL07	ADCA0TTIN107	Pin	ADTRG10	
	ADCA0T1SEL08	ADCA0TTIN108	Pin	ADTRG11	
	ADCA0T1SEL09	ADCA0TTIN109	Not connected	–	
	ADCA0T1SEL10	ADCA0TTIN110	Not connected	–	
	ADCA0T1SEL11	ADCA0TTIN111	Not connected	–	
	ADCA0T1SEL12	ADCA0TTIN112	Not connected	–	
	ADCA0T1SEL13	ADCA0TTIN113	Not connected	–	
	ADCA0T1SEL14	ADCA0TTIN114	Not connected	–	
	ADCA0T1SEL15	ADCA0TTIN115	Not connected	–	

Table 25-2 What the Hardware Trigger Is Connected to (2/2)

ADCAn Channel Group	Start Trigger Bit (ADCAnTSELi Register)	Trigger Input Signal			ADCAn Trigger Signal
		Name	Connected to		
			Unit	Signal	
CG2	ADCA0T2SEL00	ADCA0TTIN200	PIC	ADOPA2ADCATTIN00	ADCA0TTRG2
	ADCA0T2SEL01	ADCA0TTIN201	PIC	ADOPA2ADCATTIN01	
	ADCA0T2SEL02	ADCA0TTIN202	TAPA0	TAPATADOUT1	
	ADCA0T2SEL03	ADCA0TTIN203	TAPA1	TAPATADOUT1	
	ADCA0T2SEL04	ADCA0TTIN204	ENCA0	ENCATINT1	
	ADCA0T2SEL05	ADCA0TTIN205	ENCA1	ENCATINT1	
	ADCA0T2SEL06	ADCA0TTIN206	Pin	ADTRG00	
	ADCA0T2SEL07	ADCA0TTIN207	Pin	ADTRG20	
	ADCA0T2SEL08	ADCA0TTIN208	Pin	ADTRG21	
	ADCA0T2SEL09	ADCA0TTIN209	Not connected	–	
	ADCA0T2SEL10	ADCA0TTIN210	Not connected	–	
	ADCA0T2SEL11	ADCA0TTIN211	Not connected	–	
	ADCA0T2SEL12	ADCA0TTIN212	Not connected	–	
	ADCA0T2SEL13	ADCA0TTIN213	Not connected	–	
	ADCA0T2SEL14	ADCA0TTIN214	Not connected	–	
	ADCA0T2SEL15	ADCA0TTIN215	Not connected	–	

25.1.1 Conditions for transitioning to the stand-by mode

This microcontroller does not support the stand-by mode.

25.1.2 Operation while in the power saving mode

- HALT mode

  A/D conversion operations continue.

25.2 Function Overview

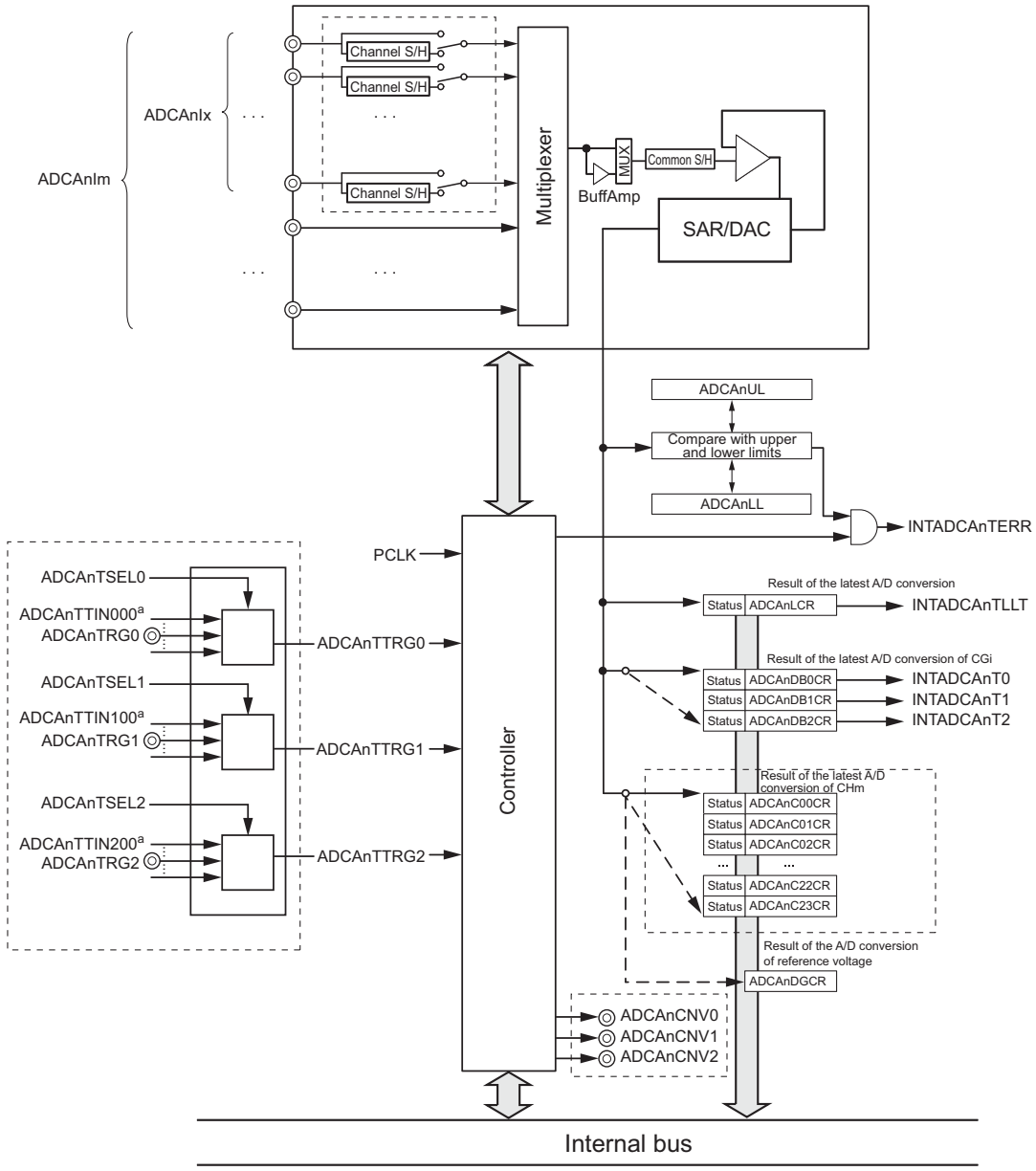
A/D converter A (ADCA) converts analog input signals to digital values.

Function overview

ADCAn has the following functions:

- Support for 10-bit and 12-bit resolution
- Successive approximation method
- Up to 24 analog input signals (depending on the product)
- Up to three differently-prioritized channel groups
- One-shot conversion mode, continuous conversion mode (only channel group 0)
- Auto-repeat function (one to four repeats for one shot mode)
- Software and hardware trigger modes
- The hardware trigger source can be selected from multiple input signals.
- The channel on which to generate the A/D conversion end interrupt when A/D conversion ends can be specified.
- Three conversion result check functions
- Discharge function to discharge the capacitor before a new sampling value is converted
- On/off switch function for the buffer amplifier
- Self-diagnosis functions

The following figure shows the main components of the ADCAn.



a) The connection destinations vary for each product. For details, see the section at the beginning of this chapter.

Figure 25-1 ADCAn Block Diagram

Note Portions enclosed in dashed lines are product dependent functions.

25.3 Description of Functions

A/D converter A (ADCAn) converts up to 24 analog input signals into digital values and supports 10-bit and 12-bit resolution (depending on the product).

Note The 10-bit/12-bit resolution setting is common to all the channels. The resolution cannot be changed on a channel basis.

Channels and channel groups

Each input channel can be assigned to one of three channel groups (CGs). The list of input channels assigned to a CG is called a scan list (including the diagnostic A/D conversion of CG0). The scan list can be easily set up using one register. Note that it is also possible to set up a new scan list during operation. All A/D conversion for a scan list is called scan list conversion.

ADCAn supports up to three differently-prioritized channel groups and two conversion modes:

- One-shot conversion mode: Scan list conversion is performed only once. In the one-shot conversion mode, scan list conversion can be repeated a specified number of times (one to four times).
- Continuous conversion mode: Scan list conversion is performed repeatedly.

A/D conversion

Conversion can be started by using the software or hardware as the start trigger.

A multiplexer selects the channel to be converted and the common sample & hold circuit holds the voltage input.

The successive approximation register (SAR) holds the D/A converter output voltage value to be compared to the analog input voltage value as a 10 or 12-bit digital value.

After each conversion the INTADCAnTLLT interrupt is generated.

A/D conversion result registers

When the A/D conversion is complete, the contents of the SAR register are stored in three registers, making it possible to read the latest A/D conversion result, the latest A/D conversion result of CGi, and the latest A/D conversion result of channel m.

Depending on the configuration, ADCAn generates a conversion end interrupt after the A/D conversion of certain channels or at the end of the A/D conversion of all channels of a channel group.

Conversion result check functions (product dependent)

For ADCAn, the following functions can be used to check the A/D conversion result:

- The conversion result overwrite check function
- The conversion result read flag function
- The conversion result limit comparison function

Discharge function (product dependent)

If required, the internal capacitor of the common sample & hold circuit can be discharged before every conversion.

On/off switch function for the buffer amplifier

To reduce the load of the external analog signal source, the signal can be connected to an internal buffer amplifier.

The buffer amplifier accelerates the charge rate of the internal sampling capacitor during the A/D sampling period.

Self-diagnosis functions

Four (product dependent) self-diagnosis functions are provided to check that ADCAn works correctly and to detect analog input pins that are disconnected.

- A/D conversion circuit diagnosis
- Channel multiplexer diagnosis
- Open pin diagnosis
- Channel sample & hold circuit diagnosis (product dependent)

Configurable stabilization time

By specifying any value for the stabilization counter, the best possible stabilization time can be secured after turning the power on.

25.3.1 Basic operation

This section describes the basic A/D conversion procedure. For details, see the following sections.

1. To optimize the start-up time after turning on the power or exiting the stand-by mode, you can increase or decrease the stabilization time by modifying the stabilization counter register ADCAnCNT.
2. Before enabling the A/D converter (ADCAnCTL0.ADCAnCE = 1 is required), specify the power on, resolution, ADCAn clock, trigger mode, conversion mode, interrupt generation, channel groups, and other settings for the following registers:
   - ADCAnCTL1 register
   - ADCAnCGi register
   - ADCAnIOCi register
   - ADCAnTSELi register (product dependent)
3. To check that the A/D conversion results are within a certain value range, enable the conversion result limit comparison function for the desired channels (ADCAnCTL2.ADCAnRCKm) and specify the lower and upper limits for the ADCAnLL and ADCAnUL registers.
4. To discharge the capacitor of the common sample & hold circuit before conversion, enable the discharge function by setting ADCAnCTL1.ADCAnDISC to 1.
5. Enable or disable the buffer amplifier by specifying a value for ADCAnCTL1.ADCAnBPC.
6. Enable the A/D converter by setting ADCAnCTL0.ADCAnCE to 1.

   The A/D converter is ready for A/D conversion after the stabilization time has elapsed after turning the power on or exiting the stand-by mode.
7. Depending on the specified trigger mode, A/D conversion is started by using one of the following channel group related start triggers:
   - A software trigger (when ADCAnTRGi.ADCAnSTTi is set to 1)
   - A hardware trigger (input signal ADCATTRGi)

If the A/D conversion of multiple CGs is triggered, the order of A/D conversion depends on the priority of the CGs.

8. The A/D conversion end interrupt INTADCAnTi is generated when the conversion of the channel specified by the ADCAnIOCi register ends.
9. Read the results from the A/D conversion result registers: ADCAnLCR, ADCAnDBiCR, and ADCAnCmCR.
10. Monitor the following registers:
    - ADCAnSTR1: Use this according to your purpose to check whether the A/D conversion results were overwritten before being read.
    - ADCAnSTR0: Use this to check whether the A/D conversion results are outside the specified value range (only if the conversion result limit comparison function is enabled).
11. Disable the A/D converter if you want to reconfigure it. To do so, clear ADCAnCTL0.ADCAnCE.

Note The self-diagnosis functions are described in *25.3.13 “Self-diagnosis functions (product dependent)” on page 1713* .

25.3.2 Clock usage

The ADCAn clock ADCAnTCLK is generated from PCLK. The division ratio is specified for ADCAnCTL1.ADCAnFR[3:0].

25.3.3 Channels and channel groups

Each input channel is organized into a channel group (CG). The scan list for each CG can be created by specifying register settings, and this list can be set up again even during operation. The conversion settings of a CG are applied to all channels in the group.

ADCAn supports up to three channel groups: CGi (i = 0 to 2). The channels of CGi are specified for the ADCAnCGi register.

Note ADCAn processes A/D conversion requests for CGs only. By assigning only one input channel to a CG, it is possible to perform conversion for a single channel.

(1) Order of A/D conversion

If a CG start trigger occurs, conversion is performed for the channels specified in the scan list in ascending order (from CH00 to CH23 (product dependent)).

If A/D conversion requests for multiple CGs are pending, the CGs are converted in the following hierarchical order: CG2 (highest priority) > CG1 > CG0 (lowest priority)

The current A/D conversion is interrupted when a start trigger for a higher priority CG or an ADCHALT mode trigger is set. Depending on the ADCAnCTL1.ADCAnTRMi setting there are two options:

- The A/D conversion of CGs is interrupted immediately (when ADCAnCTL1.ADCAnTRMi is cleared).

  The A/D conversion of the interrupted channel is resumed after all pending A/D conversions of higher priority CGs have finished.

- The A/D conversion of the current channel is completed before higher priority CGs are converted (when ADCAnCTL1.ADCAnTRMi is set).

  The A/D conversion resumes starting with the next channel after all pending A/D conversions of higher priority CGs have finished.

ADCAnSTR2.ADCAnST[2:0] indicates the conversion status.

Example The figures below show an example of A/D conversion interruption. In this example, CH3, CH9, and CH20 are assigned to CG0, and CH5 and CH9 are assigned to CG2.

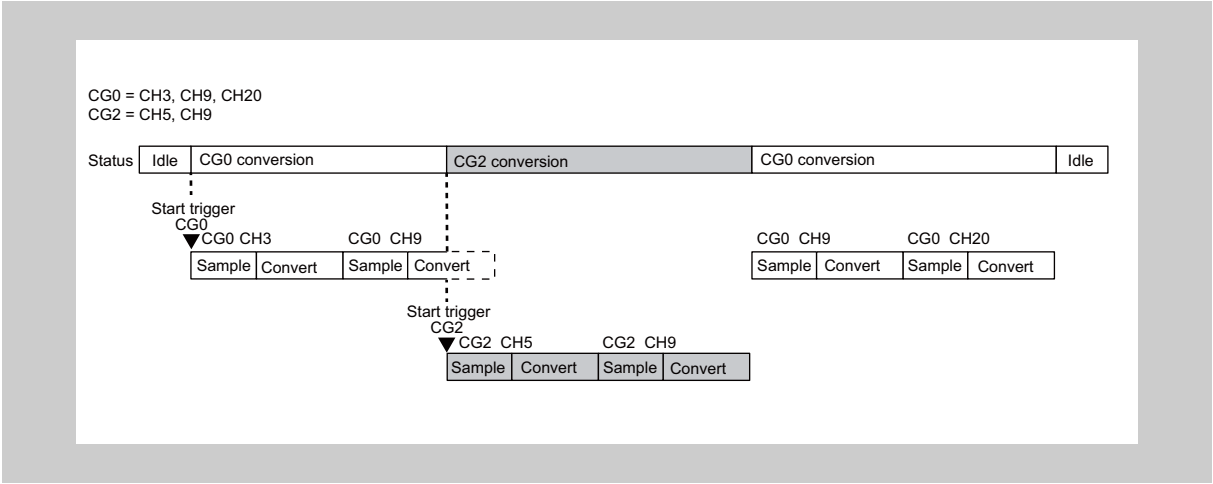


Figure 25-2 When the A/D Conversion of CG0 Is Interrupted Immediately (When ADCAnCTL1.ADCAnTRM0 Is Cleared to 0)

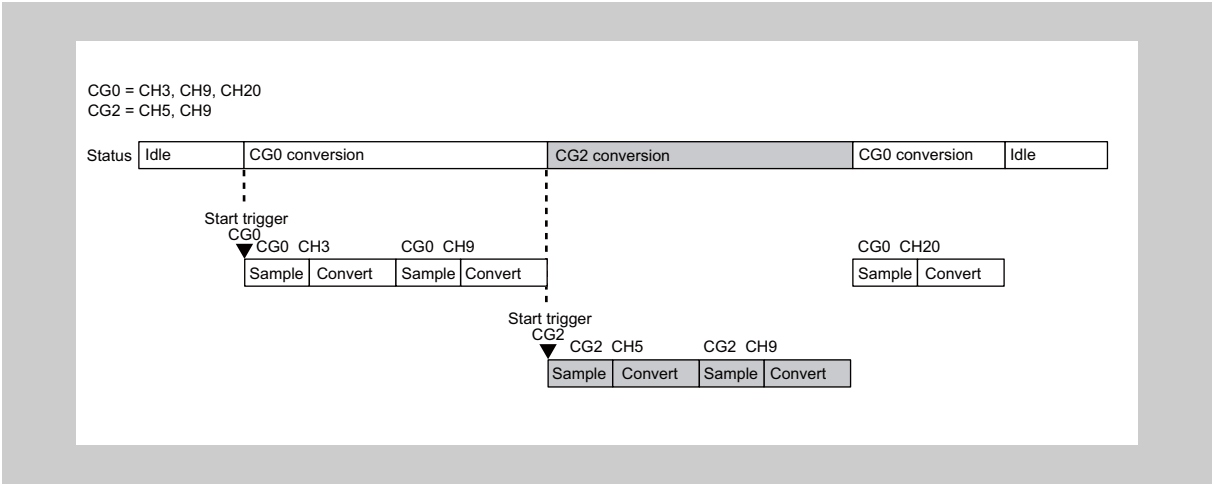


Figure 25-3 When the System Goes on Stand-By Until A/D Conversion of the Current Channel Ends (When ADCAnCTL1.ADCAnTRM0 Is Set to 1)

25.3.4 A/D conversion modes

The A/D converter has two A/D conversion modes:

Mode	Operation	Channel Group
One-shot conversion mode	Scan list conversion is performed only once. In the one-shot conversion mode, scan list conversion can be repeated a specified number of times (one to four times).	CG0, CG1, CG2
Continuous conversion mode	Scan list conversion is performed repeatedly.	CG0

Notes

1. If a running A/D conversion is interrupted by an A/D conversion request for a higher priority CG, it is automatically resumed after all requests of higher priority CGs have been finished. (For details, see *(1) "Order of A/D conversion" on page 1695*.)
2. CG1 and CG2 are operated in the one-shot conversion mode regardless of the conversion mode setting. For CG0, the A/D conversion mode can be specified for ADCAnCTL1.ADCAnMD0.

(1) One-shot conversion mode

In the one-shot conversion mode, a start trigger causes CGi scan list conversion to be performed. The number of times to repeat scan list conversion can be specified for ADCAnCTL0.ADCAnSCTi[1:0] on a CG basis as a value from one to four.

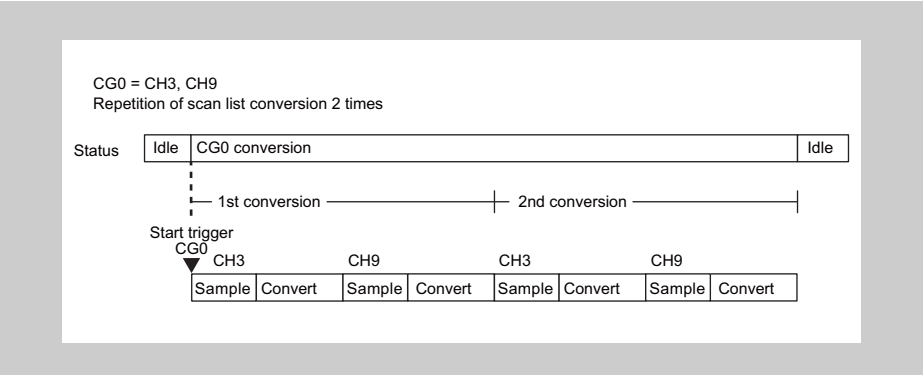


Figure 25-4 One-Shot Conversion Mode With Two Scan List Conversions

Operation when a start trigger is input before conversion ends

The A/D converter can hold up to one additional start trigger issued before conversion of a given CG ends. Therefore, if more than one additional start trigger is input before the A/D conversion of CGi caused by the first start trigger ends, A/D conversion is performed consecutively (but the second and subsequent additional start triggers are ignored).

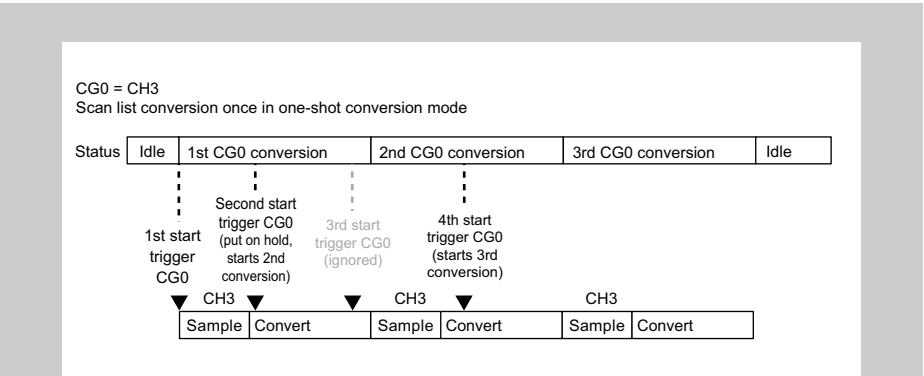


Figure 25-5 Operation When a Start Trigger Is Input Before Conversion Ends

Caution While a high priority CG is converted, any start triggers for lower priority CGs that occur before conversion completion are ignored. In contrast, if higher-priority CG conversion starts during the conversion for a lower-priority CG, the start trigger for the lower-priority CG conversion is held. Note that, if no start trigger has occurred, triggers are acknowledged even during the high-priority CG conversion period.

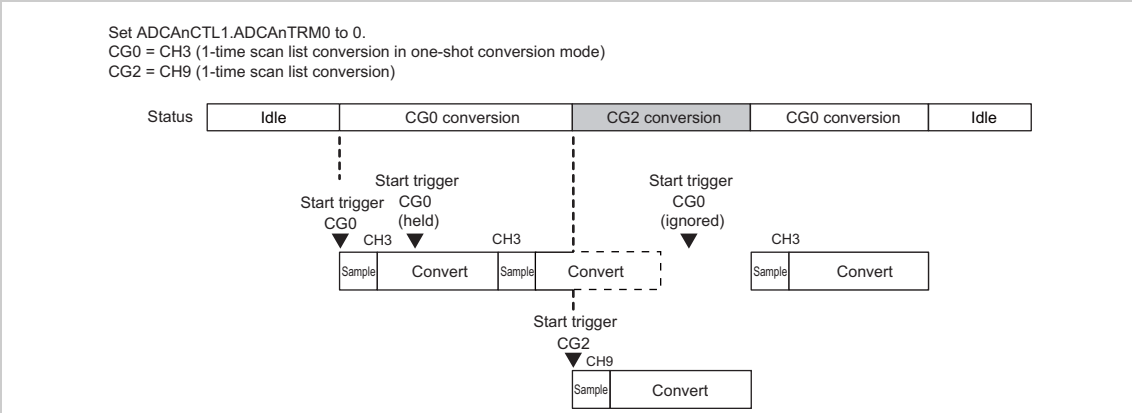


Figure 25-6 Operation When a High-Priority CG Start Trigger is Input Before Conversion of Low-Priority CG Ends

(2) Continuous conversion mode

The continuous conversion mode is available for CG0 only (and is enabled by setting ADCAnCTL1.ADCAn_MD0).

In the continuous conversion mode, a start trigger causes the channels of CG0 to be sampled and converted repeatedly until a stop trigger is generated or another stop condition occurs. (For details, see *25.3.6 “Stopping A/D conversion (stop triggers)” on page 1701* .)

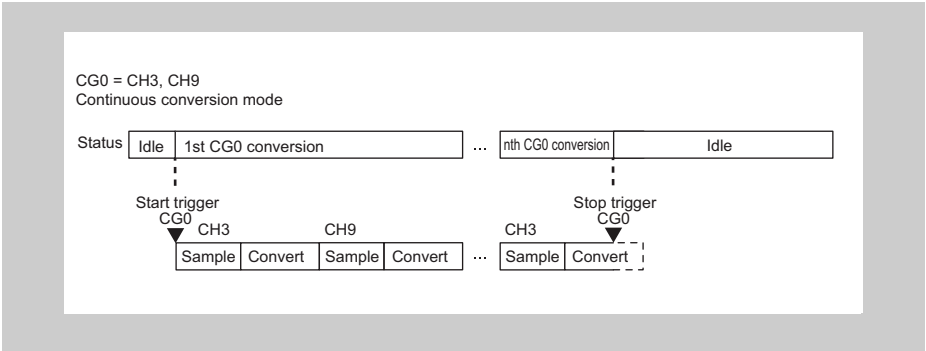


Figure 25-7 Continuous Conversion Mode

Caution After the stop trigger is generated, the system becomes idle, and sampling and conversion are not performed.

Note Start triggers for CG0 are ignored in the continuous conversion mode.

25.3.5 Starting A/D conversion (start triggers)

A/D conversion can be started by a software or hardware trigger specified for ADCAnCTL1.ADCAnMD1.

When A/D conversion is triggered for multiple CGs, the conversion order depends on the priority of the CGs. (For details, see *(1) "Order of A/D conversion" on page 1695*.)

Notes

1. ADCAn ignores start triggers for a CGi if there is no channel assigned to the scan list for the CGi. (ADCAnCGi register = 0000 0000_H).
2. In the one-shot conversion mode, the A/D converter holds up to one start trigger.

   Therefore, if more than one additional start trigger is input before the triggered A/D conversion of CGi ends, A/D conversion is performed consecutively (but the second and subsequent additional start triggers are ignored). (For details, see *Figure 25-5 "Operation When a Start Trigger Is Input Before Conversion Ends" on page 1697* .)
3. In the continuous conversion mode, additional start triggers that are generated before a stop trigger has been generated are ignored.

(1) Software start trigger

The A/D conversion of CGi is triggered by setting ADCAnTRGi.ADCAnSTTi if the A/D converter is enabled (by setting ADCAnCTL0.ADCAnCE).

Software start trigger timing example

The following figure shows the timing of a software start trigger with the following conditions:

- ADCAnTCLK clock = PCLK / 2 (and ADCAnCTL1.ADCAnFR[3:0]=0000_B)

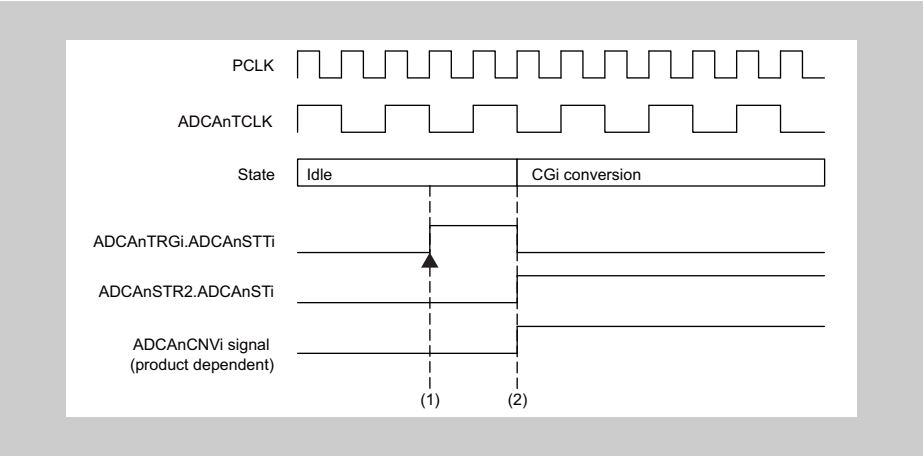


Figure 25-8 Software Start Trigger Timing Example

1. A software start trigger for CGi is written.
2. A/D conversion starts at the next falling edge of the ADCAnTCLK clock.

   The status signal ADCAnTCNVi becomes active (product dependent) and ADCAnSTR2.ADCAnSTi is set, indicating that A/D conversion of CGi is running.

(2) Hardware start trigger

The A/D conversion of CGi is triggered by valid edge detection of the ADCAnTTRGi signal if the A/D converter is enabled (by setting ADCAnCTL0.ADCAnCE) and the hardware trigger mode is specified (by setting ADCAnCTL1.ADCAnMD1).

The valid edge is specified for ADCAnCTL1.ADCAnTiETS[1:0] on a CG basis.

Hardware trigger expansion (product dependent)

If hardware trigger expansion is supported, up to 16 hardware trigger sources can be specified for each ADCAnTTRGi signal input. The ADCAnTSELi register specifies the input signals to be used as the ADCAnTTRGi signal.

Note For details about the connection destinations of the hardware start trigger function, see *Table 25-2 "What the Hardware Trigger Is Connected to"*.

Hardware start trigger timing

The A/D converter starts A/D conversion when a valid edge of the ADCAnTTRGi signal is detected.

The figure below shows the timing of a hardware start trigger with the following conditions:

- ADCAnTCLK clock = PCLK / 2 (and ADCAnCTL1.ADCAnFR[3:0] = 0000_B)
- The valid edge of the ADCAnTTRGi signal is the rising edge (ADCAnCTL1.ADCAnTiETS[1:0] = 01_B)

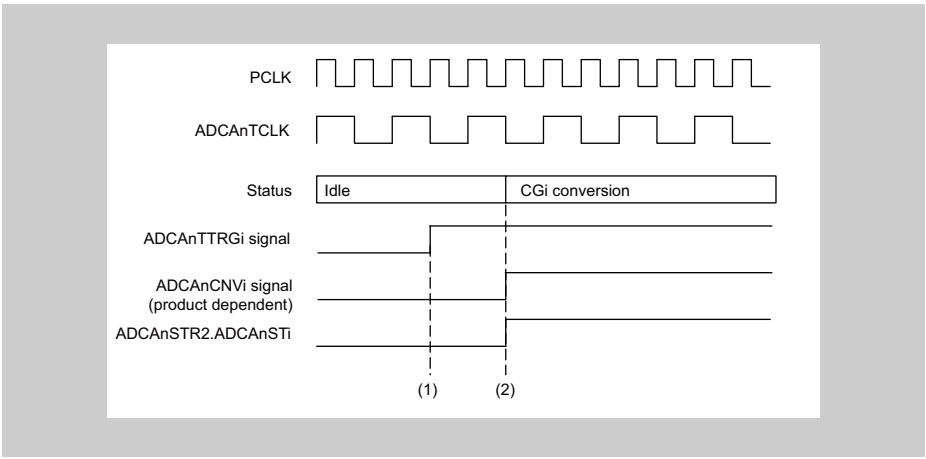


Figure 25-9 Hardware Start Trigger Timing

1. The input signal ADCAnTTRGi rises.
2. A/D conversion starts at the next falling edge of the ADCAnTCLK clock.

   The status signal ADCAnTCNVi becomes active (product dependent) and ADCAnSTR2.ADCAnSTi is set, indicating that A/D conversion of CGi is running.

25.3.6 Stopping A/D conversion (stop triggers)

(1) Stop trigger

Setting the stop trigger for CGi stops the A/D conversion of CGi (when ADCAnTRG4 + i.ADCAnSPi is set). If a stop trigger occurs before the end of A/D conversion, the A/D conversion end interrupt INTADCAnTi is not generated, and the A/D conversion result registers are not updated. If another start trigger is input after A/D conversion is stopped by a stop trigger, scan list conversion is performed from the beginning.

If using hardware start triggers, perform the following procedure:

1. Stop the generation of hardware start triggers.
2. Set the stop trigger bit (ADCAnTRG4 + i.ADCAnSPi).
3. Check the status of ADCAnSTR2.ADCAnSTi.

If the above procedure is not performed, there is a conflict between the hardware start trigger and stop trigger, and A/D conversion might not stop.

Stop trigger timing

1. A stop trigger for CGi is written.
2. A/D conversion of CGi stops at the next falling edge of the ADCAnTCLK clock.

   ADCAnSTR2.ADCAnSTi is cleared, indicating that A/D conversion of CGi has stopped.
   If the digital value of ADCAnIm is already available:
   – All A/D conversion result registers are updated.
   – The conversion end interrupt INTADCAnTi is generated as specified by the ADCAnIOCi register. (For details, see *25.3.10 “Interrupt generation” on page 1707* .)
   – If specified by ADCAnCTL2, the A/D conversion result is checked to see whether it is in the specified value range. (For details, see *25.3.12 “Result check functions” on page 1711* .)

   The A/D converter proceeds with any pending A/D conversion requests of other CGs.

The figures below show the timing of a stop trigger with the following conditions:

- ADCAnTCLK clock = PCLK / 2 (and ADCAnCTL1.ADCAnFR[3:0] = 0000_B)
- The A/D conversion end interrupt INTADCAnTi is generated at the end of the A/D conversion of CGi (ADCAnIOCi register = 0000 0000_H).

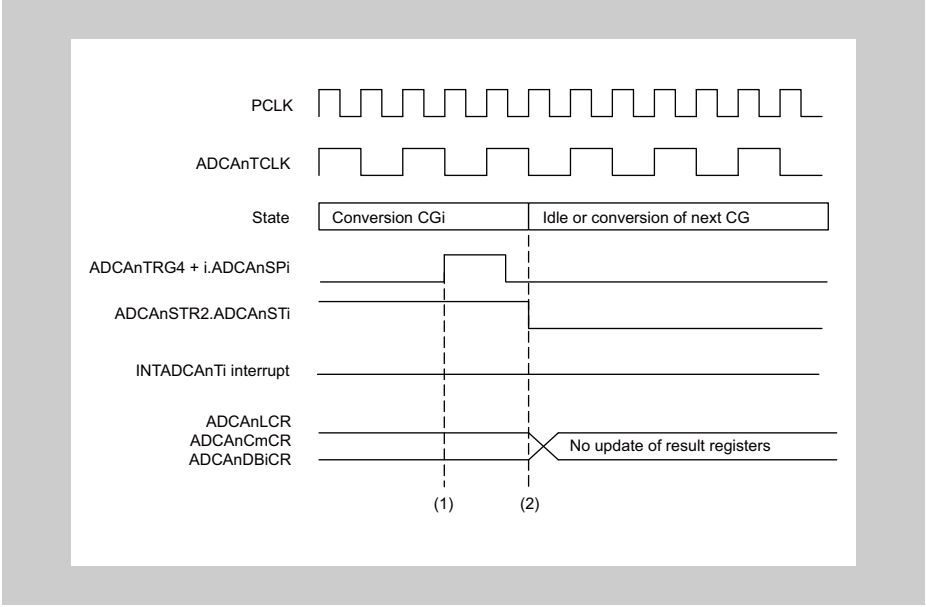


Figure 25-10 Stop Trigger Timing (When Generated Before A/D Conversion Completion)

1. The stop trigger bit (ADCAnTRG4 + i.ADCAnSPi) is set.
2. The status bit (ADCAnSTR2.ADCAnSTi) is cleared.

   The A/D conversion end interrupt INTADCAnTi is not generated, and the A/D conversion result registers are not updated.

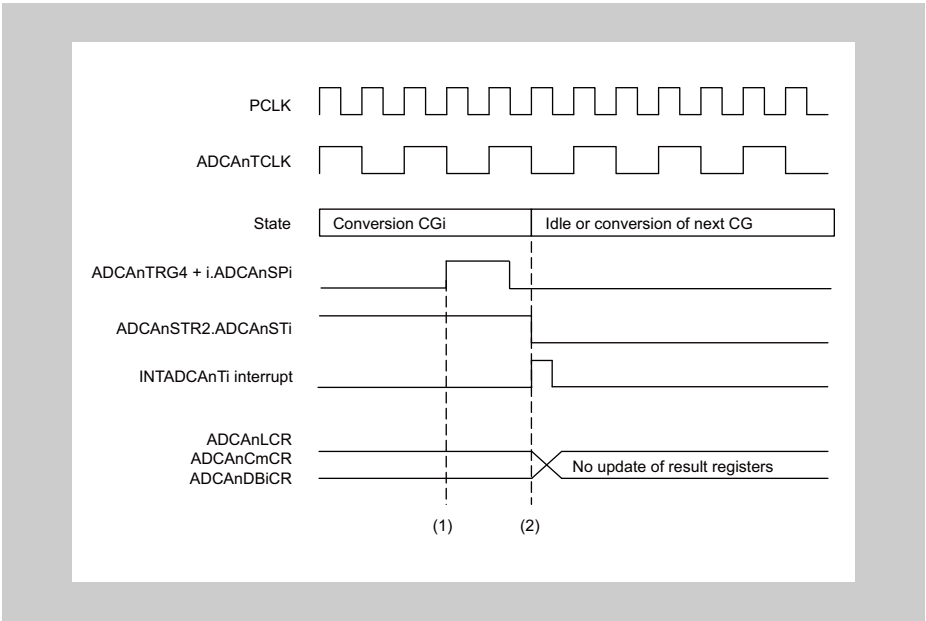


Figure 25-11 Stop Trigger Timing (When Generated After A/D Conversion Completion)

1. The stop trigger bit (ADCAnTRG4 + i.ADCAnSPi) is set.
2. The status bit (ADCAnSTR2.ADCAnSTi) is cleared.

   The A/D conversion end interrupt INTADCAnTi is generated, and the A/D conversion result registers are updated.

(2) Other stop conditions

In addition to a software stop trigger, A/D conversion stops in the following condition:

- When the A/D converter is disabled (by clearing ADCAnCTL0.ADCAnCE)

25.3.7 Stand-by mode (product dependent)

The stand-by mode is entered as described in *25.1.1 "Conditions for transitioning to the stand-by mode"*, which is at the beginning of this chapter.

The A/D converter is automatically disabled (by clearing ADCAnCTL0.ADCAnCE) while the system is in the stand-by mode.

To exit the stand-by mode:

1. Exit the related system stand-by mode.
2. Enable the A/D converter by setting ADCAnCTL1.ADCAnCE.

Note After exiting the stand-by mode, start triggers are acknowledged, but conversion will not start before the stabilization time has elapsed (stabilization counter ADCAnCNT = 00_H). For details, see *25.3.17 “Stabilization control” on page 1729* .

25.3.8 Pausing and resuming A/D conversion (ADCHALT mode) (product dependent)

The A/D converter allows the A/D conversion (of all CGs) to be paused or halted (the ADCHALT mode).

Procedure:

1. Change the system to the ADCHALT mode by setting ADCAnTRG3.ADCAnSTT3. (For details about the interrupt behavior, see *(1) "Order of A/D conversion" on page 1695*.)
   – Start triggers are ignored while in the ADCHALT mode.
   – The power consumption can be reduced by setting the sampling clock ADCAnTCLK to the low level and stopping the internal circuit.
   – While in the ADCHALT mode, the alternate functions of the analog input pin ADCAnIm can be used.
2. Exit the ADCHALT mode and resume the A/D conversion by setting ADCAnTRG7.ADCAnSP3 .

Note The ADCHALT mode has the highest priority and overrules all CGi conversions.

25.3.9 Resolution, sampling, and conversion times

The total conversion time is the sum of the sampling time and A/D conversion time.

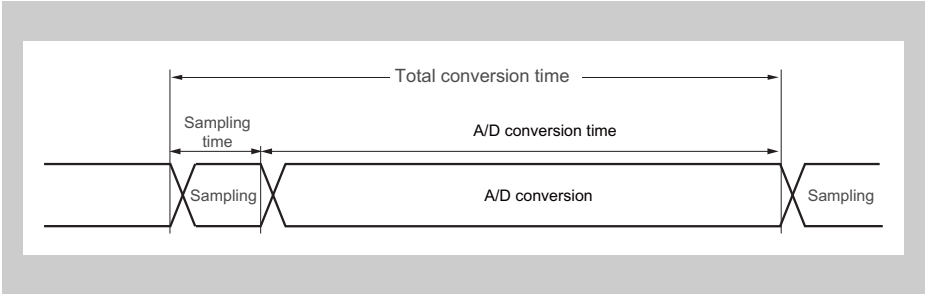


Figure 25-12 Total Conversion Time

- The sampling time is the time of connecting an analog input voltage to the common sample & hold circuit.The A/D conversion time is the time necessary to obtain one digital value from the analog input voltage.The A/D conversion time, and thus the total conversion time, depends on the conversion resolution as follows:

Table 25-3 Sampling and Conversion Times

Sampling Mode (Product Dependent)	Conversion Resolution	Sampling Time	Total Conversion Time
A	10 bit (when ADCAnCTL1.ADCAnCTYP is set)	6.5 clock cycles (ADCAnTCLK)	18 clock cycles (ADCAnTCLK)
	12 bit (when ADCAnCTL1.ADCAnCTYP is cleared)	6.5 clock cycles (ADCAnTCLK)	20 clock cycles (ADCAnTCLK)

Notes

1. Enabling the discharge function (by setting ADCAnCTL1.ADCAnDISC) increases the total conversion time by one clock cycle (ADCAnTCLK).

   For details, see *25.3.15 “Discharge function (product dependent)” on page 1728* .

2. Enabling the buffer amplifier (by setting ADCAnCTL1.ADCAnBPC) increases the total conversion time by four clock cycles (ADCAnTCLK).

   For details, see *25.3.16 “Buffer amplifier function” on page 1728* .

Table 25-4 PCLK Settings When the Resolution is 12 Bits and the Specifiable Division Ratios and Conversion Times (AV_{DD} = 4.5 V)

When 12-Bit Is Specified		PCLK <MHz> (Product Dependent)									
ADCAnFR[3:0] Bits	Division Ratio	48.000 MHz					66.667 MHz				
		ADCLK [MHz]	Amplifier Power Off		Amplifier Power On		ADCLK [MHz]	Amplifier Power Off		Amplifier Power On	
			Discharge Off	Discharge On	Discharge Off	Discharge On		Discharge Off	Discharge On	Discharge Off	Discharge On
0000B	1/2	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0001B	1/3	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0010B	1/4	12.000	1.667	Setting prohibited	2.000	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0011B	1/5	9.600	2.083	Setting prohibited	2.500	Setting prohibited	13.33	1.500	Setting prohibited	1.800	Setting prohibited
0100B	1/6	8.000	2.500	Setting prohibited	3.000	Setting prohibited	11.111	1.800	Setting prohibited	2.160	Setting prohibited
0110B	1/8	6.000	3.333	Setting prohibited	4.000	Setting prohibited	8.333	2.400	Setting prohibited	2.880	Setting prohibited
1000B	1/10	4.800	4.167	Setting prohibited	5.000	Setting prohibited	6.667	3.000	Setting prohibited	3.600	Setting prohibited
1010B	1/12	4.000	5.000	Setting prohibited	6.000	Setting prohibited	5.556	3.600	Setting prohibited	4.320	Setting prohibited
1100B	1/14	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	4.762	4.200	Setting prohibited	5.040	Setting prohibited
1110B	1/16	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	4.167	4.800	Setting prohibited	5.760	Setting prohibited

Table 25-5 PCLK Settings When the Resolution is 10 Bits and the Specifiable Division Ratios and Conversion Times (AV_{DD} = 3.3 V ±0.3 V or 5.0 V ±0.5 V)

When 10-Bit Is Specified		PCLK <MHz> (Product Dependent)									
ADCAnFR[3:0] Bits	Division Ratio	48.000 MHz					66.667 MHz				
		ADCLK [MHz]	Amplifier Power Off		Amplifier Power On		ADCLK [MHz]	Amplifier Power Off		Amplifier Power On	
			Discharge Off	Discharge On	Discharge Off	Discharge On		Discharge Off	Discharge On	Discharge Off	Discharge On
0000B	1/2	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0001B	1/3	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0010B	1/4	12.000	1.500	Setting prohibited	1.833	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0011B	1/5	9.600	1.875	Setting prohibited	2.292	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited
0100B	1/6	8.000	2.250	Setting prohibited	2.750	Setting prohibited	11.111	1.620	Setting prohibited	1.980	Setting prohibited

When 10-Bit Is Specified		PCLK <MHz> (Product Dependent)								
		48.000 MHz					66.667 MHz			
ADCAnFR[3:0] Bits	Division Ratio	ADCLK [MHz]	Amplifier Power Off		Amplifier Power On		ADCLK [MHz]	Amplifier Power Off		Amplifier Power On	
			Discharge Off	Discharge On	Discharge Off	Discharge On		Discharge Off	Discharge On	Discharge Off	Discharge On
0110B	1/8	6.000	3.000	Setting prohibited	3.667	Setting prohibited	8.333	2.160	Setting prohibited	2.640	Setting prohibited
1000B	1/10	4.800	3.750	Setting prohibited	4.583	Setting prohibited	6.667	2.700	Setting prohibited	3.300	Setting prohibited
1010B	1/12	4.000	4.500	Setting prohibited	5.550	Setting prohibited	5.556	3.240	Setting prohibited	3.960	Setting prohibited
1100B	1/14	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	4.762	3.780	Setting prohibited	4.620	Setting prohibited
1110B	1/16	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	Setting prohibited	4.167	4.320	Setting prohibited	5.280	Setting prohibited

Note The above conversion times do not include the overhead time due to A/D controller processing. The necessary overhead time is as follows:

- Conversion start request acknowledgment: One clock cycle (ADCAnTCLK)
- Initialization before starting conversion: One clock cycle (ADCAnTCLK)
- Conversion result storage processing: One clock cycle (ADCAnTCLK)

When performing consecutive conversions, no conversion start request acknowledgment processing is necessary for the second or subsequent conversions.

25.3.10 Interrupt generation

(1) A/D conversion end interrupt INTADCAnTi

The interrupt INTADCAnTi reports that new A/D conversion results have been stored in the conversion result registers.

When the A/D conversion of any CGi channel specified by the ADCAnIOCi register ends, the A/D conversion end interrupt is generated.

If no specific channel or other relevant setting is specified (ADCAnIOCi = 0000 0000_H), the interrupt INTADCAnTi is generated at the end of the A/D conversion of CGi.

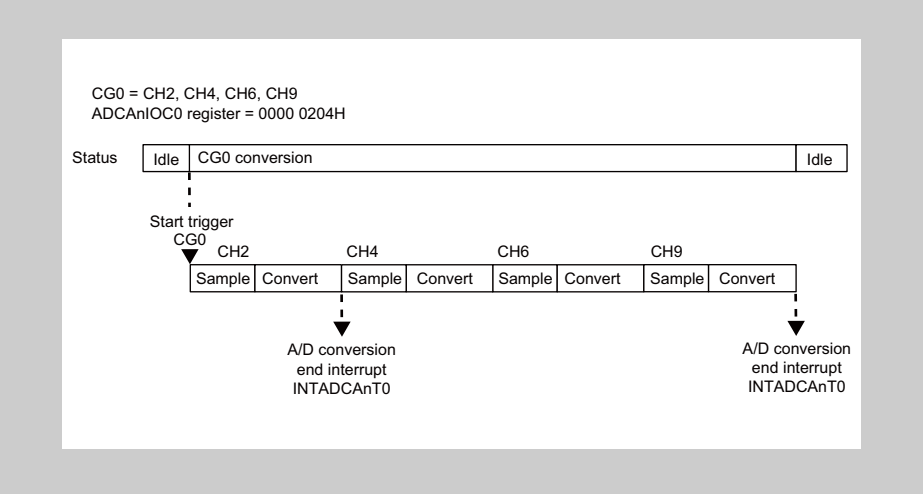


Figure 25-13 Generation of the A/D Conversion End Interrupt INTADCAnTi

Notes

1. The ADCAnIOCi register can be written at any time even when the A/D converter is enabled (by setting ADCAnCTL0.ADCAnCE). The newly specified value takes effect after the current A/D conversion of CGi has been completed.
2. Because the ADCAnIOCi register is associated with the ADCAnCGi register, their buffer registers must be updated simultaneously. Because the update is performed when the ADCAnCGi register is written to, always write to the ADCAnIOCi register before the ADCAnCGi register when changing the interrupt generation for a CG.

(2) Error interrupt INTADCAnTERR

The interrupt INTADCAnTERR is generated in the following cases:

- When the conversion result limit comparison function is enabled and an A/D conversion result of a specified channel is out of the specified range. For details, see *(3) "Conversion result limit comparison function" on page 1712*.
- The generation of the error interrupt INTADCAnTERR when the A/D conversion results in the ADCAnLCR register, ADCAnDBiCR register, or ADCAnCmCR register are overwritten before they are read can be separately controlled for each register by setting ADCAnCTL0.ADCAnOEM[4:0]. For details, see *(1) "Conversion result overwrite check function" on page 1711*.

25.3.11 Storage of A/D conversion results

(1) A/D conversion result registers

The A/D conversion results are stored in the following registers:

- ADCAnLCR register

  This register stores the latest A/D conversion results.

- ADCAnDBiCR register

  This register stores the latest CGi A/D conversion results.

- ADCAnCmCR register

  This register stores the latest A/D conversion results for channel m.

These registers store the digital value of the sampled analog input voltage in bits 15 to 00. Each register also stores A/D conversion result status flags. (For details, see *25.3.12 “Result check functions” on page 1711* .)

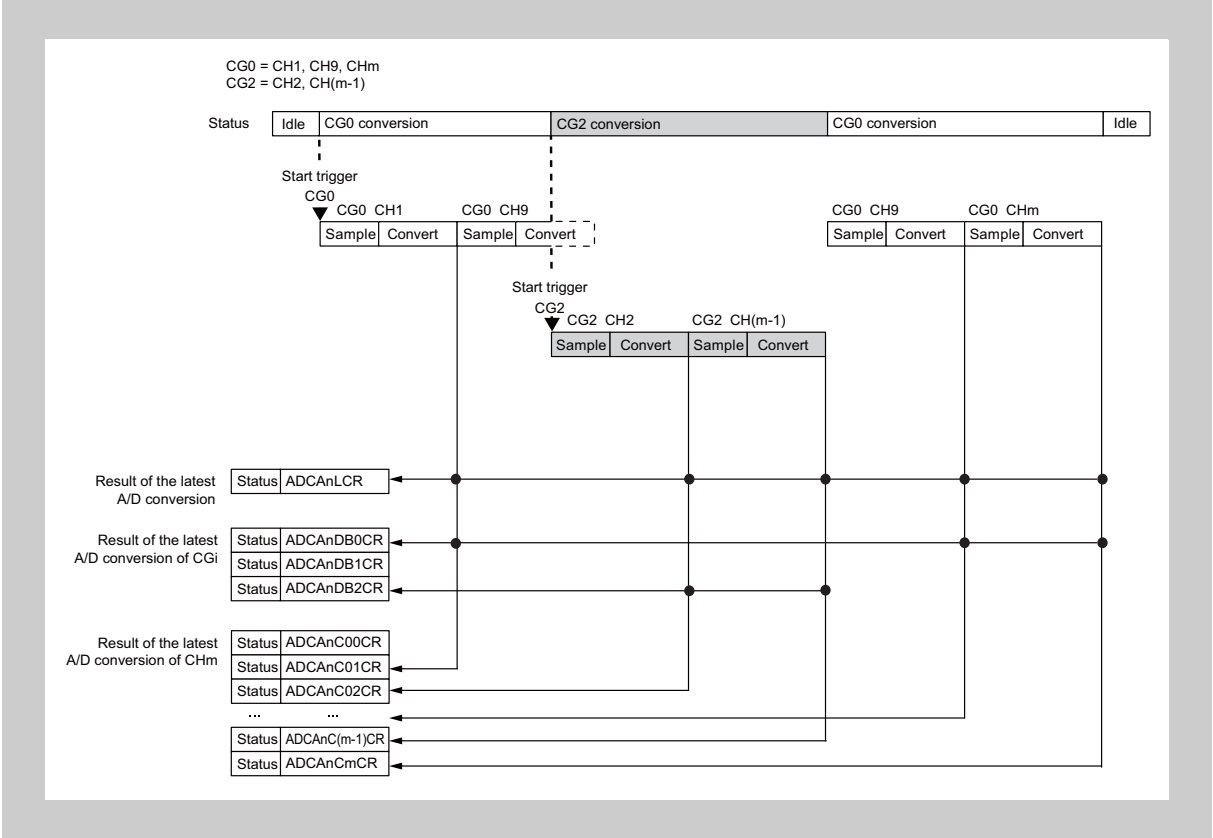


Figure 25-14 Storage of A/D Conversion Results

(2) Settings related to storing conversion results

(a) Conversion result bit position specification function

ADCAnCTL1.ADCAnCRAC specifies whether the 12-bit or 10-bit A/D conversion result is right aligned (ADCAnCRAC bit = 0) or left aligned (ADCAnCRAC bit = 1).

(b) Conversion result read and clear function

ADCAnCTL1.ADCAnRCL specifies whether the value of the A/D conversion result register ADCAnCmCR is retained after reading it or cleared by reading it.

(3) Relationship between the analog input voltage and A/D conversion results

The following equation shows the relationship between the analog input voltage input to the analog input pin (ADCAnIm) and the A/D conversion result value (the value of the ADCAnLCR[15:00] bits, ADCAnCmCR[15:00] bits, and ADCAnDBiCR[15:00] bits):

$$\text{A/D conversion result value} = \text{INT}\left(\frac{V_{IAN} - AV_{REFMn}}{AV_{REFPn} - AV_{REFMn}} \times 2^k + 0.5\right)$$

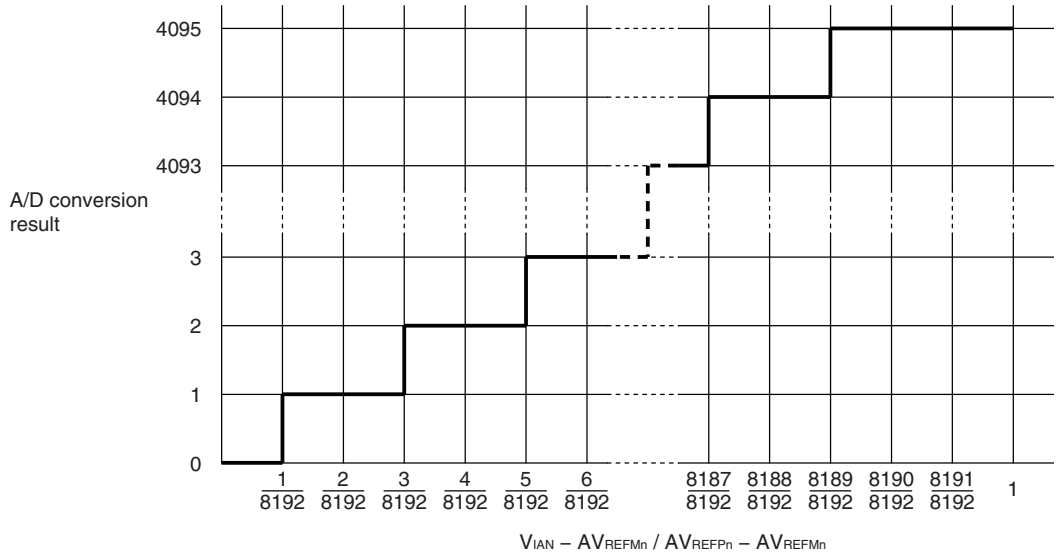
Or,

$$(\text{A/D conversion result value} - 0.5) \times \frac{AV_{REFPn} - AV_{REFMn}}{2^k} \leq V_{IAN} - AV_{REFMn} < (\text{A/D conversion result value} + 0.5) \times \frac{AV_{REFPn} - AV_{REFMn}}{2^k}$$

INT( ):	Function that returns the integer portion of the value in ( )
V_{IAN}:	Analog input voltage
AV_{REFPn}:	AV_{REFPn} pin voltage
AV_{REFMn}:	AV_{REFMn} pin voltage
A/D conversion result value:	Value of the ADCAnLCR[15:00] bits, ADCAnCmCR[15:00] bits, and ADCAnDBiCR[15:00] bits
k:	Resolution

Figure 25-15 shows the relationship between the analog input voltage and the A/D conversion results.

(i) 12-bit A/D converter conversion characteristics (when ADCAnCTL1.ADCAnCTYP is cleared)



(ii) 10-bit A/D converter conversion characteristics (when ADCAnCTL1.ADCAnCTYP is set)

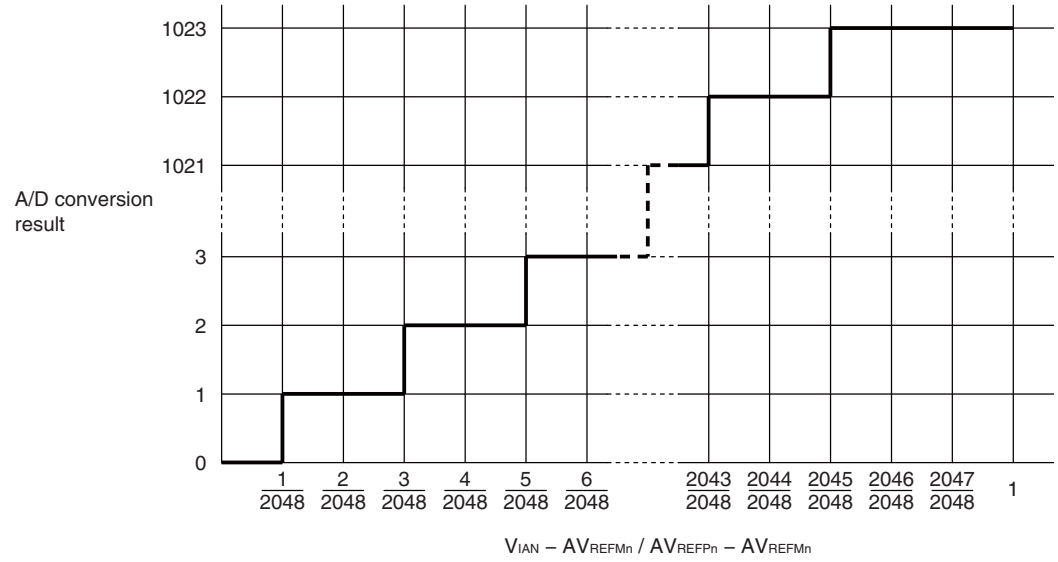


Figure 25-15 Relationship Between the Analog Input Voltage and A/D Conversion Results

25.3.12 Result check functions

For ADCAn, the following functions can be used to check the A/D conversion result:

- The conversion result overwrite check function
- The conversion result read flag function
- The conversion result limit comparison function

(1) Conversion result overwrite check function

This function makes it possible to check whether an A/D conversion result was overwritten before it was read.

Error flags Each A/D conversion result register has the following overwrite error flags:

ADCAnLCR.ADCAnLER1

ADCAnDBiCR.ADCAnDBiER1

ADCAnCmCR.ADCAnCmER1

As an example, if the A/D conversion result stored in the ADCAnCmCR register is overwritten before it is read, ADCAnCmCR.ADCAnCmER1 is set.

The ADCAnLCR and ADCAnDBiCR registers function similarly.

Note that the value of ADCAnCmCR.ADCAnCmER1 is also applied to ADCAnSTR1.ADCAnOWEm.

Error interrupt The error interrupt INTADCAnTERR is generated if an A/D conversion result in the ADCAnLCR, ADCAnDBiCR, or ADCAnCmCR register is overwritten before it is read.

A conversion result register from which the conversion result is not read can be masked by specifying ADCAnCTL0.ADCAnOEM[4:0].

Example:

- CH7 is assigned to CG1.
- ADCAnCTL0.ADCAnOEM[4:0] is cleared to 00000B so that the error interrupt INTADCAnTERR is generated whenever an A/D conversion result is overwritten in the ADCAnLCR, ADCAnDBiCR, or ADCAnCmCR register before the result is read.

(2) Conversion result read flag function

This function makes it possible to check whether the A/D conversion result has been read before or is new and has not been read yet.

Status flags The A/D conversion result registers have the following update status flags:

ADCAnLCR.ADCAnLUR

ADCAnDBiCR.ADCAnDBiUR

ADCAnCmCR.ADCAnCmUR

If these flags are set to 1, the A/D conversion result is new.
The update status flags are cleared after being read.

(3) Conversion result limit comparison function

This function makes it possible to check whether the A/D conversion results are in the specifiable range.

This function can be enabled or disabled for individual channels by using the ADCAnCTL2 register.

The A/D conversion result of a channel for which this function is enabled is compared with the previously specified lower limit (the ADCAnLL register) and upper limit (the ADCAnUL register).

Error flags

If the A/D conversion result of a specified channel is below the lower limit or above the upper limit, the error flag for that channel (ADCAnSTR0.ADCAnRCE) is set.

The ADCAnSTR0 register indicates the error status of the latest A/D conversion result limit comparison for every channel. This register makes it possible to check which A/D conversion results are outside the specified range.

Note that the value of the result check error flag ADCAnSTR0.ADCAnRCE is also applied to ADCAnCmCR.ADCAnCmER0.

Error interrupt

The error interrupt INTADCAnTERR is generated if an A/D conversion result of a specified channel is outside the specified range.

25.3.13 Self-diagnosis functions (product dependent)

The following self-diagnosis functions can be used to check whether ADCAn is working:

(1) A/D conversion circuit diagnosis

(2) Channel multiplexer diagnosis

(3) Open pin diagnosis

(4) Channel sample & hold circuit diagnosis (product dependent)

The following figure provides an overview of the self-diagnosis functions, which are described in detail in the following chapters.

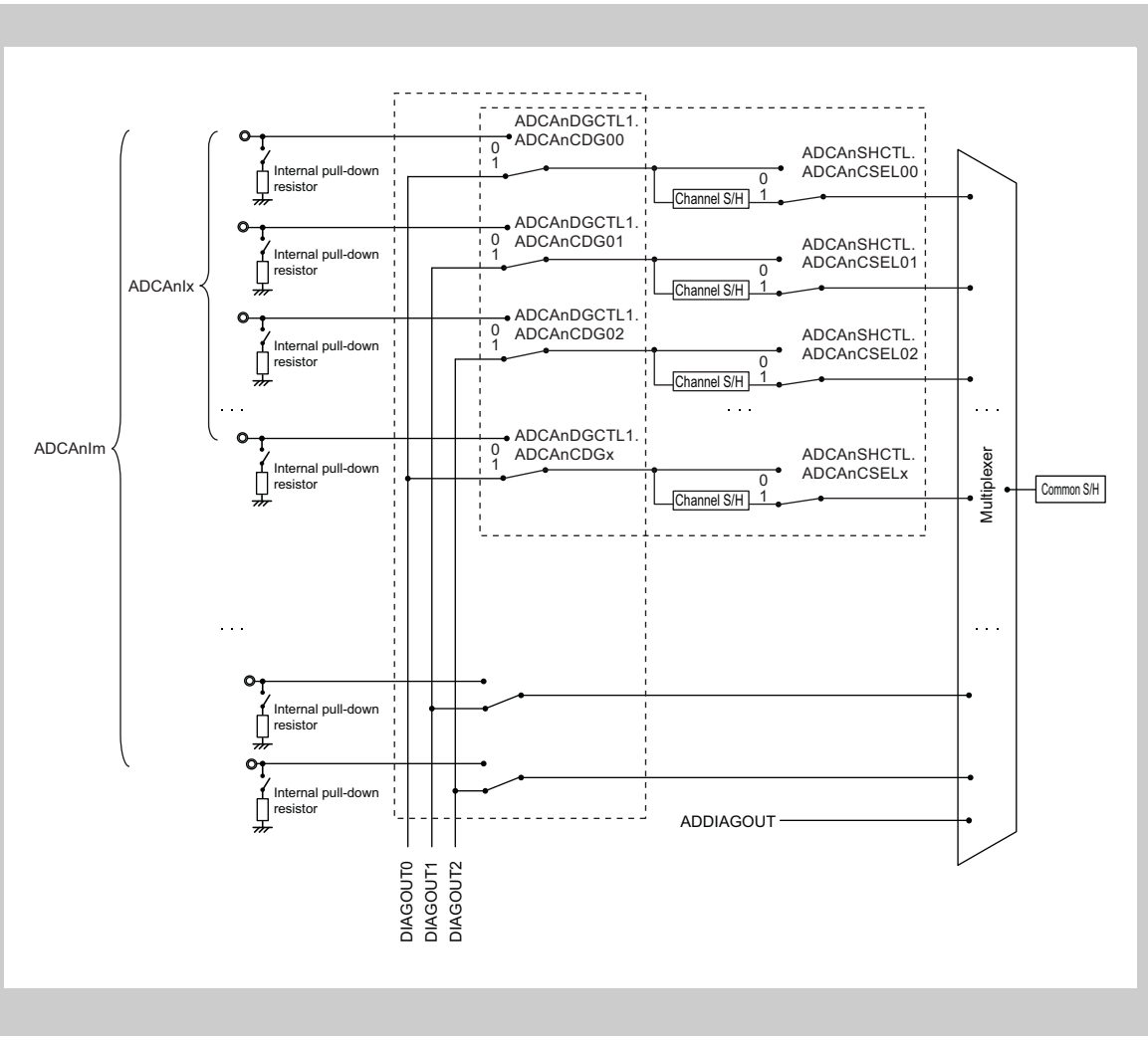


Figure 25-16 Overview of the Self-Diagnosis Functions

Note Portions enclosed in dashed lines are product dependent functions.

The following table shows when each diagnosis function can be specified:

Table 25-6 Self-Diagnosis Function Settings

Self-Diagnosis Function	ADCAnCE Bit = 0	ADCAnCE Bit = 1
A/D conversion circuit diagnosis	Specifiable	Specifiable
Channel multiplexer diagnosis	Specifiable	Not specifiable
Open pin diagnosis	Specifiable	Not specifiable
Channel sample & hold circuit diagnosis (product dependent)	Specifiable	Specifiable

(1) A/D conversion circuit diagnosis

This function makes it possible to diagnose the A/D conversion circuit operation.

The A/D conversion circuit can be diagnosed during normal A/D conversion operation. The reference voltage signal ADDIAGOUT is also converted at the end of the A/D conversion of CG0. If the result of this diagnostic A/D conversion differs significantly from the expected value, the hardware might be abnormal or operating incorrectly.

Diagnostic A/D conversion is enabled by setting ADCAnCG0.ADCAnDIAG.

Note A/D conversion circuit diagnosis is available only for CG0.

Diagnostic A/D conversion starts after the A/D conversion of the last channel of CG0 has been completed.

- The A/D conversion results of CG0 are stored in the normal A/D conversion result registers. (For details, see *(1) "A/D conversion result registers" on page 1708*.)
- The result of the diagnostic A/D conversion is stored in the ADCAnDGCR register.

Self-diagnosis procedure

1. Turn on ADCAn by setting ADCAnCTL1.ADCAnGPS.
2. Specify CG0 and the A/D conversion mode as follows:
   - Set ADCAnCG0.ADCAnDIAG to enable diagnostic A/D conversion of the reference voltage.

     For example, specify 8000 000E_H to first convert the analog input voltages of CH1, CH2, and CH3, and then the reference voltage signal ADDIAGOUT for diagnostic purposes.
   - Set ADCAnIOC0.ADCAnCG0IDG to generate the A/D conversion end interrupt INTADCAnT0 when diagnostic A/D conversion finishes.
3. Specify the reference voltage signal ADDIAGOUT for ADCAnDGCTL0.ADCAnPSEL[2:0].

   For example, set ADCAnDGCTL0.ADCAnPSEL[2:0] to 010_B to apply the reference voltage 1/2 AV_{DD}.
4. Enable ADCAn by setting ADCAnCTL0.ADCAnCE.
5. Generate a software or hardware start trigger to start A/D conversion.
6. When the A/D conversion end interrupt INTADCAnT0 is generated, read the diagnostic A/D conversion results from the ADCAnDGCR register.

ADCAnDGCTL0.ADCAnPSEL[2:0] can be written to even during A/D conversion. The following figure shows the operation when writing during A/D conversion:

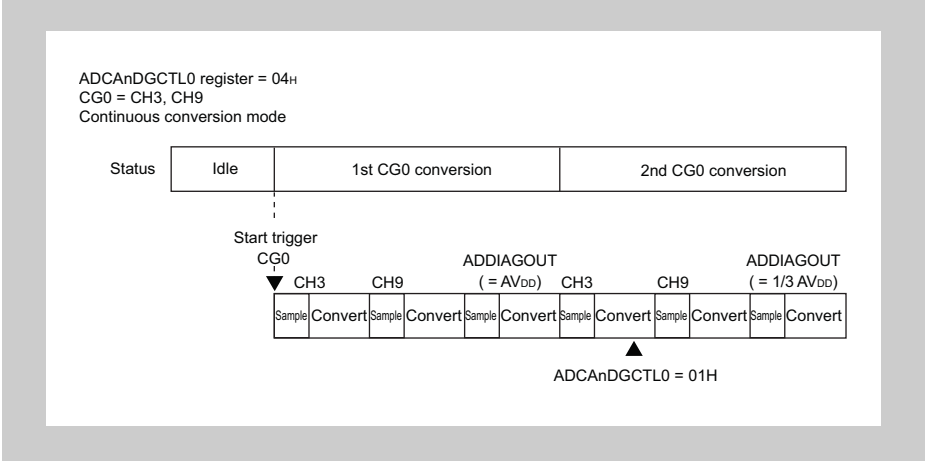


Figure 25-17 Writing During A/D Conversion

Note The value specified for ADCAnDGCTL0.ADCAnPSEL[2:0] is applied after conversion of the current channel finishes. Therefore, specify the reference voltage for the next diagnostic A/D conversion before that conversion starts.

(2) Channel multiplexer diagnosis

This function makes it possible to diagnose channel switching and selection.

Each channel is assigned one of three reference voltages. The values of the reference voltages can be changed using ADCAnDGCTL0.ADCAnPSEL[2:0].

Table 25-7 Channel assignment of reference voltages

Reference Voltage	Assigned Channels (Product Dependent)
DIAGOUT0	21, 18, 15, 12, 9, 6, 3, 0
DIAGOUT1	22, 19, 16, 13, 10, 7, 4, 1
DIAGOUT2	23, 20, 17, 14, 11, 8, 5, 2

To diagnose the channel multiplexer, different reference voltages can be input to each channel.

Self-diagnosis procedure

1. Turn on ADCAn by setting ADCAnCTL1.ADCAnGPS.
2. Specify the channels of CG0 for the ADCAnCG0 register.

   For example, specify 0000 $000E_H$ to use CH1, CH2, and CH3 for diagnosis.
3. Specify 0000 0000_H for the ADCAnI0C0 register to generate the A/D conversion end interrupt INTADCAnTi at the end of all A/D conversions for CG0.
4. Specify the other A/D conversion settings as necessary.

5. Specify the reference voltages for ADCAnDGCTL0.ADCAnPSEL[2:0].

   For example, set ADCAnDGCTL0.ADCAnPSEL[2:0] to 010_B to use the following reference voltages:
   - DIAGOUT0 = 1/2 AV_{DD}
   - DIAGOUT1 = 2/3 AV_{DD}
   - DIAGOUT2 = 1/3 AV_{DD}

6. Specify which CG0 channels a reference voltage is applied to (instead of the analog input voltage ADCAnIm) for the ADCAnDGCTL1 register. For example, set the ADCAnDGCTL1 register to $0000\ 0006_H$ to use the following channels:
   - DIAGOUT1 (2/3 AV_{DD}) is applied to CH1.
   - DIAGOUT2 (1/3 AV_{DD}) is applied to CH2.
   - The analog input voltage is applied to CH3.

7. Enable ADCAn by setting ADCAnCTL0.ADCAnCE.
8. Generate a software or hardware start trigger to start A/D conversion.
9. When the A/D conversion end interrupt INTADCAnT0 is received, read the A/D conversion results of CG0.

Caution When ADCAnCTL0.ADCAnCE is set, changing the setting of the ADCAnDGCTL1 register is prohibited.

(3) Open pin diagnosis

When an input pin is open, the correct A/D conversion results cannot be obtained.

The internal pull-down resistor can be connected to diagnose the analog input ADCAnIm.

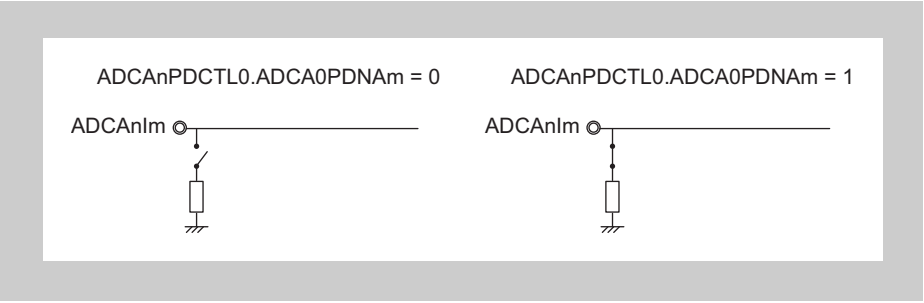


Figure 25-18 Internal Pull-Down Resistor

When the internal pull-down resistor is connected to the analog input ADCAnIm (by setting ADCAnPDCTL0.ADCAnPDNAm) and ADCAnIm is open, the A/D conversion result is almost 0 V.

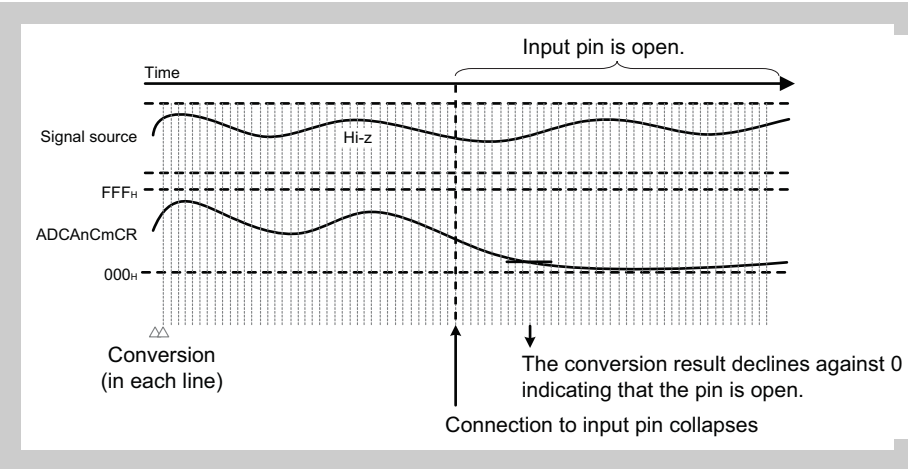


Figure 25-19 Detection of an Open Input Pin

Notes

1. If the analog input ADCAnIm is less than 0.2 V (target), an open pin cannot be detected. (For details, see *Table 37 "Electrical Specifications"*.)
2. Do not connect the internal pull-down resistor during normal A/D conversion operation. Connecting this resistor might lead to a drop in the input voltage, and it might not be possible to obtain the correct A/D conversion results.

Self-diagnosis procedure

To diagnose open input pins:

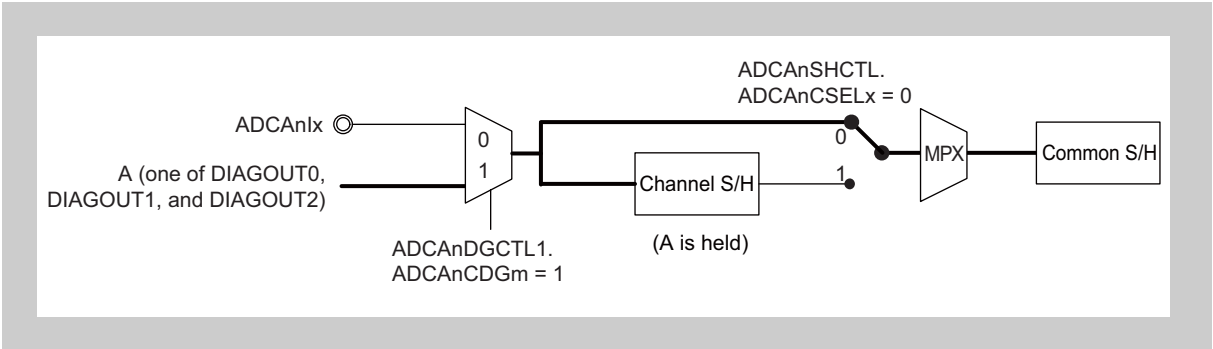
1. Specify the CG and A/D conversion settings as usual.
2. Set the ADCAnPDCTL0 register to connect the internal pull-down resistor.
3. Execute multiple A/D conversions (target).
4. Monitor the channel's A/D conversion results and check whether any result is near 0 V.

(4) Channel sample & hold circuit diagnosis (product dependent)

For details about the channel sample & hold function, see *25.3.14 “Channel sample & hold circuit function (product dependent)” on page 1721* .

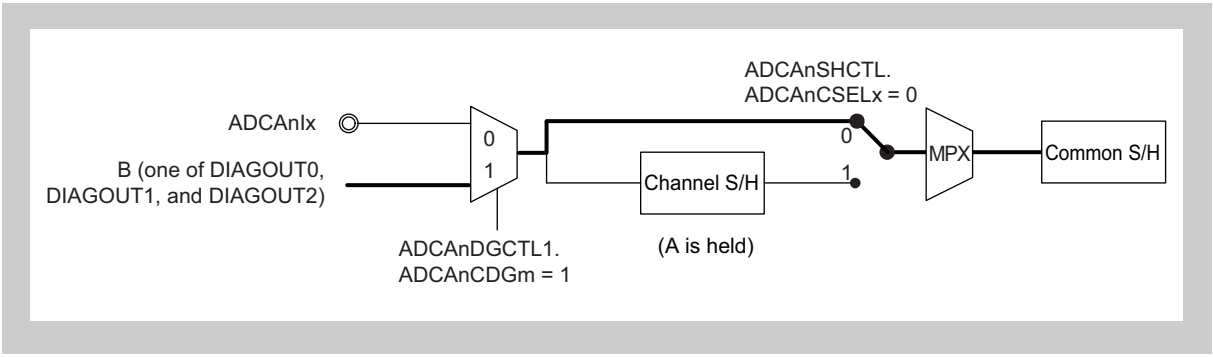
This function makes it possible to diagnose the channel sample & hold circuit. The following provides an overview.

1. One of the reference voltage signals from DIAGOUT0 to DIAGOUT2 is used.
2. The voltage "A" is specified for one of the reference voltage signals from DIAGOUT0 to DIAGOUT2. The channel sample & hold circuit holds “A”, and conversion is performed without using this circuit.



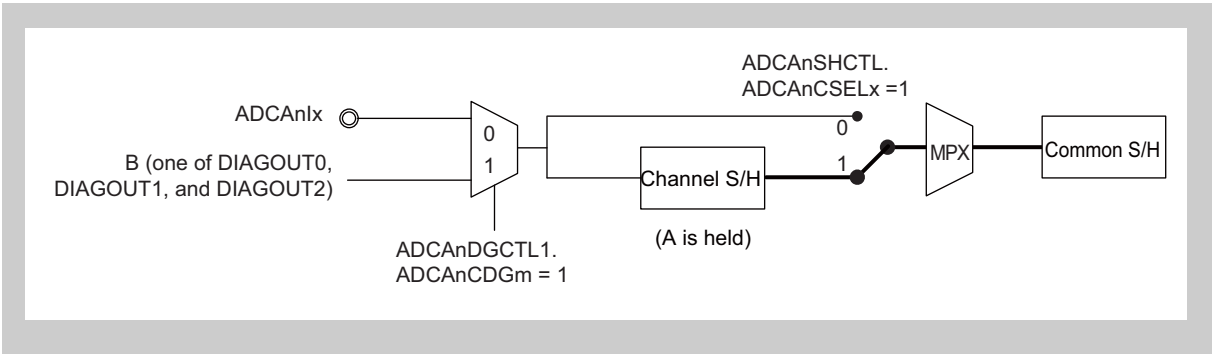
3. The voltage “B” is specified for one of the reference voltage signals from DIAGOUT0 to DIAGOUT2.

   Conversion is performed without using the channel sample & hold circuit. The conversion result is “B”.
   The channel sample & hold circuit continues holding “A”.



4. Conversion is performed by using the channel sample & hold circuit.

   The channel sample & hold circuit continues holding “A”. The conversion result is “A”.



5. The results are as follows:
   – The first conversion result (in (3)) is “B”.
   – The second conversion result (in (4)) is “A”.

The diagnostic flow for the channel sample & hold circuit is shown below.

Initial setup

The following is an example for the ADCAnI1 pin:

- Set ADCAnCTL1.ADCAnGPS.
- Specify a software trigger as the start trigger, and set the A/D conversion mode to the one-shot conversion mode (number of repetitions: 1).
- Set the ADCAnCG0 register to 0000 0002_H (to select ADCAnI1, which has the channel sample & hold circuit).
- Set the ADCAnIOC0 register to 0000 0002_H to generate the A/D conversion end interrupt INTADCAnT0. (An interrupt must be generated when conversion ends.)
- Set ADCAnDGCTL0.ADCAnPSEL[2:0] to 001_B (to select 1/3 AV_{DD} for the reference voltage DIAGOUT1).
- Set ADCAnDGCTL1.ADCAnCDG01.
- Set ADCAnCTL0.ADCAnCE.

Operation flow

Step 1

Input the first software start trigger for CG0. (1/3 AV_{DD} is held by the channel sample & hold circuit.)

Next, change the ADCAnPSEL[2:0] bits to 010_B. (After the first A/D conversion ends, 2/3 AV_{DD} is selected for the reference voltage DIAGOUT1.)

Input the second software start trigger for CG0. (The start trigger is held.)

The procedure up to this point must be performed before the first A/D conversion ends.

Step 2

The first A/D conversion ends, and the A/D conversion end interrupt INTADCAnT0 is generated.

Change the ADCAnCSEL01 bit to 1. (During the next A/D conversion, the voltage held by the channel sample & hold circuit is converted.)

Input the third software start trigger for CG0. (The start trigger is held.)

The procedure up to this point must be performed before the second A/D conversion ends.

Step 3

The second A/D conversion ends, and the A/D conversion end interrupt INTADCAnT0 is generated.

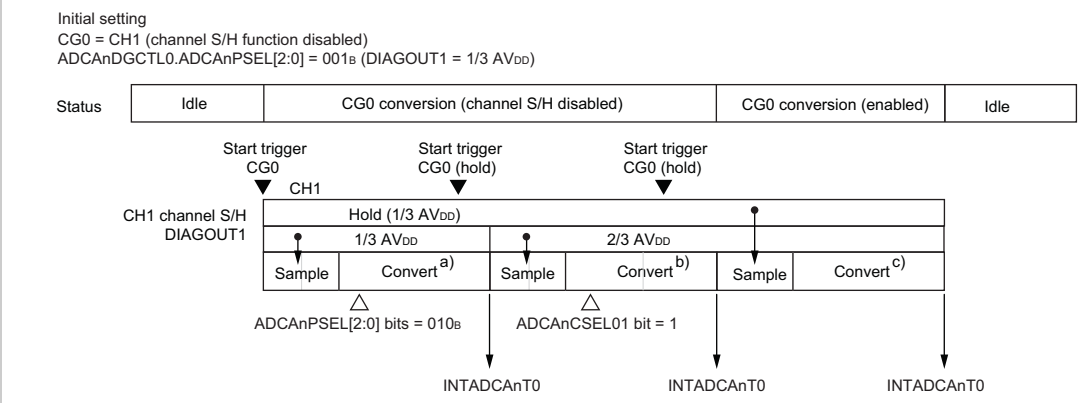
Read the ADCAnC01CR register, and make sure that the expected value (2/3 AV_{DD}) has been stored there.

Step 4

The third A/D conversion ends, and the A/D conversion end interrupt INTADCAnT0 is generated.

Read the ADCAnC01CR register, and make sure that the expected value (1/3 AV_{DD}) has been stored there.

An example of timing for the channel sample & hold circuit is shown below.

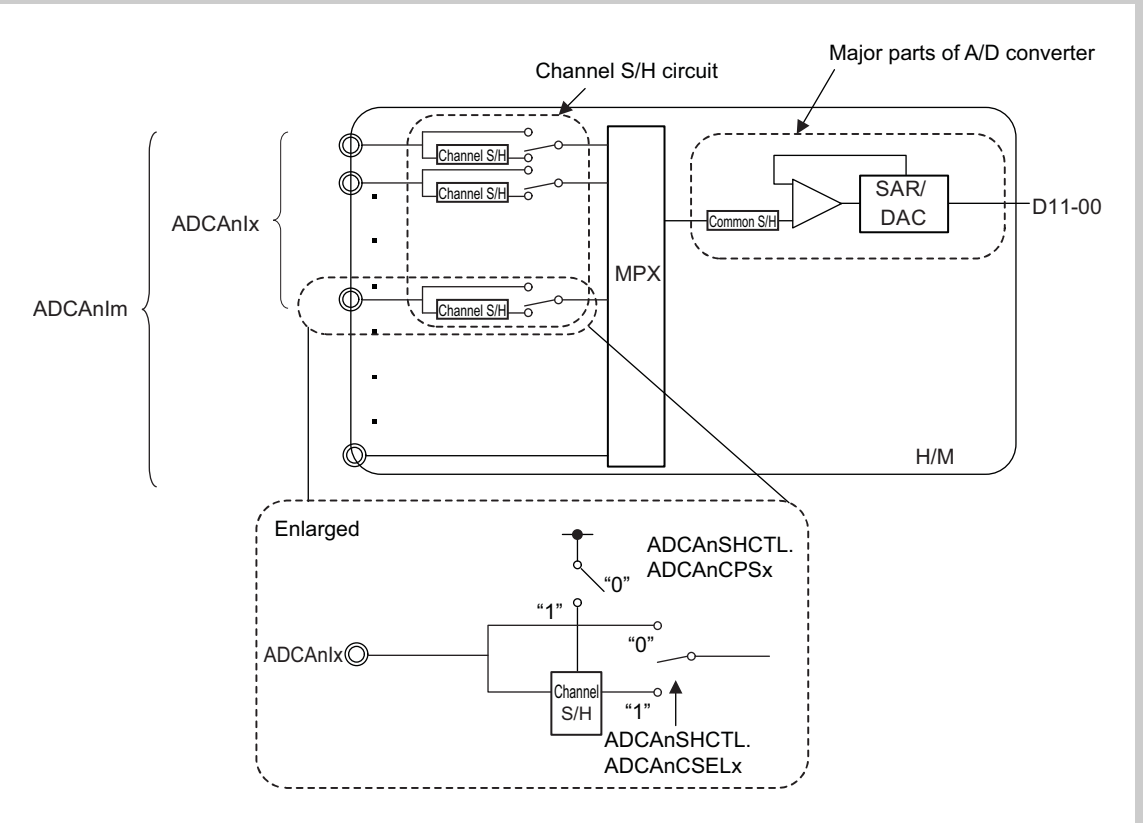


a) Convert DIAGOUT1.The conversion result is 1/3 AV_{DD}.
b) Convert DIAGOUT1.The conversion result is 2/3 AV_{DD}.
c) Convert the input voltage held by the channel sample & hold circuit. The conversion result is 1/3 AV_{DD}.

25.3.14 Channel sample & hold circuit function (product dependent)

(1) Channel sample & hold circuit function

The channel sample & hold circuit is shown below.



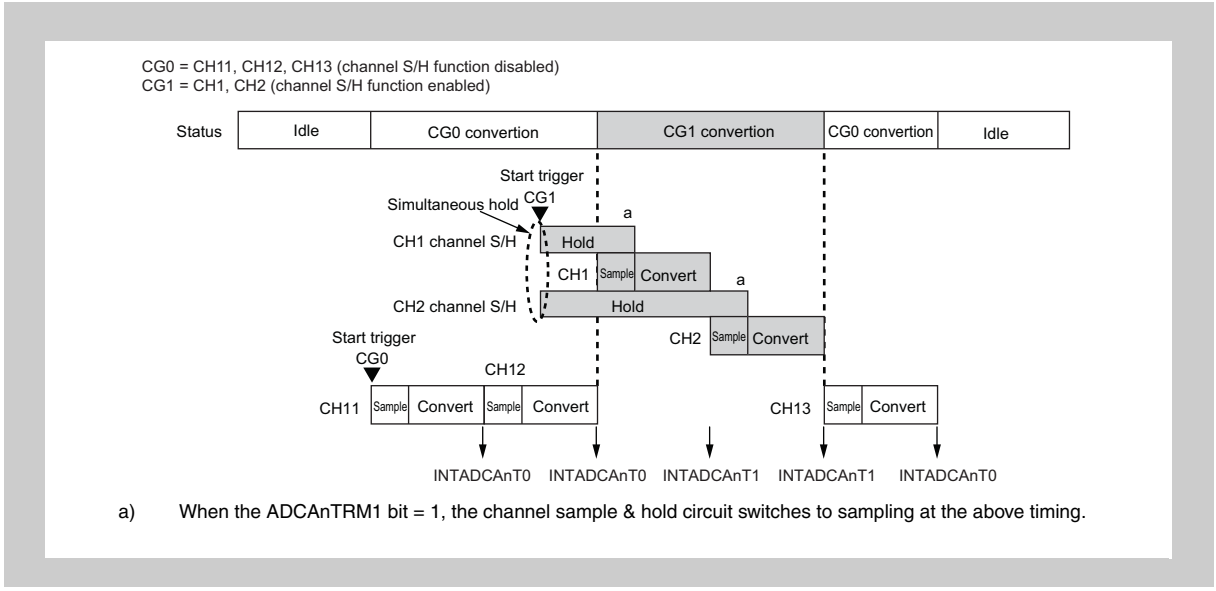
The channel sample & hold function can be used for CG0 in the one-shot conversion mode (no repetition), as well as for CG1 and CG2 (no repetition).

ADCAnSHCTL.ADCAnCPSx is used to control the supply of power to the channel sample & hold circuit, and DCAnSHCTL.ADCAnCSELx is used to enable or disable the channel sample & hold function.

Caution Make sure to turn on the amplifier power when using the channel S/H circuit.

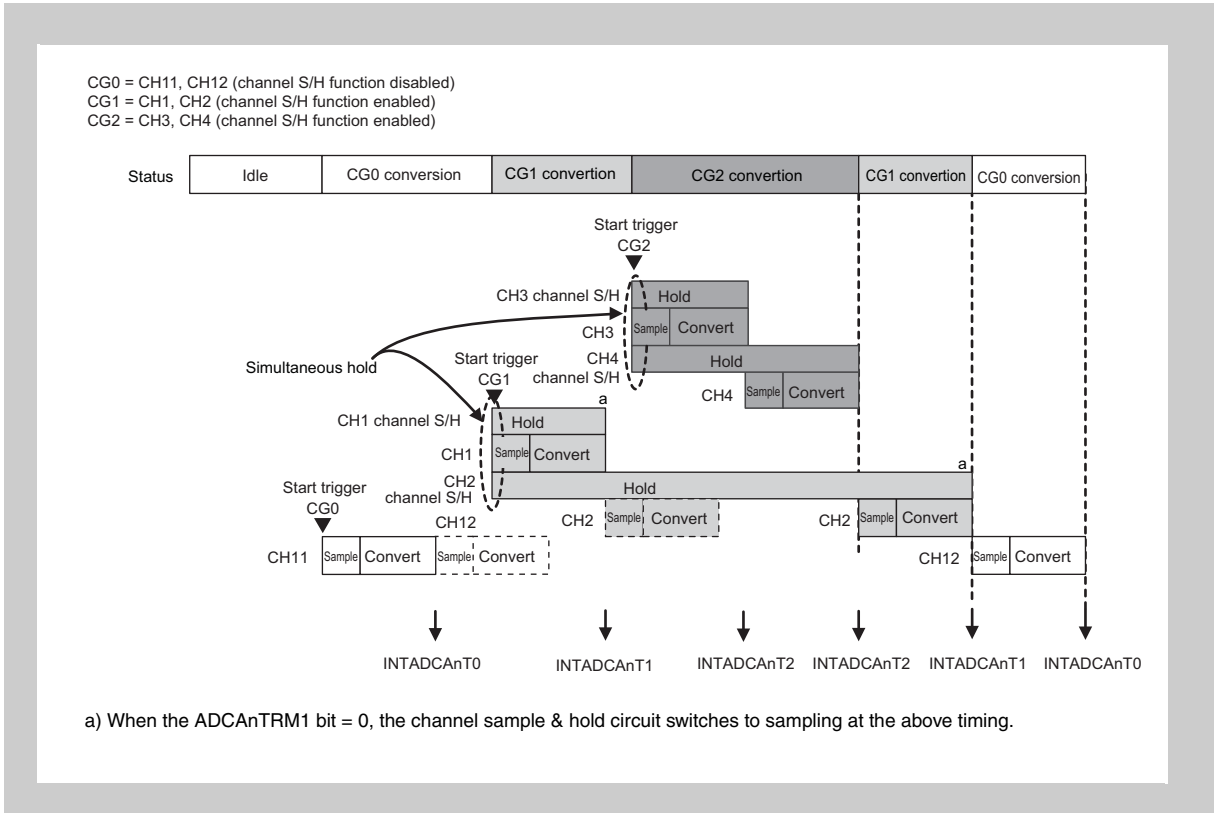
When a hardware or software start trigger is generated, the channel sample & hold circuit holds the analog input signal of the channel for which the channel sample & hold function was enabled by using ADCAnSHCTL.ADCAnCSELx. Next, scan list conversion is started according to the ADCAnCTL1.ADCAnTRMi setting.

Example 1: When a high-priority start trigger is generated (the ADCAnTRM0 bit = 1 and the ADCAnTRM1 bit = 1)



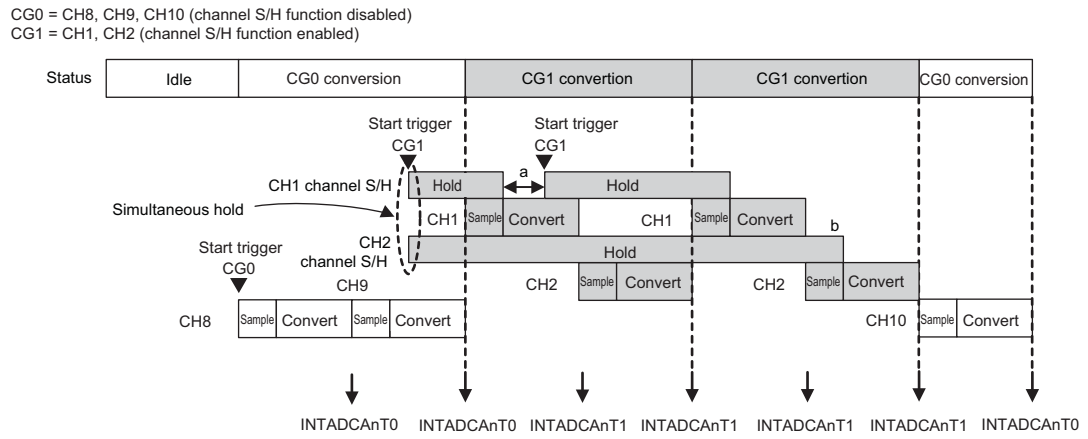
a) When the ADCAnTRM1 bit = 1, the channel sample & hold circuit switches to sampling at the above timing.

Example 2: When a high-priority start trigger is generated (the ADCAnTRM0 bit = 0 and the ADCAnTRM1 bit = 0)



a) When the ADCAnTRM1 bit = 0, the channel sample & hold circuit switches to sampling at the above timing.

Example 3: When an additional start trigger is generated (the ADCAnTRM0 bit = 1 and the ADCAnTRM1 bit = 1)

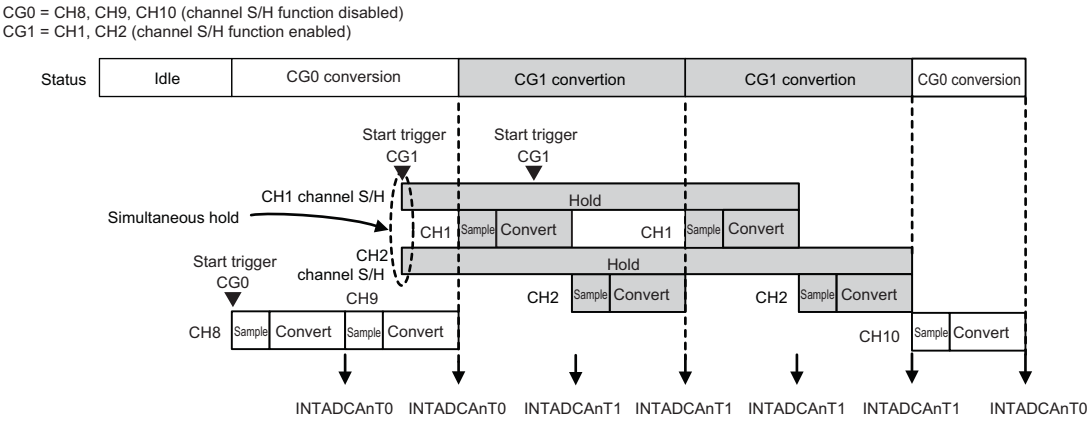


a) It is necessary to consider the sampling time for the channel sample & hold circuit. Secure a time of at least 1.5 µs.

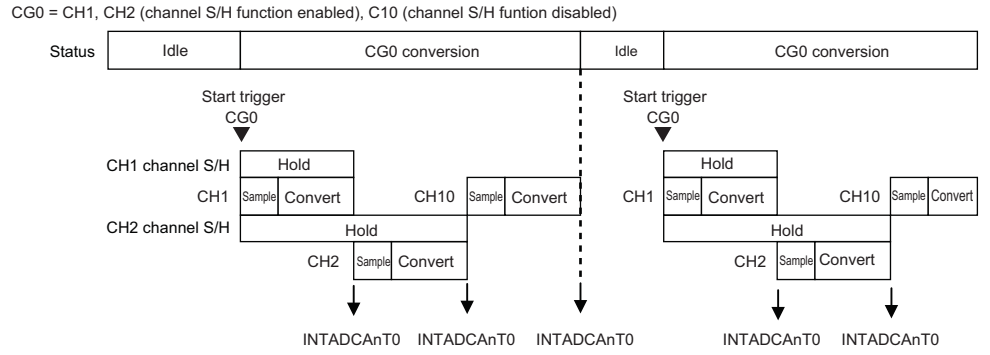
b) When a start trigger is generated, the channel sample & hold circuit continues holding the current analog input until

Example 4: When an additional start trigger is generated (the ADCAnTRM0 bit = 1 and the ADCAnTRM1 bit = 0)

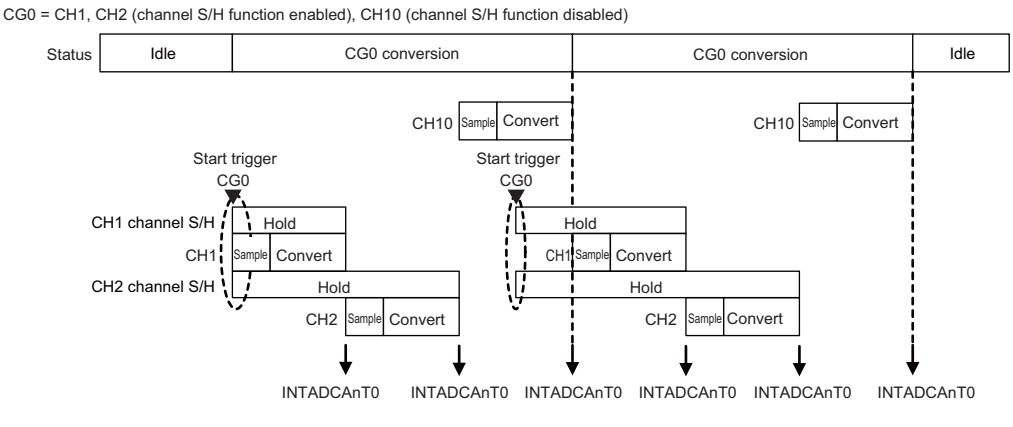
When an additional start trigger is generated during A/D conversion, the channel sample & hold circuit does not perform new sampling.



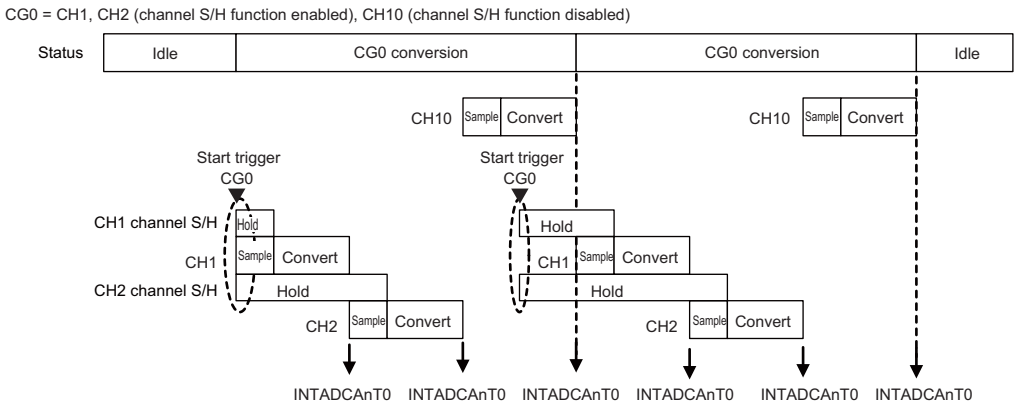
Example 5: When CG0 is in the one-shot conversion mode (no repetition) (the ADCAnTRM0 bit = 0)



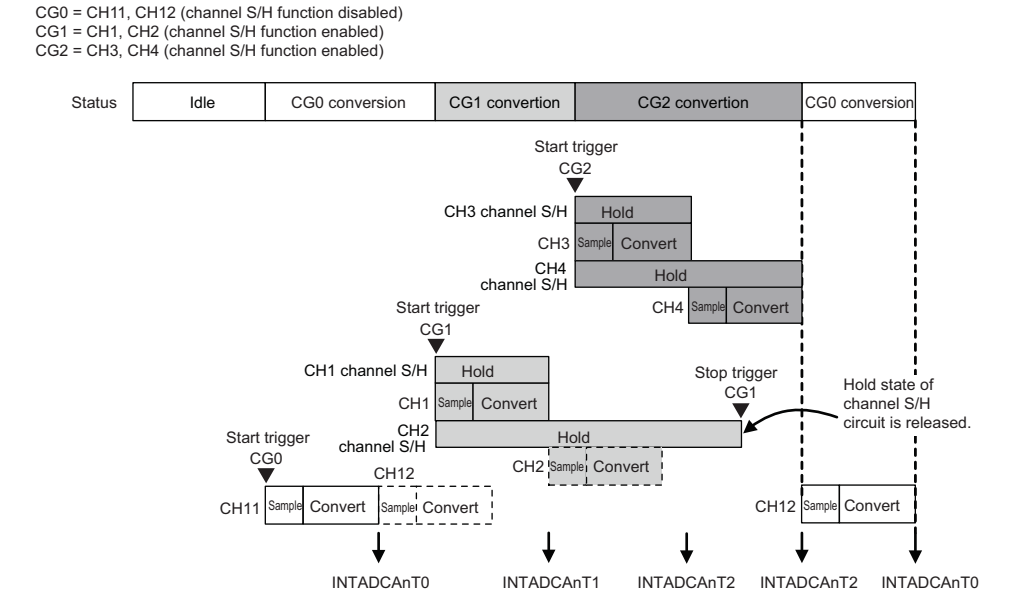
Example 6: When CG0 is in the one-shot conversion mode (no repetition), and an additional start trigger is generated (the ADCAnTRM0 bit = 0)



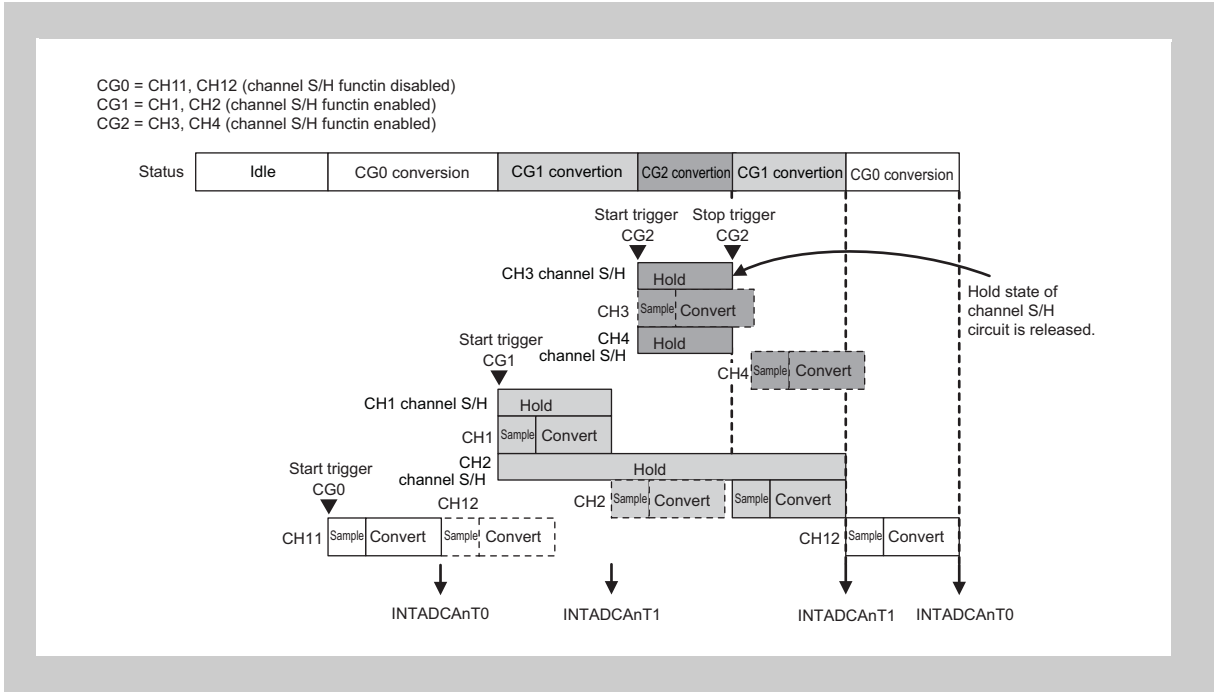
Example 7: When CG0 is in the one-shot conversion mode (no repetition), and an additional start trigger is generated (the ADCAnTRM0 bit = 1)



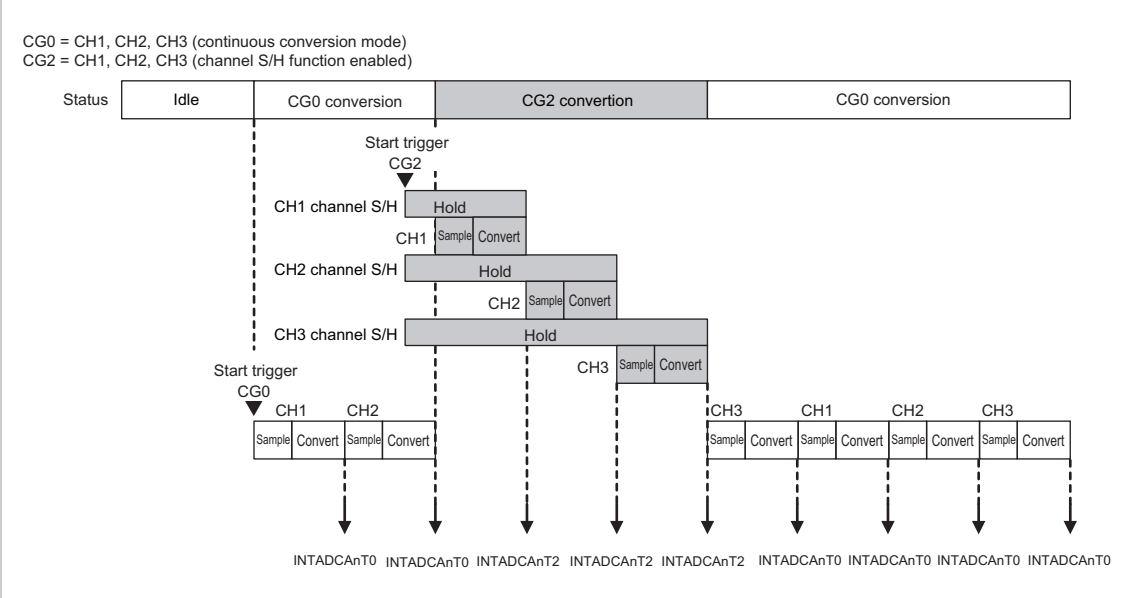
Example 8: When different channels are specified for CG1 and CG2, and a CG1 stop trigger is generated (the ADCAnTRM0 bit = 0 and the ADCAnTRM1 bit = 0)



Example 9: When different channels are specified for CG1 and CG2, and a CG2 stop trigger is generated (the ADCAnTRM0 bit = 0 and the ADCAnTRM1 bit = 0)



Example 10: When CG0 is in the continuous conversion mode, and the same channel is specified for CG0 and CG2 (the ADCAnTRM0 bit = 1 and the ADCAnTRM2 bit = 0)



(2) Restrictions when using the channel sample & hold function

The following describes the restrictions when using the channel sample & hold function.

1. Do not specify the channel used for the channel sample & hold function for multiple CGs at the same time.

Example 1: When using the channel sample & hold function for CG0, CG1, and CG2 (and CG0 is in the one-shot conversion mode)

The following combinations can be specified:

– CG0 (with CH1 selected), CG1 (with CH2 selected), and CG2 (with CH3 selected)

Specifying the following combinations is prohibited:

– CG0 (with CH1 selected), CG1 (with CH1 selected), and CG2 (with CH2 and CH3 selected)

Example 2: When using the channel sample & hold function for CG1 and CG2 (and CG0 is in the continuous conversion mode)

The following combinations can be specified:

– CG0 (with CH1 selected), CG1 (with CH1 and CH2 selected), and CG2 (with CH3 selected)

– CG0 (with CH1, CH2, and CH3 selected), CG1 (with CH1 selected), and CG2 (with CH2 and CH3 selected)

Specifying the following combinations is prohibited:

– CG0 (with CH1, CH2, and CH3 selected), CG1 (with CH1 and CH2 selected), and CG2 (with CH2 and CH3 selected)

2. When changing the scan list during operation for a CGi using the channel sample & hold function, do not change a channel subject to the function.

Example: When using the channel sample & hold function for CH1, CH2, and CH3

The following combinations can be specified:

– Changing CG0 (CH1, CH2, and CH3) to CG0 (CH1, CH2, CH3, CH10, and CH11)

– Changing CG0 (CH1, CH2, CH3, CH10, and CH11) to CG0 (CH1, CH2, and CH3)

– Changing CG0 (CH7, CH8, and CH9) to CG0 (CH10, CH11, and CH12)

Specifying the following combinations is prohibited:

– Changing CG0 (CH1, CH2, and CH3) to CG0 (CH1 and CH2)

– Changing CG0 (CH1) to CG0 (CH1, CH2, and CH3)

– Changing CG0 (CH7, CH8, and CH9) to CG0 (CH1, CH7, CH8, and CH9)

– Changing CG0 (CH1, CH2, and CH9) to CG0 (CH9, CH10, and CH11)

3. Using repetition for CGs using the channel sample & hold function is prohibited. When using the channel sample & hold function for CGi, clear the bits that specify the number of repetitions for that CG (ADCAnCTL0.ADCAnSCTi[1:0]) to 00_B.

25.3.15 Discharge function (product dependent)

If required, the internal capacitor of the common sample & hold circuit can be discharged before every conversion.

Note Enabling the discharge function increases the total conversion time by one clock cycle (ADCAnTCLK). (For details, see *25.3.9 "Resolution, sampling, and conversion times" on page 1704* .)

Setting To enable the discharge function, set ADCAnCTL1.DCAnDISC.

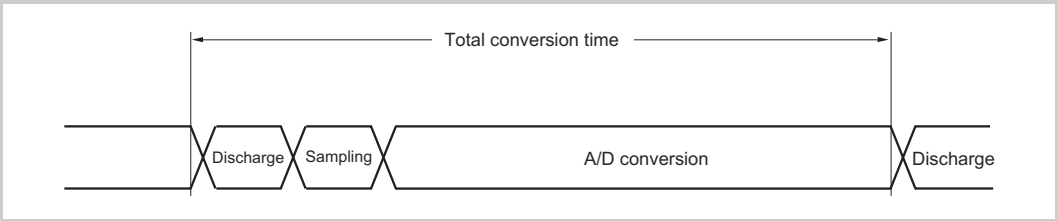


Figure 25-20 Conversion Timing When the Discharge Function Is Enabled

25.3.16 Buffer amplifier function

If required, the analog input signal can be connected to an internal buffer amplifier. The buffer amplifier accelerates the charge rate of the internal sampling capacitor during the A/D sampling period.

To enable the buffer amplifier function, set ADCAnCTL1.ADCAnBPC.

Note Enabling the buffer amplifier function increases the total conversion time by four clock cycles (ADCAnTCLK). (For details, see *25.3.9 "Resolution, sampling, and conversion times" on page 1704* .)

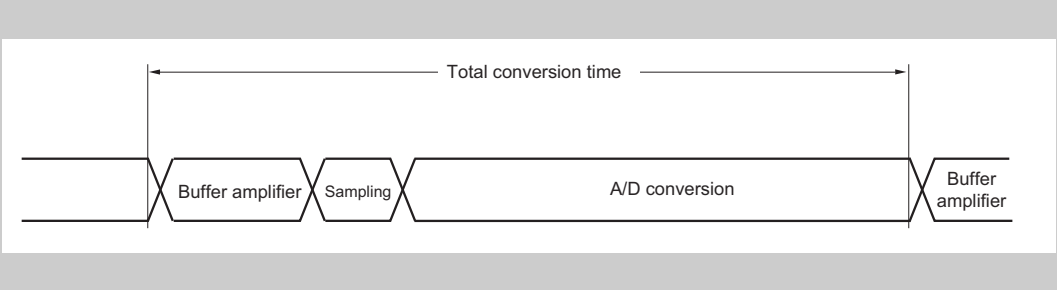


Figure 25-21 Conversion Timing When the Buffer Amplifier Function Is Enabled

25.3.17 Stabilization control

The A/D converter needs time for stabilization in the following cases:

- When the A/D converter is turned on (by setting ADCAnCTL1.ADCAnGPS)
- When the stand-by mode is exited

Start triggers are acknowledged even during the stabilization time, but conversion will not start until the stabilization time elapses.

To secure the minimum stabilization time, the stabilization time counter ADCAnCNT must be set.

25.4 Registers

This section describes all the registers of ADCAn.

25.4.1 Overview of the ADCAn registers

ADCAn is controlled and operated by the registers in the table below.

- If there is one register per channel, the channel number is indicated by an “m”.
- If there is one register per CG, the CG number is indicated by an “i” (where i = 0 to 2).

Table 25-8 ADCAn Registers (Product Dependent) (1/2)

Register Name	Symbol	Address
Control registers		
A/D converter mode control register 0	ADCAnCTL0	<ADCAn_base_OS> + 100_H
A/D converter mode control register 1	ADCAnCTL1	<ADCAn_base_OS> + 104_H
A/D converter CG register i	ADCAnCGi	<ADCAn_base_USER> + i×4_H
A/D converter interrupt control register i	ADCAnIOCi	<ADCAn_base_USER> + C_H + i×4_H
A/D converter trigger select control register i	ADCAnTSELi	<ADCAn_base_OS> + 108_H + i×4_H
A/D converter stabilization counter	ADCAnCNT	<ADCAn_base_OS> + 114_H
Conversion status registers		
A/D converter overwrite error flag register	ADCAnSTR1	<ADCAn_base_USER> + 28_H
ADCAnSTR1 flag clear register	ADCAnSTC1	<ADCAn_base_USER> + 34_H
A/D converter status flag register 2	ADCAnSTR2	<ADCAn_base_USER> + $2C_H$
ADCAnSTR2 flag clear register	ADCAnSTC2	<ADCAn_base_USER> + 38_H
Software trigger registers		
A/D converter software trigger register i	ADCAnTRGi	<ADCAn_base_USER> + $A4_H$ + i×4_H
A/D converter software trigger register 3	ADCAnTRG3	<ADCAn_base_USER> + $B0_H$
A/D converter software trigger register 4+i	ADCAnTRG4+i	<ADCAn_base_USER> + $B4_H$ + i×4_H
A/D converter software trigger register 7	ADCAnTRG7	<ADCAn_base_USER> + $C0_H$
A/D conversion result registers		
A/D converter latest conversion result register	ADCAnLCR	<ADCAn_base_USER> + $A0_H$
A/D converter conversion result register m	ADCAnCmCR	<ADCAn_base_USER>?3C_H + m×4_H
A/D converter CGi buffer result register i	ADCAnDBiCR	<ADCAn_base_USER> + $C4_H$ + i×4_H
DMA Buffer register of CGi conversion register (product dependent)	ADCAnDBiCRL	<ADCAn_base_USER> + $D0_H$ + i×4_H
A/D converter diagnostic conversion result register	ADCAnDGCR	<ADCAn_base_USER> + $9C_H$
A/D conversion limit comparison registers		
A/D converter result check register	ADCAnCTL2	<ADCAn_base_USER> + 18_H
A/D converter result check (upper limit)	ADCAnUL	<ADCAn_base_USER> + $1C_H$
A/D converter result check (lower limit)	ADCAnLL	<ADCAn_base_USER> + 20_H
A/D converter result check error flag	ADCAnSTR0	<ADCAn_base_USER> + 24_H
ADCAnSTR0 flag clear register	ADCAnSTC0	<ADCAn_base_USER> + 30_H

Table 25-8 ADCAn Registers (Product Dependent) (2/2)

Register Name	Symbol	Address
Diagnostic function control registers (product dependent)		
A/D converter self-diagnosis function control register 0	ADCAnDGCTL0	<ADCAn_base_USER> + DC_H
A/D converter self-diagnosis function control register 1	ADCAnDGCTL1	<ADCAn_base_OS> + $11C_H$
A/D converter internal pull-down resistor control register 0	ADCAnPDCTL0	<ADCAn_base_OS> + 120_H
Channel sample & hold function setup register (product dependent)		
A/D converter channel sample & hold control register (product dependent)	ADCAnSHCTL	<ADCAn_base_OS> + 118_H

25.4.2 Control registers

(1) ADCAnCTL0 - A/D converter mode control register 0

This register enables or disables the A/D converter. In addition, it specifies the number of repetitions in the one-shot conversion mode and whether to generate error interrupt requests when an A/D conversion result is overwritten before it is read.

Access This register can be read or written in 16-bit units.

Address <ADCAn_base_OS> + 100_H

Initial value 0000_H. This register is initialized by any reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	ADCAn OEM4	ADCAnOEM[3:1]			ADCAn OEM0	ADCAn CE	0	ADCAnSCT2 [1:0]		ADCAnSCT1 [1:0]		ADCAnSCT0 [1:0]	
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

Table 25-9 ADCAnCTL0 Register Contents (1/2)

Bit Position	Bit Name	Description
12	ADCAn OEM4	This bit specifies whether the error interrupt INTADCAnTERR is generated when an A/D conversion result in the ADCAnLCR register is overwritten before it is read. 0: Generate the error interrupt INTADCAnTERR when an A/D conversion result is overwritten. 1: Do not generate the error interrupt INTADCAnTERR. For details, see *(1) "Conversion result overwrite check function" on page 1711*.
11 to 9	ADCAn OEM[3:1]	These bits specify whether the error interrupt INTADCAnTERR is generated when an A/D conversion result in an ADCAnDBiCR register is overwritten before it is read. 0: Generate the error interrupt INTADCAnTERR when an A/D conversion result is overwritten. 1: Do not generate the error interrupt INTADCAnTERR. CGI is controlled by the ADCAnOEM(i+1) bit. For details, see *(1) "Conversion result overwrite check function" on page 1711*.

Table 25-9 ADCAnCTL0 Register Contents (2/2)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Description</th></tr>
<tr><td>8</td><td>ADCAn OEM0</td><td>This bit specifies whether the error interrupt INTADCAnTERR is generated when an A/D conversion result in the ADCAnCmCR register is overwritten before it is read.
0: Generate the error interrupt INTADCAnTERR when an A/D conversion result is overwritten.
1: Do not generate the error interrupt INTADCAnTERR.
For details, see (1) "Conversion result overwrite check function" on page 1711.</td></tr>
<tr><td>7</td><td>ADCAnCE</td><td>This bit enables or disables the A/D converter.
0: Disable the A/D converter.
1: Enable the A/D converter.
Note that A/D conversion only starts when there is a hardware or software trigger (ADCAnTRGi.ADCAnSTTi) if ADCAnCTL0.ADCAnCE is set to 1. Also note that the A/D converter needs time to stabilize after it has been enabled. Start triggers are acknowledged even immediately after turning the power on. A/D conversion starts after the stabilization counter ADCAnCNT reaches 00_H.</td></tr>
<tr><td>5 to 0</td><td>ADCAn SCTi[1:0]</td><td>These bits specify the number of scan list conversions for CG1, CG2, and CG0 while it is in the one-shot conversion mode.
<table>
<tr><th>ADCAn SCTi1</th><th>ADCAn SCTi0</th><th>Number of CGi Scan List Conversions</th></tr>
<tr><td>0</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>2</td></tr>
<tr><td>1</td><td>0</td><td>3</td></tr>
<tr><td>1</td><td>1</td><td>4</td></tr>
</table>
</td></tr>
</table>

(2) ADCAnCTL1 – A/D converter mode control register 1

This register specifies the conversion mode and controls the conversion operations.

Access This register can be read or written in 32-bit units.

Address <ADCAn_base_OS> + 104_H

Initial value $0100\ 0008_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADCAnT2ETS [1:0]		ADCAnT1ETS [1:0]		ADCAnT0ETS [1:0]		0	ADCAn CRAC	0	0	ADCAn MD1	ADCAn MD0	0	0	ADCAn DISC	ADCAn RCL
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCAn CTYP	0	0	ADCA nSTL	ADCAnFR[3:0]				0	ADCAnTRM[2:0]			ADCAn BPC	0	0	ADCAn GPS
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25-10 ADCAnCTL1 Register Contents (1/3)

Bit Position	Bit Name	Description
31 to 26	ADCAn TiETS[1:0]	These bits specify the valid edge of the hardware trigger signal ADCAnTTRGi. ADCAn TiETS1 / ADCAn TiETS0 / Valid Edge 0 / 0 / No valid edge detection (no acknowledgment) 0 / 1 / Rising edge 1 / 0 / Falling edge 1 / 1 / Rising and falling edges
24	ADCAn CRAC	This bit specifies the alignment of the A/D conversion and diagnostic conversion results. 0: Right-aligned 1: Left-aligned
21	ADCAnMD1	This bit specifies the A/D conversion start trigger for all CGs. 0: Software trigger 1: Hardware trigger and software trigger This setting is valid for all CGs. Triggers are only detected when the A/D converter is enabled. For details, see *25.3.5 "Starting A/D conversion (start triggers)" on page 1699* .
20	ADCAnMD0	This bit specifies the A/D conversion mode for CG0. 0: One-shot conversion mode The number of repetitions is specified by ADCAnCTL0.ADCAnSCTi[1:0] for each CG. 1: Continuous conversion mode This setting applies to the A/D conversion of CG0 only. CG1 and CG2 are always operated in the one-shot conversion mode. For details, see *25.3.4 "A/D conversion modes" on page 1696* .
17	ADCAnDISC	This bit enables or disables the discharge function. 0: Disable 1: Enable For details, see *25.3.15 "Discharge function (product dependent)" on page 1728* .

Table 25-10 ADCAnCTL1 Register Contents (2/3)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Description</th></tr>
<tr><td>16</td><td>ADCAnRCL</td><td>This bit specifies whether the A/D conversion results in ADCAnCmCR and ADCAnDBiCR are retained after reading them.
0: Retain the A/D conversion result until it is overwritten by the next A/D conversion result.
1: Clear the A/D conversion result after reading it.</td></tr>
<tr><td>15</td><td>ADCAnCTYP</td><td>This bit specifies the resolution mode.
0: 12-bit resolution (product dependent)
1: 10-bit resolution</td></tr>
<tr><td>12</td><td>ADCAnSTL</td><td>This bit specifies the ADCAnCNVi signal level.
0: When ADCAnCNVi = L, CGi is not undergoing conversion.
When ADCAnCNVi = H, CGi is undergoing conversion.
1: When ADCAnCNVi = H, CGi is not undergoing conversion.
When ADCAnCNVi = L, CGi is undergoing conversion.</td></tr>
<tr><td>11 to 8</td><td>ADCAnFR [3:0]</td><td>These bits specify the ADCAn clock ADCAnTCLK.
<table>
<tr><th>ADCAnFR[3:0]</th><th>ADCAn Clock</th></tr>
<tr><td>0000</td><td>PCLK/2</td></tr>
<tr><td>0001</td><td>PCLK/3</td></tr>
<tr><td>0010</td><td>PCLK/4</td></tr>
<tr><td>0011</td><td>PCLK/5</td></tr>
<tr><td>0100</td><td>PCLK/6</td></tr>
<tr><td>0110</td><td>PCLK/8</td></tr>
<tr><td>1000</td><td>PCLK/10</td></tr>
<tr><td>1010</td><td>PCLK/12</td></tr>
<tr><td>1100</td><td>PCLK/14</td></tr>
<tr><td>1110</td><td>PCLK/16</td></tr>
<tr><td>Other than the above</td><td>Setting prohibited</td></tr>
</table></td></tr>
</table>

Table 25-10 ADCAnCTL1 Register Contents (3/3)

Bit Position	Bit Name	Description
6 to 4	ADCAnTRMi (product dependent)	These bits specify the interrupt behavior when the start trigger for the A/D conversion of a higher priority CG is input (or when transitioning to the ADCHALT mode is requested). 0: Interrupt the current A/D conversion of CGi, and start the A/D conversion of the higher priority CG (or enter the ADCHALT mode). 1: Finish the current CGi channel conversion, interrupt A/D conversion of the CG, and start the A/D conversion of the higher priority CG (or enter the ADCHALT mode). A/D conversion of CGi is continued as soon as all pending A/D conversions of higher priority CGs have been completed (or the ADCHALT mode has been exited). The priority is as follows: ADCHALT > CG2 > CG1 > CG0 For details, see *(1) "Order of A/D conversion" on page 1695*.
3	ADCAnBPC	This bit enables or disables the buffer amplifier function. 0: Disable 1: Enable For details, see *25.3.16 “Buffer amplifier function” on page 1728* .
0	ADCAnGPS	This bit turns ADCAn on or off. 0: Power off 1: Power on The A/D converter needs time to stabilize after being turned on. (For details, see *25.3.17 “Stabilization control” on page 1729*).

(3) ADCAnCGi – A/D converter channel group register i

This register is used to create a scan list for the corresponding CG. The channels specified in the scan list are converted in ascending order. For details, see *25.3.3 “Channels and channel groups” on page 1694* .

In addition, ADCAnCG0.ADCAnDIAG can be used to enable or disable the diagnosis of A/D conversion that uses the reference voltage signal ADDIAGOUT. For details, see *(1) "A/D conversion circuit diagnosis" on page 1714*.

Access This register can be read or written in 32-bit units. Because this register has a master/slave configuration, a new A/D conversion channel can be specified for the master register during A/D conversion. The timing at which the master register value is transferred to the slave register is as follows:

- If CGi is not undergoing A/D conversion, the value is transferred one clock cycle (PCLK) after writing to the master register.
- If CGi is undergoing A/D conversion, the value is transferred when the CGi scan list conversion currently being executed ends.
- If the CGi stop trigger bit (the ADCAnSPi bit) is set after a write to this register, the value is transferred when A/D conversion stops.

Address <ADCAn_base_USER> + i × 4_H

Initial value 0000 0000_H. This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADCAn DIAG	0	0	0	0	0	0	0	ADCAnCGiS[23:16]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCAnCGiS[15:00]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25-11 ADCAnCGi Register Contents

Bit Position	Bit Name	Description
31	ADCAnDIAG	This bit enables or disables the diagnostic A/D conversion that uses the reference voltage signal ADDIAGOUT and is executed at the end of the A/D conversion of CG0. 0: Disable A/D conversion that uses the ADDIAGOUT signal. 1: Convert the ADDIAGOUT signal. This bit can only be specified for ADCAnCG0. Clear this bit for the ADCAnCG1 and ADCAnCG2 registers.
23 to 00	ADCAnCGiS [23:00]	These bits specify the analog input signals to be converted for CGi. 0: Do not convert the analog input ADCAnIm. 1: Convert the analog input ADCAnIm. **Note**Clear bits corresponding to channels that are not provided by this microcontroller.

(4) ADCAnIOCi - A/D converter interrupt control register i

The A/D conversion end interrupt INTADCAnTi can be generated when the A/D conversion of a certain channel has been completed.

This register specifies the channels for which the interrupt INTADCAnTi is generated on A/D conversion completion.

If ADCAnIOCi is cleared to 0000 0000_H, the interrupt INTADCAnTi is automatically generated on the completion of A/D conversion of CGi.

Access This register can be read or written in 32-bit units. It can be written at any time, even when the A/D converter is enabled (by setting ADCAnCTL0.ADCAnCE to 1). The new value takes effect after the current A/D conversion of CGi has been completed.

Address <ADCAn_base_USER> + $0C_H$ + i × 4_H

Initial value 0000 0000_H. This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
ADCAnCG0IDG	0	0	0	0	0	0	0	ADCAnCGil[23:16]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCAnCGil[15:00]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25-12 ADCAnIOCi Register Contents

Bit Position	Bit Name	Description
31	ADCAn CG0IDG	This bit specifies whether the interrupt INTADCAnTi is generated on completion of the A/D conversion that uses the reference voltage when the diagnostic mode is enabled for CG0 (by setting ADCAnCG0.ADCAnDIAG to 1). 0: Do not generate the A/D conversion end interrupt INTADCAnTi. 1: Generate the A/D conversion end interrupt INTADCAnTi. This bit can only be specified for ADCAnIOC0. Clear this bit for the ADCAnIOC1 and ADCAnIOC2 registers. For details, see *(1) "A/D conversion circuit diagnosis" on page 1714*.
23 to 00	ADCAn CGil[23:00]	These bits specify whether the A/D conversion end interrupt INTADCAnTi is generated on A/D conversion completion of channel m. 0: Do not generate the A/D conversion end interrupt INTADCAnTi. 1: Generate the A/D conversion end interrupt INTADCAnTi. **Note**Clear bits corresponding to channels that are not provided by this microcontroller.

Note Because the ADCAnIOCi register is associated with the ADCAnCGi register, their buffer registers must be updated simultaneously. Because the update is performed when the ADCAnCGi register is written to, always write to the ADCAnIOCi register before the ADCAnCGi register when changing the interrupt generation for a CG.

(5) ADCAnCNT – A/D converter stabilization counter

This register specifies the stabilization time.

Access This register can be read or written in 8-bit units.

Address <ADCAn_base_OS> + 114_H

Initial value 00_H. This register is initialized by any reset.

7	6	5	4	3	2	1	0
ADCAnCNT[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25-13 ADCAnCNT Register Contents

Bit Position	Bit Name	Description
7 to 0	ADCAn CNT[7:0]	These bits specify the stabilization counter value. Stabilization time = ADCAnCNT[7:0] × clock cycles (PCLK)

(6) ADCAnTSELi - A/D converter trigger select control register 0 (product dependent)

This register specifies the input signals to be used in combination with the hardware start trigger signal ADCAnTTRGi. Multiple trigger sources can be used simultaneously.

Access This register can be read or written in 16-bit units. It can only be written when the A/D converter is disabled (by clearing ADCAnCTL0.ADCAnCE).

Address <ADCAn_base_OS> + 108_H + i × 4_H

Initial value 0000_H. This register is initialized by any reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCAnTiSEL[15:00]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25-14 ADCAnTSELi Register Contents

Bit Position	Bit Name	Description
15 to 0	ADCAn TiSEL[15:00]	These bits specify whether the corresponding input signal is to be used as a hardware start trigger. 0: Do not use the signal as a hardware start trigger. 1: Use the signal as a hardware start trigger. **Note** Clear bits corresponding to triggers that are not provided by this microcontroller.

Note For details about the connection destinations of hardware start triggers, see *Table 25-2 "What the Hardware Trigger Is Connected to"*.

25.4.3 Conversion status registers

(1) ADCAnSTR1 – A/D converter overwrite error flag

This register indicates whether the latest A/D conversion result in the ADCAnCmCR register was overwritten before it was read.

Access This register can be read in 32-bit units.

Address <ADCAn_base_USER> + 28_H

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	ADCAnOWE[23:16]							
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCAnOWE[15:00]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25-15 ADCAnSTR1 Register Contents

Bit Position	Bit Name	Description
23 to 0	ADCAn OWE[23:00]	These bits indicate whether the A/D conversion result of channel m was overwritten before it was read. 0: Not overwritten 1: Overwritten These error flags are cleared by setting ADCAnSTC1.ADCAnOWECm. **Note**Clear bits corresponding to channels that are not provided by this microcontroller.

Note The value of ADCAnSTR1.ADCAnOWEm is applied to the following overwrite error flag:

- The error flag in the register that has the latest A/D conversion result of channel m (ADCAnCmCR.ADCAnCmER1)

(2) ADCAnSTC1 – ADCAnSTR1 flag clear register

This register clears the ADCAnSTR1 register.

Access This register can be written in 32-bit units.

When this register is read, 0000 0000_H is always returned.

Address <ADCAn_base_USER> + 34_H

Initial value 0000 0000_H. This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	ADCAnOWEC[23:16]							
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCAnOWEC[15:00]															
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 25-16 ADCAnSTC1 Register Contents

Bit Position	Bit Name	Description
23 to 0	ADCAn OWEC[23:00]	0: No function 1: Clear the corresponding ADCAnSTR1.ADCAnOWEm. **Note**Clear bits corresponding to channels that are not provided by this microcontroller.

(3) ADCAnSTR2 – A/D converter status flag 2

This register indicates the current conversion status.

Access This register can be read in 16-bit units.

Address <ADCAn_base_USER> + $2C_H$

Initial value 0000_H. This register is initialized by any reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	ADCAnRQ3	ADCAnRQ[2:0]			0	0	0	0	ADCAnST3	ADCAnST[2:0]		
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25-17 ADCAnSTR2 Register Contents

Bit Position	Bit Name	Description
11	ADCAnRQ3 (product dependent)	This bit indicates whether an ADCHALT request is pending: 0: No ADCHALT request is pending. 1: An ADCHALT request is pending.
10 to 8	ADCAnRQ[2:0]	These bits indicate whether an A/D conversion request for CGi is pending: 0: No A/D conversion request for CGi is pending. 1: An A/D conversion request for CGi is pending.
3	ADCAnST3 (product dependent)	This bit indicates whether A/D conversion is currently in the ADCHALT status due to a software trigger (ADCAnTRG3.ADCAnSTT3). 0: Conversion is not in the ADCHALT status. 1: Conversion is in the ADCHALT status. This bit is cleared when the A/D converter is disabled (by clearing ADCAnCTL0.ADCAnCE).
2 to 0	ADCAnST[2:0]	These bits indicate whether A/D conversion of CGi is currently being performed. 0: A/D conversion is not currently being performed (including when A/D conversion is interrupted due to A/D conversion of a higher priority CG). 1: A/D conversion is currently being performed. These bits are cleared when the A/D converter is disabled (by clearing ADCAnCTL0.ADCAnCE).

(4) ADCAnSTC2 – A/D converter status flag clear register 2

This register clears the overwrite and result check status flags of the ADCAnLCR and ADCAnDBiCR registers.

Access This register can be written in 8-bit units.

Address <ADCAn_base_USER> + 38_H

Initial value 00_H. This register is initialized by any reset.

7	6	5	4	3	2	1	0
ADCAn LERC1	ADCAn LERC0	ADCAn DB2ERC1	ADCAn DB2ERC0	ADCAn DB1ERC1	ADCAn DB1ERC0	ADCAn DB0ERC1	ADCAn DB0ERC0
W	W	W	W	W	W	W	W

Table 25-18 ADCAnSTC2 Register Contents

Bit Position	Bit Name	Description
7	ADCAn LERC1	This bit clears the overwrite flag ADCAnLCR.ADCAnLER1. 0: No function 1: Clear ADCAnLCR.ADCAnLER1.
6	ADCAn LERC0	This bit clears the result check error flag ADCAnLCR.ADCAnLER0. 0: No function 1: Clear ADCAnLCR.ADCAnLER0.
5, 3, 1	ADCAn DBiERC1	These bits clear the overwrite flag ADCAnDBiCR.ADCAnDBiER1. 0: No function 1: Clear ADCAnDBiCR.ADCAnDBiER1.
4, 2, 0	ADCAn DBiERC0	These bits clear the result check error flag ADCAnDBiCR.ADCAnDBiER0. 0: No function 1: Clear ADCAnDBiCR.ADCAnDBiER0.

25.4.4 Software trigger registers

(1) ADCAnTRGi – A/D converter software trigger register i

This register is the trigger register for starting the A/D conversion of CGi.

Access This register can be written in 8-bit units. When this register is read, 00_H is always returned.

Address <ADCAn_base_USER> + $A4_H + i \times 4_H$

Initial value 00_H. This register is initialized by any reset.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	ADCAn STTi
W	W	W	W	W	W	W	W

Table 25-19 ADCAnTRGi Register Contents

Bit Position	Bit Name	Description
0	ADCAnSTTi	This bit starts the A/D conversion of CGi. 0: No function 1: Start the A/D conversion of CGi.

For details, see *25.3.5 “Starting A/D conversion (start triggers)” on page 1699* .

(2) ADCAnTRG3 – A/D converter software trigger register 3 (product dependent)

This trigger register is used to transition to the ADCHALT mode.

Access This register can be written in 8-bit units. When this register is read, 00_H is always returned.

Address <ADCAn_base_USER> + $B0_H$

Initial value 00_H. This register is initialized by any reset.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	ADCAn STT3
W	W	W	W	W	W	W	W

Table 25-20 ADCAnTRG3 Register Contents

Bit Position	Bit Name	Description
0	ADCAnSTT3	0: No function 1: Transition to the ADCHALT mode.

For details, see *25.3.8 “Pausing and resuming A/D conversion (ADCHALT mode) (product dependent)” on page 1703* .

(3) ADCAnTRG4+i – A/D converter software trigger register 4+i

This software trigger register is used to stop the A/D conversion of CGi.

Access This register can be written in 8-bit units. When this register is read, 00_H is always returned.

Address <ADCAn_base_USER> + $B4_H + i \times 4_H$

Initial value 00_H. This register is initialized by any reset.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	ADCAnSPi
W	W	W	W	W	W	W	W

Table 25-21 ADCAnTRG4+i Register Contents

Bit Position	Bit Name	Description
0	ADCAnSPi	0: No function 1: Stop the A/D conversion of CGi.

For details, see *25.3.6 “Stopping A/D conversion (stop triggers)” on page 1701* .

(4) ADCAnTRG7 – A/D converter software trigger register 7 (product dependent)

This software trigger register is used to exit the ADCHALT mode and resume A/D conversion.

Access This register can be written in 8-bit units. When this register is read, 00_H is always returned.

Address <ADCAn_base_USER> + $C0_H$

Initial value 00_H. This register is initialized by any reset.

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	ADCAn SP3
W	W	W	W	W	W	W	W

Table 25-22 ADCAnTRG7 Register Contents

Bit Position	Bit Name	Description
0	ADCAnSP3	0: No function 1: Resume A/D conversion.

For details, see *25.3.8 “Pausing and resuming A/D conversion (ADCHALT mode) (product dependent)” on page 1703* .

25.4.5 A/D conversion result registers

(1) ADCAnLCR – A/D converter latest conversion result register

This register stores the result and status of the latest A/D conversion.

This register makes it possible to read the latest A/D conversion results.

Access This register can be read in 32-bit units.

- The upper 16 bits store the A/D conversion result status.
- The lower 16 bits store the A/D conversion result.

Address <ADCAn_base_USER> + $A0_H$

Initial value $0300\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	ADCAn LCG[1:0]		ADCAn LER1	ADCAn LER0	ADCAn LUR	ADCAnLCN[4:0]				
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCAnLCR[15:00]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25-23 ADCAnLCR Register Contents (1/2)

Bit Position	Bit Name	Description
25, 24	ADCAn LCG[1:0]	These bits indicate the CG to which the conversion result stored in ADCAnLCR[15:00] belongs.
23	ADCAn LER1	This bit indicates the overwrite error status. 0: Not overwritten 1: Overwritten This error flag is cleared by setting ADCAnSTC2.ADCAnLERC1.
22	ADCAn LER0	This bit indicates the status of the A/D conversion result limit comparison. 0: The conversion results are within the specified range. 1: The conversion results are not within the specified range. This error flag is cleared by setting ADCAnSTC2.ADCAnLERC0.
21	ADCAn LUR	This bit indicates the update status of the A/D conversion result. 0: The A/D conversion result has been read from the ADCAnLCR register. 1: The A/D conversion result is new and has not been read from the ADCAnLCR register. This bit is cleared by reading it.
20 to 16	ADCAn LCN[4:0]	These bits indicate the channel number to which the conversion result stored in the ADCAnLCR[15:00] bits belongs. 00001 × m = CHm

ADCAn LCG1	ADCAn LCG0	Channel Group
0	0	CG0
0	1	CG1
1	0	CG2
1	1	None

Table 25-23 ADCAnLCR Register Contents (2/2)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Description</th></tr>
<tr><td>15 to 0</td><td>ADCAn LCR [15:00]</td><td>These bits indicate the A/D conversion result.
The resolution and alignment depend on ADCAnCTL1.ADCAnCTYP and ADCAnCTL1.ADCAnCRAC as follows:
<table>
<tr><th>ADCAn CTL1. ADCAn CTYP</th><th>ADCAn CTL1. ADCAn CRAC</th><th>Resolution and Alignment</th><th>A/D Conversion Result Value Bit Position</th></tr>
<tr><td>0</td><td>0</td><td>12-bit resolution, right-aligned</td><td>[11:00] of ADCAnLCR[15:00]</td></tr>
<tr><td>0</td><td>1</td><td>12-bit resolution, left-aligned</td><td>[15:04] of ADCAnLCR[15:00]</td></tr>
<tr><td>1</td><td>0</td><td>10-bit resolution, right-aligned</td><td>[09:00] of ADCAnLCR[15:00]</td></tr>
<tr><td>1</td><td>1</td><td>10-bit resolution, left-aligned</td><td>[15:06] of ADCAnLCR[15:00]</td></tr>
</table></td></tr>
</table>

Note When A/D conversion is performed by using the internal reference voltage, the A/D conversion result is stored in the ADCAnDGCR register, not in the ADCAnLCR, ADCAnCmCR, and ADCAnDBiCR registers. (For details, see *(4) "ADCAnDBiCRL – DMA buffer register of CGi (product dependent)" on page 1751*.)

(2) ADCAnCmCR – A/D converter conversion result register for channel m

This register stores the result and status of the latest A/D conversion for channel m.

This makes it possible to read the A/D conversion results for a specified channel (m).

Access This register can be read in 32-bit units.

- The upper 16 bits store the A/D conversion result status.
- The lower 16 bits store the A/D conversion result.

Address <ADCAn_base_USER> + $3C_H$ + m × 4_H

Initial value $0300\ 0000_H$ + m × $0001\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	ADCAn CmCG[1:0]		ADCAn CmER1	ADCAn CmER0	ADCAn CmUR	ADCAnCmCN[4:0]				
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCAnCmCR[15:00]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Notes

1. The functions of each bit are the same as those of the ADCAnLCR register, except that here they affect the latest A/D conversion result for a specific channel instead of the latest A/D conversion results for all channels. (For details, see *Table 25-23 "ADCAnLCR Register Contents" on page 1745*.)
2. After a reset, the ADCAnCmCG[1:0] bits are set to 11_B.
3. If ADCAnCTL1.ADCAnRCL is cleared, the A/D conversion result in the ADCAnCmCR[15:00] bits is kept until it is overwritten by the next A/D conversion result.

   If ADCAnCTL1.ADCAnRCL is set, the ADCAnCmCR[15:00] bits are cleared by reading them.

Table 25-24 ADCAnCmCR Register Contents (1/2)

Bit Position	Bit Name	Description
25, 24	ADCAn CmCG [1:0]	These bits indicate the CG to which the conversion result stored in ADCAnCmCR[15:00] belongs.

ADCAn CmCG1	ADCAn CmCG0	Channel Group
0	0	CG0
0	1	CG1
1	0	CG2
1	1	None

Table 25-24 ADCAnCmCR Register Contents (2/2)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Description</th></tr>
<tr><td>23</td><td>ADCAn CmER1</td><td>This bit indicates the overwrite error status.
0: Not overwritten
1: Overwritten
This error flag reflects the value of ADCAnSTR1.ADCAnOWEm and is cleared by setting ADCAnSTC1.ADCAnQWECm.</td></tr>
<tr><td>22</td><td>ADCAn CmER0</td><td>This bit indicates the status of the A/D conversion result limit comparison.
0: The conversion results are within the specified range.
1: The conversion results are not within the specified range.
This error flag reflects the value of ADCAnSTR0.ADCAnRCEm and is cleared by setting ADCAnSTC0.ADCAnRCECm.</td></tr>
<tr><td>21</td><td>ADCAn CmUR</td><td>This bit indicates the update status of the A/D conversion result.
0: The A/D conversion result has been read from the ADCAnCmCR register.
1: The A/D conversion result is new and has not been read from the ADCAnCmCR register.
This bit is cleared by reading it.</td></tr>
<tr><td>20 to 16</td><td>ADCAn CmCN [4:0]</td><td>These bits indicate the channel number to which the conversion result stored in the ADCAnCmCR[15:00] bits belongs.
00001 × m = CHm</td></tr>
<tr><td>15 to 0</td><td>ADCAn CmCR [15:00]</td><td>These bits indicate the A/D conversion result.
The resolution and alignment depend on ADCAnCTL1.ADCAnCTYP and ADCAnCTL1.ADCAnCRAC as follows:
<table>
<tr><th>ADCAn CTL1. ADCAn CTYP</th><th>ADCAn CTL1. ADCAn CRAC</th><th>Resolution and Alignment</th><th>A/D Conversion Result Value Bit Position</th></tr>
<tr><td>0</td><td>0</td><td>12-bit resolution, right-aligned</td><td>[11:00] of ADCAnCmCR[15:00]</td></tr>
<tr><td>0</td><td>1</td><td>12-bit resolution, left-aligned</td><td>[15:04] of ADCAnCmCR[15:00]</td></tr>
<tr><td>1</td><td>0</td><td>10-bit resolution, right-aligned</td><td>[09:00] of ADCAnCmCR[15:00]</td></tr>
<tr><td>1</td><td>1</td><td>10-bit resolution, left-aligned</td><td>[15:06] of ADCAnCmCR[15:00]</td></tr>
</table>
</td></tr>
</table>

Note When A/D conversion is performed by using the internal reference voltage, the A/D conversion result is stored in the ADCAnDGCR register, not in the ADCAnLCR, ADCAnCmCR, and ADCAnDBiCR registers. (For details, see *(4) "ADCAnDBiCRL – DMA buffer register of CGi (product dependent)" on page 1751*.)

(3) ADCAnDBiCR – DMA buffer register of CGi

This register stores the result and status of the latest A/D conversion of CGi. This makes it possible to read the A/D conversion results for all channels of CGi.

Access This register can be read in 32-bit units.

- The upper 16 bits store the A/D conversion result status.
- The lower 16 bits store the A/D conversion result.

Address <ADCAn_base_USER> + $C4_H$ + i × 4_H

Initial value 0000 0000_H + i × 0100 0000_H. This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	ADCAn DBiCG[1:0]		ADCAn DBiER1	ADCAn DBiER0	ADCAn DBiUR	ADCAnDBiCN[4:0]				
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCAnDBiCR[15:00]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Note The functions of each bit are the same as those of the ADCAnLCR register, except that here they affect the latest A/D conversion result for CGi instead of the latest A/D conversion results for all CGs. (For details, see *Table 25-23 "ADCAnLCR Register Contents" on page 1745*.)

Table 25-25 ADCAnDBiCR Register Contents (1/2)

Bit Position	Bit Name	Description
25, 24	ADCAn DBiCG [1:0]	These bits indicate the CG to which the conversion result stored in ADCAnDBiCR[15:00] belongs. ADCAn DBiCG1 / ADCAn DBiCG0 / Channel Group 0 / 0 / CG0 0 / 1 / CG1 1 / 0 / CG2 1 / 1 / None The values of these bits are fixed because the conversion results and status of the same CG are always saved.
23	ADCAn DBiER1	This bit indicates the overwrite error status. 0: Not overwritten 1: Overwritten This error flag is cleared by setting ADCAnSTC2.ADCAnDBiERC1.
22	ADCAn DBiER0	This bit indicates the status of the A/D conversion result limit comparison. 0: The conversion results are within the specified range. 1: The conversion results are not within the specified range. This error flag is cleared by setting ADCAnSTC2.ADCAnDBiERC0.
21	ADCAn DBiUR	This bit indicates the update status of the A/D conversion result. 0: The A/D conversion result has been read from the ADCAnDBiCR register. 1: The A/D conversion result is new and has not been read from the ADCAnDBiCR register. This bit is cleared by reading it.

Table 25-25 ADCAnDBiCR Register Contents (2/2)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Description</th></tr>
<tr><td>20 to 16</td><td>ADCAn DBiCN [4:0]</td><td>These bits indicate the channel number to which the conversion result stored in the ADCAnDBiCR[15:00] bits belongs.
00001 × m = CHm</td></tr>
<tr><td>15 to 0</td><td>ADCAn DBiCR [15:00]</td><td>These bits indicate the A/D conversion result.
The resolution and alignment depend on ADCAnCTL1.ADCAnCTYP and ADCAnCTL1.ADCAnCRAC as follows:
<table>
<tr><th>ADCAn CTL1. ADCAn CTYP</th><th>ADCAn CTL1. ADCAn CRAC</th><th>Resolution and Alignment</th><th>A/D Conversion Result Value Bit Position</th></tr>
<tr><td>0</td><td>0</td><td>12-bit resolution, right-aligned</td><td>[11:00] of ADCAnDBiCR[15:00]</td></tr>
<tr><td>0</td><td>1</td><td>12-bit resolution, left-aligned</td><td>[15:04] of ADCAnDBiCR[15:00]</td></tr>
<tr><td>1</td><td>0</td><td>10-bit resolution, right-aligned</td><td>[09:00] of ADCAnDBiCR[15:00]</td></tr>
<tr><td>1</td><td>1</td><td>10-bit resolution, left-aligned</td><td>[15:06] of ADCAnDBiCR[15:00]</td></tr>
</table>
</td></tr>
</table>

Note When A/D conversion is performed by using the internal reference voltage, the A/D conversion result is stored in the ADCAnDGCR register, not in the ADCAnLCR, ADCAnCmCR, ADCAnDBiCR, and ADCAnDBiCRL registers. (For details, see *(5) "ADCAnDGCR – Diagnostic conversion result register" on page 1752*.)

(4) ADCAnDBiCRL – DMA buffer register of CGi (product dependent)

This register stores the result of the latest A/D conversion of CGi. This makes it possible to read the A/D conversion results for all channels of CGi. This register reflects the lower 16 bits of the DMA buffer register of CGi (ADCAnDBiCR).

Access This register can be read in 16-bit units.

Address <ADCAn_base_USER> + $D0_H$ + i×4_H

Initial value 0000_H. This register is initialized by any reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCAnDBiCR[15:00]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25-26 ADCAnDBiCRL Register Contents

Bit Position	Bit Name	Description
15 to 0	ADCAn DBiCRL [15:00]	These bits indicate the A/D conversion result. The resolution and alignment depend on ADCAnCTL1.ADCAnCTYP and ADCAnCTL1.ADCAnCRAC as follows:

ADCAn CTL1. ADCAn CTYP	ADCAn CTL1. ADCAn CRAC	Resolution and Alignment	A/D Conversion Result Value Bit Position
0	0	12-bit resolution, right-aligned	[11:00] of ADCAnDBiCR[15:00]
0	1	12-bit resolution, left-aligned	[15:04] of ADCAnDBiCR[15:00]
1	0	10-bit resolution, right-aligned	[09:00] of ADCAnDBiCR[15:00]
1	1	10-bit resolution, left-aligned	[15:06] of ADCAnDBiCR[15:00]

Note When A/D conversion is performed by using the internal reference voltage, the A/D conversion result is stored in the ADCAnDGCR register, not in the ADCAnLCR, ADCAnCmCR, ADCAnDBiCR, and ADCAnDBiCRL registers. (For details, see *(5) "ADCAnDGCR – Diagnostic conversion result register" on page 1752*.)

(5) ADCAnDGCR – Diagnostic conversion result register

This register stores the result of the A/D conversion that uses the reference voltage signal ADDIAGOUT (when ADCAnCG0.ADCAnDIAG is set to 1).

Diagnostic conversion starts after the A/D conversion of the last channel of CG0 has been completed.

Access This register can be read in 16-bit units.

Address <ADCAn_base_USER> + $9C_H$

Initial value 0000_H. This register is initialized by any reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCAnDGCR[15:00]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25-27 ADCAnDGCR Register Contents

Bit Position	Bit Name	Description
15 to 0	ADCAn DGCR[15:00]	These bits indicate the result of the diagnostic A/D conversion. The resolution and alignment depend on ADCAnCTL1.ADCAnCTYP and ADCAnCTL1.ADCAnCRAC (as for the normal A/D conversion result registers).

25.4.6 A/D conversion result limit comparison registers

(1) ADCAnCTL2 – A/D converter result check register

This register makes it possible to enable or disable the conversion result limit comparison function for individual channels.

For details, see *25.3.12 “Result check functions” on page 1711* .

Access This register can be read or written in 32-bit units. It can only be written when the A/D converter is disabled (by clearing ADCAnCTL0.ADCAnCE).

Address <ADCAn_base_USER> + 18_H

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	ADCAnRCK[23:16]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCAnRCK[15:00]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25-28 ADCAnCTL2 Register Contents

Bit Position	Bit Name	Description
23 to 00	ADCAn RCK[23:00]	These bits enable or disable result limit comparison for CHm. 0: Do not check whether the A/D conversion result of CHm is within the specified range. 1: Check whether the A/D conversion result of CHm is within the specified range. **Note**Clear bits corresponding to channels that are not provided by this microcontroller.

Note This setting is valid for A/D conversions of every CG.

(2) ADCAnUL – A/D converter result limit comparison (upper limit)

This register specifies the upper limit of the A/D conversion result.

For details, see *25.3.12 "Result check functions" on page 1711* .

Access This register can be read or written in 16-bit units. It can only be written when the A/D converter is disabled (by clearing ADCAnCTL0.ADCAnCE).

Address <ADCAn_base_USER> + $1C_H$

Initial value 0000_H. This register is initialized by any reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCAnUL[11:00]												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

Table 25-29 ADCAnUL Register Contents

Bit Position	Bit Name	Description
15 to 4	ADCAn UL[11:00]	These bits specify the upper limit of the A/D conversion result. If using the 10-bit function, use ADCAnUL[11:02] for this specification.

(3) ADCAnLL – A/D converter result limit comparison (lower limit)

This register specifies the lower limit of the A/D conversion result.

For details, see *25.3.12 "Result check functions" on page 1711* .

Access This register can be read or written in 16-bit units. It can only be written when the A/D converter is disabled (by clearing ADCAnCTL0.ADCAnCE).

Address <ADCAn_base_USER> + 20_H

Initial value 0000_H. This register is initialized by any reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCAnLL[11:00]												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

Table 25-30 ADCAnLL Register Contents

Bit Position	Bit Name	Description
15 to 4	ADCAn LL[11:00]	These bits specify the lower limit of the A/D conversion result. If using the 10-bit function, use ADCAnLL[11:02] for this specification.

(4) ADCAnSTR0 - A/D converter result limit comparison error flag

This register indicates the error status of the latest A/D conversion result limit comparison for the channels specified by the ADCAnCTL2 register. This

register makes it possible to check which A/D conversion results are outside the specified range.

For details, see *25.3.12 “Result check functions” on page 1711* .

Access This register can be read in 32-bit units.

Address <ADCAn_base_USER> + 24_H

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	ADCAnRCE[23:16]							
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCAnRCE[15:00]															
R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Table 25-31 ADCAnSTR0 Register Contents

Bit Position	Bit Name	Description
23 to 0	ADCAn RCE[23:00]	These bits indicate whether the A/D conversion results are within the specified value range. 0: The conversion results are within the specified range. 1: At least one conversion result is out of the specified range. This error flag is cleared by setting ADCAnSTC0.ADCAnRCECm. **Note**Clear bits corresponding to channels that are not provided by this microcontroller.

Note The value of ADCAnSTR0.ADCAnRCEm is applied to the following A/D conversion result error flag:

- The error flag in the register that has the latest A/D conversion result of channel m (ADCAnCmCR.ADCAnCmER0)

(5) ADCAnSTC0 - ADCAnSTR0 flag clear register

This register clears ADCAnSTR0.

Access This register can be written in 32-bit units. When this register is read, 0000 0000_H is always returned.

Address <ADCAn_base_USER> + 30_H

Initial value 0000 0000_H. This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	ADCAnRCEC[23:16]							
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCAnRCEC[15:00]															
W	W	W	W	W	W	W	W	W	W	W	W	W	W	W	W

Table 25-32 ADCAnSTC0 Register Contents

Bit Position	Bit Name	Description
23 to 0	ADCAn RCEC[23:00]	0: No function 1: Clear the corresponding ADCAnSTR0.ADCAnRCEm. **Note**Clear bits corresponding to channels that are not provided by this microcontroller.

25.4.7 Diagnostic function control registers (product dependent)

(1) ADCAnDGCTL0 – Self-diagnosis function control register 0

This register specifies the reference voltages to apply in order to diagnose the A/D conversion circuit.

This register can be written to even when ADCAnCTL0.ADCAnCE is set to 1.

Access This register can be read or written in 16-bit units.

Address <ADCAn_base_USER> + DC_H

Initial value 0000_H. This register is initialized by any reset.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	ADCAnPSEL[2:0]		
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

Table 25-33 ADCAnDGCTL0 Register Contents

Bit Position	Bit Name	Description
2 to 0	ADCAn PSEL[2:0]	These bits specify the reference voltages.

ADCAnPSEL2	ADCAnPSEL1	ADCAnPSEL0	ADDIAGOUT Signal	DIAGOUT2 Signal	DIAGOUT1 Signal	DIAGOUT0 Signal
0	0	0	AV_{SS}	2/3 AV_{DD}	1/2 AV_{DD}	1/3 AV_{DD}
0	0	1	1/3 AV_{DD}	1/2 AV_{DD}	1/3 AV_{DD}	2/3 AV_{DD}
0	1	0	1/2 AV_{DD}	1/3 AV_{DD}	2/3 AV_{DD}	1/2 AV_{DD}
0	1	1	2/3 AV_{DD}	Hi-Z	Hi-Z	Hi-Z
1	0	0	AV_{DD}	2/3 AV_{DD}	1/2 AV_{DD}	1/3 AV_{DD}
1	0	1	AV_{DD}	1/2 AV_{DD}	1/3 AV_{DD}	2/3 AV_{DD}
1	1	0	AV_{DD}	1/3 AV_{DD}	2/3 AV_{DD}	1/2 AV_{DD}
1	1	1	AV_{DD}	Hi-Z	Hi-Z	Hi-Z

When Hi-Z is selected and conversion is performed, the A/D conversion result is undefined.

For details, see *(1) "A/D conversion circuit diagnosis" on page 1714*.

(2) ADCAnDGCTL1 – Self-diagnosis function control register 1

This register specifies the channels to which an internal reference voltage is applied (instead of the analog input signal ADCAnIm).

This register can only be written to when ADCAnCTL0.ADCAnCE is cleared to 0.

Access This register can be read or written in 32-bit units.

Address <ADCAn_base_OS> + $11C_H$

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	ADCAnCDG[23:16]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCAnCDG[15:00]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25-34 ADCAnDGCTL1 Register Contents

Bit Position	Bit Name	Description
23 to 0	ADCAn CDG[23:00]	These bits specify which input voltage to use. 0: Use the analog input voltage pin ADCAnIm. 1: Use the following reference voltage: DIAGOUT0 when m = 21, 18, 15, 12, 9, 6, 3, 0 DIAGOUT1 when m = 22, 19, 16, 13, 10, 7, 4, 1 DIAGOUT2 when m = 23, 20, 17, 14, 11, 8, 5, 2 **Note** Clear bits corresponding to channels that are not provided by this microcontroller.

(3) ADCAnPDCTL0 – Pull-down resistor control register 0

This register specifies the channels to which the ADCAnIm pin-internal pull-down resistor is connected. For details, see *(3) "Open pin diagnosis" on page 1717*.

This register can only be written to when ADCAnCTL0.ADCAnCE is cleared to 0.

Access This register can be read or written in 32-bit units.

Address <ADCAn_base_OS> + 120_H

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0	ADCAnPDNA[23:16]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADCAnPDNA[15:00]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 25-35 ADCAnPDCTL0 Register Contents

Bit Position	Bit Name	Description
23 to 0	ADCAn PDNA[23:00]	These bits specify whether the internal pull-down register is connected to CHm. 0: Do not connect the internal pull-down resistor. 1: Connect the internal pull-down resistor. **Note**Clear bits corresponding to channels that are not provided by this microcontroller.

25.4.8 Channel sample & hold function setup register (product dependent)

(1) ADCAnSHCTL – A/D converter channel sample & hold control register

This register controls whether power is supplied to the channel sample & hold circuit, as well as whether the circuit is enabled or disabled.

It can only be written when the A/D converter is disabled (by clearing ADCAnCTL0.ADCAnCE).

However, when performing channel sample & hold circuit diagnosis, the register can be written even when ADCAnCTL0.ADCAnCE is set to 1.

Access This register can be read or written in 32-bit units.

Address <ADCAn_base_OS> + 118_H

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0	0	0	ADCAnCPS[12:00]												
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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	ADCAnCSEL[12:00]												
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

Table 25-36 ADCAnSHCTL Register Contents

Bit Position	Bit Name	Description
28 to 16	ADCAn CPS[12:00]	These bits control whether power is supplied to the channel sample & hold circuit. 0: Turn off the channel sample & hold circuit. 1: Turn on the channel sample & hold circuit. To reduce the power consumed by the channel sample & hold circuit, turn the circuit off when not using it. **Note** Clear bits corresponding to channels that are not provided by this microcontroller.
12 to 00	ADCAn CSEL[12:00]	These bits enable or disable the channel sample & hold function. 0: Disable the channel sample & hold function. 1: Enable the channel sample & hold function. When performing channel sample & hold circuit diagnosis, ADCAnSHCTL.ADCAnCSELx can be changed even if ADCAnCE = 1. **Note** Clear bits corresponding to channels that are not provided by this microcontroller.

25.5 Precautions on Usage

25.5.1 Channel input voltage range

Caution Use input voltages to ADCAnIm that are within the range of the ratings. If a voltage higher than AV_{DD} or lower than AV_{SS} is input to a channel, the conversion value of the channel is saturated, and the electrical characteristics of the other channels may also be affected.

25.5.2 Stopping conversion operation

If “0” is written in ADCAnCTL0.ADCAnCE during conversion, the conversion stops and no conversion result is stored into ADCAnCmCR.

25.5.3 Restrictions when using the channel S/H function

For details, see *(2) "Restrictions when using the channel sample & hold function" on page 1727*.

25.5.4 Notes on application design

(1) Analog input pin (ADCAnIm)

1. Be sure to input a voltage within the rated range to the ADCAnIm pin. It is recommended to clamp this pin using a diode with a V_F of 0.3 V (target) or less to prevent a voltage higher than AV_{REFPn} or lower than AV_{REFMn} from being applied to the pin. If a voltage higher than AV_{REFPn} or a voltage lower than AV_{REFMn} is input to ADCAnIm, the conversion value of that channel becomes undefined and is not guaranteed. In addition, the conversion value of other channels may be also affected.
2. To eliminate noise, connect a resistor Re between the ADCAnIm pin and the external analog signal input source, and a capacitor Ce between the ADCAnIm pin and AV_{SSn} pin.
3. Do not cross an analog signal line with a digital signal line or place an analog signal line in the vicinity of a digital signal line as this may degrade the A/D conversion characteristics due to noise induction.
4. It is recommended to avoid inputting and outputting a high driving current to or from a port close to the ADCAnIm pin and switching it by toggling.

(2) Wiring of power supply

To minimize the influence of switching noise of a digital circuit on the accuracy of the A/D converter, it is recommended to take the following measures.

1. Route the power line on a single side, or use as thick a pattern as possible and connect the power line in grid.

2. Insert a bypass capacitor close to the lead between a power supply pin (EV_{DD}, $OSCV_{DD}$, FV_{DD}, $PLLV_{DD}$ (product dependent), $CPUPLLV_{DD}$ (product dependent), $FRPLLV_{DD}$ (some products only, product dependent), CV_{DD} (product dependent), V_{DDM} (product dependent), V_{DDC} (product dependent), AV_{DDn}) and a ground pin (EV_{SS}, $OSCV_{SS}$, FV_{SS}, $PLLV_{SS}$ (product dependent), $CPUPLLV_{SS}$ (product dependent), $FRPLLV_{SS}$ (some products only, product dependent), CV_{SS} (product dependent), V_{SSM} (product dependent), V_{SSC} (product dependent), AV_{SSn}). A laminated ceramic capacitor of approx. 0.1 F (target) or a tantalum electrolytic capacitor of 4.7 F (target) or higher is recommended for use as this bypass capacitor.
3. It is recommended to separate the analog power line (AV_{DDn}) from the digital power lines (EV_{DD}, $OSCV_{DD}$, FV_{DD}, $PLLV_{DD}$ (product dependent), $CPUPLLV_{DD}$ (product dependent), $FRPLLV_{DD}$ (some products only, product dependent), CV_{DD} (product dependent), V_{DDM} (product dependent), V_{DDC} (product dependent)) and to supply analog power from a series regulator. To share the analog power line with the digital power lines, short-circuit the analog power supply and digital power supply at one point with an electrolytic capacitor at the power source, and separately wire the patterns on the board.

   In addition, insertion of a chip inductor at the inlet of the analog power is recommended. It is also recommended to ground the analog ground and digital ground at the ground source at one point with an electrolytic capacitor, and separate the wiring of the patterns on the board.

(3) Analog reference voltage input pins (AV_{REFPn}, AV_{REFMn})

Insert a bypass capacitor close to the pin lead between the AV_{REFPn} pin and AV_{REFMn} pin. A laminated ceramic capacitor of approx. 0.1μF (target) or a tantalum electrolytic capacitor of 4.7 μF (target) or higher is recommended for use as this bypass capacitor.

(4) Variation of A/D conversion result

The result of A/D conversion may vary due to fluctuations in the supply voltage and noise. Noise at an analog input pin (ADCAnIm) or reference voltage input pin (AV_{REFPn}, AV_{REFMn}) may cause an invalid conversion result.

Mitigate these variations by software processing to protect the system from adverse influences due to these variations and illegal conversion results.

The following are examples of software processing.

- Use the average value of the results of multiple A/D conversions as the A/D conversion result.
- Perform multiple consecutive A/D conversions and use the conversion results, with the exception of any abnormal conversion results that are obtained.
- If an A/D conversion result is obtained, from which it is judged that an abnormality has occurred in the system, do not perform abnormality processing at once but perform it upon reconfirming the occurrence of an abnormality.

(5) Hysteresis characteristics of A/D conversion

The successive approximation A/D converter holds the analog input voltage on a capacitor for the internal common sample & hold, and then executes A/D conversion. Even after A/D conversion has been completed, the analog input voltage remains on the capacitor for the internal common sample & hold. Consequently, the following phenomena may occur.

1. If the voltage rises above or falls below the voltage for the A/D conversion immediately before when A/D conversion is to be executed with a single channel, hysteresis characteristics, in which the conversion result is affected by the value immediately before, may emerge, and the conversion results may differ even at the same voltage. The hysteresis characteristics tend to increase if the signal source impedance or resistor Re of the external circuit of the analog input pin is high, or if the value of capacitor Ce is small.
2. When the analog input channel is changed, hysteresis characteristics, in which the conversion result is affected by the value of the channel immediately before, may emerge, and the conversion results may differ even at the same voltage, because A/D conversion is performed by using one A/D converter.

To obtain a more accurate conversion result, therefore, perform A/D conversion of the same channel two times in a row, and discard the first A/D conversion result.

25.6 How to Read A/D Converter Characteristics Table

This section describes the terms related to the A/D converter.

(1) Resolution

The minimum analog input voltage that can be identified, i.e. the ratio of the analog input voltage to 1 digital output is called 1 LSB (Least Significant Bit). The ratio of 1 LSB to the full scale is expressed as %FSR (full-scale range). %FSR is the ratio, in percentage, of the range in which an analog input voltage can be converted, and is expressed as follows regardless of the resolution.

$$1\%FSR = (\text{Maximum value of convertible analog input voltage} - \text{Minimum value of convertible analog input voltage})/100$$
$$= (AV_{REFP} - AV_{REFM})/100$$

1 LSB is as follows at a resolution of 10 bits:

$$1\ LSB = 1/2^{10}$$
$$= 1/1024$$
$$= 0.098\ \%FSR$$

1 LSB is as follows at a resolution of 12 bits:

$$1\ LSB = 1/2^{12}$$
$$= 1/4096$$
$$= 0.024\ \%FSR$$

The accuracy is determined by the total error, regardless of the resolution.

(2) Total error

This is the maximum value of the difference between the actually measured value and the theoretical value.

It is the total of the zero-scale error, full-scale error, linearity error, and a combination of these errors.

The total error shown in the characteristics table does not include the quantization error.

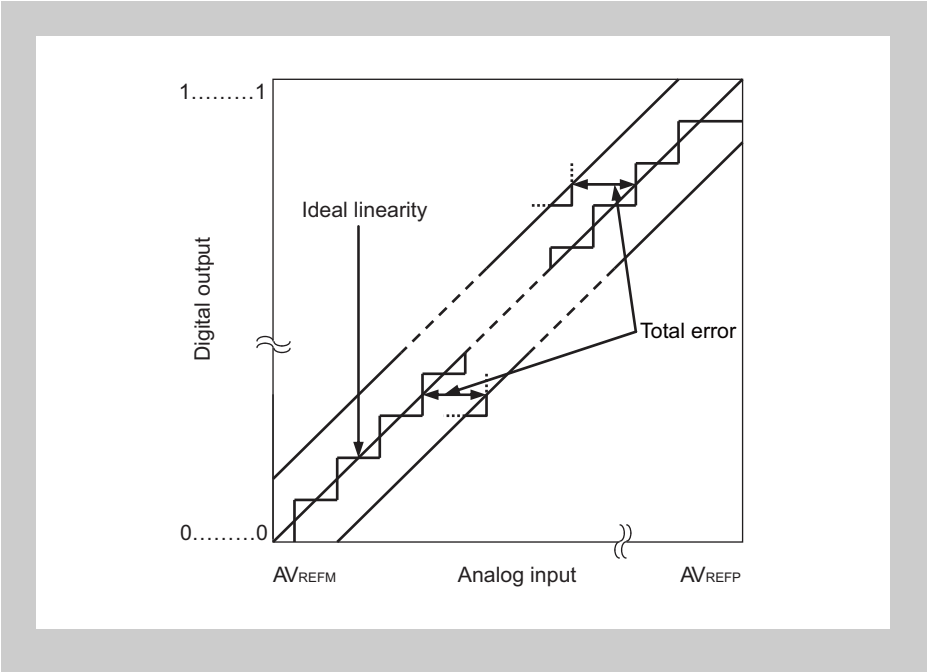


Figure 25-22 Total Error

(3) Quantization error

This is the error of ±1/2 LSB that always occurs when an analog value is converted into a digital value. Because the A/D converter converts an analog input voltage in a range of ±1/2 LSB into the same digital code, the quantization error is unavoidable.

Note that this error is not included in the total error, zero-scale error, full-scale error, integral linearity error, and differential linearity error in the characteristics table.

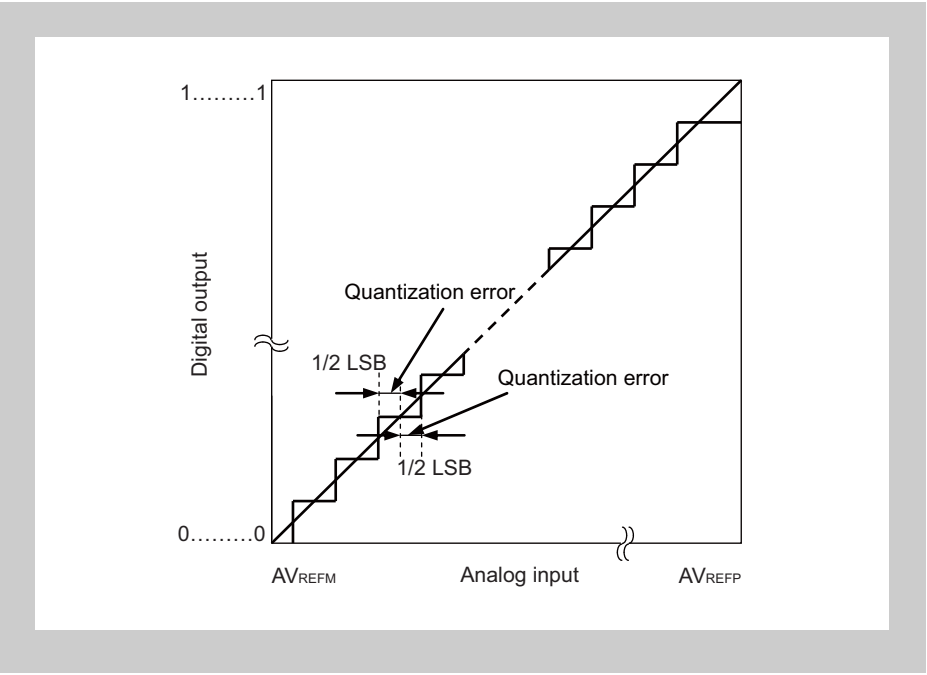


Figure 25-23 Quantization Error

(4) Zero-scale error

This is the difference between the actually measured value of the analog input voltage and the theoretical value (1/2 LSB) when the digital output changes from 0…000 to 0…001.

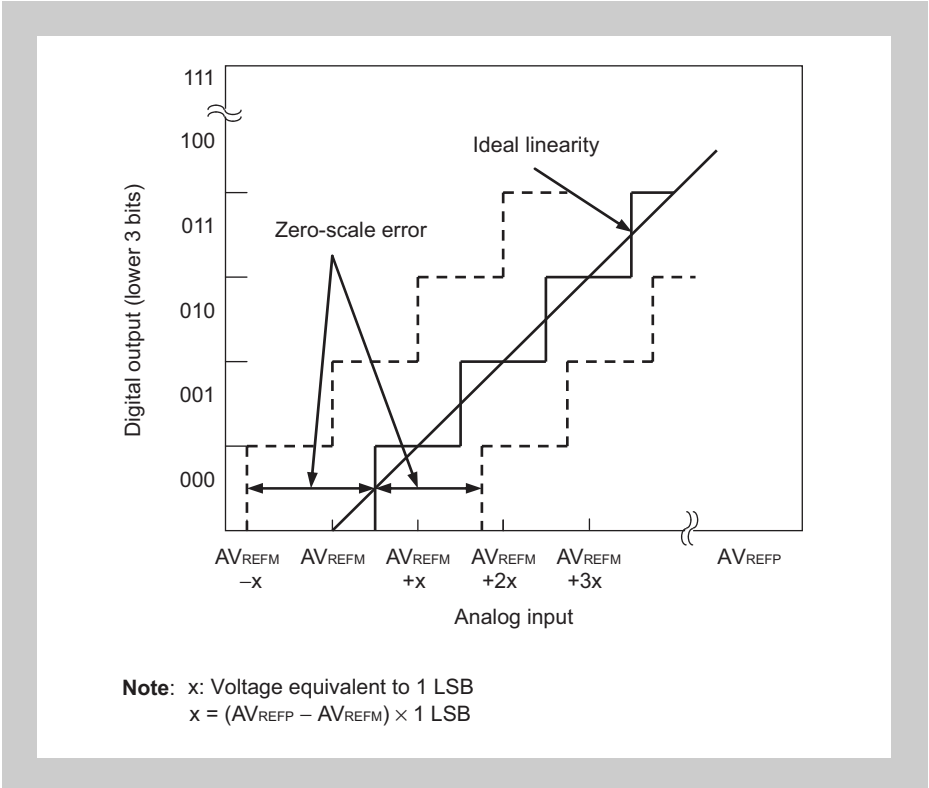


Figure 25-24 Zero-Scale Error

(5) Full-scale error

This is the difference between the actually measured value of the analog input voltage and the theoretical value (full scale – 3/2 LSB) when the digital output changes from 1…110 to 1…111.

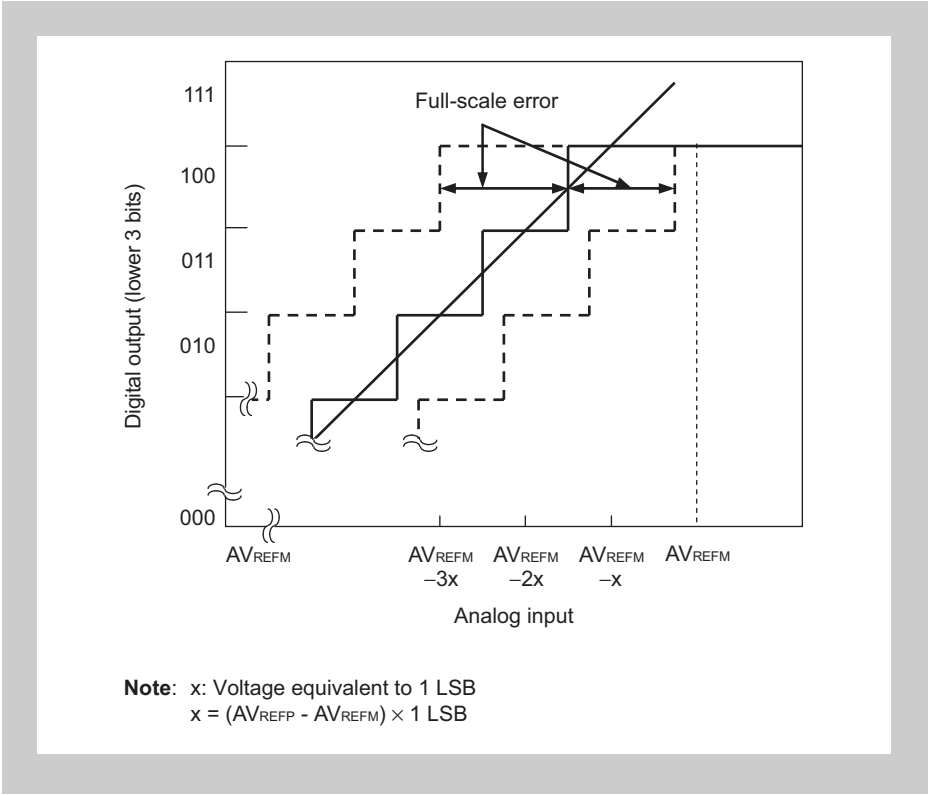


Figure 25-25 Full-Scale Error

(6) Differential linearity error

Ideally, the width at which a specific code is output is 1 LSB. The differential linearity error is the difference between the actually measured value of the width at which a specific code is output and the ideal value.

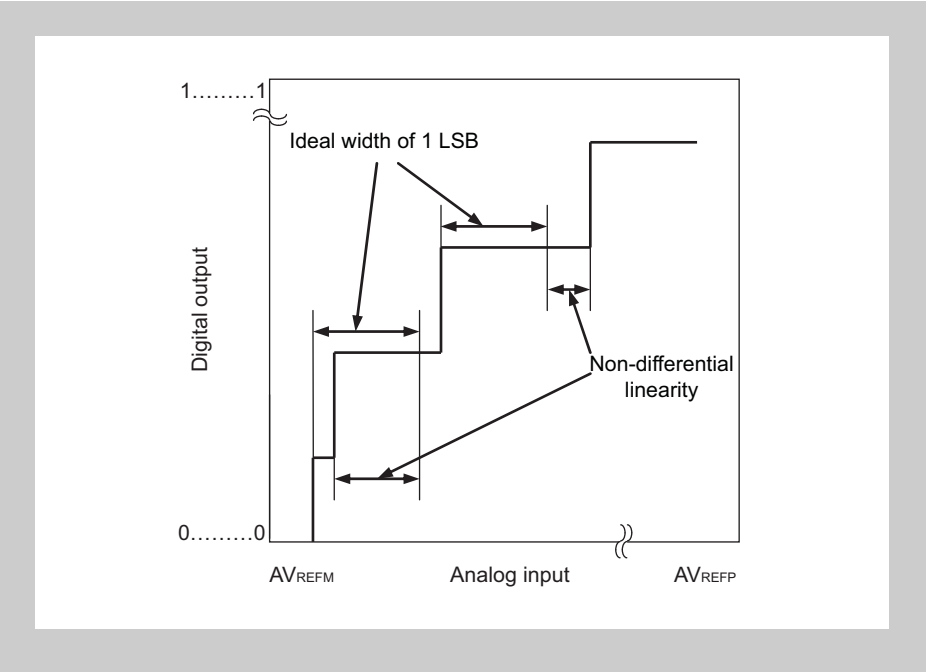


Figure 25-26 Differential Linearity Error

(7) Integral linearity error

This error indicates the extent to which the conversion characteristics differ from the ideal linear relationship. It indicates the maximum value of the difference between the actually measured value and its theoretical value where the zero-scale error and full-scale error are 0.

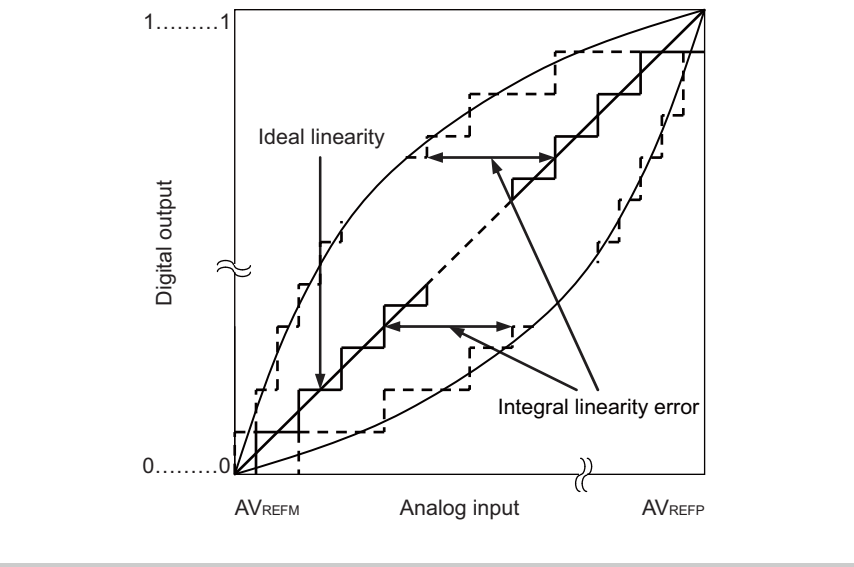


Figure 25-27 Integral Linearity Error

(8) Conversion time

This is the time from when an analog voltage is input until digital output is produced.

The conversion time in the characteristics table includes sampling time.

(9) Sampling time

This is the time during which the analog switch is on to input the analog voltage to the sample & hold circuit.

(10) A/D start time

This is the time from the A/D conversion trigger to the start of A/D conversion.

Chapter 26 H-Bus-Shared Memory

26.1 Features

- 64 KB
- Bank configuration (32 KB x 2)
- The memory can be accessed from both the internal system bus and H bus.
- The memory can be read or written in 8, 16, or 32-bit units.

26.2 Configuration

The H-bus-shared memory is RAM that can be accessed from both the internal system bus and H bus. This memory can be read or written in 8, 16, or 32-bit units. The memory uses a bank configuration (32 KB x 2), and the bus master of the internal system bus and bus master of the H bus can access the different banks at the same time. Note that arbitration is performed if there is a conflict when trying to access the same bank.

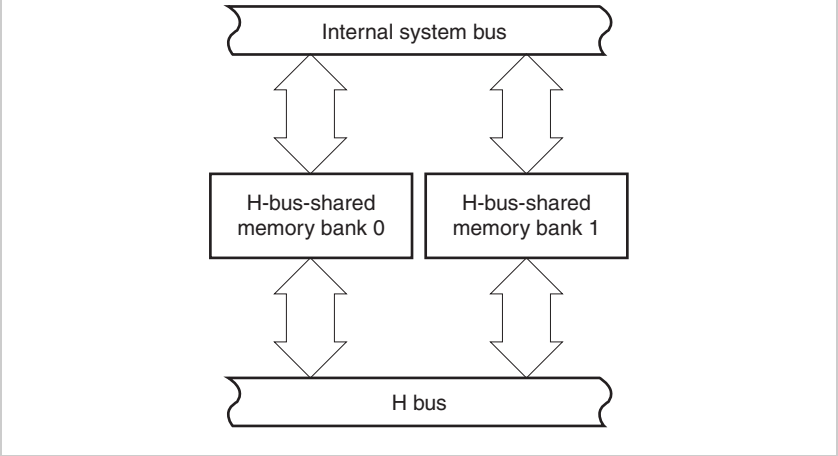


Figure 26-1 H-Bus-Shared Memory

The address spaces for each bank are shown below.

Table 26-1 H-Bus-Shared Memory

Bank	Size	Address
Bank 0	32 KB	$F9800000_H$ to $F9807FFF_H$
Bank 1	32 KB	$F9808000_H$ to $F980FFFF_H$

26.3 Operation

26.3.1 Parallel operation

The H-bus-shared memory is RAM that can be accessed from both the internal system bus and H bus. The memory uses a bank configuration (32 KB x 2), and the bus master of the internal system bus and bus master of the H bus can access the different banks at the same time.

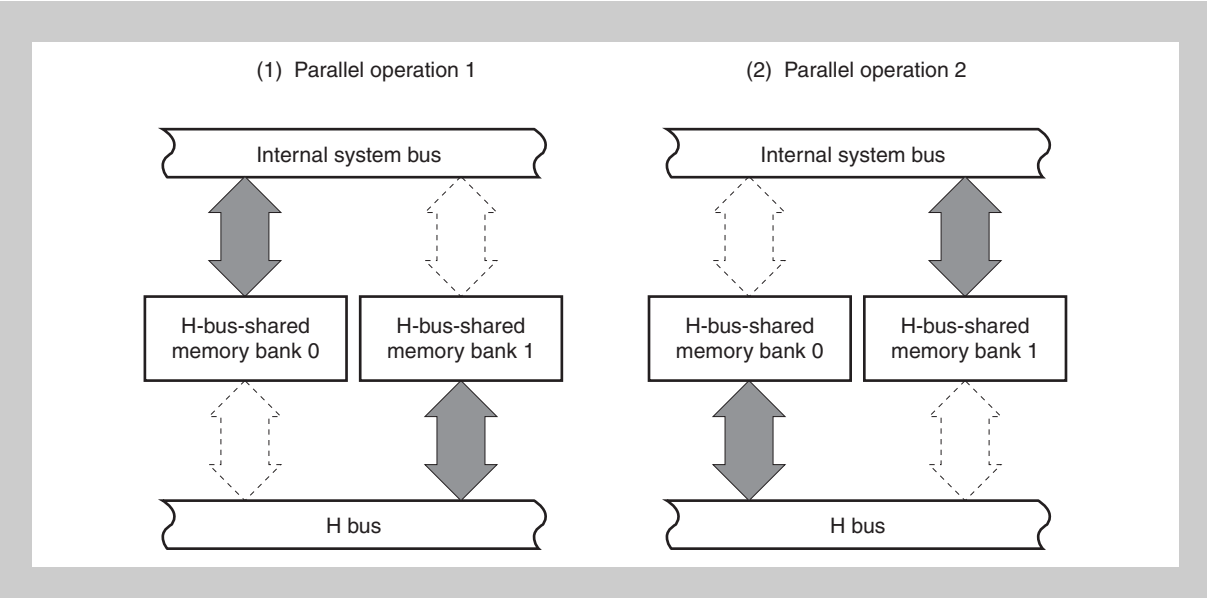


Figure 26-2 Parallel Operation Example

26.3.2 Arbitration

If there is a conflict when the bus master of the internal system bus and bus master of the H bus both attempt to access the same memory bank, round-robin access arbitration is performed to report only one of the access requests.

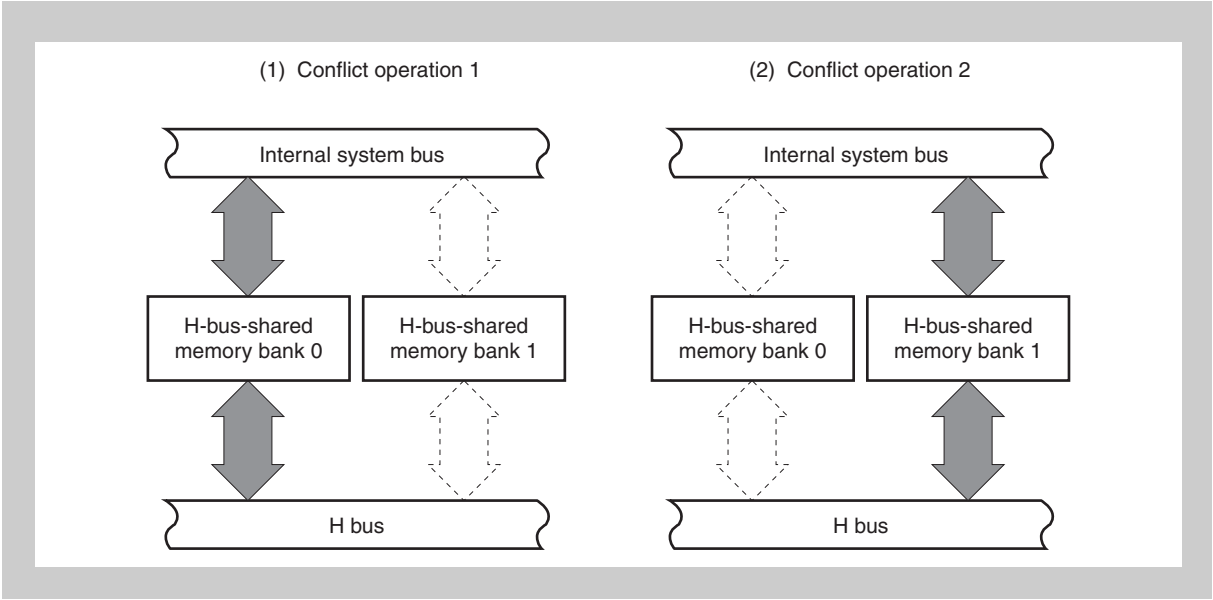


Figure 26-3 Conflict Operation Example

Chapter 27 H-Bus Memory Side Cache

27.1 Features

- Can be assigned to H-bus external memory area
- Unified 16-KB 4-way-set associative
- Sequential/critical word first access
- Flush clearing
- Pseudo-LRU algorithm used
- Write through operation

27.2 Configuration

The H-bus memory side cache uses the unified cache configuration, in which instructions and data are controlled in the same cache, not separated. The H-bus memory side cache is a 16-KB 4-way set associative unified cache, which has four ways consisting of 256 entry blocks each, where one line consists of 4 words, to make up 16 KB of capacity. If a cache miss occurs, the cache is refilled in line units (= 16 bytes).

27.3 Control Registers

27.3.1 ETA setting register (ETACFG)

This register is used to specify operation settings for the H-bus memory side cache functions.

Access This register can be read or written in 16-bit units.

Address $FFFF7100_H$

Initial value 0980_H. This register is initialized by any reset.

Caution Be sure to set bits 11 and 8 to “1”, and clear bits 15 to 12, 10, 9, and 6 to 0 to “0”.

15	14	13	12	11	10	9	8
0	0	0	0	BFM	0	BPM1	BPM0
R	R	R	R	R/W	R	R/W	R/W
7	6	5	4	3	2	1	0
CFM	0	CWM1	CWM0	0	0	0	END
R/W	R	R/W	R/W	R	R	R	R/W

Table 27-1 ETACFG register contents

Bit Position	Bit Name	Description
11	BFM	This bit sets up buffer filling. Be sure to set this to “1”.
9, 8	BPM[1:0]	These bits set up buffer prefetching. Be sure to set these bits to “01”.
7	CFM	This bit sets up cache filling. 0: Sequential 1: Critical words first
5, 4	CWM	These bits set up cache writing. 00: Write through Others: Setting prohibited
0	END	This bit specifies little endian for the H bus. Be sure to clear this to “0”.

27.3.2 ETA command register (ETACMD)

This register is used to flush the cache.

Access This register can be read or written in 16-bit units. However, if the lower 8 bits are used as the ETACMDL register, this register can be read or written in 8- or 1-bit units, and the ETACMDCFL bit can be referenced or updated by using bit manipulation instructions.

Address $FFFF7102_H$

Initial value 0000_H. This register is initialized by any reset.

Caution Be sure to clear bits 15 to 1 to “0”.

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
7	6	5	4	3	2	1	0
0	0	0	0	0	0	ETACMD BFL	ETACMD CFL
R	R	R	R	R	R	R/W	R/W

Table 27-2 ETACMD register contents

Bit Position	Bit Name	Description
1	ETACMD BFL	This bit is used to flush the buffer. Be sure to clear this to “0”.
0	ETACMD CFL	This bit is used to flush the cache. If this bit is set to 1, flushing the cache starts. After flushing ends, this bit is automatically cleared to “0”. 0: Not flushing or stopped/flushing ended 1: Flushing/flushing started

27.3.3 ETA area n setting register (ETARCFGn)

These registers are used to specify whether areas 1 to 3 are enabled or disabled and their operation modes.

Access These registers can be read or written in 16-bit units.

Address ETARCFG1: $FFFF7142_H$, ETARCFG2: $FFFF7144_H$,

ETARCFG3: $FFFF7146_H$

Initial value 0000_H. These registers are initialized by any reset.

Caution Be sure to set bit 5 to “1”, and clear bits 15 to 6 and 4 to 1 to “0”.

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
7	6	5	4	3	2	1	0
MODE				0	0	0	EN
R/W	R/W	R/W	R/W	R	R	R	R/W

Table 27-3 ETARCFGn register contents

Bit Position	Bit Name	Description
7:4	MODE	These bits specify the operation mode for area n. Set these bits to “0010”. All other values are prohibited.
0	EN	This bit specifies whether area n is enabled or disabled. 0: Disable 1: Enable

27.3.4 ETA area n address register (ETARADRSn)

These registers are used to specify the base addresses that specify areas 1 to 3.

Access These registers can be read or written in 32-bit units. However, if the higher 16 bits of the ETARADRSn register are used as the ETARADRSnH register, and the lower 16 bits are used as the ETARADRSnL register, the ETARADRSn register can be read or written in 16-bit units.

Address ETARADRS1: $FFFF7158_H$, ETARADRS1L: $FFFF7158_H$, ETARADRS1H: $FFFF715A_H$,

ETARADRS2: $FFFF7160_H$, ETARADRS2L: $FFFF7160_H$, ETARADRS2H: $FFFF7162_H$,

ETARADRS3: $FFFF7168_H$, ETARADRS3L: $FFFF7168_H$, ETARADRS3H: $FFFF716A_H$

Initial value After power-on: Undefined

After a reset: The previous value is retained. These registers are initialized by any reset.

31	30	29	28	27	26	25	24
ETARADRS[31:24]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
ETARADRS[23:16]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
ETARADRS[15:8]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
ETARADRS[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 27-4 ETARADRSn register contents

Bit Position	Bit Name	Description
31:0	ETARADRS [31:0]	These bits specify the base address that specifies area n. Bit 28 is regarded as the sign bit for bits 31 to 29, in which a sign extended value is stored. Be sure to clear bits 11 to 0 to “0”.

27.3.5 ETA area n mask register (ETARMASKn)

These registers are used to specify mask values for the base addresses that specify areas 1 to 3.

When setting these registers, be sure to specify a value that consists of consecutive 1s from the lower bits.

Access These registers can be read or written in 32-bit units. However, if the higher 16 bits of the ETARMASKn register are used as the ETARMASKnH register, and the lower 16 bits are used as the ETARMASKnL register, the ETARMASKn register can be read or written in 16-bit units.

Address ETARMASK1: $FFFF715C_H$, ETARMASK1L: $FFFF715C_H$, ETARMASK1H: $FFFF715E_H$,

ETARMASK2: $FFFF7164_H$, ETARMASK2L: $FFFF7164_H$, ETARMASK2H: $FFFF7166_H$,

ETARMASK3: $FFFF716C_H$, ETARMASK3L: $FFFF716C_H$, ETARMASK3H: $FFFF716E_H$

Initial value After power-on: Undefined

After a reset: The previous value is retained.These registers are initialized by any reset.

31	30	29	28	27	26	25	24
ETARMASK[31:24]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
ETARMASK[23:16]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
ETARMASK[15:8]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
ETARMASK[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 27-5 ETARMASKn register contents

Bit Position	Bit Name	Description
31:0	ETARMASK [31:0]	These bits are used to specify a mask value for the base address that specifies area n. Be sure to clear bits 31 to 29 to “0”. Be sure to set bits 11 to 0 to “1”.

27.4 Operation

27.4.1 Target

The H-bus memory side cache is used for accessing a cache valid area from the following bus masters:

- PE1
- PE2
- DMA

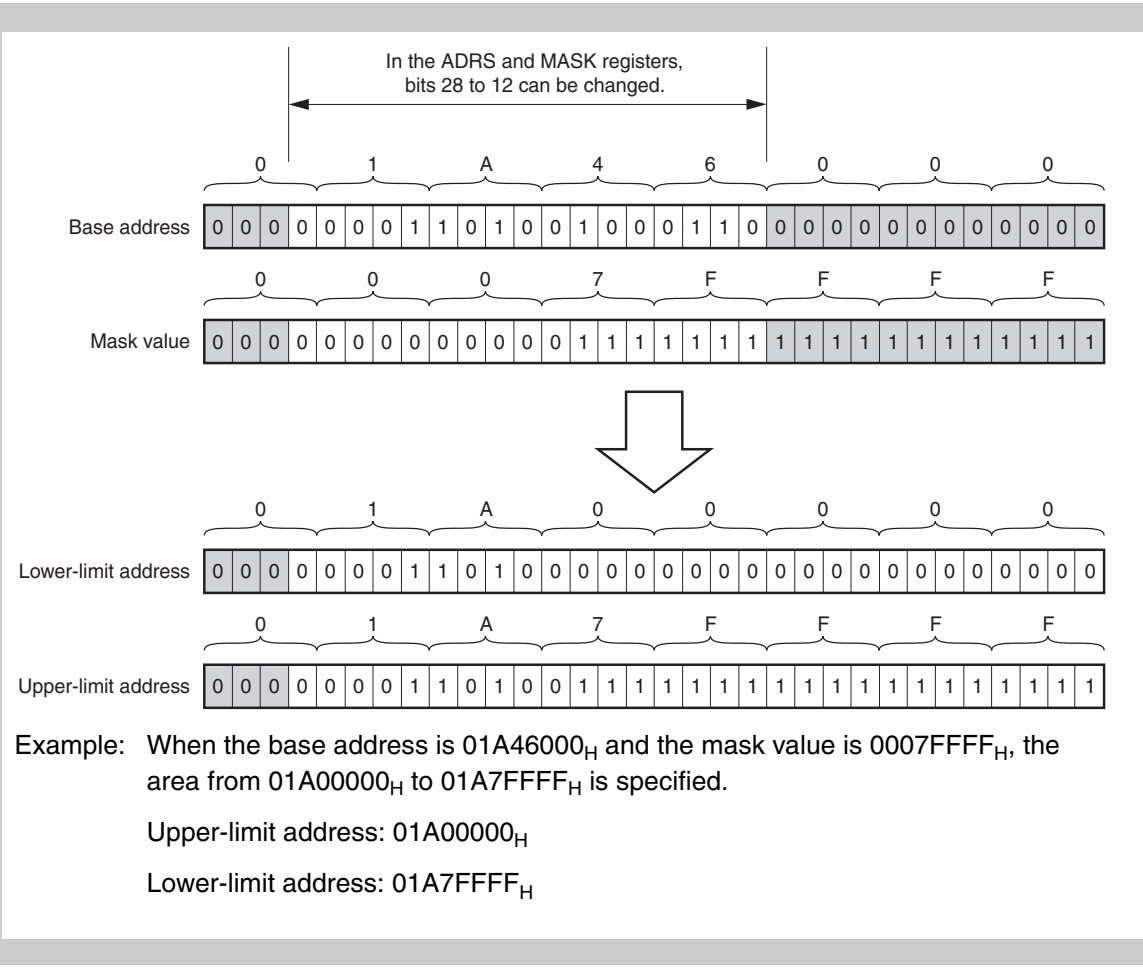
27.4.2 Area specification

The H-bus memory side cache can be assigned to the H-bus external memory area. Up to a maximum of three areas to which to assign the cache can be specified. Do not specify an area that is out of the H-bus external memory area.

An area to which to assign the cache is determined by specifying a base address by using the ETARADRSn register (n = 1 to 3) and a mask value by using the ETARMASKn register (n = 1 to 3). In other words, the lower-limit and upper-limit addresses of the cache area can be specified by the base address and mask value.

The base address whose bits corresponding to “1” of the mask value are replaced with “0” is regarded as the lower-limit address, and the base address whose bits corresponding to “1” of the mask value are replaced with “1” is regarded as the upper-limit address.

Only specify a mask value that consists of consecutive 1s from the lower bits. If a value in which 1 and 0 are placed alternately (for example, $0005FFFF_H$) is specified, the operation is not guaranteed. *Figure 27-1 “Address area specification example”* shows an example of cache area specification.



Example: When the base address is $01A46000_H$ and the mask value is $0007FFFF_H$, the area from $01A00000_H$ to $01A7FFFF_H$ is specified.

Upper-limit address: $01A00000_H$

Lower-limit address: $01A7FFFF_H$

Figure 27-1 Address area specification example

When specifying a cache area, be careful not to overlap other cache areas.

27.4.3 Operation method

(1) Write through

Only the write-through operation is supported for data writing. During the write-through operation, data is transferred to both the cache line and external memory line. The external memory therefore always saves the latest data. Note that caching is not performed when a cache miss hit occurs during a write.

(2) Cache filling

If a cache miss hit occurs during a read, data is used to fill the cache. The filling operation differs depending on the following two filling modes:

- Critical word access

  Starting from the target address of the read request, a burst transfer is performed in this mode, where one line of data is read by way of the H bus. If the requested data has been read, the read data is output to the internal

system bus without waiting for the end of the burst transfer. When one line of data has been read, the one line of data in the cache is updated all at once.

- Sequential access

  Starting from the 0th address on the line to which the target address of the read request belongs, a burst transfer is performed in this mode, where one line of data is read by way of the H bus. If the requested data has been read, the read data is output to the internal system bus without waiting for the end of the burst transfer. When one line of data has been read, the one line of data in the cache is updated all at once.

27.4.4 Read

(1) When a cache hit occurs

<1> If the target address of the read request is in an area assigned for the cache, the content of the corresponding index is read out from the cache.

<2> If the target address is in the cache (a cache hit), the data is output to the internal system bus.

(2) When a cache miss hit occurs

<1> If the target address of the read request is in an area assigned for the cache, the content of the corresponding index is read out from the cache.

<2> If the target address is not in the cache (a cache miss hit), one line of data is burst transferred to the external memory by way of the H bus.

<3> When the data at the target address has been read, the read data is output to the internal system bus.

<4> When one line of data has been read, the one line of data in the cache is updated all at once.

27.4.5 Write

(1) When a cache hit occurs

<1> If the target address of the write request is in an area assigned for the cache, the content of the corresponding index is read out from the cache.

<2> The data is written to the external memory by way of the H bus (write through operation).

<3> If the target address is in the cache (a cache hit), the one line of data in the cache is updated all at once.

(2) When a cache miss hit occurs

<1> If the target address of the write request is in an area assigned for the cache, the content of the corresponding index is read out from the cache.

<2> The data is written to the external memory by way of the H bus (write through operation). The cache is not updated because this is a cache miss hit.

(3) Flush clearing

The flush clearing function clears all the data in the cache at once to invalidate it. Clearing is enabled by writing “1” to the ETACMD.ETACMDCFL bit.

(4) Pseudo LRU

The default priority of the ways is Way0, Way1, Way2, and Way3, from the upper side (to be replaced).

27.5 Setting Procedure

To enable the H-bus memory side cache, perform the following as the initial setup by using a user-created program immediately after a system reset.

<1> Specify the operation mode of the area assigned to the cache by using the ETARCFGn register.

- Set the MODE bits (bits 7 to 4) to “0010” (used for the cache).

<2> Specify the base address of the area by using the ETARADRSn register.

- Specify the base address of the target area by using bits 28 to 12.

<3> Specify the area size by using the ETARMASKn register

- Specify the mask value for the base address of the target area by using bits 28 to 12.

<4> Enable the area assigned to the cache by using the ETARCFGn register.

- Set the EN bit (bit 0) to “1” (enable).

Chapter 28 Secondary Memory Controller (SMEMC)

28.1 Features

An SRAM/SDRAM memory controller is incorporated.

- Separate bus mode (supporting connection of SRAM and SDRAM)
- Chip select output function with 5 areas (4 areas for SRAM, 1 area for SDRAM)
- Bus sizing function: 16/32 bit selectable
- Little endian

28.1.1 Connectable memory

(1) SRAM connection function

- Read cycle requiring at least four cycles
- Write cycle requiring at least five cycles
- Up to 15 address setup wait states can be inserted by register setting.
- Up to 15 data wait states can be inserted by register setting.
- Up to 15 write trigger wait states can be inserted by register setting.
- Up to 15 idle wait states can be inserted by register setting.
- Data wait states can be inserted by an external pin input.

(2) SDRAM connection function

- Only a single access (burst length = 1) can be made (when a burst access is requested from the bus master such as a CPU core, however, a read/write command is issued per clock cycle to make a pseudo-burst access).
- Support of CAS latencies of 2 and 3
  A read or write command can be executed in one clock cycle during a DMA transfer of 128 bits or more (the data size is specified by using the DDS and SDS bits of the SCHCFGn register).A read or write command cannot be executed in one clock cycle during normal CPU operation.
  When a DMA transfer of 128 bits or more is executed for an SDRAM controlled by the secondary memory controller dedicated DMA function, it can be executed as a burst transfer.
- Address multiplex capability
- The address multiplex width (8, 9, 10, or 11 bits) can be changed by register setting.
- Up to three wait states can be inserted by register setting.
- Start of an SDRAM initialization cycle (mode register setting cycle)
- When the dynamic memory control register (DMCn) is written to, the SDRAM is initialized. Upon completion of initialization, the DMCn.IST bit is set.
- Issuance of a CBR (CAS before RAS) refresh command
  At the refresh interval set by the dynamic memory refresh control register (RFCn), a refresh command is issued. The refresh command is issued when a memory access and a bus hold operation are completed.

28.1.2 Pin Descriptions

The table below lists the pins related to the memory controller.

Table 28-1 Memory interface pins

Pin Name	I/O	Description
S_SDCKE	Output	SDRAM clock enable output signal of SMEMC
S_BUSCLK	Output	Bus clock output of SMEMC
$\overline{\text{S\_SDCAS}}$	Output	Column address strobe signal output for SDRAM of SMEMC
$\overline{\text{S\_SDRAS}}$	Output	Row address strobe signal output for SDRAM of SMEMC
S_LLDQM	Output	I/O mask signal output for SDRAM (D0 to D7) of SMEMC
S_LUDQM	Output	I/O mask signal output for SDRAM (D8 to D15) of SMEMC
S_ULDQM	Output	I/O mask signal output for SDRAM (D16 to D23) of SMEMC
S_UUDQM	Output	I/O mask signal output for SDRAM (D0 to D7) of SMEMC
$\overline{\text{S\_LLWR}}$	Output	Write strobe signal output for external data bus (D0 to D7) of SMEMC
$\overline{\text{S\_LUWR}}$	Output	Write strobe signal output for external data bus (D8 to 15) of SMEMC
$\overline{\text{S\_ULWR}}$	Output	Write strobe signal output for external data bus (D16 to D23) of SMEMC
$\overline{\text{S\_UUWR}}$	Output	Write strobe signal output for external data bus (D24 to D31) of SMEMC
$\overline{\text{S\_RD}}$	Output	Read strobe signal output for external data bus of SMEMC
$\overline{\text{S\_WR}}$	Output	Write strobe signal output for external data bus of SMEMC
$\overline{\text{S\_LLBE}}$	Output	Byte enable signal output for external data bus (D0 to D7) of SMEMC
$\overline{\text{S\_LUBE}}$	Output	Byte enable signal output for external data bus (D8 to D15) of SMEMC
$\overline{\text{S\_ULBE}}$	Output	Byte enable signal output for external data bus (D16 to D23) of SMEMC
$\overline{\text{S\_UUBE}}$	Output	Byte enable signal output for external data bus (D24 to D31) of SMEMC
$\overline{\text{S\_SDWE}}$	Output	Write enable signal output for SDRAM of SMEMC
$\overline{\text{S\_BCYST}}$	Output	Strobe signal output indicating bus cycle start of SMEMC
$\overline{\text{S\_CS0}}$	Output	Chip select signal output for external SRAM of SMEMC
$\overline{\text{S\_CS1}}$		
$\overline{\text{S\_CS2}}$		
$\overline{\text{S\_CS3}}$		
$\overline{\text{S\_WAIT}}$	Input	External wait request input of SMEMC
$\overline{\text{S\_HLDAK}}$	Output	Bus hold acknowledge output of SMEMC
$\overline{\text{S\_HLDRQ}}$	Input	Bus hold request input of SMEMC
$\overline{\text{S\_REFRQ}}$	Output	Refresh request signal output for SDRAM of SMEMC
$\overline{\text{S\_SDCS}}$	Output	Chip select signal output for external SDRAM of SMEMC
S_A[1:26]	Output	26-bit address bus for external memory of SMEMC
S_D[0:31]	Input/Output	32-bit data bus for external memory of SMEMC

28.2 Registers

Table 28-2 External memory access control registers

Address	Register Name	Symbol	R/W	Manipulatable Bit Unit			After Reset
				8	16	32	
$F9900004_H$	Bus size control register	SBSC	R/W	–	–	ü	$0000FFFF_H$
$F9900008_H$	Static memory control register 0	SSMC0	R/W	–	–	ü	$0000FFFF_H$
$F990000C_H$	Static memory control register 1	SSMC1	R/W	–	–	ü	$0000FFFF_H$
$F9900010_H$	Static memory control register 2	SSMC2	R/W	–	–	ü	$0000FFFF_H$
$F9900014_H$	Static memory control register 3	SSMC3	R/W	–	–	ü	$0000FFFF_H$
$F990001C_H$	Dynamic memory control register 0	SDMC0	R/W[a]	–	–	ü	$20C00000_H$
$F990002C_H$	Dynamic memory refresh control register 0	SRFC0	R/W	–	–	ü	$001F0000_H$

a) Bit 15 is read-only.

28.2.1 Bus size control register (SBSC)

This register is used to specify the data bus width for the memory to be accessed. Bits SBS[3:0] correspond to the external device chip select signals of the SRAM interface ($\overline{S\_CS}$[3:0]), and bit DBS0 to the chip select signal of SDRAM ($\overline{S\_SDCS}$), respectively.

Access This register can be read or written in 32-bit units.

Address $F9900004_H$

Initial value $0000FFFF_H$. This register is initialized by any reset.

Caution Be sure to clear bits 31 to 16 to “0”. Be sure to set bits 15 to 10, 8, 6, 4, 2, and 0 to “1”.

31	30	29	28	27	26	25	24
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8
1	1	1	1	1	1	DBS0	1
R	R	R	R	R	R	R/W	R
7	6	5	4	3	2	1	0
SBS3	1	SBS2	1	SBS1	1	SBS0	1
R/W	R	R/W	R	R/W	R	R/W	R

Table 28-3 SBSC register contents

Bit position	Bit name	Function
9	DBC0	This bit sets up the data bus width for SDRAM. 0: 16 bits 1: 32 bits
7, 5, 3, 1	SBS[3:0]	These bits set the data bus width for SRAM for each chip select. 0: 16 bits 1: 32 bits

28.2.2 Static memory control register n (SSMCn)

These registers are used to specify the wait states for each SRAM interface external device chip select ($\overline{S\_CS}$[3:0]).

The wait state insertion state is entered for the entire area immediately after a system reset, so change the setting value as needed.

Access These registers can be read or written in 32-bit units.

Address SSMC0: $F9900008_H$, SSMC1: $F990000C_H$, SSMC2: $F9900010_H$, SSMC3: $F9900014_H$

Initial value $0000FFFF_H$. These registers are initialized by any reset.

Caution Be sure to clear bits 31 to 16 to “0”.

31	30	29	28	27	26	25	24
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8
IWn3	IWn2	IWn1	IWn0	WWn3	WWn2	WWn1	WWn0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
DWn3	DWn2	DWn1	DWn0	ACn3	ACn2	ACn1	ACn0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 28-4 SSMCn register contents (1/4)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>15:12</td><td>IWn[3:0]</td><td>These bits set up the number of idle wait states inserted when the $\overline{\text{S\_CS}}$[3:0] area is read.
An idle wait state is a state that is inserted in cases such as when the data float delay time during the read cycle is long and a bus conflict occurs immediately after the write cycle, in order to easily achieve an interface for low-speed devices.
The ideal wait state interval is the interval from the rising edge of the $\overline{\text{S\_CS}}$[3:0] signal (from low to high) until the next falling edge of the S_BCYST signal (from high to low).
<table>
<tr><th>IWn3</th><th>IWn2</th><th>IWn1</th><th>IWn0</th><th>Number of idle wait states</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>2</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>3</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>4</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>5</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td><td>6</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td>7</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>8</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td><td>9</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>0</td><td>10</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td><td>11</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>0</td><td>12</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>1</td><td>13</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>0</td><td>14</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>15</td></tr>
</table>
</td></tr>
</table>

Table 28-4 SSMCn register contents (2/4)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>11:8</td><td>WWn[3:0]</td><td>These bits set up the number of write recovery wait states inserted when the $\overline{\text{S\_CS}}$[3:0] area is read.
A write recovery wait state is the state from the rising edge of the $\overline{\text{S\_WR}}$ signal (from low to high) until the rising edge of the $\overline{\text{S\_CS}}$[3:0] signal (from low to high).
This state is inserted to support devices that require an interval between writes, such as low-speed devices.

Caution: The number of write recovery wait states cannot be set to “0”.
<table>
<tr><th>WWn3</th><th>WWn2</th><th>WWn1</th><th>WWn0</th><th>Number of write recovery wait states</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>2</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>3</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>4</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>5</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td><td>6</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td>7</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>8</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td><td>9</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>0</td><td>10</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td><td>11</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>0</td><td>12</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>1</td><td>13</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>0</td><td>14</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>15</td></tr>
</table>
</td></tr>
</table>

Table 28-4 SSMCn register contents (3/4)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>7:4</td><td>DWn[3:0]</td><td>These bits set up the number of data wait states inserted when the $\overline{\text{S\_CS}}$[3:0] area is read or written.
This extends the active period of the $\overline{\text{S\_RD}}$/$\overline{\text{S\_WR}}$ pin.
Up to 15 data wait states can be inserted for each bus cycle started for each CS area in order to easily achieve an interface for low-speed devices.

Caution: To use external wait insertion ($\overline{\text{S\_WAIT}}$ pin), set the number of data wait states to “1” or more.
<table>
<tr><th>DWn3</th><th>DWn2</th><th>DWn1</th><th>DWn0</th><th>Number of data wait states</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>2</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>3</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>4</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>5</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td><td>6</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td>7</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>8</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td><td>9</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>0</td><td>10</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td><td>11</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>0</td><td>12</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>1</td><td>13</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>0</td><td>14</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>15</td></tr>
</table>
</td></tr>
</table>

Table 28-4 SSMCn register contents (4/4)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>3:0</td><td>ACn[3:0]</td><td>These bits set up the number of address setup wait states inserted when the $\overline{\text{S\_CS}}$[3:0] area is read or written.
An address setup wait state is the state from the falling edge of the $\overline{\text{S\_CS}}$[3:0] signal (from high to low) until the falling edge of the $\overline{\text{S\_RD}}$/$\overline{\text{S\_WR}}$ signal (from high to low). Address setup wait states are inserted to access devices that require a setup time for the address or chip select signal for the read/write strobe.

Caution: The number of address setup wait states cannot be set to “0” during write access. (It must be set to 1 or more.)
<table>
<tr><th>ACn3</th><th>ACn2</th><th>ACn1</th><th>ACn0</th><th>Number of address setup wait states</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0 (read access)
1 (write access)</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>0</td><td>2</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td><td>3</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>4</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>1</td><td>5</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td><td>6</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td>7</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>8</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>1</td><td>9</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>0</td><td>10</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td><td>11</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>0</td><td>12</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>1</td><td>13</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>0</td><td>14</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>15</td></tr>
</table>
</td></tr>
</table>

28.2.3 Dynamic memory control register 0 (SDMC0)

This register is used to specify the latency, number of wait states, row address width, and multiplex width of the SDRAM connected to the SDRAM chip select pin (S_SDCS).

Access This register can be read or written in 32-bit units. However, the IST0 bit is read-only, and values written to it are ignored.

Address $F990001C_H$

Initial value $20C00000_H$. This register is initialized by any reset.

Notes

1. Be sure to clear bits 31, 30, 27 to 24, 21, 20, and 14 to 0 to “0”.
2. Access the SDRAM after checking that the ST0 bit is set. If the SDRAM is accessed before the ST0 bit is set, access is pended because the state is the no ready state. Set a value to the SDMC0 register during memory controller initialization. Do not change the register setting after initialization.
3. To write a value again to the SDMC0 register, do so when mode register write has been completed (IST0 bit = 1).

31	30	29	28	27	26	25	24
0	0	LTM01	LTM00	0	0	0	0
R	R	R/W	R/W	R	R	R	R
23	22	21	20	19	18	17	16
BCW01	BCW00	0	0	RAW01	RAW00	SAW01	SAW00
R/W	R/W	R	R	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
IST0	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R
7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R

Table 28-5 SDMC0 register contents (1/2)

Bit position	Bit name	Function
29:28	LTM[01:00]	These bits specify the CAS latency value applied when the SDRAM is read.
23:22	BCW[01:00]	These bits specify the interval (number of wait states) between the issuance of a bank active command and the issuance of a read/write command, or between the issuance of a precharge command and the issuance of a bank active command.

LTM01	LTM00	CAS latency
1	0	2
1	1	3
Other than above		Setting prohibited

BCW01	BCW00	Number of wait states
0	0	Setting prohibited
0	1	1
1	0	2
1	1	3

Table 28-5 SDMC0 register contents (2/2)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>19:18</td><td>RAW[01:00]</td><td>These bits specify the row address width.
<table>
<tr><th>RAW01</th><th>RAW00</th><th>Row address width</th></tr>
<tr><td>0</td><td>0</td><td>11 bits</td></tr>
<tr><td>0</td><td>1</td><td>12 bits</td></tr>
<tr><td>1</td><td>0</td><td>13 bits</td></tr>
<tr><td>1</td><td>1</td><td>Setting prohibited</td></tr>
</table></td></tr>
<tr><td>17:16</td><td>SAW[01:00]</td><td>These bits specify the address multiplex width.
<table>
<tr><th>SAW01</th><th>SAW00</th><th>Address multiplex width</th></tr>
<tr><td>0</td><td>0</td><td>8 bits</td></tr>
<tr><td>0</td><td>1</td><td>9 bits</td></tr>
<tr><td>1</td><td>0</td><td>10 bits</td></tr>
<tr><td>1</td><td>1</td><td>11 bits</td></tr>
</table></td></tr>
<tr><td>15</td><td>IST0</td><td>This bit indicates the mode register write status.
0: Mode register write not completed
1: Mode register write completed
The completion of the mode register write cycle can be confirmed by reading the IST0 bit. Even if, after the SDMC0 has been written to, the SDMC0 register is to be written to again, execute this write operation after reading the IST0 bit to confirm that the register write cycle has been completed. If the SDMC0 register is written to while this bit is set, the IST0 bit is cleared once and is set again upon completion of the mode register write cycle.
The IST0 bit is read-only, and values written to it are ignored.</td></tr>
</table>

28.2.4 Dynamic memory refresh control register 0 (SRFC0)

This register is used to enable refresh operations and specify the refresh interval.

Access This register can be read or written in 32-bit units.

Address $F990002C_H$

Initial value $001F0000_H$. This register is initialized by any reset.

Notes

1. Be sure to clear bits 31 to 21 and 14 to 10 to “0”. If these bits are set to other than 0, the operation cannot be guaranteed.
2. A refresh operation is caused when the value of the internal interval timer and the refresh interval register setting match. An actual refresh operation, however, may be kept waiting depending on the external bus status (cycle execution is in progress, or the current state is the bus hold state). Therefore, the refresh interval to be specified must be shorter than the memory specification considering the system bus utilization.
3. Specify the refresh interval (set up by the RCCn and RIN bits) so that it is longer than the refresh interval in the mode register setting cycle (set up by the RFP bit).

31	30	29	28	27	26	25	24
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
23	22	21	20	19	18	17	16
0	0	0	RFP04	RFP03	RFP02	RFP01	RFP00
R	R	R	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
REN	0	0	0	0	0	RCC01	RCC00
R/W	R	R	R	R	R	R/W	R/W
7	6	5	4	3	2	1	0
RIN07	RIN06	RIN05	RIN04	RIN03	RIN02	RIN01	RIN00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 28-6 SFRC0 register contents (1/2)

Bit position	Bit name	Function
20:16	RFP[04:00]	These bits specify the refresh command interval in the mode register setting cycle.
15	REN0	This bit enables or disables refresh operations for SDRAM. 0: Disables refresh operations. 1: Enables refresh operations.

RFP04	RFP03	RFP02	RFP01	RFP00	Refresh interval
0	0	0	0	0	Setting prohibited
0	0	0	0	1	2
0	0	0	1	0	3
0	0	0	1	1	4
:					:
1	1	1	1	0	3
1	1	1	1	1	32

Table 28-6 SFRC0 register contents (2/2)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>9:8</td><td>RCC[01:00]</td><td>These bits specify the source clock factor for the refresh interval counter.
<table>
<tr><th>RCC01</th><th>RCC00</th><th>Count source clock factor</th></tr>
<tr><td>0</td><td>0</td><td>32</td></tr>
<tr><td>0</td><td>1</td><td>128</td></tr>
<tr><td>1</td><td>0</td><td>256</td></tr>
<tr><td>1</td><td>1</td><td>Setting prohibited</td></tr>
</table></td></tr>
<tr><td>7:0</td><td>RIN[07:00]</td><td>These bits specify the refresh interval factor.
<table>
<tr><th>RIN[07:00]</th><th>Interval factor</th></tr>
<tr><td>00000000</td><td>1</td></tr>
<tr><td>00000001</td><td>2</td></tr>
<tr><td>00000010</td><td>3</td></tr>
<tr><td>00000011</td><td>4</td></tr>
<tr><td>:</td><td>:</td></tr>
<tr><td>11111101</td><td>254</td></tr>
<tr><td>11111110</td><td>255</td></tr>
<tr><td>1111111</td><td>256</td></tr>
</table></td></tr>
</table>

An example of specifying the refresh interval (4096/64 ms) is given below. The source clock factor and interval factor for the refresh interval specification are specified according to the operating clock used.

Table 28-7 Example of specifying the SDRAM refresh interval setting

Refresh interval specification (μs)	Source clock factor (RCCn1 and RCCn0)	Interval factor (RINn7 to RINn0)
		f_{SDCLK} = 50 MHz
64 × 1000 / 4096 ≈15.6	00	24 (15.4)
	01	6 (15.4)
	10	3 (15.4)

Notes 1. Calculated refresh intervals (μs) are enclosed in parentheses.

2. f_{SDCLK}: Operating clock (HCLK)

28.3 Operation

28.3.1 Chip select

S_CS3 and S_SDCS share the same pin (P10_7/INTP26). Therefore, when S_SDCS is used, S_CS3 of the (P10_7/S_CS3/S_SDCS/INTP26) pin cannot be used.

In this case, to use S_CS3, use the (P13_1/S_DMAAK0/ADTRG1/S_CS3/INTP0) pin.

28.3.2 Bus hold function

When it is acknowledged that the S_HLDRQ signal becomes active as a bus acquisition request from another bus master, the memory controller goes into the bus hold state. After the memory controller state has changed to the bus hold state, the S_HLDAK signal becomes active. During the bus hold period, the S_HLDAK signal remains active.

When the S_HLDRQ signal becomes inactive, the memory controller cancels the bus hold state and the S_HLDAK signal becomes inactive.

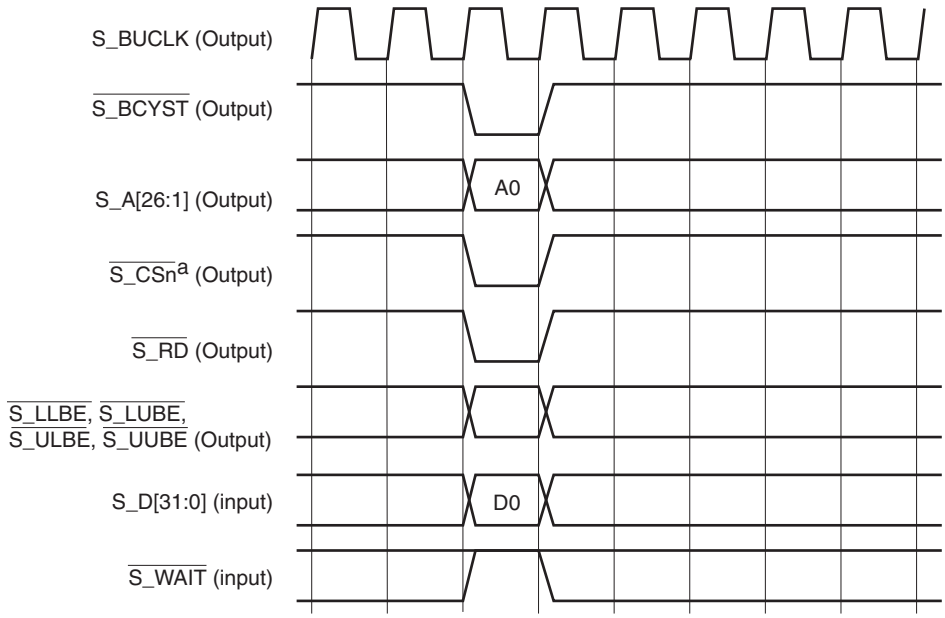
If the S_HLDRQ signal becomes active while an external memory cycle is executed, the state changes to the bus hold state upon completion of the bus cycle, and the S_HLDAK signal becomes active.

When a refresh request is output to SDRAM during the bus hold period, the memory controller sets the S_REFRQ signal to the active level to notify the external bus master of the generation of a transfer request.

Caution During the bas hold period, the external bus master must execute the bank precharge commend before accessing SDRAM. When the bus hold state is released, however, the external bus master need not execute any particular command for the SDRAM. Following return from the bus hold, the memory controller executes the first access to SDRAM using the all-bank precharge command.

28.3.3 SRAM

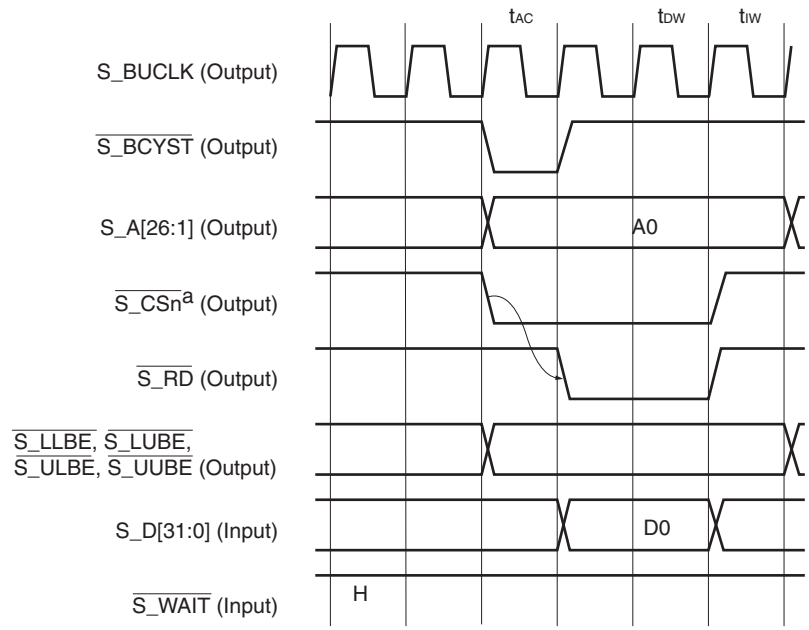
(1) SRAM access



a) Only the specified S_CSn (n = 0 to 3) pin is active.

Note: SRAM: 32 bits (SBSC.SBS[3:0] = 1111)
Idle wait state: 0 (SSMCn.IWn[3:0] = 0000)
Data wait state: 0 (SSMCn.DWn[3:0] = 0000)
Address setup wait state: 0 (SSMCn.ACn[3:0] = 0000)

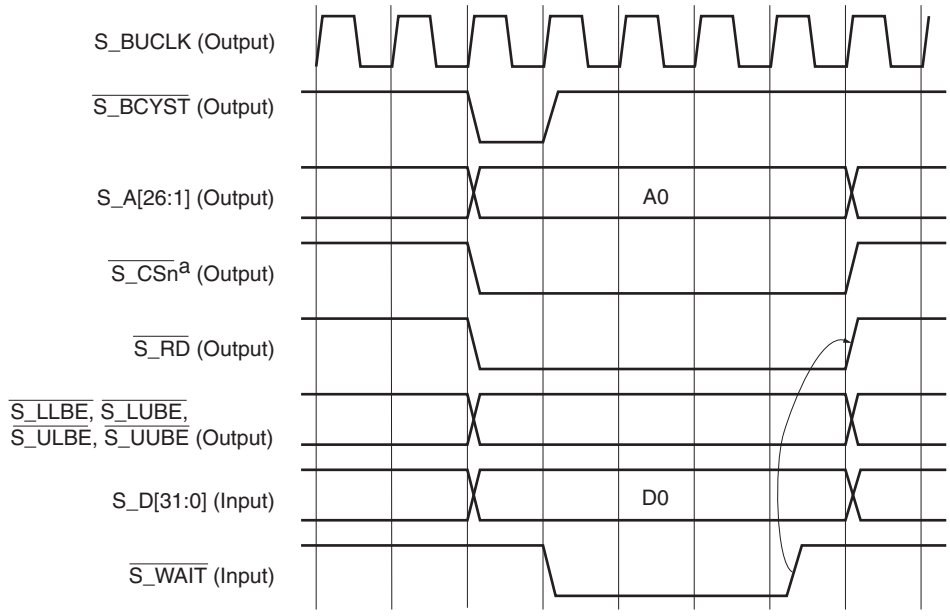
Figure 28-1 SRAM read cycle (with no wait setting)



a) Only the specified S_CSn (n = 0 to 3) pin is active.

Note: SRAM: 32 bits (SBSC.SBS[3:0] = 1111)
Idle wait state (t_{IW}): 1 (SSMCn.IWn[3:0] = 0001)
Data wait state (t_{DW}): 1 (SSMCn.DWn[3:0] = 0001)
Address setup wait state (t_{AC}): 1 (SSMCn.ACn[3:0] = 0001)

Figure 28-2 SRAM read cycle (with wait setting)

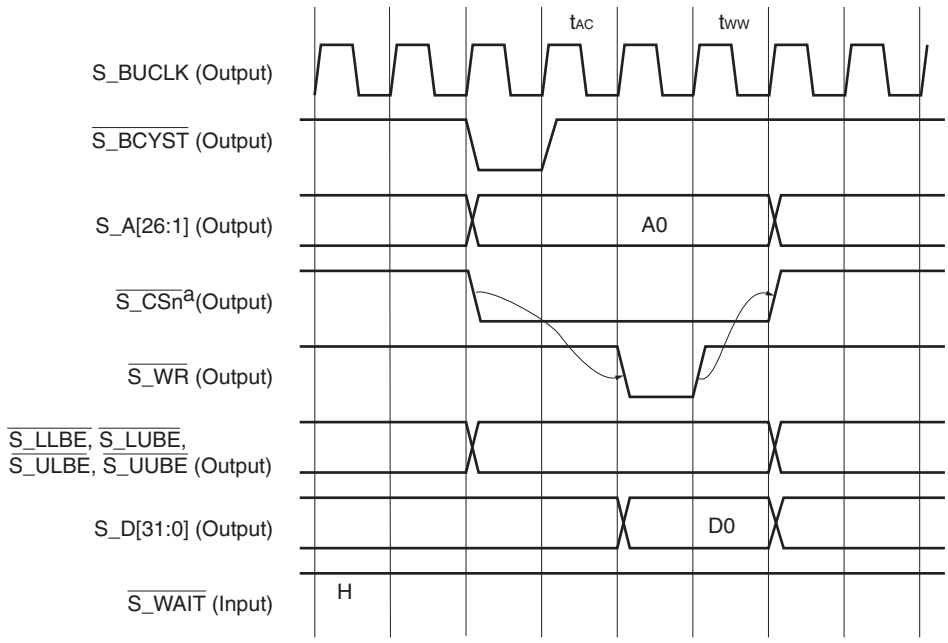


a) Only the specified S_CSn (n = 0 to 3) pin is active.

Note: SRAM: 32 bits (SBSC.SBS[3:0] = 1111)
Idle wait state: 0 (SSMCn.IWn[3:0] = 0000)
Data wait state: 1 (SSMCn.DWn[3:0] = 0001)
Address setup wait state: 0 (SSMCn.ACn[3:0] = 0000)

Figure 28-3 SRAM read cycle (external wait states inserted)

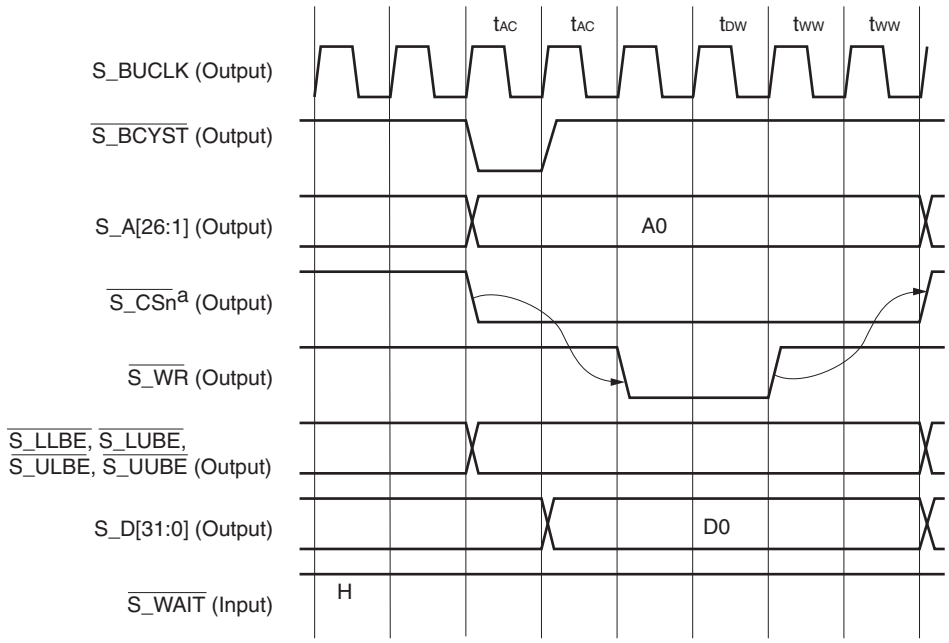
Caution To enable the external wait pin (S_WAIT), the number of data wait states (SSMCn.DWn[3:0] = 0001) must be set to “1” or more.



a) Only the specified S_CSn (n = 0 to 3) pin is active.

Note: SRAM: 32 bits (SBSC.SBS[3:0] = 1111)
Write recovery wait state (t_{WW}): 1 (SSMCn.WWn[3:0] = 0000/0001)
Data wait state: 0 (SSMCn.DWn[3:0] = 0000)
Address setup wait state (t_{AC}): 1 (SSMCn.ACn[3:0] = 0000/0001)

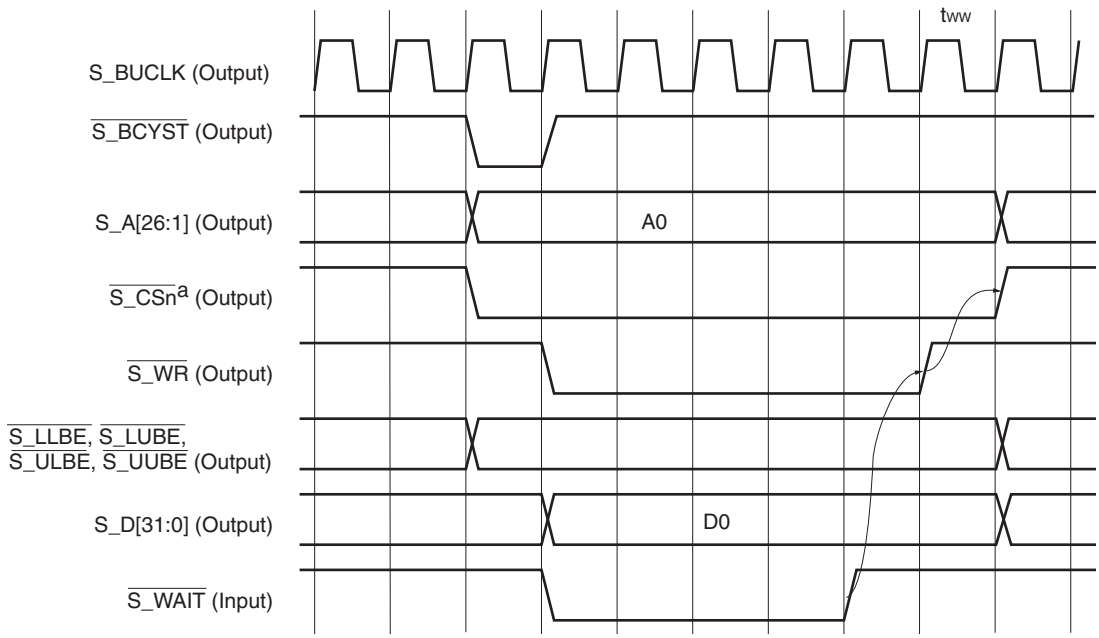
Figure 28-4 SRAM write cycle (with no wait)



a) Only the specified $\overline{\text{S\_CSn}}$ (n = 0 to 3) pin is active.

Note: SRAM: 32 bits (SBSC.SBS[3:0] = 1111)
Write recovery wait state (t_{WW}): 2 (SSMCn.WWn[3:0] = 0010)
Data wait state (t_{DW}): 1 (SSMCn.DWn[3:0] = 0001)
Address setup wait state (t_{AC}): 2 (SSMCn.ACn[3:0] = 0010)

Figure 28-5 SRAM write cycle (with wait)



a) Only the specified S_CSn (n = 0 to 3) pin is active.

Note: SRAM: 32 bits (SBSC.SBS[3:0] = 1111)
Write recovery wait state (t_{WW}): 1 (SSMCn.WWn[3:0] = 0000/0001)
Data wait state: 1 (SSMCn.DWn[3:0] = 0001)
Address setup wait state: 1 (SSMCn.ACn[3:0] = 0000/0001)

Figure 28-6 SRAM write cycle (external wait states inserted)

Caution To enable the external wait pin (S_WAIT), the number of data wait states (SSMCn.DWn[3:0] = 0001) must be set to “1” or more.

28.3.4 SDRAM

(1) SDRAM control

The secondary memory controller controls reading and writing data to/from the SDRAM. While transferring data with the SDRAM, the secondary memory controller state changes as shown in *Figure 28-7 "SDRAM access state changes"*.

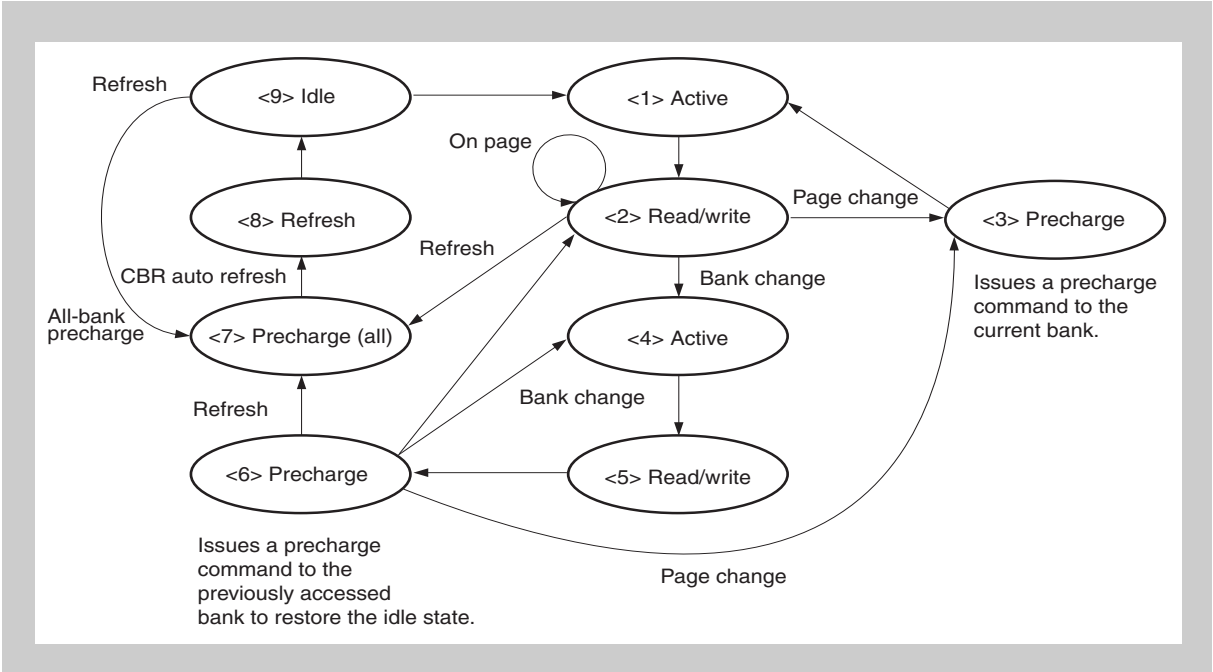


Figure 28-7 SDRAM access state changes

Based on the settings in the dynamic memory control register (SDMC0) and bus size control register (SBSC), the V850E2/MN4 recognizes the bank address, row address, and column address. For the address mapping according to the register settings, see *Table 28-12 "Address mapping"*.

The V850E2/MN4 compares the current address to be accessed with the previously accessed address to select the command to be issued to SDRAM. For the SDRAM connected to the same channel, a selected row address is present only in one bank.

(a) When an access is made after initialization, after refresh operation, or after the bus hold state is released

At the time of initialization or refresh operation, an all-bank precharge command is always issued, so all SDRAMs are placed in the idle state. Therefore, in the first access made after the initialization or refresh operation, an active command and a read or write command are issued sequentially to complete the access (<9> to <1> to <2>).

When an access is made for the first time after the bus hold state is released, a precharge command is issued to the bank to be accessed, then an active command and a read or write command are issued sequentially to complete the access.

After the access ends, the accessed SDRAM bank is in the state <2> in which a row address is selected.

If the refresh counter is hit in this state, an all-bank precharge command is issued before the refresh operation. So all SDRAMs are placed in the idle state again (<2> to <7> to <8> to <9>).

(b) On-page access

In cases other than the access described in (a), if the previously accessed bank address and row address and the currently accessed bank address and row address are the same, an on-page access is determined. Since the accessed SDRAM bank is in the state <2> in which a row address is selected, only a read or write command is issued while a page match is observed (<2> to <2> to ... to <2>).

After the access ends, the accessed SDRAM bank is in the state <2> in which a row address is selected.

If the refresh counter is hit in this state, the state changes from <2> to <7> to <8> to <9>, and all SDRAMs are placed in the idle state again.

(c) Page change access

In cases other than the access described in (a), if the previously accessed bank address and the currently accessed bank address are the same but the previously accessed row address and the currently accessed row address differ, a page change access is determined. Since the SDRAM bank to be accessed is in the state <2> in which a row address is selected, a precharge command is issued first to the bank to be accessed to enter the idle state. Subsequently, an active command is used to issue a new row address, then a read or write command is issued to complete the access (<2> to <3> to <1> to <2>).

After the access ends, the accessed SDRAM bank is in the state <2> in which a row address is selected.

If the refresh counter is hit in this state, the state changes from <2> to <7> to <8> to <9>, and all SDRAMs are placed in the idle state again.

(d) Bank change access

In cases other than the access described in (a), if the previously accessed bank address and the currently accessed bank address differ, a bank change access is determined. The SDRAM bank to be accessed is in the idle state, so an active command and a read or write command are issued sequentially to complete the access. Furthermore, in the previously accessed SDRAM bank, a row address is selected. So, a precharge command is issued to the previously accessed bank to enter the idle state (<4> to <5> to <6>). Therefore, there is always only one SDRAM bank in which a row address is selected.

After the access ends, the accessed SDRAM bank is in the state <6> (or <5>) in which a row address is selected. When an on-page access is made next, the state changes from <6> to <2>; when a page change access is made, the state changes from <6> to <3> to <1> to <2>; when a bank change access is made, the state changes from <6> to <4> to <5> to <6>. If the refresh counter is hit in this state, the state changes from <2> to <7> to <8> to <9>, and all SDRAMs are placed in the idle state again.

(2) Address control

The following describes address control in the dynamic memory cycle.

Table 28-8 When the active command is issued (a row address is output)

Bit setting		Address pin																	
SAW1	SAW0	A27-A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1
0	0	a27-a18	a25	a24	a23	a22	a21	a20	a19	a18	a17	a16	a15	a14	a13	a12	a11	a10	a9
0	1	a28-a18	a26	a25	a24	a23	a22	a21	a20	a19	a18	a17	a16	a15	a14	a13	a12	a11	a10
1	0	a28-a18	a27	a26	a25	a24	a23	a22	a21	a20	a19	a18	a17	a16	a15	a14	a13	a12	a11
1	1	a28-a18	a28	a27	a26	a25	a24	a23	a22	a21	a20	a19	a18	a17	a16	a15	a14	a13	a12

Table 28-9 When the all-bank precharge command is issued (A10 =1 in SDRAM is assumed)

Bit setting	Address pin																	
DBSn	A27-A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1
0	a27-a18	a17	a16	a15	a14	a13	a12	1	a10	a9	a8	a7	a6	a5	a4	a3	a2	a1
1	a28-a18	a17	a16	a15	a14	a13	1	a11	a10	a9	a8	a7	a6	a5	a4	a3	a2	a1

Table 28-10 When the register write command is issued

Bit setting	Address pin																	
DBSn	A27-A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1
0	0	0	0	0	0	0	0	0	0	0	0	LTMn2	LTMn1	LTMn0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	LTMn2	LTMn1	LTMn0	0	0	0	0	0

Table 28-11 When the read/write command is issued (A column address is output. A10 = 0 in SDRAM is assumed, and an auto precharge command is not issued)

Bit setting	Address pin																	
DBSn	A27-A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1
0	a27-a18	a17	a16	a15	a14	a12	a11	0	a10	a9	a8	a7	a6	a5	a4	a3	a2	a1
1	a28-a18	a17	a16	a15	a14	a12	0	a11	a10	a9	a8	a7	a6	a5	a4	a3	a2	a1

(3) Address mapping

An address is determined based on the settings of the following registers:

- RAWn[1:0] (row address width) and SAWn[1:0] (address multiplex width) of the SDMC0 register
- DBS[3:0] (memory bus width) of the SBSC register

Table 28-12 "Address mapping" shows address mapping by register settings. The secondary memory controller compares the previous address with the current address to be accessed and selects a command to be issued to the SDRAM.

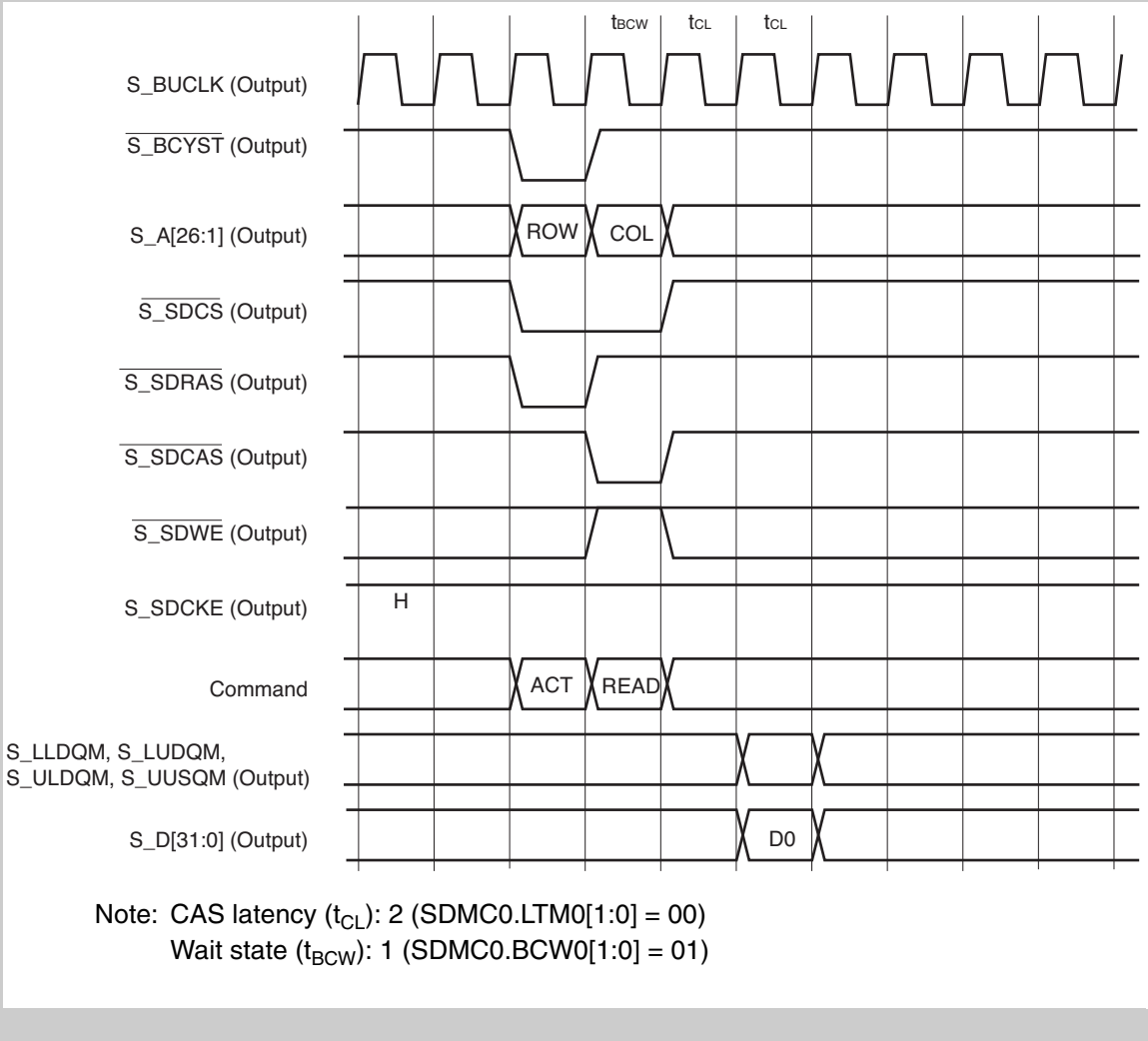
- If the previous bank address and row address and the current bank address and row address are the same, an on-page access is determined.
- If the previous bank address and the current bank address are the same but the previous row address and the current row address differ, a page-change access is determined.
- If the previous bank address and the current bank address differ, a bank-change access is determined.

Table 28-12 Address mapping

RA width	CA width	Memory bus width	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
11	8	16	BA							RA (11 bits)											CA (8 bits)								–
12	8	16	BA							RA (12 bits)											CA (8 bits)								–
13	8	16	BA							RA (13 bits)											CA (8 bits)								–
11	9	16	BA							RA (11 bits)											CA (9 bits)								–
12	9	16	BA							RA (12 bits)											CA (9 bits)								–
13	9	16	BA							RA (13 bits)											CA (9 bits)								–
11	10	16	BA							RA (11 bits)											CA (10 bits)								–
12	10	16	BA							RA (12 bits)											CA (10 bits)								–
13	10	16	BA							RA (13 bits)											CA (10 bits)								–
11	11	16	BA							RA (11 bits)											CA (11 bits)								–
12	11	16	BA							RA (12 bits)											CA (11 bits)								–
13	11	16	BA							RA (13 bits)											CA (11 bits)								–
11	8	32	BA							RA (11 bits)											CA (8 bits)							–	–
12	8	32	BA							RA (12 bits)											CA (8 bits)							–	–
13	8	32	BA							RA (13 bits)											CA (8 bits)							–	–
11	9	32	BA							RA (11 bits)											CA (9 bits)							–	–
12	9	32	BA							RA (12 bits)											CA (9 bits)							–	–
13	9	32	BA							RA (13 bits)											CA (9 bits)							–	–
11	10	32	BA							RA (11 bits)											CA (10 bits)							–	–
12	10	32	BA							RA (12 bits)											CA (10 bits)							–	–
13	10	32	BA							RA (13 bits)											CA (10 bits)							–	–
11	11	32	BA							RA (11 bits)											CA (11 bits)							–	–
12	11	32	BA							RA (12 bits)											CA (11 bits)							–	–

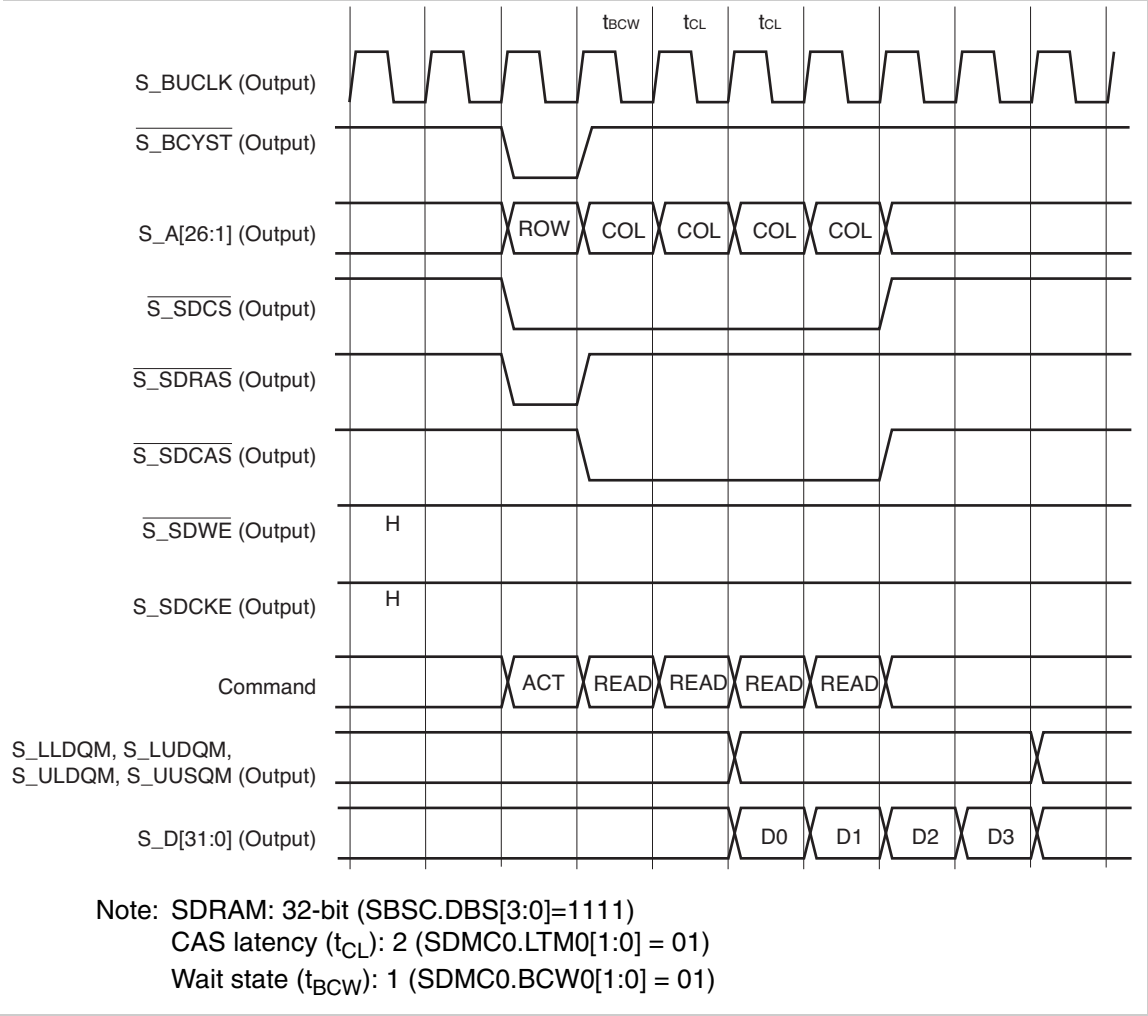
Note BA: Bank address
RA: Row address
CA: Column address

(4) SDRAM access



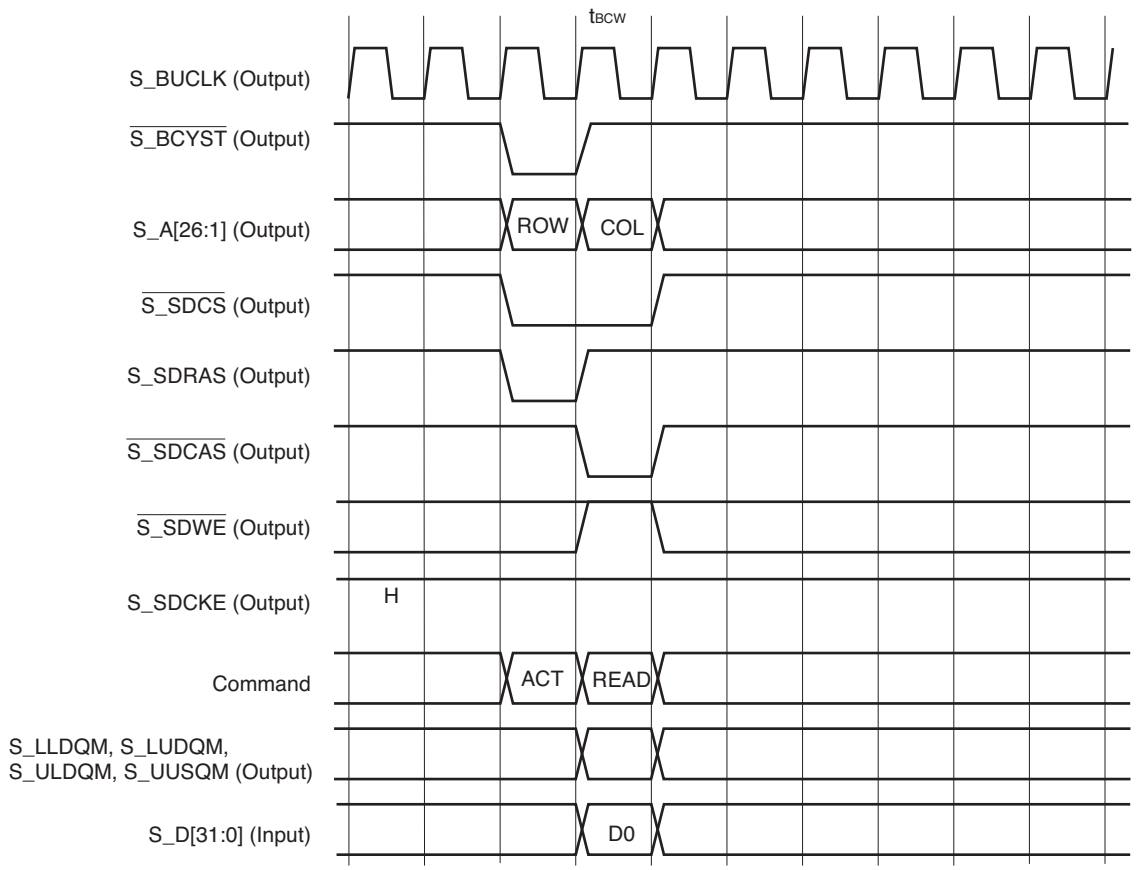
Note: CAS latency (t_{CL}): 2 (SDMC0.LTM0[1:0] = 00)
Wait state (t_{BCW}): 1 (SDMC0.BCW0[1:0] = 01)

Figure 28-8 SDRAM read cycle (single transfer)



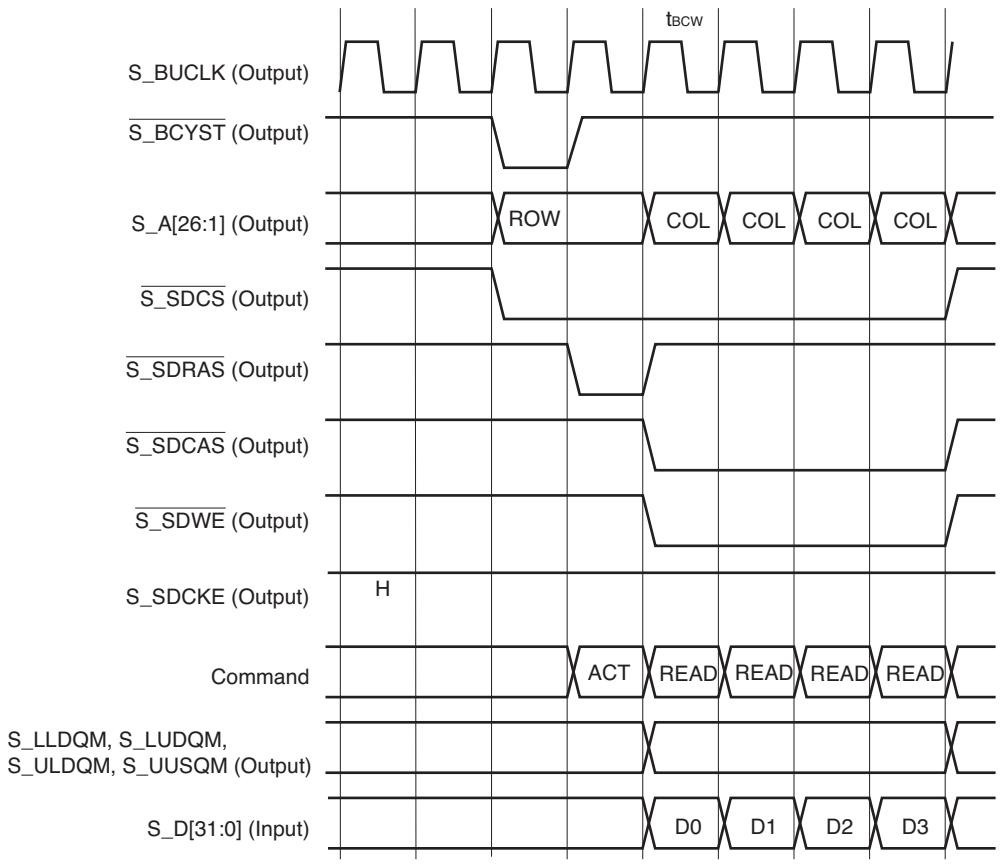
Note: SDRAM: 32-bit (SBSC.DBS[3:0]=1111)
CAS latency (t_{CL}): 2 (SDMC0.LTM0[1:0] = 01)
Wait state (t_{BCW}): 1 (SDMC0.BCW0[1:0] = 01)

Figure 28-9 SDRAM read cycle (4 times continuous transfer)



Note: CAS latency: 2 (SDMC0.LTM0[1:0] = 01)
Wait state (t_{BCW}): 1 (SDMC0.BCW0[1:0] = 01)

Figure 28-10 SDRAM write cycle (single transfer)



Note: SDRAM: 32 bits (SBSC.DBS[3:0]=1111)
CAS latency: 2 (SDMC0.LTM0[1:0] = 01)
Wait state (t_{BCW}): 1 (SDMC0.BCW0[1:0] = 01)

Figure 28-11 SDRAM write cycle (4 times continuous transfer)

Chapter 29 Secondary Memory Controller (Dedicated DMA)

29.1 Features

29.1.1 Function

The functions of the secondary memory controller dedicated DMA are described below.

(1) Transfer setting value loading method

The setting data used for DMA transfer is set to internal registers using the following two methods.

Register mode In this mode, DMA transfer is performed using the set values in the internal registers written by the CPU.

Link mode In this mode, the DMA automatically loads the set values (descriptor data) on the external memory placed by the CPU, and performs DMA transfer based on this data. DMA transfer of multiple data can be performed sequentially by specifying the descriptor addresses for the subsequent transfers in the descriptor and setting the values for multiple DMA transfers on the memory.

Moreover, continuation or stopping of the next DMA transfer can be specified in the header (information field) of the descriptor.

(2) Trigger method

DMA transfer can be triggered using one of the following two methods.

Software trigger DMA transfer is triggered using software via an internal register.

Hardware trigger DMA transfer is triggered according to the state of the $\overline{\text{S\_DMARQm}}$ input pin. The following detection modes are supported.

- Rising edge detection
- Falling edge detect
- Change point detection
- High-level detection
- Low-level detection
- Detection mask

In the case of hardware trigger, the $\overline{\text{S\_DMAAKm}}$ output is made active to be used as the handshake signal. The following modes are supported for the $\overline{\text{S\_DMAAKm}}$ output.

- Active for one-pulse cycle at the start of transfer
- Active until the $\overline{\text{S\_DMARQm}}$ input pin level becomes inactive
- Active during a bus cycle period
- Active mask

After the transfer of the set size, if the DMA transfer of the total number of transfer bytes specified by the control register is completed, the $\overline{\text{S\_DMATCm}}$ terminal count pin is made active (and can also be masked).

It is also possible to switch between executing one read/write operation per trigger, and executing transfer of the total number of transfer bytes that has been set.

(3) Interrupt

Upon completion of transfer, the INTHDMAn transfer completion interrupt is generated (and can also be masked).

If a bus error occurs, the INTHDMAERR error interrupt is generated.

(4) Transfer

The DMA transfer size can be selected in the range of 8 to 512 bits.

Both a mode that increments the transfer address at each transfer and a mode that keeps the transfer address always fixed are supported.

(5) SKIP (Scatter/Gather)

The continuous access size and discrete access size can each be set for the areas that are accessed with DMA transfer. Following access of the set size, it is possible to skip to the next address to be accessed by the set size.

(6) Intra-buffer data output

Data that has already been loaded to the buffer can be forcibly output to the destination area.

- Sweep mode: The data in the buffer can be output when the DMAC is forcibly stopped.
- Software forced emission request: The data in the buffer can be output by software. DMA transfer is continued following data output.

(7) Suspend

The execution of a DMA transaction that is currently being executed can be paused (suspended).

(8) Interval

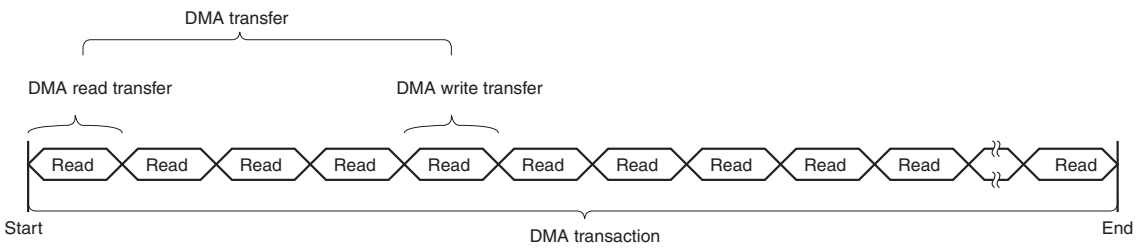
The interval between DMA transfers can be set in order to adjust the bus usage rate.

29.2 Definitions of Terms

The terms used in this chapter are defined below.

Table 29-1 Definitions of Terms

Term	Definition
Burst	Refers to one bus cycle.
DMA transfer	Refers to the read/write transfer of one-burst data performed by the DMAC.
DMA read transfer	Refers to the read transfer of one-burst data performed by the DMAC.
DMA write transfer	Refers to the write transfer of one-burst data performed by the DMAC.
DMA transaction	Refers to the DMA transfers of the total number of transfer bytes set to the DMAC, in other words the interval during which a series of DMA transfers is completed.
Register set	Refers to a group of registers.
Descriptor	Refers to the data that consists of the DMA transfer settings, which are read by the DMAC in the link mode.
DMAC	Refers to this macro.
DMA trigger	Refers to the triggers for starting DMA transfer, such as hardware requests, software triggers, and those generated internally during the block mode.
Aligned	Refers to the state in which the specified address indicates the start of the size boundary for transfer. Concretely, this is the state in which bit [(log2SIZE – 1) : 0] of the specified start address is 0 (SIZE: transfer size in bytes). Bit aligned: Refers to the state in which the address indicates the start of the alignment boundary of the data size. Word aligned: Refers to the state in which the address indicates the start of the word boundary.
Unaligned	Refers to the state in which the specified address indicates other than the start of the size boundary for transfer. Concretely, this is the state in which bit [(log2SIZE – 1) : 0] of the specified start address is not 0 (SIZE: transfer size in bytes). Beat unaligned: Refers to the state in which the address indicates the start of the alignment boundary of the data size.
[SEL]	Indicates the bit position of the $\overline{\text{S\_DMARQ}}$[0:3], $\overline{\text{S\_DMAAK}}$[0:3], and $\overline{\text{S\_DMATC}}$[0:3] pins set by the SCHCFGn.SEL[1:0] bits.
INTHDMAn	Indicates the INTDMA signal corresponding to channel n.
n register	Indicates the channel n register.
Reserved	Provided for future function expansion. The functions indicated as “Reserved” are not supported in this version.
I/F	Abbreviation of "interface"



One read/write transfer executed by the DMAC is called a DMA transfer.

The execution of a set series of DMA transfers is called a DMA transaction.

Figure 29-1 DMA Transfer State Names

Table 29-2 DMA Pins

Pin Name	I/O	Description
$\overline{\text{S\_DMARQ[0:3]}}$	Input	DMA transaction request input Connect to the unit issuing the transfer request to the DMAC.
$\overline{\text{S\_DMAAK[0:3]}}$	Output	DMA acknowledge output Connect to the unit issuing the transfer request to the DMAC.
$\overline{\text{S\_ DMATC[0:3]}}$	Output	DMA terminal count output (pulse) Connect to the unit issuing the transfer request to the DMAC.

Table 29-3 Interrupt Signals

Interrupt	Description
INTHDMA[7:0]	DMA transaction completion output When SDCTRL.LVINT = 0, the INTHDMA[7:0] signal behaves as pulse output, in which the signal automatically becomes inactive (low) one-clock cycle after being activated (high) once. When SDCTRL.LVINT = 1, the INTHDMA[7:0] signal behaves as level output, in which the output level is kept. Upon completion of the DMA transaction, INTHDMA[7:0] becomes active. If INTHDMA[7:0] becomes active while LVINT = 1, the active level is kept until the SCHSTATn.END bit is cleared. Also, when the descriptor is read in the link mode while SCHCFGn.DRRP is 0, INTHDMA[7:0] becomes active if LV in the header is 0. Once INTHDMA[7:0] becomes active while LVINT = 1, the active level is kept until the SCHSTATn.DER bit is cleared. INTHDMA[7:0] can be masked temporarily by setting the SCHSTATn.INTM bit.
INTHDMAERR	Error response (ERROR) interrupt output When SDCTRL.LVINT = 0, the INTHDMAERR signal behaves as pulse output, in which the signal automatically becomes inactive (low) one-clock cycle after being activated (high) once. When SDCTRL.LVINT = 1, the INTHDMAERR signal behaves as level output, in which the output level is kept. INTHDMAERR becomes active if an error response is returned during bus access. Once INTHDMAERR becomes active while LVINT = 1, the active level is kept until the SCHSTATn.ER bit is cleared.

29.3 Control Registers

29.3.1 Next register set

(1) SN0SA0 to SN0SA7: Next 0 source address registers 0 to 7

The SN0SAn register specifies the DMA transfer source address for DMA channel n (n = 0 to 7).

This register is used for the next 0 register set.

In the write-only mode (SCHCFGn.WONLY = 1), it is used for setting write data. (For details, see *(3) "Write-only mode" on page 1874*.)

Access This register can be read or written in 32-bit units.

Address SN0SA0: $F9900400_H$, SN0SA1: $F9900440_H$, SN0SA2: $F9900480_H$, SN0SA3: $F99004C0_H$, SN0SA4: $F9900500_H$, SN0SA5: $F9900540_H$, SN0SA6: $F9900580_H$, SN0SA7: $F99005C0_H$

Initial value $0000\ 0000_H$. This register is initialized by any reset.

Caution The SN0SAn register is overwritten with data read from the descriptor during DMA transfer in the link mode.

(a) Normal mode

31	30	29	28	27	26	25	24
SA031	SA030	SA029	SA028	SA027	SA026	SA025	SA0240
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
SA023	SA022	SA021	SA020	SA019	SA018	SA017	SA016
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
SA015	SA014	SA013	SA012	SA011	SA010	SA09	SA08
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
SA07	SA06	SA05	SA04	SA03	SA02	SA01	SA00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 29-4 Next 0 Source Address Register n (SN0SAn) Contents (Normal Mode)

Bit Position	Bit Name	Description
31:0	SA0[31:0]	Source address These bits specify the start address of the DMA transfer source.

(b) Write-only mode

31	30	29	28	27	26	25	24
WD031	WD030	WD029	WD028	WD027	WD026	WD025	WD024
W	W	W	W	W	W	W	W
23	22	21	20	19	18	17	16
WD023	WD022	WD021	WD020	WD019	WD018	WD017	WD016
W	W	W	W	W	W	W	W
15	14	13	12	11	10	9	8
WD015	WD014	WD013	WD012	WD011	WD010	WD09	WD08
W	W	W	W	W	W	W	W
7	6	5	4	3	2	1	0
WD07	WD06	WD05	WD04	WD03	WD02	WD01	WD00
W	W	W	W	W	W	W	W

Table 29-5 Next 0 Source Address Register n (SN0SAn) Contents (Write-Only Mode)

Bit Position	Bit Name	Description
31:0	WD0[31:0]	Write data These bits specify the data to be written in the write-only mode.

(2) SN0DA0 to SN0DA7: Next 0 destination address registers 0 to 7

The SN0DAn register specifies the DMA transfer destination address for DMA channel n (n = 0 to 7).

This register is used for the next 0 register set.

Access This register can be read or written in 32-bit units.

Address SN0DA0: $F9900404_H$, SN0DA1: $F9900444_H$, SN0DA2: $F9900484_H$, SN0DA3: $F99004C4_H$, SN0DA4: $F9900504_H$, SN0DA5: $F9900544_H$, SN0DA6: $F9900584_H$, SN0DA7: $F99005C4_H$

Initial value $0000\ 0000_H$. This register is initialized by any reset.

Caution The SN0DAn register is overwritten with data read from the descriptor during DMA transfer in the link mode.

31	30	29	28	27	26	25	24
DA031	DA030	DA029	DA028	DA027	DA026	DA025	DA024
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
DA023	DA022	DA021	DA020	DA019	DA018	DA017	DA016
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
DA015	DA014	DA013	DA012	DA011	DA010	DA09	DA08
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
DA07	DA06	DA05	DA04	DA03	DA02	DA01	DA00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 29-6 Next 0 Destination Address Register n (SN0DAn) Contents

Bit Position	Bit Name	Description
31:0	DA0[31:0]	Destination address These bits specify the start address of the DMA transfer destination.

(3) SN0TB0 to SN0TB7: Next 0 transaction byte registers 0 to 7

The SN0TBn register specifies the total number of transfer bytes (DMA transaction) of DMA channel n (n = 0 to 7).

This register is used for the next 0 register set.

Access This register can be read or written in 32-bit units.

Address SN0TB0: $F9900408_H$, SN0TB1: $F9900448_H$, SN0TB2: $F9900488_H$, SN0TB3: $F99004C8_H$, SN0TB4: $F9900508_H$, SN0TB5: $F9900548_H$, SN0TB6: $F9900588_H$, SN0TB7: $F99005C8_H$

Initial value $0000\ 0000_H$. This register is initialized by any reset.

Caution The SN0TBn register is overwritten with data read from the descriptor during DMA transfer in the link mode.

31	30	29	28	27	26	25	24
TB031	TB030	TB029	TB028	TB027	TB026	TB025	TB024
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
TB023	TB022	TB021	TB020	TB019	TB018	TB017	TB016
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
TB015	TB014	TB013	TB012	TB011	TB010	TB09	TB08
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
TB07	TB06	TB05	TB04	TB03	TB02	TB01	TB00
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 29-7 Next 0 Transaction Byte Register n (SN0TBn) Contents

Bit Position	Bit Name	Description
31:0	TB0[31:0]	Transaction byte These bits specify the total number of transfer bytes.

Caution Do not start the DMA transaction when 0 is set with this register.

29.3.2 Current register set

The current register set indicates the transfer source address, transfer destination address, and total number of transfer bytes for DMA transfer.

In the register mode, the set values are loaded from the next 0 register set and the next 1 register set, and in the link mode, they are loaded from the descriptor read data. The current register set cannot be written using software.

(1) SCRSA0 to SCRSA7: Current source address registers 0 to 7

The SCRSAn register indicates the DMA transfer source address for DMA channel n (n = 0 to 7).

Access This register can be read or written in 32-bit units.

Address SCRSA0: F9900418$_H$, SCRSA1: F9900458$_H$, SCRSA2: F9900498$_H$, SCRSA3: F99004D8$_H$, SCRSA4: F9900518$_H$, SCRSA5: F9900558$_H$, SCRSA6: F9900598$_H$, SCRSA7: F99005D8$_H$

Initial value 0000 0000$_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
CRSA31	CRSA30	CRSA29	CRSA28	CRSA27	CRSA26	CRSA25	CRSA24
R	R	R	R	R	R	R	R
23	22	21	20	19	18	17	16
CRSA23	CRSA22	CRSA21	CRSA20	CRSA19	CRSA18	CRSA17	CRSA16
R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8
CRSA15	CRSA14	CRSA13	CRSA12	CRSA11	CRSA10	CRSA9	CRSA8
R	R	R	R	R	R	R	R
7	6	5	4	3	2	1	0
CRSA7	CRSA6	CRSA5	CRSA4	CRSA3	CRSA2	CRSA1	CRSA0
R	R	R	R	R	R	R	R

Table 29-8 Current Source Address Register n (SCRSAn) Contents

Bit Position	Bit Name	Description
31:0	CRSA [31:0]	Current source address These bits indicate the read address of the next DMA transaction. The value is automatically incremented during DMA transaction. (The value is fixed if SCHCFGn.SAD = 1, and undefined if SCHCFGn.WONLY = 1.) The initial value is loaded from the following register: In register mode:The transfer source address is loaded from SN0SAn. In link mode:The transfer source address is loaded from the descriptor. (The descriptor read data is assigned to the SN0SAn register, and then assigned to the SCRSAn register during transfer.) The value is incremented upon completion of read transfer. The SCRSAn register must be read after the DMA has stopped (by clearing SCHSTATn.EN). (Treat a value during DMA operation as reference.)

(2) SCRDA0 to SCRDA7: Current destination address registers 0 to 7

The SCRDAn register indicates the DMA transfer destination address for DMA channel n (n = 0 to 7).

Access This register can be read or written in 32-bit units.

Address SCRDA0: F990041C$_H$, SCRDA1: F990045C$_H$, SCRDA2: F990049C$_H$, SCRDA3: F99004DC$_H$, SCRDA4: F990051C$_H$, SCRDA5: F990055C$_H$, SCRDA6: F990059C$_H$, SCRDA7: F99005DC$_H$

Initial value 0000 0000$_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
CRDA31	CRDA30	CRDA29	CRDA28	CRDA27	CRDA26	CRDA25	CRDA24
R	R	R	R	R	R	R	R
23	22	21	20	19	18	17	16
CRDA23	CRDA22	CRDA21	CRDA20	CRDA19	CRDA18	CRDA17	CRDA16
R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8
CRDA15	CRDA14	CRDA13	CRDA12	CRDA11	CRDA10	CRDA9	CRDA8
R	R	R	R	R	R	R	R
7	6	5	4	3	2	1	0
CRDA7	CRDA6	CRDA5	CRDA4	CRDA3	CRDA2	CRDA1	CRDA0
R	R	R	R	R	R	R	R

Table 29-9 Current Destination Address Register n (SCRDAn) Contents

Bit Position	Bit Name	Description
31:0	CRDA [31:0]	Current destination address These bits indicate the read address of the next DMA transaction. The value is automatically incremented during DMA transaction. (The value is fixed if SCHCFGn.DAD = 1, and undefined if SCHCFGn.WONLY = 1.) The initial value is loaded from the following register: In register mode:The transfer destination address is loaded from SN0DAn. In link mode:The transfer destination address is loaded from the descriptor. (The descriptor read data is assigned to the SN0DAn register, and then assigned to the SCRDAn register during transfer.) The value is incremented upon completion of write transfer. The SCRDAn register must be read after the DMA has stopped (by clearing SCHSTATn.EN). (Treat a value during DMA operation as reference.)

(3) SCRTB0 to SCRTB7: Current transaction byte registers 0 to 7

The SCRTBn register indicates the total number of transfer bytes (DMA transaction) of DMA channel n (n = 0 to 7).

The value is reset to 0 upon completion of transfer.

Access This register can be read or written in 32-bit units.

Address SCRTB0: $F9900420_H$, SCRTB1: $F9900460_H$, SCRTB2: $F99004A0_H$, SCRTB3: $F99004E0_H$, SCRTB4: $F9900520_H$, SCRTB5: $F9900560_H$, SCRTB6: $F99005A0_H$, SCRTB7: $F99005E0_H$

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
CRTB31	CRTB30	CRTB29	CRTB28	CRTB27	CRTB26	CRTB25	CRTB24
R	R	R	R	R	R	R	R
23	22	21	20	19	18	17	16
CRTB23	CRTB22	CRTB21	CRTB20	CRTB19	CRTB18	CRTB17	CRTB16
R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8
CRTB15	CRTB14	CRTB13	CRTB12	CRTB11	CRTB10	CRTB9	CRTB8
R	R	R	R	R	R	R	R
7	6	5	4	3	2	1	0
CRTB7	CRTB6	CRTB5	CRTB4	CRTB3	CRTB2	CRTB1	CRTB0
R	R	R	R	R	R	R	R

Table 29-10 Current Transaction Byte Register n (SCRTBn) Contents

Bit Position	Bit Name	Description
31:0	CRTB [31:0]	Current transaction byte These bits indicate the number of remaining transfer bytes of the DMA transaction that is currently being executed. The value is automatically decremented during DMA transaction. The initial value is loaded from the following register: In register mode:The number of transfer bytes is loaded from the SN0TBn register. In link mode:The number of transfer bytes is loaded from the descriptor. (The descriptor read data is assigned to the SN0TBn register, and then assigned to the SCRTBn register during transfer.) The value is decremented upon completion of write transfer. The SCRTBn register must be read after the DMA has stopped (by clearing SCHSTATn.EN). (Treat a value during DMA operation as reference.)

29.3.3 Channel register set

(1) SCHSTAT0 to SCHSTAT7: Channel status registers 0 to 7

The SCHSTATn register indicates the state of DMA channel n (n = 0 to 7).

Access This register can be read or written in 32-bit units.

Address SCHSTAT0: F9900424$_H$, SCHSTAT1: F9900464$_H$, SCHSTAT2: F99004A4$_H$, SCHSTAT3: F99004E4$_H$, SCHSTAT4: F9900524$_H$, SCHSTAT5: F9900564$_H$, SCHSTAT6: F99005A4$_H$, SCHSTAT7: F99005E4$_H$

Initial value 0000 0000$_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	INTMSK
R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8
0	0	0	0	MODE	DER	DW	DL
R	R	R	R	R	R	R	R
7	6	5	4	3	2	1	0
0	TC	END	CRSA4ER	CRSA3SUS	CRSA2TACT	CRSA1RQST	CRSA0EN
R	R	R	R	R	R	R	R

Table 29-11 Channel Status Register n (SCHSTATn) Contents (1/4)

Bit Position	Bit Name	Description
16	INTMSK	This bit indicates the temporary mask status of the INTHDMA[n] interrupt pin output. 1: Masked 0: Unmasked This bit is set when: • SETINTMSK is set. This bit is cleared when: • CLRINTMSK is set. • SWRST is set.
11	MODE	DMA mode This bit indicates the DMA mode. This reflects the set value of the SCHCFGn.DMS bit. • 0: Register mode • 1: Link mode

Table 29-11 Channel Status Register n (SCHSTATn) Contents (2/4)

Bit Position	Bit Name	Description
10	DER	Descriptor error This bit indicates whether the data read from the descriptor is invalid (LV = 0). (This does not depend on the set value of the SCHCFGn.DIM bit.) 0: No descriptor error has occurred. 1: A descriptor error has occurred. This bit is set when: • Data is loaded from the descriptor in the link mode and LV is 0 while SCHCFGn.DRRP is 0. This bit is cleared when: • SCHCTRLn.CLRDER is set. • SCHCTRLn.SWRST is set.
9	DW	Descriptor write back This bit indicates whether data is being written back to the descriptor. If a bus error has occurred while data is written back to the descriptor, this bit holds 1. 0: The header is not being written back to in the link mode. 1: When ER = 0: The header is being written back to in the link mode. When ER = 1: A bus error has occurred while the header is being written back to in the link mode. This bit is set when: • Writing back of the header starts in the link mode. This bit is cleared when: • The header has been written back in the link mode and an OK response is returned. • SCHCTRLn.SWRST is set.
8	DL	Descriptor load This bit indicates whether data is being loaded from the descriptor. If a bus error has occurred while data is being loaded from the descriptor, this bit holds 1. 0: No data is being loaded from the descriptor. 1: When ER = 0: Data is being loaded from the descriptor in the link mode. When ER = 1: A bus error has occurred while data is being loaded from the descriptor in the link mode. This bit is set when: • Loading data from the descriptor starts in the link mode. This bit is cleared when: • Data has been loaded from the descriptor in the link mode and an OK response is returned. • SCHCTRLn.SWRST is set.

Table 29-11 Channel Status Register n (SCHSTATn) Contents (3/4)

Bit Position	Bit Name	Description
6	TC	Terminal count This is a status bit that indicates whether DMA transaction has completed. This bit is set only when SCHCFGn.TCM is cleared. 0: DMA transfer has not completed. 1: DMA transfer has completed. This bit is set when: • The total number of transfer bytes specified for the SCRTBn register has been transferred completely in the register mode. • The total number of transfer bytes specified for the SCRTBn register has been transferred completely with WBD of the descriptor header = 1 in the link mode. • Data has been written back to the descriptor with WBD of the descriptor header = 0 in the link mode This bit is cleared when: • SCHCTRLn.CLRTC is set. • SCHCTRLn.SWRST is set.
5	END	INTHDMAn interrupt This bit indicates whether the DMA transaction has completed and the INTHDMAn interrupt has been generated. 0: DMA transfer has not completed. 1: DMA transfer has completed. This bit is set when: • The TC bit set condition is met and SCHCFGn.DEM is cleared. • The descriptor is read in the link mode and LV of the header is 0, SCHCFGn.DRRP is 0, and DIM is 0. This bit is cleared when: • SCHCTRLn.CLREND is set. • SCHCTRLn.SWRST is set.
4	ER	Error bit This bit indicates whether an error response was returned during DMA transfer and the INTHDMAERR interrupt has been generated. 0: No error response was returned. 1: An error response was returned. This bit is set when: • An error response was returned in a bus cycle. This bit is cleared when: • SCHCTRLn.SWRST is set.
3	SUS	Suspend This bit indicates whether the channel is suspended. 0: Channel n is not suspended. 1: Channel n is suspended. This bit is set when: • SETSUS is set during DMA transfer of channel n and the channel is suspended. This bit is cleared when: • SCHCTRLn.CLRSUS is set. • SCHCTRLn.CLREN is set.

Table 29-11 Channel Status Register n (SCHSTATn) Contents (4/4)

Bit Position	Bit Name	Description
2	TACT	Transaction active This bit indicates whether the DMAC is in progress. It is used to verify that the channel has completely stopped. 0: DMA on channel n has stopped or is in the trigger wait state. 1: DMA on channel n is in progress. This set is set when: • DMA transaction for channel n starts. This bit is cleared when: • DMA transaction has completed.
1	RQST	Request This bit indicates whether a transfer request has been acknowledged. 0: No DMA transfer request has been acknowledged. 1: A DMA transfer request has been acknowledged. This bit is set when: • SCHCTRLn.STG is set. • A transfer request has been acknowledged from the S_DMARQn pin specified for the SCHCFGn register. This bit is cleared when: • SCHCTRLn.SWRST is set. • SCHCTRLn.CLRRQ is set. • Data transfer at the side (source or destination) specified by SCHCFGn.REQD has completed in the single transfer mode (TM = 0). • DMA transaction has completed in the register mode. • DMA transfer of the last descriptor (LE = 1) has completed in the link mode. • The descriptor read operation stopped (LV = 0 and DRRP = 0) in the link mode. • DMA transaction has finished in the link mode with DEM = 0. • The master interface received a bus error.
0	EN	Enable This bit indicates whether the DMA channel n operation is enabled or stopped. 0: Operation is stopped. 1: Operation is enabled. This bit is set when: • SCHCTRLn.SETEN is set. This bit is cleared when: • SCHCTRLn.SWRST is set. • SCHCTRLn.CLREN is set. • An error response is returned during transfer. • DMA transaction has completed in the register mode. • DMA transfer (write back when WBD = 0) of the last descriptor (LE = 1) has finished in the link mode. • The descriptor read operation stopped (LV = 0 and DRRP = 0) in the link mode.

Cautions

1. If the ER bit is set during DMA transfer, the entire transfer must be regarded to be invalid.
2. To interrupt a DMA transaction, either mask and clear the transfer request or clear the EN bit. For the procedure, see *29.5.14 “Transfer halt function” on page 1896*.
3. If a transfer request by a DMA transfer request pin ($\overline{\text{S\_DMARQm}}$) input and a request by software (by setting the STG bit) are used in combination, it is not possible to determine which source triggered the transfer. Therefore, make sure that the system uses only one of these transfer requests.
4. To perform a transfer request by software, execute the next STG bit operation after the previously requested DMA transfer operation has been completed (this can be checked using the current register.).

(2) SCHCTRL0 to SCHCTRL7: Channel control registers 0 to 7

The SCHCTRLn register controls the DMA transfer of DMA channel n (n = 0 to 7).

Access This register can be read or written in 32-bit units.

Address SCHCTRL0: F9900428$_H$, SCHCTRL1: F9900468$_H$, SCHCTRL2: F99004A8$_H$, SCHCTRL3: F99004E8$_H$, SCHCTRL4: F9900528$_H$, SCHCTRL5: F9900568$_H$, SCHCTRL6: F99005A8$_H$, SCHCTRL7: F99005E8$_H$

Initial value 0000 0000$_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
23	22	21	20	19	18	17	16
0	0	0	0	0	0	CLRINTMSK	SETINTMSK
R	R	R	R	R	R	R/W	R/W
15	14	13	12	11	10	9	8
0	0	0	0	0	0	CLRSUS	SETSUS
R	R	R	R	R	R	R/W	R/W
7	6	5	4	3	2	1	0
CLRDER	CLRTC	CLREND	CLRRQ	SWRST	STG	CLREN	SETEN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 29-12 Channel Control Register n (SCHCTRLn) Contents (1/2)

Bit Position	Bit Name	Description
17	CLRINTMSK	Setting the CLRINTMSK bit cancels masking of the INTHDMAn interrupt. Also, the SCHSTATn.INTMSK bit is cleared when this bit is set. If this bit is set while DCTRL.LVINT is 1 and SCHSTATnEND is 1, the INTHDMAn interrupt is generated. Note that the INTHDMAn interrupt is not generated if this bit is set while DCTRL.LVINT is 0. Reading this bit returns 0. 1: Cancel masking specified for the SETINTMSK bit. 0: No effect on the operation
16	SETINTMSK	Setting the SETINTMSK bit temporarily masks the INTHDMAn interrupt. Also, the SCHSTATn.INTMSK bit is set when this bit is set. Reading this bit returns 0. 1: Mask the INTHDMAn interrupt. 0: No effect on the operation
9	CLRSUS	Clear suspend This bit cancels the suspended state. Setting the CLRSUS bit while the SCHSTATn.SUS bit is 1 cancels the suspended state. Reading this bit returns 0. 1: Cancel suspension of the DMA transfer under execution. 0: No effect on the operation
8	SETSUS	Set suspend This bit suspends the DMA transfer under execution. Setting the SETSUS bit while the SCHSTATn.EN bit is 1 suspends the DMA transfer under execution. Reading this bit returns 0. 1: Suspend the DMA transfer under execution. 0: No effect on the operation

Table 29-12 Channel Control Register n (SCHCTRLn) Contents (2/2)

Bit Position	Bit Name	Description
7	CLRDER	Clear DER bit Setting the CLRDER bit clears the SCHSTATn.DER bit. Also, the INTHDMA interrupt pin is set to low level when this bit is set. Reading this bit returns 0. 1: Clear the DER bit. 0: No effect on the operation
6	CLRTC	Clear TC bit Setting the CLRTC bit clears the SCHSTATn.TC bit. Reading this bit returns 0. 1: Clear the TC bit. 0: No effect on the operation
5	CLREND	Clear END bit Setting the CLREND bit clears the SCHSTATn.END bit. Also, the INTHDMA interrupt pin is set to low level when this bit is set. Reading this bit returns 0. 1: Clear the END bit. 0: No effect on the operation
4	CLRRQ	Clear RQST bit Setting the CLRRQ bit clears the SCHSTATn.RQST bit. Reading this bit returns 0. 1: Clear the RQST bit. 0: No effect on the operation
3	SWRST	Software reset Setting the SWRST bit clears the status register. Set this bit when both the SCHSTATn.EN and SCHSTATn.TACT bits are 0. Reading this bit returns 0. 1: Reset the channel status register. 0: No effect on the operation
2	STG	Software trigger Setting the STG bit sets an internal transfer request (software trigger). If this bit and the SWRST bit are set at the same time, the SWRST bit takes priority. Reading this bit returns 0. 1: Set a transfer request by software. (The RQST bit is set.) 0: No effect on the operation
1	CLREN	Clear enable Setting the CLREN bit clears the SCHSTATn.EN bit. (For details, see *29.5.14 "Transfer halt function" on page 1896*.) Reading this bit returns 0. 1: Stop DMA transfer. (The EN bit is cleared.) 0: No effect on the operation
0	SETEN	Set enable Setting the SETEN bit enables DMA transfer of DMA channel n. If this bit and the SWRST bit are set at the same time, the SWRST bit takes priority. Reading this bit returns 0. 1: Enable DMA transfer. (The EN bit is set.) 0: No effect on the operation

(3) SCHCFG0 to SCHCFG7: Channel configuration registers 0 to 7

The SCHCFGn register configures the DMA transfer of DMA channel n (n = 0 to 7).

Access This register can be read or written in 32-bit units.

Address SCHCFG0: F990042C$_H$, SCHCFG1: F990046C$_H$, SCHCFG2: F99004AC$_H$, SCHCFG3: F99004EC$_H$, SCHCFG4: F990052C$_H$, SCHCFG5: F990056C$_H$, SCHCFG6: F99005AC$_H$, SCHCFG7: F99005EC$_H$

Initial value 0000 0000$_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
DMS	0	0	0	SBE	DIM	TCM	DEM
R/W	R	R	R	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
WONLY	TM	DAD	SAD	DDS3	DDS2	DDS1	DDS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
SDS3	SDS2	SDS1	SDS0	DRRP	AM2	AM1	AM0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
–	LVL	LOEN	HIEN	REQD	0	SEL1	SEL0
R	R/W	R/W	R/W	R/W	R	R/W	R/W

Table 29-13 Channel Configuration Register n (SCHCFGn) Contents (1/4)

Bit Position	Bit Name	Description
31	DMS	DMA mode select This bit specifies the DMA mode. 0: Register mode (Default) 1: Link mode
27	SBE	Sweep buffer enable This bit sets whether to sweep out (write) the data already stored in the buffer before stopping the DMA transfer when the SCHSTATn.EN bit is cleared during a DMA transaction. The sweep mode can be used only when REQD is 0. 0: Stop the transfer without sweeping out the buffer data. (Default) 1: Stop the transfer after sweeping out the buffer data.
26	DIM	Descriptor interrupt mask This bit specifies whether to mask INTHDMAn (n: interrupt selected for SEL[1:0] bits) if LV = 0 when the header of the descriptor is read. 0: Do not mask INTHDMAn. (Default) 1: Mask INTHDMAn.
25	TCM	$\overline{\text{S\_DMATCn}}$ mask This bit masks the $\overline{\text{S\_DMATCn}}$ interrupt. The $\overline{\text{S\_DMATCn}}$ interrupt is disabled if it is output while this bit is 1. At this time, this bit is automatically cleared. Use this bit to control DMA transfer by software. 0: Do not mask the interrupt. (Default) 1: Mask the interrupt. This bit is cleared when: • DMA transaction has completed while TCM is 1.

Table 29-13 Channel Configuration Register n (SCHCFGn) Contents (2/4)

Bit Position	Bit Name	Description
24	DEM	INTHDMAn mask This bit masks the INTHDMAn (n: selected by SEL[1:0] bits) interrupt output during transfer in the register mode. The INTHDMAn interrupt is disabled if it is output while this bit is 1. At this time, this bit is automatically cleared. 0: Do not mask the interrupt. (Default) 1: Mask the interrupt. This bit is cleared when: • DMA transaction has completed while DEM is 1.
23	WONLY	Write-only mode This bit specifies the write-only mode. For details about the write-only mode, see *(3) "Write-only mode" on page 1874*. 0: Normal operation (Default) 1: Write-only mode
22	TM	Transfer mode This bit specifies the DMA transfer mode. 0: Single transfer mode (Default) 1: Block transfer mode
21	DAD	This bit specifies the count direction of the transfer destination address of DMA channel n. 0: Increment (Default) 1: Fixed Do not set this bit if the skip mode is used on the destination side or if the destination side is beat unaligned.
20	SAD	This bit specifies the count direction of the transfer source address of DMA channel n. 0: Increment (Default) 1: Fixed Do not set this bit if the skip mode is used on the source side or if the source side is beat unaligned.
19:16	DDS[3:0]	Destination data size These bits specify the DMA transfer size. The normal mode and skip mode are switched with DDS3. 0: Normal mode (Default) 1: Skip mode The transfer size is set with DDS[2:0]. For details about the setting range, see *Table 29-14 "Data Size Ranges Settable with the SDS and DDS Bits"*.

DDS [2:0]	Transfer Size	Remarks
000	8 bits	(Default)
001	16 bits	
010	32 bits	
011	Setting prohibited	
100	128 bits	
101	256 bits	
110	512 bits	
111	Setting prohibited	

Table 29-13 Channel Configuration Register n (SCHCFGn) Contents (3/4)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Description</th></tr>
<tr><td>15:12</td><td>SDS[3:0]</td><td>Source data size
These bits specify the DMA transfer size.
The normal mode and skip mode are switched with SDS3.
0: Normal mode (Default)
1: Skip mode

The transfer size is set with SDS[2:0]. For details about the setting range, see Table 29-14 “Data Size Ranges Settable with the SDS and DDS Bits”.

<table>
<tr><th>SDS [2:0]</th><th>Transfer Size</th><th>Remarks</th></tr>
<tr><td>000</td><td>8 bits</td><td>(Default)</td></tr>
<tr><td>001</td><td>16 bits</td><td></td></tr>
<tr><td>010</td><td>32 bits</td><td></td></tr>
<tr><td>011</td><td>Setting prohibited</td><td></td></tr>
<tr><td>100</td><td>128 bits</td><td></td></tr>
<tr><td>101</td><td>256 bits</td><td></td></tr>
<tr><td>110</td><td>512 bits</td><td></td></tr>
<tr><td>111</td><td>Setting prohibited</td><td></td></tr>
</table>
</td></tr>
<tr><td>11</td><td>DRRP</td><td>Descriptor read repeat
This bit switches the operation when the header's LV = 0 while the descriptor is read. (For details, see (a) “Link mode operation flow” on page 1864.)
0: Set the SCHSTATn.DER bit to stop the operation. (Default)
1: Continue to read the same descriptor until LV becomes 1, and when LV becomes 1, starts DMA transfer using this descriptor value. The descriptor read interval is controlled by the SDSCITVL register.</td></tr>
<tr><td>10:8</td><td>AM[2:0]</td><td>ACK mode
These bits specify the $\overline{\text{S\_DMAACn}}$ output mode.

<table>
<tr><th>AM[2:0]</th><th>$\overline{\text{S\_DMAACn}}$ Output Mode</th><th>Remarks</th></tr>
<tr><td>000</td><td>Pulse mode (active during 1 clock)</td><td>(Default)</td></tr>
<tr><td>001</td><td>Level mode (active until the selected $\overline{\text{S\_DMARQn}}$ input becomes inactive)</td><td></td></tr>
<tr><td>01x</td><td>Bus cycle mode
(active while the DMA transfer is in a bus cycle)</td><td></td></tr>
<tr><td>1xx</td><td>Do not output $\overline{\text{S\_DMAACn}}$.</td><td></td></tr>
</table>
</td></tr>
<tr><td>6</td><td>LVL</td><td>Level
This bit selects whether to detect DMA requests with the signal level or the edge.
0: Detect a DMA request by a signal rising/falling edge. (Default)
1: Detect a DMA request by the signal level.</td></tr>
</table>

Table 29-13 Channel Configuration Register n (SCHCFGn) Contents (4/4)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Description</th></tr>
<tr><td>5</td><td>LOEN</td><td>Low enable
This bit selects whether to detect DMA requests with the signal's low level or falling edge.

When LVL = 0:
1: Recognize a request if the signal falls.
0: Do not recognize a request even if the signal falls. (Default)

When LVL = 1:
1: Recognize a request if the signal is low level.
0: Do not recognize a request even if the signal is low level. (Default)</td></tr>
<tr><td>4</td><td>HIEN</td><td>High enable
This bit selects whether to detect DMA requests with the signal's high level or rising edge.

When LVL = 0:
1: Recognize a request if the signal rises.
0: Do not recognize a request even if the signal falls. (Default)

When LVL = 1:
1: Recognize a request if the signal is high level.
0: Do not recognize a request even if the signal is high level. (Default)</td></tr>
<tr><td>3</td><td>REQD</td><td>Request direction
This bit selects whether the $\overline{\text{S\_DMARQm}}$ signal selected for the SEL[1:0] bits is source side or destination side. The timing at which $\overline{\text{S\_DMAACn}}$ becomes active can also be selected with this bit.

0: Specify the source side. $\overline{\text{S\_DMAACn}}$ becomes active during read. (Default)
1: Specify the destination side. $\overline{\text{S\_DMAACn}}$ becomes active during write.</td></tr>
<tr><td>1:0</td><td>SEL[1:0]</td><td>Terminal select
These bits select one among the four $\overline{\text{S\_DMARQm}}$, $\overline{\text{S\_DMAACn}}$, and $\overline{\text{S\_DMATCn}}$ signals.

<table>
<tr><th>SEL [1:0]</th><th>Selected Signal</th><th>Remarks</th></tr>
<tr><td>000</td><td>$\overline{\text{S\_DMARQ0}}$, $\overline{\text{S\_DMAAC0}}$, $\overline{\text{S\_DMATC0}}$</td><td>(Default)</td></tr>
<tr><td>001</td><td>$\overline{\text{S\_DMARQ1}}$, $\overline{\text{S\_DMAAC1}}$, $\overline{\text{S\_DMATC1}}$</td><td></td></tr>
<tr><td>010</td><td>$\overline{\text{S\_DMARQ2}}$, $\overline{\text{S\_DMAAC2}}$, $\overline{\text{S\_DMATC2}}$</td><td></td></tr>
<tr><td>011</td><td>$\overline{\text{S\_DMARQ3}}$, $\overline{\text{S\_DMAAC3}}$, $\overline{\text{S\_DMATC3}}$</td><td></td></tr>
<tr><td>Other than above</td><td>Setting prohibited</td><td></td></tr>
</table>
</td></tr>
</table>

Caution If a macro with a different clock is the DMA transfer target and $\overline{\text{S\_DMAACn}}$ is required, $\overline{\text{S\_DMAACn}}$ may not be correctly received owing to synchronous clock issues. In this case, set AM[2:0] to either 001 or 010 in order to use a mode in which $\overline{\text{S\_DMAACn}}$ is active for a longer time.

The settable data size range specified for SDS and DDS depends on the data bus width and the number of buffer stages that is implemented.

The settable ranges are listed below. (These are the ranges in which the macro functions correctly. Set the actual values appropriately according to the access destination macro.)

Table 29-14 Data Size Ranges Settable with the SDS and DDS Bits

Data Bus Width (Bits)	Number of Buffer Stages	Transfer Address	Data Size Range Settable with SDS[2:0] and DDS[2:0]
32	16	Beat aligned	8 or 32 bits (000 to 010) 128 to 512 bits (100 to 1010)
	16	Beat unaligned	8 or 32 bits (000 to 010) 128 bits (100)

(4) SCHITVL0 to SCHITVL7: Channel interval registers 0 to 7

The SCHITVLn register specifies the transfer interval of DMA channel n (n = 0 to 7).

For details, see *29.5.10 “Interval count function” on page 1893*.

Access This register can be read or written in 32-bit units.

Address SCHITVL0: $F9900430_H$, SCHITVL1: $F9900470_H$, SCHITVL2: $F99004B0_H$, SCHITVL3: $F99004F0_H$, SCHITVL4: $F9900530_H$, SCHITVL5: $F9900570_H$, SCHITVL6: $F99005B0_H$, SCHITVL7: $F99005F0_H$

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8
ITVL15	ITVL14	ITVL13	ITVL12	ITVL11	ITVL10	ITVL09	ITVL08
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
ITVL7	ITVL6	ITVL5	ITVL4	ITVL3	ITVL2	ITVL1	ITVL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 29-15 Channel Interval Register n (SCHITVLn) Contents

Bit Position	Bit Name	Description
16:0	ITVL[15:0]	These bits specify the channel transfer interval.

(5) SCHEXT0 to SCHEXT7: Channel extension registers 0 to 7

The SCHEXTn register is the expansion register for DMA channel n (n = 0 to 7).

It is used to specify the bus protocol dependent information.

Access This register can be read or written in 32-bit units.

Address SCHEXT0: $F9900434_H$, SCHEXT1: $F9900474_H$, SCHEXT2: $F99004B4_H$, SCHEXT3: $F99004F4_H$, SCHEXT4: $F9900534_H$, SCHEXT5: $F9900574_H$, SCHEXT6: $F99005B4_H$, SCHEXT7: $F99005F4_H$

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
23	22	21	20	19	18	17	16
0	0	0	0	GPO3	GPO2	GPO1	GPO0
R	R	R	R	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
0	0	0	0	DPR3	DPR2	DPR1	DPR0
R	R	R	R	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
0	0	0	0	SPR3	SPR2	SPR1	SPR0
R	R	R	R	R/W	R/W	R/W	R/W

Table 29-16 Channel Extension Register n (SCHEXTn) Contents

Bit Position	Bit Name	Description
19:16	GPO[7:0]	GPO The value specified for these bits is output from the GPOCn (n = 0 to 7 (channel number)) pins.
11:8	DPR[7:0]	Destination PROT These bits specify the value to be output to MHPROT[3:0] for DMA write transfer. The default value is 0H.
3:0	SPR[7:0]	Source PROT These bits specify the value to be output to MHPROT[3:0] for DMA read transfer. The default value is 0H.

29.3.4 Link register set

This register set indicates the link destination in the link mode.

(1) SNXLA0 to SNXLA7: Next link address registers 0 to 7

The SNXLAn register holds the link address of DMA channel n (n = 0 to 7).

For details about the link mode, see *(2) "Link mode" on page 1863*.

Access This register can be read or written in 32-bit units.

Address SNXLA0: $F9900438_H$, SNXLA1: $F9900478_H$, SNXLA2: $F99004B8_H$, SNXLA3: $F99004F8_H$, SNXLA4: $F9900538_H$, SNXLA5: $F9900578_H$, SNXLA6: $F99005B8_H$, SNXLA7: $F99005F8_H$

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
NXLA31	NXLA30	NXLA29	NXLA28	NXLA27	NXLA26	NXLA25	NXLA24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
NXLA23	NXLA22	NXLA21	NXLA20	NXLA19	NXLA18	NXLA17	NXLA16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
NXLA15	NXLA14	NXLA13	NXLA12	NXLA11	NXLA10	NXLA9	NXLA8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
NXLA7	NXLA6	NXLA5	NXLA4	NXLA3	NXLA2	NXLA1	NXLA0
R/W	R/W	R/W	R/W	R/W	R/W	R	R

Table 29-17 Next Link Address Register n (SNXLAn) Contents

Bit Position	Bit Name	Description
31:0	NXLZ[31:0]	These bits specify the address of the link destination. The lower two bits are masked with 0s. Only word-aligned addresses can be specified.

(2) SCRLA0 to SCRLA7: Current link address registers 0 to 7

The SCRLAn register holds the link address of DMA channel n (n = 0 to 7).

For details about the link mode, see *(2) "Link mode" on page 1863*.

Access This register can be read or written in 32-bit units.

Address SCRLA0: F990043C$_H$, SCRLA1: F990047C$_H$, SCRLA2: F99004BC$_H$, SCRLA3: F99004FC$_H$, SCRLA4: F990053C$_H$, SCRLA5: F990057C$_H$, SCRLA6: F99005BC$_H$, SCRLA7: F99005FC$_H$

Initial value 0000 0000$_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
CRLA31	CRLA30	CRLA29	CRLA28	CRLA27	CRLA26	CRLA25	CRLA24
R	R	R	R	R	R	R	R
23	22	21	20	19	18	17	16
CRLA23	CRLA22	CRLA21	CRLA20	CRLA19	CRLA18	CRLA17	CRLA16
R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8
CRLA15	CRLA14	CRLA13	CRLA12	CRLA11	CRLA10	CRLA9	CRLA8
R	R	R	R	R	R	R	R
7	6	5	4	3	2	1	0
CRLA7	CRLA6	CRLA5	CRLA4	CRLA3	CRLA2	CRLA1	CRLA0
R	R	R	R	R	R	R	R

Table 29-18 Current Link Address Register n (SCRLAn) Contents

Bit Position	Bit Name	Description
31:0	CRLA[31:0]	These bits indicate the address of the descriptor currently being executed.

(3) SSCNT0 to SSCNT7: Source continuous registers 0 to 7

The SSCNTn register specifies the size of the space to be continuously accessed during source address access of DMA channel n (n = 0 to 7).

The SSCNTn register is used in combination with the SSSKPn register. (For details, see *Figure 29-2 "Relationship between SSSKP and SSCNT"*.)

To use the skip mode, set the SCHCFGn.SDS[3] bit.

When performing skip transfer on the source side, do not specify SAD = 1 (fixed). Also, do not perform skip transfer with SSCNTn = 0.

Access This register can be read or written in 32-bit units.

Address SSCNT0: $F9900600_H$, SSCNT1: $F9900620_H$, SSCNT2: $F9900640_H$, SSCNT3: $F9900660_H$, SSCNT4: $F9900680_H$, SSCNT5: $F99006A0_H$, SSCNT6: $F99006C0_H$, SSCNT7: $F99006E0_H$

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
SCNT31	SCNT30	SCNT29	SCNT28	SCNT27 S	CNT26 S	CNT25	SCNT24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
SCNT23	SCNT22	SCNT21	SCNT20	SCNT19	SCNT18	SCNT17	SCNT16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
SCNT15	SCNT14	SCNT13	SCNT12	SCNT11	SCNT10	SCNT9	SCNT8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
SCNT7	SCNT6	SCNT5	SCNT4	SCNT3	SCNT2	SCNT1	SCNT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 29-19 Source Continuous Register n (SSCNTn) Contents

Bit Position	Bit Name	Description
31:0	SCNT[31:0]	Source continuous These bits specify the size of the continuous access space during source address access (unit: byte).

(4) SSSKP0 to SSSKP7: Source skip registers 0 to 7

During source address access of DMA channel n, the next source address is skipped by the amount specified for the SSSKPn register, following access using the data size specified for the SSCNTn register (n = 0 to 7).

The SSSKPn register is used in combination with the SSCNTn register. (For details, see *Figure 29-2 "Relationship between SSSKP and SSCNT"*.)

To use the skip mode, set the SCHCFGn.SDS[3] bit.

When performing skip transfer on the source side, do not specify SAD = 1 (fixed).

Access This register can be read or written in 32-bit units.

Address SSSKP0: $F9900604_H$, SSSKP1: $F9900624_H$, SSSKP2: $F9900644_H$, SSSKP3: $F9900664_H$, SSSKP4: $F9900684_H$, SSSKP5: $F99006A4_H$, SSSKP6: $F99006C4_H$, SSSKP7: $F99006E4_H$

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
SSKP31	SSKP30	SSKP29	SSKP28	SSKP27	SSKP26	SSKP25	SSKP24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
SSKP23	SSKP22	SSKP21	SSKP20	SSKP19	SSKP18	SSKP17	SSKP16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
SSKP15	SSKP14	SSKP13	SSKP12	SSKP11	SSKP10	SSKP9	SSKP8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
SSKP7	SSKP6	SSKP5	SSKP4	SSKP3	SSKP2	SSKP1	SSKP0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 29-20 Source Skip Register n (SSSKPn) Contents

Bit Position	Bit Name	Description
31:0	SSKP[31:0]	Source skip These bits specify the amount of skip during source address access (unit: byte).

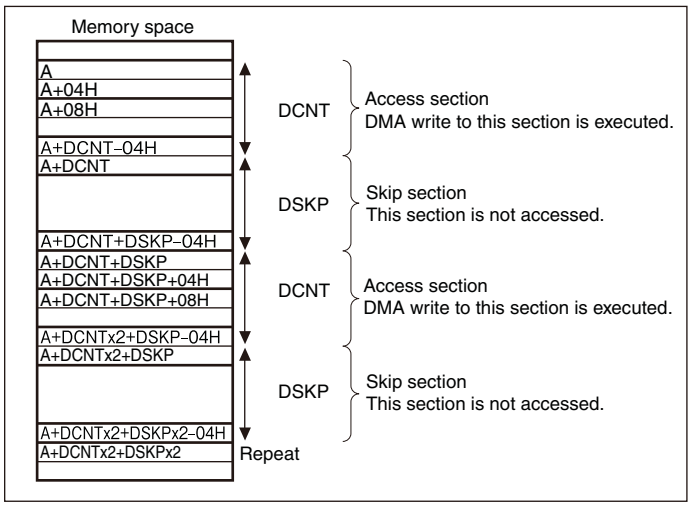


Figure 29-2 Relationship between SSSKP and SSCNT

The values of SSCNT and SSSKP can be specified independently of the setting values of the source address and SCHCFGn.SDS (Source Data Size) bits. The DMAC performs access in the data size specified for the SDS bits, and loads only valid data to the buffer.

(5) SDCNT0 to SSCNT7: Destination continuous registers 0 to 7

The SDCNTn register specifies the size of the space to be continuously accessed during destination address access of DMA channel n (n = 0 to 7).

The SDCNTn register is used in combination with the SDSKPn register. (For details, see *Figure 29-3 "Relationship between SDSKP and SDCNT"*.)

To use the skip mode, set the SCHCFGn.DDS[3] bit.

When performing skip transfer on the destination side, do not specify DAD = 1 (fixed).

Also, do not perform skip transfer with SDCNTn = 0.

Access This register can be read or written in 32-bit units.

Address SDCNT0: $F9900608_H$, SDCNT1: $F9900628_H$, SDCNT2: $F9900648_H$, SDCNT3: $F9900668_H$, SDCNT4: $F9900688_H$, SDCNT5: $F99006A8_H$, SDCNT6: $F99006C8_H$, SDCNT7: $F99006E8_H$

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
DCNT31	DCNT30	DCNT29	DCNT28	DCNT27	DCNT26	DCNT25	DCNT24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
DCNT23	DCNT22	DCNT21	DCNT20	DCNT19	DCNT18	DCNT17	DCNT16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
DCNT15	DCNT14	DCNT13	DCNT12	DCNT11	DCNT10	DCNT9	DCNT8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
DCNT7	DCNT6	DCNT5	DCNT4	DCNT3	DCNT2	DCNT1	DCNT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 29-21 Destination Continuous Register n (SDCNTn) Contents

Bit Position	Bit Name	Description
31:0	DCNT[31:0]	Destination continuous These bits specify the size of the continuous access space during destination address access (unit: byte).

(6) SDSKP0 to SDSKP7: Destination skip registers 0 to 7

After accessing the destination of DMA channel n using the data size specified for the SDCNTn register, the next destination address is skipped by the amount specified for the SDSKPn register (n = 0 to 7).

The SDSKPn register is used in combination with the SDCNTn register. (For details, see *Figure 29-3 "Relationship between SDSKP and SDCNT"*.)

To use the skip mode, set the SCHCFGn.DDS[3] bit.

When performing skip transfer on the destination side, do not specify DAD = 1 (fixed).

Access This register can be read or written in 32-bit units.

Address SDSKP0: $F990060C_H$, SDSKP1: $F990062C_H$, SDSKP2: $F990064C_H$, SDSKP3: $F990066C_H$, SDSKP4: $F990068C_H$, SDSKP5: $F99006AC_H$, SDSKP6: $F99006CC_H$, SDSKP7: $F99006EC_H$

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
DSKP31	DSKP30	DSKP29	DSKP28	DSKP27	DSKP26	DSKP25	DSKP24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
DSKP23	DSKP22	DSKP21	DSKP20	DSKP19	DSKP18	DSKP17	DSKP16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
DSKP15	DSKP14	DSKP13	DSKP12	DSKP11	DSKP10	DSKP9	DSKP8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
DSKP7	DSKP6	DSKP5	DSKP4	DSKP3	DSKP2	DSKP1	DSKP0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 29-22 Destination Skip Register n (SDSKPn) Contents

Bit Position	Bit Name	Description
31:0	DSKP[31:0]	Destination skip These bits specify the amount of skip during destination address access (unit: byte).

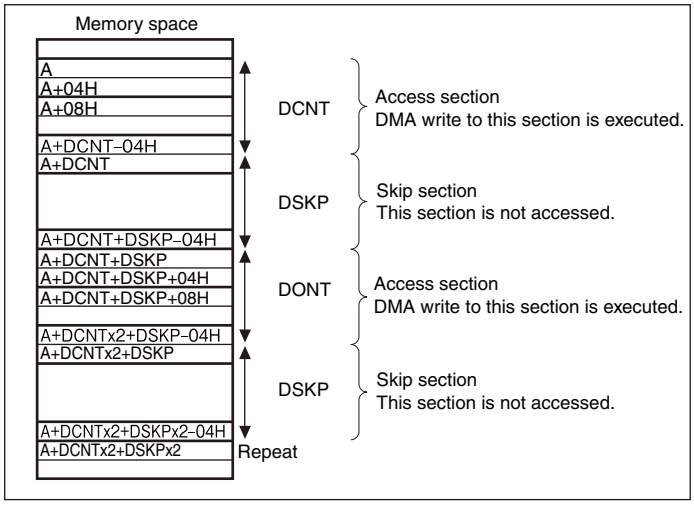


Figure 29-3 Relationship between SDSKP and SDCNT

The values of SDCNT and SDSKP can be specified independently of the setting values of the destination address and SCHCFGn.DDS (Destination Data Size) bits. The DMAC performs write access to the specified space only in data size equal to or less than that specified for the DDS bits, based on the setting combination.

29.3.5 DMA register set

The registers described in this section are common to all the channels.

(1) SDCTRL: DMA control register

The SDCTRL register specifies the transfer type during descriptor access and the arbitration among channels.

Access This register can be read or written in 32-bit units.

Address $F9900700_H$

Initial value $001F0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
0	0	0	0	LWPR3	LWPR2	LWPR1	LWPR0
R	R	R	R	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
0	0	0	0	LDPR3	LDPR2	LDPR1	LDPR0
R	R	R	R	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
7	6	5	4	3	2	1	0
0	0	0	0	0	0	LVINT	PR
R	R	R	R	R	R	R/W	R/W

Table 29-23 DMA Control Register (SDCTRL) Contents

Bit Position	Bit Name	Description
27:24	LWPR[3:0]	Link writeback plot These bits specify the value to be output to MHPROT[3:0] while data is written back to the descriptor in the link mode.
19-16	LDPR[3:0]	Link descriptor plot These bits specify the value to be output to MHPROT[3:0] while data is loaded from the descriptor in the link mode.
1	LVINT	This bit specifies pulse output or level output for INTHDMA[7:0] and INTHDMAERR. 0: Pulse output 1: Level output
0	PR	This bit specifies the transfer priority control mode. (For details, see *29.5.2 "Priority order control of DMA channels" on page 1878*.) 0: Fixed priority mode 1: Round-robin mode

(2) SDSCITVL: Descriptor interval register

When the SCHCFGn.DRRP bit is set, the descriptor is continuously read until LV = 1. This register sets the read interval used at this time.

Access This register can be read or written in 32-bit units.

Address $F9900704_H$

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8
DIVTL7	DIVTL6	DIVTL5	DIVTL4	DIVTL3	DIVTL2	DIVTL1	DIVTL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R

Table 29-24 Descriptor Interval Register (SDSCITVL) Contents

Bit Position	Bit Name	Description
15-8	DIVTL[7:0]	Descriptor interval These bits specify the descriptor read interval. The descriptor is repeatedly read at intervals of [DITVL × 256] cycles.

(3) SDSTAT_EN: DMA status EN register

The SDSTAT_EN register indicates the EN bit states of all DMA channels.

The value of each bit remains unchanged even if this register is written to.

Access This register can be read or written in 32-bit units.

Address $F9900710_H$

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
7	6	5	4	3	2	1	0
EN7	EN6	EN5	EN4	EN3	EN2	EN1	EN0
R	R	R	R	R	R	R	R

Table 29-25 DMA Status EN Register (SDSTAT_EN) Contents

Bit Position	Bit Name	Description
7	EN7	This bit indicates the EN bit state of DMA channel 7.
6	EN6	This bit indicates the EN bit state of DMA channel 6.
5	EN5	This bit indicates the EN bit state of DMA channel 5.
4	EN4	This bit indicates the EN bit state of DMA channel 4.
3	EN3	This bit indicates the EN bit state of DMA channel 3.
2	EN2	This bit indicates the EN bit state of DMA channel 2.
1	EN1	This bit indicates the EN bit state of DMA channel 1.
0	EN0	This bit indicates the EN bit state of DMA channel 0.

Caution The EN bits for the channels that do not exist in the channel configuration are 0.

(4) SDSTAT_ER: DMA status ER register

The SDSTAT_ER register indicates the ER bit states of all DMA channels.

The value of each bit remains unchanged even if this register is written to.

Access This register can be read or written in 32-bit units.

Address $F9900714_H$

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
7	6	5	4	3	2	1	0
ER7	ER6	ER5	ER4	ER3	ER2	ER1	ER0
R	R	R	R	R	R	R	R

Table 29-26 DMA Status ER Register (SDSTAT_ER) Contents

Bit Position	Bit Name	Description
7	ER7	This bit indicates the ER bit state of DMA channel 7.
6	ER6	This bit indicates the ER bit state of DMA channel 6.
5	ER5	This bit indicates the ER bit state of DMA channel 5.
4	ER4	This bit indicates the ER bit state of DMA channel 4.
3	ER3	This bit indicates the ER bit state of DMA channel 3.
2	ER2	This bit indicates the ER bit state of DMA channel 2.
1	ER1	This bit indicates the ER bit state of DMA channel 1.
0	ER0	This bit indicates the ER bit state of DMA channel 0.

Caution The ER bits for the channels that do not exist in the channel configuration are 0.

(5) SDSTAT_END: DMA status END register

The SDSTAT_END register indicates the END bit states of all DMA channels.

The value of each bit remains unchanged even if this register is written to.

Access This register can be read or written in 32-bit units.

Address $F9900718_H$

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
7	6	5	4	3	2	1	0
END7	END6	END5	END4	END3	END2	END1	END0
R	R	R	R	R	R	R	R

Table 29-27 DMA Status END Register (SDSTAT_END) Contents

Bit Position	Bit Name	Description
7	END7	This bit indicates the END bit state of DMA channel 7.
6	END6	This bit indicates the END bit state of DMA channel 6.
5	END5	This bit indicates the END bit state of DMA channel 5.
4	END4	This bit indicates the END bit state of DMA channel 4.
3	END3	This bit indicates the END bit state of DMA channel 3.
2	END2	This bit indicates the END bit state of DMA channel 2.
1	END1	This bit indicates the END bit state of DMA channel 1.
0	END0	This bit indicates the END bit state of DMA channel 0.

Caution The END bits for the channels that do not exist in the channel configuration are 0.

(6) SDSTAT_TC: DMA status TC register

The SDSTAT_TC register indicates the TC bit states of all DMA channels.

The value of each bit remains unchanged even if this register is written to.

Access This register can be read or written in 32-bit units.

Address $F990071C_H$

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
7	6	5	4	3	2	1	0
TC7	TC6	TC5	TC4	TC3	TC2	TC1	TC0
R	R	R	R	R	R	R	R

Table 29-28 DMA Status TC Register (SDSTAT_TC) Contents

Bit Position	Bit Name	Description
7	TC7	This bit indicates the TC bit state of DMA channel 7.
6	TC6	This bit indicates the TC bit state of DMA channel 6.
5	TC5	This bit indicates the TC bit state of DMA channel 5.
4	TC4	This bit indicates the TC bit state of DMA channel 4.
3	TC3	This bit indicates the TC bit state of DMA channel 3.
2	TC2	This bit indicates the TC bit state of DMA channel 2.
1	TC1	This bit indicates the TC bit state of DMA channel 1.
0	TC0	This bit indicates the TC bit state of DMA channel 0.

Caution The TC bits for the channels that do not exist in the channel configuration are 0.

(7) SDSTAT_SUS: DMA status SUS register

The SDSTAT_SUS register indicates the SUS bit states of all DMA channels.

The value of each bit remains unchanged even if this register is written to.

Access This register can be read or written in 32-bit units.

Address $F9900720_H$

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
7	6	5	4	3	2	1	0
SUS7	SUS6	SUS5	SUS4	SUS3	SUS2	SUS1	SUS0
R	R	R	R	R	R	R	R

Table 29-29 DMA Status SUS Register (SDSTAT_SUS) Contents

Bit Position	Bit Name	Description
7	SUS7	This bit indicates the SUS bit state of DMA channel 7.
6	SUS6	This bit indicates the SUS bit state of DMA channel 6.
5	SUS5	This bit indicates the SUS bit state of DMA channel 5.
4	SUS4	This bit indicates the SUS bit state of DMA channel 4.
3	SUS3	This bit indicates the SUS bit state of DMA channel 3.
2	SUS2	This bit indicates the SUS bit state of DMA channel 2.
1	SUS1	This bit indicates the SUS bit state of DMA channel 1.
0	SUS0	This bit indicates the SUS bit state of DMA channel 0.

Caution The SUS bits for the channels that do not exist in the channel configuration are 0.

(8) DMA transfer interface control register: DMAIFCm (m = 0 to 3)

The DMAIFCm register specifies the active width of the DMA acknowledge output signals ($\overline{\text{S\_DMAAK[0:3]}}$) and internal mask width of the DMA transfer request input signal ($\overline{\text{S\_DMARQ[0:3]}}$).

Access This register can be read or written in 32-bit units.

Address DMAIFC0: $F9900C00_H$ DMAIFC1: $F9900C04_H$
DMAIFC2: $F9900C08_H$ DMAIFC3: $F9900C0C_H$

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
DIFENm	0	0	0	0	0	0	0
R/W	R	R	R	R	R	R	R
23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8
0	0	0	RQMKm4	RQMKm3	RQMKm2	RQMKm1	RQMKm0
R	R	R	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
SUS7	SUS6	SUS5	AKWDm4	AKWDm3	AKWDm2	AKWDm1	AKWDm0
R	R	R	R/W	R/W	R/W	R/W	R/W

Table 29-30 DMAIFCm Register Contents (1/2)

Bit Position	Bit Name	Description
31	DIFENn	This bit enables or disables the DMA transfer interface signal control function. 0: Disable (default) 1: Enable
12:8	RQMKm[4:0]	These bits specify mask width[a] of the DMA request signal ($\overline{\text{S\_DMARQm}}$).

RQMKm4	RQMKm3	RQMKm2	RQMKm1	RQMKm0	Mask width[a] of $\overline{\text{S\_DMARQm}}$ signal
0	0	0	0	0	No mask
0	0	0	0	1	$1/f_{SDCLK}$
0	0	0	1	0	$2/f_{SDCLK}$
:	:	:	:	:	:
1	1	1	1	0	$30/f_{SDCLK}$
1	1	1	1	1	$31/f_{SDCLK}$

Table 29-30 DMAIFCm Register Contents (2/2)

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Description</th></tr>
<tr><td>4:0</td><td>AKWDm[4:0]</td><td>These bits specify the mask width of the DMA acknowledge signal (S_DMACKm).
<table>
<tr><th>RQMKm4</th><th>RQMKm3</th><th>RQMKm2</th><th>RQMKm1</th><th>RQMKm0</th><th>Active level width[b] of S_DMACKm signal</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>$1/f_{SDCLK}$</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>$2/f_{SDCLK}$</td></tr>
<tr><td>:</td><td>:</td><td>:</td><td>:</td><td>:</td><td>:</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>$30/f_{SDCLK}$</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>$31/f_{SDCLK}$</td></tr>
</table>
</td></tr>
</table>

a) Masking of the S_DMARQm signal starts at the rising edge of the S_DMACKm signal (in other words, when this signal becomes inactive).

b) The active width of the S_DMACKm signal is based on the acknowlege signal specified for SCHCFGn.AMn[2:0]. For details, refer to *29.3.3 (3) "SCHCFG0 to SCHCFG7: Channel configuration registers 0 to 7" on page 1834*.

29.4 Operation

29.4.1 DMA mode

(1) Register mode

The register mode is the mode in which DMA transfer is executed using the values set in internal registers.

The transfer source address, transfer destination address, and number of transfer bytes can be set.

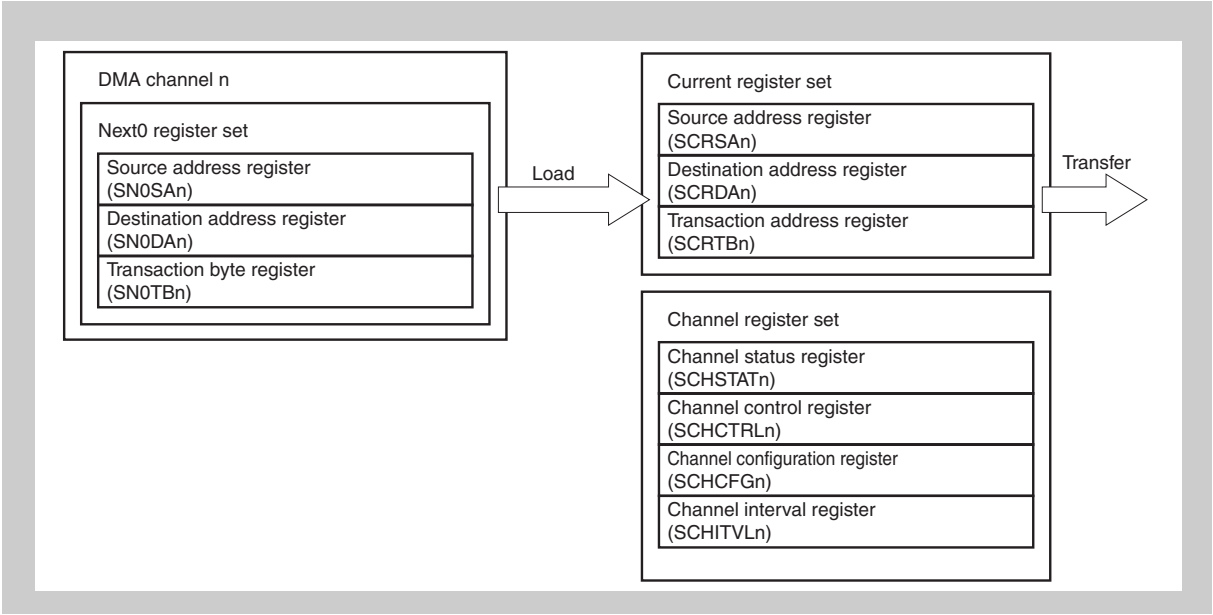


Figure 29-4 Register Mode Operation Schematic

(a) Register Mode Operation Flow

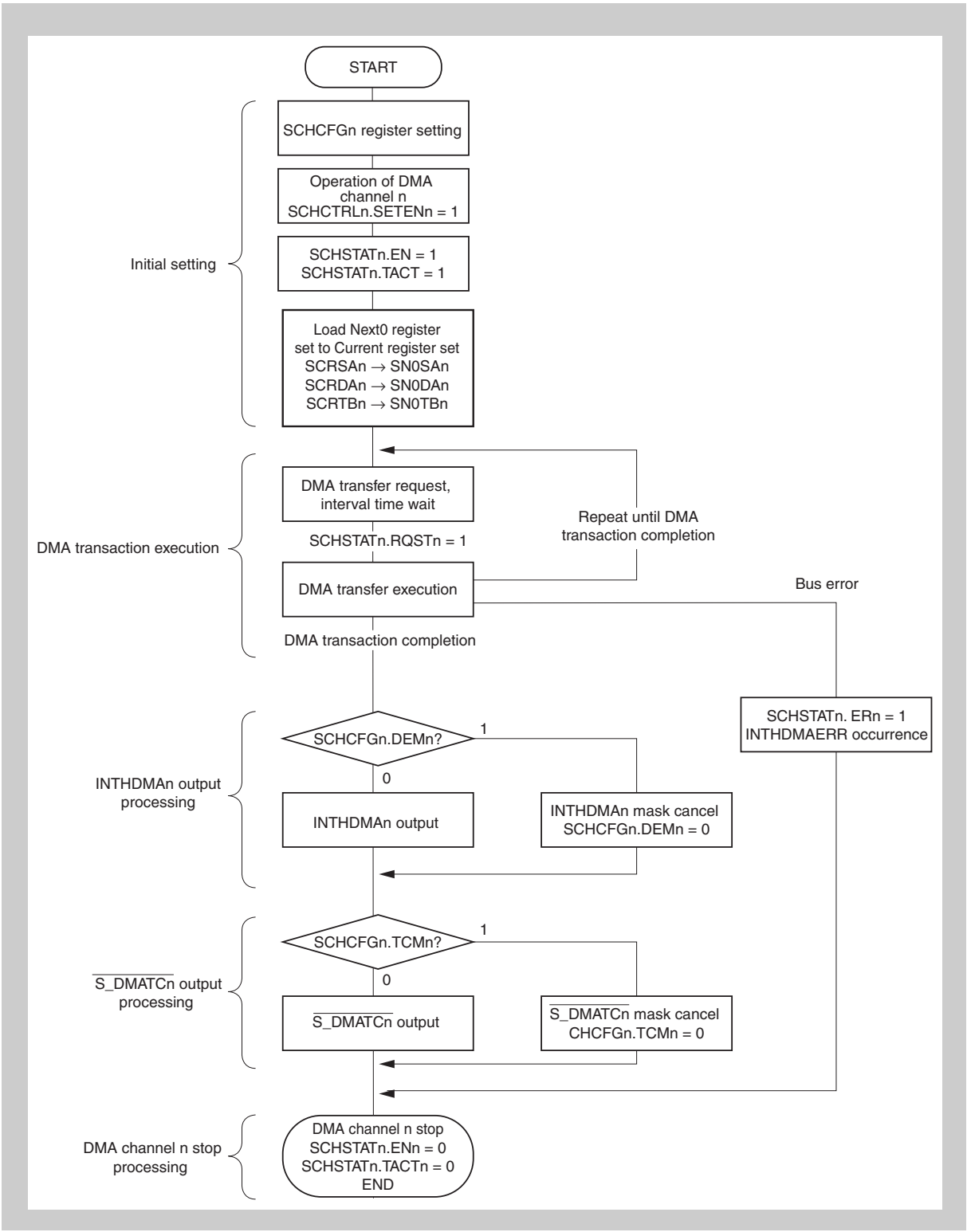


Figure 29-5 Register Mode Operation Flow

Initial setting The transfer destination address, transfer source address, and total number of transfer bytes are specified to the next 0 register set. Also, the S_DMARQm pin, S_DMAAKm pin, and transfer amount are specified to the channel register set. (For details, see *29.5 "DMA Transfer" on page 1875*.)

DMA transaction DMA transactions are performed according to the set values. For details about transfer, see *29.5 "DMA Transfer" on page 1875*.

INTHDMAn output processing The INTHDMAn interrupt is masked according to the SCHCFGn.DEM bit setting. If DEM is set, the INTHDMAn interrupt is masked. Immediately after, DEM is automatically cleared.

S_DMATCn mask The S_DMATCn pin output is masked according to the SCHCFGn.TCM bit setting. If TCM is set, the S_DMATCn pin output is masked. Immediately after, TCM is cleared.

(b) Register settings in the register mode

Link mode selection (SCHCFGn.DMS) The link mode is selected with the SCHCFGn.DMS bit.

The DMS bit cannot be changed using the descriptor.

Table 29-31 Link Mode Selection

SCHCFGn.DMS	Operation
1 (Link mode)	Operation in link mode

Selection of INTHDMAn operation (SCHCFGn.DEM) The INTHDMAn operation upon completion of a DMA transaction (series of DMA transfers) in the register mode can be specified for the SCHCFGn.DEM bit.

Table 29-32 INTHDMAn Operation Selection

SCHCFGn.DEM	Operation	
0	Enable INTHDMAn output. (INTHDMAn is not masked.)	INTHDMAn is output upon completion of a DMA transaction (series of DMA transfers).
1	Disable INTHDMAn output. (INTHDMAn is masked.)	INTHDMAn is not output upon completion of a DMA transaction (series of DMA transfers). The DEM bit is then automatically cleared and INTHDMAn output is enabled.

Mask setting (SCHCFGn.TCM) of terminal count output ($\overline{\text{S\_DMATCm}}$)

Whether to mask the terminal count ($\overline{\text{S\_DMATCm}}$ (m: specified for SCHCFGn.SEL[1:0])) output upon completion of a DMA transaction (series of DMA transfers) in the register mode can be specified for the SCHCFGn.TCM bit (bit 25).

Table 29-33 Terminal Count Output ($\overline{\text{S\_DMATCm}}$) Mask Setting

SCHCFGn.DEM	Operation	
0	Enable terminal count ($\overline{\text{S\_DMATCm}}$) output. ($\overline{\text{SDMATCm}}$ is not masked.)	$\overline{\text{S\_DMATCm}}$ is output upon completion of a DMA transaction (series of DMA transfers).
1	Disable terminal count ($\overline{\text{S\_DMATCm}}$) output. ($\overline{\text{SDMATCm}}$ is masked.)	$\overline{\text{S\_DMATCm}}$ is not output upon completion of a DMA transaction (series of DMA transfers). The TCM bit is then automatically cleared and $\overline{\text{S\_DMATCm}}$ output is enabled.

(c) Example of register settings in the register mode

Table 29-34 Register Setting Example

SCHCFGn.DMS	SCHCFGn.DEM	SCHCFGn.TCMn
0	0	0
Register mode	INTHDMAn not masked	$\overline{\text{S\_DMATCm}}$ not masked

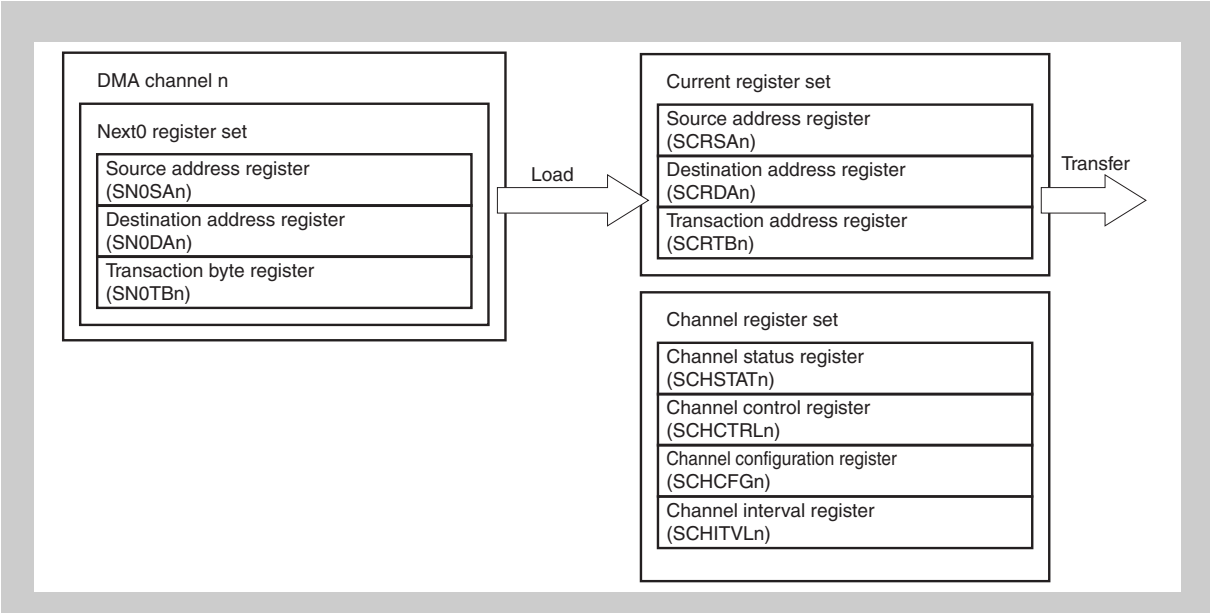


Figure 29-6 Register Mode

<1> When SCHSTATn.EN and SCHSTATn.TACTn are set, the contents of the next 0 register set are loaded to the current register set.

<2> The DMA transaction (series of DMA transfers) is executed according to the values of the current register set and the channel register set.

<3> INTHDMAn is output upon completion of the DMA transaction (series of DMA transfers) because SCHCFGn.DEM is 0.

<4> $\overline{\text{S\_DMATCm}}$ is output upon completion of the DMA transaction (series of DMA transfers) because SCHCFGn.TCM is 0.

(2) Link mode

The link mode is a mode in which the “descriptor” placed in the memory connected internally/externally to the V850E2/MN4 is loaded as the set values and DMA transactions (series of DMA transfers) are executed accordingly.

Inside the DMAC, a next link address register and a current link address register are provided for each channel. The next link address register is used to specify the descriptor address to be executed next. The current link address register is used to indicate the descriptor address of the DMA transaction (series of DMA transfers) currently being executed.

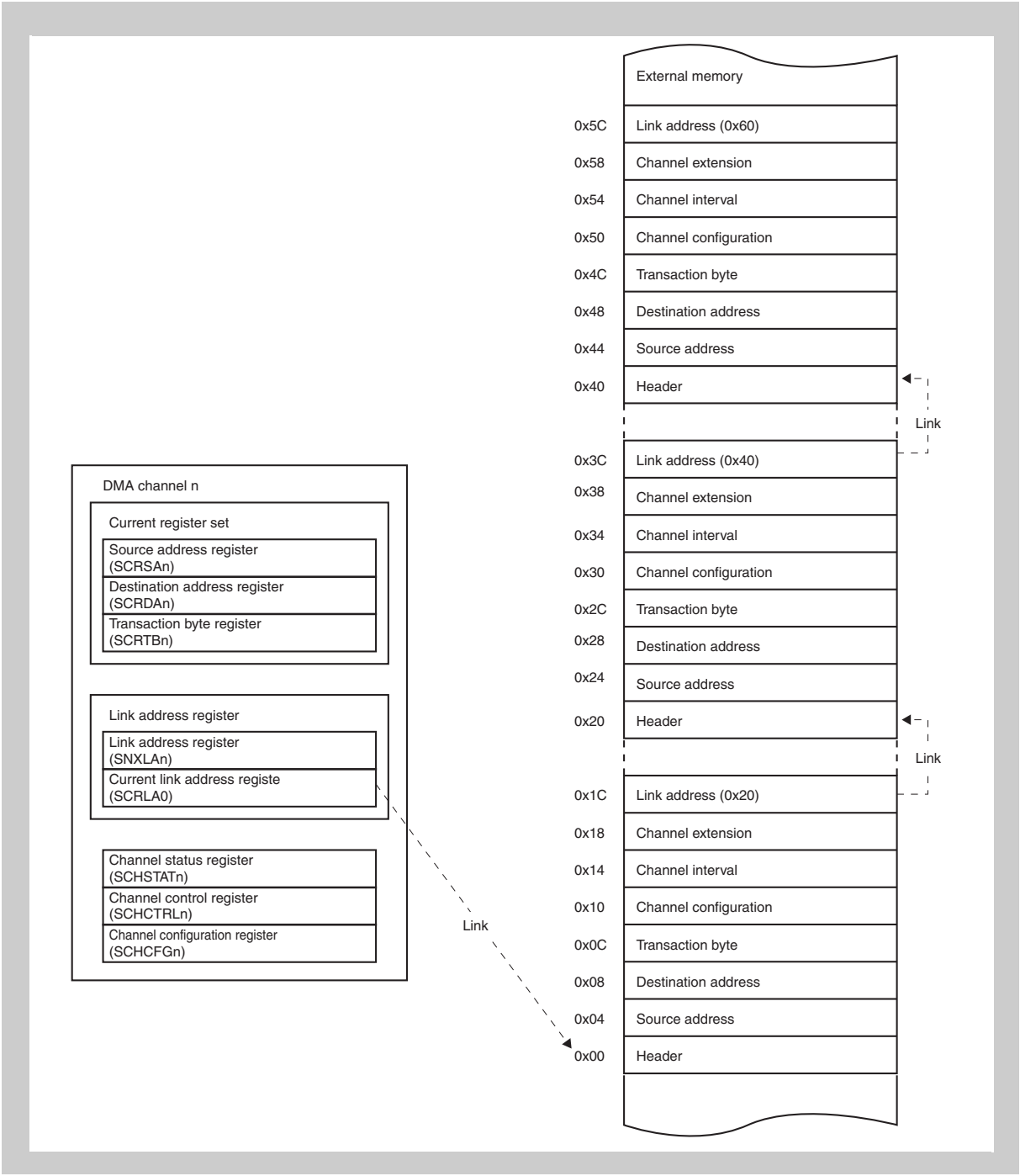


Figure 29-7 Link Mode Schematic

(a) Link mode operation flow

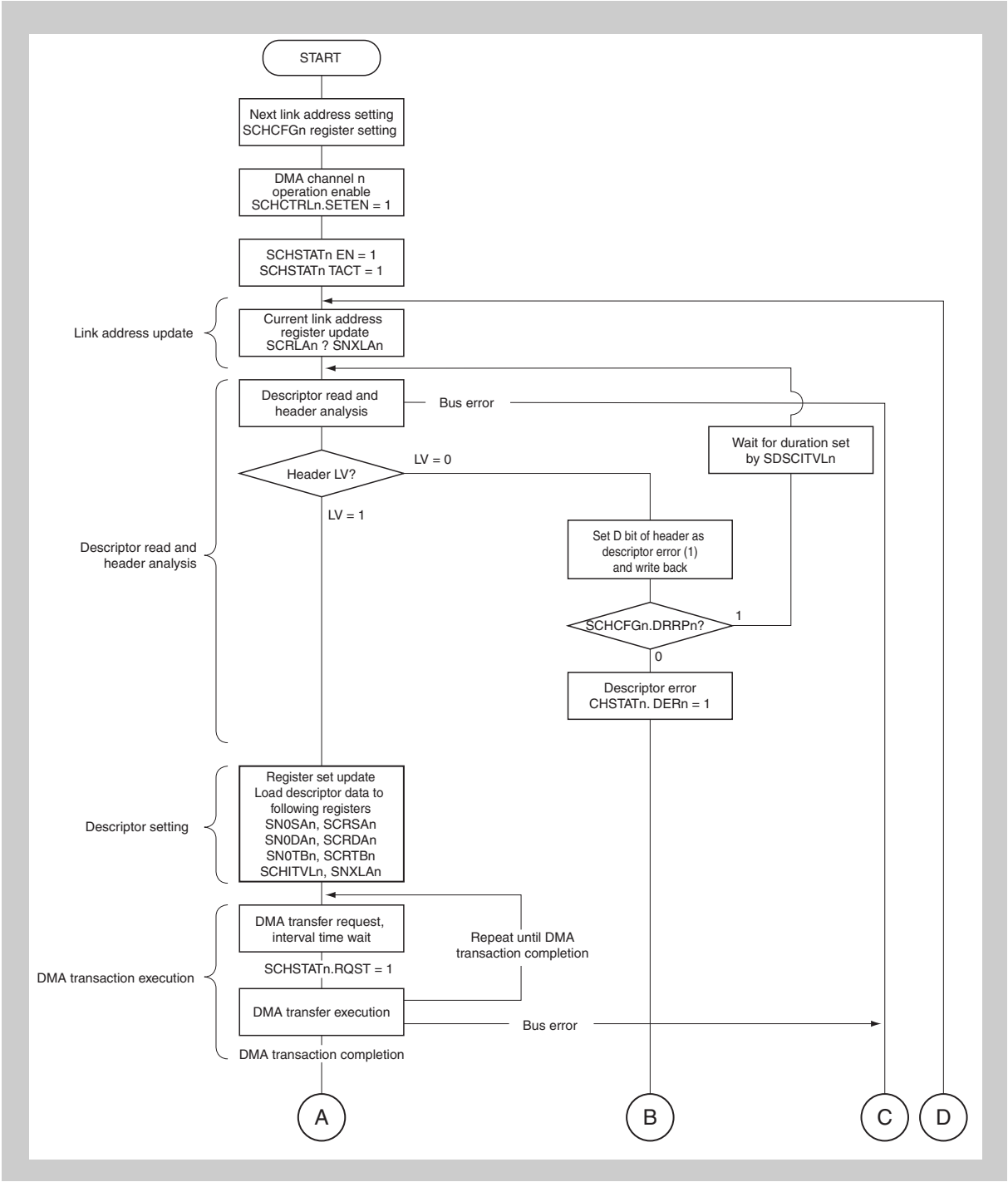


Figure 29-8 Link Mode Operation Flow (1/2)

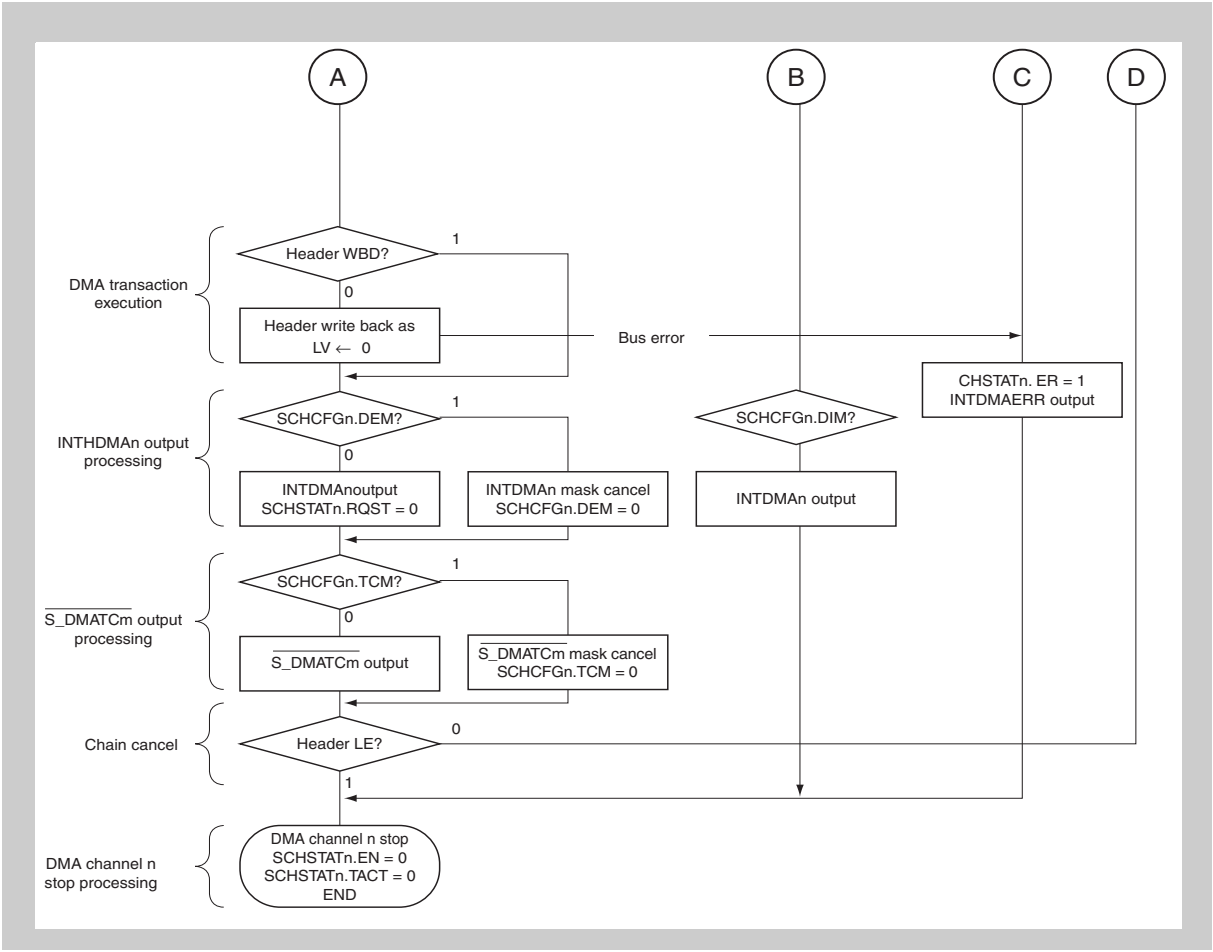


Figure 29-9 Link Mode Operation Flow (2/2)

Channel setting

The start address of the link destination is set to SNXLAn.

Link address update

When the SCHCTRLn.SETEN bit is set, SCHSTATn.EN and SCHSTATn.TACT are set and the link address set to the SNXLAn register is loaded to SCRLAn.

Descriptor read and header judgment

Loading data from the descriptor starts and the DMAC checks the contents of the header. If LV is 0, the D bit of the header is set and written back.

If SCHCFGn.DRRP is 1, the same descriptor is read again after the lapse of the time specified for the SDSCITVL register.

If SCHCFGn.DRRP = 0, the value of SCHSTATn.DER becomes 1 and the state changes to the end state (EN = 0, TACT = 0). If at this time SCHCFGn.DIM is 0, INTHDMAn is output.

Descriptor setting

The data loaded from the descriptor is set to the current register set and the channel register set. Also, the next link destination is set to SNXLAn.

DMA transaction execution

A DMA transactions is executed according to the set value.

If a DMA transfer error occurs during this time, INTHDMAERR is output and DMA transfer ends.

Header write back

If the header's WBD is 0, the DMAC clears the header's LV bit and writes back the header.

INTHDMAn output processing

The INTHDMAn interrupt is masked according to the SCHCFGn.DEM bit setting.

If DEM is 1, INTHDMAn is masked and not output.

S_DMATCm output processing

The S_DMATCm (m: specified for SCHCFGn.SEL[1:0] bits) output is masked according to the SCHCFGn.TCM bit setting. If TCMn is 1, S_DMATCm is masked and not output.

Link end judgment

If the header's LV is 1, upon completion of the DMA transaction with the descriptor setting, EN and TACT are cleared and the processing ends. If LE is 0, the current register set is updated and loading of the next descriptor starts.

(b) Register settings in link mode

Selection of link mode (SCHCFGn.DMS)

The link mode is selected with the SCHCFGn.DMS bit.

The DMS bit cannot be changed using the descriptor.

Table 29-35 Link Mode Selection

SCHCFGn.DMS	Operation
1 (Link mode)	Operation in link mode

Link address setting (SNXLAn)

The registers that indicate the link destination are the next link address register n (SNXLAn) and the current link address register n (SCRLAn). When starting the link mode, set the link destination address to the SNXLAn register. The SNXLAn register indicates the next link destination upon completion of descriptor loading. The SCRLAn register indicates the link address that is currently being executed.

Table 29-36 Link Address Register Set

SCHCFGn.DMS	Operation
Next link address register (SNXLAn)	This register indicates the next link destination. Before starting the link mode, specify the link destination address to this register.
Current link address register (SCRLAn)	This register indicates the link address currently being executed. This register is read-only.

Caution

In the link mode, the settings can be changed through descriptor read, but the timing to change the settings cannot be synchronized with DMA transfer request (S_DMARQm or interrupt signal) via hardware. Therefore, when using DMA transfer requests via hardware, specify SCHCFGn.AM[2:0], LVL, HIEN, LOEN, and SEL[1:0] before setting the EN bit. Do not use the descriptor to change these bits.

Descriptor setting

The DMAC supports two descriptor formats. To switch the formats, specify the DSCFM field of bits 32 to 28 in the first word (header) of the descriptor. The table below shows the correspondence between the DSCFM values and the descriptor formats. The meaning of each symbol used in this table is described in *Table 29-38 "Meaning of Symbols"*.

Table 29-37 Descriptor Formats

DSCFM Field Value	0001_b	0011_b
Descriptor size	8 words	4 words
Link address	○	○
Channel interval	○	– (Reload)
Channel configuration	○	– (Reload)
Transaction size	○	– (Reload)
Destination size	○	○
Source address	○	○
Header	○	○ (STS)

Cautions

1. Do not specify a value other than those listed above to the DSCFM field.
2. The descriptor cannot be used to change the SCHCFGn.DMS bit (the mode is fixed to the link mode).

Table 29-38 Meaning of Symbols

Field	Symbol	Description
Link address	○	Specify the next descriptor address (link address) to be read following the DMA transfer with the current descriptor.
Channel interval and channel configuration	○	Specify the channel interval and channel configuration.
	– (Reload)	Omit the channel interval and channel configuration specification and use the previously set values.
Transaction size	○	Specify the transaction byte size.
	– (Reload)	Omit the transaction byte size specification and use the STS field value in the header as the total number of transfer bytes. The maximum value is 65,536 (bytes) because the STS field is 16 bytes.
Destination address	○	Specify the transfer destination address.
Source address	○	Specify the transfer source address.
Header	○ (no STS)	The STS field, bits 15 to 0 of the header, is invalid. The transaction size of the descriptor is used as the total number of transfer bytes.
	○ (STS)	The STS field, bits 15 to 0 of the header, is valid. The value set in the STS field is used as the total number of transfer bytes.

Header setting The header indicates the state of the descriptor, etc. The header is read before the start of DMA transfer in the link mode. It is written back upon completion of a DMA transaction (series of DMA transfers).

Caution To sequentially add descriptors during the DMAC operation, set the LV bit through byte access write. Because the DMAC writes back the D bit through byte access write, conflict between setting the LV bit using software and writeback of the D bit by the DMAC can be avoided through this operation.

31	30	29	28	27	26	25	24
DSCFM				0	WBD	LE	LV

If WBD is 0 during header read, only the LV bit is cleared and written back upon completion of the DMA transaction (series of DMA transfers) according to this descriptor.

23	22	21	20	19	18	17	16
D	0	0	0	0	0	0	0

If LV is 0 during header read, only the D bit is set and written back, as a descriptor error.

15	14	13	12	11	10	9	8
STS							

7	6	5	4	3	2	1	0
STS							

Table 29-39 Header Contents (1/2)

Bit Position	Bit Name	Description
31:28	DSCFM	Descriptor Format These bits specify one of the two descriptor formats shown in *Table 29-37 "Descriptor Formats"*.
26	WBD	Write Back Disable This bit specifies the LV bit writeback operation. 0: Write back 0 to the LV bit upon completion of DMA transaction (series of DMA transfers). 1: Do not write back 0 to the LV bit upon completion of DMA transaction (series of DMA transfers).
25	LE	Link End This bit indicates the link continuation status of the DMA transaction (series of DMA transfers) of this descriptor. Set this bit at the end of the link. 0: Link continuation 1: Link end
24	LV	Link Valid This bit indicates whether the descriptor is valid or invalid. If WBD is 0, the LV bit is cleared and written back upon completion of the DMA transaction (series of DMA transfers) according to the descriptor. Set this bit for the header setting. 0: The descriptor is invalid. 1: The descriptor is valid.

Table 29-39 Header Contents (2/2)

Bit Position	Bit Name	Description
23	D	Descriptor Error This bit indicates a descriptor error. If LV is 0 (descriptor invalid) when the header is read, the DMAC sets this bit and performs writeback. 0: No descriptor error has occurred. 1: A descriptor error has occurred.
15:0	STS	Short Transaction Size If 0011_b is set to the DSCFM field, these bits set the total number of DMA transfer bytes. The maximum value is 65,536 bytes. At this time, 0 cannot be set for STS.

Caution To sequentially add descriptors during the DMAC operation, set the LV bit through byte access write. Because the DMAC writes back the D bit through byte access write, conflict between setting the LV bit using software and writeback of the D bit by the DMAC can be avoided through this operation.

Descriptor setting other than header

The descriptor data fields other than the header have the same specifications as the DMAC internal registers.

Table 29-40 "Correspondence Between Data Fields of Descriptor and DMAC Internal Registers" shows the correspondence between the data fields of the descriptor and the DMAC internal registers.

Table 29-40 Correspondence Between Data Fields of Descriptor and DMAC Internal Registers

Offset Address of Descriptor	Data Fields of Descriptor	DMAC Internal Registers
$+04_H$	Source address	Source address register (SCRSAn)
$+08_H$	Destination address	Destination address register (SCRDAn)
$+0C_H$	Transaction byte	Transaction byte register (SCRTBn)
$+10_H$	Channel configuration	Channel configuration register (SCHCFn)
$+14_H$	Channel interval	Channel interval register (SCHITVLn)
$+18_H$	Be sure to specify $0000\ 0000_H$.	–

Caution The descriptor cannot be used to change the SCHCFGn.DMS bit.

(c) Outline of descriptor areas and DMA transfer areas

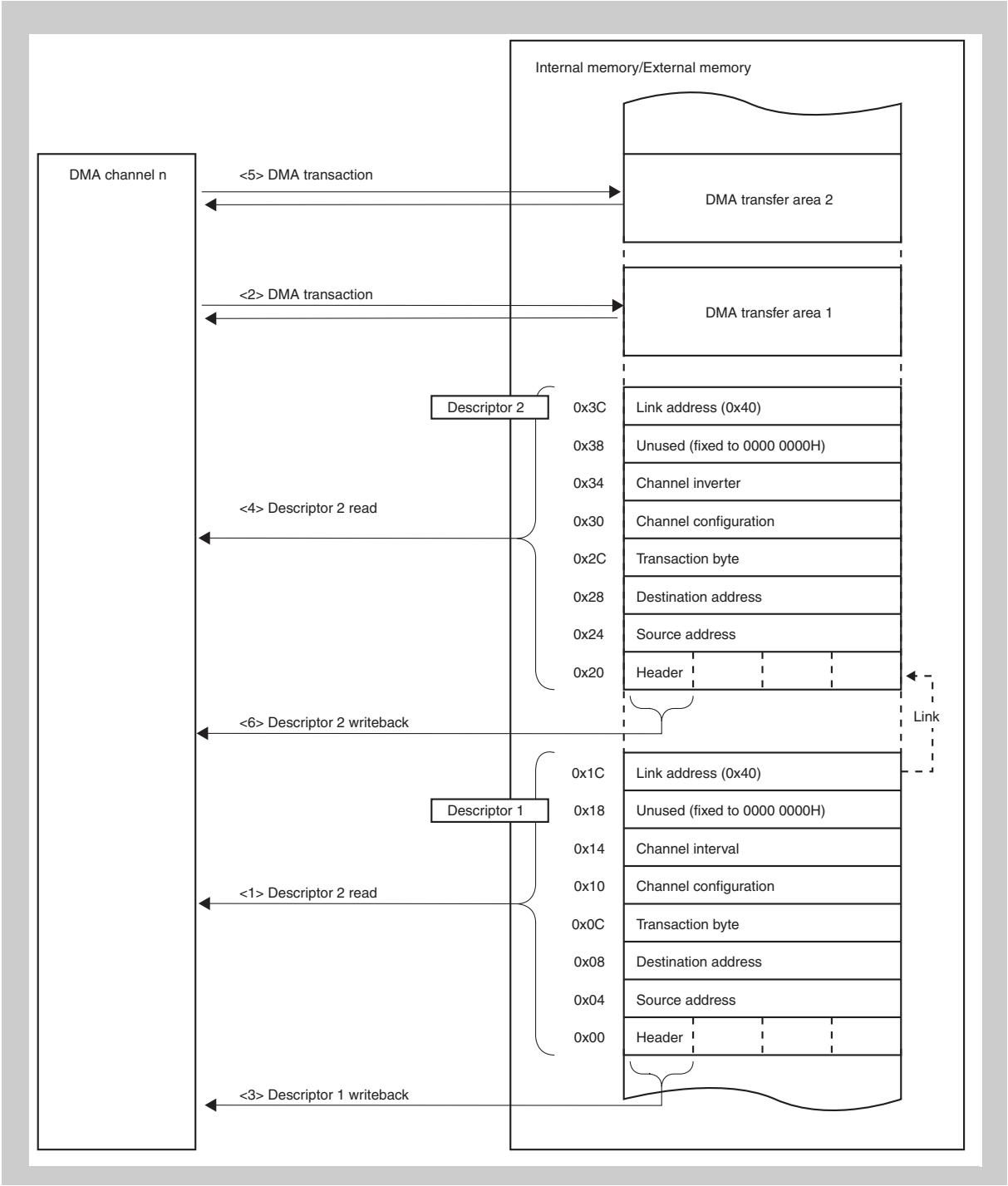


Figure 29-10 Outline of Descriptor Areas and DMA Transfer Areas

<1> Descriptor read

The value set to the next link address register (SNXLAn) in the DMAC is loaded to the current link address register (SCRLAn) and the descriptor is read from "Descriptor 1" on the memory space indicated by the SCRLAn register.

<2> DMA transfer (DMA transaction)

If LV in the header of the descriptor is 1, DMA transfer is executed according to the descriptor information.

<3> Descriptor writeback

If WBD in the header is 0 after completion of DMA transaction for the set number of bytes, data is written back to bits 31 to 24 in the header of descriptor 1 with LV = 0. For other data fields, the value read in step <1> is written back through byte write.

<4> Descriptor read

If LE in the header of the descriptor read in step <1> is 0, the next descriptor is read from the address (descriptor 2) indicated by the next link address in the descriptor.

<5> DMA transfer (DMA transaction)

If LV in the header of the descriptor is 1, DMA transfer is executed according to the descriptor information.

<6> Descriptor writeback

If WBD in the header is 0 after completion of DMA transaction for the set number of bytes, data is written back to bits 31 to 24 in the header of descriptor 2 with LV = 0. For other data fields, the value read in step <4> is written back through byte access.

Steps <4> to <6> repeated

Notes

1. If the header's LE is 1 and WBD is 0, the DMA transaction is executed with this descriptor's settings and the processing is ended after writeback with LV = 0.
2. If the header's LE and WBE are 1, the DMA transaction is executed with this descriptor's settings and the processing is ended without writeback.
3. When the header's LV is 0, data is written back to the header with D = 1. Then, if SCHCFGn.DRRP is 1, the descriptor is read again following the lapse of the interval specified for the SDSCITVLn.DITVL bit. If SCHCFGn.DRRP is 0, DMA is stopped.

(d) Cautions regarding descriptor

- In the link mode, the settings can be changed through descriptor read, but the timing to change the settings cannot be synchronized with DMA transfer requests via hardware. Therefore, when using DMA transfer requests via hardware, specify the SCHCFGn.AM[2:0], LVL, HEN, LEN, and SEL[1:0] bits before setting the SCHCTRLn.SETEN bit. Do not use the descriptor to change these bits.
- The descriptor cannot be used to change the SCHCFGn.DMS bit (the mode is fixed to the link mode).
- The DMAC judges whether the descriptor is valid or invalid by referencing the DSCFM and LV bits in the header. It is therefore necessary to initialize the memory areas corresponding to the DSCFM and LV bits of the descriptor (by specifying 0001b or 0011b to DSCFM and clearing LV) before the DMAC accesses them.
- To set the next descriptor on the memory during DMA operation, write “1” to the LV bit after the descriptor data fields following the header, such the source address, destination address, next link address, have been set. If the DMAC starts reading the descriptor while the descriptor is being set by the CPU, resulting in a conflict, and in this case the descriptor value before setting might be used to execute DMA transfer. The measures described above is thus required to avoid this problem.
- To leave the information of writeback to the D bit in the header, use byte access to write “1” to the header's LV bit.

(e) Link configuration example

In the link mode, the descriptor can be configured either using the list configuration or the loop configuration, as described below.

List configuration

The link is ended by setting the LE bit in the header of the last descriptor. In the list configuration, the LE bit of the last descriptor must be set.

Loop configuration

The descriptor is set in the loop configuration by setting the link destination of the last descriptor to the address of the start descriptor. To end the loop, either set the header's LE bit or stop the DMAC according to the transfer halt procedure before the DMAC starts reading the descriptor.

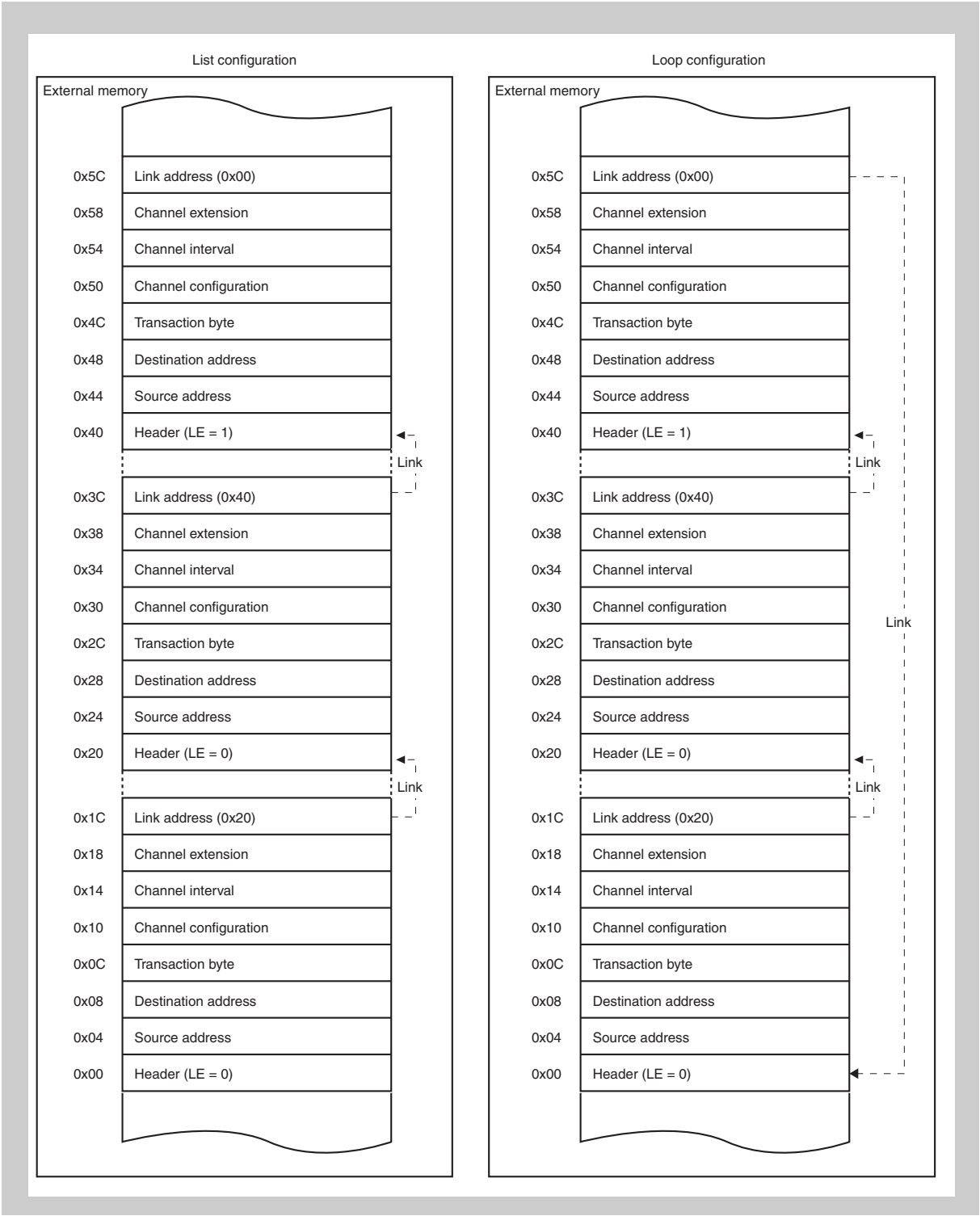


Figure 29-11 Link Mode Configuration Example

(3) Write-only mode

The write-only mode is set by setting the SCHCFGn.WONLY bit.

DMA read transfer cannot be executed in the write-only mode. However, the descriptor is read in the link mode. The value set in the SN0SAn register is used as the write data.

Use the write-only mode when initializing the memory area, etc.

Table 29-41 Write-Only Mode Settings

SCHCFGn.WONLY	Mode	Operation
0	Normal mode	DMA transfer is performed using the values set in the next register set.
1	Write-only mode	DMA read transfer is not performed and only DMA write transfer is performed.

29.5 DMA Transfer

29.5.1 Transfer mode

The DMAC supports the single transfer mode and the block transfer mode.

Use the SCHCFGn.TM bit to select the DMAT transfer mode for each channel.

The single transfer mode is used for DMA transfer that is triggered by a DMA transfer request from the S_DMARQm pin or an interrupt request generated from internal peripheral functions.

The block transfer mode is used for DMA transfer that is triggered using software.

Table 29-42 DMA Transfer Mode Selection

SCHCFGn.TMn	Mode	Operation
0	Single transfer mode	In this mode, a single DMA transfer is executed for an a DMA transfer request from S_DMARQm or an interrupt request generated from internal peripheral functions. This mode is used for DMA transfer that is triggered by the S_DMARQm input or an interrupt request generated from internal peripheral functions.
1	Block transfer mode	In this mode, DMA transfer is repeatedly executed for a single software trigger, until completion of the DMA transaction (series of DMA transfers). This mode is used for DMA transfer that is triggered using software.

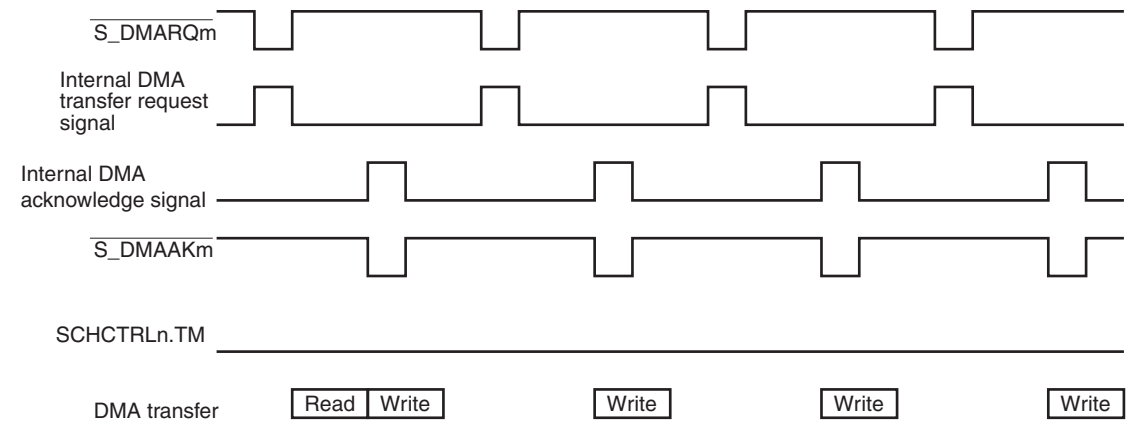
(1) Single transfer mode

This mode is used for DMA transfer that is triggered by the S_DMARQm input.

When a DMA transfer request is acknowledged, DMA transfer is executed once at the side (transfer source or destination) specified for the SCHCFGn.REQD bit, and S_DMAAKm becomes active at the timing specified for the SCHCFGn.AM[2:0] bits.

One DMA transfer is performed each time a transfer request is acknowledged, and this operation is repeated for the number of bytes loaded to the current transaction byte register (SCRTBn). (Arbitration among channels is performed at each DMA transfer.)

The output timing of S_DMAAKm and the count timing of the SCRTBn register differ depending on the SCHCFGn.REQD bit setting and the transfer sizes specified for the DDSn and SDSn bits. For details, see *29.5.11 “Operation differences depending on transfer size” on page 1893*.



- DMA transfer request: Rising edge detection, request from transfer destination
- DMA acknowledge output: Pulse mode
- SDS[3:0] > DDS[3:0] (as an example, when the transfer size of the transfer source is four times the transfer size of the transfer destination)

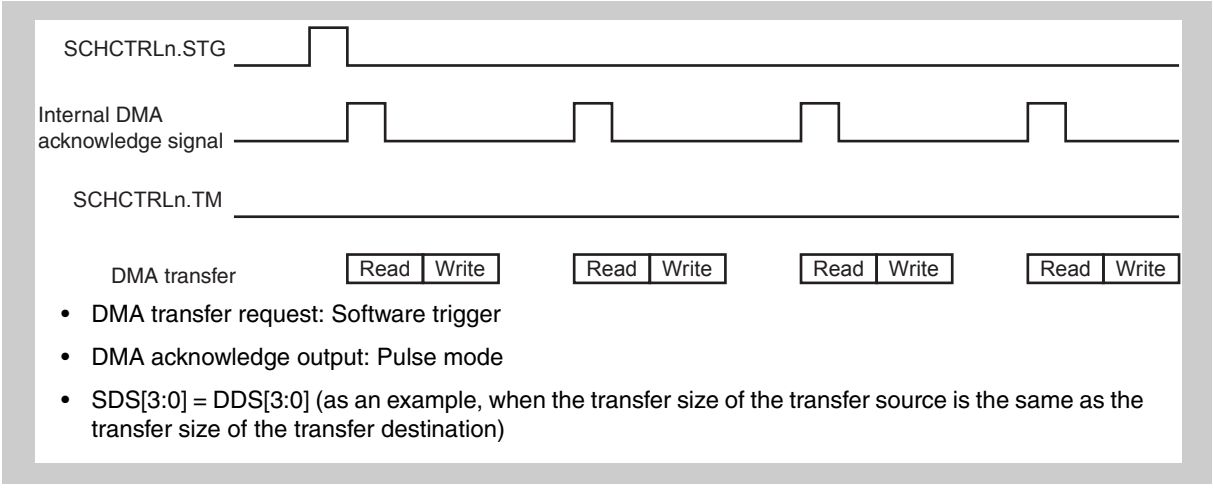
Figure 29-12 Single Transfer Mode Example

Note The DMA interface signals ($\overline{\text{S\_DMARQm}}$, $\overline{\text{S\_DMAAKm}}$, and $\overline{\text{S\_DMATCm}}$) use negative logic. They are inverted internally into positive logic and connected to the DMAC (m = 0 to 3).

(2) Block transfer mode

This mode is used for DMA transfer that is triggered using software. This mode is started up by setting the SCHCTRLn.STG bit.

When a DMA transfer request is acknowledged, transfer is performed repeatedly until the number of bytes loaded to the SCRTBn register has been transferred (DMA transaction completion). (Arbitration among channels is performed at each DMA transfer.)



- DMA transfer request: Software trigger
- DMA acknowledge output: Pulse mode
- SDS[3:0] = DDS[3:0] (as an example, when the transfer size of the transfer source is the same as the transfer size of the transfer destination)

Figure 29-13 Block Transfer Mode Example

29.5.2 Priority order control of DMA channels

The fixed priority mode and round-robin mode are supported for control of the priority order of channels.

Use the PR bit of the DMAC control register (DCTRL) to specify the mode.

Table 29-43 Selection of DMA Channel Priority Order Control

SDCTRL.PR	Mode	Operation
0	Fixed priority order	The priority is controlled with a fixed priority order (High: CH0 > CH1 > CH2 > CH3 > CH4 > CH5 > CH6 > CH7: Low). Use this mode when there is a priority order among the channels.
1	Round robin	The priority is controlled with a round robin. Use this mode to execute DMA transfer evenly among the channels.

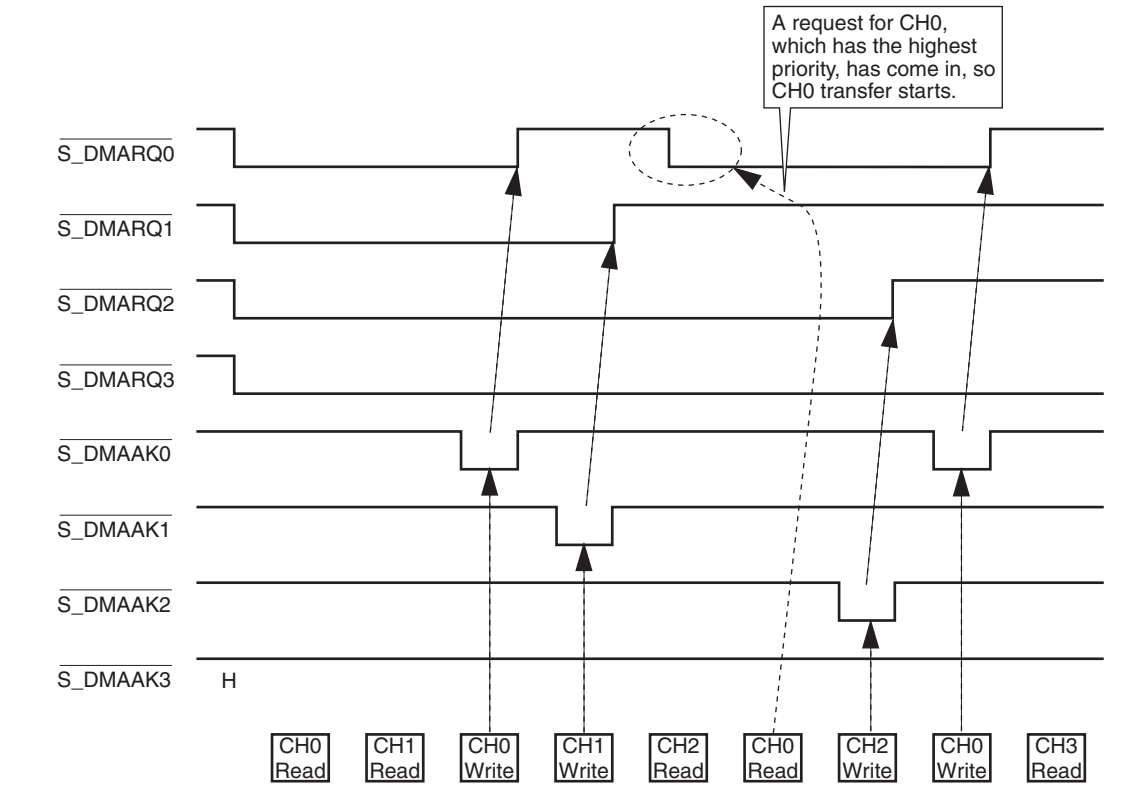
(1) Fixed priority mode

In the fixed priority mode, the priority order of the channels is fixed as follows.

(High) CH0 > CH1 > CH2 > CH3 > CH4 > CH5 > CH6 > CH7 (Low)

If DMA transfer requests are simultaneously output for several channels, the DMA transfer request for the lowest number channel takes priority. The figure below shows an example where a DMA transfer request that has higher priority is output to other channels during DMA transfer execution in the fixed priority mode. (In this figure, four channels are represented for the sake of convenience.)

Caution Unlike in the case of a central DMAC, priority order among DMA channels is controlled even between the read cycle from the transfer source and the write cycle to the transfer destination.



- DMA transfer request: High-level detection, request from transfer destination
- DMA acknowledge output: Level mode

Figure 29-14 Fixed Priority Mode Example

(2) Round-Robin Mode

In the round-robin mode, each time a DMA transfer request for a channel is acknowledged, the priority of the channel on which transfer was performed immediately before is changed to the lowest priority.

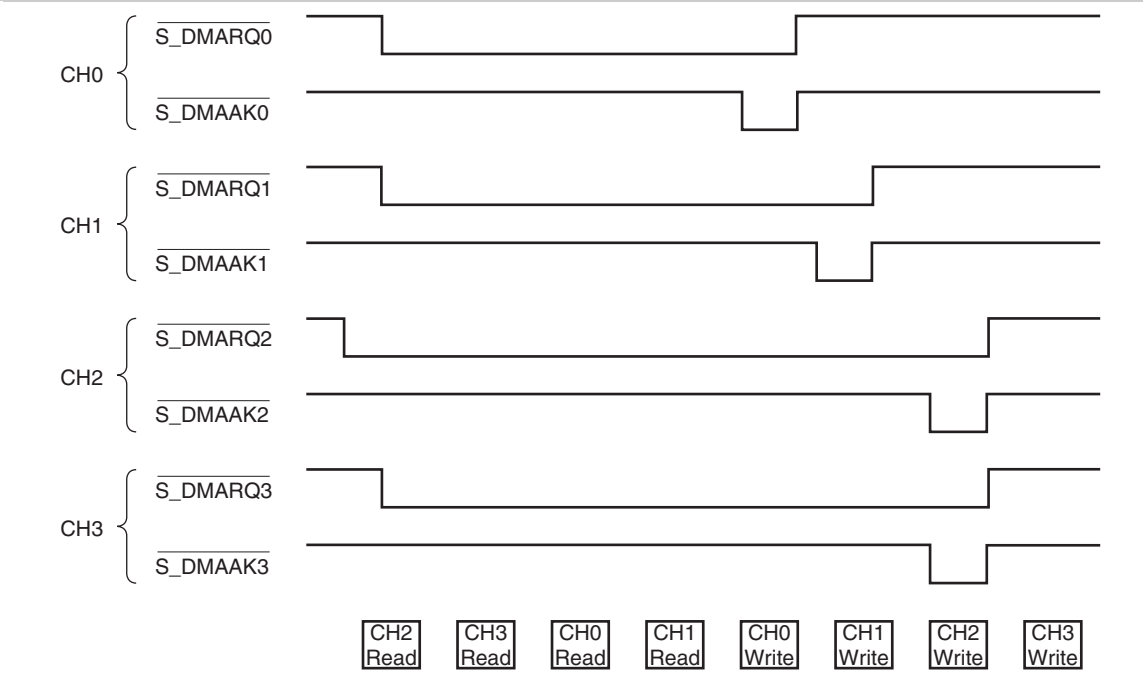
The priority order immediately after reset release is as follows, like in the fixed priority mode.

(High) CH0 > CH1 > CH2 > CH3 > CH4 > CH5 > CH6 > CH7 (Low)

In this state, if there is no transfer request for DMA channel 0 and there is a transfer request for DMA channel 2, transfer on DMA channel 2 is performed. Then, at the end of the transfer, the priority order becomes as follows.

(High) CH3 > CH4 > CH5 > CH6 > CH7 > CH0 > CH1 > CH2 (Low)

The figure below shows an example of DMA transfer in the round-robin mode. (In this figure, four channels are represented for the sake of convenience.)



- DMA transfer request: High-level detection, request from transfer destination
- DMA acknowledge output: Level mode

Figure 29-15 Round-Robin Mode

Note Arbitration is performed among the read channels and among the write channels.

29.5.3 DMA transfer trigger source

There are two types of DMA transfer trigger source, as shown below:

(1) Request from external pin (S_DMARQm)

Requests from the S_DMARQm pin are sampled at each rising edge of the CLKOUT signal.

Hold requests from the S_DMARQm pin until the corresponding S_DMAAKm signal becomes active.

When the SCHCTRLn.EN bit is set, the S_DMARQm signal in the TI state becomes valid. When the S_DMARQm signal in the TI state becomes active, the T0 state is entered and DMA transfer starts.

(2) Request by software

When the SCHCTRLn.STG and EN bits are set as follows, DMA transfer starts.

- STG bit = 1
- EN bit = 1

29.5.4 DMA transfer request

The relationships between DMA transfer requests and the various channels can be changed with the SCHCFGn.SEL[1:0] bits.

Caution The internal DMA interface uses positive logic, except the DMA interface pins including S_DMARQm, S_DMAAKm, and S_DMATCm (m = 0 to 3), which use negative logic. The DMA interface pin signals are inverted internally and connected to the DMAC. The falling edge of S_DMARQm is detected when the rising edge is set in this field. Likewise, the internal signals of S_DMAAKm and S_DMATCm are active high. These signals are inverted to negative logic before they are output to pins.

(1) Allocation of DMA channels and DMA transfer requests

In the round-robin mode, in which priority is evenly allocated among the DMA channels, select the same channel and pin names using the SCHCFGn.SEL[1:0] bits. For example, select S_DMARQ2, S_DMAAK2, and S_DMATC2 for the DMA interface signals of channel 2.

In the fixed priority mode, change the relationship between the DMA channels and DMA interface signals by using the SCHCFGn.SEL[1:0] bits according to the required DMA transfer request priority level.

(2) Specification of detection operation for each DMA transfer request source

The method to detect DMA transfer requests is in some cases dictated by the DMA transfer request source.

Based on the table below, specify the correct detection operation by using the LVL, LEN, and HEN bits of the SCHCFGn register:

Table 29-44 DMA Transfer Request Signal Detection Methods

LVL	HEN	LEN	DMA Transfer Request Signal Detection Methods	
0	0	0	Edge detection	Detection disabled
0	0	1		Falling edge detection
0	1	0		Rising edge detection
0	1	1		Rising and falling edge detection
1	0	0	Level detection	Detection disabled
1	0	1		Low-level detection
1	1	0		High-level detection
1	1	1		DMA transfer is started when the SCHCTRLn.SETEN bit is set, regardless of the DMA transfer request input level.

(3) Edge detection

Edge detection is selected by clearing the SCHCFGn.LVL bit.

Rising edge detection is enabled by setting the SCHCFGn.HEN bit, and falling edge detection is enabled by setting the LEN bit.

To use the $\overline{\text{S\_DMARQ0}}$ to $\overline{\text{S\_DMARQ3}}$ signals for DMA transfer requests, output the next DMA transfer request ($\overline{\text{S\_DMARQ0}}$ to $\overline{\text{S\_DMARQ3}}$) after detecting a DMA acknowledge signal ($\overline{\text{S\_DMAAK0}}$ to $\overline{\text{S\_DMAAK3}}$).

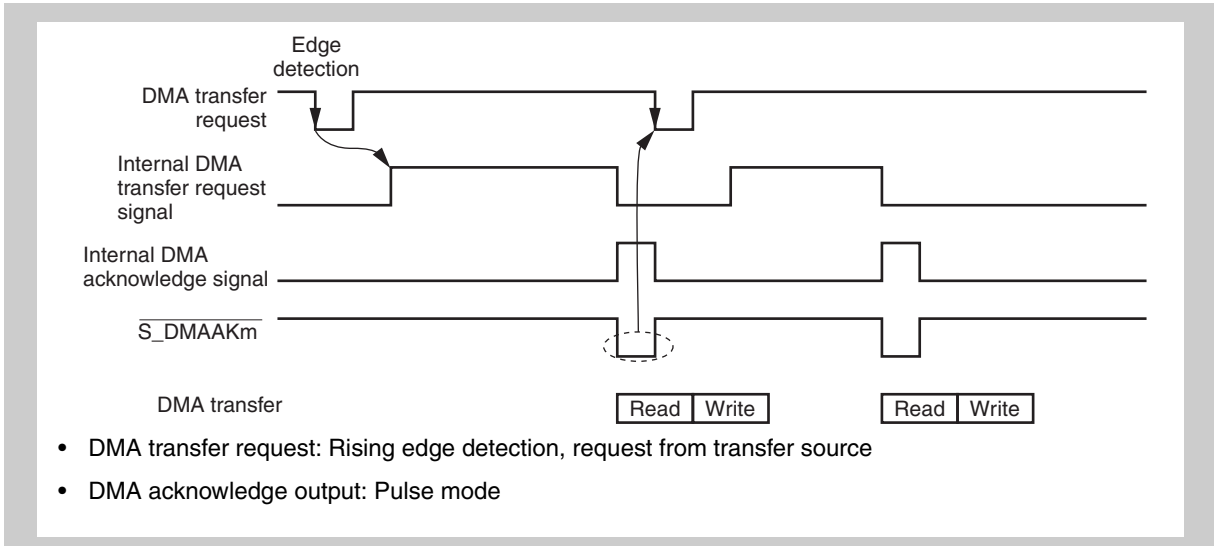
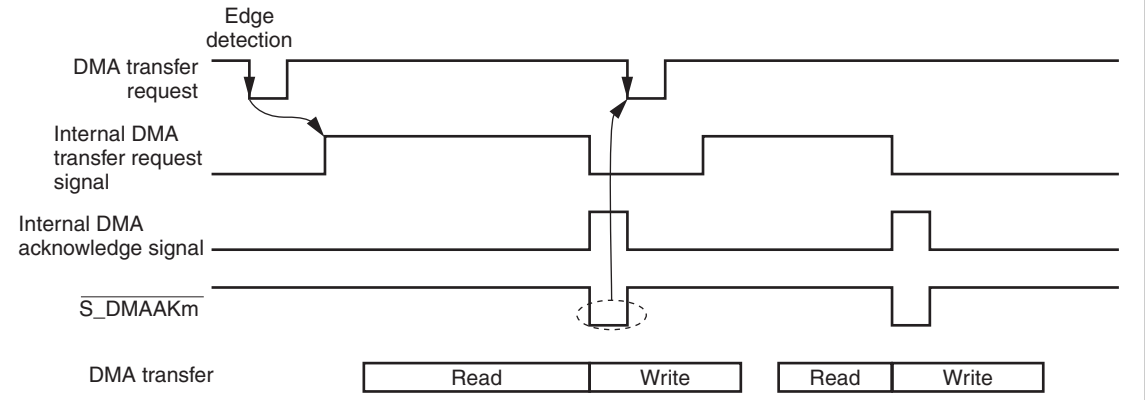


Figure 29-16 Edge Detection Mode Operation Example 1



- DMA transfer request: Rising edge detection, request from transfer destination
- DMA acknowledge output: Pulse mode

Figure 29-17 Edge Detection Mode Operation Example 2

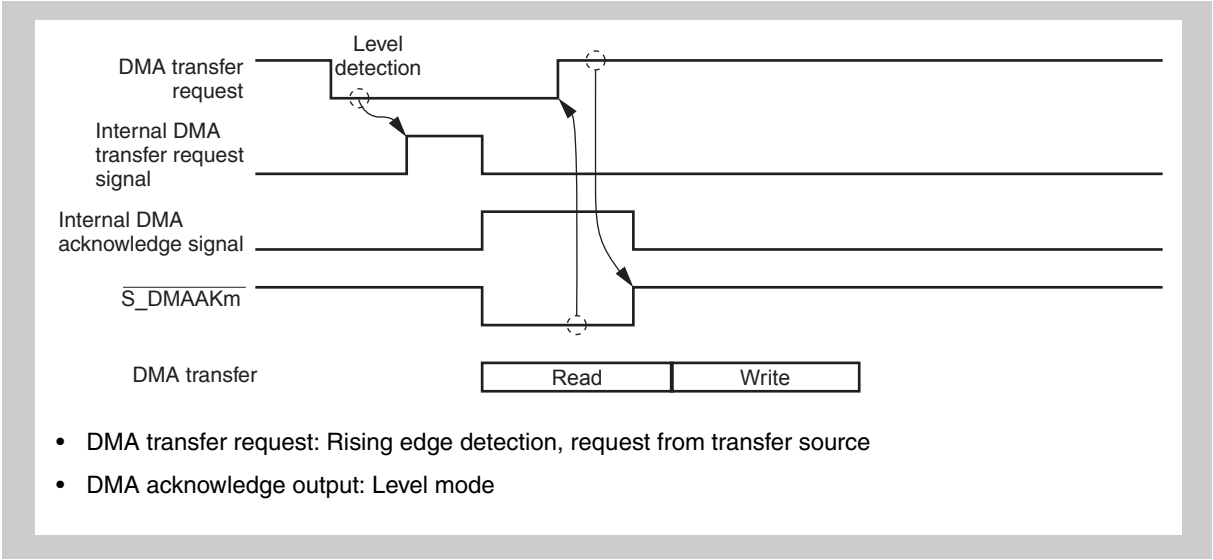
(4) Level detection

Level detection is selected by setting the SCHCFGn.LVL bit.

To use the $\overline{\text{S\_DMARQ0}}$ to $\overline{\text{S\_DMARQ3}}$ signals for DMA transfer requests, secure the effective level of these signals (specified for SCHCFGn.HEN and SCHCFGn.LEN) for twice the BUSCLK (SBUSCLK) width.

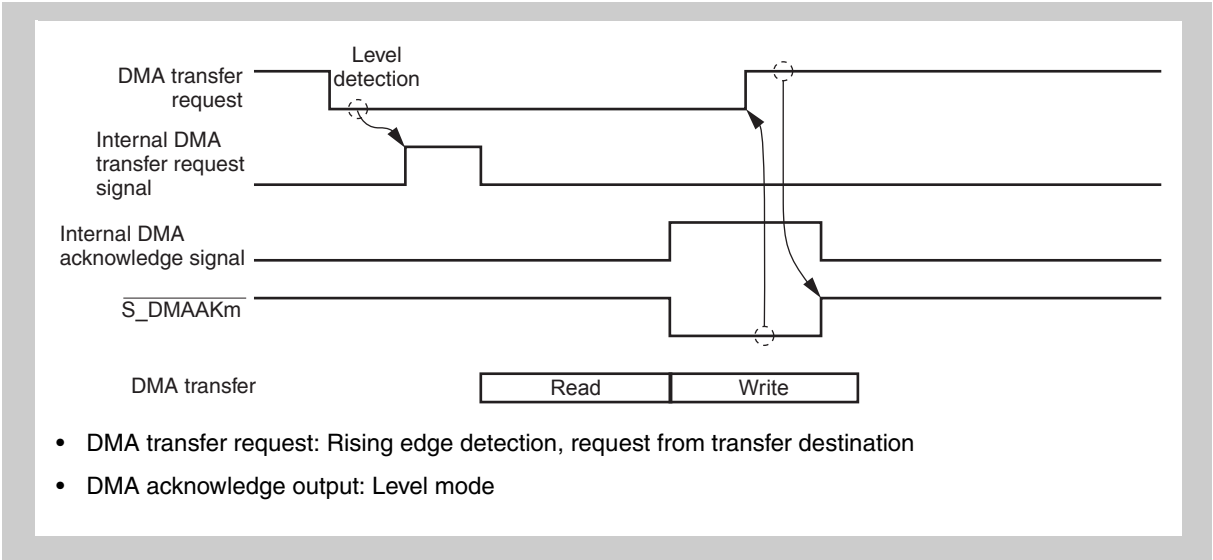
If the level mode is selected for the DMA acknowledge signal, $\overline{\text{S\_DMAAKm}}$ retains low level until $\overline{\text{S\_DMARQm}}$ becomes inactive. When the pulse mode is selected, $\overline{\text{S\_DMAAKm}}$ is output for one BUSCLK (SBUSCLK) pulse width.

To use the $\overline{\text{S\_DMARQ0}}$ to $\overline{\text{S\_DMARQ3}}$ signals for DMA transfer requests, output the next DMA transfer request ($\overline{\text{S\_DMARQ0}}$ to $\overline{\text{S\_DMARQ3}}$) after detecting a DMA acknowledge signal ($\overline{\text{S\_DMAAK0}}$ to $\overline{\text{S\_DMAAK3}}$).



- DMA transfer request: Rising edge detection, request from transfer source
- DMA acknowledge output: Level mode

Figure 29-18 Level Detection Mode Operation Example 1



- DMA transfer request: Rising edge detection, request from transfer destination
- DMA acknowledge output: Level mode

Figure 29-19 Level Detection Mode Operation Example 2

29.5.5 DMA acknowledge output function

A DMA acknowledge signal is output as a response signal for a DMA transfer request.

To use the S_DMARQ0 to S_DMARQ3 signals for DMA transfer requests, use the S_DMAAK0 to S_DMAAK3 signals as the DMA acknowledge signal.

The relationships between DMA transfer requests and the various channels can be changed with the SCHCFGn.SEL[1:0] bits.

(1) Allocation of DMA channels and DMA acknowledge signals

In the round-robin mode, in which priority is evenly allocated among the DMA channels, select the same channel and pin names using the SCHCFGn.SEL[1:0] bits. For example, select S_DMARQ2, S_DMAAK2, and S_DMATC2 for the DMA interface signals of channel 2.

In the fixed priority mode, change the relationship between the DMA channels and DMA interface signals by using the SCHCFGn.SEL[1:0] bits according to the required DMA transfer request priority level.

(2) Specification of acknowledge signal mode for each DMA transfer request source

The DMA acknowledge signal output mode is in some cases dictated by the source.

Based on the table below, specify the correct detection operation by using the SCHCFGn.AM[2:0] bits.

Table 29-45 DMA Acknowledge Signal (S_DMAAKm) Output Modes

AM2	AM1	AM0	DMA Acknowledge Signal (S_DMAAKm) Output Modes
0	0	0	Pulse mode[a] (default)
0	0	1	Level mode The DMA acknowledge signal retains the active level until the DMA transfer request (S_DMARQm) becomes inactive.
0	1	Any	Bus cycle mode The DMA acknowledge signal retains the active level during the DMA transfer bus cycle.
1	Any	Any	DMA acknowledge signal (S_DMAAKm) output prohibited

a) A pulse of 1 × BUSCLK cycle is output as the S_DMAAKm signal.

Cautions

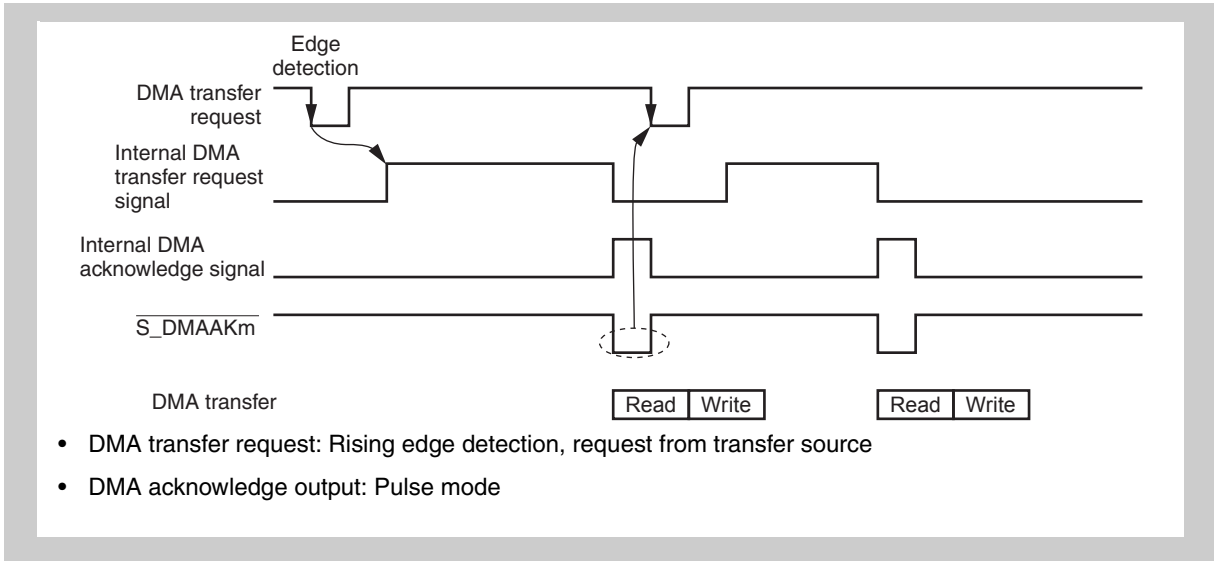
1. If the interrupt request signal of an internal peripheral function or an external interrupt input is selected using the DTFR register, the setting of AM[2:0] does not affect the operation.
2. The AM[2:0] settings and DMAIFCn register can be multiply set, but as the general usage method, when the operation of the S_DMAAKm signal is set to the level mode, use the DMAIFCn register with the initial value as is. Reversely, when the DMAIFCn register is used to expand the S_DMAAKm pulse width or mask the S_DMARQm, select the pulse mode for AM[2:0].

(3) Pulse output

Pulse output is selected for the DMA acknowledge signal ($\overline{\text{S\_DMAAKm}}$) by setting the SCHCFGn.AM[2:0] bits to 000b.

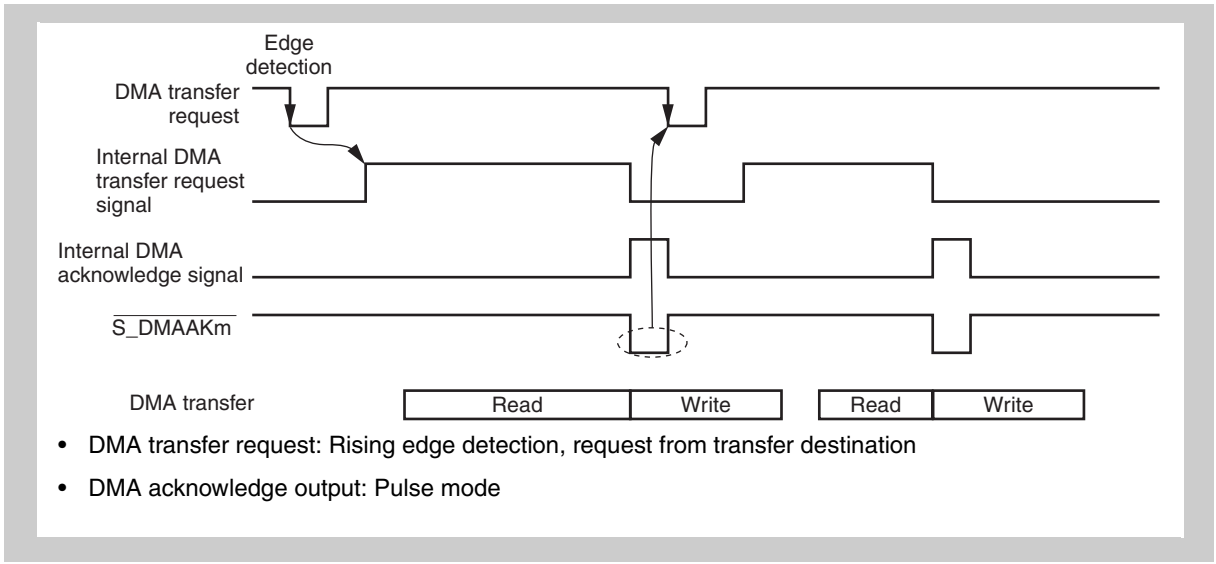
A low-level pulse of 1 × BUSCLK (SBUSCLK) width is output.

If the pulse width is short at the DMA transfer request source, the $\overline{\text{S\_DMAAKm}}$ width can be adjusted from 1 to 32 times BUSCLK (SBUSCLK) by specifying the AKWDn4 to AKWDn0 bits of the DMA transfer interface signal control registers 0 to 3 (DMAIFC0 to DMAIFC3).



- DMA transfer request: Rising edge detection, request from transfer source
- DMA acknowledge output: Pulse mode

Figure 29-20 Edge Detection Mode Operation Example 1



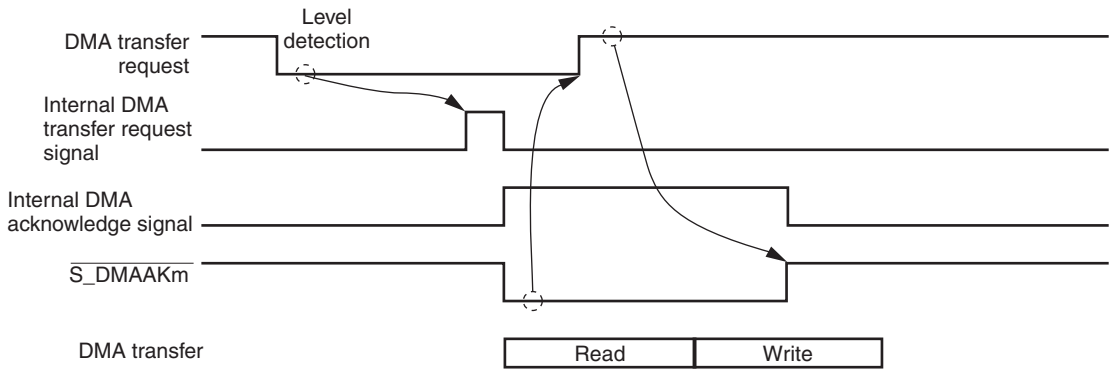
- DMA transfer request: Rising edge detection, request from transfer destination
- DMA acknowledge output: Pulse mode

Figure 29-21 Edge Detection Mode Operation Example 2

(4) Level output

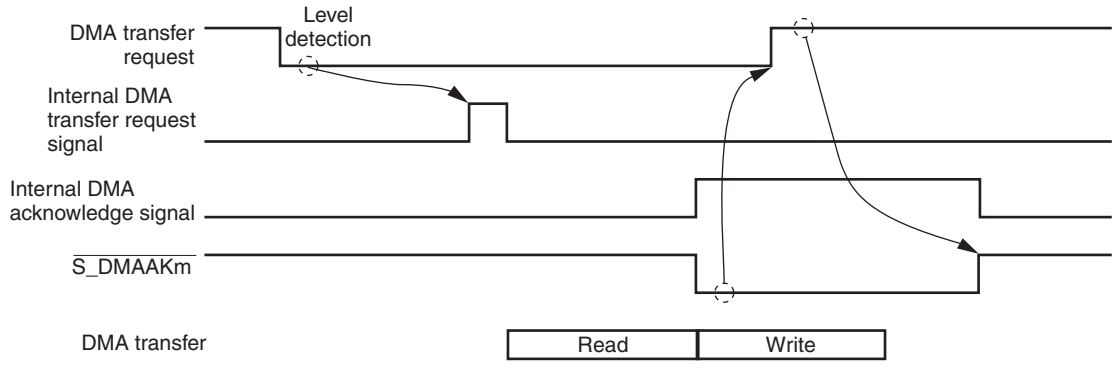
Level output is selected for the DMA acknowledge signal ($\overline{\text{S\_DMAAKm}}$) by setting the SCHCFGn.AM[2:0] bits to 001_b. The $\overline{\text{S\_DMAAKm}}$ signal retains the active level until the $\overline{\text{S\_DMARQm}}$ signal becomes inactive.

If level output is selected for the DMA acknowledge signal, use the initial values of DMA transfer interface signal control registers 0 to 3 (DMAIFC0 to DMAIFC3) as is, and do not expand the $\overline{\text{S\_DMAAKm}}$ width.



- DMA transfer request: High-level detection, request from transfer source
- DMA acknowledge output: Level mode

Figure 29-22 Level Detection Mode Operation Example 1



- DMA transfer request: High-level detection, request from transfer destination
- DMA acknowledge output: Level mode

Figure 29-23 Level Detection Mode Operation Example 2

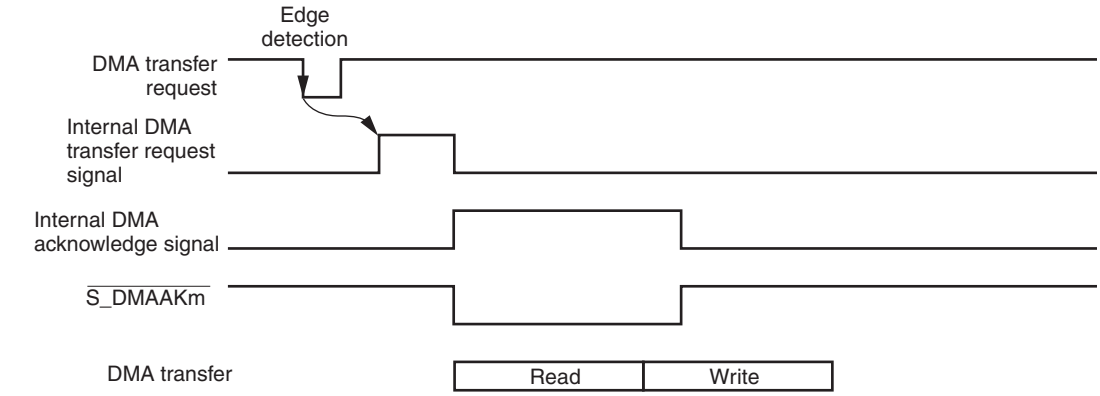
(5) Bus cycle output

Bus cycle output is selected for the DMA acknowledge signal ($\overline{\text{S\_DMAAKm}}$) by setting the SCHCFGn.AM[2:0] bits to 010_b.

The $\overline{\text{S\_DMAAKm}}$ signal is active (low level) during the bus cycle period. The DMA acknowledge signal output destination depends on the DMA transfer request source. If the transfer is requested from the transfer source, the DMA acknowledge signal is output to a read cycle, and if requested from the transfer destination, the DMA acknowledge signal is output to a write cycle. If, owing to bus size differences between the transfer source and the transfer destination, multiple reads/writes occur during one DMA transfer (transaction), $\overline{\text{S\_DMAAKm}}$ is active during this time.

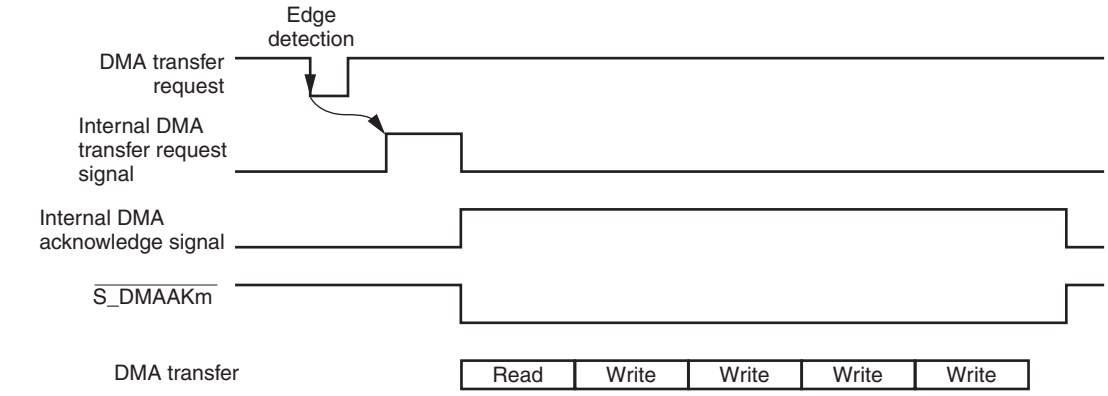
If bus cycle output is selected for the DMA acknowledge signal, use the initial values of DMA transfer interface signal control registers 0 to 3 (DMAIFC0 to DMAIFC3) as is, and do not expand the $\overline{\text{S\_DMAAKm}}$ width.

Caution In the bus cycle output mode, no $\overline{\text{S\_DMARQm}}$ can be acknowledged during the DMA transfer bus cycle period and the following one BUSCLK (SBUSCLK) cycle.



- DMA transfer request: Rising edge detection, request from transfer source
- DMA acknowledge output: Level mode

Figure 29-24 Bus Cycle Output Mode Operation Example 1



- DMA transfer request: Rising edge detection, request from transfer destination
- DMA acknowledge output: Pulse mode
- SDS[3:0] > DDS[3:0] (example where the transfer size of the transfer source is four times the transfer size of the transfer destination)

Figure 29-25 Edge Detection Mode Operation Example 2

29.5.6 DMA transfer completion interrupt

Upon completion of a DMA transaction (series of DMA transfers), INTHDMAn (n = 0 to 7) is generated.

INTHDMAn corresponds to DMA channel n.

Upon completion of the transfer of the total number of bytes loaded to the SCRTBn register, the SCHSTATn.END bit is set. If, at this time, the SCHCFGn.DEM bit is 0, INTHDMAn is generated.

If writeback is performed in the link mode, INTHDMAn is generated after writeback. If the descriptor is read while SCHCFGn.DRRP is 0 and the read LV in the header is 0, the SCHSTATn.DER bit is set. At this time, INTHDMAn is generated if SCHCFGn.DIM is 0.

Table 29-46 INTHDMAn Active Conditions

Cause	Condition	INTHDMAn mask setting bit
DMA transaction	Upon normal completion of the transfer of the total number of bytes loaded to the SCRTBn register (After writeback if writeback is performed in the link mode)	SCHCFGn.DEM
Descriptor invalid (Header's LV = 0)	In the link mode, when the LV bit of the read descriptor is 0 while SCHCFGn.DRRP and SCHCFGn.DIM are 0.	SCHCFGn.DIM

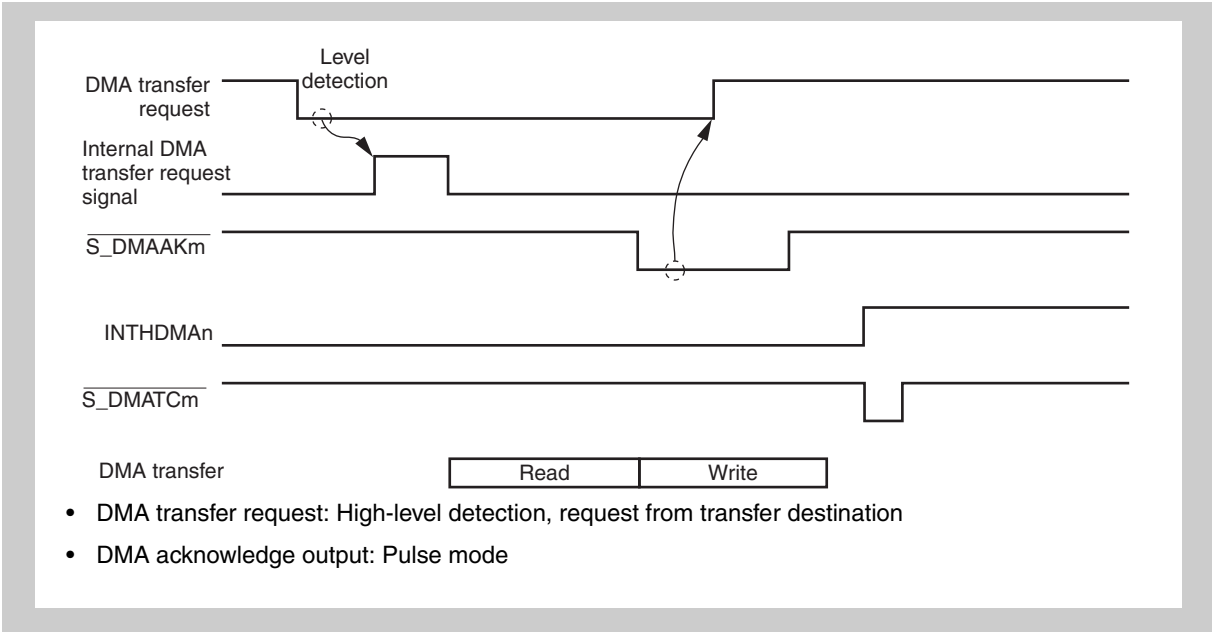


Figure 29-26 INTHDMAn Output Operation Example

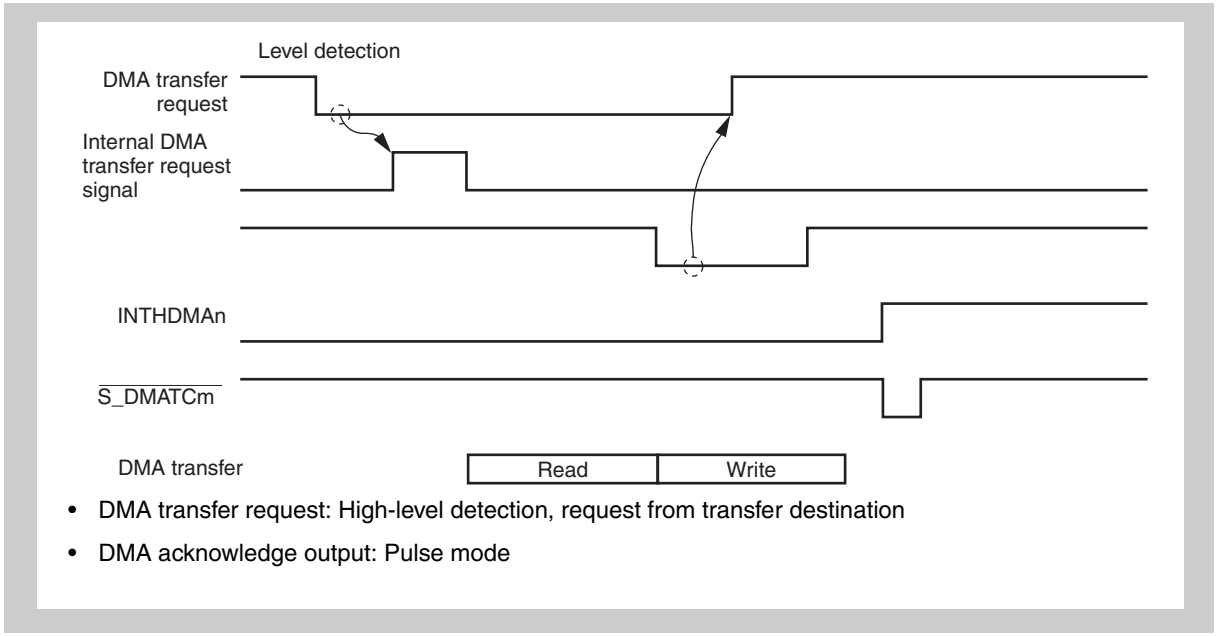
29.5.7 DMA terminal count output function

The DMA terminal count signal is output as the DMA transaction (series of DMA transfers) completion signal.

To use the $\overline{\text{S\_DMARQ0}}$ to $\overline{\text{S\_DMARQ3}}$ signals for DMA transfer requests, use the $\overline{\text{S\_DMATC0}}$ to $\overline{\text{S\_DMATC3}}$ signals as the DMA terminal count signal.

Although no DMA terminal count signal is used if an external interrupt or an interrupt request from internal peripheral functions is used for DMA transfer requests, the DMA terminal count signal specified for the SCHCFGn.AM[2:0] bits is output in this case.

Upon normal completion of transfer of the total number of bytes loaded to the SCRTBn register, the SCHSTATn.TC bit is set, and the DMA terminal count signal ($\overline{\text{S\_DMATCm}}$) is output as a low level signal for the duration of 1 × BUSCLK (SBUSCLK) cycle.



- DMA transfer request: High-level detection, request from transfer destination
- DMA acknowledge output: Pulse mode

Figure 29-27 INTHDMAn Output Operation Example

(1) Mask function of DMA terminal count signal

The DMA terminal count signal can be masked by using the SCHCFGn.TCM bit. Generally, the DMA terminal count signal is masked if the DMA transfer is triggered using software (by setting SCHCTRLn.STG bit).

The relationships between DMA transfer requests and the various channels can be changed with the SCHCFGn.SEL[2:0] bits. The DMA terminal count signal is also output in coordination with this selection.

Table 29-47 DMA Terminal Count Output Settings

SCHCFGn.TCMm	Operation	Application
0	DMA terminal count output enabled	Use this setting for DMA transfer triggered by hardware. Use this setting to detect: • Count end • Link mode end
1	DMA terminal count output masked	Use this setting for DMA transfer triggered by software. Upon completion of a DMA transaction (series of DMA transfers), TCM is cleared and the DMA terminal count output is enabled.

(2) Allocation of DMA channels and DMA terminal count output signals

In the round-robin mode, in which priority is evenly allocated among the DMA channels, select the same channel and pin names using the SCHCFGn.SEL[1:0] bits. For example, select S_DMARQ2, S_DMAAK2, and S_DMATC2 for the DMA interface signals of channel 2.

In the fixed priority mode, change the relationship between the DMA channels and DMA interface signals by using the SCHCFGn.SEL[1:0] bits according to the required DMA transfer request priority level.

29.5.8 Forced sweep function

When the SCHCTRLn.SETSSWPRQ bit is set, the DMAC forcibly sweeps (writes) out the data of the internal buffers to the transfer destination. The DMA transfer is then resumed.

The forced sweep function cannot be used if the SCHCFGn.REQD bit is set and S_DMAAKm becomes active when it is written. This is because there is the possibility that erroneous operation may occur at the transfer destination if data transfer is performed while the DMA transfer request (S_DMARQm) is not active.

Sweep is also performed when the SCHCFGn.SBE bit is set. In this case, stop the DMA operation by clearing the SCHSTATn.EN bit after sweep. If forced sweep is started by setting the SETSSWPRQm bit, DMA transfer continues even after the sweep.

29.5.9 DMA error interrupt

DMA transfer is interrupted if an error occurs during DMA transfer or descriptor access.

When an error occurs, the SCHSTATn.EN bit is cleared and the ER bit is set. Also, INTDMAERR is generated.

The data of the series of transfers for which the error occurred cannot be guaranteed. To restart DMA transfer, set the SCHCTRLn.SWRST bit, reset DMA channel n, and set the various registers again.

29.5.10 Interval count function

The execution interval of DMA transfers can be adjusted by specifying the SCHITVLn.ITVL[15:0] bits. The interval can be set in the range of system bus clock (HCLK) period × ITVL[15:0] setting value. Through this, the bus usage rate can be adjusted. Upon completion of one read or write operation, a countdown begins from the value specified for the SCHITVLn register, and the next internal DMA transfer request is held until the count value reaches 0.

29.5.11 Operation differences depending on transfer size

(1) If the transfer size of the transfer source is smaller than that of the transfer destination

After data of the size specified for the SCHCFGn.DDS[3:0] bits has been read, a write operation to the transfer destination starts. The number of writes is transfer destination size/transfer source size.

The timing chart when the transfer size of the source is 16 bits and that of the destination is 64 bits is shown below.

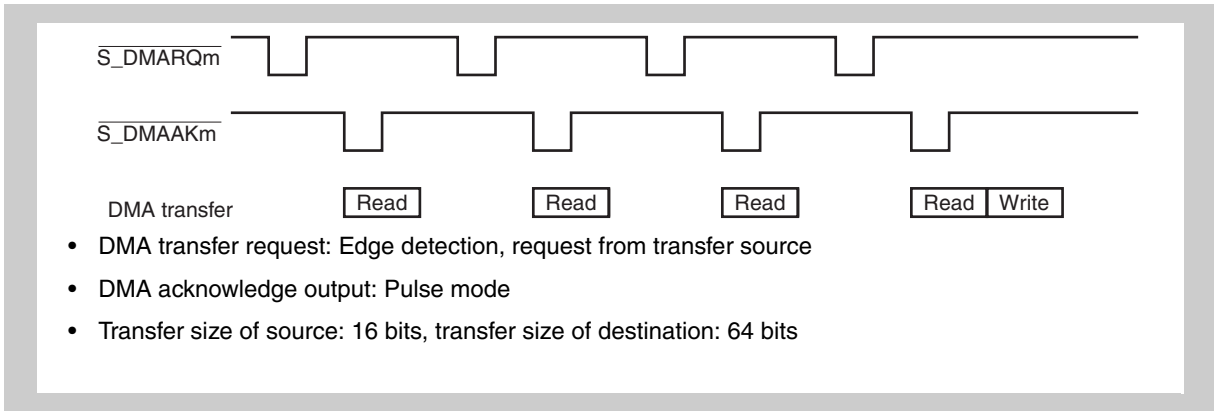
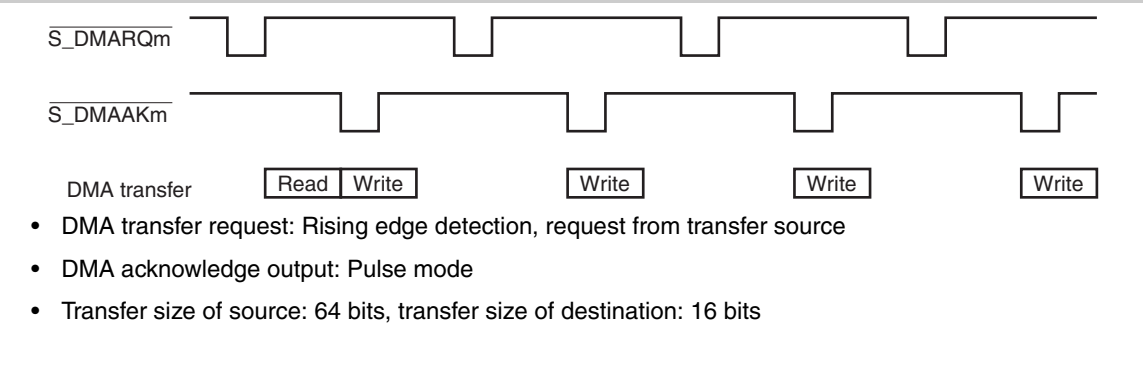


Figure 29-28 Example Where Transfer Size of Transfer Source Is Smaller than That of Transfer Destination

(2) If the transfer size of the transfer destination is smaller than that of the transfer source

The number of writes equal to transfer source/transfer destination takes place because the transfer destination size is smaller.

The timing chart when the transfer size of the source is 64 bits and that of the destination is 16 bits is shown below.

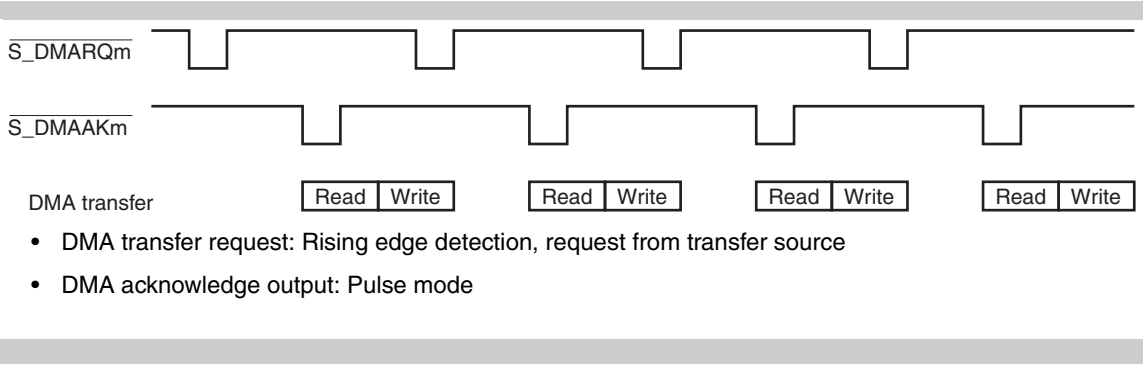


- DMA transfer request: Rising edge detection, request from transfer source
- DMA acknowledge output: Pulse mode
- Transfer size of source: 64 bits, transfer size of destination: 16 bits

Figure 29-29 Example Where Transfer Size of Transfer Destination Is Smaller than That of Transfer Source

(3) When the transfer size of the transfer destination is equal to that of the transfer source

Read from the transfer source and write to the transfer destination are performed each time a DMA transfer request is detected.



- DMA transfer request: Rising edge detection, request from transfer source
- DMA acknowledge output: Pulse mode

Figure 29-30 Example Where Transfer Size of Transfer Source Is Equal to That of Transfer Destination

29.5.12 Transfer status

The transfer execution status for DMA channel n can be checked with the SCHSTATn register.

The SCHSTATn.TACT bit indicates whether channel n is currently operating. The TACT bit is set when the SCHCTRLn.SETEN bit is set. The TACT bit is also set during access to the descriptor in the link mode or while a DMA transfer request is being waited for.

The TACT bit is cleared when the clear condition for the SCHSTAT.EN bit is met and DMA transfer has completed. Note that the TACT bit is not cleared even after completion of a DMA transaction, unless the clear condition for the SCHSTAT.EN bit is met (for example, the next descriptor is accessed in the link mode).

The transfer state is updated each DMA transfer.

29.5.13 Suspend function

DMA transfer can be suspended by setting the SCHCTRLn.SETSUS bit. If a bus cycle is currently being executed at that time, the suspended state is entered following completion of that bus cycle. The suspended state is cancelled by setting the SCHCTRLn.CLRSUS bit.

To check the suspended state, first set the SETSUS bit, and then read the SCHSTATn register or the SDSTAT_SUS register to check whether the SUS bit of the corresponding channel has been set.

29.5.14 Transfer halt function

By setting the SCHCTRLn.CLREN bit during a DMA transaction (series of DMA transfers), the DMA transaction (series of DMA transfers) of that DMA channel can be halted.

The processing after the DMA transaction has been halted can be selected from between the mode in which the remaining data is swept to a buffer at the halt timing (by setting the SCHCFGn.SBE bit), and the mode in which sweep is not performed (by clearing the SBE bit).

If, when the sweep mode is enabled and a DMA transaction (series of DMA transfers) has been halted by setting CLREN = 1, data remains in the DMAC buffer, that data is swept and the DMA transaction is completed.

No INTHDMAn interrupt is generated when DMA transfer is halted.

Following DMA transfer halt, be sure to set the SCHCTRLn.SWRST bit to reset the internal status of the DMA channel before specifying the next transfer setting.

Caution DMA transfer may still be in progress even after the SCHCTRLn.CLREN bit is set and the EN bit is cleared. To check whether the DMA channel operation is stopped, check that SCHSTATn.EN and SCHSTATn.TACT are 0.

(1) When buffer sweep is disabled (SBE = 0)

If the CLREN bit is set during a DMA transaction (series of DMA transfers), DMA transfer is halted and stopped.

The DMA transfer halt timing depends on the SCHCFGn.REQD bit setting. If the DMA transfer is requested from the transfer source, the transfer halts after a read cycle, and if requested from the transfer destination, the transfer halts after a write cycle.

(2) When buffer sweep is enabled (SBE = 1)

If the CLREN bit is set during a DMA transaction (series of DMA transfers), DMA transfer is halted and stopped. If REQD is 0, the DMA transfer stops after sweeping (writing) out the already read data. Do not use the sweep mode if REQD is 1 and DMA transfer requests via hardware are used.

Chapter 30 USB Function Controller (USBF)

This microcontroller has an internal USB function controller (USBF) conforming to the Universal Serial Bus Specification. Data communication using the polling method is performed between the USB function controller and external host device by using a token-based protocol.

Caution Use the following procedure to read the USB Function controller registers.

- When connecting an external clock to the USBCLK pin

  (1) Set the EPC_RST bit of the EPC macro control register (USFA0EPCCTL) to 0 to cancel the reset.

  (2) Read the required USB function controller registers.

- When an external clock is not to be connected to the USBCLK pin

  [1] When supplying an external clock is specified in the SFRCTL3 register (When the USBDIV1 and USBDIV0 bits of SFRCTL3 are 00b)

  (1) Set the USBDIV1 and USBDIV0 bits of the SFRCTL3 register to 11b to specify an internal clock of fxx/4.

  (2) Set the EPC_RST bit of the EPC macro control register (USFA0EPCCTL) to 0 to cancel the reset.

  (3) Read the required USB function controller registers.

  [2] When supplying an internal clock is specified in the SFRCTL3 register (When the USBDIV1 and USBDIV0 bits of SFRCTL3 are 10b or 11b)

  (1) Set the EPC_RST bit of the EPC macro control register (USFA0EPCCTL) to 0 to cancel the reset.

  (2) Read the required USB function controller registers.

30.1 V850E2/MN4 USBF Features

Register addresses

All USBF register addresses are given as addresses offset from the base address <USBF_base>. Each <USBF_base> address of USBF is shown in the following table:

Table 30-1 Register base addresses <USBF_base>

USBF function	<USBF_base> address
EPC control registers	F993 1000_H
EPC data hold registers	F993 1000_H
EPC request data registers	F993 1000_H
Bridge registers	F993 1000_H

Clock supply

The following clocks can be input to USBF:

Table 30-2 USBF clock supply

USBF clock (f_{USB})	
Internal	9.6 MHz external clock x internal clock multiplied by 20/4 = 48 MHz internal clock
	7.2 MHz external clock x internal clock multiplied by 20/3 = 48 MHz internal clock
External	External clock input from the UCLK pin (f_{USB} = 48 MHz)

Interrupts

USBF can generate the following interrupt requests.

Table 30-3 USBF interrupt requests

USBF signal	Function	Connected to:
INTUSFA0I0	Bridge interrupt	• Interrupt controller 187 (INTUSFA0I0)
INTUSFA0I1	EPC interrupt	• Interrupt controller 188 (INTUSFA0I1)
INTUSFA0I2	USB Resume detection	• Interrupt controller 189 (INTUSFA0I2)

I/O signals

The USBF I/O signals are shown in the following table.

Table 30-4 USBF I/O signals

USBF signal	Function	Connected to:
UDMF	USBF data I/O (–)	UDMF pin
UDPF	USBF data I/O (+)	UDPF pin

30.1.1 V850E2/MN4 USBF control register

Before using the USB function controller, set up the register below.

(1) USBF buffer control register (USFBC)

This register controls whether to enable or disable the input buffer for the USB function controller and whether to enable or disable the floating countermeasure.

Access This registers can be read or written in 32-bit units.

Address F990 1004_H

Initial value $0000\ 0000_H$. This register is initialized by any reset.

	31	30	29	28	27	26	25	24
USFBC	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R
	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R
	15	14	13	12	11	10	9	8
	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	UBFIEN	UBFIOR
	R	R	R	R	R	R	R/W	R/W

Table 30-5 USFBC register contents

Bit position	Bit name	Function
1	UBFIEN	This bit controls whether to use the USB buffer. 0: Disable the buffer. 1: Enable the buffer. **Caution: If not using USBF, be sure to clear the UBFIEN bit (to 0).**
0	UBFIOR	This bit controls whether to use the floating countermeasure for the USB buffer. 0: Disable the floating countermeasure. 1: Enable the floating countermeasure. When the cable is not connected (when the data input is floating), this countermeasure prevents the incorrect recognition of Bus Reset, Suspend, Resume, and other signals due to undefined values. If this bit is set, control the floating countermeasure by using a VBUS or other signal to recognize the cable connection.

30.2 Overview

- Conforms to the Universal Serial Bus Specification
- Supports 12 Mbps (full-speed) transfers
- Endpoint for transfer incorporated

Endpoint name	FIFO size (bytes)	Transfer type	Remark
Endpoint0 Read	64	Control transfer	–
Endpoint0 Write	64	Control transfer	–
Endpoint1	64 × 2	Bulk 1 transfer (IN)	Two-buffer configuration
Endpoint2	64 × 2	Bulk 1 transfer (OUT)	Two-buffer configuration
Endpoint3	64 × 2	Bulk 2 transfer (IN)	Two-buffer configuration
Endpoint4	64 × 2	Bulk 2 transfer (OUT)	Two-buffer configuration
Endpoint7	64	Interrupt transfer	–
Endpoint8	64	Interrupt transfer	–

- A bulk transfer (IN/OUT) can be executed as a DMA transfer (2-cycle single-transfer mode).

Caution The registers in the USB function controller register list (in *30.7.1 “USB function controller registers” on page 1917*) must be accessed after specifying that the internal clock or the external clock is to be used as the USB clock and enabling clock supply to the USB function controller.

30.3 Configuration

30.3.1 Block diagram

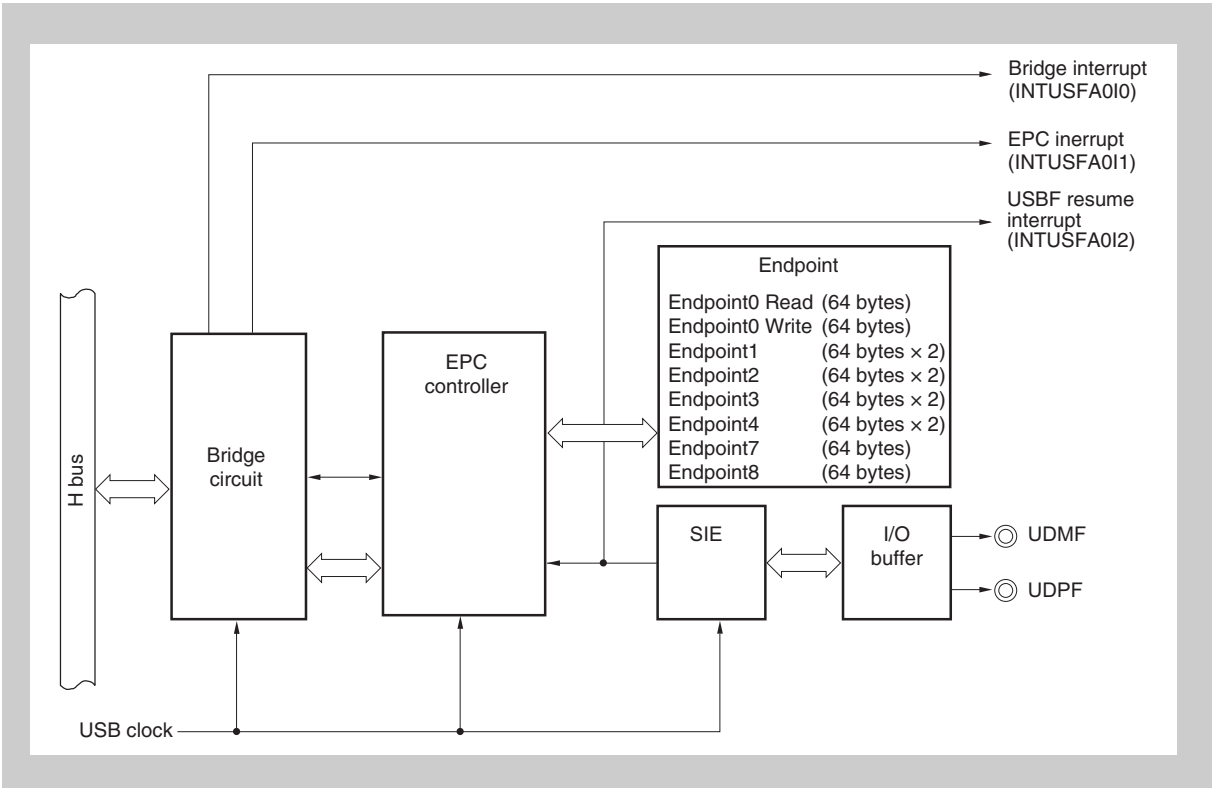


Figure 30-1 Block diagram for USB function controller

30.4 External Circuit Configuration

30.4.1 Overview

For USB transmission, when communication is performed with the host controller and function controller facing each other, pull-up/pull-down resistors must be connected to the USB signal (D+/D–) to identify the communication partner. For this microcontroller, series resistors must also be connected.

Because this microcontroller does not include these pull-up/pull-down resistors and series resistors, be sure to connect them externally.

The following shows the schematic configuration of the USB transmission line. For details about the external configuration, see the description provided in each section.

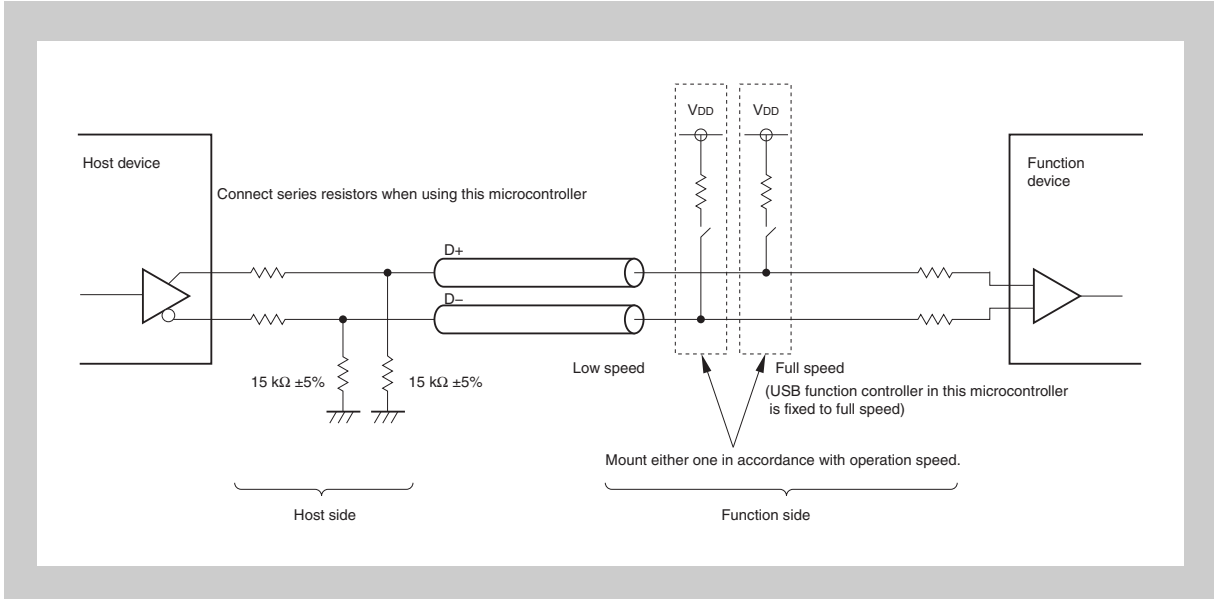


Figure 30-2 Schematic configuration of pull-up, pull-down, and series resistors in USB transmission line

30.4.2 Connection configuration

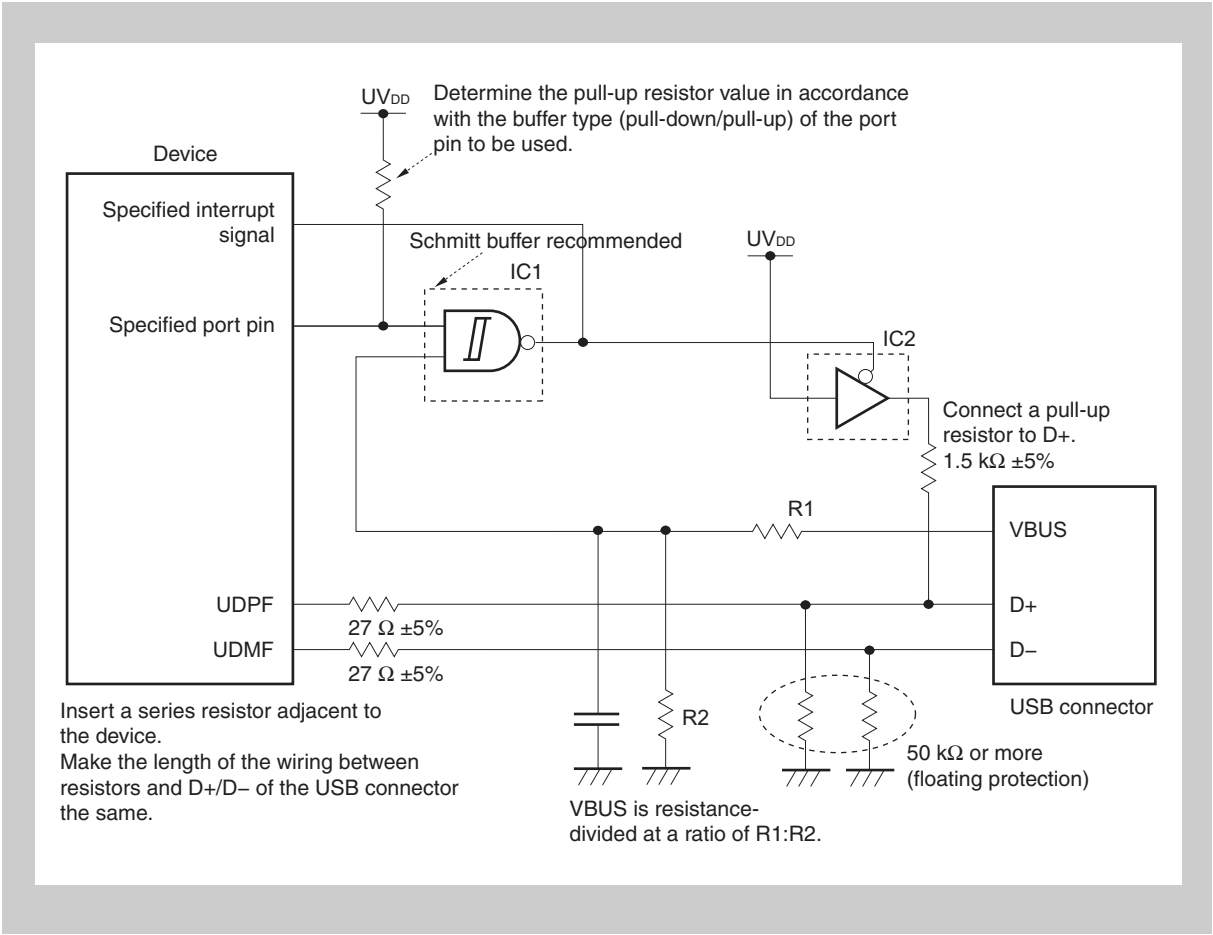


Figure 30-3 Example of USB function controller connections

(1) Series resistor connection to D+/D–

Connect series resistors of 27 Ω±5% to the D+/D– pins (UFDP, UFDM) of the USB function controller in this microcontroller. If they are not connected, the impedance rating cannot be satisfied and the output waveform might be disturbed.

Place the series resistors adjacent to the microcontroller, and make the length of the wiring between the series resistors and the USB connectors the same, to make the impedance of D+ and D– equal. (A differential with 90 Ω±5% is recommended.)

(2) Pull-up control of D+

Because the function controller of this microcontroller is fixed to full speed (FS), be sure to pull up the D+ pin (UFDP) by 1.5 kΩ ±5% to UVDD.

To disable a connection report (D+ pull up) to the USB host controller/HUB (such as during high priority servicing or initialization), control the pull-up resistor of D+ via a general-purpose port in the system. For a circuit such as the one shown in *Figure 30-3 "Example of USB function controller connections"*, control the pull-up control signal and the VBUS input signal of the D+ pin by using a general-purpose port and the USB cable VBUS (AND circuit). In *Figure 30-3 "Example of USB function controller connections"*, if the general-purpose port is low level, pulling up D+ is prohibited. For the IC2 in

Figure 30-3 “Example of USB function controller connections”, use an IC to which voltage can be applied when the system power is off.

(3) Detection of USB cable connection/disconnection

The USB function controller (USBF) requires a VBUS input signal to recognize whether the USB cable is connected or disconnected, because the state of the USBF is controlled by hardware. The voltage (5 V) from the USB host controller or HUB is applied as the VBUS input signal when the USB cable VBUS is connected to the USB host controller or HUB while the USBF power is off. For the IC2 in *Figure 30-3 “Example of USB function controller connections”*, use an IC to which voltage can be applied when the system power is off. When disconnecting the USB cable in the circuit in *Figure 30-3 “Example of USB function controller connections”*, the input signal to a specified interrupt might be unstable while the VBUS voltage is dropping. It is therefore recommended to use a Schmitt buffer for IC1 in *Figure 30-3 “Example of USB function controller connections”*.

(4) Floating protection during initialization or when USBF is unused

When the USB function controller is initialized or unused, to avoid a floating status, pull the D+/D– pins (UFDP, UFDM) down using a resistor of 50 kΩ or higher.

30.5 Cautions

<R> **(1) Clock accuracy**

When using the internal clock, use a resonator that satisfies the following accuracy:

When the system operates at 192 MHz: 9.6 MHz±500 ppm

When the system operates at 144 MHz: 7.2 MHz±100 ppm

Supply an external clock with an accuracy of 48 MHz±500 ppm or less to the UCLK pin. If the accuracy of the USB clock decreases, the transmitted data can no longer satisfy the USB Specification.

(2) Stopping the USB clock

The USB clock must be stopped when stopping the USB function controller. Stop the USB function controller, and then stop the clock supply (by clearing SFRCTL3.USBCKE2 to 0).

If the USB clock (f_{USB}) is stopped without stopping the USB function controller, malfunctions might occur due to clock noise (spike) when the clock is supplied again.

30.6 Requests

The USB standard has a request command that reports requests from the host device to the function device to execute response processing.

The requests are received in the SETUP stage of control transfer, and most can be automatically processed via the hardware of the USB function controller (USBF).

30.6.1 Automatic requests

(1) Decode

Table 30-6 "Request format" shows the request format and *Table 30-7 "Correspondence between requests and decoded values"* shows the correspondence between requests and decoded values.

Table 30-6 Request format

Offset	Field name	
0	bmRequestType	
1	bRequest	
2	wValue	Lower side
3		Higher side
4	wIndex	Lower side
5		Higher side
6	wLength	Lower side
7		Higher side

Table 30-7 Correspondence between requests and decoded values

Offset / Request	Decoded value								Response			Data stage
	bmRequestType	bRequest	wValue		wIndex		wLength		Df	Ad	Cf	
	0	1	3	2	5	4	7	6				
GET_INTERFACE	81_H	$0A_H$	00_H	00_H	00_H	$0n_H$	00_H	01_H	STALL	STALL	ACK NAK	✓
GET_CONFIGURATION	80_H	08_H	00_H	00_H	00_H	00_H	00_H	01_H	ACK NAK	ACK NAK	ACK NAK	✓
GET_DESCRIPTOR Device	80_H	06_H	01_H	00_H	00_H	00_H	XX_H	XX_H[a]	ACK NAK	ACK NAK	ACK NAK	✓
GET_DESCRIPTOR Configuration	80_H	06_H	02_H	00_H	00_H	00_H	XX_H	XX_H[a]	ACK NAK	ACK NAK	ACK NAK	✓
GET_STATUS Device	80_H	00_H	00_H	00_H	00_H	00_H	00_H	02_H	ACK NAK	ACK NAK	ACK NAK	✓
GET_STATUS Endpoint 0	82_H	00_H	00_H	00_H	00_H	00_H 80_H	00_H	02_H	ACK NAK	ACK NAK	ACK NAK	✓
GET_STATUS Endpoint X	82_H	00_H	00_H	00_H	00_H	$\$\$_H$	00_H	02_H	STALL	STALL	ACK NAK	✓
CLEAR_FEATURE Device[b]	00_H	01_H	00_H	01_H	00_H	00_H	00_H	00_H	ACK NAK	ACK NAK	ACK NAK	–
CLEAR_FEATURE Endpoint 0[b]	02_H	01_H	00_H	00_H	00_H	00_H 80_H	00_H	00_H	ACK NAK	ACK NAK	ACK NAK	–
CLEAR_FEATURE Endpoint X[b]	02_H	01_H	00_H	00_H	00_H	$\$\$_H$	00_H	00_H	STALL	STALL	ACK NAK	–
SET_FEATURE Device[c]	00_H	03_H	00_H	01_H	00_H	00_H	00_H	00_H	ACK NAK	ACK NAK	ACK NAK	–
SET_FEATURE Endpoint 0[c]	02_H	03_H	00_H	00_H	00_H	00_H 80_H	00_H	00_H	ACK NAK	ACK NAK	ACK NAK	–
SET_FEATURE Endpoint X[c]	02_H	03_H	00_H	00_H	00_H	$\$\$_H$	00_H	00_H	STALL	STALL	ACK NAK	–
SET_INTERFACE	01_H	$0B_H$	00_H	$0\#_H$	00_H	$0?_H$	00_H	00_H	STALL	STALL	ACK NAK	–
SET_CONFIGURATION[d]	00_H	09_H	00_H	00_H 01_H	00_H	00_H	00_H	00_H	ACK NAK	ACK NAK	ACK NAK	–
SET_ADDRESS	00_H	05_H	XX_H	XX_H	00_H	00_H	00_H	00_H	ACK NAK	ACK NAK	ACK NAK	–

a) If the wLength value is less than the prepared value, the value up to wLength is returned. If the wLength value is the prepared value or greater, the value up to the prepared one is returned.

b) The CLEAR_FEATURE request clears the device status register (USFA0DST) and EPn status register (USFA0EnS) (n = 0 to 4, 7) when ACK is received in the status stage.

c) The SET_FEATURE request sets the device status register (USFA0DST) and EPn status register (USFA0EnS) (n = 0 to 4, 7) when ACK is received in the status stage.
If the E0HALT bit of the USFA0E0S register is set, a STALL response is returned in the status stage or data stage during control transfer for a request other than the GET_STATUS Endpoint0 request, SET_FEATURE Endpoint0 request, and a request generated by the CPUDEC interrupt request, until the CLEAR_FEATURE Endpoint0 request is received. A STALL response to an unsupported request does not set the E0HALT bit of the USFA0E0S register to 1, and the STALL response is cleared when the next SETUP token is received.

d) A STALL response is automatically returned if the wValue value is not the value above.

Cautions

1. The sequence of control transfer defined by the Universal Serial Bus Specification is not satisfied under the conditions below. The operation is not guaranteed under these conditions.
   - If an IN/OUT token is suddenly received without a SETUP stage
   - If DATA PID1 is sent in the data phase of the SETUP stage
   - If a token of 128 addresses or more is received
   - If the request data transmitted in the SETUP stage is of less than 8 bytes
2. In the status stage, an ACK response is returned even when the host transmits data other than a Null packet.
3. If the wLength value is 00_H during a control transfer (read) of FW processing, a Null packet is automatically transmitted for a control transfer (without data). No Null packet will be automatically transmitted for FW requests.

Notes

1. Df: Default state, Ad: Addressed state, Cf: Configured state
2. ✓: Data stage
   –: No data stage
3. n = 0 to 4

   For a request with interface number 1 to 4, whether the correct response or a STALL response is returned depends on the validity of the corresponding interface number specified by the active interface number register (USFA0AIFN).
4. $$: Valid endpoint number including the transfer direction

   Whether the target Endpoint is valid is determined by the currently specified Alternative Setting number. (For details, see *(36) "Active alternative setting register (USFA0AAS)" on page 1978* and *(38) "Endpoint1 interface mapping register (USFA0E1IM)" on page 1980* through *(42) "Endpoint7 interface mapping register (USFA0E7IM)" on page 1984* in *30.7.2 "EPC control registers"*.)
5. ? and #: Value transmitted from host (information on interface numbers 0 to 4)

   For an alternate setting request corresponding to an interface number, whether the correct response or a STALL response is returned depends on the interface number and the validity of the alternate setting, which are specified by the active interface number register (USFA0AIFN) and active alternative setting register (USFA0AAS), respectively.

(2) Processing

The processing of an automatic request in the Default state, Addressed state, and Configured state is described below.

Note Default state: State in which an operation is performed with the Default address

Addressed state: State after an address has been allocated

Configured state: State after SET_CONFIGURATION wValue = 1 has been correctly received

(a) CLEAR_FEATURE() request

For the CLEAR_FEATURE() request, a STALL response is returned in the status stage if clearing has failed, the specified feature does not exist, or the request is for an interface or nonexistent endpoint. A STALL response is also returned if the wLength value is not 0.

- Default state:

  When the CLEAR_FEATURE() request is received, the correct response is returned only if the request is for a device or Endpoint0. A STALL response is returned in the status stage in all other cases.

- Addressed state:

  When the CLEAR_FEATURE() request is received, the correct response is returned only if the request is for a device or Endpoint0. A STALL response is returned in the status stage in all other cases.

- Configured state:

  When the CLEAR_FEATURE() request is received, the correct response is returned only if the request is for a device or existent endpoint. A STALL response is returned in the status stage in all other cases.

When the CLEAR_FEATURE() request is correctly processed, the corresponding bit of the CLR request register (USFA0CLR) is set to 1, the EnHALT bit of the EPn status register (USFA0EnS) is cleared to 0, and an interrupt is issued (n = 0 to 4, 7, 8). If the CLEAR_FEATURE() request is received when the subject is an endpoint, the toggle bit (that controls switching between DATA0 and DATA1) of the corresponding endpoint is always re-set to DATA0.

(b) GET_CONFIGURATION() request

A STALL response is returned in the data stage if wValue, wIndex, or wLength is not the value shown in *Table 30-7 "Correspondence between requests and decoded values"*.

- Default state:

  The value stored in the configuration register (USFA0CNF) is returned when the GET_CONFIGURATION() request is received.

- Addressed state:

  The value stored in the USFA0CNF register is returned when the GET_CONFIGURATION() request is received.

- Configured state:

  The value stored in the USFA0CNF register is returned when the GET_CONFIGURATION() request is received.

(c) GET_DESCRIPTOR() request

If the subject descriptor has a length that is a multiple of wMaxPacketSize, a Null packet is returned to indicate the end of the data stage. If the length of the descriptor at this time is less than the wLength value, the entire descriptor is returned. If the length of the descriptor is the wLength value or greater, the descriptor up to the wLength value is returned.

- Default state:

  The values stored in the device descriptor register n (USFA0DDn) and configuration/interface/endpoint descriptor register m (USFA0CIEm) are returned (n = 0 to 17, m = 0 to 255) when the GET_DESCRIPTOR() request is received.

- Addressed state:

  The values stored in the USFA0DDn register and USFA0CIEm register are returned when the GET_DESCRIPTOR() request is received.

- Configured state:

  The values stored in the USFA0DDn register and USFA0CIEm register are returned when the GET_DESCRIPTOR() request is received.

A descriptor of up to 256 bytes can be stored in the USFA0CIEm register. To transmit a descriptor of more than 256 bytes, set the CDCGDST bit of the USFA0MODC register and process the GET_DESCRIPTOR() request using FW.

Store the value of (the total number of bytes of the descriptor specified by the USFA0CIEm register 1) in the descriptor length register (USFA0DSCL). The transfer data is controlled by the value of this data + 1 and wLength.

(d) GET_INTERFACE() request

If wValue or wLength is not the value shown in *Table 30-7 "Correspondence between requests and decoded values"*, or if wIndex is not the value specified by the active interface number register (USFA0AIFN), a STALL response is returned in the data stage.

- Default state:

  A STALL response is returned in the data stage when the GET_INTERFACE() request is received.

- Addressed state:

  A STALL response is returned in the data stage when the GET_INTERFACE() request is received.

- Configured state:

  The value stored in the interface n register (USFA0IFn, n = 0 to 4) corresponding to the wIndex value is returned when the GET_INTERFACE() request is received.

(e) GET_STATUS() request

A STALL response is returned in the data stage if wValue, wIndex, or wLength is not the value shown in *Table 30-7 "Correspondence between requests and decoded values"*. A STALL response is also returned in the data stage if the request is for an interface or nonexistent endpoint.

- Default state:

  When the GET_STATUS() request is received, the value stored in the target status register[a] is returned only if the request is for a device or Endpoint0. A STALL response is returned in the data stage in all other cases.

- Addressed state:

  When the GET_STATUS() request is received, the value stored in the target status register[a] is returned only if the request is for a device or Endpoint0. A STALL response is returned in the data stage in all other cases.

- Configured state:

  When the GET_STATUS() request is received, the value stored in the target status register[a] is returned only if the request is for a device or existent endpoint. A STALL response is returned in the data stage in all other cases.

a) The target status register is as follows.

- If the target is a device: Device status register (USFA0DST)
- If the target is Endpoint 0: EP0 status register (USFA0E0S)
- If the target is Endpoint n: EPn status register (USFA0EnS) (n = 1 to 4, 7, 8)

(f) SET_ADDRESS() request

A STALL response is returned in the status stage if wIndex or wLength is not the value shown in *Table 30-7 "Correspondence between requests and decoded values"*. A STALL response is returned if the specified device address is greater than 127.

- Default state:

  If the specified address is not 0 when the SET_ADDRESS() request is received, the device enters the Addressed state and the USB Address value to input to SIE is changed to a specified address value. If the specified address is 0, the device remains in the Default state.

- Addressed state:

  If the specified address is 0 when the SET_ADDRESS() request is received, the device enters the Default state and the USB Address value to input to SIE is returned to the default address. If the specified address is not 0, the device remains in the Addressed state, and the USB Address value to input to SIE is changed to a specified new address value.

- Configured state:

  The device remains in the Configured state and returns the USB Address value to be input to SIE to the default address if the specified address is 0 when the SET_ADDRESS() request has been received. In this case, the endpoints other than endpoint 0 remain valid, and control transfers (IN or OUT), bulk transfers, and interrupt transfers for an endpoint other than endpoint 0 are also acknowledged. If the specified address is not 0, the device remains in the Configured state and changes the USB Address value to be input to SIE to a specified new address value.

(g) SET_CONFIGURATION() request

A STALL response is made in the data stage if wValue, wIndex, or wLength is not the value shown in *Table 30-7 "Correspondence between requests and decoded values"*.

- Default state:

  The CONF bit of the mode status register (USFA0MODS) and the configuration register (USFA0CNF) are set to 1 if the specified configuration value is 1 when the SET_CONFIGURATION() request has been received. If the specified configuration value is 0, the CONF bit of the USFA0MODS register and USFA0CNF register are cleared to 0. In other words, the device skips the Addressed state and moves to the Configured state in which it responds to the Default address.

- Addressed state:

  The CONF bit of the USFA0MODS register and USFA0CNF register are set to 1 and the device enters the Configured state if the specified configuration value is 1 when the SET_CONFIGURATION() request has been received. If the specified configuration value is 0, the device remains in the Addressed state.

- Configured state:

  The CONF bit of the USFA0MODS register and USFA0CNF register are set to 1 and the device returns to the Addressed state if the specified configuration value is 0 when the SET_CONFIGURATION() request has been received. If the specified configuration value is 1, the device remains in the Configured state.

If the SET_CONFIGURATION() request has been correctly processed, the target bit of the SET request register (USFA0SET) is set to 1, and an interrupt is issued. All Halt Features are cleared after the SET_CONFIGURATION() request has been completed even if the specified configuration value is the same as the current configuration value. If the SET_CONFIGURATION() request has been correctly processed, the data toggle of all endpoints is always initialized to DATA0 again (it is defined that the default status, Alternative Setting 0, is set from when the SET_CONFIGURATION request is received to when the SET_INTERFACE request is received).

(h) SET_FEATURE() request

A STALL response is returned in the status stage if setting has failed, the specified feature does not exist, or the request is for an interface or nonexistent endpoint. A STALL response is also returned if the wLength value is not 0.

- Default state:

  When the SET_FEATURE() request is received, the correct response is returned only if the request is for a device or Endpoint0. A STALL response is returned in the status stage in all other cases.

- Addressed state:

  When the SET_FEATURE() request is received, the correct response is returned only if the request is for a device or Endpoint0. A STALL response is returned in the status stage in all other cases.

- Configured state:

  When the SET_FEATURE() request is received, the correct response is returned only if the request is for a device or existent endpoint. A STALL response is returned in the status stage in all other cases.

When the SET_FEATURE() request is correctly processed, the corresponding bit of the SET request register (USFA0SET) and the EnHALT bit of the EPn status register (USFA0EnS) are set to 0, and an interrupt is issued (n = 0 to 4, 7, 8).

(i) SET_INTERFACE() request

If wLength is not the value shown in *Table 30-7 "Correspondence between requests and decoded values"*, wIndex is not the value specified by the active interface number register (USFA0AIFN), or wValue is not the value specified by the active alternative setting register (USFA0AAS), a STALL response is returned in the status stage.

- Default state:

  A STALL response is returned in the status stage when the SET_INTERFACE() request is received.

- Addressed state:

  A STALL response is returned in the status stage when the SET_INTERFACE() request is received.

- Configured state:

  A Null packet is transmitted in the status stage when the SET_INTERFACE() request is received.

When the SET_INTERFACE() request is correctly processed, an interrupt is issued. All the Halt Features of the endpoint linked to the target interface are cleared after the SET_INTERFACE() request has been cleared. The data toggle of all the endpoints related to the target interface number is always reinitialized to DATA0. When the currently selected Alternative Setting is to be changed by correctly processing the SET_INTERFACE() request, the FIFO of the endpoint that is affected is completely cleared, and all the related interrupt sources are also initialized.

When the SET_INTERFACE() request is completed, the FIFO of all the endpoints linked to the target interface are cleared. At the same time, Halt Feature and Data PID are initialized, and the related INT status n register (USFA0ISn) is cleared to 0 (n = 0 to 4). (Only Halt Feature and Data PID are initialized when the SET_CONFIGURATION request is completed.)

If the target endpoint is not supported by the SET_INTERFACE() request during a DMA transfer, the DMA request signal is immediately deasserted, and the FIFO of the endpoint that was linked when the SET_INTERFACE() request was completed is completely cleared. As a result of clearing the FIFO, transferring data by way of DMA is not correctly processed.

30.6.2 Other requests

(1) Response and processing

The following table shows how other requests are responded to and processed.

Table 30-8 Response and processing of other requests

Request	Response and processing
GET_DESCRIPTOR String	Generation of CPUDEC interrupt request
GET_STATUS Interface	Automatic STALL response
CLEAR_FEATURE Interface	Automatic STALL response
SET_FEATURE Interface	Automatic STALL response
all SET_DESCRIPTOR	Generation of CPUDEC interrupt request
All other requests	Generation of CPUDEC interrupt request

30.7 Register Configuration

30.7.1 USB function controller registers

(1) EPC control registers

Table 30-9 EPC control registers (1/2)

Offset address	Register name	Symbol	R/W	Manipulatable bit unit				Initial value
				1	8	16	32	
000_H	EP0NAUSFA0K register	USFA0E0N	R/W				✓	$0000\ 0000_H$
004_H	EP0NAKALL register	USFA0E0NA	R/W				✓	$0000\ 0000_H$
008_H	EPNAK register	USFA0EN	R/W				✓	$0000\ 0000_H$
$00C_H$	EPNAK mask register	USFA0ENM	R/W				✓	$0000\ 0000_H$
010_H	SNDSIE register	USFA0SDS	R/W				✓	$0000\ 0000_H$
014_H	CLR request register	USFA0CLR	R				✓	$0000\ 0000_H$
018_H	SET request register	USFA0SET	R				✓	$0000\ 0000_H$
$01C_H$	EP status 0 register	USFA0EPS0	R				✓	$0000\ 0000_H$
020_H	EP status 1 register	USFA0EPS1	R				✓	$0000\ 0000_H$
024_H	EP status 2 register	USFA0EPS2	R				✓	$0000\ 0000_H$
040_H	INT status 0 register	USFA0IS0	R				✓	$0000\ 0000_H$
044_H	INT status 1 register	USFA0IS1	R				✓	$0000\ 0000_H$
048_H	INT status 2 register	USFA0IS2	R				✓	$0000\ 0000_H$
$04C_H$	INT status 3 register	USFA0IS3	R				✓	$0000\ 0000_H$
050_H	INT status 4 register	USFA0IS4	R				✓	$0000\ 0000_H$
$05C_H$	INT mask 0 register	USFA0IM0	R/W				✓	$0000\ 0000_H$
060_H	INT mask 1 register	USFA0IM1	R/W				✓	$0000\ 0000_H$
064_H	INT mask 2 register	USFA0IM2	R/W				✓	$0000\ 0000_H$
068_H	INT mask 3 register	USFA0IM3	R/W				✓	$0000\ 0000_H$
$06C_H$	INT mask 4 register	USFA0IM4	R/W				✓	$0000\ 0000_H$
078_H	INT clear 0 register	USFA0IC0	W				✓	$FFFF\ FFFF_H$
$07C_H$	INT clear 1 register	USFA0IC1	W				✓	$FFFF\ FFFF_H$
080_H	INT clear 2 register	USFA0IC2	W				✓	$FFFF\ FFFF_H$
084_H	INT clear 3 register	USFA0IC3	W				✓	$FFFF\ FFFF_H$
088_H	INT clear 4 register	USFA0IC4	W				✓	$FFFF\ FFFF_H$
098_H	INT & DMARQ register	USFA0IDR	R/W				✓	$0000\ 0000_H$
$09C_H$	DMA status 0 register	USFA0DMS0	R				✓	$0000\ 0000_H$
$0A0_H$	DMA status 1 register	USFA0DMS1	R				✓	$0000\ 0000_H$
$0C0_H$	FIFO clear 0 register	USFA0FIC0	W				✓	$0000\ 0000_H$
$0C4_H$	FIFO clear 1 register	USFA0FIC1	W				✓	$0000\ 0000_H$
$0D4_H$	Data end register	USFA0DEND	R/W				✓	$0000\ 0000_H$
$0DC_H$	GPR register	USFA0GPR	W				✓	$0000\ 0000_H$
$0E8_H$	Mode control register	USFA0MODC	R/W				✓	$0000\ 0000_H$
$0F0_H$	Mode status register	USFA0MODS	R				✓	$0000\ 0000_H$
100_H	Active interface number register	USFA0AIFN	R/W				✓	$0000\ 0000_H$

Table 30-9 EPC control registers (2/2)

Offset address	Register name	Symbol	R/W	Manipulatable bit unit				Initial value
				1	8	16	32	
104_H	Active alternative setting register	USFA0AAS	R/W				✓	0000 0000_H
108_H	Alternative setting status register	USFA0ASS	R				✓	0000 0000_H
$10C_H$	Endpoint 1 interface mapping register	USFA0E1IM	R/W				✓	0000 0000_H
110_H	Endpoint 2 interface mapping register	USFA0E2IM	R/W				✓	0000 0000_H
114_H	Endpoint 3 interface mapping register	USFA0E3IM	R/W				✓	0000 0000_H
118_H	Endpoint 4 interface mapping register	USFA0E4IM	R/W				✓	0000 0000_H
124_H	Endpoint 7 interface mapping register	USFA0E7IM	R/W				✓	0000 0000_H
128_H	Endpoint 8 interface mapping register	USFA0E8IM	R/W				✓	0000 0000_H

(2) EPC data hold registers

Table 30-10 EPC data hold registers

Offset address	Register name	Symbol	R/W	Manipulatable bit unit				Initial value
				1	8	16	32	
200_H	EP0 read register	USFA0E0R	R				✓	Undefined
204_H	EP0 length register	USFA0E0L	R				✓	0000 0000_H
208_H	EP0 setup register	USFA0E0ST	R				✓	0000 0000_H
$20C_H$	EP0 write register	USFA0E0W	W				✓	Undefined
210_H	Bulk-out 1 register	USFA0BO1	R				✓	Undefined
214_H	Bulk-out 1 length register	USFA0BO1L	R				✓	0000 0000_H
218_H	Bulk-out 2 register	USFA0BO2	R				✓	Undefined
$21C_H$	Bulk-out 2 length register	USFA0BO2L	R				✓	0000 0000_H
220_H	Bulk-in 1 register	USFA0BI1	W				✓	Undefined
224_H	Bulk-in 2 register	USFA0BI2	W				✓	Undefined
228_H	Interrupt 1 register	USFA0INT1	W				✓	Undefined
$22C_H$	Interrupt 2 register	USFA0INT2	W				✓	Undefined

(3) EPC request data registers

Table 30-11 EPC request data registers (1/12)

Offset address	Register name	Symbol	R/W	Manipulatable bit unit				Initial value
				1	8	16	32	
288_H	Device status register	USFA0DST	R/W				✓	0000 000000_H
298_H	EP0 status register	USFA0E0S	R/W				✓	0000 000000_H
$2A0_H$	EP1 status register	USFA0E1S	R/W				✓	0000 000000_H
$2A8_H$	EP2 status register	USFA0E2S	R/W				✓	0000 000000_H
$2B0_H$	EP3 status register	USFA0E3S	R/W				✓	0000 000000_H
$2B8_H$	EP4 status register	USFA0E4S	R/W				✓	0000 000000_H
$2D0_H$	EP7 status register	USFA0E7S	R/W				✓	0000 000000_H
$2D8_H$	EP8 status register	USFA0E8S	R/W				✓	0000 000000_H
300_H	Address register	USFA0ADRS	R				✓	0000 000000_H
304_H	Configuration register	USFA0CNF	R				✓	0000 000000_H
308_H	Interface 0 register	USFA0IF0	R				✓	0000 000000_H
$30C_H$	Interface 1 register	USFA0IF1	R				✓	0000 000000_H
310_H	Interface 2 register	USFA0IF2	R				✓	0000 000000_H
314_H	Interface 3 register	USFA0IF3	R				✓	0000 000000_H
318_H	Interface 4 register	USFA0IF4	R				✓	0000 000000_H
340_H	Descriptor length register	USFA0DSCL	R/W				✓	0000 000000_H
344_H	Device descriptor register 0	USFA0DD0	R/W				✓	Undefined
348_H	Device descriptor register 1	USFA0DD1	R/W				✓	Undefined
$34C_H$	Device descriptor register 2	USFA0DD2	R/W				✓	Undefined
350_H	Device descriptor register 3	USFA0DD3	R/W				✓	Undefined
354_H	Device descriptor register 4	USFA0DD4	R/W				✓	Undefined
358_H	Device descriptor register 5	USFA0DD5	R/W				✓	Undefined
$35C_H$	Device descriptor register 6	USFA0DD6	R/W				✓	Undefined
360_H	Device descriptor register 7	USFA0DD7	R/W				✓	Undefined
364_H	Device descriptor register 8	USFA0DD8	R/W				✓	Undefined
368_H	Device descriptor register 9	USFA0DD9	R/W				✓	Undefined
$36C_H$	Device descriptor register 10	USFA0DD10	R/W				✓	Undefined
370_H	Device descriptor register 11	USFA0DD11	R/W				✓	Undefined
374_H	Device descriptor register 12	USFA0DD12	R/W				✓	Undefined
378_H	Device descriptor register 13	USFA0DD13	R/W				✓	Undefined
$37C_H$	Device descriptor register 14	USFA0DD14	R/W				✓	Undefined
380_H	Device descriptor register 15	USFA0DD15	R/W				✓	Undefined
384_H	Device descriptor register 16	USFA0DD16	R/W				✓	Undefined
388_H	Device descriptor register 17	USFA0DD17	R/W				✓	Undefined
$38C_H$	Configuration/interface/endpoint descriptor register 0	USFA0CIE0	R/W				✓	Undefined
390_H	Configuration/interface/endpoint descriptor register 1	USFA0CIE1	R/W				✓	Undefined
394_H	Configuration/interface/endpoint descriptor register 2	USFA0CIE2	R/W				✓	Undefined
398_H	Configuration/interface/endpoint descriptor register 3	USFA0CIE3	R/W				✓	Undefined

Table 30-11 EPC request data registers (2/12)

Offset address	Register name	Symbol	R/W	Manipulatable bit unit				Initial value
				1	8	16	32	
$39C_H$	Configuration/interface/endpoint descriptor register 4	USFA0CIE4	R/W				✓	Undefined
$3A0_H$	Configuration/interface/endpoint descriptor register 5	USFA0CIE5	R/W				✓	Undefined
$3A4_H$	Configuration/interface/endpoint descriptor register 6	USFA0CIE6	R/W				✓	Undefined
$3A8_H$	Configuration/interface/endpoint descriptor register 7	USFA0CIE7	R/W				✓	Undefined
$3AC_H$	Configuration/interface/endpoint descriptor register 8	USFA0CIE8	R/W				✓	Undefined
$3B0_H$	Configuration/interface/endpoint descriptor register 9	USFA0CIE9	R/W				✓	Undefined
$3B4_H$	Configuration/interface/endpoint descriptor register 10	USFA0CIE10	R/W				✓	Undefined
$3B8_H$	Configuration/interface/endpoint descriptor register 11	USFA0CIE11	R/W				✓	Undefined
$3BC_H$	Configuration/interface/endpoint descriptor register 12	USFA0CIE12	R/W				✓	Undefined
$3C0_H$	Configuration/interface/endpoint descriptor register 13	USFA0CIE13	R/W				✓	Undefined
$3C4_H$	Configuration/interface/endpoint descriptor register 14	USFA0CIE14	R/W				✓	Undefined
$3C8_H$	Configuration/interface/endpoint descriptor register 15	USFA0CIE15	R/W				✓	Undefined
$3CC_H$	Configuration/interface/endpoint descriptor register 16	USFA0CIE16	R/W				✓	Undefined
$3D0_H$	Configuration/interface/endpoint descriptor register 17	USFA0CIE17	R/W				✓	Undefined
$3D4_H$	Configuration/interface/endpoint descriptor register 18	USFA0CIE18	R/W				✓	Undefined
$3D8_H$	Configuration/interface/endpoint descriptor register 19	USFA0CIE19	R/W				✓	Undefined
$3DC_H$	Configuration/interface/endpoint descriptor register 20	USFA0CIE20	R/W				✓	Undefined
$3E0_H$	Configuration/interface/endpoint descriptor register 21	USFA0CIE21	R/W				✓	Undefined
$3E4_H$	Configuration/interface/endpoint descriptor register 22	USFA0CIE22	R/W				✓	Undefined
$3E8_H$	Configuration/interface/endpoint descriptor register 23	USFA0CIE23	R/W				✓	Undefined
$3EC_H$	Configuration/interface/endpoint descriptor register 24	USFA0CIE24	R/W				✓	Undefined
$3F0_H$	Configuration/interface/endpoint descriptor register 25	USFA0CIE25	R/W				✓	Undefined
$3F4_H$	Configuration/interface/endpoint descriptor register 26	USFA0CIE26	R/W				✓	Undefined
$3F8_H$	Configuration/interface/endpoint descriptor register 27	USFA0CIE27	R/W				✓	Undefined

Table 30-11 EPC request data registers (3/12)

Offset address	Register name	Symbol	R/W	Manipulatable bit unit				Initial value
				1	8	16	32	
$3FC_H$	Configuration/interface/endpoint descriptor register 28	USFA0CIE28	R/W				✓	Undefined
400_H	Configuration/interface/endpoint descriptor register 29	USFA0CIE29	R/W				✓	Undefined
404_H	Configuration/interface/endpoint descriptor register 30	USFA0CIE30	R/W				✓	Undefined
408_H	Configuration/interface/endpoint descriptor register 31	USFA0CIE31	R/W				✓	Undefined
$40C_H$	Configuration/interface/endpoint descriptor register 32	USFA0CIE32	R/W				✓	Undefined
410_H	Configuration/interface/endpoint descriptor register 33	USFA0CIE33	R/W				✓	Undefined
414_H	Configuration/interface/endpoint descriptor register 34	USFA0CIE34	R/W				✓	Undefined
418_H	Configuration/interface/endpoint descriptor register 35	USFA0CIE35	R/W				✓	Undefined
$41C_H$	Configuration/interface/endpoint descriptor register 36	USFA0CIE36	R/W				✓	Undefined
420_H	Configuration/interface/endpoint descriptor register 37	USFA0CIE37	R/W				✓	Undefined
424_H	Configuration/interface/endpoint descriptor register 38	USFA0CIE38	R/W				✓	Undefined
428_H	Configuration/interface/endpoint descriptor register 39	USFA0CIE39	R/W				✓	Undefined
$42C_H$	Configuration/interface/endpoint descriptor register 40	USFA0CIE40	R/W				✓	Undefined
430_H	Configuration/interface/endpoint descriptor register 41	USFA0CIE41	R/W				✓	Undefined
434_H	Configuration/interface/endpoint descriptor register 42	USFA0CIE42	R/W				✓	Undefined
438_H	Configuration/interface/endpoint descriptor register 43	USFA0CIE43	R/W				✓	Undefined
$43C_H$	Configuration/interface/endpoint descriptor register 44	USFA0CIE44	R/W				✓	Undefined
440_H	Configuration/interface/endpoint descriptor register 45	USFA0CIE45	R/W				✓	Undefined
444_H	Configuration/interface/endpoint descriptor register 46	USFA0CIE46	R/W				✓	Undefined
448_H	Configuration/interface/endpoint descriptor register 47	USFA0CIE47	R/W				✓	Undefined
$44C_H$	Configuration/interface/endpoint descriptor register 48	USFA0CIE48	R/W				✓	Undefined
450_H	Configuration/interface/endpoint descriptor register 49	USFA0CIE49	R/W				✓	Undefined
454_H	Configuration/interface/endpoint descriptor register 50	USFA0CIE50	R/W				✓	Undefined
458_H	Configuration/interface/endpoint descriptor register 51	USFA0CIE51	R/W				✓	Undefined

Table 30-11 EPC request data registers (4/12)

Offset address	Register name	Symbol	R/W	Manipulatable bit unit				Initial value
				1	8	16	32	
$45C_H$	Configuration/interface/endpoint descriptor register 52	USFA0CIE52	R/W				✓	Undefined
460_H	Configuration/interface/endpoint descriptor register 53	USFA0CIE53	R/W				✓	Undefined
464_H	Configuration/interface/endpoint descriptor register 54	USFA0CIE54	R/W				✓	Undefined
468_H	Configuration/interface/endpoint descriptor register 55	USFA0CIE55	R/W				✓	Undefined
$46C_H$	Configuration/interface/endpoint descriptor register 56	USFA0CIE56	R/W				✓	Undefined
470_H	Configuration/interface/endpoint descriptor register 57	USFA0CIE57	R/W				✓	Undefined
474_H	Configuration/interface/endpoint descriptor register 58	USFA0CIE58	R/W				✓	Undefined
478_H	Configuration/interface/endpoint descriptor register 59	USFA0CIE59	R/W				✓	Undefined
$47C_H$	Configuration/interface/endpoint descriptor register 60	USFA0CIE60	R/W				✓	Undefined
480_H	Configuration/interface/endpoint descriptor register 61	USFA0CIE61	R/W				✓	Undefined
484_H	Configuration/interface/endpoint descriptor register 62	USFA0CIE62	R/W				✓	Undefined
488_H	Configuration/interface/endpoint descriptor register 63	USFA0CIE63	R/W				✓	Undefined
$48C_H$	Configuration/interface/endpoint descriptor register 64	USFA0CIE64	R/W				✓	Undefined
490_H	Configuration/interface/endpoint descriptor register 65	USFA0CIE65	R/W				✓	Undefined
494_H	Configuration/interface/endpoint descriptor register 66	USFA0CIE66	R/W				✓	Undefined
498_H	Configuration/interface/endpoint descriptor register 67	USFA0CIE67	R/W				✓	Undefined
$49C_H$	Configuration/interface/endpoint descriptor register 68	USFA0CIE68	R/W				✓	Undefined
$4A0_H$	Configuration/interface/endpoint descriptor register 69	USFA0CIE69	R/W				✓	Undefined
$4A4_H$	Configuration/interface/endpoint descriptor register 70	USFA0CIE70	R/W				✓	Undefined
$4A8_H$	Configuration/interface/endpoint descriptor register 71	USFA0CIE71	R/W				✓	Undefined
$4AC_H$	Configuration/interface/endpoint descriptor register 72	USFA0CIE72	R/W				✓	Undefined
$4B0_H$	Configuration/interface/endpoint descriptor register 73	USFA0CIE73	R/W				✓	Undefined
$4B4_H$	Configuration/interface/endpoint descriptor register 74	USFA0CIE74	R/W				✓	Undefined
$4B8_H$	Configuration/interface/endpoint descriptor register 75	USFA0CIE75	R/W				✓	Undefined

Table 30-11 EPC request data registers (5/12)

Offset address	Register name	Symbol	R/W	Manipulatable bit unit				Initial value
				1	8	16	32	
$4BC_H$	Configuration/interface/endpoint descriptor register 76	USFA0CIE76	R/W				✓	Undefined
$4C0_H$	Configuration/interface/endpoint descriptor register 77	USFA0CIE77	R/W				✓	Undefined
$4C4_H$	Configuration/interface/endpoint descriptor register 78	USFA0CIE78	R/W				✓	Undefined
$4C8_H$	Configuration/interface/endpoint descriptor register 79	USFA0CIE79	R/W				✓	Undefined
$4CC_H$	Configuration/interface/endpoint descriptor register 80	USFA0CIE80	R/W				✓	Undefined
$4D0_H$	Configuration/interface/endpoint descriptor register 81	USFA0CIE81	R/W				✓	Undefined
$4D4_H$	Configuration/interface/endpoint descriptor register 82	USFA0CIE82	R/W				✓	Undefined
$4D8_H$	Configuration/interface/endpoint descriptor register 83	USFA0CIE83	R/W				✓	Undefined
$4DC_H$	Configuration/interface/endpoint descriptor register 84	USFA0CIE84	R/W				✓	Undefined
$4E0_H$	Configuration/interface/endpoint descriptor register 85	USFA0CIE85	R/W				✓	Undefined
$4E4_H$	Configuration/interface/endpoint descriptor register 86	USFA0CIE86	R/W				✓	Undefined
$4E8_H$	Configuration/interface/endpoint descriptor register 87	USFA0CIE87	R/W				✓	Undefined
$4EC_H$	Configuration/interface/endpoint descriptor register 88	USFA0CIE88	R/W				✓	Undefined
$4F0_H$	Configuration/interface/endpoint descriptor register 89	USFA0CIE89	R/W				✓	Undefined
$4F4_H$	Configuration/interface/endpoint descriptor register 90	USFA0CIE90	R/W				✓	Undefined
$4F8_H$	Configuration/interface/endpoint descriptor register 91	USFA0CIE91	R/W				✓	Undefined
$4FC_H$	Configuration/interface/endpoint descriptor register 92	USFA0CIE92	R/W				✓	Undefined
500_H	Configuration/interface/endpoint descriptor register 93	USFA0CIE93	R/W				✓	Undefined
504_H	Configuration/interface/endpoint descriptor register 94	USFA0CIE94	R/W				✓	Undefined
508_H	Configuration/interface/endpoint descriptor register 95	USFA0CIE95	R/W				✓	Undefined
$50C_H$	Configuration/interface/endpoint descriptor register 96	USFA0CIE96	R/W				✓	Undefined
510_H	Configuration/interface/endpoint descriptor register 97	USFA0CIE97	R/W				✓	Undefined
514_H	Configuration/interface/endpoint descriptor register 98	USFA0CIE98	R/W				✓	Undefined
518_H	Configuration/interface/endpoint descriptor register 99	USFA0CIE99	R/W				✓	Undefined

Table 30-11 EPC request data registers (6/12)

Offset address	Register name	Symbol	R/W	Manipulatable bit unit				Initial value
				1	8	16	32	
$51C_H$	Configuration/interface/endpoint descriptor register 100	USFA0CIE100	R/W				✓	Undefined
520_H	Configuration/interface/endpoint descriptor register 101	USFA0CIE101	R/W				✓	Undefined
524_H	Configuration/interface/endpoint descriptor register 102	USFA0CIE102	R/W				✓	Undefined
528_H	Configuration/interface/endpoint descriptor register 103	USFA0CIE103	R/W				✓	Undefined
$52C_H$	Configuration/interface/endpoint descriptor register 104	USFA0CIE104	R/W				✓	Undefined
530_H	Configuration/interface/endpoint descriptor register 105	USFA0CIE105	R/W				✓	Undefined
534_H	Configuration/interface/endpoint descriptor register 106	USFA0CIE106	R/W				✓	Undefined
538_H	Configuration/interface/endpoint descriptor register 107	USFA0CIE107	R/W				✓	Undefined
$53C_H$	Configuration/interface/endpoint descriptor register 108	USFA0CIE108	R/W				✓	Undefined
540_H	Configuration/interface/endpoint descriptor register 109	USFA0CIE109	R/W				✓	Undefined
544_H	Configuration/interface/endpoint descriptor register 110	USFA0CIE110	R/W				✓	Undefined
548_H	Configuration/interface/endpoint descriptor register 111	USFA0CIE111	R/W				✓	Undefined
$54C_H$	Configuration/interface/endpoint descriptor register 112	USFA0CIE112	R/W				✓	Undefined
550_H	Configuration/interface/endpoint descriptor register 113	USFA0CIE113	R/W				✓	Undefined
554_H	Configuration/interface/endpoint descriptor register 114	USFA0CIE114	R/W				✓	Undefined
558_H	Configuration/interface/endpoint descriptor register 115	USFA0CIE115	R/W				✓	Undefined
$55C_H$	Configuration/interface/endpoint descriptor register 116	USFA0CIE116	R/W				✓	Undefined
560_H	Configuration/interface/endpoint descriptor register 117	USFA0CIE117	R/W				✓	Undefined
564_H	Configuration/interface/endpoint descriptor register 118	USFA0CIE118	R/W				✓	Undefined
568_H	Configuration/interface/endpoint descriptor register 119	USFA0CIE119	R/W				✓	Undefined
$56C_H$	Configuration/interface/endpoint descriptor register 120	USFA0CIE120	R/W				✓	Undefined
570_H	Configuration/interface/endpoint descriptor register 121	USFA0CIE121	R/W				✓	Undefined
574_H	Configuration/interface/endpoint descriptor register 122	USFA0CIE122	R/W				✓	Undefined
578_H	Configuration/interface/endpoint descriptor register 123	USFA0CIE123	R/W				✓	Undefined

Table 30-11 EPC request data registers (7/12)

Offset address	Register name	Symbol	R/W	Manipulatable bit unit				Initial value
				1	8	16	32	
$57C_H$	Configuration/interface/endpoint descriptor register 124	USFA0CIE124	R/W				✓	Undefined
580_H	Configuration/interface/endpoint descriptor register 125	USFA0CIE125	R/W				✓	Undefined
584_H	Configuration/interface/endpoint descriptor register 126	USFA0CIE126	R/W				✓	Undefined
588_H	Configuration/interface/endpoint descriptor register 127	USFA0CIE127	R/W				✓	Undefined
$58C_H$	Configuration/interface/endpoint descriptor register 128	USFA0CIE128	R/W				✓	Undefined
590_H	Configuration/interface/endpoint descriptor register 129	USFA0CIE129	R/W				✓	Undefined
594_H	Configuration/interface/endpoint descriptor register 130	USFA0CIE130	R/W				✓	Undefined
598_H	Configuration/interface/endpoint descriptor register 131	USFA0CIE131	R/W				✓	Undefined
$59C_H$	Configuration/interface/endpoint descriptor register 132	USFA0CIE132	R/W				✓	Undefined
$5A0_H$	Configuration/interface/endpoint descriptor register 133	USFA0CIE133	R/W				✓	Undefined
$5A4_H$	Configuration/interface/endpoint descriptor register 134	USFA0CIE134	R/W				✓	Undefined
$5A8_H$	Configuration/interface/endpoint descriptor register 135	USFA0CIE135	R/W				✓	Undefined
$5AC_H$	Configuration/interface/endpoint descriptor register 136	USFA0CIE136	R/W				✓	Undefined
$5B0_H$	Configuration/interface/endpoint descriptor register 137	USFA0CIE137	R/W				✓	Undefined
$5B4_H$	Configuration/interface/endpoint descriptor register 138	USFA0CIE138	R/W				✓	Undefined
$5B8_H$	Configuration/interface/endpoint descriptor register 139	USFA0CIE139	R/W				✓	Undefined
$5BC_H$	Configuration/interface/endpoint descriptor register 140	USFA0CIE140	R/W				✓	Undefined
$5C0_H$	Configuration/interface/endpoint descriptor register 141	USFA0CIE141	R/W				✓	Undefined
$5C4_H$	Configuration/interface/endpoint descriptor register 142	USFA0CIE142	R/W				✓	Undefined
$5C8_H$	Configuration/interface/endpoint descriptor register 143	USFA0CIE143	R/W				✓	Undefined
$5CC_H$	Configuration/interface/endpoint descriptor register 144	USFA0CIE144	R/W				✓	Undefined
$5D0_H$	Configuration/interface/endpoint descriptor register 145	USFA0CIE145	R/W				✓	Undefined
$5D4_H$	Configuration/interface/endpoint descriptor register 146	USFA0CIE146	R/W				✓	Undefined
$5D8_H$	Configuration/interface/endpoint descriptor register 147	USFA0CIE147	R/W				✓	Undefined

Table 30-11 EPC request data registers (8/12)

Offset address	Register name	Symbol	R/W	Manipulatable bit unit				Initial value
				1	8	16	32	
$5DC_H$	Configuration/interface/endpoint descriptor register 148	USFA0CIE148	R/W				✓	Undefined
$5E0_H$	Configuration/interface/endpoint descriptor register 149	USFA0CIE149	R/W				✓	Undefined
$5E4_H$	Configuration/interface/endpoint descriptor register 150	USFA0CIE150	R/W				✓	Undefined
$5E8_H$	Configuration/interface/endpoint descriptor register 151	USFA0CIE151	R/W				✓	Undefined
$5EC_H$	Configuration/interface/endpoint descriptor register 152	USFA0CIE152	R/W				✓	Undefined
$5F0_H$	Configuration/interface/endpoint descriptor register 153	USFA0CIE153	R/W				✓	Undefined
$5F4_H$	Configuration/interface/endpoint descriptor register 154	USFA0CIE154	R/W				✓	Undefined
$5F8_H$	Configuration/interface/endpoint descriptor register 155	USFA0CIE155	R/W				✓	Undefined
$5FC_H$	Configuration/interface/endpoint descriptor register 156	USFA0CIE156	R/W				✓	Undefined
600_H	Configuration/interface/endpoint descriptor register 157	USFA0CIE157	R/W				✓	Undefined
604_H	Configuration/interface/endpoint descriptor register 158	USFA0CIE158	R/W				✓	Undefined
608_H	Configuration/interface/endpoint descriptor register 159	USFA0CIE159	R/W				✓	Undefined
$60C_H$	Configuration/interface/endpoint descriptor register 160	USFA0CIE160	R/W				✓	Undefined
610_H	Configuration/interface/endpoint descriptor register 161	USFA0CIE161	R/W				✓	Undefined
614_H	Configuration/interface/endpoint descriptor register 162	USFA0CIE162	R/W				✓	Undefined
618_H	Configuration/interface/endpoint descriptor register 163	USFA0CIE163	R/W				✓	Undefined
$61C_H$	Configuration/interface/endpoint descriptor register 164	USFA0CIE164	R/W				✓	Undefined
620_H	Configuration/interface/endpoint descriptor register 165	USFA0CIE165	R/W				✓	Undefined
624_H	Configuration/interface/endpoint descriptor register 166	USFA0CIE166	R/W				✓	Undefined
628_H	Configuration/interface/endpoint descriptor register 167	USFA0CIE167	R/W				✓	Undefined
$62C_H$	Configuration/interface/endpoint descriptor register 168	USFA0CIE168	R/W				✓	Undefined
630_H	Configuration/interface/endpoint descriptor register 169	USFA0CIE169	R/W				✓	Undefined
634_H	Configuration/interface/endpoint descriptor register 170	USFA0CIE170	R/W				✓	Undefined
638_H	Configuration/interface/endpoint descriptor register 171	USFA0CIE171	R/W				✓	Undefined

Table 30-11 EPC request data registers (9/12)

Offset address	Register name	Symbol	R/W	Manipulatable bit unit				Initial value
				1	8	16	32	
$63C_H$	Configuration/interface/endpoint descriptor register 172	USFA0CIE172	R/W				✓	Undefined
640_H	Configuration/interface/endpoint descriptor register 173	USFA0CIE173	R/W				✓	Undefined
644_H	Configuration/interface/endpoint descriptor register 174	USFA0CIE174	R/W				✓	Undefined
648_H	Configuration/interface/endpoint descriptor register 175	USFA0CIE175	R/W				✓	Undefined
$64C_H$	Configuration/interface/endpoint descriptor register 176	USFA0CIE176	R/W				✓	Undefined
650_H	Configuration/interface/endpoint descriptor register 177	USFA0CIE177	R/W				✓	Undefined
654_H	Configuration/interface/endpoint descriptor register 178	USFA0CIE178	R/W				✓	Undefined
658_H	Configuration/interface/endpoint descriptor register 179	USFA0CIE179	R/W				✓	Undefined
$65C_H$	Configuration/interface/endpoint descriptor register 180	USFA0CIE180	R/W				✓	Undefined
660_H	Configuration/interface/endpoint descriptor register 181	USFA0CIE181	R/W				✓	Undefined
664_H	Configuration/interface/endpoint descriptor register 182	USFA0CIE182	R/W				✓	Undefined
668_H	Configuration/interface/endpoint descriptor register 183	USFA0CIE183	R/W				✓	Undefined
$66C_H$	Configuration/interface/endpoint descriptor register 184	USFA0CIE184	R/W				✓	Undefined
670_H	Configuration/interface/endpoint descriptor register 185	USFA0CIE185	R/W				✓	Undefined
674_H	Configuration/interface/endpoint descriptor register 186	USFA0CIE186	R/W				✓	Undefined
678_H	Configuration/interface/endpoint descriptor register 187	USFA0CIE187	R/W				✓	Undefined
$67C_H$	Configuration/interface/endpoint descriptor register 188	USFA0CIE188	R/W				✓	Undefined
680_H	Configuration/interface/endpoint descriptor register 189	USFA0CIE189	R/W				✓	Undefined
684_H	Configuration/interface/endpoint descriptor register 190	USFA0CIE190	R/W				✓	Undefined
688_H	Configuration/interface/endpoint descriptor register 191	USFA0CIE191	R/W				✓	Undefined
$68C_H$	Configuration/interface/endpoint descriptor register 192	USFA0CIE192	R/W				✓	Undefined
690_H	Configuration/interface/endpoint descriptor register 193	USFA0CIE193	R/W				✓	Undefined
694_H	Configuration/interface/endpoint descriptor register 194	USFA0CIE194	R/W				✓	Undefined
698_H	Configuration/interface/endpoint descriptor register 195	USFA0CIE195	R/W				✓	Undefined

Table 30-11 EPC request data registers (10/12)

Offset address	Register name	Symbol	R/W	Manipulatable bit unit				Initial value
				1	8	16	32	
$69C_H$	Configuration/interface/endpoint descriptor register 196	USFA0CIE196	R/W				✓	Undefined
$6A0_H$	Configuration/interface/endpoint descriptor register 197	USFA0CIE197	R/W				✓	Undefined
$6A4_H$	Configuration/interface/endpoint descriptor register 198	USFA0CIE198	R/W				✓	Undefined
$6A8_H$	Configuration/interface/endpoint descriptor register 199	USFA0CIE199	R/W				✓	Undefined
$6AC_H$	Configuration/interface/endpoint descriptor register 200	USFA0CIE200	R/W				✓	Undefined
$6B0_H$	Configuration/interface/endpoint descriptor register 201	USFA0CIE201	R/W				✓	Undefined
$6B4_H$	Configuration/interface/endpoint descriptor register 202	USFA0CIE202	R/W				✓	Undefined
$6B8_H$	Configuration/interface/endpoint descriptor register 203	USFA0CIE203	R/W				✓	Undefined
$6BC_H$	Configuration/interface/endpoint descriptor register 204	USFA0CIE204	R/W				✓	Undefined
$6C0_H$	Configuration/interface/endpoint descriptor register 205	USFA0CIE205	R/W				✓	Undefined
$6C4_H$	Configuration/interface/endpoint descriptor register 206	USFA0CIE206	R/W				✓	Undefined
$6C8_H$	Configuration/interface/endpoint descriptor register 207	USFA0CIE207	R/W				✓	Undefined
$6CC_H$	Configuration/interface/endpoint descriptor register 208	USFA0CIE208	R/W				✓	Undefined
$6D0_H$	Configuration/interface/endpoint descriptor register 209	USFA0CIE209	R/W				✓	Undefined
$6D4_H$	Configuration/interface/endpoint descriptor register 210	USFA0CIE210	R/W				✓	Undefined
$6D8_H$	Configuration/interface/endpoint descriptor register 211	USFA0CIE211	R/W				✓	Undefined
$6DC_H$	Configuration/interface/endpoint descriptor register 212	USFA0CIE212	R/W				✓	Undefined
$6E0_H$	Configuration/interface/endpoint descriptor register 213	USFA0CIE213	R/W				✓	Undefined
$6E4_H$	Configuration/interface/endpoint descriptor register 214	USFA0CIE214	R/W				✓	Undefined
$6E8_H$	Configuration/interface/endpoint descriptor register 215	USFA0CIE215	R/W				✓	Undefined
$6EC_H$	Configuration/interface/endpoint descriptor register 216	USFA0CIE216	R/W				✓	Undefined
$6F0_H$	Configuration/interface/endpoint descriptor register 217	USFA0CIE217	R/W				✓	Undefined
$6F4_H$	Configuration/interface/endpoint descriptor register 218	USFA0CIE218	R/W				✓	Undefined
$6F8_H$	Configuration/interface/endpoint descriptor register 219	USFA0CIE219	R/W				✓	Undefined

Table 30-11 EPC request data registers (11/12)

Offset address	Register name	Symbol	R/W	Manipulatable bit unit				Initial value
				1	8	16	32	
$6FC_H$	Configuration/interface/endpoint descriptor register 220	USFA0CIE220	R/W				✓	Undefined
700_H	Configuration/interface/endpoint descriptor register 221	USFA0CIE221	R/W				✓	Undefined
704_H	Configuration/interface/endpoint descriptor register 222	USFA0CIE222	R/W				✓	Undefined
708_H	Configuration/interface/endpoint descriptor register 223	USFA0CIE223	R/W				✓	Undefined
$70C_H$	Configuration/interface/endpoint descriptor register 224	USFA0CIE224	R/W				✓	Undefined
710_H	Configuration/interface/endpoint descriptor register 225	USFA0CIE225	R/W				✓	Undefined
714_H	Configuration/interface/endpoint descriptor register 226	USFA0CIE226	R/W				✓	Undefined
718_H	Configuration/interface/endpoint descriptor register 227	USFA0CIE227	R/W				✓	Undefined
$71C_H$	Configuration/interface/endpoint descriptor register 228	USFA0CIE228	R/W				✓	Undefined
720_H	Configuration/interface/endpoint descriptor register 229	USFA0CIE229	R/W				✓	Undefined
724_H	Configuration/interface/endpoint descriptor register 230	USFA0CIE230	R/W				✓	Undefined
728_H	Configuration/interface/endpoint descriptor register 231	USFA0CIE231	R/W				✓	Undefined
$72C_H$	Configuration/interface/endpoint descriptor register 232	USFA0CIE232	R/W				✓	Undefined
730_H	Configuration/interface/endpoint descriptor register 233	USFA0CIE233	R/W				✓	Undefined
734_H	Configuration/interface/endpoint descriptor register 234	USFA0CIE234	R/W				✓	Undefined
738_H	Configuration/interface/endpoint descriptor register 235	USFA0CIE235	R/W				✓	Undefined
$73C_H$	Configuration/interface/endpoint descriptor register 236	USFA0CIE236	R/W				✓	Undefined
740_H	Configuration/interface/endpoint descriptor register 237	USFA0CIE237	R/W				✓	Undefined
744_H	Configuration/interface/endpoint descriptor register 238	USFA0CIE238	R/W				✓	Undefined
748_H	Configuration/interface/endpoint descriptor register 239	USFA0CIE239	R/W				✓	Undefined
$74C_H$	Configuration/interface/endpoint descriptor register 240	USFA0CIE240	R/W				✓	Undefined
750_H	Configuration/interface/endpoint descriptor register 241	USFA0CIE241	R/W				✓	Undefined
754_H	Configuration/interface/endpoint descriptor register 242	USFA0CIE242	R/W				✓	Undefined
758_H	Configuration/interface/endpoint descriptor register 243	USFA0CIE243	R/W				✓	Undefined

Table 30-11 EPC request data registers (12/12)

Offset address	Register name	Symbol	R/W	Manipulatable bit unit				Initial value
				1	8	16	32	
$75C_H$	Configuration/interface/endpoint descriptor register 244	USFA0CIE244	R/W				✓	Undefined
760_H	Configuration/interface/endpoint descriptor register 245	USFA0CIE245	R/W				✓	Undefined
764_H	Configuration/interface/endpoint descriptor register 246	USFA0CIE246	R/W				✓	Undefined
768_H	Configuration/interface/endpoint descriptor register 247	USFA0CIE247	R/W				✓	Undefined
$76C_H$	Configuration/interface/endpoint descriptor register 248	USFA0CIE248	R/W				✓	Undefined
770_H	Configuration/interface/endpoint descriptor register 249	USFA0CIE249	R/W				✓	Undefined
774_H	Configuration/interface/endpoint descriptor register 250	USFA0CIE250	R/W				✓	Undefined
778_H	Configuration/interface/endpoint descriptor register 251	USFA0CIE251	R/W				✓	Undefined
$77C_H$	Configuration/interface/endpoint descriptor register 252	USFA0CIE252	R/W				✓	Undefined
780_H	Configuration/interface/endpoint descriptor register 253	USFA0CIE253	R/W				✓	Undefined
784_H	Configuration/interface/endpoint descriptor register 254	USFA0CIE254	R/W				✓	Undefined
788_H	Configuration/interface/endpoint descriptor register 255	USFA0CIE255	R/W				✓	Undefined

(4) Bridge registers

Table 30-12 Bridge registers

Offset address	Register name	Symbol	R/W	Manipulatable bit unit				Initial value
				1	8	16	32	
808_H	H-bus bridge interrupt status register	USFA0BRGINT	R/W				✓	$0000\ 0000_H$
$80C_H$	H-bus bridge interrupt enable register	USFA0BRGINTE	R/W				✓	$0000\ 0000_H$
810_H	EPC interrupt status register	USFA0EPCINT	R				✓	$0000\ 0000_H$
814_H	EPC interrupt enable register	USFA0EPCINTE	R/W				✓	$0000\ 0000_H$
818_H	EPC macro control register	USFA0EPCCTL	R/W				✓	00000001_H
910_H	Endpoint1 DMA control register 1	USFA0E1DC1	R/W				✓	$0000\ 0000_H$
914_H	Endpoint1 DMA control register 2	USFA0E1DC2	R/W				✓	$0000\ 0000_H$
918_H	Endpoint1 DMA transfer address register	USFA0E1TAD	R/W				✓	$0000\ 0000_H$
920_H	Endpoint2 DMA control register 1	USFA0E2DC1	R/W				✓	$0000\ 0002_H$
928_H	Endpoint2 DMA transfer address register	USFA0E2TAD	R/W				✓	$0000\ 0000_H$
930_H	Endpoint3 DMA control register 1	USFA0E3DC1	R/W				✓	$0000\ 0000_H$
934_H	Endpoint3 DMA control register 2	USFA0E3DC2	R/W				✓	$0000\ 0000_H$
938_H	Endpoint3 DMA transfer address register	USFA0E3TAD	R/W				✓	$0000\ 0000_H$
940_H	Endpoint4 DMA control register 1	USFA0E4DC1	R/W				✓	$0000\ 0002_H$
944_H	Endpoint4 DMA control register 2	USFA0E4DC2	R/W				✓	$0000\ 0000_H$
948_H	Endpoint4 DMA transfer address register	USFA0E4TAD	R/W				✓	$0000\ 0000_H$

30.7.2 EPC control registers

(1) EP0NAK register (USFA0E0N)

This register controls NAK of Endpoint0 (except an automatically executed request).

It takes five USB clocks to apply the status to this register after the USFA0FIC0 and USFA0FIC1 registers have been set. If it is necessary to read the status correctly, therefore, separate a write signal that accesses the USFA0FIC0 and USFA0FIC1 registers from a read signal that accesses the USFA0EPS0, USFA0EPS1, USFA0EPS2, USFA0E0N, and USFA0EN registers by at least four USB clocks.

While NAK is being transmitted to Endpoint0 Read, Endpoint2, and Endpoint4, a write access to the EP0NKR bit is ignored.

Access This register can be read or written in 8-bit units. (However, bit 0 can only be read.)

Address 000_H

Initial value 00_H. This register is initialized by any reset.

	7	6	5	4	3	2	1	0
USFA0E0N	0	0	0	0	0	0	EP0NKR	EP0NKW
	R	R	R	R	R	R	R/W	R

Table 30-13 USFA0E0N register contents

Bit position	Bit name	Function
1	EP0NKR	This bit controls NAK to the OUT token to Endpoint0 (except an automatically executed request). It is automatically set to 1 by hardware when Endpoint0 correctly receives data. It is automatically cleared to 0 by hardware when the data of the USFA0E0R register is read by FW (counter value = 0). 1: Transmit NAK. 0: Do not transmit NAK. (Default) To not receive data from the USB bus for some reason even if USBF is ready to receive data, set this bit to 1 using FW. In this case, USBF continues transmitting NAK until this bit is cleared to 0 by FW. This bit is cleared to 0 at the same time when the USFA0E0R register is cleared.
0	EP0NKW	This bit indicates how NAK to the IN token to Endpoint0 is controlled (except an automatically executed request). It is automatically cleared to 0 by hardware when the data of Endpoint0 is transmitted and the host correctly receives the transmitted data. The data of the USFA0E0W register is retained until this bit is cleared. Therefore, it is not necessary to rewrite this bit even in the case of a retransmission request that is made if the host could not receive data correctly. To send a short packet, be sure to set the E0DED bit of the USFA0DEND register to 1. This bit is automatically set to 1 when the FIFO is full. This bit is automatically set to 1 at the same time when the E0DED bit of the USFA0DEND register is set to 1. 1: Do not transmit NAK. 0: Transmit NAK. (Default) If a control transfer enters the status stage while ACK cannot be correctly received in the data stage, this bit is cleared to 0 at the same time when the USFA0E0W register is cleared. This bit is also cleared to 0 when USFA0E0W is cleared by FW.

The procedure for a SETUP transaction that uses IN/OUT tokens is described below.

(a) When an IN token is used (except a request automatically executed by hardware)

Use FW to clear the PROT bit of the USFA0IS1 register to 0 after receiving the CPUDEC interrupt and before reading data from the USFA0E0ST register. Next, perform processing in accordance with the request and, if it is necessary to return data by using an IN token, write data to the USFA0E0W register. After data has been written completely, confirm that the PROT bit of the USFA0IS1 register is 0, and then set the E0DED bit of the USFA0DEND register to 1. The hardware sends out data at the first IN token after the EP0NKW bit has been set to 1. If the PROT bit of the USFA0IS1 register is 1, it indicates that a SETUP transaction occurred again before the completion of the control transfer. In this case, clear the PROT bit of the USFA0IS1 register to 0 by clearing the PROTC bit of the USFA0IC1 register to 0, and then read data from the USFA0E0ST register again. A request received later can be read.

(b) When an OUT token is used (except a request automatically executed by hardware)

Use FW to clear the PROT bit of the USFA0IS1 register to 0 after receiving the CPUDEC interrupt and before reading data from the USFA0E0ST register. Confirm that the PROT bit of the USFA0IS1 register is 0 before reading data from the USFA0E0R register. If the PROT bit is 1, it indicates that invalid data is retained. Clear the FIFO by using FW. (The EP0NKR bit is automatically cleared to 0.) If the PROT bit of the USFA0IS1 register is 0, read the data of the USFA0E0L register, and then read the set number of data items from the USFA0E0R register. When data has been read from the USFA0E0R register completely, (when the counter of the USFA0E0R register is cleared to 0), the hardware automatically clears the EP0NKR bit to 0.

(2) EP0NAKALL register (USFA0E0NA)

This register controls NAK to all the requests of Endpoint0. It is also valid for automatically executed requests.

Access This register can be read or written in 8-bit units.

Address 004_H

Initial value 00_H. This register is initialized by any reset.

	7	6	5	4	3	2	1	0
USFA0E0NA	0	0	0	0	0	0	0	EP0NKA
	R	R	R	R	R	R	R	R/W

Table 30-14 USFA0E0NA register contents

Bit position	Bit name	Function
0	EP0NKA	This bit controls NAK to Endpoint0 of a transaction other than a SETUP transaction (including an automatically executed request). This bit is manipulated by FW. 1: Transmit NAK. 0: Do not transmit NAK. (Default) This register is used to prevent a conflict between a write access by FW and a read access from SIE when the data used for an automatically executed request is to be changed. It postpones applying write access to this bit from FW during access from SIE. Before rewriting the request data register from FW, confirm that this bit has been correctly set to 1. Setting this bit to 1 is applied only in the following cases: - Immediately after USBF is reset and no SETUP token has been received - Immediately after receiving Bus Reset and no SETUP token has been received - When a PID of a SETUP token is detected - When the stage is changed to the status stage Clearing this bit to 0 is applied immediately, except while an IN token is being received and a NAK response is being sent. Setting the EP0NKA bit to 1 is applied only in the above four cases during an Endpoint0 transfer, but it is applied immediately after data has been written during a different transfer.

(3) EPNAK register (USFA0EN)

This register controls NAK of endpoints other than Endpoint0.

The BKO2NK bit can be written only when the BKO2NKM bit of the USFA0ENM register is 1, and the BKO1NK bit can be written only when the BKO1NKM bit of the USFA0ENM register is 1.

The related bits are invalid if the corresponding endpoint is not supported according to the setting of the USFA0EnIM register (n = 1 to 4, 7, 8) and the current interface.

It takes five USB clocks to apply the status to this register after the USFA0FIC0 and USFA0FIC1 registers have been set. If it is necessary to read the status correctly, therefore, separate a write signal that accesses the USFA0FIC0 and USFA0FIC1 registers from a read signal that accesses the USFA0EPS0, USFA0EPS1, USFA0EPS2, USFA0E0N, and USFA0EN registers by at least four USB clocks.

While NAK is being transmitted to Endpoint0 Read, Endpoint2, and Endpoint4, writing to the BKO1NK and BKO2NK bits is ignored.

Access This register can be read or written in 8-bit units. (However, bits 5, 4, and 1 can only be read.)

Address 008_H

Initial value 00_H. This register is initialized by any reset.

Caution Be sure to clear bits 7 and 6 to “0”.

	7	6	5	4	3	2	1	0
USFA0EN	0	0	IT2NK	IT1NK	BKO2NK	BKO1NK	BKI2NK	BKI1NK
	R	R	R	R	R/W	R/W	R	R

Table 30-15 USFA0EN register contents (1/3)

Bit position	Bit name	Function
5	IT2NK	This bit controls NAK to Endpoint8 (interrupt 2 transfer). It is automatically set to 1 and transmission is started when the USFA0INT2 register becomes full as a result of writing data to it. To send a short packet that does not make the FIFO full, set the IT2DEND bit of the USFA0DEND register to 1. When the IT2DEND bit is set to 1, this bit is automatically set to 1 at the same time. 1: Do not transmit NAK. 0: Transmit NAK. (Default) This bit is also cleared to 0 at the same time when the USFA0INT2 register is cleared.
4	IT1NK	This bit controls NAK to Endpoint7 (interrupt 1 transfer). It is automatically set to 1 and transmission is started when the USFA0INT1 register becomes full as a result of writing data to it. To send a short packet that does not make the FIFO full, set the IT1DEND bit of the USFA0DEND register to 1. When the IT1DEND bit is set to 1, this bit is automatically set to 1 at the same time. 1: Do not transmit NAK. 0: Transmit NAK. (Default) This bit is cleared to 0 at the same time when the USFA0INT1 register is cleared.

Table 30-15 USFA0EN register contents (2/3)

Bit position	Bit name	Function
3	BKO2NK	This bit controls NAK to Endpoint4 (bulk 2 transfer (OUT)). 1: Transmit NAK. 0: Do not transmit NAK. (Default) This bit is set to 1 only when the FIFO connected to the SIE side of the USFA0BO2 register (a 64-byte FIFO in a bank configuration) cannot receive data. It is cleared to 0 when a toggle operation is performed. The bank is changed (toggle operation) when the following conditions are satisfied: - Correctly received data is stored in the FIFO connected to the SIE side. - The value of the FIFO counter connected to the CPU side is 0 (completion of reading). Use FW to read data of the USFA0BO2L register when it has received the BLKO2DT interrupt request and read as many data items from the USFA0BO2 register as the value of that data. To not receive data from the USB bus for some reason even if USBF is ready to receive data, set this bit to 1 using FW. In this case, USBF keeps transmitting NAK until the FW clears this bit to 0. This bit is also cleared to 0 at the same time when the USFA0BO2 register is cleared.
2	BKO1NK	This bit controls NAK to Endpoint2 (bulk 1 transfer (OUT)). 1: Transmit NAK. 0: Do not transmit NAK. (Default) This bit is set to 1 only when the FIFO connected to the SIE side of the USFA0BO1 register (a 64-byte FIFO in a bank configuration) cannot receive data. It is cleared to 0 when a toggle operation is performed. The bank is changed (toggle operation) when the following conditions are satisfied: - Correctly received data is stored in the FIFO connected to the SIE side. - The value of the FIFO counter connected to the CPU side is 0 (completion of reading). Use FW to read data of the USFA0BO1L register when it has received the BLKO1DT interrupt request and read as many data items from the USFA0BO1 register as the value of that data. To not receive data from the USB bus for some reason even if USBF is ready to receive data, set this bit to 1 using FW. In this case, USBF keeps transmitting NAK until the FW clears this bit to 0. This bit is also cleared to 0 at the same time when the USFA0BO1 register is cleared.
1	BKI2NK	This bit controls NAK to Endpoint3 (bulk 2 transfer (IN)). 1: Do not transmit NAK. 0: Transmit NAK. (Default) This bit is cleared to 0 only when the FIFO that is connected to the SIE side and has transmitted the data of the USFA0BI2 register (a 64-byte FIFO in a bank configuration) cannot receive data. It is set to 1 when a toggle operation is performed. (The data of the USFA0BI2 register is retained until transmission has been correctly completed.) The bank is changed (toggle operation) when the following conditions are satisfied: - Data is correctly written to the FIFO connected to the CPU bus side. (Writing has been completed and the FIFO is full or the USFA0DEND register is set.) - The value of the FIFO counter connected to the SIE side is 0. This bit is automatically set to 1 and data transmission is started when the FIFO on the CPU side becomes full and a FIFO toggle operation is performed as a result of writing data to the FIFO. However, if the FIFO on the CPU side becomes full as a result of writing data to it by DMA while the BKI2T bit of the USFA0DEND register is cleared to 0, the toggle operation is not performed because the condition of the toggle operation is not satisfied until the BKI2DED bit of the USFA0DEND register is set to 1. To send a short packet that does not make the FIFO on the CPU side full, set the BKI2DED bit to 1 after completing writing data. When the BKI2DED bit is set to 1, a toggle operation is performed and, at the same time, this bit is automatically set to 1. This bit is also cleared to 0 at the same time when the USFA0BI2 register is cleared.

Table 30-15 USFA0EN register contents (2/3)

Bit position	Bit name	Function
3	BKO2NK	This bit controls NAK to Endpoint4 (bulk 2 transfer (OUT)). 1: Transmit NAK. 0: Do not transmit NAK. (Default) This bit is set to 1 only when the FIFO connected to the SIE side of the USFA0BO2 register (a 64-byte FIFO in a bank configuration) cannot receive data. It is cleared to 0 when a toggle operation is performed. The bank is changed (toggle operation) when the following conditions are satisfied: - Correctly received data is stored in the FIFO connected to the SIE side. - The value of the FIFO counter connected to the CPU side is 0 (completion of reading). Use FW to read data of the USFA0BO2L register when it has received the BLKO2DT interrupt request and read as many data items from the USFA0BO2 register as the value of that data. To not receive data from the USB bus for some reason even if USBF is ready to receive data, set this bit to 1 using FW. In this case, USBF keeps transmitting NAK until the FW clears this bit to 0. This bit is also cleared to 0 at the same time when the USFA0BO2 register is cleared.
2	BKO1NK	This bit controls NAK to Endpoint2 (bulk 1 transfer (OUT)). 1: Transmit NAK. 0: Do not transmit NAK. (Default) This bit is set to 1 only when the FIFO connected to the SIE side of the USFA0BO1 register (a 64-byte FIFO in a bank configuration) cannot receive data. It is cleared to 0 when a toggle operation is performed. The bank is changed (toggle operation) when the following conditions are satisfied: - Correctly received data is stored in the FIFO connected to the SIE side. - The value of the FIFO counter connected to the CPU side is 0 (completion of reading). Use FW to read data of the USFA0BO1L register when it has received the BLKO1DT interrupt request and read as many data items from the USFA0BO1 register as the value of that data. To not receive data from the USB bus for some reason even if USBF is ready to receive data, set this bit to 1 using FW. In this case, USBF keeps transmitting NAK until the FW clears this bit to 0. This bit is also cleared to 0 at the same time when the USFA0BO1 register is cleared.
1	BKI2NK	This bit controls NAK to Endpoint3 (bulk 2 transfer (IN)). 1: Do not transmit NAK. 0: Transmit NAK. (Default) This bit is cleared to 0 only when the FIFO that is connected to the SIE side and has transmitted the data of the USFA0BI2 register (a 64-byte FIFO in a bank configuration) cannot receive data. It is set to 1 when a toggle operation is performed. (The data of the USFA0BI2 register is retained until transmission has been correctly completed.) The bank is changed (toggle operation) when the following conditions are satisfied: - Data is correctly written to the FIFO connected to the CPU bus side. (Writing has been completed and the FIFO is full or the USFA0DEND register is set.) - The value of the FIFO counter connected to the SIE side is 0. This bit is automatically set to 1 and data transmission is started when the FIFO on the CPU side becomes full and a FIFO toggle operation is performed as a result of writing data to the FIFO. However, if the FIFO on the CPU side becomes full as a result of writing data to it by DMA while the BKI2T bit of the USFA0DEND register is cleared to 0, the toggle operation is not performed because the condition of the toggle operation is not satisfied until the BKI2DED bit of the USFA0DEND register is set to 1. To send a short packet that does not make the FIFO on the CPU side full, set the BKI2DED bit to 1 after completing writing data. When the BKI2DED bit is set to 1, a toggle operation is performed and, at the same time, this bit is automatically set to 1. This bit is also cleared to 0 at the same time when the USFA0BI2 register is cleared.

Table 30-15 USFA0EN register contents (3/3)

Bit position	Bit name	Function
0	BKI1NK	This bit controls NAK to Endpoint1 (bulk 1 transfer (IN)). 1: Do not transmit NAK. 0: Transmit NAK. (Default) This bit is cleared to 0 only when the FIFO that is connected to the SIE side and has transmitted the data of the USFA0BI1 register (a 64-byte FIFO in a bank configuration) cannot receive data. It is set to 1 when a toggle operation is performed. (The data of the USFA0BI1 register is retained until transmission has been correctly completed.) The bank is changed (toggle operation) when the following conditions are satisfied: - Data is correctly written to the FIFO connected to the CPU bus side. (Writing has been completed and the FIFO is full or the USFA0DEND register is set.) - The value of the FIFO counter connected to the SIE side is 0. This bit is automatically set to 1 and data transmission is started when the FIFO on the CPU side becomes full and a FIFO toggle operation is performed as a result of writing data to the FIFO. However, if the FIFO on the CPU side becomes full as a result of writing data to it by DMA while the BKI1T bit of the USFA0DEND register is cleared to 0, the toggle operation is not performed because the condition of the toggle operation is not satisfied until the BKI1DED bit of the USFA0DEND register is set to 1. To send a short packet that does not make the FIFO on the CPU side full, set the BKI1DED bit to 1 after completing writing data. When the BKI1DED bit is set to 1, a toggle operation is performed and, at the same time, this bit is automatically set to 1. This bit is also cleared to 0 at the same time when the USFA0BI1 register is cleared.

Cautions

1. If DMA is enabled while data is being read from the USFA0BO2 register in the PIO mode, a DMA request is immediately issued.
2. If the last data of the FIFO on the CPU side is read in the DMA transfer mode, the DMA request signal becomes inactive.
3. If the TC signal is received in the DMA transfer mode, the DMA request signal becomes inactive.
4. If 64-byte data is written in the DMA transfer mode, the DMA request signal becomes inactive. If the BKI2NK bit is then set to 1, data is transmitted in synchronization with an IN token. The DMA request signal becomes active again when the FIFO is toggled, as long as the DMA request is not masked. If the BKI2NK bit is not set, data is not transmitted even if an IN token is received. In this case, set the BKI2DED bit of the USFA0DEND register to 1.
5. If the TC signal is received in the DMA transfer mode, the DMA request signal becomes inactive. At the same time, the DMA request is masked. If the BKI2NK bit is not set to 1, data is not transmitted even if an IN token is received. When the BKI2DED bit of the USFA0DEND register is set to 1 by FW, data is transmitted in synchronization with an IN token. To execute another DMA transfer, unmask the DMA request.

(4) EPNAK mask register (USFA0ENM)

This register controls masking writing to the USFA0EN register.

Access This register can be read or written in 8-bit units.

Address $00C_H$

Initial value 00_H. This register is initialized by any reset.

Caution Be sure to clear bits 7 to 4, 1, and 0 to “0”.

	7	6	5	4	3	2	1	0
USFA0ENM	0	0	0	0	BKO2NKM	BKO1NKM1	0	0
	R	R	R	R	R/W	R/W	R	R

Table 30-16 USFA0ENM register contents

Bit position	Bit name	Function
3	BKO2NKM	Specify whether to mask writing to bit 3 (BKO2NK) of the USFA0EN register. 1: Do not mask writing. 0: Mask writing. (Default)
2	BKO1NKM	Specify whether to mask writing to bit 2 (BKO1NK) of the USFA0EN register. 1: Do not mask writing. 0: Mask writing. (Default)

(5) SNDSIE register (USFA0SDS)

This register performs manipulation such as no handshake. It can directly manipulate the SIE pins.

Be sure to clear bit 2 to “0”. If it is set to 1, the operation is not guaranteed.

Access This register can be read or written in 8-bit units.

Address 010_H

Initial value 00_H. This register is initialized by any reset.

Caution Be sure to clear bits 7 to 4, 1, and 0 to “0”.

	7	6	5	4	3	2	1	0
USFA0SDS	0	0	0	0	SNDSTL	0	0	RSUMIN
	R	R	R	R	R/W	R	R	R/W

Table 30-17 USFA0SDS register contents

Bit position	Bit name	Function
3	SNDSTL	This bit makes Endpoint0 issue a STALL handshake. Setting this bit to 1 if a request for CPUDEC processing is not supported by the system results in a STALL handshake response. If an unsupported wValue is sent by the SET_CONFIGURATION or SET_INTERFACE request, the hardware sets this bit to 1. This bit is also set to 1 if a problem occurs in Endpoint0 due to overrun of an automatically executed request. However, the E0HALT bit of the USFA0E0S register is not set to 1. 1: Respond with a STALL handshake. 0: Do not respond with a STALL handshake. (Default) This bit is cleared to 0 and the handshake response to the bus is not STALL when the next SETUP token is received. To set the SNDSTL bit to 1 by using FW, do not write data to the USFA0E0W register. Depending on the timing of setting this bit, the STALL response might not be made in time, and it might be sent to the next transfer after a NAK response has been sent. Setting this bit is valid only while an FW-executed request is under execution when this bit is set to 1. It is automatically cleared to 0 when the next SETUP token is received. **Note** The SNDSTL bit is valid only for an FW-executed request.
0	RSUMIN	This bit outputs the Resume signal to the USB bus. Writing this bit is invalid unless the RMWK bit of the USFA0DST register is set to 1. 1: Generate the Resume signal. 0: Do not generate the Resume signal. (Default) While this bit is set to 1, the Resume signal continues to be generated. Clear this bit to 0 using FW after a specific time has elapsed. Because sampling is internally performed with the clock, the operation is guaranteed only when CLK is supplied. Care must be exercised when the system CLK is stopped.

(6) CLR request register (USFA0CLR)

This register indicates the target of the received CLEAR_FEATURE request.

This register is meaningful only when an interrupt request is generated. Each bit is set to 1 after completion of the status stage, and automatically cleared to 0 when this register is read.

The related bits are invalid if the corresponding endpoint is not supported according to the setting of the USFA0EnIM register (n = 1 to 4, 7, 8) and the current interface.

Access This register is read-only, in 8-bit units.

Address 014_H

Initial value 00_H. This register is initialized by any reset.

	7	6	5	4	3	2	1	0
USFA0CLR	CLREP8	CLREP7	CLREP4	CLREP3	CLREP2	CLREP1	CLREP0	CLRDEV
	R	R	R	R	R	R	R	R

Table 30-18 USFA0CLR register contents

Bit position	Bit name	Function
7:1	CLREPn	These bits indicate that a CLEAR_FEATURE Endpoint n request was received and automatically processed. 1: A request was automatically processed. 0: No request was automatically processed. (Default)
0	CLRDEV	This bit indicates that a CLEAR_FEATURE Device request was received and automatically processed. 1: A request was automatically processed. 0: No request was automatically processed. (Default)

Note n = 0 to 4, 7, 8

(7) SET request register (USFA0SET)

This register indicates the target of an automatically processed SET_XXXX (except SET_INTERFACE) request.

This register is meaningful only when an interrupt request is generated. Each bit is set to 1 after completion of the status stage, and automatically cleared to 0 when this register is read.

Access This register is read-only, in 8-bit units.

Address 018_H

Initial value 00_H. This register is initialized by any reset.

	7	6	5	4	3	2	1	0
USFA0SET	SETCON	0	0	0	0	SETEP	0	SETDEV
	R	R	R	R	R	R	R	R

Table 30-19 USFA0SET register contents

Bit position	Bit name	Function
7	SETCON	This bit indicates that a SET_CONFIGURATION request was received and automatically processed. 1: A request was automatically processed. 0: No request was automatically processed. (Default)
2	SETEP	This bit indicates that a SET_FEATURE Endpoint n (n = 0 to 4, 7, 8) request was received and automatically processed. 1: A request was automatically processed. 0: No request was automatically processed. (Default)
0	SETDEV	This bit indicates that a SET_FEATURE Device request was received and automatically processed. 1: A request was automatically processed. 0: No request was automatically processed. (Default)

(8) EP status 0 register (USFA0EPS0)

This register indicates the USB bus status and whether there is register data.

The related bits are invalid if the corresponding endpoint is not supported according to the setting of the USFA0EnIM register (n = 1 to 4, 7, 8) and the current interface.

It takes five USB clocks to apply the status to this register after the USFA0FIC0 and USFA0FIC1 registers have been set. If it is necessary to read the status correctly, therefore, separate write access to the USFA0FIC0 and USFA0FIC1 registers from read access to the USFA0EPS0, USFA0EPS1, USFA0EPS2, USFA0E0N, and USFA0EN registers by at least four USB clocks.

Access This register is read-only, in 8-bit units.

Address $01C_H$

Initial value 00_H. This register is initialized by any reset.

	7	6	5	4	3	2	1	0
USFA0EPS0	IT2	IT1	BKOUT2	BKOUT1	BKIN2	BKIN1	EP0W	EP0R
	R	R	R	R	R	R	R	R

Table 30-20 USFA0EPS0 register contents (1/2)

Bit position	Bit name	Function
7, 6	ITn	These bits indicate whether data is in the USFA0INTn register (FIFO). By setting the ITnDEND bit of the USFA0DEND register to 1, the status in which data is in the USFA0INTn register can be set even if data is not written to the register (Null data transmission). When the ITnDEND bit of the USFA0DEND register is set to 1, this bit is also set to 1 by hardware at the same time even if the counter of the USFA0INTn register is 0. This bit is cleared to 0 after successful transmission. 1: There is data in the register. 0: There is no data in the register. (Default)
5, 4	BKOUTn	These bits indicate whether data is in the USFA0BOn register (FIFO) connected to the CPU side. When the FIFO that makes up the USFA0BOn register is toggled, this bit is automatically set to 1 by hardware. It is automatically cleared to 0 by hardware when reading the USFA0BOn register (FIFO) connected to the CPU side has been completed (counter value = 0). It is not set to 1 when Null data is received (nor does toggling the FIFO does not take place). 1: There is data in the register. 0: There is no data in the register. (Default)
3, 2	BKINn	These bits indicate whether data is in the USFA0BIn register (FIFO) connected to the CPU side. By setting the BKInDED bit of the USFA0DEND register to 1, the status in which data is in the USFA0BIn register can be set even if data is not written to the register (Null data transmission). When the BKInDED bit of the USFA0DEND register is set to 1, this bit is also set to 1 by hardware at the same time even if the counter of the USFA0BIn register is 0. It is cleared to 0 when a toggle operation is performed. 1: There is data in the register. 0: There is no data in the register. (Default)

Table 30-20 USFA0EPS0 register contents (2/2)

Bit position	Bit name	Function
1	EP0W	This bit indicates that data is in the USFA0E0W register (FIFO). By setting the E0DED bit of the USFA0DEND register to 1, the status in which data is in the USFA0E0W register can be set even if data is not written to the register (Null data transmission). When the E0DED bit of the USFA0DEND register is set to 1, this bit is also set to 1 by hardware at the same time even if the counter of the USFA0E0W register is 0. This bit is cleared to 0 after successful transmission. 1: There is data in the register. 0: There is no data in the register. (Default)
0	EP0R	This bit indicates that data is in the USFA0E0R register (FIFO). It is automatically cleared to 0 by hardware when reading the USFA0E0R register (FIFO) has been completed (counter value = 0). It is not set to 1 if Null data is received. 1: There is data in the register. 0: There is no data in the register. (Default)

Note n = 1 or 2

(9) EP status 1 register (USFA0EPS1)

This register indicates the USB bus status and whether there is register data.

Access This register is read-only, in 8-bit units.

Address 020_H

Initial value 00_H. This register is initialized by any reset.

	7	6	5	4	3	2	1	0
USFA0EPS1	RSUM	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R

Table 30-21 USFA0EPS1 register contents

Bit position	Bit name	Function
7	RSUM	This bit indicates that the USB bus is in the Resume status. This bit is meaningful only when an interrupt request is generated. 1: The bus entered the Suspend status. 0: The bus entered the Resume status. (Default) Because sampling is internally performed with the clock, the operation is guaranteed only when CLK is supplied. Care must be exercised when the system CLK is stopped. The INTUSFA0I2 signal of SIE operates even when CLK is stopped. It can therefore be supported by making the interrupt control register (UFIC1) valid or lowering the frequency of CLK for the USBF. This bit is automatically cleared to 0 when it is read.

(10) EP status 2 register (USFA0EPS2)

This register indicates the USB bus status and whether there is register data.

The related bits are invalid if the corresponding endpoint is not supported according to the setting of the USFA0EnIM register (n = 1 to 4, 7, 8) and the current interface.

Access This register is read-only, in 8-bit units.

Address 024_H

Initial value 00_H. This register is initialized by any reset.

	7	6	5	4	3	2	1	0
USFA0EPS2	0	HALT8	HALT7	HALT4	HALT3	HALT2	HALT1	HALT0
	R	R	R	R	R	R	R	R

Table 30-22 USFA0EPS2 register contents

Bit position	Bit name	Function
6:0	HALTn	These bits indicate that Endpoint n is currently stalled. They are set to 1 when a stall condition, such as the occurrence of an overrun or reception of an undefined request, is satisfied. These bits are automatically set to 1 by hardware. 1: The endpoint is stalled. 0: The endpoint is not stalled. (Default) The SNDSTL bit is set to 1 at the same time when the HALT0 bit is set to 1 as a result of the occurrence of an overrun or reception of an undefined request. If the next SETUP token is received in this status, the SNDSTL bit is cleared to 0 and, therefore, the HALT0 bit is also cleared to 0. If Endpoint0 is stalled by the SET_FEATURE Endpoint0 request, the HALT0 bit is not cleared to 0 until the CLEAR_FEATURE Endpoint0 request is received or Halt Feature is cleared by FW. If the GET_STATUS Endpoint0, CLEAR_FEATURE Endpoint0, or SET_FEATURE Endpoint0 request is received, or if a request to be processed by FW is received due to the CPUDEC interrupt request, the HALT0 bit is masked and cleared to 0 until the next SETUP token is received. The HALTn bit is not cleared to 0 until Endpoint n receives the CLEAR_FEATURE Endpoint request, Halt Feature is cleared by the SET_INTERFACE or SET_CONFIGURATION request to the interface to which the endpoint is linked, or Halt Feature is cleared by FW. When the SET_INTERFACE or SET_CONFIGURATION request is correctly processed, the Halt Features of all the target endpoints, except Endpoint0, are cleared after the request has been processed, even if the wValue is the same as the currently set value, and these bits are also cleared to 0. Halt Feature of Endpoint0 cannot be cleared if it is set because a STALL response is returned in response to the SET_INTERFACE and SET_CONFIGURATION requests.

Note n = 0 to 4, 7, 8

(11) INT status 0 register (USFA0IS0)

This register indicates the interrupt source. If the contents of this register are changed, the USFA0EPCINT.EPC_INT0 bit is set to 1.

If an interrupt request (INTUSFA0I1) is generated from USBF, the FW must read this register to identify the interrupt source.

Each bit of this register is forcibly cleared to 0 when 0 is written to the corresponding bit of the USFA0IC0 register.

Access This register is read-only, in 8-bit units.

Address 040_H

Initial value 00_H. This register is initialized by any reset.

Caution In the USBF, multiple interrupt sources, such as Bus Reset, Resume, and Short, are ORed internally and issued as a single interrupt request (INTUSFA0I1). Therefore, even if multiple interrupt sources occur, they are ORed and issued as an INTUSFA0I1 interrupt request. For example, if a Bus Reset interrupt source and Resume interrupt source occur, the two sources are ORed and an INTUSFA0I1 interrupt request is issued. Under these conditions, if the Bus Reset interrupt source is cleared to 0 (USFA0IC0.BUSRSTC = 0), the INTUSFA0I1 interrupt request for this product might remain set to 1 because the Resume interrupt source will still remain. The new interrupt request flag (US0BIC.US0BIF), therefore, might not be set to 1. In this case, after performing clear processing for each interrupt request with the INTUSFA0I1 interrupt servicing routine, confirm the flag status for the USFA0IS0 and USFA0IS1 registers again, and, if there are any interrupt sources for which the flags are set to 1, clear the flags. (Clear only the applicable bits, not all the bits.)

	7	6	5	4	3	2	1	0
USFA0IS0	BUSRST	RSUSPD	0	SHORT	DMAED	SETRQ	CLRRQ	EPHALT
	R	R	R	R	R	R	R	R

Table 30-23 USFA0IS0 register contents (1/2)

Bit position	Bit name	Function
7	BUSRST	This bit indicates that Bus Reset has occurred. 1: Bus Reset has occurred. (An interrupt request has been generated.) 0: Bus Reset has not occurred. (Default)
6	RSUSPD	This bit indicates that the Resume or Suspend status has occurred. Reference bit 7 of the USFA0EPS1 register by using FW. 1: The Resume or Suspend status has occurred. (An interrupt request has been generated.) 0: The Resume or Suspend status has not occurred. (Default)
4	SHORT	This bit indicates that data is read from the FIFO of either the USFA0BO1 or USFA0BO2 register and that the USBSPnB signal (n = 2, 4) is active. It is valid only when the FIFO is not full in the DMA mode. 1: The USBSPnB signal has been activated. (An interrupt request has been generated.) 0: The USBSPnB signal has not been activated. (Default) Use the USFA0DMS1 register to determine which endpoint the operation is actually performed on. This bit is not automatically cleared to 0 even when the USFA0DMS1 register is read by FW.

Table 30-23 USFA0IS0 register contents (2/2)

Bit position	Bit name	Function
3	DMAED	This bit indicates that the DMA end (TC) signal for Endpoint n (n = 1 to 4, 7) is active. 1: The DMA end signal for Endpoint n has been input. (An interrupt request has been generated.) 0: The DMA end signal for Endpoint n has not been input. (Default) When this bit is set to 1, the DMA request signal for Endpoint n becomes inactive. The DMA request signal for Endpoint n does not become active unless FW enables DMA transfers. Use the USFA0DMS0 register to determine which endpoint the operation is actually performed on. This bit is not automatically cleared to 0 even when the USFA0DMS0 register is read by FW.
2	SETRQ	This bit indicates that the SET_XXXX request to be automatically processed has been received and automatically processed (XXXX = CONFIGURATION or FEATURE). 1: The SET_XXXX request to be automatically processed has been received. (An interrupt request has been generated.) 0: The SET_XXXX request to be automatically processed has not been received. (Default) This bit is set to 1 upon completion of the status stage. Reference the USFA0SET register to identify the target of the request. This bit is not automatically cleared to 0 even if the USFA0SET register is read by FW. The EPHALT bit is also set to 1 when the SET_FEATURE Endpoint request is received.
1	CLRRQ	This bit indicates that the CLEAR_FEATURE request has been received and automatically processed. 1: The CLEAR_FEATURE request has been received. (An interrupt request has been generated.) 0: The CLEAR_FEATURE request has not been received. (Default) This bit is set to 1 upon completion of the status stage. Reference the USFA0CLR register to identify the target of the request. This bit is not automatically cleared to 0 even if the USFA0CLR register is read by FW.
0	EPHALT	This bit indicates that an endpoint has stalled. 1: The endpoint has stalled. (An interrupt request has been generated.) 0: The endpoint has not stalled. (Default) This bit is also set to 1 when an endpoint has been caused to stall by setting FW. Identify the endpoint that has stalled by referencing the USFA0EPS2 register. This bit is not automatically cleared to 0 even when a CLEAR_FEATURE Endpoint, SET_INTERFACE, or SET_CONFIGURATION request is received. In addition, it is not automatically cleared to 0 if the next SETUP token is received in the case of an Endpoint0 overrun. **Caution** Even if Halt Feature of Endpoint0 is set and an interrupt request is generated, bit 0 of the USFA0EPS2 register is masked and cleared to 0 between when a SET_FEATURE Endpoint0, CLEAR_FEATURE Endpoint0, GET_STATUS Endpoint0 request, or FW-processed request is received and when a SETUP token other than the above is received.

(12) INT status 1 register (USFA0IS1)

This register indicates the interrupt source. If the contents of this register are changed, the USFA0EPCINT.EPC_INT0 bit is set to 1.

If an interrupt request (INTUSFA0I1) is generated from USBF, the FW must read this register to identify the interrupt source.

Each bit of this register is forcibly cleared to 0 when 0 is written to the corresponding bit of the USFA0IC1 register. However, the SUCES and STG bits of the USFA0IS1 register are automatically cleared to 0 when the next SETUP token is received.

Access This register is read-only, in 8-bit units.

Address 044_H

Initial value 00_H. This register is initialized by any reset.

Caution In the USBF, multiple interrupt sources, such as Bus Reset, Resume, and Short, are ORed internally and issued as a single interrupt request (INTUSFA0I1). Therefore, even if multiple interrupt sources occur, they are ORed and issued as an INTUSFA0I1 interrupt request. For example, if a Bus Reset interrupt source and Resume interrupt source occur, the two sources are ORed and an INTUSFA0I1 interrupt request is issued. Under these conditions, if the Bus Reset interrupt source is cleared to 0 (USFA0IC0.BUSRSTC = 0), the INTUSFA0I1 interrupt request for this product might remain set to 1 because the Resume interrupt source will still remain. The new interrupt request flag (US0BIC.US0BIF), therefore, might not be set to 1. In this case, after performing clear processing for each interrupt request with the INTUSFA0I1 interrupt servicing routine, confirm the flag status for the USFA0IS0 and USFA0IS1 registers again, and, if there are any interrupt sources for which the flags are set to 1, clear the flags. (Clear only the applicable bits, not all the bits.)

In addition, if an interrupt from the USBF is detected, the value of the interrupt request is stored in the EICn.EIRFn bit (n = 187 to 189). Therefore, even if the CPU acknowledges the interrupt and the EIRFn bit of the interrupt control register (EIC187 to EIC189) is cleared to 0, the EICn.EIRFn bit is immediately set to 0 and an interrupt is continuously generated. To prevent this from happening, execute the following processing in the interrupt service routine, and forcibly clear the EICn.EIRFn bit to 0.

- For EIC187 (when a bridge interrupt is detected)
  - (1) Clear the interrupt source flag in the H-bus bridge interrupt status register (USFA0BRGINT) to 0.
  - (2) Clear the EIC187.EIRF187 bit to 0.
- For EIC188 (when an EPC interrupt is detected)
  - (1) Clear the interrupt source flag in the INT status x register (USFA0ISx; x = 0 to 4) to 0.
  - (2) Clear the interrupt source flag in the EPC interrupt status register (USFA0EPCINT) to 0.
  - (3) Clear the EIC188.EIRF188 bit to 0.
- For EIC189 (when USB resume is detected)
  - (1) Clear the EIC189.EIRF189 bit to 0.

	7	6	5	4	3	2	1	0
USFA0IS1	0	E0IN	E0INDT	E0ODT	SUCES	STG	PROT	CPUDEC
	R	R	R	R	R	R	R	R

Table 30-24 USFA0IS1 register contents (1/2)

Bit position	Bit name	Function
6	E0IN	This bit indicates that an IN token for Endpoint0 has been received and the hardware has automatically transmitted NAK. 1: An IN token has been received and NAK has been transmitted. (An interrupt request has been generated.) 0: No IN token has been received. (Default)
5	E0INDT	This bit indicates that data has been correctly transmitted from the USFA0E0W register. 1: Transmission from the USFA0E0W register is complete. (An interrupt request has been generated.) 0: Transmission from the USFA0E0W register is not complete. (Default) Data is transmitted in synchronization with the IN token next to the one that set the EP0NKW bit of the USFA0E0N register to 1. This bit is automatically set to 1 by hardware when the host correctly receives that data. This bit is set to 1 even when the data is a Null packet. This bit is automatically cleared to 0 by hardware when the USFA0E0W register is written to the first time.
4	E0ODT	This bit indicates that data has been correctly received in the USFA0E0R register. 1: There is data in the USFA0E0R register. (An interrupt request has been generated.) 0: There is no data in the USFA0E0R register. (Default) This bit is automatically set to 1 by hardware when data has been correctly received. At the same time, the EP0R bit of the USFA0EPS0 register is set to 1. If a Null packet has been received, this bit is not set to 1. It is automatically cleared to 0 by hardware when the FW reads the USFA0E0R register and the value of the USFA0E0L register becomes 0.
3	SUCES	This bit indicates that either an FW-processed or hardware-processed request has been received and the status stage has been correctly completed. 1: A control transfer has been correctly processed. (An interrupt request has been generated.) 0: Control transfer processing has not ended correctly. (Default) This bit is set to 1 upon completion of the status stage. It is automatically cleared to 0 by hardware when the next SETUP token is received. This bit is also set to 1 when data with Data PID of 0 (Null data) is received in the status stage of a control transfer.
2	STG	This bit is set to 1 when the stage of a control transfer changes to the status stage. It is valid for both FW-processed and hardware-processed requests. This bit is also set to 1 when the stage of a control transfer (without data) changes to the status stage. 1: The system entered the status stage. (An interrupt request has been generated.) 0: The system has not entered the status stage. (Default) This bit is automatically cleared to 0 by hardware when the next SETUP token is received. It is also set to 1 when the stage of a control transfer changes to the status stage while ACK cannot be correctly received in the data stage. In this case, the EP0NKW bit of the USFA0E0N register is also cleared to 0 at the same time when the USFA0E0W register is cleared, if the FW is processing a control transfer (read).

Table 30-24 USFA0IS1 register contents (2/2)

Bit position	Bit name	Function
1	PROT	This bit indicates that a SETUP token has been received. It is valid for both FW-processed and hardware-processed requests. 1: A SETUP token has been correctly received. (An interrupt request has been generated.) 0: No SETUP token has been received. (Default) This bit is set to 1 when data has been correctly received in the USFA0E0ST register. Clear this bit to 0 by using FW when the USFA0E0ST register is read the first time. If it is not cleared to 0 by FW, reception of the next SETUP token cannot be correctly recognized. This bit is used to accurately recognize that a SETUP transaction has been executed again during a control transfer. If the SETUP transaction is re-executed during a control transfer and a second request is executed by hardware, the CPUDEC bit is not set to 1, but the PROT bit can be used to recognize the re-execution.
0	CPUDEC	This bit indicates that the USFA0E0ST register has a request that is to be decoded by FW. 1: There is data in the USFA0E0ST register. (An interrupt request has been generated.) 0: There is no data in the USFA0E0ST register. (Default) This bit is automatically cleared to 0 by hardware when all the data of the USFA0E0ST register is read.

(13) INT status 2 register (USFA0IS2)

This register indicates the interrupt source. If the contents of this register are changed, the USFA0EPCINT.EPC_INT1 bit is set to 1.

If an interrupt request (INTUSFA0I1) is generated from USBF, the FW must read this register to identify the interrupt source.

Each bit of this register is forcibly cleared to 0 when 0 is written to the corresponding bit of the USFA0IC2 register.

The related bits are invalid if the corresponding endpoint is not supported according to the setting of the USFA0EnIM register (n = 1, 3, 7, 8) and the current interface.

Access This register is read-only, in 8-bit units.

Address 048_H

Initial value 00_H. This register is initialized by any reset.

	7	6	5	4	3	2	1	0
USFA0IS2	BKI2IN	BKI2DT	BKI1IN	BKI1DT	0	0	IT2DT	IT1DT
	R	R	R	R	R	R	R	R

Table 30-25 USFA0IS2 register contents

Bit position	Bit name	Function
7, 5	BKInIN	These bits indicate that an IN token has been received in the USFA0BIn register (Endpoint m) and that NAK has been returned. 1: An IN token has been received and NAK has been transmitted. (An interrupt request has been generated.) 0: No IN token has been received. (Default)
6, 4	BKInDT	These bits indicate that the FIFO of the USFA0BIn register (Endpoint m) has been toggled. This means that data can be written to Endpoint m. 1: The FIFO has been toggled. (An interrupt request has been generated.) 0: The FIFO has not been toggled. (Default) The data written to Endpoint m is transmitted in synchronization with the IN token next to the one that set the BKInNK bit of the USFA0EN register to 1. When the FIFO has been toggled and then data can be written from the CPU, this bit is automatically set to 1 by hardware. It is also set to 1 when the FIFO has been toggled, even if the data is a Null packet. This bit is automatically cleared to 0 by hardware when the USFA0BIn register is written to the first time.
1, 0	ITnDT	These bits indicate that data has been correctly received from the USFA0INTn register (Endpoint x). 1: Transmission is complete. (An interrupt request has been generated.) 0: Transmission is not complete. (Default) Data is transmitted in synchronization with the IN token next to the one that set the ITnNK bit of the USFA0EN register to 1. These bits are automatically set to 1 by hardware when the host has correctly received the data. These bits are automatically cleared to 0 by hardware when the USFA0INTn register is written to the first time. They are set to 1 even when the data is a Null packet.

Note n = 1 or 2

m = 1 and x = 7 when n = 1

m = 3 and x = 8 when n = 2

(14) INT status 3 register (USFA0IS3)

This register indicates the interrupt source. If the contents of this register are changed, the USFA0EPCINT.EPC_INT1 bit is set to 1.

If an interrupt request (INTUSFA0I1) is generated from USBF, the FW must read this register to identify the interrupt source.

Each bit of this register is forcibly cleared to 0 when 0 is written to the corresponding bit of the USFA0IC3 register.

The related bits are invalid if the corresponding endpoint is not supported according to the setting of the USFA0EnIM register (n = 2 or 4) and the current interface.

Access This register is read-only, in 8-bit units.

Address $04C_H$

Initial value 00_H. This register is initialized by any reset.

	7	6	5	4	3	2	1	0
USFA0IS3	BKO2FL	BKO2NL	BKO2NAK	BKO2DT	BKO1FL	BKO1NL	BKO1NAK	BKO1DT
	R	R	R	R	R	R	R	R

Table 30-26 USFA0IS3 register contents (1/2)

Bit position	Bit name	Function
7, 3	BKOnFL	These bits indicate that data has been correctly received in the USFA0BOn register (Endpoint m) and the FIFOs of both the CPU and SIE hold the data. 1: Received data is in both FIFOs of the USFA0BOn register. (An interrupt request has been generated.) 0: Received data is not in the FIFO on the SIE side of the USFA0BOn register. (Default). If data is held in the FIFOs of both the CPU and SIE, these bits are automatically set to 1 by hardware. They are automatically cleared to 0 by hardware when the FIFO is toggled.
6, 2	BKOnNL	These bits indicate that a Null packet (a packet with a length of 0) has been received in the USFA0BOn register (Endpoint m). 1: A Null packet has been received. (An interrupt request has been generated.) 0: No Null packet has been received. (Default) These bits are set to 1 immediately after the reception of a Null packet when the FIFO is empty. They are set to 1 when the FIFO on the CPU side has been completely read if data is in that FIFO.
5, 1	BKOnNAK	These bits indicate that an OUT token has been received in the USFA0BOn register (Endpoint m) and NAK has been returned. 1: An OUT token has been received and NAK has been transmitted. (An interrupt request has been generated.) 0: No OUT token has been received. (Default)

Table 30-26 USFA0IS3 register contents (2/2)

Bit position	Bit name	Function
4, 0	BKOnDT	These bits indicate that data has been correctly received in the USFA0BOn register (Endpoint m). 1: Reception finished correctly. (An interrupt request has been generated.) 0: Reception is not complete. (Default) These bits are automatically set to 1 by hardware when data has been correctly received and the FIFO has been toggled. At the same time, the corresponding bits of the USFA0EPS0 register are set to 1. They are not set to 1 when the data is a Null packet. These bits are automatically cleared to 0 by hardware when the value of the USFA0BOnL register becomes 0 as a result of reading the USFA0BOn register using FW. These bits are automatically cleared to 0 when all the contents of the FIFO on the CPU side have been read. However, the interrupt request is not cleared if data is in the FIFO on the SIE side at this time, and the INTUSFA0I2 signal does not become inactive. The signal is kept active if data is successively received.

Note n = 1 or 2

m = 2 when n = 1

m = 4 when n = 2

(15) INT status 4 register (USFA0IS4)

This register indicates the interrupt source. If the contents of this register are changed, USFA0EPCINT.EPC_INT2 is set to 1.

If an interrupt request (INTUSFA0I1) is generated from USBF, the FW must read this register to identify the interrupt source.

Each bit of this register is forcibly cleared to 0 when 0 is written to the corresponding bit of the USFA0IC4 register.

The related bits are invalid if the corresponding endpoint is not supported according to the setting of the USFA0EnIM register (n = 1 to 4, 7, 8) and the current interface.

Access This register is read-only, in 8-bit units.

Address 050_H

Initial value 00_H. This register is initialized by any reset.

	7	6	5	4	3	2	1	0
USFA0IS4	0	0	SETINT	0	0	0	0	0
	R	R	R	R	R	R	R	R

Table 30-27 USFA0IS4 register contents

Bit position	Bit name	Function
5	SETINT	This bit indicates that the SET_INTERFACE request has been received and automatically processed. 1: The request was automatically processed. (An interrupt request has been generated.) 0: No request was automatically processed. (Default) The current setting of this bit can be identified by reading the USFA0ASS or USFA0IFn register (n = 0 to 4).

(16) INT mask 0 register (USFA0IM0)

This register controls the masking of the interrupt sources indicated by the USFA0IS0 register.

FW can mask the occurrence of an interrupt request from USBF (INTUSFA0I1) by writing 1 to the corresponding bit of this register. The status is applied even when interrupt requests are masked.

Access This register can be read or written in 8-bit units.

Address $05C_H$

Initial value 00_H. This register is initialized by any reset.

	7	6	5	4	3	2	1	0
USFA0IM0	BUSRSTM	RSUSPDM	0	SHORTM	DMAEDM	SETRQM	CLRRQM	EPHALTM
	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W

Table 30-28 USFA0IM0 register contents

Bit position	Bit name	Function
7	BUSRSTM	This bit masks the Bus Reset interrupt. 1: Mask the interrupt. 0: Do not mask the interrupt. (Default)
6	RSUSPDM	This bit masks the Resume/Suspend interrupt. 1: Mask the interrupt. 0: Do not mask the interrupt. (Default)
4	SHORTM	This bit masks the Short interrupt 1: Mask the interrupt. 0: Do not mask the interrupt. (Default)
3	DMAEDM	This bit masks the DMA_END interrupt. 1: Mask the interrupt. 0: Do not mask the interrupt. (Default)
2	SETRQM	This bit masks the SET_RQ interrupt. 1: Mask the interrupt. 0: Do not mask the interrupt. (Default)
1	CLRRQM	This bit masks the CLR_RQ interrupt. 1: Mask the interrupt. 0: Do not mask the interrupt. (Default)
0	EPHALTM	This bit masks the EP_Halt interrupt. 1: Mask the interrupt. 0: Do not mask the interrupt. (Default)

(17) INT mask 1 register (USFA0IM1)

This register controls the masking of the interrupt sources indicated by the USFA0IS1 register.

FW can mask the occurrence of an interrupt request from USBF (INTUSFA0I1) by writing 1 to the corresponding bit of this register. The status is applied even when interrupt requests are masked.

Access This register can be read or written in 8-bit units.

Address 060_H

Initial value 00_H. This register is initialized by any reset.

	7	6	5	4	3	2	1	0
USFA0IM1	0	E0INM	E0INDTM	E0ODTM	SUCESM	STGM	PROTM	CPUDECM
	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30-29 USFA0IM1 register contents

Bit position	Bit name	Function
6	E0INM	This bit masks the EP0IN interrupt. 1: Mask the interrupt. 0: Do not mask the interrupt. (Default)
5	E0INDTM	This bit masks the EP0INDT interrupt. 1: Mask the interrupt. 0: Do not mask the interrupt. (Default)
4	E0ODTM	This bit masks the EP0OUTDT interrupt. 1: Mask the interrupt. 0: Do not mask the interrupt. (Default)
3	SUCESM	This bit masks the Success interrupt. 1: Mask the interrupt. 0: Do not mask the interrupt. (Default)
2	STGM	This bit masks the Stg interrupt. 1: Mask the interrupt. 0: Do not mask the interrupt. (Default)
1	PROTM	This bit masks the Protect interrupt. 1: Mask the interrupt. 0: Do not mask the interrupt. (Default)
0	CPUDECM	This bit masks the CPUDEC interrupt. 1: Mask the interrupt. 0: Do not mask the interrupt. (Default)

(18) INT mask 2 register (USFA0IM2)

This register controls the masking of the interrupt sources indicated by the USFA0IS2 register.

FW can mask the occurrence of an interrupt request from USBF (INTUSFA0I1) by writing 1 to the corresponding bit of this register. The status is applied even when interrupt requests are masked.

The related bits are invalid if the corresponding endpoint is not supported according to the setting of the USFA0EnIM register (n = 1, 3, 7, 8) and the current interface.

Access This register can be read or written in 8-bit units.

Address 064_H

Initial value 00_H. This register is initialized by any reset.

	7	6	5	4	3	2	1	0
USFA0IM2	BKI2INM	BKI2DTM	BKI1INM	BKI1DTM	0	0	IT2DTM	IT1DTM
	R/W	R/W	R/W	R/W	R	R	R/W	R/W

Table 30-30 USFA0IM2 register contents

Bit position	Bit name	Function
7, 5	BKInINM	These bits mask the BLKInIN interrupt. 1: Mask the interrupt. 0: Do not mask the interrupt. (Default)
6, 4	BKInDTM	These bits mask the BLKInDT interrupt. 1: Mask the interrupt. 0: Do not mask the interrupt. (Default)
1, 0	ITnDTM	These bits mask the INTnDT interrupt. 1: Mask the interrupt. 0: Do not mask the interrupt. (Default)

Note n = 1 or 2

(19) INT mask 3 register (USFA0IM3)

This register controls the masking of the interrupt sources indicated by the USFA0IS3 register.

FW can mask the occurrence of an interrupt request from USBF (INTUSFA0I1) by writing 1 to the corresponding bit of this register. The status is applied even when interrupt requests are masked.

The related bits are invalid if the corresponding endpoint is not supported according to the setting of the USFA0EnIM register (n = 2 or 4) and the current interface.

Access This register can be read or written in 8-bit units.

Address 068_H

Initial value 00_H. This register is initialized by any reset.

	7	6	5	4	3	2	1	0
USFA0IM3	BKO2FLM	BKO2NLM	BKO2NAKM	BKO2DTM	BKO1FLM	BKO1NLM	BKO1NAKM	BKO1DTM
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30-31 USFA0IM3 register contents

Bit position	Bit name	Function
7, 3	BKOnFLM	These bits mask the BLKOnFL interrupt. 1: Mask the interrupt. 0: Do not mask the interrupt. (Default)
6, 2	BKOnNLM	These bits mask the BLKOnNL interrupt. 1: Mask the interrupt. 0: Do not mask the interrupt. (Default)
5, 1	BKOnNAKM	These bits mask the BLKOnNK interrupt. 1: Mask the interrupt. 0: Do not mask the interrupt. (Default)
4, 0	BKOnDTM	These bits mask the BLKOnDT interrupt. 1: Mask the interrupt. 0: Do not mask the interrupt. (Default)

Note n = 1 or 2

(20) INT mask 4 register (USFA0IM4)

This register controls the masking of the interrupt sources indicated by the USFA0IS4 register.

FW can mask the occurrence of an interrupt request from USBF (INTUSFA0I1) by writing 1 to the corresponding bit of this register. The status is applied even when interrupt requests are masked.

The related bits are invalid if the corresponding endpoint is not supported according to the setting of the USFA0EnIM register (n = 1 to 4, 7, 8) and the current interface.

Access This register can be read or written in 8-bit units.

Address $06C_H$

Initial value 00_H. This register is initialized by any reset.

	7	6	5	4	3	2	1	0
USFA0IM4	0	0	SETINTM	0	0	0	0	0
	R	R	R/W	R	R	R	R	R

Table 30-32 USFA0IM4 register contents

Bit position	Bit name	Function
5	SETINTM	This bit masks the SET_INT interrupt. 1: Mask 0: Do not mask the interrupt. (Default)

(21) INT clear 0 register (USFA0IC0)

This register controls clearing the interrupt sources indicated by the USFA0IS0 register.

FW can clear an interrupt source by writing 0 to the corresponding bit of this register. Even a bit that is automatically cleared to 0 by hardware can be cleared by FW before it is cleared by hardware. Writing 0 to a bit of this register automatically sets the bit to 1. Writing 1 to the bit is invalid.

Access This register is write-only, in 8-bit units. If this register is read, FF_H is read.

Address 078_H

Initial value FF_H. This register is initialized by any reset.

	7	6	5	4	3	2	1	0
USFA0IC0	BUSRSTC	RSUSPDC	1	SHORTC	DMAEDC	SETRQC	CLRRQC	EPHALTC
	W	W	W	W	W	W	W	W

Table 30-33 USFA0IC0 register contents

Bit position	Bit name	Function
7	BUSRSTC	This bit clears the Bus Reset interrupt. 0: Clear the interrupt.
6	RSUSPDC	This bit clears the Resume/Suspend interrupt. 0: Clear the interrupt.
4	SHORTC	This bit clears the Short interrupt. 0: Clear the interrupt.
3	DMAEDC	This bit clears the DMA_END interrupt. 0: Clear the interrupt.
2	SETRQC	This bit clears the SET_RQ interrupt. 0: Clear the interrupt.
1	CLRRQC	This bit clears the CLR_RQ interrupt. 0: Clear the interrupt.
0	EPHALTC	This bit clears the EP_Halt interrupt. 0: Clear the interrupt.

(22) INT clear 1 register (USFA0IC1)

This register controls clearing the interrupt sources indicated by the USFA0IS1 register.

FW can clear an interrupt source by writing 0 to the corresponding bit of this register. Even a bit that is automatically cleared to 0 by hardware can be cleared by FW before it is cleared by hardware. Writing 0 to a bit of this register automatically sets the bit to 1. Writing 1 to the bit is invalid.

Access This register is write-only, in 8-bit units. If this register is read, FF_H is read.

Address $07C_H$

Initial value FF_H. This register is initialized by any reset.

	7	6	5	4	3	2	1	0
USFA0IC1	1	E0INC	E0INDTC	E0ODTC	SUCESC	STGC	PROTC	CPUDECC
	W	W	W	W	W	W	W	W

Table 30-34 USFA0IC1 register contents

Bit position	Bit name	Function
6	E0INC	This bit clears the EP0IN interrupt. 0: Clear the interrupt.
5	E0INDTC	This bit clears the EP0INDT interrupt. 0: Clear the interrupt.
4	E0ODTC	This bit clears the EP0OUTDT interrupt. 0: Clear the interrupt.
3	SUCESC	This bit clears the Success interrupt. 0: Clear the interrupt.
2	STGC	This bit clears the Stg interrupt. 0: Clear the interrupt.
1	PROTC	This bit clears the Protect interrupt. 0: Clear the interrupt.
0	CPUDECC	This bit clears the CPUDEC interrupt. 0: Clear the interrupt.

(23) INT clear 2 register (USFA0IC2)

This register controls clearing the interrupt sources indicated by the USFA0IS2 register.

FW can clear an interrupt source by writing 0 to the corresponding bit of this register. Even a bit that is automatically cleared to 0 by hardware can be cleared by FW before it is cleared by hardware. Writing 0 to a bit of this register automatically sets the bit to 1. Writing 1 to the bit is invalid.

The related bits are invalid if the corresponding endpoint is not supported according to the setting of the USFA0EnIM register (n = 1, 3, 7, 8) and the current interface.

Access This register is write-only, in 8-bit units. If this register is read, FF_H is read.

Address 080_H

Initial value FF_H. This register is initialized by any reset.

	7	6	5	4	3	2	1	0
USFA0IC2	BKI2INC	BKI2DTC	BKI1INC	BKI1DTC	1	1	IT2DTC	IT1DTC
	W	W	W	W	W	W	W	W

Table 30-35 USFA0IC2 register contents

Bit position	Bit name	Function
7, 5	BKInINC	These bits clear the BLKInIN interrupt. 0: Clear the interrupt.
6, 4	BKInDTC	These bits clear the BLKInDT interrupt. 0: Clear the interrupt.
1, 0	ITnDTC	These bits clear the INTnDT interrupt. 0: Clear the interrupt.

Note n = 1 or 2

(24) INT clear 3 register (USFA0IC3)

This register controls clearing the interrupt sources indicated by the USFA0IS3 register.

FW can clear an interrupt source by writing 0 to the corresponding bit of this register. Even a bit that is automatically cleared to 0 by hardware can be cleared by FW before it is cleared by hardware. Writing 0 to a bit of this register automatically sets the bit to 1. Writing 1 to the bit is invalid.

The related bits are invalid if the corresponding endpoint is not supported according to the setting of the USFA0EnIM register (n = 2 or 4) and the current interface.

Access This register is write-only, in 8-bit units. If this register is read, FF_H is read.

Address 084_H

Initial value FF_H. This register is initialized by any reset.

	7	6	5	4	3	2	1	0
USFA0IC3	BKO2FLC	KO2NLC	BKO2NAKC	BKO2DTC	BKO1FLC	BKO1NLC	BKO1NAKC	BKO1DTC
	W	W	W	W	W	W	W	W

Table 30-36 USFA0IC3 register contents

Bit position	Bit name	Function
7, 3	BKOnFLC	These bits clear the BLKOnFL interrupt. 0: Clear the interrupt.
6, 2	BKOnNLC	These bits clear the BLKOnNL interrupt. 0: Clear the interrupt.
5, 1	BKOnNAKC	These bits clear the BLKOnNK interrupt. 0: Clear the interrupt.
4, 0	BKOnDTC	These bits clear the BLKOnDT interrupt. 0: Clear the interrupt.

Note n = 1 or 2

(25) INT clear 4 register (USFA0IC4)

This register controls clearing the interrupt sources indicated by the USFA0IS4 register.

FW can clear an interrupt source by writing 0 to the corresponding bit of this register. Even a bit that is automatically cleared to 0 by hardware can be cleared by FW before it is cleared by hardware. Writing 0 to a bit of this register automatically sets the bit to 1. Writing 1 to the bit is invalid.

The related bits are invalid if the corresponding endpoint is not supported according to the setting of the USFA0EnIM register (n = 1 to 4, 7, 8) and the current interface.

Access This register is write-only, in 8-bit units. If this register is read, FF_H is read.

Address 088_H

Initial value FF_H. This register is initialized by any reset.

	7	6	5	4	3	2	1	0
USFA0IC4	1	1	SETINTC	1	1	1	1	1
	W	W	W	W	W	W	W	W

Table 30-37 USFA0IC4 register contents

Bit position	Bit name	Function
5	SETINTC	This bit clears the SET_INT interrupt. 0: Clear the interrupt.

(26) INT & DMARQ register (USFA0IDR)

This register selects reporting by way of an interrupt request or starting DMA.

If data exists in either the USFA0BO1 or USFA0BO2 register, or if data can be written to the USFA0BI1 or USFA0BI2 register, this register selects whether it is reported to the FW by an interrupt request, or whether starting DMA is requested. If starting DMA is requested, the DMA transfer mode can be selected according to the setting of bits 0 and 1.

The related bits are invalid if the corresponding endpoint is not supported according to the setting of the USFA0EnIM register (n = 1 to 4) and the current interface.

Be sure to clear bits 3 and 2 to “0”. If they are set to 1, the operation is not guaranteed.

Access This register can be read or written in 8-bit units.

Address 098_H

Initial value 00_H. This register is initialized by any reset.

Caution If the target endpoint is not supported by the SET_INTERFACE request during a DMA transfer, the DMA request signal becomes inactive immediately, and the corresponding bit is automatically cleared to 0 by hardware.

	7	6	5	4	3	2	1	0
USFA0IDR	DQBI2MS	DQBI1MS	DQBO2MS	DQBO1MS	0	0	MODE1	MODE0
	R/W	R/W	R/W	R/W	R	R	R/W	R/W

Table 30-38 USFA0IDR register contents (1/2)

Bit position	Bit name	Function
7, 6	DQBInMS	These bits enable a write DMA transfer request (DMA request signal for Endpoint m) to the USFA0BIn register. When these bits are set to 1, the DMA request signal for Endpoint m becomes active while writing data can be acknowledged. If the DMA end signal for Endpoint m is input (if the DMA controller issues TC), these bits are automatically cleared to 0 by hardware. To continue DMA transfers, re-set these bits to 1 by FW. 1: Enable the active DMA request signal for Endpoint m (and mask the BKInDT interrupt). 0: Disable the active DMA request signal for Endpoint m. (Default)
5, 4	DQBOnMS	These bits enable a read DMA transfer request (DMA request signal for Endpoint x) to the USFA0BOn register. When these bits are set to 1, the DMA request signal for Endpoint x becomes active if the data to be read is prepared in the USFA0BOn register. If the DMA end signal for Endpoint x is input (if the DMA controller issues TC), these bits are automatically cleared to 0 by hardware. They are also cleared to 0 when the USBSPxB signal is active. To continue DMA transfers, re-set these bits to 1 by FW. 1: Enable the active DMA request signal for Endpoint x (and mask the BKOnDT interrupt). 0: Disable the active DMA request signal for Endpoint x. (Default)

Table 30-38 USFA0IDR register contents (2/2)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>1, 0</td><td>MODE1,
MODE0</td><td>These bits select the DMA transfer mode.
<table>
<tr><th>MODE1</th><th>MODE0</th><th>Mode</th><th>Note</th></tr>
<tr><td>1</td><td>0</td><td>Demand mode</td><td>The DMA request signal is active as long as there is data. It becomes inactive if there is no more data.</td></tr>
<tr><td colspan="2">Other than the above</td><td colspan="2">Setting prohibited</td></tr>
</table></td></tr>
</table>

Note n = 1 or 2

m = 1 and x = 2 when n = 1

m = 3 and x = 4 when n = 2

(27) DMA status 0 register (USFA0DMS0)

This register indicates the DMA status of Endpoint1 to Endpoint4.

The related bits are invalid if the corresponding endpoint is not supported according to the setting of the USFA0EnIM register (n = 1 to 4) and the current interface.

Access This register is read-only, in 8-bit units.

Address $09C_H$

Initial value 00_H. This register is initialized by any reset.

	7	6	5	4	3	2	1	0
USFA0DMS0	0	0	DQE4	DQE3	DQE2	DQE1	0	0
	R	R	R	R	R	R	R	R

Table 30-39 USFA0DMS0 register contents

Bit position	Bit name	Function
5	DQE4	This bit indicates that a DMA read request is being issued from Endpoint4 to memory. 1: A DMA read request from Endpoint4 is being issued. 0: No DMA read request from Endpoint4 is being issued. (Default)
4	DQE3	This bit indicates that a DMA write request is being issued from memory to Endpoint3. Note that, even if data is in Endpoint3 (when the FIFO is not full and after the BKI2DED bit has been set to 1), the DMA request signal becomes active immediately and the DMA transfer is started when the DQBI2MS bit of the USFA0IDR register is set to 1. 1: A DMA write request for Endpoint3 is being issued. 0: No DMA write request for Endpoint3 is being issued. (Default)
3	DQE2	This bit indicates that a DMA read request is being issued from Endpoint2 to memory. 1: A DMA read request from Endpoint2 is being issued. 0: No DMA read request from Endpoint2 is being issued. (Default)
2	DQE1	This bit indicates that a DMA write request is being issued from memory to Endpoint1. Note that, even if data is in Endpoint1 (when the FIFO is not full and after the BKI1DED bit has been set to 1), the DMA request signal becomes active immediately and the DMA transfer is started when the DQBI1MS bit of the USFA0IDR register is set to 1. 1: A DMA write request for Endpoint1 is being issued. 0: No DMA write request for Endpoint1 is being issued. (Default)

(28) DMA status 1 register (USFA0DMS1)

This register indicates the DMA status of Endpoint1 to Endpoint4.

The related bits are invalid if the corresponding endpoint is not supported according to the setting of the USFA0EnIM register (n = 1 to 4) and the current interface.

Each bit is automatically cleared to 0 when this register is read. Even when this register is read, however, bits 4 and 3 of the USFA0IS0 register are not cleared to 0. If the target endpoint is no longer supported by the SET_INTERFACE request, each bit is automatically cleared to 0 by hardware. (However, the DMA_END interrupt request and Short interrupt request are not cleared.)

Access This register is read-only, in 8-bit units.

Address $0A0_H$

Initial value 00_H. This register is initialized by any reset.

	7	6	5	4	3	2	1	0
USFA0DMS1	DEDE4	DSPE4	DEDE3	DEDE2	DSPE2	DEDE1	0	0
	R	R	R	R	R	R	R	R

Table 30-40 USFA0DMS1 register contents

Bit position	Bit name	Function
7, 5, 4, 2	DEDEn	These bits indicate that the DMA end (TC) signal for Endpoint n becomes active and DMA is stopped while a DMA read request is being issued from Endpoint n to memory. 1: The DMA end signal for Endpoint n is active. 0: The DMA end signal for Endpoint n is inactive. (Default)
6, 3	DSPEm	These bits indicate that, although a DMA read request was being issued from Endpoint m to memory, DMA has been stopped because the received data is a short packet and there is no more data to be transferred. 1: The DMASTOP_EPm signal is active. 0: The DMASTOP_EPm signal is inactive. (Default)

Note n = 1 to 4

m = 2 or 4

(29) FIFO clear 0 register (USFA0FIC0)

This register clears each FIFO.

FW can clear the target FIFO by writing 1 to the corresponding bit of this register. A bit to which 1 has been written is automatically cleared to 0. Writing 0 to the bit is invalid.

The related bits are invalid if the corresponding endpoint is not supported according to the setting of the USFA0EnIM register (n = 1, 3, 7, 8) and the current interface.

Access This register is write-only, in 8-bit units. If this register is read, 00_H is read.

Address $0C0_H$

Initial value 00_H. This register is initialized by any reset.

	7	6	5	4	3	2	1	0
USFA0FIC0	BKI2SC	BKI2CC	BKI1SC	BKI1CC	0	ITR1C	EP0WC	EP0RC
	W	W	W	W	W	W	W	W

Table 30-41 USFA0FIC0 register contents

Bit position	Bit name	Function
7, 5	BKInSC	These bits clear only the FIFO on the SIE side of the USFA0BIn register (reset the counter). 1: Clear the FIFO. Writing to these bits is invalid while an IN token for Endpoint m is being processed with the BKInNK bit set to 1. The BKInNK bit is automatically cleared to 0 by clearing the FIFO. Make sure that the FIFO on the CPU side is empty when these bits are used.
6, 4	BKInCC	These bits clear only the FIFO on the CPU side of the USFA0BIn register (reset the counter). 1: Clear the FIFO.
2	ITR1C	This bit clears the USFA0INT1 register (resets the counter). 1: Clear the register. Writing to these bits is invalid while an IN token for Endpoint 7 is being processed with the IT1NK bit set to 1. The IT1NK bit is automatically cleared to 0 by clearing the FIFO.
1	EP0WC	This bit clears the USFA0E0W register (resets the counter). 1: Clear the register. Writing to this bit is invalid while an IN token for Endpoint0 is being processed with the EP0NKW bit set to 1. The EP0NKW bit is automatically cleared to 0 by clearing the FIFO.
0	EP0RC	This bit clears the USFA0E0R register (resets the counter). 1: Clear the register. When the EP0NKR bit is set to 1 (except when it has been set by FW), the EP0NKR bit is automatically cleared to 0 by clearing the FIFO.

Note n = 1 or 2

m = 1 and x = 7 when n = 1

m = 3 and x = 8 when n = 2

(30) FIFO clear 1 register (USFA0FIC1)

This register clears each FIFO.

FW can clear the target FIFO by writing 1 to the corresponding bit of this register. A bit to which 1 has been written is automatically cleared to 0. Writing 0 to the bit is invalid.

The related bits are invalid if the corresponding endpoint is not supported according to the setting of the USFA0EnIM register (n = 2 or 4) and the current interface.

Access This register is write-only, in 8-bit units. If this register is read, 00_H is read.

Address $0C4_H$

Initial value 00_H. This register is initialized by any reset.

	7	6	5	4	3	2	1	0
USFA0FIC1	0	0	0	0	BKO2C	BKO2CC	BKO1C	BKO1CC
	R	R	R	R	W	W	W	W

Table 30-42 USFA0FIC1 register contents

Bit position	Bit name	Function
3, 1	BKOnC	These bits clear the FIFOs on both the SIE and CPU sides of the USFA0BOn register (reset the counter). 1: Clear the FIFOs. When the BKOnNK bit is set to 1 (except when it has been set by FW), the BKOnNK bit is automatically cleared to 0 by clearing the FIFO.
2, 0	BKOnCC	These bits clear only the FIFO on the CPU side of the USFA0BOn register (reset the counter). 1: Clear the FIFO. When the BKOnNK bit is set to 1 (except when it has been set by FW), the BKOnNK bit is automatically cleared to 0 by clearing the FIFO.

Note n = 1 or 2

(31) Data end register (USFA0DEND)

This register specifies the toggle setting for the transmission system FIFO and reports the end of writing.

By writing 1 to each BKnINT (n = 0 or 1) bit, the FIFO can automatically be toggled when the target endpoint FIFO becomes full.

By writing 1 to the ITnDEND (n = 0 or 1), BKInDEND (n = 0 or 1), and E0DNED bits, a target endpoint data transfer can be started. A bit to which 1 has been written is automatically cleared to 0. Writing 0 to the bit is invalid.

The related bits are invalid if the corresponding endpoint is not supported according to the setting of the USFA0EnIM register (n = 1, 3, 7, 8) and the current interface.

Access This register can be read or written in 8-bit units. (However, bits 4 to 0 are write-only and return 0 when read.)

Address $0D4_H$

Initial value 00_H. This register is initialized by any reset.

	7	6	5	4	3	2	1	0
USFA0DEND	BKI2T	BKI1T	0	IT2DEND	IT1DEND	BKI2DED	BKI1DED	E0DED
	R/W	R/W	R	W	W	W	W	W

Table 30-43 USFA0DEND register contents (1/2)

Bit position	Bit name	Function
7, 6	BKInT	These bits specify whether toggling the FIFO is automatically executed if the FIFO on the CPU side of the USFA0BIn register becomes full as a result of DMA. 1: Automatically toggle the FIFO as soon as the FIFO has become full. 0: Do not automatically toggle the FIFO even if the FIFO becomes full. (Default)
4, 3	ITnDEND	Set these bits to 1 to transmit the data of the USFA0INTn register. When these bits are set to 1, the ITnNK bit is set to 1 and data is transferred. 1: Transmit a short packet. 0: Do not transmit a short packet. (Default) If the ITRnC bit of the USFA0FIC0 register is set to 1 and then these bits are set to 1 (the counter of the USFA0INTn register = 0 and the corresponding bit of the USFA0EPS0 register = 1), a Null packet (with a data length of 0) is transmitted. If data exists in the USFA0INTn register and these bits are set to 1 (the counter of the USFA0INTn register is not 0 and the corresponding bit of the USFA0EPS0 register = 1), a short packet is transmitted. These bits are automatically controlled by hardware when the FIFO is full.

Table 30-43 USFA0DEND register contents (2/2)

Bit position	Bit name	Function
2, 1	BKInDED	Set these bits to 1 when writing transmitted data to the USFA0BIn register has been completed. When these bits are set to 1, the FIFO is toggled as soon as possible, the BKInNK bit is set to 1, and data is transferred. 1: Transmit a short packet. 0: Do not transmit a short packet. (Default) These bits control the FIFO on the CPU side. If the BKInCC bit of the USFA0FIC0 register is set to 1 and then these bits are set to 1 (the counter of the USFA0BIn register = 0), a Null packet (with a data length of 0) is transmitted. If data exists in the USFA0BIn register and these bits are set to 1 (the counter of the USFA0BIn register is not 0), and if the FIFO is not full, a short packet is transmitted. If the FIFO on the CPU side of the USFA0BIn register becomes full as a result of DMA, with the PIO or BKInT bit set to 1, the hardware starts data transmission even if these bits are not set to 1. If the FIFO on the CPU side of the USFA0BIn register becomes full as a result of DMA, with the BKInT bit cleared to 0, be sure to set these bits to 1. (For details, see *(3) "EPNAK register (USFA0EN)"* in *30.7.2 "EPC control registers"*.)
0	E0DED	Set this bit to 1 to transmit the data of the USFA0E0W register. When this bit is set to 1, the EP0NKW bit is set to 1 and data is transferred. 1: Transmit a short packet. 0: Do not transmit a short packet. (Default) If the EP0WC bit of the USFA0FIC0 register is set to 1 and if this bit is set to 1 (the counter of the USFA0E0W register = 0 and bit 1 of the USFA0EPS0 register = 1), a Null packet (with a data length of 0) is transmitted. If data exists in the USFA0E0W register and this bit is set to 1 (the counter of the USFA0E0W register is not 0 and bit 1 of the USFA0EPS0 register = 1), and if the FIFO is not full, a short packet is transmitted.

Note n = 1 or 2

(32) GPR register (USFA0GPR)

This register controls the USBF and USB interface.

FW can reset the USBF by writing 1 to bit 0 of this register. A bit to which 1 has been written is automatically cleared to 0. Writing 0 to the bit is invalid.

Access This register is write-only, in 8-bit units. If this register is read, 00_H is read.

Address $0DC_H$

Initial value 00_H. This register is initialized by any reset.

Caution Be sure to clear bits 7 to 1 to “0”.

	7	6	5	4	3	2	1	0
USFA0GPR	0	0	0	0	0	0	0	MRST
	R	R	R	R	R	R	R	W

Table 30-44 USFA0GPR register contents

Bit position	Bit name	Function
0	MRST	Set this bit to 1 to reset the USBF. 1: Reset the USBF. Actually, USBF is reset two USB clocks after this bit has been set to 1 by FW and the write signal has become inactive. Resetting the USBF by using the MRST bit while the system clock is operating has the same result as resetting by using the $\overline{\text{RESET}}$ pin (hardware reset) (and the register value returns to the default value).

(33) Mode control register (USFA0MODC)

This register controls CPUDEC processing.

By setting each bit of this register, the setting of the USFA0MODS register can be changed. The bit of this register is automatically cleared to 0 only upon a hardware reset and when the MRST bit of the USFA0GRP register has been set to 1.

Even if the bit of this register has automatically been set to 1 by hardware, the setting by FW takes precedence.

Access This register can be read or written in 8-bit units.

Address $0E8_H$

Initial value 00_H. This register is initialized by any reset.

Cautions

1. This register is provided for debugging purposes. Usually, do not set this register except for verifying the operation or when a special mode is used.
2. Be sure to clear bits 7 and 5 to 0 to “0”. If they are set to 1, the operation is not guaranteed.

	7	6	5	4	3	2	1	0
USFA0MODC	0	CDCGDST	0	0	0	0	0	0
	R	R/W	R	R	R	R	R	R

Table 30-45 USFA0MODC register contents

Bit position	Bit name	Function
6	CDCGDST	Set this bit to 1 to switch the GET_DESCRIPTOR Configuration request to CPUDEC processing. By setting this bit to 1, the CDCGD bit of the USFA0MODS register can be forcibly set to 1. 1: Forcibly change the GET_DESCRIPTOR Configuration request to CPUDEC processing (set the CDCGD bit of the USFA0MODS register to 1). 0: Automatically process the GET_DESCRIPTOR Configuration request. (Default)

(34) Mode status register (USFA0MODS)

This register indicates the configuration status.

Access This register is read-only, in 8-bit units.

Address $0F0_H$

Initial value 00_H. This register is initialized by any reset.

	7	6	5	4	3	2	1	0
USFA0MODS	0	CDCGD	0	MPACK	DFLT	CONF	0	0
	R	R	R	R	R	R	R	R

Table 30-46 USFA0MODS register contents

Bit position	Bit name	Function
6	CDCGD	This bit specifies whether CPUDEC processing is performed for the GET_DESCRIPTOR Configuration request. 1: Forcibly change the GET_DESCRIPTOR Configuration request to CPUDEC processing. 0: Automatically process the GET_DESCRIPTOR Configuration request. (Default)
4	MPACK	This bit indicates the transmitted packet size of Endpoint0. 1: Transmit a packet of a size other than 8 bytes. 0: Transmit an 8-byte packet. (Default). This bit is automatically set to 1 by hardware after the GET_DESCRIPTOR Device request has been processed (on normal completion of the status stage). It is not cleared to 0 until the USBF has been reset (and is not cleared to 0 by Bus Reset). If this bit is not set to 1, the hardware transfers only the automatically-executed request in 8-byte units. Therefore, even if data of more than 8 bytes is sent by using the OUT token to be processed by FW before completion of the GET_DESCRIPTOR Device request, the data is correctly received. This bit is ignored if the size of Endpoint0 is 8 bytes.
3	DFLT	This bit indicates the default status (DFLT bit = 1). 1: Responses are enabled. 0: Responses are disabled (never a response). (Default) This bit is automatically set to 1 by Bus Reset. No transaction for an endpoint is responded to until this bit is set to 1.
2	CONF	This bit indicates whether the SET_CONFIGURATION request has been completed. 1: The SET_CONFIGURATION request has been completed. 0: The SET_CONFIGURATION request has not been completed. (Default) This bit is set to 1 when Configuration value = 1 is received by the SET_CONFIGURATION request. Unless this bit is set to 1, access to an endpoint other than Endpoint0 is ignored. This bit is cleared to 0 when Configuration value = 0 is received by the SET_CONFIGURATION request. It is also cleared to 0 when Bus Reset is detected.

(35) Active interface number register (USFA0AIFN)

This register specifies the valid Interface number that correctly responds to the GET/SET_INTERFACE request. Because Interface 0 is always valid, Interfaces 1 to 4 can be selected.

Access This register can be read or written in 8-bit units.

Address 100_H

Initial value 00_H. This register is initialized by any reset.

	7	6	5	4	3	2	1	0
USFA0AIFN	ADDIF	0	0	0	0	0	IFNO1	IFNO0
	R/W	R	R	R	R	R	R/W	R/W

Table 30-47 USFA0AIFN register contents

Bit position	Bit name	Function
7	ADDIF	This bit allows the use of Interfaces with numbers other than 0. 1: Support up to the Interface number specified by the IFNO1 and IFNO0 bits. 0: Support only Interface 0. (Default) Setting bits 1 and 0 of this register is invalid when this bit is not set to 1.
1, 0	IFNO1, IFNO0	These bits specify the range of Interface numbers to be supported.

IFNO1	FNO0	Valid Interface No.
1	1	0, 1, 2, 3, 4
1	0	0, 1, 2, 3
0	1	0, 1, 2
0	0	0, 1

(36) Active alternative setting register (USFA0AAS)

This register specifies a link between the Interface number and Alternative Setting.

For the USBF of this microcontroller, a five-series Alternative Setting (for which Alternate Setting 0, 1, 2, 3, and 4 can be defined) and a two-series Alternative Setting (for which Alternative Setting 0 and 1 can be defined) can be specified for one Interface each.

Access This register can be read or written in 8-bit units.

Address 104_H

Initial value 00_H. This register is initialized by any reset.

	7	6	5	4	3	2	1	0
USFA0AAS	ALT2	IFAL21	IFAL20	ALT2EN	ALT5	IFAL51	IFAL50	ALT5EN
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30-48 USFA0AAS register contents

Bit position	Bit name	Function
7, 3	ALTn	These bits specify whether an n-series Alternative Setting is linked with Interface 0. When these bits are set to 1, the setting of the IFALn1 and IFALn0 bits is invalid. 1: Link an n-series Alternative Setting with Interface 0. 0: Do not link an n-series Alternative Setting with Interface 0. (Default)
6, 5, 2, 1	IFALn1, IFALn0	These bits specify the Interface number to be linked with the n-series Alternative Setting. If the linked Interface number is outside the range specified by the USFA0AIFN register, the n-series Alternative Setting is invalid (ALTnEN bit = 0). IFALn1 / IFALn0 / Interface Number to Link: 1 / 1 / Link Interface 4. 1 / 0 / Link Interface 3. 0 / 1 / Link Interface 2. 0 / 0 / Link Interface 1. Do not link a five-series Alternative Setting and a two-series Alternative Setting with the same Interface number.
4, 0	ALTnEN	These bits validate the n-series Alternative Setting. Unless these bits are set to 1, the setting of the ALTn, IFALn1, and IFALn0 bits is invalid. 1: Validate the n-series Alternative Setting. 0: Do not validate the n-series Alternative Setting. (Default)

Note n = 2 or 5

For example, when the USFA0AIFN register is set to 82_H and the USFA0AAS register is set to 15_H, Interfaces 0, 1, 2, and 3 are valid. Interfaces 0 and 2 support only Alternative Setting 0, Interface 1 supports Alternative Setting 0 and 1, and Interface 3 supports Alternative Setting 0, 1, 2, 3, and 4. With this setting, the requests GET_INTERFACE wIndex = 0/1/2/3, SET_INTERFACE wValue = 0 & wIndex = 0/2, SET_INTERFACE wValue = 0/1 & wIndex = 1, and SET_INTERFACE wValue = 0/1/2/3/4 & wIndex = 3 are automatically responded to, and a STALL response is made to the other GET/SET_INTERFACE requests.

(37) Alternative setting status register (USFA0ASS)

This register indicates the current status of the Alternative Setting.

Check this register when the SET_INT interrupt request has been issued. The value received by the SET_INTERFACE request is applied to the USFA0IFn register (n = 0 to 4) as well as to this register.

Access This register is read-only, in 8-bit units.

Address 108_H

Initial value 00_H. This register is initialized by any reset.

	7	6	5	4	3	2	1	0
USFA0ASS	0	0	0	0	AL5ST3	AL5ST2	AL5ST1	AL2ST
	R	R	R	R	R	R	R	R

Table 30-49 USFA0ASS register contents

Bit position	Bit name	Function
3:1	AL5ST[3:1]	These bits indicate the current status of the five-series Alternative Setting.
0	AL2ST	This bit indicates the current status of the two-series Alternative Setting (the selected Alternative Setting number). 1: Alternative Setting 1 0: Alternative Setting 0

AL5ST3	AL5ST2	AL5ST1	Selected Alternative Setting Number
1	0	0	Alternative Setting 4
0	1	1	Alternative Setting 3
0	1	0	Alternative Setting 2
0	0	1	Alternative Setting 1
0	0	0	Alternative Setting 0

(38) Endpoint1 interface mapping register (USFA0E1IM)

This register specifies the Interface and Alternative Setting for which Endpoint1 is valid.

The setting of this register and the Alternative Setting selected by the SET_INTERFACE request indicate whether Endpoint1 is currently valid, and the hardware determines how the GET_STATUS/CLEAR_FEATURE/ SET_FEATURE Endpoint1 request and the IN transaction to Endpoint1 are responded to, and whether the related bits are valid or invalid.

Access This register can be read or written in 8-bit units.

Address $10C_H$

Initial value 00_H. This register is initialized by any reset.

	7	6	5	4	3	2	1	0
USFA0E1IM	E1EN2	E1EN1	E1EN0	E12AL1	E15AL4	E15AL3	E15AL2	E15AL1
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30-50 USFA0E1IM register contents

Bit position	Bit name	Function
7:5	E1EN[2:0]	These bits set up a link between the interface of Endpoint1 and the two-/five-series Alternative Setting. The endpoint is linked with Alternative Setting 0. The endpoint linked with Alternative Setting 0 cannot be excluded from Alternative Setting 1 to 4. When these bits are set to 110 or 111, they are invalid even if the E12AL1 bit is cleared to 0. If the endpoint is linked, it indicates that Endpoint1 is valid when the CONF bit of the USFA0MODS register is set to 1.
4	E12AL1	This bit validates Endpoint1 when the two-series Alternative Setting and the Alternative Setting of the linked interface are set to 1. 1: Validate the endpoint when Alternative Setting 1 is set with CONF bit = 1. 0: Do not validate the endpoint even when Alternative Setting 1 is set with CONF bit = 1. (Default) This bit is valid when the E15AL4 to E15AL1 bits are 0000.
3:0	E15ALn	These bits validate Endpoint1 when the five-series Alternative Setting and the Alternative Setting of the linked interface are set to n. 1: Validate the endpoint when Alternative Setting n is set with CONF bit = 1. 0: Do not validate the endpoint even when Alternative Setting n is set with CONF bit = 1. (Default)

E1EN2	E1EN1	E1EN0	Link Status
1	1	1	Do not link with an Interface.
1	1	0	
1	0	1	Link with Interface 4 and Alternative Setting 0.
1	0	0	Link with Interface 3 and Alternative Setting 0.
0	1	1	Link with Interface 2 and Alternative Setting 0.
0	1	0	Link with Interface 1 and Alternative Setting 0.
0	0	1	Link with Interface 0 and Alternative Setting 0.
0	0	0	Do not link with an Interface. (Default)

Note n = 1 to 4

(39) Endpoint2 interface mapping register (USFA0E2IM)

This register specifies the Interface and Alternative Setting for which Endpoint2 is valid.

The setting of this register and the Alternative Setting selected by the SET_INTERFACE request indicate whether Endpoint2 is currently valid, and the hardware determines how the GET_STATUS/CLEAR_FEATURE/ SET_FEATURE Endpoint2 request and the OUT transaction to Endpoint2 are responded to, and whether the related bits are valid or invalid.

Access This register can be read or written in 8-bit units.

Address 110_H

Initial value 00_H. This register is initialized by any reset.

	7	6	5	4	3	2	1	0
USFA0E2IM	E2EN2	E2EN1	E2EN0	E22AL1	E25AL4	E25AL3	E25AL2	E25AL1
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30-51 USFA0E2IM register contents

Bit position	Bit name	Function
7:5	E2EN[2:0]	These bits set up a link between the interface of Endpoint2 and the two-/five-series Alternative Setting. The endpoint is linked with Alternative Setting 0. The endpoint linked with Alternative Setting 0 cannot be excluded from Alternative Setting 1 to 4.

E2EN2	E2EN1	E2EN0	Link Status
1	1	1	Do not link with an Interface.
1	1	0	
1	0	1	Link with Interface 4 and Alternative Setting 0.
1	0	0	Link with Interface 3 and Alternative Setting 0.
0	1	1	Link with Interface 2 and Alternative Setting 0.
0	1	0	Link with Interface 1 and Alternative Setting 0.
0	0	1	Link with Interface 0 and Alternative Setting 0.
0	0	0	Do not link with an Interface. (Default)

When these bits are set to 110 or 111, they are invalid even if the E22AL1 bit is cleared to 0.
If the endpoint is linked, it indicates that Endpoint2 is valid when the CONF bit of the USFA0MODS register is set to 1.

4	E22AL1	This bit validates Endpoint2 when the two-series Alternative Setting and the Alternative Setting of the linked interface are set to 1. 1: Validate the endpoint when Alternative Setting 1 is set with CONF bit = 1. 0: Do not validate the endpoint even when Alternative Setting 1 is set with CONF bit = 1. (Default) This bit is valid when the E25AL4 to E25AL1 bits are 0000.
3:0	E25ALn	These bits validate Endpoint2 when the five-series Alternative Setting and the Alternative Setting of the linked interface are set to n. 1: Validate the endpoint when Alternative Setting n is set with CONF bit = 1. 0: Do not validate the endpoint even when Alternative Setting n is set with CONF bit = 1. (Default)

Note n = 1 to 4

(40) Endpoint3 interface mapping register (USFA0E3IM)

This register specifies the Interface and Alternative Setting for which Endpoint3 is valid.

The setting of this register and the Alternative Setting selected by the SET_INTERFACE request indicate whether Endpoint3 is currently valid, and the hardware determines how the GET_STATUS/CLEAR_FEATURE/ SET_FEATURE Endpoint3 request and the IN transaction to Endpoint3 are responded to, and whether the related bits are valid or invalid.

Access This register can be read or written in 8-bit units.

Address 114_H

Initial value 00_H. This register is initialized by any reset.

	7	6	5	4	3	2	1	0
USFA0E3IM	E3EN2	E3EN1	E3EN0	E32AL1	E35AL4	E35AL3	E35AL2	E35AL1
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30-52 USFA0E3IM register contents

Bit position	Bit name	Function
7:5	E3EN[2:0]	These bits set up a link between the interface of Endpoint3 and the two-/five-series Alternative Setting. The endpoint is linked with Alternative Setting 0. The endpoint linked with Alternative Setting 0 cannot be excluded from Alternative Setting 1 to 4.

E3EN2	E3EN1	E3EN0	Link Status
1	1	1	Do not link with an Interface.
1	1	0	
1	0	1	Link with Interface 4 and Alternative Setting 0.
1	0	0	Link with Interface 3 and Alternative Setting 0.
0	1	1	Link with Interface 2 and Alternative Setting 0.
0	1	0	Link with Interface 1 and Alternative Setting 0.
0	0	1	Link with Interface 0 and Alternative Setting 0.
0	0	0	Do not link with an Interface. (Default)

When these bits are set to 110 or 111, they are invalid even if the E32AL1 bit is cleared to 0.
If the endpoint is linked, it indicates that Endpoint3 is valid when the CONF bit of the USFA0MODS register is set to 1.

4	E32AL1	This bit validates Endpoint3 when the two-series Alternative Setting and the Alternative Setting of the linked interface are set to 1. 1: Validate the endpoint when Alternative Setting 1 is set with CONF bit = 1. 0: Do not validate the endpoint even when Alternative Setting 1 is set with CONF bit = 1. (Default) This bit is valid when the E35AL4 to E35AL1 bits are 0000.
3:0	E35ALn	These bits validate Endpoint3 when the five-series Alternative Setting and the Alternative Setting of the linked interface are set to n. 1: Validate the endpoint when Alternative Setting n is set with CONF bit = 1. 0: Do not validate the endpoint even when Alternative Setting n is set with CONF bit = 1. (Default)

Note n = 1 to 4

(41) Endpoint4 interface mapping register (USFA0E4IM)

This register specifies the Interface and Alternative Setting for which Endpoint4 is valid.

The setting of this register and the Alternative Setting selected by the SET_INTERFACE request indicate whether Endpoint4 is currently valid, and the hardware determines how the GET_STATUS/CLEAR_FEATURE/ SET_FEATURE Endpoint4 request and the OUT transaction to Endpoint4 are responded to, and whether the related bits are valid or invalid.

Access This register can be read or written in 8-bit units.

Address 118_H

Initial value 00_H. This register is initialized by any reset.

	7	6	5	4	3	2	1	0
USFA0E4IM	E4EN2	E4EN1	E4EN0	E42AL1	E45AL4	E45AL3	E45AL2	E45AL1
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30-53 USFA0E4IM register contents

Bit position	Bit name	Function
7:5	E4EN[2:0]	These bits set up a link between the interface of Endpoint4 and the two-/five-series Alternative Setting. The endpoint is linked with Alternative Setting 0. The endpoint linked with Alternative Setting 0 cannot be excluded from Alternative Setting 1 to 4. When these bits are set to 110 or 111, they are invalid even if the E42AL1 bit is cleared to 0. If the endpoint is linked, it indicates that Endpoint4 is valid when the CONF bit of the USFA0MODS register is set to 1.
4	E42AL1	This bit validates Endpoint4 when the two-series Alternative Setting and the Alternative Setting of the linked interface are set to 1. 1: Validate the endpoint when Alternative Setting 1 is set with CONF bit = 1. 0: Do not validate the endpoint even when Alternative Setting 1 is set with CONF bit = 1. (Default) This bit is valid when the E45AL4 to E45AL1 bits are 0000.
3:0	E45ALn	These bits validate Endpoint4 when the five-series Alternative Setting and the Alternative Setting of the linked interface are set to n. 1: Validate the endpoint when Alternative Setting n is set with CONF bit = 1. 0: Do not validate the endpoint even when Alternative Setting n is set with CONF bit = 1. (Default)

E4EN2	E4EN1	E4EN0	Link Status
1	1	1	Do not link with an Interface.
1	1	0	
1	0	1	Link with Interface 4 and Alternative Setting 0.
1	0	0	Link with Interface 3 and Alternative Setting 0.
0	1	1	Link with Interface 2 and Alternative Setting 0.
0	1	0	Link with Interface 1 and Alternative Setting 0.
0	0	1	Link with Interface 0 and Alternative Setting 0.
0	0	0	Do not link with an Interface. (Default)

Note n = 1 to 4

(42) Endpoint7 interface mapping register (USFA0E7IM)

This register specifies the Interface and Alternative Setting for which Endpoint7 is valid.

The setting of this register and the Alternative Setting selected by the SET_INTERFACE request indicate whether Endpoint7 is currently valid, and the hardware determines how the GET_STATUS/CLEAR_FEATURE/ SET_FEATURE Endpoint7 request and the IN transaction to Endpoint7 are responded to, and whether the related bits are valid or invalid.

Access This register can be read or written in 8-bit units.

Address 124_H

Initial value 00_H. This register is initialized by any reset.

	7	6	5	4	3	2	1	0
USFA0E7IM	E7EN2	E7EN1	E7EN0	E72AL1	E75AL4	E75AL3	E75AL2	E75AL1
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30-54 USFA0E7IM register contents

Bit position	Bit name	Function
7:5	E7EN[2:0]	These bits set up a link between the interface of Endpoint7 and the two-/five-series Alternative Setting. The endpoint is linked with Alternative Setting 0. The endpoint linked with Alternative Setting 0 cannot be excluded from Alternative Setting 1 to 4. When these bits are set to 110 or 111, they are invalid even if the E72AL1 bit is cleared to 0. If the endpoint is linked, it indicates that Endpoint7 is valid when the CONF bit of the USFA0MODS register is set to 1.
4	E72AL1	This bit validates Endpoint7 when the two-series Alternative Setting and the Alternative Setting of the linked interface are set to 1. 1: Validate the endpoint when Alternative Setting 1 is set with CONF bit = 1. 0: Do not validate the endpoint even when Alternative Setting 1 is set with CONF bit = 1. (Default) This bit is valid when the E75AL4 to E75AL1 bits are 0000.
3:0	E75ALn	These bits validate Endpoint7 when the five-series Alternative Setting and the Alternative Setting of the linked interface are set to n. 1: Validate the endpoint when Alternative Setting n is set with CONF bit = 1. 0: Do not validate the endpoint even when Alternative Setting n is set with CONF bit = 1. (Default)

E7EN2	E7EN1	E7EN0	Link Status
1	1	1	Do not link with an Interface.
1	1	0	
1	0	1	Link with Interface 4 and Alternative Setting 0.
1	0	0	Link with Interface 3 and Alternative Setting 0.
0	1	1	Link with Interface 2 and Alternative Setting 0.
0	1	0	Link with Interface 1 and Alternative Setting 0.
0	0	1	Link with Interface 0 and Alternative Setting 0.
0	0	0	Do not link with an Interface. (Default)

Note n = 1 to 4

(43) Endpoint8 interface mapping register (USFA0E8IM)

This register specifies the Interface and Alternative Setting for which Endpoint8 is valid.

The setting of this register and the Alternative Setting selected by the SET_INTERFACE request indicate whether Endpoint8 is currently valid, and the hardware determines how the GET_STATUS/CLEAR_FEATURE/ SET_FEATURE Endpoint8 request and the IN transaction to Endpoint8 are responded to, and whether the related bits are valid or invalid.

Access This register can be read or written in 8-bit units.

Address 128_H

Initial value 00_H. This register is initialized by any reset.

	7	6	5	4	3	2	1	0
USFA0E8IM	E8EN2	E8EN1	E8EN0	E82AL1	E85AL4	E85AL3	E85AL2	E85AL1
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30-55 USFA0E8IM register contents

Bit position	Bit name	Function
7:5	E8EN[2:0]	These bits set up a link between the interface of Endpoint8 and the two-/five-series Alternative Setting. The endpoint is linked with Alternative Setting 0. The endpoint linked with Alternative Setting 0 cannot be excluded from Alternative Setting 1 to 4.

E8EN2	E8EN1	E8EN0	Link Status
1	1	1	Do not link with an Interface.
1	1	0	
1	0	1	Link with Interface 4 and Alternative Setting 0.
1	0	0	Link with Interface 3 and Alternative Setting 0.
0	1	1	Link with Interface 2 and Alternative Setting 0.
0	1	0	Link with Interface 1 and Alternative Setting 0.
0	0	1	Link with Interface 0 and Alternative Setting 0.
0	0	0	Do not link with an Interface. (Default)

When these bits are set to 110 or 111, they are invalid even if the E82AL1 bit is cleared to 0.
If the endpoint is linked, it indicates that Endpoint8 is valid when the CONF bit of the USFA0MODS register is set to 1.

4	E82AL1	This bit validates Endpoint8 when the two-series Alternative Setting and the Alternative Setting of the linked interface are set to 1. 1: Validate the endpoint when Alternative Setting 1 is set with CONF bit = 1. 0: Do not validate the endpoint even when Alternative Setting 1 is set with CONF bit = 1. (Default) This bit is valid when the E85AL4 to E85AL1 bits are 0000.
3:0	E85ALn	These bits validate Endpoint8 when the five-series Alternative Setting and the Alternative Setting of the linked interface are set to n. 1: Validate the endpoint when Alternative Setting n is set with CONF bit = 1. 0: Do not validate the endpoint even when Alternative Setting n is set with CONF bit = 1. (Default)

Note n = 1 to 4

30.7.3 Data hold registers

(1) EP0 read register (USFA0E0R)

The USFA0E0R register is a 64-byte FIFO that stores the OUT data sent from the host during the data stage of a control transfer to/from Endpoint0.

The hardware automatically transfers data to the USFA0E0R register when it has received the data from the host. When the data has been correctly received, the E0ODT bit of the USFA0IS1 register is set to 1, the USFA0E0L register holds the quantity of the received data, and an interrupt request (INTUSFA0I1) is issued. The USFA0E0L register always updates the received data length while it is receiving data. If the final transfer is correct reception, the interrupt request is generated. If the reception is abnormal, the USFA0E0L register is cleared to 0 and the interrupt request is not generated.

Use FW to read the data held by the USFA0E0R register, up to the value of the amount of data read by the USFA0E0L register. Make sure that all data has been read by using the EP0R bit of the USFA0EPS0 register. (The EP0R bit = 0 when all data has been read.) If the value of the USFA0E0L register is 0, the EP0NKR bit of the USFA0E0N register is cleared to 0, and the USFA0E0R register is ready for reception. The USFA0E0R register is cleared when the next SETUP token has been received.

Access This register is read-only, in 8-bit units. Writing to this register is ignored.

Address 200_H

Initial value Undefined.

Caution Be sure to read all the stored data. Clear the FIFO to discard some data.

	7	6	5	4	3	2	1	0
USFA0E0R	E0R7	E0R6	E0R5	E0R4	E0R3	E0R2	E0R1	E0R0
	R	R	R	R	R	R	R	R

Table 30-56 USFA0E0R register contents

Bit position	Bit name	Function
7:0	E0R[7:0]	These bits store the OUT data sent from the host during the data stage of a control transfer to/from Endpoint0.

The operation of the USFA0E0R register is shown below.

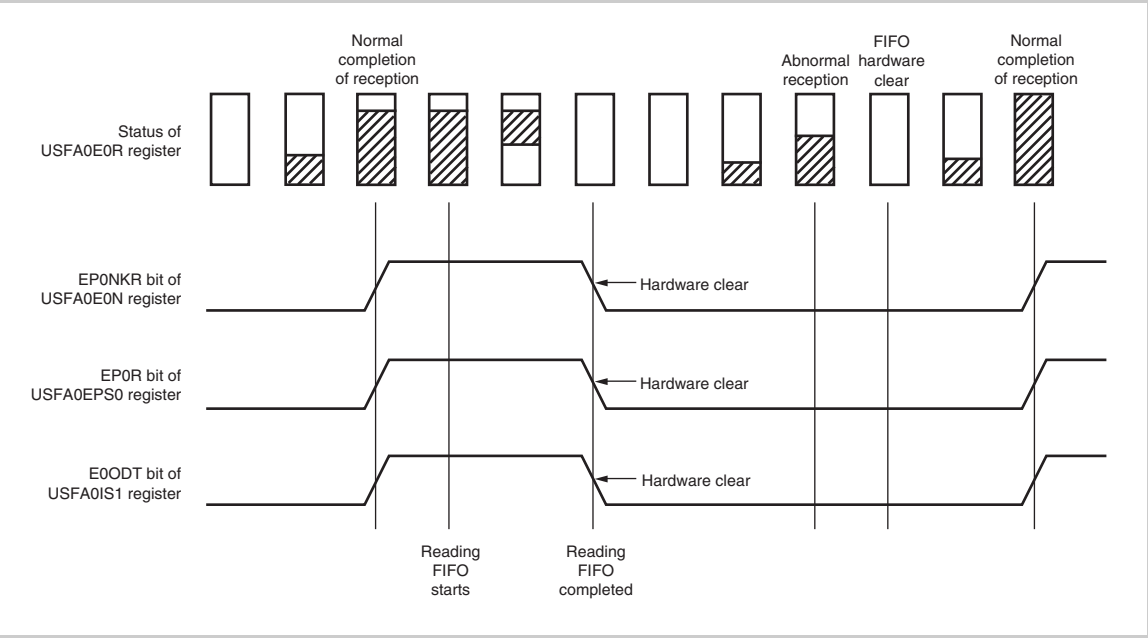


Figure 30-4 USFA0E0R register operation

(2) EP0 length register (USFA0E0L)

The USFA0E0L register stores the length of the data held by the USFA0E0R register.

The USFA0E0L register always updates the received data length while it is receiving data. If the final transfer is abnormal reception, the USFA0E0L register is cleared to 0 and the interrupt request is not generated. If the reception is normal, the interrupt request is generated, and FW can read as much data from the USFA0E0R register as the value read from the USFA0E0L register. The value of the USFA0E0L register is decremented each time the USFA0E0R register has been read.

Access This register is read-only, in 8-bit units. Writing to this register is ignored.

Address 204_H

Default value 00_H. This register is initialized by any reset.

	7	6	5	4	3	2	1	0
USFA0E0L	E0L7	E0L6	E0L5	E0L4	E0L3	E0L2	E0L1	E0L0
	R	R	R	R	R	R	R	R

Table 30-57 USFA0E0L register contents

Bit position	Bit name	Function
7:0	E0L[7:0]	These bits store the length of the data held by the USFA0E0R register.

(3) EP0 setup register (USFA0E0ST)

The USFA0E0ST register holds the SETUP data sent from the host.

The USFA0E0ST register always writes data when a SETUP transaction has been received. The hardware sets the PROT bit of the USFA0IS1 register when it has correctly received the SETUP transaction. It sets the CPUDEC bit of the USFA0IS1 register in the case of an FW-processed request. Next, an interrupt request (INTUSFA0I1) is issued. In the case of an FW-processed request, be sure to read the request in 8-byte units. If it is not read in 8-byte units, the subsequent requests cannot be correctly decoded. The read counter of the USFA0E0ST register is not cleared even when Bus Reset is received.

Always read this counter in 8-byte units regardless of whether Bus Reset is received. Because the USFA0E0ST register always enables writing, the hardware overwrites data to this register even if a SETUP transaction is received while the data of the register is being read. Even if the SETUP transaction cannot be correctly received, the CPUDEC interrupt request and Protect interrupt request are not generated, but the previous data is discarded. If a SETUP token of less than 8 bytes is received, however, the received SETUP token is discarded, and the previously received SETUP data is retained. If the SETUP token is received more than once when one control transfer is executed, be sure to check the PROT bit of the USFA0IS1 register under the conditions below. If PROT bit = 1, read the USFA0E0ST register again because the SETUP transaction has been received more than once.

Access This register is read-only, in 8-bit units. Writing to this register is ignored.

Address 208_H

Initial value 00_H. This register is initialized by any reset.

(1) If a request is decoded by FW and the USFA0E0R register is read or the USFA0E0W register is written
(2) When preparing for a STALL response for a request to which the decode result does not correspond

Caution Be sure to read all the stored data. The USFA0E0ST register is always updated by the request in the SETUP transaction.

	7	6	5	4	3	2	1	0
USFA0E0ST	E0S7	E0S6	E0S5	E0S4	E0S3	E0S2	E0S1	E0S0
	R	R	R	R	R	R	R	R

Table 30-58 USFA0E0ST register contents

Bit position	Bit name	Function
7:0	E0S[7:0]	These bits hold the SETUP data sent from the host.

The operation of the USFA0E0ST register is shown below.

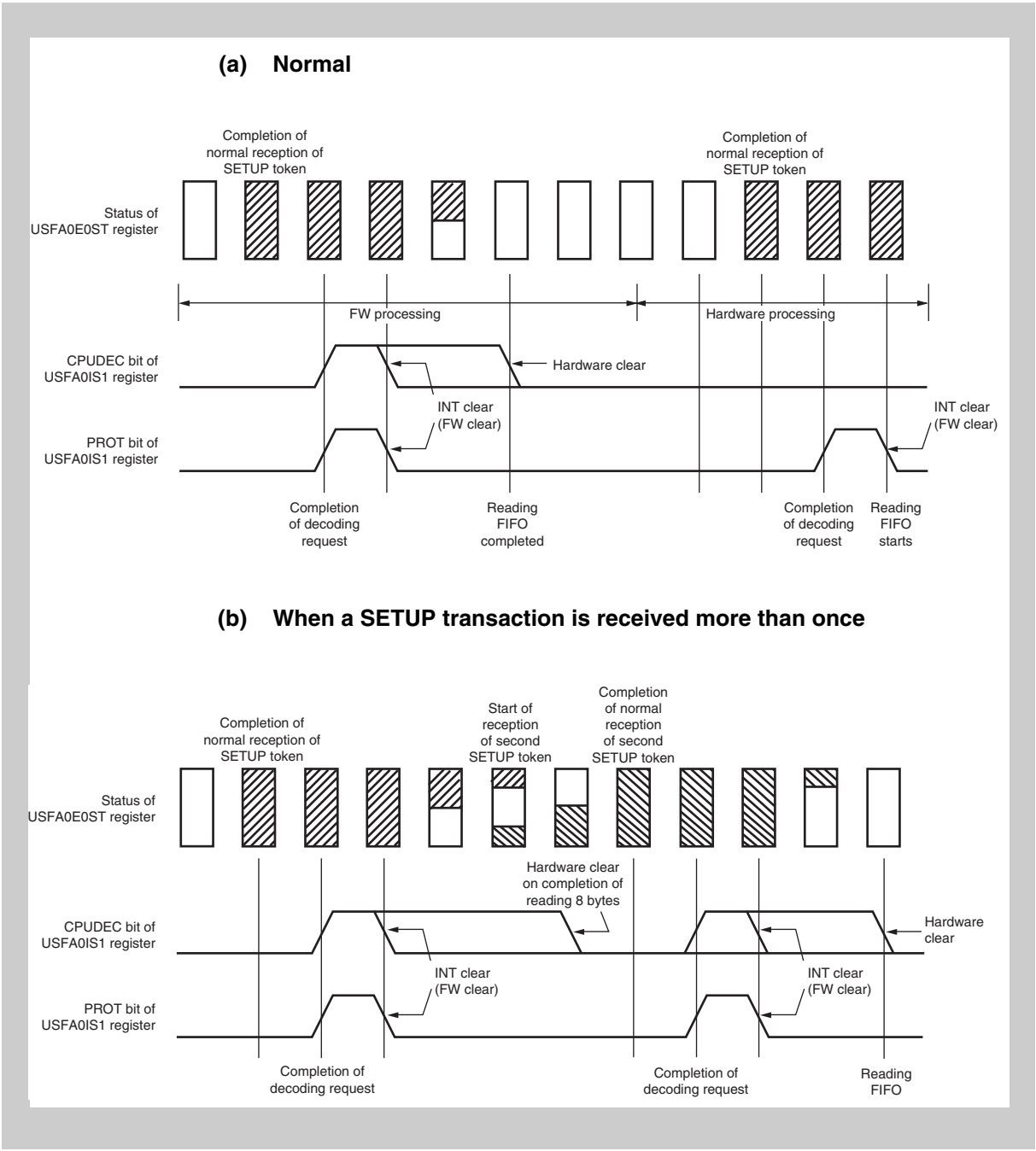


Figure 30-5 USFA0E0ST register operation

(4) EP0 write register (USFA0E0W)

The USFA0E0W register is a 64-byte FIFO that stores the IN data (passes it to SIE) sent to the host during the data stage to Endpoint0.

The hardware transmits data to the USB bus in synchronization with an IN token only when the EP0NKW bit of the USFA0E0N register is set to 1 (when NAK is not transmitted). When data is transmitted and when the host correctly receives the data, the EP0NKW bit of the USFA0E0N register is automatically cleared to 0 by hardware. A short packet is transmitted when data is written to the USFA0E0W register and the E0DED bit of the USFA0DEND register is set to 1 (EP0W bit of the USFA0EPS0 register = 1 (data exists)). A Null packet is transmitted when the USFA0E0W register is cleared and the E0DED bit of the USFA0DEND register is set to 1 (EP0W bit of the USFA0EPS0 register = 1 (data exists)).

The USFA0E0W register is cleared to 0 when the next SETUP token is received before transmission has been completed. If the stage of a control transfer (read) changes to the status stage while ACK has not been correctly received in the data stage, the USFA0E0W register is automatically cleared to 0. It is also cleared to 0 if the EP0NKW bit of the USFA0E0N register is 1 at the same time.

If the USFA0E0W register is read while no data is in it, 00_H is read.

Access This register is write-only, in 8-bit units. If this register is read, 00H is read.

Address $20C_H$

Initial value Undefined.

	7	6	5	4	3	2	1	0
USFA0E0W	E0W7	E0W6	E0W5	E0W4	E0W3	E0W2	E0W1	E0W0
	W	W	W	W	W	W	W	W

Table 30-59 USFA0E0W register contents

Bit position	Bit name	Function
7:0	E0W[7:0]	These bits store the IN data sent to the host during the data stage to Endpoint0.

The operation of the USFA0E0W register is shown below.

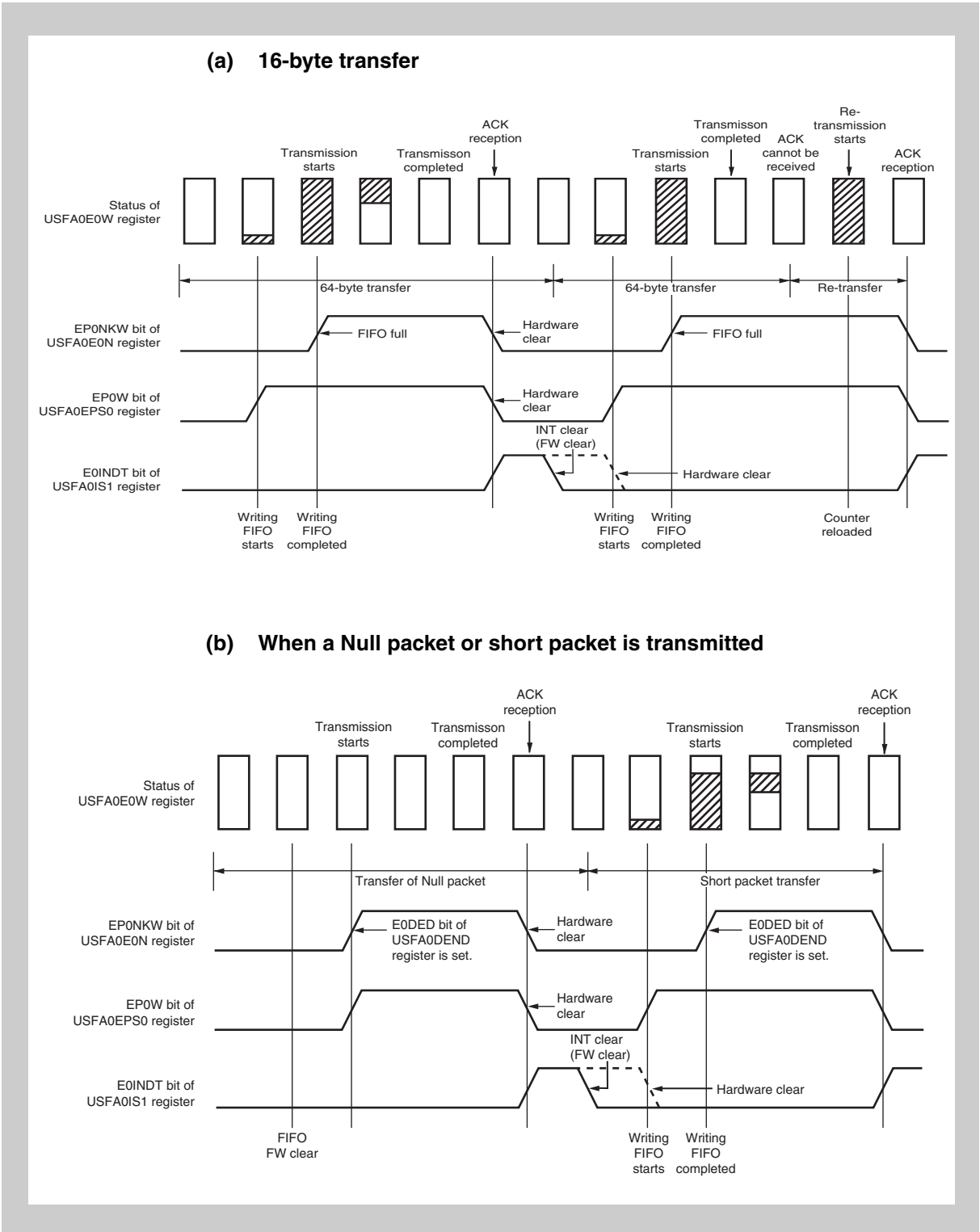


Figure 30-6 USFA0E0W register operation

(5) Bulk-out 1 register (USFA0BO1)

The USFA0BO1 register is a pair of 64-byte FIFOs that store data for Endpoint2. The USFA0BO1 register consists of two 64-byte FIFOs in a bank configuration, each of which performs a toggle operation and repeatedly connects the buses on the SIE and CPU sides. The toggle operation takes place when there is data in the FIFO on the SIE side but not in the FIFO on the CPU side (counter value = 0).

When the hardware receives data for Endpoint2 from the host, it automatically transfers the data to the USFA0BO1 register. When the register correctly receives the data, a FIFO toggle operation occurs. As a result, the BKO1DT bit of the USFA0IS3 register is set to 1, the amount of received data is held by the USFA0BO1L register, and an interrupt request or DMA request is issued to the CPU. Whether the interrupt request or DMA request is issued can be selected by using the DQBO1MS bit of the USFA0IDR register.

Use FW to read the data held by the USFA0BO1 register, up to the amount of data read by the USFA0BO1L register. When the correct received data is held by the FIFO connected to the SIE side and the value of the USFA0BO1L register reaches 0, the FIFO toggle operation occurs, and the BKO1NK bit of the USFA0EN register is automatically cleared to 0. If data greater than the value of the USFA0BO1L register is read and the FIFO toggle condition is satisfied, the FIFO toggle operation occurs. As a result, the next packet might be read by mistake. Note that, if the toggle condition is not satisfied, the first data is repeatedly read.

If overrun data is received while data is held by the FIFO connected to the CPU side, Endpoint2 stalls, and the FIFO on the CPU side is cleared.

If the USFA0BO1 register is read while no data is in it, an undefined value is read.

Access This register is read-only, in 8-bit units. Writing to this register is ignored.

Address 210_H

Initial value Undefined.

Caution Be sure to read all the stored data.

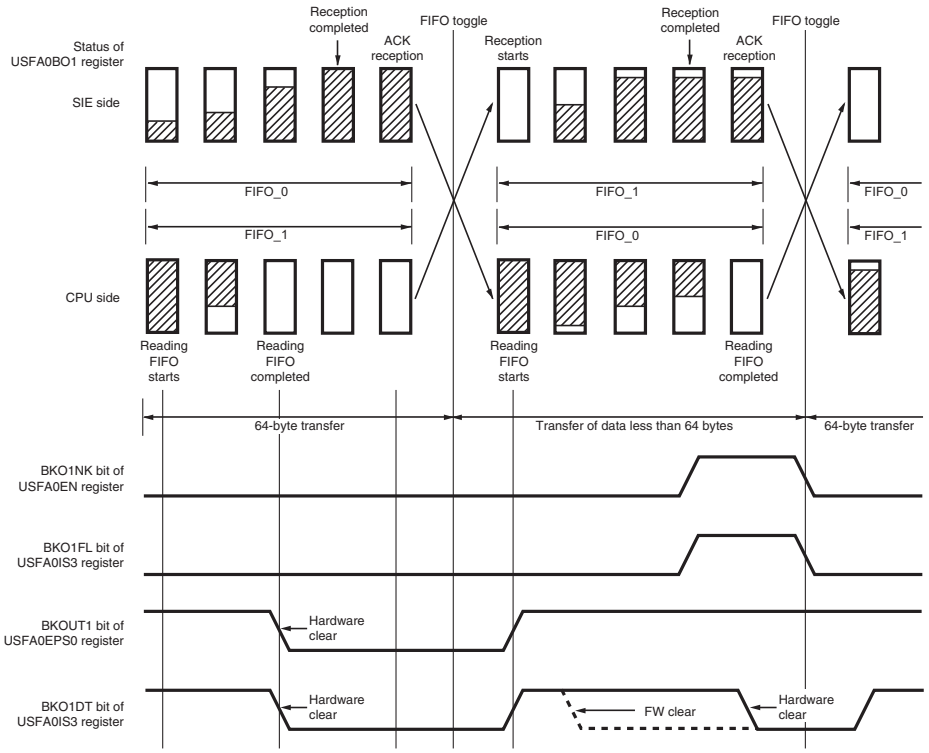
	7	6	5	4	3	2	1	0
USFA0BO1	BKO17	BKO16	BKO15	BKO14	BKO13	BKO12	BKO11	BKO10
	R	R	R	R	R	R	R	R

Table 30-60 USFA0BO1 register contents

Bit position	Bit name	Function
7:0	BKO[17:10]	These bits store data for Endpoint2.

The operation of the USFA0BO1 register is shown below.

(a) Operation example 1



(b) Operation example 2

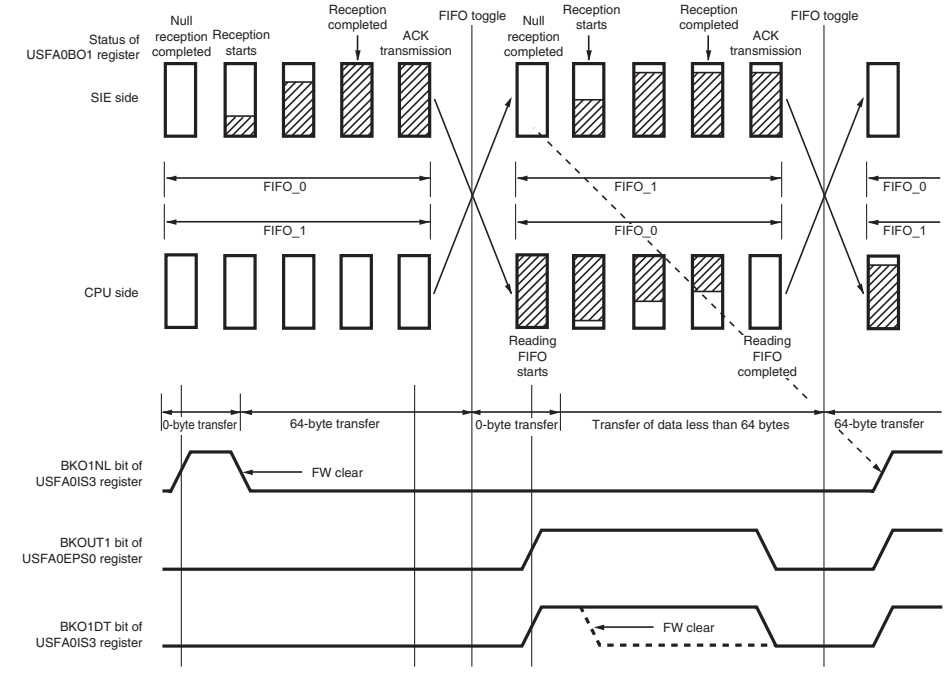


Figure 30-7 USFA0BO1 register operation

(6) Bulk-out 1 length register (USFA0BO1L)

The USFA0BO1L register stores the length of the data held by the USFA0BO1 register.

The USFA0BO1L register always updates the received data length while it is receiving data. If the final transfer results in abnormal reception, the USFA0BO1L register is cleared to 00_H, and an interrupt request is not generated. If the reception is normal, the interrupt request is generated, and FW can read as much data from the USFA0BO1 register as the value read from the USFA0BO1L register. The value of the USFA0BO1L register is decremented each time the USFA0BO1 register has been read.

Access This register is read-only, in 8-bit units. Writing to this register is ignored.

Address 214_H

Initial value 00_H. This register is initialized by any reset.

	7	6	5	4	3	2	1	0
USFA0BO1L	BKO1L7	BKO1L6	BKO1L5	BKO1L4	BKO1L3	BKO1L2	BKO1L1	BKO1L0
	R	R	R	R	R	R	R	R

Table 30-61 USFA0BO1L register contents

Bit position	Bit name	Function
7:0	BKO1L[7:0]	These bits store the length of the data held by the USFA0BO1 register.

(7) Bulk-out 2 register (USFA0BO2)

The USFA0BO2 register is a pair of 64-bit FIFOs that store data for Endpoint4. The USFA0BO2 register consists of two 64-byte FIFOs in a bank configuration, each of which performs a toggle operation and repeatedly connects the buses on the SIE and CPU sides. The toggle operation takes place when there is data in the FIFO on the SIE side but not in the FIFO on the CPU side (counter value = 0).

When the hardware receives data for Endpoint4 from the host, it automatically transfers the data to the USFA0BO2 register. When the register correctly receives the data, a FIFO toggle operation occurs. As a result, the BKO2DT bit of the USFA0IS3 register is set to 1, the amount of received data is held by the USFA0BO2L register, and an interrupt request or DMA request is issued to the CPU. Whether the interrupt request or DMA request is issued can be selected by using the DQBO2MS bit of the USFA0IDR register.

Use FW to read the data held by the USFA0BO2 register, up to the amount of data read by the USFA0BO2L register. When the correct received data is held by the FIFO connected to the SIE side and the value of the USFA0BO2L register reaches 0, the FIFO toggle operation occurs, and the BKO2NK bit of the USFA0EN register is automatically cleared to 0. If data greater than the value of the USFA0BO2L register is read and the FIFO toggle condition is satisfied, the FIFO toggle operation occurs. As a result, the next packet might be read by mistake. Note that, if the toggle condition is not satisfied, the first data is repeatedly read.

If overrun data is received while data is held by the FIFO connected to the CPU side, Endpoint4 stalls, and the FIFO on the CPU side is cleared.

If the USFA0BO2 register is read while no data is in it, an undefined value is read.

Access This register is read-only, in 8-bit units. Writing to this register is ignored.

Address 218_H

Initial value Undefined.

Caution Be sure to read all the stored data.

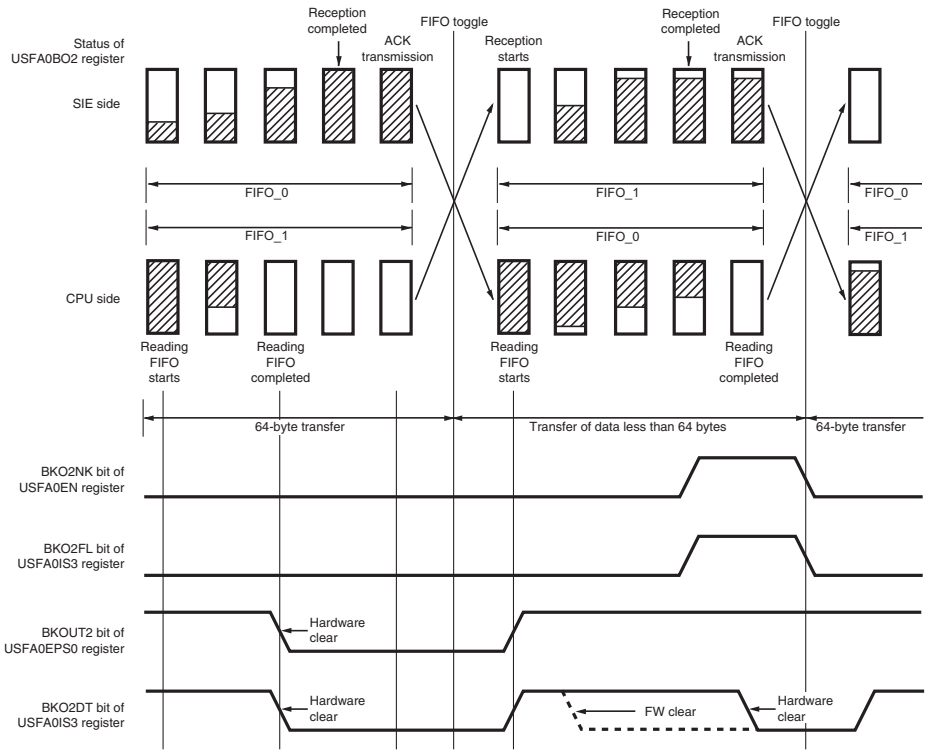
	7	6	5	4	3	2	1	0
USFA0BO2	BKO27	BKO26	BKO25	BKO24	BKO23	BKO22	BKO21	BKO20
	R	R	R	R	R	R	R	R

Table 30-62 USFA0BO2 register contents

Bit position	Bit name	Function
7:0	BKO[27:20]	These bits store data for Endpoint4.

The operation of the USFA0BO2 register is shown below.

(a) Operation example 1



(b) Operation example 2

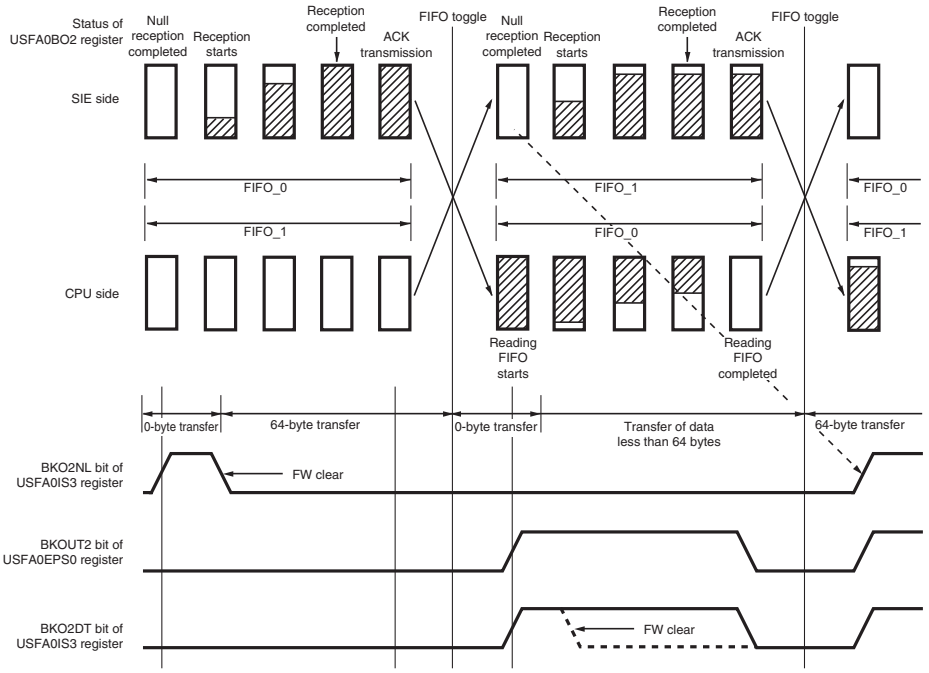


Figure 30-8 USFA0BO2 register operation

(8) Bulk-out 2 length register (USFA0BO2L)

The USFA0BO2L register stores the length of the data held by the USFA0BO2 register.

The USFA0BO2L register always updates the received data length while it is receiving data. If the final transfer results in abnormal reception, the USFA0BO2L register is cleared to 00_H, and an interrupt request is not generated. If the reception is normal, the interrupt request is generated, and FW can read as much data from the USFA0BO2 register as the value read from the USFA0BO2L register. The value of the USFA0BO2L register is decremented each time the USFA0BO2 register has been read.

Access This register is read-only, in 8-bit units. Writing to this register is ignored.

Address $21C_H$

Initial value 00_H. This register is initialized by any reset.

	7	6	5	4	3	2	1	0
USFA0BO2L	BKO2L7	BKO2L6	BKO2L5	BKO2L4	BKO2L3	BKO2L2	BKO2L1	BKO2L0
	R	R	R	R	R	R	R	R

Table 30-63 USFA0BO2L register contents

Bit position	Bit name	Function
7:0	BKO2L[7:0]	These bits store the length of the data held by the USFA0BO2 register.

(9) Bulk-in 1 register (USFA0BI1)

The USFA0BI1 register is a pair of 64-byte FIFOs that store data for Endpoint1. The USFA0BI1 register consists of two 64-byte FIFOs in a bank configuration, each of which performs a toggle operation and repeatedly connects the buses on the SIE and CPU sides. The toggle operation takes place when there is no data in the FIFO on the SIE side (counter value = 0) and when the FIFO on the CPU side is correctly written (FIFO full or BKI1DED bit = 1).

The hardware transmits data to the USB bus in synchronization with the IN token for Endpoint1 only when the BKI1NK bit of the USFA0EN register is set to 1 (when NAK is not transmitted). The address at which data is to be written or read is managed by the hardware. Therefore, FW can transmit data to the host only by writing the data to the USFA0BI1 register sequentially. A short packet is transmitted when data is written to the USFA0BI1 register and the BKI1DED bit of the USFA0DEND register is set to 1 (BKIN1 bit of USFA0EPS0 register = 1 (data exists)). A Null packet is transmitted when the USFA0BI1 register is cleared and the BKI1DED bit of the USFA0DEND register is set to 1 (BKIN1 bit of the USFA0EPS0 register = 1 (data exists)). When the data is transmitted correctly, a FIFO toggle operation occurs, the BKI1DT bit of the USFA0IS2 register is set to 1, and an interrupt request is generated for the CPU. An interrupt request or DMA request can be selected by using the DQBI1MS bit of the USFA0IDR register.

Access This register is write-only, in 8-bit units. If this register is read, 00_H is read.

Address 220_H

Initial value Undefined.

	7	6	5	4	3	2	1	0
USFA0BI1	BKI17	BKI16	BKI15	BKI14	BKI13	BKI12	BKI11	BKI10
	W	W	W	W	W	W	W	W

Table 30-64 USFA0BI1 register contents

Bit position	Bit name	Function
7:0	BKI[17:10]	These bits store data for Endpoint1.

The operation of the USFA0BI1 register is shown below.

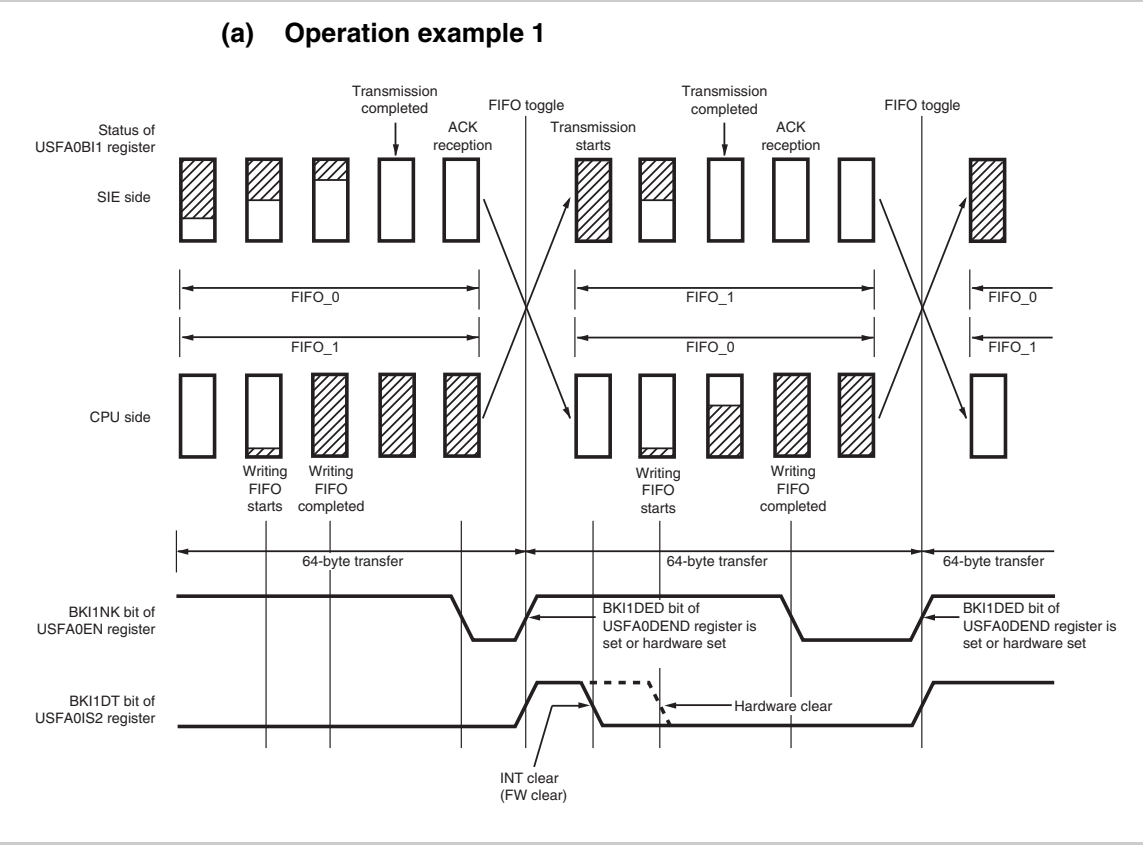


Figure 30-9 USFA0BI1 register operation - 1

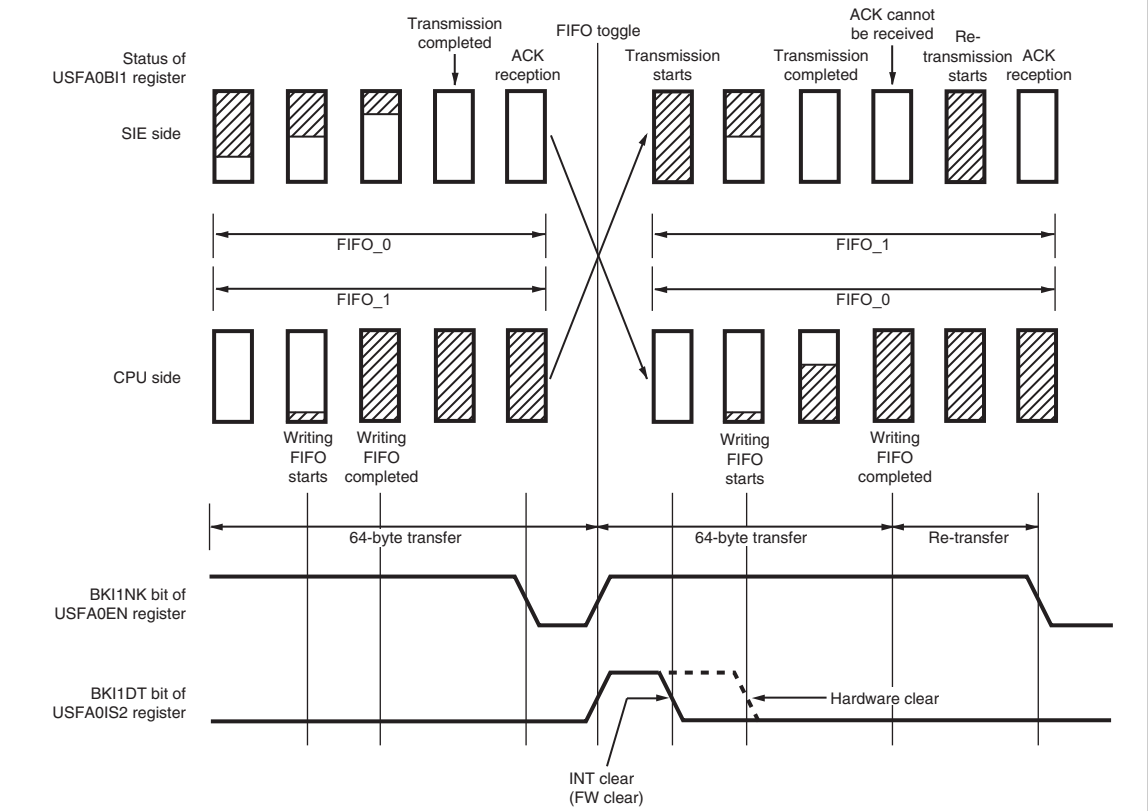


Figure 30-10 USFA0BI1 register operation - 2

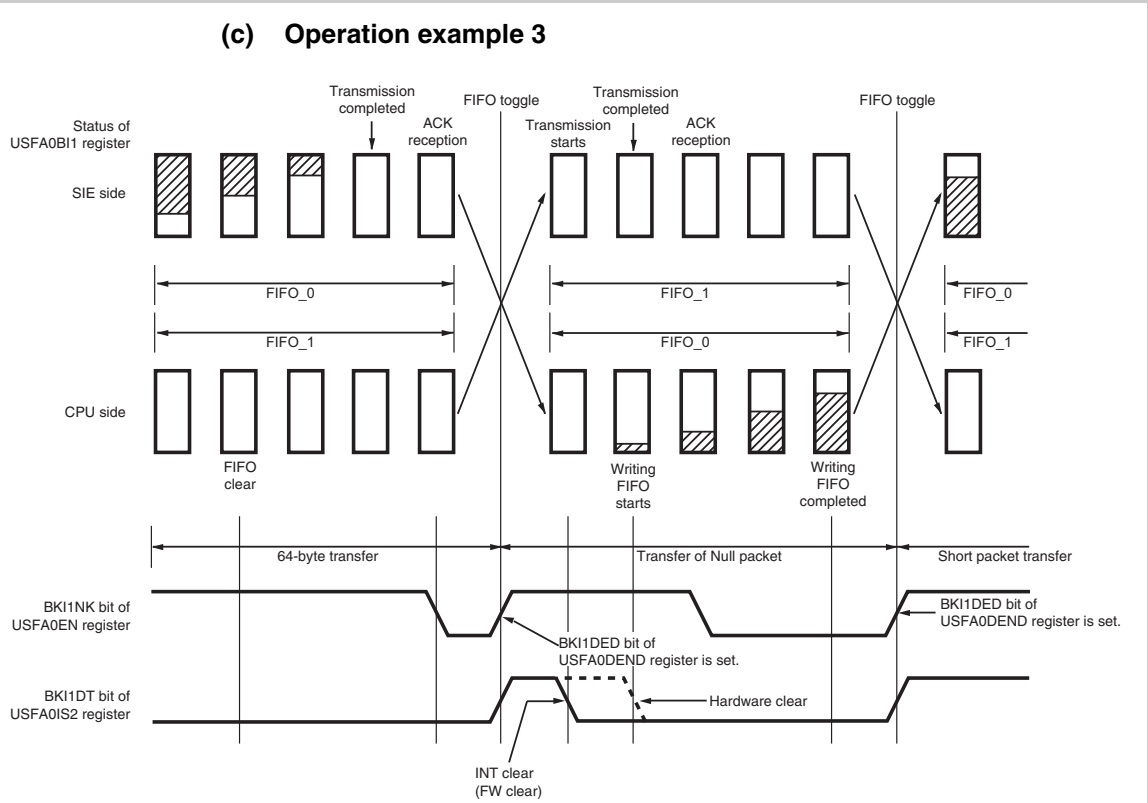


Figure 30-11 USFA0BI1 register operation - 3

(10) Bulk-in 2 register (USFA0BI2)

The USFA0BI2 register is a pair of 64-byte FIFOs that store data for Endpoint3. The USFA0BI2 register consists of two 64-byte FIFOs in a bank configuration, each of which performs a toggle operation and repeatedly connects the buses on the SIE and CPU sides. The toggle operation takes place when there is no data in the FIFO on the SIE side (counter value = 0) and when the FIFO on the CPU side is correctly written (FIFO full or BKI2DED bit = 1).

The hardware transmits data to the USB bus in synchronization with the IN token for Endpoint3 only when the BKI2NK bit of the USFA0EN register is set to 1 (when NAK is not transmitted). The address at which data is to be written or read is managed by the hardware. Therefore, FW can transmit data to the host only by writing the data to the USFA0BI2 register sequentially. A short packet is transmitted when data is written to the USFA0BI2 register and the BKI1DED bit of the USFA0DEND register is set to 1 (BKIN2 bit of USFA0EPS0 register = 1 (data exists)). A Null packet is transmitted when the USFA0BI2 register is cleared and the BKI2DED bit of the USFA0DEND register is set to 1 (BKIN2 bit of the USFA0EPS0 register = 1 (data exists)). When the data is transmitted correctly, a FIFO toggle operation occurs, the BKI2DT bit of the USFA0IS2 register is set to 1, and an interrupt request is generated for the CPU. An interrupt request or DMA request can be selected by using the DQBI2MS bit of the USFA0IDR register.

Access This register is write-only, in 8-bit units. If this register is read, 00_H is read.

Address 224_H

Initial value Undefined.

	7	6	5	4	3	2	1	0
USFA0BI2	BKI27	BKI26	BKI25	BKI24	BKI23	BKI22	BKI21	BKI20
	W	W	W	W	W	W	W	W

Table 30-65 USFA0BI2 register contents

Bit position	Bit name	Function
7:0	BKI[27:20]	These bits store data for Endpoint3.

The operation of the USFA0BI2 register is shown below.

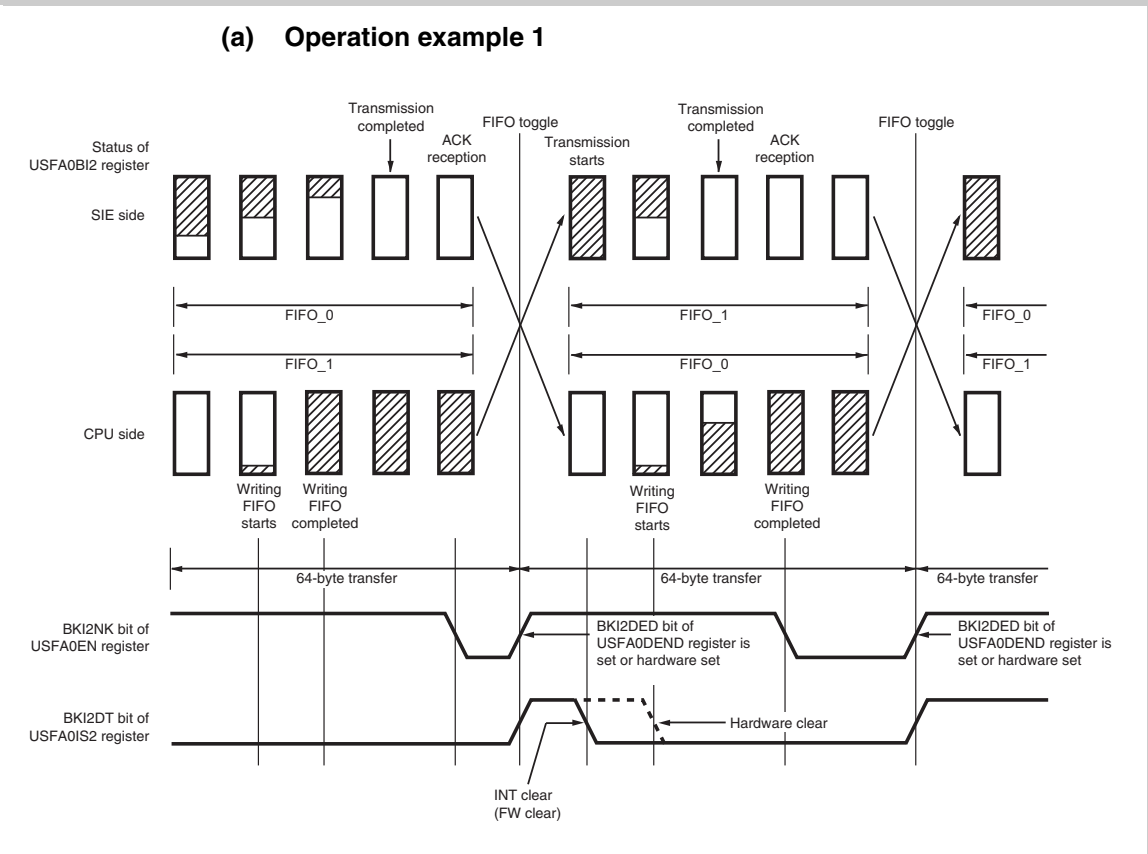


Figure 30-12 USFA0BI2 register operation - 1

(b) Operation example 2

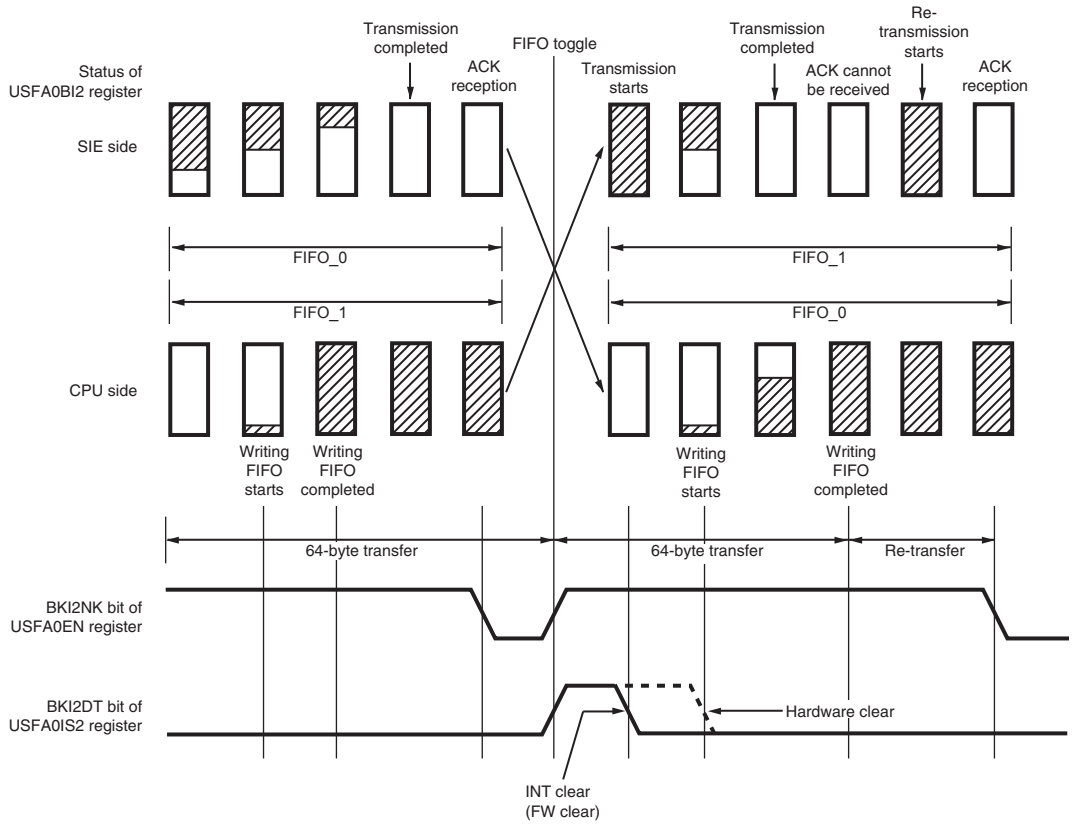


Figure 30-13 USFA0BI2 register operation - 2

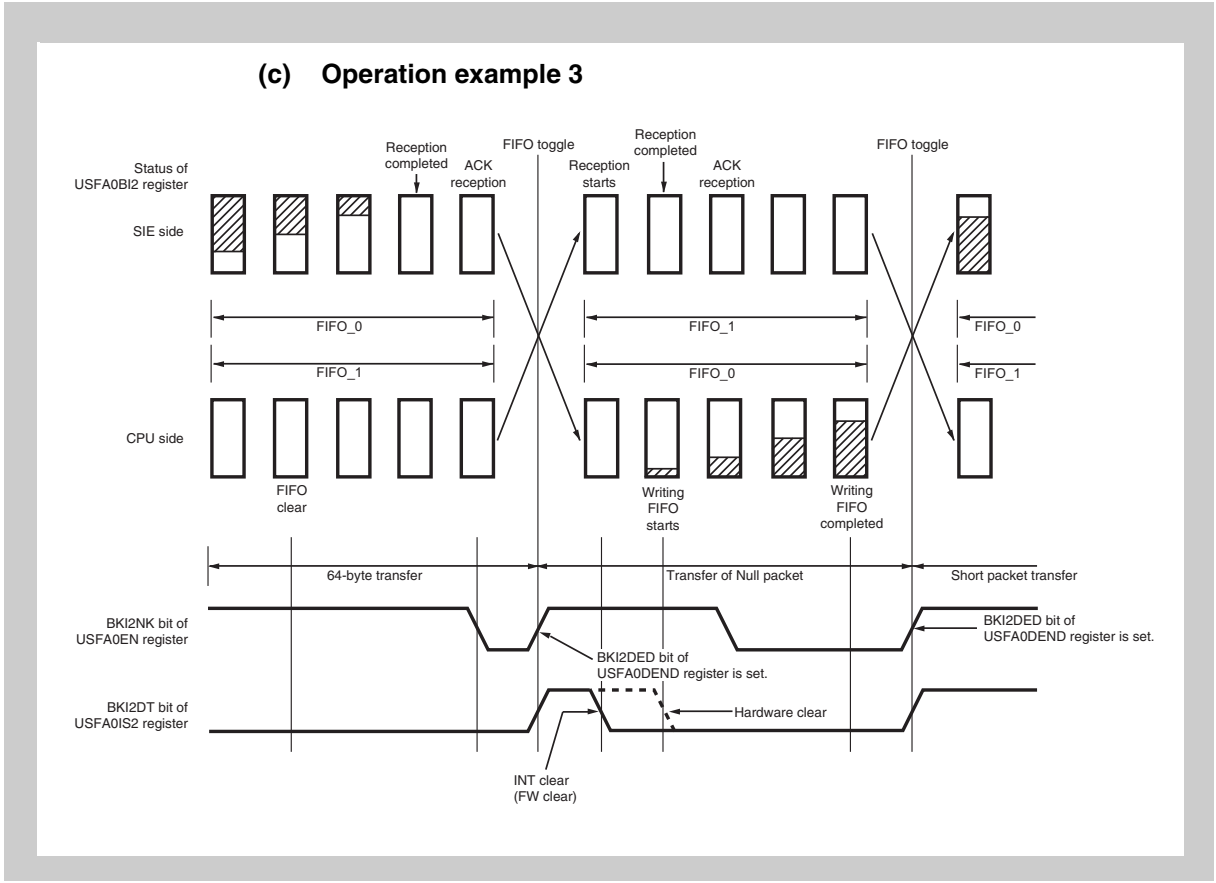


Figure 30-14 USFA0BI2 register operation - 3

(11) Interrupt 1 register (USFA0INT1)

The USFA0INT1 register is a 64-byte FIFO that stores data for Endpoint7 (to be passed to SIE).

The hardware transmits data to the USB bus in synchronization with the IN token for Endpoint7 only when the IT1NK bit of the USFA0EN register is set to 1 (when NAK is not transmitted). When the data is transmitted and the host correctly receives it, the IT1NK bit of the USFA0EN register is automatically cleared to 0 by hardware. A short packet is transmitted when data is written to the USFA0INT1 register and the IT1DEND bit of the USFA0DEND register is set to 1 (IT1 bit of the USFA0EPS0 register = 1 (data exists)). A Null packet is transmitted when the USFA0INT1 register is cleared and the IT1DEND bit of the USFA0DEND register is set to 1 (IT1 bit of the USFA0EPS0 register = 1 (data exists)).

Access This register is write-only, in 8-bit units. If this register is read, 00_H is read.

Address 228_H

Initial value Undefined.

	7	6	5	4	3	2	1	0
USFA0INT1	IT17	IT16	IT15	IT14	IT13	IT12	IT11	IT10
	W	W	W	W	W	W	W	W

Table 30-66 USFA0INT1 register contents

Bit position	Bit name	Function
7:0	IT[17:10]	These bits store data for Endpoint7.

The operation of the USFA0INT1 register is shown below.

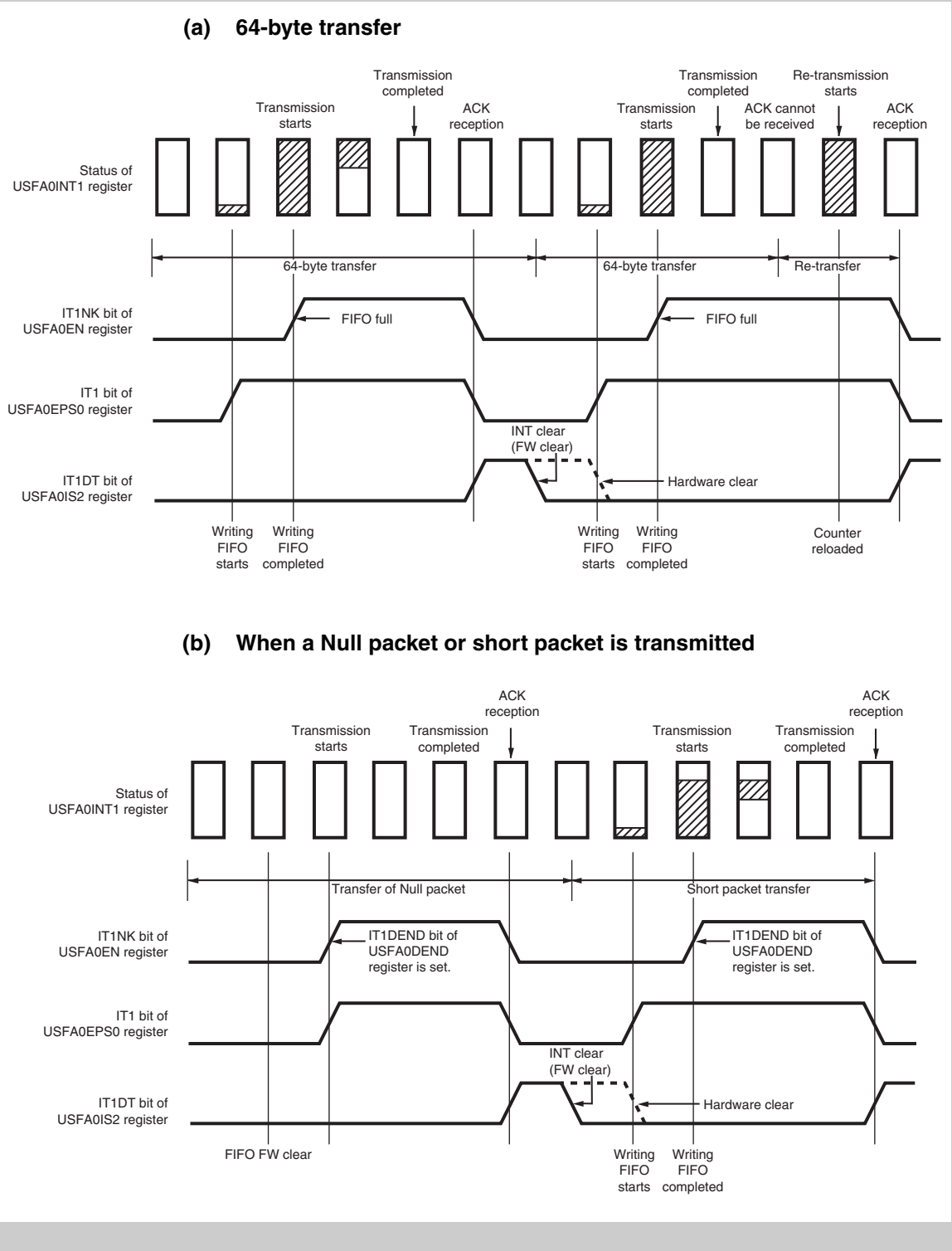


Figure 30-15 USFA0INT1 register operation

(12) Interrupt 2 register (USFA0INT2)

The USFA0INT2 register is a 64-byte FIFO that stores data for Endpoint8 (to be passed to SIE).

The hardware transmits data to the USB bus in synchronization with the IN token for Endpoint8 only when the IT2NK bit of the USFA0EN register is set to 1 (when NAK is not transmitted). When the data is transmitted and the host correctly receives it, the IT2NK bit of the USFA0EN register is automatically cleared to 0 by hardware. A short packet is transmitted when data is written to the USFA0INT2 register and the IT2DEND bit of the USFA0DEND register is set to 1 (IT2 bit of the USFA0EPS0 register = 1 (data exists)). A Null packet is transmitted when the USFA0INT2 register is cleared and the IT2DEND bit of the USFA0DEND register is set to 1 (IT2 bit of the USFA0EPS0 register = 1 (data exists)).

Access This register is write-only, in 8-bit units. If this register is read, 00_H is read.

Address $22C_H$

Initial value Undefined.

	7	6	5	4	3	2	1	0
USFA0INT2	IT27	IT26	IT25	IT24	IT23	IT22	IT21	IT20
	W	W	W	W	W	W	W	W

Table 30-67 USFA0INT2 register contents

Bit position	Bit name	Function
7:0	IT[27:20]	These bits store data for Endpoint8.

The operation of the USFA0INT2 register is shown below.

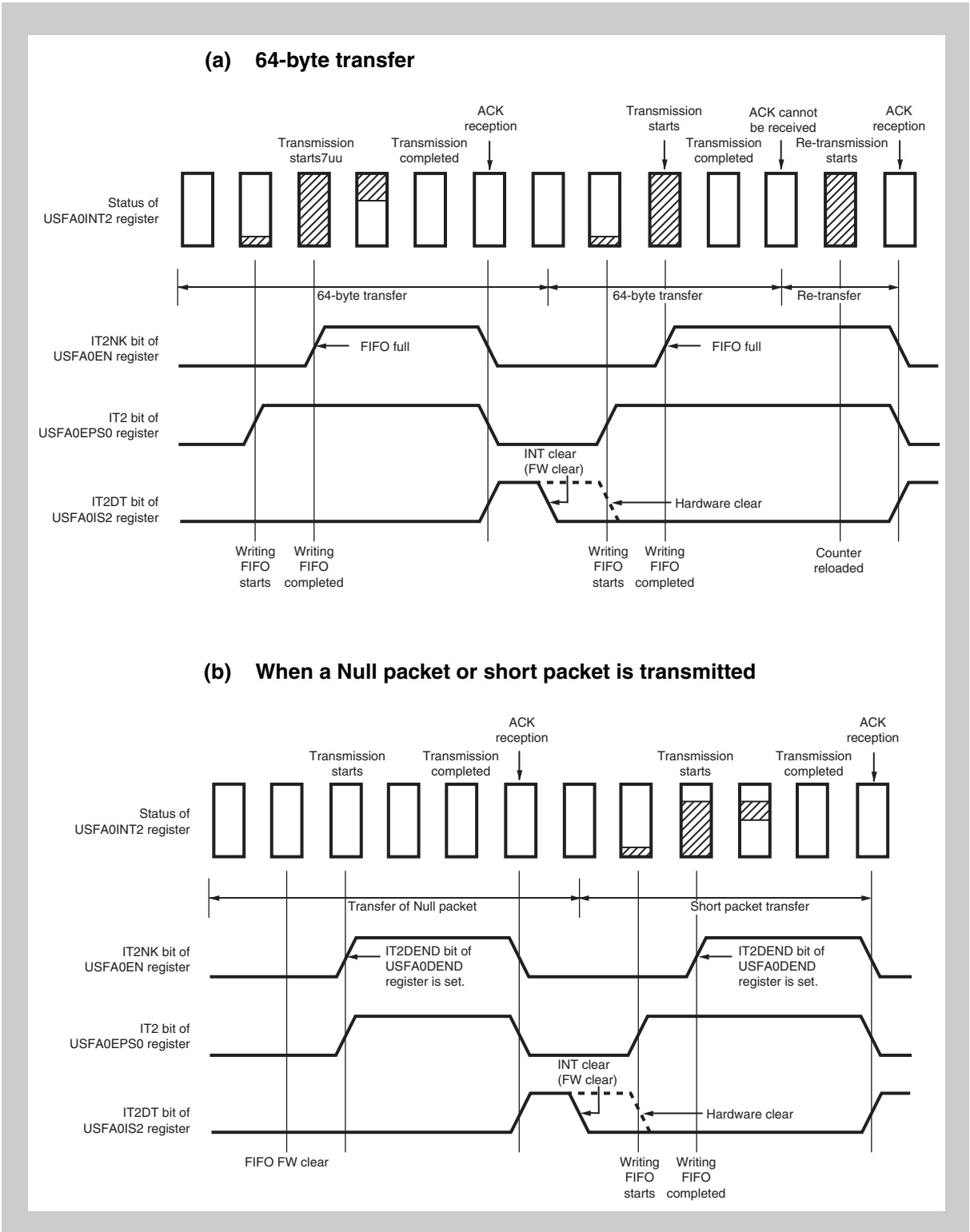


Figure 30-16 USFA0INT2 register operation

30.7.4 EPC request data registers

(1) Device status register (USFA0DST)

This register stores the value that is to be returned in response to the GET_STATUS Device request.

The hardware automatically transmits the contents of this register to the host when it has received the GET_STATUS Device request.

Access These registers can be read or written in 8-bit units.

Address 288_H

Initial value 00_H. This register is initialized by any reset.

Caution To rewrite this register, set the EP0NKA bit to 1 before reading the register contents, and then rewrite the register contents after confirming that the bit has been set, in order to prevent a conflict between reading and writing.

	7	6	5	4	3	2	1	0
USFA0DST	0	0	0	0	0	0	RMWK	SFPW
	R	R	R	R	R	R	R/W	R/W

Table 30-68 USFA0DST register contents

Bit position	Bit name	Function
1	RMWK	This bit specifies whether the remote wakeup function of the device is used. 1: Enabled 0: Disabled If the device supports the remote wakeup function, this bit is set to 1 by hardware when the SET_FEATURE Device request has been received, and is cleared to 0 by hardware when the CLEAR_FEATURE Device request has been received. If the device does not support the remote wakeup function, make sure that the SET_FEATURE Device request is not issued from the host.
0	SFPW	This bit indicates whether the device is self-powered or bus-powered. 1: Self-powered 0: Bus-powered

(2) EP0 status register (USFA0E0S)

This register stores the value that is to be returned in response to the GET_STATUS Endpoint0 request.

If an error occurs in USBF, the E0HALT bit is set to 1 by FW. Writing to this register is ignored while a USB-side access to Endpoint0 is being received.

When the E0HALT bit is set to 1 by FW, it is not applied until the next SETUP token is received if the control transfer immediately before is for SET_FEATURE Endpoint0, CLEAR_FEATURE Endpoint0, a GET_STATUS Endpoint0 request, or an FW-processed request.

The hardware automatically transmits the contents of this register to the host when it has received the GET_STATUS Endpoint0 request. If Endpoint0 has stalled, the USFA0E0W and USFA0E0R registers are cleared, and the EP0NKW and EP0NKR bits of the USFA0E0N register are cleared to 0.

Access This register can be read or written in 8-bit units. However, data can be written to this register only when the EP0NKA bit is set to 1.

Address 298_H

Initial value 00_H. This register is initialized by any reset.

Caution To rewrite this register, set the EP0NKA bit to 1 before reading the register contents, and then rewrite the register contents after confirming that the bit has been set, in order to prevent a conflict between reading and writing.

	7	6	5	4	3	2	1	0
USFA0E0S	0	0	0	0	0	0	0	E0HALT
	R	R	R	R	R	R	R	R/W

Table 30-69 USFA0E0S register contents

Bit position	Bit name	Function
0	E0HALT	This bit indicates the status of Endpoint0. 1: Endpoint0 has stalled. 0: Endpoint0 has not stalled. This bit is set to 1 by hardware when the SET_FEATURE Endpoint0 request has been received, and cleared to 0 by hardware when the CLEAR_FEATURE Endpoint0 request has been received. DATA PID is initialized to DATA0.

(3) EP1 status register (USFA0E1S)

This register stores the value that is to be returned in response to the GET_STATUS Endpoint1 request.

If an error occurs in Endpoint1, the E1HALT bit is set to 1. Writing to this register is ignored while a USB-side access to Endpoint1 is being received.

The hardware automatically transmits the contents of this register to the host when it has received the GET_STATUS Endpoint1 request. If Endpoint1 has stalled, the USFA0BI1 register is cleared and the BKI1NK bit is cleared to 0.

Because writing to this register is always masked when a transfer to Endpoint1, rather than a control transfer, is executed, be sure to check whether data has been correctly written to this register.

Access This register can be read or written in 8-bit units. However, data can be written to this register only when the EP0NKA bit is set to 1.

Address $2A0_H$

Initial value 00_H. This register is initialized by any reset.

Caution To rewrite this register, set the EP0NKA bit to 1 before reading the register contents, and then rewrite the register contents after confirming that the bit has been set, in order to prevent a conflict between reading and writing.

	7	6	5	4	3	2	1	0
USFA0E1S	0	0	0	0	0	0	0	E1HALT
	R	R	R	R	R	R	R	R/W

Table 30-70 USFA0E1S register contents

Bit position	Bit name	Function
0	E1HALT	This bit indicates the status of Endpoint1. 1: Endpoint1 has stalled. 0: Endpoint1 has not stalled. This bit is set to 1 by hardware when the SET_FEATURE Endpoint1 request has been received. It is cleared to 0 by hardware when the CLEAR_FEATURE Endpoint1 request, SET_CONFIGURATION request, or SET_INTERFACE request for the interface to which Endpoint1 is linked has correctly been received. DATA PID is initialized to DATA0.

(4) EP2 status register (USFA0E2S)

This register stores the value that is to be returned in response to the GET_STATUS Endpoint2 request.

If an error occurs in Endpoint2, the E2HALT bit is set to 1. Writing to this register is ignored while a USB-side access to Endpoint2 is being received.

The hardware automatically transmits the contents of this register to the host when it has received the GET_STATUS Endpoint2 request. If Endpoint2 has stalled, the USFA0BO1 register is cleared and the BKO1NK bit is cleared to 0.

Because writing to this register is always masked when a transfer to Endpoint2, rather than a control transfer, is executed, be sure to check whether data has been correctly written to this register.

Access This register can be read or written in 8-bit units. However, data can be written to this register only when the EP0NKA bit is set to 1.

Address $2A8_H$

Initial value 00_H. This register is initialized by any reset.

Caution To rewrite this register, set the EP0NKA bit to 1 before reading the register contents, and then rewrite the register contents after confirming that the bit has been set, in order to prevent a conflict between reading and writing.

	7	6	5	4	3	2	1	0
USFA0E2S	0	0	0	0	0	0	0	E2HALT
	R	R	R	R	R	R	R	R/W

Table 30-71 USFA0E2S register contents

Bit position	Bit name	Function
0	E2HALT	This bit indicates the status of Endpoint2. 1: Endpoint2 has stalled. 0: Endpoint2 has not stalled. This bit is set to 1 by hardware when the SET_FEATURE Endpoint2 request has been received. It is cleared to 0 by hardware when the CLEAR_FEATURE Endpoint2 request, SET_CONFIGURATION request, or SET_INTERFACE request for the interface to which Endpoint2 is linked has correctly been received. DATA PID is initialized to DATA0.

(5) EP3 status register (USFA0E3S)

This register stores the value that is to be returned in response to the GET_STATUS Endpoint3 request.

If an error occurs in Endpoint3, the E3HALT bit is set to 1. Writing to this register is ignored while a USB-side access to Endpoint3 is being received.

The hardware automatically transmits the contents of this register to the host when it has received the GET_STATUS Endpoint3 request. If Endpoint3 has stalled, the USFA0BI2 register is cleared and the BKI2NK bit is cleared to 0.

Because writing to this register is always masked when a transfer to Endpoint3, rather than a control transfer, is executed, be sure to check whether data has been correctly written to this register.

Access This register can be read or written in 8-bit units. However, data can be written to this register only when the EP0NKA bit is set to 1.

Address $2B0_H$

Initial value 00_H. This register is initialized by any reset.

Caution To rewrite this register, set the EP0NKA bit to 1 before reading the register contents, and then rewrite the register contents after confirming that the bit has been set, in order to prevent a conflict between reading and writing.

	7	6	5	4	3	2	1	0
USFA0E3S	0	0	0	0	0	0	0	E3HALT
	R	R	R	R	R	R	R	R/W

Table 30-72 USFA0E3S register contents

Bit position	Bit name	Function
0	E3HALT	This bit indicates the status of Endpoint3. 1: Endpoint3 has stalled. 0: Endpoint3 has not stalled. This bit is set to 1 by hardware when the SET_FEATURE Endpoint3 request has been received. It is cleared to 0 by hardware when the CLEAR_FEATURE Endpoint3 request, SET_CONFIGURATION request, or SET_INTERFACE request for the interface to which Endpoint3 is linked has correctly been received. DATA PID is initialized to DATA0.

(6) EP4 status register (USFA0E4S)

This register stores the value that is to be returned in response to the GET_STATUS Endpoint4 request.

If an error occurs in Endpoint4, the E4HALT bit is set to 1. Writing to this register is ignored while a USB-side access to Endpoint4 is being received.

The hardware automatically transmits the contents of this register to the host when it has received the GET_STATUS Endpoint4 request. If Endpoint4 has stalled, the USFA0BO2 register is cleared and the BKO2NK bit is cleared to 0.

Because writing to this register is always masked when a transfer to Endpoint4, rather than a control transfer, is executed, be sure to check whether data has been correctly written to this register.

Access This register can be read or written in 8-bit units. However, data can be written to this register only when the EP0NKA bit is set to 1.

Address $2B8_H$

Initial value 00_H. This register is initialized by any reset.

Caution To rewrite this register, set the EP0NKA bit to 1 before reading the register contents, and then rewrite the register contents after confirming that the bit has been set, in order to prevent a conflict between reading and writing.

	7	6	5	4	3	2	1	0
USFA0E4S	0	0	0	0	0	0	0	E4HALT
	R	R	R	R	R	R	R	R/W

Table 30-73 USFA0E4S register contents

Bit position	Bit name	Function
0	E4HALT	This bit indicates the status of Endpoint4. 1: Endpoint4 has stalled. 0: Endpoint4 has not stalled. This bit is set to 1 by hardware when the SET_FEATURE Endpoint4 request has been received. It is cleared to 0 by hardware when the CLEAR_FEATURE Endpoint4 request, SET_CONFIGURATION request, or SET_INTERFACE request for the interface to which Endpoint4 is linked has correctly been received. DATA PID is initialized to DATA0.

(7) EP7 status register (USFA0E7S)

This register stores the value that is to be returned in response to the GET_STATUS Endpoint7 request.

If an error occurs in Endpoint7, the E7HALT bit is set to 1. Writing to this register is ignored while a USB-side access to Endpoint7 is being received.

The hardware automatically transmits the contents of this register to the host when it has received the GET_STATUS Endpoint7 request. If Endpoint7 has stalled, the USFA0INT1 register is cleared and the IT1NK bit is cleared to 0.

Because writing to this register is always masked when a transfer to Endpoint7, rather than a control transfer, is executed, be sure to check whether data has been correctly written to this register.

Access This register can be read or written in 8-bit units. However, data can be written to this register only when the EP0NKA bit is set to 1.

Address $2D0_H$

Initial value 00_H. This register is initialized by any reset.

Caution To rewrite this register, set the EP0NKA bit to 1 before reading the register contents, and then rewrite the register contents after confirming that the bit has been set, in order to prevent a conflict between reading and writing.

	7	6	5	4	3	2	1	0
USFA0E7S	0	0	0	0	0	0	0	E7HALT
	R	R	R	R	R	R	R	R/W

Table 30-74 USFA0E7S register contents

Bit position	Bit name	Function
0	E7HALT	This bit indicates the status of Endpoint7. 1: Endpoint7 has stalled. 0: Endpoin7 has not stalled. This bit is set to 1 by hardware when the SET_FEATURE Endpoint7 request has been received. It is cleared to 0 by hardware when the CLEAR_FEATURE Endpoint7 request, SET_CONFIGURATION request, or SET_INTERFACE request for the interface to which Endpoint7 is linked has correctly been received. DATA PID is initialized to DATA0.

(8) EP8 status register (USFA0E8S)

This register stores the value that is to be returned in response to the GET_STATUS Endpoint8 request.

If an error occurs in Endpoint8, the E8HALT bit is set to 1. Writing to this register is ignored while a USB-side access to Endpoint8 is being received.

The hardware automatically transmits the contents of this register to the host when it has received the GET_STATUS Endpoint8 request. If Endpoint8 has stalled, the USFA0INT2 register is cleared and the IT2NK bit is cleared to 0.

Because writing to this register is always masked when a transfer to Endpoint8, rather than a control transfer, is executed, be sure to check whether data has been correctly written to this register.

Access This register can be read or written in 8-bit units. However, data can be written to this register only when the EP0NKA bit is set to 1.

Address $2D8_H$

Initial value 00_H. This register is initialized by any reset.

Caution To rewrite this register, set the EP0NKA bit to 1 before reading the register contents, and then rewrite the register contents after confirming that the bit has been set, in order to prevent a conflict between reading and writing.

	7	6	5	4	3	2	1	0
USFA0E8S	0	0	0	0	0	0	0	E8HALT
	R	R	R	R	R	R	R	R/W

Table 30-75 USFA0E8S register contents

Bit position	Bit name	Function
0	E8HALT	This bit indicates the status of Endpoint8. 1: Endpoint8 has stalled. 0: Endpoint8 has not stalled. This bit is set to 1 by hardware when the SET_FEATURE Endpoint8 request has been received. It is cleared to 0 by hardware when the CLEAR_FEATURE Endpoint8 request, SET_CONFIGURATION request, or SET_INTERFACE request for the interface to which Endpoint8 is linked has correctly been received. DATA PID is initialized to DATA0.

(9) Address register (USFA0ADRS)

This register stores the device address.

The device address sent by the SET_ADDRESS request is analyzed and the resulting value is automatically written to this register. If the SET_ADDRESS request is processed by FW, the value of this register is applied as the device address when the SUCCESS signal is received in the status stage.

Access This register is read-only, in 8-bit units.

Address 300_H

Initial value 00_H. This register is initialized by any reset.

Caution Do not write to this register. If the register is written to, the operation is not guaranteed.

	7	6	5	4	3	2	1	0
USFA0ADRS	0	ADRS6	ADRS5	ADRS4	ADRS3	ADRS2	ADRS1	ADRS0
	R	R	R	R	R	R	R	R

Table 30-76 USFA0ADRS register contents

Bit position	Bit name	Function
6:0	ADRS[6:0]	These bits hold the SIE device address.

(10) Configuration register (USFA0CNF)

This register stores the value that is to be returned in response to the GET_CONFIGURATION request.

When the SET_CONFIGURATION request is received, its wValue is automatically written to this register.

When a change of the value of this register from 00_H to other than 00_H is detected, the CONF bit of the USFA0MODS register is set to 1. If the SET_CONFIGURATION request is processed by FW, the status of this register is immediately applied to the USFA0MODS register as soon as data has been written to this register (CONF bit = 1 before completion of the status stage).

Access This register is read-only, in 8-bit units.

Address 304_H

Initial value 00_H. This register is initialized by any reset.

Caution Do not write to this register. If the register is written to, the operation is not guaranteed.

	7	6	5	4	3	2	1	0
USFA0CNF	0	0	0	0	0	0	CONF1	CONF0
	R	R	R	R	R	R	R	R

Table 30-77 USFA0CNF register contents

Bit position	Bit name	Function
1, 0	CONF1, CONF0	These bits hold the data to be returned in response to the GET_CONFIGURATION request.

(11) Interface 0 register (USFA0IF0)

This register stores the value that is to be returned in response to the GET_INTERFACE wIndex = 0 request.

When the SET_INTERFACE request is received, its wValue is automatically written to this register.

If the SET_INTERFACE request is processed by FW, wIndex and wValue are decoded, and the endpoint setting is automatically changed. At this time, the status bit of the target endpoint and DPID are automatically cleared to 0, depending on the setting. The FIFO is not cleared automatically.

Access This register is read-only, in 8-bit units.

Address 308_H

Initial value 00_H. This register is initialized by any reset.

Caution Do not write to this register. If the register is written to, the operation is not guaranteed.

	7	6	5	4	3	2	1	0
USFA0IF0	0	0	0	0	0	IF02	IF01	IF00
	R	R	R	R	R	R	R	R

Table 30-78 USFA0IF0 register contents

Bit position	Bit name	Function
2:0	IF0[2:0]	These bits hold the data to be returned in response to the GET_INTERFACE wIndex = 0 request.

(12) Interface 1 to 4 registers (USFA0IF1 to USFA0IF4)

These registers store the value that is to be returned in response to the GET_INTERFACE wIndex = n request (n = 1 to 4).

When the SET_INTERFACE request is received, its wValue is automatically written to these registers.

These registers are disabled according to the setting of the USFA0AIFN and USFA0AAS registers.

If the SET_INTERFACE request is processed by FW, wIndex and wValue are decoded, and the endpoint setting is automatically changed. At this time, the status bit of the target endpoint and DPID are automatically cleared to 0, depending on the setting. The FIFO is not cleared automatically.

Access These registers are read-only, in 8-bit units.

Address $30C_H$: USFA0IF1

310_H: USFA0IF2

314_H: USFA0IF3

318_H: USFA0IF4

Initial value 00H. These registers are initialized by any reset.

Caution Do not write to these registers. If the registers are written to, the operation is not guaranteed.

	7	6	5	4	3	2	1	0
USFA0IF1	0	0	0	0	0	IF12	IF11	IF10
	R	R	R	R	R	R	R	R
USFA0IF2	0	0	0	0	0	IF22	IF21	IF20
	R	R	R	R	R	R	R	R
USFA0IF3	0	0	0	0	0	IF32	IF31	IF30
	R	R	R	R	R	R	R	R
USFA0IF4	0	0	0	0	0	IF42	IF41	IF40
	R	R	R	R	R	R	R	R

Table 30-79 USFA0IF1 to USFA0IF4 register contents

Bit position	Bit name	Function
2:0	IFn[2:0]	These bits hold the data to be returned in response to the GET_INTERFACE wIndex = n request.

Note n = 1 to 4

(13) Descriptor length register (USFA0DSCL)

This register stores the length of the value that is to be returned in response to the GET_DESCRIPTOR Configuration request. The value of this register is the number of bytes of all the descriptors set by the USFA0CIEn register minus 1 (n = 0 to 255). The total descriptor length that is to be returned in response to the GET_DESCRIPTOR Configuration request is determined according to the value of this register.

wLength processing is automatically controlled. If this register is set to 00_H, it means the descriptor to be returned is 1 byte long. If the register is set to FF_H, a descriptor length of 256 bytes is returned. When a descriptor exceeding 256 bytes in length is used, set the CDCGDST bit of the USFA0MODC register to 1 and process the GET_DESCRIPTOR request by using FW. (At this time, the CDCGD bit of the USFA0MODS register is also set to 1.)

Access This register can be read or written in 8-bit units. However, data can be written to this register only when the EP0NKA bit is set to 1.

Address 340_H

Initial value 00_H. This register is initialized by any reset.

Caution To rewrite this register, set the EP0NKA bit to 1 before reading the register contents, and then rewrite the register contents after confirming that the bit has been set, in order to prevent a conflict between reading and writing.

	7	6	5	4	3	2	1	0
USFA0DSCL	DPL7	DPL6	DPL5	DPL4	DPL3	DPL2	DPL1	DPL0
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30-80 USFA0DSCL register contents

Bit position	Bit name	Function
7:0	DPL[7:0]	These bits specify the value of the number of bytes of all the descriptors to be returned in response to the GET_DESCRIPTOR Configuration request minus 1.

(14) Device descriptor registers 0 to 17 (USFA0DD0 to USFA0DD17)

These registers store the value to be returned in response to the GET_DESCRIPTOR Device request.

Access These registers can be read or written in 8-bit units. However, data can be written to these registers only when the EP0NKA bit is set to 1.

Address For details, see *Table 30-81 "Mapping and data of device descriptor registers"*.

Initial value Undefined.

Cautions

1. To rewrite these registers, set the EP0NKA bit to 1 before reading the register contents, and then rewrite the register contents after confirming that the bit has been set, in order to prevent a conflict between reading and writing.
2. Use the value defined by USB Specification Ver. 2.0 and the latest Class Specification as the setting.

	7	6	5	4	3	2	1	0
USFA0DDn								
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 30-81 Mapping and data of device descriptor registers

Symbol	Address	Field Name	Description
USFA0DD0	344_H	bLength	Size of this descriptor
USFA0DD1	348_H	bDescriptorType	Device descriptor type
USFA0DD2	$34C_H$	bcdUSB	Value after the decimal point of the revision number in the USB specifications
USFA0DD3	350_H		Value before the decimal point of the revision number in the USB specifications
USFA0DD4	354_H	bDeviceClass	Class code
USFA0DD5	358_H	bDeviceSubClass	Subclass code
USFA0DD6	$35C_H$	bDeviceProtocol	Protocol code
USFA0DD7	360_H	bMaxPacketSize0	Maximum packet size of Endpoint0
USFA0DD8	364_H	idVendor	Lower value of the vendor ID
USFA0DD9	368_H		Higher value of the vendor ID
USFA0DD10	$36C_H$	idProduct	Lower value of the product ID
USFA0DD11	370_H		Higher value of the product ID
USFA0DD12	374_H	bcdDevice	Lower value of the device release number
USFA0DD13	378_H		Higher value of the device release number
USFA0DD14	$37C_H$	iManufacturer	Index of the string descriptor describing the manufacturer
USFA0DD15	380_H	iProduct	Index of the string descriptor describing the product
USFA0DD16	384_H	ISerialNumber	Index of the string descriptor describing the device serial number
USFA0DD17	388_H	BNumConfigurations	Number of settable configurations

(15) Configuration/interface/endpoint descriptor registers 0 to 255 (USFA0CIE0 to USFA0CIE255)

These registers store the value to be returned in response to the GET_DESCRIPTOR Configuration request.

Up to 256 bytes of descriptor information can be stored in these registers. Store each descriptor in the order of Configuration, Interface, and Endpoint (see *Table 30-82 "Mapping of USFA0CIEn Registers"*). If there are multiple Interfaces, repeatedly store the data following the Interface descriptor.

Access These registers can be read or written in 8-bit units. However, data can be written to these registers only when the EP0NKA bit is set to 1.

Address $38C_H$ to 788_H

Initial value Undefined.

Table 30-82 Mapping of USFA0CIEn Registers

Address	Register	Stored Descriptor
$38C_H$	USFA0CIE0	Configuration descriptor 0
…	…	…
$3AC_H$	USFA0CIE8	Configuration descriptor 8
$3B0_H$	USFA0CIE9	Interface0 descriptor 0
…	…	…
$3D0_H$	USFA0CIE17	Interface0 descriptor 0
$3D4_H$	USFA0CIE18	Endpoint1 descriptor 0
…	…	…
$3EC_H$	USFA0CIE24	Endpoint1 descriptor 6
$3F0_H$	USFA0CIE25	Endpoint2 descriptor 0
…	…	…
408_H	USFA0CIE31	Endpoint2 descriptor 6
$40C_H$	USFA0CIE32	Endpoint3 descriptor 0
…	…	…
XXX_H	USFA0CIExxx	Interfacex descriptor 0 (Create the descriptor according to the endpoints supported by the interface.)
…	…	…
$XXX_H + 20_H$	USFA0CIExxx + 8	Interfacex descriptor 8
$XXX_H + 24_H$	USFA0CIExxx + 9	Endpoint1 descriptor 0
…	…	…
$XXX_H + 3C_H$	USFA0CIExxx + 15	Endpoint1 descriptor 6
$XXX_H + 40_H$	USFA0CIExxx + 16	Endpoint2 descriptor 0
…	…	…
$XXX_H + 58_H$	USFA0CIExxx + 22	Endpoint2 descriptor 6
$XXX_H + 5C_H$	USFA0CIExxx + 23	Interfacex descriptor 0
…	…	…

The range of the valid data that can be specified for these registers varies according to the setting of the USFA0DSCL register. In addition to the descriptors listed in *Table 30-82 "Mapping of USFA0CIEn Registers"*, descriptors specific to classes and vendors can also be stored.

If all the values are fixed, they can be stored in ROM.

Cautions

1. To rewrite these registers, set the EP0NKA bit to 1 before reading the register contents, and then rewrite the register contents after confirming that the bit has been set, in order to prevent a conflict between reading and writing.
2. Use the value defined by USB Specification Ver. 2.0 and the latest Class Specification as the setting.

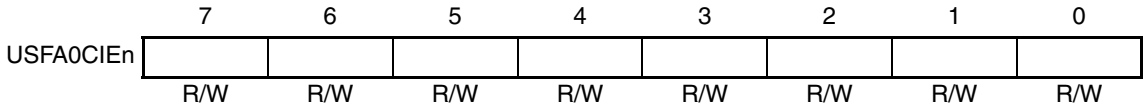


Table 30-83 USFA0CIEn register data

(a) Configuration descriptor (9 bytes)

Offset	Field Name	Description
0	bLength	Size of this descriptor
1	bDescriptorType	Descriptor type
2	wTotalLength	Lower value of the total number of bytes of the Configuration descriptor, all Interface descriptors, and all Endpoint descriptors
3		Higher value of the total number of bytes of the Configuration descriptor, all interface descriptors, and all endpoint descriptors
4	bNumInterface	Number of interfaces
5	bConfigurationValue	Value to select this Configuration
6	iConfiguration	Index of the string descriptor describing this Configuration
7	bmAttributes	Configuration features (self-powered, no remote wakeup)
8	MaxPower	Maximum power consumption of this Configuration (unit: mA)[a]

a) The power consumption is shown in 2 mA units. (Example: 50 = 100 mA)

(b) Interface descriptor (9 bytes)

Offset	Field Name	Description
0	bLength	Size of this descriptor
1	bDescriptorType	Descriptor type
2	bInterfaceNumber	Value of this interface
3	bAlternateSetting	Value to select the alternative interface setting
4	bNumEndpoints	Number of usable endpoints
5	bInterfaceClass	Class code
6	bInterfaceSubClass	Subclass code
7	bInterfaceProtocol	Protocol code
8	Interface	Index of the string descriptor describing this interface

(c) Endpoint descriptor (7 bytes)

Offset	Field Name	Description
0	bLength	Size of this descriptor
1	bDescriptorType	Descriptor type
2	bEndpointAddress	Address/transfer direction of this endpoint
3	bmAttributes	Transfer type
4	wMaxPaketSize	Lower value of the maximum data transfer number
5		Higher value of the maximum data transfer number
6	bInterval	Transfer interval

30.7.5 Bridge registers

(1) H-bus bridge interrupt status register (USFA0BRGINT)

This register indicates the status of the interrupts generated by the H-bus bridge.

Access This register can be read or written in 32-bit units.

Address 808_H

Initial value $0000\ 0000_H$. This register is initialized by any reset.

	31	30	29	28	27	26	25	24
USFA0BRGINT	0	0	0	DMA_ERRINT[4:1]				0
	R	R	R	R/W	R/W	R/W	R/W	R
	23	22	21	20	19	18	17	16
	0	0	0	DMA_ENDINT[4:1]				0
	R	R	R	R/W	R/W	R/W	R/W	R
	15	14	13	12	11	10	9	8
	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R
	7	6	5	4	3	2	1	0
	0	MBUS_ ERRINT	SBUS_ ERRINT1	SBUS_ ERRINT0	ERR_MASTER[3:0]			
	R	R/W	R/W	R/W	R	R	R	R

Table 30-84 USFA0BRGINT register contents

Bit position	Bit name	Function
28:25	DMA_ ERRINT [4:1]	These bits are set to 1 when, during a DMA transfer with EPC, the epc_drqb signal becomes inactive faster than the maximum packet size in bursts specified for EPn_MPKT[8:0]. Writing 1 to the DMA_ERRINT[4:1] bits clears them to 0.
20:17	DMA_ ENDINT [4:1]	These bits are set to 1 when a DMA transfer finishes successfully or such a transfer terminates with an error. Writing 1 to the DMA_ENDINT[4:1] bits clears them to 0.
6	MBUS_ ERRINT	This bit is set to 1 if an error response is received when attempting to access the H bus during master operation. Writing 1 to the MBUS_ERRINT bit clears it to 0.
5	SBUS_ ERRINT1	When there is access from the master specified by NOT_RETRY_MASTER, an error response is generated and this bit is set to 1. Writing 1 to the SBUS_ERRINT1 bit clears it to 0.
4	SBUS_ ERRINT0	When there is access that uses a bus width of 32 bits or more, an error response is generated and this bit is set to 1. Writing 1 to the SBUS_ERRINT0 bit clears it to 0.
3:0	ERR_ MASTER [3:0]	These bits store the master number in the error response generated when the SBUS_ERRINT1 or SBUS_ERRINT0 bit is set to 1. The ERR_MASTER[3:0] bits retain their current value until SBUS_ERRINT[1:0] is cleared to 00, even if another source is generated.

(2) H-bus bridge interrupt enable register (USFA0BRGINTE)

This register controls the interrupts generated by the H-bus bridge.

Access This register can be read or written in 32-bit units.

Address $80C_H$

Initial value $0000\ 0000_H$. This register is initialized by any reset.

	31	30	29	28	27	26	25	24
USFA0BRGINTE	0	0	0	DMA_ERRINTEN[4:1]				0
	R	R	R	R/W	R/W	R/W	R/W	R
	23	22	21	20	19	18	17	16
	0	0	0	DMA_ENDINTEN[4:1]				0
	R	R	R	R/W	R/W	R/W	R/W	R
	15	14	13	12	11	10	9	8
	0	0	0	VBUS_INTEN	0	0	0	0
	R	R	R	R/W	R	R	R	R
	7	6	5	4	3	2	1	0
	0	MBUS_ERRINTEN	SBUS_ERRINT1EN	SBUS_ERRINT0EN	0	0	0	0
	R	R/W	R/W	R/W	R	R	R	R

Table 30-85 USFA0BRGINTE register contents

Bit position	Bit name	Function
28:25	DMA_ERRINTEN[4:1]	These bits control whether to output an interrupt when the DMA_ERRINT[4:1] bits are set to 1. 0: Do not output the interrupt. 1: Output the interrupt.
20:17	DMA_ENDINTEN[4:1]	These bits control whether to output an interrupt when the DMA_ENDINT[4:1] bits are set to 1. 0: Do not output the interrupt. 1: Output the interrupt.
12	VBUS_INTEN	This bit controls whether to output an interrupt when the VBUS_INT bit is set to 1. 0: Do not output the interrupt. 1: Output the interrupt.
6	MBUS_ERRINT	This bit controls whether to output an interrupt when the MBUS_ERRINT bit is set to 1. 0: Do not output the interrupt. 1: Output the interrupt.
5	SBUS_ERRINT1EN	This bit controls whether to output an interrupt when the SBUS_ERRINT1 bit is set to 1. 0: Do not output the interrupt. 1: Output the interrupt.
4	SBUS_ERRINT0EN	This bit controls whether to output an interrupt when the SBUS_ERRINT0 bit is set to 1. 0: Do not output the interrupt. 1: Output the interrupt.

(3) EPC interrupt status register (USFA0EPCINT)

This register indicates the status of the interrupts generated by EPC.

Access This register is read-only, in 32-bit units.

Address 810_H

Initial value $0000\ 0000_H$. This register is initialized by any reset.

	31	30	29	28	27	26	25	24
USFA0EPCINT	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R
	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R
	15	14	13	12	11	10	9	8
	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R
	7	6	5	4	3	2	1	0
	0	0	0	0	0	EPC_INT2	EPC_INT1	EPC_INT0
	R	R	R	R	R	R	R	R

Table 30-86 USFA0EPCINT register contents

Bit position	Bit name	Function
2	EPC_INT2	This bit indicates the interrupt source of the INT status 4 register (USFA0IS4). If the USFA0IS4 register has changed, 1 is set.
1	EPC_INT1	This bit indicates the interrupt source of the INT status 2 register (USFA0IS2). If the USFA0IS2 register has changed, 1 is set. The bit also indicates the interrupt source of the INT status 3 register. Like USFA0IS3, if the USFA0IS3 register has changed, 1 is set.
0	EPC_INT0	This bit indicates the interrupt source of the INT status 0 register (USFA0IS0). If the USFA0IS0 register has changed, 1 is set. The bit also indicates the interrupt source of the INT status 1 register. Like USFA0IS0, if the USFA0IS register has changed, 1 is set.

(4) EPC interrupt enable register (USFA0EPCINTE)

This register controls the interrupts generated by EPC.

Access This register can be read or written in 32-bit units.

Address 814_H

Initial value $0000\ 0000_H$. This register is initialized by any reset.

	31	30	29	28	27	26	25	24
USFA0EPCINTE	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R
	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R
	15	14	13	12	11	10	9	8
	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R
	7	6	5	4	3	2	1	0
	0	0	0	0	0	EPC_ INT2EN	EPC_ INT1EN	EPC_ INT0EN
	R	R	R	R	R	R/W	R/W	R/W

Table 30-87 USFA0EPCINTE register contents

Bit position	Bit name	Function
2	EPC_ INT2EN	This bit controls whether to output an interrupt when the EPC_INT2 bit is set to 1. 0: Do not output the interrupt. 1: Output the interrupt.
1	EPC_ INT1EN	This bit controls whether to output an interrupt when the EPC_INT1 bit is set to 1. 0: Do not output the interrupt. 1: Output the interrupt.
0	EPC_ INT0EN	This bit controls whether to output an interrupt when the EPC_INT0 bit is set to 1. 0: Do not output the interrupt. 1: Output the interrupt.

(5) EPC macro control register (USFA0EPCCTL)

This register controls the reset signal to the EPC circuit.

Access This register can be read or written in 32-bit units.

Address 818_H

Initial value 00000001_H. This register is initialized by any reset.

	31	30	29	28	27	26	25	24
USFA0EPCCTL	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R
	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R
	15	14	13	12	11	10	9	8
	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	EPC_RST
	R	R	R	R	R	R	R	R/W

Table 30-88 USFA0EPCCTL register contents

Bit position	Bit name	Function
0	EPC_RST	This bit controls the reset signal to EPC. The initial value is 1 (the reset status). 0: Cancel the reset. 1: Reset EPC.

(6) Endpoint n DMA control register 1 (USFA0EnDC1)

This register sets up DMA transfer control.

Access This register can be read or written in 32-bit units.

Address 910_H (USFA0E1DC1)

920_H (USFA0E2DC1)

930_H (USFA0E3DC1)

940_H (USFA0E4DC1)

Initial value 0000 0000_H. This register is initialized by any reset.

	31	30	29	28	27	26	25	24
USFA0EnDC1	EPn_DSCNT[15:8]							
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	23	22	21	20	19	18	17	16
	EPn_DSCNT[7:0]							
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	15	14	13	12	11	10	9	8
	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	EPn_DIR0	EPn_REQEN
	R	R	R	R	R	R	R	R/W

Table 30-89 USFA0EnDC1 register contents

Bit position	Bit name	Function
31:16	EPn_ DSCNT [15:0]	These bits specify the number of packets (not bytes) on which to perform a DMA transfer. The value of these bits is decremented each time a DMA transfer is performed on one packet. **EPn_DSCNT[15:0] / Number of packets** 0000_H / 65536 0001_H / 65535 … / … $FFFE_H$ / 1 $FFFF_H$ / 0 Specify values for the EPnDCNT[15:0] bits only when the EPnREQEN bit is 0.
1	EPnDIR0	This bit indicates the EPC direction. 0: IN transfer (USFA0E1DC1 and EP3DCR3 registers) 1: OUT transfer (EP2DCR1and EP4DCR3 registers)
0	EPn_REQEN	This bit controls DMA requests from EPC. 0: Mask DMA requests. 1: Enable DMA requests. **Note** If even one of the following occurs, the EPnREQEN bit is cleared to 0: - Transferring the number of packets specified by the EPnDSCNT[15:0] bits finishes. - An end signal (epc_stopb) is received from EPC. - DMA_ERRINT is generated.

(7) Endpoint n DMA control register 2 (USFA0EnDC2)

This register sets up DMA transfer control.

Access This register can be read or written in 32-bit units.

Address 914_H (USFA0E1DC2)

924_H (USFA0E2DC2)

934_H (USFA0E3DC2)

944_H (USFA0E4DC2)

Initial value $0000\ 0000_H$. This register is initialized by any reset.

	31	30	29	28	27	26	25	24
EPnDCR2	0	0	0	0	0	0	0	EPn_LMPKT8
	R	R	R	R	R	R	R	R/W
	23	22	21	20	19	18	17	16
	EPn_LMPKT[7:0]							
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	15	14	13	12	11	10	9	8
	0	0	0	0	0	0	0	EPn_MPKT8
	R	R	R	R	R	R	R	R/W
	7	6	5	4	3	2	1	0
	EPn_MPKT[7:0]							
	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Table 30-90 USFA0EnDC2 register contents

Bit position	Bit name	Function
24:16	EPn_LMPKT[8:0]	These bits specify the maximum size of the EPn end packet. The operation differs for IN transfers (USFA0EnDC1.EPnDIR0 = 0) and OUT transfers (USFA0EnDC1.EPnDIR0 = 1). - For an IN transfer (USFA0EnDC1.EPnDIR0 = 0) These bits specify the number of bytes in the end packet. After transferring the specified number of bytes during a DMA transfer, epc_endb is generated for EPC and the DMA transfer ends. - For an OUT transfer (USFA0EnDC1.EPnDIR0 = 1) These bits indicate the number of bytes in an end packet specified for a DMA transfer. Specify values for the EPnLMPKT[8:0] bits only when the EPnREQEN bit is 0.
8:0	EPn_MPKT[8:0]	These bits specify the maximum EPn packet size. The lower two bits are fixed to 00. Specify values for the EPnMPKT[8:0] bits only when the EPnREQEN bit is 0. To specify a maximum packet size of 64 bytes, specify 040_H.

(8) Endpoint n DMA transfer address register 1 (USFA0EnTAD)

This register specifies the starting address for performing a DMA transfer to the H bus side of each Endpoint.

Access This register can be read or written in 32-bit units.

Address 918_H (USFA0E1TAD)

928_H (USFA0E2TAD)

938_H (USFA0E3TAD)

948_H (USFA0E4TAD)

Initial value $0000\ 0000_H$. This register is initialized by any reset.

	31	30	29	28	27	26	25	24
USFA0EnTAD	HB_STATADR[31:24]							
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	23	22	21	20	19	18	17	16
	HB_STATADR[23:16]							
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	15	14	13	12	11	10	9	8
	HB_STATADR[15:8]							
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	7	6	5	4	3	2	1	0
	HB_STATADR[7:0]							
	R/W	R/W	R/W	R/W	R/W	R/W	R	R

Table 30-91 USFA0EnTAD register contents

Bit position	Bit name	Function
31:0	HBn_ TADR[31:0]	These bits specify the starting address for performing a transfer to the H bus side. Except for when the end packet is a short packet, transfers are performed in 32-bit units for the H bus, so the lower two bits are fixed to 00. The value of the USFA0EnTAD register is incremented each time a DMA transfer is performed. In addition, even when transferring short packets for which the number of bytes is not divisible by 4, the value of the USFA0EnTAD register is incremented in 4-byte units. For example, when 63 bytes are transferred, the value is incremented by 64 (40_H). Specify values for the HBnSTATADR[31:0] bits only when the EPnREQEN bit is 0.

The following flowcharts show the program execution when the host is disconnected and then reconnected, as well as the program execution when power is supplied.

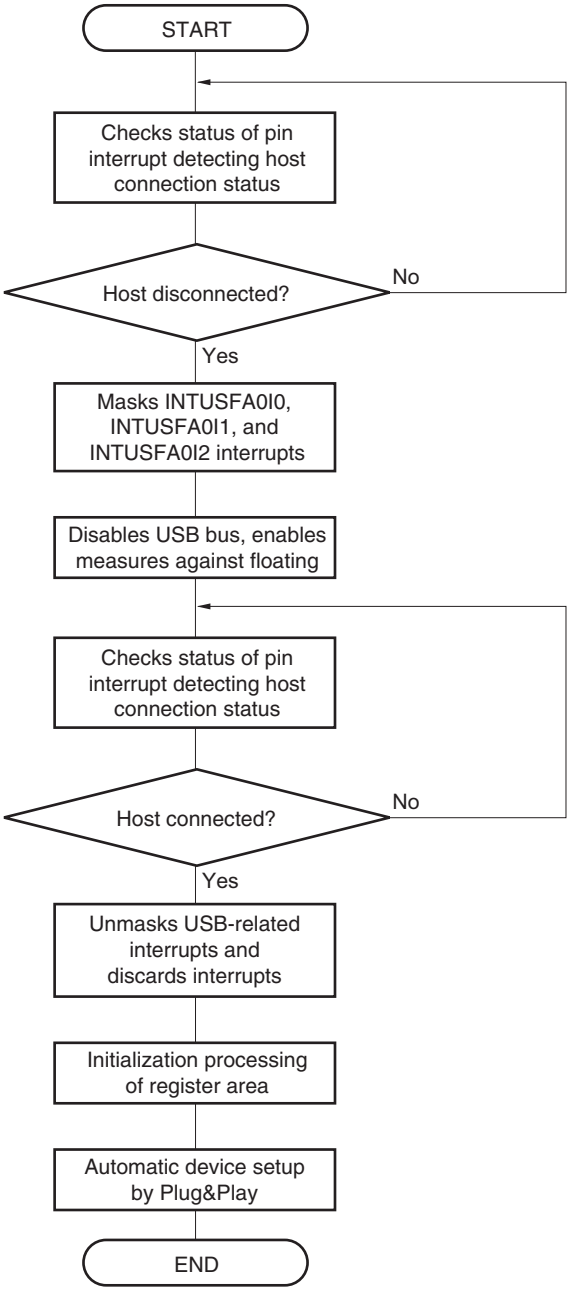


Figure 30-17 Flowchart of program when host is disconnected and then reconnected

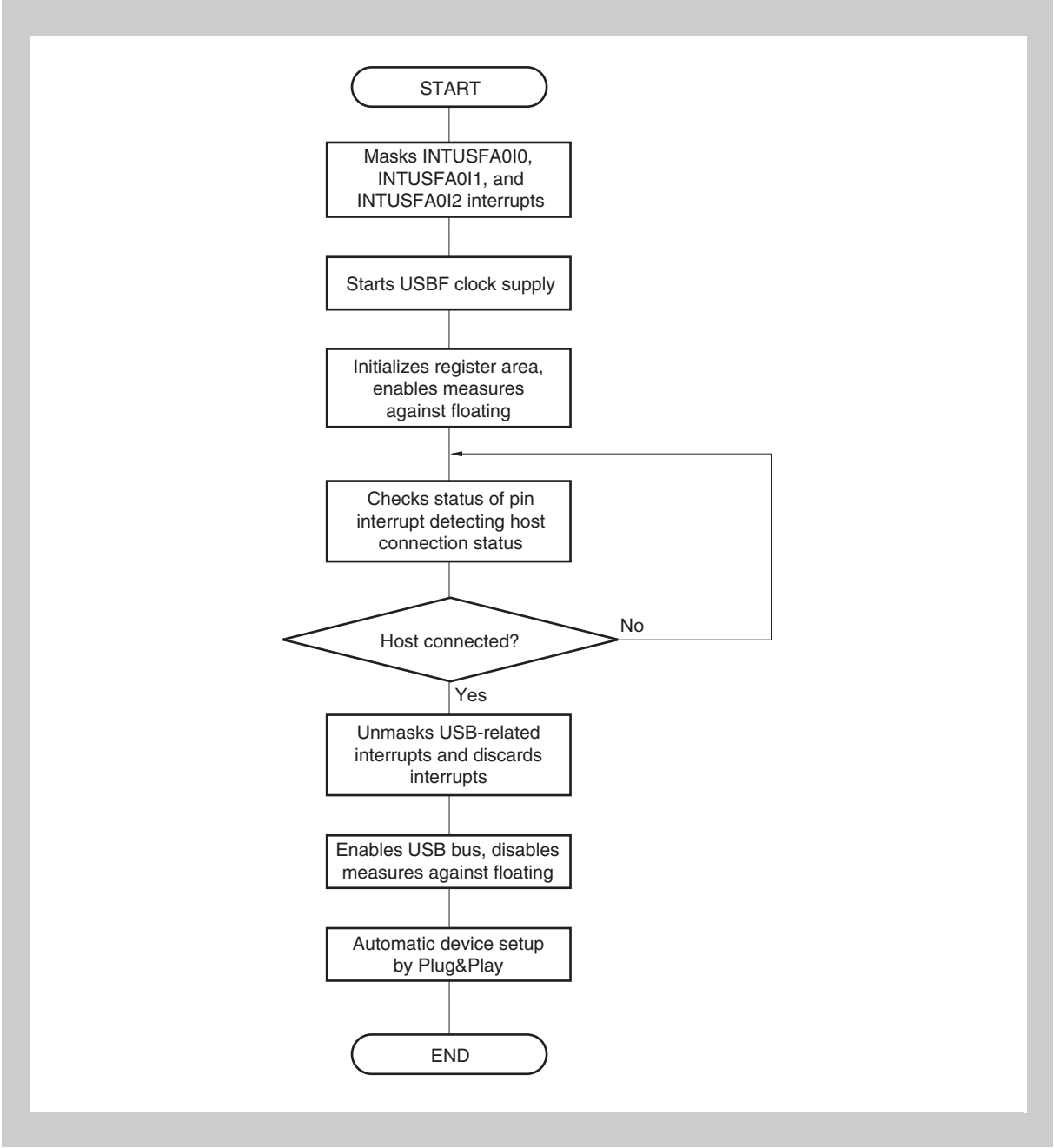


Figure 30-18 Flowchart of program when power is supplied

30.8 STALL Handshake or No Handshake

USBF error handling is defined as follows:

Transfer type	Transaction	Target packet	Error type	Function response	Processing
Control transfer, bulk transfer, interrupt transfer	IN/OUT/SETUP	Token	Endpoint not supported	No response	None
			Endpoint transfer direction mismatch	No response	
			CRC error	No response	
			Bit stuffing error	No response	
Control transfer, bulk transfer	OUT/SETUP	Data	Timeout	No response	None
			PID check error	No response	
			Unsupported PID (other than data PID)	No response	
			CRC error	No response	Received data is discarded.
			Bit stuffing error	No response	
	OUT	Data	Data PID mismatch	ACK	
Control transfer (SETUP stage)	SETUP	Data	Overrun	No response	Received data is discarded.
Control transfer (data stage)	OUT	Data	Overrun	No response[a]	The SNDSTL bit of USFA0SDS register is set to 1 and the received data is discarded.
Control transfer (status stage)	OUT	Data	Overrun	ACK or no response[b]	
Bulk transfer	OUT	Data	Overrun	No response[a]	The EnHALT bit of the USFA0EnS register (n = 0 to 4, 7) is set to 1.
Control transfer, bulk transfer, interrupt transfer	IN	Hand-shake	PID check error	–	Transferred data is held and re-transferred.[c].
			Unsupported PID (other than ACK PID)	–	
			Timeout	–	

a) A STALL is returned in response to re-transfer by the host.

b) An ACK response is returned if the transfer data is of MaxPacketSize or less and the data received in the status stage is discarded. If MaxPacketSize is exceeded, no response is returned, the SNDSTL bit of the USFA0SDS register is set to 1, and the received data is discarded.

c) If an OUT transaction indicating a change from the data stage to the status stage is received during a control transfer, an error is not handled and it is assumed that reception has been correctly completed.

Cautions

1. The currently specified Alternative Setting number is used to judge whether the target endpoint is valid or invalid.
2. For details about the response to the request included in a control transfer to or from Endpoint0, see *30.6 "Requests"*.

30.9 Register Values in Specific Status

Table 30-92 Register values in a specific status (1/2)

Register name	After a CPU reset ($\overline{RESET}$)	After a bus reset
USFA0E0N register	0000 0000_H	The value is held.
USFA0E0NA register	0000 0000_H	The value is held.
USFA0EN register	0000 0000_H	The value is held.
USFA0ENM register	0000 0000_H	The value is held.
USFA0SDS register	0000 0000_H	The value is held.
USFA0CLR register	0000 0000_H	The value is held.
USFA0SET register	0000 0000_H	The value is held.
USFA0EPS0 register	0000 0000_H	The value is held.
USFA0EPS1 register	0000 0000_H	The value is held.
USFA0EPS2 register	0000 0000_H	The value is held.
USFA0IS0 register	0000 0000_H	The value is held.
USFA0IS1 register	0000 0000_H	The value is held.
USFA0IS2 register	0000 0000_H	The value is held.
USFA0IS3 register	0000 0000_H	The value is held.
USFA0IS4 register	0000 0000_H	The value is held.
USFA0IM0 register	0000 0000_H	The value is held.
USFA0IM1 register	0000 0000_H	The value is held.
USFA0IM2 register	0000 0000_H	The value is held.
USFA0IM3 register	0000 0000_H	The value is held.
USFA0IM4 register	0000 0000_H	The value is held.
USFA0IC0 register	FFFF $FFFF_H$	The value is held.
USFA0IC1 register	FFFF $FFFF_H$	The value is held.
USFA0IC2 register	FFFF $FFFF_H$	The value is held.
USFA0IC3 register	FFFF $FFFF_H$	The value is held.
USFA0IC4 register	FFFF $FFFF_H$	The value is held.
USFA0IDR register	0000 0000_H	The value is held.
USFA0DMS0 register	0000 0000_H	The value is held.
USFA0DMS1 register	0000 0000_H	The value is held.
USFA0FIC0 register	0000 0000_H	The value is held.
USFA0FIC1 register	0000 0000_H	The value is held.
USFA0DEND register	0000 0000_H	The value is held.
USFA0GPR register	0000 0000_H	The value is held.
USFA0MODC register	0000 0000_H	The value is held.
USFA0MODS register	0000 0000_H	Bit 2 (CONF): Cleared to 0, Other bits: Values are held.
USFA0AIFN register	0000 0000_H	The value is held.
USFA0AAS register	0000 0000_H	The value is held.
USFA0ASS register	0000 0000_H	0000 0000_H
USFA0E1IM register	0000 0000_H	The value is held.

Table 30-92 Register values in a specific status (2/2)

Register name	After a CPU reset ($\overline{RESET}$)	After a bus reset
USFA0E2IM register	0000 0000_H	The value is held.
USFA0E3IM register	0000 0000_H	The value is held.
USFA0E4IM register	0000 0000_H	The value is held.
USFA0E7IM register	0000 0000_H	The value is held.
USFA0E0R register	Undefined[a]	The value is held.
USFA0E0L register	0000 0000_H	The value is held.
USFA0E0ST register	0000 0000_H	0000 0000_H
USFA0E0W register	Undefined[a]	The value is held.
USFA0BO1 register	Undefined[a]	The value is held.
USFA0BO1L register	0000 0000_H	The value is held.
USFA0BO2 register	Undefined[a]	The value is held.
USFA0BO2L register	0000 0000_H	The value is held.
USFA0BI1 register	Undefined[a]	The value is held.
USFA0BI2 register	Undefined[a]	The value is held.
USFA0INT1 register	Undefined	The value is held.
USFA0DST register	0000 0000_H	0000 0000_H
USFA0E0S register	0000 0000_H	0000 0000_H
USFA0E1S register	0000 0000_H	0000 0000_H
USFA0E2S register	0000 0000_H	0000 0000_H
USFA0E3S register	0000 0000_H	0000 0000_H
USFA0E4S register	0000 0000_H	0000 0000_H
USFA0E7S register	0000 0000_H	0000 0000_H
USFA0E8S register	0000 0000_H	0000 0000_H
USFA0ADRS register	0000 0000_H	0000 0000_H
USFA0CNF register	0000 0000_H	0000 0000_H
USFA0IF0 register	0000 0000_H	0000 0000_H
USFA0IF1 register	0000 0000_H	0000 0000_H
USFA0IF2 register	0000 0000_H	0000 0000_H
USFA0IF3 register	0000 0000_H	0000 0000_H
USFA0IF4 register	0000 0000_H	0000 0000_H
USFA0DSCL register	0000 0000_H	The value is held.
USFA0DDn registers (n = 0 to 17)	[b]	[b]
USFA0CIEn registers (n = 0 to 255)	[b]	[b]

a) This register is cleared to 0 when the $\overline{RESET}$ signal is asserted because the data in this register, including the write pointer, counter, and read pointer, is controlled by using a FIFO, in the same manner as when the USFA0FICn register is cleared.

b) These registers cannot be cleared to 0. Because data can be written to them by using FW, however, any value can be written to the registers. (Before doing so, however, be sure to set the EP0NKA bit of the USFA0E0NA register to 1.)

30.10 FW Processing

The following FW processing is performed:

- Setup processing on the device side for the SET_CONFIGURATION, SET_INTERFACE, SET_FEATURE, and CLEAR_FEATURE requests during enumeration processing
- Analysis and processing of XXXXStandard, XXXXClass, and XXXXVendor requests not subject to automatic processing
- Reading data following a bulk-transferred OUT token from the Rx buffer
- Writing data to be returned in response to a bulk-transferred IN token
- Writing data to be returned in response to an interrupt-transferred token

The following table lists the requests supported by FW:

Table 30-93 FW-supported standard requests

Request	Reception side	Processing /frequency	Description
CLEAR_FEATURE	Interface	Automatic STALL response	Though this request is reserved for bmRequestType, it is assumed that this request does not come to the interface because no function selector value is specified. When this request is received, the hardware makes an automatic STALL response.
SET_FEATURE	Interface	Automatic STALL response	Though it the request is reserved for bmRequestType, it is assumed that this request does not come to the interface because no function selector value is specified. When this request is received, the hardware makes an automatic STALL response.
GET_DESCRIPTOR	String	FW	FW returns the string descriptor. When this request is received by using the SETUP token, the hardware generates the CPUDEC interrupt request for FW. FW decodes the contents of the received request, and then writes the data to be returned to the host to the USFA0E0W register.
SET_DESCRIPTOR	Device	FW	FW rewrites the device descriptor. When this request is received by using the SETUP token, the hardware generates the CPUDEC interrupt request for FW. FW decodes the contents of the received request, and then writes the data for the next control transfer (OUT) to the USFA0DDn register (n = 0 to 17).
SET_DESCRIPTOR	Configuration	FW	FW rewrites the configuration descriptor. When this request is received by using the SETUP token, the hardware generates the CPUDEC interrupt request for FW. FW decodes the contents of the received request, and then writes the data for the next control transfer (OUT) to the USFA0CIEn register (n = 0 to 255).
SET_DESCRIPTOR	String	FW	FW rewrites the string descriptor. When this request is received by using the SETUP token, the hardware generates the CPUDEC interrupt request for FW. FW decodes the contents of the received request, and then loads the data for the next control transfer (OUT).
Other	NA	FW	When a request other than the above is received by using the SETUP token, the hardware generates the CPUDEC interrupt request for FW. FW decodes the contents of the received request, and then performs the necessary processing.

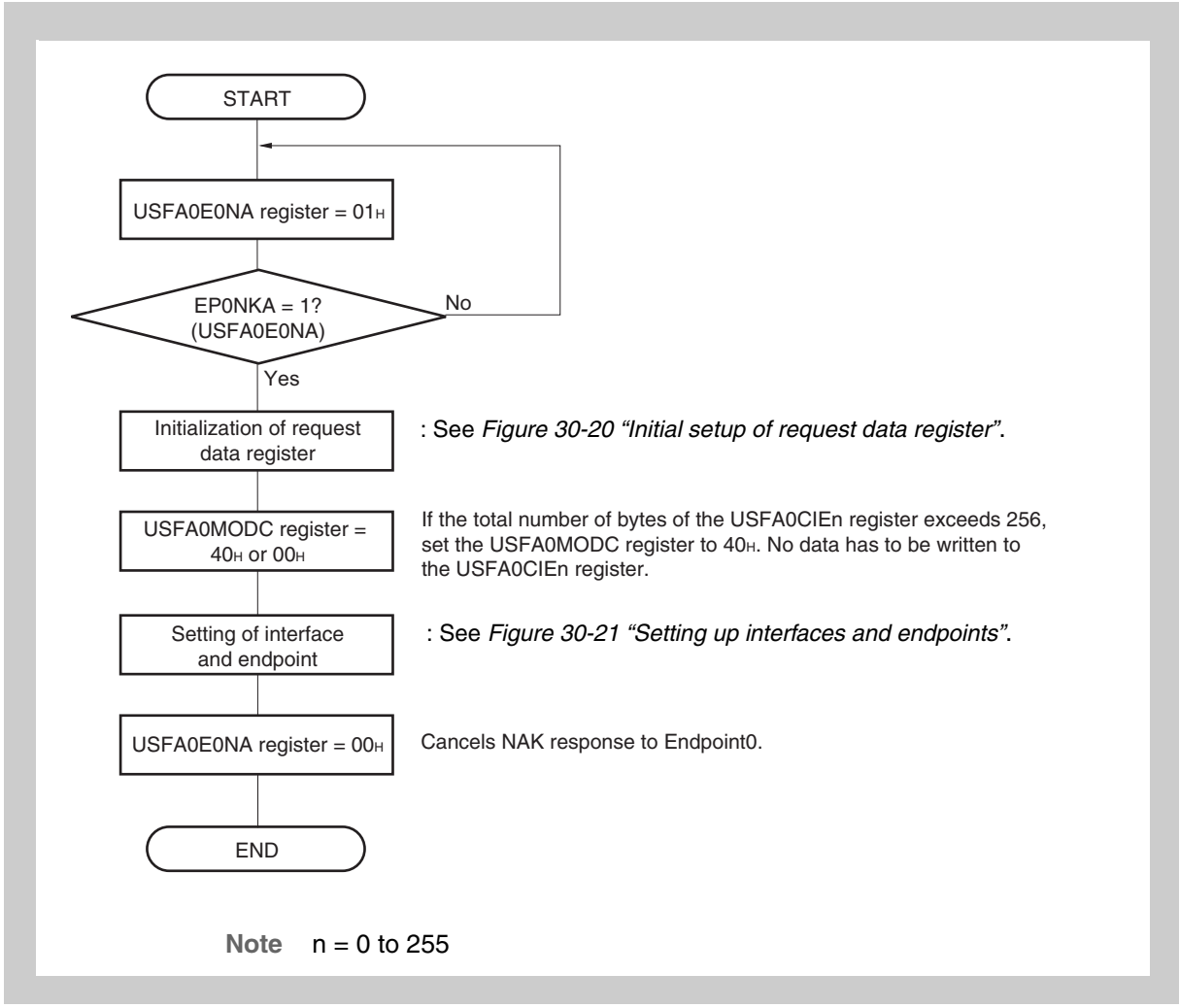
30.10.1 Initialization

Initialization is executed in the following two ways:

- Initializing the request data register
- Setting up interrupts

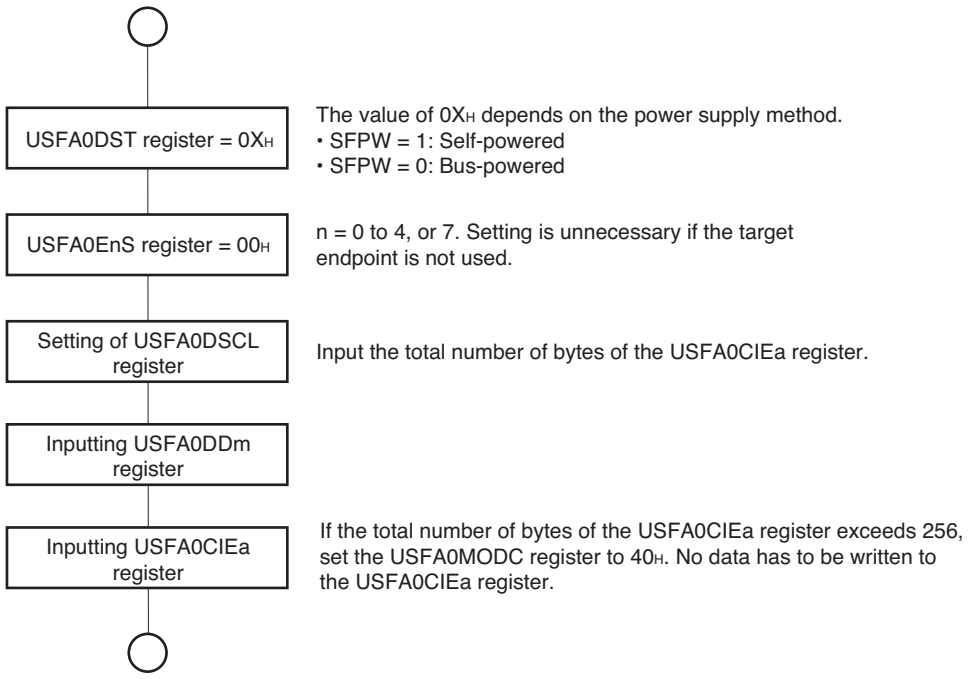
During initialization of request data registers, data for the GET_XXXX request to which a response is automatically returned is written, and an endpoint is assigned to the corresponding interface. In the interrupt settings, the interrupt sources that do not have to be checked can be masked by using the USFA0IMn register (n = 0 to 4).

The following flowcharts show the related processing.



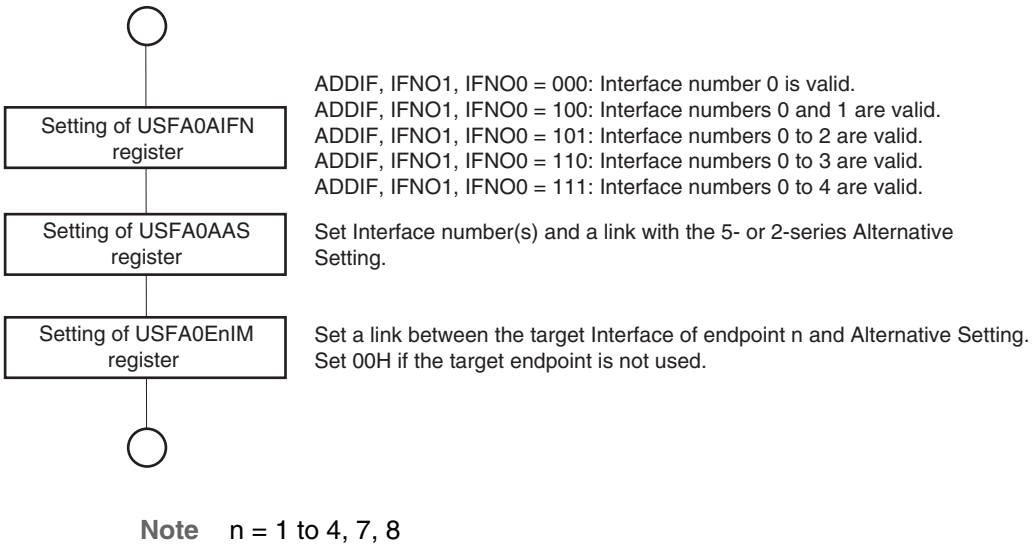
Note n = 0 to 255

Figure 30-19 Initializing request data registers



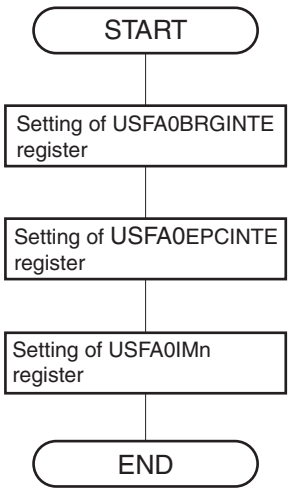
Note m = 0 to 17
a = 0 to 255

Figure 30-20 Initial setup of request data register



Note n = 1 to 4, 7, 8

Figure 30-21 Setting up interfaces and endpoints



Control the interrupts generated by the H bus-EPC bridge (for which the status is reflected in the USFA0BRGINT register).

Control the interrupts generated by EPC (for which the status is reflected in the USFA0EPCINT register).

Mask the interrupt source to avoid issuance of an unnecessary interrupt request (IINTUSFA0I1).

Note n = 0 to 4

Figure 30-22 Setting up interrupts

30.10.2 Interrupt servicing

The following flowcharts show the related processing.

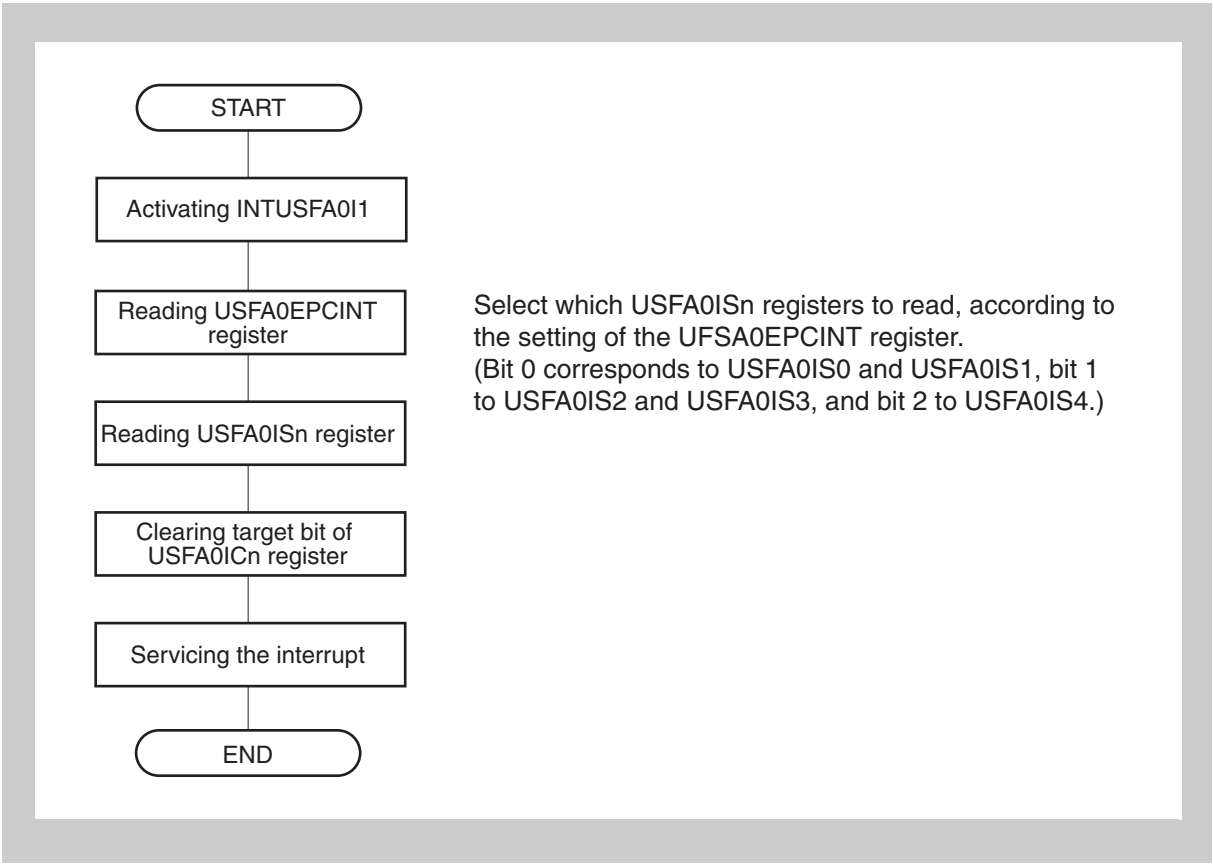


Figure 30-23 EPC interrupt detection

The following bits of the USFA0ISn register are cleared by hardware when a given condition is satisfied (n = 0 to 4):

- The E0INDT, E0ODT, SUCES, STG, and CPUDEC bits of the USFA0IS1 register
- The BKI2DT, BKI1DT, and IT1DT bits of USFA0IS2 register
- The BKO2FL, BKO2DT, BKO1FL, and BKO1DT bits of USFA0IS3 register

Because clearing an interrupt source by using the USFA0ICn register is given a lower priority than setting an interrupt source by hardware, the interrupt source might not be cleared depending on the timing (n = 0 to 4).

30.10.3 Main USB processing

The main USB processing involves processing USB transactions. The types of transactions to be processed are as follows:

- Fully automatically processed requests for control transfers
- Automatically processed requests for control transfers (SET_CONFIGURATION, SET_INTERFACE, SET_FEATURE, and CLEAR_FEATURE)
- CPUDEC requests for control transfers
- Processing for bulk transfers (IN)
- Processing for bulk transfers (OUT)
- Processing for interrupt transfers (IN)

The processing for endpoint n involves reading or writing data for transfers. The flowchart shown below is for PIO.

(1) Fully automatically processed requests for control transfers

Because a fully automatically processed request for a control transfer is executed by hardware, FW cannot reference it. Therefore, FW does not have to perform any special processing for this request.

(2) Automatically processed requests for control transfers

(SET_CONFIGURATION, SET_INTERFACE, SET_FEATURE, and CLEAR_FEATURE)

Processing to write a register for automatically processed requests for control transfers, such as SET_CONFIGURATION, SET_INTERFACE, SET_FEATURE, and CLEAR_FEATURE requests, is automatically executed by hardware, but an interrupt request is issued for recognition on the device side. This processing might be ignored if there is no special processing to execute.

The following flowcharts show the related processing.

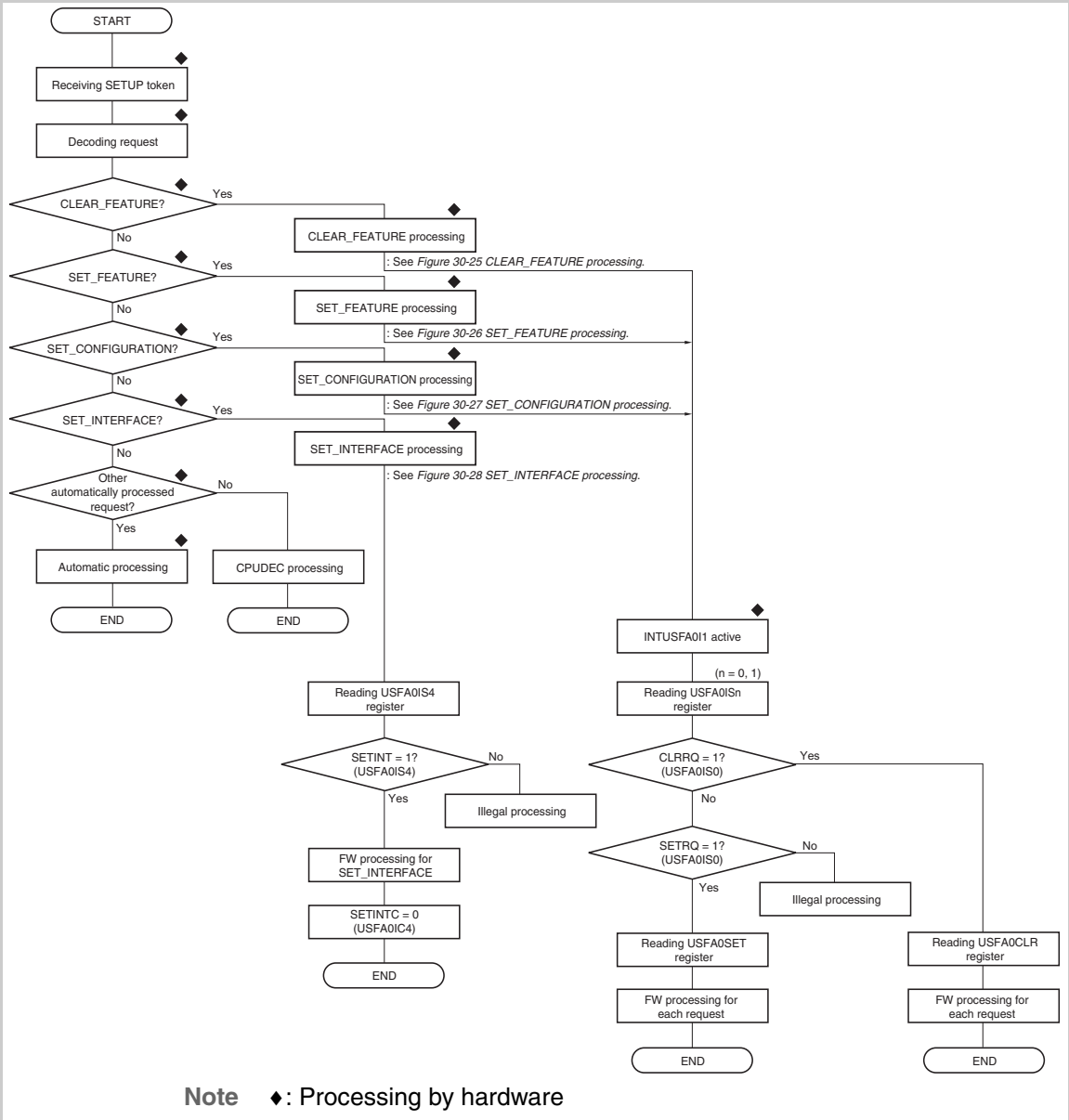
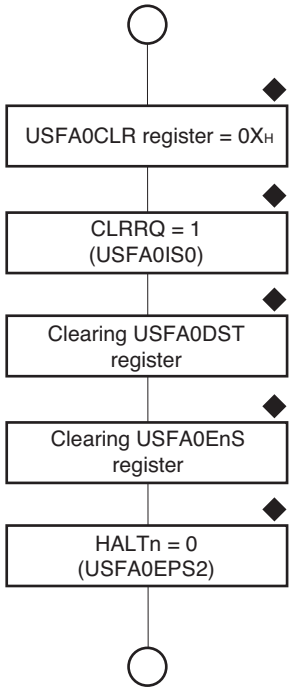


Figure 30-24 Automatically processed requests for control transfers

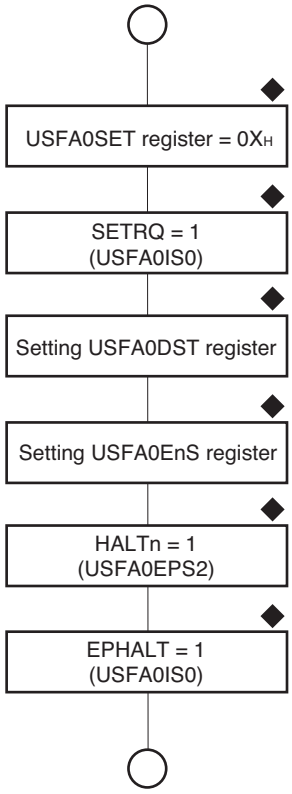


Set the corresponding bit for the value of $0X_H$.
The EPHALT bit of the USFA0IS0 register is cleared to 0 only when all Halt Features are cleared.

Notes 1. n = 0 to 4, 7

2. ♦: Processing by hardware

Figure 30-25 CLEAR_FEATURE processing

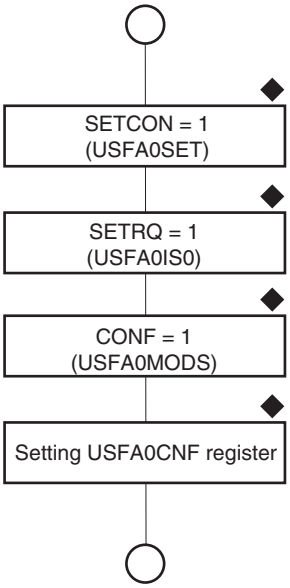


Set the corresponding bit for the value of $0X_H$.
The EPHALT bit of the USFA0IS0 register is not set to 1 by setting the USFA0DST register.

Notes 1. n = 0 to 4, 7

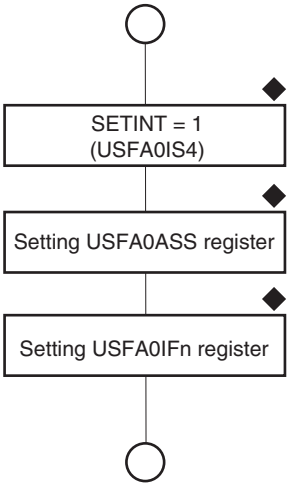
2. ♦: Processing by hardware

Figure 30-26 SET_FEATURE processing



Note ♦: Processing by hardware

Figure 30-27 SET_CONFIGURATION processing



Notes 1. n = 0 to 4

2. ♦: Processing by hardware

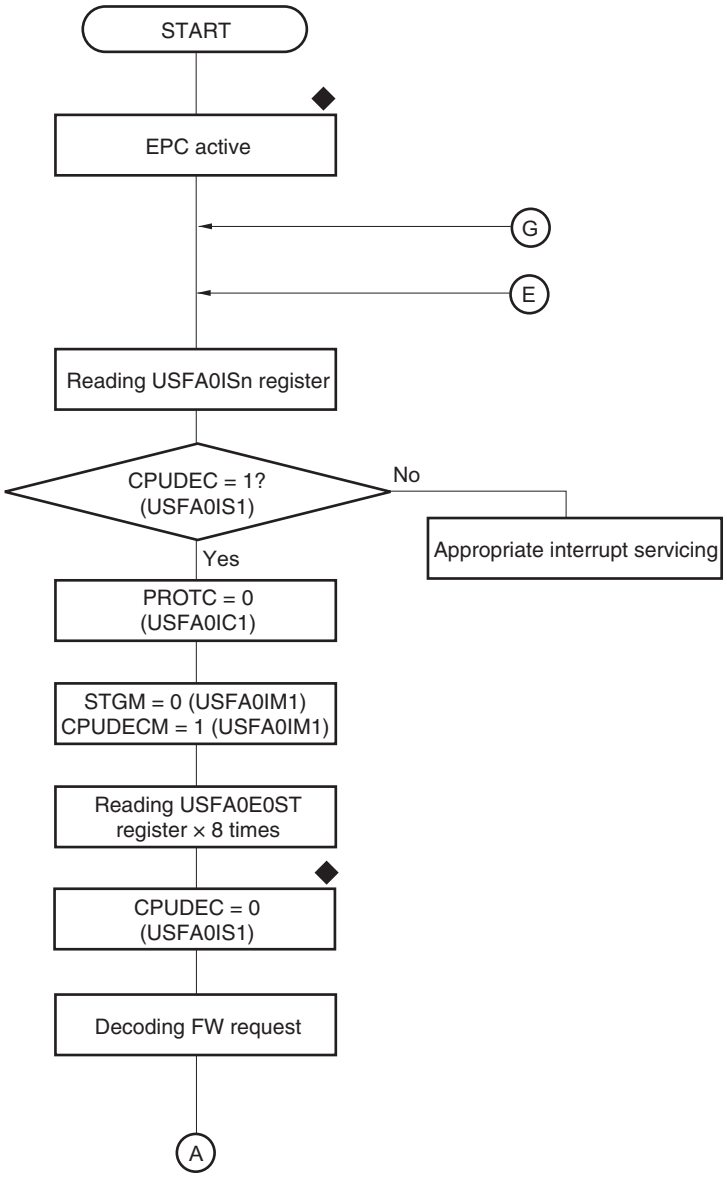
Figure 30-28 SET_INTERFACE processing

(3) CPUDEC requests for control transfers

There are three types of CPUDEC requests: requests for control transfer (write), control transfer (read), and control transfer (without data). Control transfer (write) is a request that uses the OUT transaction in the data stage (such as SET_DESCRIPTOR), control transfer (read) is a request that uses the IN transaction in the data stage (such as GET_DESCRIPTOR), and control transfer (without data) is a request that has no data stage (such as SET_CONFIGURATION).

The following flowcharts show the related processing.

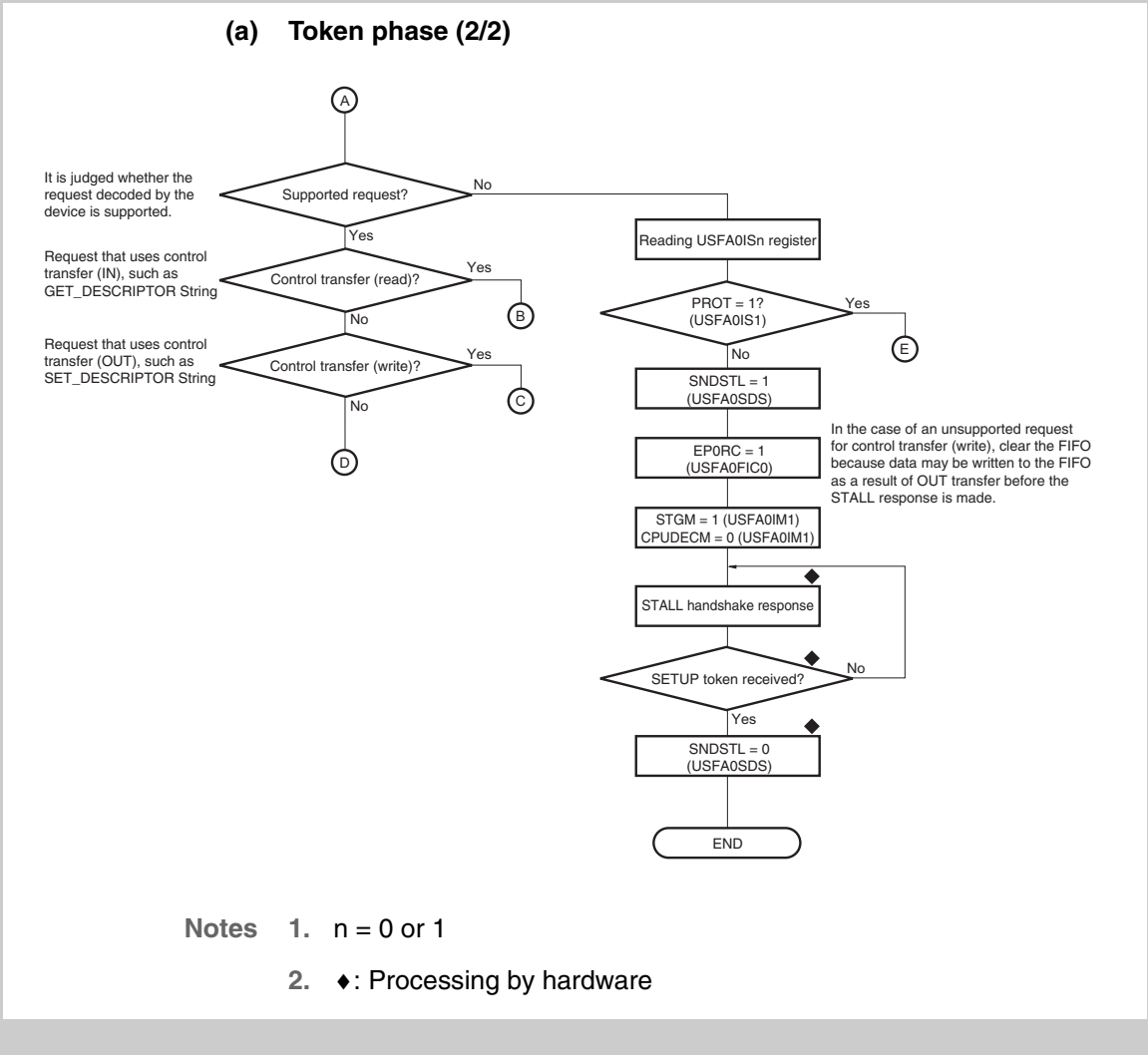
(a) Token phase (1/2)



Notes 1. n = 0 or 1

2. ♦: Processing by hardware

Figure 30-29 CPUDEC requests for control transfers (1/12)

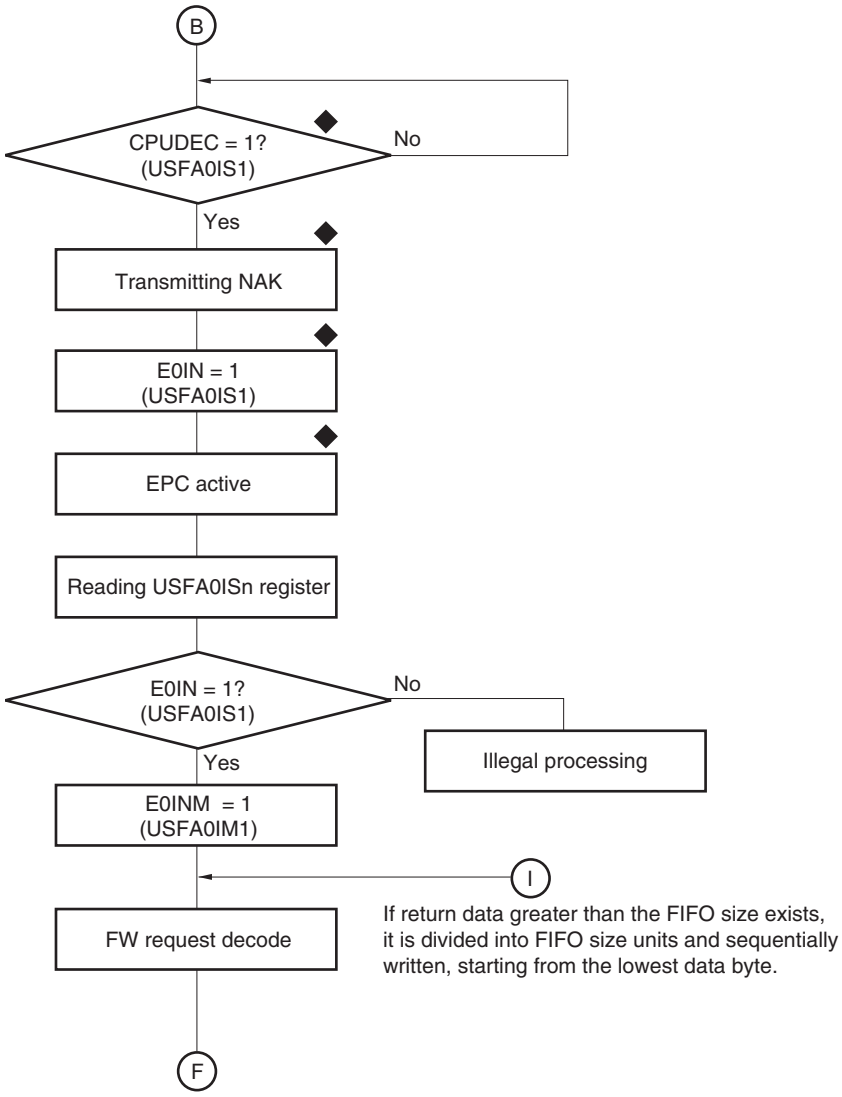


Notes 1. n = 0 or 1

2. ♦: Processing by hardware

Figure 30-29 CPUDEC requests for control transfers (2/12)

(b) Control transfer (read) (1/4)

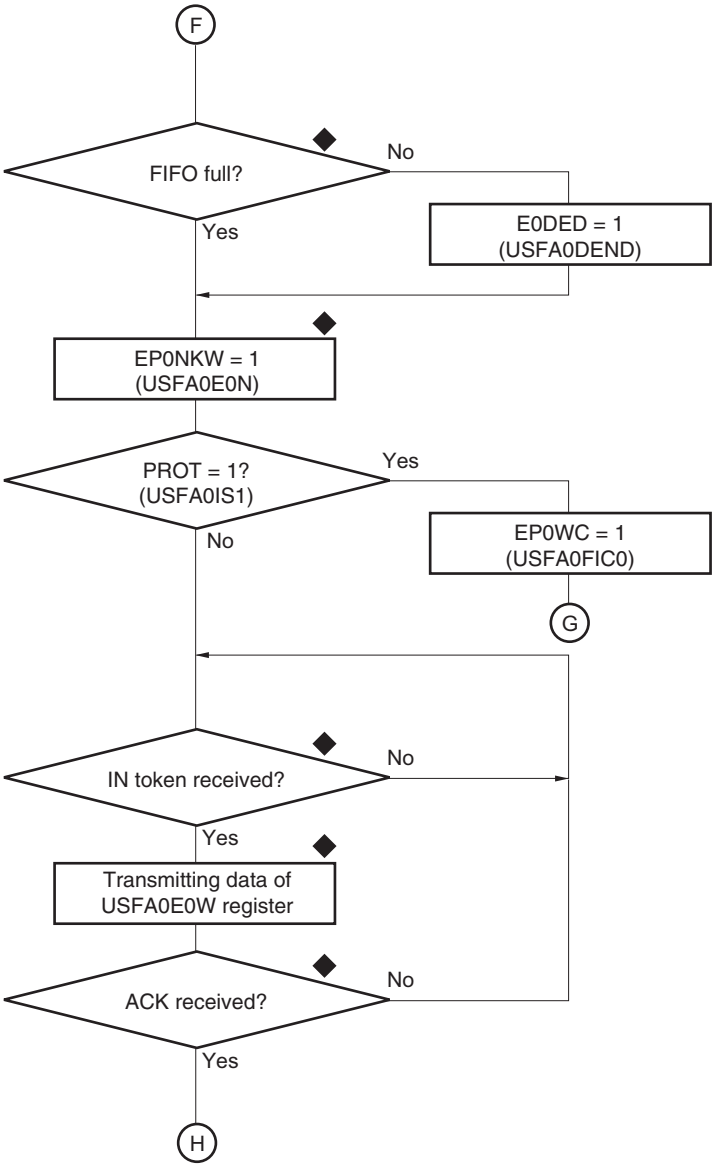


Notes 1. n = 0 or 1

2. ♦: Processing by hardware

Figure 30-29 CPUDEC requests for control transfers (3/12)

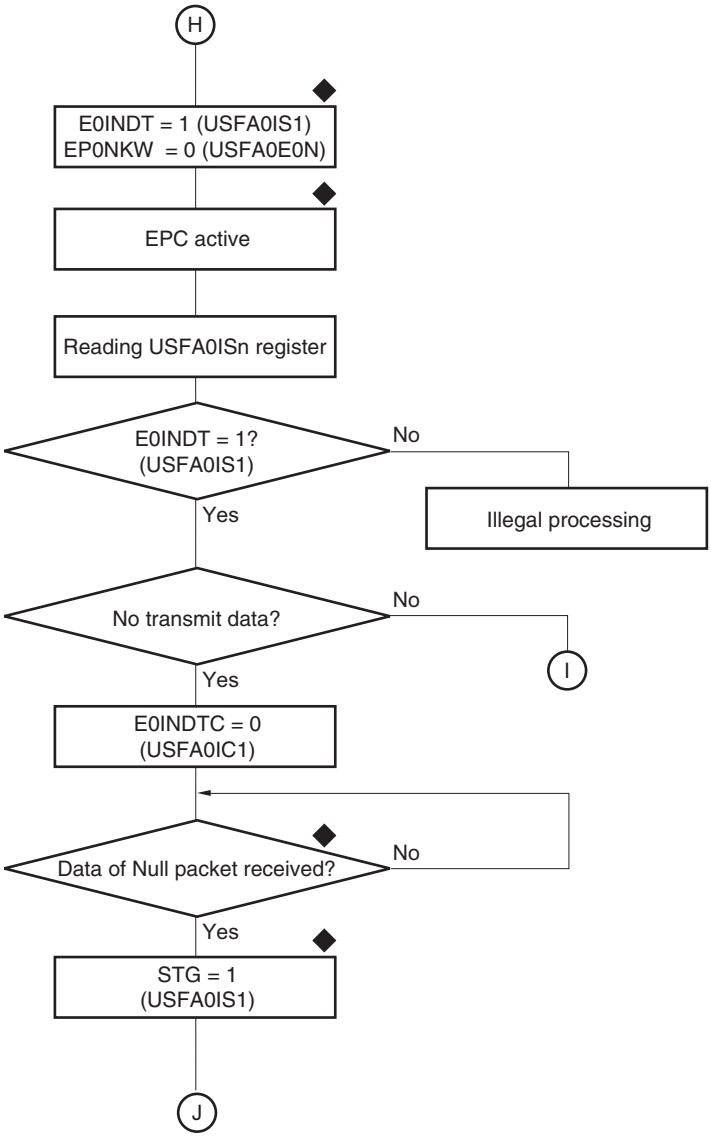
(b) Control transfer (read) (2/4)



Note ♦: Processing by hardware

Figure 30-29 CPUDEC requests for control transfers (4/12)

(b) Control transfer (read) (3/4)

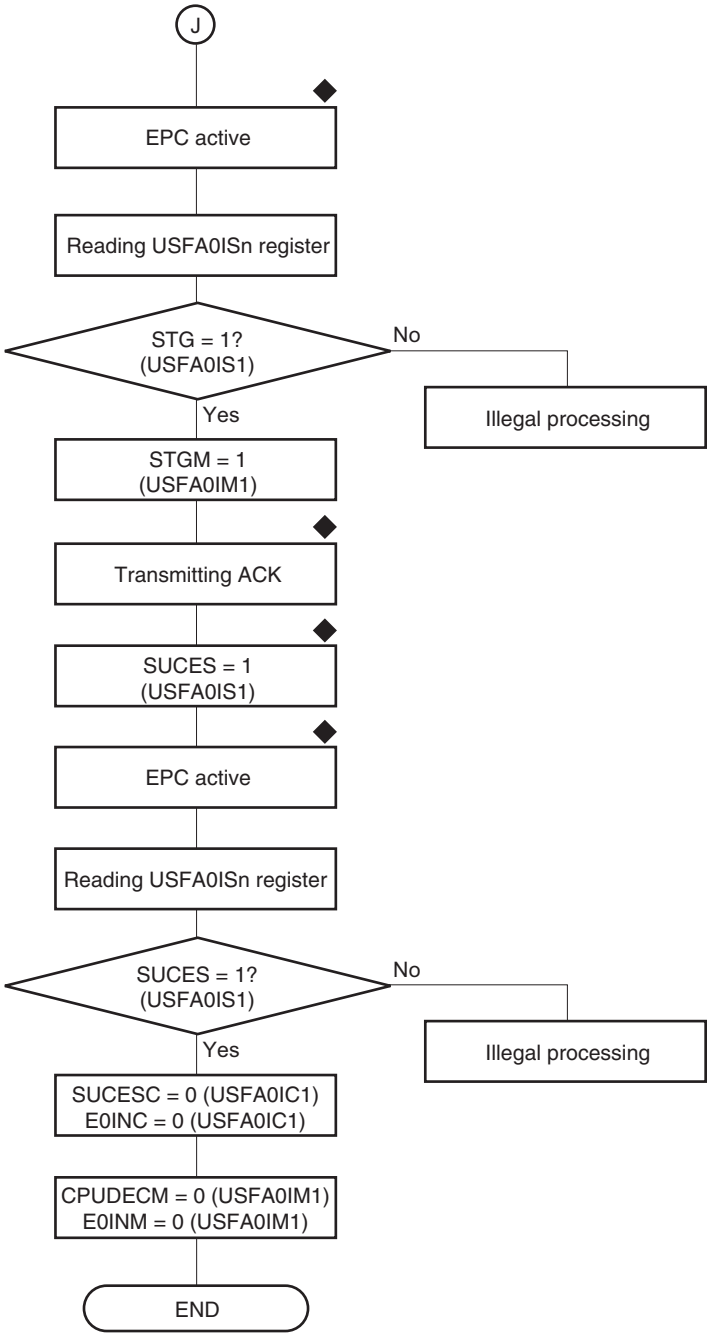


Notes 1. n = 0 or 1

2. ♦: Processing by hardware

Figure 30-29 CPUDEC requests for control transfers (5/12)

(b) Control transfer (read) (4/4)

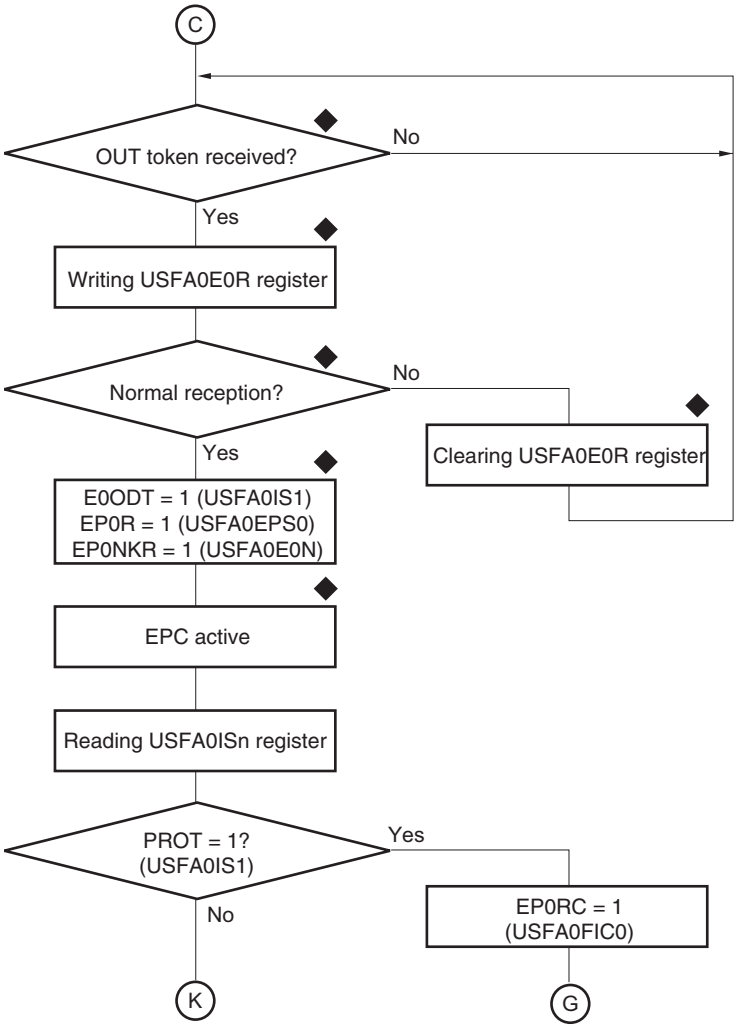


Notes 1. n = 0 or 1

2. ♦: Processing by hardware

Figure 30-29 CPUDEC requests for control transfers (6/12)

(c) Control transfer (write) (1/4)

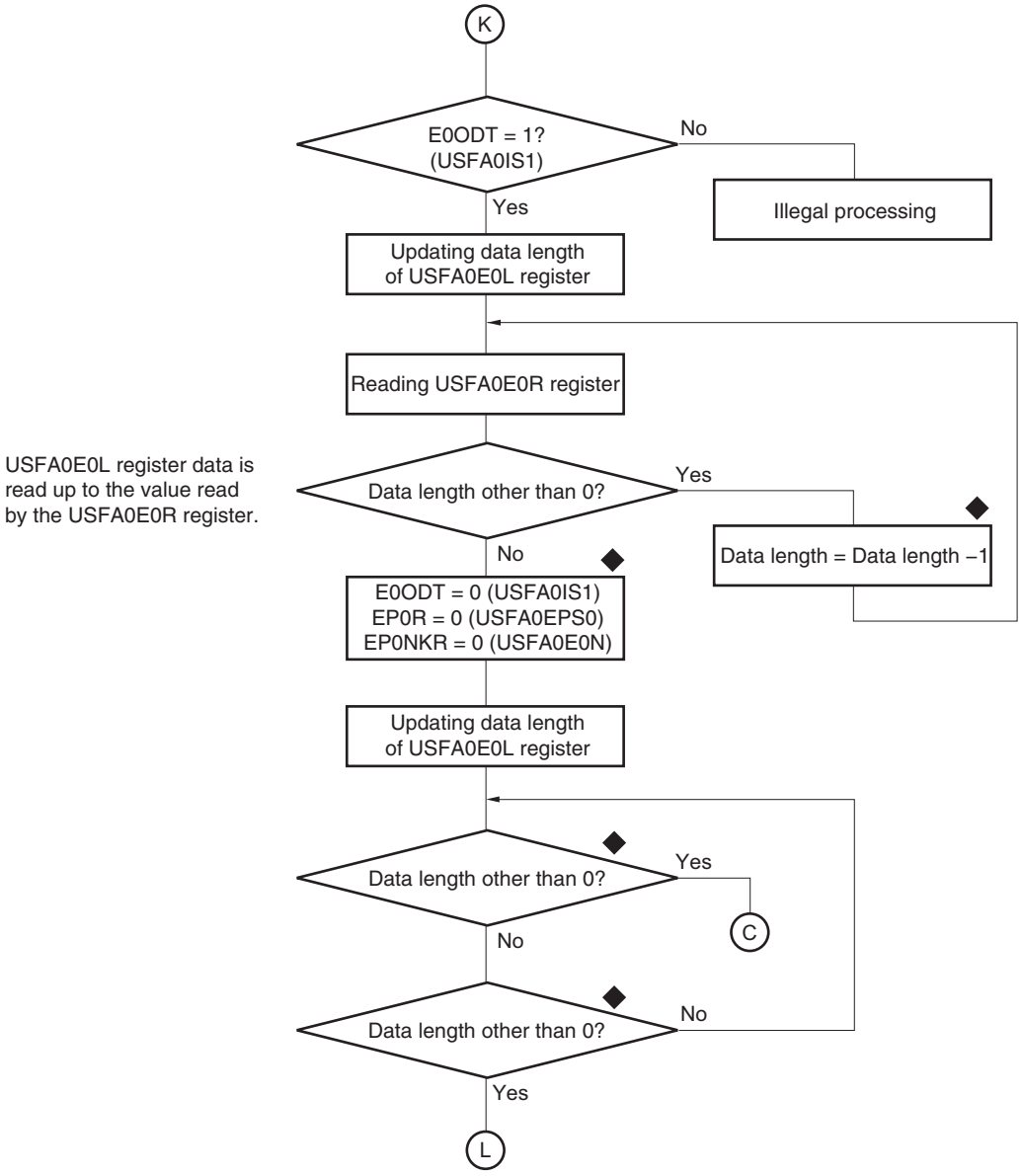


Notes 1. n = 0 or 1

2. ♦: Processing by hardware

Figure 30-29 CPUDEC requests for control transfers (7/12)

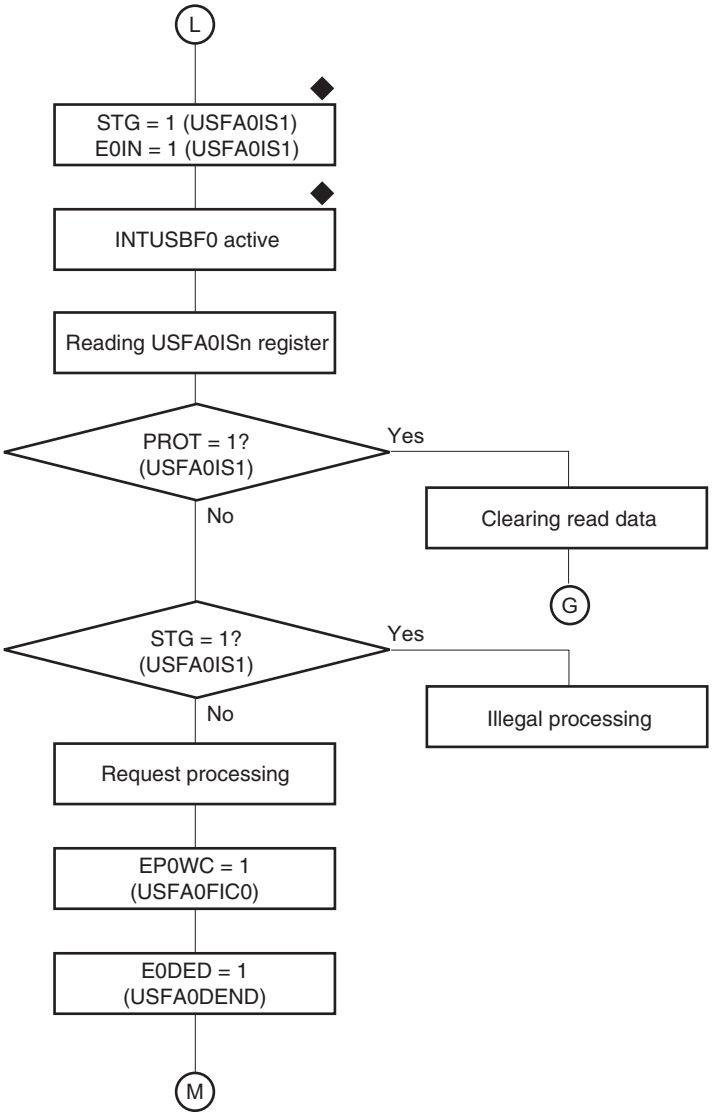
(c) Control transfer (write) (2/4)



Note ♦: Processing by hardware

Figure 30-29 CPUDEC requests for control transfers (8/12)

(c) Control transfer (write) (3/4)

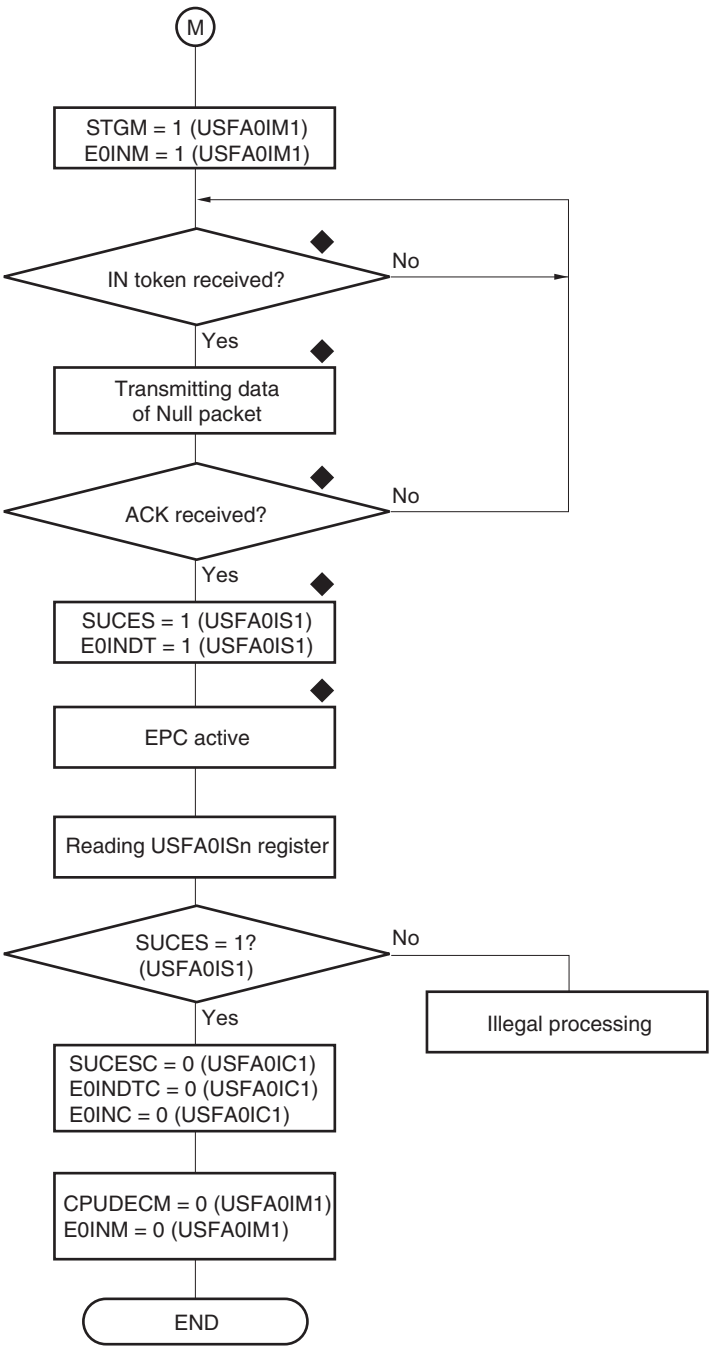


Notes 1. n = 0 or 1

2. ♦: Processing by hardware

Figure 30-29 CPUDEC requests for control transfers (9/12)

(c) Control transfer (write) (4/4)

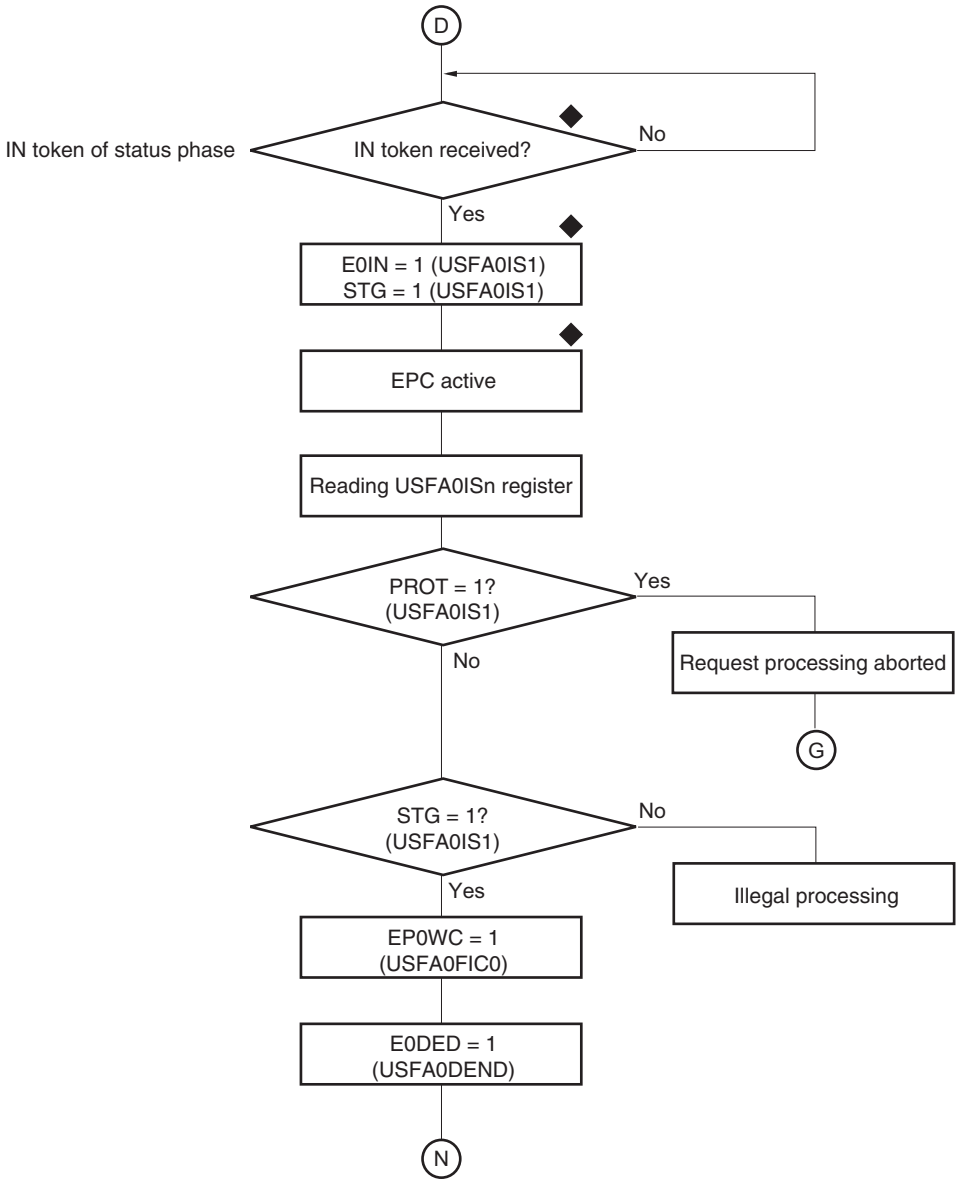


Notes 1. n = 0 or 1

2. ♦: Processing by hardware

Figure 30-29 CPUDEC requests for control transfers (10/12)

(d) Control transfer (no data stage) (1/2)

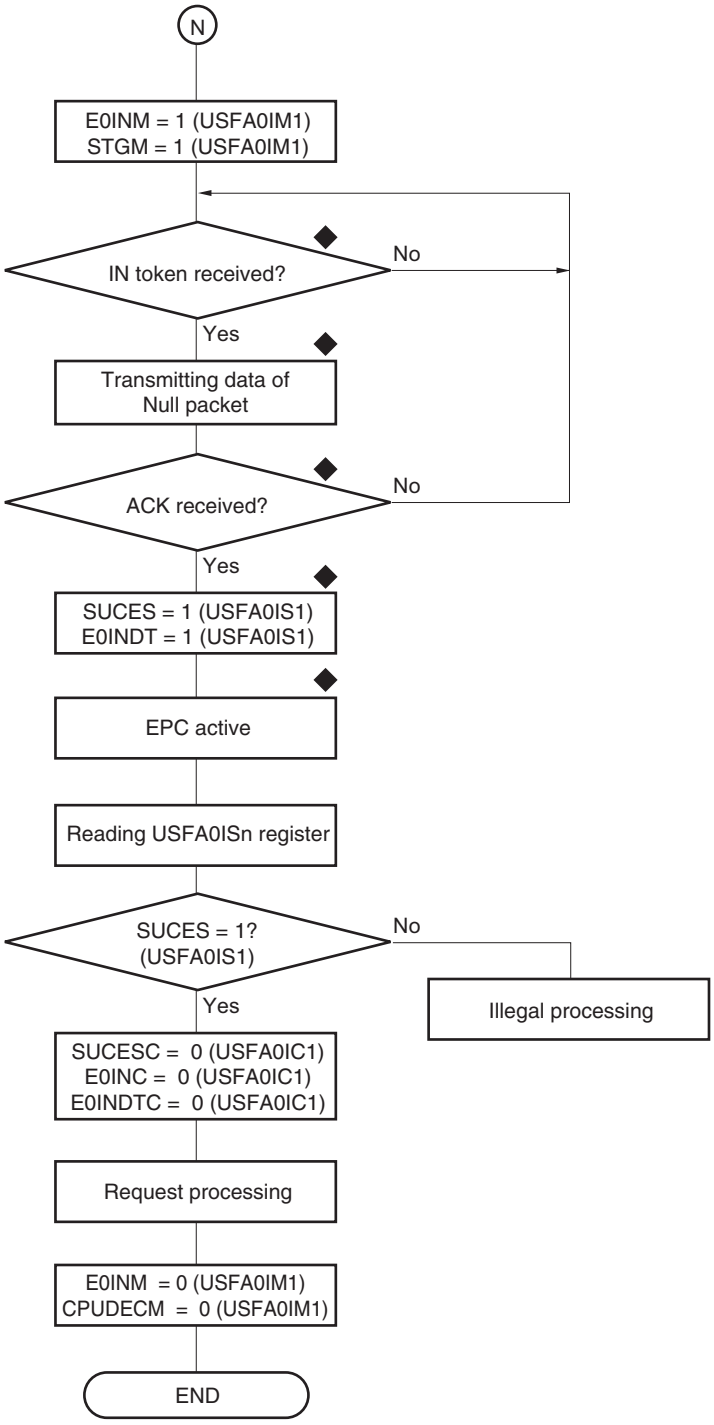


Notes 1. n = 0 or 1

2. ♦: Processing by hardware

Figure 30-29 CPUDEC requests for control transfers (11/12)

(d) Control transfer (no data stage) (2/2)



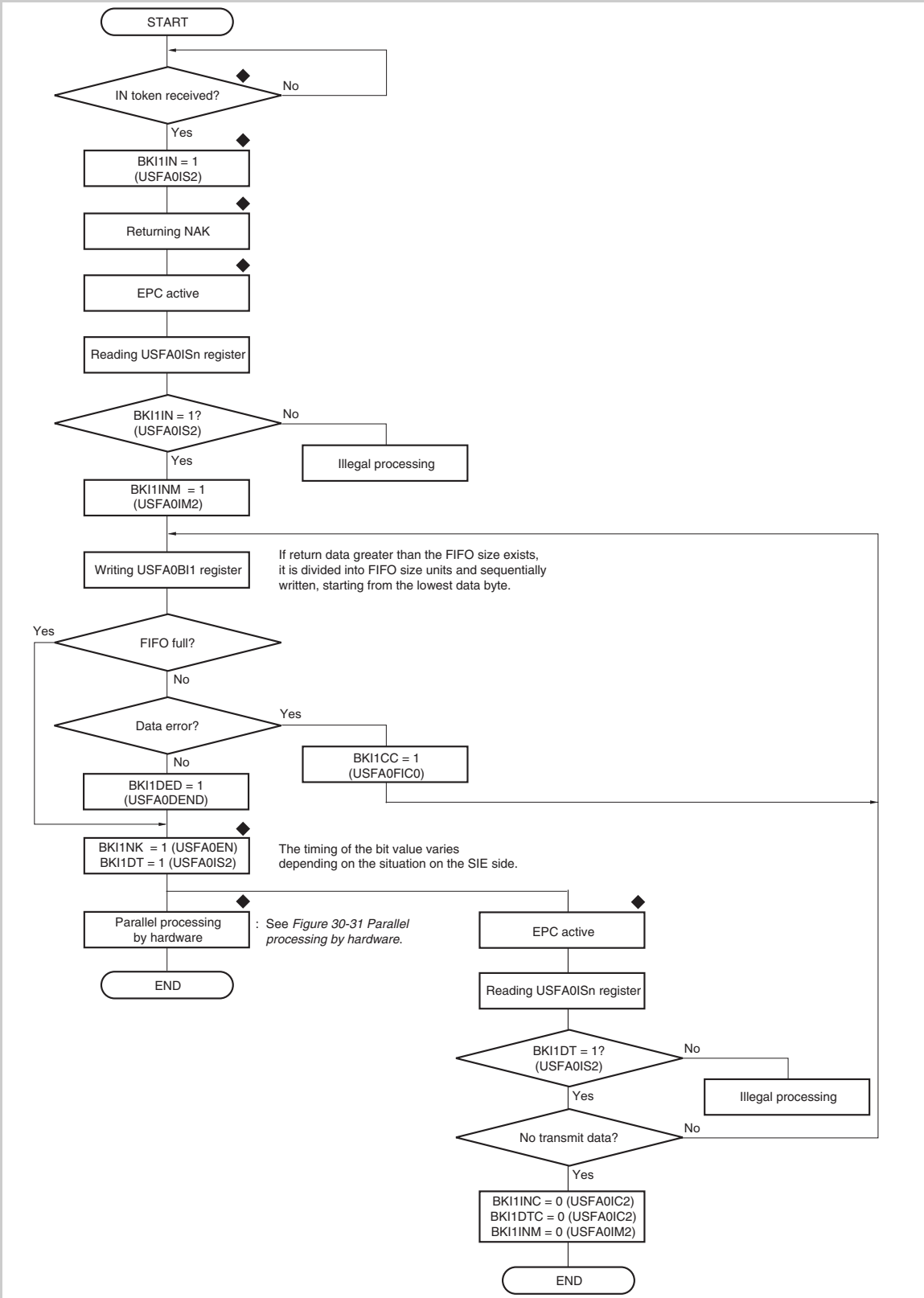
Notes 1. n = 0 or 1

2. ♦: Processing by hardware

Figure 30-29 CPUDEC requests for control transfers (12/12)

(4) Processing for bulk transfers (IN)

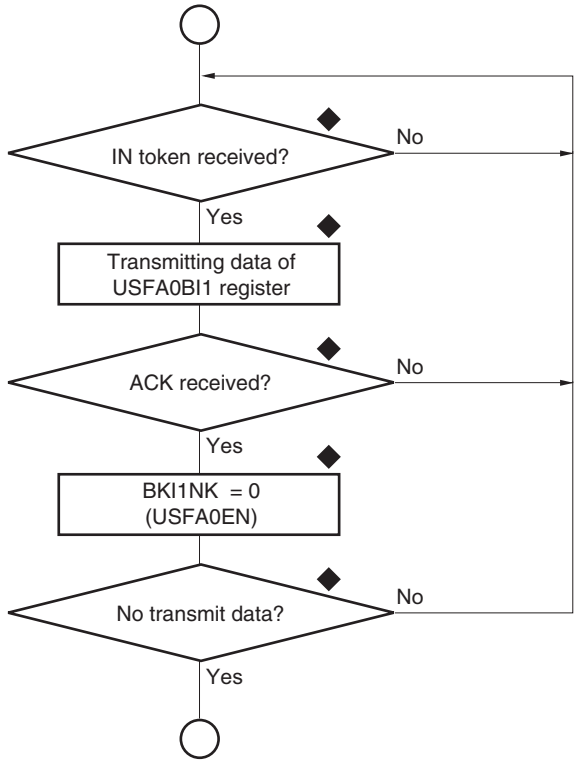
A bulk transfer (IN) is assigned to Endpoints1 and 3. The flowchart below shows how Endpoint1 is controlled. Endpoint3 can also be controlled using the same sequence. To use this flowchart as the control flow of Endpoint3, therefore, read the bit names of Endpoint1 in the flowchart as those of Endpoint3.



Notes 1. n = 2 or 3

2. ♦: Processing by hardware

Figure 30-30 Processing for Bulk transfers (IN) (Endpoint1)

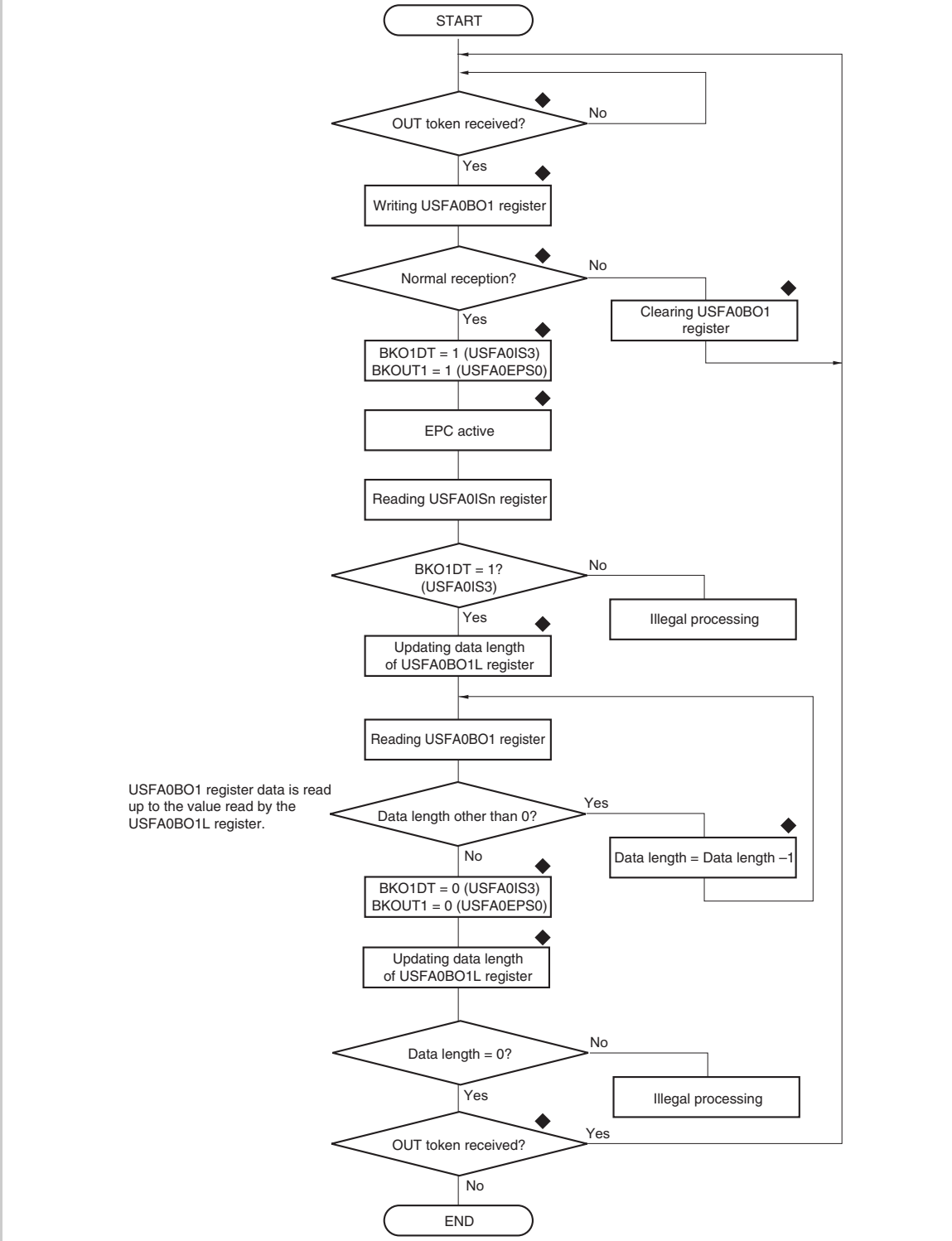


Note ♦: Processing by hardware

Figure 30-31 Parallel processing by hardware

(5) Processing for bulk transfers (OUT)

A bulk transfer (OUT) is assigned to Endpoints2 and 4. The flowchart below shows Endpoint2 is controlled. Endpoint4 can also be controlled using the same sequence. To use this flowchart as the control flow of Endpoint4, therefore, read the bit names of Endpoint2 in the flowchart as those of Endpoint4.

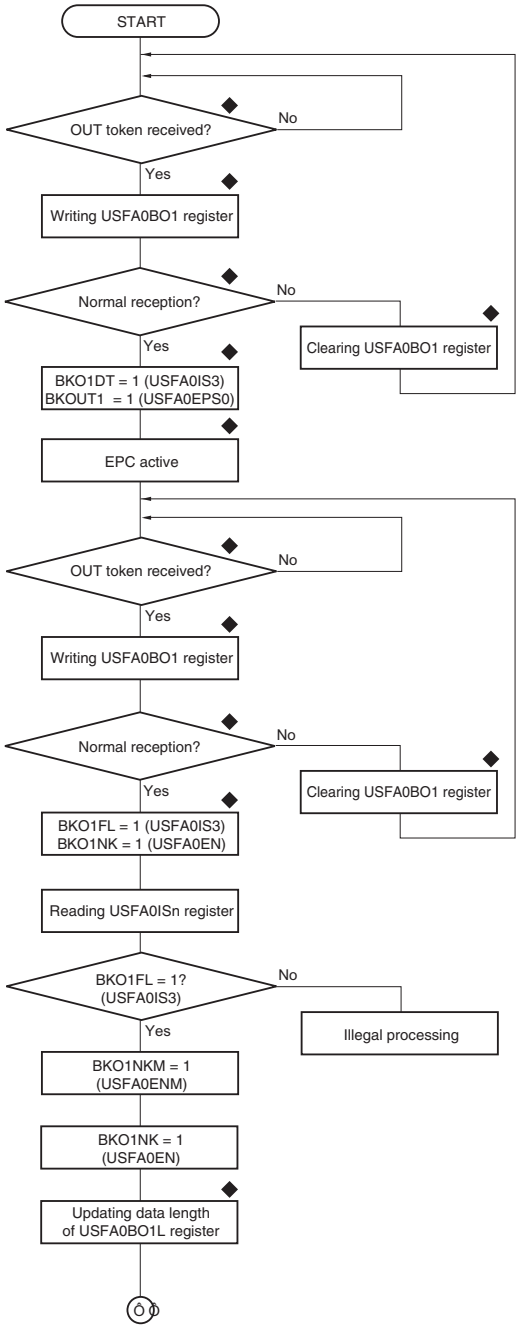


Notes 1. n = 2 or 3

2. ♦: Processing by hardware

Figure 30-32 Normal processing for bulk transfers (OUT) (Endpoint2)

During a bulk transfer (OUT), more data than is expected by the system might be transmitted from the host. To allow for this, endpoints 2 and 4 for bulk transfers (OUT) in this product consist of two 64-byte buffers so data can be read from the CPU side even while the bus side is being accessed, preventing excessive NAK responses and improving the transfer rate of the USB bus. If the host sends more data than is expected by the system, up to 128 bytes of extra data might be received in the worst case. In this case, change the control flow from that of normal processing of endpoints 2 and 4 to the flow shown below when the system is expecting to receive only two more packets of data. This flowchart shows how Endpoint2 is controlled. Endpoint4 can also be controlled using the same sequence. To use this flowchart as the control flow of Endpoint4, therefore, read the bit names of Endpoint2 in the flowchart as those of Endpoint4.



Notes 1. n = 2 or 3

2. ♦: Processing by hardware

Figure 30-33 Processing if more data than is expected by system is transmitted (Endpoint2) (1/2)

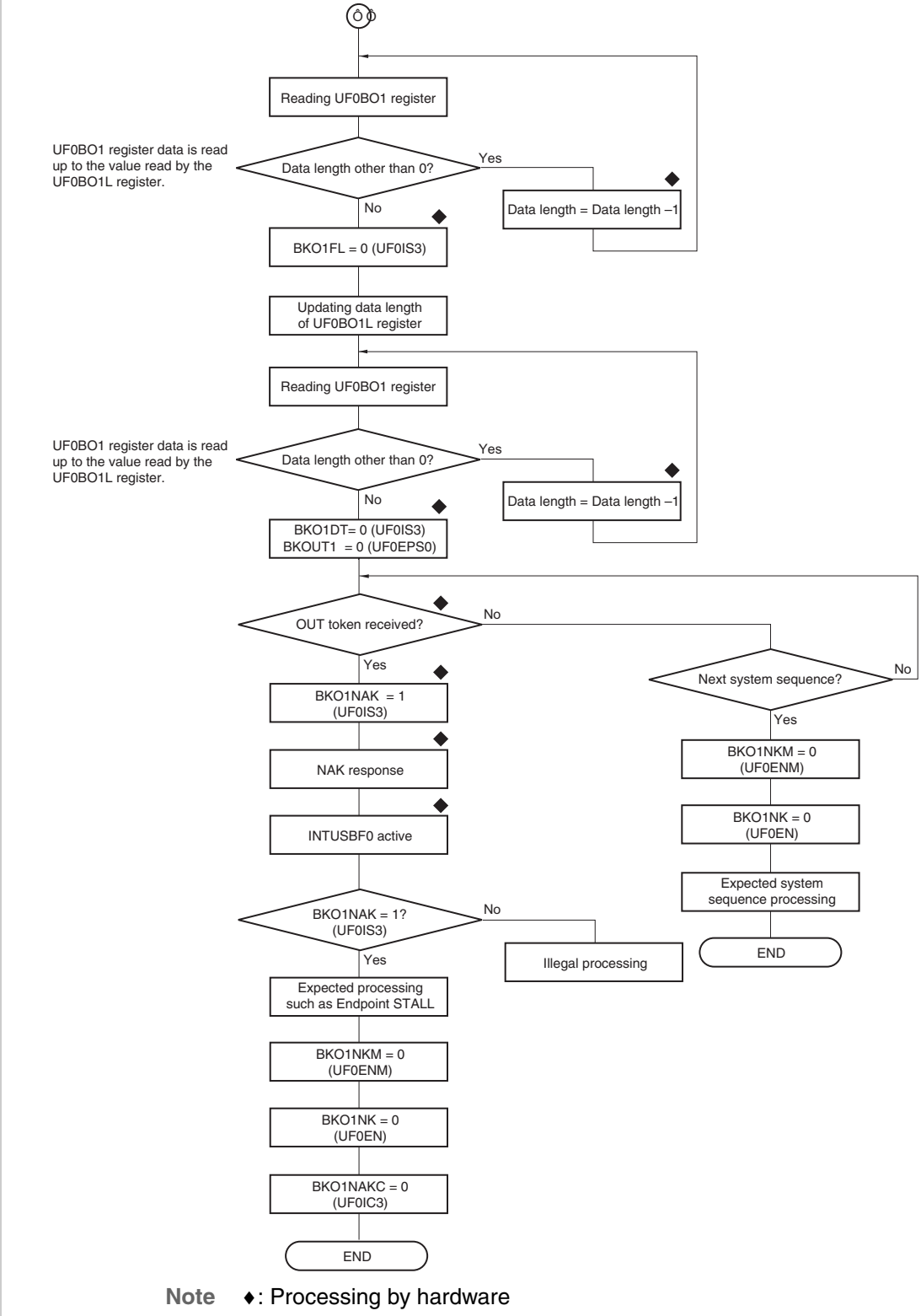
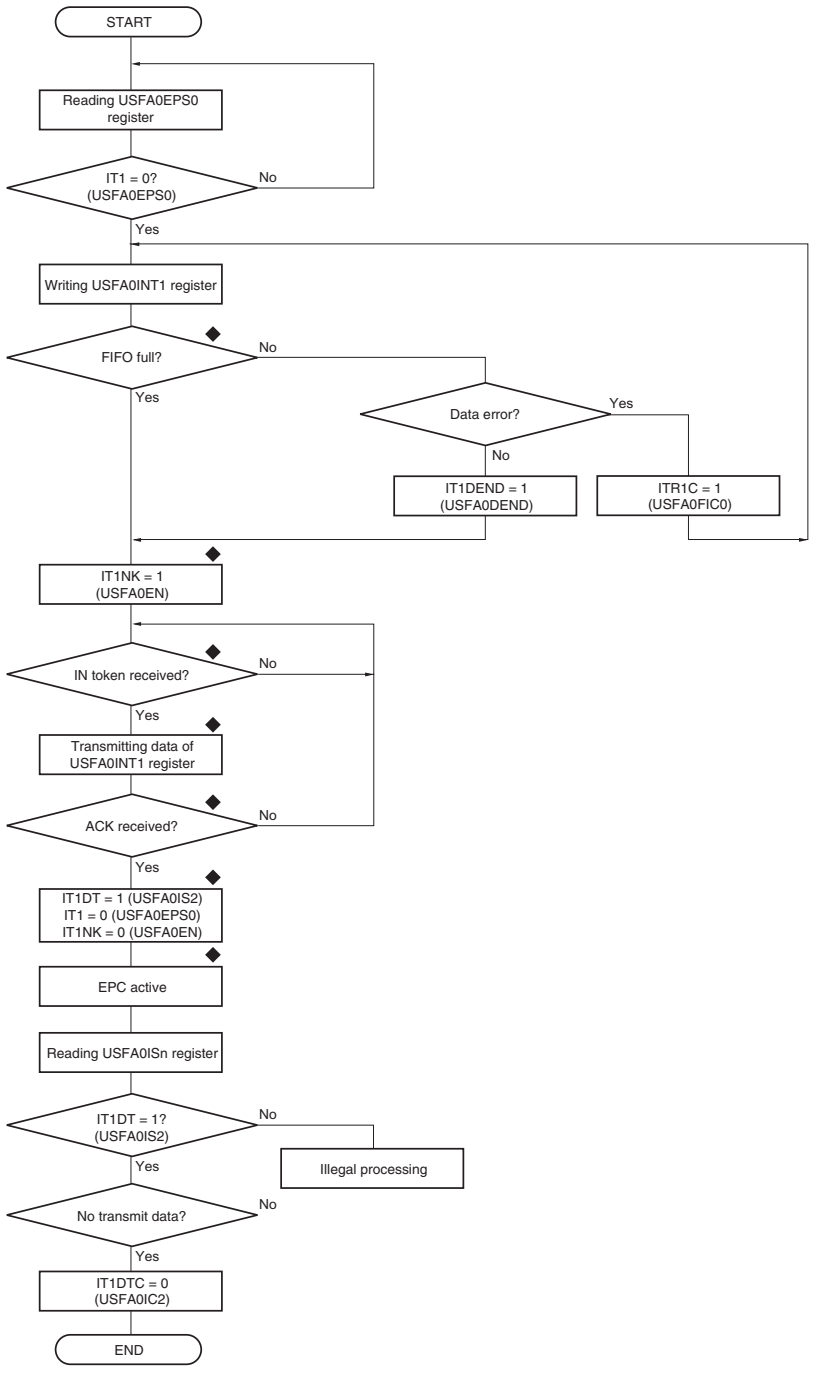


Figure 30-34 Pocessing if more data than is expected by system is transmitted (Endpoint2) (2/2)

(6) Processing for interrupt transfers (IN)

An interrupt transfer (IN) is assigned to Endpoint7. The flowchart in *Figure 30-35 "Processing for interrupt transfers (IN) (Endpoint7)"* shows the related processing.



Notes 1. n = 2 or 3

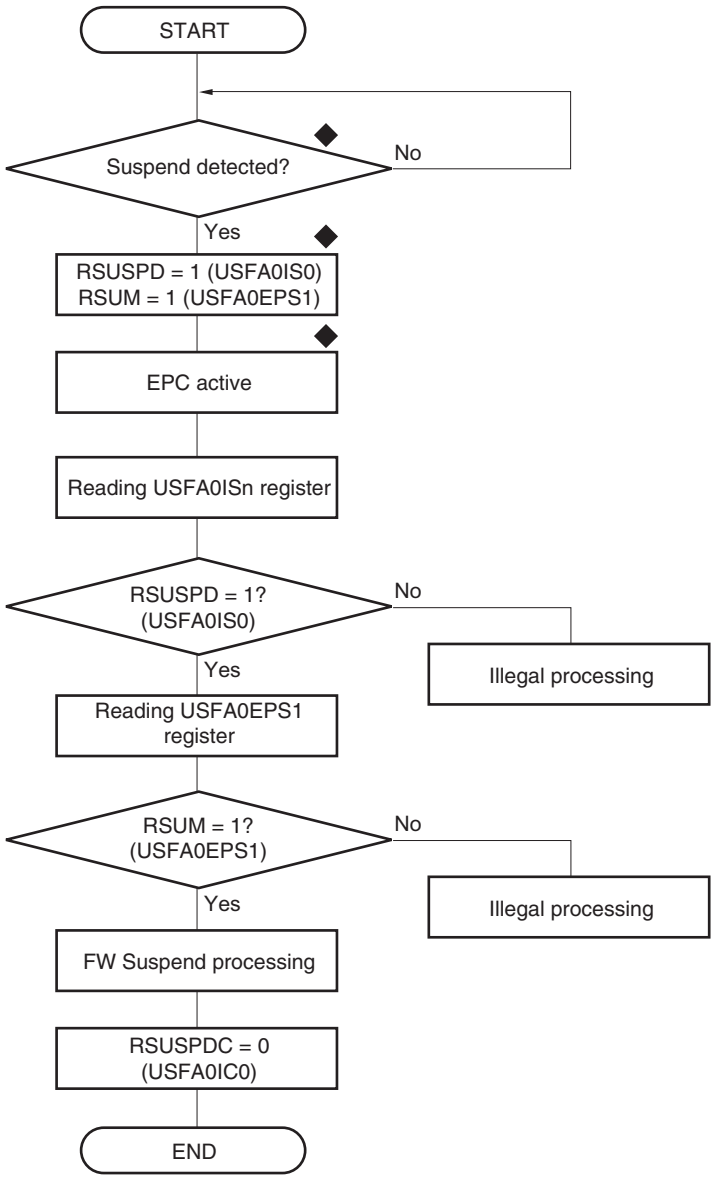
2. ♦: Processing by hardware

Figure 30-35 Processing for interrupt transfers (IN) (Endpoint7)

30.10.4 Suspend/Resume processing

How Suspend/Resume processing is performed depends on the system configuration. One example is given below.

(a) Example of Suspend processing

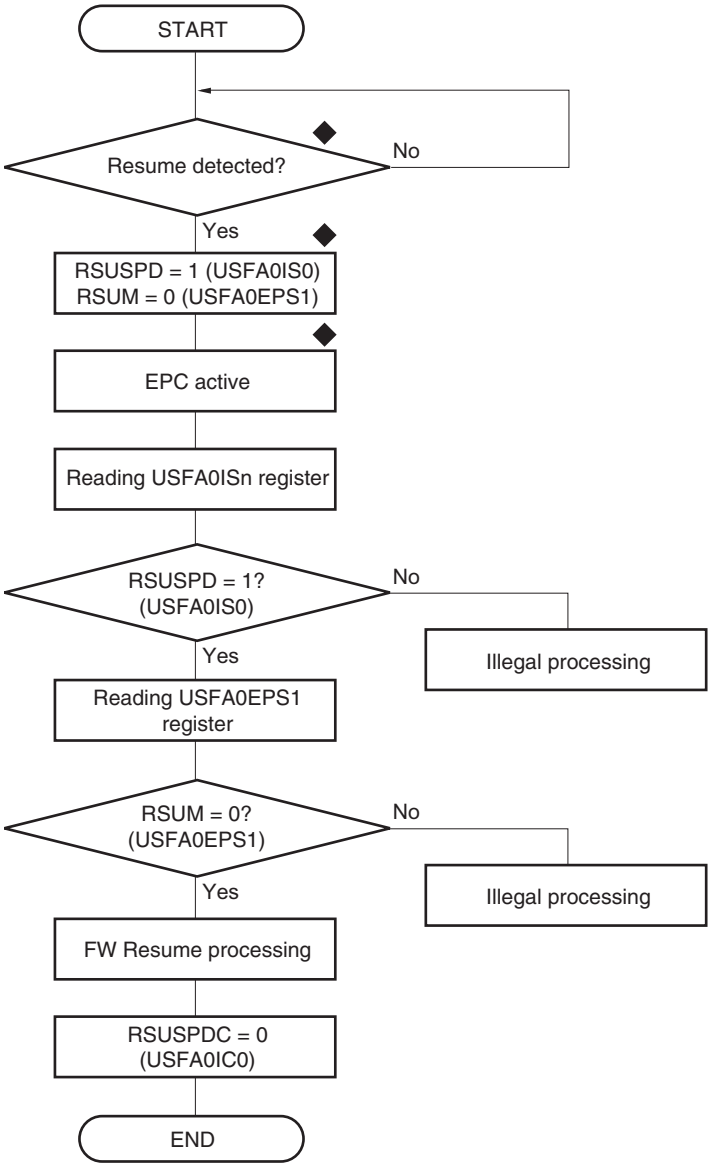


Notes 1. n = 0 or 1

2. ♦: Processing by hardware

Figure 30-36 Example of Suspend/Resume processing (1/3)

(b) Example of Resume processing

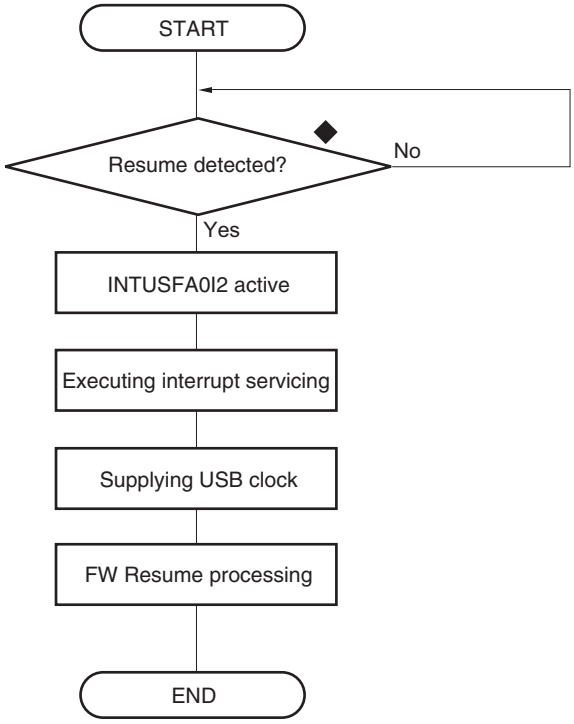


Notes 1. n = 0 or 1

2. ♦: Processing by hardware

Figure 30-37 Example of Suspend/Resume processing (2/3)

(c) Example of Resume processing (when supply of USB clock to USBF is stopped)



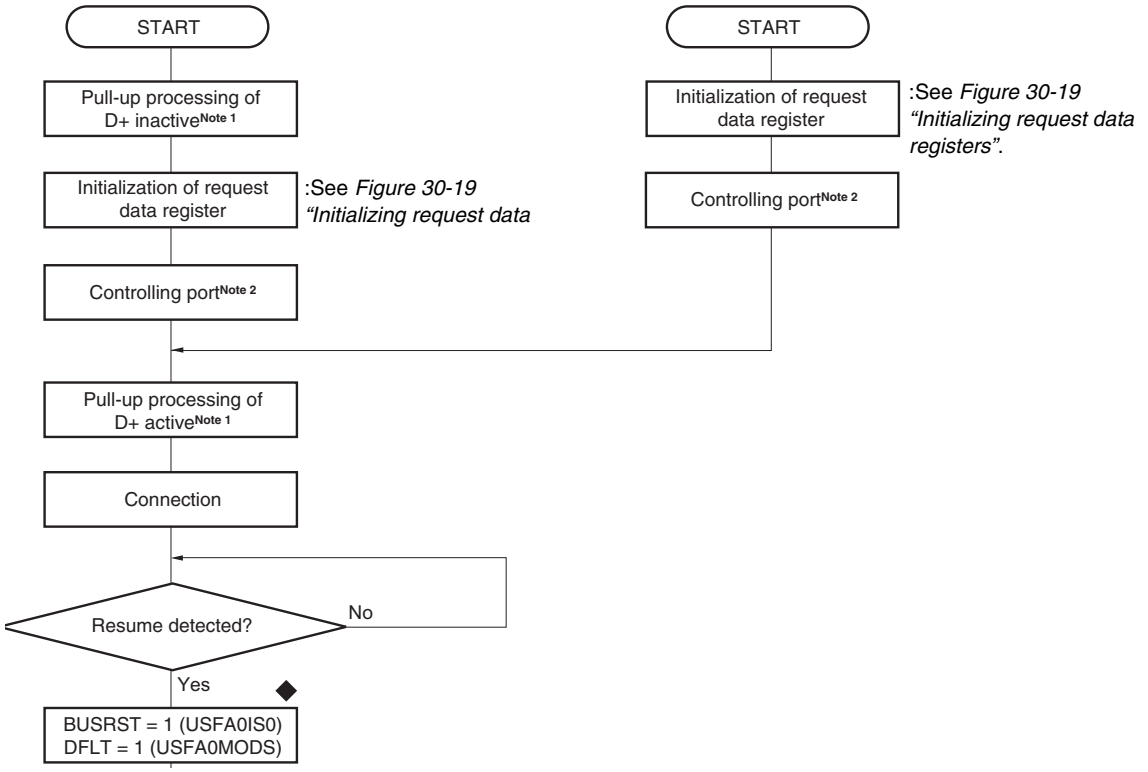
Note ♦: Processing by hardware

Figure 30-38 Example of Suspend/Resume processing (3/3)

30.10.5 Processing after power application

The processing to be performed after power application depends on the system configuration. One example is given below.

(a) Processing after power application (1/2)



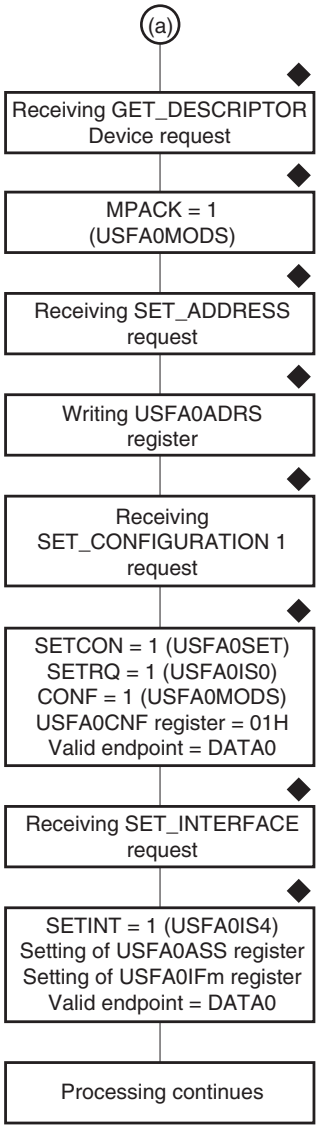
a) Use one general-purpose port pin for the signal that controls switching the pull-up resistor of the USB bus.

b) The input mode or control mode might be selected for the general-purpose port pin assigned in a) by default. Therefore, make sure the active level of the port is high or low when pulling up D+ on power application.

Note ♦: Processing by hardware

Figure 30-39 Example of processing after power application and shutdown (1/3)

(a) Processing after power application (2/2)

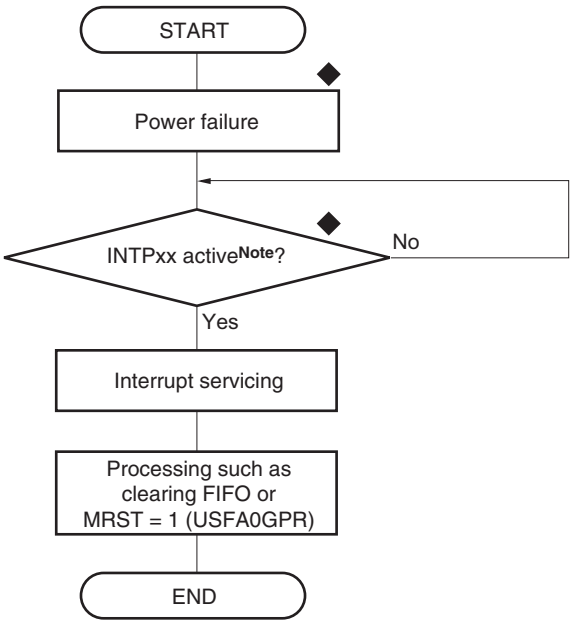


Notes 1. m = 0 to 4

2. ♦: Processing by hardware

Figure 30-40 Example of processing after power application and shutdown (2/3)

(b) Processing on power shutdown



a) INTPxx refers to the external interrupt pins (INTP00 to INTP27) for this product. Assign one external interrupt pin for the following applications.

Detect disconnection of the connector in the case of self-powered mode (SFPW bit of USFA0DST register = 1), The VDD line of the USB connector is monitored, and if disconnection of the connector is detected, a signal is input to the external interrupt pin at the signal edge. Note that the noise elimination time of the interrupt input pin differs depending on the timer.
Detecting turning off the power from the hub when the device is mounted on the same board as a hub chip.

Note ♦: Processing by hardware

Figure 30-41 Example of processing after power application and shutdown (3/3)

Chapter 31 USB Host Controller (USBH)

This microcontroller has an internal USB host controller (USBH) conforming to the Universal Serial Bus Specification. Data communication using the polling method is performed between the USB host controller and external host device by using a token-based protocol.

Caution Use the following procedure to read the USB host controller registers.

- When connecting an external clock to the USBCLK pin

  (1) Set the PCICRST bit of the PCI control register_H (USHA0PCICTRL_H) to 0, and then cancel the reset.

  (2) Set the BMASEN bit of the PCI command register (USHA0PCICMD) to 1 to enable the bus master.

  (3) Read the required USB host controller registers.

- When an external clock is not to be connected to the USBCLK pin

  [1] When supplying an external clock is specified in the SFRCTL3 register (When the USBDIV1 and USBDIV0 bits of SFRCTL3 are 00b)

  (1) Set the USBDIV1 and USBDIV0 bits of the SFRCTL3 register to 11b to specify an internal clock of fxx/4.

  (2) Set the PCICRST bit of the PCI control register-H (USHA0PCICTRL_H) to 0 to cancel the reset.

  (3) Set the BMASEN bit of the PCI command register (USHA0PCICMD) to 1 to enable the bus master.

  (4) Read the required USB host controller registers.

  [2] When supplying an internal clock is specified in the SFRCTL3 register (When the USBDIV1 and USBDIV0 bits of SFRCTL3 are 10b or 11b)

  (1) Set the PCICRST bit of the PCI control register-H (USHA0PCICTRL_H) to 0 to cancel the reset.

  (2) Set the BMASEN bit of the PCI command register (USHA0PCICMD) to 1 to enable the bus master

  (3) Read the required USB host controller registers.

31.1 V850E2/MN4 USBH Features

Instances

This microcontroller has following number of instances of the USB host controller.

Table 31-1 Instances of USBH

USB host controller	V850E2/MN4
Instance	1

Register addresses

All USBH register addresses are given as addresses offset from the individual base address <USHA0_base_OHCI> or <USHA0_base_PCI> for USBH. The <USHA0_base_OHCI> and <USHA0_base_PCI> addresses of USBH are given in the following table:

Table 31-2 Register base addresses <USAH0_base>

USBH registers	USBH features	<USHA0_base> addresses
OHCI operational registers	USAH0_base_OHCI	F993 0000_H
PCI bridge registers	USAH0_base_PCI	F993 0800_H

Clock supply

The following clocks can be input to USBH:

Table 31-3 USBH clock supply

Clock	Internal/ external	Description
USBH clock	Internal	{9.6 MHz external clock × internal clock multiplied by 20}/4 = 48 MHz internal clock
		{7.2 MHz external clock × internal clock multiplied by 20}/3 = 48 MHz internal clock
	External	External clock input from the USBCLK pin (f_{USB} = 48 MHz)
PCI clock	Internal	The frequency from 25 to 33.33 MHz can be specified by using the SFRCTL3.PCIDIV[2:0] bits.

Interrupts

USBH can generate the following interrupt requests:

Table 31-4 USBH interrupt requests

USBH signals	Description
INTUSHA0I0	Bridge error interrupt
INTUSHA0I1	USBH core interrupt
INTUSHA0PME	USBH wakeup interrupt

I/O signals

The USBH I/O signals are given in the following table:

Table 31-5 USBH I/O signals

USBH signals	Description	Connected to
UDMH	USBH data I/O (–)	Port UDMH
UDPH	USBH data I/O (+)	Port UDPH
$\overline{OCI}$	Overcurrent detection input	Port $\overline{OCI}$
PPON	USB power supply output	Port PPON

Host controller communication area (HCCA)

The following emmory can be used as shared memory called the *host controller communication area (HCCA)*:

- H-bus shared memory
- External memory connected to the secondary memory controller

31.1.1 USB host controller memory map

The registers of the USB host controller are allocated to the following memory spaces:

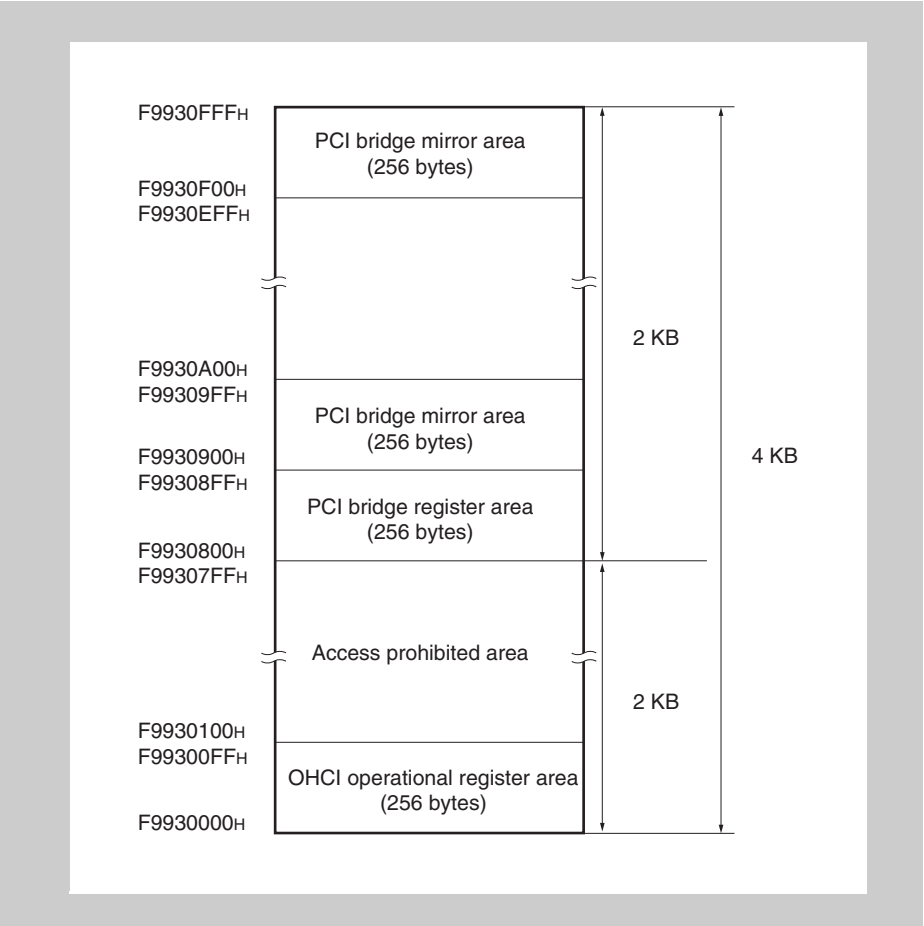


Figure 31-1 USB host controller memory map

31.1.2 V850E2/MN4 USB control register

Before using the USB host controller, set up the register below:

(1) USBH buffer control register (USHBC)

The USHBC register enables and disables the USB function controller input buffers and controls the input buffer floating state.

Access This register can be read or written in 32-bit units.

Address F990 1000_H

Initial value 0000 0000_H. This register is initialized by any reset.

	31	30	29	28	27	26	25	24
USHBC	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R
	23	22	21	20	19	18	17	16
	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R
	15	14	13	12	11	10	9	8
	0	0	0	0	0	0	0	0
	R	R	R	R	R	R	R	R
	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	UBHIEN	UBHIOR
	R	R	R	R	R	R	R/W	R/W

Table 31-6 USHBC register contents

Bit position	Bit name	Function
1	UBHIEN	Specifies whether to enable the USB buffers. 0: Disable buffers. 1: Enable buffers. Caution: When not using the USB, clear the UBHIEN bit to 0.
0	UBHIOR	Specifies whether to enable the use of a measure to prevent the USB buffer inputs from floating. 0: Disable the use of a floating prevention measure. 1: Enable the use of a floating prevention measure. When no cable is connected (data input is in the floating state), it is necessary to take measures to prevent a signal that has an undefined value from being recognized as a bus reset, suspend, or resume signal. By setting this bit to 1, floating can be prevented by using a signal such as VBUS (for recognizing cable connection).

31.2 Overview

- Conforms to the Universal Serial Bus Specification Revision xx
- Supports 12 Mbps (full-speed) transfer
- Supports the open host controller interface (OHCI) 1.0a (with a restriction on changing USB ports from disabled to enabled)
- Supports control transfer, bulk transfer, interrupt transfer, and isochronous transfer[a]
- Has a built-in root hub function and is equipped with 1 downstream port channel

a) Isochronous transfer puts a heavy load on the system performance, so be sure to sufficiently evaluate performance with the target system in advance.

31.3 Configuration

31.3.1 Block diagram

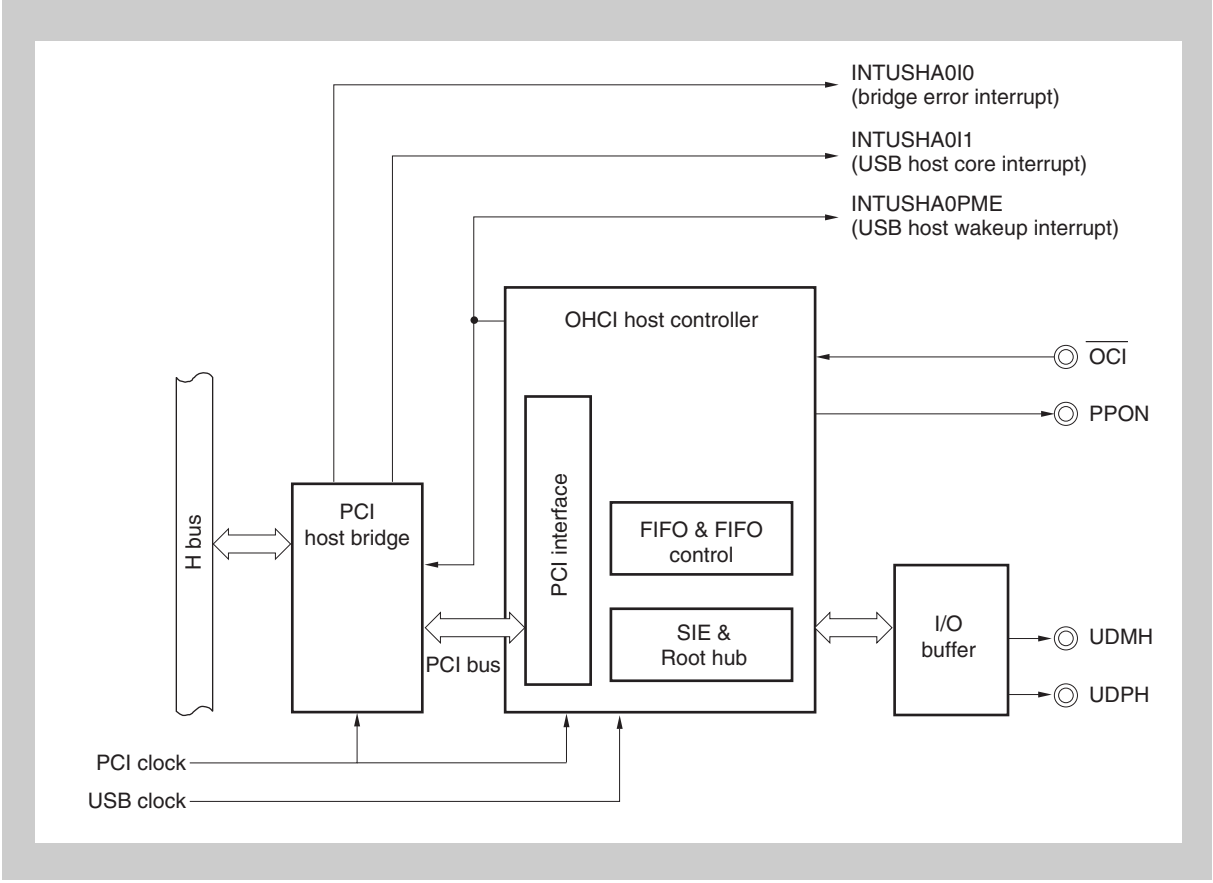


Figure 31-2 Block diagram of USB host controller

31.4 External circuit configuration

31.4.1 Overview

In USB transmission, when communication is performed with the host controller and function controller facing each other, pull-up/pull-down resistors must be connected to the USB signal pins (D+ and D–) to identify the communication partner. Moreover, in this microcontroller, series resistors must also be connected.

Because this microcontroller does not include pull-up, pull-down, or series resistors, be sure to connect them externally.

Figure 31-3 "Schematic configuration of pull-up, pull-down, and series resistors in USB transmission line" shows the outline configuration of the USB transmission line. For details about the external circuit configuration, see *31.4.2 "Connection configuration"*.

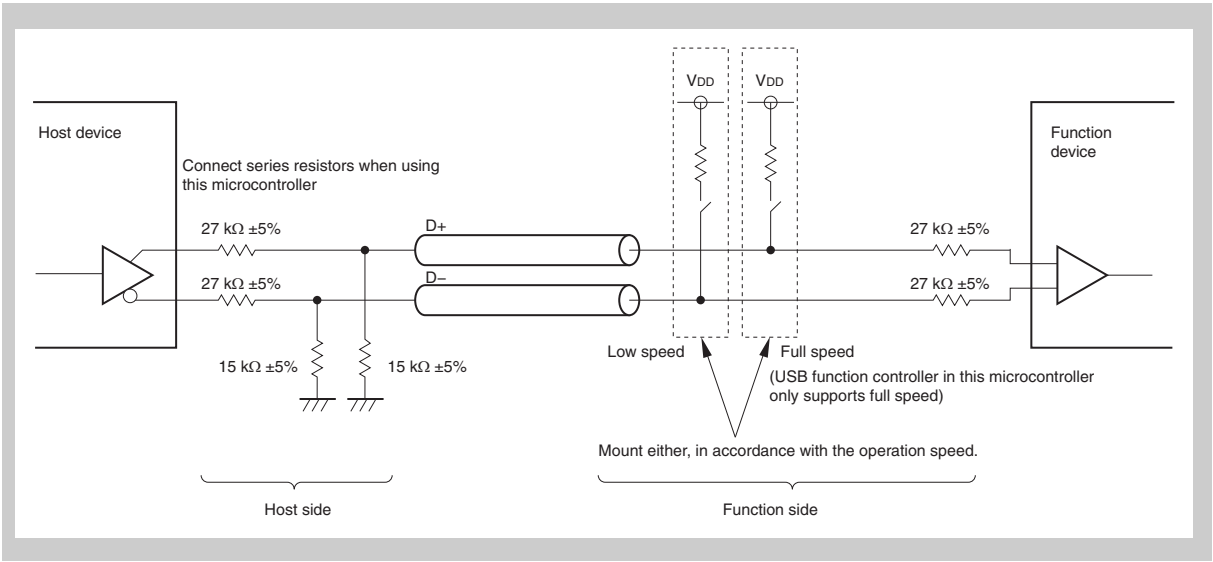


Figure 31-3 Schematic configuration of pull-up, pull-down, and series resistors in USB transmission line

31.4.2 Connection configuration

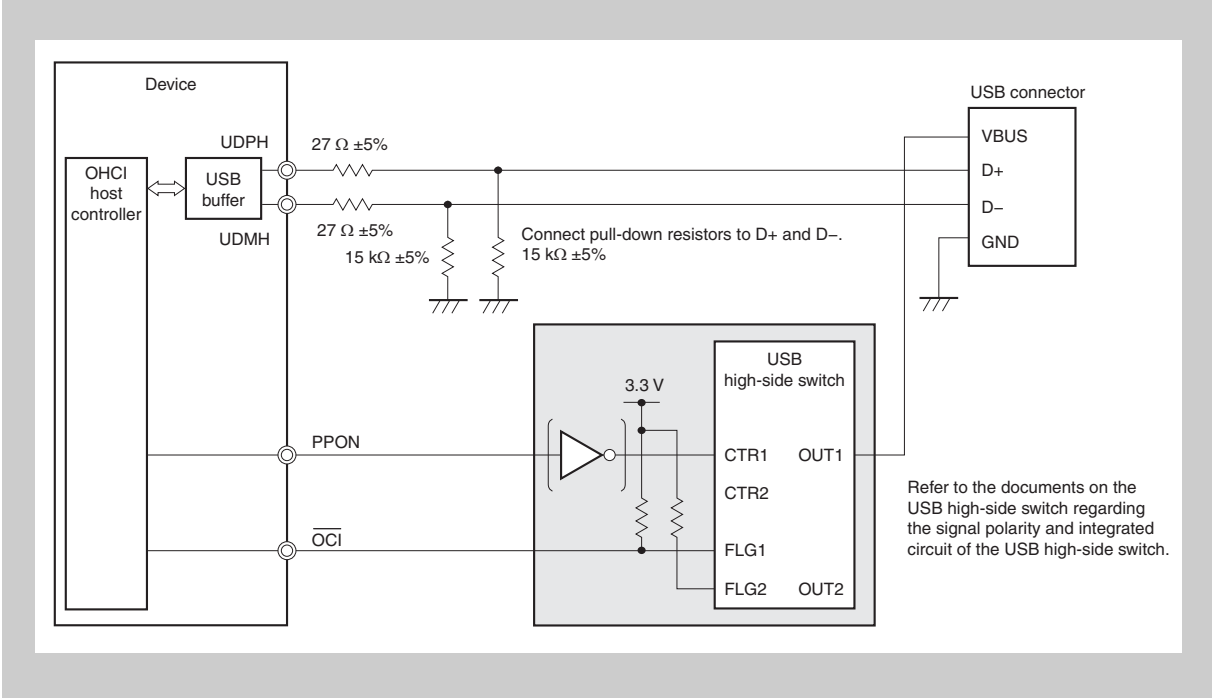


Figure 31-4 Example of USB host controller connection

(1) Series resistor connection to D+ and D–

Connect series resistors of 27 Ω ±5% to the D+ and D– pins (UDPH, UDMH) of the USB host controller in this microcontroller. If they are not connected, the impedance specification cannot be satisfied and the output waveform might be disturbed.

Allocate the series resistors adjacent to this microcontroller as much as possible, and make the length of the wiring between the series resistors and the USB connectors the same, to make the impedance of D+ and D– equal (a differential with 90 Ω ±5% is recommended).

(2) Pull-down resistor connection to D+ and D–

Be sure to pull down the D+ and D– pins (UDPH and UDMH) to GND using a resistance of 15 kΩ ±5%.

In this case, the configuration is the same as when no function device is connected.

31.4.3 USB power supply

(1) Overcurrent detection and power control

This microcontroller does not have a circuit for detecting overcurrent at USB ports or controlling the USB port power supply. To support these functions in the system, use an external circuit such as a USB high-side switch to configure these functions and connect it with the $\overline{\text{OCI}}$ and PPON pins.

The following table shows the operations related to the $\overline{\text{OCI}}$ and PPON pins used for controlling the external circuit for detecting overcurrent on USB ports or controlling the USB port power.

Table 31-7 OCI and PPON Signals

Pin	I/O	Level	Description
$\overline{\text{OCI}}$	Input	1	Overcurrent not detected
		0	Overcurrent detected
PPON	Output	1	Power supply to VBUS is on
		0	Power supply to VBUS is off

Figure 31-4 "Example of USB host controller connection" shows an example of connecting a circuit to detect overcurrent and control the power (VBUS). When the USB bus is not used, power consumption can be reduced by stopping VBUS supply to the port, though its availability depends on the connection to the USB connector. If the connected USB function device is bus-powered, using a high-side switch as shown in the connection example is recommended.

(2) VBUS control specification according to root hub register settings

Control of the PPS bit for controlling VBUS varies depending on the settings of the root hub control registers HcRhDescriptorA and HcRhDescriptorB.

To control VBUS for each port, the HcRhDescriptorA.NPS and HcRhDescriptorA.PSM bits must be set to 1.

31.5 Cautions

(1) Clock accuracy

To operate the USB host controller, a 48 MHz internal clock ({9.6 MHz external clock × internal clock multiplied by 20}/4 or {7.2 MHz external clock × internal clock multiplied by 20}/3) or a 48 MHz external clock (f_{USB}) (input to the USBCLK pin) must be used as the USB clock. When the external clock is used as the USB clock, supply a clock with an accuracy of 48 MHz ±500 ppm max. (target) to the USBCLK pin. If the USB clock accuracy drops, the transmission data cannot satisfy the USB specifications.
The clock accuracy defined in the USB specifications is not guaranteed if an internal clock is used. Make sure to use an external clock to satisfy the USB specifications.

(2) Hub connection

Use of the external clock (crystal resonator) is recommended when connecting a device via the hub. When the internal clock is used, the USB specifications might not be satisfied because of the effects of clock jitter in the microcontroller's internal circuits. One-to-one connection with the host device is recommended when the internal clock is used.

31.6 PCI host bridge

31.6.1 PCI bridge registers

The table below lists the PCI bridge registers. For details about the base address, see *Table 31-2 "Register base addresses <USAH0_base>"*.

Address	Register name	Symbol	R/W	Manipulatable bit unit		Initial value
				16	32	
<USHA0_base_PCI> + 04_H	PCI command register	USHA0PCICMD	R/W	✓		0006_H
<USHA0_base_PCI> + 06_H	PCI status register	USHA0PCISTS	R/W	✓		0280_H
<USHA0_base_PCI> + 10_H	PCI base address register 0	USHA0BAR0	R/W		✓	$0000\ 0000_H$
<USHA0_base_PCI> + $A0_H$	PCI address control register 0	USHA0ACR0	R/W		✓	$0000\ 0000_H$
<USHA0_base_PCI> + $C0_H$	Error register 1	USHA0ERR1	R/W		✓	$0000\ 0000_H$
<USHA0_base_PCI> + CC_H	Read burst type register	USHA0RBTYPE	R/W		✓	$0500\ 0000_H$
<USHA0_base_PCI> + $E4_H$	PCI control register	USHA0PCICTRL_H	R/W		✓	$0000\ 0000_H$
<USHA0_base_PCI> + EC_H	PCI BAR enable register	USHA0PCIBARE	R/W		✓	$0000\ DFFF_H$
<USHA0_base_PCI> + $F8_H$	PCI configuration address register	USHA0CNFIGADDR	R/W		✓	$0000\ 0000_H$
<USHA0_base_PCI> + FC_H	PCI configuration data register	USHA0CNFIGDATA	R/W		✓	$0000\ 0000_H$

(1) PCI command register (USHA0PCICMD)

The USHA0PCICMD register is used to specify the operating conditions of USBH on the PCI bus.

Access This register can be read or written in 16-bit units.

Address <USHA0_base_PCI> + 04_H

Initial value 0006_H. This register is initialized by any reset.

Caution Be sure to clear bits 15 to 3 and 0 to “0”.

15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
0	0	0	0	0	BMASEN	MEMEN	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 31-8 USHA0PCICMD register contents

Bit position	Bit name	Description
2	BMASEN	BUS Master Enable Specifies whether USBH operates as the PCI bus master. 0: USBH operates as the PCI bus master. 1: USBH does not operate as the PCI bus master.
1	MEMEN	Memory Space Enable Specifies whether USBH can access the memory space. 0: USBH cannot access the memory space. 1: USBH can access the memory space.

(2) PCI status register (USHA0PCISTS)

The USHA0PCISTS register is the PCI status register.

Access This register can be read or written in 16-bit units.

Address <USHA0_base_PCI> + 06_H

Initial value 0280_H. This register is initialized by any reset.

Caution Be sure to clear bits 15, 14, 11 to 0 to “0”.

15	14	13	12	11	10	9	8
0	0	RMA	RTA	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	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

Table 31-9 USHA0PCISTS register contents

Bit position	Bit name	Description
13	RMA	Received Master-Abort This bit indicates whether master abort is received as the PCI master. This bit is cleared if 1 is written to. Writing 0 to this bit is ignored. 1: Master abort is received. 0: Master abort is cleared (default).
12	RTA	Received Target-Abort This bit indicates whether target abort is received as the PCI master. This bit is cleared if 1 is written to. Writing 0 to this bit is ignored. 1: Target abort is received. 0: Target abort is cleared (default).

(3) PCI base address register 0 (USHA0BAR0)

The USHA0BAR0 register is used to set up address window 0 when USBH operates as the PCI slave.

Access This register can be read or written in 32-bit units.

Address <USHA0_base_PCI> + 10_H

Initial value $0000\ 0000_H$. This register is initialized by any reset.

Notes

1. Before specifying a value for the USHA0BAR0 register, specify the address space size by using PCI address control register 0 (USHA0ACR0).
2. This register is cleared to 0 after a reset, so the value must be specified again when next using this register.
3. If no space is used, set the USHA0ACR0.BARMASK bits to 0000b.
4. If the USHA0BAR0 register setting is changed while the USB host controller is accessing the specified memory space via the PCI bus, the operation is undefined. Set up the USHA0BAR0 register during system initialization.

31	30	29	28	27	26	25	24
BASEADD31	BASEADD30	BASEADD29	BASEADD28	BASEADD27	BASEADD26	BASEADD25	BASEADD24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
BASEADD23	BASEADD22	BASEADD21	BASEADD20	BASEADD19	BASEADD18	BASEADD17	BASEADD16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
BASEADD15	BASEADD14	BASEADD13	BASEADD12	BASEADD11	BASEADD10	BASEADD9	BASEADD8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
BASEADD7	BASEADD6	BASEADD5	BASEADD4	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 31-10 USHA0BAR0 register contents

Bit position	Bit name	Function
31:4	BASEADD [31:4]	Specify the PCI slave start address (reset value: 0000000_H).

(4) PCI address control register 0 (USHA0ACR0)

The USHA0ACR0 register is used to specify the size of the address space starting at the address specified by the USHA0BAR0 register and the address to be translated and passed to the AHB side.

Specify 64 KB as the address space size. If no space is used, set the BARMASK bits to 0000_b.

Access This register can be read or written in 32-bit units.

Address <USHA0_base_PCI> + $A0_H$

Initial value $0000\ 0000_H$. This register is initialized by any reset.

Notes

1. Set the USHA0ACR0 register to $0000\ 0100_H$.
2. Before specifying a value for the USHA0BAR0 register, specify the address space size by using the USHA0ACR0 register.
3. If the USHA0ACR0 register setting is changed while the USB host controller is accessing the specified memory space via the PCI bus, the operation is undefined. Set up the USHA0ACR0 register during system initialization.

31	30	29	28	27	26	25	24
0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	BARMASK4
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
BARMASK3	BARMASK2	BARMASK1	BARMASK0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 31-11 USHA0ACR0 register contents

Bit position	Bit name	Function
8:4	BARMASK [4:0]	Base Address Mask Specify the valid size of address window 0. Specify 10000_B (64 KB).

(5) Error register 1 (USHA0ERR1)

This bit indicates the error generation status when an error response is returned to the USB host controller from the H-bus shared memory. This bit also possesses a control signal for asserting a bridge error interrupt (INTUSHA0I0) when an error is generated.

Access This register can be read or written in 32-bit units.

Address <USHA0_base_PCI> + $C0_H$

Initial value $0000\ 0000_H$. This register is initialized by any reset.

Caution Be sure to clear bits 31 to 10, 8 to 6, and 4 to0 to “0”.

31	30	29	28	27	26	25	24
0	0	0	0	0	1	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
0	0	0	0	0	0	AMEn	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
0	0	AMEr	0	0	0	0	0/1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 31-12 USHA0ERR1 register contents

Bit position	Bit name	Function
9	AMEn	This bit enables notification using the bridge error interrupt (INTUSHA0I0) when the H-bus master unit in the PCI host bridge acknowledges an error response (when the H-bus shared memory returns an error response). 0: Do not assert the bridge error interrupt (INTUSHA0I0), even if the H-bus master unit in the PCI host bridge acknowledges an error response. 1: Assert the bridge error interrupt (INTUSHA0I0) when the H-bus master unit in the PCI host bridge acknowledges an error response.
5	AMEr	This bit is asserted when the H-bus master unit in the PCI host bridge acknowledges an error response. The bit is cleared to 0 when 1 is written to the AMEr bit. 0: The H-bus master unit in the PCI host bridge has not acknowledged an error response. 1: The H-bus master unit in the PCI host bridge has acknowledged an error response.

(6) Read burst type register (USHA0RBTYPE)

The USHA0RBTYPE register is used to specify the format of the access burst to be issued to the external bus during read transfers. Set up the USHA0RBTYPE register during initialization and do not change the setting while executing a transfer.

Access This register can be read or written in 32-bit units.

Address <USHA0_base_PCI> + CC_H

Initial value $0500\ 0000_H$. This register is initialized by any reset.

Caution Be sure to specify $0500\ 0001_H$.

31	30	29	28	27	26	25	24
0	0	0	0	0	1	0	1
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	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

(7) PCI control register 1 (USHA0PCICTL_H)

The USHA0PCICTRL_H register is used to specify the operating conditions of the PCI bus.

Access This register can be read or written in 32-bit units.

Address <USHA0_base_PCI> + $E4_H$

Initial value $0000\ 0000_H$. This register is initialized by any reset.

Cautions

1. If the PCI master on the OHCI host controller is reset when the PCI host bridge sets the read data (data written from the CPU) by using a read request from the PCI master, accessing the PCI host bridge is not retried and the PCI bus might lock up. Therefore, if you are changing the settings of PCICRST from 0 to 1, you must first confirm that access to the PCI unit on the OHCI host controller has finished and stop access to the PCI unit on the OHCI host controller, or, change PCICRST by using the configuration register via the PCI in order to prevent the PCI host bridge from being reset during read data processing. In addition, when writing to the PCI unit on the OHCI host controller, use the PCI configuration address register (USHA0CNFIGADDR) and the PCI configuration data register (USHA0CNFIGDATA) to issue a configuration access to the PCI unit on the OHCI host controller. After writing to the PCICRST bit, read the register to confirm that the PCICRST bit has been set. Even if 1 is written to the PCICRST bit, the PCI host bridge is not reset.
2. Be sure to clear bits 30, 29, 27 to 0 to “0”.

31	30	29	28	27	26	25	24
PCICRST	0	0	CNFG DONE	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	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

Table 31-13 USHA0PCICTL_H register contents

Bit position	Bit name	Function
31	PCICRST	This bit controls the PCI interface on the OHCI host controller. Setting 1 resets the PCI interface on the OHCI host controller.
28	CNFGDONE	Specify the response for the access from the PCI on the OHCI host controller. 0: Issue a retry continuously for all accesses as the PCI on the OHCI host controller. 1: Acknowledge access as the PCI on the OHCI host controller.

(8) PCI BAR enable register (USHA0PCIBARE)

The USHA0PCIBARE register is used to enable or disable a PCI window. This register can be used to disable the window specified by using PCI address control register 0 (USHA0ACR0). If the window is disabled, even if an attempt is made to access the specified USHA0BAR0 space, the access is ignored .

Access This register can be read or written in 32-bit units.

Address <USHA0_base_PCI> + EC_H

Initial value 0000 $DFFF_H$. This register is initialized by any reset.

Notes

1. Be sure to clear bits 31 to 16 to ”0”. Although the initial value of bits 11 to 1 is 1, be sure to clear these bits to 0.
2. The USHA0PCIBARE register cannot be used to enable a USHA0BAR0 space disabled by using the USHA0ACR0 register.
3. If the USHA0PCIBARE register setting is changed while the USB host controller is accessing the specified memory space via the PCI bus, the operation is undefined. Set up the USHA0PCIBARE register during system initialization.
4. Be sure to set bits 15 to 12 to “1101”.
5. Be sure to clear bits 11 to 1 to “0”.

31	30	29	28	27	26	25	24
0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
1	1	0	1	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	BAR0En
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 31-14 USHA0PCIBARE register contents

Bit position	Bit name	Function
0	BAR0En	USHA0BAR0 Enable Specifies whether to enable window 1 specified by using the USHA0BAR0 register. Be sure to set this bit to 1 to use the window. 0: Disable 1: Enable

(9) PCI configuration address register (USHA0CNFIGADDR)

The USHA0CNFIGADDR register is used to specify the address used to access the configuration of the PCI slaves on the OHCI host controller.

Set the CNFEN bit of the USHA0CNFIGADDR register to 1, and then access the USHA0CNFIGDATA register to perform a PCI configuration space access.

The USHA0CNFIGADDR and USHA0CNFIGDATA registers can only be accessed by using the AHB bus. Accessing these registers by using the PCI bus is prohibited.

Access This register can be read or written in 32-bit units.

Address <USHA0_base_PCI> + $F8_H$

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
CNFEN	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
BUSNUM7	BUSNUM6	BUSNUM5	BUSNUM4	BUSNUM3	BUSNUM2	BUSNUM1	BUSNUM0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
DEVNUM4	DEVNUM3	DEVNUM2	DEVNUM1	DEVNUM0	FNCNUM2	FNCNUM1	FNCNUM0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
REGNUM5	REGNUM4	REGNUM3	REGNUM2	REGNUM1	REGNUM0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 31-15 USHA0CNFIGADDR register contents

Bit position	Bit name	Function
31	CNFEN	Config Enable Specifies whether to enable a PCI configuration space access. Before attempting a configuration space access by using the USHA0CNFIGDATA register, set this bit to 1. 0: Disable a configuration space access to a PCI device even if the USHA0CNFIGDATA register is accessed. 1: Enable a configuration space access to a PCI device if the USHA0CNFIGDATA register is accessed.
23:16	BUSNUM [23:16]	Bus Number Specify the number of the bus used for configuration space access. Set these bits to 0000 0000b. Type 0 configuration space access is specified.
15:11	DEVNUM [4:0]	Device Number Specify the device number. 00000: OHCI host controller configuration 00001: PCI host bridge internal registers Other than the above: Setting prohibited Accessing an PCI host bridge internal register by using these bits is required only when executing a PCI reset by using the PCICRST bit of the USHA0PCICTRL-H register.
10:8	FNCNUM [2:0]	Function Number Specify the function number. Set these bits to 000.
7:2	REGNUM [5:0]	Register Number Specify the OHCI configuration register number.

(10) PCI configuration data register (USHA0CNFIGDATA)

The USHA0CNFIGDATA register stores data used to access the configuration space of an PCI slaveon the OHCI host controller..

Set the CNFEN bit of the USHA0CNFIGADDR register to 1, and then access the USHA0CNFIGDATA register to perform a PCI configuration space access.

When writing data to the configuration space of a PCI device, the data stored in the USHA0CNFIGDATA register is written. When reading data from the configuration space of a PCI device, the data read from the corresponding external PCI configuration register is stored in USHA0CNFIGDATA register.

The USHA0CNFIGADDR and USHA0CNFIGDATA registers can only be accessed by using the AHB bus. Accessing these registers by using the PCI bus is prohibited.

Access This register can be read or written in 32-bit units.

Address <USHA0_base_PCI> + FC_H

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
CNFDATA31	CNFDATA30	CNFDATA29	CNFDATA28	CNFDATA27	CNFDATA26	CNFDATA25	CNFDATA24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
CNFDATA23	CNFDATA22	CNFDATA21	CNFDATA20	CNFDATA19	CNFDATA18	CNFDATA17	CNFDATA16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
CNFDATA15	CNFDATA14	CNFDATA13	CNFDATA12	CNFDATA11	CNFDATA10	CNFDATA9	CNFDATA8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
CNFDATA7	CNFDATA6	CNFDATA5	CNFDATA4	CNFDATA3	CNFDATA2	CNFDATA1	CNFDATA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 31-16 USHA0CNFIGDATA register contents

Bit position	Bit name	Function
31:0	CNFDATA [31:0]	Config Data These bits are used to store the data read from or written to the configuration space of a PCI device.

31.7 OHCI host controller

31.7.1 OHCI host controller features

The OHCI host controller has the following features:

- Conforms to OpenHCI Specification Release 1.0a
- Conforms to Universal Serial Bus Specification Revision 1.1

  Supports transfer at full-speed (12 Mbps)
- Incorporates 1 channel root hub and supports one downstream port
- USB clock: 48 MHz, PCI clock: 25 to 33 MHz
- Memory space
  - 256-byte PCI memory area (OHCI operational registers) allocated
  - Host controller communication area (HCCA) allocated
- Communication with CPU

  Performs communication via operational registers in the OHCI host controller and the host controller communication area (HCCA).

Two communication channels are connected between the CPU and the OHCI host controller. The communication channel consists of OHCI operational registers, and the OHCI host controller serves as a target (slave) for this communication. The Base Address register (10_H) in the OHCI host configuration registers functions as a pointer to the OHCI operational registers.

The second communication channel is the host controller communication area (HCCA) in the OHCI operational registers, which functions as a pointer to the shared memory. The OHCI host controller serves as a master for this communication.

Descriptor information used for communication is managed by the OHCI operational registers and the HCCA area.

31.7.2 OHCI host configuration registers

The OHCI host configuration registers configure a 256-byte register space, and are incorporated in the OHCI host controller. These registers are accessed from the CPU system via PCI host bridge registers (USHA0CNFIGDATA and USHA0CNFIGADDR). For details about the base address, see *31.1.1 "USB host controller memory map"*.

Table 31-17 OHCI host configuration registers

Bit / Address	31 to 24	23 to 16	15 to 8	7 to 0
00_H	Reserved		Reserved	
04_H	Status		Command	
08_H	Class Code			Revision ID
$0C_H$	BIST	Header Code	Latency Timer	Cache Line Size
10_H	Base Address			
14_H	Reserved			
18_H				
$1C_H$				
20_H				
24_H				
28_H	Reserved			
$2C_H$	Reserved		Reserved	
30_H	Reserved			
34_H	Reserved			Cap_ptr
38_H	Reserved			
$3C_H$	Max_lat	Min_Gnt	Interrupt Pin	Interrupt Line
40_H	PMC		Next_Item_Ptr	Cap_ID
44_H	Data	DMCSR_BSE	PMCSR	
$E0_H$	Reserved			

(1) Command and Status registers (Offset: 04_H)

Access These registers can be read or written in 32-bit units.

Address 04_H

Initial value $0210\ 0000_H$. These registers are initialized by any reset.

31	30	29	28	27	26	25	24
Detected Parity Error	Signaled System Error	Received Master Abort	Received Target Abort	Signaled Target Abort	Devsel Timing1	Devsel Timing0	Data Parity Detected
R/W	R/W	R/W	R/W	R/W	R	R	R/W
23	22	21	20	19	18	17	16
Fast Back to Back Capable	0	0	Capabilities	0	0	0	0
R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8
0	0	0	0	0	0	Fast Back to Back Enable	SERR Enable
R	R	R	R	R	R	R	R/W
7	6	5	4	3	2	1	0
Wait Cycle Control	Parity Error Response	VGA Pallet Snoop	Memory Write and Invalidate	Special Cycle	Bus Master	Memory Space	I/O Space
R	R/W	R	R	R	R/W	R/W	R

Table 31-18 Command and Status register contents (1/2)

Bit position	Bit name	Function
31	Detected Parity Error	Indicates the parity error status. This bit is set when an address or data parity error is detected. This bit is cleared to 0 when 1 is written via the PCI bus.
30	Signaled System Error	Indicates the SERR status. This bit is set when a system error occurs. This bit is cleared to 0 when 1 is written via the PCI bus.
29	Received Master Abort	Indicates the master's master abort status. This bit is set when master operation is terminated by master abort. This bit is cleared to 0 when 1 is written via the PCI bus.
28	Received Target Abort	Indicates the master's target abort status. This bit is set when master operation is terminated by target abort. This bit is cleared to 0 when 1 is written via the PCI bus.
27	Signaled Target Abort	Indicates the slave's target abort status. This bit is set when slave operation is terminated by target abort. This bit is cleared to 0 when 1 is written via the PCI bus.
26:25	Devsel Timing[1:0]	Indicates the DEVSEL response speed. This field is fixed to 01 because only the medium mode is supported. These bits are read-only.
24	Data Parity Detected	This bit is set when a parity error is detected during master operation. This bit is cleared to 0 when 1 is written via the PCI bus. This bit is fixed to 0 when the parity error response is disabled by using the Parity Error Response bit (Command register).
23	Fast Back to Back Capable	Indicates whether Fast Back-to-Back is supported. This bit is fixed to 0 because Fast Back-to-Back is not supported. This bit is read-only.
22:21	–	Reserved (Be sure to write 0 to these bits.)

Table 31-18 Command and Status register contents (2/2)

Bit position	Bit name	Function
20	Capabilities	Indicates that the power management mode is supported. This bit is fixed to 1. This bit is read-only.
19:10	–	Reserved (Be sure to write 0 to these bits.)
9	Fast Back to Back Enable	This bit is used to enable Fast Back-to-Back. This bit is fixed to 0 because the USB host controller does not support Fast Back to Back. This bit is read-only.
8	SERR Enable	This bit is used to enable SERR. Set this bit to 1 when reporting system errors by using the SERR signal.
7	Wait Cycle Control	This bit is used to enable wait cycle control. This bit is fixed to 0 because the USB host controller does not support address/data stepping. This bit is read-only.
6	Parity Error Response	This bit is used to enable parity error responses. Set this bit to 1 when checking parity errors.
5	VGA Pallet Snoop	This bit is used to enable VGA palette snoop. This bit is fixed to 0 because the USB host controller does not support VGA palette snoop. This bit is read-only.
4	Memory Write and Invalidate	This bit is used to enable Memory Write and Invalidate. This bit is fixed to 0 because the USB host controller does not support Memory Write and Invalidate. This bit is read-only.
3	Special Cycle	This bit is used to enable special cycles. This bit is fixed to 0 because the USB host controller does not support Special Cycle. This bit is read-only.
2	Bus Master	This bit is used to enable bus master operation. This bit enables bus master accesses for the PCI bus and must be set to 1 before accessing SRAM via the system bus. Set this bit to 1 during host controller initialization.
1	Memory Space	This bit is used to enable memory space access. This bit enables memory accesses as defined in the PCI Specification, and must be set to 1 before accessing registers. Set this bit to 1 during host controller initialization.
0	I/O Space	This bit is used to enable I/O space access. This bit enables I/O accesses as defined in the PCI Specification, but this bit is fixed to 0 because the USB host controller does not use I/O accesses. This bit is read-only.

(2) Revision ID and Class Code registers (Offset: 08_H)

Access These registers are read-only, in 32-bit units.

Address 08_H

Initial value $0C03\ 1042_H$. These registers are initialized by any reset.

31	30	29	28	27	26	25	24
Base Class7	Base Class6	Base Class5	Base Class4	Base Class3	Base Class2	Base Class1	Base Class0
R	R	R	R	R	R	R	R
23	22	21	20	19	18	17	16
Sub Class7	Sub Class6	Sub Class5	Sub Class4	Sub Class3	Sub Class2	Sub Class1	Sub Class0
R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8
Program-ming I/F7	Program-ming I/F6	Program-ming I/F5	Program-ming I/F4	Program-ming I/F3	Program-ming I/F2	Program-ming I/F1	Program-ming I/F0
R	R	R	R	R	R	R	R
7	6	5	4	3	2	1	0
Revision ID7	Revision ID6	Revision ID5	Revision ID4	Revision ID3	Revision ID2	Revision ID1	Revision ID0
R	R	R	R	R	R	R	R

Table 31-19 Revision ID and Class Code register contents

Bit position	Bit name	Function
31:24	Base Class [7:0]	Indicates the base class defined in the PCI Specification (Class Code). This field is fixed to $0C_H$, which indicates the serial peripheral interface bus controller.
23:16	Sub Class [7:0]	Indicates the subclass defined in the PCI Specification (Class Code). This field is fixed to 03_H, which indicates the USB device.
15:8	Program-ming I/F [7:0]	Indicates the program interface defined in the PCI Specification (Class Code). This field is fixed to 10_H, which indicates an OHCI-specification USB.
7:0	Revision ID [7:0]	Indicates the revision of the USB host controller (Class Code). These bits are fixed to 42_H.

(3) Cache Line Size, Latency Timer, Header Type, and BIST registers (Offset: $0C_H$)

Access These registers can be read or written in 32-bit units.

Address $0C_H$

Initial value 00000800_H. These registers are initialized by any reset.

31	30	29	28	27	26	25	24
BIST7	BIST6	BIST5	BIST4	BIST3	BIST2	BIST1	BIST0
R	R	R	R	R	R	R	R
23	22	21	20	19	18	17	16
Header Type7	Header Type6	Header Type5	Header Type4	Header Type3	Header Type2	Header Type1	Header Type0
R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8
Latency Timer7	Latency Timer6	Latency Timer5	Latency Timer4	Latency Timer3	Latency Timer2	Latency Timer1	Latency Timer0
R	R	R	R	R	R	R	R
7	6	5	4	3	2	1	0
Cache Line Size7	Cache Line Size6	Cache Line Size5	Cache Line Size4	Cache Line Size3	Cache Line Size2	Cache Line Size1	Cache Line Size0
R	R	R	R	R	R	R	R

Table 31-20 Cache Line Size, Latency Timer, Header Type, and BIST register contents

Bit position	Bit name	Function
31:24	BIST[7:0]	Field for self-testing. These bits are fixed to 00_H. These bits are read-only.
23:16	Header Type[7:0]	Field for reporting the header type to the system This field is fixed to 00_H because the device is a PCI device. Particularly, bit 23 is fixed to 0 because multifunction is not supported. These bits are read-only.
15:8	Latency Timer[7:0]	Field for reporting the latency timer to the system The lower 2 bits are fixed to 00. These bits are read-only.
7:0	Cache Line Size[7:0]	Field for reporting the cache line size to the system These bits are fixed to 00_H. These bits are read-only.

(4) OHCI Base Address register (Offset: 10_H)

Access This register can be read only in 32-bit units.

Address 10_H

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
OHCI Base Address27	OHCI Base Address26	OHCI Base Address25	OHCI Base Address24	OHCI Base Address23	OHCI Base Address22	OHCI Base Address21	OHCI Base Address20
R	R	R	R	R	R	R	R
23	22	21	20	19	18	17	16
OHCI Base Address19	OHCI Base Address18	OHCI Base Address17	OHCI Base Address16	OHCI Base Address15	OHCI Base Address14	OHCI Base Address13	OHCI Base Address12
R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8
OHCI Base Address11	OHCI Base Address10	OHCI Base Address9	OHCI Base Address8	OHCI Base Address7	OHCI Base Address6	OHCI Base Address5	OHCI Base Address4
R	R	R	R	R	R	R	R
7	6	5	4	3	2	1	0
OHCI Base Address3	OHCI Base Address2	OHCI Base Address1	OHCI Base Address0	Prefetchable	Type1	Type0	Memory Space Indicator
R	R	R	R	R	R	R	R

Table 31-21 OHCI Base Address register contents

Bit position	Bit name	Function
31:4	OHCI Base Address [27:0]	Specify the OHCI operational register address by using bits 31 to 12. Specify the OHCI operational register base address defined by the system during initialization. Bits 11 to 4 are fixed to 00_H. These bits are read-only.
3	Prefetchable	Field indicating that the field specified by using the OHCI Base Address bits is a memory space. This bit is fixed to 0. This bit is fixed to 0, which indicates that prefetching from the field is prohibited. This bit is read-only.
2:1	Type[1:0]	Field that indicates the base address type. This field is fixed to 00. Indicates that the address specified by using the OHCI Base Address bits is a specific position in a 32-bit space. These bits are read-only.
0	Memory Space Indicator	Field indicating that the field specified by using the OHCI Base Address bits is a memory space. This bit is fixed to 0. This bit is read-only.

(5) Capability Pointer register (Offset: 34_H)

Access This register is read-only, in 32-bit units.

Address 34_H

Initial value $0000\ 0040_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
7	6	5	4	3	2	1	0
Capability Pointer7	Capability Pointer6	Capability Pointer5	Capability Pointer4	Capability Pointer3	Capability Pointer2	Capability Pointer1	Capability Pointer0
R	R	R	R	R	R	R	R

Table 31-22 Capability Pointer register contents

Bit position	Bit name	Function
31:10	–	Reserved (Be sure to write 0 to these bits.)
7:0	Capability Pointer[7:0]	These bits are fixed to 40_H.

(6) Interrupt Line, Interrupt Pin, Min Gnt, and Max Latency registers (Offset: $3C_H$)

Access These registers can be read only in 32-bit units.

Address $3C_H$

Initial value $2A01\ 0100_H$. These registers are initialized by any reset.

31	30	29	28	27	26	25	24
Max Latency7	Max Latency6	Max Latency5	Max Latency4	Max Latency3	Max Latency2	Max Latency1	Max Latency0
R	R	R	R	R	R	R	R
23	22	21	20	19	18	17	16
Min Gnt7	Min Gnt6	Min Gnt5	Min Gnt4	Min Gnt3	Min Gnt2	Min Gnt1	Min Gnt0
R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8
Interrupt Pin7	Interrupt Pin6	Interrupt Pin5	Interrupt Pin4	Interrupt Pin3	Interrupt Pin2	Interrupt Pin1	Interrupt Pin0
R	R	R	R	R	R	R	R
7	6	5	4	3	2	1	0
Interrupt Line7	Interrupt Line6	Interrupt Line5	Interrupt Line4	Interrupt Line3	Interrupt Line2	Interrupt Line1	Interrupt Line0
R	R	R	R	R	R	R	R

Table 31-23 Interrupt Line, Interrupt Pin, Min Gnt, and Max Latency register contents

Bit position	Bit name	Function
31:24	MaxLatency [7:0]	Indicates the maximum latency. These bits are fixed to $2A_H$. These bits are read-only.
23:16	Min Gnt [7:0]	Indicates the minimum grant time. These bits are fixed to 01_H. These bits are read-only.
15:8	Interrupt Pin[7:0]	Indicates the interrupt output pin. Fixed to 01_H because INTA is used. These bits are read-only.
7:0	Interrupt Line[7:0]	Indicates the interrupt line. These bits are fixed to 00_H.

(7) Capability Identifier, Next Item Pointer, and Power Management Capabilities registers (Offset: 40_H)

Access These registers are read-only, in 32-bit units.

Address 40_H

Initial value $7E02\ 0001_H$. These registers are initialized by any reset.

31	30	29	28	27	26	25	24
PME Support4	PME Support3	PME Support2	PME Support1	PME Support0	D2 Support	D1 Support	Aux Current2
R	R	R	R	R	R	R	R
23	22	21	20	19	18	17	16
Aux Current1	Aux Current0	DSI	0	PME CLK	Version2	Version1	Version0
R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8
Next Item Pointer7	Next Item Pointer6	Next Item Pointer5	Next Item Pointer4	Next Item Pointer3	Next Item Pointer2	Next Item Pointer1	Next Item Pointer0
R	R	R	R	R	R	R	R
7	6	5	4	3	2	1	0
Capability Identifier7	Capability Identifier6	Capability Identifier5	Capability Identifier4	Capability Identifier3	Capability Identifier2	Capability Identifier1	Capability Identifier0
R	R	R	R	R	R	R	R

Table 31-24 Capability Identifier, Next Item Pointer, and Power Management Capabilities register contents

Bit position	Bit name	Function
31	PME Support [4:0]	Indicates whether the D3 Cold state is supported (Power Management Capabilities). This bit is fixed to 0 because the D3 Cold state is not supported.
30:27		Indicates that PME (Power Management Capabilities) is supported in all PCI power states (D0 to D3). These bits are fixed to 1111.
26	D2 Support	Indicates that D2 of the PCI power states is supported (Power Management Capabilities). This bit is fixed to 1.
25	D1 Support	Indicates that D1 of the PCI power states is supported (Power Management Capabilities). This bit is fixed to 1.
24:22	Aux Current [2:0]	Indicates that asserting PME interrupts in the D3 Cold state is not supported (Power Management Capabilities). These bits are fixed to 000.
21	DSI	Indicates that no special initialization is required when using Power Management (Power Management Capabilities). This bit is fixed to 0.
19	PME CLK	Indicates that PCLK is not required for generating PME interrupts (Power Management Capabilities). This bit is fixed to 0.
18:16	Version[2:0]	Indicates the power management version (Power Management Capabilities). These bits are fixed to 010 according to the circuit configuration implemented in the OHCI host controller.
15:8	Next Item Pointer[7:0]	Indicates that the next item does not exist. These bits are fixed to 00_H.
7:0	Capability Identifier [7:0]	Indicates the Power Management register ID These bits are fixed to 01_H.

(8) Power Management Control/Status and PMCSR Bridge Support Extensions registers (Offset: 44_H)

Access These registers can be read or written in 32-bit units.

Address 44_H

Initial value $0000\ 0000_H$. These registers are initialized by any reset.

31	30	29	28	27	26	25	24
Data7	Data6	Data5	Data4	Data3	Data2	Data1	Data0
R	R	R	R	R	R	R	R
23	22	21	20	19	18	17	16
BPCC Enable	B2_B3	0	0	0	0	0	0
R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8
PME Status	Data Scale1	Data Scale0	Data Select3	Data Select2	Data Select1	Data Select0	PME Enable
R/W	R	R	R	R	R	R	R/W
7	6	5	4	3	2	1	0
0	0	0	0	0	0	Power State1	Power State0
R	R	R	R	R	R	R	R

Table 31-25 Power Management Control/Status and PMCSR Bridge Support Extensions register contents

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>31:24</td><td>Data[7:0]</td><td>This field is optional in the PCI Specification, and is not supported by the OHCI host controller (Data).
These bits are fixed to 00.
These bits are read-only.</td></tr>
<tr><td>23</td><td>BPCC Enable</td><td>Bit for PCI bridges, which is not supported by the OHCI host controller (PMCSR Bridge Support Extensions).
This bit is fixed to 0.
This bit is read-only.</td></tr>
<tr><td>22</td><td>B2_B3</td><td>Bit for bridges, which is not supported by the OHCI host controller (PMCSR Bridge Support Extensions).
This bit is fixed to 0.
This bit is read-only.</td></tr>
<tr><td>15</td><td>PME Status</td><td>Indicates the PME interrupt status (Power Management Control/Status).
This bit is set to 1 when conditions for asserting PME are satisfied.
This bit is cleared to 0 when 1 is written to it via the PCI bus.</td></tr>
<tr><td>14:13</td><td>Data Scale [1:0]</td><td>This field is optional in the PCI Specification, and is not supported by the OHCI host controller (Power Management Control/Status).
These bits are fixed to 00.
These bits are read-only.</td></tr>
<tr><td>12:9</td><td>Data Select [3:0]</td><td>This field is optional in the PCI Specification, and is not supported by the OHCI host controller (Power Management Control/Status).
These bits are fixed to 0000.
These bits are read-only.</td></tr>
<tr><td>8</td><td>PME Enable</td><td>Bit for setting whether to use external pin PME (Power Management Control/Status).
If this bit is set to 1, a PME interrupt occurs when returning from power management mode.</td></tr>
<tr><td>1:0</td><td>Power State[1:0]</td><td>Indicates the PCI power status (Power Management Control/Status).
The status varies as follows depending on the setting of these bits:
<table>
<tr><th>Power State1</th><th>Power State0</th><th>Field status</th></tr>
<tr><td>0</td><td>0</td><td>D0 State</td></tr>
<tr><td>0</td><td>1</td><td>D1 State</td></tr>
<tr><td>1</td><td>0</td><td>D2 State</td></tr>
<tr><td>1</td><td>1</td><td>D3 hot State</td></tr>
</table></td></tr>
</table>

31.7.3 OHCI operational registers

The OHCI operational registers are incorporated in the OHCI host controller, and consist of the registers listed in *Table 31-26 “OHCI operational registers”*. For details, see OpenHCI Specification Release 1.0a.

Because OpenHCI Specification Release 1.0a defines the port number as [1: number of ports], this section observes this rule. For example, port 1 corresponds to host channel 0 and port 2 corresponds to host channel 1. For details about the base address, see *Table 31-2 "Register base addresses <USAH0_base>"*.

Table 31-26 OHCI operational registers

Bit / Address	31 24	23 16	15 8	7 0
00_H	HcRevision			
04_H	HcControl			
08_H	HcCommandStatus			
$0C_H$	HcInterruptStatus			
10_H	HcInterruptEnable			
14_H	HcInterruptDisable			
18_H	HcHCCA			
$1C_H$	HcPeriodCurrentED			
20_H	HcControlHeadED			
24_H	HcControlCurrentED			
28_H	HcBulkHeadED			
$2C_H$	HcBulkCurrentED			
30_H	HcDoneHead			
34_H	HcFmInterval			
38_H	HcFmRemaining			
$3C_H$	HcFmNumber			
40_H	HcPeriodicStart			
44_H	HcLSThreshold			
48_H	HcRhDescriptorA			
$4C_H$	HcRhDescriptorB			
50_H	HcRhStatus			
54_H	HcRhPortStatus1			
$5C_H$ to FF_H	Reserved			

Note HC: OHCI host controller
HCD: OHCI host controller driver
ED: Endpoint descriptor
TD: Transfer descriptor
EOP: End of packet
SOF: Start of frame

(1) HcRevision register (Offset: 00_H)

Access This register is read-only, in 32-bit units.

Address <USHA0_base_OHCI> + 00_H

Initial value $0000\ 0010_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	Legacy
R	R	R	R	R	R	R	R
7	6	5	4	3	2	1	0
Revision							
R	R	R	R	R	R	R	R

Table 31-27 HcRevision register contents

Bit position	Bit name	Function
31:9	–	Reserved (Be sure to write 0 to these bits.)
8	Legacy	Indicates whether legacy-support registers are implemented in the OHCI host controller. This bit is fixed to 0 because the OHCI host controller does not support the legacy function.
7:0	Revision	Indicates the version of the OHCI Specification implemented in the OHCI host controller These bits are fixed to 10_H because the OHCI host controller is compliant with OHCI Specification 1.0a.

(2) HcControl register (Offset: 04_H)

Access This register can be read or written in 32-bit units.

Address <USHA0_base_OHCI> + 04_H

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8
0	0	0	0	0	RWE	RWC	IR
R	R	R	R	R	R/W	R/W	R/W
7	6	5	4	3	2	1	0
HCFS1	HCFS0	BLE	CLE	IE	PLE	CBSR1	CBSR0
R	R	R/W	R/W	R/W	R/W	R/W	R/W

Table 31-28 HcControl register contents (1/2)

Bit position	Bit name	Function
31:11	–	Reserved (Be sure to write 0 to these bits.)
10	RWE	Remote WakeUp Enable Specifies how to detect an upstream resume signal. 1: Detect the resume signal as Remote Wakeup. 0: Do not detect the resume signal as Remote Wakeup.
9	RWC	Remote WakeUp Connect Indicates whether the OHCI host controller supports Remote Wakeup. This bit must be set during initialization if remote wakeup is supported in the system. 1: Remote Wakeup is supported. 0: Remote Wakeup is not supported.
8	IR	Interrupt Routing Indicates the method by which the OHCI host controller outputs interrupts. Specify how to report to the HcInterruptStatus register the source of an interrupt that has occurred. 1: Report the interrupt by using SMMI. 0: Report the interrupt by using INTA.
7:6	HCFS[1:0]	Host Controller Functional State Indicates the OHCI host controller operating status. **HCFS** / **USB status** 00 / USB Reset 01 / USB Resume 10 / USB Operational 11 / USB Suspend When the OHCI host controller transitions the USB Operational state, it starts management of frames in 1 ms units. This status is always controlled by the host controller driver (HCD) except for when the status transitions to USB Resume by remote wakeup during USB Suspend. This field is set to 00 after a hardware reset, or 11 after a software reset.

Table 31-28 HcControl register contents (2/2)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>5</td><td>BLE</td><td>Bulk List Enable
Specify whether to perform bulk list processing.
1: Perform bulk list processing.
0: Do not perform bulk list processing.
The setting of this bit becomes valid from the next frame of bulk list processing.
This bit must be 0 when modifying a bulk list.</td></tr>
<tr><td>4</td><td>CLE</td><td>Control List Enable
Specify whether to perform control list processing.
1: Perform control list processing.
0: Do not perform control list processing.
The setting of this bit becomes valid from the next frame of control list processing.
This bit must be 0 when modifying a control list.</td></tr>
<tr><td>3</td><td>IE</td><td>Isochronous Enable
Specify whether to perform isochronous ED processing.
This bit is checked when an isochronous ED is detected during list processing to determine whether to perform isochronous ED processing.
1: Perform isochronous transfer processing.
0: Do not perform isochronous transfer processing.
The setting of this bit becomes valid from the next frame of isochronous transfer processing.</td></tr>
<tr><td>2</td><td>PLE</td><td>Periodic List Enable
Specify whether to perform periodic list processing.
1: Perform periodic list processing.
0: Do not perform periodic list processing.
The setting of this bit becomes valid from the next frame of periodic list processing.</td></tr>
<tr><td>1:0</td><td>CBSR[1:0]</td><td>Control Bulk Service Ratio
Specify the ratio of control transfer and bulk transfer services.
The ratio specified in this field is retained and transfer is performed using this ratio during periodic list processing.
<table>
<tr><th>CBSR</th><th>ED Service ratio (Bulk ED:Control ED)</th></tr>
<tr><td>00</td><td>1:1</td></tr>
<tr><td>01</td><td>2:1</td></tr>
<tr><td>10</td><td>3:1</td></tr>
<tr><td>11</td><td>4:1</td></tr>
</table>
</td></tr>
</table>

(3) HcCommandStatus register (Offset: 08_H)

Access This register can be read or written in 32-bit units.

Address <USHA0_base_OHCI> + 08_H

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
23	22	21	20	19	18	17	16
0	0	0	0	0	0	SOC1	SOC0
R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
7	6	5	4	3	2	1	0
0	0	0	0	OCR	BLF	CLF	HCR
R	R	R	R	R/W	R/W	R/W	R/W

Table 31-29 HcCommandStatus register contents (1/2)

Bit position	Bit name	Function
31:18	–	Reserved (Be sure to write 0 to these bits.)
17:16	SOC[1:0]	Scheduling Overrun Count This field is used to count the number of schedule overruns. This counter is incremented each time a schedule overrun occurs. This counter continues to be incremented even if the SO bit of the HcInterruptStatus register is set to 1. These bits are read-only.
15:4	–	Reserved (Be sure to write 0 to these bits.)
3	OCR	Ownership Change Request This bit is used to request to change the ownership of the OHCI host controller.

Table 31-29 HcCommandStatus register contents (2/2)

Bit position	Bit name	Function
2	BLF	Bulk List Filled Indicates whether a TD exists in a bulk list. The host controller driver (HCD) sets this bit to 1 each time a TD is added to the ED of a bulk list. The OHCI host controller checks this bit when starting bulk list head processing. Bulk list processing is not started if this bit is 0. If this bit is 1, the OHCI host controller clears it to 0 and bulk list processing starts. If a TD is detected in a bulk list, this bit is set to 1 to continue bulk list processing. The host controller driver must set this bit before reconfiguring the list, setting the BLE bit of the HcCommand register, and starting list processing.
1	CLF	Control List Filled Indicates whether a control list exists. The host controller driver (HCD) sets this bit to 1 each time a TD is added to the ED of a control list. The OHCI host controller checks this bit when starting control list head processing. Control list processing is not started if this bit is 0. If this bit is 1, the OHCI host controller clears it to 0 and control list processing starts. If a TD is detected in a control list, this bit is set to 1 to continue control list processing. The host controller driver must set this bit before reconfiguring the list, setting the CLE bit of the HcCommand register, and starting list processing.
0	HCR	Host Controller Reset This bit is used to execute a software reset for the OHCI host controller. When this bit is set, the USB Suspend state is entered regardless of the OHCI host controller function state. The OHCI host controller clears this bit to 0 when the reset operation finishes.

(4) HcInterruptStatus register (Offset: $0C_H$)

Access This register can be read or written in 32-bit units.

Address <USHA0_base_OHCI> + $0C_H$

Initial value $0000\ 0000_H$. This register is initialized by any reset.

Caution In addition, if an interrupt from the USB hostcontroller is detected, the value of the interrupt request is stored in the EICn.EIRFn bit (n = 190 to 192). Therefore, even if the CPU acknowledges the interrupt and the EIRFn bit of the interrupt control register (INTUSHA0xxx) is cleared to 0, the EICn.EIRFn bit is immediately set to 0 and an interrupt is continuously generated. To prevent this from happening, clear the status flag of each interrupt request in the interrupt service routine, and then forcibly clear the EICn.EIRFn bit to 0.

31	30	29	28	27	26	25	24
0	OC	0	0	0	0	0	0
R	R/W	R	R	R	R	R	R
23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
7	6	5	4	3	2	1	0
0	RHSC	FNO	UE	RD	SF	WDH	SO
R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 31-30 HcInterruptStatus register contents (1/2)

Bit position	Bit name	Function
31	–	Reserved (Be sure to write 0 to these bits.)
30	OC	Ownership Change Indicates that an Ownership Change interrupt has occurred. This bit is set to 1 when the OCR field of the HcCommandStatus register is set to 1. An SMMI interrupt occurs if the interrupt source is not masked. 1: An OC interrupt has occurred. 0: No OC interrupt has occurred. Writing 1 to this bit clears the interrupt source.
29:7	–	Reserved (Be sure to write 0 to these bits.)
6	RHSC	Root Hub Status Change Indicates that the HcRhPortStatus register setting has been changed. This bit is set to 1 when the status of the HcRhPortStatus register has been changed due to a hardware interrupt. 1: An RHSC interrupt has occurred. 0: No RHSC interrupt has occurred. Writing 1 to this bit clears the interrupt source.

Table 31-30 HcInterruptStatus register contents (2/2)

Bit position	Bit name	Function
5	FNO	Frame Number Overflow Indicates that the MSB of a frame number has been changed. This bit is set to 1 after HccaFrameNumber is updated in a frame whose frame number MSB changes from 0 to 1, or 1 to 0. 1: An FNO interrupt has occurred. 0: No FNO interrupt has occurred. Writing 1 to this bit clears the interrupt source.
4	UE	Unrecoverable Error Indicates that a system error has been detected on the PCI bus not related to the USB. 1: A UE interrupt has occurred. 0: No UE interrupt has occurred. Writing 1 to this bit clears the interrupt source.
3	RD	Resume Detected Indicates that a resume signal has been detected. This bit is set to 1 when asserting the resume signal by a device on the USB bus is detected. This bit is not set if USB Resume is issued by the host controller driver (HCD). 1: An RD interrupt has occurred. 0: No RD interrupt has occurred. Writing 1 to this bit clears the interrupt source.
2	SF	StartOfFrame Indicates that HccaFrameNumber was updated at the beginning of a frame. The OHCI host controller transmits the SOF packet and updates HccaFrameNumber. 1: An SF interrupt has occurred. 0: No SF interrupt has occurred. Writing 1 to this bit clears the interrupt source.
1	WDH	Writeback Done Head Indicates that the OHCD host controller updated HccaDoneHead. The OHCI host controller sets this bit to 1 immediately after updating HccaDoneHead, and does not update HccaDoneHead until this bit is cleared to 0. 1: A WDH interrupt has occurred. 0: No WDH interrupt has occurred. Writing 1 to this bit clears the interrupt source.
0	SO	Scheduling Overrun Indicates that the USB scheduling in a frame has overrun. This bit is set to 1 after updating of FrameNumberUpdate of the frame following the frame in which the USB scheduling overruns. When this bit is set to 1, the SOC field of the HcCommandStatus register is incremented. 1: An SO interrupt has occurred. 0: No SO interrupt has occurred. Writing 1 to this bit clears the interrupt source.

(5) HcInterruptEnable register (Offset: 10_H)

Access This register can be read or written in 32-bit units.

Address <USHA0_base_OHCI> + 10_H

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
MIE	OCE	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R
23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
7	6	5	4	3	2	1	0
0	RHSCE	FNOE	UEE	RDE	SFE	WDHE	SOE
R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 31-31 HcInterruptEnable register contents (1/2)

Bit position	Bit name	Function
31	MIE	Master Interrupt Enable Specify whether to enable interrupt sources specified by using bits 30 and 6 to 0. 1: Enable all the specified interrupt sources. 0: Invalid (writing 0 is ignored). To disable the interrupt, write 1 to the corresponding bit of the HcInterruptDisable register.
30	OCE	Ownership Change Enable Specifies whether to enable OC as an interrupt source. 1: Enable OC as an interrupt source. 0: Invalid (writing 0 is ignored). Writing 1 to this bit specifies OC as an interrupt source. To disable the interrupt, write 1 to the corresponding bit of the HcInterruptDisable register.
29:7	–	Reserved (Be sure to write 0 to these bits.)
6	RHSCE	Root Hub Status Change Enable Specify whether to enable RHSC as an interrupt source. 1: Enable RHSC as an interrupt source. 0: Invalid (writing 0 is ignored). Writing 1 to this bit specifies RHSC as an interrupt source. To disable the interrupt, write 1 to the corresponding bit of the HcInterruptDisable register.
5	FNOE	Frame Number Overflow Enable Specify whether to enable FNO as an interrupt source. 1: Enable FNO as an interrupt source. 0: Invalid (writing 0 is ignored). Writing 1 to this bit specifies FNO as an interrupt source. To disable the interrupt, write 1 to the corresponding bit of the HcInterruptDisable register.

Table 31-31 HcInterruptEnable register contents (2/2)

Bit position	Bit name	Function
4	UEE	Unrecoverable Error Enable Specify whether to enable UE as an interrupt source. 1: Enable UE as an interrupt source. 0: Invalid (writing 0 is ignored). Writing 1 to this bit specifies UE as an interrupt source. To disable the interrupt, write 1 to the corresponding bit of the HcInterruptDisable register.
3	RDE	Resume Detected Enable Specify whether to enable RD as an interrupt source. 1: Enable RD as an interrupt source. 0: Invalid (writing 0 is ignored). Writing 1 to this bit specifies RD as an interrupt source. To disable the interrupt, write 1 to the corresponding bit of the HcInterruptDisable register.
2	SFE	State Of Frame Enable Specify whether to enable SF as an interrupt source. 1: Enable SF as an interrupt source. 0: Invalid (writing 0 is ignored). Writing 1 to this bit specifies SF as an interrupt source. To disable the interrupt, write 1 to the corresponding bit of the HcInterruptDisable register.
1	WDHE	Writeback Done Head Enable Specify whether to enable WDH as an interrupt source. 1: Enable WDH as an interrupt source. 0: Invalid (writing 0 is ignored). Writing 1 to this bit specifies WDH as an interrupt source. To disable the interrupt, write 1 to the corresponding bit of the HcInterruptDisable register.
0	SOE	Scheduling Overrun Enable Specify whether to enable SO as an interrupt source. 1: Enable SO as an interrupt source. 0: Invalid (writing 0 is ignored). Writing 1 to this bit specifies SOE as an interrupt source. To disable the interrupt, write 1 to the corresponding bit of the HcInterruptDisable register.

(6) HcInterruptDisable register (Offset: 14_H)

Access This register can be read or written in 32-bit units.

Address <USHA0_base_OHCI> + 14_H

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
MID	OCD	0	0	0	0	0	0
R/W	R/W	R	R	R	R	R	R
23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
7	6	5	4	3	2	1	0
0	RHSCD	FNOD	UED	RDD	SFD	WDHD	SOD
R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 31-32 HcInterruptDisable register contents (1/2)

Bit position	Bit name	Function
31	MID	Master Interrupt Disable Specify whether to disable interrupt sources specified by using bits 30 and 6 to 0 of the HcInterruptEnable register. 1: Disable all the specified interrupt sources. 0: Invalid (writing 0 is ignored). When this bit is read, the value of the corresponding bit of the HcInterruptEnable register is read. To enable the interrupt, write 1 to the corresponding bit of the HcInterruptEnable register.
30	OCD	Ownership Change Disable Specifies whether to remove OC from the interrupt sources. 1: Disable OC as an interrupt source. 0: Invalid (writing 0 is ignored). When this bit is read, the value of the corresponding bit of the HcInterruptEnable register is read. Writing 1 to this bit removes OC from the interrupt sources. To enable the interrupt, write 1 to the corresponding bit of the HcInterruptEnable register.
29:7	–	Reserved (Be sure to write 0 to these bits.)
6	RHSCD	Root Hub Status Change Disable Specify whether to remove RHSC from the interrupt sources. 1: Disable RHSC as an interrupt source. 0: Invalid (writing 0 is ignored). When this bit is read, the value of the corresponding bit of the HcInterruptEnable register is read. Writing 1 to this bit removes RHSC from the interrupt sources. To enable the interrupt, write 1 to the corresponding bit of the HcInterruptEnable register.

Table 31-32 HcInterruptDisable register contents (2/2)

Bit position	Bit name	Function
5	FNOD	Frame Number Overflow Disable Specify whether to remove FNO from the interrupt sources. 1: Disable FNO as an interrupt source. 0: Invalid (writing 0 is ignored). When this bit is read, the value of the corresponding bit of the HcInterruptEnable register is read. Writing 1 to this bit removes FNO from the interrupt sources. To enable the interrupt, write 1 to the corresponding bit of the HcInterruptEnable register.
4	UED	Unrecoverable Error Disable Specify whether to remove UE from the interrupt sources. 1: Disable UE as an interrupt source. 0: Invalid (writing 0 is ignored). When this bit is read, the value of the corresponding bit of the HcInterruptEnable register is read. Writing 1 to this bit removes to UE from the interrupt sources. To enable the interrupt, write 1 to the corresponding bit of the HcInterruptEnable register.
3	RDD	Resume Detected Disable Specify whether to remove RD from the interrupt sources. 1: Disable RD as an interrupt source. 0: Invalid (writing 0 is ignored). When this bit is read, the value of the corresponding bit of the HcInterruptEnable register is read. Writing 1 to this bit removes RD from the interrupt sources. To enable the interrupt, write 1 to the corresponding bit of the HcInterruptEnable register.
2	SFD	StartOfFrame Disable Specify removing of SF from interrupt sources. 1: Disable SF as an interrupt source. 0: Invalid (writing 0 is ignored). When this bit is read, the value of the corresponding bit of the HcInterruptEnable register is read. Writing 1 to this bit removes SF from the interrupt sources. To enable the interrupt, write 1 to the corresponding bit of the HcInterruptEnable register.
1	WDHD	Writeback Done Head Disable Specify whether to remove WDH from the interrupt sources. 1: Disable WDH as an interrupt source. 0: Invalid (writing 0 is ignored). When this bit is read, the value of the corresponding bit of the HcInterruptEnable register is read. Writing 1 to this bit removes WDH from the interrupt sources. To enable the interrupt, write 1 to the corresponding bit of the HcInterruptEnable register.
0	SOD	Scheduling Overrun Disable Specify whether to remove SO from interrupt sources. 1: Disable SO as an interrupt source. 0: Invalid (writing 0 is ignored). When this bit is read, the value of the corresponding bit of the HcInterruptEnable register is read. Writing 1 to this bit removes SO from the interrupt sources. To enable the interrupt, write 1 to the corresponding bit of the HcInterruptEnable register.

(7) HcHCCA register (Offset: 18_H)

Access This register can be read or written in 32-bit units.

Address <USHA0_base_OHCI> + 18_H

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
HcHCCA23	HcHCCA22	HcHCCA21	HcHCCA20	HcHCCA19	HcHCCA18	HcHCCA17	HcHCCA16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
HcHCCA15	HcHCCA14	HcHCCA13	HcHCCA12	HcHCCA11	HcHCCA10	HcHCCA9	HcHCCA8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
HcHCCA7	HcHCCA6	HcHCCA5	HcHCCA4	HcHCCA3	HcHCCA2	HcHCCA1	HcHCCA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R

Table 31-33 HcHCCA register contents

Bit position	Bit name	Function
31:8	HcHCCA [23:0]	Specify the base address of the RAM to which the OHCI host controller communication area is allocated. This field must be set up during initialization. The OHCI host controller requests the 256-byte area from the specified base address, as the HCCA.

(8) HcPeriodCurrentED register (Offset: $1C_H$)

Access This register is read-only, in 32-bit units.

Address <USHA0_base_OHCI> + $1C_H$

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
Period CurrentED27	Period CurrentED26	Period CurrentED25	Period CurrentED24	Period CurrentED23	Period CurrentED22	Period CurrentED21	Period CurrentED20
R	R	R	R	R	R	R	R
23	22	21	20	19	18	17	16
Period CurrentED19	Period CurrentED18	Period CurrentED17	Period CurrentED16	Period CurrentED15	Period CurrentED14	Period CurrentED13	Period CurrentED12
R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8
Period CurrentED11	Period CurrentED10	Period CurrentED9	Period CurrentED8	Period CurrentED7	Period CurrentED6	Period CurrentED5	Period CurrentED4
R	R	R	R	R	R	R	R
7	6	5	4	3	2	1	0
Period CurrentED3	Period CurrentED2	Period CurrentED1	Period CurrentED0	0	0	0	0
R	R	R	R	R	R	R	R

Table 31-34 HcPeriodCurrentED register contents

Bit position	Bit name	Function
31:4	Period CurrentED [27:0]	Indicates the processing address included in a periodic list. The OHCI host controller updates the value of this field each time a periodic list processing sequence ends.

(9) HcControlHeadED register (Offset: 20_H)

Access This register can be read or written in 32-bit units.

Address <USHA0_base_OHCI> + 20_H

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
Control HeadED27	Control HeadED26	Control HeadED25	Control HeadED24	Control HeadED23	Control HeadED22	Control HeadED21	Control HeadED20
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
Control HeadED19	Control HeadED18	Control HeadED17	Control HeadED16	Control HeadED15	Control HeadED14	Control HeadED13	Control HeadED12
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
Control HeadED11	Control HeadED10	Control HeadED9	Control HeadED8	Control HeadED7	Control HeadED6	Control HeadED5	Control HeadED4
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
Control HeadED3	Control HeadED2	Control HeadED1	Control HeadED0	0	0	0	0
R/W	R/W	R/W	R/W	R	R	R	R

Table 31-35 HcControlHeadED register contents

Bit position	Bit name	Function
31:4	Control HeadED [27:0]	Specify the start address of the list ED for control transfer. To perform control transfer, this field must be set to 1 before setting the CLE bit of the HcControl register.

(10) HcControlCurrentED register (Offset: 24_H)

Access This register can be read or written in 32-bit units.

Address <USHA0_base_OHCI> + 24_H

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
Control CurrentED27	Control CurrentED26	Control CurrentED25	Control CurrentED24	Control CurrentED23	Control CurrentED22	Control CurrentED21	Control CurrentED20
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
Control CurrentED19	Control CurrentED18	Control CurrentED17	Control CurrentED16	Control CurrentED15	Control CurrentED14	Control CurrentED13	Control CurrentED12
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
Control CurrentED11	Control CurrentED10	Control CurrentED9	Control CurrentED8	Control CurrentED7	Control CurrentED6	Control CurrentED5	Control CurrentED4
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
Control CurrentED3	Control CurrentED2	Control CurrentED1	Control CurrentED0	0	0	0	0
R/W	R/W	R/W	R/W	R	R	R	R

Table 31-36 HcControlCurrentED register contents

Bit position	Bit name	Function
31:4	Control CurrentED [27:0]	Indicates the processing address included in a control list. The OHCI host controller updates the value of this field each time a control ED processing sequence ends. When configuring a new list, set this field $0000\ 0000_H$, which indicates the end of the list. When suspending or resuming a transfer, it must be guaranteed that the ED indicated by this field exists.

(11) HcBulkHeadED register (Offset: 28_H)

Access This register can be read or written in 32-bit units.

Address <USHA0_base_OHCI> + 28_H

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
BulkHead ED27	BulkHead ED26	BulkHead ED25	BulkHead ED24	BulkHead ED23	BulkHead ED22	BulkHead ED21	BulkHead ED20
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
BulkHead ED19	BulkHead ED18	BulkHead ED17	BulkHead ED16	BulkHead ED15	BulkHead ED14	BulkHead ED13	BulkHead ED12
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
BulkHead ED11	BulkHead ED10	BulkHead ED9	BulkHead ED8	BulkHead ED7	BulkHead ED6	BulkHead ED5	BulkHead ED4
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
BulkHead ED3	BulkHead ED2	BulkHead ED1	BulkHead ED0	0	0	0	0
R/W	R/W	R/W	R/W	R	R	R	R

Table 31-37 HcBulkHeadED register contents

Bit position	Bit name	Function
31-4	BulkHead ED[27:0]	Specify the start address of a list ED for bulk transfer. To perform a bulk transfer, this field must be set to 1 before setting the BLE bit of the HcControl register.

(12) HcBulkCurrentED register (Offset: $2C_H$)

Access This register can be read or written in 32-bit units.

Address <USHA0_base_OHCI> + $2C_H$

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
BulkCurrentED27	BulkCurrentED26	BulkCurrentED25	BulkCurrentED24	BulkCurrentED23	BulkCurrentED22	BulkCurrentED21	BulkCurrentED20
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
BulkCurrentED19	BulkCurrentED18	BulkCurrentED17	BulkCurrentED16	BulkCurrentED15	BulkCurrentED14	BulkCurrentED13	BulkCurrentED12
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
BulkCurrentED11	BulkCurrentED10	BulkCurrentED9	BulkCurrentED8	BulkCurrentED7	BulkCurrentED6	BulkCurrentED5	BulkCurrentED4
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
BulkCurrentED3	BulkCurrentED2	BulkCurrentED1	BulkCurrentED0	0	0	0	0
R/W	R/W	R/W	R/W	R	R	R	R

Table 31-38 HcBulkCurrentED register contents

Bit position	Bit name	Function
31:4	BulkCurrentED[27:0]	Indicates the address at which bulk list processing is being executed. The OHCI host controller updates the value of this field each time a bulk ED processing sequence ends. When configuring a new list, set this field $0000\ 0000_H$, which indicates the end of the list. When suspending or resuming a transfer, it must be guaranteed that the ED indicated by this field exists.

(13) HcDoneHead register (Offset: 30_H)

Access This register is read-only, in 32-bit units.

Address <USHA0_base_OHCI> + 30_H

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
DoneHead27	DoneHead26	DoneHead25	DoneHead24	DoneHead23	DoneHead22	DoneHead21	DoneHead20
R	R	R	R	R	R	R	R
23	22	21	20	19	18	17	16
DoneHead19	DoneHead18	DoneHead17	DoneHead16	DoneHead15	DoneHead14	DoneHead13	DoneHead12
R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8
DoneHead11	DoneHead10	DoneHead9	DoneHead8	DoneHead7	DoneHead6	DoneHead5	DoneHead4
R	R	R	R	R	R	R	R
7	6	5	4	3	2	1	0
DoneHead3	DoneHead2	DoneHead1	DoneHead0	0	0	0	0
R	R	R	R	R	R	R	R

Table 31-39 HcDoneHead register contents

Bit position	Bit name	Function
31:4	DoneHead [27:0]	Indicates the address of HcDoneHead of the OHCI host controller.

(14) HcFmInterval register (Offset: 34_H)

Access This register can be read or written in 32-bit units.

Address <USHA0_base_OHCI> + 34_H

Initial value $00002EDF_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
FIT	FSMPS14	FSMPS13	FSMPS12	FSMPS11	FSMPS10	FSMPS9	FSMPS8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
FSMPS7	FSMPS6	FSMPS5	FSMPS4	FSMPS3	FSMPS2	FSMPS1	FSMPS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
0	0	FI13	FI12	FI11	FI10	FI9	FI8
R	R	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
FI7	FI6	FI5	FI4	FI3	FI2	FI1	FI0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 31-40 HcFmInterval register contents

Bit position	Bit name	Function
31	FIT	Frame Interval Toggle This bit is used to synchronize the frame setting values of the host controller driver (HCD) and OHCI host controller. Toggle the setting of this bit when writing to the FI field by using the HCD. The OHCI host controller applies the FIT value to the FRT bit of the HcFmRemaining register when loading the FI field. The HCD can check whether the new FI field value has been applied by comparing the FIT bit value that was specified when it was written to the FI field and the FRT bit value that was read.
30:16	FSMPS [14:0]	FSLagest Data Packet Specify the maximum data amount that can be transmitted or received without causing a schedule overrun. The current frame position and the specified value are compared, and judged up to which position of a frame can be transferred. The result depends on the system bus performance, so this value is specified by the host controller driver (HCD).
15:14	–	Reserved (Be sure to write 0 to these bits.)
13:0	FI[13:0]	FrameInterval Specify the bit time for the interval of two successive SOFs in Full-Speed mode. This field must be set to $2EDF_H$ to satisfy the length of one frame (= 1 ms) prescribed by the USB Specification.

(15) HcFmRemaining register (Offset: 38_H)

Access This register is read-only, in 32-bit units.

Address <USHA0_base_OHCI> + 38_H

Initial value $00002EDF_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
FRT	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8
0	0	FR13	FR12	FR11	FR10	FR9	FR8
R	R	R	R	R	R	R	R
7	6	5	4	3	2	1	0
FR7	FR6	FR5	FR4	FR3	FR2	FR1	FR0
R	R	R	R	R	R	R	R

Table 31-41 HcFmRemaining register contents

Bit position	Bit name	Function
31	FRT	Frame Remaining Toggle This bit is used to synchronize the frame setting values of the host controller driver (HCD) and OHCI host controller. The OHCI host controller copies the FIT field value to this bit when the FR field is set to 0_H and the FI field value is reloaded. The HCD can check whether the value set to the FI field has been applied to the FR field by comparing the FIT bit and FRT bit values.
13:0	FR [13:0]	Frame Remaining Indicates the current frame value. The value of this field is decremented as time elapses. Because the frame value is reloaded when the value of FI field becomes 0_H, the FR field value is copied to this field and decrementing starts again.

(16) HcFmNumber register (Offset: $3C_H$)

Access This register is read-only, in 32-bit units.

Address <USHA0_base_OHCI> + $3C_H$

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8
Frame Number15	Frame Number14	Frame Number13	Frame Number12	Frame Number11	Frame Number10	Frame Number9	Frame Number8
R	R	R	R	R	R	R	R
7	6	5	4	3	2	1	0
Frame Number7	Frame Number6	Frame Number5	Frame Number4	Frame Number3	Frame Number2	Frame Number1	Frame Number0
R	R	R	R	R	R	R	R

Table 31-42 HcFmNumber register contents

Bit position	Bit name	Function
15:0	Frame Number [15:0]	Indicates the number of frames that have been transferred. This field is incremented when the FR field value becomes 0_H.

(17) HcPeriodicStart register (Offset: 40_H)

Access This register can be read or written in 32-bit units.

Address <USHA0_base_OHCI> + 40_H

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8
0	0	Periodic Start13	Periodic Start12	Periodic Start11	Periodic Start10	Periodic Start9	Periodic Start8
R	R	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
Periodic Start7	Periodic Start6	Periodic Start5	Periodic Start4	Periodic Start3	Periodic Start2	Periodic Start1	Periodic Start0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 31-43 HcPeriodicStart register contents

Bit position	Bit name	Function
31:14	–	Reserved (Be sure to write 0 to these bits.)
13:0	Periodic Start[13:0]	These bits are used to determine the transfer ratio of periodic and async lists in a PeriodicStart field frame. This field must be set by the host controller driver (HCD) during initialization of the OHCI host controller. If the value of the FR field is larger than the value set to this field, the non-periodic list takes precedence over the periodic list. The OHCI Specification recommends setting this value to about 10% of the FI field value, and the relevant value is $3E67_H$.

(18) HcLSThreshold register (Offset: 44_H)

Access This register can be read or written in 32-bit units.

Address <USHA0_base_OHCI> + 44_H

Initial value 00000628_H. This register is initialized by any reset.

31	30	29	28	27	26	25	24
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8
0	0	0	0	HcLSThres-hold11	HcLSThres-hold10	HcLSThres-hold9	HcLSThres-hold8
R	R	R	R	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
HcLSThres-hold7	HcLSThres-hold6	HcLSThres-hold5	HcLSThres-hold4	HcLSThres-hold3	HcLSThres-hold2	HcLSThres-hold1	HcLSThres-hold0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 31-44 HcLSThreshold register contents

Bit position	Bit name	Function
31:12	–	Reserved (Be sure to write 0 to these bits.)
11:0	HcLSThres-hold[11:0]	Specify the threshold value for whether another transfer can be made before the remaining frame time elapses during an LS transfer. If the FR field value is larger than the value set to this field, an LS transfer can be started.

(19) HRDA (HcRhDescriptorA) register (Offset: 48_H)

Access This register can be read or written in 32-bit units.

Address <USHA0_base_OHCI> + 48_H

Initial value $FF00\ 0902_H$. This register is initialized by any reset.

HRDA

31	30	29	28	27	26	25	24
POTPGT7	POTPGT6	POTPGT5	POTPGT4	POTPGT3	POTPGT2	POTPGT1	POTPGT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8
0	0	0	NOCP	OCPM	DT	NPS	PSM
R	R	R	R/W	R/W	R	R/W	R/W
7	6	5	4	3	2	1	0
NDP7	NDP6	NDP5	NDP4	NDP3	NDP2	NDP1	NDP0
R	R	R	R	R	R	R	R

Table 31-45 HRDA (HcRhDescriptorA) register contents

Bit position	Bit name	Function
31:24	POTPGT [7:0]	Power On To Power Good Time Specify the wait time until the host controller driver (HCD) accesses a root hub powered on (unit: 2 ms). The wait time is calculated by POTPGT field value × 2 ms.
23:13	–	Reserved (Be sure to write 0 to these bits.)
12	NOCP	No Over Current Protection Specify whether to support root hub overcurrent protection. 1: Do not support overcurrent protection. 0: Support overcurrent protection.
11	OCPM	Over Current Protection Mode Specifies how to report the overcurrent state of the root hub. The mode specified for this bit must be the same as the mode specified for the PSM bit. 1: Report the overcurrent state in port units. 0: Report the overcurrent state at all ports simultaneously. The setting of this bit becomes valid only when the NOCP bit is 0.
10	DT	Device Type Indicates that the root hub is not a combined device. This field always returns 0 because the root hub must not be a combined device. This bit is read-only.
9	NPS	No Power Switching Specifies whether to support power switching or whether the power for the ports is constantly on. 1: Power is constantly on while the OHCI host controller is running. 0: Power for ports can be switched.
8	PSM	Power Switching Mode Specifies how to control root hub port power switching. 1: Control the power for ports individually. 0: Control the power for all ports simultaneously. Ports are only controlled by the SPP and CPP bits of the HcRhPortStatus register if the PPCM field of the HcRhDescriptorB register is 1. If this field is cleared, ports are controlled by the SGP and CGP bits. The setting of this bit becomes valid only when the NPS bit is 0.
7:0	NDP[7:0]	Number Downstream Port Specify the number of downstream ports supported by the root hub of the OHCI host controller. This field is fixed to 02_H because the OHCI host controller incorporates two downstream ports. These bits are read-only.

(20) HcRhDescriptorB register (Offset: $4C_H$)

Access This register can be read or written in 32-bit units.

Address <USHA0_base_OHCI> + $4C_H$

Initial value $0006\ 0000_H$. This register is initialized by any reset.

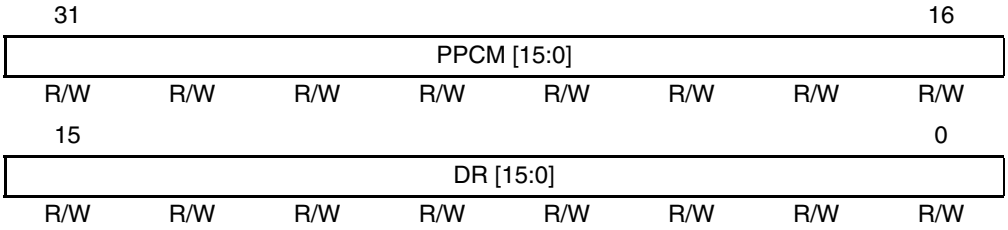


Table 31-46 HcRhDescriptorB register contents

Bit position	Bit name	Function
31:16	PPCM [15:0]	Port Power Control Mask Indicates whether ports are controlled by the Set/ClearGlobalPower. The setting of this bit becomes valid only when the PSM bit is 1. • Field Bit / Description: 0 Reserved; 1 Setting of device connected to port 1.; 2 Setting of device connected to port 2.; 15:3 Reserved • Value 1: Ports are only controlled by the Set/ClearPortPower. 0: Ports are controlled by the Set/ClearGlobalPower.
15:0	DR [15:0]	Device Removable Indicates whether the device connected to the OHCI host controller port is removable. • Field Bit / Description: 0 Reserved; 1 Setting of device connected to port 1.; 2 Setting of device connected to port 2.; 15:3 Reserved • Value 1: The connected device is not removable. 0: The connected device is removable.

(21) HcRhStatus register (Offset: 50_H)

Access This register can be read or written in 32-bit units.

Address <USHA0_base_OHCI> + 50_H

Initial value $0000\ 0000_H$. This register is initialized by any reset.

(a) When read

31	30	29	28	27	26	25	24
CRWE	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
23	22	21	20	19	18	17	16
0	0	0	0	0	0	OCIC	LPSC
R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8
DRWE	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
7	6	5	4	3	2	1	0
0	0	0	0	0	0	OCI	LPS
R	R	R	R	R	R	R	R

(b) When written

31	30	29	28	27	26	25	24
CRWE	0	0	0	0	0	0	0
W	W	W	W	W	W	W	W
23	22	21	20	19	18	17	16
0	0	0	0	0	0	OCIC	SGP
W	W	W	W	W	W	W	W
15	14	13	12	11	10	9	8
SRWE	0	0	0	0	0	0	0
W	W	W	W	W	W	W	W
7	6	5	4	3	2	1	0
0	0	0	0	0	0	OCI	CGP
W	W	W	W	W	W	W	W

Table 31-47 HcRhStatus register contents

Bit position	R/W	Bit name	Function
31	–	CRWE	Clear Remote Wakeup Enable Use this bit to clear the DRWE bit to 0. Writing 1 to this bit clears the DRWE bit to 0. Writing 0 to this bit is ignored. Value 0 is always read from this bit. This bit is write-only.
30:18	–	–	Reserved (Be sure to write 0 to these bits.)
17	–	OCIC	Over Current Indicate Change This bit is used to report a change in the OCI bit. It is set to 1 when the overcurrent state has changed. This bit is cleared to 0 when 1 is written while the OCIC bit is set (1). 1: The overcurrent state has changed. 0: The overcurrent state has not changed.
16	R	LPSC	Local Power Status Change This bit is fixed to 0 because the local power status is not supported.
	W	SGP	Set Global Power Use this bit to turn on the power to all ports in the global power mode. Setting this bit to 1 turns the power on to all ports. When the PSM bit of the HcRhDescriptorA register is set to 1, power is turned on to ports whose PPCM field is cleared (0).
15	R	DRWE	Device Remote Wakeup Enable Indicates whether a connection status change is included in the remote wakeup events. 1: A connection status change is a remote wakeup source. 0: A connection status change is not a remote wakeup source. If a connection status change event occurs while this bit is 1, the state changes from USB Suspend to USB Resume, and a Resume Detected interrupt occurs.
	W	SRWE	Set Remote Wakeup Enable Use this bit to set the DRWE bit to 1. Writing 1 to this bit sets the DRWE bit to 1. Writing 0 to this bit is ignored.
14:2	–	–	Reserved (Be sure to write 0 to these bits.)
1	–	OCI	Over Current Indicator This bit is used to report the overcurrent state in the global overcurrent detection mode. 1: The port is in the overcurrent state. 0: Ports are in the normal state. This bit is fixed to 0 when overcurrent is reported in port units. This bit is read-only.
0	R	LPS	Local Power Status This bit is fixed to 0 because the local power status is not supported.
	W	CGP	Clear Global Power Use this bit to turn off the power for all ports in the global power mode. Setting this bit to 1 turns off the power to all ports. When the PSM bit of the HcRhDescriptorA register is set to 1, the power is turned off to ports whose PPCM field is cleared (0).

(22) HcRhPortStatus register 1 (Offset: 54_H)

Access These registers can be read or written in 32-bit units.

Address <USHA0_base_OHCI> + 54_H

Initial value $0000\ 0000_H$. These registers are initialized by any reset.

(a) When read

31	30	29	28	27	26	25	24
0	0	0	0	0	0	0	0
R	R	R	R	R	R	R	R
23	22	21	20	19	18	17	16
0	0	0	PRSC	OCIC	PSSC	PESC	CSC
R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8
0	0	0	0	0	0	LSDA	PPS
R	R	R	R	R	R	R	R
7	6	5	4	3	2	1	0
0	0	0	PRS	POCI	PSS	PES	CSC
R	R	R	R	R	R	R	R

(b) When written

31	30	29	28	27	26	25	24
0	0	0	0	0	0	0	0
W	W	W	W	W	W	W	W
23	22	21	20	19	18	17	16
0	0	0	PRSC	OCIC	PSSC	PESC	CSC
W	W	W	W	W	W	W	W
15	14	13	12	11	10	9	8
0	0	0	0	0	0	CPP	SPP
W	W	W	W	W	W	W	W
7	6	5	4	3	2	1	0
0	0	0	SPR	CSS	SPS	SPE	CPE
W	W	W	W	W	W	W	W

Table 31-48 HcRhPortStatus register 1 contents (1/3)

Bit position	R/W	Bit name	Function
30:21	–	–	Reserved (Be sure to write 0 to these bits.)
20	–	PRSC	Port Reset Status Change Indicates the completion of a port reset. 1: The port reset is complete. 0: The port reset status has not changed. This bit is set to 1 when a 10 ms hardware reset ends. This bit is cleared to 0 when 1 is written by the host controller driver (HCD).
19	–	OCIC	Over Current Indicate Change This bit is set to 1 when the overcurrent status is detected at a port. 1: The overcurrent state has changed. 0: The overcurrent state has not changed. This bit is cleared to 0 when 1 is written by the host controller driver (HCD).
18	–	PSSC	Port Suspend Status Change Indicates that the resume sequence has completed. 1: The resume sequence has completed. 0: Port Suspend Status has not changed. This bit is set when all of resume processing is completed by hardware. This bit is cleared to 0 when 1 is written by the host controller driver (HCD).
17	–	PESC	Port Enable Status Change Indicates that the PES bit has been cleared to 0. 1: The PES bit status has changed. (PES cleared) 0: The PES bit status has not changed. This bit is set to 1 when the PES bit is cleared to 0 because the port state is changed from Enable to Disable due to a hardware event such as an overcurrent state, device disconnection, power-off, or bubble error. This bit is cleared (0) when 1 is written by the host controller driver (HCD).
16	–	CSC	Connect Status Change Indicates that the CCS bit status has changed. 1: The CCS bit status has changed. 0: The CCS bit status has not changed. This bit is set to 1 when the CCS bit status is changed due to connection or disconnection of a USB device. This bit is set to 1 when a request for port reset, port suspend, or port enable is issued while a USB device is disconnected, so that the driver can re-evaluate the device connection. This bit is cleared to 0 when 1 is written by the host controller driver (HCD).
15:10	–	–	Reserved (Be sure to write 0 to these bits.)
9	R	LSDA	Low Speed Device Attached Indicates the speed of the device connected to the USB port. 1: A low-speed device is connected. 0: A full-speed device is connected. The bit is enabled only when the CCS bit is set.
	W	CPP	Clear Port Power Use this bit to turn off the power to the ports. Writing 1 to this bit turns the port power off. Writing 0 to this bit is ignored.

Table 31-48 HcRhPortStatus register 1 contents (2/3)

Bit position	R/W	Bit name	Function
8	R	PPS	Port Power Status Indicates the port power status. 1: Port power on 0: Port power off The control method depends on the power switch time.
	W	SPP	Set Port Power Use this bit to turn on the power to a port when the power is controlled in port units. Writing 1 to this bit turns the port power on. Writing 0 to this bit is ignored.
7:5	–	–	Reserved (Be sure to write 0 to these bits.)
4	R	PRS	Port Reset Status Indicates whether a downstream port is being reset or not. 1: The port is being reset. 0: The port is not being reset. This bit is cleared to 0 together with setting of the PRSC bit when a 10 ms port reset ends. This bit cannot be set when the CSC bit has been cleared to 0 (no device connected).
	W	SPR	Set Port Reset Use this bit to issue a port reset request to a downstream port. Writing 1 to this bit starts a 10 ms port reset. When 1 is written to this bit while the CCS bit is 0, the CSC bit is set and it is reported to the host controller driver that an attempt was made to reset the disconnected port. Writing 0 to this bit is ignored.
3	R	POCI	Port Over Current Indicator Indicates the overcurrent status at a downstream port. 1: The port is in the overcurrent state. 0: The port is in the normal state.
	W	CPS	Clear Port Suspend Use this bit to exit the suspend mode and start the resume sequence. Writing 1 to this bit starts the resume sequence. Writing 0 to this bit is ignored. The resume sequence starts only when the PSS bit has been set.
2	R	PSS	Port Suspend Status Indicates that ports are in the suspend state or the resume sequence is being executed. 1: Ports are in the suspend state. 0: Ports are in the normal transfer state. This bit cannot be set when the CCS bit has been cleared to 0 (no device connected). Writing to the SPS bit sets ports to the suspend state. This bit is cleared to 0 when the resume sequence ends, port reset ends, or when the state shifts to USB Resume.
	W	SPS	Set Port Suspend Use this bit to shift the port status to the suspend state. When 1 is written to this bit, ports enter the suspend state. Writing 0 to this bit is ignored. Writing to the SPS bit sets ports to the suspend state. If this bit is written when the CCS bit has been cleared to 0, the CSC bit is set and it is reported to the host controller driver that an attempt was made to suspend the disconnected port.

Table 31-48 HcRhPortStatus register 1 contents (3/3)

Bit position	R/W	Bit name	Function
1	R	PES	Port Enable Status Indicates whether ports are enabled or disabled. 1: Ports are enabled. 0: Ports are disabled. This bit cannot be set when the CSC bit has been cleared to 0 (no device connected). Ports are enabled after a port reset ends. This bit is cleared to 0 by hardware when an event such as an overcurrent state, device disconnection, power-off, or bubble error is detected.
	W	SPE	Set Port Enable Use this bit to set the PES bit to 1. Writing 0 to this bit is ignored. Change the port status by setting the SPR bit. In this USB host controller, ports cannot be enabled by using this bit because the USB Specification does not support this feature, even though the OHCI Specification does.
0	R	CCS	Current Connect Status Indicates the current connection status at a downstream port. 1: A device is connected. 0: No device is connected.
	W	CPE	Clear Port Enable Use this bit to clear the CPE bit. When 1 is written to this bit, ports are disabled. Writing 0 to this bit is ignored.

31.7.4 Interruption from USB host controller

The USB host controller collects and sorts interrupts sent from the OHCI host controller and reports them to the system as one of the following three types of interrupts:

Table 31-49 Interruption from USB host controller

Signals that report interrupts to the system	Interrupt report signal generated by OHCI host controller
INTUSBH0	USBH status interrupt (INTA, SMMI, or PME)
INTUSBH1	USBH PCI cycle error
INTUSBH2	USBH PME interrupt

The details of each interrupt are described below.

(1) USB status interrupt (INTUSBH0)

(a) Interrupt routing

The OHCI host controller reports an occurrence of the INTA or SMMI interrupt to the system, according to the setting of the IR bit of the OHCI operational register HcControl. After a reset, INTA is routed to report the interrupt by the initial setting of the IR bit.

INTA and SMMI are used as the interrupt sources, except for Ownership Change requests.

Table 31-50 Routing of interrupts INTA and SMMI

IR bit of HcControl register	Interrupt report signal
0	INTA (initial value)
1	SMMI

To use the INTA and SMMI interrupt, the inta_en and int_smmi_en bits of the PCI interrupt control register of PCI host bridge registers must be set to 1.

(b) Interrupt sources

Interrupts defined in the OHCI Specification are supported.

Use the HcInterruptEnable register to specify interrupt sources reported to the system. Interrupts are reported via the route specified by using the IR bit.

The interrupt sources are shown below.

Table 31-51 Sources of interrupts INTA and SMMI

<table>
<tr><th>Interrupt source</th><th>Function</th></tr>
<tr><td>Scheduling Overrun</td><td>Indicates that USB scheduling has overrun in a frame.</td></tr>
<tr><td>Writeback DoneHead</td><td>Indicates that the USB host controller has received the TD and written it back.</td></tr>
<tr><td>Start Of Frame</td><td>Indicates the update of HccaFmNumber upon a start of a frame.</td></tr>
<tr><td>Resume Detected</td><td>Indicates that a resume signal sent from a USB device was detected.</td></tr>
<tr><td>Unrecoverable Error</td><td>Indicates that an error not related to the USB (PCI abort) was detected.</td></tr>
<tr><td>Frame Number Overflow</td><td>Indicates that the FrameNumber15 bit of the HcFmNumber register has changed (from 0 to 1 or 1 to 0).</td></tr>
<tr><td>Root Hub Status Change</td><td>Indicates that the HcRhStatus or HcRhPortStatus status has changed.
This interrupt source is classified based on the following event details:
<table>
<tr><th>OverCurrentIndicateChange</th><th>Indicates that an overcurrent occurred.</th></tr>
<tr><td>Connect Status Change</td><td>Indicates that a device has been connected to or disconnected from the USB bus.</td></tr>
<tr><td>Port EnableStatusChange</td><td>Indicates that the port is disabled due to a USB error.</td></tr>
<tr><td>Port Suspend Status Change</td><td>Indicates that the resume sequence has finished.</td></tr>
<tr><td>Port Reset Status Change</td><td>Indicates that a USB reset has finished.</td></tr>
</table></td></tr>
<tr><td>Ownership Change</td><td>Indicates that an ownership request has been issued.
This interrupt is reported to SMMI only.</td></tr>
</table>

(2) PCI cycle error interrupt (INTUSBH1)

The PCI cycle error interrupt (INTUSBH1) occurs when a data access that causes an error is performed for the OHCI or PCI bus bridge register area (in 32-bit access). This interrupt is used during debugging; it is not used during normal operation.

(3) PME interrupt (INTUSBH2)

The PME interrupt is an interrupt signal for power management, which can be used to report the USB bus status change to the system when PCLK is not supplied.

The following lists the events that might occur on the USB bus and whether each interrupt is supported:

Table 31-52 PME interrupt sources

Event	Occurrence of interrupts
Over Current Indicate	Does not occur.
Connect	Occurs.
Disconnect	Occurs.
Resume (RemoteWakeUp)	Occurs.

To use the PME interrupt, the PME Enable bit of the Power Management Control/Status register in OHCI host configuration registers and the int_pme_en bit of the PCI interrupt control register in PCI host bridge registers must be set to 1 in advance.

Chapter 32 Ethernet Controller

32.1 General

The Ethernet controller includes a 10/100 Mbps Ethernet Media Access Controller (MAC) conforming to IEEE802.3, a FIFO controller for flow control, and a checksum calculation unit (only for received packets) conforming to RFC1071.

32.1.1 V850E2/MN4 Ethernet controller features

Instances

This microcontroller has following number of instances of the Ethernet controllers.

Table 32-1 Instances of Ethernet controller

Ethernet controller	μPD70F3510	μPD70F3512, 70F3514, 70F3515
Number of instances	Not integrated	1

Register addresses

All Ethernet controller register addresses are given as addresses offset from the individual base address <ETHA0_base_USER> or <ETHA0C_base>. The base addresses <ETHA0_base> and <ETHA0C_base> of each Ethernet controller are listed in the following table:

Table 32-2 Register base addresses

Ethernet function	Base address	Address
MAC control register	ETHA0_base	F993 2000_H
Statistics counter	ETHA0_base	F993 2000_H
FIFO controller control register	ETHA0_base	F993 2000_H
Ethernet controller dedicated DMAC control register	ETHA0_base	F993 2000_H
Transmission checksum dedicated DMAC control register	ETHA0C_base	F993 3000_H

Interrupts The Ethernet controller can generate the following interrupt requests:

Table 32-3 Ethernet controller interrupt requests

Ethernet controller signals	Function	Connected to
INTETMRQ	Ethernet receive data ready interrupt	Interrupt controller INTETHA0SRX
INTETMRX	Ethernet packet reception interrupt	Interrupt controller INTETHA0SCRX DTS controller trigger 120
INTETMTX	Ethernet packet transmission interrupt	Interrupt controller INTETHA0SCTX DTS controller trigger 121
INTETMFS	Ethernet FIFO status interrupt	Interrupt controller INTETHA0FS
INTETMTS	Ethernet transmission status interrupt	Interrupt controller INTETHA0TS
INTETMRS	Ethernet reception status interrupt	Interrupt controller INTETHA0RS
INTETMOV	Ethernet MAC interrupt	Interrupt controller INTETHA0MAC
IRQSCTX_TCH	Ethernet transmit data calculation completion interrupt	Interrupt controller INTETHA0SCRXTCH
IRQSCRX_TCH	Ethernet transmit checksum interrupt	Interrupt controller INTETHA0SCTXTCH

Caution If an interrupt from the the Ethernet controller is detected, the value of the interrupt request is stored in the EICn.EIRFn bit (n = 178 to 186). Therefore, even if the CPU acknowledges the interrupt and the EIRFn bit of the interrupt control register (EICn) is automatically cleared to 0, the EICn.EIRFn bit is immediately set to 1 and an interrupt is continuously generated. To prevent this from happening, clear the status flag of each interrupt request in the interrupt service routine, and then forcibly clear the EICn.EIRFn bit to 0. For Ethernet reception data ready interrupts, after the received packet is read, forcibly clear the EIC178.EIRF178 bit to 0.

Assignment of descriptor and data buffer Descriptors and data buffers can be assigned to the following areas:

- H-bus shared memory
- External memory connected to the secondary memory controller

32.1.2 Functions

(1) MAC

- 10/100 Mbps full-duplex communication, half-duplex communication, and flow control conforming to IEEE802.3 supported
- MII supported as physical layer device (PHY) interface
- Accessing PHY registers via serial management interface supported
- Statistics counter to support RMON/SNMP (RFC2665, RFC2819)
- Packet filtering based on address types
- VLAN frame detection

(2) FIFO

- Transmit/receive FIFO size: Transmit FIFO = 2 KB, receive FIFO = 2 KB
- FIFO status register
- Interrupts generated according to transmission/reception status and FIFO status

(3) DMAC for Ethernet controller

- Data transfer (DMA)
- Reception status DMA transfer
- Reading (in pointer chain format), analyzing, and writing back buffer descriptors
- Controlling interrupts in packet transfers

(4) Checksum calculation

- Transmit checksum calculation function conforming to RFC1071

  DMAC for transmit checksum can calculate multiple checksums successively and save the result to any address.
- Receive checksum calculation conforming to RFC1071

  The MAC header and FCS of a received packet are automatically identified and the checksum for verifying the received packet (excluding the dummy header) is generated.

32.2 Configuration

32.2.1 System configuration

The Ethernet controller transmits and receives data by using a dedicated direct memory access controller (DMAC). The Ethernet controller supports the MII (Media Independent Interface) of IEEE802.3 and can create a 10 Mbps or 100 Mbps Ethernet environment when it is connected to a PHY device conforming to MII. In addition, data can be communicated in full-duplex or half-duplex mode, which can be selected.

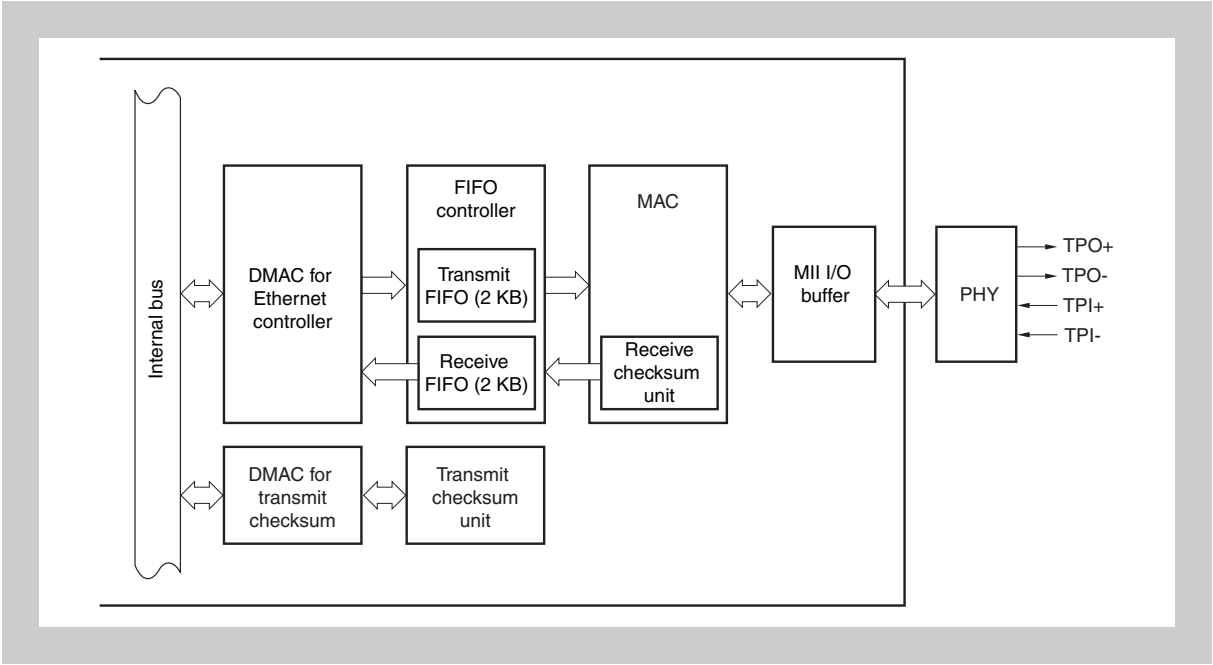


Figure 32-1 Configuration of Ethernet controller

(1) MAC

This unit has MAC functions and supports MII-based interfacing with an external PHY device.

- Receive checksum unit
  This unit calculates the receive checksum.

(2) FIFO controller

This unit controls the transmit/receive FIFO buffers.

2 KB FIFO buffers are separately available for transmission and reception.

(3) DMAC for Ethernet controller

This DMA controller controls data transmission and reception with the internal bus.

Caution The DMAC for the Ethernet controller processes all the data the Ethernet controller transmits and receives. Data cannot be transmitted or received in packet units by reading or writing a register.

(4) Transmit Checksum DMAC

This DMA controller that interfaces with the internal bus is dedicated to the transmit checksum function.

(5) Transmit checksum unit

This unit can only calculate the transmit checksum. Independent of the data transmission/reception interface, the transmit checksum unit calculates and transfers transmit data by using the transmit checksum DMA function and descriptor.

32.2.2 Interrupt Requests and Sources

The Ethernet controller interrupt requests and their sources are listed below.

Table 32-4 Interrupt requests

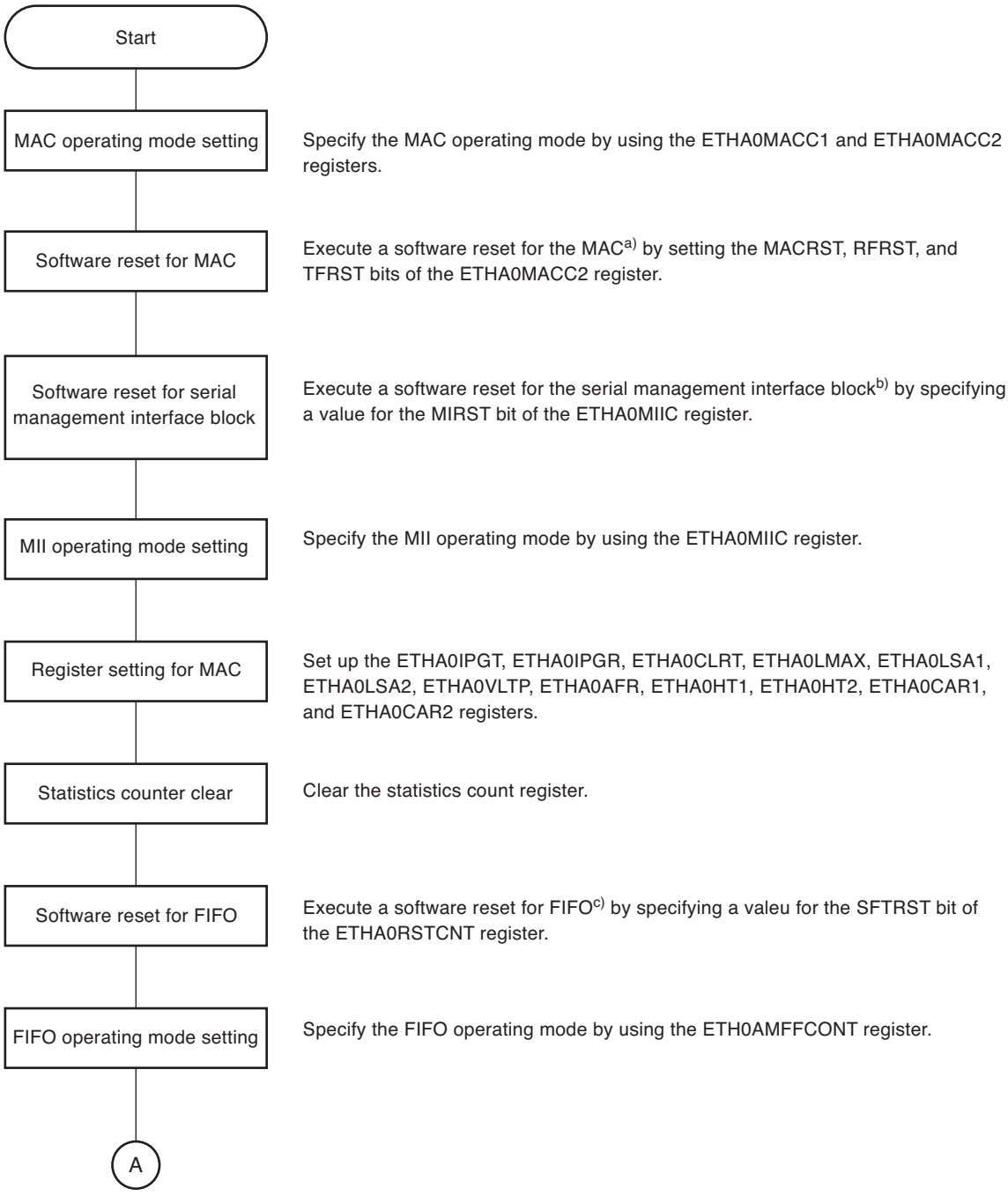
Interrupt request		Interrupt source
INTETHA0SRX	Ethernet receive data ready interrupt	Received packet read request
INTETHA0SCRX	Ethernet packet reception interrupt	Packet reception (DMA) completion interrupt (RXI)
		Reception (DMA) end of chain interrupt (RECI)
		Receive data buffer access error interrupt (RBEI)
		Pause interrupt (RUPI) triggered by the “U” (used) bit in the receive descriptor
INTETHA0SCTX	Ethernet packet transmission interrupt	Packet transmission (DMA) completion interrupt (TXI)
		Transmission (DMA) end of chain interrupt (TECI)
		Transmit data buffer access error interrupt (TBEI)
		Pause interrupt (TUPI) triggered by the “U” (used) bit in the transmit descriptor
INTETHA0FS	Ethernet FIFO status interrupt	FIFO status (ETHA0FSTATUS) interrupt
INTETHA0TS	Ethernet transmission status interrupt	Transmission status (ETHA0FTXSTATUS) interrupt
INTETHA0RS	Ethernet reception status interrupt	Reception status (ETHA0RXSTATUS) interrupt
INTETHA0MAC	Ethernet MAC interrupt	Statistics counter overflow (CARRY status)
INTETHA0SCTXTCH	Ethernet transmit data calculation completion interrupt	1 transmit checksum calculation completion interrupt (TCH_TXI)
		All transmit checksums calculation completion interrupt (TCH_TECI)
		Transmit checksum buffer access error interrupt (TCH_TBEI)
		Transmit checksum calculation pause interrupt (TCH_RUPI)
INTETHA0SCRXTCH	Ethernet transmit checksum interrupt	1 transmit checksum writing completion interrupt (TCH_RXI)
		All transmit checksums calculation writing completion interrupt (TCH_RECI)
		Data write error interrupt (TCH_RBEI)

- Each interrupt source can be masked. If an interrupt source is generated while the interrupt is masked, the corresponding bit in the status register is set but the interrupt request is not generated.
- It is recommended to read the interrupt register if multiple sources have been generated concurrently.

32.3 Initialization

Use the following procedure to perform initialization before using the V850E2/MN4 Ethernet controller:

1. Enable Ethernet controller operation.
2. Initialize the media access controller (MAC).
3. Initialize the FIFO controller.
4. Initialize the DMA controller.
5. Set up interrupts.

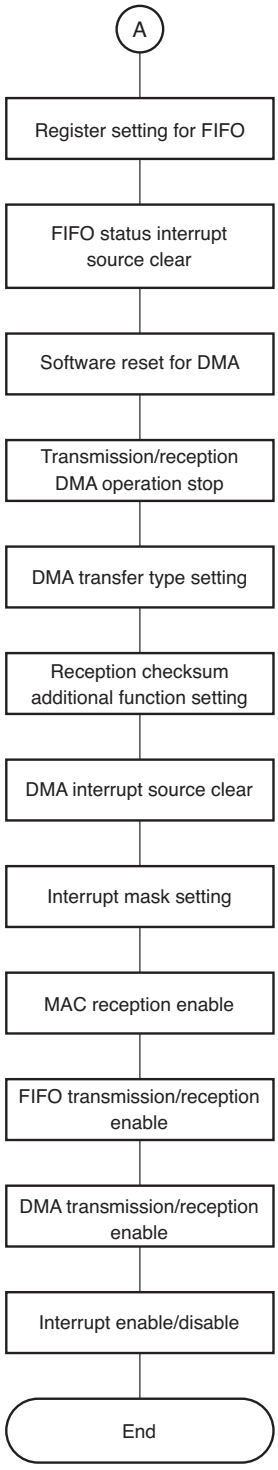


a) To cancel the software reset for the MAC, set the MCRST, RFRST, and TFRST bits of the ETHA0MACC2 register at the same time, and then clear them at the same time. Manipulate these bits at an interval of five or more RXCLK or TXCLK clock cycles.

b) To cancel the software reset for the serial management interface block, set the MIRST bit of the ETHA0MIIC register, and then clear it. Manipulate this bit at an interval of five or more Ethernet control clock cycles (f_{EC}).

c) The SFTRST bit of the ETHA0RSTCNT register is a self-clear bit that is automatically cleared after it is set.

Figure 32-2 Initializing Ethernet controller (1/2)



Set up the ETHA0FLOWTH, ETHA0PAUSETM, ETHA0RXERSEL, ETHA0TXABTCNT, and ETHA0RXABTCNT registers.

Clear the ETHA0FSTATUS, ETHA0TXSTATUS, and ETHA0RXSTATUS registers[a].

Execute a software reset for the DMA controller by specifying a value for the SFTRST bit of the ETHA0SFTRST register[b].

Stop the transmission/reception DMA operation by specifying a value for the TXEN and RXEN bits of the ETHA0TRANSCTL register.
The DMA operating status applied to the TXEN_STA and RXEN_STA bits of the ETHA0TRANSCTL register.

Specify the DMA transfer type for the DMA controller by using the ETHA0DMACM register.

Enable or disable the reception checksum function by specifying a value for the RXCHKSMEN bit of the ETHA0TRANSCTL register.

Clear the ETHA0INTMS register[c].

Specify the interrupt mask settings by using the ETHA0FSTATMK, ETHA0TXSTATMK, ETHA0RXSTATMK, and ETHA0INTMS registers.

Enable the reception data interface function for the MAC by setting the SRXEN bit of the ETHA0MACC1 register.

Enable the transmission/reception function for FIFO by setting the RXEN and TXEN bits of the ETHA0MFFCONT register.

Enable the transmission/reception DMA transfer function for the DMA controller by setting the RXEN and TXEN bits of the ETHA0TRANSCTL register.

Set up the interrupt controller control registers.

a) The ETHA0FSTATUS, ETHA0TXSTATUS, and ETHA0RXSTATUS are read-clear registers.
b) The SFTRST bit of the ETHA0SFTRST register is a self-clear bit that is automatically cleared after it is set.
c) The source bits of the ETHA0INTMS register RBEI, RECI, RXI, TBEI, TECI, and TXI are read-clear bits.

Note For details about the interrupt controller control register, see *Chapter 9 "Interrupt Functions"*.

Figure 32-3 Initializing Ethernet controller (2/2)

32.3.1 Software reset

The Ethernet controller can be initialized by using a software reset if the system has become unstable or to protect against being effects from the previous mode when the operating mode has changed.

When executing a software reset, the MAC control registers are not initialized, but the DMAC registers for the FIFO controller and Ethernet controller are initialized, requiring the operating mode and interrupt mask to be specified again.

Discard packets being used with FIFO when executing a software reset.

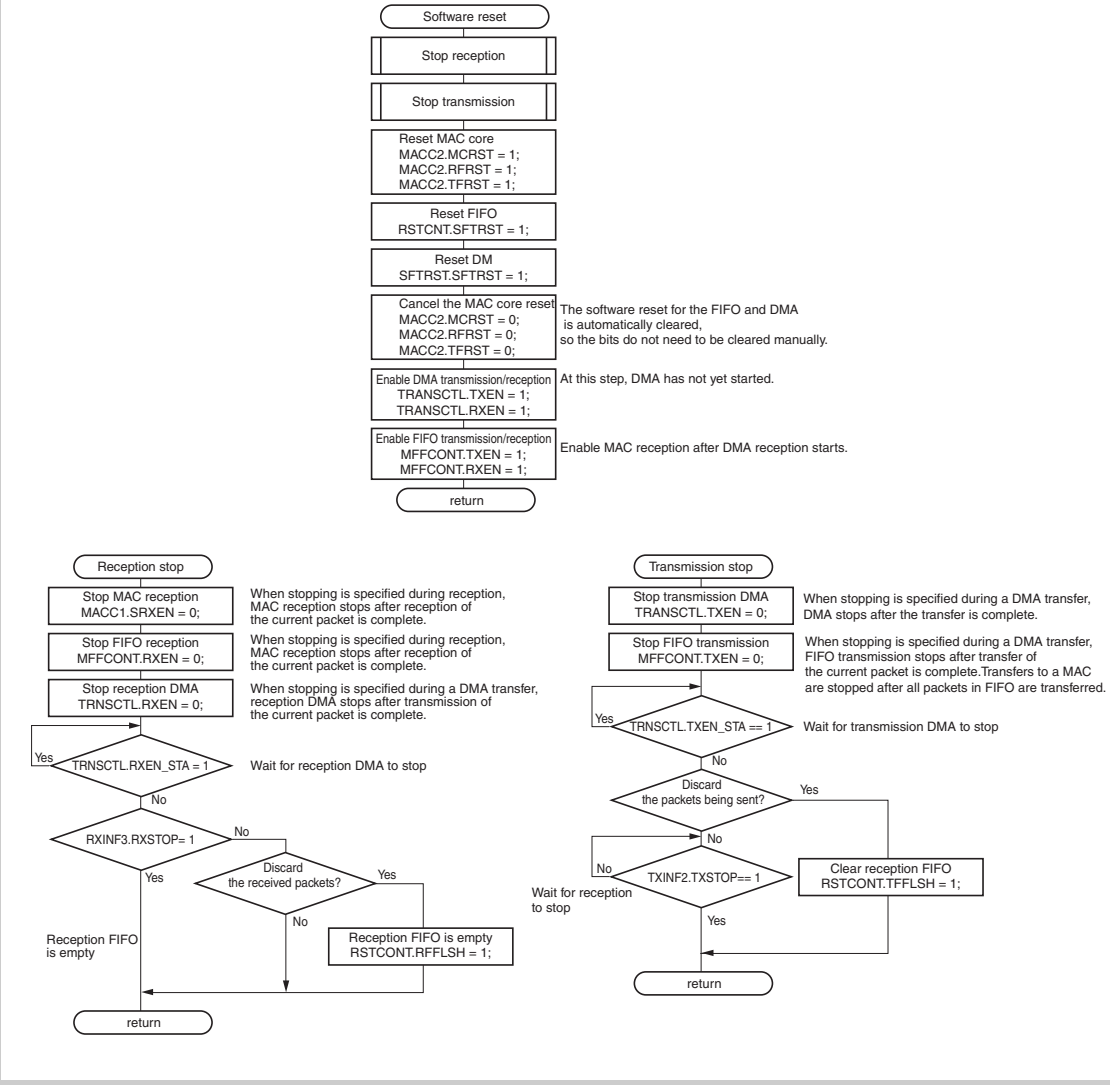


Figure 32-4 Software reset and restart process

32.4 Registers for Controlling the Ethernet Controller

(1) Register setting procedure

To update the values of the control registers, make sure that transmission and reception of frames and DMA are stopped.

If these registers are updated while frames are being transmitted or received, or while DMA is in progress, the operation is not guaranteed.

Caution When using the Ethernet controller, the FIFO controller control register (ETHA0MFFCONT) must be set up first. For details, refer to *(1) "ETHA0MFFCONT - FIFO controller control register" on page 2232 in 32.4.3 "FIFO controller control registers"*.

(2) Register list

Table 32-5 MAC control register list

Offset address	Symbol	Register name	R/W	Bit manipulation unit			Default value
				8	16	32	
0000_H	ETHA0MACC1	MAC setting register 1	R/W			✓	Undefined
0004_H	ETHA0MACC2	MAC setting register 2	R/W			✓	Undefined
0008_H	ETHA0IPGT	Back-to-back IPG register	R/W			✓	Undefined
$000C_H$	ETHA0IPGR	Non back-to-back IPG register	R/W			✓	Undefined
0010_H	ETHA0CLRT	Collision register	R/W			✓	Undefined
0014_H	ETHA0LMAX	Maximum packet length register	R/W			✓	Undefined
0054_H	ETHA0LSA1	Station address register 1	R/W			✓	Undefined
0058_H	ETHA0LSA2	Station address register 2	R/W			✓	00000000_H
$005C_H$	ETHA0PTVR	Pause timer value read register	R			✓	Undefined
0064_H	ETHA0VLTP	VLAN type register	R/W			✓	Undefined
0080_H	ETHA0MIIC	Serial management interface configuration register	R/W			✓	Undefined
0094_H	ETHA0MCMD	MII command register	W			✓	Undefined
0098_H	ETHA0MADR	MII address register	R/W			✓	Undefined
$009C_H$	ETHA0MWTD	MII write data register	R/W			✓	Undefined
$00A0_H$	ETHA0MRDD	MII read data register	R			✓	Undefined
$00A4_H$	ETHA0MIND	MII indicator register	R			✓	Undefined
$00C8_H$	ETHA0AFR	Address filter register	R/W			✓	Undefined
$00CC_H$	ETHA0HT1	Hash table register 1	R/W			✓	00000000_H
$00D0_H$	ETHA0HT2	Hash table register 2	R/W			✓	00000000_H
$00DC_H$	ETHA0CAR1	Carry register 1	R/W			✓	Undefined
$00E0_H$	ETHA0CAR2	Carry register 2	R/W			✓	Undefined
0130_H	ETHA0CAM1	Carry mask register 1	R/W			✓	Undefined
0134_H	ETHA0CAM2	Carry mask register 2	R/W			✓	Undefined

Table 32-6 Statistics counter register list

Offset address	Symbol	Register name	R/W	Bit manipulation unit			Default value
				8	16	32	
0140_H	ETHA0RBYT	Reception byte counter	R/W			✓	00000000_H
0144_H	ETHA0RPKT	Reception packet counter	R/W			✓	00000000_H
0148_H	ETHA0RFCS	Reception FCS error frame counter	R/W			✓	00000000_H
$014C_H$	ETHA0RMCA	Reception multicast packet counter	R/W			✓	00000000_H
0150_H	ETHA0RBCA	Reception broadcast packet counter	R/W			✓	00000000_H
0154_H	ETHA0RXCF	Reception control frame packet counter	R/W			✓	00000000_H
0158_H	ETHA0RXPF	Reception pause frame packet counter	R/W			✓	00000000_H
$015C_H$	ETHA0RXUO	Reception undefined control packet counter	R/W			✓	00000000_H
0160_H	ETHA0RALN	Reception alignment error counter	R/W			✓	00000000_H
0164_H	ETHA0RFLR	Reception frame length error counter	R/W			✓	00000000_H
0168_H	ETHA0RCDE	Reception code error counter	R/W			✓	00000000_H
$016C_H$	ETHA0RFCR	Reception false carrier counter	R/W			✓	00000000_H
0170_H	ETHA0RUND	Reception undersize packet counter	R/W			✓	00000000_H
0174_H	ETHA0ROVR	Reception oversize packet counter	R/W			✓	00000000_H
0178_H	ETHA0RFRG	Reception fragment counter	R/W			✓	00000000_H
$017C_H$	ETHA0RJBR	Reception jabber counter	R/W			✓	00000000_H
0180_H	ETHA0R64	Receive 64-byte frame counter	R/W			✓	00000000_H
0184_H	ETHA0R127	Receive 65- to 127-byte frame counter	R/W			✓	00000000_H
0188_H	ETHA0R255	Receive 128- to 255-byte frame counter	R/W			✓	00000000_H
$018C_H$	ETHA0R511	Receive 256- to 511-byte frame counter	R/W			✓	00000000_H
0190_H	ETHA0R1K	Receive 512- to 1023-byte frame counter	R/W			✓	00000000_H
0194_H	ETHA0RMAX	Receive 1024- to RMAX-byte frame counter	R/W			✓	00000000_H
0198_H	ETHA0RVBT	Receive valid byte counter	R/W			✓	00000000_H
$01C0_H$	ETHA0TBYT	Transmission byte counter	R/W			✓	00000000_H
$01C4_H$	ETHA0TPKT	Transmission packet counter	R/W			✓	00000000_H
$01C8_H$	ETHA0TFCS	Transmission FCS error frame counter	R/W			✓	00000000_H
$01CC_H$	ETHA0TMCA	Transmission multicast packet counter	R/W			✓	00000000_H
$01D0_H$	ETHA0TBCA	Transmission broadcast packet counter	R/W			✓	00000000_H
$01D4_H$	ETHA0TUCA	Transmission unicast packet counter	R/W			✓	00000000_H
$01D8_H$	ETHA0TXPF	Transmission pause control frame counter	R/W			✓	00000000_H
$01DC_H$	ETHA0TDFR	Transmission delay packet counter	R/W			✓	00000000_H
$01E0_H$	ETHA0TXDF	Transmission excessive delay packet counter	R/W			✓	00000000_H
$01E4_H$	ETHA0TSCL	Transmission single collision packet counter	R/W			✓	00000000_H
$01E8_H$	ETHA0TMCL	Transmission multiple collision packet counter	R/W			✓	00000000_H
$01EC_H$	ETHA0TLCL	Transmission late collision packet counter	R/W			✓	00000000_H
$01F0_H$	ETHA0TXCL	Transmission excessive collision packet counter	R/W			✓	00000000_H
$01F4_H$	ETHA0TNCL	Transmission total collision counter	R/W			✓	00000000_H
$01F8_H$	ETHA0TCSE	Transmission carrier sense error counter	R/W			✓	00000000_H
$01FC_H$	ETHA0TIME	MAC internal error counter	R/W			✓	00000000_H

Table 32-7 FIFO controller register list

Offset address	Symbol	Register name	R/W	Bit manipulation unit			Default Value
				8	16	32	
0200_H	ETHA0MFFCONT	FIFO controller control register	R/W			✓	Undefined
0204_H	ETHA0RSTCNT	Software reset control register	R/W			✓	Undefined
0218_H	ETHA0FLOWTH	Flow control threshold value register	R/W			✓	Undefined
$021C_H$	ETHA0PAUSETM	Pause timer value register	R/W			✓	$7FFFFFFF_H$
0220_H	ETHA0RXERSEL	Receive error selection register	R/W			✓	Undefined
0230_H	ETHA0TXSTMONI1	Transmission status monitor 1 register	R			✓	Undefined
0234_H	ETHA0TXSTMONI2	Transmission status monitor 2 register	R			✓	00000000_H
0238_H	ETHA0TXFINF1	Transmission status 1 register	R			✓	Undefined
$023C_H$	ETHA0TXFINF2	Transmission status 2 register	R			✓	Undefined
0240_H	ETHA0RXSTMONI	Reception status monitor register	R			✓	00000000_H
0244_H	ETHA0RXFINF1	Reception status 1 register	R			✓	Undefined
0248_H	ETHA0RXFINF2	Reception status 2 register	R			✓	Undefined
$024C_H$	ETHA0RXFINF3	Reception status 3 register	R			✓	Undefined
0250_H	ETHA0FSTATUS	FIFO status interrupt register	R			✓	Undefined
0254_H	ETHA0FSTATMK	FIFO status interrupt mask register	R/W			✓	Undefined
0258_H	ETHA0TXSTATUS	Transmission status interrupt register	R			✓	Undefined
$025C_H$	ETHA0TXSTATMK	Transmission status interrupt mask register	R/W			✓	Undefined
0260_H	ETHA0RXSTATUS	Reception status interrupt register	R			✓	Undefined
0264_H	ETHA0RXSTATMK	Reception status interrupt mask register	R			✓	Undefined
0270_H	ETHA0TXABTCNT	TX abort counter	R/W			✓	Undefined
0274_H	ETHA0RXABTCNT	RX abort counter	R			✓	Undefined

Table 32-8 DMAC for Ethernet controller register list

Offset address	Symbol	Register name	R/W	Bit manipulation unit			Default value
				8	16	32	
0300_H	ETHA0MODE	Core function setting register	R/W			✓	Undefined
0304_H	ETHA0INTMS	Interrupt control register	R/W			✓	Undefined
0308_H	ETHA0TRANSCTL	Transfer control register	R/W			✓	Undefined
$030C_H$	ETHA0SFTRST	Software reset control register	R/W			✓	Undefined
0310_H	ETHA0DMACM	DMA mode control register	R/W			✓	Undefined
0320_H	ETHA0RXDP	Receive descriptor pointer register	R/W			✓	$FFFFFFFC_H$
0324_H	ETHA0LSTRXDP	Last receive descriptor pointer register	R			✓	$FFFFFFFC_H$
0328_H	ETHA0TXDP	Transmit descriptor pointer register	R/W			✓	$FFFFFFFC_H$
$032C_H$	ETHA0LSTTXDP	Last transmit descriptor pointer register	R			✓	$FFFFFFFC_H$

Table 32-9 DMAC control for transmit checksum register list

Offset address	Symbol	Register name	R/W	Bit manipulation unit			Default value
				8	16	32	
0300_H	ETHA0CMODE	Transmit checksum unit function setting register	R/W			✓	Undefined
0304_H	ETHA0CINTMS	Transmit checksum interrupt register	R/W			✓	Undefined
0308_H	ETHA0CTRANSCTL	Transmit checksum transfer control register	R/W			✓	Undefined
$030C_H$	ETHA0CSFTRST	Transmit checksum software reset register	R/W			✓	Undefined
0310_H	ETHA0CDMACM	Transmit checksum DMA control mode setting register	R/W			✓	Undefined
0320_H	ETHA0CRXDP	Transmit checksum receive descriptor pointer register	R/W			✓	$FFFFFFFC_H$
0324_H	ETHA0CLSTRXDP	Transmit checksum last receive descriptor pointer register	R/W			✓	$FFFFFFFC_H$
0328_H	ETHA0CTXDP	Transmit checksum transmit descriptor pointer register	R/W			✓	$FFFFFFFC_H$
$032C_H$	ETHA0CLSTTXDP	Transmit checksum last transmit descriptor pointer register	R/W			✓	$FFFFFFFC_H$

32.4.1 MAC control registers

(1) ETHA0MACC1 - MAC setting register

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + 0000_H

Initial value Undefined

Cautions

1. Be sure to execute a software reset after setting the operation mode. To execute a software reset, set the MCRST, RFRST, and TFRST bits of the ETHA0MACC2 register at the same time. Cancel the software reset by clearing these bits at the same time.
2. Be sure to set bits 31 to 15, 13, 12, and 4 to “0”.

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
–	MACLB	–	–	TXFC	RXFC	SRXEN	PARF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
PUREP	FLCHT	NOBO	–	CRCEN	PADEN	FULLD	HUGEN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-10 ETHA0MACC1 register contents (1/2)

Bit position	Bit name	Function
14	MACLB	MAC loopback 0: Disables loopback operation. 1: Operation loops back from transmission block to reception block in MAC. To execute the loopback operation, set the FULLD bit to enable full-duplex operation.
11	TXFC	Transmission flow control enable 0: Disables transmission of a pause control frame executed by inputting the TPCF signal. 1: Enables transmission of a pause control frame executed by inputting the TPCF signal.
10	RXFC	Reception flow control enable 0: A pause operation is not executed. 1: A pause operation is executed for the pause period set to the pause timer. The value of the pause timer is updated, regardless of the setting of this bit, when a valid pause control frame is received.

Table 32-10 ETHA0MACC1 register contents (2/2)

Bit position	Bit name	Function
9	SRXEN	Reception enable 0: Reception is disabled. 1: The function of the reception data interface is enabled. If the setting of this bit is changed while the CRS signal is asserted, the new setting becomes valid after the CRS signal has been deasserted, regardless of the setting of the FULLD bit.
8	PARF	Control packet pass 0: A control frame is judged as a control frame. 1: No received packet, including a control frame, is judged as a control frame. The value of the pause timer is not updated even if a valid pause control frame is received, regardless of the setting of the RXFC bit.
7	PUREP	Pure preamble 0: The data of a preamble is not checked. 1: A reception status interrupt is generated if an illegal preamble is detected.
6	FLCHT	Length field check 0: The length field is not checked. 1: The value of the length field and data field length are checked, and a status interrupt is generated.
5	NOBO	No backoff 0: Packets are transmitted by using the backoff algorithm. 1: Packets are always transmitted without using the backoff algorithm.
3	CRCEN	CRC appending 0: CRC is not appended. The end of the transmitted packet must be a valid frame check sequence (FCS). The MAC checks the FCS, and, if the FCS value is not correct, the MAC generates a transmission status interrupt (ETHA0TXSTATUS) to report an error. 1: CRC is automatically appended to the end of a packet. An internally generated frame check sequence (FCS) is appended to the end of the transmit packet.
2	PADEN	PAD appending 0: PAD is not appended. 1: PAD is appended to packets that are less than 64 bytes long. At this time, CRC is automatically appended to the end of the packet regardless of the setting of the CRCEN bit.
1	FULLD	Full-duplex enable 0: Half-duplex operation 1: Full-duplex operation
0	HUGEN	Huge packet enable 0: Transmission/reception of a packet that exceeds the value of the maximum packet length register (ETHA0LMAX) is stopped. 1: Transmission/reception of a packet that exceeds the value of the maximum packet length register (ETHA0LMAX) is not stopped.

(2) ETHA0MACC2 - MAC setting register

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + 0004_H

Initial value Undefined

Cautions

1. Be sure to execute a software reset after setting the operation mode. To execute a software reset, set the MCRST, RFRST, and TFRST bits of the ETHA0MACC2 register at the same time. Cancel the software reset by clearing these bits at the same time.

   Manipulate these reset bits at an interval of five or more RXCLK or TXCLK cycles.

2. Be sure to set bits 31 to 11, 7, and 3 to 0 to “0”.

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
–	–	–	–	–	MCRST	RFRST	TFRST
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
–	BPNB	APD	VPD	–	–	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-11 ETHA0MACC2 register contents

Bit position	Bit name	Function
10	MCRST	MAC control block software reset 0: Cancels a software reset for the MAC control block. 1: Executes a software reset for the MAC control block.
9	RFRST	Reception block software reset 0: Cancels a software reset for the reception block. 1: Executes a software reset for the reception block.
8	TFRST	Transmission block software reset 0: Cancels a software reset for the transmission block. 1: Executes a software reset for the transmission block.
6	BPNB	No backoff after back pressure When this bit is set, backoff is not performed for a transmission after back pressure.
5	APD	Auto VLAN PAD If a packet that matches the VLAN type registered to the ETHA0VLTP register is transmitted, it is treated as a VLAN packet and PAD is appended.
4	VPD	VLAN pad mode The packet to be transmitted is always treated as a VLAN packet and PAD is appended.

(3) ETHA0IPGT - Back-to-back IPG register

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + 0008_H

Initial value Undefined

Caution Be sure to set bits 31 to 7 to “0”.

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
–	IPGT						
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-12 ETHA0IPGT register contents

Bit position	Bit name	Function
6 to 0	IPGT	IPG in back-to-back transmission These bits specify the gap between packets (inter-packet gap (IPG)) in back-to-back transmission. The expression used to calculate the IPG is as follows. • IPG = (5 + IPGT) x time required to transmit 4 bits (Time required to transmit 1 bit = 100 ns when the data rate is 10 Mbps or 10 ns when the data rate is 100 Mbps) Set the IPG to the time required to transmit at least 96 bits to satisfy the specification of IEEE802.3 (refer to *(5) “Inter-packet gap (IPG)” on page 2285 in 32.5.2 “Frame transmission”*).

(4) ETHA0IPGR - Non back-to-back IPG register

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + $000C_H$

Initial value Undefined

Caution Be sure to set bits 31 to 15 and 7 to “0”.

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
–	IPGR1						
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
–	IPGR2						
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-13 ETHA0IPGR register contents

Bit position	Bit name	Function
14 to 8	IPGR1	Carrier sense period These bits specify the carrier sense period of the first half of the IPG in transmission other than back-to-back transmission. The calculation expression used to calculate the carrier sense period is as follows. • Carrier sense period = (2 + IPGR1) x time required to transmit 4 bits Set the carrier sense period to 2/3IPG to satisfy the specification of IEEE802.3 (refer to *(5) “Inter-packet gap (IPG)” on page 2285 in 32.5.2 “Frame transmission”*).
6 to 0	IPGR2	IPG in transmission other than back-to-back transmission These bits specify the IPG in transmission other than back-to-back transmission. The expression used to calculate the IPG is as follows. • IPG = (5 + IPGR2) x time required to transmit 4 bits The carrier sense period specified by the IPGR1 bits is included in the IPG specified by the IPGR2 bits. Set the IPG to the time required to transmit at least 96 bits to satisfy the specification of IEEE802.3 (refer to *(5) “Inter-packet gap (IPG)” on page 2285 in 32.5.2 “Frame transmission”*).

(5) ETHA0CLRT - Collision register

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + 0010_H

Initial value Undefined

Caution Be sure to set bits 31 to 14 and 7 to 4 to “0”.

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
–	–	LCOL					
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
–	–	–	–	RETRY			
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-14 ETHA0CLRT register contents

Bit position	Bit name	Function
13 to 8	LCOL	Collision window These bits specify the collision window width. The width of the collision window to be set is calculated by the following expression. • Collision window width = (LCOL + 8) x time required to transmit 8 bits The IEEE802.3 defines the collision window width as the time required to transmit 512 bits.
3 to 0	RETRY	Maximum number of times of retransmission in case of collision These bits specify the maximum number of times to attempt retransmission when a collision occurs. If retransmission does not finish within the value specified by these bits, transmission is aborted. This value indicates the maximum number of collisions. The IEEE802.3 defines the maximum number of collisions as 15.

(6) ETHA0LMAX - Maximum packet length register

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + 0014_H

Initial value Undefined

Caution Be sure to set bits 31 to 16 to “0”.

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
MAXF15	MAXF14	MAXF13	MAXF12	MAXF11	MAXF10	MAXF9	MAXF8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
MAXF7	MAXF6	MAXF5	MAXF4	MAXF3	MAXF2	MAXF1	MAXF0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-15 ETHA0LMAX register contents

Bit position	Bit name	Function
15:0	MAXF[15:0]	Maximum packet length (bytes) When the ETHA0MACC1.HUGEN bit is 0, the transmit and receive packet length is limited by the value specified by these bits. During reception: Reception is terminated immediately when the receive frame length exceeds the value specified by the MAXF bits. During transmission: Transmission is aborted immediately when the transmit frame length exceeds the value specified by the MAXF bits.

(7) ETHA0LSA1 - Station address register 1

This register is used to compare a source address when a pause control frame is assembled and a destination address when address filtering is used. This register is used in combination with the ETHA0LSA2 register as a 48-bit register.

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + 0054_H

Initial value Undefined

Caution Be sure to set bits 31 to 16 to “0”.

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
LSA115	LSA114	LSA113	LSA112	LSA111	LSA110	LSA19	LSA18
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
LSA17	LSA16	LSA15	LSA14	LSA13	LSA12	LSA11	LSA10
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-16 ETHA0LSA1 register contents

Bit position	Bit name	Function
15:0	LSA1[15:0]	Station address (SA) (47:32) The SA bits (47:0) are used to compare a source address when a pause control frame is assembled and a destination address when address filtering is used (refer to *(a) “Filtering of unicast addresses” on page 2301 in (1) "Overview of address filtering" of 32.5.7 “Address filtering”*).

(8) ETHA0LSA2 - Station address register 2

This register is used to compare a source address when a pause control frame is assembled and a destination address when address filtering is used. This register is used in combination with the ETHA0LSA1 register as a 48-bit register.

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + 0058_H

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
LSA231	LSA230	LSA229	LSA228	LSA227	LSA226	LSA225	LSA224
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
LSA223	LSA222	LSA221	LSA220	LSA219	LSA218	LSA21	LSA216
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
LSA215	LSA214	LSA213	LSA212	LSA211	LSA210	LSA29	LSA28
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
LSA27	LSA26	LSA25	LSA24	LSA23	LSA22	LSA21	LSA20
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-17 ETHA0LSA2 register contents

Bit position	Bit name	Function
31:0	LSA2[31:0]	Station address (SA) (31:0) The SA bits (47:0) are used to compare a source address when a pause control frame is assembled and a destination address when address filtering is used (refer to *(a) “Filtering of unicast addresses” on page 2301 in (1) "Overview of address filtering" of 32.5.7 “Address filtering”*).

(9) ETHA0PTVR - Pause timer value read register

This register is used to read the value of the pause timer counter.

Access This register is read-only, in 32-bit units.

Address <ETHA0_base> + $005C_H$

Initial value Undefined

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
R	R	R	R	R	R	R	R
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8
PTCT15	PTCT14	PTCT13	PTCT12	PTCT11	PTCT10	PTCT9	PTCT8
R	R	R	R	R	R	R	R
7	6	5	4	3	2	1	0
PTCT7	PTCT6	PTCT5	PTCT4	PTCT3	PTCT2	PTCT1	PTCT0
R	R	R	R	R	R	R	R

Table 32-18 ETHA0PTVR register contents

Bit position	Bit name	Function
15:0	PTCT[15:0]	Pause timer counter These bits indicate the value currently set in the pause timer. The value of this register is valid only when reception flow control is enabled (the ETHA0MACC1.RXFC bit is 1) (refer to *(1) "Flow control" on page 2291 in 32.5.4 "MAC control function"*).

(10) ETHA0VLTP - VLAN type register

This register is used to specify the operation to be performed on a VLAN frame.

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + 0064_H

Initial value Undefined

Caution Be sure to set bits 31 to 16 to “0”.

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
VLTP15	VLTP14	VLTP13	VLTP12	VLTP11	VLTP10	VLTP9	VLTP8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
VLTP7	VLTP6	VLTP5	VLTP4	VLTP3	VLTP2	VLTP1	VLTP0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-19 ETHA0VLTP register contents

Bit position	Bit name	Function
15:0	VLTP[15:0]	VLAN frame operation These bits specify the operation to be performed or a VLAN frame (refer to *(3) “Operations related to VLAN frame” on page 2294 in 32.5.4 “MAC control function”*). During reception: The value of VLTP[15:0] is compared with the value of the TPID field (2 bytes following the source address) of a frame to detect a VLAN frame. During transmission: If the value of the VLAN field matches the value of VLTP[15:0] when the ETHA0MACC2.APD bit is 1, PAD is appended to the VLAN frame.

(11) ETHA0MIIC - Serial management interface configuration register

This register is used to set the operation mode of the serial management interface block.

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + 0080_H

Initial value Undefined

Cautions

1. Manipulate the MIRST bit at an interval of five or more Ethernet controller clock cycles (f_{EC}).
2. Be sure to set bits 31 to 16, 14 to 5, and 0 to “0”.

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
MIRST	–	–	–	–	–	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
–	–	–	CLKS2	CLKS1	CLKS0	PHYSEL	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-20 ETHA0MIIC register contents

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>15</td><td>MIRST</td><td>Serial management interface block software reset
0: Cancels a software reset for the serial management interface block.
1: Executes a software reset for the serial management interface block.</td></tr>
<tr><td>4 to 2</td><td>CLKS[2:0]</td><td>MDC division ratio
These bits select a division ratio according to an Ethernet controller clock (f_{EC}) to be used (refer to (a) “MDC clock” on page 2297 in (1) "Overview of serial management interface" of 32.5.6 “Serial management interface”).
To satisfy the specification of IEEE802.3, set a division ratio so that the MDC frequency is 2.5 MHz or less.
<table>
<tr><th>CLKS2</th><th>CLKS1</th><th>CLKS0</th><th>Input frequency of f_{EC}</th></tr>
<tr><td>0</td><td>0</td><td>1</td><td>33 MHz or less (division ratio: 14)</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>50 MHz or less (division ratio: 20)</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>66 MHz or less (division ratio: 28)</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>100 MHz or less (division ratio: 40)</td></tr>
<tr><td colspan="3">Other than above</td><td>Setting prohibited</td></tr>
</table>
</td></tr>
<tr><td>1</td><td>PHYSEL</td><td>MDC output setting
Set this bit if data is not correctly transferred during communication with PHY when the MDC is stopped.
1: The MDC is always output for any frames other than the management frame.
0: The MDC is stopped for frames other than the management frame.</td></tr>
</table>

(12) ETHA0MCMD - MII command register

This register is used to read an external PHY device by using the SCAN command and MII management interface.

Access This register is write-only, in 32-bit units.

The value written to the ETHA0MCMD register must be read from the ETHA0MIND register.

Address <ETHA0_base> + 0094_H

Initial value Undefined

Caution Be sure to set bits 31 to 2 to “0”. Bits 1 and 0 can only be written.

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
W	W	W	W	W	W	W	W
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
W	W	W	W	W	W	W	W
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
W	W	W	W	W	W	W	W
7	6	5	4	3	2	1	0
–	–	–	–	–	–	SCANC	RSTAT
W	W	W	W	W	W	W	W

Table 32-21 ETHA0MCMD register contents

Bit position	Bit name	Function
1	SCANC	SCAN command When this bit is set, the SCAN command is executed.
0	RSTAT	MII management read When this bit is set, the MII management interface reads the external PHY device.

(13) ETHA0MADR - MII address register

This register is used to set a PHY address and a PHY register address.

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + 0098_H

Initial value Undefined

Caution Be sure to set bits 31 to 13 and 7 to 5 to “0”.

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
–	–	–	FIAD				
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
–	–	–	RGAD				
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-22 ETHA0MADR register contents

Bit position	Bit name	Function
12:8	FIAD	PHY address These bits specify a PHY address. One Ethernet controller can control up to 31 PHY devices.
4:0	RGAD	PHY register address These bits specify the address of the register to be accessed. The Ethernet controller can access 32 16-bit registers in one PHY device.

(14) ETHA0MWTD - MII write data register

This register is used to set the data to be written to an external PHY device when the MII management interface writes a PHY device.

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + $009C_H$

Initial value Undefined

Caution Be sure to set bits 31 to 16 to “0”.

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
CTLD15	CTLD14	CTLD13	CTLD12	CTLD11	CTLD10	CTLD9	CTLD8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
CTLD7	CTLD6	CTLD5	CTLD4	CTLD3	CTLD2	CTLD1	CTLD0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-23 ETHA0MWTD register contents

Bit position	Bit name	Function
15:0	CTLD[15:0]	MII write data This is a write data field when the MII management interface writes an external PHY device.

(15) ETHA0MRDD - MII read data register

This register is used to read data that has been read from an external PHY device by the MII management interface.

Access This register is read-only, in 32-bit units.

Address <ETHA0_base> + $00A0_H$

Initial value Undefined

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
R	R	R	R	R	R	R	R
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8
PRSD15	PRSD14	PRSD13	PRSD12	PRSD11	PRSD10	PRSD9	PRSD8
R	R	R	R	R	R	R	R
7	6	5	4	3	2	1	0
PRSD7	PRSD6	PRSD5	PRSD4	PRSD3	PRSD2	PRSD1	PRSD0
R	R	R	R	R	R	R	R

Table 32-24 ETHA0MRDD register contents

Bit position	Bit name	Function
15:0	PRSD[15:0]	MII read data This is a read data field when the MII management interface reads an external PHY device.

(16) ETHA0MIND - MII indicator register

This register indicates the statuses of SCAN command execution and MII management interface access.

Access This register is read-only, in 32-bit units.

Address <ETHA0_base> + $00A4_H$

Initial value Undefined

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
R	R	R	R	R	R	R	R
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
R	R	R	R	R	R	R	R
7	6	5	4	3	2	1	0
–	–	–	–	–	NVALID	SCANA	BUSY
R	R	R	R	R	R	R	R

Table 32-25 ETHA0MIND register contents

Bit position	Bit name	Function
2	NVALID	SCAN command start status 1: The SCAN command is under execution and the first read access has not finished. 0: Normal status
1	SCANA	SCAN command active 1: The SCAN command is under execution. 0: Normal status
0	BUSY	BUSY 1: The MII management interface is accessing an external PHY device. 0: The MII management interface is not accessing an external PHY device.

(17) ETHA0AFR - Address filter register

This register is used to set the conditions under which a receive packet is received.

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + $00C8_H$

Initial value Undefined

Caution Be sure to set bits 31 to 4 to “0”.

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
–	–	–	–	PRO	PRM	AMC	ABC
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-26 ETHA0AFR register contents

Bit position	Bit name	Function
3	PRO	Promiscuous mode All packets are valid in this mode.
2	PRM	Multicast reception In this mode, all multicast packets are valid and other packets are discarded.
1	AMC	Conditional multicast reception In this mode, multicast packets that satisfy the conditions are valid and other packets are discarded. Only multicast packets whose multicast address matches the values of the hash table specified by the ETHA0HT1 and ETHA0HT2 registers are received. The hash table is specified by the ETHA0HT1 and ETHA0HT2 registers.
0	ABC	Broadcast reception In this mode, broadcast packets are valid and other packets are discarded.

For details of the settings of the ETHA0AFR register and the packets to be filtered, refer to *Table 32-114 “Settings of ETHA0AFR register and packets to be filtered” on page 2304*.

(18) ETHA0HT1 - Hash table register 1

This register is used to specify the hash table used for conditional multicast packet detection.

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + $00CC_H$

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
HT131	HT130	HT129	HT128	HT127	HT126	HT125	HT124
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
HT123	HT122	HT121	HT120	HT119	HT118	HT11	HT116
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
HT115	HT114	HT113	HT112	HT111	HT110	HT19	HT18
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
HT17	HT16	HT15	HT14	HT13	HT12	HT11	HT10
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-27 ETHA0HT1 register contents

Bit position	Bit name	Function
31:0	HT1[31:0]	Hash table 1 The hash table is used for conditional multicast packet detection. These bits indicate the higher 32 bits of the hash table. HT (63:32)

(19) ETHA0HT2 - Hash table register 2

This register is used to specify the hash table used for conditional multicast packet detection.

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + $00D0_H$

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
HT231	HT230	HT229	HT228	HT227	HT226	HT225	HT224
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
HT223	HT222	HT221	HT220	HT219	HT218	HT217	HT216
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
HT215	HT214	HT213	HT212	HT211	HT210	HT29	HT28
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
HT27	HT26	HT25	HT24	HT23	HT22	HT21	HT20
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-28 ETHA0HT2 register contents

Bit position	Bit name	Function
31:0	HT2[31:0]	Hash table 2 The hash table is used for conditional multicast packet detection. These bits indicate the lower 32 bits of the hash table. HT (31:0)

(20) ETHA0CAR1 - Carry register 1

This register indicates that a statistics counter has overflowed. Each bit of this register corresponds to a statistics counter. When a statistics counter overflows, the corresponding bit in this register is set.

Each bit is cleared when it is read.

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + $00DC_H$

Initial value Undefined

Caution Be sure to set bits 31 to 16 to “0”.

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
C1VT	C1UT	C1BT	C1MT	C1PT	C1TB	C1MX	C11K
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
C1FE	C1TF	C1OT	C1SF	C1BR	C1MR	C1PR	C1RB
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-29 ETHA0CAR1 register contents

Bit position	Bit name	Function
15	C1VT	Overflow of ETHA0RVBT counter 0: Counter did not overflow. 1: Counter overflowed.
14	C1UT	Overflow of ETHA0TUCA counter 0: Counter did not overflow. 1: Counter overflowed.
13	C1BT	Overflow of ETHA0TBCA counter 0: Counter did not overflow. 1: Counter overflowed.
12	C1MT	Overflow of ETHA0TMCA counter 0: Counter did not overflow. 1: Counter overflowed.
11	C1PT	Overflow of ETHA0TPKT counter 0: Counter did not overflow. 1: Counter overflowed.
10	C1TB	Overflow of ETHA0TBYT counter 0: Counter did not overflow. 1: Counter overflowed.
9	C1MX	Overflow of ETHA0RMAX counter 0: Counter did not overflow. 1: Counter overflowed.
8	C11K	Overflow of ETHA0R1K counter 0: Counter did not overflow. 1: Counter overflowed.
7	C1FE	Overflow of ETHA0R511 counter 0: Counter did not overflow. 1: Counter overflowed.
6	C1TF	Overflow of ETHA0R255 counter 0: Counter did not overflow. 1: Counter overflowed.
5	C1OT	Overflow of ETHA0R127 counter 0: Counter did not overflow. 1: Counter overflowed.
4	C1SF	Overflow of ETHA0R64 counter 0: Counter did not overflow. 1: Counter overflowed.
3	C1BR	Overflow of ETHA0RBCA counter 0: Counter did not overflow. 1: Counter overflowed.
2	C1MR	Overflow of ETHA0RMCA counter 0: Counter did not overflow. 1: Counter overflowed.
1	C1PR	Overflow of ETHA0RPKT counter 0: Counter did not overflow. 1: Counter overflowed.
0	C1RB	Overflow of ETHA0RBYT counter 0: Counter did not overflow. 1: Counter overflowed.

(21) ETHA0CAR2 - Carry register 2

This register indicates that a statistics counter has overflowed. Each bit of this register corresponds to a statistics counter. When a statistics counter overflows, the corresponding bit in this register is set.

Each bit is cleared when it is read.

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + $00E0_H$

Initial value Undefined

Caution Be sure to set bits 30 to 23 to “0”.

31	30	29	28	27	26	25	24
C2DV	–	–	–	–	–	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
–	C2IM	C2CS	C2NC	C2XC	C2LC	C2MC	C2SC
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
C2XD	C2DF	C2XF	C2TE	C2JB	C2FG	C2OV	C2UN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
C2FC	C2CD	C2FO	C2AL	C2UO	C2PF	C2CF	C2RE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-30 ETHA0CAR2 register contents (1/2)

Bit position	Bit name	Function
31	C2DV	Overrunning of status vector 0: Status vector does not overrun. 1: Status vector overruns.
22	C2IM	Overflow of ETHA0TIME counter 0: Counter did not overflow. 1: Counter overflowed.
21	C2CS	Overflow of ETHA0TCSE counter 0: Counter did not overflow. 1: Counter overflowed.
20	C2NC	Overflow of ETHA0TNCL counter 0: Counter did not overflow. 1: Counter overflowed.
19	C2XC	Overflow of ETHA0TXCL counter 0: Counter did not overflow. 1: Counter overflowed.
18	C2LC	Overflow of ETHA0TLCL counter 0: Counter did not overflow. 1: Counter overflowed.
17	C2MC	Overflow of ETHA0TMCL counter 0: Counter did not overflow. 1: Counter overflowed.

Table 32-30 ETHA0CAR2 register contents (2/2)

Bit position	Bit name	Function
16	C2SC	Overflow of ETHA0TSCL counter 0: Counter did not overflow. 1: Counter overflowed.
15	C2XD	Overflow of ETHA0TXDF counter 0: Counter did not overflow. 1: Counter overflowed.
14	C2DF	Overflow of ETHA0TDFR counter 0: Counter did not overflow. 1: Counter overflowed.
13	C2XF	Overflow of ETHA0TXPF counter 0: Counter did not overflow. 1: Counter overflowed.
12	C2TE	Overflow of ETHA0TFCS counter 0: Counter did not overflow. 1: Counter overflowed.
11	C2JB	Overflow of ETHA0TPKT counter 0: Counter did not overflow. 1: Counter overflowed.
10	C2FG	Overflow of ETHA0RFRG counter 0: Counter did not overflow. 1: Counter overflowed.
9	C2OV	Overflow of ETHA0ROVR counter 0: Counter did not overflow. 1: Counter overflowed.
8	C2UN	Overflow of ETHA0RUND counter 0: Counter did not overflow. 1: Counter overflowed.
7	C2FC	Overflow of ETHA0RFCR counter 0: Counter did not overflow. 1: Counter overflowed.
6	C2CD	Overflow of ETHA0RCDE counter 0: Counter did not overflow. 1: Counter overflowed.
5	C2FO	Overflow of ETHA0RFLR counter 0: Counter did not overflow. 1: Counter overflowed.
4	C2AL	Overflow of ETHA0RALN counter 0: Counter did not overflow. 1: Counter overflowed.
3	C2UO	Overflow of ETHA0RXUO counter 0: Counter did not overflow. 1: Counter overflowed.
2	C2PF	Overflow of ETHA0RXPF counter 0: Counter did not overflow. 1: Counter overflowed.
1	C2CF	Overflow of ETHA0RXCF counter 0: Counter did not overflow. 1: Counter overflowed.
0	C2RE	Overflow of ETHA0RFCS counter 0: Counter did not overflow. 1: Counter overflowed.

(22) ETHA0CAM1 - Carry mask register 1

This register is used to mask the INTETMOV signal that is generated when a statistics counter has overflowed and the corresponding bit in the ETHA0CAR1 register has been set.

Each bit of this register can be masked.

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + 0130_H

Initial value Undefined

Caution Be sure to set bits 31 to 16 to “0”.

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
M1VT	M1UT	M1BT	M1MT	M1PT	M1TB	M1MX	M11K
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
M1FE	M1TF	M1OT	M1SF	M1BR	M1MR	M1PR	M1RB
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-31 ETHA0CAM1 register contents

Bit position	Bit name	Function
15	M1VT	ETHA0RVBT counter overflow interrupt mask bit 0: Interrupt generation is enabled (not masked). 1: Interrupt generation is disabled (masked).
14	M1UT	ETHA0TUCA counter overflow interrupt mask bit 0: Interrupt generation is enabled (not masked). 1: Interrupt generation is disabled (masked).
13	M1BT	ETHA0TBCA counter overflow interrupt mask bit 0: Interrupt generation is enabled (not masked). 1: Interrupt generation is disabled (masked).
12	M1MT	ETHA0TMCA counter overflow interrupt mask bit 0: Interrupt generation is enabled (not masked). 1: Interrupt generation is disabled (masked).
11	M1PT	ETHA0TPKT counter overflow interrupt mask bit 0: Interrupt generation is enabled (not masked). 1: Interrupt generation is disabled (masked).
10	M1TB	ETHA0TBYT counter overflow interrupt mask bit 0: Interrupt generation is enabled (not masked). 1: Interrupt generation is disabled (masked).
9	M1MX	ETHA0RMAX counter overflow interrupt mask bit 0: Interrupt generation is enabled (not masked). 1: Interrupt generation is disabled (masked).
8	M11K	ETHA0R1K counter overflow interrupt mask bit 0: Interrupt generation is enabled (not masked). 1: Interrupt generation is disabled (masked).
7	M1FE	ETHA0R511 counter overflow interrupt mask bit 0: Interrupt generation is enabled (not masked). 1: Interrupt generation is disabled (masked).
6	M1TF	ETHA0R255 counter overflow interrupt mask bit 0: Interrupt generation is enabled (not masked). 1: Interrupt generation is disabled (masked).
5	M1OT	ETHA0R127 counter overflow interrupt mask bit 0: Interrupt generation is enabled (not masked). 1: Interrupt generation is disabled (masked).
4	M1SF	ETHA0R64 counter overflow interrupt mask bit 0: Interrupt generation is enabled (not masked). 1: Interrupt generation is disabled (masked).
3	M1BR	ETHA0RBCA counter overflow interrupt mask bit 0: Interrupt generation is enabled (not masked). 1: Interrupt generation is disabled (masked).
2	M1MR	ETHA0RMCA counter overflow interrupt mask bit 0: Interrupt generation is enabled (not masked). 1: Interrupt generation is disabled (masked).
1	M1PR	ETHA0RPKT counter overflow interrupt mask bit 0: Interrupt generation is enabled (not masked). 1: Interrupt generation is disabled (masked).
0	M1RB	ETHA0RBYT counter overflow interrupt mask bit 0: Interrupt generation is enabled (not masked). 1: Interrupt generation is disabled (masked).

(23) ETHA0CAM2 - Carry mask register 2

This register is used to mask the CAINT signal that is generated when a statistics counter has overflowed and the corresponding bit in the ETHA0CAR2 register has been set.

Each bit of this register can be masked.

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + 0134_H

Initial value Undefined

Caution Be sure to set bits 30 to 23 to “0”.

31	30	29	28	27	26	25	24
M2DV	–	–	–	–	–	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
–	M2IM	M2CS	M2NC	M2XC	M2LC	M2MC	M2SC
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
M2XD	M2DF	M2XF	M2TE	M2JB	M2FG	M2OV	M2UN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
M2FC	M2CD	M2FO	M2AL	M2UO	M2PF	M2CF	M2RE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-32 ETHA0CAM2 register contents (1/2)

Bit position	Bit name	Function
31	M2DV	Status vector overrun interrupt mask bit 0: Interrupt generation is enabled (not masked). 1: Interrupt generation is disabled (masked).
22	M2IM	ETHA0TIME counter overflow interrupt mask bit 0: Interrupt generation is enabled (not masked). 1: Interrupt generation is disabled (masked).
21	M2CS	ETHA0TCSE counter overflow interrupt mask bit 0: Interrupt generation is enabled (not masked). 1: Interrupt generation is disabled (masked).
20	M2NC	ETHA0TNCL counter overflow interrupt mask bit 0: Interrupt generation is enabled (not masked). 1: Interrupt generation is disabled (masked).
19	M2XC	ETHA0TXCL counter overflow interrupt mask bit 0: Interrupt generation is enabled (not masked). 1: Interrupt generation is disabled (masked).
18	M2LC	ETHA0TLCL counter overflow interrupt mask bit 0: Interrupt generation is enabled (not masked). 1: Interrupt generation is disabled (masked).
17	M2MC	ETHA0TMCL counter overflow interrupt mask bit 0: Interrupt generation is enabled (not masked). 1: Interrupt generation is disabled (masked).

Table 32-32 ETHA0CAM2 register contents (2/2)

Bit position	Bit name	Function
16	M2SC	ETHA0TSCL counter overflow interrupt mask bit 0: Interrupt generation is enabled (not masked). 1: Interrupt generation is disabled (masked).
15	M2XD	ETHA0TXDF counter overflow interrupt mask bit 0: Interrupt generation is enabled (not masked). 1: Interrupt generation is disabled (masked).
14	M2DF	ETHA0TDFR counter overflow interrupt mask bit 0: Interrupt generation is enabled (not masked). 1: Interrupt generation is disabled (masked).
13	M2XF	ETHA0TXPF counter overflow interrupt mask bit 0: Interrupt generation is enabled (not masked). 1: Interrupt generation is disabled (masked).
12	M2TE	ETHA0TFCS counter overflow interrupt mask bit 0: Interrupt generation is enabled (not masked). 1: Interrupt generation is disabled (masked).
11	M2JB	ETHA0TPKT counter overflow interrupt mask bit 0: Interrupt generation is enabled (not masked). 1: Interrupt generation is disabled (masked).
10	M2FG	ETHA0RFRG counter overflow interrupt mask bit 0: Interrupt generation is enabled (not masked). 1: Interrupt generation is disabled (masked).
9	M2OV	ETHA0ROVR counter overflow interrupt mask bit 0: Interrupt generation is enabled (not masked). 1: Interrupt generation is disabled (masked).
8	M2UN	ETHA0RUND counter overflow interrupt mask bit 0: Interrupt generation is enabled (not masked). 1: Interrupt generation is disabled (masked).
7	M2FC	ETHA0RFCR counter overflow interrupt mask bit 0: Interrupt generation is enabled (not masked). 1: Interrupt generation is disabled (masked).
6	M2CD	ETHA0RCDE counter overflow interrupt mask bit 0: Interrupt generation is enabled (not masked). 1: Interrupt generation is disabled (masked).
5	M2FO	ETHA0RFLR counter carry mask bit 0: Interrupt generation is enabled (not masked). 1: Interrupt generation is disabled (masked).
4	M2AL	ETHA0RALN counter carry mask bit 0: Interrupt generation is enabled (not masked). 1: Interrupt generation is disabled (masked).
3	M2UO	ETHA0RXUO counter carry mask bit 0: Interrupt generation is enabled (not masked). 1: Interrupt generation is disabled (masked).
2	M2PF	ETHA0RXPF counter carry mask bit 0: Interrupt generation is enabled (not masked). 1: Interrupt generation is disabled (masked).
1	M2CF	ETHA0RXCF counter carry mask bit 0: Interrupt generation is enabled (not masked). 1: Interrupt generation is disabled (masked).
0	M2RE	ETHA0RFCS counter carry mask bit 0: Interrupt generation is enabled (not masked). 1: Interrupt generation is disabled (masked).

32.4.2 Statistics counters

(1) ETHA0RBYT - Reception byte counter

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + 0140_H

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
RBYT31	RBYT30	RBYT29	RBYT28	RBYT27	RBYT26	RBYT25	RBYT24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
RBYT23	RBYT22	RBYT21	RBYT20	RBYT19	RBYT18	RBYT17	RBYT16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
RBYT15	RBYT14	RBYT13	RBYT12	RBYT11	RBYT10	RBYT9	RBYT8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
RBYT7	RBYT6	RBYT5	RBYT4	RBYT3	RBYT2	RBYT1	RBYT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-33 ETHA0RBYT register contents

Bit position	Bit name	Function
31:0	RBYT[31:0]	This counter indicates the number of bytes in a received packet. It counts from the destination address byte to the FCS byte. It continues counting bytes even if an error occurs. If a packet longer than the value specified by the ETHA0LMAX register is received when the ETHA0MACC1.HUGEN bit is 0, this counter is incremented because the packet length is assumed to be that specified by the ETHA0LMAX register.

(2) ETHA0RPKT - Reception packet counter

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + 0144_H

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
RPKT31	RPKT30	RPKT29	RPKT28	RPKT27	RPKT26	RPKT25	RPKT24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
RPKT23	RPKT22	RPKT21	RPKT20	RPKT19	RPKT18	RPKT1	RPKT16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
RPKT15	RPKT14	RPKT13	RPKT12	RPKT11	RPKT10	RPKT9	RPKT8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
RPKT7	RPKT6	RPKT5	RPKT4	RPKT3	RPKT2	RPKT1	RPKT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-34 ETHA0RPKT register contents

Bit position	Bit name	Function
31:0	RPKT[31:0]	This counter is incremented when any packet, including packets in which an error has occurred, all unicast packets, all multicast packets, and broadcast packets, is received.

(3) ETHA0RFCS - Reception FCS error frame counter

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + 0148_H

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
RFCS31	RFCS30	RFCS29	RFCS28	RFCS27	RFCS26	RFCS25	RFCS24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
RFCS23	RFCS22	RFCS21	RFCS20	RFCS19	RFCS18	RFCS17	RFCS16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
RFCS15	RFCS14	RFCS13	RFCS12	RFCS11	RFCS10	RFCS9	RFCS8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
RFCS7	RFCS6	RFCS5	RFCS4	RFCS3	RFCS2	RFCS1	RFCS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-35 ETHA0RFCS register contents

Bit position	Bit name	Function
31:0	RFCS[31:0]	This counter is incremented if a CRC error occurs in a receive packet. If a packet longer than the value specified by the ETHA0LMAX register is received when the ETHA0MACC1.HUGEN bit is 0, a CRC check is executed when the packet length reaches the value specified by the ETHA0LMAX register, so this counter might be incremented because the received packet is assumed to contain a CRC error.

(4) ETHA0RMCA - Reception multicast packet counter

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + $014C_H$

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
RMCA31	RMCA30	RMCA29	RMCA28	RMCA27	RMCA26	RMCA25	RMCA24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
RMCA23	RMCA22	RMCA21	RMCA20	RMCA19	RMCA18	RMCA17	RMCA16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
RMCA15	RMCA14	RMCA13	RMCA12	RMCA11	RMCA10	RMCA9	RMCA8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
RMCA7	RMCA6	RMCA5	RMCA4	RMCA3	RMCA2	RMCA1	RMCA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-36 ETHA0RMCA register contents

Bit position	Bit name	Function
31:0	RMCA[31:0]	This counter is incremented when a multicast packet whose length is 64 to 1,518 bytes (64 to 1,522 bytes for a VLAN frame) is received. It is not incremented when a multicast packet in which a CRC error has occurred or a broadcast packet is received.

(5) ETHA0RBCA - Reception broadcast packet counter

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + 0150_H

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
RBCA31	RBCA30	RBCA29	RBCA28	RBCA27	RBCA26	RBCA25	RBCA24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
RBCA23	RBCA22	RBCA21	RBCA20	RBCA19	RBCA18	RBCA17	RBCA16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
RBCA15	RBCA14	RBCA13	RBCA12	RBCA11	RBCA10	RBCA9	RBCA8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
RBCA7	RBCA6	RBCA5	RBCA4	RBCA3	RBCA2	RBCA1	RBCA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-37 ETHA0RBCA register contents

Bit position	Bit name	Function
31:0	RBCA[31:0]	This counter is incremented when a broadcast packet whose length is 64 to 1,518 bytes (64 to 1,522 bytes for a VLAN frame) is received. It is not incremented when a broadcast packet in which a CRC error has occurred or a multicast packet is received.

(6) ETHA0RXCF - Reception control frame packet counter

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + 0154_H

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
RXCF31	RXCF30	RXCF29	RXCF28	RXCF27	RXCF26	RXCF25	RXCF24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
RXCF23	RXCF22	RXCF21	RXCF20	RXCF19	RXCF18	RXCF17	RXCF16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
RXCF15	RXCF14	RXCF13	RXCF12	RXCF11	RXCF10	RXCF9	RXCF8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
RXCF7	RXCF6	RXCF5	RXCF4	RXCF3	RXCF2	RXCF1	RXCF0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-38 ETHA0RXCF register contents

Bit position	Bit name	Function
31:0	RXCF[31:0]	This counter is incremented when any control frame, including pause control frames and unsupported control frames is received. It is not incremented when a control frame in which a CRC error has been detected is received.

(7) ETHA0RXPF - Reception pause frame packet counter

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + 0158_H

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
RXPF31	RXPF30	RXPF29	RXPF28	RXPF27	RXPF26	RXPF25	RXPF24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
RXPF23	RXPF22	RXPF21	RXPF20	RXPF19	RXPF18	RXPF17	RXPF16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
RXPF15	RXPF14	RXPF13	RXPF12	RXPF11	RXPF10	RXPF9	RXPF8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
RXPF7	RXPF6	RXPF5	RXPF4	RXPF3	RXPF2	RXPF1	RXPF0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-39 ETHA0RXPF register contents

Bit position	Bit name	Function
31:0	RXPF[31:0]	This counter is incremented when a valid pause control frame is received.

(8) ETHA0RXUO - Reception undefined control packet counter

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + $015C_H$

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
RXUO31	RXUO30	RXUO29	RXUO28	RXUO27	RXUO26	RXUO25	RXUO24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
RXUO23	RXUO22	RXUO21	RXUO20	RXUO19	RXUO18	RXUO17	RXUO16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
RXUO15	RXUO14	RXUO13	RXUO12	RXUO11	RXUO10	RXUO9	RXUO8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
RXUO7	RXUO6	RXUO5	RXUO4	RXUO3	RXUO2	RXUO1	RXUO0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-40 ETHA0RXUO register contents

Bit position	Bit name	Function
31:0	RXUO[31:0]	This counter is incremented when a control frame that has an opcode other than PAUSE or a pause control frame that has an invalid destination address is received. It is not incremented when a pause control frame in which a CRC error has been detected is received.

(9) ETHA0RALN - Reception alignment error counter

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + 0160_H

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
RALN31	RALN30	RALN29	RALN28	RALN27	RALN26	RALN25	RALN24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
RALN23	RALN22	RALN21	RALN20	RALN19	RALN18	RALN17	RALN16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
RALN15	RALN14	RALN13	RALN12	RALN11	RALN10	RALN9	RALN8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
RALN7	RALN6	RALN5	RALN4	RALN3	RALN2	RALN1	RALN0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-41 ETHA0RALN register contents

Bit position	Bit name	Function
31:0	RALN[31:0]	This counter is incremented if a CRC error and a dribble nibble occur in a received packet. If a packet longer than the value specified by the ETHA0LMAX register is received when the ETHA0MACC1.HUGEN bit is 0, an alignment error check is executed when the packet length reaches the length (in bytes) specified by the ETHA0LMAX register, so this counter is not incremented.

(10) ETHA0RFLR - Reception frame length error counter

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + 0164_H

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
RFLR31	RFLR30	RFLR29	RFLR28	RFLR27	RFLR26	RFLR25	RFLR24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
RFLR23	RFLR22	RFLR21	RFLR20	RFLR19	RFLR18	RFLR17	RFLR16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
RFLR15	RFLR14	RFLR13	RFLR12	RFLR11	RFLR10	RFLR9	RFLR8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
RFLR7	RFLR6	RFLR5	RFLR4	RFLR3	RFLR2	RFLR1	RFLR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-42 ETHA0RFLR register contents

Bit position	Bit name	Function
31:0	RFLR[31:0]	This counter is incremented if the value of the length field of a receive packet does not match the data field length of a packet actually received. If the value of the length field is 1,501 or more (for example, when the bytes equivalent to the length field are used as the Ethernet type field), this counter is not incremented.

(11) ETHA0RCDE - Reception code error counter

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + 0168_H

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
RCDE31	RCDE30	RCDE29	RCDE28	RCDE27	RCDE26	RCDE25	RCDE24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
RCDE23	RCDE22	RCDE21	RCDE20	RCDE19	RCDE18	RCDE17	RCDE16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
RCDE15	RCDE14	RCDE13	RCDE12	RCDE11	RCDE10	RCDE9	RCDE8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
RCDE7	RCDE6	RCDE5	RCDE4	RCDE3	RCDE2	RCDE1	RCDE0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-43 ETHA0RCDE register contents

Bit position	Bit name	Function
31:0	RCDE[31:0]	This counter is incremented if an illegal data symbol has been detected at least once while a carrier is being detected.

(12) ETHA0RFCR - Reception false carrier counter

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + $016C_H$

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
RFCR31	RFCR30	RFCR29	RFCR28	RFCR27	RFCR26	RFCR25	RFCR24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
RFCR23	RFCR22	RFCR21	RFCR20	RFCR19	RFCR18	RFCR17	RFCR16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
RFCR15	RFCR14	RFCR13	RFCR12	RFCR11	RFCR10	RFCR9	RFCR8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
RFCR7	RFCR6	RFCR5	RFCR4	RFCR3	RFCR2	RFCR1	RFCR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-44 ETHA0RFCR register contents

Bit position	Bit name	Function
31:0	RFCR[31:0]	If a false carrier is generated in the idle status, this counter is incremented after the next packet is received. It is assumed that a false carrier has occurred if 1110_B is input as nibble data from RXD while RXER is high. This counter is incremented only once even if multiple false carriers are generated in the idle status.

(13) ETHA0RUND - Reception undersize packet counter

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + 0170_H

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
RUND31	RUND30	RUND29	RUND28	RUND27	RUND26	RUND25	RUND24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
RUND23	RUND22	RUND21	RUND20	RUND19	RUND18	RUND1	RUND16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
RUND15	RUND14	RUND13	RUND12	RUND11	RUND10	RUND9	RUND8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
RUND7	RUND6	RUND5	RUND4	RUND3	RUND2	RUND1	RUND0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-45 ETHA0RUND register contents

Bit position	Bit name	Function
31:0	RUND[31:0]	This counter is incremented if the receive packet length is less than 64 bytes and the packet contains a valid FCS field.

(14) ETHA0ROVR - Reception oversize packet counter

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + 0174_H

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
ROVR31	ROVR30	ROVR29	ROVR28	ROVR27	ROVR26	ROVR25	ROVR24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
ROVR23	ROVR22	ROVR21	ROVR20	ROVR19	ROVR18	ROVR17	ROVR16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
ROVR15	ROVR14	ROVR13	ROVR12	ROVR11	ROVR10	ROVR9	ROVR8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
ROVR7	ROVR6	ROVR5	ROVR4	ROVR3	ROVR2	ROVR1	ROVR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-46 ETHA0ROVR register contents

Bit position	Bit name	Function
31:0	ROVR[31:0]	This counter is incremented if the receive packet length exceeds 1,518 bytes (1,522 bytes for a VLAN frame) and the packet contains a valid FCS field. If a packet longer than the value specified by the ETHA0LMAX register is received when the ETHA0MACC1.HUGEN bit is 0, a CRC check is executed when the packet length reaches the value specified by the ETHA0LMAX register. Therefore, a CRC error might be detected and this counter might not be incremented.

(15) ETHA0RFRG - Reception fragment counter

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + 0178_H

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
RFRG31	RFRG30	RFRG29	RFRG28	RFRG27	RFRG26	RFRG25	RFRG24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
RFRG23	RFRG22	RFRG21	RFRG20	RFRG19	RFRG18	RFRG17	RFRG16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
RFRG15	RFRG14	RFRG13	RFRG12	RFRG11	RFRG10	RFRG9	RFRG8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
RFRG7	RFRG6	RFRG5	RFRG4	RFRG3	RFRG2	RFRG1	RFRG0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-47 ETHA0RFRG register contents

Bit position	Bit name	Function
31:0	RFRG[31:0]	This counter is incremented if the receive packet length is less than 64 bytes and the packet contains a CRC error or an alignment error.

(16) ETHA0RJBR - Reception jabber counter

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + $017C_H$

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
RJBR31	RJBR30	RJBR29	RJBR28	RJBR27	RJBR26	RJBR25	RJBR24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
RJBR23	RJBR22	RJBR21	RJBR20	RJBR19	RJBR18	RJBR17	RJBR16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
RJBR15	RJBR14	RJBR13	RJBR12	RJBR11	RJBR10	RJBR9	RJBR8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
RJBR7	RJBR6	RJBR5	RJBR4	RJBR3	RJBR2	RJBR1	RJBR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-48 ETHA0RJBR register contents

Bit position	Bit name	Function
31:0	RJBR[31:0]	This counter is incremented if the receive packet length exceeds 1,518 bytes (1,522 bytes for a VLAN frame) and the packet contains a CRC error or an alignment error. If a packet longer than the value specified by the ETHA0LMAX register is received when the ETHA0MACC1.HUGEN bit is 0, a CRC check is executed when the packet length reaches the value specified by the ETHA0LMAX register. Therefore, a CRC error might be detected and this counter might be incremented.

(17) ETHA0R64 - Receive 64-byte frame counter

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + 0180_H

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
R6431	R6430	R6429	R6428	R6427	R6426	R6425	R6424
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
R6423	R6422	R6421	R6420	R6419	R6418	R6417	R6416
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
R6415	R6414	R6413	R6412	R6411	R6410	R649	R648
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
R647	R646	R645	R644	R643	R642	R641	R640
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-49 ETHA0R64 register contents

Bit position	Bit name	Function
31:0	R64[31:0]	This counter is incremented if the receive packet length is 64 bytes. It is incremented even if the receive packet contains a CRC error, symbol error, or length/type error.

(18) ETHA0R127 - Receive 65- to 127-byte frame counter

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + 0184_H

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
R12731	R12730	R12729	R12728	R12727	R12726	R12725	R12724
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
R12723	R12722	R12721	R12720	R12719	R12718	R12717	R12716
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
R12715	R12714	R12713	R12712	R12711	R12710	R1279	R1278
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
R1277	R1276	R1275	R1274	R1273	R1272	R1271	R1270
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-50 ETHA0R127 register contents

Bit position	Bit name	Function
31:0	R127[31:0]	This counter is incremented if the receive packet length is 64 to 127 bytes. It is incremented even if the receive packet contains a CRC error, symbol error, or length/type error.

(19) ETHA0R255 - Receive 128- to 255-byte frame counter

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + 0188_H

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
R25531	R25530	R25529	R25528	R25527	R25526	R25525	R25524
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
R25523	R25522	R25521	R25520	R25519	R25518	R25517	R25516
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
R25515	R25514	R25513	R25512	R25511	R25510	R2559	R2558
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
R2557	R2556	R2555	R2554	R2553	R2552	R2551	R2550
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-51 ETHA0R255 register contents

Bit position	Bit name	Function
31:0	R255[31:0]	This counter is incremented if the receive packet length is 128 to 255 bytes. It is incremented even if the receive packet contains a CRC error, symbol error, or length/type error.

(20) ETHA0R511 - Receive 256- to 511-byte frame counter

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + $018C_H$

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
R51131	R51130	R51129	R51128	R51127	R51126	R51125	R51124
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
R51123	R51122	R51121	R51120	R51119	R51118	R51117	R51116
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
R51115	R51114	R51113	R51112	R51111	R51110	R5119	R5118
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
R5117	R5116	R5115	R5114	R5113	R5112	R5111	R5110
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-52 ETHA0R511 register contents

Bit position	Bit name	Function
31:0	R511[31:0]	This counter is incremented if the receive packet length is 256 to 511 bytes. It is incremented even if the receive packet contains a CRC error, symbol error, or length/type error.

(21) ETHA0R1K - Receive 512- to 1023-byte frame counter

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + 0190_H

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
R1K31	R1K30	R1K29	R1K28	R1K27	R1K26	R1K25	R1K24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
R1K23	R1K22	R1K21	R1K20	R1K19	R1K18	R1K17	R1K16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
R1K15	R1K14	R1K13	R1K12	R1K11	R1K10	R1K9	R1K8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
R1K7	R1K6	R1K5	R1K4	R1K3	R1K2	R1K1	R1K0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-53 ETHA0R1K register contents

Bit position	Bit name	Function
31:0	R1K[31:0]	This counter is incremented if the receive packet length is 512 to 1,023 bytes. It is incremented even if the receive packet contains a CRC error, symbol error, or length/type error.

(22) ETHA0RMAX - Receive 1024- to RMAX-byte frame counter

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + 0194_H

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
RMAX31	RMAX30	RMAX29	RMAX28	RMAX27	RMAX26	RMAX25	RMAX24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
RMAX23	RMAX22	RMAX21	RMAX20	RMAX19	RMAX18	RMAX17	RMAX16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
RMAX15	RMAX14	RMAX13	RMAX12	RMAX11	RMAX10	RMAX9	RMAX8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
RMAX7	RMAX6	RMAX5	RMAX4	RMAX3	RMAX2	RMAX1	RMAX0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-54 ETHA0RMAX register contents

Bit position	Bit name	Function
31:0	RMAX[31:0]	This counter is incremented if the receive packet length is 1,024 to 1,518 bytes (1,024 to 1,522 bytes for a VLAN frame). It is incremented even if the receive packet contains a CRC error, symbol error, or length/type error.

(23) ETHA0RVBT - Receive valid byte counter

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + 0198_H

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
RVBT31	RVBT30	RVBT29	RVBT28	RVBT27	RVBT26	RVBT25	RVBT24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
RVBT23	RVBT22	RVBT21	RVBT20	RVBT19	RVBT18	RVBT17	RVBT16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
RVBT15	RVBT14	RVBT13	RVBT12	RVBT11	RVBT10	RVBT9	RVBT8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
RVBT7	RVBT6	RVBT5	RVBT4	RVBT3	RVBT2	RVBT1	RVBT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-55 ETHA0RVBT register contents

Bit position	Bit name	Function
31:0	RVBT[31:0]	This counter indicates the byte count of a valid packet. It counts from the destination address byte to the FCS byte.

(24) ETHA0TBYT - Transmission byte counter

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + $01C0_H$

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
TBYT31	TBYT30	TBYT29	TBYT28	TBYT27	TBYT26	TBYT25	TBYT24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
TBYT23	TBYT22	TBYT21	TBYT20	TBYT19	TBYT18	TBYT17	TBYT16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
TBYT15	TBYT14	TBYT13	TBYT12	TBYT11	TBYT10	TBYT9	TBYT8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
TBYT7	TBYT6	TBYT5	TBYT4	TBYT3	TBYT2	TBYT1	TBYT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-56 ETHA0TBYT register contents

Bit position	Bit name	Function
31:0	TBYT[31:0]	This counter indicates the number of bytes in a transmit packet. When a collision occurs before transmission is finished or aborted, the transmission byte at which the collision occurred is also counted. The preamble and SFD are not included in the byte count indication.

(25) ETHA0TPKT - Transmission packet counter

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + $01C4_H$

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
TPKT31	TPKT30	TPKT29	TPKT28	TPKT27	TPKT26	TPKT25	TPKT24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
TPKT23	TPKT22	TPKT21	TPKT20	TPKT19	TPKT18	TPKT17	TPKT16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
TPKT15	TPKT14	TPKT13	TPKT12	TPKT11	TPKT10	TPKT9	TPKT8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
TPKT7	TPKT6	TPKT5	TPKT4	TPKT3	TPKT2	TPKT1	TPKT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-57 ETHA0TPKT register contents

Bit position	Bit name	Function
31:0	TPKT[31:0]	This counter is incremented when any packet is transmitted. This includes when a packet in which an error has occurred, unicast packet, multicast packet, or broadcast packet is transmitted.

(26) ETHA0TFCS - Transmission FCS error frame counter

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + $01C8_H$

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
TFCS31	TFCS30	TFCS29	TFCS28	TFCS27	TFCS26	TFCS25	TFCS24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
TFCS23	TFCS22	TFCS21	TFCS20	TFCS19	TFCS18	TFCS17	TFCS16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
TFCS15	TFCS14	TFCS13	TFCS12	TFCS11	TFCS10	TFCS9	TFCS8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
TFCS7	TFCS6	TFCS5	TFCS4	TFCS3	TFCS2	TFCS1	TFCS0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-58 ETHA0TFCS register contents

Bit position	Bit name	Function
31:0	TFCS[31:0]	This counter is incremented if a CRC error is detected in the FCS field that is appended to a transmit packet. It is not incremented if transmission is aborted.

(27) ETHA0TMCA - Transmission multicast packet counter

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + $01CC_H$

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
TMCA31	TMCA30	TMCA29	TMCA28	TMCA27	TMCA26	TMCA25	TMCA24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
TMCA23	TMCA22	TMCA21	TMCA20	TMCA19	TMCA18	TMCA17	TMCA16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
TMCA15	TMCA14	TMCA13	TMCA12	TMCA11	TMCA10	TMCA9	TMCA8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
TMCA7	TMCA6	TMCA5	TMCA4	TMCA3	TMCA2	TMCA1	TMCA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-59 ETHA0TMCA register contents

Bit position	Bit name	Function
31:0	TMCA[31:0]	This counter is incremented when a multicast packet is transmitted. It is not incremented when a broadcast packet is transmitted. Nor is it incremented if transmission is aborted or if a CRC error is detected.

(28) ETHA0TBCA - Transmission broadcast packet counter

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + $01D0_H$

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
TBCA31	TBCA30	TBCA29	TBCA28	TBCA27	TBCA26	TBCA25	TBCA24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
TBCA23	TBCA22	TBCA21	TBCA20	TBCA19	TBCA18	TBCA17	TBCA16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
TBCA15	TBCA14	TBCA13	TBCA12	TBCA11	TBCA10	TBCA9	TBCA8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
TBCA7	TBCA6	TBCA5	TBCA4	TBCA3	TBCA2	TBCA1	TBCA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-60 ETHA0TBCA register contents

Bit position	Bit name	Function
31:0	TBCA[31:0]	This counter is incremented when a broadcast packet is transmitted. It is not incremented when a multicast packet is transmitted. Nor is it incremented if transmission is aborted or if a CRC error is detected.

(29) ETHA0TUCA - Transmit unicast packet counter

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + $01D4_H$

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
TUCA31	TUCA30	TUCA29	TUCA28	TUCA27	TUCA26	TUCA25	TUCA24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
TUCA23	TUCA22	TUCA21	TUCA20	TUCA19	TUCA18	TUCA17	TUCA16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
TUCA15	TUCA14	TUCA13	TUCA12	TUCA11	TUCA10	TUCA9	TUCA8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
TUCA7	TUCA6	TUCA5	TUCA4	TUCA3	TUCA2	TUCA1	TUCA0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-61 ETHA0TUCA register contents

Bit position	Bit name	Function
31:0	TUCA[31:0]	This counter is incremented when a unicast packet is transmitted. It is not incremented if transmission is aborted or if a CRC error is detected.

(30) ETHA0TXPF - Transmission pause control frame counter

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + $01D8_H$

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
TXPF31	TXPF30	TXPF29	TXPF28	TXPF27	TXPF26	TXPF25	TXPF24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
TXPF23	TXPF22	TXPF21	TXPF20	TXPF19	TXPF18	TXPF17	TXPF16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
TXPF15	TXPF14	TXPF13	TXPF12	TXPF11	TXPF10	TXPF9	TXPF8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
TXPF7	TXPF6	TXPF5	TXPF4	TXPF3	TXPF2	TXPF1	TXPF0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-62 ETHA0TXPF register contents

Bit position	Bit name	Function
31:0	TXPF[31:0]	This counter is incremented each time a pause control frame is transmitted when the maximum amount of data has been stored in the receive FIFO.

(31) ETHA0TDFR - Transmission delay packet counter

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + $01DC_H$

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
TDFR31	TDFR30	TDFR29	TDFR28	TDFR27	TDFR26	TDFR25	TDFR24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
TDFR23	TDFR22	TDFR21	TDFR20	TDFR19	TDFR18	TDFR17	TDFR16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
TDFR15	TDFR14	TDFR13	TDFR12	TDFR11	TDFR10	TDFR9	TDFR8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
TDFR7	TDFR6	TDFR5	TDFR4	TDFR3	TDFR2	TDFR1	TDFR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-63 ETHA0TDFR register contents

Bit position	Bit name	Function
31:0	TDFR[31:0]	This counter is incremented if a transmit delay occurs because of carrier detection when transmission is about to start. It is not incremented if a collision occurs during transmission that was started after the delay occurred.

(32) ETHA0TXDF - Transmission excessive delay packet counter

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + $01E0_H$

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
TXDF31	TXDF30	TXDF29	TXDF28	TXDF27	TXDF26	TXDF25	TXDF24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
TXDF23	TXDF22	TXDF21	TXDF20	TXDF19	TXDF18	TXDF17	TXDF16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
TXDF15	TXDF14	TXDF13	TXDF12	TXDF11	TXDF10	TXDF9	TXDF8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
TXDF7	TXDF6	TXDF5	TXDF4	TXDF3	TXDF2	TXDF1	TXDF0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-64 ETHA0TXDF register contents

Bit position	Bit name	Function
31:0	TXDF[31:0]	This counter is incremented if transmission is aborted by an excessive delay.

(33) ETHA0TSCL - Transmission single collision packet counter

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + $01E4_H$

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
TSCL31	TSCL30	TSCL29	TSCL28	TSCL27	TSCL26	TSCL25	TSCL24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
TSCL23	TSCL22	TSCL21	TSCL20	TSCL19	TSCL18	TSCL17	TSCL16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
TSCL15	TSCL14	TSCL13	TSCL12	TSCL11	TSCL10	TSCL9	TSCL8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
TSCL7	TSCL6	TSCL5	TSCL4	TSCL3	TSCL2	TSCL1	TSCL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-65 ETHA0TSCL register contents

Bit position	Bit name	Function
31:0	TSCL[31:0]	This counter is incremented if a transmission finishes successfully after a single collision occurred during the transmission.

(34) ETHA0TMCL - Transmission multiple collision packet counter

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + $01E8_H$

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
TMCL31	TMCL30	TMCL29	TMCL28	TMCL27	TMCL26	TMCL25	TMCL24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
TMCL23	TMCL22	TMCL21	TMCL20	TMCL19	TMCL18	TMCL17	TMCL16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
TMCL15	TMCL14	TMCL13	TMCL12	TMCL11	TMCL10	TMCL9	TMCL8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
TMCL7	TMCL6	TMCL5	TMCL4	TMCL3	TMCL2	TMCL1	TMCL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-66 ETHA0TMCL register contents

Bit position	Bit name	Function
31:0	TMCL[31:0]	This counter is incremented if a transmission finishes successfully after a collision occurred multiple times (but less times than the value specified by ETHA0CLRT.RETRY) during the transmission.

(35) ETHA0TLCL - Transmission late collision packet counter

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + $01EC_H$

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
TLCL31	TLCL30	TLCL29	TLCL28	TLCL27	TLCL26	TLCL25	TLCL24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
TLCL23	TLCL22	TLCL21	TLCL20	TLCL19	TLCL18	TLCL17	TLCL16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
TLCL15	TLCL14	TLCL13	TLCL12	TLCL11	TLCL10	TLCL9	TLCL8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
TLCL7	TLCL6	TLCL5	TLCL4	TLCL3	TLCL2	TLCL1	TLCL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-67 ETHA0TLCL register contents

Bit position	Bit name	Function
31:0	TLCL[31:0]	This counter is incremented if a late collision occurs during transmission.

(36) ETHA0TXCL - Transmission excessive collision packet counter

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + $01F0_H$

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
TXCL31	TXCL30	TXCL29	TXCL28	TXCL27	TXCL26	TXCL25	TXCL24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
TXCL23	TXCL22	TXCL21	TXCL20	TXCL19	TXCL18	TXCL17	TXCL16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
TXCL15	TXCL14	TXCL13	TXCL12	TXCL11	TXCL10	TXCL9	TXCL8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
TXCL7	TXCL6	TXCL5	TXCL4	TXCL3	TXCL2	TXCL1	TXCL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-68 ETHA0TXCL register contents

Bit position	Bit name	Function
31:0	TXCL[31:0]	This counter is incremented if collision occurs more than the times specified by ETHA0CLRT.RETRY in a single transmission.

(37) ETHA0TNCL - Transmission total collision counter

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + $01F4_H$

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
TNCL31	TNCL30	TNCL29	TNCL28	TNCL27	TNCL26	TNCL25	TNCL24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
TNCL23	TNCL22	TNCL21	TNCL20	TNCL19	TNCL18	TNCL17	TNCL16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
TNCL15	TNCL14	TNCL13	TNCL12	TNCL11	TNCL10	TNCL9	TNCL8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
TNCL7	TNCL6	TNCL5	TNCL4	TNCL3	TNCL2	TNCL1	TNCL0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-69 ETHA0TNCL register contents

Bit position	Bit name	Function
31:0	TNCL[31:0]	This counter counts the number of collisions after which transmission finishes successfully.

(38) ETHA0TCSE - Transmission carrier sense error counter

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + $01F8_H$

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
TCSE31	TCSE30	TCSE29	TCSE28	TCSE27	TCSE26	TCSE25	TCSE24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
TCSE23	TCSE22	TCSE21	TCSE20	TCSE19	TCSE18	TCSE17	TCSE16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
TCSE15	TCSE14	TCSE13	TCSE12	TCSE11	TCSE10	TCSE9	TCSE8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
TCSE7	TCSE6	TCSE5	TCSE4	TCSE3	TCSE2	TCSE1	TCSE0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-70 ETHA0TCSE register contents

Bit position	Bit name	Function
31:0	TCSE[31:0]	This counter is incremented if a carrier sense error occurs during transmission.

(39) ETHA0TIME - MAC internal error counter

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + $01FC_H$

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
TIME31	TIME30	TIME29	TIME28	TIME27	TIME26	TIME25	TIME24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
TIME23	TIME22	TIME21	TIME20	TIME19	TIME18	TIME17	TIME16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
TIME15	TIME14	TIME13	TIME12	TIME11	TIME10	TIME9	TIME8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
TIME7	TIME6	TIME5	TIME4	TIME3	TIME2	TIME1	TIME0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-71 ETHA0TIME register contents

Bit position	Bit name	Function
31:0	TIME[31:0]	This counter is incremented if an error occurs in MAC during transmission or if a packet longer than the value specified by the ETHA0LMAX register is transmitted.

32.4.3 FIFO controller control registers

(1) ETHA0MFFCONT - FIFO controller control register

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + 0200_H

Initial value Undefined

Cautions

1. Be sure to set the following bits to the values specified below (fixed values). If other values are set, the correct operation cannot be guaranteed.
   - RXSDMA[1:0] = 10
   - ASOE = 0
   - APS = 1
   - APL = 1
   - RXTHRC = 0
   - TXTHRC = 0
2. Be sure to set bits 29, 28, 23 to 19, 13, and 7 to 3 to “0”.

31	30	29	28	27	26	25	24
LOOPBACK	RCSEL	–	–	IMLP3	IMLP2	IMLP1	IMLP0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
–	–	–	–	–	FLOWCNT	IVPAUSE	ZEROPAUSE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
RXSDMA1	RXSDMA0	–	ASOE	APS	APL	RXTHRC	RXEN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
–	–	–	–	–	TABT	TXTHRC	TXEN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-72 ETHA0MFFCONT register contents

Bit position	Bit name	Function
31	LOOPBACK	Loopback mode Loopback between the transmit FIFO and receive FIFO is performed. 0: Normal mode 1: Loopback mode
30	RCSEL	RXCLK selection TXCLK is selected to be internally connected, instead of RXCLK. Specify a value for this bit if it is necessary to switch RXCLK to TXCLK when the MAC or the FIFO controller is in the loopback mode. 0: Normal mode 1: Clock switch mode (RXCLK switched to TXCLK)
27 to 24	IMLP[3:0]	Set these bits to 0000.
18	FLOWCNT	Flow control enable/disable 0: Flow control is disabled. 1: Flow control is enabled.
17	IVPAUSE	Interval pause packet transmission control This bit specifies the method of retransmitting a pause packet. 0: Retransmission by threshold value of FIFO Internal pause timer (ETHA0PAUSETM.IPTIME) is not used. 1: Retransmission by internal pause timer Internal pause timer (ETHA0PAUSETM.IPTIME) is used.
16	ZEROPAUSE	Zero pause packet output enable/disable 0: Zero pause packet output is disabled. 1: Zero pause packet output is enabled.
15, 14	RXSDMA[1:0]	Fix these bits to 10.
12	ASOE	Fix this bit to 0.
11	APS	Fix this bit to 1.
10	APL	Fix this bit to 1.
9	RXTHRC	Fix this bit to 0.
8	RXEN	Reception enable 0: Disables reception. 1: Enables reception. [Timing for disabling reception] If this bit is cleared to disable reception while a packet is being written from the MAC to the receive FIFO, the receive FIFO write circuit is stopped after the packet has been written. The receive FIFO of the system stops after all the packets written in the receive FIFO have been read. The flow control circuit is not stopped by this bit.
2	TABT	Transmission abort control This bit retransmits a packet that has been aborted by the MAC. 0: Discards the packet. 1: Retransmits the packet.
1	TXTHRC	Fix this bit to 0.
0	TXEN	Transmission enable 0: Disables transmission. 1: Enables transmission. [Timing for disabling transmission] If this bit is cleared to disable transmission while a packet is being written to the transmit FIFO, the transmit FIFO write circuit is stopped after the packet has been written (after the END flag has been set), cancelling a request to write the next packet. Packet transfer to the MAC is stopped after all the packets in the transmit FIFO have been transferred (after the empty status is indicated).

(2) ETHA0RSTCNT - Software reset control register

This register is used to control software reset.

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + 0204_H

Initial value Undefined

Caution Be sure to set bits 31 to 17, 15 to 9, and 7 to 1 to “0”.

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	RFFLSH
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	TFFLSH
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	SFTRST
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-73 ETHA0RSTCNT register contents

Bit position	Bit name	Function
16	RFFLSH	Receive FIFO clear (flush) This bit clears the receive FIFO, reception control circuit, flow control circuit, reception status register, and interrupt registers related to reception. Writing 1 to this bit starts the reset operation, and then this bit is automatically cleared. This bit is always read as 0.
8	TFFLSH	Transmit FIFO clear (flush) This bit clears the transmit FIFO, transmission control circuit, transmission status register, and interrupt registers related to transmission. Writing 1 to this bit starts the reset operation, and then this bit is automatically cleared.
0	SFTRST	Software reset This bit resets all the circuits of the FIFO controller (MFF). Writing 1 to this bit starts the reset operation, and then this bit is automatically cleared.

(3) ETHA0FLOWTH - Flow control threshold value register

This register is used to specify a threshold value for the receive FIFO to start flow control, and a threshold value to transmit a zero pause control frame.

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + 0218_H

Initial value Undefined

Caution Be sure to set bits 31 to 27 and 15 to 11 to “0”.

31	30	29	28	27	26	25	24
–	–	–	–	–	FLOWTHR10	FLOWTHR9	FLOWTHR8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
FLOWTHR7	FLOWTHR6	FLOWTHR5	FLOWTHR4	FLOWTHR3	FLOWTHR2	FLOWTHR1	FLOWTHR0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
–	–	–	–	–	ZPTHR10	ZPTHR9	ZPTHR8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
ZPTHR7	ZPTHR6	ZPTHR5	ZPTHR4	ZPTHR3	ZPTHR2	ZPTHR1	ZPTHR0
R/W	R/W	R/W	R/W	R/W	R/W	R	R

Table 32-74 ETHA0FLOWTH register contents

Bit position	Bit name	Function
26:16	FLOWTHR[10:0]	These bits specify the threshold value in bytes for the receive FIFO to start flow control. Flow control starts when the amount of data in the receive FIFO reaches the value specified by these bits. Back pressure transmission is executed in the half-duplex mode and pause control packets are transmitted in the full-duplex mode. Writing bit 17 or 16 is ignored because the receive FIFO can only be written in 32-bit (4-byte) units. Bits 17 and 16 are always read as 0.
10:0	ZPTHR[10:0]	These bits specify the threshold value in bytes for transmitting a zero pause control packet. When zero pause packet transmission is enabled by setting the MFFCNT.ZEROPAUSE bit to 1 (high level) and flow is controlled by pause control packets, a zero pause packet is transmitted if the amount of data in the receive FIFO falls below the threshold value specified by these bits. Writing bit 1 or 0 is ignored because the receive FIFO can only be written in 32-bit (4-byte) units. Bits 1 and 0 are always read as 0.

(4) ETHA0PAUSETM - Pause timer value register

This register is used to set the pause time.

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + $021C_H$

Initial value $7FFF\ FFFF_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
IPTIME15	IPTIME14	IPTIME13	IPTIME12	IPTIME11	IPTIME10	IPTIME9	IPTIME8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
IPTIME7	IPTIME6	IPTIME5	IPTIME4	IPTIME3	IPTIME2	IPTIME1	IPTIME0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
PAUSETM_MAX15	PAUSETM_MAX14	PAUSETM_MAX13	PAUSETM_MAX12	PAUSETM_MAX11	PAUSETM_MAX10	PAUSETM_MAX9	PAUSETM_MAX8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
PAUSETM_MAX7	PAUSETM_MAX6	PAUSETM_MAX5	PAUSETM_MAX4	PAUSETM_MAX3	PAUSETM_MAX2	PAUSETM_MAX1	PAUSETM_MAX0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-75 ETHA0PAUSETM register contents

Bit position	Bit name	Function
31:16	IPTIME[15:0]	Interval pause packet timer value These bits specify the interval for transmitting a pause packet when interval pause packet transmission is enabled by setting the MFFCNT.IVPAUSE bit. Time required to transmit one packet = Time required to transmit 512 bits (circuit size) = 128 TXCLK cycles Default value: About 168 ms when the data rate is 100 Mbps, or about 1.68 seconds when the data rate is 10 Mbps
15:0	PAUSETM_MAX[15:0]	Pause control timer value of MAX pause packet These bits specify the value of TPTV[15:0] when a pause control request is issued to the MAC. Time required to transmit one packet = Time required to transmit 512 bits (circuit size) = 128 TXCLK cycles Default value: About 336 ms when the data rate is 100 Mbps, or about 3.36 seconds when the data rate is 10 Mbps

(5) ETHA0RXERSEL - Receive error selection register

This register is used to specify whether to accept or discard the packet if a reception error occurs.

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + 0220_H

Initial value Undefined

Caution Be sure to set bits 25 to 23 and 15 to 2 to “0”.

31	30	29	28	27	26	25	24
RLENE	VLAN	USOP	RPCF	RCFR	DBNB	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
–	RLOR	RLER	RCRCE	RXER	CEPS	REPS	PAIG
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
–	–	–	–	–	–	TXRX	DVCF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-76 ETHA0RXERSEL register contents (1/2)

Bit position	Bit name	Function
31	RLENE	Specifies whether to accept a packet that contains a packet length error. 0: Accepts the packet. 1: Discards the packet.
30	VLAN	Specifies whether to accept a VLAN packet. 0: Accepts the packet. 1: Discards the packet.
29	USOP	Specifies whether to accept an undefined opcode control packet. 0: Accepts the packet. 1: Discards the packet.
28	RPCF	Specifies whether to accept a pause control packet. 0: Accepts the packet. 1: Discards the packet.
27	RCFR	Specifies whether to accept a control packet. 0: Accepts the packet. 1: Discards the packet.
26	DBNB	Specifies whether to accept a packet that contains dribble nibble. 0: Accepts the packet. 1: Discards the packet.
22	RLOR	Specifies whether to accept a packet that has a length field exceeding 1,500 bytes. 0: Accepts the packet. 1: Discards the packet.

Table 32-76 ETHA0RXERSEL register contents (2/2)

Bit position	Bit name	Function
21	RLER	Specifies whether to accept a packet whose length field does not match the data field length. 0: Accepts the packet. 1: Discards the packet.
20	RCRCE	Specifies whether to accept a packet in which a CRC error has occurred. 0: Accepts the packet. 1: Discards the packet.
19	RXER	Specifies whether to accept a packet in which RXER has been detected. 0: Accepts the packet. 1: Discards the packet.
18	CEPS	Specifies whether to accept a packet in which a false carrier has been detected. 0: Accepts the packet. 1: Discards the packet.
17	REPS	Specifies whether to accept a packet in which the total size of the preamble and SFD or the data field size is one nibble. 0: Accepts the packet. 1: Discards the packet.
16	PAIG	Specifies whether to accept a packet for which one of the following events occurred after the previous packet was received. • A carrier longer than 6,072 nibbles (3,036 bytes) has been detected. • The next packet with IFG + preamble + SFD has been received before the time required to transmit 80 bits has elapsed after a packet has been received. • An illegal preamble or SFD has been received while the pure preamble data check is enabled (the ETHA0MACC1.PUREP bit is set). 0: Accepts the packet. 1: Discards the packet.
1	TXRX	Specifies whether to accept a packet in which a collision has been detected by the MAC. 0: Accepts the packet. 1: Discards the packet.
0	DVCF	Specifies whether to accept a packet received by the MAC that is judged to be a valid control packet. 0: Accepts the packet. 1: Discards the packet.

(6) ETHA0TXSTMONI1 - Transmission status monitor 1 register

Access This register is read-only, in 32-bit units.

Address <ETHA0_base> + 0230_H

Initial value Undefined

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
–	–	–	CSE	TBP	TPP	TPCF	TCFR
R/W	R/W	R/W	R	R	R	R	R
15	14	13	12	11	10	9	8
TTBC15	TTBC14	TTBC13	TTBC12	TTBC11	TTBC10	TTBC9	TTBC8
R	R	R	R	R	R	R	R
7	6	5	4	3	2	1	0
TTBC7	TTBC6	TTBC5	TTBC4	TTBC3	TTBC2	TTBC1	TTBC0
R	R	R	R	R	R	R	R

Table 32-77 ETHA0TXSTMONI1 register contents

Bit position	Bit name	Function
20	CSE	Detection of carrier loss during transmission
19	TBP	Occurrence of a collision due to the back pressure function after the previous transmission[a]
18	TPP	End of transmission of a transmit packet requested during pausing[b]
17	TPCF	Transmission of a pause control packet
16	TCFR	Transmission of a control packet
15:0	TTBC[15:0]	Number of bytes of transmitted data, including packets in which collisions occurred

a) This bit is set if a collision has occurred between when the transmission status was previously updated and when the status is updated this time.

b) This bit is not set if the packet requested during pausing is a control frame.

(7) ETHA0TXSTMONI2 - Transmission status monitor 2 register

Access This register is read-only, in 32-bit units.

Address <ETHA0_base> + 0234_H

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
TUDR	TGNT	LCOL	ECOL	TEDFR	TDFR	TBRO	TMUL
R	R	R	R	R	R	R	R
23	22	21	20	19	18	17	16
TDONE	TFLOR	TFLER	TCRCE	TCBC3	TCBC2	TCBC1	TCBC0
R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8
TBYT15	TBYT14	TBYT13	TBYT12	TBYT11	TBYT10	TBYT9	TBYT8
R	R	R	R	R	R	R	R
7	6	5	4	3	2	1	0
TBYT7	TBYT6	TBYT5	TBYT4	TBYT3	TBYT2	TBYT1	TBYT0
R	R	R	R	R	R	R	R

Table 32-78 ETHA0TXSTMONI2 register contents

Bit position	Bit name	Function
31	TUDR	Detection of a transmit packet underrun[a]
30	TGNT	Transmission of a packet longer than the value specified by ETHA0LMAX[b]
29	LCOL	Occurrence of a late collision
28	ECOL	Occurrence of collisions exceeding the specified maximum number
27	TEDFR	Detection of an excessive transmission delay
26	TDFR	Occurrence of a transmission delay
25	TBRO	Transmission of a broadcast packet
24	TMUL	Transmission of a multicast packet
23	TDONE	End of transmission[c]
22	TFLOR	Detection of a length field greater than 1,500 bytes[d]
21	TFLER	Detection of a packet whose length field does not match the data field length[d, e]
20	TCRCE	Occurrence of a CRC error when CRC automatic appending mode was disabled
19:16	TCBC[3:0]	Number of times retransmission occurred due to collisions[f]
15:0	TBYT[15:0]	Transmit packet length (number of bytes) when transmission finished normally[f]

a) This bit is set only if no collision has occurred.
b) This bit is set only if ETHA0MACC1.HUGEN is 0.
c) This bit is not set if transmission has been aborted.
d) This bit is not set if ETHA0MACC1.FLCHT is 0.
e) If the length field exceeds 1,500 bytes, TFLOR is set and TFLER is not.
f) This value is not correct if transmission has been aborted.

(8) ETHA0TXFINF1 - Transmission status 1 register

Access This register is read-only, in 32-bit units.

Address <ETHA0_base> + 0238_H

Initial value Undefined

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	TPCNT8
R	R	R	R	R	R	R	R
23	22	21	20	19	18	17	16
TPCNT7	TPCNT6	TPCNT5	TPCNT4	TPCNT3	TPCNT2	TPCNT1	TPCNT0
R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8
–	–	–	–	TREMAIN11	TREMAIN10	TREMAIN9	TREMAIN8
R	R	R	R	R	R	R	R
7	6	5	4	3	2	1	0
TREMAIN7	TREMAIN6	TREMAIN5	TREMAIN4	TREMAIN3	TREMAIN2	TREMAIN1	TREMAIN0
R	R	R	R	R	R	R	R

Table 32-79 ETHA0TXFINF1 register contents

Bit position	Bit name	Function
24:16	TPCNT[8:0]	Number of packets in the transmit FIFO These bits indicate the number of packets (start flag to end flag) that exist in the transmit FIFO. The value is incremented when one packet has been written by the system. The value is decremented when one packet has been read by the MAC (upon completion or halting of transmission).
11:0	TREMAIN[11:0]	These bits indicate the remaining capacity of the transmit FIFO (in bytes). Bits 1 and 0 are always 00 because the data in the transmit FIFO are aligned in 32-bit (4-byte) units.

(9) ETHA0TXFINF2 - Transmission status 2 register

Access This register is read-only, in 32-bit units.

Address <ETHA0_base> + $023C_H$

Initial value Undefined

Caution Before rewriting a mode register related to transmission, be sure to confirm that the ETHA0TXFINF2.TXSTOP bit is 1.

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
R	R	R	R	R	R	R	R
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
R	R	R	R	R	R	R	R
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	TXSTOP
R	R	R	R	R	R	R	R

Table 32-80 ETHA0TXFINF2 register contents

Bit position	Bit name	Function
0	TXSTOP	This bit is set if no data is in the transmit FIFO while transmission is stopped (by clearing ETHA0MFFCONT.TXEN). Before rewriting a mode setting register related to transmission, be sure to confirm that this bit is 1. 0: The transmit FIFO is operating. 1: The transmit FIFO is stopped.

(10) ETHA0RXSTMONI - Reception status monitor register

Access This register is read-only, in 32-bit units.

Address <ETHA0_base> + 0240_H

Initial value $0000\ 0000_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
RLENE	VLAN	USOP	RPCF	RCFR	DBNB	RBRO	RMUL
R	R	R	R	R	R	R	R
23	22	21	20	19	18	17	16
RXOK	RLOR	RLER	RCRCE	RXER	CEPS	REPS	PAIG
R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8
RBYT15	RBYT14	RBYT13	RBYT12	RBYT11	RBYT10	RBYT9	RBYT8
R	R	R	R	R	R	R	R
7	6	5	4	3	2	1	0
RBYT7	RBYT6	RBYT5	RBYT4	RBYT3	RBYT2	RBYT1	RBYT0
R	R	R	R	R	R	R	R

Table 32-81 ETHA0RXSTMONI register contents

Bit position	Bit name	Function
31	RLENE	Occurrence of a receive packet length error This bit is set if the received packet size is less than 64 bytes or greater than 1,518 bytes (or less than 64 bytes or greater than 1,522 bytes for a VLAN packet).
30	VLAN	Reception of a VLAN packet This bit is set if a packet whose TPID field matches ETHA0VLTP is received.[a]
29	USOP	Reception of an undefined opcode control packet[b]
28	RPCF	Reception of a pause control packet[b]
27	RCFR	Reception of a control packet[b]
26	DBNB	Reception of a packet containing dribble nibble
25	RBRO	Reception of a broadcast packet
24	RMUL	Reception of a multicast packet
23	RXOK	End of reception[a]
22	RLOR	Reception of a packet that has a length field exceeding 1,500 bytes[c]
21	RLER	Detection of a packet whose length field does not match the data field length[c, d]
20	RCRCE	Occurrence of a CRC error
19	RXER	Detection of RXER
18	CEPS	Detection of a false carrier [e]
17	REPS	Reception of a packet in which the total size of the preamble and SFD or the data field size is one nibble[e, f]
16	PAIG	This bit is set if one of the following events occurred after the previous packet was received.[e] • A carrier longer than 6,072 nibbles (3,036 bytes) has been detected. • The next packet with IFG + preamble + SFD has been received before the time required to transmit 80 bits has elapsed after a packet has been received.[f] • An illegal preamble or SFD has been received while the pure preamble data check is enabled (the ETHA0MACC1.PUREP bit is set).[f]
15:0	RBYT[15:0]	Number of received bytes

a) This bit is not set if a CRC error or RXER has occurred.
b) This bit is not set if a CRC error has occurred.
c) This bit is not set if ETHA0MACC1.FLCHT is 0.
d) If the length field exceeds 1,500 bytes, RLOR is set and RLER is not.
e) The bit is set if a false carrier is detected between when the reception status was updated previously and when it is updated this time.
f) A packet in which this event has occurred is ignored and not transferred to the upstream system.
The ETHA0RXSTMONI register is updated when a DMA transfer of the receive packet has finished. (This register indicates the status of the packet transferred by DMA.) The ETHA0RXFINF1 register is also updated at the same time.

The ETHA0RXSTATUS register is updated when a DMA transfer of the receive packet has finished. The reception status register 1 (ETHA0RXFINF1) is also updated at the same time.

(11) ETHA0RXFINF1 - Reception status 1 register

Access This register is read-only, in 32-bit units.

Address <ETHA0_base> + 0244_H

Initial value Undefined

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
R	R	R	R	R	R	R	R
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8
RPLEN15	RPLEN14	RPLEN13	RPLEN12	RPLEN11	RPLEN10	RPLEN9	RPLEN8
R	R	R	R	R	R	R	R
7	6	5	4	3	2	1	0
RPLEN7	RPLEN6	RPLEN5	RPLEN4	RPLEN3	RPLEN2	RPLEN1	RPLEN0
R	R	R	R	R	R	R	R

Table 32-82 ETHA0RXFINF1 register contents

Bit position	Bit name	Function
15:0	RPLEN[15:0]	These bits indicate the receive packet length in bytes. The Ethernet controller uses the value of RPLEN for the size field when writing back the receive descriptor.

(12) ETHA0RXFINF2 - Reception status 2 register

Access This register is read-only, in 32-bit units.

Address <ETHA0_base> + 0248_H

Initial value Undefined

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	RPCNT8
R	R	R	R	R	R	R	R
23	22	21	20	19	18	17	16
RPCNT7	RPCNT6	RPCNT5	RPCNT4	RPCNT3	RPCNT2	RPCNT1	RPCNT0
R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8
–	–	–	–	RREMAIN11	RREMAIN10	RREMAIN9	RREMAIN8
R	R	R	R	R	R	R	R
7	6	5	4	3	2	1	0
RREMAIN7	RREMAIN6	RREMAIN5	RREMAIN4	RREMAIN3	RREMAIN2	RREMAIN1	RREMAIN0
R	R	R	R	R	R	R	R

Table 32-83 ETHA0RXFINF2 register contents

Bit position	Bit name	Function
24:16	RPCNT[8:0]	Number of packets in the receive FIFO These bits indicate the number of packets (start flag to end flag) that exist in the receive FIFO. The value is incremented when one packet has been written by the MAC. (A packet that is discarded upon being received does not increment this value.) If a packet has been read from the DMAC for the Ethernet controller or if the packet is discarded, this value is decremented after the internal operation to discard the packet finishes.
11:0	RREMAIN[11:0]	These bits indicates the remaining receive FIFO capacity (in bytes). Bits 1 and 0 are always 00 because the data in the receive FIFO is aligned in 32-bit (4-byte) units.

(13) ETHA0RXFINF3 - Reception status 3 register

This register indicates the status of the receive FIFO while reception is stopped.

Access This register is read-only, in 32-bit units.

Address <ETHA0_base> + $024C_H$

Initial value Undefined

Caution Before rewriting a mode setting register related to reception or flow control, be sure to confirm that the ETHA0RXFINF3.RXSTOP bit is 1.

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
R	R	R	R	R	R	R	R
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
R	R	R	R	R	R	R	R
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	RXSTOP
R	R	R	R	R	R	R	R

Table 32-84 ETHA0RXFINF3 register contents

Bit position	Bit name	Function
0	RXSTOP	This bit is set if no data is in the receive FIFO while reception is stopped (by clearing ETHA0MFFCONT.RXEN). Before rewriting a mode setting register related to reception or flow control, confirm that the RXSTOP bit is 1. 0: Receive FIFO is operating. 1: Receive FIFO is stopped.

(14) ETHA0FSTATUS - FIFO status interrupt register

If an interrupt source that is not masked by the ETHA0FSTATMK register is generated, the INTETMFS interrupt (FIFO status interrupt) is generated. The INTETMFS interrupt signal is asserted until all bits in this register are cleared.

If an interrupt source masked by the ETHA0FSTATMK register is generated, the corresponding bit in this register is set. All the bits of the ETHA0FSTATUS register are cleared when the register is read.

Access This register is read-only, in 32-bit units.

Address <ETHA0_base> + 0250_H

Initial value Undefined

Caution The FIFO status interrupt status register is cleared when it is read. It is recommended to copy interrupt sources to variables so that multiple interrupt sources generated concurrently can be detected.

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	TACOF
R	R	R	R	R	R	R	R
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	RACOF
R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8
–	–	–	TSUP	TFNRTY	TFWE	–	–
R	R	R	R	R	R	R	R
7	6	5	4	3	2	1	0
RFFE	RSUP	–	RFWE	RFOF	–	RFFLW	RFZP
R	R	R	R	R	R	R	R

Table 32-85 ETHA0FSTATUS register contents

Bit position	Bit name	Function
24	TACOF	This bit is set when the ETHA0TXABTCNT register (TX abort counter) overflows.
16	RACOF	This bit is set when the ETHA0RXABTCNT register (RX abort counter) overflows.
12	TSUP	TX status update This bit is set when the transmission status is updated in the ETHA0TXSTMONI1 or ETHA0TXSTMONI2 register.
11	TFNRTY	Transmit FIFO abort (transmit FIFO no retry) This bit is set if transmission fails and the data in the transmit FIFO is discarded. In this case, ETHA0TXABTCNT is incremented.
10	TFWE	This bit is set if a transmit FIFO write error has occurred.
7	RFFE	Receive FIFO flag error This bit is set if handshaking is not correctly performed when receive data is written from the MAC to the receive FIFO. The receive packet and reception status are invalid but reception is not canceled. • If the reception status is updated before all receive data is stored in the receive FIFO, the end of the packet is assumed as soon as the reception status has been updated. • If the reception status is not updated after all receive data has been stored in the receive FIFO, the reception status is assumed to be all 0.
6	RSUP	This bit is set when the reception status monitor register (ETHA0RXSTMONI) is updated. A valid value can be read from the ETHA0RXSTMONI or ETHA0RXFINF1 register when this bit is 1.
4	RFWE	Receive FIFO write error This bit is set if a packet whose size is less than 32 bits (4 bytes) has been received and could not be written to the receive FIFO.
3	RFOF	This bit is set if the receive FIFO overflows.
1	RFFLW	This bit is set if the data in the receive FIFO reaches or exceeds the value specified by the ETHA0FLOWTH.FLOWTHR bits.
0	RFZP	This bit is set if the data in the receive FIFO reaches or exceeds the value specified by the ETHA0FLOWTH.ZPTHR bits.

(15) ETHA0FSTATMK - FIFO status interrupt mask register

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + 0254_H

Initial value Undefined

Caution Be sure to set bits 31 to 25, 23 to 17, and 15 to 13, 9, 8, 5, and 2 to “1”.

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	TACOF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	RACOF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
–	–	–	TSUP	TFNRTY	TFWE	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
RFFE	RSUP	–	RFWE	RFOF	–	RFFLW	RFZP
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-86 ETHA0FSTATMK register contents

Bit position	Bit name	Function
24	TACOF	0: Does not mask interrupt. 1: Masks interrupt.
16	RACOF	0: Does not mask interrupt. 1: Masks interrupt.
12	TSUP	0: Does not mask interrupt. 1: Masks interrupt.
11	TFNRTY	0: Does not mask interrupt. 1: Masks interrupt.
10	TFWE	0: Does not mask interrupt. 1: Masks interrupt.
7	RFFE	0: Does not mask interrupt. 1: Masks interrupt.
6	RSUP	0: Does not mask interrupt. 1: Masks interrupt.
4	RFWE	0: Does not mask interrupt. 1: Masks interrupt.
3	RFOF	0: Does not mask interrupt. 1: Masks interrupt.
1	RFFLW	0: Does not mask interrupt. 1: Masks interrupt.
0	RFZP	0: Does not mask interrupt. 1: Masks interrupt.

(16) ETHA0TXSTATUS - Transmission status interrupt register

This register stores the cumulative result of the transmission status. If an interrupt that is not masked by the ETHA0TXSTATMK register is generated, the INTETMTS interrupt (transmission status interrupt) is generated. The INTETMTS interrupt signal is asserted until all bits in this register are cleared.

If an interrupt source masked by the ETHA0TXSTATMK register is generated, the corresponding bit in this register is set. All the bits of the ETHA0TXSTATUS register are cleared when the register is read.

Access This register is read-only, in 32-bit units.

Address <ETHA0_base> + 0258_H

Initial value Undefined

Caution The transmission status interrupt register is cleared when it is read. It is recommended to copy interrupt sources to variables so that multiple interrupt sources generated concurrently can be detected.

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
R	R	R	R	R	R	R	R
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	TAB
R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
R	R	R	R	R	R	R	R
7	6	5	4	3	2	1	0
TGNT	LCOL	ECOL	TEDFR	TDFR	TFLOR	TFLER	TCRCE
R	R	R	R	R	R	R	R

Table 32-87 ETHA0TXSTATUS register contents

Bit position	Bit name	Function
16	TAB	This bit is set if transmission has been aborted.
7	TGNT	This bit is set if a packet longer than the value specified by ETHA0LMAX has been transmitted (TAB source). This bit is not set if ETHA0MACC1.HUGEN is 1.
6	LCOL	This bit is set if a late collision has been detected (TAB source).
5	ECOL	This bit is set if collisions have occurred exceeding the specified maximum number (TAB source).
4	TEDFR	This bit is set if an excessive transmission delay has been detected (TAB source).
3	TDFR	This bit is set if a transmission delay has occurred.
2	TFLOR	This bit is set if a packet whose length field is greater than 1,500 bytes is detected. This bit is also set when a VLAN packet pause control frame is transmitted. This bit is not set if ETHA0MACC1.FLCHT is 0.
1	TFLER	This bit is set if a packet whose length field does not match the data field length is detected. This bit is not set if ETHA0MACC1.FLCHT is 0. If the length field exceeds 1,500 bytes, TFLOR is set and TFLER is not.
0	TCRCE	This bit is set if a CRC error occurred. This bit is set if transmission is performed with the CRC automatic appending mode disabled (the ETHA0MACC1.PADEN and ETHA0MACC1.CRCEN bits are 0).

(17) ETHA0TXSTATMK - Transmission status interrupt mask register

This register is used to mask the transmission status interrupt (INTCTS).

If an interrupt source that is not masked by this register is generated, the INTCTS interrupt is generated. The INTCTS interrupt signal is asserted until all the related interrupt sources are cleared. If an interrupt source masked by the ETHA0TXSTATMK register is generated, the corresponding bit in the ETHA0TXSTATUS register is set.

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + $025C_H$

Initial value Undefined

Caution Be sure to set bits 31 to 17 and 15 to 8 to “1”.

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	TAB
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
TGNT	LCOL	ECOL	TEDFR	TDFR	TFLOR	TFLER	TCRCE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-88 ETHA0TXSTATMK register contents

Bit position	Bit name	Function
16	TAB	0: Does not mask interrupt. 1: Masks interrupt.
7	TGNT	0: Does not mask interrupt. 1: Masks interrupt.
6	LCOL	0: Does not mask interrupt. 1: Masks interrupt.
5	ECOL	0: Does not mask interrupt. 1: Masks interrupt.
4	TEDFR	0: Does not mask interrupt. 1: Masks interrupt.
3	TDFR	0: Does not mask interrupt. 1: Masks interrupt.
2	TFLOR	0: Does not mask interrupt. 1: Masks interrupt.
1	TFLER	0: Does not mask interrupt. 1: Masks interrupt.
0	TCRCE	0: Does not mask interrupt. 1: Masks interrupt.

(18) ETHA0RXSTATUS - Reception status interrupt register

This register stores the cumulative result of the reception status. If an interrupt source that is not masked by the ETHA0RXSTATMK register is generated, the INTETMRS interrupt is generated. The INTETMRS interrupt signal is asserted until all bits in this register are cleared.

If an interrupt source masked by the ETHA0RXSTATMK register is generated, the corresponding bit in this register is set.

This register is not affected by the setting of ETHA0RXERSEL (receive error selection register).

All the bits of the ETHA0RXSTATUS register are cleared when the register is read.

Access This register is read-only, in 32-bit units.

Address <ETHA0_base> + 0260_H

Initial value Undefined

Caution The reception status interrupt status register is cleared when it is read. It is recommended to copy interrupt sources to variables so that multiple interrupt sources generated concurrently can be detected.

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
R	R	R	R	R	R	R	R
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8
–	RLENE	VLAN	USOP	RPCF	RCFR	DBNB	RLOR
R	R	R	R	R	R	R	R
7	6	5	4	3	2	1	0
RLER	RCRCE	RXER	CEPS	REPS	PAIG	TXRX	DVCF
R	R	R	R	R	R	R	R

Table 32-89 ETHA0RXSTATUS register contents

Bit position	Bit name	Function
14	RLENE	Occurrence of a receive packet length error This bit is set if the received packet size is less than 64 bytes or greater than 1,518 bytes (or less than 64 bytes or greater than 1,522 bytes for a VLAN packet).
13	VLAN	Reception of a VLAN packet This bit is set if a packet whose TPID field matches ETHA0VLTP.[a]
12	USOP	Reception of an undefined opcode control packet[b]
11	RPCF	Reception of a pause control packet[b]
10	RCFR	Reception of a control packet[b]
9	DBNB	Reception of a packet containing dribble nibble
8	RLOR	Reception of a packet that has a length field exceeding 1,500 bytes[c]
7	RLER	Detection of a packet whose length field does not match the data field length[c, d]
6	RCRCE	Occurrence of a receive CRC error
5	RXER	Detection of RXER
4	CEPS	Detection of a false carrier[e]
3	REPS	Reception of a packet in which the total size of the preamble and SFD or the data field size is one nibble[e, f]
2	PAIG	This bit is set if one of the following events occurred after the previous packet was received.[e] • A carrier longer than 6,072 nibbles (3,036 bytes) has been detected. • The next packet with IFG + preamble + SFD has been received before the time required to transmit 80 bits has elapsed after a packet has been received. • An illegal preamble or SFD has been received while the pure preamble data check is enabled (the ETHA0MACC1.PUREP bit is set).
1	TXRX	This bit is set if transmission starts (a collision occurs) during half-duplex reception (immediately after starting reception).
0	DVCF	This bit is set if the received packet is a valid control packet (that does not contain an error).

a) This bit is not set if a CRC error or RXER has occurred.
b) This bit is not set if a CRC error has occurred.
c) This bit is not set if ETHA0MACC1.FLCHT is 0.
d) If the length field exceeds 1,500 bytes, RLOR is set and RLER is not.
e) This bit indicates that this event occurred between when the reception status was updated previously and when it is updated this time.
f) A packet in which this event has occurred is ignored and not transferred to the upstream system.

(19) ETHA0RXSTATMK - Reception status interrupt mask register

This register is used to mask the reception status interrupt (INTETMRS).

If an interrupt source that is not masked by this register is generated, INTETMRS is generated. The INTETMRS interrupt signal is asserted until all the related interrupt sources are cleared. If an interrupt source masked by this register is generated, the corresponding bit in the ETHA0RXSTATUS register is set.

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + 0264_H

Initial value Undefined

Caution Be sure to set bits 31 to 15 to “1”.

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
–	RLENE	VLAN	USOP	RPCF	RCFR	DBNB	RLOR
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
RLER	RCRCE	RXER	CEPS	REPS	PAIG	TXRX	DVCF
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-90 ETHA0RXSTATMK register contents

Bit position	Bit name	Function
14	RLENE	0: Does not mask interrupt. 1: Masks interrupt.
13	VLAN	0: Does not mask interrupt. 1: Masks interrupt.
12	USOP	0: Does not mask interrupt. 1: Masks interrupt.
11	RPCF	0: Does not mask interrupt. 1: Masks interrupt.
10	RCFR	0: Does not mask interrupt. 1: Masks interrupt.
9	DBNB	0: Does not mask interrupt. 1: Masks interrupt.
8	RLOR	0: Does not mask interrupt. 1: Masks interrupt.
7	RLER	0: Does not mask interrupt. 1: Masks interrupt.
6	RCRCE	0: Does not mask interrupt. 1: Masks interrupt.
5	RXER	0: Does not mask interrupt. 1: Masks interrupt.
4	CEPS	0: Does not mask interrupt. 1: Masks interrupt.
3	REPS	0: Does not mask interrupt. 1: Masks interrupt.
2	PAIG	0: Does not mask interrupt. 1: Masks interrupt.
1	TXRX	0: Does not mask interrupt. 1: Masks interrupt.
0	DVCF	0: Does not mask interrupt. 1: Masks interrupt.

(20) ETHA0TXABTCNT - TX abort counter

This is a transmission abort counter. It counts the number of packets that have resulted in a MAC transmission error (including an underrun).

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + 0270_H

Initial value Undefined

Caution Be sure to set bits 31 to 16 to “0”.

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
TABCNT15	TABCNT14	TABCNT13	TABCNT12	TABCNT11	TABCNT10	TABCNT9	TABCNT8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
TABCNT7	TABCNT6	TABCNT5	TABCNT4	TABCNT3	TABCNT2	TABCNT1	TABCNT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-91 ETHA0TXABTCNT register contents

Bit position	Bit name	Function
15:0	TABCNT[15:0]	Transmission abort count These bits count the number of packets that have resulted in a MAC transmission error (including an underrun). This counter is not incremented when ETHA0MFFCONT.TABT is set to retransmit an aborted packet. Packets are counted after 68 bytes have been transferred because they are not retransmitted by a retry request (usually, the MAC does not issue a retry request after 64 bytes have been transferred). If the count value overflows, the value of these bits returns to 0 and the ETHA0FSTATUS.TACOF bit is set. These bits are not cleared by resetting the transmission circuit (by setting TFRST and TFFLSH).

(21) ETHA0RXABTCNT - RX abort counter

This is a reception abort counter. These bits count the number of receive packets that have been discarded because of the status of the receive packet, the status of the receive FIFO, address filtering by the MAC, and reception of a control packet.

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + 0274_H

Initial value Undefined

Caution Be sure to set bits 31 to 16 to “0”.

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
TABCNT15	TABCNT14	TABCNT13	TABCNT12	TABCNT11	TABCNT10	TABCNT9	TABCNT8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
TABCNT7	TABCNT6	TABCNT5	TABCNT4	TABCNT3	TABCNT2	TABCNT1	TABCNT0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-92 ETHA0RXABTCNT register contents

Bit position	Bit name	Function
15:0	TABCNT[15:0]	Reception abort count These bits count the number of receive packets that have been discarded because of the status of the receive packet, the status of the receive FIFO, address filtering by the MAC, and reception of a control packet. If the count value overflows, the value of these bits returns to 0 and the ETHA0FSTATUS.RACOF bit is set. These bits are not cleared by resetting the reception circuit (by setting RFRST and RFFLSH).

32.4.4 DMAC control registers for Ethernet controller

(1) ETHA0MODE - Core function control register

This register is used to analyze the reception start descriptor and transmission start descriptor.

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + 0300_H

Initial value Undefined

Caution Be sure to set bits 31 to 19 and 16 to 0 to “0”.

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
–	–	–	–	–	RXS	TXS	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-93 ETHA0MODE register contents

Bit position	Bit name	Function
18	RXS	Reception enable bit The RXS bit is used to start analyzing the reception start descriptor. This bit is automatically cleared after 1 is written to it.
17	TXS	Transmission enable bit The TXS bit is used to start analyzing the transmission start descriptor. This bit is automatically cleared after 1 is written to it.

(2) ETHA0INTMS - Interrupt control register

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + 0304_H

Initial value Undefined

Caution Be sure to set bits 31 to 28, 23 to 20, 15 to 11, and 7 to 3 to “0”.

Note The RBEI, RECI, RXI, TBEI, TECI, and TXI bits of the ETHA0INTMS register are cleared when they are read.

31	30	29	28	27	26	25	24
–	–	–	–	–	RBEMSK	RECMSK	RXMSK
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
–	–	–	–	–	RBEI	RECI	RXI
R/W	R/W	R/W	R/W	R/W	R	R	R
15	14	13	12	11	10	9	8
–	–	–	–	–	TBEMSK	TECMSK	TXMSK
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
–	–	–	–	–	TBEI	TECI	TXI
R/W	R/W	R/W	R/W	R/W	R	R	R

Table 32-94 ETHA0INTMS register contents (1/2)

Bit position	Bit name	Function
27	RUSMSK	This bit specifies whether to mask the RUPI interrupt indicated by bit 19. 0: Does not mask interrupt. 1: Masks interrupt.
26	RBEMSK	This bit specifies whether to mask the RBEI interrupt indicated by bit 18. 0: Does not mask interrupt. 1: Masks interrupt.
25	RECMSK	This bit specifies whether to mask the RECI interrupt indicated by bit 17. 0: Does not mask interrupt. 1: Masks interrupt.
24	RXMSK	This bit specifies whether to mask the RXI interrupt indicated by bit 16. 0: Does not mask interrupt. 1: Masks interrupt.
19	RUPI	This bit indicates whether a pause interrupt specified by the “U” (used) bit of a receive descriptor has been generated. It is cleared when read. 0: No interrupt generated. 1: Interrupt generated.
18	RBEI	This bit indicates whether a receive data buffer access error interrupt has been generated. It is cleared when read. 0: No interrupt generated. 1: Interrupt generated.

Table 32-94 ETHA0INTMS register contents (2/2)

Bit position	Bit name	Function
17	RECI	This bit indicates whether a reception (DMA) end of chain interrupt has been generated. It is cleared when read. 0: No interrupt generated. 1: Interrupt generated.
16	RXI	This bit indicates whether a packet reception (DMA) completion interrupt has been generated. It is cleared when read. 0: No interrupt generated. 1: Interrupt generated.
11	TUSMSK	This bit specifies whether to mask the TUPI interrupt indicated by bit 3. 0: Does not mask interrupt. 1: Masks interrupt.
10	TBEMSK	This bit specifies whether to mask the TBEI interrupt indicated by bit 2. 0: Does not mask interrupt. 1: Masks interrupt.
9	TECMSK	This bit specifies whether to mask the TECI interrupt indicated by bit 1. 0: Does not mask interrupt. 1: Masks interrupt.
8	TXMSK	This bit specifies whether to mask the TXI interrupt indicated by bit 0. 0: Does not mask interrupt. 1: Masks interrupt.
3	TUPI	This bit indicates whether a pause interrupt specified by the “U” (used) bit of a transmit descriptor has been generated. It is cleared when read. 0: No interrupt generated. 1: Interrupt generated.
2	TBEI	This bit indicates whether a transmit data buffer access error interrupt has been generated. It is cleared when read. 0: No interrupt generated. 1: Interrupt generated.
1	TECI	This bit indicates whether a transmission (DMA) end of chain interrupt has been generated. It is cleared when read. 0: No interrupt generated. 1: Interrupt generated.
0	TXI	This bit indicates whether a packet transmission (DMA) completion interrupt has been generated. It is cleared when read. 0: No interrupt generated. 1: Interrupt generated.

(3) ETHA0TRANSCTL - Transfer control register

This register is used to control transfer by the DMAC for the Ethernet controller.

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + 0308_H

Initial value Undefined

Caution Be sure to set bits 31 to 26, 23 to 18, and 15 to 1 to “0”.

31	30	29	28	27	26	25	24
–	–	–	–	–	–	RXEN_STA	TXEN_STA
R/W	R/W	R/W	R/W	R/W	R/W	R	R
23	22	21	20	19	18	17	16
–	–	–	–	–	–	RXEN	TXEN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	RXCHKSMEN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-95 ETHA0TRANSCTL register contents

Bit position	Bit name	Function
25	RXEN_STA	This bit indicates the reception status. 0: Reception is not in progress (IDLE status). 1: Reception is in progress. Reception stops if an RBEI or RECI interrupt, which is controlled by the interrupt register (ETHA0INTMS), is generated. At this time, this bit is cleared.
24	TXEN_STA	This bit indicates the transmission status. 0: Transmission is not in progress (IDLE status). 1: Transmission is in progress. Transmission stops if a TBEI or TECI interrupt, which is controlled by the interrupt register (ETHA0INTMS), is generated. At this time, this bit is cleared.
17	RXEN	This bit specifies whether to enable reception. 0: Disables reception (DMA reception stops.) 1: Enables reception.
16	TXEN	This bit specifies whether to enable transmission. 0: Disables transmission (DMA transmission stops). 1: Enables transmission.
0	RXCHKSMEN	This bit enables or disables the receive checksum appending function. 0: Disables the receive check sum appending function. 1: Enables the receive check sum appending function.

(4) ETHA0SFTRST - Software reset setting register

This register is used to execute a software reset for the DMAC for the Ethernet controller.

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + $030C_H$

Initial value Undefined

Caution Be sure to set bits 31 to 1 to “0”.

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	SFTRST
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-96 ETHA0SFTRST register contents

Bit position	Bit name	Function
0	SFTRST	Software reset When this bit is set, the DMAC for the Ethernet controller is reset. The receive checksum unit is also reset. This bit is automatically cleared after 1 is written to it.

(5) ETHA0DMACM - DMA mode control register

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + 0310_H

Initial value Undefined

Caution Be sure to set bits 31 to 11, 7 to 5, and 3 to 0 to “0”, and bit 4 to “1”.

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
–	–	–	–	–	BURST2	BURST1	BURST0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-97 ETHA0DMACM register contents

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>10:8</td><td>BURST[2:0]</td><td>These bits specify the type of burst transfer.
<table>
<tr><th>BURST2</th><th>BURST1</th><th>BURST0</th><th>Type</th><th>Operation</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>SINGLE</td><td>Single transfer mode</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>INCR4</td><td>4-beat incremental burst transfer mode</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>INCR8</td><td>8-beat incremental burst transfer mode</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>INCR16</td><td>16-beat incremental burst transfer mode</td></tr>
<tr><td colspan="3">Other than above</td><td colspan="2">Setting prohibited</td></tr>
</table></td></tr>
</table>

(6) ETHA0RXDP - Receive descriptor pointer register

This register is used to specify the pointer position of the receive descriptor of the DMAC for the Ethernet controller.

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + 0320_H

Initial value FFFF $FFFC_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
RXDP31	RXDP30	RXDP29	RXDP28	RXDP27	RXDP26	RXDP25	RXDP24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
RXDP23	RXDP22	RXDP21	RXDP20	RXDP19	RXDP18	RXDP17	RXDP16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
RXDP15	RXDP14	RXDP13	RXDP12	RXDP11	RXDP10	RXDP9	RXDP8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
RXDP7	RXDP6	RXDP5	RXDP4	RXDP3	RXDP2	RXDP1	RXDP0
R/W	R/W	R/W	R/W	R/W	R/W	R	R

Table 32-98 ETHA0RXDP register contents

Bit position	Bit name	Function
31:0	RXDP[31:0]	These bits specify the pointer position of the receive descriptor. Specify the first address of the receive descriptor chain. Bits 1 and 0 are fixed to 00.

(7) ETHA0LSTRXDP - Last receive descriptor pointer register

This register indicates the last receive descriptor address of the DMAC for the Ethernet controller.

Access This register is read-only, in 32-bit units.

Address <ETHA0_base> + 0320_H

Initial value FFFF $FFFC_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
LSTRXDP31	LSTRXDP30	LSTRXDP29	LSTRXDP28	LSTRXDP27	LSTRXDP26	LSTRXDP25	LSTRXDP24
R	R	R	R	R	R	R	R
23	22	21	20	19	18	17	16
LSTRXDP23	LSTRXDP22	LSTRXDP21	LSTRXDP20	LSTRXDP19	LSTRXDP18	LSTRXDP17	LSTRXDP16
R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8
LSTRXDP15	LSTRXDP14	LSTRXDP13	LSTRXDP12	LSTRXDP11	LSTRXDP10	LSTRXDP9	LSTRXDP8
R	R	R	R	R	R	R	R
7	6	5	4	3	2	1	0
LSTRXDP7	LSTRXDP6	LSTRXDP5	LSTRXDP4	LSTRXDP3	LSTRXDP2	LSTRXDP1	LSTRXDP0
R	R	R	R	R	R	R	R

Table 32-99 ETHA0LSTRXDP register contents

Bit position	Bit name	Function
31:0	LSTRXDP[31:0]	These bits indicate the last receive descriptor pointer address. They hold the address of the last accessed receive descriptor. Bits 1 and 0 are fixed to 00.

(8) ETHA0TXDP - Transmit descriptor pointer register

This register is used to specify the pointer position of the transmit descriptor of the DMAC for the Ethernet controller.

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + 0328_H

Initial value FFFF $FFFC_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
TXDP31	TXDP30	TXDP29	TXDP28	TXDP27	TXDP26	TXDP25	TXDP24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
TXDP23	TXDP22	TXDP21	TXDP20	TXDP19	TXDP18	TXDP17	TXDP16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
TXDP15	TXDP14	TXDP13	TXDP12	TXDP11	TXDP10	TXDP9	TXDP8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
TXDP7	TXDP6	TXDP5	TXDP4	TXDP3	TXDP2	TXDP1	TXDP0
R/W	R/W	R/W	R/W	R/W	R/W	R	R

Table 32-100 ETHA0TXDP register contents

Bit position	Bit name	Function
31:0	TXDP[31:0]	These bits specify the pointer position of the transmit descriptor. Specify the first address of the transmit descriptor chain. Bits 1 and 0 are fixed to 00.

(9) ETHA0LSTTXDP - Last transmit descriptor pointer register

This register indicates the last transmit descriptor address of the DMAC for the Ethernet controller.

Access This register is read-only, in 32-bit units.

Address <ETHA0_base> + $032C_H$

Initial value $FFFF\ FFFC_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
LSTTXDP31	LSTTXDP30	LSTTXDP29	LSTTXDP28	LSTTXDP27	LSTTXDP26	LSTTXDP25	LSTTXDP24
R	R	R	R	R	R	R	R
23	22	21	20	19	18	17	16
LSTTXDP23	LSTTXDP22	LSTTXDP21	LSTTXDP20	LSTTXDP19	LSTTXDP18	LSTTXDP17	LSTTXDP16
R	R	R	R	R	R	R	R
15	14	13	12	11	10	9	8
LSTTXDP15	LSTTXDP14	LSTTXDP13	LSTTXDP12	LSTTXDP11	LSTTXDP10	LSTTXDP9	LSTTXDP8
R	R	R	R	R	R	R	R
7	6	5	4	3	2	1	0
LSTTXDP7	LSTTXDP6	LSTTXDP5	LSTTXDP4	LSTTXDP3	LSTTXDP2	LSTTXDP1	LSTTXDP0
R	R	R	R	R	R	R	R

Table 32-101 ETHA0LSTTXDP register contents

Bit position	Bit name	Function
31:0	LSTTXDP[31:0]	These bits indicate the last transmit descriptor pointer address. They hold the address of the last accessed transmit descriptor. Bits 1 and 0 are fixed to 00.

32.4.5 Control registers of DMAC for transmit checksum

(1) ETHA0CMODE - Transmit checksum unit function setting register

This register is used to analyze the reception start descriptor and transmission start descriptor.

Access This register can be read/written in 32-bit units.

Address <ETHA0C_base> + 0300_H

Initial value Undefined

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
–	–	–	–	–	–	TCH_RXS	TCH_TXS
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-102 ETHA0CMODE register contents

Bit position	Bit name	Function
17	TCH_RXS	The TCH_RXS bit is used to start analyzing the reception start descriptor. This bit is cleared immediately after 1 is written to it.
16	TCH_TXS	The TCH_TXS bit is used to start analyzing the transmission start descriptor. This bit is cleared immediately after 1 is written to it.

(2) ETHA0CINTMS - Transmit checksum interrupt register

This register indicates the status of and masks the INTSCRXTCH and INTSCTXTCH interrupts of the DMAC for the transmit checksum.

Access This register can be read/written in 32-bit units. Bits 18 to 16 and 2 to 0 can only be read, however.

Address <ETHA0C_base> + 0304_H

Initial value Undefined

Caution The TCH_RBEI, TCH_RECI, TCH_RXI, TCH_TEBI, TCH_TECI, and TCH_TXI bits of the transmit checksum interrupt register are cleared when they are read. It is recommended to copy interrupt sources to variables so that multiple interrupt sources generated concurrently can be detected.

31	30	29	28	27	26	25	24
–	–	–	–	–	TCH_RBEMSK	TCH_RECMSK	TCH_RXMSK
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
–	–	–	–	–	TCH_RBEI	TCH_RECI	TCH_RXI
R/W	R/W	R/W	R/W	R/W	R	R	R
15	14	13	12	11	10	9	8
–	–	–	–	–	TCH_TBEMSK	TCH_TECMSK	TCH_TXMSK
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
–	–	–	–	–	TCH_TBEI	TCH_TECI	TCH_TXI
R/W	R/W	R/W	R/W	R/W	R	R	R

Table 32-103 ETHA0CINTMS register contents

Bit position	Bit name	Function
26	TCH_RBEMSK	This bit specifies whether to mask the TCH_RBEI interrupt indicated by bit 18. 0: Does not mask interrupt. 1: Masks interrupt.
25	TCH_RECMSK	This bit specifies whether to mask the TCH_RECI interrupt indicated by bit 17. 0: Does not mask interrupt. 1: Masks interrupt.
24	TCH_RXMSK	This bit specifies whether to mask the TCH_RXI interrupt indicated by bit 16. 0: Does not mask interrupt. 1: Masks interrupt.
18	TCH_RBEI	This bit indicates whether a receive data buffer access error interrupt of INTSCRXTCH has been generated. It is cleared (0) when read. 0: No interrupt generated. 1: Interrupt generated.
17	TCH_RECI	This bit indicates whether a receive DMA end of chain interrupt of INTSCRXTCH has been generated. It is cleared (0) when read. 0: No interrupt generated. 1: Interrupt generated.
16	TCH_RXI	This bit indicates whether a packet reception DMA transfer completion interrupt of INTSCRXTCH has been generated. It is cleared (0) when read. 0: No interrupt generated. 1: Interrupt generated.
10	TCH_TBEMSK	This bit specifies whether to mask the TCH_TBEI interrupt indicated by bit 2. 0: Does not mask interrupt. 1: Masks interrupt.
9	TCH_TECMSK	This bit specifies whether to mask the TCH_TECI interrupt indicated by bit 1. 0: Does not mask interrupt. 1: Masks interrupt.
8	TCH_TXMSK	This bit specifies whether to mask the TCH_TXI interrupt indicated by bit 0. 0: Does not mask interrupt. 1: Masks interrupt.
2	TCH_TBEI	This bit indicates whether a transmit data buffer access error interrupt of INTSCTXTCH has been generated. It is cleared (0) when read. 0: No interrupt generated. 1: Interrupt generated.
1	TCH_TECI	This bit indicates whether a transmit DMA end of chain interrupt of INTSCTXTCH has been generated. It is cleared (0) when read. 0: No interrupt generated. 1: Interrupt generated.
0	TCH_TXI	This bit indicates whether a packet transmission DMA transfer completion interrupt of INTSCTXTCH has been generated. It is cleared (0) when read. 0: No interrupt generated. 1: Interrupt generated.

(3) ETHA0CTRANSCTL - Transmit checksum transfer control register

This register is used to control transmission/reception of the DMAC for the transmit checksum.

Access This register can be read/written in 32-bit units. Bits 25 and 24 can only be read, however.

Address <ETHA0C_base> + 0308_H

Initial value Undefined

31	30	29	28	27	26	25	24
–	–	–	–	–	–	TCH_RXEN_STA	TCH_TXEN_STA
R/W	R/W	R/W	R/W	R/W	R/W	R	R
23	22	21	20	19	18	17	16
–	–	–	–	–	–	TCH_RXEN	TCH_TXEN
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-104 ETHA0CTRANSCTL register contents

Bit position	Bit name	Function
25	TCH_RXEN_STA	This bit indicates the reception status. 0: Reception is not in progress (IDLE status). 1: Reception is in progress. Reception stops if a TCH_RBEI or TCH_RECI interrupt, which is controlled by the transmit checksum interrupt register (ETHA0CINTMS), is generated. At this time, this bit is cleared.
24	TCH_TXEN_STA	This bit indicates the transmission status. 0: Transmission is not in progress (IDLE status). 1: Transmission is in progress. Transmission stops if a TCH_TBEI or TCH_TECI interrupt, which is controlled by the transmit checksum interrupt register (ETHA0CINTMS), is generated. At this time, this bit is cleared.
17	TCH_RXEN	This bit specifies whether to enable reception. 0: Disables reception (DMA reception stops.) 1: Enables reception.
16	TCH_TXEN	This bit specifies whether to enable transmission. 0: Disables transmission (DMA transmission stops). 1: Enables transmission.

(4) ETHA0CSFTRST - Transmit checksum software reset register

This register is used to execute a software reset for the DMAC for the transmit checksum.

Access This register can be read/written in 32-bit units.

Address <ETHA0C_base> + $030C_H$

Initial value Undefined

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
–	–	–	–	–	–	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	TCH_ SFTRST
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-105 ETHA0CSFTRST register contents

Bit position	Bit name	Function
0	TCH_SFTRST	This is a software reset bit for the DMAC for the transmit checksum. When this bit is set, the DMAC for the transmit checksum and the transmit checksum unit are reset. This bit is automatically cleared after the reset is executed.

(5) ETHA0CDMACM - Transmit checksum DMAC control mode setting register

This register is used to specify the burst transfer type of DMA by the DMAC for the transmit checksum.

Access This register can be read/written in 32-bit units.

Address <ETHA0C_base> + 0310_H

Initial value Undefined

31	30	29	28	27	26	25	24
–	–	–	–	–	–	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
–	–	–	–	–	–	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
–	–	–	–	–	BURST2	BURST1	BURST0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
–	–	–	–	–	–	–	–
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 32-106 ETHA0CDMACM register contents

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>10:8</td><td>BURST [2:0]</td><td>These bits specify the type of burst transfer for the internal system bus.
<table>
<tr><th>BURST2</th><th>BURST1</th><th>BURST0</th><th>Transfer type</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>Single transfer mode</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>4-beat incremental burst transfer mode</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>8-beat incremental burst transfer mode</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>16-beat incremental burst transfer mode</td></tr>
<tr><td colspan="3">Other than above</td><td>Setting prohibited</td></tr>
</table>
</td></tr>
</table>

(6) ETHA0CRXDP - Transmit checksum receive descriptor pointer register

This register is used to specify the pointer position of the receive descriptor of the DMAC for the transmit checksum. The lower 2 bits are fixed to 00_B.

Access This register can be read/written in 32-bit units.

Address <ETHA0C_base> + 0320_H

Initial value FFFF $FFFC_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
TCH_RXDP31	TCH_RXDP30	TCH_RXDP29	TCH_RXDP28	TCH_RXDP27	TCH_RXDP26	TCH_RXDP25	TCH_RXDP24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
TCH_RXDP23	TCH_RXDP22	TCH_RXDP21	TCH_RXDP20	TCH_RXDP19	TCH_RXDP18	TCH_RXDP17	TCH_RXDP16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
TCH_RXDP15	TCH_RXDP14	TCH_RXDP13	TCH_RXDP12	TCH_RXDP11	TCH_RXDP10	TCH_RXDP9	TCH_RXDP8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
TCH_RXDP7	TCH_RXDP6	TCH_RXDP5	TCH_RXDP4	TCH_RXDP3	TCH_RXDP2	TCH_RXDP1	TCH_RXDP0
R/W	R/W	R/W	R/W	R/W	R/W	R	R

Table 32-107 ETHA0CRXDP register contents

Bit position	Bit name	Function
31:0	TCH_RXDP[31:0]	These bits specify the pointer position of the receive descriptor. Specify the first address of the receive descriptor chain. Bits 1 and 0 are fixed to 00.

(7) ETHA0CLSTRXDP - Transmit checksum last receive descriptor pointer register

This register indicates the last receive descriptor address of the DMAC for the transmit checksum. The lower 2 bits are fixed to 00_B.

Access This register can be read/written in 32-bit units.

Address <ETHA0C_base> + 0324_H

Initial value FFFF FFFC$_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
TCH_LSTRXDP31	TCH_LSTRXDP30	TCH_LSTRXDP29	TCH_LSTRXDP28	TCH_LSTRXDP27	TCH_LSTRXDP26	TCH_LSTRXDP25	TCH_LSTRXDP24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
TCH_LSTRXDP23	TCH_LSTRXDP22	TCH_LSTRXDP21	TCH_LSTRXDP20	TCH_LSTRXDP19	TCH_LSTRXDP18	TCH_LSTRXDP17	TCH_LSTRXDP16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
TCH_LSTRXDP15	TCH_LSTRXDP14	TCH_LSTRXDP13	TCH_LSTRXDP12	TCH_LSTRXDP11	TCH_LSTRXDP10	TCH_LSTRXDP9	TCH_LSTRXDP8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
TCH_LSTRXDP7	TCH_LSTRXDP6	TCH_LSTRXDP5	TCH_LSTRXDP4	TCH_LSTRXDP3	TCH_LSTRXDP2	TCH_LSTRXDP1	TCH_LSTRXDP0
R/W	R/W	R/W	R/W	R/W	R/W	R	R

Table 32-108 ETHA0CLSTRXDP register contents

Bit position	Bit name	Function
31:0	TCH_LSTRXDP[31:0]	These bits indicate the last receive descriptor address. They hold the address of the last accessed receive descriptor. Bits 1 and 0 are fixed to 00.

(8) ETHA0CTXDP - Transmit checksum transmit descriptor pointer register

This register is used to specify the pointer position of the transmit descriptor of the DMAC for the transmit checksum. The lower 2 bits are fixed to 00_B.

Access This register can be read/written in 32-bit units.

Address <ETHA0C_base> + 0328_H

Initial value $FFFF\ FFFC_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
TCH_TXDP31	TCH_TXDP30	TCH_TXDP29	TCH_TXDP28	TCH_TXDP27	TCH_TXDP26	TCH_TXDP25	TCH_TXDP24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
TCH_TXDP23	TCH_TXDP22	TCH_TXDP21	TCH_TXDP20	TCH_TXDP19	TCH_TXDP18	TCH_TXDP17	TCH_TXDP16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
TCH_TXDP15	TCH_TXDP14	TCH_TXDP13	TCH_TXDP12	TCH_TXDP11	TCH_TXDP10	TCH_TXDP9	TCH_TXDP8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
TCH_TXDP7	TCH_TXDP6	TCH_TXDP5	TCH_TXDP4	TCH_TXDP3	TCH_TXDP2	TCH_TXDP1	TCH_TXDP0
R/W	R/W	R/W	R/W	R/W	R/W	R	R

Table 32-109 ETHA0CTXDP register contents

Bit position	Bit name	Function
31:0	TCH_TXDP[31:0]	These bits specify the pointer position of the transmit descriptor. Specify the first address of the transmit descriptor chain. Bits 1 and 0 are fixed to 00.

(9) ETHA0CLSTTXDP - Transmit checksum last transmit descriptor pointer register

This register indicates the last transmit descriptor address of the DMAC for the transmit checksum. The lower 2 bits are fixed to 00_B.

Access This register can be read/written in 32-bit units.

Address <ETHA0_base> + $032C_H$

Initial value FFFF $FFFC_H$. This register is initialized by any reset.

31	30	29	28	27	26	25	24
TCH_LSTTXDP31	TCH_LSTTXDP30	TCH_LSTTXDP29	TCH_LSTTXDP28	TCH_LSTTXDP27	TCH_LSTTXDP26	TCH_LSTTXDP25	TCH_LSTTXDP24
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
23	22	21	20	19	18	17	16
TCH_LSTTXDP23	TCH_LSTTXDP22	TCH_LSTTXDP21	TCH_LSTTXDP20	TCH_LSTTXDP19	TCH_LSTTXDP18	TCH_LSTTXDP17	TCH_LSTTXDP16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
15	14	13	12	11	10	9	8
TCH_LSTTXDP15	TCH_LSTTXDP14	TCH_LSTTXDP13	TCH_LSTTXDP12	TCH_LSTTXDP11	TCH_LSTTXDP10	TCH_LSTTXDP9	TCH_LSTTXDP8
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
7	6	5	4	3	2	1	0
TCH_LSTTXDP7	TCH_LSTTXDP6	TCH_LSTTXDP5	TCH_LSTTXDP4	TCH_LSTTXDP3	TCH_LSTTXDP2	TCH_LSTTXDP1	TCH_LSTTXDP0
R/W	R/W	R/W	R/W	R/W	R/W	R	R

Table 32-110 ETHA0CLSTTXDP register contents

Bit position	Bit name	Function
31:0	TCH_LSTTXDP[31:0]	These bits indicate the last transmit descriptor pointer address. They hold the address of the last accessed transmit descriptor. Bits 1 and 0 are fixed to 00.

32.5 MAC/FIFO/DMAC Function

32.5.1 Frame format

With Ethernet/IEEE802.3, data is transmitted and received as a packet or frame.

The Ethernet controller supports the following three types of frames.

- Basic frame
- VLAN frame
- Pause control frame

(1) Basic frame

The basic frame used with Ethernet consists of a preamble (PA), frame start delimiter (SFD), destination address (DA), source address (SA), type/length field (TYPE/LEN), data field (DATA), and frame check sequence (FCS).

The packet size is defined to be 64 bytes minimum and 1,518 bytes maximum, excluding the preamble (PA) and frame start delimiter (SFD).

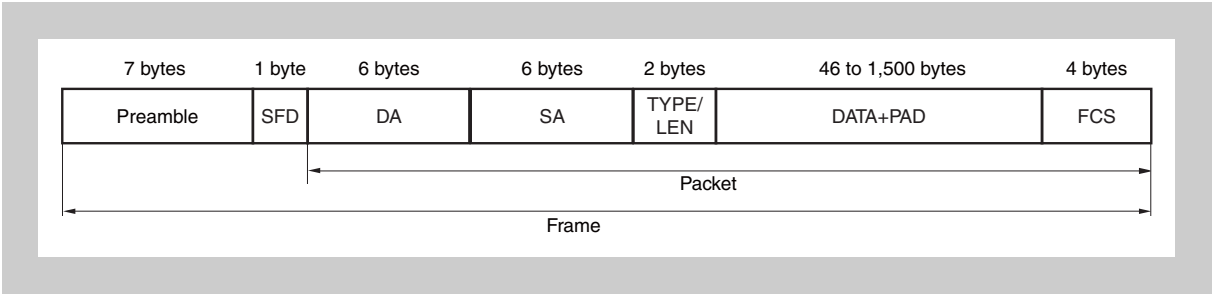


Figure 32-5 Basic frame structure

(a) Preamble and frame start delimiter (SFD)

The preamble and SFD consist of a repetition of 10 for 62 bits followed by 11 at the end, indicating the header of each frame.

(b) Destination address (DA)

The destination address field indicates the destination MAC address. A unicast address, multicast address, or broadcast address is written in this field.

(c) Source address (SA)

The source MAC address is written in this field.

(d) Type/length field

As an Ethernet frame, this field is a type field that indicates a protocol type. As an IEEE802.3 frame, this field is a length field that indicates the length of the data field.

(e) Data field

The data field size ranges from 46 bytes to 1,500 bytes.

Depending on the communication protocol, the data field might be divided to insert special header information. The Ethernet controller uses the data in this field only for calculating the CRC (Cycle Redundancy Check) for the FCS and does not check its contents.

(f) Frame check sequence (FCS)

The frame check sequence field is used to write a 32-bit CRC code to check the transfer data.

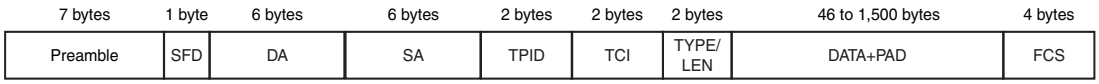
The Ethernet controller can automatically append a CRC to a transmit frame.

(2) VLAN frame

The structure of a VLAN frame (Qtag frame) is slightly different from that of a basic frame.

A 4-byte VLAN header is inserted immediately after the source address field. As a result, the minimum packet length of a VLAN frame is 64 bytes and the maximum packet length is 1,522 bytes.

The Ethernet controller has a VLAN frame detection function. If a transmit or receive packet is detected to be a VLAN frame, packet processing is performed based on the receive packet length.



Note TPID: Tag Protocol ID
TCI: Tag Control Information
TPID + TCI: VLAN header

Figure 32-6 VLAN frame structure

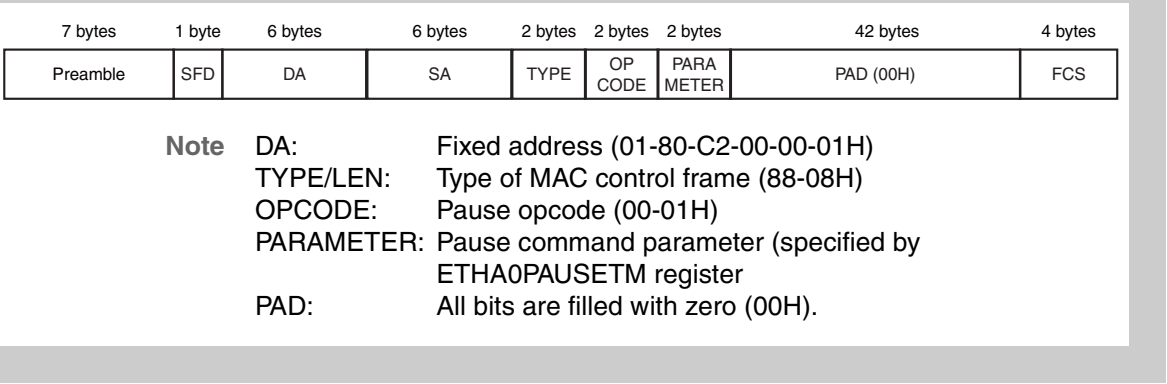
Caution The Ethernet controller recognizes the value set to the ETHA0VLTP.VLTP[15:0] bits as a VLAN frame (TPID). The default value is 0000_H. For details, refer to *(10) "ETHA0VLTP - VLAN type register" on page 2174 in 32.4.1 "MAC control registers"*.

(3) Pause control frame

The pause control frame is a 64-byte packet that has a dedicated format.

The destination address field has a fixed value of 01-80-C2-00-00-01_H.

The type/length field has a value of 8808H, which indicates a control frame, and the opcode has a value of 0001H, which indicates a pause control frame. The parameter field stores the value specified by the ETHA0PAUSETM register. The unused area following the parameter field is filled with PAD data consisting of zeros.



Note
DA: Fixed address (01-80-C2-00-00-01H)
TYPE/LEN: Type of MAC control frame (88-08H)
OPCODE: Pause opcode (00-01H)
PARAMETER: Pause command parameter (specified by ETHA0PAUSETM register
PAD: All bits are filled with zero (00H).

Figure 32-7 Pause control frame structure

The Ethernet controller can automatically transmit a pause control frame according to the amount of the remaining in the receive FIFO.

When a frame is received, the frame type is identified by the DA, TYPE, and OPCODE fields as shown in *Table 32-111 "Frame reception"*.

Table 32-111 Frame reception

DA	TYPE	OPCODE	Frame identified
01-80-C2-00-00-01	8808_H	0001_H	Pause frame
01-80-C2-00-00-01	8808_H	Other than 0001_H	Not supported
01-80-C2-00-00-01	Other than 8808_H	xxxx	Data frame
Unicast (station address)	8808_H	0001_H	Pause frame
Unicast (station address)	8808_H	Other than 0001_H	Not supported
Unicast (station address)	Other than 8808_H	xxxx	Data frame
Multicast	8808_H	xxxx	Not supported
Multicast	Other than 8808_H	xxxx	Data frame
Unicast (station address)	8808_H	xxxx	Not supported
Unicast (station address)	Other than 8808_H	xxxx	Data frame

(4) Pause control frame containing VLAN tag

The Ethernet controller does not support a pause control frame containing a VLAN tag.

It receives such a frame as an ordinary VLAN packet, but the RBRO and RLOR flags of the reception status monitor register (ETHA0RXSTMONI) are set (the RPCF and RCFR flags are not set).

(5) Envelope frame

The envelope frame format was added to IEEE802.3as (2005). Because it is a frame for 1,000 Mbps half-duplex communication, the Ethernet controller does not support this type of frame. If an envelope frame containing an EXTENSION field is received, a CRC error or length field mismatch occurs, or the receive FIFO overflows. Check the reception status and discard such a frame.

The Ethernet controller cannot transmit an envelope frame containing an EXTENSION field.

32.5.2 Frame transmission

The Ethernet controller generates a transmit frame as defined in IEEE802.3 from the transmit packet data stored in the transmit FIFO by the DMAC for the Ethernet controller via a DMA transfer, and then outputs that frame to a PHY device. If a collision is detected, the frame is retransmitted by using a random back-off algorithm. The status information of each transmit frame, such as excessive transmit delays and collisions exceeding the maximum number, is reflected in the ETHA0TXSTATUS register, and the number of times each event has occurred in all transmit frames is counted by statistics counters.

(1) Transmit frame

The transmit frame defined by IEEE802.3 consists of the following six fields (refer to *Figure 32-5 "Basic frame structure"*).

- Preamble (PA)
- Frame start delimiter (SFD)
- Destination address (DA)
- Source address (SA)
- Length field (LEN)
- Data and frame check sequence (FCS)

During transmission, the Ethernet controller generates a preamble, frame start delimiter, and FCS data.

(2) Transmission clock

The Ethernet controller operates in synchronization with the transmission clock (TXCLK) supplied by an external PHY device. Transmit packet data stored in the transmit FIFO via a DMA transfer is synchronized with TXCLK in the FIFO buffer, and then output to the PHY device. IEEE802.3 defines the frequency of TXCLK as 25 MHz±100 ppm at a data transfer rate of 100 Mbps, or 2.5 MHz±100 ppm at 10 Mbps.

(3) Carrier sense signal (CRS)

During half-duplex communication, if a carrier is detected (CRS = 1) after the Ethernet controller has stored transmit data in the FIFO buffer and transmission is enabled, the Ethernet controller postpones transmission until the end of the carrier (CRS = 0). After the carrier ends, transmission starts when the inter-packet gap (IPG) count specified by the ETHA0IPGT register has been reached.

If no carrier is detected (CRS = 0) when transmission is enabled and the IPG count is reached after the end of the previous carrier, transmission starts immediately.

When a frame is transmitted from the local side, the transmitted carrier sense signal is looped back from the external PHY device. If the transmitted carrier sense signal is masked by the external PHY device in the user-defined system,

the Ethernet controller detects a carrier sense error, but this does not affect the transmission.

(4) Collision detection (COL) and retransmission

If the Ethernet controller detects a collision during half-duplex communication, it transmits jam data (an error CRC) and then stops transmission.

If less than the maximum number of collisions (default value: 15) are detected in the collision window, transmission is kept waiting by a random back-off algorithm and data in the transmit FIFO is retransmitted. (In this case, no data is captured into the FIFO buffer again by DMA.)

If collisions exceeding the maximum number are detected or if a late collision (a collision detected outside the collision window) occurs, transmission is aborted and the transmit data is discarded.

(5) Inter-packet gap (IPG)

The IPG for successive data transmission from the local side is specified by the ETHA0IPGT register, and that for other cases is specified by the ETHA0IPGR register.

When transmission from the local side or the communicating device is finished, the Ethernet controller starts counting the IPG. If a request for the next transmission is issued from the FIFO buffer after transmission from the local side is finished and before the IPG count reaches the value of the ETHA0IPGT register, it is assumed that the data is to be transmitted successively (back to back), and transmission starts as soon as counting finishes.

If a packet is to be transmitted after transmission from the communicating device is finished, the IPG count is controlled by the ETHA0IPGR register. In the ETHA0IPGR register, the entire period of the IPG is specified for the IPGR2 bits and the interval time to detect a carrier in the first half of the IPG is specified for the IPGR1 bits. If a carrier is detected during the period specified for the IPGR1 bits, the Ethernet controller waits for the end of the carrier and then starts IPG counting from the beginning. If no carrier is detected during the period specified for the IPGR1 bits, transmission starts after the IPG period specified for the IPGR2 bits elapses.

If transmission does not start within the time required to transmit 24,288 bits (2.43 ms at 10 Mbps or 243.88 μs at 100 Mbps) after the next transmission request is received from the FIFO buffer, an excessive transmission delay is assumed, transmission is aborted, and the transmit data is discarded.

The set value of the ETHA0IPGT and ETHA0IPGR registers and the actual IPG period are calculated by the following expression.

[At 100 Mbps]

Back-to-back transmission:

IPG = (5 + IPGT) × 40 ns (default value: 960 ns)

Non back-to-back transmission:

IPG = (5 + IPGT) × 40 ns (default value: 960 ns)

Carrier sense time: (2 + IPGR1) × 40 ns (default value: 640 ns)

[At 10 Mbps]

Back-to-back transmission:

IPG = (5 + IPGT) × 400 ns (default value: 9.6 μs)

Non back-to-back transmission:

IPG = (5 + IPGR2) × 400 ns (default value: 9.6 μs)

Carrier sense time: (2 + IPGR1) × 400 ns (default value: 6.4 μs)

Caution According to the specification of IEEE802.3, set the IPG to 960 ns or more at a data transfer rate of 100 Mbps, or 9.6 μs or more at 10 Mbps.

The default value of the ETHA0IPGT and ETHA0IPGR registers is the minimum rated value and may be used as is.

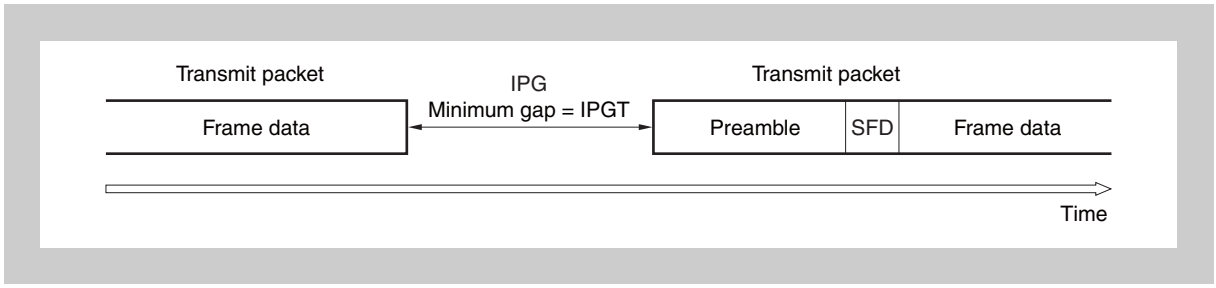


Figure 32-8 IPG during back-to-back transmission

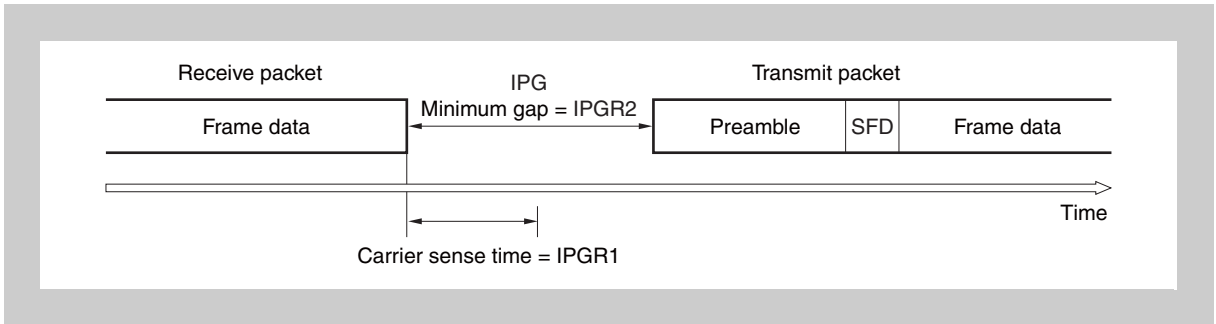


Figure 32-9 IPG during non back-to-back transmission

(6) Appending preamble, CRC, and PAD

A 7-byte preamble and a 1-byte frame start delimiter (SFD) are appended to the beginning of the transmit packet supplied from the FIFO buffer.

ETHA0MACC1.CRCEN	Operation
0	The transmit packet must end with a valid frame check sequence (FCS). The MAC checks the FCS, and if the FCS value is not correct, the MAC generates a transmission status interrupt (INTETMTS) to report an error.
1	An internally generated frame check sequence (FCS) is appended to the end of the transmit packet.

If the ETHA0MACC1.CRCEN bit is set, a frame check sequence (FCS) that has been internally generated is appended to the end of the transmit packet.

If the ETHA0MACC1.CRCEN bit is cleared, the transmit packet must end with a valid FCS. The Ethernet controller can check the FCS. If the value of the FCS is not correct, the Ethernet controller generates an Ethernet transmission status interrupt.

If the ETHA0MACC1.PADEN bit is set, zeros (PAD) are appended to a transmit packet shorter than 64 bytes (this is known as padding). In this case, the Ethernet controller appends the correct FCS to the end of the frame, regardless of the setting of the CRCEN bit.

If the ETHA0MACC2.APD or ETHA0MACC2.VPD bit is set when the ETHA0MACC1.PADEN bit is 1, PAD is appended to a VLAN frame. If the APD bit is set, only a packet that matches the VLAN type specified by the ETHA0VLTP register is regarded as a VLAN frame and is padded. If the VPD bit is set, all frames are regarded as VLAN frames and padded. A packet regarded as a VLAN frame is padded to extend its length to 68 bytes. The data that is appended as a pad is all 0.

(7) Aborting transmission

The Ethernet controller aborts transmission under the following conditions.

It does not abort transmission if the transmit FIFO underruns within the normal operating range.

- If more than the maximum number of collisions (MAX collision) occur
- If collision occurs outside the collision window (late collision)
- If there is an excessive transmission delay
- If a packet exceeding the frame length specified by the ETHA0LMAX register is to be transmitted.

  (If the ETHA0MACC1.HUGEN bit is 1, however, the transmit frame length is not limited.)

(8) Full-duplex operation

A full-duplex operation is enabled when the ETHA0MACC1.FULLD bit is set. The IPG is always the value specified by the ETHA0IPGT register. The FULLD signal is asserted if the ETHA0MACC1.FULLD bit is set, reporting to the external circuit that full-duplex operation is specified.

(9) Flow control and back pressure functions

The Ethernet controller has a flow control function (see *(1) "Flow control" on page 2291 in 32.5.4 "MAC control function"*) and a back pressure function (see *(2) "Back pressure" on page 2293 in 32.5.4 "MAC control function"*) associated with the receive FIFO. These functions are automatically activated to prevent the FIFO buffer from overflowing when the vacant capacity of the receive FIFO runs short.

(10) Transmission status update timing

The transmission status is updated in the timing shown below.

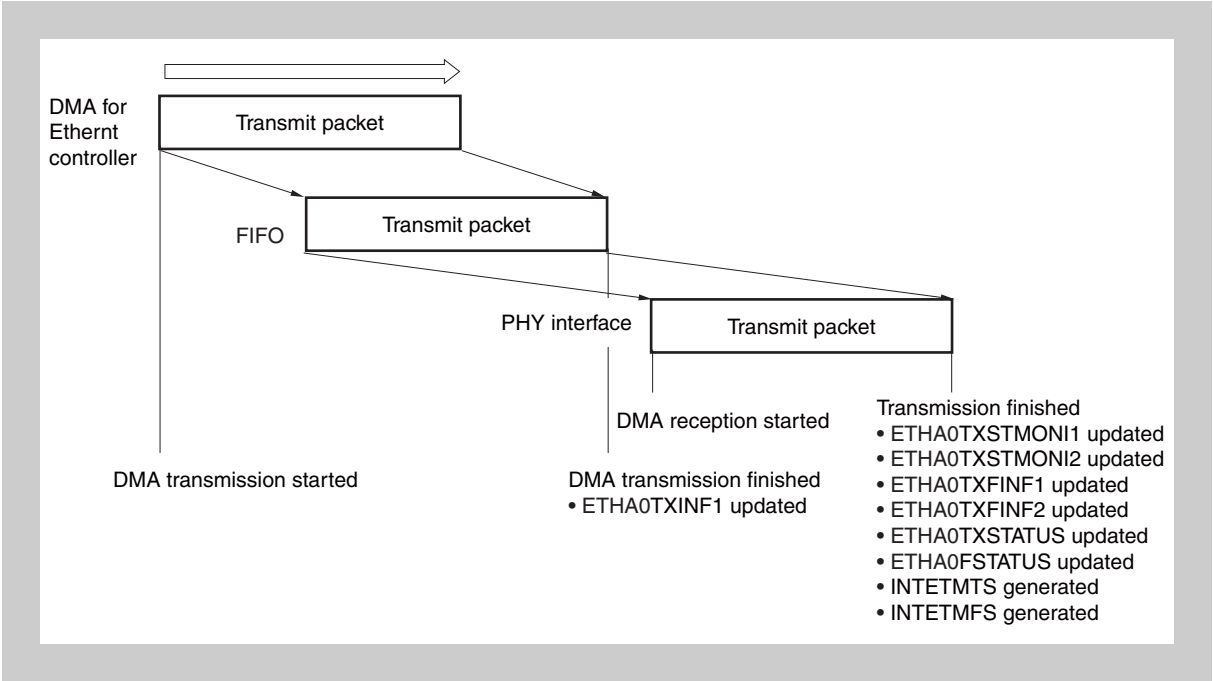


Figure 32-10 Transmission status update timing

32.5.3 Frame reception

The Ethernet controller generates a receive packet from a receive frame to be stored in the FIFO buffer, detects the SFD, checks the length field and FCS, and identifies whether the frame is a VLAN frame.

The status information of each receive packet is set to the reception status monitor register (ETHA0RXSTMONI) and the number of times each event has occurred in all receive frames is counted by statistics counters.

(1) Receive clock

The Ethernet controller receives data in synchronization with the reception clock (RXCLK) supplied by an external (PHY) device.

IEEE802.3 defines the frequency of RXCLK as 25 MHz±100 ppm at a data transfer rate of 100 Mbps, or 2.5 MHz±100 ppm at 10 Mbps.

(2) Reception of MII data

The Ethernet controller recognizes the RXD signal data as receive frames while the RXD[3:0] signal is asserted, and recognizes the end of the frames when the RXDV signal is deasserted.

(3) Detecting preamble and SFD

The Ethernet controller detects the preamble and SFD at the beginning of a receive frame and recognizes the data that follows as a receive packet.

(4) Checking length field

The Ethernet controller counts the length of a receive packet, and checks the length of the data field assuming the 2 bytes following the source address to be a length field. The result of this check can be read as the reception status from the ETHA0RXSTMONI register. An interrupt signal can be output to report a mismatch between the counted receive packet length and the length field value.

(5) CRC

The Ethernet controller calculates the 4-byte frame check sequence (FCS) from a receive packet and compares it with the FCS data appended to the end of the receive packet. The result of the comparison can be read from the ETHA0RXSTMONI register. An interrupt signal can be output to report a mismatch between the calculated 4-byte FCS and the FCS data at the end of the receive packet.

(6) Transmitting data to FIFO

The Ethernet controller assumes that a packet of 6 bytes or more is valid and discards a packet of less than 6 bytes.

(7) Detection of huge packet

If the ETHA0MACC1.HUGEN bit is cleared, the Ethernet controller receives only packets shorter than the maximum frame length specified by the ETHA0LMAX register (default value: 1,536 bytes), and stops receiving longer packets midway.

For details about the receivable packet length, refer to *Table 32-119 "Restrictions on receive FIFO"*.

(8) Detecting VLAN frame

The Ethernet controller checks whether received packets are VLAN frames.

If the value of the TPID field (the 2 bytes following the source address) of a received packet matches the value specified by the ETHA0VLTP register, the packet is recognized as a VLAN packet and the ETHA0RXSTMONI.VLAN flag is set. In a packet recognized as a VLAN frame, the 2 bytes immediately after the VLAN header (the 4 bytes following the source address) including the TPID field are regarded as the length field.

(9) Reception status update timing

The reception status is updated in the timing shown below.

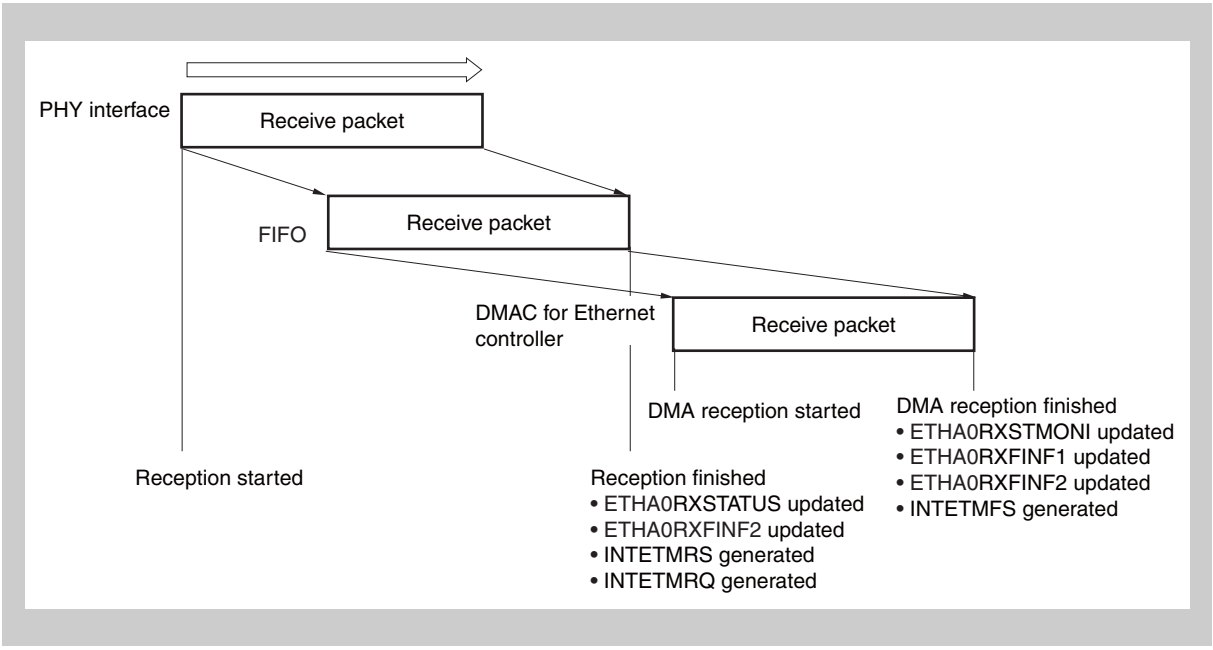


Figure 32-11 Reception status update timing

32.5.4 MAC control function

(1) Flow control

The Ethernet controller controls the flow by using the pause control frame defined in “IEEE802.3 Annex 31”.

The purpose of flow control is to decrease the frequency of frame transmission executed by the communicating device (link partner) connected point-to-point during full-duplex operation. The amount of data a system can receive and process is limited. If frames are received too frequently, system processing cannot keep up, so the receive FIFO might overflow. Flow control is used to avoid this situation.

When the Ethernet controller receives a pause control frame, it loads the value of the parameter field in the control frame to the pause timer in the MAC. If the value of the pause timer is not 0, the next transmission starts after the time set to the pause timer has elapsed.

If the value of the parameter field in the received pause control frame is 0 (a zero pause control frame), the value of the pause timer is cleared to 0 and transmission is resumed after the packet interval specified by the ETHA0IPGR register has elapsed.

To suppress data transmission from the link partner, the reserved multicast address (01-80-C2-00-00-01), pause opcode (00-01), and the pause timer value of the ETHA0PAUSETM register (PAUSETM_MAX) are transmitted as a pause control frame.

Starting transmission of a pause frame takes precedence over starting transmission of a basic frame. If a condition for transmitting a pause frame is satisfied while a basic frame is being transmitted, however, the pause frame is transmitted after transmission of the basic frame has finished.

The Ethernet controller controls the flow by setting ETHA0MFFCONT.FLOWCNT.

The necessity of transmitting a pause control frame request is judged according to the amount of data in the receive FIFO.

If the ETHA0MFFCONT.IVPAUSE bit is 0 in the full-duplex communication mode, the Ethernet controller monitors the amount of data in the receive FIFO during reception (see *(a)* in *Figure 32-12 “Flow control”*). And if the amount of data in the receive FIFO exceeds the value specified by the ETHA0FLOWTH.FLOWTHR bits, a pause control frame is transmitted (see *(b)* in *Figure 32-12 “Flow control”*).

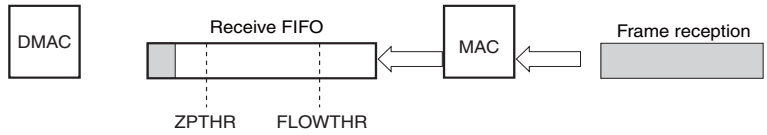
If the ETHA0MFFCONT.IVPAUSE bit is 1, the pause control frame is continually transmitted at the interval specified by the ETHA0PAUSETM.IPTIME bits as long as the amount of data in the receive FIFO exceeds the value specified by the ETHA0FLOWTH.FLOWTHR bits.

The Ethernet controller monitors the quantity of data in the receive FIFO even while receive data is being transferred by DMA (see *(c)* in *Figure 32-12 “Flow control”*).

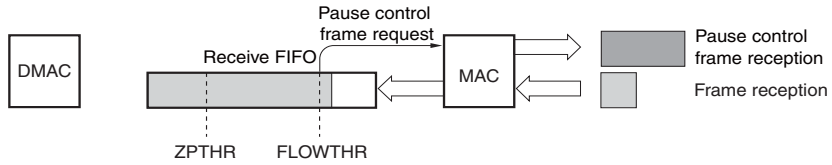
If the ETHA0MFFCONT.ZEROPAUSE bit is 1, a zero pause control frame is transmitted if the amount of data in the receive FIFO falls below the value specified by the ETHA0FLOWTH.ZPTHR bits (see *(d)* in *Figure 32-12 “Flow control”*).

If the ETHA0MFFCONT.ZEROPAUSE bit is 0, a zero pause control frame is not transmitted.

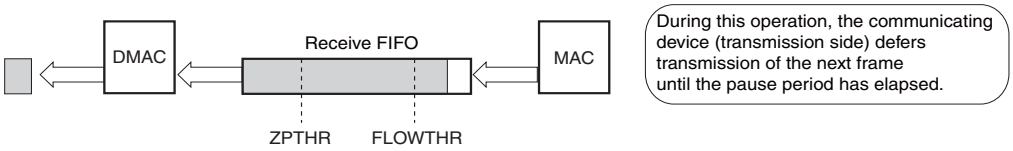
(a) When a frame is received, data is stored in the receive FIFO.



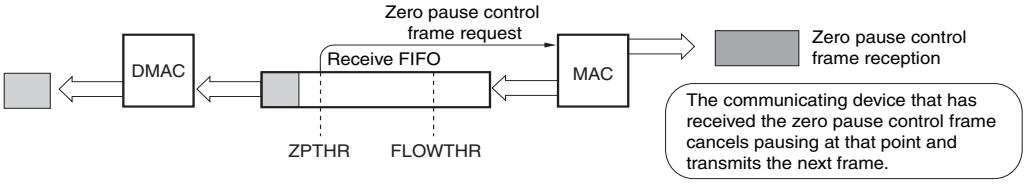
(b) When the amount of data in the receive FIFO exceeds the value specified by FLOWTHR, a pause control frame is transmitted.



(c) The data in the receive FIFO is read by DMA.



(d) When the amount of data in the receive FIFO falls below the value specified by ZPTHR, a zero pause control frame is transmitted.



Note The shaded part ▭ of the receive FIFO in the above figures indicates the amount of data in the receive FIFO, not how data is actually stored.

Figure 32-12 Flow control

(2) Back pressure

This function is available only during half-duplex operation.

If the amount of data in the receive FIFO exceeds the value specified by the ETHA0FLOWTH.FLOWTHR bits when the ETHA0MFFCONT.FLOWCNT bit is 1 and the ETHA0MACC1.FULLD bit is 0, the back pressure function is enabled (see *(b)* in *Figure 32-13 "Back pressure control"*).

If the next frame is received in this status, a collision is intentionally generated by transmitting a dummy packet, prompting the communicating device to retransmit the frame (see *(c)* in *Figure 32-13 "Back pressure control"*).

A collision that occurs in the back pressure status is not included in the number of collisions.

The back pressure status is released if the amount of data in the receive FIFO falls below the value specified by the FLOWTHR bits (see *(d)* in *Figure 32-13 "Back pressure control"*).

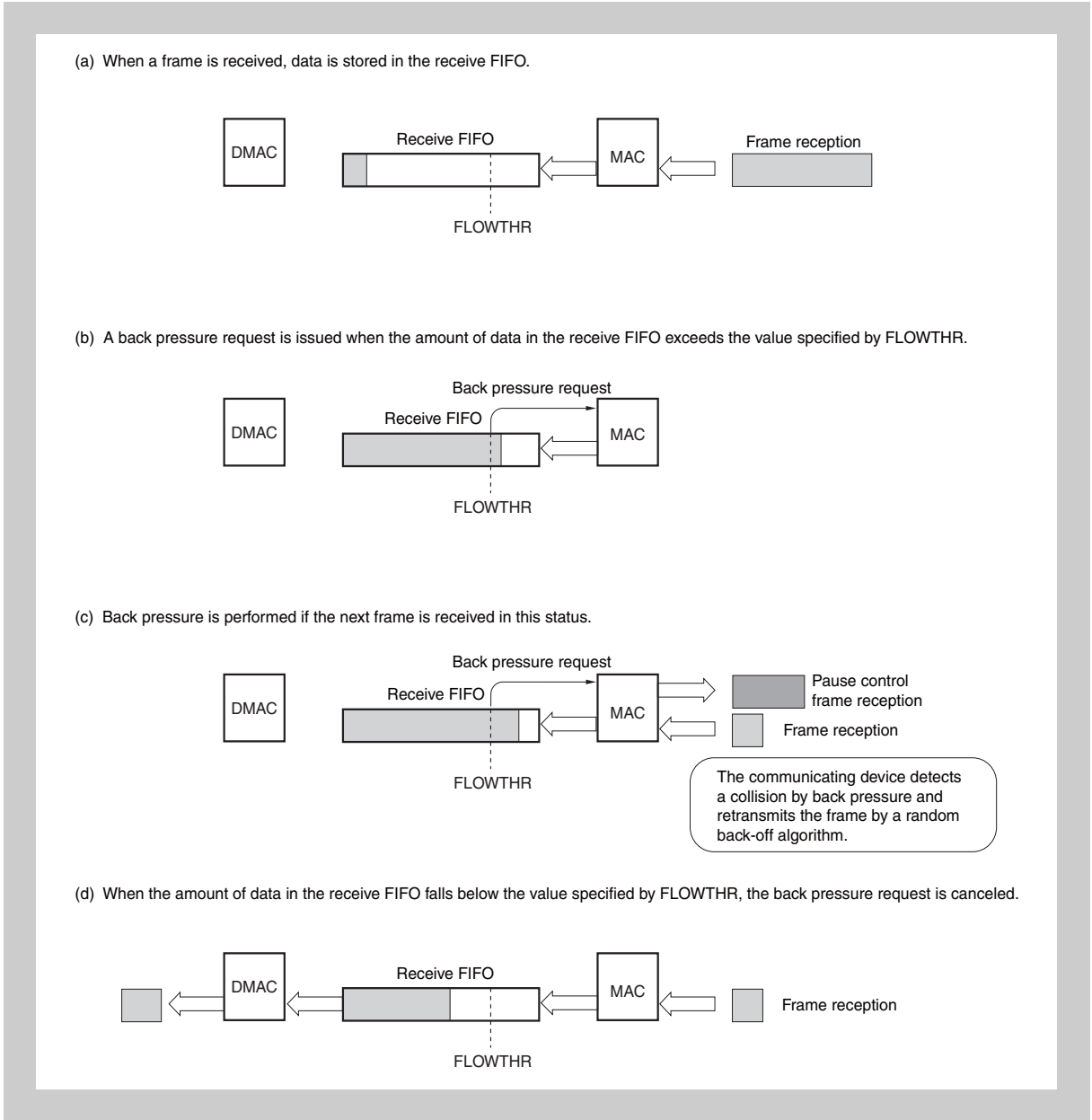


Figure 32-13 Back pressure control

(3) Operations related to VLAN frame

The Ethernet controller detects a VLAN frame by comparing the TPID field in a receive/transmit packet with the value of the VLAN type register (ETHA0VLTP). Operations related to the VLAN frame are described below.

(a) Detection of VLAN frame

The Ethernet controller constantly monitors the value of the 2-byte TPID field that follows the source address in a receive packet.

During transmission, the value of the TPID field is checked when the ETHA0MACC2.APD or ETHA0MACC2.VPD bit is 1.

The Ethernet controller recognizes a packet whose TPID field matches the value of the VLAN type register (ETHA0VLTP) as a VLAN frame.

(b) Reception of VLAN frame

If the value of the TPID field of a received packet matches the value of the VLAN type register (ETHA0VLTP), all judgments concerning the frame size are made based on the VLAN frame size (MAX: 1,522 bytes, MIN: 64 bytes).

(c) Transmission of VLAN frame

If a frame whose TPID field value matches the value of the VLAN type register (ETHA0VLTP) is transmitted from an upper layer when the ETHA0MACC2.APD bit is 1, the frame is recognized as a VLAN frame and is padded to extend its length to 68 bytes.

When the ETHA0MACC2.VPD bit is 1, all frames are recognized as VLAN frames and are padded with “0” to extend their lengths to 68 bytes.

32.5.5 DMAC

The dedicated DMAC for the Ethernet controller is a DMA function for the internal system bus of the Ethernet controller.

It is dedicated for the transmission/reception Ethernet controller.

All transmission and reception data is transferred by the dedicated DMAC for the Ethernet controller.

(1) DMA transfer mode

The following settings can be specified by using the ETHA0DMACM register.

Transfer mode

- Single transfer mode
- 4-beat incremental burst transfer mode
- 8-beat incremental burst transfer mode
- 16-beat incremental burst transfer mode

After the transfer mode has been specified by the ETHA0DMACM register, it will be applied starting with the next DMA transfer.

Cautions

1. Undefined length burst transfer mode cannot be specified using the register.
   Undefined length burst transfer is automatically used by the Ethernet controller to internally process fractional data during a DMA transfer.
   It is therefore not possible to transfer all transfer data in this mode intentionally.
2. The register that specifies the transfer mode is not locked during a DMA transfer. If the setting of this register is changed during a DMA transfer, therefore, the current DMA cycle becomes illegal. Do not change the setting of the register during a DMA transfer (when RXEN_STA or TXEN_STA is 1).

(2) Areas accessible by DMA transfer

The following two areas are subject to DMA transfers:

- H-bus shared memory
- External memory connected to the secondary memory controller

(3) DMA address boundary

With the DMA for the Ethernet controller, the address boundary does not have to be considered when setting the start address of the data buffer and the number of transfer bytes.

If there is fractional data during a burst transfer, the fraction is automatically processed before the data is transferred.

For reception, however, because it is not possible to predict where the data to be received will end, the last transfer might be a dummy transfer when a burst transfer is used.

Note In the 4-, 8-, or 16-beat incremental burst transfer mode, if the last data is shorter than the fixed length, it is automatically transferred in undefined length burst transfer mode.

Byte access for byte alignment is always executed in the single transfer mode.

(4) DMA arbitration

Because the Ethernet controller supports full-duplex transfer, DMA transmission and DMA reception might be executed together. If DMA requests for transmission and reception are issued at the same time, the reception request takes precedence.

32.5.6 Serial management interface

The Ethernet controller has a pair of serial management interfaces which can be used to set a PHY device, to read statuses, and for communicating with the PHY device when auto-negotiation is used.

Set the address of the PHY device to be connected by Ethernet controller to the ETHA0MADR register before using the serial management interface.

(1) Overview of serial management interface

(a) MDC clock

The management data clock (MDC) is generated by dividing the Ethernet control clock (f_{EC}).

The division ratio is specified by the ETHA0MIIC.CLKS bits.

Table 32-112 ETHA0MIIC register: CLKS bits and f_{EC} frequency

ETHA0MIIC.CLKS bits			Frequency range of f_{EC} input
Bit 4	Bit 3	Bit 2	
0	0	0	Setting prohibited
0	0	1	33 MHz or less
0	1	0	50 MHz or less
0	1	1	66 MHz or less
1	0	0	100 MHz or less
1	0	1	Setting prohibited
1	1	0	Setting prohibited
1	1	1	Setting prohibited

If the ETHA0MIIC.PHYSEL bit is 0 (the default value), the MDC is output only when a management frame is transmitted or received.

The MDC is always output when the ETHA0MIIC.PHYSEL bit is set.

If communication with the PHY device fails when the ETHA0MIIC.PHYSEL bit is 0, set this bit.

(b) Serial management frame structure

The Ethernet controller generates a serial management frame shown below by writing a value to the ETHA0MCMD or ETHA0MWTD register.

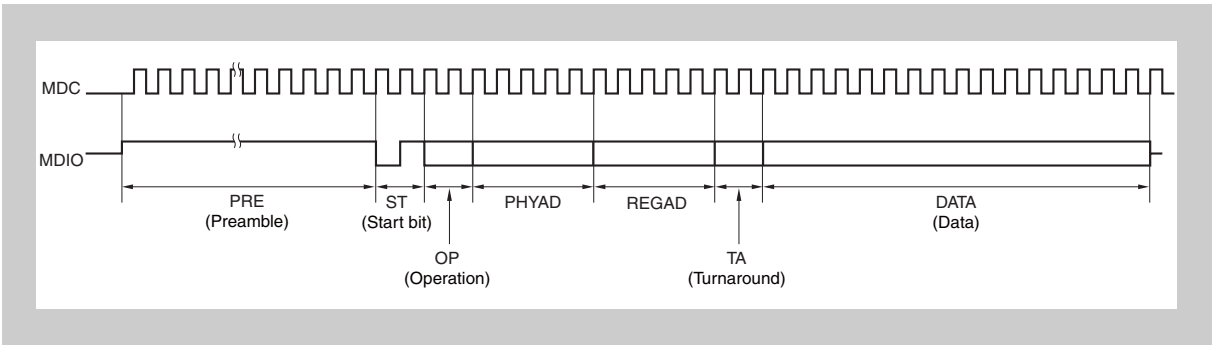
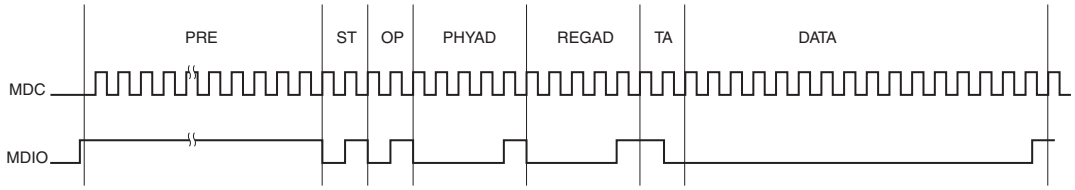


Figure 32-14 Serial management frame structure

A 32-bit preamble, 2-bit start bit field, and 2-bit opcode that indicates whether a register in the PHY device is read or written, are automatically appended to the serial management frame. PHYAD and REGAD indicate the address of the externally connected PHY device and the address of a register in that PHY device, respectively. The values set to the ETHA0MADR.FIAD and ETHA0MADR.RGAD bits are appended to PHYAD and REGAD, respectively.

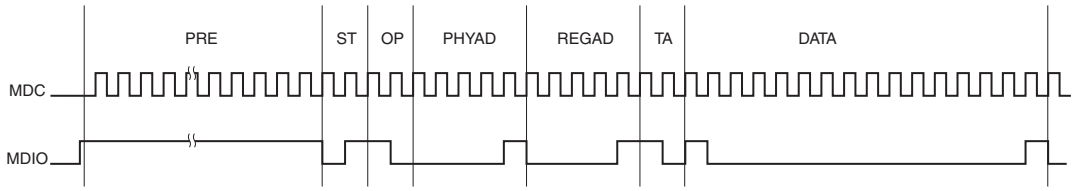
The Ethernet controller serially outputs data from the preamble to REGAD, and after a 2-bit turnaround, the data set to the ETHA0MWTD.CTLD bits is output for a write access. For a read access, serial data is input by the MDI signal and written to the ETHA0MRDD.PRSD bits.

While the MDO signal is being output, the MDOEN signal is asserted to 1.



Note Setting conditions: PHYAD = 01H, REGAD = 01H, CTLD = 0001H

Figure 32-15 Timing of MII management interface signals (write access)



Note Setting conditions: PHYAD = 01H, REGAD = 01H, CTLD = 8001H

Figure 32-16 Timing of MII management interface signals (read access)

(c) SCAN command

The Ethernet controller has a SCAN command to successively read a specific PHY register. When the ETHA0MCMD.SCANC bit is set, read accesses are generated successively. By reading the ETHA0MRDD.PRSD bit, the specific PHY register can be polled.

(2) Procedure for transmitting or receiving a serial management frame

Serial management frames are transmitted and received as follows.

First, the ETHA0MIND.SCANA bit is checked to see whether the SCAN command is under execution.

If not, the ETHA0MIND.BUSY bit is checked to see whether the serial management frame is being accessed. If the BUSY bit is 1, the Ethernet controller waits until it is cleared. If the SCAN command is being executed, the ETHA0MCMD.SCANC bit is cleared and then the Ethernet controller waits until the BUSY bit is cleared.

Next, the address of the external PHY device to which the frame is to be transmitted and the address of a register in the PHY device are set to the ETHA0MADR.FIAD and ETHA0MADR.RGAD bits, respectively.

Write access is started by writing to the ETHA0MWTD.CTLD bits.

The BUSY bit is set when data has been written to the ETHA0MWTD register and cleared when writing has finished.

Read access is started by writing 1 to the ETHA0MCMD.RSTAT bit. When the RSTAT bit is set, the BUSY bit is set. The BUSY bit is cleared after completion of reading. The host system can obtain the data of the PHY register by confirming that the BUSY bit is 0 and then reading the ETHA0MRDD.PRSD bits.

To execute the SCAN command, set the ETHA0MCMD.SCANC bit. While this bit is 1, reading is repeatedly executed. The ETHA0MIND.SCANA bit holds 1 while the SCAN command is executed. The ETHA0MIND.NVALID bit holds 1 until the first read access finishes after the SCAN command is executed. The ETHA0MIND.BUSY bit is set when the SCAN command is executed. If the SCAN command is disabled (by clearing the ETHA0MCMD.SCANC bit), the ETHA0MIND.BUSY bit is cleared after the current read access finishes.

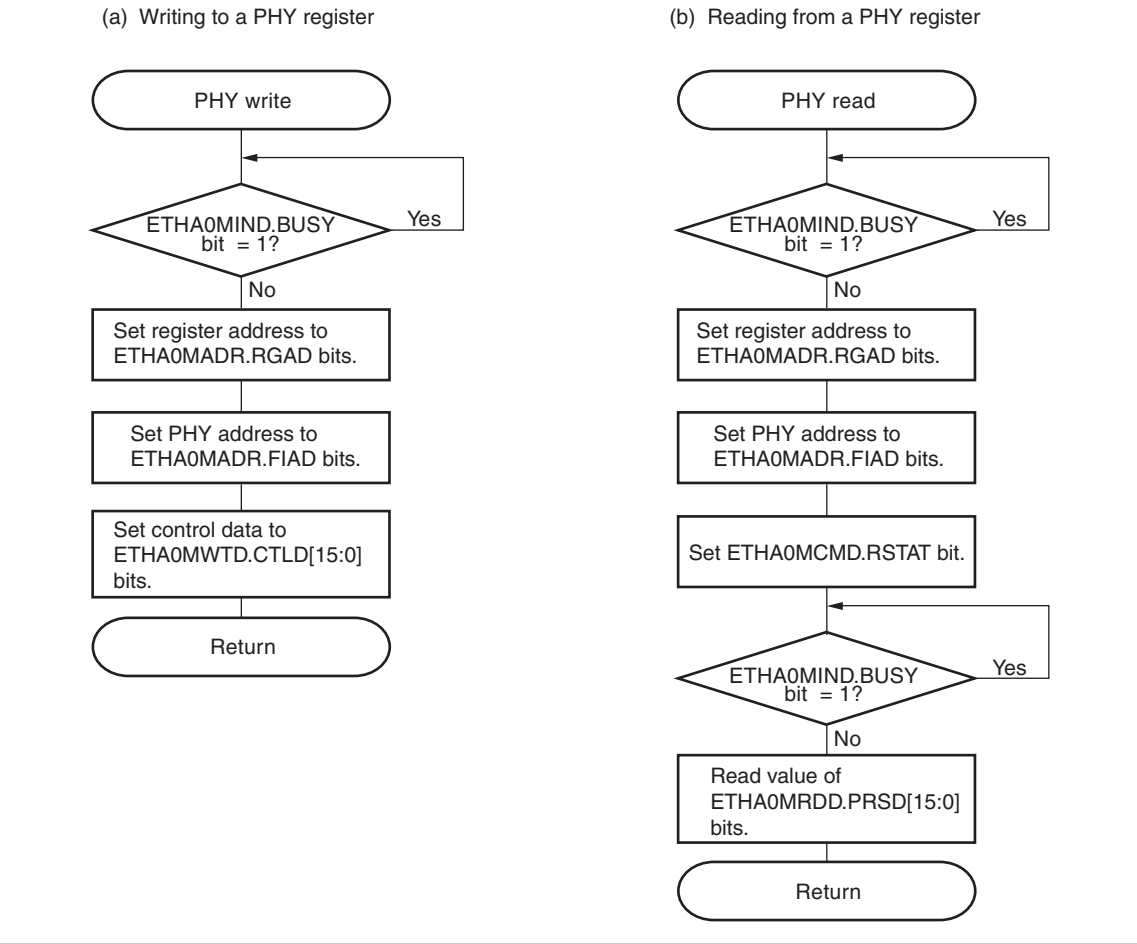


Figure 32-17 Accessing PHY register

32.5.7 Address filtering

(1) Overview of address filtering

The Ethernet controller filters addresses by using the destination address of a received packet and, based on the filtering result, decides whether to accept or discard the received packet.

The filtering conditions can be specified by the ETHA0AFR register. Conditions can be individually specified for unicast addresses, multicast addresses, and broadcast addresses, or conditions can be combined.

(a) Filtering of unicast addresses

The address set to the ETHA0LSA1 and ETHA0LSA2 registers is compared with the destination address of a received packet as a unicast address. A received packet is accepted if its destination address matches the address specified for these registers, and is discarded if not. Each receive packet is checked to see if its destination address matches the set unicast address.

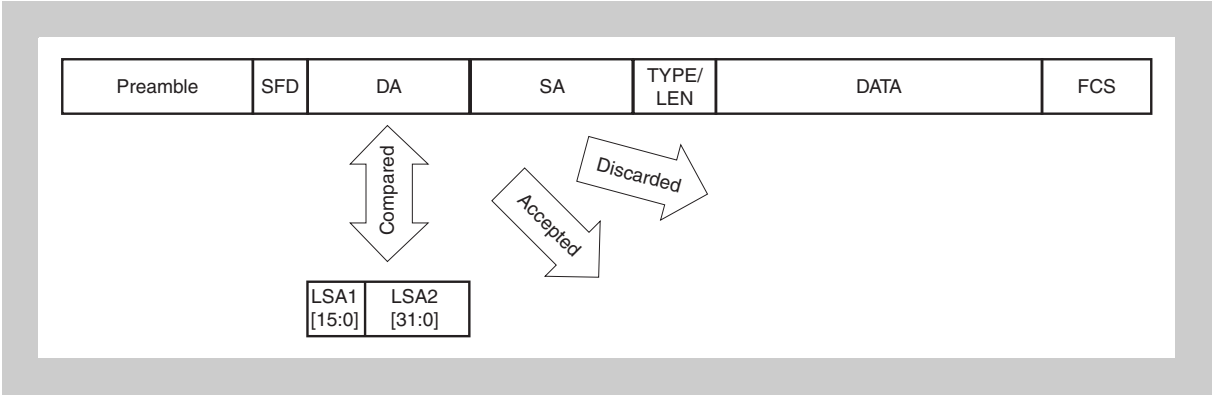


Figure 32-18 Image of filtering by unicast address during reception

(b) Filtering of multicast addresses

A multicast address is filtered in two ways. If the ETHA0AFR.PRM bit is set, all received packets that have a multicast address as the DA are accepted.

If the ETHA0AFR.AMC bit is set, only a received packet that has a multicast address that matches the hash table set up by the ETHA0HT1 and ETHA0HT2 registers is accepted, and a received packet whose multicast address does not match is discarded.

The hash table is used as follows for multicast address filtering.

The hash table is referenced by using bits [28:32] of the 32 bits of the CRC calculation result of the received multicast address. The following polynomial is used for calculating the CRC.

$$CRC(x) = X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^{8} + X^{7} + X^{5} + X^{4} + X^{2} + X + 1$$

If 1 is specified at the bit position indicated by the value resulting from decoding the above 6 bits in the ETHA0HT1 and ETHA0HT2 registers (shown in *Table 32-113 "Correspondence between HT bits and hash values (ETHA0HT1 and ETHA0HT2)"*), a receive packet that has that multicast address will be accepted. To set the hash table, it is necessary to execute a CRC calculation on a multicast address defined in advance, and set the corresponding bits.

Table 32-113 Correspondence between HT bits and hash values (ETHA0HT1 and ETHA0HT2)

	CRC[25:23]							
CRC[28:26]	**111_B**	**110_B**	**101_B**	**100_B**	**011_B**	**010_B**	**001_B**	**000_B**
111_B	HT1[31]	HT1[30]	HT1[29]	HT1[28]	HT1[27]	HT1[26]	HT1[25]	HT1[24]
110_B	HT1[23]	HT1[22]	HT1[21]	HT1[20]	HT1[19]	HT1[18]	HT1[17]	HT1[16]
101_B	HT1[15]	HT1[14]	HT1[13]	HT1[12]	HT1[11]	HT1[10]	HT1[9]	HT1[8]
100_B	HT1[7]	HT1[6]	HT1[5]	HT1[4]	HT1[3]	HT1[2]	HT1[1]	HT1[0]
011_B	HT2[31]	HT2[30]	HT2[29]	HT2[28]	HT2[27]	HT2[26]	HT2[25]	HT2[24]
010_B	HT2[23]	HT2[22]	HT2[21]	HT2[20]	HT2[19]	HT2[18]	HT2[17]	HT2[16]
001_B	HT2[15]	HT2[14]	HT2[13]	HT2[12]	HT2[11]	HT2[10]	HT2[9]	HT2[8]
000_B	HT2[7]	HT2[6]	HT2[5]	HT2[4]	HT2[3]	HT2[2]	HT2[1]	HT2[0]

An example of a program that executes hash table calculations is shown below.

If DA = 123456789ABC, for example, CRC = D4E88056, CRC[28:26] = 5, and CRC[25:23] = 1. If HT1[9] in *Table 32-113 "Correspondence between HT bits and hash values (ETHA0HT1 and ETHA0HT2)"* is set, a multicast packet with the target DA is received. If the value of both the ETHA0HT1 and ETHA0HT2 registers is 00000000H, all packets are discarded.

```
#include <stdio.h>

// Calculate CRC(Little endian)
unsigned long crc32( unsigned char *data, int size )
{
 const unsigned long poly = 0x04C11DB7ul; // MSB=X^31+X^30+X^29...
 unsigned long crc = 0xffffffful;
 int i;
 unsigned char c;

 while( size-- != 0 ) {
   c = *data++;
   for ( i=0; i<8; i++, c >>= 1) {
     crc = (crc<<1)^( (((crc&0x80000000ul)?1:0)^(c&1))? poly : 0ul );
   }
  }
  return crc;
}

//=============================================================================
// Hash table value calculated by performing a CRC32 operation on the MAC address of ETHER-SS
//=============================================================================
int main( void )
{

  unsigned char temp[6] = { 0x12,0x34,0x56,0x78,0x9A,0xBC };
  unsigned long crc;
  unsigned int lp;

    printf("------------------------------------------------\n");
    printf(" MAC=%02X:%02X:%02X:%02X:%02X:%02X\n",
                temp[0],temp[1],temp[2],temp[3],temp[4],temp[5] );
    printf("------------------------------------------------\n");

    // CRC calculation method 1 for the MAC address
    crc = crc32( temp, sizeof(temp) );
    temp[5] = (char)(crc >> 26)&0x07;
    temp[4] = (char)(crc >> 23)&0x07;
    for( lp = 0; lp < 4; lp++, crc>>=8 ) temp[lp] = (char)(crc & 0xff);
    printf(" CRC=%02X, %02X, %02X, %02X \n", temp[3], temp[2], temp[1], temp[0] );
    if( temp[5] >4 ) printf(" HT1=%08X, HT2=00000000\n", (unsigned long)(1<<temp[4])<<(8*(temp[5]&0x03)) );
        else printf(" HT1=00000000, HT2=%08X\n", (unsigned long)(1<<temp[4])<<(8*(temp[5]&0x03)) );
    printf("------------------------------------------------\n");
  return 0;
}
```

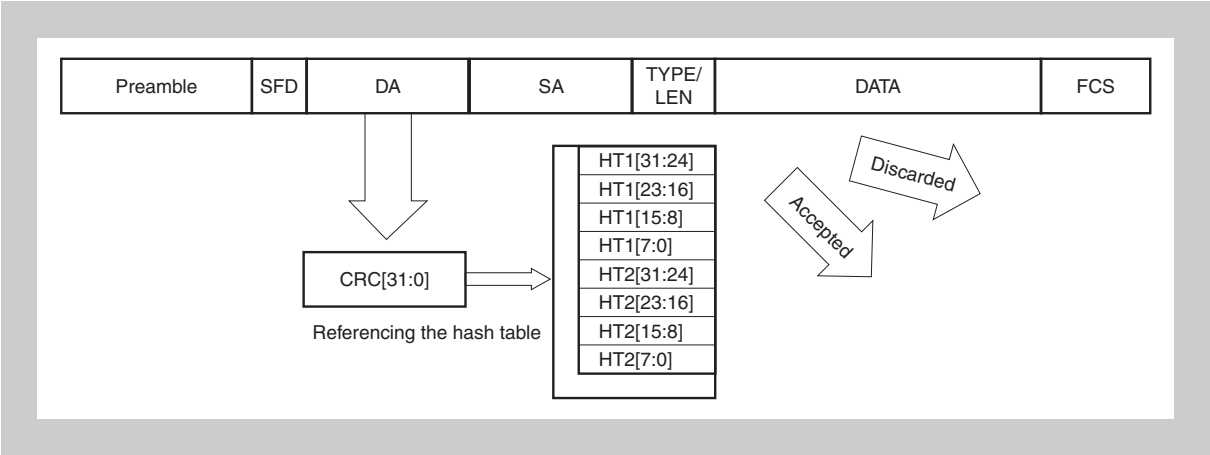


Figure 32-19 Image of filtering by referencing hash table

(c) Filtering of broadcast addresses

When the ETHA0AFR.ABC bit is set, a packet that has a broadcast address is received.

(d) Promiscuous mode

When the ETHA0AFR.PRO bit is set, the promiscuous mode is set and all packets are received.

If none of reception conditions (a) to (d) above is satisfied, the received packet is discarded.

For the combinations of the above conditions, refer to *Table 32-114 "Settings of ETHA0AFR register and packets to be filtered"*.

Table 32-114 Settings of ETHA0AFR register and packets to be filtered

Setting of ETHA0AFR register				Receive packet				
PRO	PRM	AMC	ABC	LSA mismatch, unicast	LSA match, unicast	HT mismatch, multicast	HT match, multicast	Broadcast packet
1	–	–	–	Accepted	Accepted	Accepted	Accepted	Accepted
0	1	–	–	Discarded				
0	0	1	1			Discarded		
0	0	1	0					Discarded[a]
0	0	0	1				Discarded	Accepted
0	0	0	0					Discarded

a) The broadcast packet can be received if the corresponding bit in the hash table is set because the broadcast address is included in a multicast address.

Note –: Any

(2) Setting address filtering conditions

Set address filtering as follows.

First, clear the ETHA0MACC1.SRXEN bit. When the SRXEN bit is 0, the receive data interface is disabled. Next, set a station address to the ETHA0LSA1 and ETHA0LSA2 registers. Set a combination of the necessary filtering conditions to the ETHA0AFR register. To conditionally receive multicast packets, a hash table must be set up by using the ETHA0HT1 and ETHA0HT2 registers. After making the above settings, enable packet reception by setting the ETHA0MACC1.SRXEN bit.

32.5.8 Statistics counters

The Ethernet controller has 39 statistics counters to check the communication quality and other line statuses.

Each time communication of one frame has been finished (or aborted), the communication status is checked and the corresponding statistics counter is updated. The statistics counters cannot be stopped. To avoid using a statistics counter, set the corresponding bit in the ETHA0CAM1 or ETHA0CAM2 register to mask the interrupt from that counter.

The statistics counters can be read at any time even during communication.

If a counter overflows, the corresponding bit in the ETHA0CAR1 or ETHA0CAR2 register is set, and, if the relevant interrupt is not masked by the ETHA0CAM1 and ETHA0CAM2 registers, an Ethernet MAC interrupt is generated. The ETHA0CAM1 and ETHA0CAM2 registers can be used to specify whether to mask the overflow interrupt of each counter.

To clear a statistics counter, write 0 to it. At this time, the current communication does not have to be stopped. If updating a statistics counter and writing data to it occur at the same time and conflict, updating takes precedence, and the counter is written after it has been updated.

Note that the statistics counters cannot be stopped. To avoid using a statistics counter, mask the counter by setting the corresponding bit in carry mask register 1 or 2 (ETHA0CAM1 and ETHA0CAM2) to prevent the INTETMOV interrupt from being generated.

The statistics counters can be read/written in 32-bit units.

Cautions

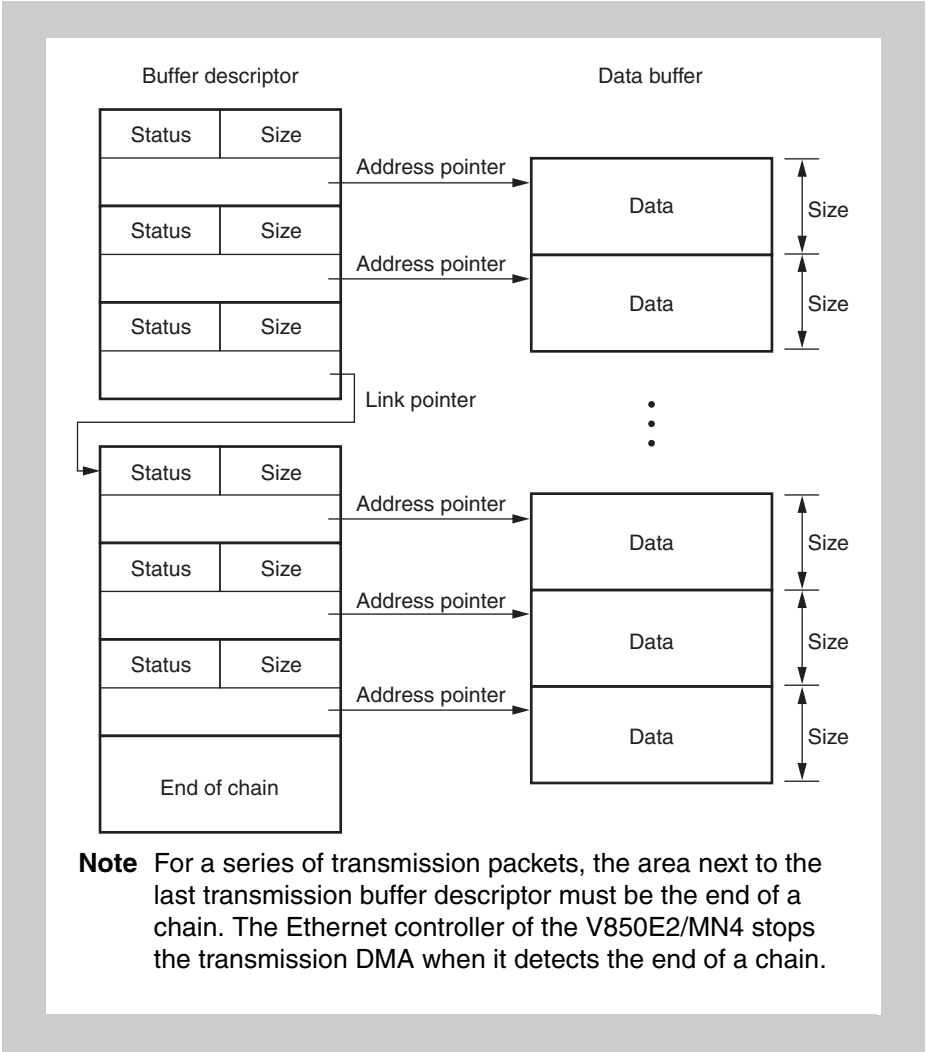
1. The Ethernet controller updates the statistics counters based on the Ethernet controller clock (f_{EC}). If the Ethernet controller clock (f_{EC}) is considerably slower than the communication clock (TXCLK/RXCLK), the counters might miscount statistics information. If the statistics information is miscounted, a status vector overrun occurs, the C2DV bit in carry register 2 (ETHA0CAR2) is set, and an INTETMOV interrupt is generated.
2. Carry registers 1 and 2 (ETHA0CAR1 and ETHA0CAR2) are cleared when they are read.

Note Aside from the statistics counters, a transmission abort counter (ETHA0TXABTCNT) and a reception abort counter (ETHA0RXABTCNT) are available for counting the number of times transmission/reception has been aborted.

32.6 Data Transfer

32.6.1 Buffer structure

The Ethernet controller buffer of the V850E2/MN4 consists of a buffer descriptor and data buffer.



Note For a series of transmission packets, the area next to the last transmission buffer descriptor must be the end of a chain. The Ethernet controller of the V850E2/MN4 stops the transmission DMA when it detects the end of a chain.

Figure 32-20 Ethernet controller buffer structure (transmission)

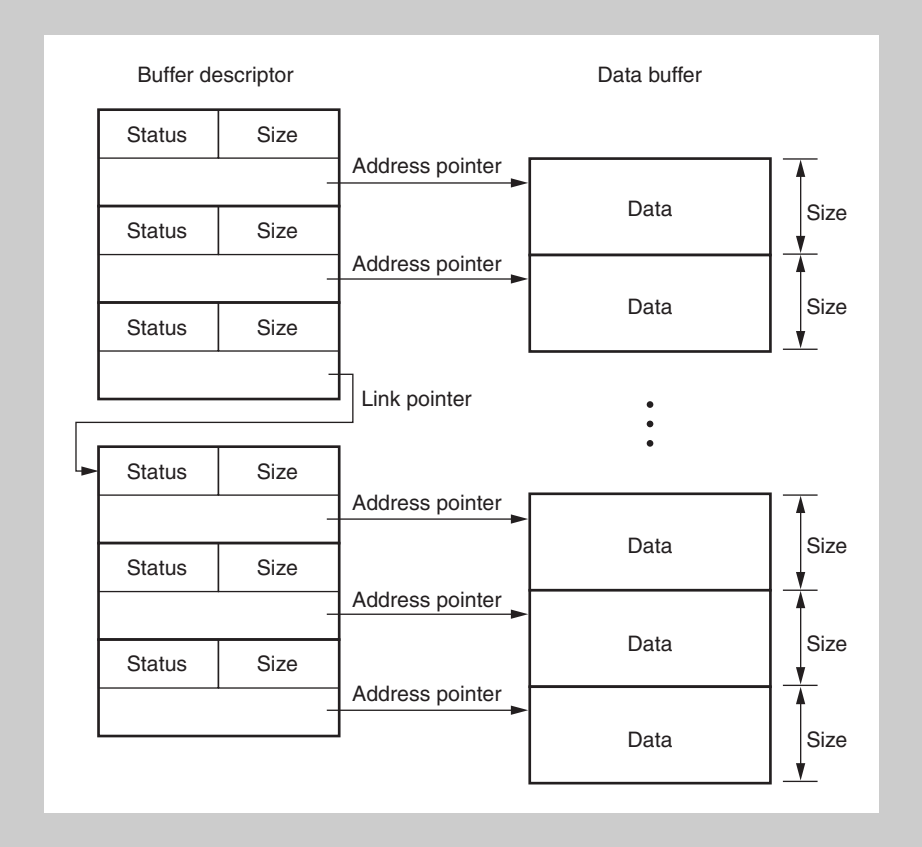


Figure 32-21 Ethernet controller buffer structure (reception)

32.6.2 Descriptor mechanism

The Ethernet controller supports a descriptor mechanism to support a situation where the memory space that stores transmit data and receive data is not contiguous.

The Ethernet controller uses the following three types of descriptors.

- Buffer descriptor
- Link pointer
- End of chain

Each descriptor consists of two word-aligned data (64 bits).

The Ethernet controller can consecutively process multiple descriptors in one DMA transfer (refer to *(6) "Descriptor chain" on page 2315 in 32.6.2 "Descriptor mechanism"*).

A reception DMA transfer or transmission DMA transfer is started by setting the first address of a receive descriptor chain to ETHA0RXDP or the first address of a transmit descriptor chain to ETHA0TXDP, and then setting the RXS or TXS bit of the ETHA0MODE register.

A descriptor chain must end with the descriptor of an end of chain.

(1) Format of buffer descriptor

A buffer descriptor is configured of 2 words (64 bits). The lower word consists of control bits. The higher word indicates the start address of the data buffer indicated by the descriptor.

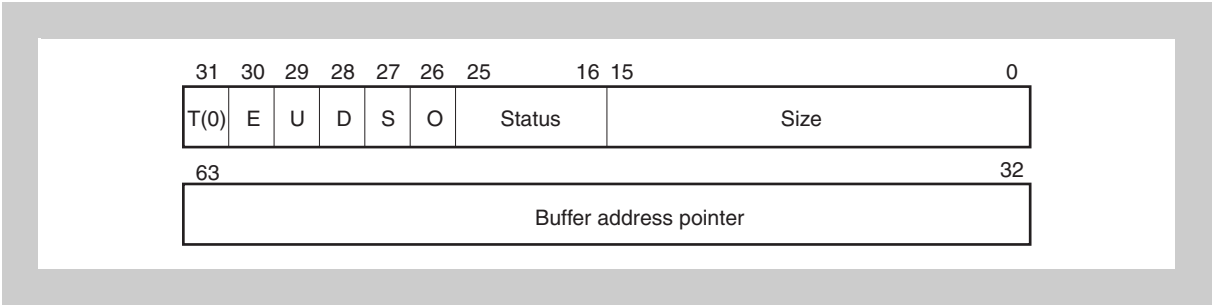


Figure 32-22 Buffer descriptor format

Table 32-115 Bit configuration of buffer descriptor (1/2)

Bit position	Bit name	Function
63 to 32	BAP	These bits specify an address pointer that indicates the start address of the data buffer. Byte alignment can be specified for BAP.
31	T	Descriptor type This bit indicates the type of the descriptor. Clear this bit for a buffer descriptor.
30	E	Last buffer flag This is a control bit that indicates the end of packet data. 0: Normal buffer data (not last data) 1: Last buffer data of the current packet If this bit is set for transmission, a TXI interrupt is generated when the data of the corresponding data buffer has been transferred, and then the next descriptor processing starts. Clear this bit for reception. When the last data of a frame has been written, this bit is set when the data is written back. Next, an RXI interrupt is generated, and then the next descriptor processing starts.
29	U	Used bit This bit indicates whether a DMA transfer has finished (including transfer in progress). 0: Transfer not finished (including transfer in progress) 1: Transfer finished The CPU clears this bit when it creates or obtains buffer data (a descriptor). When a DMA transfer to the buffer area indicated by this descriptor has finished, the Ethernet controller sets this bit. The Ethernet controller issues the TECI or RECI interrupt and stops DMA if it reads a descriptor whose U bit is set. If a bus error or overflow error occurs during reception, the U bit of either the first descriptor in the packet or the descriptor where the overflow occurred is set.
28	D	This bit indicates an access error in the data buffer. 0: No error 1: Access error in the data buffer The CPU clears this bit when it creates or obtains buffer data (a descriptor). If an access error occurs, the Ethernet controller sets control bit D of the first descriptor that indicates the current packet, and control bit D of the descriptor that was responsible for the access error.
27	S[a]	This bit indicates that reception status information has been written to the Status field (only control bit S in the first descriptor of a received packet is valid). 0: Status information is not included. 1: Status information of the received packet is included. The CPU clears this bit when it creates or obtains buffer data (a descriptor). When a received packet is transferred via DMA, the Ethernet controller writes a valid value to the Status field of the first descriptor of the current packet and sets control bit S each time one packet has been transferred.

Table 32-115 Bit configuration of buffer descriptor (2/2)

<table>
<tr><th>Bit position</th><th>Bit name</th><th>Function</th></tr>
<tr><td>26</td><td>O[a]</td><td>This bit indicates occurrence of an overflow error during reception.
0: No overflow
1: Overflow
The CPU clears this bit when it creates or obtains buffer data (a descriptor). If an overflow error occurs during reception, the Ethernet controller writes back 1 to the control bit O of the first descriptor of the packet, and sets control bit E of the descriptor in which the overflow error occurred. No interrupt is generated.</td></tr>
<tr><td>25:16</td><td>Status[a]</td><td>These bits form a Status field that indicates status information during reception. If control bit S is 1, the value of the Status field is valid. The CPU clears this bit when it creates or obtains buffer data (a descriptor). During a DMA transfer of a receive packet, the Ethernet controller writes a valid value to the Status field of the first descriptor of the current packet and sets control bit S each time transfer of one packet has finished.
<table>
<tr><th>Bit position</th><th colspan="2">Bit name</th></tr>
<tr><td>16</td><td colspan="2">CEPS</td></tr>
<tr><td>17</td><td colspan="2">RCV</td></tr>
<tr><td>18</td><td colspan="2">RCRCE</td></tr>
<tr><td>19</td><td colspan="2">RLOR</td></tr>
<tr><td>20</td><td colspan="2">DBNB</td></tr>
<tr><td>21</td><td colspan="2">RXOK</td></tr>
<tr><td>23:25</td><td>FYTP[0:2]</td><td>000: RBRO
001: RMUL
010: USOP
011: VLAN
100: RPCF
101: RCFR
110: “Normal”
111: “Reserved”</td></tr>
</table>
</td></tr>
<tr><td>15:0</td><td>Size</td><td>These bits form a Size field that indicates the size (in bytes) of the buffer data indicated by this descriptor. During a DMA transfer of a receive packet, the Ethernet controller writes the length of one transferred packet to the Size field of the last descriptor of the current packet each time transfer of one packet has finished.</td></tr>
</table>

a) This bit is not used during transmission. Clear this bit.

Note The Size field is 16 bits. Setting 0 to this field is prohibited. If 0 is set, an error interrupt is generated.
If FFFFH is set to this field, transfer of 64K – 1 bytes is executed.

(2) Format of link pointer

A link pointer consists of 2 words. The lower word consists of control bits. The higher word indicates the address of the next descriptor.

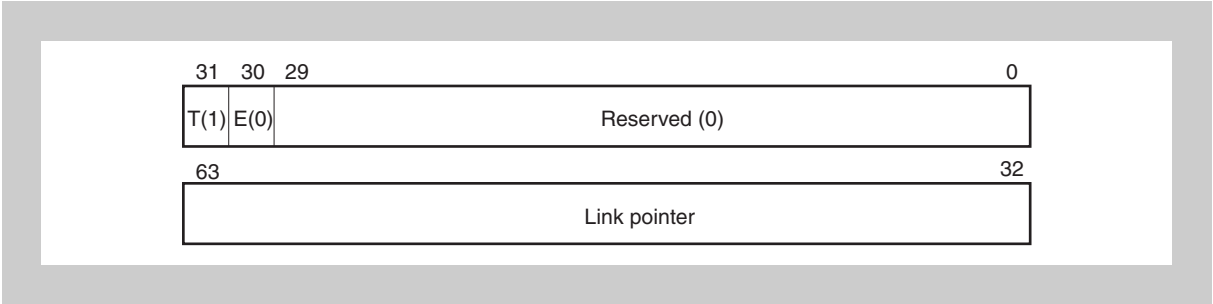


Figure 32-23 Link pointer format

Table 32-116 Bit configuration of link pointer

Bit position	Bit name	Function
63:32	Link Pointer	These bits indicate the address of the next descriptor. The lower 2 bits are ignored (word aligned).
31	T	Set this bit for a link pointer.
30	E	Clear this bit for a link pointer.
29:0	Reserved	These bits are reserved. Clear these bits.

(3) Format of end of chain

An end of chain consists of 2 words. The lower word consists of control bits. The higher word indicates 0.

When the Ethernet controller detects an end of chain, it finishes the DMA transfer and generates an RECI or TECI interrupt.

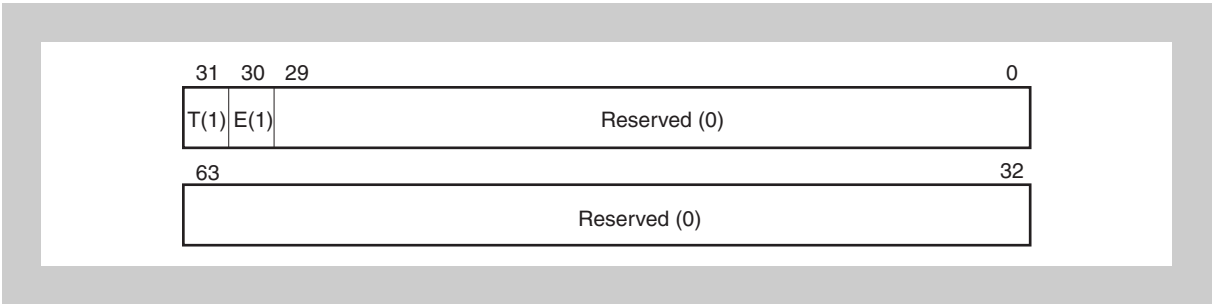


Figure 32-24 End of chain format

Table 32-117 Bit configuration of end of chain

Bit position	Bit name	Function
63:32	BAP	These bits are set to null (all zero) for an end of chain.
31	T	Set this bit for an end of chain.
30	E	Set this bit for an end of chain.
29:0	Reserved	These bits are reserved. Clear these bits.

(4) Writing back the status

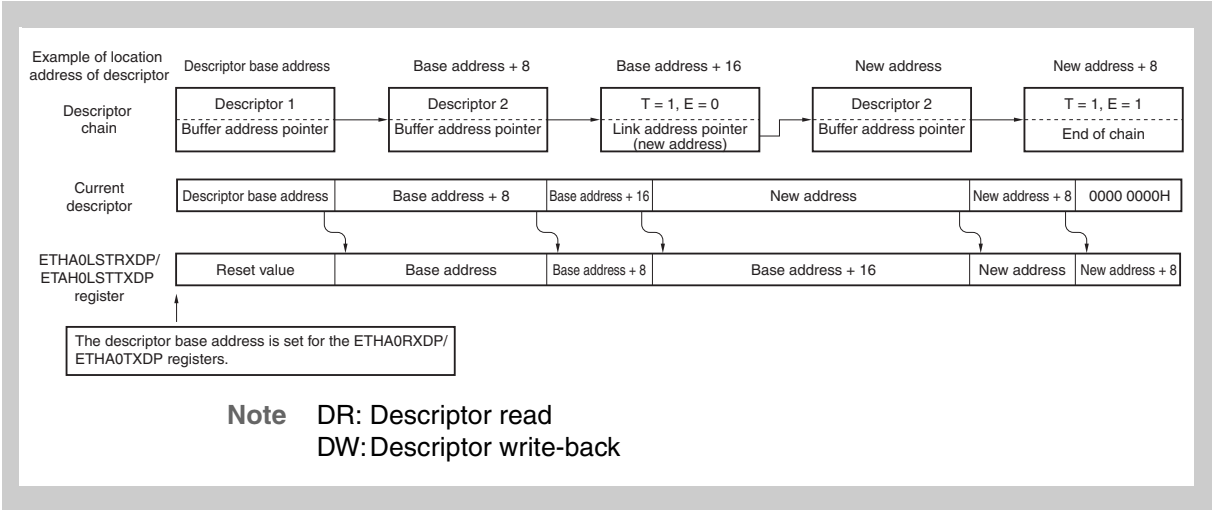
When DMA reception is executed, the reception status is written back to the first descriptor of the packet, and the length of the packet transferred via DMA is written back to the last descriptor. The status is written back as described in the *Status* field in *Table 32-115 "Bit configuration of buffer descriptor"*.

(5) Reporting last descriptor

The current descriptor can be reported. Two registers, ETHA0LSTRXDP and ETHA0LSTTXDP, hold the address of the descriptors processed by the Ethernet controller. The address of the descriptor that was processed immediately before can be obtained by reading these two registers via software.

The timing to save the address of a descriptor to ETHA0LSTRXDP or ETHA0LSTTXDP is shown in *Figure 32-25 "Timing to copy last descriptor"* below.

When the link pointer is read, the address of the next descriptor can be read from the BAP bits. The address of the link pointer is then copied to ETHA0LSTRXDP or ETHA0LSTTXDP.



Note DR: Descriptor read
DW: Descriptor write-back

Figure 32-25 Timing to copy last descriptor

If a descriptor chain is configured in a ring buffer, the descriptor can be updated by reading ETHA0LSTRXDP or ETHA0LSTTXDP, using the TXI flag of the INTSCTX interrupt (the RXI flag of the INTSCRX interrupt) as a trigger.

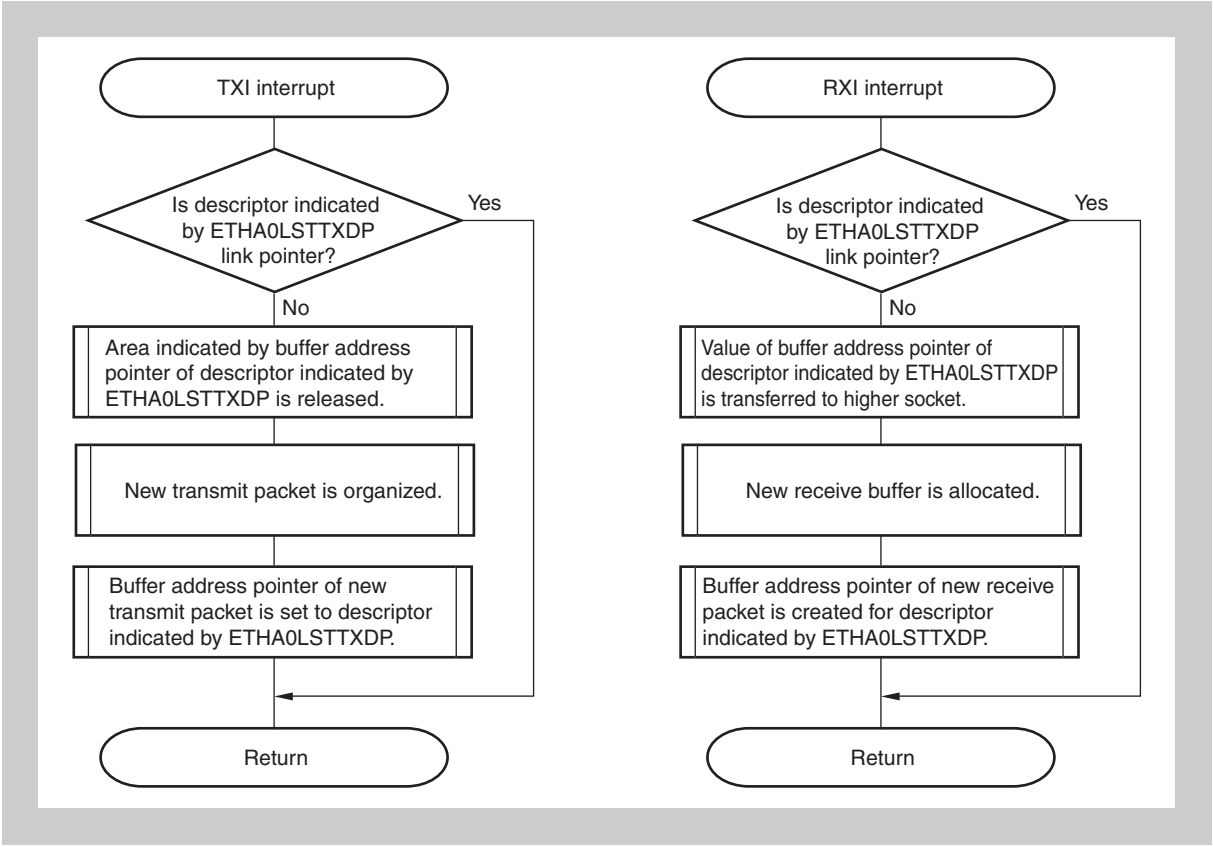


Figure 32-26 Updating descriptor chain by using ETHA0LSTRXDP or ETHA0LSTTXDP

(6) Descriptor chain

A descriptor uses a chain structure to indicate a data buffer (of an undefined length).

An image is as shown below.

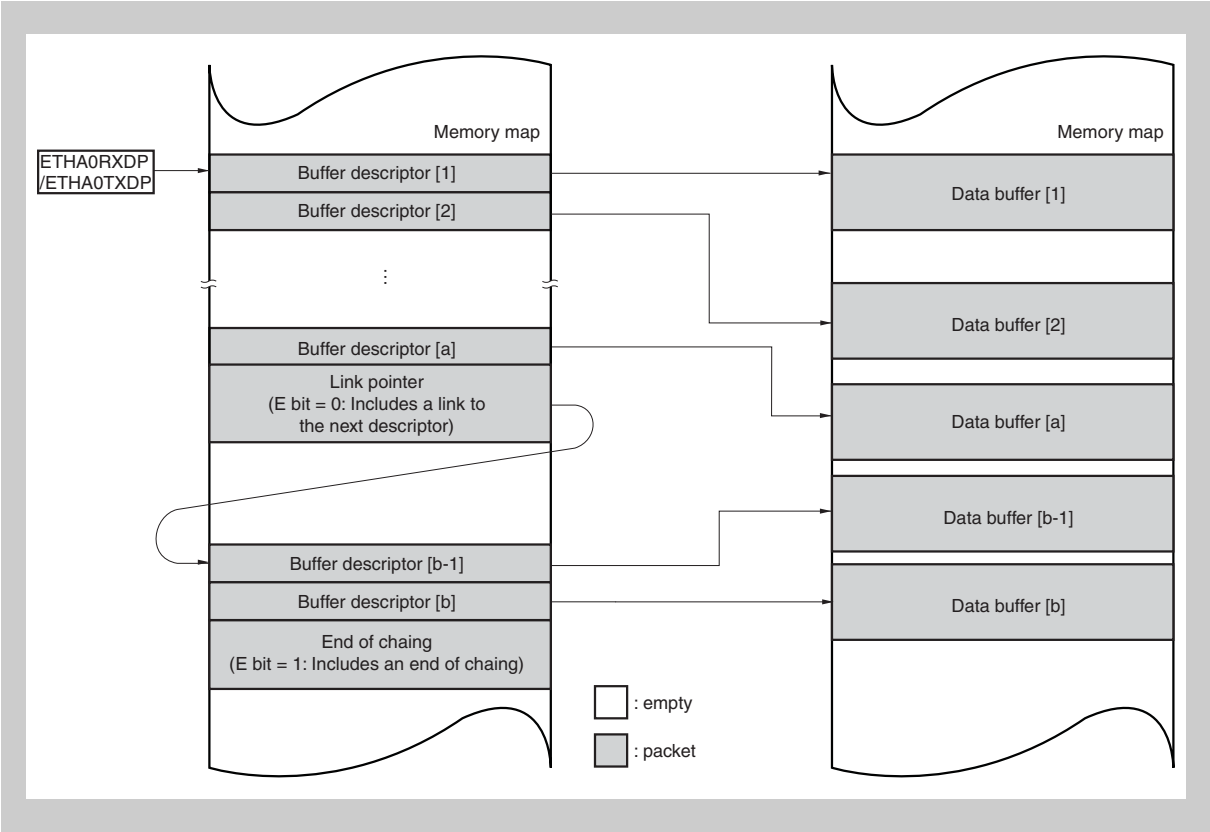


Figure 32-27 Overview of descriptor chain using ETHA0LSTRXDP or ETHA0LSTTXDP

Allocate a descriptor to a successive memory area. When a link pointer is used, a descriptor can also be allocated to a non-successive memory area by specifying the location address of the next buffer descriptor for the link pointer. The descriptor chain can be finished by specifying the end of a chain.

A ring descriptor chain may be configured by using the last link pointer to specify the memory address of the first buffer descriptor.

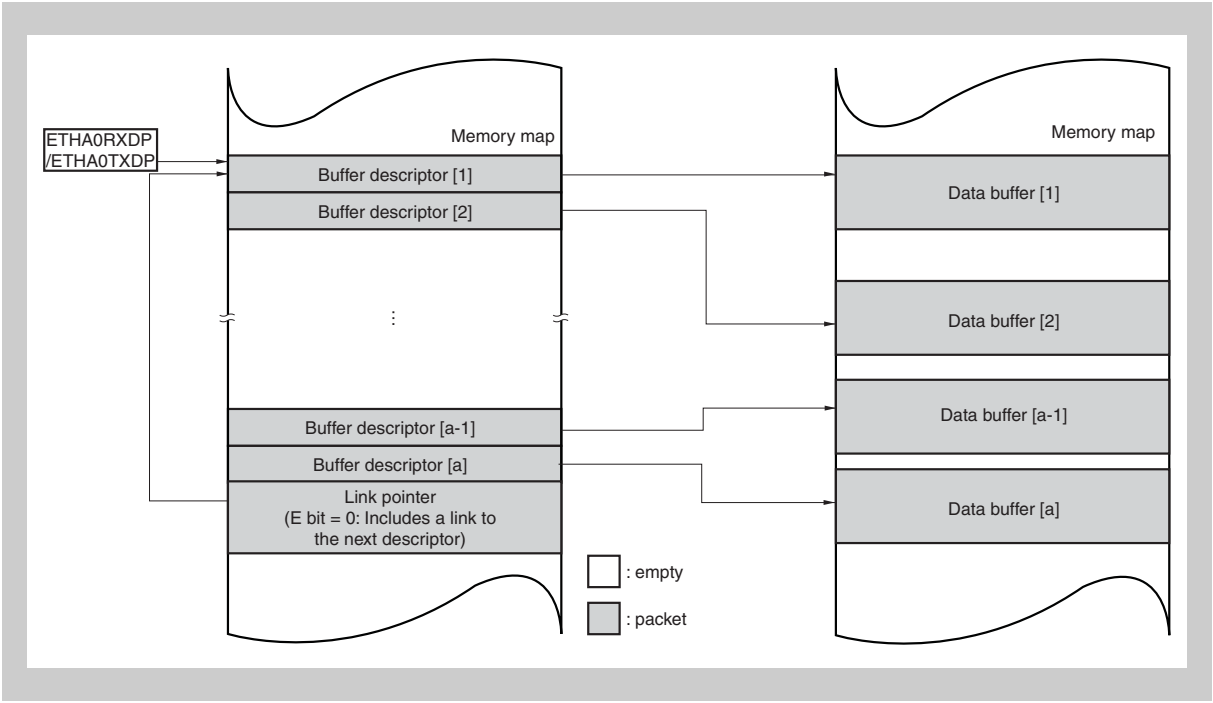


Figure 32-28 Overview of ring buffer formed by descriptor chain

A ring buffer is configured if the beginning of a descriptor chain is specified by a link pointer. In a ring buffer, if a descriptor whose U bit is set is read, the Ethernet controller generates a RECI or TECI interrupt in the same manner as when detecting the end of a chain, and then stops DMA.

Caution Handling of U bit

If the U bit of a transmit descriptor is set, it indicates that the processing for that descriptor has finished, so the CPU can specify a new descriptor by clearing the U bit.

For a receive descriptor that has the U bit set to 1, however, its content might be updated later due to status write back or error occurrence, so a new descriptor cannot be specified unless the completion of packet reception is confirmed. But if the E bit is set, it indicates that packet reception has finished, so the descriptor chain can be specified as a new descriptor.

(7) Byte alignment and word boundary

Although a descriptor must be word-aligned, the data buffer can be set at a byte-aligned address.

The Ethernet controller automatically identifies an address, executes a single transfer up to a word boundary and an undefined length word transfer up to the burst boundary, and then executes a burst transfer.

32.6.3 Frame transmission

When the CPU prepares the transmission descriptor and data for the transfer target of the DMAC for the Ethernet controller, the transmission descriptor register (ETHA0TXDP) is set up, and then the TXS bit of the ETHA0MODE register is set, the dedicated DMAC fetches the transmission buffer descriptor from the address specified for the descriptor register, reads the transmission data from the data buffer, and then transfers it to the transmission FIFO.

The data transferred to the FIFO buffer is then output to PHY in synchronization with TXCLK, in the order of the preamble, SFD, and then frame data.

If the CRCEN bit of the ETHA0MACC1 register is set, FCS is added to the end of the data.

If the PADEN bit of the ETHA0MACC1 register is set, short frames are automatically padded with 0s when they are transmitted.

If the current descriptor does not contain the end of the frame, the next descriptor is read and data is read from the data buffer indicated by the read descriptor.

When the transmission finishes, the transmission status is written to the last descriptor. Then, the next transmission buffer descriptor is fetched, and the transmission resumes in the same manner when the next data can be transmitted.

Each time DMA processing for a buffer descriptor finishes, an interrupt that indicates the end of transmission DMA (TXI) is generated.

If the next transmission buffer descriptor is the end of a chain descriptor, an interrupt that indicates this (TECI) is generated and transmission DMA stops.

To resume transmission DMA, set up the ETHA0TXDP register and buffer descriptors again.

The frame transmission procedure is described below.

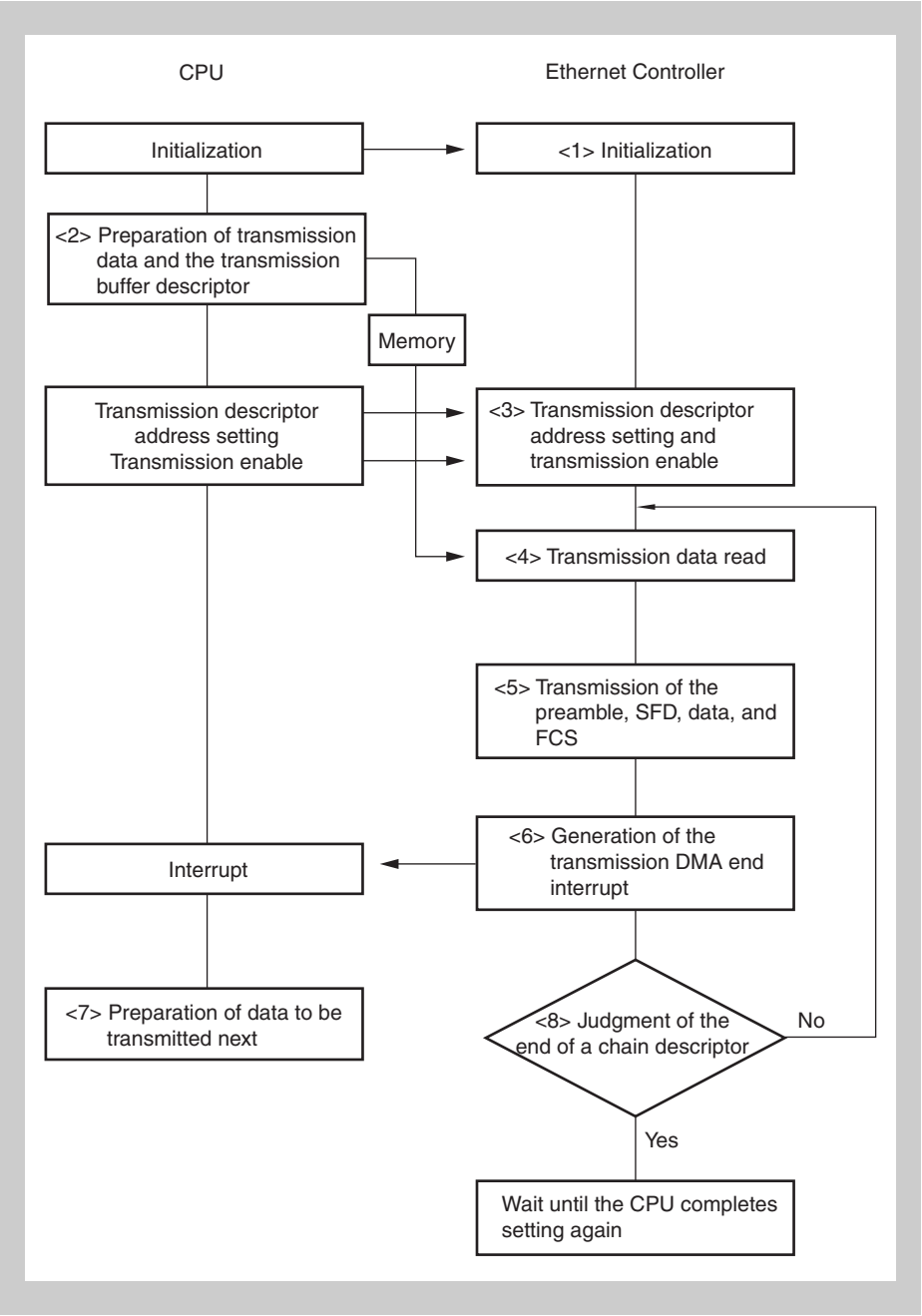


Figure 32-29 Frame transmission procedure example

1. Initialize the Ethernet controller.
   For details about the initialization procedure, see *32.3 “Initialization” on page 2156*.
2. Create transmission data and the transmission buffer descriptor in the data RAM.

   When creating the descriptor, specify values for the E bit (set to 1 if the last packet data needs to be kept), T bit (cleared to 0), U bit (cleared to 0), and Size bits of the transmission buffer descriptor.
3. Set up the transmission buffer descriptor register and enable transmission.

   Specify the transmission descriptor address for the ETHA0TXDP register, and set the TXS bit of the ETHA0MODE register to enable transmission.
4. Read the transmission data.

   The transmission data is read from memory by way of DMA.
   The next descriptor is read if the E bit of the transmission buffer descriptor is 0.
5. Transmit a packet.

   The preamble, SFD, data, and FCS are transmitted.
6. Report the end of DMA transmission.

   A TXI interrupt request is generated to report the end of transmission to the CPU.
7. Prepare the next data.

   The CPU checks the transmission status and prepares the data to be transmitted next.
8. Judge the end of a chain descriptor.

   A TECI interrupt request is generated to report to the CPU that the end of the chain descriptor has been reached.

The frame transmission operation is described below, using a specific descriptor chain as an example.

When the TXS bit of the core function setting register ETHA0MODE is set by software, the first descriptor is read from the address ($0028\ 0000_H$) specified by the transmission descriptor pointer ETHA0TXDP, and transmission descriptor analysis starts. Specify the buffer address pointer ($0028\ 1000_H$) as the DMA transfer start address, and then transfer the data in the buffer to the transmission FIFO.

Because the E bit of the transmission descriptor is 0, which indicates that it is not the last data, the next buffer descriptor ($0028\ 0008_H$) is read, the buffer address pointer ($0028\ 1800_H$) is specified, and then the data in the buffer is transferred to the transmission FIFO.

If the data in the buffer indicated by the buffer descriptor whose E bit is 1 has been transferred completely, the U bit is set and the transmission processing finishes. At this time, an interrupt request TXI is generated.

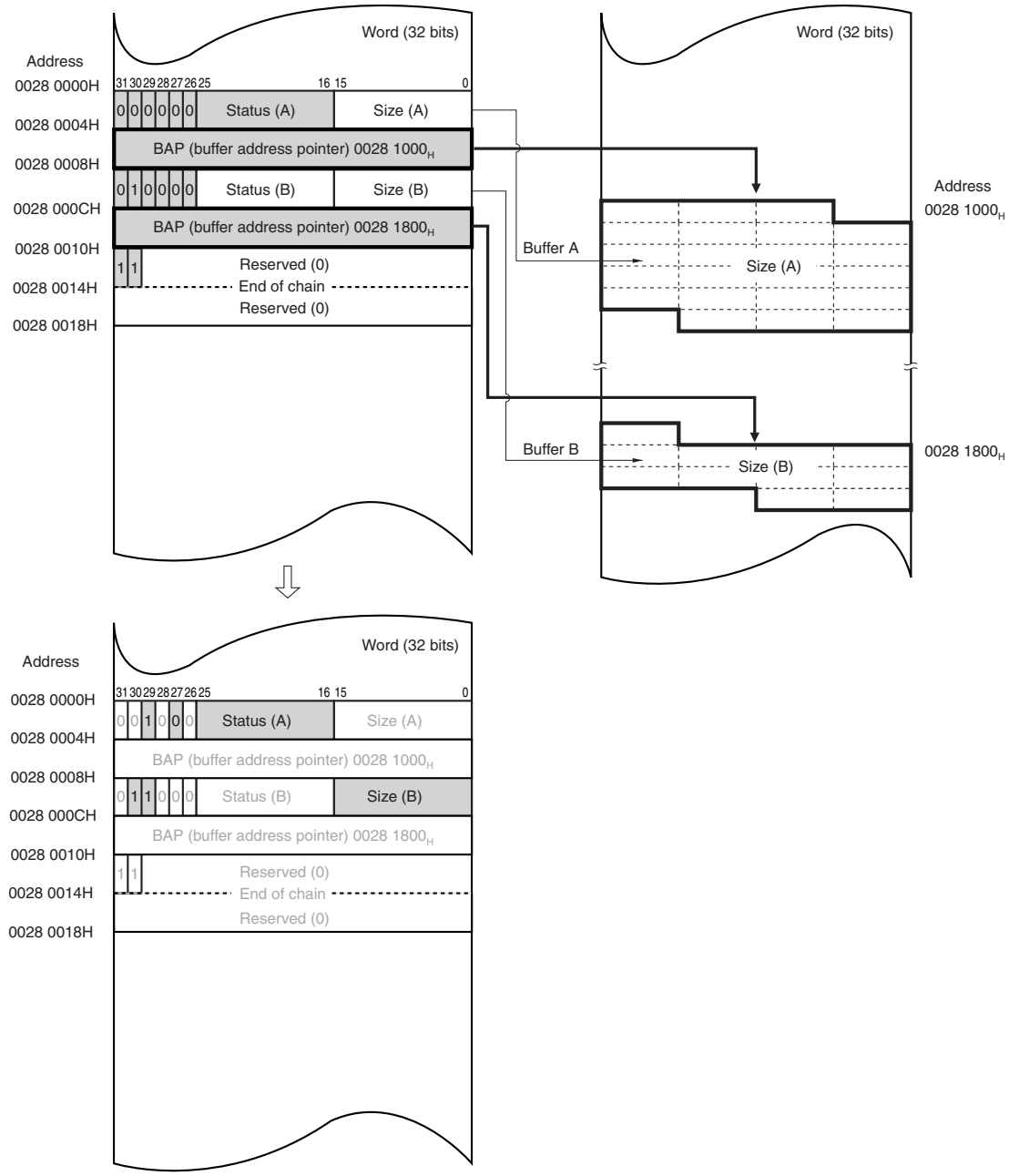


Figure 32-30 Descriptor chain configuration example (packet transmission)

32.6.4 Frame reception

When the of SRXEN bit (reception enable) of the MAC configuration register (ETHA0MACC1) and the RXS bit (reception DMA enable) of the ETHA0MODE register are set and the descriptor pointer register ETHA0RXDP is set up, reception frame processing starts immediately when the MAC receives data.

When data is received, the preamble and frame start delimiter (SFD) are checked to determine whether they are valid.

If they are valid, processing of the received frame starts.

If either is invalid, the frame is ignored.

If a frame conflict occurs or a frame is corrupted due to address filtering, the data is not written to the reception buffer.

When the frame is normally received and not corrupted by address filtering, it is transferred to the data buffer specified by the reception buffer descriptor.

During reception, the Ethernet controller checks whether the frame length is correct.

At the end of the frame, the FCS is checked and written to the buffer descriptor.

A frame of 64 bytes or shorter (a short packet) is DMA transferred.

When the frame has been received completely, the E and U bits of the last descriptor are set and the number of data bytes transferred to the Size field is written back.

When the all packet data has been received, the U and S bits of the first descriptor are set and the reception status is written back to the Status field.

At this time, an interrupt request RXI is generated.

The frame reception procedure is described below.

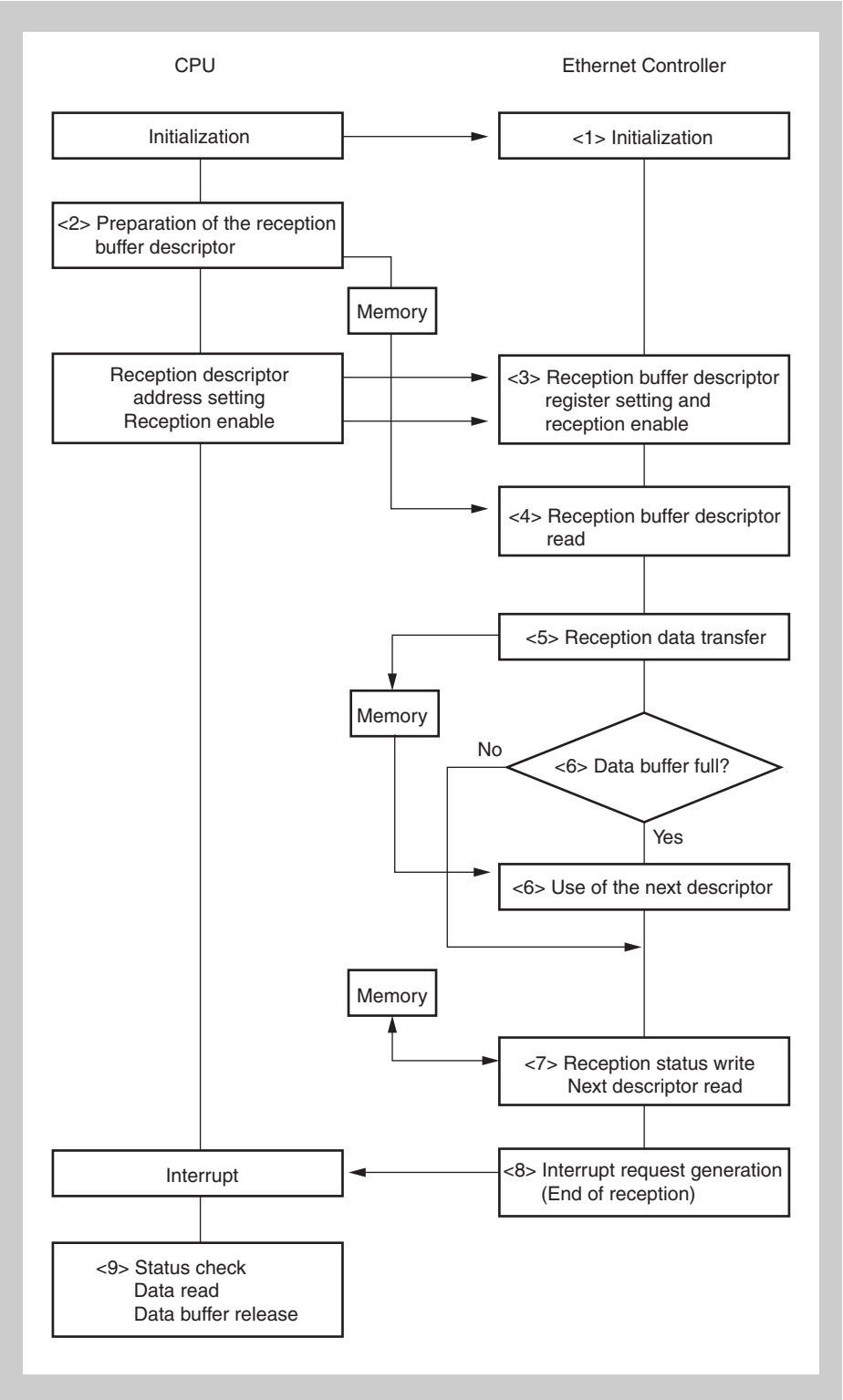


Figure 32-31 Frame reception procedure example

1. Initialize theEthernet controller.
   For details about the initialization procedure, see *32.3 “Initialization” on page 2156*.
2. Creating a reception buffer descriptor in memory.

   When creating the descriptor, specify values for the T bit (cleared to 0), U bit (cleared to 0), and Size bits (data buffer size) of the reception buffer descriptor.
3. Set up the reception buffer descriptor register and enable reception.

   Specify the reception descriptor address for the ETHA0RXDP register, and set the RXS bit of the ETHA0MODE register to enable reception.
4. Read the reception buffer descriptor.

   The reception buffer descriptor is read by way of DMA.
5. Transfer reception data.

   The data is transferred to the data RAM by way of DMA.
6. Judge whether the reception data buffer is full.

   The next descriptor is read if the current data buffer is full.
7. Receive a packet.

   Reading of the buffer descriptor and data transfers are repeatedly performed. When the end of the frame is received, 1 is written to the E bit of the last reception buffer descriptor and the number of bytes of transferred data is written to the Size field.
8. Report the end of reception.

   An RXI interrupt request is generated to report the end of reception to the CPU (when the interrupt is not masked).
9. Prepare the next data.

   The CPU checks the reception status, releases the data buffer, and then prepares the next data.

The frame reception operation is described below, using a specific descriptor chain as an example.

When the RXS bit of the core function setting register ETHA0MODE is set by software, the first descriptor is read from the address (0028 0000_H) specified by the reception descriptor pointer ETHA0RXDP, and reception descriptor analysis starts.

Specify the first buffer address pointer (0028 1000_H) as the DMA transfer start address, and then transfer the reception data in the FIFO buffer to buffer A.

When buffer A becomes full during the subsequent reception, the next descriptor (0028 0008_H) is read, the buffer address pointer (0028 1800_H) is specified as the DMA transfer start address, and then the reception data in the FIFO buffer is transferred to buffer C.

The E and U bits of the last descriptor are set and the number of bytes of data transferred to the Size field is written back.

When all packet data has been transferred, the U and S bits of the first descriptor are set and the reception status information is written back to the Status (A) field.

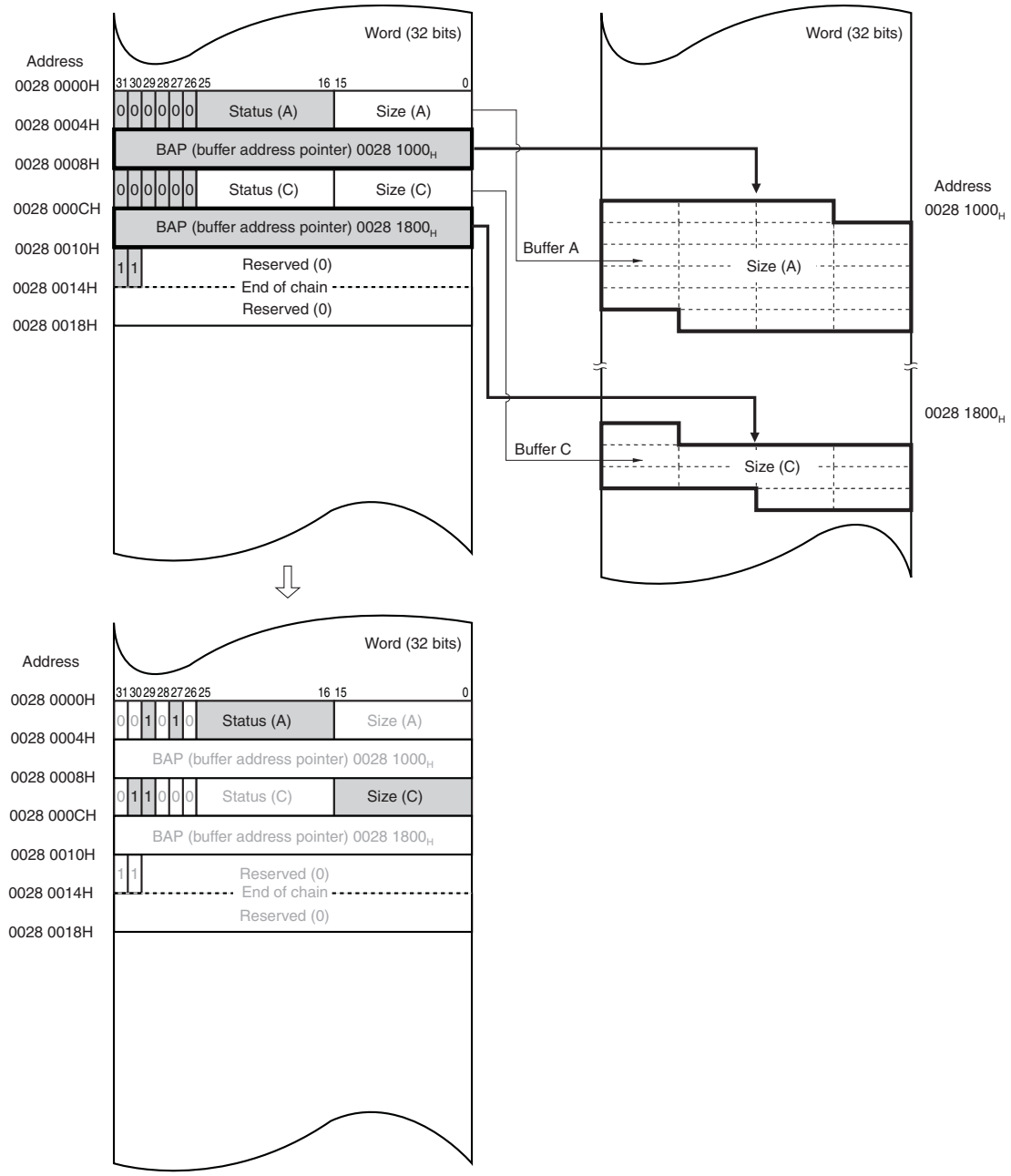


Figure 32-32 Descriptor chain configuration example (packet transmission)

32.6.5 Error processing

(1) Error write back

If a bus error occurs when accessing the data buffer during reception or transmission, an error interrupt is asserted and DMA is stopped. In addition, the U and D bits of the first descriptor in the packet are set (the U bit might already be set), and the U, D, and E bits of the descriptor where the error occurred are set.

If an overflow error occurs during reception, the U and O bits of the first descriptor in the packet are set (the U bit might already be set), and the U and E bits of the descriptor where the error occurred are set.

(2) Error interrupt

An error interrupt is not asserted only by a data buffer access error but also by a descriptor access error. Whether an error interrupt is generated can be checked using the ETHA0INTMS.RBEI and ETHA0INTMS.TBEI bits.

If a data buffer access error or descriptor access error has occurred, the descriptor chain that includes the descriptor where the error occurred must be set up again.

During transmission:

If a data buffer access error or descriptor access error has occurred, the TBEI bit is set and the DMA is stopped.
Transmission will not be performed until the TXS bit is set again.

During reception:

If a data buffer access error or descriptor access error has occurred, the RBEI bit is set and the DMA is stopped.
Reception will not be performed until the RXS bit is set again.
At this time, the packet being transferred from the FIFO buffer is canceled.
Packet cancellation is not executed if no packet is being transferred.
If a bus error occurred during write back due to a reception overflow, the processing is performed in the same way as when a descriptor access error occurs.

32.7 Receive Checksum

The Ethernet controller has a receive checksum unit that calculates the receive checksum.

Receive checksum calculation is enabled and disabled by setting the RXCHKSMEN bit of the ETHA0TRANSCTL register.

A checksum is appended to the end of receive data. Allocate enough memory for the amount of received data, which is the packet length plus 2 bytes.

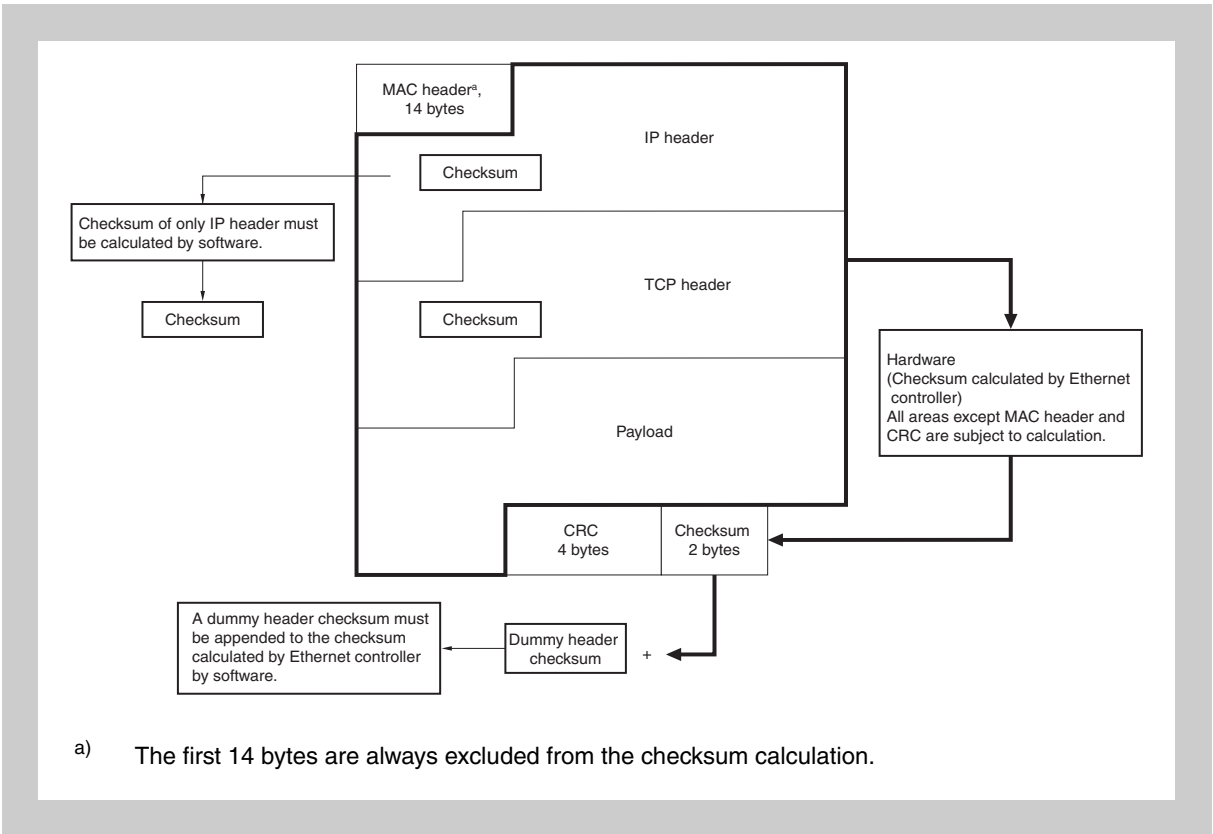
When checksum calculation is enabled, all parts (payload) of a receive frame, except the MAC header (first 14 bytes) and CRC (last 4 bytes), are subject to checksum calculation. If the number of bytes subject to calculation is odd, 00H is added to the last byte for checksum calculation.

The minimum receive packet length subject to checksum calculation is 19 bytes (payload = 1 byte). If the receive packet length is 18 bytes (payload = 0 bytes) or less, 0 is output as the checksum. However, the length information increases by 2 bytes.

Before changing the value of the RXCHKSMEN bit, confirm that transfer of the receive frame is stopped.

32.7.1 Processing by software

The first 14 bytes of a packet are treated as the MAC header and are always excluded from checksum calculations. Therefore, calculation starts from the 15th byte. If the MAC header exceeds 14 bytes, such as in a VLAN and huge frame, correction by software is necessary.



a) The first 14 bytes are always excluded from the checksum calculation.

Figure 32-33 Checksum calculation

The minimum packet length is 64 bytes. When padding a short packet, many systems use 00_H, but some systems use a different code for padding. In this case, the checksum calculation is executed including the padding data, resulting in a mismatch between the calculation result and the checksum in the header. In this case, the checksum of the padding data must be corrected by software.

Caution According to RFC1071, the correct checksum comparison result can be obtained if the checksum result is compared using the same endian format (either big endian or little endian).

32.8 Transmit Checksum

The Ethernet controller has a transmit checksum function. This function has a dedicated DMAC (transmit checksum DMAC), which accesses the internal system bus independently of the internal Ethernet controller DMAC.

The algorithm for a transmit checksum complies with RFC1071, a payload is added in a 32-bit width, and a 2-byte checksum is output to the end of the data.

The minimum transmit payload subject to checksum calculations is 1 byte, and the maximum is 1,500 bytes as defined in IEEE802.3.

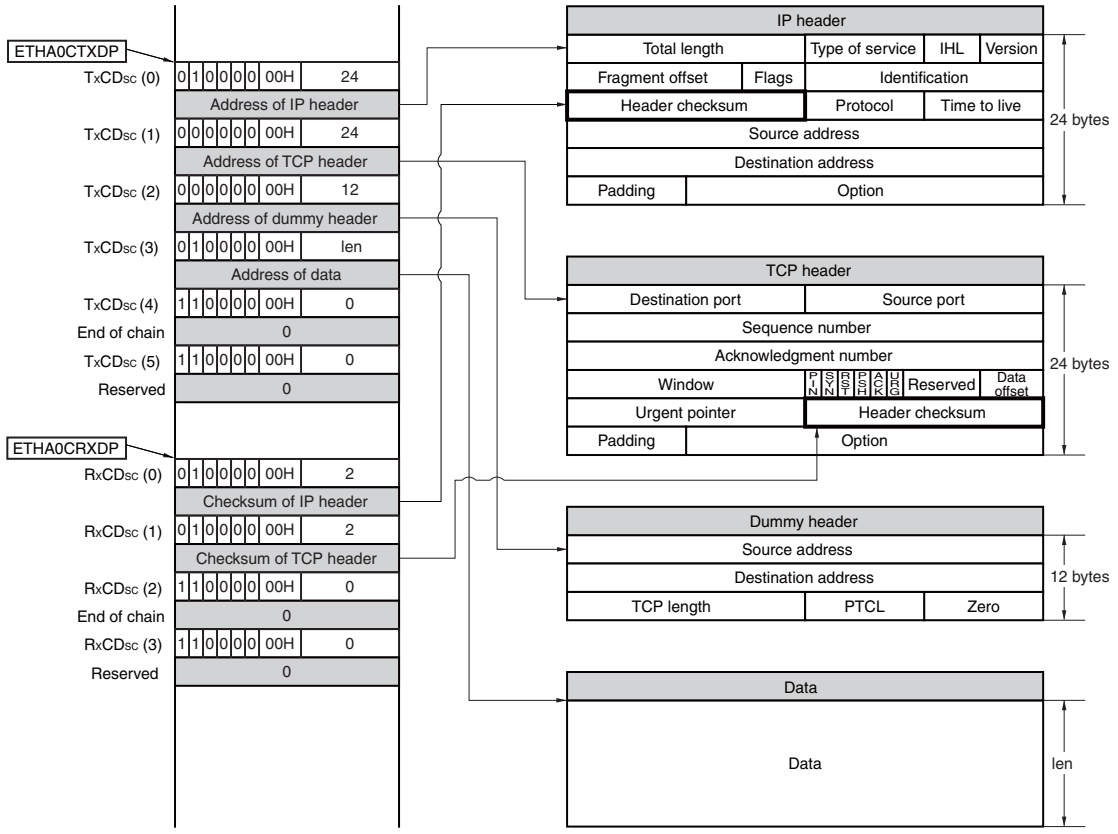
A checksum is calculated and the result is formatted in little endian format.

32.8.1 Configuration of transmit checksum descriptor

Calculation of a transmit checksum is controlled by a descriptor.

Depending on the configuration of the descriptor, the IP header and TCP header can be calculated separately or all at once.

The descriptor configuration for calculating the IP header and TCP header all at once is shown below.



The address to which the checksum of the IP header is to be written is obtained by adding 10 bytes to the IP header address, and the address to which the checksum of the TCP header is to be written is obtained by adding 16 bytes to the TCP header address.

Figure 32-34 Configuration of transmit checksum descriptor

32.9 Cautions

32.9.1 Cautions on FIFO

Note the following restrictions on the internal FIFO buffers of the Ethernet controller.

Table 32-118 Restrictions on transmit FIFO

Maximum FIFO capacity	DMA transfer condition	Retry/abort	Condition to transmit data to PHY	Feature	Note
2,044 bytes or less	Data in transmit FIFO is less than 1,536 bytes[a].	Data is automatically retransmitted/ aborted when a collision is detected.	At least one packet is in the transmit FIFO.	Data can be retransmitted without underrun.	The transmit packet length must be 1,536 bytes or less.

a) Multiple packets may be stored in the transmit FIFO as long as its capacity is not exceeded. However, if the data in the transmit FIFO reaches 1,536 bytes, DMA transmission stops, preventing the transmit FIFO from overflowing. However, because the Ethernet controller starts transmission after one packet of data has been stored in the transmit FIFO, the transmit FIFO is locked if the length of one packet is more than 1,536 bytes. Be sure to observe the rated size of one packet to use (1,518 bytes or less for non-VLAN frames or 1,522 bytes or less for VLAN frames).

Table 32-119 Restrictions on receive FIFO

Maximum FIFO capacity	DMA transfer condition	Condition to transmit pause control frame[a]	Feature	Note
2,036 bytes or less	At least one packet is in the receive FIFO.	• Transmission of pause control frame: Data in the receive FIFO is greater than the size specified by the FLOWTHR bits. • Transmission of 0 pause control frame: Data in the receive FIFO is less than the size specified by the ZPTHR bits.	All error packets can be discarded.	The receive packet length must be 2,036 bytes or less.

a) Control by a pause control frame does not completely prevent the receive FIFO from overflowing. If the receive FIFO overflows, receive packets that are not accepted are discarded.

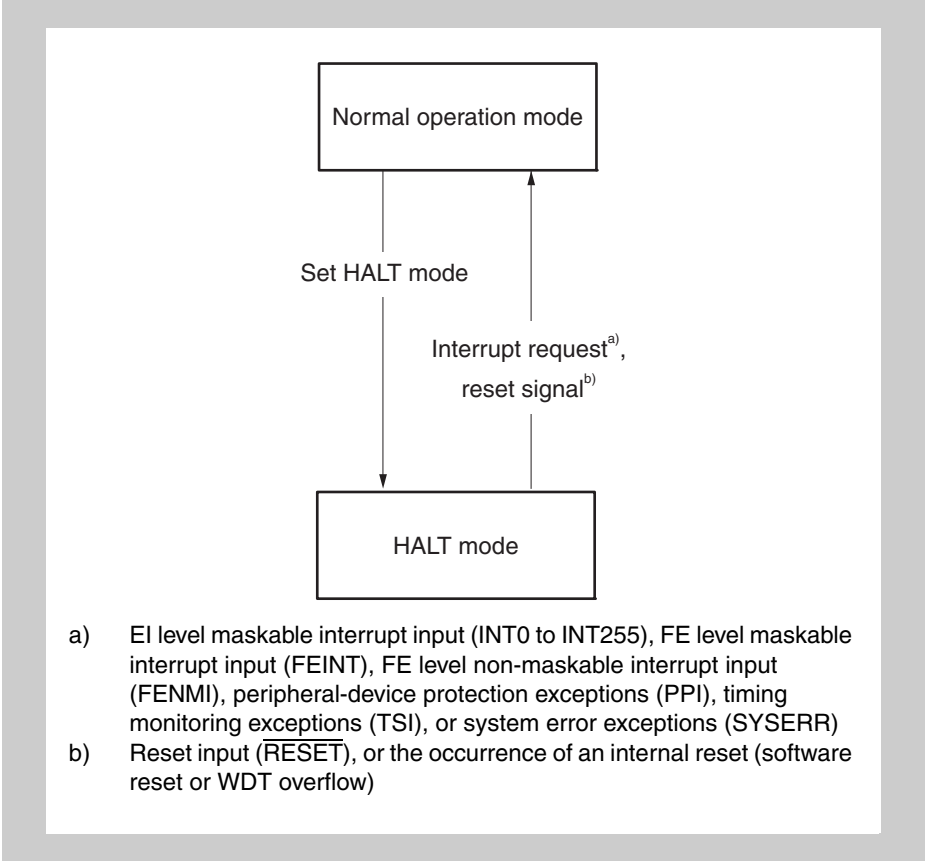
Chapter 33 Standby Function

33.1 Features

This microcontroller has a HALT mode as a standby mode.

Table 33-1 Standby Mode

Mode	Function Overview
HALT mode	Only the CPU operation clock is stopped.



a) EI level maskable interrupt input (INT0 to INT255), FE level maskable interrupt input (FEINT), FE level non-maskable interrupt input (FENMI), peripheral-device protection exceptions (PPI), timing monitoring exceptions (TSI), or system error exceptions (SYSERR)
b) Reset input ($\overline{\text{RESET}}$), or the occurrence of an internal reset (software reset or WDT overflow)

Figure 33-1 State Transition

33.2 HALT Mode

33.2.1 Setting and operation status

The HALT mode is set when a dedicated instruction (HALT) is executed in the normal operation mode.

In the HALT mode, the clock generator continues operating. Only supplying a clock to the CPU is stopped. Supplying a clock to the other on-chip peripheral functions continues.

As a result, program execution is stopped, and the internal RAM including the shared H-bus memory retains the contents before the HALT mode was set. The on-chip peripheral functions that are independent of instruction processing by the CPU continue operating.

The average current consumption of the system can be reduced by using the HALT mode in combination with the normal operation mode.

Table 33-2 Operation Status of Each Function in the HALT Mode

Function	Status
Main clock	Operates
PLL	Operates
Clock generator	Operates
CPU clock (f_{CPU})	Operates
P-bus clock (f_{PCLK})	Operates
H-bus clock (f_{HCLK})	Operates
Ethernet controller	Operates
USB function	Operates
CPU	Stops
I/O	Operates
P-bus peripheral function	Operates
WDT	Operates
A/D converter	Operates
H-bus peripheral function	Operates
Primary bus interface (P_MEMC)	Operates
Secondary bus interface (S_MEMC)	Operates
Internal data	The internal data that will not be overwritten while the CPU stops, such as the CPU registers, statuses, data, and the contents of the internal RAM (including shared H-bus memory), retains the contents from before the HALT mode was set.

33.2.2 Exiting of the HALT mode

The state transition from the HALT mode to the normal operation mode is triggered by a reset input, interrupt, or occurrence of an exception.

If an interrupt or exception is acknowledged in the HALT mode, the PC indicated by the instruction following the HALT instruction is restored after the interrupt or exception is processed. (For details, see the description of the HALT instruction in the *V850E2M Architecture Users Manual*.)

The request to exit the HALT mode is input upon the occurrence of the next reset input, interrupt, or exception request.

(1) Reset inputs, interrupts, and exceptions that exit the HALT mode

The HALT mode is exited by the following sources:

- Reset input ($\overline{\text{RESET}}$)
- Occurrence of an internal reset (a software reset or WDT overflow)
- EI level maskable interrupt input (INT0 to INT255)
- FE level maskable interrupt input (FEINT)
- FE level non-maskable interrupt input (FENMI)
- Peripheral-device protection exception (PPI)
- Timing monitoring exception (TSI)
- System error exception (SYSERR)

Even if the conditions for acknowledging an exception above (the ID and NP values) are not satisfied, the HALT mode is existed if a request has been issued. For example, the HALT mode is exited upon the occurrence of INT0 even if ID is 1.

Note that the HALT mode exiting operation differs as follows if the HALT mode is set within an exception processing routine:

- If an exception request that has a priority level equal to or lower than that of the currently processed exception request is generated, the HALT mode is exited and this request is held.
- If an exception request that has a priority level higher than that of the currently processed exception request is generated, the HALT mode is exited and this request is acknowledged.

Note that the HALT mode will not be exited if servicing of the interrupt inputs including INT0 to INT255, FEINT, and FENMI is prohibited by the following interrupt control registers:

- EI level interrupt control registers (EIC0 to EIC255)
- EI level interrupt mask registers (IMR0 to IMR15)

Note For details about the interrupt and exception requests, see *Chapter 12 “Timer Array Unit A (TAUA)” on page 550*. For details about the reset input types, see *Chapter 8 “Port Functions” on page 191*.

Chapter 34 Flash Memory

Type of flash memory

The V850E2/MN4 microcontroller includes "Type05" flash memory. For details, see the descriptions of "Type05" flash memory in *Self Programming* and other documents.

This microcontroller has internal flash memory.

Table 34-1 Flash Memory

Product Name	Internal Flash Memory
μPD70F3510F1-HN6-A	1 MB
μPD70F3512F1-HN6-A	
μPD70F3514F1-HN6-A	
μPD70F3515F1-HN6-A	2 MB

The flash memory can be rewritten by using the following methods:

- Off-board programming: Mounting the device onto a dedicated program adapter (FA series) and rewriting with a dedicated flash memory programmer through communication with a serial interface
- On-board programming: Mounting the device onto a target system and rewriting with a dedicated flash memory programmer through communication with a serial interface
- Self programming: Rewriting the flash memory by using a user-created program (application)

34.1 Features

- Erasing individual blocks is possible (block: 4 KB).
- Communication with a dedicated flash memory programmer by way of a serial interface (on-board programming)
- On-board and off-board programming are possible.
- Programming the flash memory through self programming is possible.[a]
- Erase/write prohibition (a security function): Supported
- Boot swapping: Supported

[a)] For μPD70F3514 or 70F3515, self programming can only be executed using PE1. Do not use PE2.

Table 34-2 Flash Memory Configuration

	Block 511 (4 KB)	**001F FFFF$_H$** **001F 0FFF$_H$**	**Address**
	...	**...**	
	Block 384 (4 KB)	**0018 0FFF$_H$** **0018 0000$_?$**	
	Block 383 (4 KB)	**0017 FFFF$_H$** **0017 F000$_H$**	
	...	**...**	
	Block 256 (4 KB)	**0010 0FFF$_H$** **0010 0000$_H$**	
Block 255 (4 KB)	Block 255 (4 KB)	**000F FFFF$_H$** **000F F000$_H$**	
...	...	**...**	
Block 1 (4 KB)	Block 1 (4 KB)	**0000 1FFF$_H$** **0000 1000$_H$**	
Block 0 (4 KB)	Block 0 (4 KB)	**0000 0FFF$_H$** **0000 0000$_H$**	
1 MB	2 MB	**Flash memory size**	
4, 8, 16, 32, 64, 128, or 256 KB		**Boot swap cluster size**	
μPD70F3510 μPD70F3512 μPD70F3514	μPD70F3515	**Product name**	

34.2 Rewriting by Using a Dedicated Flash Memory Programmer

The flash memory can be rewritten by using a dedicated flash memory programmer after the V850E2/MN4 is mounted on the target system (on-board programming). By using a dedicated program adapter (FA series), the flash memory can also be rewritten before the device is mounted on the target system (off-board programming).

Note The FA series is a product of Naito Densei Machida Mfg. Co., Ltd.

34.2.1 Communication methods

Communication between the dedicated flash memory programmer and the V850E2/MN4 is performed through serial communication by a one-wired UART or three-wired-HS CSI.

- One-wired UART

Pin No.	Pin Name	Function	Alternate Functions
W14	FLRXD	Pin for the flash memory programmer	TDI/FLSI

Caution Communication cannot be performed at 2 Mbps.

- Three-wired-HS CSI

Pin No.	Pin Name	Function	Alternate Functions
W13	FLSO	Pin for the flash memory programmer	TDO
W14	FLSI	Pin for the flash memory programmer	TDI/FLRXD
AB13	FLSCK	Pin for the flash memory programmer	TCK

34.3 Rewriting by Self Programming

34.3.1 Overview

The V850E2/MN4 supports a flash macro service that allows a user-created program to rewrite the internal flash memory by itself. By using this interface and a self programming library that is used to rewrite the flash memory with a user-created application program, the flash memory can be rewritten by a user-created application transferred in advance to the internal RAM or external memory. Therefore, the user-created program can be upgraded and constant data can be rewritten in the field.

(1) Boot swapping

Boot swapping makes it possible to safely change flash memory programs and maintains a software version that can be run if changing a program fails (such as due to a power failure).

(2) Protection

During flash memory programming, a series of protection flags can be specified to prohibit reading, rewriting, erasing, or other types of access to the flash memory. This protects the flash memory contents from being read or rewritten by users without permission.

Note For details about boot swapping and protection, see *34.5.4 “Sefe self programming (boot swap cluster)” on page 2353*.

34.3.2 Erasing or rewriting the flash memory

Erase The flash memory can be erased by using the following two commands, depending on the block structure:

- Erasing all the blocks at once (erasing the chip, only in the serial programming mode)
  All the blocks are erased at once.
- Erasing individual blocks
  4 KB flash memory blocks are separately erased.
  In the self programming mode, any number of sequential flash memory blocks can be erased at once.

Write In the self programming or serial programming mode, the flash memory can be written to in units smaller than one block. When all the blocks are erased, the flash memory can be written to in 16-byte units. After erasing all the blocks, data can be written once per 16-byte unit.

Erase/write The erase/write command can be used to erase a 32 KB block of flash memory and then write to a different 32 KB block.

34.4 Flash Memory Programming by Using a Flash Memory Programmer

The flash memory can be written to in the serial programming mode by using a dedicated flash memory programmer.

Serial programming

During serial programming, the microcontroller is mounted onto a target board that has a connector, and the flash memory programmer is connected to the target microcontroller.

The microcontroller must provide a complete range of operations.

- All external power supplies must be activated.
- An external oscillator must be connected to the X1 and X2 pins.

All necessary microcontroller functions are set up and executed by using the on-board firmware.

Caution When connecting the flash memory programmer to the on-board microcontroller, conflicts with other signals might occur. For details, see the tips in *(2) "Conflicts with on-board signals" on page 2345*.

34.4.1 Programming environment

The recommended environment for writing data to the microcontroller's flash memory is shown below.

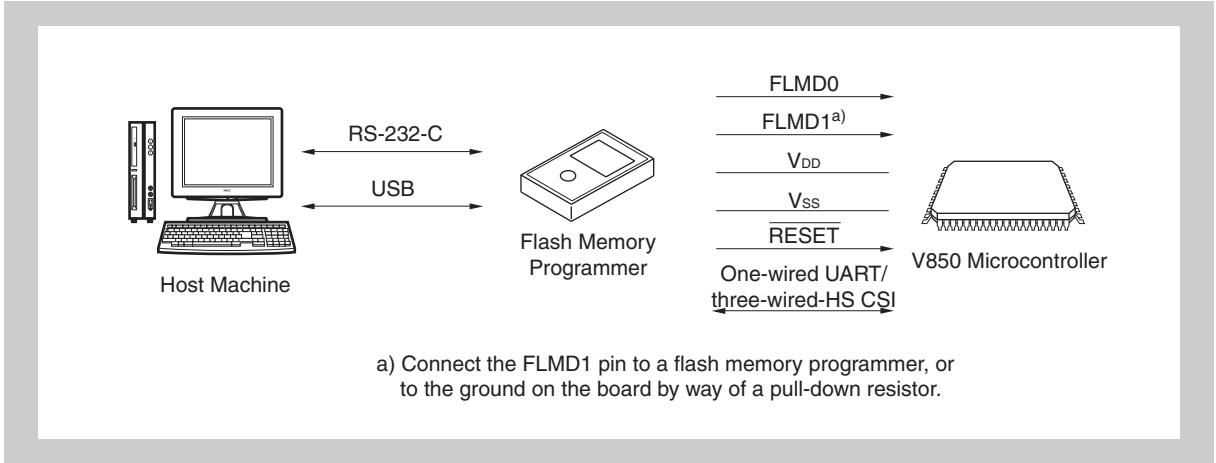


Figure 34-1 Environment for Writing Programs to the Flash Memory

A host machine is necessary for setting up the flash memory programmer. However, no host machine is necessary to use the flash memory programmer because the programmer can function in the standalone mode.

The dedicated microcontroller serial interfaces below can be used as an interface between the flash memory programmer and the microcontroller.

- Single-wire asynchronous serial programming interface: One-wired UART
- Synchronous serial programming interface: Three-wired-HS CSI

Note The one-wired UART and three-wired-HS CSI serial interfaces can only be used in the flash memory programming mode. They cannot be used in the normal operation mode.

34.4.2 Communication mode

(1) Single-wire asynchronous serial programming interface: One-wired UART

Connect the one-wired UART single-wire asynchronous serial programming interface to the flash memory programmer by using the following pin:

- Reception/transmission data: FLRXD

Various baud rates can be selected for the external flash memory programmer.

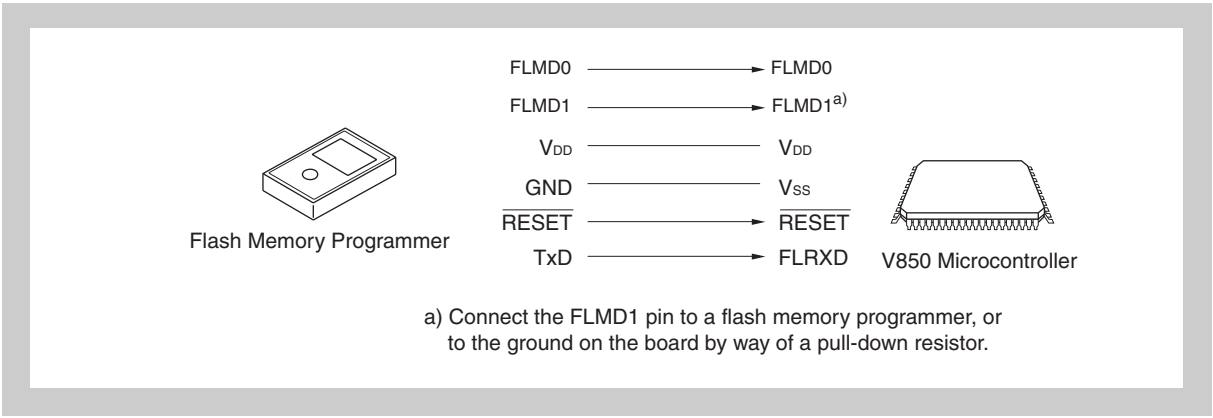


Figure 34-2 Communication with the Flash Memory Programmer by Using the One-Wired UART

Caution Communication cannot be performed at 2 Mbps.

(2) Synchronous serial programming interface: Three-wired-HS CSI

Connect the three-wired-HS CSI synchronous serial programming interface to the flash memory programmer by using the following pins:

- Serial data input: FLSI
- Serial data output: FLSO
- Serial data clock input: FLSCK

Various clock speeds can be selected for the external flash memory programmer.

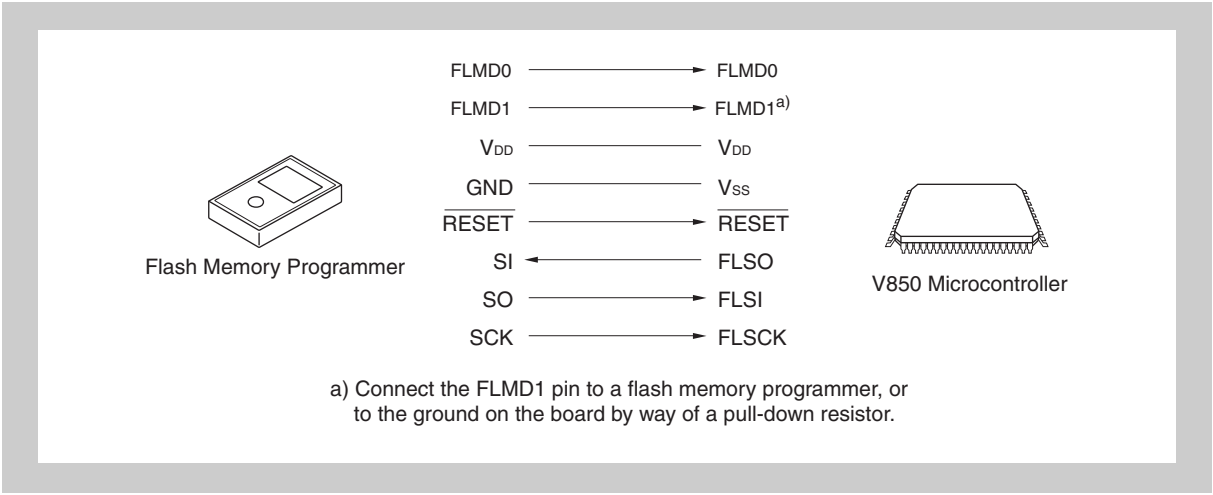


Figure 34-3 Communication with the Flash Memory Programmer by Way of the Three-Wired-HS CSI

The flash memory programmer outputs the serial data clock SCK, and the microcontroller runs as a slave.

34.4.3 Pin connections with the flash memory programmer PG-FP5

It is necessary to attach a connector to the target system and connect a flash memory programmer for serial programming. Functions for switching between the normal operation mode and flash memory programming mode, as well as for controlling the microcontroller's $\overline{\text{RESET}}$ pin, must also be provided on the board.

When the microcontroller flash memory programming mode is specified, all pins not used for flash memory programming have the same status as immediately after a reset.

If using PG-FP5 as the flash memory programmer, connect the PG-FP5 target interface connector to the microcontroller as shown below.

Table 34-3 Microcontroller Flash Memory Programmer PG-FP5 Connections

Flash Memory Programmer PG-FP5 Connection Pin			Microcontroller Signal (Alternate Function) Name			
Signal Name	I/O	Function	One-Wired UART		Three-Wired-HS CSI	
			Signal	Port	Signal	Port
SO/TxD	O	• Transmission/reception data • Transmission data	FLRXD	TDI/FLSI	FLSI	TDI/FLRXD
SI/RxD	I	Reception data	Open		FLSO	TDO
SCK	O	Transfer clock	Open		FLSCK	TCK
CLK	O	Clock to the microcontroller	Open		Open	
			Open		Open	
$\overline{\text{RESET}}$	O	Reset signal	$\overline{\text{RESET}}$	–	$\overline{\text{RESET}}$	–
FLMD0	I	Mode selection	FLMD0	–	FLMD0	–
FLMD1	I	Mode selection	FLMD1[a]	–	FLMD1[a]	–
H/S	I	Handshake signal	Open		Open	
V_{DD}	O	Power supply	EV_{DD}		EV_{DD}	
V_{DD2}	O	Power supply	IV_{DD}		IV_{DD}	
V_{PP}	–	Flash memory programming voltage	Open		Open	
GND	–	Ground	VSS		VSS	
VDE	–	Reserved	Open		Open	
RFU-1	–	Reserved	Open		Open	

a) If FLMD1 is fixed to the low level on the target board, it is not necessary to connect the FLMD1 signal.

For details, see the PG-FP5 user's manual R20UT0008E.

34.4.4 Flash memory programming control

The flash memory programming procedure is shown below.

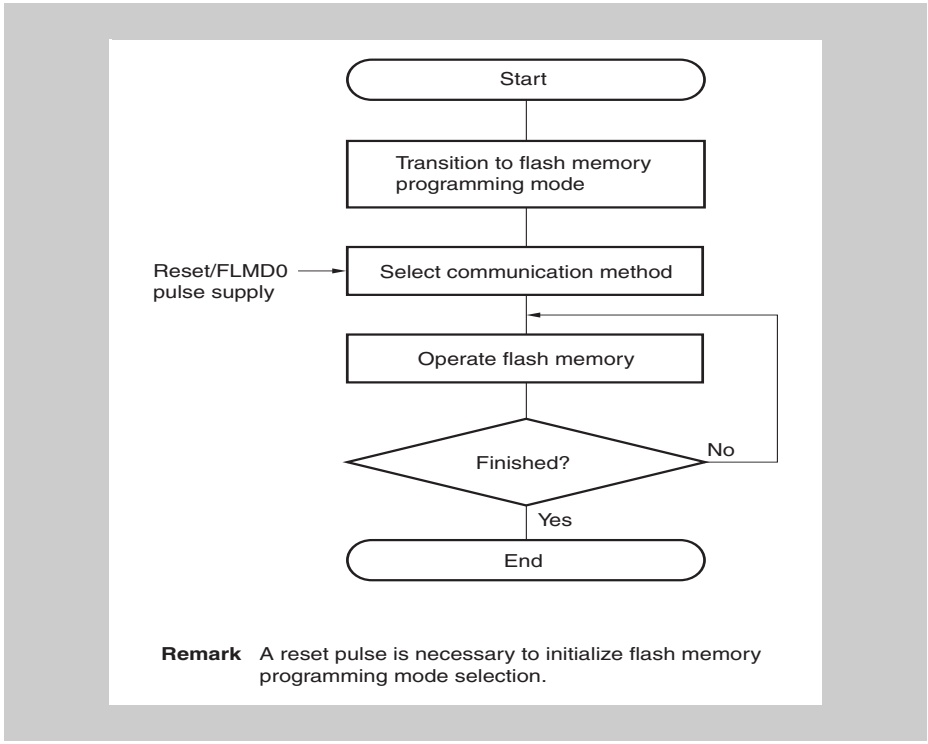


Figure 34-4 Flowchart of Flash Memory Programming Procedure

(1) Operation mode control

To use the flash memory programmer to rewrite the contents of flash memory, specify the flash memory programming mode for the microcontroller.

To specify this mode, set up the FLMD0 and FLM1 pins as shown in *Table 34-4 "Operation Mode Selection" on page 2344* and cancel $\overline{\text{RESET}}$.

During the normal operation mode, VSS is input to the FLMD0 pin. When no flash memory programmer is connected, the operation in the normal operation mode is secured by the pull-down resistor connected to the FLMD0 pin.

To specify the serial flash memory programming mode (on-board programming using an external flash memory programmer), VDD and VSS are supplied to the FLMD0 and FLMD1 pins respectively when $\overline{\text{RESET}}$ is canceled.

Table 34-4 Operation Mode Selection

Pin				Operation Mode
FLMD0	FLMD1	MODE2	MODE3	
L	L	L	L	Normal operation mode
H	L	L	L	Flash memory programming mode
	H	L	L	Boundary scan mode
Other than above				Setting prohibited

An example of connecting the FLMD0 and FLMD1 pins is shown below. The FLMD1 pin can be connected to the ground by way of a resistor. The FLMD1 pin can also be connected directly to the FLMD1 pin of the flash memory programmer.

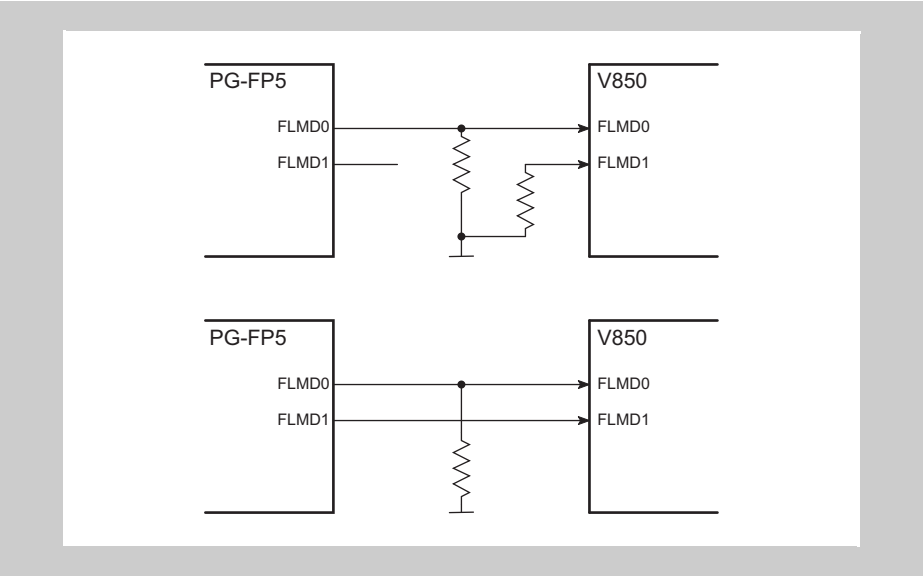


Figure 34-5 Example of Flash Memory Programmer PG-FP5 Connection

If the device is started in the normal operation mode (FLMD0 = 0), self programming can be enabled using the FLMD0 pin. For details, see *34.5 "Flash Memory Self Programming" on page 2349*.

(2) Conflicts with on-board signals

Serial I/O signals

If another device is connected to the serial interface pin used for flash memory programming in the serial programming mode, take action to prevent relevant signals from conflicting with signals from the flash memory programmer or microcontroller. Isolate the output pins of the connected device, or make them high impedance. Such action is necessary to prevent the flash memory programmer signals from causing the connected device to malfunction.

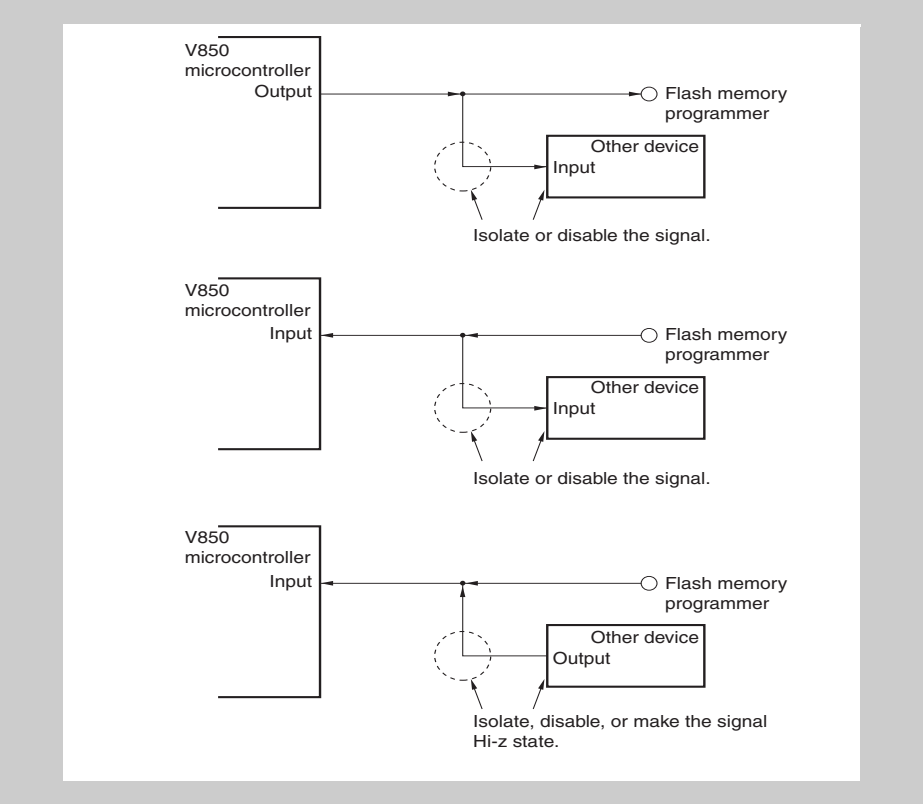


Figure 34-6 Conflict with Serial Interface Signals

$\overline{\text{RESET}}$ signal

Special care is required when connecting the $\overline{\text{RESET}}$ signal of the flash memory programmer to the on-board reset generator. Isolate or disable the reset signal output from the reset generator because it might cancel the flash memory programming processing.

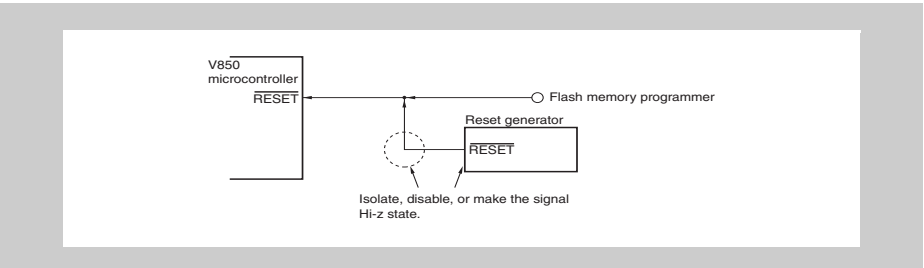


Figure 34-7 Conflict with $\overline{\text{RESET}}$ signal

Port signals During serial programming, the port pins of the V850 microcontrollers function as follows.

The pin used for programming is specified as the FLRXD or FLSO/FLSI/FLSCK pin.

The pins not used for programming are kept in their default state after a reset ends.

If the default state of a pin not used for programming is an input port or high-impedance output port, be careful of a device connected using this pin. If this device requires the level specified for the pin, connect this port to VDD or VSS by way of a resistor.

Oscillator Connect all oscillators in the same way during the normal operation mode.

Power supply Supply the same power as while in the normal operation mode to all power supply pins, including those for the reference voltage and voltage regulator.

(3) Communication mode selection

The communication interface is determined by applying the specified number of pulses to the FLMD0 pins after a reset ends. Note that this is processed using the flash memory programmer.

If a one-wired UART is selected after the FLMD0 pulses are received at 9,600 bps, the baud rate for the flash memory programmer is changed according to the user's selection specified by way of the flash memory programmer's user interface.

(4) Communication command

The flash memory programmer sends commands to the microcontroller. Upon receiving a command, the microcontroller returns the status information of required data.

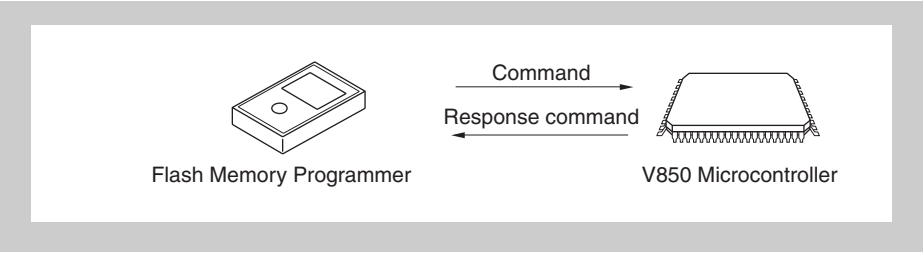


Figure 34-8 Communication command transfer

The table below lists the flash memory control commands for the microcontroller. These commands are all sent from the flash memory programmer, and the microcontroller performs the corresponding processing in response to them.

Table 34-5 Flash Memory Control Commands (1/2)

Type	Command Name	Supported Interface		Function
		One-Wired UART	Three-Wired-HS CSI	
Blank check	Chip blank check	Supported	Supported	Checks the data erase status of the flash option[a] and all areas of the code flash memory.
	Block blank check	Supported	Supported	Checks the data erase status of the specified block in the code flash memory.
Erase	Chip erase	Supported	Supported	Erases the flash option[a] and data in all areas of code flash memory.
	Block erase	Supported	Supported	Erases the contents of the specified block in the code flash memory.
Write	Write	Supported	Supported	Writes data to the specified block in the code flash memory.
Erase/write	Block erase and write	Supported	Supported	Erases the contents of the specified block in the code flash memory and writes data to that block at the same time, enabling fast processing.
Read	Read	Supported	Supported	Reads the data of the specified block in the code flash memory.
Verify	Block verify	Supported	Supported	Compares the contents of the specified block in the code flash memory with the data transmitted by the programmer.
On-chip debug security ID specification	On-chip debug security ID specification	Supported	Supported	Sets an on-chip debug security ID to flash option[a].
	On-chip debug security ID setting acquisition	Supported	Supported	Obtains the on-chip debug security ID from flash option[a].
CRC	Chip CRC	Supported	Supported	Calculates the checksum of flash option[a] and all areas of the code flash memory.
	Block CRC	Supported	Supported	Calculates the checksum of the specified block in the code flash memory.

Table 34-5 Flash Memory Control Commands (2/2)

Type	Command Name	Supported Interface		Function
		One-Wired UART	Three-Wired-HS CSI	
Option byte specification	Option byte specification	Supported	Supported	Sets the option byte setting to the flash option[a].
	Option byte setting acquisition	Supported	Supported	Obtains the option byte setting from the flash option[a].
Security specification	Security specification	Supported	Supported	Specifies the chip erase prohibit, block erase prohibit, write prohibit, read prohibit, boot area write prohibit, and flash memory shield window settings.
	Security setting acquisition	Supported	Supported	Obtains the security settings.
System setting and control	Reset	Supported	Supported	Used for communication synchronization.
	Oscillation frequency specification	Supported	Supported	Specifies an oscillation frequency.
	Baud rate specification	Not supported	Supported	Specifies a baud rate for using UART.
	Silicon signature	Supported	Supported	Obtains the signature information.

a) Flash option is a collective term indicating the security, write protection, reset vector handling function, option byte, and on-chip debug security ID settings.

Note For details about the on-chip debug ID, see the operation manual of the software tool used.

34.5 Flash Memory Self Programming

This V850 microcontroller supports a flash macro service that allows a user-created program to rewrite the internal flash memory by itself.

By using the flash macro service and self programming library (FSL) in a user-created application program, the flash memory can be rewritten by the data transferred in advance to the internal RAM or external memory.

This makes it possible to upgrade the user-created program and rewrite constant data fields.

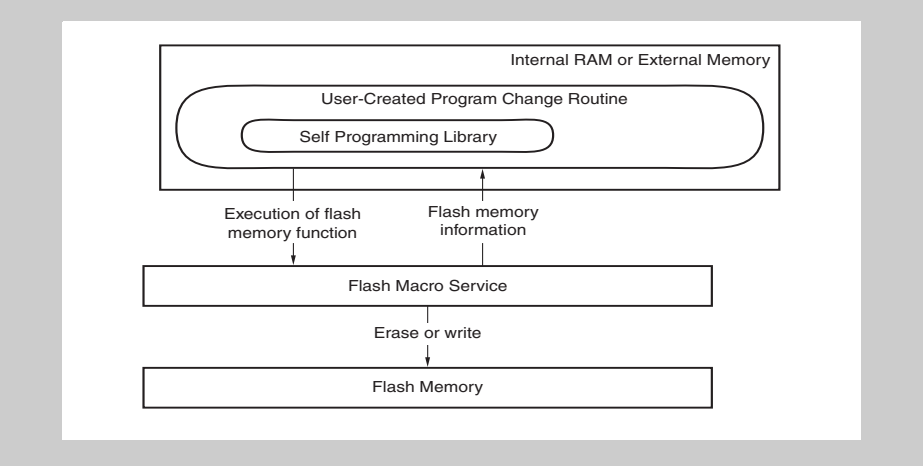


Figure 34-9 Self Programming Schematic

Note that the flash memory cannot be accessed during self programming. Therefore, the program can be executed only by fetching an instruction from the internal RAM or external memory.

Before activating self programming, therefore, it is necessary to copy the instructions of the software routine that changes the user-created program while in the self programming procedure from the flash memory to the internal RAM or external memory. Because no interrupt can be serviced by an interrupt vector in the flash memory during self programming, a special function for acknowledging interrupts in the internal RAM is provided. (For details, see *34.5.5 “Interrupt service during flash memory self programming” on page 2357*.)

Caution When using the flash self programming library, there are notes concerning data access using the gp and ep registers during interrupt processing. The notes regarding the gp register are applicable to this product.

For general information about flash memory self programming, see the application notes that describe self programming.

34.5.1 Activating self programming

The self programming function can be activated from the normal user mode of the microcontroller.

- Set the external FLMD0 pin to high.
  To do this, some external components must be set up, wiring is required (such as connecting an output port to the FMLD0 pin), and the FLMD0 must be pulled down (a 100 kΩ pull-down resistor is required).
- Specify the internal register bit FLMDCNT.FLMDPUP.
  At this time, pull down the FLMD0 pin (a 100 kΩ pull-down resistor is recommended).

The following register is used to internally enable self programming by using software.

(1) FLMDCNT: FLMD control register

This register is used to control the on-chip pull-up and pull-down resistors that are connected to the FLMD0 pin and are used to enable and disable self programming, respectively.

Protection Writing to this register is protected by a special sequence of instructions by using the protection command register FLMDPCM

Access This register can be read/written in 8-bit units.

Address FF43 8000_H

Initial value 00_H

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	FLMDPUP
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 34-6 FLMDCNT register contents

Bit position	Bit name	Function
0	FLMDPUP	This bit controls pull-up and pull-down of the FLMD0 pin. 0: Activate the pull-down resistor connected to the FLMD0 pin (self programming mode disabled). 1: Activate the pull-up resistor connected to the FLMD0 pin (self programming mode enabled).

(2) FLMDPCMD: FLMD protection command register

This register is the protection command register for the FLMDCNT register.

Access This register can be written in 8-bit units.

Address $FF438004_H$

Initial value Undefined

7	6	5	4	3	2	1	0
FLMDPC[7:0]							
W	W	W	W	W	W	W	W

Table 34-7 FLMDPCMD register contents

Bit position	Bit name	Function
7 to 0	FLMDPC[7:0]	These bits function as a protection command to enable writing to FLMDCNT.

(3) FLMDPS: FLMD protection error status register

This register indicates whether data was successfully written to the write-protected register FLMDCNT.

Access This register can be written in 8-bit units.

Address $FF438008_H$

Initial value 00_H

7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	FLMD PRERR
R	R	R	R	R	R	R	R

Table 34-8 FLMDPS register contents

Bit position	Bit name	Function
0	FLMDPRERR	These bits indicate whether data write to the write-protected register FLMDCNT was successful. 0: Successful 1: Failed

34.5.2 Self programming library functions

Flash memory self programming by the user-created program is supported by the self programming library.

This library provides a set of C function calls that execute the following basic features:

- Blank check, erase, rewrite, and verify for the flash memory
- Boot swap cluster (including boot cluster specification)
- Protection flag setting
- Acquisition of various types of information related to the flash memory

For details about how to use the library, see the application notes that describe self programming.

34.5.3 Occupation of internal RAM by self programming

During self programming, the upper 3.5 KB of the internal RAM (from $FEDFF200_H$ to $FEDFFFFF_H$) are occupied to process the self programming. Therefore, the contents of the 2 KB RAM are changed during self programming, and it might be necessary to restore the contents using the user-created program.

Note In addition, it might be necessary to use the RAM as intermediate memory for storing user data and code when data is copied from the flash memory to the RAM during self programming.

34.5.4 Sefe self programming (boot swap cluster)

The V850 flash microcontroller supports a mechanism that swaps the cluster in the flash memory block starting from 0000 0000_H with the upper adjacent cluster of the same size.

Boot swap cluster

Because the cluster of the boot block to be swapped, which starts from the group address 0000 0000_H, contains the entry point of the user-created program at the default reset vector 0000 0000_H, it is called the *active boot swap cluster*.

Boot swapping flag

Which of the two clusters is the boot swap cluster is specified using the self programming library during flash memory programming. *Figure 34-10 "Boot Swap Cluster Swapping" on page 2353* shows an example of the boot block swap function for a flash memory block of cluster size 4. When `boot_flag` is inverted, it becomes `not (boot_flag)` and blocks 4 to 7 become the active boot cluster. At this time, the user-created program starts from the new boot swap cluster the next time a reset ends.

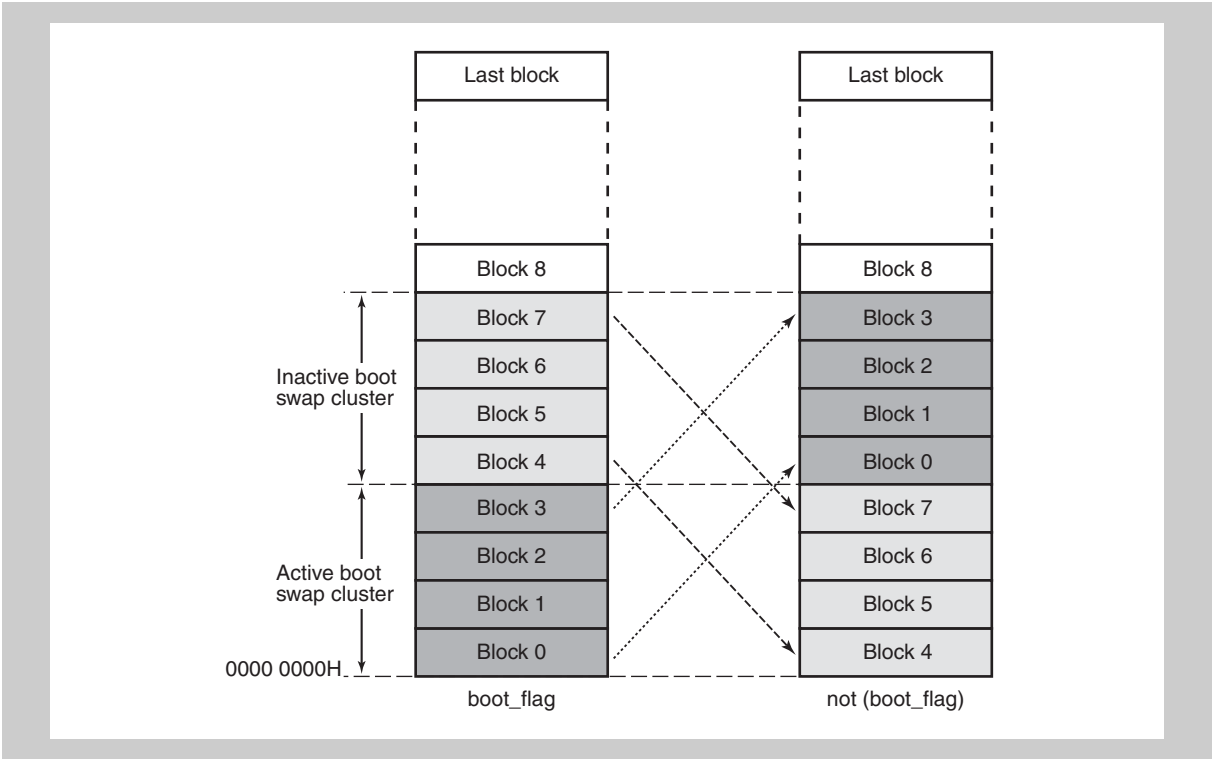


Figure 34-10 Boot Swap Cluster Swapping

Safe self programming

The boot swap cluster function makes self programming safe. When a boot code is rewritten, the new code is written to the boot cluster, and `boot_flag` retains the previous contents. When the boot cluster is rewritten normally, `boot_flag` is inverted and the new boot code become active. Even if the new boot code failed to be rewritten due to a power failure, unexpected reset, or other error, rewriting can be executed again because the old boot code remains active.

Boot cluster

The boot code might be smaller than that of the boot swap cluster.

A flash memory block, which is a part of the boot code, is called a *boot cluster*. The number of boot blocks that are cluster members can be specified by using the self programming library during self programming.

The size of the boot swap cluster is determined by that of the boot cluster. In other words, it is determined according to the number of boot blocks specified during self programming.

Table 34-9 “Correspondence between Boot Block and Boot Swap Cluster” on page 2355 shows the correspondence between the number of boot blocks, boot cluster size, and boot swap cluster.

Number of boot blocks

The number of boot blocks must be specified by the user during self programming. The specified number of boot blocks determine the target boot block for boot cluster protection that protects the block from being erased or written.

Boot block protection

To prohibit boot block rewriting, specify the boot cluster protection setting using the self programming library during flash memory programming. Once this setting is specified, the active boot cluster block is not erased or written, and is also protected from the boot swap cluster function. Note that only the active boot cluster block is protected. In *Figure 34-11 “Boot Swap Cluster Example” on page 2356*, for example, blocks 0 an 1 are protected from being erased or written while blocks 2 and 3 are not.

Caution Once boot cluster protection is activated, it cannot be inactivated.

Table 34-9 Correspondence between Boot Block and Boot Swap Cluster

Last address of boot block	Boot swap cluster		Cluster to be replaced with boot swap cluster	Boot block cluster	
	Size	Address	Address	Size	Address
00_H	4 KB	$0000\ 0000_H$ to $0000\ 0FFF_H$	$0000\ 1000_H$ to $0000\ 1FFF_H$	4 KB	$0000\ 0000_H$ to $0000\ 0FFF_H$
01_H	8 KB	$0000\ 0000_H$ to $0000\ 1FFF_H$	$0000\ 2000_H$ to $0000\ 3FFF_H$	8 KB	$0000\ 0000_H$ to $0000\ 1FFF_H$
02_H	16 KB	$0000\ 0000_H$ to $0000\ 3FFF_H$	$0000\ 4000_H$ to $0000\ 7FFF_H$	12 KB	$0000\ 0000_H$ to $0000\ 2FFF_H$
03_H				16 KB	$0000\ 0000_H$ to $0000\ 3FFF_H$
04_H	32 KB	$0000\ 0000_H$ to $0000\ 7FFF_H$	$0000\ 8000_H$ to $0000\ FFFF_H$	20 KB	$0000\ 0000_H$ to $0000\ 4FFF_H$
...				...	...
07_H				32 KB	$0000\ 0000_H$ to $0000\ 7FFF_H$
08_H	64 KB	$0000\ 0000_H$ to $0000\ FFFF_H$	$0001\ 0000_H$ to $0001\ FFFF_H$	36 KB	$0000\ 0000_H$ to $0000\ 8FFF_H$
...				...	...
$0F_H$				64 KB	$0000\ 0000_H$ to $0000\ FFFF_H$
10_H	128 KB	$0000\ 0000_H$ to $0001\ FFFF_H$	$0002\ 0000_H$ to $0003\ FFFF_H$	68 KB	$0000\ 0000_H$ to $0001\ 0FFF_H$
...				...	...
$1F_H$				128 KB	$0000\ 0000_H$ to $0001\ FFFF_H$
20_H	256 KB	$0000\ 0000_H$ to $0003\ FFFF_H$	$0004\ 0000_H$ to $0007\ FFFF_H$	132 KB	$0000\ 0000_H$ to $0002\ 0FFF_H$
...				...	...
$3F_H$				256 KB	$0000\ 0000_H$ to $0003\ FFFF_H$
40_H	Boot swap disabled			260 KB	$0000\ 0000_H$ to $0004\ 0FFF_H$
...				...	...
FF_H				1024 KB	$0000\ 0000_H$ to $000F\ FFFF_H$

Maximum size of boot swap cluster

The maximum size of the boot cluster is 256 KB. Therefore, a flash memory area whose size exceeds 512 KB cannot be used for boot swap cluster.

Figure 34-11 "Boot Swap Cluster Example" on page 2356 shows a boot swap example in which the cluster consists of four flash memory blocks. When boot_flag is inverted, it becomes not (boot_flag), and blocks 4 to 7 make up an active boot block cluster. Therefore, the user-defined program starts from the replaced boot swap cluster after the next reset.

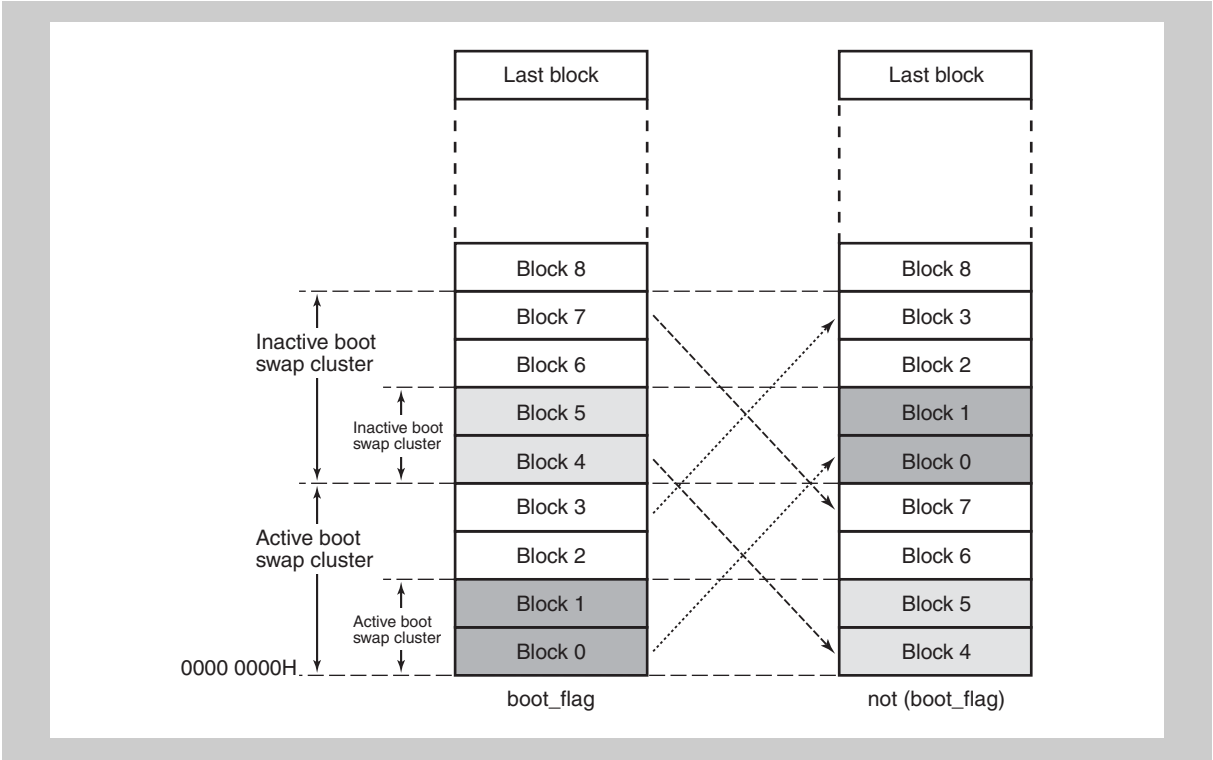


Figure 34-11 Boot Swap Cluster Example

Secure self programming

The boot swap function enables secure self programming. When rewriting a boot code, the new code is written into the inactive cluster (blocks 4 to 7) while boot_flag remains in its original state (the cluster of blocks 0 to 3 is active). Once the inactive cluster (blocks 4 to 7) is successfully rewritten, `boot_flag` is changed (boot swap) and the new boot code (the cluster of blocks 4 to 7) becomes active.
For example, even if the new boot code rewrite failed due to a blackout or unintentional reset, rewrite can be restarted because the original boot code remains active.

Boot block cluster

The boot block cluster is a boot area for the application program, and area selection is possible.
The security setting to disable rewriting can be specified for the boot block cluster.

Caution Once boot cluster protection is activated, it cannot be inactivated.

34.5.5 Interrupt service during flash memory self programming

This microcontroller provides a function that maintains interrupt servicing during self programming.

When self programming is active, neither interrupt vector tables nor interrupt handler routines placed in the flash memory can be accessed. It is therefore necessary to acknowledge interrupts somewhere other than the flash memory, such as the internal RAM.

Interrupt handler addresses can be switched by using software.

For details, see *6.4 "Exception handler adders switching function"* in Volume 2 of the V850E2M Architecture User's Manual (R01US0001E).

For general information about flash memory self programming, see the application notes that describe self programming.

Chapter 35 On-chip Debugging Unit (OCD)

This microcontroller has an on-chip debugging function. By using an on-chip debugging emulator, programs can be debugged on the microcontroller mounted on the target system.

The debugging function provided for this microcontroller complies with the Nexus debugging interface standards IEEE-ISTO 5001™2003 Class 1. This function also complies with Class 3 when trace pins are used.

Caution Although the debugging function described in this chapter is supported by this microcontroller, whether it can be used depends on the debugger. For details about the debugger, see its operation manual.

35.1 V850E2/MN4 On-chip Debugging Function

35.1.1 Peripheral break

The module performs one of the following operations when the debugger stops the microcontroller's operation during a peripheral break:

- Always stops (unconditional peripheral break)
- Optionally stops (peripheral break function)
- Continues operation

Peripheral break For peripheral break, the following points are referenced during a debugging session:

- Break point hit
- Manual break

(1) Module subject to unconditional peripheral break

The table below shows the module that always stops when a peripheral break occurs.

Table 35-1 Module That Stops upon a Peripheral Break (Unconditional Peripheral Break)

Module
Window Watchdog Timer A (WDTAn)

(2) Modules that continue operation upon a peripheral break

The table below shows the modules that continue operation when a peripheral break occurs.

Table 35-2 Modules That Continue Operation upon a Peripheral Break

Module
Timers (TAUA, TAUJ, ENCA, and OSTM)
Serial interface modules (UARTE, CSIH, and I^2CB)
A/D converter (ADCA)

35.1.2 Signal mask

The external signals of the V850E2/MN4 below are not affected by on-chip debugging because they can be masked. This microcontroller is controlled by the on-chip debugging unit.

- $\overline{\text{RESET}}$
- $\overline{\text{P\_HLDRQ}}$
- $\overline{\text{P\_WAIT}}$

35.2 Description

The following provides an overview of the on-chip debugging function.

(1) Debugging interface

This interface is used to communicate with the host machine by way of an on-chip debugging emulator by using $\overline{\text{TRST}}$, TCK, TMS, TDI, TDO, and $\overline{\text{TRDY}}$ signals.

(2) Debug monitoring function

If the monitor program stored in the memory area is executed and debugging is performed while the user-created program is stopped, the following basic debugging functions can be monitored:

- Downloading user-created programs
- Reading and writing memory and registers
- Executing user-created programs started from any address

(3) Hardware break function

Up to four break points for an instruction or data can be specified. If a break point for an instruction is specified, execution can be interrupted at a specified address. If a break point for data is specified, execution can be interrupted when data at a specified address is accessed.

The break conditions can be combined by using up to two level sequences.

(4) Software break function

Execution of a user-created program stored in RAM can be interrupted at a specified address.

(5) Forced break function

Execution of a user-created program can be forcibly interrupted.

(6) Forced reset function

The microcontroller can be forcibly reset.

(7) Real-time RAM monitor (RRM)

The memory can be read during program execution. Because this read access uses the DMA dedicated to debugging, the effects on program execution can be minimized.

(8) Dynamic memory modification (DMM)

The memory can be written during program execution. Because this write access uses the DMA dedicated to debugging, the effects on program execution can be minimized.

(9) Timer function

A 32-bit counter is used to measure the time to execute the user-created program based on the clock that is obtained by dividing the DCUTCK signal frequency by two.

(10) Mask function

Some external dedicated signals are not affected by on-chip debugging because they can be masked. This microcontroller is controlled by the on-chip debugging unit.

For a list of these signals, see *"Signal mask"* at the beginning of this chapter.

(11) Selection of whether to continue or stop peripheral module operation upon a break

The microcontroller's modules perform one of the following operations when a break point is reached:

- Always stop operation upon a break
- Operation upon a break can be specified by using user options and the module emulation registers.
- Always continue operation upon a break

The operations above only apply to the corresponding modules.

(12) Hot attach function

When an on-chip debugging emulator is connected, debugging can be started without resetting the running CPU.

(13) Security function

A 96-bit ID code can be written to the microcontroller to prevent the contents of the flash memory from being read by unauthorized users. The flash memory cannot be accessed until the code entered by the user when the debugger is started matches the ID code written to the microcontroller. If "0" is specified for the last bit (bit 95), however, the flash memory cannot be accessed regardless of whether the entered code matches the specified ID code.

For details about how to specify an ID code, see the operation manual of the software tool used.

35.3 Connection with on-chip debug emulator

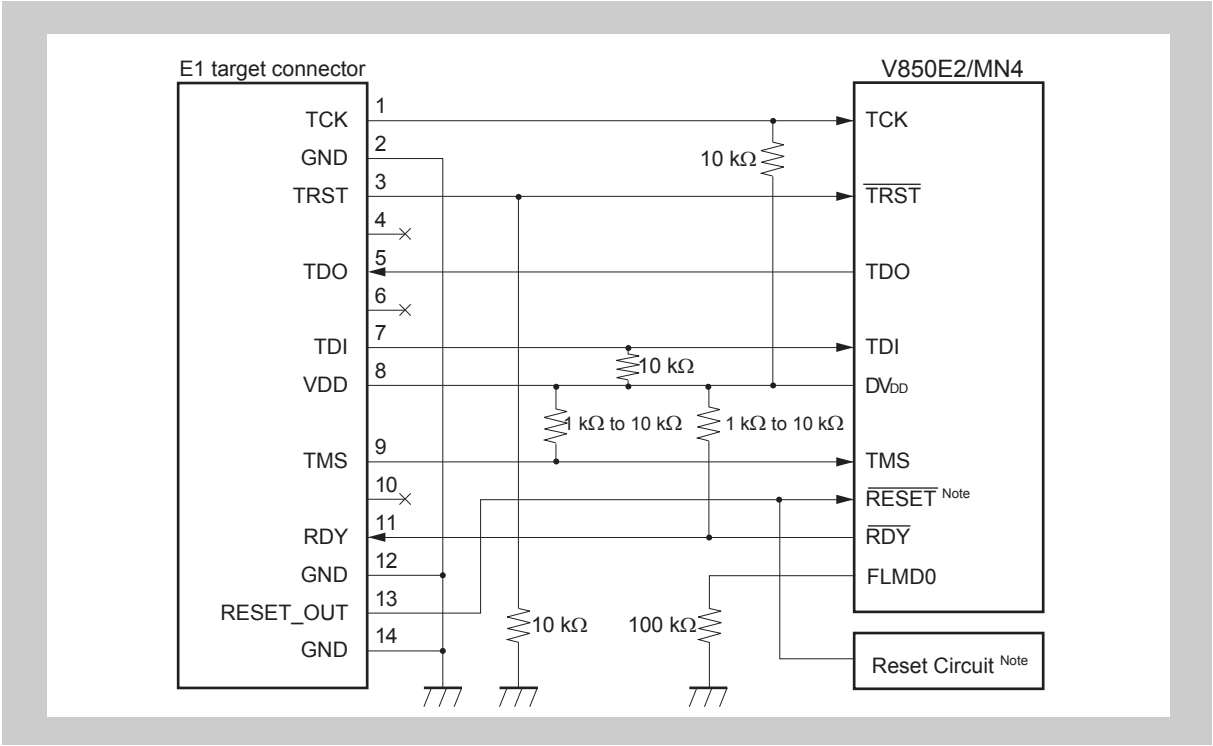


Figure 35-1 Connection with on-chip debug emulator

Table 35-3 Pins used for connection with on-chip debug emulator

Pin name	Description
VDD	Signal used to detect the power supply voltage of the target system and that of the buffer in the on-chip debug emulator
$\overline{\text{TRST}}$	Signal used to asynchronously reset the debugging function of the microcontroller
TCK	Clock signal used for debugging
TMS	Signal used to select data communication transfer mode
TDI	Data signal input to the microcontroller
TDO	Data signal output from the microcontroller
$\overline{\text{TRDY}}$	Synchronization signal for data communication
$\overline{\text{RESET}}$	Reset signal for the microcontroller. When this pin is connected, the microcontroller retains the reset status until the system is turned on and the debugger starts.
FLMD0	Mode signal used to rewrite the flash memory of the microcontroller

35.4 Cautions on Using On-chip Debugging

(1) Usage of devices used for debugging

Do not mount a device that was used for debugging onto a mass-produced product, because the flash memory was rewritten during debugging and the number of rewrites of the flash memory cannot be guaranteed.

Chapter 36 Boundary Scan

36.1 Features

A boundary scan is a test method specified by IEEE 1149.1 that is used to sequentially test the external I/O pins of a device and then input or output the test data. This makes it possible to check the connections between devices mounted on an application system board.

36.2 Boundary Scan Environment

The host machine is communicated with by way of the JTAG test tool by using the $\overline{\text{TRST}}$, TCK, TMS, TDI, and TDO signals.

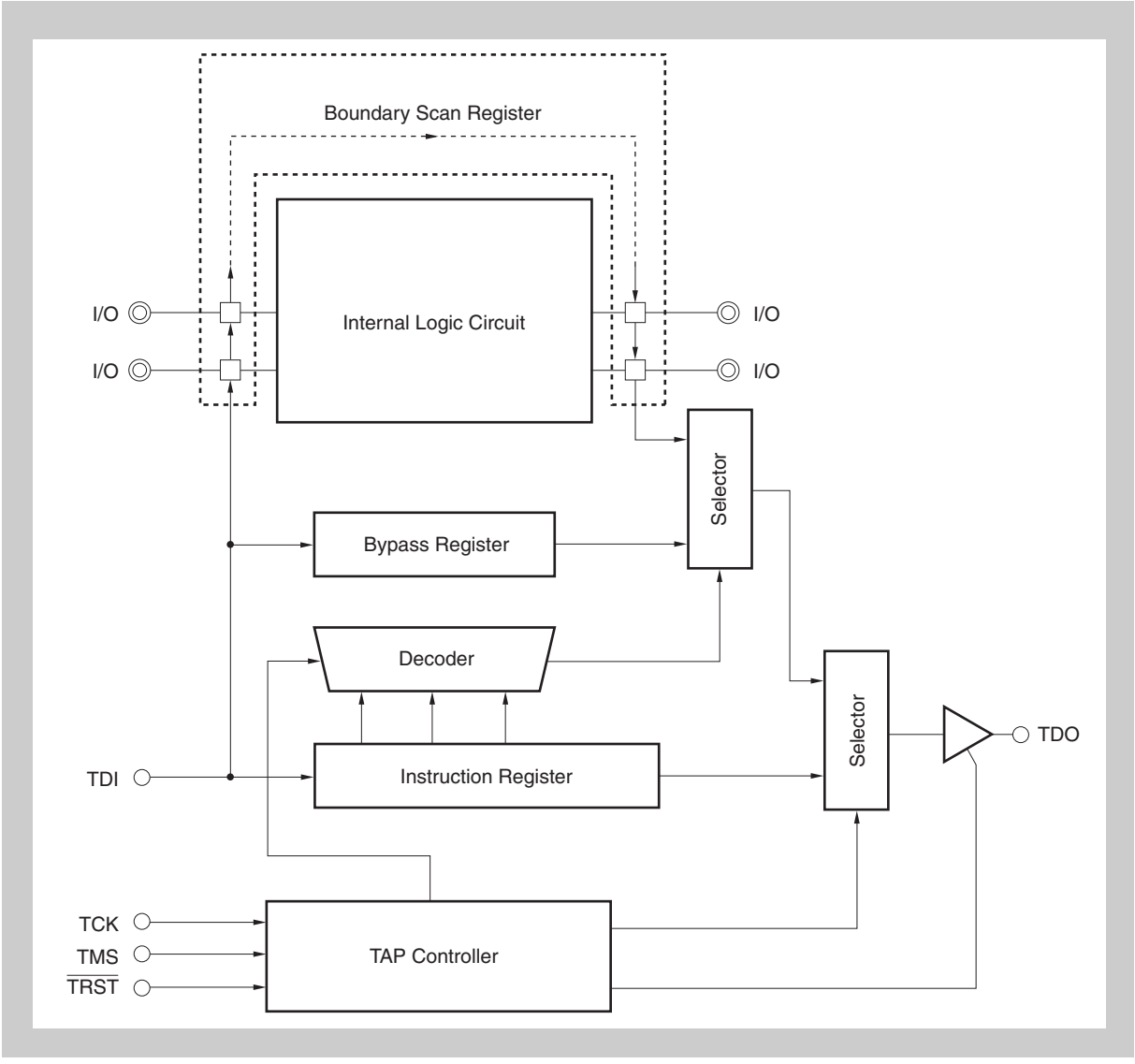


Figure 36-1 Boundary Scan Block Diagram

36.3 Boundary Scan Mode

When performing a boundary scan, set the system to the boundary scan mode.

To switch to this mode, set up the FLMD0, FLMD1, MODE2, and MODE3 pins, and then cancel the reset.

Table 36-1 Boundary Scan Mode Settings

FLMD0	FLMD1	MODE2	MODE3	Operation Mode
L	L	L	L	Normal operation mode
H	L	L	L	Flash memory programming mode
H	H	L	L	Boundary scan mode
Other than the above				Setting prohibited

Note L: Low level H: High level

36.4 TAP (Test Access Port)

Use the following pins for boundary scan I/O.

(1) $\overline{\text{TRST}}$

This is the test reset input pin. This pin acknowledges asynchronous input and resets the boundary scan circuit when low-level input is received.

The $\overline{\text{TRST}}$ pin is pulled down on the chip by using a resistor.

(2) TCK

This is the test clock input pin. All input signals are captured at the rising edge of TCK, and output starts at the falling edge of TCK.

(3) TMS

This is the test mode selection pin. It inputs commands to the TAP controller.

(4) TDI

This is the test data input pin. It functions as the input for the serial registers placed between TDI and TDO.

The status of these registers is determined by the TAP controller status and instructions captured by the TAP controller.

(5) TDO

This is the test data output pin. It functions as the output for the serial registers lined up between TDI and TDO.

The output changes at the falling edge of TCK.

36.5 TAP Controller

36.5.1 Target pins

All pins are subject to boundary scans except those in *Table 36-2 "Pins Not Subject to Boundary Scans"*.

Table 36-2 Pins Not Subject to Boundary Scans

Type	Pin Name
JTAG I/F	$\overline{\text{TRST}}$, TCK, TMS, TDI, TDO, and RDY
MODE pins	FLMD0, FLMD1, MODE2, and MODE3
Analog input pins	ANI01 to ANI05
Clock pins	X1 and X2
Flash test pins	VPPTS2 and VPPTS3
Power supply pins	V_{DD}, EV_{DD}, DV_{DD}, $OSCV_{DD}$, $PLLV_{DD}$, UV_{DD}, AV_{DD}, and AV_{REFP}
GND pins	V_{SS}, EV_{SS}, DV_{SS}, $OSCV_{SS}$, $PLLV_{SS}$, UV_{SS}, AV_{SS}, and AV_{REFM}
USB pins	UDPF, UDMF, UDPH, and UDMH

36.5.2 Instructions

The V850E2/MN4 supports the BYPASS, EXTEST, SAMPLE, and PRELOAD instructions.

Table 36-3 Instruction Codes

Instructions	Instruction Codes		Remark
	20…16	15……0	
BYPASS	1 1111	1111 1111 1111 1111	
EXTEST	1 1111	1111 1111 1110 1000	
SAMPLE	1 1111	1111 1111 1111 1000	Same code as PRELOAD
PRELOAD	1 1111	1111 1111 1111 1000	Same code as SAMPLE

(1) BYPASS

When the BYPASS instruction is captured by the instruction register, the TAP controller enters the shift-DR status and places the bypass register between TDI and TDO.

The board-level scan path for the BYPASS instruction makes it easy to test the scan paths of other devices by using the shortest possible route.

(2) EXTEST

The EXTEST instruction makes it possible to test an external circuit from a JTAG circuit.

The test vector is specified for the boundary scan register cell on the output pin side, and the test results are captured on the input pin side. Normally, the PRELOAD instruction is executed before the EXTEST instruction to specify the first test vector for the boundary scan register cell. The result of this is that, if the TAP controller enters the Update-IR status during EXTEST instruction execution, the output driver is enabled, and the PRELOAD data is output from the output pin.

(3) SAMPLE/PRELOAD

SAMPLE/PRELOAD is a general instruction for which the execution details are specified by IEEE 1149.1.

When the SAMPLE/PRELOAD instruction is captured by the instruction register, the TAP controller enters the capture-DR status, and the data input to the I/O pin is captured by the boundary scan register.

36.5.3 Scan registers

(1) Instruction register

The instruction register is used to store instructions to be issued by the TAP controller.

When the instruction register is placed between TDI and TDO, instructions are captured by the register.

When the power supply is turned on, the instruction register captures an IDCODE instruction. At this time, the TAP controller is reset to the Test-Logic-Reset status.

(2) Bypass register

The bypass register is a 1-bit register placed between TDI and TDO.

The bypass register transfers sequential test data to other devices along the shortest possible route by way of the TAP controller.

(3) ID register

The ID register is a 32-bit register.

When the TAP controller enters the capture-DR status after the instruction register captures an IDCODE instruction, the ID register captures a 32-bit device code and manufacturer code.

When the TAP controller enters the shift-DR status, the ID register is placed between TDI and TDO.

Table 36-4 ID Register Codes

Device Name	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
μPD70F3510	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	1	1	0	0	0	0	0	0	1	0	0	0	0	1
μPD70F3512	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	1
μPD70F3514	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	1	0	1	0	0	0	0	0	0	1	0	0	0	0	1
μPD70F3515	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	1

(4) Boundary scan register

The boundary scan register is controlled by the TAP controller.

When the TAP controller enters the capture-DR status, the boundary scan register captures the contents of the RAM I/O ring. When the TAP controller enters the shift-DR status, the boundary scan register is placed between TDI and TDO. Several TAP instructions use the boundary scan register.

36.5.4 Status transitions

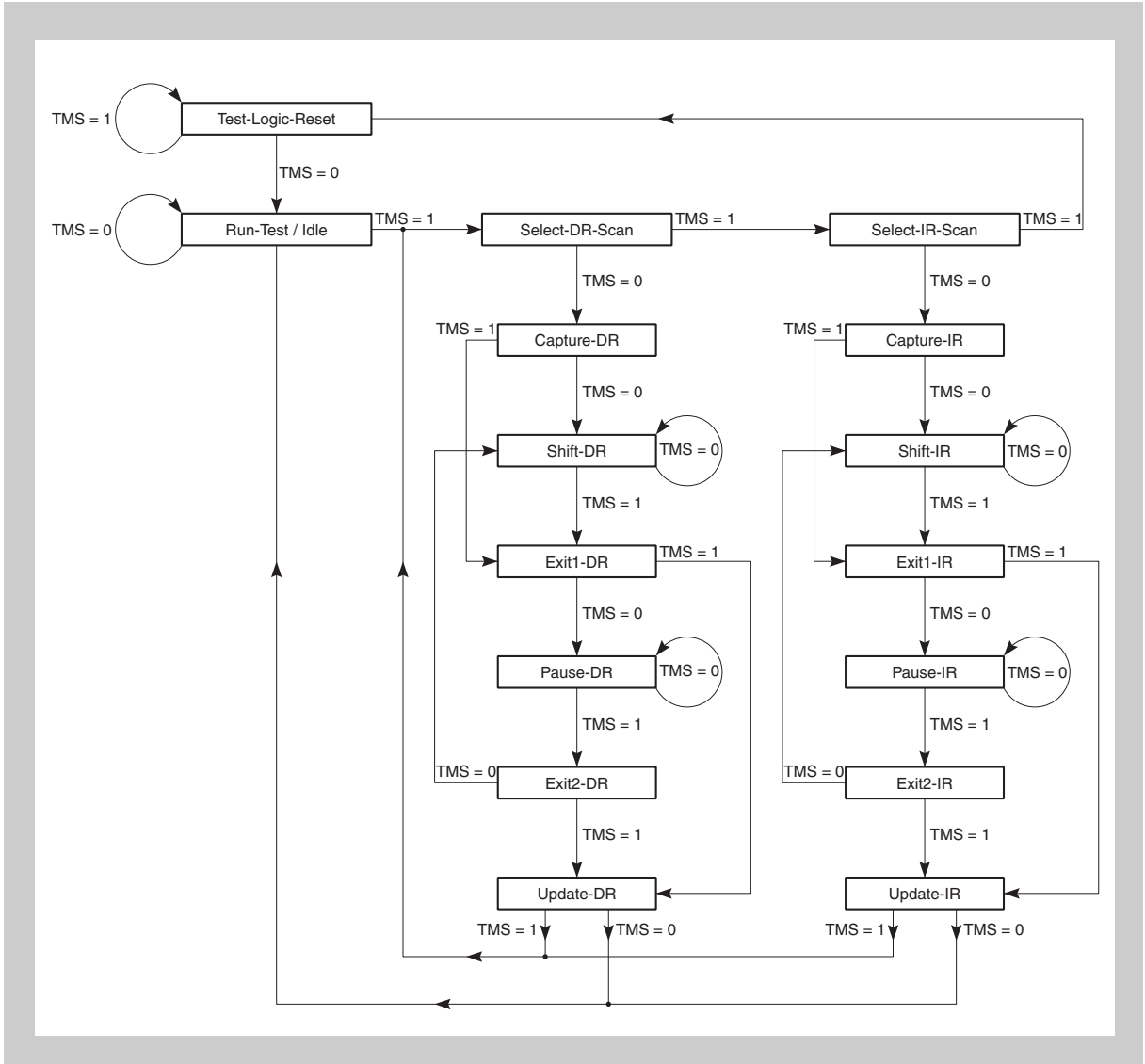


Figure 36-2 TAP Controller Status Transitions

Chapter 37 Electrical Specifications

37.1 Absolute Maximum Ratings

T_A = 25°C

Item	Symbol	Conditions	Rating	Unit
Power supply	IV_{DD}	IV_{DD} pin	–0.5 to +1.6	V
	EV_{DD}	EV_{DD} pin	–0.5 to +4.1	V
	$OSCV_{DD}$	$OSCV_{DD}$ pin	–0.5 to +4.1	V
	$OSCV_{SS}$	$OSCV_{SS}$ pin	–0.5 to +0.5	V
	$PLLV_{DD}$	$PLLV_{DD}$ pin	–0.5 to +1.6	V
	$PLLV_{SS}$	$PLLV_{SS}$ pin	–0.5 to +0.5	V
	AV_{DD}	AV_{DD} pin	–0.5 to +6.0	V
	AVss	AVss pin	–0.5 to +0.5	V
	UV_{DD}	UV_{DD} pin	–0.5 to +4.1	V
	DV_{DD}	DV_{DD} pin	–0.5 to +4.1	V
Input voltage	V_I	The X1 pin, 5 V tolerant pin[a], and analog input pin[b] are excluded.	–0.5 to +4.1[c]	V
		5-V tolerant pin[a] (except during output)	–0.5 to +6.0	V
Clock input voltage	V_K	X1 pin	–0.5 to $OSCV_{DD}$ + 0.5[d]	V
Low-level output current	I_{OL}	Per pin	10.0	mA
		Total of all pins	160	mA
Output current, high	I_{OH}	Per pin	–10.0	mA
		Total of all pins	–160	mA
Output voltage	V_O		–0.5 to EV_{DD} + 0.5	V
Analog input voltage	V_{WASN}	Analog input pin[b] $AV_{REFP} \leq AV_{DD}$ AV_{DD} must not exceed +0.3 V.	–0.3 to AV_{REFP} + 0.3	V
A/D converter reference input voltage	AV_{REFP}	Do not exceed 6.0 V.	–0.3 to AV_{DD} + 0.3	V
	AV_{REFM}		–0.3 to +0.3	V
Package surface temperature	T_t	Normal operation	–40 to +100	°C
		During flash memory rewriting	–40 to +100	°C
Storage temperature	T_{stg}		–65 to +150	°C

a) The 5-V tolerant pin is assigned to one of the following pin numbers: Y2, AA1, AA2, AB2.
b) The analog input pin is assigned to one of the following pin numbers: V5, V6, V7, V8, W5, W6, W7, W8, AA5, AA8, AB5, AB8.
c) Do not exceed the absolute maximum rating of EV_{DD} (the MAX value).
d) Do not exceed the absolute maximum rating of the $OSCV_{DD}$ pin (the MAX value).

Cautions

1. Do not connect the output (an I/O pin in the output status) of an IC product to a different output pin (including an I/O pin in the output status), and do not directly connect power supply pins such as IV_{DD} and EV_{DD} or GND pins. However, direct connection of the I/O pins between an IC product and an external circuit is possible, if the I/O pins can be set to the high-impedance state and the output timing of the external circuit is designed to avoid output conflict.
2. Product quality might suffer if the absolute maximum rating is exceeded even momentarily for any parameter. That is, the absolute maximum ratings are the rated values at which the product is on the verge of suffering physical damage. Use the product so as to stay away from these rated values as much as possible. The ratings and conditions indicated for DC characteristics and AC characteristics represent the quality assurance range during normal operation.

37.2 Capacitance

(T_t = –40 to 100°C, IV_{DD} = $PLLV_{DD}$ = EV_{DD} = UV_{DD} = $OSCV_{DD}$ = DV_{DD} = AV_{DD} = V_{SS} = AV_{SS} = 0 V)

Item	Symbol	Conditions	MIN.	TYP	MAX.	Unit
Input capacitance	C_I	fc = 1 MHz Unmeasured pins returned to 0 V			15	pF
I/O capacitance	C_{IO}				15	pF
Output capacitance	C_O				15	pF

37.3 Operating Conditions

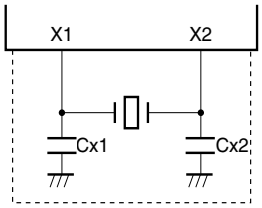
Internal Operating Frequency (f_{XX})	Package Surface Temperature (T_t)	Power Supply Voltage (IV_{DD}, $PLLV_{DD}$, EV_{DD}, UV_{DD}, DV_{DD}, $OSCV_{DD}$, AV_{DD})
144 MHz to 200 MHz	–40 to +100°C	IV_{DD} = $PLLV_{DD}$ = 1.2 V ± 0.1 V EV_{DD} = UV_{DD} = $OSCV_{DD}$ = DV_{DD} = 3.3 V ± 0.3 V AV_{DD} = 3.3 V ± 0.3 V or 5.0 V ± 0.5 V

37.4 Oscillator Characteristics

Item	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Oscillation frequency	f_X		7.2		10	MHz

37.4.1 Recommended resonator

(1) KYOCERA Crystal Device Corporation: Crystal resonator

Type	Circuit Example	Part Number		Oscillation Frequency (MHz)	Recommended Circuit Constant		Load Capacitance of crystal resonator (pF)
		For consumer applications	For industrial/ high-reliability applications		Cx1 (pF)	Cx2 (pF)	
Surface mounting		CX1255GB CX8045GB	CX1255GA CX8045GA	7.2 MHz	10 pF	10 pF	8 pF
		CX1255GB CX8045GB CX5032GB	CX1255GA CX8045GA CX5032SA CX5032GA	7.2 MHz	10 pF	10 pF	8 pF
		CX1255GB CX8045GB CX5032GB	CX1255GA CX8045GA CX5032SA CX5032GA	9.0 MHz	10 pF	10 pF	8 pF
		CX1255GB CX8045GB CX5032GB	CX1255GA CX8045GA CX5032SA CX5032GA	9.6 MHz	10 pF	10 pF	8 pF
		CX1255GB CX8045GB CX5032SB CX5032GB CX3225GB	CX1255GA CX8045GA CX5032SA CX5032GA CX3225SA	10.0 MHz	10 pF	10 pF	8 pF

37.5 DC Characteristics

37.5.1 Pin characteristics

(T_t = –40 to 100°C , IV_{DD} = $PLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = UV_{DD} = $OSCV_{DD}$ = DV_{DD} = 3.0 to 3.6 V, AV_{DD} = 3.0 to 3.6 V or AV_{DD} = 4.5 to 5.5 V, V_{SS} = AV_{SS} = $OSCV_{SS}$ = $PLLV_{SS}$ = DV_{SS} = 0 V)

Item	Symbol	Conditions		MIN.	TYP	MAX.	Unit
Input voltage, high	V_{IH1}	$\overline{RESET}$		1.7		EV_{DD} + 0.3	V
		$\overline{TRST}$		1.7		EV_{DD} + 0.3	V
		CMOS	a	0.7EV_{DD}		EV_{DD} + 0.3	V
		Schmitt 1	a	0.7EV_{DD}		EV_{DD} + 0.3	V
		Schmitt 2	a	0.8 EV_{DD}		EV_{DD} + 0.3	V
		LVTTL	a	2.0		EV_{DD} + 0.3	V
Input voltage, high (5-V tolerant pin[b])	V_{IH2}	CMOS	a	0.7EV_{DD}		5.5	V
		Schmitt 1	a	0.7EV_{DD}		5.5	V
		Schmitt 2	a	0.8EV_{DD}		5.5	V
Input voltage, high (port 14[c])	V_{IH3}			0.7AV_{DD}		AV_{DD} + 0.3	V
Input voltage, low	V_{IL1}	$\overline{RESET}$		–0.5		0.7	V
		$\overline{TRST}$		–0.5		0.7	V
		CMOS0	a	–0.5		0.3 EV_{DD}	V
		Schmitt 1	a	–0.5		0.3EV_{DD}	V
		Schmitt 2	a	–0.5		0.2EV_{DD}	V
		LVTTL	a	–0.5		0.8	V
Input voltage, low (5-V tolerant pin[b])	V_{IL2}	CMOS	a	–0.5		0.3EV_{DD}	V
		Schmitt 1	a	–0.5		0.3EV_{DD}	V
		Schmitt 2	a	–0.5		0.2EV_{DD}	V
Input voltage, low (port 14[c])	V_{IL3}			–0.5		0.3AV_{DD}	V
Schmitt trigger input hysteresis width	V_T^+- V_T^-		d	0.3	0.6		V

a) This is determined by the used pin function. For details, see *2.6 "Pin Input Circuit Types"*.

b) The 5-V tolerant pin is assigned to one of the following pin numbers: Y2, AA1, AA2, AB2.

c) The port 14 pin is assigned to one of the following pin numbers: V5, V6, V7, V8, W5, W6, W7, W8, AA5, AA8, AB5, AB8.

d) MNI, INTP0, INTP1, INTP2, INTP3, INTP4, UCLK, $\overline{OCI}$, RXD0, SI0, SCK0, RXD1, SI1, SCK1, RXD2, SI2, SCK2, RXD3, SI3, SCK3, RXD4, SI4, SCK4, RXD5, SI5, SCK5, RXD0F, SI0F, SCK0F, RXD1F, SI1F, SCK1F, RXD2F, SI2F, SCK2F, RXD3F, SI3F, SCK3F, FLSI, FLRXD, FLSCK

(T_t = –40 to 100°C , IV_{DD} = $PLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = UV_{DD} = $OSCV_{DD}$ = DV_{DD} = 3.0 to 3.6 V, AV_{DD} = 3.0 to 3.6 V or AV_{DD} = 4.5 to 5.5 V, V_{SS} = AV_{SS} = $OSCV_{SS}$ = $PLLV_{SS}$ = DV_{SS} = 0 V)

Item	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Output voltage, high	V_{OH1}	I_{OH} = –3 mA	EV_{DD} – 1.0			V
	V_{OH2}	I_{OH} = –100 *m*A	EV_{DD} – 0.5			V
Output voltage, low	V_{OL}	I_{OL} = 3 mA			0.4	V
Input leakage current, high	I_{LIH}	V_I = EV_{DD} = UV_{DD} = $OCSV_{DD}$ = DV_{DD} = AV_{DD}			10	μA
Input leakage current, low	I_{LIL}	V_I = 0 V			–10	μA
Analog pin input leakage current	I_{LIAN}				±10	μA
Internal pull-down resistance	R_L		20	40	100	kΩ
Internal pull-up resistance	R_H		20	40	100	kΩ

37.5.2 Power supply current

(1) μPD70F3510, 70F3512

(T_t = –40 to 100°C , IV_{DD} = $PLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = UV_{DD} = $OSCV_{DD}$ = DV_{DD} = 3.0 to 3.6 V, AV_{DD} = 3.0 to 3.6 V or AV_{DD} = 4.5 to 5.5 V, V_{SS} = AV_{SS} = $OSCV_{SS}$ = $PLLV_{SS}$ = DV_{SS} = 0 V)

Item	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Power supply current	I_{DD}	Normal operation	IV_{DD} + $PLLV_{DD}$ pin			680	mA
	EI_{DD}		EV_{DD} + $OSCV_{DD}$ + UV_{DD} + DV_{DD} pin			[a]	mA
	I_{DDH}	While in the HALT status	IV_{DD} + $PLLV_{DD}$ pin			490	mA

a) Keep the external power supply current at 160 mA or less, including the EV_{DD} current (40 mA to 70 mA) consumed internally.

Note Leave sufficient margins during power supply design.

(2) μPD70F3514, 70F3515

(T_t = –40 to 100°C , IV_{DD} = $PLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = UV_{DD} = $OSCV_{DD}$ = DV_{DD} = 3.0 to 3.6 V, AV_{DD} = 3.0 to 3.6 V or AV_{DD} = 4.5 to 5.5 V, V_{SS} = AV_{SS} = $OSCV_{SS}$ = $PLLV_{SS}$ = DV_{SS} = 0 V)

Item	Symbol	Conditions		MIN.	TYP.	MAX.	Unit
Power supply current	I_{DD}	Normal operation	IV_{DD} + $PLLV_{DD}$ pin			950	mA
	EI_{DD}		EV_{DD} + $OSCV_{DD}$ + UV_{DD} + DV_{DD} pin			[a]	mA
	I_{DDH}	While in the HALT status	IV_{DD} + $PLLV_{DD}$ pin			540	mA

a) Keep the external power supply current at 160 mA or less, including the EV_{DD} current (40 mA to 70 mA) consumed internally.

Note Leave sufficient margins during power supply design.

37.6 AC Characteristics

37.6.1 AC measurement points

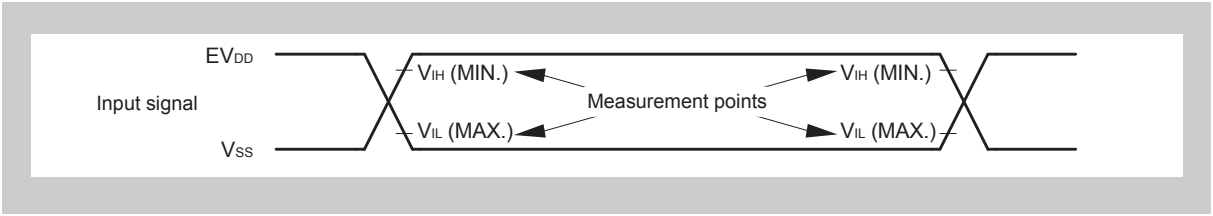


Figure 37-1 AC Test Input Measurement Points

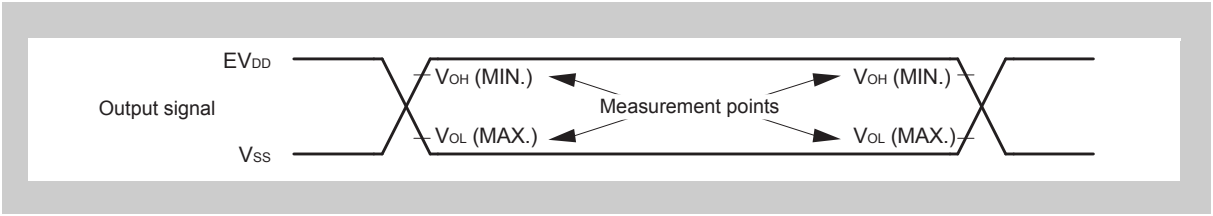


Figure 37-2 AC Test Output Measurement Points

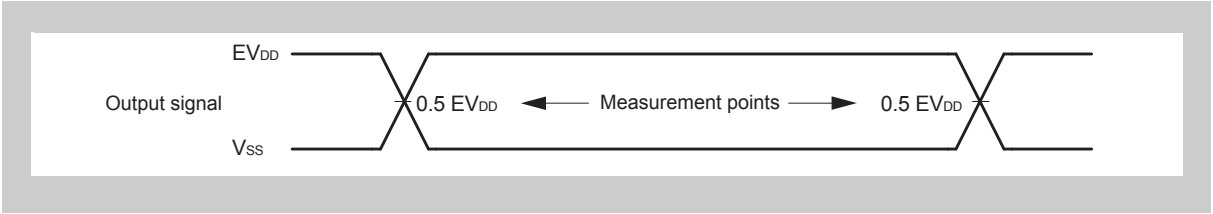


Figure 37-3 External Bus Access AC Test I/O Measurement Points

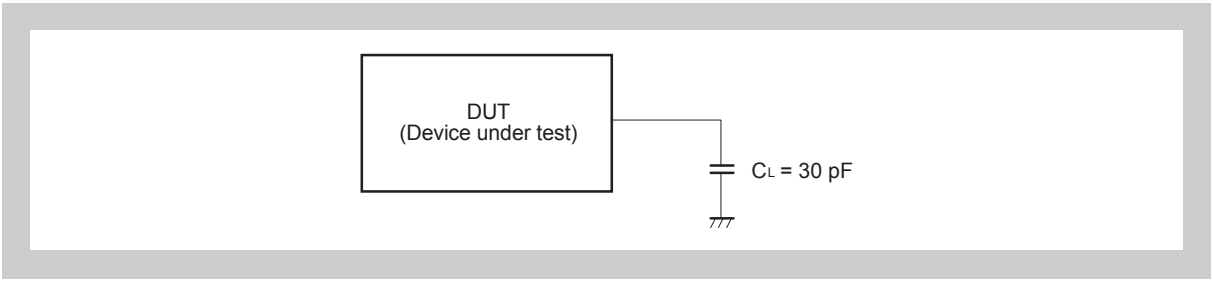


Figure 37-4 Load Condition

Caution If the load capacitance exceeds 30 pF due to the circuit configuration, bring the load condition of the device to 30 pF or less by inserting a buffer or by some other means.

37.6.2 External bus interface

(1) Primary memory controller, SRAM read access timing

(T_t = –40 to 100°C , IV_{DD} = $PLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = UV_{DD} = $OSCV_{DD}$ = DV_{DD} = 3.0 to 3.6 V, AV_{DD} = 3.0 to 3.6 V or AV_{DD} = 4.5 to 5.5 V, V_{SS} = AV_{SS} = $OSCV_{SS}$ = $PLLV_{SS}$ = DV_{SS} = 0 V)

Item	Symbol	MIN.	MAX.	Unit
Address, $\overline{\text{P\_CS[3:1]}}$, $\overline{\text{P\_xxBE}}$ output delay time (from P_BUSCLK↑)	t_{DKA}	1.5	14	ns
Address, $\overline{\text{P\_CS[3:1]}}$, $\overline{\text{P\_xxBE}}$ output hold time (to P_BUSCLK↑)	t_{HKA}	1.5	14	ns
$\overline{\text{P\_BCYST}}$↓ delay time (from P_BUSCLK↑)	t_{DKBSL}	1.5	14	ns
$\overline{\text{P\_BCYST}}$↑ delay time (from P_BUSCLK↑)	t_{DKBSH}	1.5	14	ns
$\overline{\text{P\_RD}}$↓ delay time (from P_BUSCLK↑)	t_{DKRDL}	1.5	14	ns
$\overline{\text{P\_RD}}$↑ delay time (from P_BUSCLK↑)	t_{DKRDH}	1.5	14	ns
Data output delay time (from P_BUSCLK↑)	t_{DKOD}	1.5	14	ns
Data float delay time (from P_BUSCLK↑)	t_{HKOD2}		14	ns
$\overline{\text{P\_WAIT}}$ setup time (to P_BUSCLK↑)	t_{SWK}	15 + 1.5f_{CLK}		ns
$\overline{\text{P\_WAIT}}$ hold time (from P_BUSCLK↑)	t_{HKW}	2 – 1.5f_{CLK}		ns
Data input setup time (to P_BUSCLK↑)	t_{SKID}	13		ns
Data input hold time (to P_BUSCLK↑)	t_{HKID}	0		ns
Data input setup time (to address, $\overline{\text{P\_CS[3:1]}}$, and $\overline{\text{P\_xxBE}}$)	t_{SAID}		(w + w_D + 2) T – 27	ns
Data input setup time (to $\overline{\text{P\_RD}}$↓)	t_{SRDID}		(w + w_D + 1) T – 27	ns
$\overline{\text{P\_RD}}$ low-level width	t_{WRDL}	(w + w_D + 1) T – 10		ns
$\overline{\text{P\_RD}}$↓ delay time from address, $\overline{\text{P\_CS[3:1]}}$, and $\overline{\text{P\_xxBE}}$	t_{DARD}	T – 10		ns
Address, $\overline{\text{P\_CS[3:1]}}$, and $\overline{\text{P\_xxBE}}$ delay time from $\overline{\text{P\_RD}}$↑	t_{DRDA}	–6		ns
Data input hold time (to $\overline{\text{P\_RD}}$↑)	t_{HRDID}	0		ns
Data output delay time from $\overline{\text{P\_RD}}$↑	t_{DRDOD}	iT – 14		ns

Notes
1. xx: LL, LU, UL, UU
2. w: Number of wait cycles inserted by $\overline{\text{P\_WAIT}}$
3. w_D: Number of wait cycles specified by the DWC0 register
4. T: t_{BCYK} (P_BUSCLK cycle)
5. i: Number of idle states
6. f_{CLK}: Internal system clock

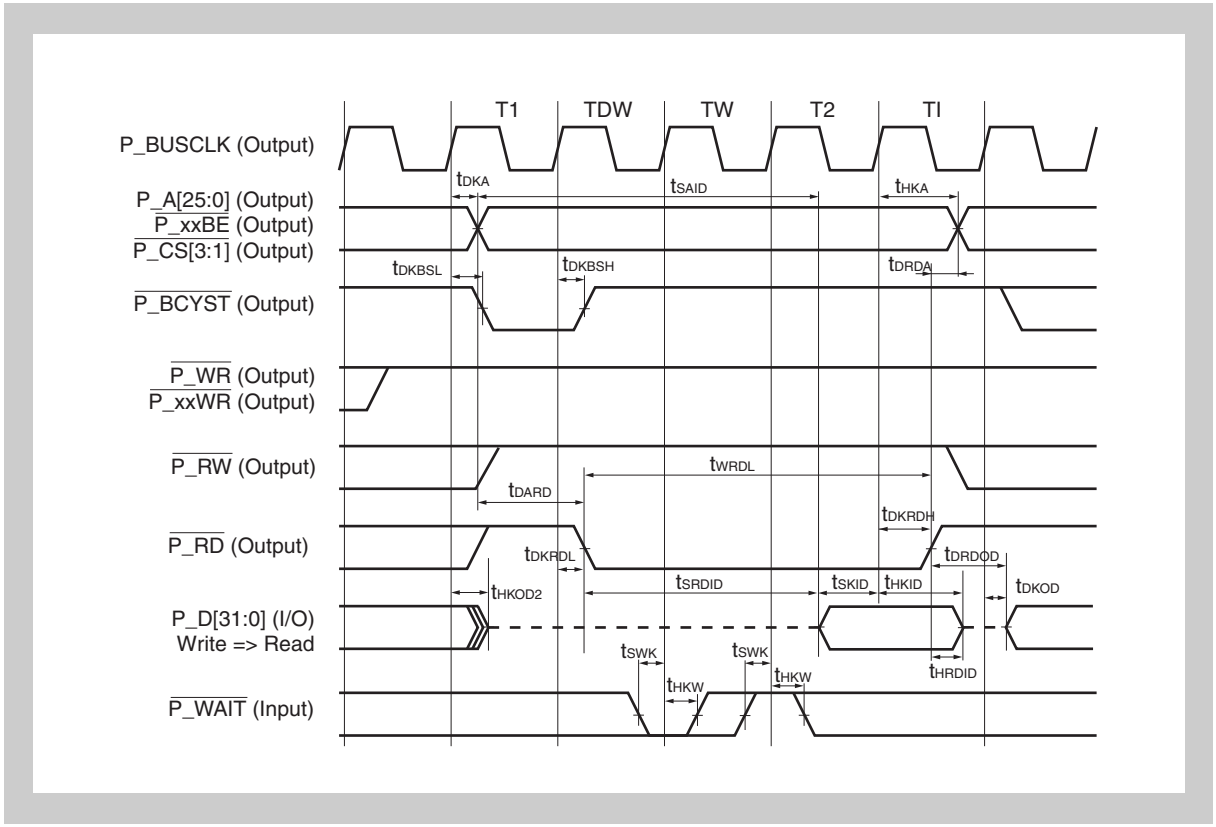


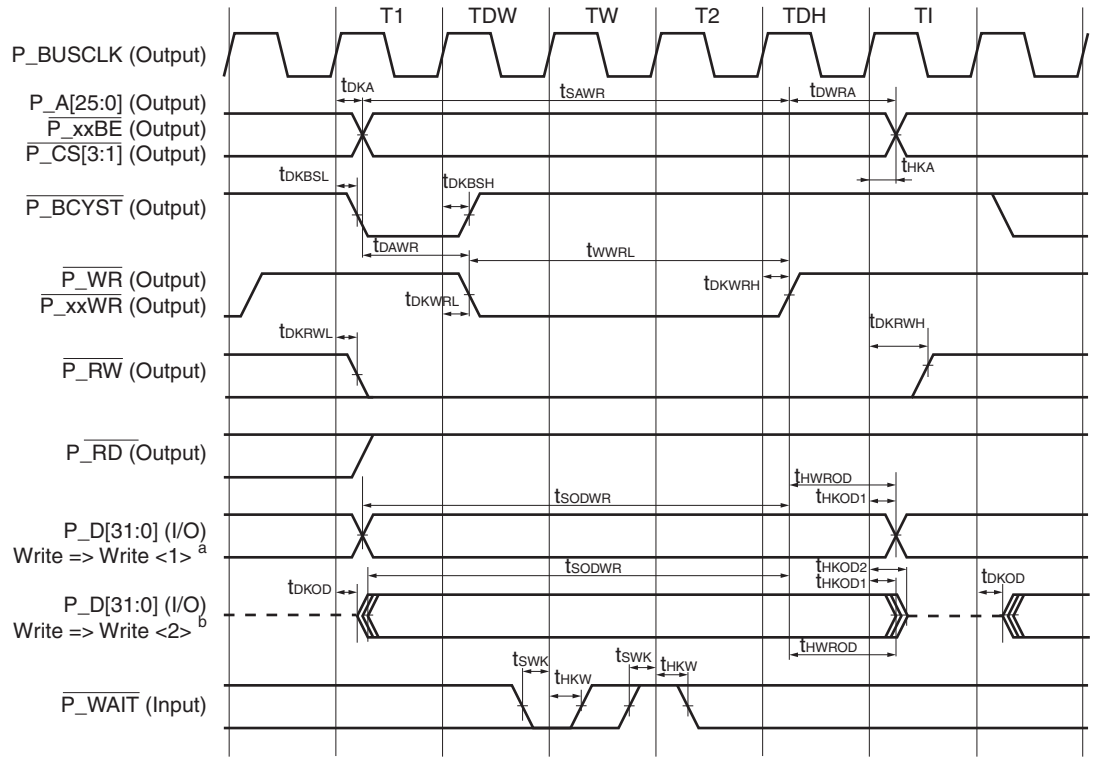
Figure 37-5 Primary Memory Controller, SRAM Read Access Timing

(2) Primary memory controller, SRAM write access timing

(T_t = –40 to 100°C , IV_{DD} = $PLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = UV_{DD} = $OSCV_{DD}$ = DV_{DD} = 3.0 to 3.6 V, AV_{DD} = 3.0 to 3.6 V or AV_{DD} = 4.5 to 5.5 V, V_{SS} = AV_{SS} = $OSCV_{SS}$ = $PLLV_{SS}$ = DV_{SS} = 0 V)

Item	Symbol		MIN.	MAX.	Unit
Address, $\overline{\text{P\_CS[3:1]}}$, and $\overline{\text{P\_xxBE}}$ delay time (from P_BUSCLK↑)		t_{DKA}	1.5	14	ns
Address, $\overline{\text{P\_CS[3:1]}}$, and $\overline{\text{P\_xxBE}}$ hold time (to P_BUSCLK↑)		t_{HKA}	1.5	14	ns
$\overline{\text{P\_BCYST}}$↓ delay time (from P_BUSCLK↑)		t_{DKBSL}	1.5	14	ns
$\overline{\text{P\_BCYST}}$↑ delay time (from P_BUSCLK↑)		t_{DKBSH}	1.5	14	ns
$\overline{\text{P\_xxWR}}$ and $\overline{\text{P\_WR}}$↓ delay time (from P_BUSCLK↑)		t_{DKWRL}	1.5	14	ns
$\overline{\text{P\_xxWR}}$ and $\overline{\text{P\_WR}}$↑ delay time (from P_BUSCLK↑)		t_{DKWRH}	1.5	14	ns
$\overline{\text{P\_RW}}$↓ delay time (from P_BUSCLK↑)		t_{DKRWL}	1.5	14	ns
$\overline{\text{P\_RW}}$↑ delay time (from P_BUSCLK↑)		t_{DKRWH}	1.5	14	ns
Data output delay time (from P_BUSCLK↑)		t_{DKOD}	1.5	14	ns
Data output hold time (from P_BUSCLK↑)		t_{HKOD1}	1.5		ns
Data float delay time (from P_BUSCLK↑)		t_{HKOD2}		14	ns
$\overline{\text{P\_WAIT}}$ setup time (to P_BUSCLK↑)		t_{SWK}	$15 + 1.5f_{CLK}$		ns
$\overline{\text{P\_WAIT}}$ hold time (from P_BUSCLK↑)		t_{HKW}	$2 – 1.5f_{CLK}$		ns
Address, $\overline{\text{P\_CS[3:1]}}$, and $\overline{\text{P\_xxBE}}$ delay time (from $\overline{\text{P\_xxWR}}$, $\overline{\text{P\_WR}}$↓)		t_{DAWR}	T – 10		ns
Address, $\overline{\text{P\_CS[3:1]}}$, and $\overline{\text{P\_xxBE}}$ setup time (to $\overline{\text{P\_xxWR}}$, $\overline{\text{P\_WR}}$↑)		t_{SAWR}	$(w + w_D + 2)$ T – 10		ns
Address, $\overline{\text{P\_CS[3:1]}}$, $\overline{\text{P\_xxBE}}$ delay time from $\overline{\text{P\_xxWR}}$ and $\overline{\text{P\_WR}}$↑		t_{DWRA}	w_{DH}T – 10		ns
$\overline{\text{P\_xxWR}}$, $\overline{\text{P\_WR}}$ low level width		t_{WWRL}	$(w + w_D + 1)$ T – 10		ns
Data output setup time (to $\overline{\text{P\_xxWR}}$ and $\overline{\text{P\_WR}}$↑)		t_{SODWR}	$(w + w_D + 2)$ T – 10		ns
Data output hold time (from $\overline{\text{P\_xxWR}}$ and $\overline{\text{P\_WR}}$↑)		t_{HWROD}	w_{DH}T – 10		ns

Notes
1. xx: LL, LU, UL, UU
2. w: Number of wait cycles inserted by $\overline{\text{P\_WAIT}}$
3. w_D: Number of wait cycles specified by the DWC0 register
4. w_{DH}: Number of wait cycles specified by the DHC register
5. T: t_{BCYK} (P_BUSCLK cycle)
6. i: Number of idle states
7. f_{CLK}: Internal system clock



a) During continuous write-to-write operation, when no TI cycle is inserted

b) During continuous write-to-write operation, when a TI cycle is inserted

Figure 37-6 Primary Memory Controller, SRAM Write Access Timing

(3) Primary memory controller, SDRAM read timing

(T_t = –40 to 100°C , IV_{DD} = $PLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = UV_{DD} = $OSCV_{DD}$ = DV_{DD} = 3.0 to 3.6 V, AV_{DD} = 3.0 to 3.6 V or AV_{DD} = 4.5 to 5.5 V, V_{SS} = AV_{SS} = $OSCV_{SS}$ = $PLLV_{SS}$ = DV_{SS} = 0 V)

Item	Symbol		MIN.	MAX.	Unit
Address output delay time (from P_BUSCLK↑)		t_{DKA}	1.5	14	ns
Address output hold time (from P_BUSCLK↑)		t_{HKA}	1.5	14	ns
$\overline{\text{P\_CS4}}$ delay time (from P_BUSCLK↑)		t_{DKCS}	1.5	14	ns
$\overline{\text{P\_SDRAS}}$ delay time (from P_BUSCLK↑)		t_{DKRAS}	1.5	14	ns
$\overline{\text{P\_SDCAS}}$ delay time (from P_BUSCLK↑)		t_{DKCAS}	1.5	14	ns
P_xxDQM delay time (from P_BUSCLK↑)		t_{DKDQM}	1.5	14	ns
P_SDCKE delay time (from P_BUSCLK↑)		t_{DKCKE}	1.5	14	ns
$\overline{\text{P\_WE}}$ delay time (from P_BUSCLK↑)		t_{DKWE}	1.5	14	ns
Data input setup time (SDRAM read time, to P_BUSCLK↑)		t_{SDRMK}	13		ns
Data input hold time (SDRAM read time, to P_BUSCLK↑)		t_{HKDRM}	0		ns
Data output delay from P_BUSCLK↑		t_{DSDOD}	(1 + i) T		ns

Notes
1. xx: LL, LU, UL, UU
2. T: t_{BCYK} (P_BUSCLK cycle)
3. i: Number of idle states

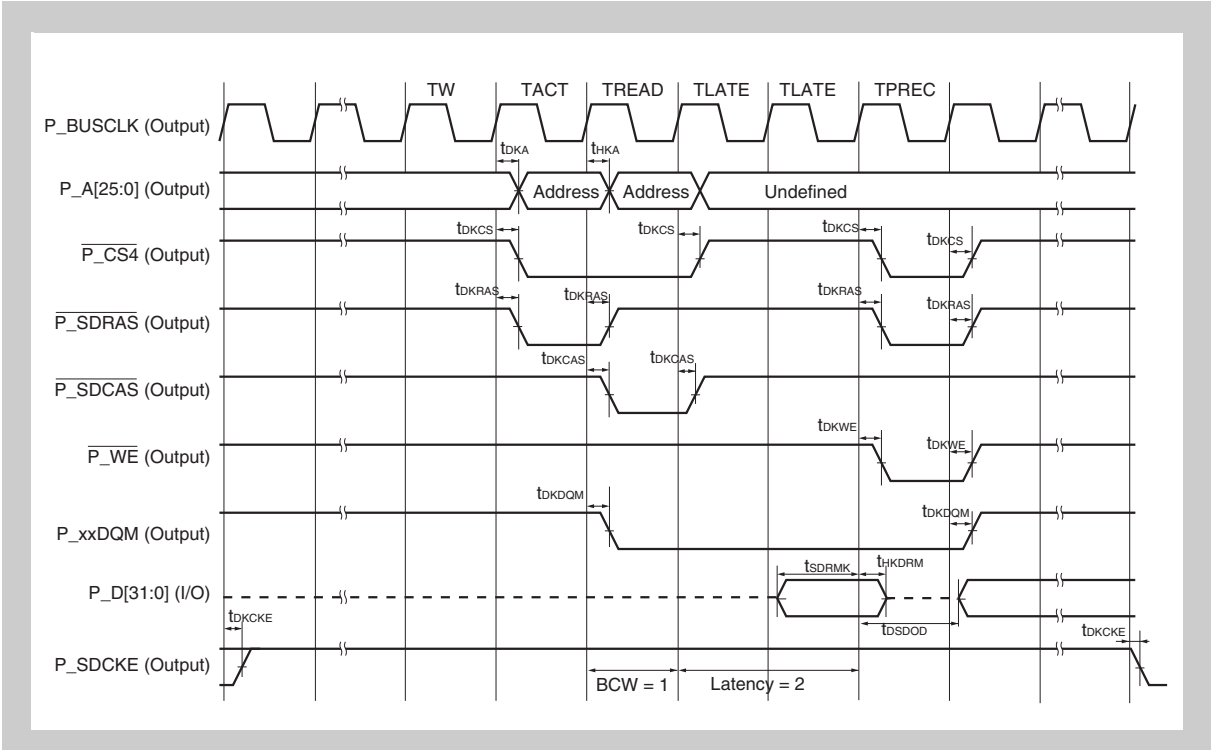


Figure 37-7 Primary Memory Controller, SDRAM Read Timing

(4) Primary memory controller, SDRAM write timing

(T_t = –40 to 100°C , IV_{DD} = $PLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = UV_{DD} = $OSCV_{DD}$ = DV_{DD} = 3.0 to 3.6 V, AV_{DD} = 3.0 to 3.6 V or AV_{DD} = 4.5 to 5.5 V, V_{SS} = AV_{SS} = $OSCV_{SS}$ = $PLLV_{SS}$ = DV_{SS} = 0 V)

Item	Symbol		MIN.	MAX.	Unit
Address output delay time (from P_BUSCLK↑)		t_{DKA}	1.5	14	ns
Address output hold time (from P_BUSCLK↑)		t_{HKA}	1.5	14	ns
$\overline{\text{P\_CS4}}$ delay time (from P_BUSCLK↑)		t_{DKCS}	1.5	14	ns
$\overline{\text{P\_SDRAS}}$ delay time (from P_BUSCLK↑)		t_{DKRAS}	1.5	14	ns
$\overline{\text{P\_SDCAS}}$ delay time (from P_BUSCLK↑)		t_{DKCAS}	1.5	14	ns
P_xxDQM delay time (from P_BUSCLK↑)		t_{DKDQM}	1.5	14	ns
P_SDCKE delay time (from P_BUSCLK↑)		t_{DKCKE}	1.5	14	ns
$\overline{\text{P\_WE}}$ delay time (from P_BUSCLK↑)		t_{DKWE}	1.5	14	ns
Data output delay time (from P_BUSCLK↑)		t_{DKDT}	1.5	14	ns
Data output hold time (from P_BUSCLK↑)		t_{HZKDT1}	1.5		ns
Data float delay time (from P_BUSCLK↑)		t_{HZKDT2}		14	ns

Note xx: LL, LU, UL, UU

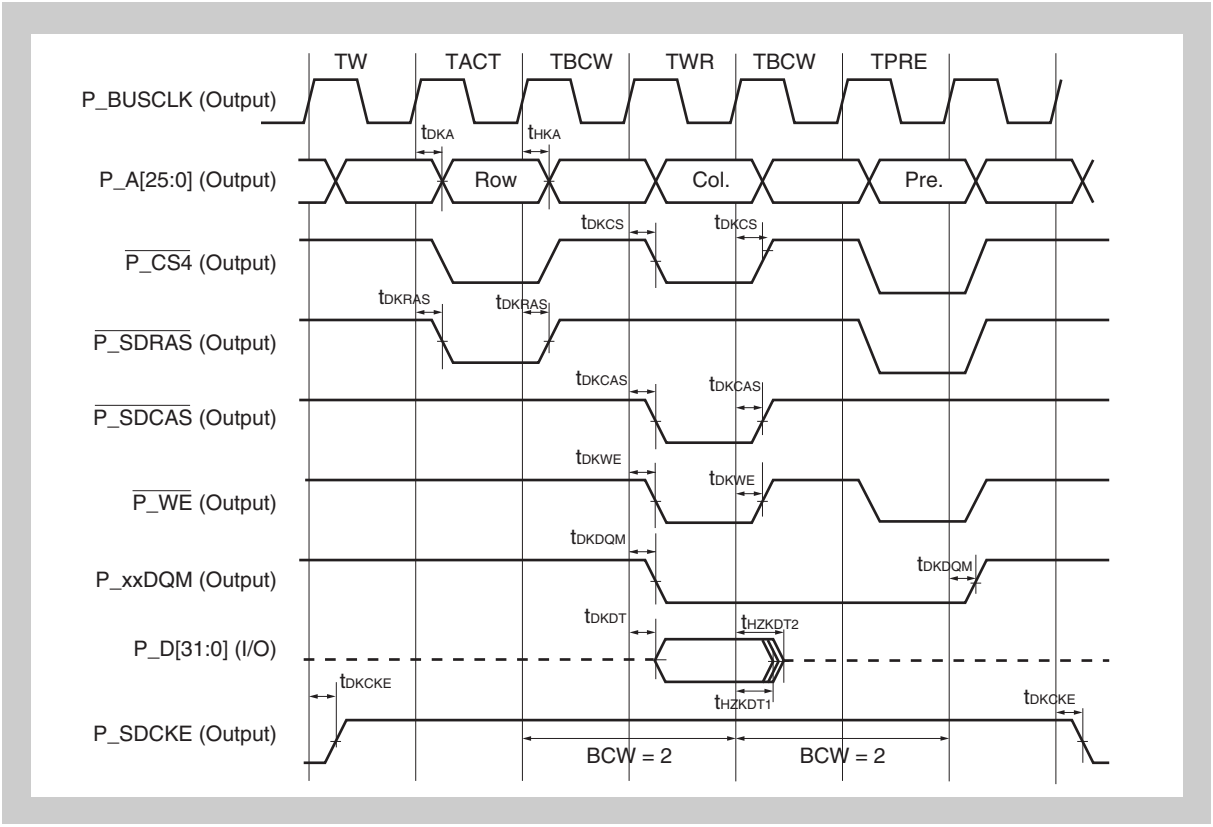


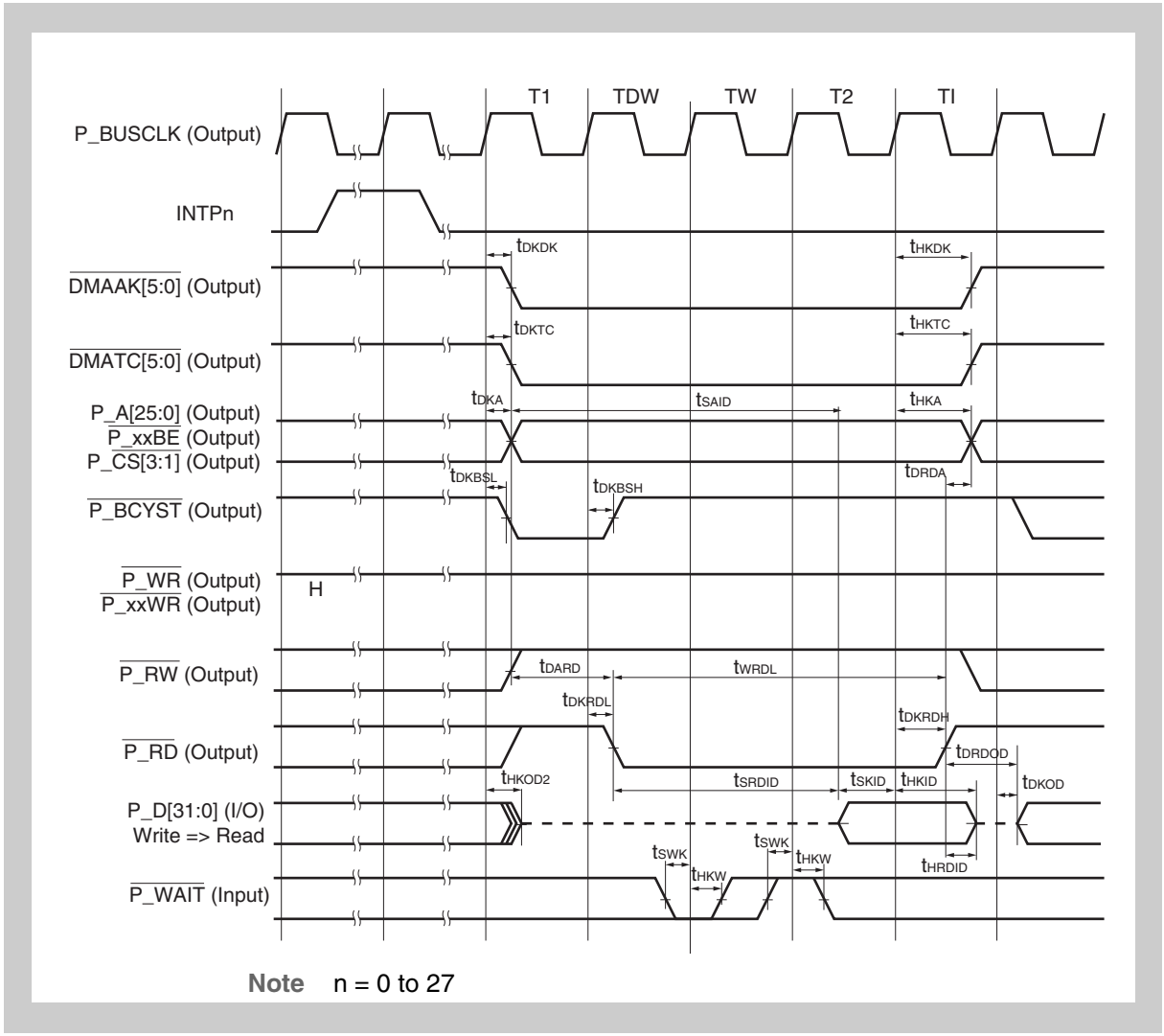
Figure 37-8 Primary Memory Controller, SDRAM Write Timing

(5) Primary memory controller, SRAM DMA read access timing

(T_t = –40 to 100°C , IV_{DD} = $PLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = UV_{DD} = $OSCV_{DD}$ = DV_{DD} = 3.0 to 3.6 V, AV_{DD} = 3.0 to 3.6 V or AV_{DD} = 4.5 to 5.5 V, V_{SS} = AV_{SS} = $OSCV_{SS}$ = $PLLV_{SS}$ = DV_{SS} = 0 V)

Item	Symbol	MIN.	MAX.	Unit
$\overline{\text{DMAAK[5:0]}}$ output delay time (from P_BUSCLK↑)	t_{DKDK}	1.5	14	ns
$\overline{\text{DMAAK[5:0]}}$ output hold time (from P_BUSCLK↑)	t_{HKDK}	1.5	14	ns
$\overline{\text{DMATC[5:0]}}$ output delay time (from P_BUSCLK↑)	t_{DKTC}	1.5	14	ns
$\overline{\text{DMATC[5:0]}}$ output hold time (from P_BUSCLK↑)	t_{HKTC}	1.5	14	ns
Address, $\overline{\text{P\_CS[3:1]}}$, and $\overline{\text{P\_xxBE}}$ delay time (from P_BUSCLK↑)	t_{DKA}	1.5	14	ns
Address, $\overline{\text{P\_CS[3:1]}}$, and $\overline{\text{P\_xxBE}}$ hold time (from P_BUSCLK↑)	t_{HKA}	1.5	14	ns
$\overline{\text{P\_BCYST}}$↓ delay time (from P_BUSCLK↑)	t_{DKBSL}	1.5	14	ns
$\overline{\text{P\_BCYST}}$↑ delay time (from P_BUSCLK↑)	t_{DKBSH}	1.5	14	ns
Data output delay time (from P_BUSCLK↑)	t_{DKOD}	1.5	14	ns
Data float delay time (from P_BUSCLK↑)	t_{HKOD2}		14	ns
$\overline{\text{P\_WAIT}}$ setup time (to P_BUSCLK↑)	t_{SWK}	15 + 1.5f_{CLK}		ns
$\overline{\text{P\_WAIT}}$ hold time (from P_BUSCLK↑)	t_{HKW}	2 – 1.5f_{CLK}		ns
$\overline{\text{P\_RD}}$↓ delay time (from P_BUSCLK↑)	t_{DKRDL}	1.5	14	ns
$\overline{\text{P\_RD}}$↑ delay time (from P_BUSCLK↑)	t_{DKRDH}	1.5	14	ns
Data input setup time (to P_BUSCLK↑)	t_{SKID}	13		ns
Data input hold time (from P_BUSCLK↑)	t_{HKID}	0		ns
Data input setup time (to address, $\overline{\text{P\_CS[3:1]}}$, $\overline{\text{P\_xxBE}}$, $\overline{\text{P\_DMATC[5:0]}}$, and $\overline{\text{P\_DMAAK[5:0]}}$)	t_{SAID}		(w + w_D + 2) T – 27	ns
Data input setup time (to $\overline{\text{P\_RD}}$↓)	t_{SRDID}		(w + w_D + 1) T – 27	ns
$\overline{\text{P\_RD}}$ low-level width	t_{WRDL}	(w + w_D + 1) T – 10		ns
$\overline{\text{P\_RD}}$↓ delay time from address, $\overline{\text{P\_CS[3:1]}}$, $\overline{\text{P\_xxBE}}$, $\overline{\text{P\_DMATC[5:0]}}$, and $\overline{\text{P\_DMAAK[5:0]}}$	t_{DARD}	T – 10		ns
Address, $\overline{\text{P\_CS[3:1]}}$, $\overline{\text{P\_xxBE}}$, $\overline{\text{P\_DMATC[5:0]}}$, and $\overline{\text{DMAAK[5:0]}}$ delay time from $\overline{\text{P\_RD}}$↑	t_{DRDA}	–6		ns
Data input hold time (from $\overline{\text{P\_RD}}$↑)	t_{HRDID}	0		ns
Data output delay time from $\overline{\text{P\_RD}}$↑	t_{DRDOD}	iT – 14		ns

Notes
1. xx: LL, LU, UL, UU
2. w: Number of wait cycles inserted by $\overline{\text{P\_WAIT}}$
3. w_D: Number of wait cycles specified by the DWC0 register
4. T: t_{BCYK} (P_BUSCLK cycle)
5. i: Number of idle states
6. f_{CLK}: Internal system clock



Note n = 0 to 27

Figure 37-9 Primary Memory Controller, DMA Read Access Timing

(6) Primary memory controller, DMA write access timing (without a TI cycle)

(T_t = –40 to 100°C , IV_{DD} = $PLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = UV_{DD} = $OSCV_{DD}$ = DV_{DD} = 3.0 to 3.6 V, AV_{DD} = 3.0 to 3.6 V or AV_{DD} = 4.5 to 5.5 V, V_{SS} = AV_{SS} = $OSCV_{SS}$ = $PLLV_{SS}$ = DV_{SS} = 0 V)

Item	Symbol	MIN.	MAX.	Unit
$\overline{\text{DMAAK[5:0]}}$ output delay time (from P_BUSCLK↑)	t_{DKDK}	1.5	14	ns
$\overline{\text{DMAAK[5:0]}}$ output hold time (from P_BUSCLK↑)	t_{HKDK}	1.5	14	ns
$\overline{\text{DMATC[5:0]}}$ output delay time (from P_BUSCLK↑)	t_{DKTC}	1.5	14	ns
$\overline{\text{DMATC[5:0]}}$ output hold time (from P_BUSCLK↑)	t_{HKTC}	1.5	14	ns
Address, $\overline{\text{P\_CS[3:1]}}$, and $\overline{\text{P\_xxBE}}$ delay time (from P_BUSCLK↑)	t_{DKA}	1.5	14	ns
Address, $\overline{\text{P\_CS[3:1]}}$, and $\overline{\text{P\_xxBE}}$ hold time (from P_BUSCLK↑)	t_{HKA}	1.5	14	ns
$\overline{\text{P\_BCYST}}$↓ delay time (from P_BUSCLK↑)	t_{DKBSL}	1.5	14	ns
$\overline{\text{P\_BCYST}}$↑ delay time (from P_BUSCLK↑)	t_{DKBSH}	1.5	14	ns
$\overline{\text{P\_WR}}$↓ delay time (from P_BUSCLK↑)	t_{DKWRL}	1.5	14	ns
$\overline{\text{P\_WR}}$↑ delay time (from P_BUSCLK↑)	t_{DKWRH}	1.5	14	ns
$\overline{\text{P\_RW}}$↓ delay time (from P_BUSCLK↑)	t_{DKRWL}	1.5	14	ns
$\overline{\text{P\_RW}}$↑ delay time (from P_BUSCLK↑)	t_{DKRWH}	1.5	14	ns
Data output delay time (from P_BUSCLK↑)	t_{DKOD}	1.5	14	ns
Data output hold time (from P_BUSCLK↑)	t_{HKOD1}	1.5		ns
$\overline{\text{P\_WAIT}}$ setup time (to P_BUSCLK↑)	t_{SWK}	15 + 1.5f_{CLK}		ns
$\overline{\text{P\_WAIT}}$ hold time (from P_BUSCLK↑)	t_{HKW}	2 – 1.5f_{CLK}		ns
Address, $\overline{\text{P\_CS[3:1]}}$, $\overline{\text{P\_xxBE}}$, $\overline{\text{P\_DMATC}}$, and $\overline{\text{P\_DMAAK}}$ delay time (from $\overline{\text{P\_xxWR}}$ and $\overline{\text{P\_WR}}$ ↓)	t_{DAWR}	T – 10		ns
Address, $\overline{\text{P\_CS[3:1]}}$, $\overline{\text{P\_xxBE}}$, $\overline{\text{P\_DMATC}}$, and $\overline{\text{P\_DMAAK}}$ setup time (to $\overline{\text{P\_xxWR}}$ and $\overline{\text{P\_WR}}$↓)	t_{SAWR}	(w + w_D + 2) T – 10		ns
Address and $\overline{\text{P\_CS[3:1]}}$ delay time from $\overline{\text{P\_xxWR}}$, $\overline{\text{P\_WR}}$↑	t_{DWRA}	w_{DH}T – 10		ns
$\overline{\text{P\_xxWR}}$ and $\overline{\text{P\_WR}}$ low level width	t_{WWRL}	(w + w_D + 1) T – 10		ns
Data output setup time (to $\overline{\text{P\_xxWR}}$ and $\overline{\text{P\_WR}}$↑)	t_{SODWR}	(w + w_D + 2) T – 10		ns
Data output hold time (from $\overline{\text{P\_xxWR}}$ and $\overline{\text{P\_WR}}$↑)	t_{HWROD}	w_{DH}T – 10		ns

Notes
1. xx: LL, LU, UL, UU
2. w: Number of wait cycles inserted by $\overline{\text{P\_WAIT}}$
3. w_D: Number of wait cycles specified by the DWC0 register
4. w_{DH}: Number of wait cycles specified by the DHC register
5. T: t_{BCYK} (P_BUSCLK cycle)
6. f_{CLK}: Internal system clock

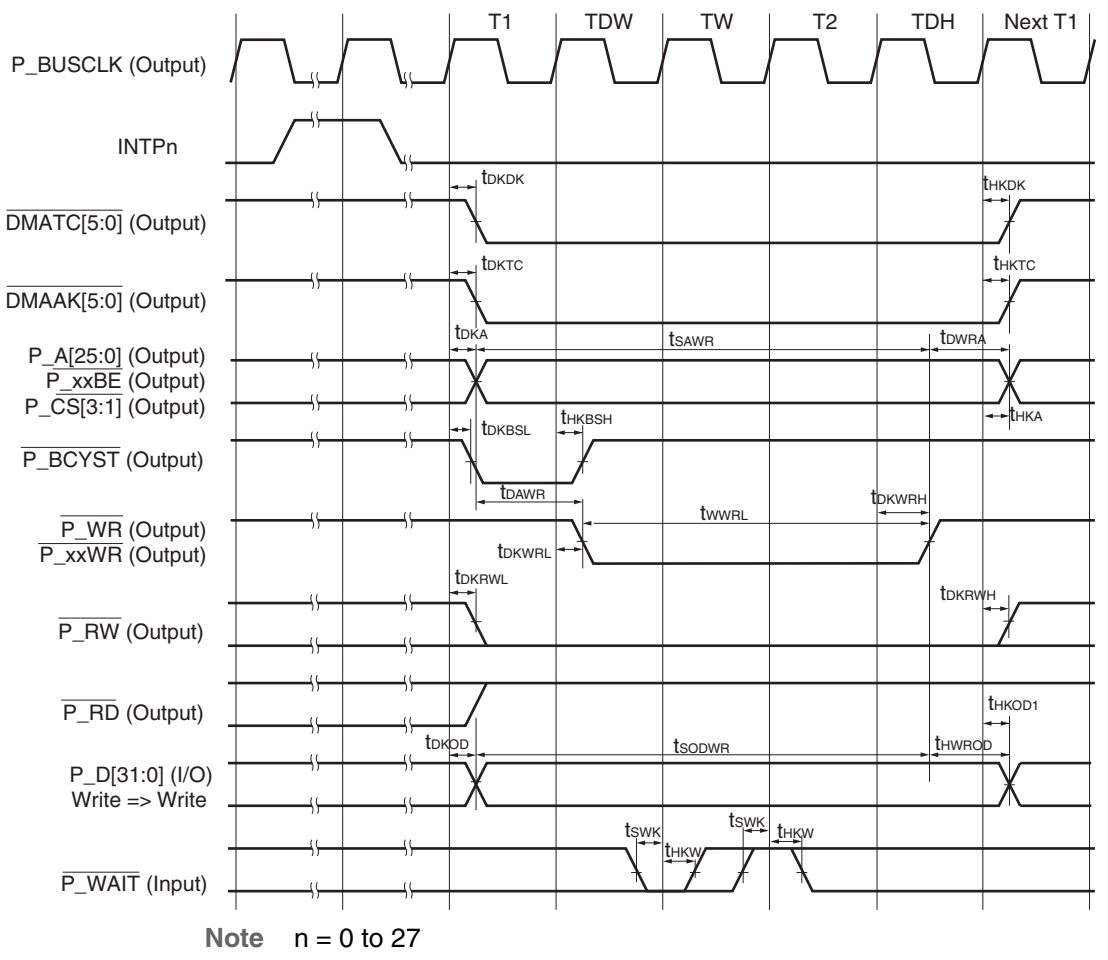


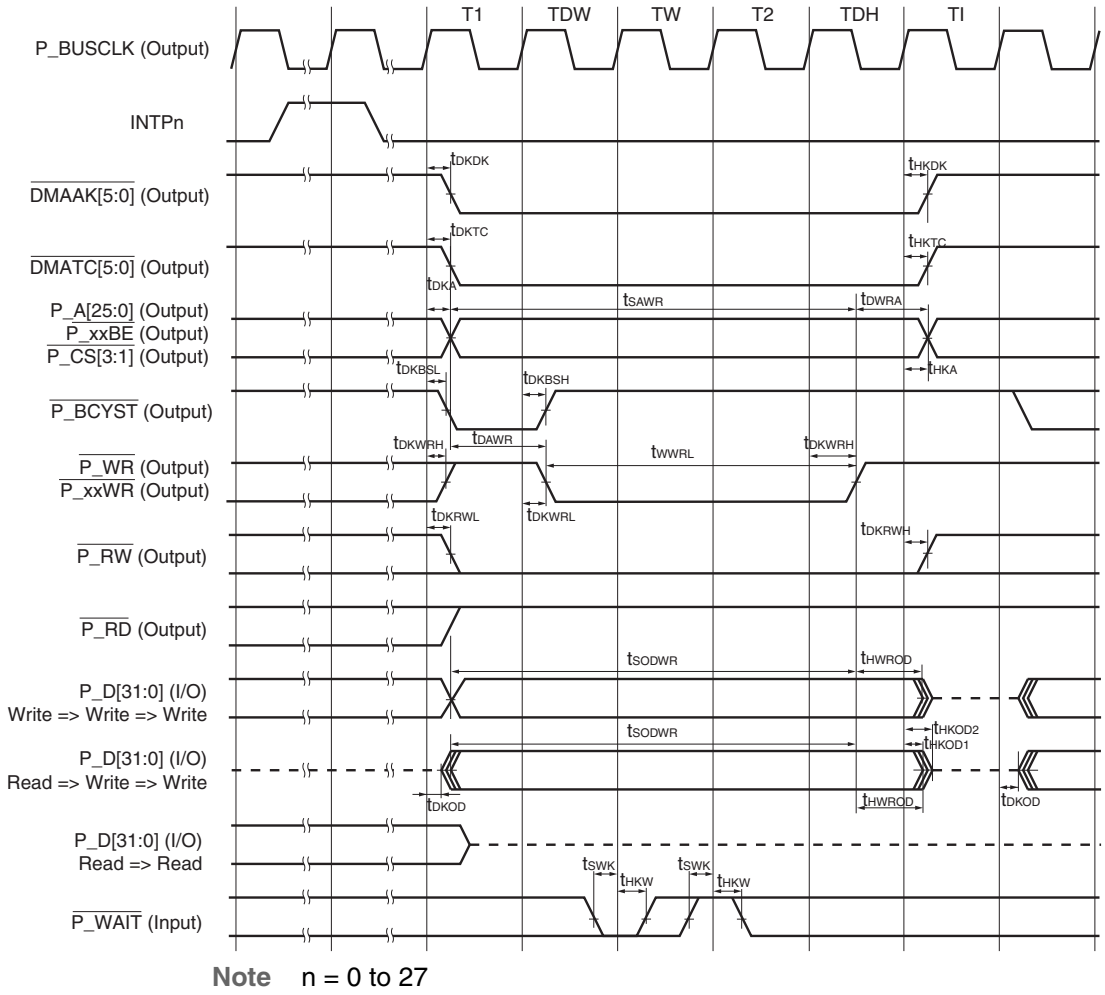
Figure 37-10 Primary Memory Controller, SRAM DMA Write Access Timing (Without a TI Cycle)

(7) Primary memory controller, SRAM DMA write access timing (with a TI cycle)

(T_t = –40 to 100°C , IV_{DD} = $PLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = UV_{DD} = $OSCV_{DD}$ = DV_{DD} = 3.0 to 3.6 V, AV_{DD} = 3.0 to 3.6 V or AV_{DD} = 4.5 to 5.5 V, V_{SS} = AV_{SS} = $OSCV_{SS}$ = $PLLV_{SS}$ = DV_{SS} = 0 V)

Item	Symbol		MIN.	MAX.	Unit
$\overline{\text{P\_DMAAK[5:0]}}$ output delay time (from P_BUSCLK↑)		t_{DKDK}	1.5	14	ns
$\overline{\text{P\_DMAAK[5:0]}}$ output hold time (from P_BUSCLK↑)		t_{HKDK}	1.5	14	ns
$\overline{\text{P\_DMATC[5:0]}}$ output delay time (from P_BUSCLK↑)		t_{DKTC}	1.5	14	ns
$\overline{\text{P\_DMATC[5:0]}}$ output hold time (from P_BUSCLK↑)		t_{HKTC}	1.5	14	ns
Address, $\overline{\text{P\_CS[3:1]}}$, and $\overline{\text{P\_xxBE}}$ output delay time (from P_BUSCLK↑)		t_{DKA}	1.5	14	ns
Address, $\overline{\text{P\_CS[3:1]}}$, $\overline{\text{P\_xxBE}}$ output hold time (from P_BUSCLK↑)		t_{HKA}	1.5	14	ns
$\overline{\text{P\_BCYST}}$↓ delay time (from P_BUSCLK↑)		t_{DKBSL}	1.5	14	ns
$\overline{\text{P\_BCYST}}$↑ delay time (from P_BUSCLK↑)		t_{DKBSH}	1.5	14	ns
$\overline{\text{P\_WR}}$↓ delay time (from P_BUSCLK↑)		t_{DKWRL}	1.5	14	ns
$\overline{\text{P\_WR}}$↑ delay time (from P_BUSCLK↑)		t_{DKWRH}	1.5	14	ns
$\overline{\text{P\_RW}}$↓ delay time (from P_BUSCLK↑)		t_{DKRWL}	1.5	14	ns
$\overline{\text{P\_RW}}$↑ delay time (from P_BUSCLK↑)		t_{DKRWH}	1.5	14	ns
Data output delay time (from P_BUSCLK↑)		t_{DKOD}	1.5	14	ns
Data output hold time (from P_BUSCLK↑)		t_{HKOD1}	1.5		ns
Data float delay time (from P_BUSCLK↑)		t_{HKOD2}		14	ns
$\overline{\text{P\_WAIT}}$ setup time (to P_BUSCLK↑)		t_{SWK}	15 + 1.5f_{CLK}		ns
$\overline{\text{P\_WAIT}}$ hold time (from P_BUSCLK↑)		t_{HKW}	2 – 1.5f_{CLK}		ns
Address, $\overline{\text{P\_CS[3:1]}}$, $\overline{\text{P\_xxBE}}$, $\overline{\text{P\_DMATC}}$, and $\overline{\text{P\_DMAAK}}$ delay time (from $\overline{\text{P\_xxWR}}$ and $\overline{\text{P\_WR}}$ ↓)		t_{DAWR}	T – 10		ns
Address, $\overline{\text{P\_CS[3:1]}}$, $\overline{\text{P\_xxBE}}$, $\overline{\text{P\_DMATC}}$, and $\overline{\text{P\_DMAAK}}$ setup time (to $\overline{\text{P\_xxWR}}$ and $\overline{\text{P\_WR}}$↓)		t_{SAWR}	(w + w_D + 2) T – 10		ns
Address and $\overline{\text{P\_CS[3:1]}}$ delay time from $\overline{\text{P\_xxWR}}$, $\overline{\text{P\_WR}}$↑		t_{DWRA}	w_{DH}T – 10		ns
$\overline{\text{P\_xxWR}}$ and $\overline{\text{P\_WR}}$ low level width		t_{WWRL}	(w + w_D + 1) T – 10		ns
Data output setup time (to $\overline{\text{P\_xxWR}}$ and $\overline{\text{P\_WR}}$↑)		t_{SODWR}	(w + w_D + 2) T – 10		ns
Data output hold time (from $\overline{\text{P\_xxWR}}$ and $\overline{\text{P\_WR}}$↑)		t_{HWROD}	w_{DH}T – 10		ns

Notes
1. xx: LL, LU, UL, UU
2. w: Number of wait cycles inserted by $\overline{\text{P\_WAIT}}$
3. w_D: Number of wait cycles specified by the DWC0 register
4. w_{DH}: Number of wait cycles specified by the DHC register
5. T: t_{BCYK} (P_BUSCLK cycle)
6. f_{CLK}: Internal system clock



Note n = 0 to 27

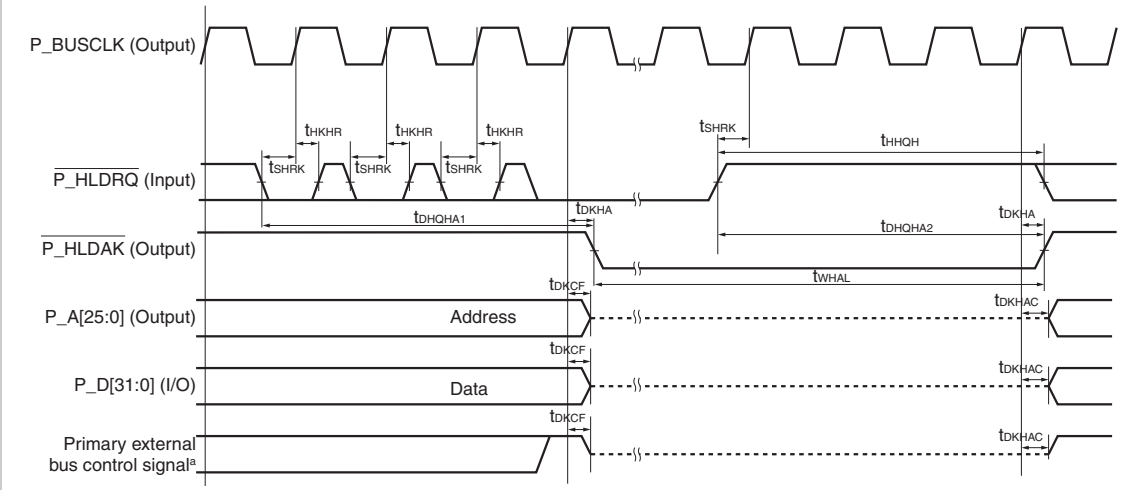
Figure 37-11 Primary Memory Controller, SRAM DMA Write Access Timing (With a TI Cycle)

(8) Primary memory controller, bus hold timing

(T_t = –40 to 100°C , IV_{DD} = $PLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = UV_{DD} = $OSCV_{DD}$ = DV_{DD} = 3.0 to 3.6 V, AV_{DD} = 3.0 to 3.6 V or AV_{DD} = 4.5 to 5.5 V, V_{SS} = AV_{SS} = $OSCV_{SS}$ = $PLLV_{SS}$ = DV_{SS} = 0 V)

Item	Symbol	MIN.	MAX.	Unit
$\overline{\text{P\_HLDRQ}}$ setup time (until P_BUSCLK↑)	t_{SHRK}	15 + 1.5f_{CLK}		ns
$\overline{\text{P\_HLDRQ}}$ hold time (from P_BUSCLK↑)	t_{HKHR}	2 – 1.5f_{CLK}		ns
$\overline{\text{P\_HLDAK}}$ delay time (from P_BUSCLK↑)	t_{DKHA}	1.5	14	ns
$\overline{\text{P\_HLDRQ}}$ high-level hold time	t_{HHQH}	to $\overline{\text{P\_HLDAK}}$↑		ns
$\overline{\text{P\_HLDAK}}$ low-level width	t_{WHAL}	T – 10		ns
Bus float delay time (from P_BUSCLK↑)	t_{DKCF}		14	ns
Bus output delay time (from P_BUSCLK↑)	t_{DKHAC}	1.5	14	ns
$\overline{\text{P\_HLDAK}}$↓ delay time from $\overline{\text{P\_HLDRQ}}$↓	t_{DHQHA1}	2T + 1.5f_{CLK}		ns
$\overline{\text{P\_HLDAK}}$↓ delay time from $\overline{\text{P\_HLDRQ}}$↑	t_{DHQHA2}	1.5f_{CLK}		ns

Note f_{CLK}: Internal system clock



a) Primary external bus control signals: $\overline{\text{P\_RW}}$, $\overline{\text{P\_xxBE}}$, $\overline{\text{P\_RD}}$, $\overline{\text{P\_xxWR}}$, $\overline{\text{P\_WR}}$, $\overline{\text{P\_CS[4:1]}}$, $\overline{\text{P\_BCYST}}$, $\overline{\text{P\_SDRAS}}$, $\overline{\text{P\_SDCAS}}$, P_SDCKE, $\overline{\text{P\_SDWE}}$, P_xxDQM

Figure 37-12 Primary Memory Controller, Bus Hold Timing

(9) Secondary memory controller, SRAM read access timing

(T_t = –40 to 100°C , IV_{DD} = $PLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = UV_{DD} = $OSCV_{DD}$ = DV_{DD} = 3.0 to 3.6 V, AV_{DD} = 3.0 to 3.6 V or AV_{DD} = 4.5 to 5.5 V, V_{SS} = AV_{SS} = $OSCV_{SS}$ = $PLLV_{SS}$ = DV_{SS} = 0 V)

Item	Symbol		MIN.	MAX.	Unit
Address output delay time (from S_BUSCLK↑)		t_{DKA}	1.5	14	ns
Address output hold time (from S_BUSCLK↑)		t_{HKA}	1.5	14	ns
$\overline{S\_CS[3:0]}$ and $\overline{S\_xxBE}$ delay time (from S_BUSCLK↑)		t_{DKCS}	1.5	14	ns
$\overline{S\_CS[3:0]}$ and $\overline{S\_xxBE}$ hold time (from S_BUSCLK↑)		t_{HKCS}	1.5	14	ns
$\overline{S\_BCYST}$↓ delay time (from S_BUSCLK↑)		t_{DKBSL}	1.5	14	ns
$\overline{S\_BCYST}$↑ delay time (from S_BUSCLK↑)		t_{DKBSH}	1.5	14	ns
Data float delay time (from S_BUSCLK↑)		t_{HKOD2}		14	ns
$\overline{S\_WAIT}$ setup time (to S_BUSCLK↓)		t_{SWK}	15		ns
$\overline{S\_WAIT}$ hold time (from S_BUSCLK↓)		t_{HKW}	2		ns
$\overline{S\_RD}$↓ delay time (from S_BUSCLK↑)		t_{DKRDL}	1.5	14	ns
$\overline{S\_RD}$↑ delay time (from S_BUSCLK↑)		t_{DKRDH}	1.5	14	ns
Data input setup time (to S_BUSCLK↑)		t_{SKID}	13		ns
Data input hold time (from S_BUSCLK↑)		t_{HKID}	0		ns
Data input setup time (to address, $\overline{S\_CS[3:0]}$, and $\overline{S\_xxBE}$)		t_{SAID}		$(w + w_D + w_{AS} + 1)$ T – 27	ns
Data input setup time (to $\overline{S\_RD}$↓)		t_{SRDID}		$(w + w_D + 1)$ T – 27	ns
$\overline{S\_RD}$ low-level width		t_{WRDL}	$(w + w_D + 1)$ T – 10		ns
$\overline{S\_RD}$↓ delay time from address, $\overline{S\_CS[3:0]}$, and $\overline{S\_xxBE}$		t_{DARD}	w_{AS}T – 10		ns
Address output delay time from $\overline{S\_RD}$↑		t_{DRDA}	iT – 6		ns
$\overline{S\_CS[3:0]}$ and $\overline{S\_xxBE}$ delay time from $\overline{S\_RD}$↑		t_{DADCS}	–6		ns
Data input hold time (from $\overline{S\_RD}$↑)		t_{HRDID}	0		ns
Data output delay time from $\overline{S\_RD}$↑		t_{DRDOD}	iT – 14		ns

Notes
1. xx: LL, LU, UL, UU
2. w: Number of wait cycles inserted by $\overline{S\_WAIT}$
3. w_D: Number of wait cycles specified by the SSMCn.DWn[3:0] bits
4. w_{AS}: Number of wait cycles specified by the SSMCn.ACn[3:0] bits
5. T: t_{SCYK} (S_BUSCLK cycle)
6. i: Number of idle states

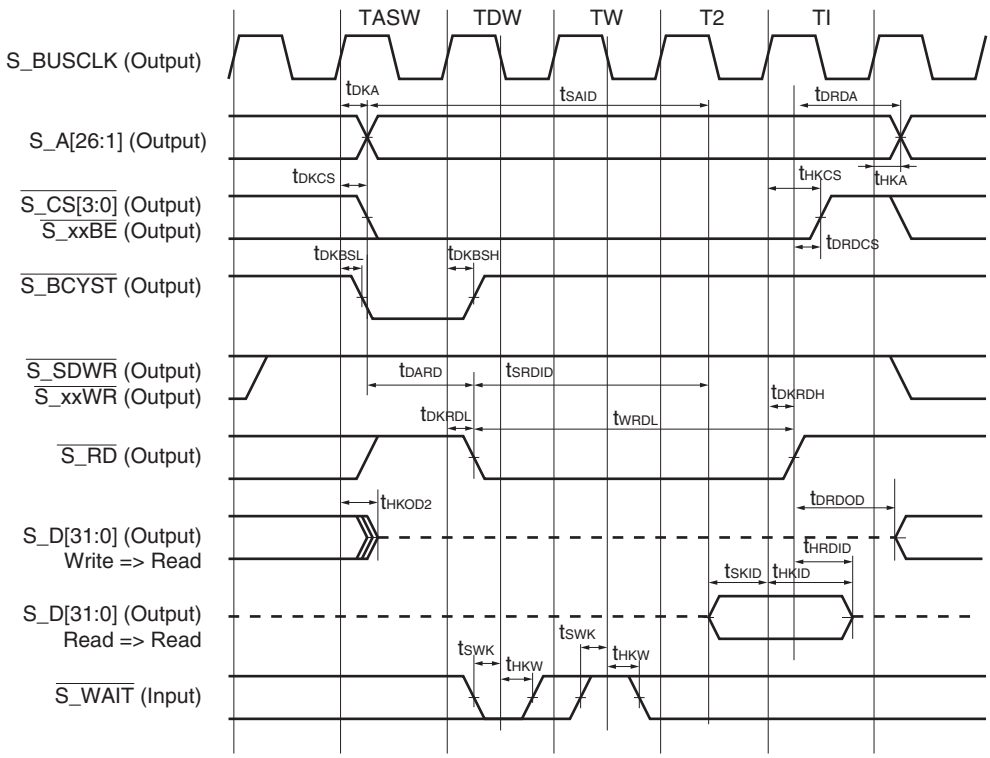


Figure 37-13 Secondary Memory Controller, SRAM Read Access Timing

(10) Secondary memory controller, SRAM write access timing

(T_t = –40 to 100°C , IV_{DD} = $PLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = UV_{DD} = $OSCV_{DD}$ = DV_{DD} = 3.0 to 3.6 V, AV_{DD} = 3.0 to 3.6 V or AV_{DD} = 4.5 to 5.5 V, V_{SS} = AV_{SS} = $OSCV_{SS}$ = $PLLV_{SS}$ = DV_{SS} = 0 V)

Item	Symbol		MIN.	MAX.	Unit
Address output delay time (from S_BUSCLK↑)		t_{DKA}	1.5	14	ns
Address output hold time (from S_BUSCLK↑)		t_{HKA}	1.5	14	ns
$\overline{S\_CS[3:0]}$ and $\overline{S\_xxBE}$ delay time (from S_BUSCLK↑)		t_{DKCS}	1.5	14	ns
$\overline{S\_CS[3:0]}$ and $\overline{S\_xxBE}$ hold time (from S_BUSCLK↑)		t_{HKCS}	1.5	14	ns
$\overline{S\_CS[3:0]}$ and $\overline{S\_xxBE}$ delay time (from $\overline{S\_xxWR}$, $\overline{S\_WR}$↑)		t_{DWRCS}	1.5	14	ns
$\overline{S\_BCYST}$↓ delay time (from S_BUSCLK↑)		t_{DKBSL}	1.5	14	ns
$\overline{S\_BCYST}$↑ delay time (from S_BUSCLK↑)		t_{DKBSH}	1.5	14	ns
$\overline{S\_xxWR}$ and $\overline{S\_WR}$↓ delay time (from S_BUSCLK↑)		t_{DKWRL}	1.5	14	ns
$\overline{S\_xxWR}$ and $\overline{S\_WR}$↑ delay time (from S_BUSCLK↑)		t_{DKWRH}	1.5	14	ns
Data output delay time (from S_BUSCLK↑)		t_{DKOD}	1.5	14	ns
Data output hold time (from S_BUSCLK↑)		t_{HKOD1}	1.5		ns
Data float delay time (from S_BUSCLK↑)		t_{HKOD2}		14	ns
$\overline{S\_WAIT}$ setup time (to S_BUSCLK↓)		t_{SWK}	15		ns
$\overline{S\_WAIT}$ hold time (from S_BUSCLK↓)		t_{HKW}	2		ns
Address, $\overline{S\_CS[3:0]}$, and $\overline{S\_xxBE}$ delay time (from $\overline{S\_xxWR}$, $\overline{S\_WR}$↓)		t_{DAWR}	$w_{AS}T$ – 10		ns
Address, $\overline{S\_CS[3:0]}$, and $\overline{S\_xxBE}$ setup time (to $\overline{S\_xxWR}$, $\overline{S\_WR}$↑)		t_{SAWR}	$(w + w_D + w_{AS} + 1)$ T – 10		ns
Address output delay time from $\overline{S\_xxWR}$ and $\overline{S\_WR}$↑		t_{DWRA}	$(w_{DH} + i)$ T – 10		ns
$\overline{S\_xxWR}$ and $\overline{S\_WR}$ low-level width		t_{WWRL}	$(w + w_{DH} + 1)$ T – 10		ns
Data output setup time (to $\overline{S\_xxWR}$ and $\overline{S\_WR}$↑)		t_{SODWR}	$(w + w_{DH} + 1)$ T – 10		ns
Data output hold time (from $\overline{S\_xxWR}$ and $\overline{S\_WR}$↑)		t_{HWROD}	$w_{DH}T$ – 10		ns

Notes
1. xx: LL, LU, UL, UU
2. w: Number of wait cycles inserted by $\overline{S\_WAIT}$
3. w_D: Number of wait cycles specified by the SSMCn.DWn[3:0] bits
4. w_{AS}: Number of wait cycles specified by the SSMCn.ACn[3:0] bits
5. w_{DH}: Number of wait cycles specified by the SSMCn.WWn[3:0] bits
6. T: t_{SCYK} (S_BUSCLK cycle)
7. i: Number of idle states

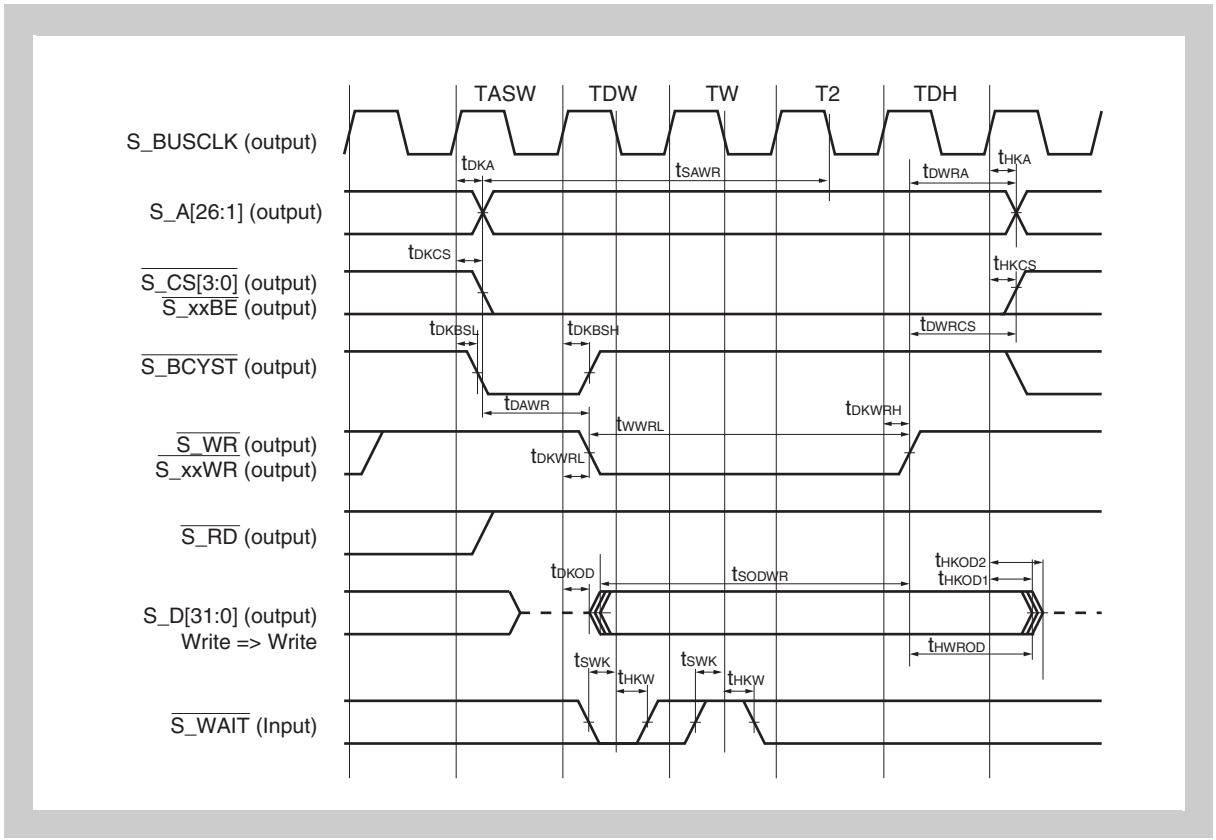


Figure 37-14 Secondary Memory Controller, SRAM Write Access Timing

(11) Secondary memory controller, SDRAM read timing

(T_t = –40 to 100°C , IV_{DD} = $PLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = UV_{DD} = $OSCV_{DD}$ = DV_{DD} = 3.0 to 3.6 V, AV_{DD} = 3.0 to 3.6 V or AV_{DD} = 4.5 to 5.5 V, V_{SS} = AV_{SS} = $OSCV_{SS}$ = $PLLV_{SS}$ = DV_{SS} = 0 V)

Item	Symbol		MIN.	MAX.	Unit
Address output delay time (from S_BUSCLK↑)		t_{DKA}	1.5	14	ns
Address output hold time (from S_BUSCLK↑)		t_{HKA}	1.5	14	ns
$\overline{\text{S\_BCYST}}$↓ delay time (from S_BUSCLK↑)		t_{DKBSL}	1.5	14	ns
$\overline{\text{S\_BCYST}}$↑ delay time (from S_BUSCLK↑)		t_{DKBSH}	1.5	14	ns
$\overline{\text{S\_SDCS}}$ delay time (from S_BUSCLK↑)		t_{DKCS}	1.5	14	ns
$\overline{\text{S\_SDRAS}}$ delay time (from S_BUSCLK↑)		t_{DKRAS}	1.5	14	ns
$\overline{\text{S\_SDCAS}}$ delay time (from S_BUSCLK↑)		t_{DKCAS}	1.5	14	ns
S_xxDQM delay time (from S_BUSCLK↑)		t_{DKDQM}	1.5	14	ns
S_SDCKE delay time (from S_BUSCLK↑)		t_{DKCKE}	1.5	14	ns
$\overline{\text{S\_SDWE}}$ delay time (from S_BUSCLK↑)		t_{DKWE}	1.5	14	ns
Data input setup time (SDRAM read time, to S_BUSCLK↑)		t_{SDRMK}	13		ns
Data input hold time (SDRAM read time, from S_BUSCLK↑)		t_{HKDRM}	0		ns
Data output delay time from S_BUSCLK↑		t_{DSDOD}	1T		ns

Note xx: LL, LU, UL, UU

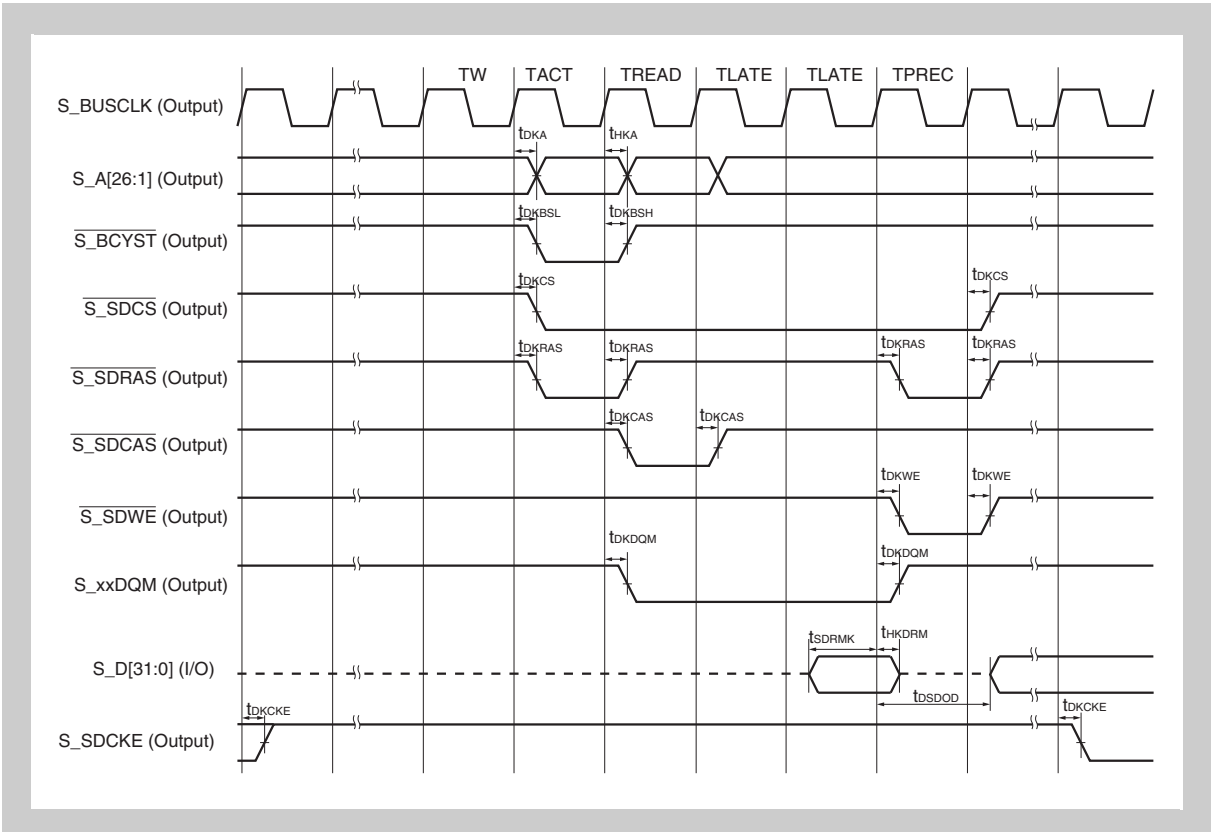


Figure 37-15 Secondary Memory Controller, SDRAM Read Timing

(12) Secondary memory controller, SDRAM write timing

(T_t = –40 to 100°C , IV_{DD} = $PLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = UV_{DD} = $OSCV_{DD}$ = DV_{DD} = 3.0 to 3.6 V, AV_{DD} = 3.0 to 3.6 V or AV_{DD} = 4.5 to 5.5 V, V_{SS} = AV_{SS} = $OSCV_{SS}$ = $PLLV_{SS}$ = DV_{SS} = 0 V)

Item	Symbol		MIN.	MAX.	Unit
Address output delay time (from S_BUSCLK↑)		t_{DKA}	1.5	14	ns
Address output hold time (from S_BUSCLK↑)		t_{HKA}	1.5	14	ns
$\overline{\text{S\_BCYST}}$↓ delay time (from S_BUSCLK↑)		t_{DKBSL}	1.5	14	ns
$\overline{\text{S\_BCYST}}$↑ delay time (from S_BUSCLK↑)		t_{DKBSH}	1.5	14	ns
$\overline{\text{S\_SDCS}}$ delay time (from S_BUSCLK↑)		t_{DKCS}	1.5	14	ns
$\overline{\text{S\_SDRAS}}$ delay time (from S_BUSCLK↑)		t_{DKRAS}	1.5	14	ns
$\overline{\text{S\_SDCAS}}$ delay time (from S_BUSCLK↑)		t_{DKCAS}	1.5	14	ns
S_xxDQM delay time (from S_BUSCLK↑)		t_{DKDQM}	1.5	14	ns
S_SDCKE delay time (from S_BUSCLK↑)		t_{DKCKE}	1.5	14	ns
$\overline{\text{S\_SDWE}}$ delay time (from S_BUSCLK↑)		t_{DKWE}	1.5	14	ns
Data output delay time (from S_BUSCLK↑)		t_{DKDT}	1.5	14	ns
Data output hold time (from S_BUSCLK↑)		t_{HKDT1}	1.5		ns
Data float delay time (from S_BUSCLK↑)		t_{HKDT2}		14	ns

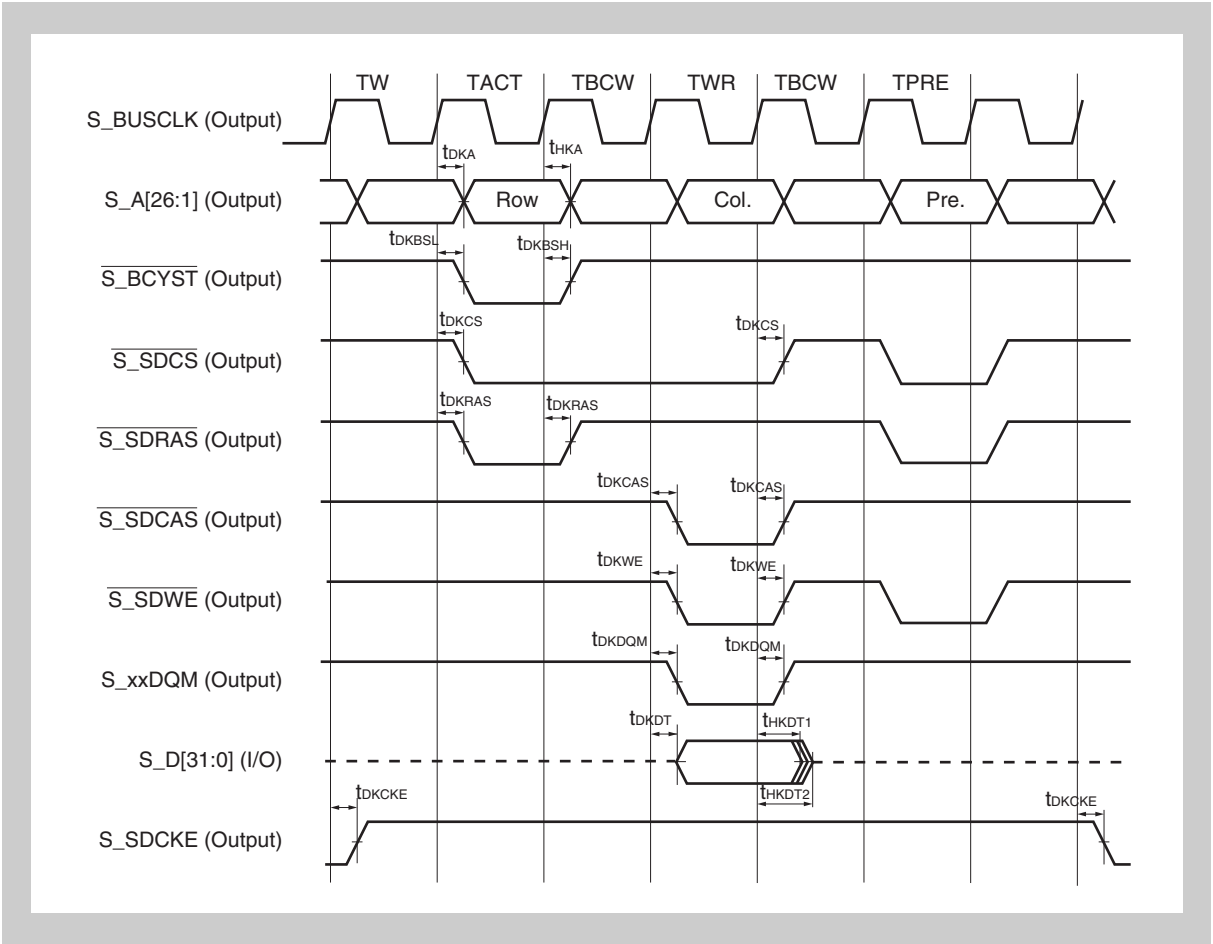


Figure 37-16 Secondary Memory Controller, SDRAM Write Timing

(13) Secondary memory controller, SRAM S_DMA access timing

(T_t = –40 to 100°C , IV_{DD} = $PLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = UV_{DD} = $OSCV_{DD}$ = DV_{DD} = 3.0 to 3.6 V, AV_{DD} = 3.0 to 3.6 V or AV_{DD} = 4.5 to 5.5 V, V_{SS} = AV_{SS} = $OSCV_{SS}$ = $PLLV_{SS}$ = DV_{SS} = 0 V)

Item	Symbol		MIN.	MAX.	Unit
$\overline{\text{S\_DMARQ[3:0]}}$ input setup time (to S_BUSCLK↑)		t_{SDRK}	13		ns
$\overline{\text{S\_DMARQ[3:0]}}$ input hold time		t_{HKDR1}	to $\overline{\text{S\_DMAAK[3:0]}}$↓		ns
$\overline{\text{S\_DMARQ[3:0]}}$ input hold time (from S_BUSCLK↑)		t_{HKDR2}		T × m – 13	ns
$\overline{\text{S\_DMAAK[3:0]}}$ output delay time (from S_BUSCLK↑)		t_{DKDA}	1.5	14	ns
$\overline{\text{S\_DMAAK[3:0]}}$ low-level width		t_{WDAL}	T – 10		ns
$\overline{\text{S\_DMAAK[3:0]}}$ ↑ output delay time (from S_BUSCLK↑)		t_{HKDA}	1.5	14	ns
$\overline{\text{S\_DMATC[3:0]}}$ ↓ output delay time (from S_BUSCLK↑)		t_{DKTC}	1.5	14	ns
$\overline{\text{S\_DMATC[3:0]}}$ ↑ output delay time (from S_BUSCLK↑)		t_{HKTC}	1.5	14	ns
$\overline{\text{S\_DMATC[3:0]}}$ low-level width		t_{WDTL}	T – 10		ns

Note T: t_{SCYK} (S_BUSCLK cycle)

m:1 to 16(Setting by DMAIFC0 to DMAIFC3 registers)

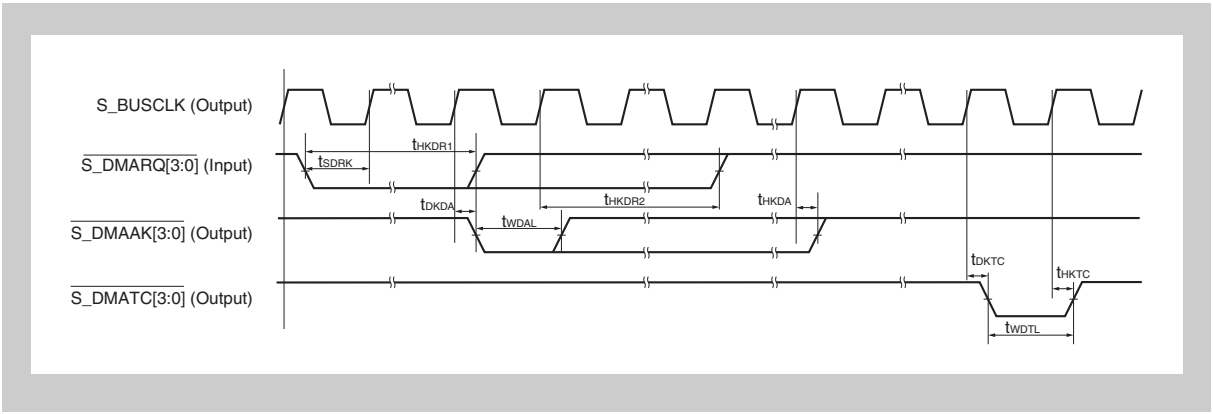


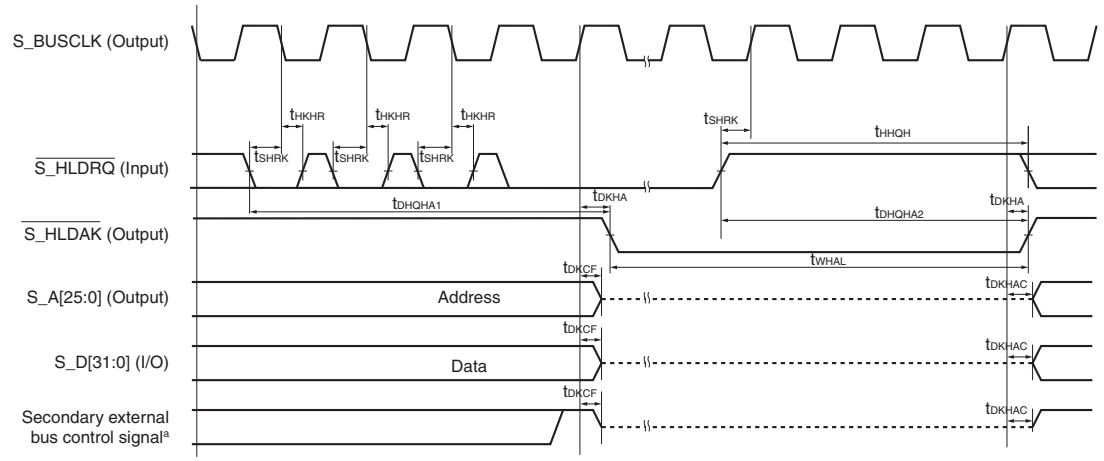
Figure 37-17 Secondary memory controller, SRAM S_DMA access timing

(14) Secondary memory controller, bus hold timing

(T_t = –40 to 100°C , IV_{DD} = $PLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = UV_{DD} = $OSCV_{DD}$ = DV_{DD} = 3.0 to 3.6 V, AV_{DD} = 3.0 to 3.6 V or AV_{DD} = 4.5 to 5.5 V, V_{SS} = AV_{SS} = $OSCV_{SS}$ = $PLLV_{SS}$ = DV_{SS} = 0 V)

Item	Symbol		MIN.	MAX.	Unit
$\overline{\text{S\_HLDRQ}}$ setup time (to S_BUSCLK↑)		t_{SHRK}	15		ns
$\overline{\text{S\_HLDRQ}}$ hold time (from S_BUSCLK↑)		t_{HKHR}	2		ns
$\overline{\text{S\_HLDAK}}$ delay time (from S_BUSCLK↑)		t_{DKHA}	1.5	14	ns
$\overline{\text{S\_HLDRQ}}$ high-level width		t_{HHQH}	to $\overline{\text{S\_HLDAK}}$↑		ns
$\overline{\text{S\_HLDAK}}$ low-level width		t_{WHAL}	T – 10		ns
Bus float delay time (from S_BUSCLK↑)		t_{DKCF}		14	ns
Bus output delay time (from S_BUSCLK↑)		t_{DKHAC}	1.5	14	ns
$\overline{\text{S\_HLDAK}}$↓ delay time from $\overline{\text{S\_HLDRQ}}$↓		t_{DHQHA1}	2.5T		ns
$\overline{\text{S\_HLDAK}}$↑ delay time from $\overline{\text{S\_HLDRQ}}$↑		t_{DHQHA2}	2.5T		ns

Note T: t_{SCYK} (S_BUSCLK cycle)



a) Secondary external bus control signals: $\overline{\text{S\_CS[3:0]}}$, $\overline{\text{S\_SDCS}}$, $\overline{\text{S\_xxBE}}$, $\overline{\text{S\_BCYST}}$, $\overline{\text{S\_xxWR}}$, $\overline{\text{S\_WR}}$, $\overline{\text{S\_RD}}$, $\overline{\text{S\_SDRAS}}$, $\overline{\text{S\_SDCAS}}$, $\overline{\text{S\_SDWE}}$, S_SDCKE, S_xxDQM

37.6.3 Clock timing

(T_t = –40 to 100°C , IV_{DD} = $PLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = UV_{DD} = $OSCV_{DD}$ = DV_{DD} = 3.0 to 3.6 V, AV_{DD} = 3.0 to 3.6 V or AV_{DD} = 4.5 to 5.5 V, V_{SS} = AV_{SS} = $OSCV_{SS}$ = $PLLV_{SS}$ = DV_{SS} = 0 V)

Item	Symbol		Conditions	MIN.	MAX.	Unit
CPU operating frequency				144	200	MHz
P_BUSCLK output cycle		t_{BCYK}	SRAM mode	15		ns
			SDRAM mode	20		ns
P_BUSCLK high-level width		t_{BWKH}	During PLL operation when m = 3, 4, or 6[a]	m × 1000 × 0.4/200	m × 1000 × 0.6/144	ns
P_BUSCLK low-level width		t_{BWKL}	During PLL operation when m = 3, 4, or 6[a]	m × 1000 × 0.4/200	m × 1000 × 0.6/144	ns
P_BUSCLK rising edge width		t_{BKR}			5	ns
P_BUSCLK falling edge width		t_{BKF}			5	ns
S_BUSCLK output cycle		t_{SCYK}	SRAM mode	15		ns
			SDRAM mode	20		ns
S_BUSCLK high-level width		t_{SWKH}	During PLL operation when m = 3, 4, or 6[a]	m × 1000 × 0.4/200	m × 1000 × 0.6/144	ns
S_BUSCLK low-level width		t_{SWKL}	During PLL operation when m = 3, 4, or 6[a]	m × 1000 × 0.4/200	m × 1000 × 0.6/144	ns
S_BUSCLK rising edge width		t_{SKR}			5	ns
S_BUSCLK falling edge width		t_{SKF}			5	ns

a) During the reset period, the main clock oscillator output is output from the P_BUSCLK and S_BUSCLK pins. During the reset period, be careful of the duty factor for the P_BUSCLK and S_BUSCLK pins.

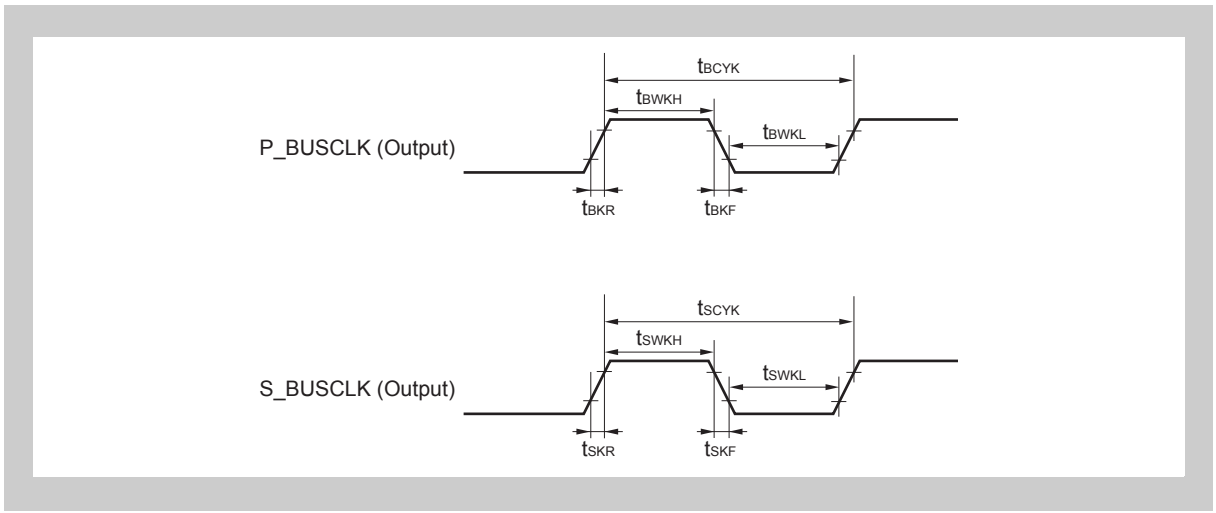


Figure 37-18 Clock Timing

37.6.4 Reset timing

(T_t = –40 to 100°C , IV_{DD} = $PLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = UV_{DD} = $OSCV_{DD}$ = DV_{DD} = 3.0 to 3.6 V, AV_{DD} = 3.0 to 3.6 V or AV_{DD} = 4.5 to 5.5 V, V_{SS} = AV_{SS} = $OSCV_{SS}$ = $PLLV_{SS}$ = DV_{SS} = 0 V)

Item	Symbol		Conditions	MIN.	MAX.	Unit
$\overline{\text{RESET}}$ high-level width		t_{WRSH}		500		ns
$\overline{\text{RESET}}$ low-level width		t_{WRSL}	When the power is on	500 + t_{OSC}		ns
			When the power is not on	500		ns

Note t_{OSC}: Main clock oscillation stabilization time (differs depending on the used oscillator)

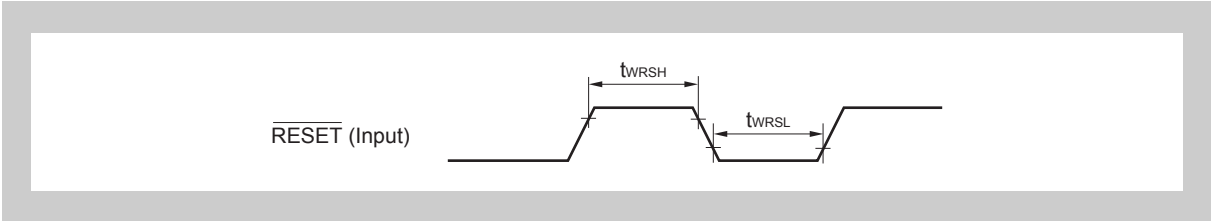


Figure 37-19 Reset Timing

37.6.5 Interrupts

(T_t = –40 to 100°C , IV_{DD} = $PLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = UV_{DD} = $OSCV_{DD}$ = DV_{DD} = 3.0 to 3.6 V, AV_{DD} = 3.0 to 3.6 V or AV_{DD} = 4.5 to 5.5 V, V_{SS} = AV_{SS} = $OSCV_{SS}$ = $PLLV_{SS}$ = DV_{SS} = 0 V)

Item	Symbol		Conditions	MIN.	TYP.	MAX.	Unit
NMI high-level width		t_{WNIH}	NMI high-level width	500			ns
NMI low-level width		t_{WNIL}	NMI low-level width	500			ns
INTPm pin high-level width		t_{WITACH}	When analog noise elimination is specified	500			ns
INTPm pin low-level width		t_{WITACL}	When analog noise elimination is specified	500			ns
INTPn pin high-level width		t_{WITDCH}	When digital noise elimination is specified	(number of specified elimination clocks –1)/ (PCLK) + 10			ns
INTPn pin low-level width		t_{WITDCL}	When digital noise elimination is specified	(number of specified elimination clocks –1)/ (PCLK) + 10			ns
ESOx high-level width		t_{ESWTCH}	When noise elimination is specified	500			ns
ESOx low-level width		t_{ESWTCL}	When noise elimination is specified	500			ns

Note m = 0 to 4, n = 5 to 27, x = 0 to 3

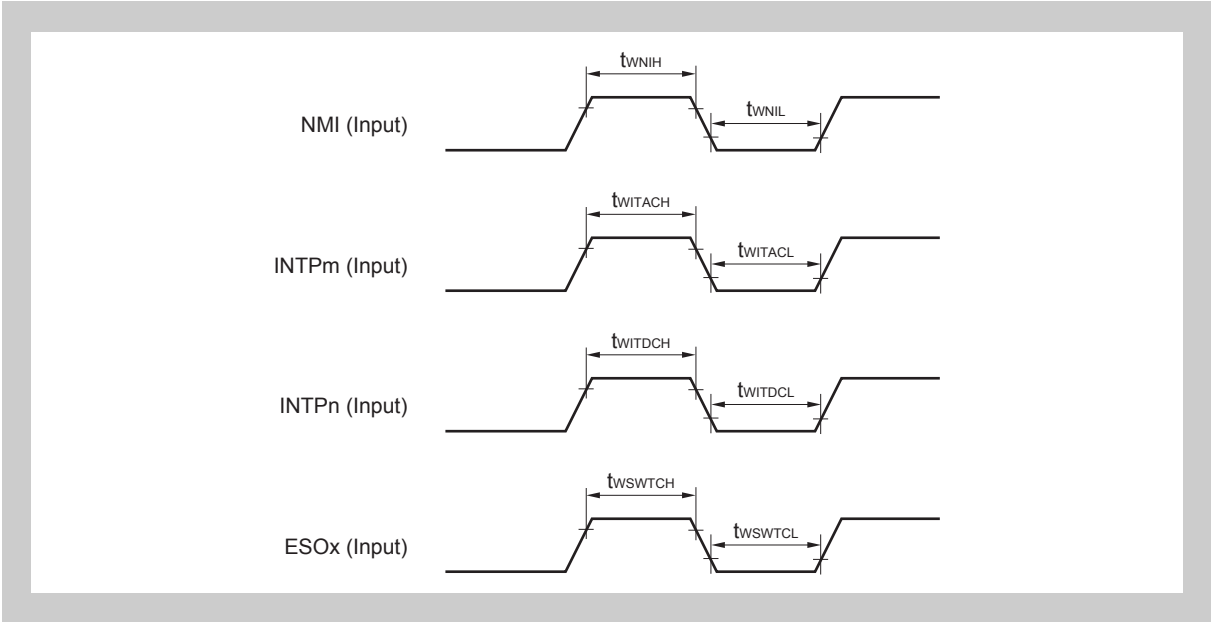


Figure 37-20 Interrupt Timing

37.6.6 TAUA

(T_t = –40 to 100°C , IV_{DD} = $PLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = UV_{DD} = $OSCV_{DD}$ = DV_{DD} = 3.0 to 3.6 V, AV_{DD} = 3.0 to 3.6 V or AV_{DD} = 4.5 to 5.5 V, V_{SS} = AV_{SS} = $OSCV_{SS}$ = $PLLV_{SS}$ = DV_{SS} = 0 V)

Item	Symbol		Conditions	MIN.	TYP.	MAX.	Unit
TAm_In high-level width		t_{AWTCH}	When noise elimination is specified	(number of specified elimination clocks –1)/ (PCLK) + 10			ns
TAm_In low-level width		t_{AWTCL}	When noise elimination is specified	(number of specified elimination clocks –1)/ (PCLK) + 10			ns

Note n = 0 to 15, m = 0 to 3

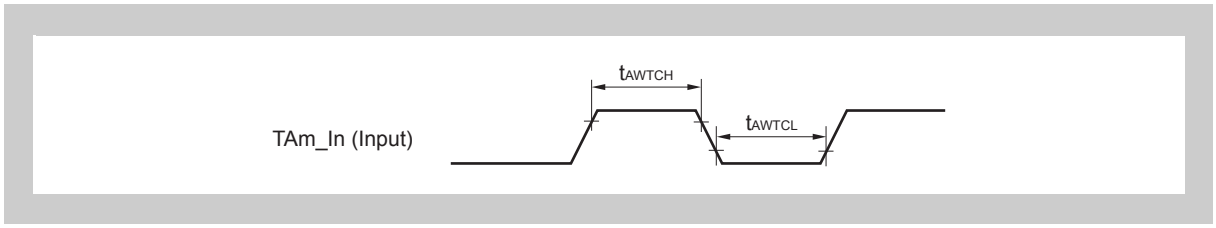


Figure 37-21 TAUA Timing

37.6.7 TAUJ

(T_t = –40 to 100°C , IV_{DD} = $PLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = UV_{DD} = $OSCV_{DD}$ = DV_{DD} = 3.0 to 3.6 V, AV_{DD} = 3.0 to 3.6 V or AV_{DD} = 4.5 to 5.5 V, V_{SS} = AV_{SS} = $OSCV_{SS}$ = $PLLV_{SS}$ = DV_{SS} = 0 V)

Item	Symbol		Conditions	MIN.	TYP.	MAX.	Unit
TJ_In high-level width		t_{JWTCH}	When noise elimination is specified	(number of specified elimination clocks –1)/ (PCLK) + 10			ns
TJ_In low-level width		t_{JWTCL}	When noise elimination is specified	(number of specified elimination clocks –1)/ (PCLK) + 10			ns

Note n = 0 to 3

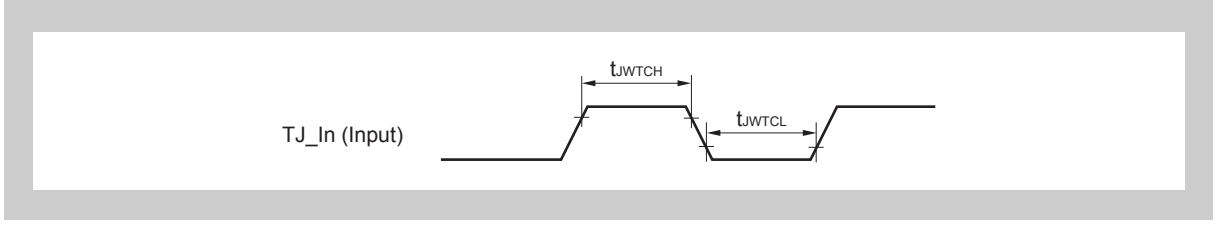


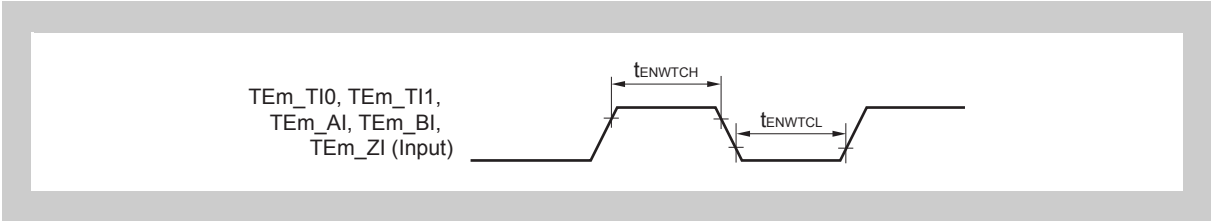
Figure 37-22 TAUJ Timing

37.6.8 ENCA

(T_t = –40 to 100°C , IV_{DD} = $PLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = UV_{DD} = $OSCV_{DD}$ = DV_{DD} = 3.0 to 3.6 V, AV_{DD} = 3.0 to 3.6 V or AV_{DD} = 4.5 to 5.5 V, V_{SS} = AV_{SS} = $OSCV_{SS}$ = $PLLV_{SS}$ = DV_{SS} = 0 V)

Item	Symbol		Conditions	MIN.	TYP.	MAX.	Unit
TEm_TI0, TEm_TI1, TEm_AI, TEm_BI, and TEm_ZI high-level width		t_{ENWTCH}	When noise elimination is specified	(number of specified elimination clocks –1)/ (PCLK) + 10			ns
TEm_TI0, TEm_TI1, TEm_AI, TEm_BI, and TEm_ZI low-level width		t_{ENWTCL}	When noise elimination is specified	(number of specified elimination clocks –1)/ (PCLK) + 10			ns

Note m = 0, 1



37.6.9 CSIH

(1) CSIH0, CSIH2, CSIH3 (8.33 Mbps (f_{XX} = 200 MHz): While in the master mode)

(T_t = –40 to 100°C , IV_{DD} = $PLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = UV_{DD} = $OSCV_{DD}$ = DV_{DD} = 3.0 to 3.6 V, AV_{DD} = 3.0 to 3.6 V or AV_{DD} = 4.5 to 5.5 V, V_{SS} = AV_{SS} = $OSCV_{SS}$ = $PLLV_{SS}$ = DV_{SS} = 0 V)

Item	Symbol		Conditions	MIN.	MAX.	Unit
Clock source		t_{KCY}	Input	15		ns
$\overline{SCKnF}$ cycle		t_{KCYM}	Output	120		ns
$\overline{SCKnF}$ high-level width		t_{KWHM}	Output	(t_{KCYM}/2) –8.0		ns
$\overline{SCKnF}$ low-level width		t_{KWLM}	Output	(t_{KCYM}/2) –8.0		ns
SInF setup time (to $\overline{SCKnF}$↑)		t_{SSIM}	Input	26.0		ns
SInF setup time (to $\overline{SCKnF}$↓)				26.0		ns
SInF hold time (from $\overline{SCKnF}$↑)		t_{HSIM}	Input	26.0		ns
SInF hold time (from $\overline{SCKnF}$↓)				26.0		ns
SOnF output delay time (from $\overline{SCKnF}$↑)		t_{DSOM}	Output		26	ns
SOnF output delay time (from $\overline{SCKnF}$↓)					26	ns
SOnF output hold time (from $\overline{SCKnF}$↑)		t_{HSOM}	Output	(t_{KCYM}/2) –10.0		ns
SOnF output hold time (from $\overline{SCKnF}$↓)				(t_{KCYM}/2) –10.0		ns
CSInF_RYI setup time (to $\overline{SCKnF}$↑): CSIHnCTL1.CSIHnSIT = x		t_{SRDYI}	Input	((t_{KCYM}/2)/t_{KCY} –2) × t_{KCY}	(Min. value) + (t_{KCYM}/2) –26	ns
CSInF_RYI setup time (to $\overline{SCKnF}$↓): CSIHnCTL1.CSIHnSIT = x				((t_{KCYM}/2)/t_{KCY} –2) × t_{KCY}	(Min. value) + (t_{KCYM}/2) –26	ns
CSInF_RYI high-level width		t_{WRYI}	Input	t_{KCY} + 10		ns
CSInF_CS[7:0] setup time (to $\overline{SCKnF}$↑): CSIHnCTL1.CSIHnDAP = 0		t_{SSCSB0}	Output	(CSSETUP + 0.5) × t_{KCYM} –26		ns
CSInF_CS[7:0] setup time (to $\overline{SCKnF}$↓): CSIHnCTL1.CSIHnDAP = 0				CSSETUP × t_{KCYM} –26		ns
CSInF_CS[7:0] setup time (to $\overline{SCKnF}$↑): CSIHnCTL1.CSIHnDAP = 1		t_{SSCSB1}	Output	CSSETUP × t_{KCYM} –26		ns
CSInF_CS[7:0] setup time (to $\overline{SCKnF}$↓): CSIHnCTL1.CSIHnDAP = 1				CSSETUP × t_{KCYM} –26		ns
CSInF_CS[7:0] hold time (from $\overline{SCKnF}$↑): CSIHnCTL1.CSIHnDAP = 0		t_{HSCSB0}	Output	(CSHOLD + 0.5) × t_{KCYM} –26		ns
CSInF_CS[7:0] hold time (from $\overline{SCKnF}$↓): CSIHnCTL1.CSIHnDAP = 0				(CSHOLD + 0.5) × t_{KCYM} –26		ns
CSInF_CS[7:0] hold time (from $\overline{SCKnF}$↑): CSIHnCTL1.CSIHnDAP = 1		t_{HSCSB1}	Output	(CSHOLD + 0.5) × t_{KCYM} –26		ns
CSInF_CS[7:0] hold time (from $\overline{SCKnF}$↓): CSIHnCTL1.CSIHnDAP = 1				(CSHOLD + 0.5) × t_{KCYM} –26		ns

Note n = 0, 2, or 3

CSSETUP: Settings for the CSIHnSP[3:0] bits of CSIHnCFG0 to CSIHnCFG7

CSHOLD: Settings for the CSIHnHD[3:0] bits of CSIHnCFG0 to CSIHnCFG7

(2) CSIH0, CSIH2, CSIH3 (8.33 Mbps (f_{XX} = 200 MHz): While in the slave mode)

(T_t = –40 to 100°C , IV_{DD} = $PLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = UV_{DD} = $OSCV_{DD}$ = DV_{DD} = 3.0 to 3.6 V, AV_{DD} = 3.0 to 3.6 V or AV_{DD} = 4.5 to 5.5 V, V_{SS} = AV_{SS} = $OSCV_{SS}$ = $PLLV_{SS}$ = DV_{SS} = 0 V)

Item	Symbol	Conditions	MIN.	MAX.	Unit
Clock source	t_{KCY}	Input	15		ns
$\overline{SCKnF}$ cycle	t_{KCYS}	Input	120		ns
$\overline{SCKnF}$ high-level width	t_{KWHS}	Input	(t_{KCYS}/2) –8.0		ns
$\overline{SCKnF}$ low-level width	t_{KWLS}	Input	(t_{KCYS}/2) –8.0		ns
SInF setup time (to $\overline{SCKnF}$↑)	t_{SSIS}	Input	26.0		ns
SInF setup time (to $\overline{SCKnF}$↓)			26.0		ns
SInF hold time (from $\overline{SCKnF}$↑)	t_{HSIS}	Input	(t_{KCYS}/2) –8.0		ns
SInF hold time (from $\overline{SCKnF}$↓)			(t_{KCYS}/2) –8.0		ns
SOnF output delay time (from $\overline{SCKnF}$↑)	t_{DSOS}	Output		26.0	ns
SOnF output delay time (from $\overline{SCKnF}$↓)				26.0	ns
SOnF output hold time (from $\overline{SCKnF}$↑)	t_{HSOS}	Output	(t_{KCYS}/2) –10.0		ns
SOnF output hold time (from $\overline{SCKnF}$↓)			(t_{KCYS}/2) –10.0		ns
CSInF_RYO output delay time (from $\overline{SCKnF}$↑)	t_{SRDYIO}	Output		26.0	ns
CSInF_RYO output delay time (from $\overline{SCKnF}$↓)				26.0	ns
CSInF_RYO output hold time (from $\overline{SCKnF}$↑)	t_{HRDYIO}	Output	(t_{KCYS}/2) –10.0		ns
CSInF_RYO output hold time (from $\overline{SCKnF}$↓)			(t_{KCYS}/2) –10.0		ns
CSInF_SSI setup time (to $\overline{SCKnF}$↑)	t_{SSSIS}	Input	26.0		ns
CSInF_SSI setup time (to $\overline{SCKnF}$↓)			26.0		ns
CSInF_SSI hold time (from $\overline{SCKnF}$↑)	t_{HSSIS}	Input	26.0		ns
CSInF_SSI hold time (from $\overline{SCKnF}$↓)			26.0		ns

Note n = 0, 2, or 3

(3) CSIH1 (8.33 Mbps (f_{XX} = 200 MHz): While in the master mode)

(T_t = –40 to 100°C , IV_{DD} = $PLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = UV_{DD} = $OSCV_{DD}$ = DV_{DD} = 3.0 to 3.6 V, AV_{DD} = 3.0 to 3.6 V or AV_{DD} = 4.5 to 5.5 V, V_{SS} = AV_{SS} = $OSCV_{SS}$ = $PLLV_{SS}$ = DV_{SS} = 0 V)

Item	Symbol		Conditions	MIN.	MAX.	Unit
Clock source		t_{KCY}	Input	15		ns
$\overline{SCK1F}$ cycle		t_{KCYM}	Output	120		ns
$\overline{SCK1F}$ high-level width		t_{KWHM}	Output	$(t_{KCYM}/2)$ –13.0		ns
$\overline{SCK1F}$ low-level width		t_{KWLM}	Output	$(t_{KCYM}/2)$ –13.0		ns
SI1F setup time (to $\overline{SCK1F}$↑)		t_{SSIM}	Input	26.0		ns
SI1F setup time (to $\overline{SCK1F}$↓)				26.0		ns
SI1F hold time (from $\overline{SCK1F}$↑)		t_{HSIM}	Input	26.0		ns
SI1F hold time (from $\overline{SCK1F}$↓)				26.0		ns
SO1F output delay time (from $\overline{SCK1F}$↑)		t_{DSOM}	Output		26	ns
SO1F output delay time (from $\overline{SCK1F}$↓)					26	ns
SO1F output hold time (from $\overline{SCK1F}$↑)		t_{HSOM}	Output	$(t_{KCYM}/2)$ –10.0		ns
SO1F output hold time (from $\overline{SCK1F}$↓)				$(t_{KCYM}/2)$ –10.0		ns
CSI1F_RYI setup time (to $\overline{SCK1F}$↑): CSIH0CTL1.CSIH0SIT = x		t_{SRDYI}	Input	$((t_{KCYM}/2)/t_{KCY}$ –2) × t_{KCY}	(Min. value) + $(t_{KCYM}/2)$ –26	ns
CSI1F_RYI setup time (to $\overline{SCK1F}$↓): CSIH0CTL1.CSIH0SIT = x				$((t_{KCYM}/2)/t_{KCY}$ –2) × t_{KCY}	(Min. value) + $(t_{KCYM}/2)$ –26	ns
CSI1F_RYI high-level width		t_{WRYI}	Input	t_{KCY} + 10		ns
CSI1F_CS[7:0] setup time (to $\overline{SCK1F}$↑): CSIH0CTL1.CSIH0DAP = 0		t_{SSCSB0}	Output	(CSSETUP + 0.5) × t_{KCYM} – 26		ns
CSI1F_CS[7:0] setup time (to $\overline{SCK1F}$↓): CSIH0CTL1.CSIH0DAP = 0				CSSETUP × t_{KCYM} –26		ns
CSI1F_CS[7:0] setup time (to $\overline{SCK1F}$↑: CSIH0CTL1.CSIH0DAP = 1		t_{SSCSB1}	Output	CSSETUP × t_{KCYM} –26		ns
CSI1F_CS[7:0] setup time (to $\overline{SCK1F}$↓): CSIH0CTL1.CSIH0DAP = 1				CSSETUP × t_{KCYM} –26		ns
CSI1F_CS[7:0] hold time (from $\overline{SCK1F}$↑): CSIH0CTL1.CSIH0DAP = 0		t_{HSCSB0}	Output	(CSHOLD + 0.5) × t_{KCYM} –26		ns
CSI1F_CS[7:0] hold time (from $\overline{SCK1F}$↓): CSIH0CTL1.CSIH0DAP = 0				(CSHOLD + 0.5) × t_{KCYM} – 26		ns
CSI1F_CS[7:0] hold time (from $\overline{SCK1F}$↑): CSIH0CTL1.CSIH0DAP = 1		t_{HSCSB1}	Output	(CSHOLD + 0.5) × t_{KCYM} –26		ns
CSI1F_CS[7:0] hold time (from $\overline{SCK1F}$↓): CSIH0CTL1.CSIH0DAP = 1				(CSHOLD + 0.5) × t_{KCYM} –26		ns

Note CSSETUP: Settings for the CSIH1SP[3:0] bits of CSIH1CFG0 to CSIH1CFG7
CSHOLD: Settings for the CSIH1HD[3:0] bits of CSIH1CFG0 to CSIH1CFG7

(4) CSIH1 (8.33 Mbps (f_{XX} = 200 MHz): While in the slave mode)

(T_t = –40 to 100°C , IV_{DD} = $PLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = UV_{DD} = $OSCV_{DD}$ = DV_{DD} = 3.0 to 3.6 V, AV_{DD} = 3.0 to 3.6 V or AV_{DD} = 4.5 to 5.5 V, V_{SS} = AV_{SS} = $OSCV_{SS}$ = $PLLV_{SS}$ = DV_{SS} = 0 V)

Item	Symbol		Conditions	MIN.	MAX.	Unit
Clock source		t_{KCY}	Input	15		ns
$\overline{SCK1F}$ cycle		t_{KCYS}	Input	120		ns
$\overline{SCK1F}$ high-level width		t_{KWHS}	Input	(t_{KCYS}/2) –8.0		ns
$\overline{SCK1F}$ low-level width		t_{KWLS}	Input	(t_{KCYS}/2) –8.0		ns
SI1F setup time (to $\overline{SCK1F}$↑)		t_{SSIS}	Input	26.0		ns
SI1F setup time (to $\overline{SCK1F}$↓)				26.0		ns
SI1F hold time (from $\overline{SCK1F}$↑)		t_{HSIS}	Input	(t_{KCYS}/2) –8.0		ns
SI1F hold time (from $\overline{SCK1F}$↓)				(t_{KCYS}/2) –8.0		ns
SO1F output delay time (from $\overline{SCK1F}$↑)		t_{DSOS}	Output		26.0	ns
SO1F output delay time (from $\overline{SCK1F}$↓)					26.0	ns
SO1F output hold time (from $\overline{SCK1F}$↑)		t_{HSOS}	Output	(t_{KCYS}/2) –10.0		ns
SO1F output hold time (from $\overline{SCK1F}$↓)				(t_{KCYS}/2) –10.0		ns
CSI1F_RYO output delay time (from $\overline{SCK1F}$↑)		t_{SRDYIO}	Output		26.0	ns
CSI1F_RYO output delay time (from $\overline{SCK1F}$↓)					26.0	ns
CSI1F_RYO output hold time (from $\overline{SCK1F}$↑)		t_{HRDYIO}	Output	(t_{KCYS}/2) –10.0		ns
CSI1F_RYO output hold time (from $\overline{SCK1F}$↓)				(t_{KCYS}/2) –10.0		ns
CSI1F_SSI setup time (to $\overline{SCK1F}$↑)		t_{SSSIS}	Input	26.0		ns
CSI1F_SSI setup time (to $\overline{SCK1F}$↓)				26.0		ns
CSI1F_SSI hold time (from $\overline{SCK1F}$↑)		t_{HSSIS}	Input	26.0		ns
CSI1F_SSI hold time (from $\overline{SCK1F}$↓)				26.0		ns

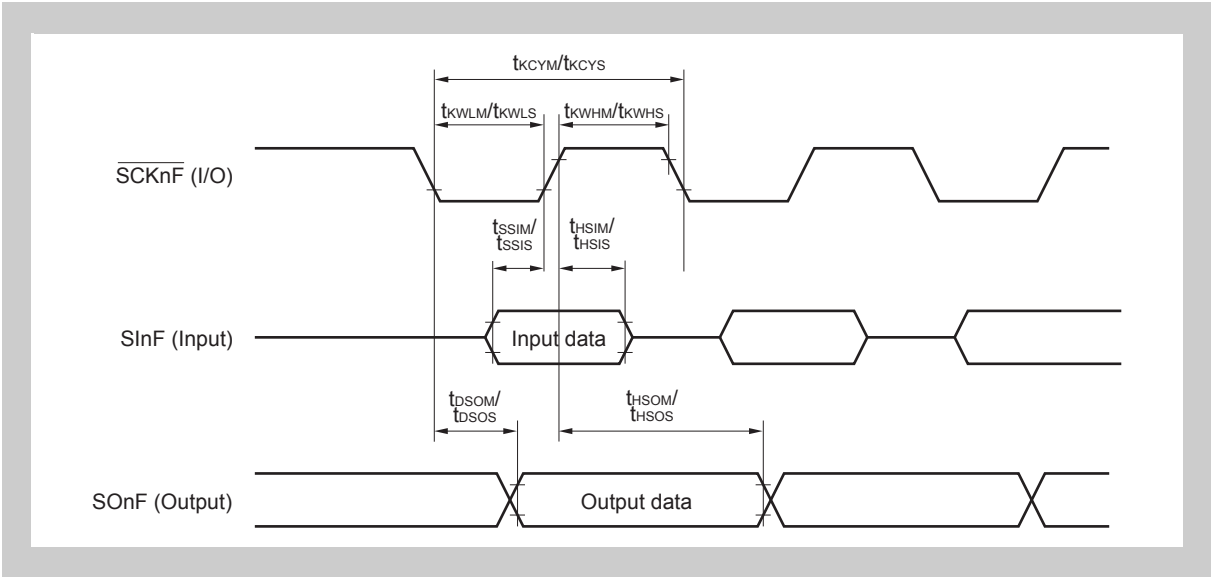


Figure 37-23 CSIH Timing

37.6.10 CSIG

(1) CSIG0 to CSIG4 (8.33 Mbps (f_{XX} = 200 MHz): While in the master mode)

(T_t = –40 to 100°C , IV_{DD} = $PLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = UV_{DD} = $OSCV_{DD}$ = DV_{DD} = 3.0 to 3.6 V, AV_{DD} = 3.0 to 3.6 V or AV_{DD} = 4.5 to 5.5 V, V_{SS} = AV_{SS} = $OSCV_{SS}$ = $PLLV_{SS}$ = DV_{SS} = 0 V)

Item	Symbol		Conditions	MIN.	MAX.	Unit
Clock source		t_{KCY}	Input	15		ns
$\overline{SCKn}$ cycle		t_{KCYM}	Output	120		ns
$\overline{SCKn}$ high-level width		t_{KWHM}	Output	(t_{KCYM}/2) –8.0		ns
$\overline{SCKn}$ low-level width		t_{KWLM}	Output	(t_{KCYM}/2) –8.0		ns
SIn setup time (to $\overline{SCKn}$↑)		t_{SSIM}	Input	26.0		ns
SIn setup time (to $\overline{SCKn}$↓)				26.0		ns
SIn hold time (from $\overline{SCKn}$↑)		t_{HSIM}	Input	26.0		ns
SIn hold time (from $\overline{SCKn}$↓)				26.0		ns
SOn output delay time (from $\overline{SCKn}$↑)		t_{DSOM}	Output		26	ns
SOn output delay time (from $\overline{SCKn}$↓)					26	ns
SOn output hold time (from $\overline{SCKn}$↑)		t_{HSOM}	Output	(t_{KCYM}/2) –10.0		ns
SOn output hold time (from $\overline{SCKn}$↓)				(t_{KCYM}/2) –10.0		ns
CSIn_RYI setup time (to $\overline{SCKn}$↑): CSIGnCTL1.CSIGnSIT = x		t_{SRDYI}	Input	((t_{KCYM}/2)/t_{KCY} –2) × t_{KCY}	(Min. value) + (t_{KCYM}/2) –26	ns
CSIn_RYI setup time (to $\overline{SCKn}$↓): CSIGnCTL1.CSIGnSIT = x				((t_{KCYM}/2)/t_{KCY} –2) × t_{KCY}	(Min. value) + (t_{KCYM}/2) –26	ns
CSIn_RYI high-level width		t_{WRYI}	Input	t_{KCY} + 10		ns

Note n = 0 to 4

CSSETUP: Settings for the CHBA0SP[3:0] bits of CHBACFG0 to CHBACFG7

CSHOLD: Settings for the CHBA0HD[3:0] bits of CHBACFG0 to CHBACFG7

(2) CSIG0 to CSIG4 (8.33 Mbps (f_{XX} = 200 MHz): While in the slave mode)

(T_t = –40 to 100°C , IV_{DD} = $PLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = UV_{DD} = $OSCV_{DD}$ = DV_{DD} = 3.0 to 3.6 V, AV_{DD} = 3.0 to 3.6 V or AV_{DD} = 4.5 to 5.5 V, V_{SS} = AV_{SS} = $OSCV_{SS}$ = $PLLV_{SS}$ = DV_{SS} = 0 V)

Item	Symbol		Conditions	MIN.	MAX.	Unit
Clock source		t_{KCY}	Input	15		ns
$\overline{SCKn}$ cycle		t_{KCYS}	Input	120		ns
$\overline{SCKn}$ high-level width		t_{KWHS}	Input	(t_{KCYS}/2) –8.0		ns
$\overline{SCKn}$ low-level width		t_{KWLS}	Input	(t_{KCYS}/2) –8.0		ns
SIn setup time (to $\overline{SCKn}$↑)		t_{SSIS}	Input	26.0		ns
SIn setup time (to $\overline{SCKn}$↓)				26.0		ns
SIn hold time (from $\overline{SCKn}$↑)		t_{HSIS}	Input	(t_{KCYS}/2) –8.0		ns
SIn hold time (from $\overline{SCKn}$↓)				(t_{KCYS}/2) –8.0		ns
SOn output delay time (from $\overline{SCKn}$↑)		t_{DSOS}	Output		26.0	ns
SOn output delay time (from $\overline{SCKn}$↓)					26.0	ns
SOn output hold time (from $\overline{SCKn}$↑)		t_{HSOS}	Output	(t_{KCYS}/2) –10.0		ns
SOn output hold time (from $\overline{SCKn}$↓)				(t_{KCYS}/2) –10.0		ns
CSIn_RYO output delay time (from $\overline{SCKn}$↑)		t_{SRDYIO}	Output		26.0	ns
CSIn_RYO output delay time (from $\overline{SCKn}$↓)					26.0	ns
CSIn_RYO output hold time (from $\overline{SCKn}$↑)		t_{HRDYIO}	Output	(t_{KCYS}/2) –10.0		ns
CSIn_RYO output hold time (from $\overline{SCKn}$↓)				(t_{KCYS}/2) –10.0		ns
CSIn_SSI setup time (to $\overline{SCKn}$↑)		t_{SSSIS}	Input	26.0		ns
CSIn_SSI setup time (to $\overline{SCKn}$↓)				26.0		ns
CSIn_SSI hold time (from $\overline{SCKn}$↑)		t_{HSSIS}	Input	26.0		ns
CSIn_SSI hold time (from $\overline{SCKn}$↓)				26.0		ns

Note n = 0 to 4

(3) CSIG5 (8.33 Mbps (f_{XX} = 200 MHz): While in the master mode)

(T_t = –40 to 100°C , IV_{DD} = $PLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = UV_{DD} = $OSCV_{DD}$ = DV_{DD} = 3.0 to 3.6 V, AV_{DD} = 3.0 to 3.6 V or AV_{DD} = 4.5 to 5.5 V, V_{SS} = AV_{SS} = $OSCV_{SS}$ = $PLLV_{SS}$ = DV_{SS} = 0 V)

Item	Symbol		Conditions	MIN.	MAX.	Unit
Clock source		t_{KCY}	Input	15		ns
$\overline{SCK5}$ cycle		t_{KCYM}	Output	120		ns
$\overline{SCK5}$ high-level width		t_{KWHM}	Output	$(t_{KCYM}/2)$ –13.0		ns
$\overline{SCK5}$ low-level width		t_{KWLM}	Output	$(t_{KCYM}/2)$ –13.0		ns
SI5 setup time (to $\overline{SCK5}$↑)		t_{SSIM}	Input	26.0		ns
SI5 setup time (to $\overline{SCK5}$↓)				26.0		ns
SI5 hold time (from $\overline{SCK5}$↑)		t_{HSIM}	Input	26.0		ns
SI5 hold time (from $\overline{SCK5}$↓)				26.0		ns
SO5 output delay time (from $\overline{SCK5}$↑)		t_{DSOM}	Output		26	ns
SO5 output delay time (from $\overline{SCK5}$↓)					26	ns
SO5 output hold time (from $\overline{SCK5}$↑)		t_{HSOM}	Output	$(t_{KCYM}/2)$ –10.0		ns
SO5 output hold time (from $\overline{SCK5}$↓)				$(t_{KCYM}/2)$ –10.0		ns
CSI5_RYI setup time (to $\overline{SCK5}$↑): CSIG0CTL1.CSIG0SIT = x		t_{SRDYI}	Input	$((t_{KCYM}/2)/t_{KCY}$ –2) × t_{KCY}	(Min. value) + $(t_{KCYM}/2)$ –26	ns
CSI5_RYI setup time (to $\overline{SCK5}$↓): CSIG0CTL1.CSIG0SIT = x				$((t_{KCYM}/2)/t_{KCY}$ –2) × t_{KCY}	(Min. value) + $(t_{KCYM}/2)$ –26	ns
CSI5_RYI high-level width		t_{WRYI}	Input	t_{KCY} + 10		ns

(4) CSIG5 (8.33 Mbps (f_{XX} = 200 MHz): While in the slave mode)

(T_t = –40 to 100°C , IV_{DD} = $PLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = UV_{DD} = $OSCV_{DD}$ = DV_{DD} = 3.0 to 3.6 V, AV_{DD} = 3.0 to 3.6 V or AV_{DD} = 4.5 to 5.5 V, V_{SS} = AV_{SS} = $OSCV_{SS}$ = $PLLV_{SS}$ = DV_{SS} = 0 V)

Item	Symbol		Conditions	MIN.	MAX.	Unit
Clock source		t_{KCY}	Input	15		ns
$\overline{SCK5}$ cycle		t_{KCYS}	Input	120		ns
$\overline{SCK5}$ high-level width		t_{KWHS}	Input	(t_{KCYS}/2) –8.0		ns
$\overline{SCK5}$ low-level width		t_{KWLS}	Input	(t_{KCYS}/2) –8.0		ns
SI5 setup time (to $\overline{SCK5}$↑)		t_{SSIS}	Input	26.0		ns
SI5 setup time (to $\overline{SCK5}$↓)				26.0		ns
SI5 hold time (from $\overline{SCK5}$↑)		t_{HSIS}	Input	(t_{KCYS}/2) –8.0		ns
SI5 hold time (from $\overline{SCK5}$↓)				(t_{KCYS}/2) –8.0		ns
SO5 output delay time (from $\overline{SCK5}$↑)		t_{DSOS}	Output		26.0	ns
SO5 output delay time (from $\overline{SCK5}$↓)					26.0	ns
SO5 output hold time (from $\overline{SCK5}$↑)		t_{HSOS}	Output	(t_{KCYS}/2) –10.0		ns
SO5 output hold time (from $\overline{SCK5}$↓)				(t_{KCYS}/2) –10.0		ns
CSI5_RYO output delay time (from $\overline{SCK5}$↑)		t_{SRDYIO}	Output		26.0	ns
CSI5_RYO output delay time (from $\overline{SCK5}$↓)					26.0	ns
CSI5_RYO output hold time (from $\overline{SCK5}$↑)		t_{HRDYIO}	Output	(t_{KCYS}/2) –10.0		ns
CSI5_RYO output hold time (from $\overline{SCK5}$↓)				(t_{KCYS}/2) –10.0		ns
CSI5_SSI setup time (to $\overline{SCK5}$↑)		t_{SSSIS}	Input	26.0		ns
CSI5_SSI setup time (to $\overline{SCK5}$↓)				26.0		ns
CSI5_SSI hold time (from $\overline{SCK5}$↑)		t_{HSSIS}	Input	26.0		ns
CSI5_SSI hold time (from $\overline{SCK5}$↓)				26.0		ns

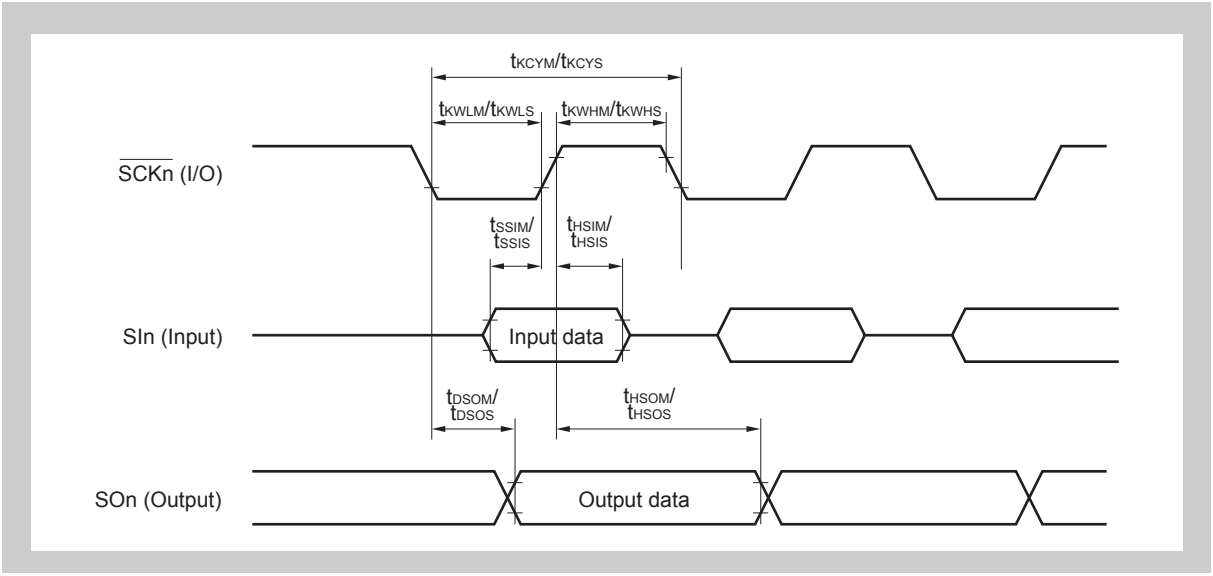
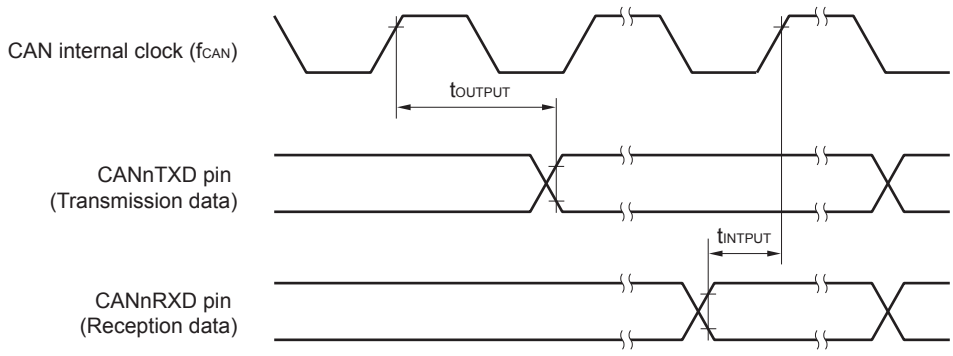


Figure 37-24 CSIG Timing

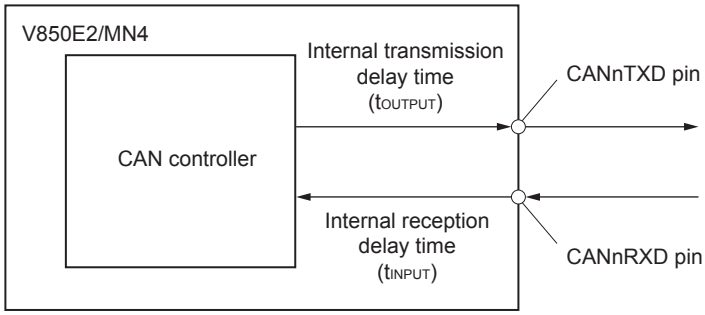
37.6.11 CAN

(T_t = –40 to 100°C , IV_{DD} = $PLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = UV_{DD} = $OSCV_{DD}$ = DV_{DD} = 3.0 to 3.6 V, AV_{DD} = 3.0 to 3.6 V or AV_{DD} = 4.5 to 5.5 V, V_{SS} = AV_{SS} = $OSCV_{SS}$ = $PLLV_{SS}$ = DV_{SS} = 0 V)

Item	Symbol		Conditions	MIN.	MAX.	Unit
Internal delay time		t_{NODE}			75	ns



Internal delay time (t_{NODE}) =
Internal transmission delay time (t_{OUTPUT}) + Internal reception delay time (t_{INPUT})



37.6.12 I^2C

(T_t = –40 to 100°C , IV_{DD} = $PLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = UV_{DD} = $OSCV_{DD}$ = DV_{DD} = 3.0 to 3.6 V, AV_{DD} = 3.0 to 3.6 V or AV_{DD} = 4.5 to 5.5 V, V_{SS} = AV_{SS} = $OSCV_{SS}$ = $PLLV_{SS}$ = DV_{SS} = 0 V)

Item		Symbol	Conditions	Standard Mode		High-Speed Mode		Unit
				MIN.	MAX.	MIN.	MAX.	
SCL clock frequency		f_{CLK}		0	100	0	400	kHz
Bus-free time between the stop condition and start condition		t_{BUF}		4.7		1.3		μs
Hold time[a]		$t_{HD:STA}$		4.0		0.6		μs
SCL clock low-level width		t_{LOW}		4.7		1.3		μs
SCL clock high-level width		t_{HIGH}		4.0		0.6		μs
Setup time for the start and restart conditions		$t_{SU:STA}$		4.7		0.6		μs
Data hold time	For a CBUS compatible master	$t_{HD:DAT}$		5.0				μs
	For an IIC bus			0		0	0.9	μs
Data setup time		$t_{SU:DAT}$		250		100		ns
SDA and SCL signal rise time		t_R			1000	20 + 0.1C_b	300	ns
SDA and SCL signal fall time		t_F			300	20 + 0.1C_b	300	ns
Stop condition setup time		$t_{SU:STO}$		4.0		0.6		μs
Pulse width of spike suppressed by input filter		t_{SP}				0	50	ns
Capacitance load of each bus line		C_b			400		400	pF

a) When there is a start condition/restart condition, the first clock pulse is generated after this period (while in the master mode).

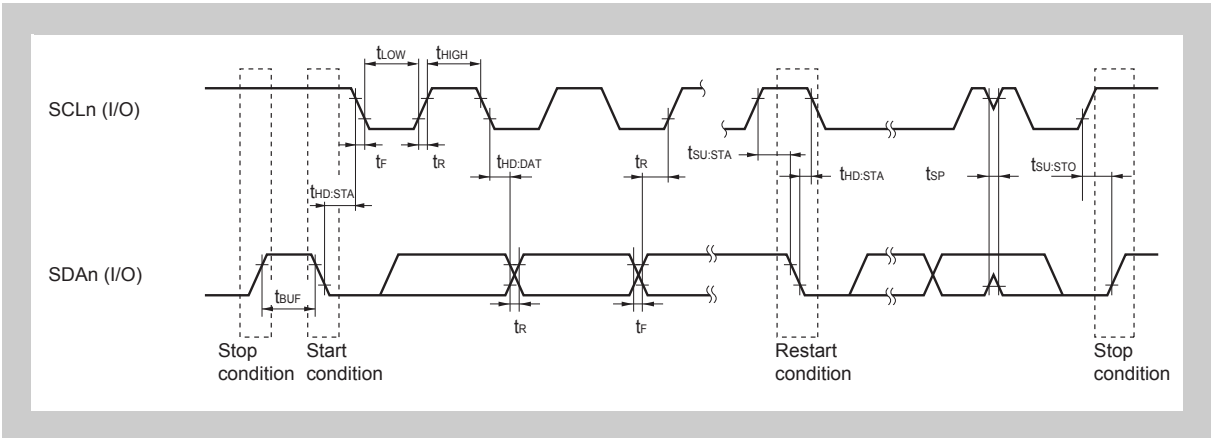


Figure 37-25 I^2C Timing

37.6.13 Ethernet controller

(T_t = –40 to 100°C , IV_{DD} = $PLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = UV_{DD} = $OSCV_{DD}$ = DV_{DD} = 3.0 to 3.6 V, AV_{DD} = 3.0 to 3.6 V or AV_{DD} = 4.5 to 5.5 V, V_{SS} = AV_{SS} = $OSCV_{SS}$ = $PLLV_{SS}$ = DV_{SS} = 0 V, output pin load capacitance: CL = 30 pF)

(1) Transmission interface

Item	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
ETH_TXD[3:0] delay time (from ETH_TXCLK↑)	t_{DTKTD}		0		25	ns
ETH_TXEN and ETH_TXER delay time (from TXCLK↑)	t_{DTKTE}		0		25	ns
ETH_TXCLK clock cycle	t_{CYTK}		40			ns
ETH_TXCLK high-level width	t_{TKH}		$0.4t_{CYTK}$		$0.6t_{CYTK}$	ns
ETH_TXCLK low-level width	t_{TKL}		$0.4t_{CYTK}$		$0.6t_{CYTK}$	ns
ETH_CRS hold time (from TXCLK↑)	t_{HCRS}		5			ns
ETH_CRS setup time (from TXCLK↑)	t_{SCRS}		5			ns
ETH_COL hold time (from TXCLK↑)	t_{HCOL}		5			ns
ETH_COL setup time (from TXCLK↑)	t_{SCOL}		5			ns

(2) Reception interface

Item	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
ETH_RXD[3:0] hold time (from ETH_RXCLK↑)	t_{HRKRD}		5			ns
ETH_RXD[3:0] setup time (from ETH_RXCLK↑)	t_{SDRK}		5			ns
ETH_RXDV and ETH_RXER hold time (from ETH_RXCLK↑)	t_{HRKRV}		5			ns
ETH_RXDV and ETH_RXER setup time (from ETH_RXCLK↑)	t_{SRVRK}		5			ns
ETH_RXCLK clock cycle	t_{CYKR}		40			ns
ETH_RXCLK high-level width	t_{RKH}		$0.4t_{CYTK}$		$0.6t_{CYTK}$	ns
ETH_RXCLK low-level width	t_{RKL}		$0.4t_{CYTK}$		$0.6t_{CYTK}$	ns

(3) Management interface

Item	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
ETH_MDC clock cycle	t_{CYMDC}		400			ns
ETH_MDIO delay time (from ETH_MDC↑)	t_{DMCMD}		0		300	ns
MDI setup time (to ETH_MDC↑)	t_{SMDMC}		50			ns
MDI hold time (from ETH_MDC↑)	t_{HMCMD}		50			ns

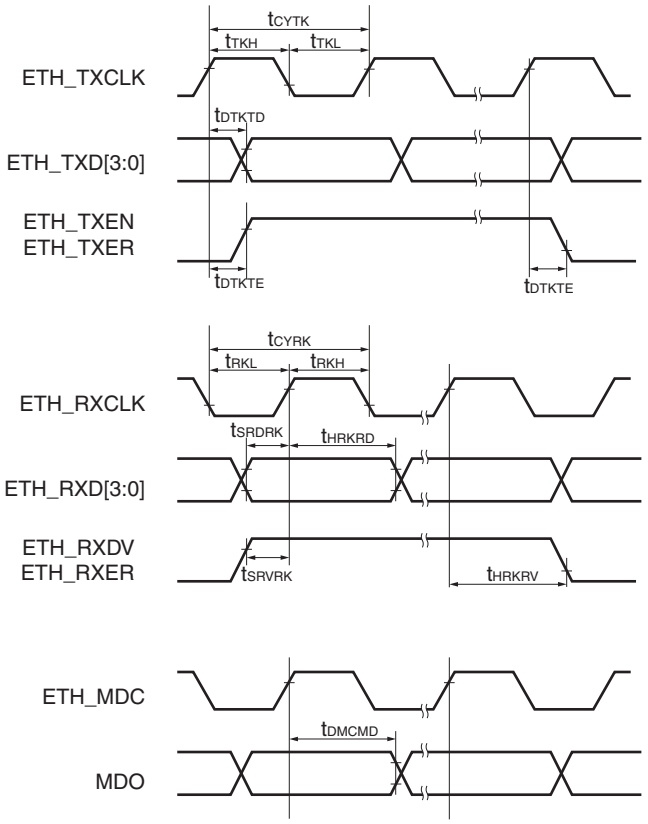


Figure 37-26 Ethernet Controller Timing

37.6.14 A/D converter (3.3 V, 10 bits)

(1) Normal operation (ANI00 to ANI11)

(T_t = –40 to 100°C , IV_{DD} = $PLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = UV_{DD} = $OSCV_{DD}$ = DV_{DD} = 3.0 to 3.6 V, AV_{DD} = AV_{REFP} = 3.0 to 3.6 V, V_{SS} = AV_{SS} = AV_{REFM} = $OSCV_{SS}$ = $PLLV_{SS}$ = DV_{SS} = 0 V)

Item	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution				10		Bits
Overall error[a]	TOE				±4	LSB
Quantization error	–				1/2	LSB
Conversion time[b]	t_{CONV}	ADCLK: 12.00 MHz (PCLK = 4MHz) Amplifier power: Off	1.5		5	μs
		ADCLK: 12.00 MHz (PCLK = 4MHz) Amplifier power: On	1.8		5	μs
Sampling time	t_{SAMP}		6.5			AD clocks
Integral nonlinear error	INL				±3	LSB
Differential nonlinear error	DNL				±2.5	LSB
Zero-scale error[a]	ZSE				±3	LSB
Full-scale error[a]	FSE				±3	LSB
AV_{DD} power supply current	AI_{DD}	Amplifier power: Off		9.1	20	mA
		Amplifier power: On		9.9	20	mA
Analog input voltage	V_{AIN}	ANI00 to ANI11	AV_{REFM}		AV_{REFP}	V
Reference power supply current	AI_{REFPn}			500		μA
ADTRGn high-level width (n = 0 or 1)	t_{WAIH1}	When using DNF	(number of specified elimination clocks –1)/(PCLK) + 10			ns
	t_{WAIH2}	When using ANF	500			ns
ADTRGn high-level width (n = 0 or 1)	t_{WAIL1}	When using DNF	(number of specified elimination clocks –1)/(PCLK) + 10			ns
	t_{WAIL2}	When using ANF	500			ns

a) The quantization error (±0.5 LSB) is not included.

b) This is the conversion time for only the analog portion. The conversion time specified for the ADCAnCTL1.ADCAnFR[3:0] bits is the value that results when the time it takes to transfer to the A/D controller is added.

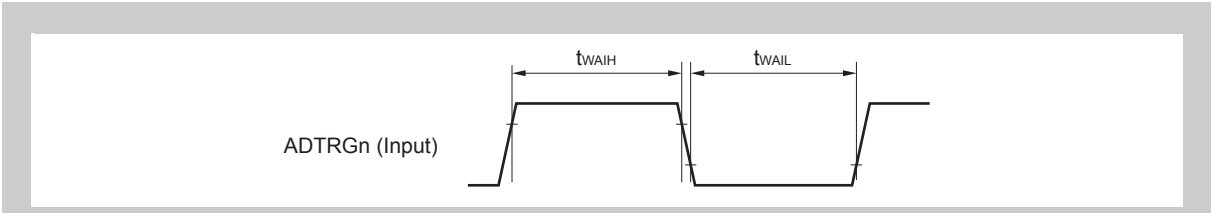


Figure 37-27 A/D Converter Timing (3.3 V, 10 bits)

(2) When using the channel sample & hold circuit (ANI00 to ANI05)

(T_t = –40 to 100°C , IV_{DD} = $PLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = UV_{DD} = $OSCV_{DD}$ = DV_{DD} = 3.0 to 3.6 V, AV_{DD} = AV_{REFP} = 3.0 to 3.6 V, V_{SS} = AV_{SS} = AV_{REFM} = $OSCV_{SS}$ = $PLLV_{SS}$ = DV_{SS} = 0 V)

Item	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution				10		Bits
Overall error[a]	TOE	V_{AIN} = 0.2 V to AV_{REFP} – 0.2 V			±7.5	LSB
Quantization error	–				1/2	LSB
Conversion time[b]	t_{CONV}	ADCLK: 12.00 MHz (PCLK = 4MHz) Amplifier power: On	1.833		5	μs
Sample hold function hold specification	t_{SAMPH}				10	μs
Sampling time	t_{SAMP}		6.5			AD clocks
Integral nonlinear error	INL	V_{AIN} = 0.2 V to AV_{REFP} – 0.2 V			±6	LSB
Differential nonlinear error	DNL	V_{AIN} = 0.2 V to AV_{REFP} – 0.2 V			±4	LSB
AV_{DD} power supply current	AI_{DD}	Amplifier power: On		9.9	20	mA
Analog input voltage	V_{AIN}	ANI00 to ANI11	AV_{REFM}		AV_{REFP}	V
Reference power supply current	AI_{REFPn}			650		μA
ADTRGn high-level width (n = 0 or 1)	t_{WAIH1}	When using DNF	(number of specified elimination clocks –1)/(PCLK) + 10			ns
	t_{WAIH2}	When using ANF	500			ns
ADTRGn high-level width (n = 0 or 1)	t_{WAIL1}	When using DNF	(number of specified elimination clocks –1)/(PCLK) + 10			ns
	t_{WAIL2}	When using ANF	500			ns

a) The quantization error (±0.5 LSB) is not included.

b) This is the conversion time for only the analog portion. The conversion time specified for the ADCAnCTL1.ADCAnFR[3:0] bits is the value that results when the time it takes to transfer to the A/D controller is added.

Caution Make sure to turn on the amplifier power when using the channel S/H circuit.

37.6.15 A/D converter (5.0 V, 12 bits)

(1) Normal operation (ANI00 to ANI11)

(T_t = –40 to 100°C , IV_{DD} = $PLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = UV_{DD} = $OSCV_{DD}$ = DV_{DD} = 3.0 to 3.6 V, AV_{DD} = AV_{REFP} = 4.5 to 5.5 V, V_{SS} = AV_{SS} = AV_{REFM} = $OSCV_{SS}$ = $PLLV_{SS}$ = DV_{SS} = 0 V)

Item	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution				12		Bits
Overall error[a]	TOE				±6.5	LSB
Quantization error	–				1/2	LSB
Conversion time[b]	t_{CONV}	ADCLK: 13.33 MHz Amplifier power: Off	1.5		5	μs
		ADCLK: 13.33 MHz Amplifier power: On	1.8		5	μs
Sampling time	t_{SAMP}		6.5			AD clocks
Integral nonlinear error	INL				±5	LSB
Differential nonlinear error	DNL				±3	LSB
Zero-scale error[a]	ZSE				±5.5	LSB
Full-scale error[a]	FSE				±5.5	LSB
AV_{DD} power supply current	AI_{DD}	Amplifier power: Off		14.3	30	mA
		Amplifier power: On		14.8	30	mA
Analog input voltage	V_{IAN}	ANI00 to ANI11	AV_{REFM}		AV_{REFP}	V
Reference power supply current	AI_{REFPn}			650		μA
ADTRGn high-level width (n = 0 or 1)	t_{WAIH1}	When using DNF	(number of specified elimination clocks –1)/(PCLK) + 10			ns
	t_{WAIH2}	When using ANF	500			ns
ADTRGn high-level width (n = 0 or 1)	t_{WAIL1}	When using DNF	(number of specified elimination clocks –1)/(PCLK) + 10			ns
	t_{WAIL2}	When using ANF	500			ns

a) The quantization error (±0.5 LSB) is not included.

b) This is the conversion time for only the analog portion. The conversion time specified for the ADCAnCTL1.ADCAnFR[3:0] bits is the value that results when the time it takes to transfer to the A/D controller is added.

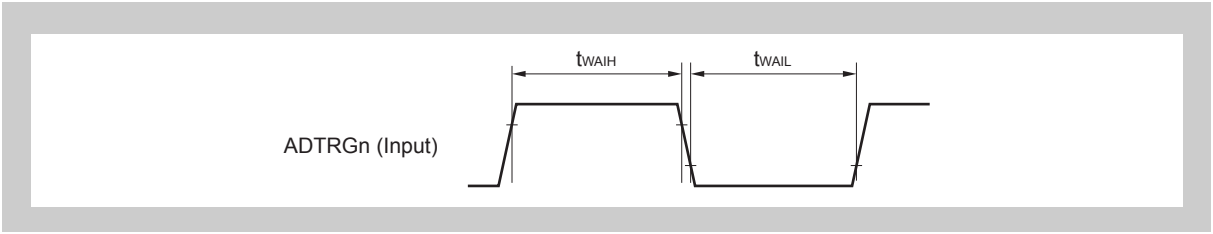


Figure 37-28 A/D Converter Timing (5.0 V, 12 bits)

(2) When using the channel sample & hold circuit (ANI00 to ANI05)

(T_t = –40 to 100°C , IV_{DD} = $PLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = UV_{DD} = $OSCV_{DD}$ = DV_{DD} = 3.0 to 3.6 V, AV_{DD} = AV_{REFP} = 4.5 to 5.5 V, V_{SS} = AV_{SS} = AV_{REFM} = $OSCV_{SS}$ = $PLLV_{SS}$ = DV_{SS} = 0 V)

Item	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Resolution				12		Bits
Overall error[a]	TOE	V_{IAN} = 0.2 V to AV_{REFP} – 0.2 V			±7.5	LSB
Quantization error	–				1/2	LSB
Conversion time[b]	t_{CONV}	ADCLK: 13.33 MHz Amplifier power: Off	1.8		5	μs
Sample hold function hold specification	t_{SAMPH}				10	μs
Sampling time	t_{SAMP}		6.5			AD clocks
Integral nonlinear error	INL	V_{AIN} = 0.2 V to AV_{REFP} – 0.2 V			±6	LSB
Differential nonlinear error	DNL	V_{AIN} = 0.2 V to AV_{REFP} – 0.2 V			±4	LSB
AV_{DD} power supply current	AI_{DD}	Amplifier power: On		14.8	30	mA
Analog input voltage	V_{AIN}	ANI00 to ANI11	AV_{REFM}		AV_{REFP}	V
Reference power supply current	AI_{REFPn}			650		μA
ADTRGn high-level width (n = 0 or 1)	t_{WAIH1}	When using DNF	(number of specified elimination clocks –1)/(PCLK) + 10			ns
	t_{WAIH2}	When using ANF	500			ns
ADTRGn high-level width (n = 0 or 1)	t_{WAIL1}	When using DNF	(number of specified elimination clocks –1)/(PCLK) + 10			ns
	t_{WAIL2}	When using ANF	500			ns

a) The quantization error (±0.5 LSB) is not included.

b) This is the conversion time for only the analog portion. The conversion time specified for the ADCAnCTL1.ADCAnFR[3:0] bits is the value that results when the time it takes to transfer to the A/D controller is added.

Caution Make sure to turn on the amplifier power when using the channel S/H circuit.

37.7 Sequences for Turning the Power Supply On or Off

There are recommended procedures for turning the power supply on or off.

(T_t = –40 to 100°C , IV_{DD} = $PLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = UV_{DD} = $OSCV_{DD}$ = DV_{DD} = 3.0 to 3.6 V, AV_{DD} = 3.0 to 3.6 V or AV_{DD} = 4.5 to 5.5 V, V_{SS} = AV_{SS} = $OSCV_{SS}$ = $PLLV_{SS}$ = DV_{SS} = 0 V)

Item	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Delay from when the internal power supply rises until the external power supply rises Rising delay time	t_{DVIE}		0			s
Delay from when the external power supply rises until the ADC power supply rises Rising delay time	t_{DVEA}		0			s
Delay from when the ADC power supply rises until $\overline{RESET}$↑ Delay time	t_{DVAR}		t_{OSC} + 500			ns
Delay from when $\overline{RESET}$↓ until the ADC power supply falls Delay time	t_{DVRA}		500			ns
Delay from when the ADC power supply falls until the external power supply falls Falling delay time	t_{DVAE}		0			ns
Delay from when the external power supply falls until the internal power supply falls Falling delay time	t_{DVEI}		0			ns

Note t_{OSC}: Main clock oscillation stabilization time (differs depending on the used oscillator)

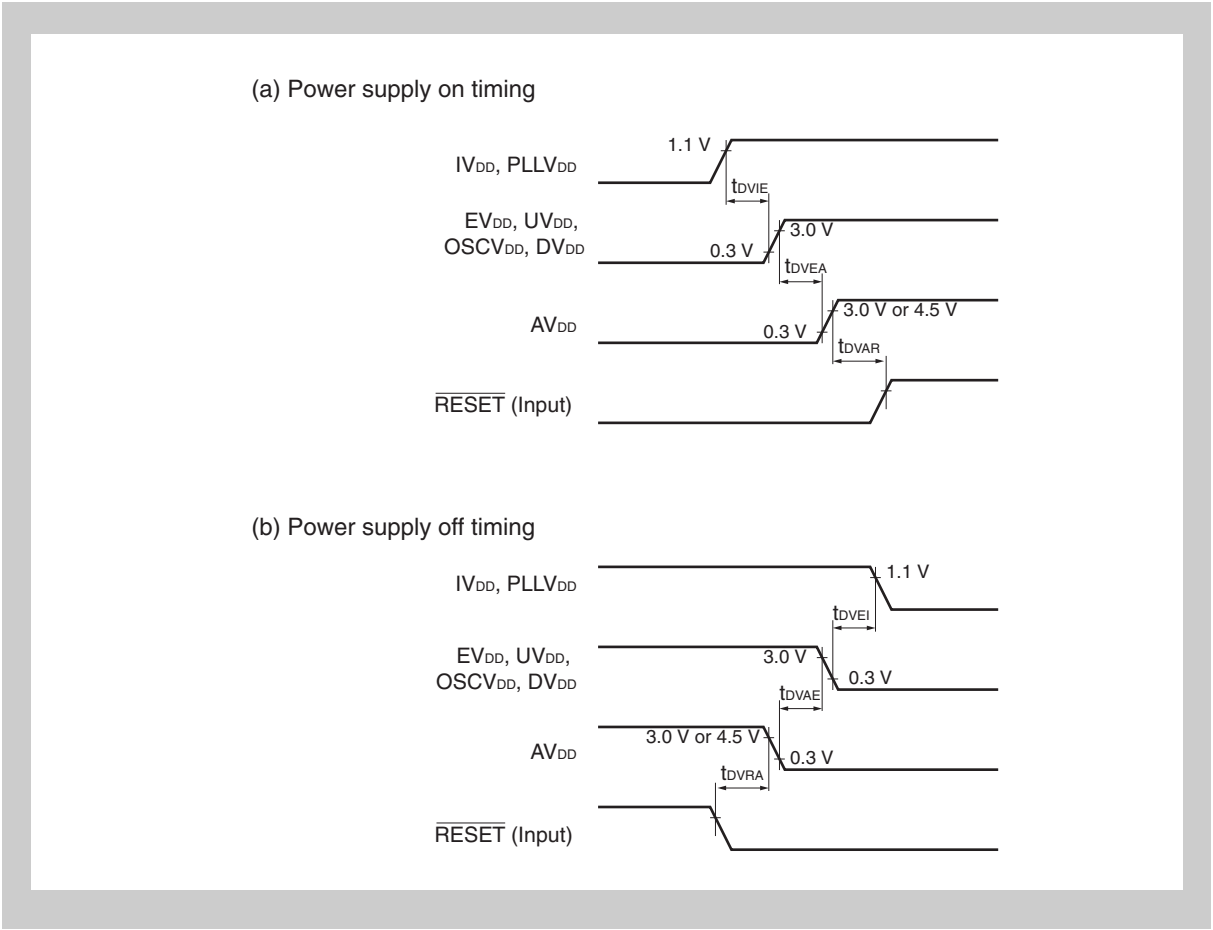


Figure 37-29 Timing of the Sequences for Turning the Power Supply On and Off

Caution The V850E2/MN4 has two power supply pins: IV_{DD} for internal units and EV_{DD} for external pins. The I/O status of the pins might be undefined outside the guaranteed operation range.

37.8 Flash Memory Programming Mode

(1) Basic characteristics

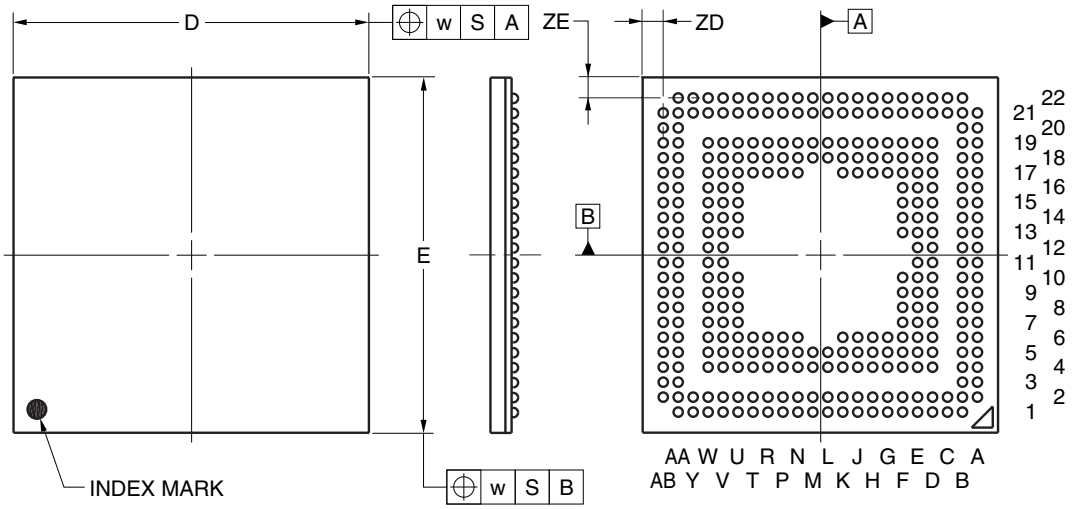
(T_t = –40 to 100°C , IV_{DD} = $PLLV_{DD}$ = 1.1 to 1.3 V, EV_{DD} = UV_{DD} = $OSCV_{DD}$ = DV_{DD} = 3.0 to 3.6 V, AV_{DD} = 3.0 to 3.6 V or AV_{DD} = 4.5 to 5.5 V, V_{SS} = AV_{SS} = $OSCV_{SS}$ = $PLLV_{SS}$ = DV_{SS} = 0 V)

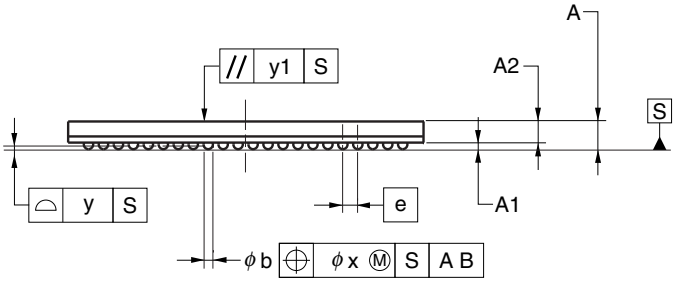
Item	Symbol	Conditions	MIN.	TYP.	MAX.	Unit
Number of rewrites	C_{ERWR}	One erasure + one post-erasure write = one rewrite [a]	100	100	100	Times/block

a) When initially writing to a shipped product, both erasures followed by writes and writes alone are counted as rewrites. Example P: Write, E: Erasure
Shipped product → E → P → E → P → E → P: Three rewrites
Shipped product → E → P → E → P → E → P: Three rewrites

Chapter 38 Package Drawing

304-PIN PLASTIC FBGA (19x19)





(UNIT:mm)

ITEM	DIMENSIONS
D	19.00±0.10
E	19.00±0.10
w	0.20
e	0.80
A	1.46±0.10
A1	0.35±0.06
A2	1.11
b	$0.50^{+0.05}_{-0.10}$
x	0.08
y	0.10
y1	0.20
ZD	1.10
ZE	1.10

P304F1-80-HN6

Chapter 39 Recomended Soldering Conditions

The V850E2/MN4 should be soldered and mounted under the following recommended conditions.

For technical information, see the following website.

Semiconductor Device Mount Manual (http://www.renesas.com/products/package/manual/index.jsp)

Table 36-1. Surface Mounting Type Soldering Conditions

(1) μPD70F3510F1-HN6-A
μPD70F3512F1-HN6-A
μPD70F3514F1-HN6-A
μPD70F3515F1-HN6-A

Soldering Method	Soldering Conditions	Recommended Condition Symbol
Infrared reflow	Package peak temperature: 250°C, Time: 60 seconds max. (at 220°C or higher), Count: Three times or less, Exposure limit: 7 days[1] (after that, prebake at 125°C for 20 to 72 hours)	IR50-207-3

1) After opening the dry pack, store it at 25°C or less and 65% RH or less for the allowable storage period.

Caution Do not use different soldering methods together (except for partial heating).

Notes
1. The V850E2/MN4 is a lead-free product.
2. For soldering methods and conditions other than those recommended above, please contact a Renesas Electronics sales representative.

(1/6)

Rev.	Date	Description	
		Page	Summary
0.03	February 28, 2011	–	First edition issued
1.00	September 28, 2011	p.27	Modification of ***1.2 Features***
		pp.223 to 226	Addition of ***8.3.6 Example port settings***
		p.313	Modification of Cautions to ***8.5.2 (2) (a) Digital filter function***
		p.500	Addition of Caution 6 to ***11.2.8 SDRAM configuration register (SDCR)***
		p.508	Addition of ***Table 11-16 List of Address Bus Connecting to SDRAM***
		p.598	Modification of Caution to***12.15.1 (1) Equations***
		p.921	Modification of Caution to***13.14.1 (1) Equations***
		p.1193	Modification of ***20.5.1 Transmission interrupt request INTUAJnTIT***
		p.1194	Modification of ***Figure 20-2 Transmission interrupt request timing***
		p.1195	Modification of ***20.5.2 Reception interrupt request INTUAJnTIR***
		p.1195	Modification of ***Figure 20-3 Reception interrupt request timing***
		p.1231	Modification of ***21.3.3 Serial clock selection Baud rate limits***
		p.1529	Addition of Caution to ***24.2.2 Configuration***
		p.1547	Modification of ***24.5.1 (3) FCNnGMABCTL - FCNn global automatic block transmission control*** register
		p.1554	Modification of Caution and Notes to ***24.5.2 (2) FCNnCMCLCTL - FCNn module control register2 Features***
		p.1555	Modification of Note to ***24.5.2 (2) FCNnCMCLCTL - FCNn module control register2 Features***
		p.1626	Modification of ***Figure 24-14 Re-initialization without using the software reset function***
		p.1626	Addition of Caution to ***Figure 24-14 Re-initialization without using the software reset function***
		p.1638	Modification of ***1Figure 24-25 Transmission abort processing (normal operation mode with ABT) - Repeat option for aborted message***
		p.1640	Addition of ***ABT transmission request abort processing (in normal operation mode with ABT) (2)***
		p.1640	Addition of Caution to ***ABT transmission request abort processing (in normal operation mode with ABT) (2)***
		p.1641	Modification of ***Figure 24-28 ABT transmission request abort processing (normal operation mode with ABT) with transmission completely finished flag***
		p.1774	Modification of ***Figure 28-3 SRAM read cycle (external wait states inserted)***
		p.1775	Modification of ***Figure 28-4 SRAM write cycle (with no wait)***
		p.2058	Modification of ***31.6.1 PCI bridge registers***
		p.2059	Modification of ***31.6.1 (1) PCI command register (USHA0PCICMD)***
		p.2067	Modification of ***31.7.1 OHCI host controller features***

Rev.	Date	Description	
		Page	Summary
1.00	September 28, 2011	p.2319	Modification of ***34.5.1 (3) FLMDPS: FLMD protection error status registers***
		p.2340	Modification of ***37.5.2 Power supply current***
		p.2354	Modification of ***37.6.2 (8) Primary memory controller, bus hold timing***
		p.2359	Modification of ***37.6.2 (11) Secondary memory controller, SDRAM read timing***
		p.2361	Modification of ***37.6.2 (13) Secondary memory controller, SRAM S_DMA access timing***
		p.2362	Modification of ***37.6.2 (14) Secondary memory controller, bus hold timing***
		p.2363	Modification of ***37.6.3 Clock timing***
		p.2382	Modification of ***37.6.14 (1) Normal operation (ANI00 to ANI11)***
		p.2383	Modification of ***37.6.14 (2) When using the channel sample & hold circuit (ANI00 to ANI05)***
		p.2384	Modification of ***37.6.15 (1) Normal operation (ANI00 to ANI11)***
		p.2385	Modification of ***37.6.15 (2) When using the channel sample & hold circuit (ANI00 to ANI05)***
2.00	July 25, 2012	p.72	Addition of Caution to ***2.3 Duplication of Pin Functions.***
		p195	Modification of Caution to ***8.2.3 Pin data input/output.***
		p.209	Modification of Caution to ***8.3.3 (1) PBDCn - Port bi-direction control register.***
		p.485	Modification of ***Table 10-60 DTSSELm register contents.***
		p.501	Modification of ***Table 11-10 SDCR register contents.***
		p.513	Modification of ***11.4.4 SDRAM setting function.***
		p.513	Modification of ***Figure 11-4 Initial register settings - register write operation flow.***
		p.539	Modification of ***Table 11-18 Data flow during byte access (little endian)***.
		p.542	Modification of ***Table 11-20 Data flow during halfword access (little endian)***.
		p.543	Modification of ***Table 11-21 Data flow during halfword access (big endian).***
		p.544	Modification of ***Table 11-22 Data flow during word access (little endian).***
		p.548	Modification of ***Table 11-23 Data flow during word access (big endian).***
		p.1193	Addition of Caution to ***20.5.1 Transmission interrupt request INTUAJnTIT.***
		p.1194	Addition of Caution to ***20.5.2 Reception interrupt request INTUAJnTIR.***
		p.1269	Modification of ***Table 22-5 Maximum CSIHn transfer speeds (baud rates).***
		p.1298	Modification of ***Table 22-10 Generation of CSIHnTIC in job mode.***
		p.1319	Modification of ***Table 22-17 CSIHnCTL0 register contents.***
		p.1524	Addition of Caution to ***24.1 FCN Features of V850E2/MN4.***
		p.1526	Modification of ***24.2 Features.***
		p.1527	Modification of ***Table 24-7 Overview of functions.***
		p.1657	Modification of ***25.1 V850E2/MN4 Features.***
		p.1692	Modification of ***25.3.14 Channel sample & hold circuit function (product dependent).***
		p.1757	Modification of ***28.1.1 (2) SDRAM connection function***

Rev.	Date	Description	
		Page	Summary
2.00	July 25, 2012	p.1868	Addition of Caution to ***Chapter 30 USB Function Controller (USBF).***
		p.1872	Modification of ***Figure 30-1 Block diagram for USB function controller.***
		p.1920	Modification of Caution to ***30.7.2 (12) INT status 1 register (USFA0IS1).***
		p.1930	Modification of Address to ***30.7.2 (19) INT mask 3 register (USFA0IM3).***
		p.1947	Modification of Address to ***30.7.2 (34) Mode status register (USFA0MODS).***
		p.1953	Modification of Address to ***(40) Endpoint3 interface mapping register (USFA0E3IM).***
		p.2000	Modification of ***Table 30-86 USFA0EPCINT register contents.***
		p.2050	Addition of Caution to ***Chapter 31 USB Host Controller (USBH).***
		p.2056	Modification of ***Figure 31-2 Block diagram of USB host controller.***
		p.2061	Modification of ***31.6.1 PCI bridge registers.***
		p.2066	Addition of ***31.6.1 (5) Error register 1 (USHA0ERR1).***
		p.2068	Addition of ***31.6.1(7) PCI control register 1 (USHA0PCICTL_H)***
		p.2069	Modification of ***31.6.1 (8) PCI BAR enable register (USHA0PCIBARE).***
		p.2070	Modification of ***Table 31-15 USHA0CNFIGADDR register contents.***
		p.2078	Modification of ***31.7.2 (4) OHCI Base Address register (Offset: 10_H).***
		p.2091	Modification of ***31.7.3 (4) HcInterruptStatus register (Offset: $0C_H$).***
		p.2122	Addition of Caution to ***Table 32-3 Ethernet controller interrupt requests.***
		p.2126	Modification of ***Table 32-4 Interrupt requests.***
		p.2130	Addition of ***32.3.1 Software reset.***
		p.2131	Modification of ***Table 32-5 MAC control register list.***
		p.2133	Modification of ***Table 32-7 FIFO controller register list.***
		p.2134	Modification of ***Table 32-8 DMAC for Ethernet controller register list.***
		p.2134	Modification of ***Table 32-9 DMAC control for transmit checksum register list.***
		p.2135	Modification of Initial value, Attribute to ***32.4.1 (1) ETHA0MACC1 - MAC setting register.***
		p.2137	Modification of Initial value, Attribute to ***32.4.1 (2) ETHA0MACC2 - MAC setting register.***
		p.2138	Modification of Initial value, Attribute to ***32.4.1 (3) ETHA0IPGT - Back-to-back IPG register.***
		p.2139	Modification of Initial value, Attribute to ***32.4.1 (4) ETHA0IPGR - Non back-to-back IPG register***
		p.2140	Modification of Initial value, Attribute to ***32.4.1 (5) ETHA0CLRT - Collision register.***
		p.2141	Modification of Initial value, Attribute to ***32.4.1 (6) ETHA0LMAX - Maximum packet length register.***
		p.2142	Modification of Initial value, Attribute to ***32.4.1 (7) ETHA0LSA1 - Station address register 1***
		p.2144	Modification of Initial value, Attribute to ***32.4.1 (9) ETHA0PTVR - Pause timer value read register.***
		p.2145	Modification of Initial value, Attribute to ***32.4.1 (10) ETHA0VLTP - VLAN type register.***

Rev.	Date	Description	
		Page	Summary
2.00	July 25, 2012	p.2146	Modification of Initial value, Attribute to ***32.4.1 (11) ETHA0MIIC - Serial management interface configuration register.***
		p.2148	Modification of Initial value, Attribute to ***32.4.1 (12) ETHA0MCMD - MII command register.***
		p.2149	Modification of Initial value, Attribute to ***32.4.1 (13) ETHA0MADR - MII address register.***
		p.2150	Modification of Initial value, Attribute to ***32.4.1 (14) ETHA0MWTD - MII write data register.***
		p.2151	Modification of Initial value, Attribute to ***32.4.1 (15) ETHA0MRDD - MII read data register.***
		p.2152	Modification of Initial value, Attribute to ***32.4.1 (16) ETHA0MIND - MII indicator register***
		p.2153	Modification of Initial value, Attribute to ***32.4.1 (17) ETHA0AFR - Address filter register.***
		p.2156	Modification of Initial value, Attribute to ***32.4.1(20) ETHA0CAR1 - Carry register 1.***
		p.2158	Modification of Initial value, Attribute to ***32.4.1 (21) ETHA0CAR2 - Carry register 2.***
		p.2160	Modification of Initial value, Attribute to ***32.4.1 (22) ETHA0CAM1 - Carry mask register 1.***
		p.2162	Modification of Initial value, Attribute to ***32.4.1 (23) ETHA0CAM2 - Carry mask register 2.***
		p.2203	Modification of Initial value, Attribute to ***32.4.3 (1) ETHA0MFFCONT - FIFO controller control register.***
		p.2205	Modification of Initial value, Attribute to ***32.4.3 (2) ETHA0RSTCNT - Software reset control register.***
		p.2206	Modification of Initial value, Attribute to ***32.4.3 (3) ETHA0FLOWTH - Flow control threshold value register.***
		p.2208	Modification of Initial value, Attribute to ***32.4.3 (5) ETHA0RXERSEL - Receive error selection register.***
		p.2210	Modification of Initial value, Attribute to ***32.4.3 (6) ETHA0TXSTMONI1 - Transmission status monitor 1 register.***
		p.2212	Modification of Initial value, Attribute to ***32.4.3 (8) ETHA0TXFINF1 - Transmission status 1 register.***
		p.2213	Modification of Initial value, Attribute to ***32.4.3 (9) ETHA0TXFINF2 - Transmission status 2 register.***
		p.2216	Modification of Initial value, Attribute to ***32.4.3 (11) ETHA0RXFINF1 - Reception status 1 register.***
		p.2217	Modification of Initial value, Attribute to ***32.4.3 (12) ETHA0RXFINF2 - Reception status 2 register***
		p.2218	Modification of Initial value, Attribute to ***32.4.3 (13) ETHA0RXFINF3 - Reception status 3 register.***
		p.2219	Modification of Initial value, Attribute to ***32.4.3 (14) ETHA0FSTATUS - FIFO status interrupt register.***
		p.2221	Modification of Caution, Initial value, Attribute to ***32.4.3 (15) ETHA0FSTATMK - FIFO status interrupt mask register.***
		p.2222	Modification of Initial value, Attribute to ***32.4.3 (16) ETHA0TXSTATUS - Transmission status interrupt register.***

Rev.	Date	Description	
		Page	Summary
2.00	July 25, 2012	p.2224	Modification of Initial value, Attribute to ***32.4.3 (17) ETHA0TXSTATMK - Transmission status interrupt mask register.***
		p.2225	Modification of Initial value, Attribute to ***32.4.3 (18) ETHA0RXSTATUS - Reception status interrupt register.***
		p.2227	Modification of Initial value, Attribute to ***32.4.3 (19) ETHA0RXSTATMK - Reception status interrupt mask register.***
		p.2229	Modification of Initial value, Attribute to ***32.4.3 (20) ETHA0TXABTCNT - TX abort counter.***
		p.2230	Modification of Initial value, Attribute to ***32.4.3 (21) ETHA0RXABTCNT - RX abort counter.***
		p.2231	Modification of Initial value, Attribute to ***32.4.4 (1) ETHA0MODE - Core function control register.***
		p.2232	Modification of Initial value, Attribute to ***32.4.4 (2) ETHA0INTMS - Interrupt control register.***
		p.2234	Modification of Initial value, Attribute to ***32.4.4 (3) ETHA0TRANSCTL - Transfer control register.***
		p.2235	Modification of Initial value, Attribute to ***32.4.4 (4) ETHA0SFTRST - Software reset setting register.***
		p.2236	Modification of Initial value, Attribute to ***32.4.4 (5) ETHA0DMACM - DMA mode control register.***
		p.2241	Modification of Initial value, Attribute to ***32.4.5 (1) ETHA0CMODE - Transmit checksum unit function setting register.***
		p.2242	Modification of Initial value, Attribute to ***32.4.5 (2) ETHA0CINTMS - Transmit checksum interrupt register***
		p.2244	Modification of Initial value, Attribute to ***32.4.5 (3) ETHA0CTRANSCTL - Transmit checksum transfer control register***
		p.2245	Modification of Initial value, Attribute to ***32.4.5 (4) ETHA0CSFTRST - Transmit checksum software reset register.***
		p.2246	Modification of Initial value, Attribute to ***32.4.5 (5) ETHA0CDMACM - Transmit checksum DMAC control mode setting*** **register**.
		p.2274	Modification of Initial value, Attribute to ***32.5.7 (1) (b) Filtering of multicast addresses***.
		p.2308	Addition of Caution to ***34.2.1 Communication method.***
		p.2311	Addition of Caution to ***34.4.2 (1) Single-wire asynchronous serial programming interface: One-wired UART.***
		p.2320	Addition of Caution to ***34.5 Flash Memory Self Programming.***
		p.2342	Addition of ***37.4.1 Recommended resonator.***
		p.2343	Modification of ***37.5.1 Pin characteristics.***
		p.2367	Modification of ***37.6.3 Clock timing.***
		p.2372	Modification of ***37.6.9 (1) CSIH0, CSIH2, CSIH3 (8.33 Mbps (f_{XX} = 200 MHz): While in the master mode).***
		p.2374	Modification of ***34.6.9 (3) CSIH1 (8.33 Mbps (f_{XX} = 200 MHz): While in the master mode).***
		p.2376	Modification of ***Figure 37-23 CSIH Timing.***
		p.2381	Modification of ***Figure 37-24 CSIG Timing.***
		p.2384	Modification of ***37.6.13 (2) Reception interface.***

Rev.	Date	Description	
		Page	Summary
2.00	July 25, 2012	p.2386	Modification of ***37.6.14 (1) Normal operation (ANI00 to ANI11).***
		p.2387	Modification of ***37.6.14 (2) When using the channel sample & hold circuit (ANI00 to ANI05)***
		p.2389	Modification of ***37.6.15 (2) When using the channel sample & hold circuit (ANI00 to ANI05)***
		p.2394	Addition of Caution to ***Chapter 39 Recomended Soldering Conditions.***
3.00	Feb 28, 2013	Through out	Change of the following function names • $\overline{\text{DRST}}$ –>$\overline{\text{TRST}}$ • DCK –>TCK • DDI –>TDI • DDO –>TDO • DMS –>TMS
		p.311	Modification of ***Figure 8-7 Digital filter block diagram.***
		p.316	Modification of ***Control for digital filters*** to ***8.5.2(3) (a)*** **FCLAnCTLm – Filter control register*r.***
		p.316	Modification of ***Table 8-46 FCLAnCTLm register contents for digital filters.***
		p.2387	Modification of ***37.6.14 (2) When using the channel sample & hold circuit (ANI00 to ANI05).***
		p.2388	Modification of ***34.6.15 (1) Normal operation (ANI00 to ANI11).***
		p.2389	Modification of ***34.6.15 (2) When using the channel sample & hold circuit (ANI00 to ANI05).***
4.00	Sep 15, 2014	p.105	Modification of ***4.4.1 CPU address space.***
		p.106	Modification of ***4.4.2 (1) Program space.***
		p.106	Modification of ***Figure 4-4 Program space wraparound.***
		p.485	Addition of Caution to ***10.13.3 DTSSELm (m = 0 to 15): DTS transfer end interrupt select register.***
		p.641	Addition of Caution to***12.16.4 (2) Equations.***
		p.713	Addition of Caution to***12.20.3 (2) Equations.***
		p.954	Addition of Caution to***13.15.4 (2) Equations.***
		p.966	Addition of Caution to***13.16.1 (2) Equations.***
		p.1225	Modification of ***Figure 20-4 Processing flow after interrupt generation.***
		p.1645	Modification of***24.13.1 Baud rate setting conditions.***
		p.1905	Modification of***30.5 (1) Clock accuracy.***

V850E2/MN4 Hardware User Manual

Publication Date: Rev.4.00 Sep 15, 2014

Published by: Renesas Electronics Corporation

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
2801 Scott Boulevard Santa Clara, CA 95050-2549, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-6503-0, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
Room 1709, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100191, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 301, Tower A, Central Towers, 555 Langao Road, Putuo District, Shanghai, P. R. China 200333
Tel: +86-21-2226-0888, Fax: +86-21-2226-0999

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2265-6688, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei 10543, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre, Singapore 339949
Tel: +65-6213-0200, Fax: +65-6213-0300

Renesas Electronics Malaysia Sdn.Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.
12F., 234 Teheran-ro, Gangnam-Ku, Seoul, 135-920, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 3.0

V850E2/MN4

Renesas Electronics Corporation

R01UH0011EJ0400