Spansion® Analog and Microcontroller Products

The following document contains information on Spansion analog and microcontroller products. Although the document is marked with the name "Fujitsu", the company that originally developed the specification, Spansion will continue to offer these products to new and existing customers.

Continuity of Specifications

There is no change to this document as a result of offering the device as a Spansion product. Any changes that have been made are the result of normal document improvements and are noted in the document revision summary, where supported. Future routine revisions will occur when appropriate, and changes will be noted in a revision summary.

Continuity of Ordering Part Numbers

Spansion continues to support existing part numbers beginning with "MB". To order these products, please use only the Ordering Part Numbers listed in this document.

For More Information

Please contact your local sales office for additional information about Spansion memory, analog, and microcontroller products and solutions.

FUJITSU SEMICONDUCTOR
CONTROLLER MANUAL

MN709-00001-1v0-E

FM4

Fujitsu
Cortex M4

32-BIT MICROCONTROLLER

FM4 Family

PERIPHERAL MANUAL

For the latest information for microcontroller supports, see the following web site.

http://edevice.fujitsu.com/micom/en-support/

FUJITSU SEMICONDUCTOR LIMITED

ARM™

FUJITSU

Preface

Thank you for your continued use of Fujitsu semiconductor products.
Read this manual and "Data Sheet" thoroughly before using products in this family.

■ Purpose of this manual and intended readers

This manual explains the functions and operations of this family and describes how it is used. The manual is intended for engineers engaged in the actual development of products using this family.
For the descriptions on Analog macro, Timer, and Communication Macro, see the respective separate peripheral manual.

* This manual explains the configuration and operation of the peripheral functions, but does not cover the specifics of each device in the series.
Users should refer to the respective data sheets of devices for device-specific details.

■ Trademark

ARM and Cortex are the trademarks of ARM Limited in the EU and other countries.

The company names and brand names herein are the trademarks or registered trademarks of their respective owners.

■ Sample programs and development environment

Fujitsu Semiconductor offers sample programs free of charge for using the peripheral functions of the FM4 family. Fujitsu Semiconductor also makes available descriptions of the development environment required for this family. Feel free to use them to verify the operational specifications and usage of this Fujitsu Semiconductor microcontroller.

· Microcontroller support information:
http://edevice.fujitsu.com/micom/en-support/

* : Note that the sample programs are subject to change without notice. Since they are offered as a way to demonstrate standard operations and usage, evaluate them sufficiently before running them on your system.
Fujitsu Semiconductor assumes no responsibility for any damage that may occur as a result of using a sample program.

■ Overall Organization of This Manual

Peripheral Manual Timer Part has 18 chapters and APPENDIXES as shown below.

CHAPTER 1 : System Overview

CHAPTER 2-1 : Clock

CHAPTER 2-1 : Peripheral Clock Gating

CHAPTER 2-3 : High-Speed CR Trimming

CHAPTER 2-4 : Low-Speed CR Prescaler

CHAPTER 3 : Clock supervisor

CHAPTER 4 : Resets

CHAPTER 5 : Low-voltage Detection (A)

CHAPTER 6 : Low Power Consumption Mode

- FUJITSU SEMICONDUCTOR LIMITED, its subsidiaries and affiliates (collectively, "FUJITSU SEMICONDUCTOR") reserves the right to make changes to the information contained in this document without notice. Please contact your FUJITSU SEMICONDUCTOR sales representatives before order of FUJITSU SEMICONDUCTOR device.
- Information contained in this document, such as descriptions of function and application circuit examples is presented solely for reference to examples of operations and uses of FUJITSU SEMICONDUCTOR device. FUJITSU SEMICONDUCTOR disclaims any and all warranties of any kind, whether express or implied, related to such information, including, without limitation, quality, accuracy, performance, proper operation of the device or non-infringement. If you develop equipment or product incorporating the FUJITSU SEMICONDUCTOR device based on such information, you must assume any responsibility or liability arising out of or in connection with such information or any use thereof. FUJITSU SEMICONDUCTOR assumes no responsibility or liability for any damages whatsoever arising out of or in connection with such information or any use thereof.
- Nothing contained in this document shall be construed as granting or conferring any right under any patents, copyrights, or any other intellectual property rights of FUJITSU SEMICONDUCTOR or any third party by license or otherwise, express or implied. FUJITSU SEMICONDUCTOR assumes no responsibility or liability for any infringement of any intellectual property rights or other rights of third parties resulting from or in connection with the information contained herein or use thereof.
- The products described in this document are designed, developed and manufactured as contemplated for general use including without limitation, ordinary industrial use, general office use, personal use, and household use, but are not designed, developed and manufactured as contemplated (1) for use accompanying fatal risks or dangers that, unless extremely high levels of safety is secured, could lead directly to death, personal injury, severe physical damage or other loss (including, without limitation, use in nuclear facility, aircraft flight control system, air traffic control system, mass transport control system, medical life support system and military application), or (2) for use requiring extremely high level of reliability (including, without limitation, submersible repeater and artificial satellite). FUJITSU SEMICONDUCTOR shall not be liable for you and/or any third party for any claims or damages arising out of or in connection with above-mentioned uses of the products.
- Any semiconductor devices fail or malfunction with some probability. You are responsible for providing adequate designs and safeguards against injury, damage or loss from such failures or malfunctions, by incorporating safety design measures into your facility, equipments and products such as redundancy, fire protection, and prevention of overcurrent levels and other abnormal operating conditions.
- The products and technical information described in this document are subject to the Foreign Exchange and Foreign Trade Control Law of Japan, and may be subject to export or import laws or regulations in U.S. or other countries. You are responsible for ensuring compliance with such laws and regulations relating to export or re-export of the products and technical information described herein.
- All company names, brand names and trademarks herein are property of their respective owners.

Related Manuals

The manuals related to this family are listed below. See the manual appropriate to the applicable conditions.
The contents of these manuals are subject to change without notice. Contact us to check the latest versions available.

■ Peripheral Manual

- **FM4 Family PERIPHERAL MANUAL (this manual)**
  (Called "PERIPHERAL MANUAL" hereafter)
- **FM4 Family PERIPHERAL MANUAL Timer Part**
  (Called "Timer Part" hereafter)
- **FM4 Family PERIPHERAL MANUAL Analog Macro Part**
  (Called "Analog Macro Part" hereafter)
- **FM4 Family PERIPHERAL MANUAL Communication Macro Part**
  (Called "Communication Macro Part" hereafter)

■ Data sheet

For details about device-specific, electrical characteristics, package dimensions, ordering information etc., see the following document.

- **32-bit Microcontroller FM4 Family DATA SHEET**

* The data sheets for each series are provided.
See the appropriate data sheet for the series that you are using.

■ CPU Programming manual

For details about ARM Cortex-M4F core, see the following documents that can be obtained from http://www.arm.com/.

- **Cortex-M4 Technical Reference Manual**
- **ARMv7-M Architecture Application Level Reference Manual**

■ Flash Programming manual

For details about the functions and operations of the built-in flash memory, see the following document.

- **FM4 Family FLASH PROGRAMMING MANUAL**

* Flash programming manuals for each series are provided.
See the appropriate flash programming manual for the series that you are using.

How to Use This Manual

■ Finding a function

The following methods can be used to search for the explanation of a desired function in this manual:

- Search from the table of the contents
  The table of the contents lists the manual contents in the order of description.

- Search from the register
  The address where each register is located is not described in the text. To verify the address of a register, see "A. Register Map" in "APPENDIXES".

■ About the chapters

Basically, this manual explains 1 peripheral function per chapter.

■ Terminology

This manual uses the following terminology.

Term	Explanation
Word	Indicates access in units of 32 bits.
Half word	Indicates access in units of 16 bits.
Byte	Indicates access in units of 8 bits.

■ Notations

- The notations in bit configuration of the register explanation of this manual are written as follows.
  - bit : bit number
  - Field : bit field name
  - Attribute : Attributes for read and write of each bit
    - R : Read only
    - W : Write only
    - R/W : Readable/Writable
    - - : Undefined
  - Initial value : Initial value of the register after reset
    - 0 : Initial value is "0"
    - 1 : Initial value is "1"
    - X : Initial value is undefined

- The multiple bits are written as follows in this manual.
  Example : bit7:0 indicates the bits from bit7 to bit0

- The values such as for addresses are written as follows in this manual.
  - Hexadecimal number : "0x" is attached in the beginning of a value as a prefix (example : 0xFFFF)
  - Binary number : "0b" is attached in the beginning of a value as a prefix (example: 0b1111)
  - Decimal number : Written using numbers only (example : 1000)

CONTENTS

MAJOR CHANGES IN THIS EDITION

Page	Section	Change Results
-	-	First edition

CHAPTER: System Overview

This chapter explains the system overview.

CODE: 3-6.0-SYSTEM_FM4-E01.0

1. Bus Architecture

This section explains the bus architecture.

For this family bus, AHB Bus Matrix circuit actualizes a multi-layer bus. Master and slave architectures are shown below:

● Master

- Cortex-M4F CPU(I-code Bus, D-code Bus, System Bus)
- Ethernet/SD-Card
- DMAC
- DSTC

● Slave

- On-chip Flash Memory (MainFlash, WorkFlash)
- On-chip SRAM (SRAM0, SRAM1, SRAM2)
- External Bus
- USB ch.0/ch.1
- AHB-AHB Bus Bridge
- AHB-APB Bus Bridge (APB0 to APB2)

See Figure 1-1 for the bus block diagram.

■ Features

● RAM Architecture

This family divides the on-chip SRAM area into three separate SRAM (SRAM0, SRAM1, and SRAM2). SRAM0 is connected to the I-Code bus and D-Code bus of the Cortex-M4 core. SRAM1 and SRAM2 are connected to the System bus of the Cortex-M4 core. Also, SRAM0, SRAM1, and SRAM2 are connected to DMAC and other bus masters. This allows for preventing conflicts to RAM by multiple bus masters such as CPU and DMAC and allows for improving the performance.
Also, because the divided RAM address areas are serial, RAM area can be utilized to the maximum extent.

● APB Extension Bus

APB1 and APB2 Peripheral Buses are APB extension bus that the following functions are originally added based on AMBA3.0. (APB0 is not included.)

- **Supporting Halfword (16 bits) and Byte(8 bits) Accesses**
  For supported registers, halfword access and byte access are enabled.
  See "A. Register Map" in "APPENDIXES" for the supported registers.

- **Adding Read-Modify-Write (RMW) Signal**
  HMASTLOCK signal in bit-band operations is used to generate.

  RMW signal is a signal added to prevent that an unrelated flag is cleared mistakenly in read-modify-write process of bit-band operations.
  The corresponding flag reads "1" in read during the read-modify-write process and is designed to ignore "1" write.
  This prevents any unrelated flag from being mistakenly cleared in the next write when the flag is set immediately after the read in the sequence from read to modify to write.

  For the corresponding flags and registers, it is described that "regardless of bit values, "1" can be read in "Read-Modify-Write".

<Notes>

- bit-band operation must not be performed to a register which RMW is prohibited.
- When Read-Modify-Write process is performed over the software without bit-band operation, RMW signal is not output.
  Therefore, in this case, the flag value can be read in read operation although a register supports RMW process, and it is necessary not to be cleared an unrelated flag mistakenly in write operation.
- For the details of bit-band operations, see the "Cortex-M4 Technical Reference Manual".

● Priority Level

A priority of the bus right is determined in round-robin fashion.

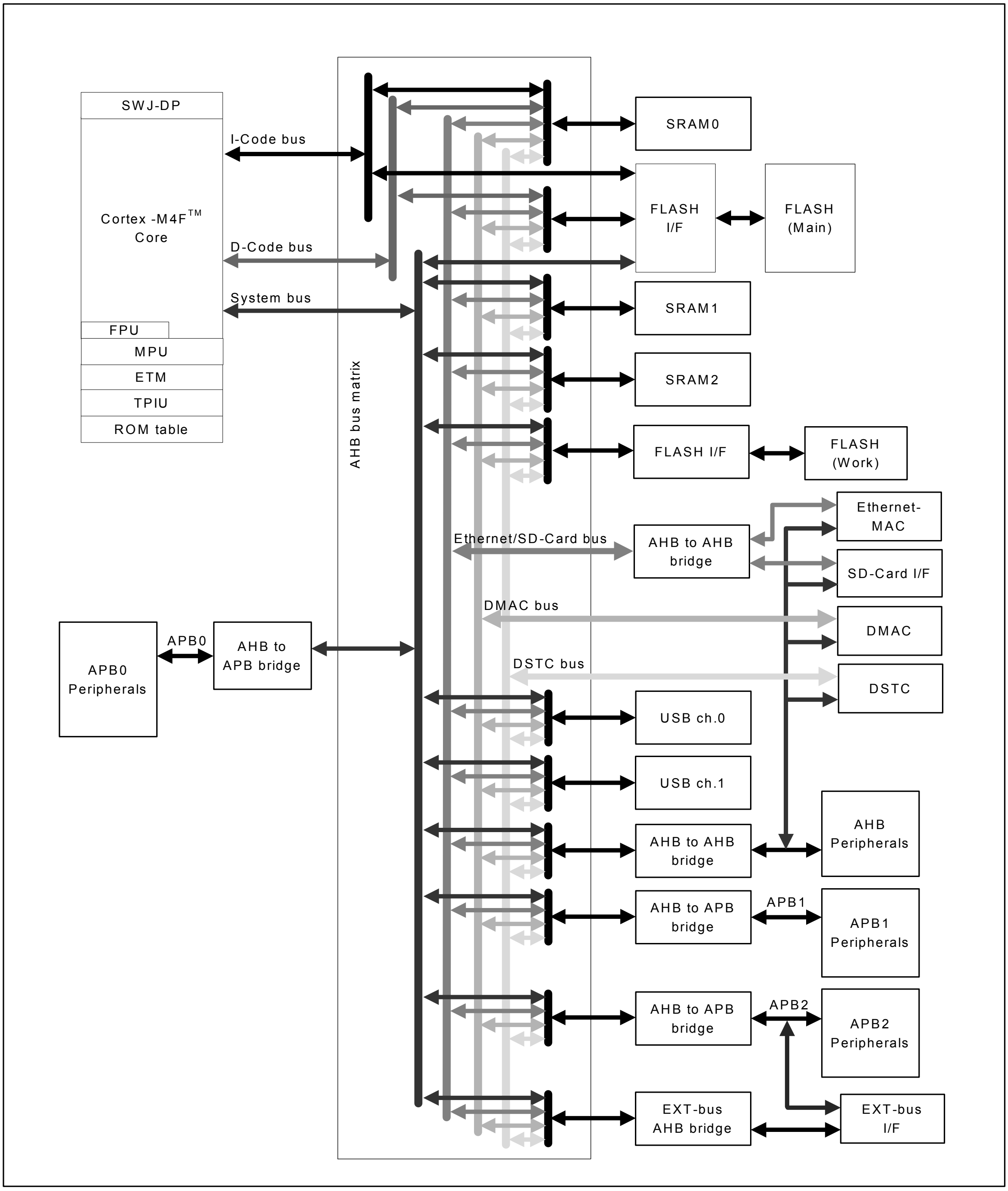
● Endian

This family uses Little endian byte order.

1.1. Bus Block Diagram

Figure 1-1 illustrates the bus block diagram.

Figure 1-1 Bus Block Diagram



<Note>

There are some areas to which no DMAC transfer can be performed. For details, see the DMAC Transfer column in Table 1-1.

1.2. Memory Architecture

This section shows the memory architecture.

For this family, 4G-byte address space is available.

Maximum 4M-byte FLASH memory, maximum 512K-byte SRAM0 area, maximum 256K-byte SRAM1 area, and maximum 256K-byte SRAM2 area are defined.

Also, as an external bus area, 2G-byte area from 0x60000000 to 0xDFFFFFFF is defined. An external memory device can be connected to this area.

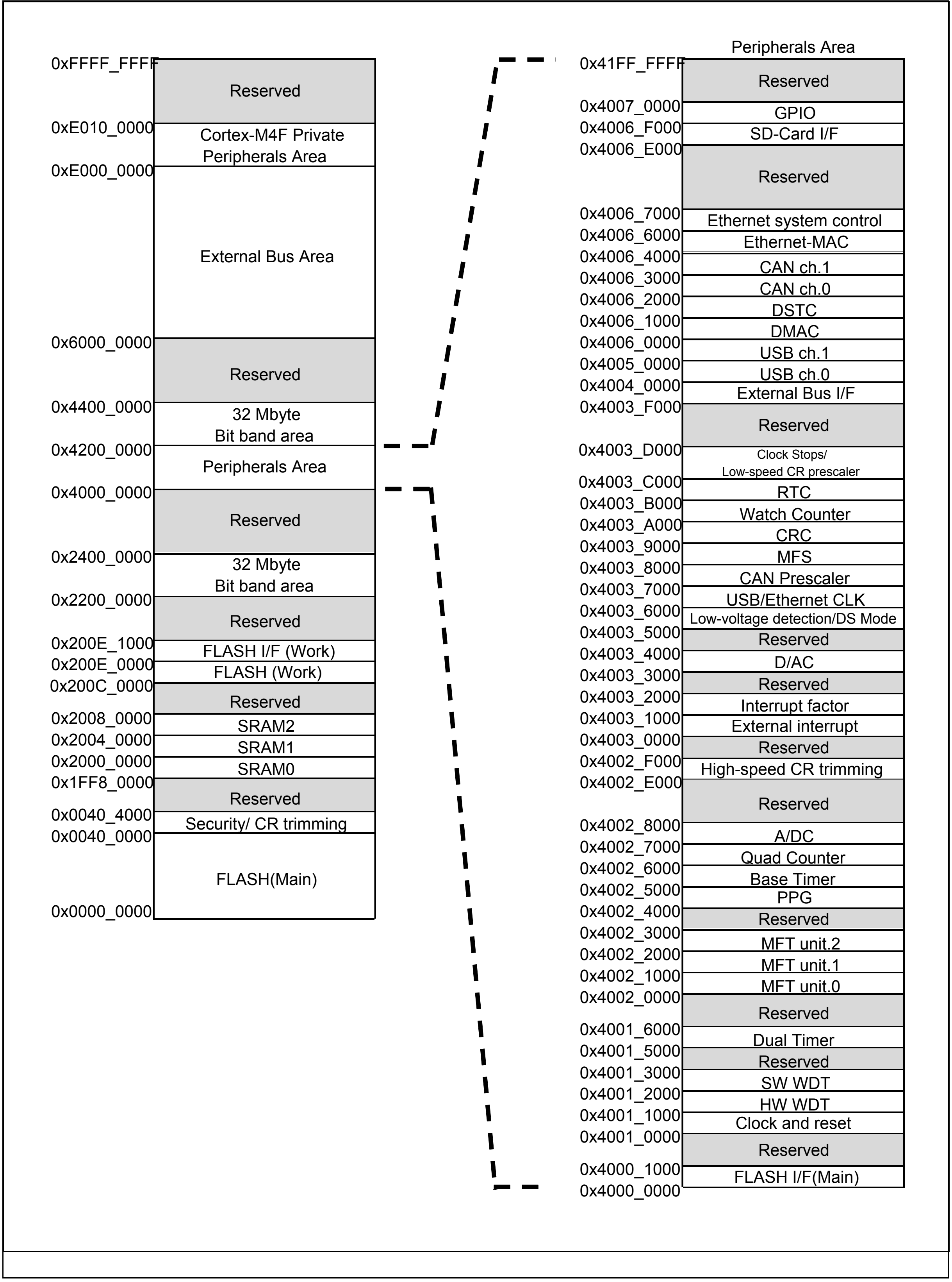
Section "1.3 Memory Map" illustrates the memory map, and Section "1.4 Peripheral Address Map" illustrates the peripheral address map.

For the details of Cortex-M4F private peripheral area and bit-band area shown in Figure 1-2, see "Cortex-M4 Technical Reference Manual".

1.3. Memory Map

Figure 1-2 illustrates the memory map.

Figure 1-2 Memory Map



0xFFFF_FFFF	
	Reserved
0xE010_0000	
	Cortex-M4F Private Peripherals Area
0xE000_0000	
	External Bus Area
0x6000_0000	
	Reserved
0x4400_0000	
	32 Mbyte Bit band area
0x4200_0000	
	Peripherals Area
0x4000_0000	
	Reserved
0x2400_0000	
	32 Mbyte Bit band area
0x2200_0000	
	Reserved
0x200E_1000	
	FLASH I/F (Work)
0x200E_0000	
	FLASH (Work)
0x200C_0000	
	Reserved
0x2008_0000	
	SRAM2
0x2004_0000	
	SRAM1
0x2000_0000	
	SRAM0
0x1FF8_0000	
	Reserved
0x0040_4000	
	Security/ CR trimming
0x0040_0000	
	FLASH(Main)
0x0000_0000	

Peripherals Area

0x41FF_FFFF	
	Reserved
0x4007_0000	
	GPIO
0x4006_F000	
	SD-Card I/F
0x4006_E000	
	Reserved
0x4006_7000	
	Ethernet system control
0x4006_6000	
	Ethernet-MAC
0x4006_4000	
	CAN ch.1
0x4006_3000	
	CAN ch.0
0x4006_2000	
	DSTC
0x4006_1000	
	DMAC
0x4006_0000	
	USB ch.1
0x4005_0000	
	USB ch.0
0x4004_0000	
	External Bus I/F
0x4003_F000	
	Reserved
0x4003_D000	
	Clock Stops/ Low-speed CR prescaler
0x4003_C000	
	RTC
0x4003_B000	
	Watch Counter
0x4003_A000	
	CRC
0x4003_9000	
	MFS
0x4003_8000	
	CAN Prescaler
0x4003_7000	
	USB/Ethernet CLK
0x4003_6000	
	Low-voltage detection/DS Mode
0x4003_5000	
	Reserved
0x4003_4000	
	D/AC
0x4003_3000	
	Reserved
0x4003_2000	
	Interrupt factor
0x4003_1000	
	External interrupt
0x4003_0000	
	Reserved
0x4002_F000	
	High-speed CR trimming
0x4002_E000	
	Reserved
0x4002_8000	
	A/DC
0x4002_7000	
	Quad Counter
0x4002_6000	
	Base Timer
0x4002_5000	
	PPG
0x4002_4000	
	Reserved
0x4002_3000	
	MFT unit.2
0x4002_2000	
	MFT unit.1
0x4002_1000	
	MFT unit.0
0x4002_0000	
	Reserved
0x4001_6000	
	Dual Timer
0x4001_5000	
	Reserved
0x4001_3000	
	SW WDT
0x4001_2000	
	HW WDT
0x4001_1000	
	Clock and reset
0x4001_0000	
	Reserved
0x4000_1000	
	FLASH I/F(Main)
0x4000_0000	

<Notes>

- Do not access to reserved area.
- For the details of flash memory, see "FLASH PROGRAMMING MANUAL" of the product used.
- Do not perform DMAC-transfer to bit-band area.

1.4. Peripheral Address Map

Table 1-1 shows the peripheral address map.

Table 1-1 Peripheral Address Map

Start Address	End Address	Bus	Access other than CPU	Peripheral	Register Map	CHAPTER
0x4000_0000	0x4000_0FFF	AHB	Disabled	FLASH IF Register (Main)/ Unique ID Register	FLASH_IF/ Unique ID	*1 Chapter17 Chapter18
0x4000_1000	0x4000_FFFF			Reserved	-	-
0x4001_0000	0x4001_0FFF	APB0	Disabled	Clock and Reset Control	Clock / Reset	Chapter 2-1 Chapter 3 Chapter 4 Chapter 6
0x4001_1000	0x4001_1FFF			Hardware Watchdog Timer	HW WDT	Chapter 1 in Timer Part
0x4001_2000	0x4001_2FFF			Software Watchdog Timer	SW WDT	
0x4001_3000	0x4001_4FFF			Reserved	-	-
0x4001_5000	0x4001_5FFF			Dual Timer	Dual_ Timer	Chapter 2 in Timer Part
0x4001_6000	0x4001_FFFF			Reserved	-	-
0x4002_0000	0x4002_0FFF	APB1		Multi-function Timer unit0	MFT	Chapter 6 in Timer Part
0x4002_1000	0x4002_1FFF			Multi-function Timer unit1		
0x4002_2000	0x4002_2FFF			Multi-function Timer unit2		
0x4002_3000	0x4002_3FFF			Reserved	-	-
0x4002_4000	0x4002_4FFF			PPG	PPG	Chapter 7-2 in Timer Part
0x4002_5000	0x4002_5FFF			Base Timer	Base Timer/ Base Timer Selector	Chapter 5-1 Chapter 5-2 in Timer Part
0x4002_6000	0x4002_6FFF			QPRC	QPRC	Chapter 8-1 Chapter 8-2 in Timer Part
0x4002_7000	0x4002_7FFF			A/D Converter	A/DC	Chapter 1-1 Chapter 1-2 Chapter 1-3 in Analog Macro Part
0x4002_8000	0x4002_DFFF			Reserved	-	-
0x4002_E000	0x4002_EFFF			High speed CR trimming	CR Trim	Chapter 2-3

*1 : For the details of "FLASH IF Register", see "FLASH PROGRAMMING MANUAL" of the product used.

Start Address	End Address	Bus	Access other than CPU	Peripheral	Register Map	CHAPTER
0x4002_F000	0x4002_FFFF	APB2	Enabled	Reserved	-	-
0x4003_0000	0x4003_0FFF			External Interrupt	EXTI	Chapter 9
0x4003_1000	0x4003_1FFF			Interrupt Factor Check Register	INT-Req READ	Chapter 8
0x4003_2000	0x4003_2FFF			Reserved	-	-
0x4003_3000	0x4003_3FFF			D/A Converter	D/AC	Chapter 2 in Analog Macro Part
0x4003_4000	0x4003_4FFF			Reserved	-	-
0x4003_5000	0x4003_50FF			Low Voltage Detection	LVD	Chapter 5
0x4003_5100	0x4003_5FFF			Deep standby control block	DS_Mode	Chapter 6
0x4003_6000	0x4003_6FFF			USB clock generation block/ USB/Ethernet clock generation block	USB Clock	Chapter 2-2 Chapter 2-3 in Communication Macro Part
0x4003_7000	0x4003_7FFF			CAN Prescaler	CAN_ Prescaler	Chapter 5-1 in Communication Macro Part
0x4003_8000	0x4003_8FFF			Multi-function serial	MFS	Chapter 1-1 Chapter 1-2 Chapter 1-3 Chapter 1-4 Chapter 1-5 in Communication Macro Part
0x4003_9000	0x4003_9FFF			CRC	CRC	Chapter 13
0x4003_A000	0x4003_AFFF			Watch counter	Watch Counter	Chapter 3-1 Chapter 3-2 in Timer Part
0x4003_B000	0x4003_BFFF			Real time clock	RTC	Chapter 4-1 Chapter 4-2 Chapter 4-3 in Timer Part
0x4003_C000	0x4003_CFFF			Clock Stop /Low Speed CR Prescaler	Clock Stop	Chapter 2-2 Chapter 2-4
0x4003_D000	0x4003_EFFF			Reserved	-	
0x4003_F000	0x4003_FFFF			External Bus I/F	EXT-Bus I/F	Chapter 14

Start Address	End Address	Bus	Access other than CPU	Peripheral	Register Map	CHAPTER
0x4004_0000	0x4004_FFFF	AHB	Enabled	USB ch.0	USB	Chapter 3-1 Chapter 3-2 in Communication Macro Part
0x4005_0000	0x4005_FFFF			USB ch.1	USB	
0x4006_0000	0x4006_0FFF			DMAC	DMAC	Chapter 10
0x4006_1000	0x4006_1FFF			DSTC	DSTC	Chapter 11
0x4006_2000	0x4006_2FFF			CAN ch.0	CAN	Chapter 5-2 in Communication Macro Part
0x4006_3000	0x4006_3FFF			CAN ch.1	CAN	
0x4006_4000	0x4006_5FFF			Ethernet MAC	Ethernet MAC	-
0x4006_6000	0x4006_6FFF			Ethernet System Control	Ethernet Control	-
0x4006_7000	0x4006_DFFF			Reserved	-	-
0x4006_E000	0x4006_EFFF			SD-Card I/F	SD Card IF	Chapter 15
0x4006_F000	0x4006_FFFF			GPIO	GPIO	Chapter 12
0x4007_0000	0x41FF_FFFF			Reserved	-	-

Start Address	End Address	Bus	Access other than CPU	Peripheral	Register Map	Details
0x200E_0000	0x200E_1000	AHB	Enabled	Flash IF register (Work)	"FLASH IF"	*2

*2 : For the details of "FLASH IF register (Work)", refer to "FLASH PROGRAMMING MANUAL" of the product used.

2. Cortex-M4F Architecture

This section explains the core architecture used in this family.

Cortex-M4F core block architecture* used in this family is as follows:

- Cortex-M4 Core
- NVIC
- FPU
- DWT
- ITM
- FPB
- MPU
- ETM
- SWJ-DP
- TPIU
- ROM Table

*: The architecture varies depending on the products. For details, see "Data Sheet" of the product used.

■ Cortex-M4 Core

High-performance 32-bit processor core (ARM Cortex-M4 core) is equipped with this family.
This peripheral manual does not describe the details of Cortex-M4 core.
For the details, see "Cortex-M4 Technical Reference Manual".

- Cortex-M4 Core Version
  For the version of Cortex-M4 core, see "Data Sheet" of the product used.

■ NVIC (Nested Vectored Interrupt Controller)

For this family, one NMI (non-maskable interrupt) and maximum 128 peripheral interrupts (IRQ0 to IRQ127)*1 can be used.
Also, interrupt priority register (from 0xE000E400) is comprised of 4 bits, and 16 interrupt priority levels can be configured.

For the details of peripheral interrupts, see the chapter of the target "Interrupts" after check the product currently used with "Configuration of interrupts", and for NMI operations, see also another chapter "External Interrupt and NMI Control Block".

NMI pin is assigned for a combined use with a general-purpose port. Its initial value after a reset release is set to the general-purpose port, and NMI input is masked.
When NMI is used, enable NMI in the port setting.
For the details, see another chapter "I/O Port".

*1: "Cortex-M4 Technical Reference Manual" defines an exception type: IRQ as an external interrupt. In this peripheral manual, to distinguish from an interrupt by an external pin "External Interrupt and NMI Control Block", the exception type: IRQ is indicated as a peripheral interrupt.

- **SysTick Timer**
  SysTick Timer is a system timer for OS task management integrated into NVIC.

  This family generates STCLK through dividing HCLK by eight and sets the values of SysTick Calibration Value Register (Address: 0xE000E01C) as shown below:

bit31	:	NOREF = 0
bit30	:	SKEW = 1
bit23:0	:	TENMS = 0x0186A0 (100000)*[1]

*1 : TENMS value is set to a value which becomes 10 ms when 1/8 clock of HCLK is input to STCLK and that HCLK is in 80 MHz (10 MHz in 1/8 case).
The value of TENMS is not always 10ms because HCLK can be changed to another frequency with the clock control block. Therefore, it is required to calculate an appropriate interrupt timing according to HCLK frequency.

■ FPU (Floating Point number processing Unit)

This family is equipped with a Cortex-M4 optional component FPU.
FPU has the following features:

- Conforms to IEEE754 Standard
- Single Precision Floating Point number processing Unit
- High Precision Fused MAC Calculation

■ DWT (Data Watchpoint & Trace Unit)

This family is equipped with DWT to use as the debug function.
DWT contains four comparators, and each comparator can be set as a hardware watchpoint.

■ ITM (Instrumentation Trace Macrocell)

This family is equipped with ITM as a debug function.
ITM is an optional application driven trace source that supports printf style debugging. The operation system (OS) and application event are traced, and the system diagnostic information is sent.

■ FPB (Flash Patch & Breakpoint)

FPB has the following functions:

- Hardware Breakpoint function
- The function of remapping from Code memory space (FLASH memory) to SRAM space.

FPB is equipped with six instruction comparators and two literal comparators.

■ MPU (Memory Protection Unit)*

This family is equipped with a Cortex-M4 optional component MPU, and maximum eight areas can be defined.

*: Some products do not have this function. For details, see "Data Sheet" of the product used.

■ ETM (Embedded Trace Macrocell)*

This family is equipped with a Cortex-M4 optional component ETM to support instruction trace.

*: Some products do not have this function. For details, see "Data Sheet" of the product used.

■ SWJ-DP

This family is equipped with SWJ-DP to support both serial wire protocol and JTAG protocol.

■ TPIU (Trace Port Interface Unit)

ETM/ITM trace information is output via TPIU.

■ ROM Table

ROM table provides the address information of a debug component to an external debug tool.

3. Mode

This section explains the function of operating modes.

In this family, the following operating modes can be used:

- User Mode
  Internal ROM (Flash memory) Startup : CPU obtains a reset vector from Flash memory and starts operations.

- Serial Writer Mode
  Serial write to Flash memory is enabled.
  * : For the details of this mode, see "FLASH PROGRAMMING MANUAL" of the product used.

Operating modes are determined after a release of respective power-on reset, low voltage detection reset, and INITX pin input reset.

* : For the details of power consumption control and clock selection modes, see other chapters "Low Power Consumption Mode" and "Clock".

■ How to Set Operating Mode

Operating modes are configured by MD pins' (MD1 and MD0) inputs.

MD Pins		Operating Mode
MD1	MD0	
-	0	User Mode Internal ROM(Flash memory) Startup
0	1	Serial Writer Mode
1	1	Setting is prohibited.

■ Startup Sequence

Processes to determine operating modes in the startup sequence are shown below:

1. MD Pin Sampling
2. Determining Operating Mode and Retaining Mode Data

The descriptions of these processes are as follows:

1. MD Pin Sampling
   Operating mode is configured by MD pin inputs (MD1, MD0). These inputs are sampled by power-on reset, low-voltage detection reset, and INITX pin input reset.
   Until each reset, which is the sampling factor, is released, MD1 and MD0 pin inputs need to be determined.

2. Determining Operating Mode and Retaining Mode Data
   MD1 and MD0 which are sampled by respective resets are retained until respective resets are input again.
   Operating modes are determined by the retained MD1 and MD0. Therefore, even MD1 and MD0 are changed after a reset is released, it does not affect an operating mode.

■ MD1 pin

MD1 pin is used also as GPIO. This pin can be continually used as GPIO after setting a mode.

CHAPTER: Clock

This chapter explains the operating clock.

CODE: 9BFCLOCK_FM4-E01.0

1. Clock Generation Unit Overview

This section provides an overview of the clock generation unit.

The clock generation unit generates various types of clocks used to operate the MCU.

Source clock is the generic name for external and internal oscillation clocks of this MCU.

The following five types of clocks are source clocks:

- Main clock (CLKMO)
- Sub clock (CLKSO)
- High-speed CR clock (CLKHC)
- Low-speed CR clock (CLKLC)
- Main PLL clock (CLKPLL)

Select one from the source clocks. In this chapter, the selected clock is referred to as the master clock. The master clock is a source of internal bus clocks used to operate this MCU.

Dividing the master clock frequency can generate a base clock. In addition, dividing the base clock can generate each bus clock.

In this chapter, the base clock and bus clocks are referred to as internal bus clocks. The following five types of clocks are internal bus clocks:

- Base clock (FCLK/HCLK)
- APB0 bus clock (PCLK0)
- APB1 bus clock (PCLK1)
- APB2 bus clock (PCLK2)
- TRACE clock (TPIUCLK)

In addition to source clocks, the master clock, and internal bus clocks, the following clocks are provided:

- USB clock
- USB/Ethernet clock
- CAN prescaler clock
- Software watchdog timer count clock

The following shows the features of the clock generation unit.

- It can set the oscillation stabilization wait time of the main clock (CLKMO).
- It can set the interrupt which generates at completing the oscillation stabilization wait time of the main clock (CLKMO).
- It can set the oscillation stabilization wait time of the sub clock (CLKSO).
- It can set the interrupt which generates at completing the oscillation stabilization wait time of the sub clock (CLKSO).
- It can set the oscillation stabilization wait time of the main PLL clock (CLKPLL).
- It can set the interrupt which generates at completing the oscillation stabilization wait time of the main PLL clock (CLKPLL).
- It can set the PLL multiplication ratio.
- It can select the master clock.
- It can set the frequency division ratio of each internal bus clock frequency.
- It can select run or stop of the APB1 and APB2 bus clocks.
- It can set the frequency division ratio of the software watchdog timer count clock frequency.
- It can set run/stop of the software watchdog timer count clock.
- It can set the watchdog timer count operation in debug mode.
- It includes registers for enabling clock-related interrupts, checking interrupt status, and clearing interrupt factors.

2. Clock Generation Unit Configuration/Block Diagram

This section explains configuration of the clock generation unit.

■ Source clocks

Source clock is the generic name for external and internal oscillation clocks of this MCU. The following five types of clocks are source clocks:

● Main clock (CLKMO)

CLKMO is generated by connecting a crystal oscillator etc. to the main clock oscillation pins (X0, X1), or input using an external clock.

● Sub clock (CLKSO)

CLKSO is generated by connecting a crystal oscillator etc. to the sub clock oscillator pins (X0A, X1A), or input using an external clock.

● High-speed CR clock (CLKHC)

CLKHC is an output clock for the high-speed CR oscillator.

● Low-speed CR clock (CLKLC)

CLKLC is an output clock for the low-speed CR oscillator.

(Notes) The low-speed CR clock is a clock after a prerscaler.
For details on the low-speed CR clock prescaler, see Chapter "Low-speed CR Clock Prescaler".

● Main PLL clock (CLKPLL)

CLKPLL is generated by multiplying the main clock (CLKMO) or high-speed CR clock (CLKHC) using the PLL Clock Multiplication Circuit (PLL Oscillation Circuit).

■ Master clock

The signal selected from source clocks are referred to as the master clock.
The master clock is a source for all bus clocks.

(Notes) See "1. Notes when high-speed CR is used for the master clock" in "B. List of Notes" when you use the following clock for the master clock.

- High-speed CR clock
- Main PLL clock (When selecting high-speed CR clock for the input clock of PLL)
- The master clock value should not be larger than the maximum value in "Internal operating clock frequency: Fcc(Base clock HCLK/FCLK)"of AC Specifications in "Data Sheet".

■ Internal bus clocks

The following signals are bus clocks generated internally.

● Base clock (HCLK/FCLK)

HCLK and FCLK are collectively called the base clock. Both HCLK and FCLK are supplied to the CPU.
HCLK is a clock for macro connected to the AHB bus.
The clock frequency can be set to between 1/1 and 1/16 frequency of the master clock.
This clock stops in timer mode, RTC mode, stop mode, deep standby RTC mode, and deep standby stop mode.
In sleep mode, the CPU stops the supply of HCLK while continuing the supply of FCLK.

● APB0 bus clock (PCLK0)

PCLK0 is a clock for peripheral macro connected to the APB0 bus.
The clock frequency can be set to between 1/1 and 1/8 frequency of the base clock.
This clock stops in timer mode, RTC mode, stop mode, deep standby RTC mode, and deep standby stop mode.

● APB1 bus clock (PCLK1)

PCLK1 is a clock for peripheral macro connected to the APB1 bus.
The clock frequency can be set to between 1/1 and 1/8 frequency of the base clock.
This clock stops in timer mode, RTC mode, stop mode, deep standby RTC mode, and deep standby stop mode.
The supply of the clock can be also stopped by setting a register.

● APB2 bus clock (PCLK2)

PCLK2 is a clock for peripheral macro connected to the APB2 bus.
The clock frequency can be set to between 1/1 and 1/8 frequency of the base clock.
This clock stops in timer mode, RTC mode, stop mode, deep standby RTC mode,,and deep standby stop mode.
The supply of the clock can be also stopped by setting a register.

● TPIU clock (TPIUCLK)

TPIUCLK is a clock for TRACE.
The clock frequency can be set to between 1/1 and 1/8 frequency of the base clock.
This clock stops in timer mode, RTC mode, stop mode, deep standby RTC mode and deep standby stop mode.
This clock output is enabled only for the products equipped with ETM.

■ Clocks other than source clocks and internal bus clocks

● USB clock

This clock generates a clock at 48 MHz, used by USB communication.
It sets the PLL oscillator for USB to generate a USB clock.
This clock stops in timer mode, RTC mode, stop mode, deep standby RTC mode, and deep standby stop mode.
This clock can be set independently regardless of the frequency of the master clock.
For USB clock operation settings, see Chapter "USB clock generation" in "Communication Macro Part".

● USB/Ethernet clock

This clock generates a clock at 48 MHz, used by USB communication.
Also, it generates a clock for Ethernet.
It sets the PLL oscillator for USB/Ethernet to generate a USB/Ethernet clock.
This clock stops in timer mode, RTC mode, stop mode, deep standby RTC mode, deep standby RTC mode, and deep standby stop mode.
This clock can be set independently regardless of the frequency of the master clock.
For the operation setting of PLL for USB/Ethernet, see the chapter "USB/Ethernet PLL Clock Generation" in "Communication Macro Part".

● CAN prescaler clock

This clock is the same clock as CLKPLL, used for CAN prescaler.
The frequency division used for the clock must be configured on the prescaler side.
This clock stops in RTC mode, stop mode, deep standby RTC mode, and deep standby stop mode.
The supply of the clock can be also stopped by setting a register.
For operation settings of CAN prescaler, see Chapter "CAN Prescaler" in "Communication Macro Part".

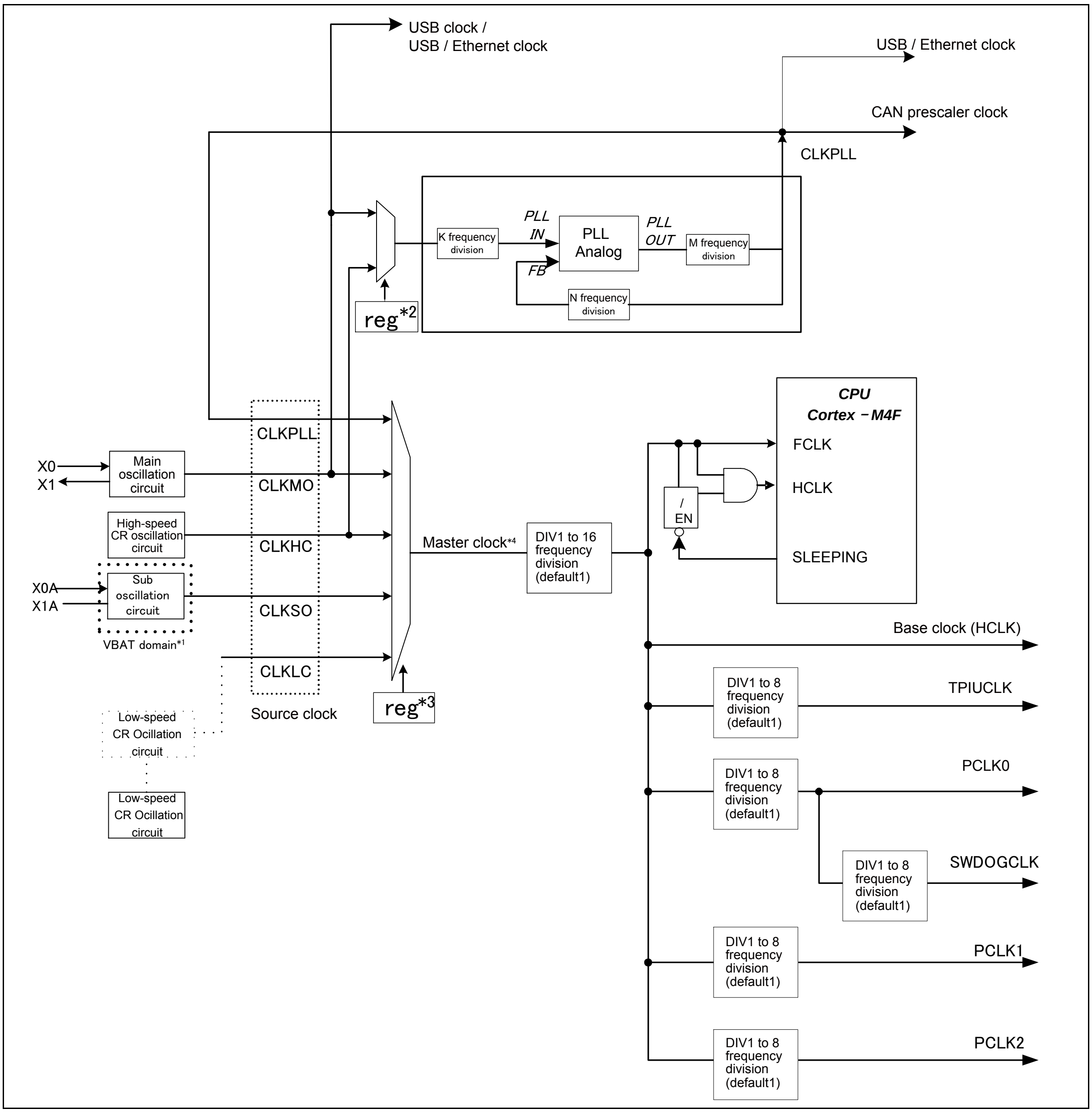
● Software watchdog timer count clock (SWDOGCLK)

SWDOGCLK is a clock for the software watchdog timer connected to the APB0 bus.
The clock frequency can be set to between 1/1 and 1/8 frequency of the APB0 bus clock.
This clock stops in timer mode, RTC mode, stop mode, deep standby RTC mode, and deep standby stop mode.
For operation settings of the software watchdog timer, see Chapter "Watchdog Timer" in "Timer Part".

■ Block diagram

Figure 2-1 shows the block diagram of the clock generation unit.

Figure 2-1 Block Diagram of Clock Generation Unit



*1: For details on VBAT Domain, see Chapter "VBAT Domain".

*2: PSW_TMR:PINC (PLL input clock select bit)

*3: SCM_CTL:RCS[2:0] (Master clock switch control bits)

*4: The master clock frequency should not be larger than the maximum frequency of base clock (HCLK / FCLK). For the maximum frequency of base clock (HCLK/FCLK), see "Data Sheet" of the product used.

3. Clock Generation Unit Operations

This section explains the clock generation unit.

3.1. Selecting the clock mode

■ Definition of clock mode (selecting the master clock)

The MCU clock mode is defined by the source clock selected by the system clock mode control register. Five types of clock modes are provided: Main clock mode, sub clock mode, high-speed CR clock mode, low-speed CR clock mode, and main PLL clock mode.

● Main clock mode

In main clock mode, the main clock (CLKMO) is used as a master clock. The clock runs a bus clock used to operate the CPU, and most peripheral functions.

Status of the PLL clock (CLKPLL) differs depending on the setting of the PLLE bit in the System Clock Mode Control Register (SCM_CTL), and the sub clock (CLKSO) depends on the SOSCE bit in the System Clock Mode Control Register (SCM_CTL). The high-speed CR clock (CLKHC) and low-speed CR clock (CLKLC) cannot be stopped by user program.

● Sub-clock mode

In sub clock mode, the sub clock (CLKSO) is used as a master clock. The clock runs a bus clock used to operate the CPU, and most peripheral functions.

The main clock (CLKMO), high-speed CR clock (CLKHC), and main PLL clock (CLKPLL) are stopped by hardware. The low-speed CR clock (CLKLC) cannot be stopped by user program.

● High-speed CR clock mode

In high-speed CR clock mode, the high-speed CR clock (CLKHC) is used as a master clock. The clock runs a bus clock used to operate the CPU, and most peripheral functions.

Statuses of the main clock (CLKMO), main PLL clock (CLKPLL), and sub clock (CLKSO) differ depending on the settings of MOSCE, PLLE, and SOSCE bits in the System Clock Mode Control Register (SCM_CTL). The high-speed CR clock (CLKHC) and low-speed CR clock (CLKLC) cannot be stopped by user program.

● Low-speed CR clock mode

In low-speed CR clock mode, the low-speed CR clock (CLKLC) is used as a master clock. The clock runs a bus clock used to operate the CPU, and most peripheral functions.

In low-speed CR clock mode, the main clock (CLKMO), high-speed CR clock (CLKHC), and main PLL clock (CLKPLL) are stopped by hardware. Status of the sub clock (CLKSO) differs depending on the setting of the SOSCE bit in the System Clock Mode Control Register (SCM_CTL).

● Main PLL clock mode

In main PLL clock mode, the main PLL clock (CLKPLL) is used as a master clock. The clock runs a bus clock used to operate the CPU, and most peripheral functions.

Status of the sub clock (CLKSO) differs depending on the setting of the SOSCE bit in the System Clock Mode Control Register (SCM_CTL). The high-speed CR clock (CLKHC) and low-speed CR clock (CLKLC) cannot be stopped by user program.

3.2. Internal bus clock frequency division control

This section explains the internal bus clock frequency division.

Frequency division ratio from the base clock can be set independently for each internal bus clock.
This function can set the operating frequency optimized for each circuit.

The maximum frequency of the internal bus clock differs by product. For details, see "Data Sheet" of the product used.

To set the frequency division ratio of internal bus clocks, use the Base Clock Prescaler Register (BSC_PSR), APB0 Prescaler Register (APBC0_PSR), APB1 Prescaler Register (APBC1_PSR), APB2 Prescaler Register (APBC2_PSR), and Trace Clock Prescaler Register (TTC_PSR). For details on each register, see "5. Lsit of Clock Generation Unit Registers".

■ Setting the bus clock frequency division

- The set frequency division ratio is not cleared by a software reset. The latest value is retained before the software reset.
- The value is initialized by a reset other than software resets.
  Before changing the initially set master clock to a faster source clock, be sure to set the frequency division ratio.
- If a combined value of master clock, PLL multiplication, and frequency division ratio settings exceeds the maximum operating frequency of each internal bus, the operation corresponding to the setting is not guaranteed.

3.3. PLL clock control

This section explains the PLL clock control.

The PLL Clock Control Circuit is used to generate the main PLL clock from the main clock or high-speed CR clock. The PLL Oscillation Circuit can enable/disable operation (oscillation), select the input clock, set the stabilization wait time, and set the multiplication.

■ PLL operation

The following explains operation of the main PLL clock.

- Configure the following settings using the PLL Clock Oscillation Stabilization Wait Time Setup Register (PSW_TMR).
  - Selecting the PLL input clock
  - Setting the main PLL clock stabilization wait time
- The PLL oscillation enable bit (PLLE) of the System Clock Mode Control Register (SCM_CTL) must be enabled to let the PLL Circuit start oscillating.
- When the PLL clock stabilization wait time has elapsed, and the "PLL oscillation stable bit" of the System Clock Mode Status Register (SCM_STR) indicates a stable state, the preparation for transition to main PLL clock mode completes.
- Master clock switch control bit (RCS[2:0]) of the System Clock Mode Control Register (SCM_CTL) must be set to main PLL clock mode (RCS[2:0]=010) to change to main PLL clock mode.

■ Setting the main PLL clock oscillation stabilization wait time

The details are given in "5.10 PLL Clock Stabilization Wait Time Setup Register (PSW_TMR)".

<Notes>

- For block diagram of the PLL Clock Control Circuit, see "2. Clock Generation Unit Configuration/Block Diagram".
- For the order of frequency division settings for each internal bus clock, see "4 Clock Setup Procedure Examples".
- For the oscillation stabilization wait time, see "3.4 Oscillation stabilization wait time".
- When selecting high-speed CR in the input clock of PLL, see "1. Notes when high-speed CR is used for the master clock"in "B. List of Notes" of "APPENDIXES".

■ Setting the multiplication ratio to generate the main PLL clock

Each frequency division clock in the PLL Multiplication Circuit must be set using PLL Control Register 1 (PLL_CTL1) and PLL Control Register 2 (PLL_CTL2). The following Table 3-1 provides example of frequency division settings.

Table 3-1 Example of PLL multiplication ratio settings

Input clock	K	PLLin	N	PLLout	M	CLKPLL
4MHz	1	4MHz	20	320MHz	4	80MHz
4MHz	1	4MHz	30	240MHz	2	120MHz
4MHz	1	4MHz	40	320MHz	2	160MHz
5MHz	1	5MHz	24	240MHz	2	120MHz
5MHz	1	5MHz	30	300MHz	2	150MHz
5MHz	1	5MHz	32	320MHz	2	160MHz
6MHz	1	6MHz	20	240MHz	2	120MHz
6MHz	1	6MHz	25	300MHz	2	150MHz
8MHz	1	8MHz	20	320MHz	2	160MHz
10MHz	1	10MHz	8	320MHz	4	80MHz
10MHz	1	10MHz	16	320MHz	2	160MHz
10MHz	1	10MHz	15	300MHz	2	150MHz
12MHz	1	12MHz	10	240MHz	2	120MHz
12MHz	1	12MHz	12	288MHz	2	144MHz
15MHz	1	15MHz	10	300MHz	2	150MHz
16MHz	1	16MHz	10	320MHz	2	160MHz
20MHz	2	10MHz	10	200MHz	2	100MHz
30MHz	2	15MHz	10	300MHz	2	150MHz
40MHz	4	10MHz	15	300MHz	2	150MHz
48MHz	3	16MHz	10	320MHz	2	160MHz
48MHz	4	12MHz	12	288MHz	2	144MHz

<Notes>

- For PLL characteristics, see "Data Sheet" of the product used.
- Set the PLLin within the value "PLL input clock frequency: f_{PLLI}" shown in the "Data Sheet".
- The value "M×N" is a multiplication ratio for the PLLin. Set this value within the range shown in the "PLL multiple rate" of the "Data Sheet".
- The frequency of the PLLin multiplied by "M×N" becomes PLLout. Set this value within the range shown in the "PLL macro oscillation clock frequency: f_{PLLO}" of the "Data Sheet".
- The value of the PLLout divided by "M" becomes CLKPLL.
  See Figure 2-1 for the configurations of PLL and divider.
- The master clock value should not be larger than the maximum value in "Internal operating clock frequency: Fcc (Base clock HCLK/FCLK)" of "Data Sheet"

3.4. Oscillation stabilization wait time

This section explains the oscillation stabilization wait time.

An oscillation stabilization wait time is required if the source clock is not in a stable operating state. During the oscillation stabilization wait time, internal and external clocks stop the supply, only the internal time counter operates to wait until the stabilization wait time passes, a time value set in the Clock Stabilization Wait Time Register (CSW_TMR) or PLL Clock Oscillation Stabilization Wait Time Setup Register (PSW_TMR). When the wait time has been passed, the corresponding oscillator is ready to operate, and the clock can be used as a master clock.

■ Setting the oscillation stabilization wait time

- Main clock (CLKMO)
  Set the stabilization wait time of the main clock using the Clock Stabilization Wait Time Register (CSW_TMR). The set time value is counted by CLKHC.

- Sub clock (CLKSO)
  Set the stabilization wait time of the sub clock using the Clock Stabilization Wait Time Register (CSW_TMR). The set time value is counted by CLKLC.

- Main PLL clock
  Configure the following settings using the PLL Clock Oscillation Stabilization Wait Time Setup Register (PSW_TMR). The set time value is counted by CLKHC.

  - Selecting the PLL input clock
  - Setting the main PLL clock stabilization wait time

■ Cause of waiting for oscillation stability

- After the oscillation is enabled via software
  If the PLLE, SOSCE, and MOSCE bits of the System Clock Mode Control Register (SCM_CTL) are set to "1", each relevant oscillator waits during the oscillation stabilization wait time.

- When returning to watch counter interrupt, RTC interrupt, and external interrupt from RTC mode
  It returns to the clock mode before RTC mode by watch counter interrupt, RTC interrupt, and external interrupt. Since a source clock other than the sub clock is stopped in RTC mode, hardware of a source clock other than the sub clock waits for the oscillation stabilization wait time automatically.

- When returning from stop mode using an external interrupt
  The status returns to clock mode, a state before stop mode, using an external clock. During stop mode, all source clocks stop and, therefore, the hardware automatically waits during the oscillation stabilization wait time.

- After PLL operation is enabled
  After PLL operation is enabled, the PLL oscillation stabilization wait time is waited.

<Notes>

- Each set value of the oscillation stabilization wait time must be changed before the clock is enabled.
- After software reset, the oscillation stabilization wait time is not applied.
- In the stabilization wait time for main clock, sub clock and main PLL clock, the high-speed CR clock (CLKHC) counts the clock as set in the Stabilization Wait Time Setup Registers. The oscillation stabilization wait completion flag will be activated when the counting is complete, so these wait times are independent of each oscillator statuses. The oscillation stabilization wait time may be completed before oscillator stabilization if the setting of the oscillation stabilization wait time is too short.
- As the stabilization wait times for main clock and sub clock oscillators depend on the type of the oscillator (crystal, ceramics, etc.), proper oscillation stabilization wait time must be chosen for the oscillator to be used.
- Set the PLL oscillation stabilization wait time by referring to PLL Clock LOCKUP Time of the electric characteristics described in "Data Sheet" of the product used.

3.5. Interrupt Factors

This section explains interrupt factors relevant to clocks.

The clock generation unit has the following interrupt factors.

■ Interrupt factors

The clock generation unit has the following four types of interrupt factors:

· FCS (anomalous frequency detection) interrupt
When the FCS (anomalous frequency detection) is enabled, and an anomalous frequency of the main clock is detected, an interrupt occurs.

· Main PLL clock oscillation stabilization wait completion interrupt
When the main PLL clock oscillation stabilization wait time ends, an interrupt occurs.

· Sub clock oscillation stabilization wait completion interrupt
When the sub clock oscillation stabilization wait time ends, an interrupt occurs.

· Main clock oscillation stabilization wait completion interrupt
When the main clock oscillation stabilization wait time ends, an interrupt occurs.

■ Registers

The following three types of registers are provided for each interrupt factor:

· Interrupt Enable Register (INT_ENR)
This register enables each interrupt.

· Interrupt Status Register (INT_STR)
This register indicates each interrupt status. This register is read-only.

· Interrupt Clear Register (INT_CLR)
This register clears each interrupt factor. This register is write-only.

4. Clock Setup Procedure Examples

This section explains procedure examples of setting up clocks.

■ Setup procedure examples

Figure 4-1 Example of clock setup procedure (Power-on -> High-speed CR run mode -> Desired clock mode)

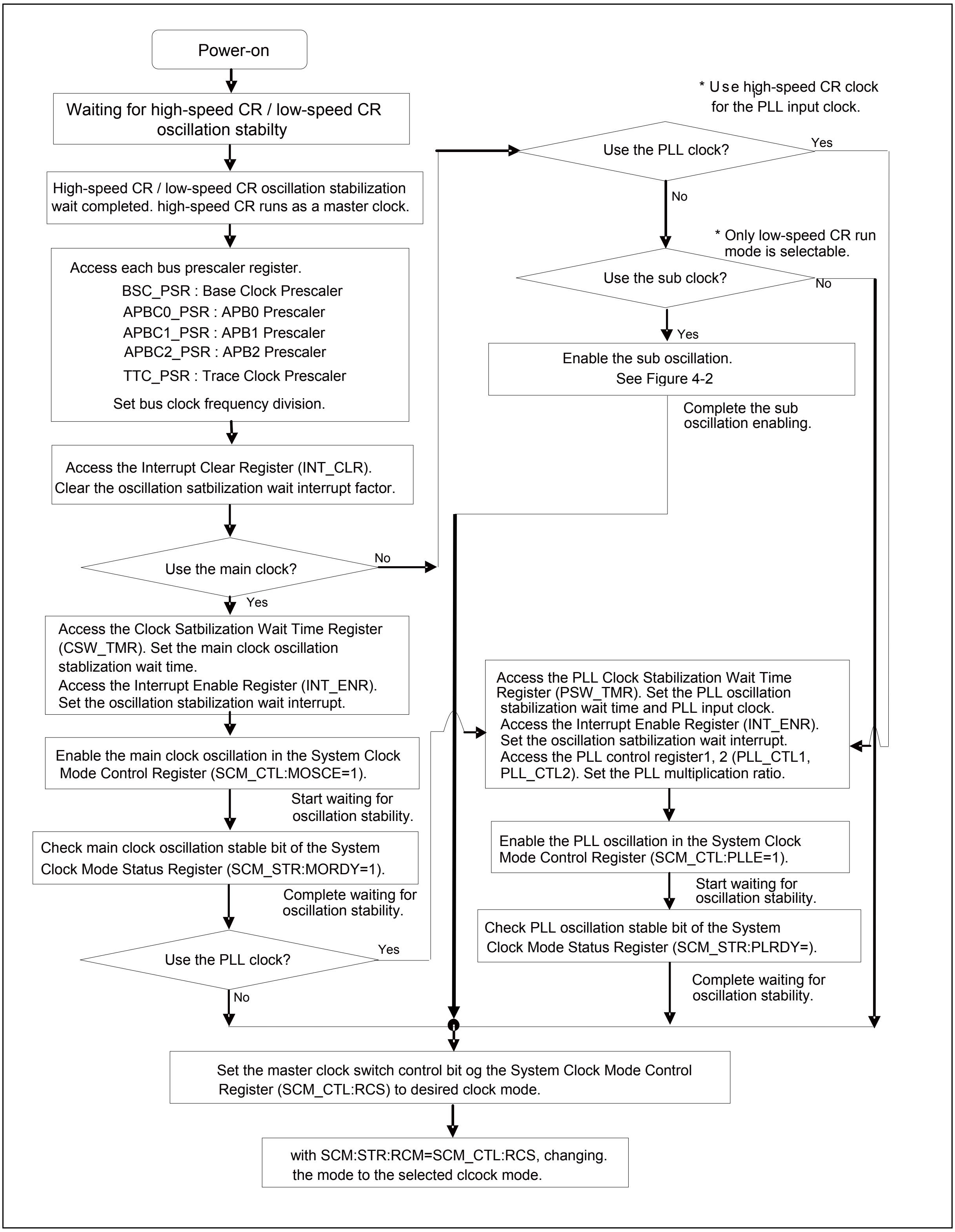
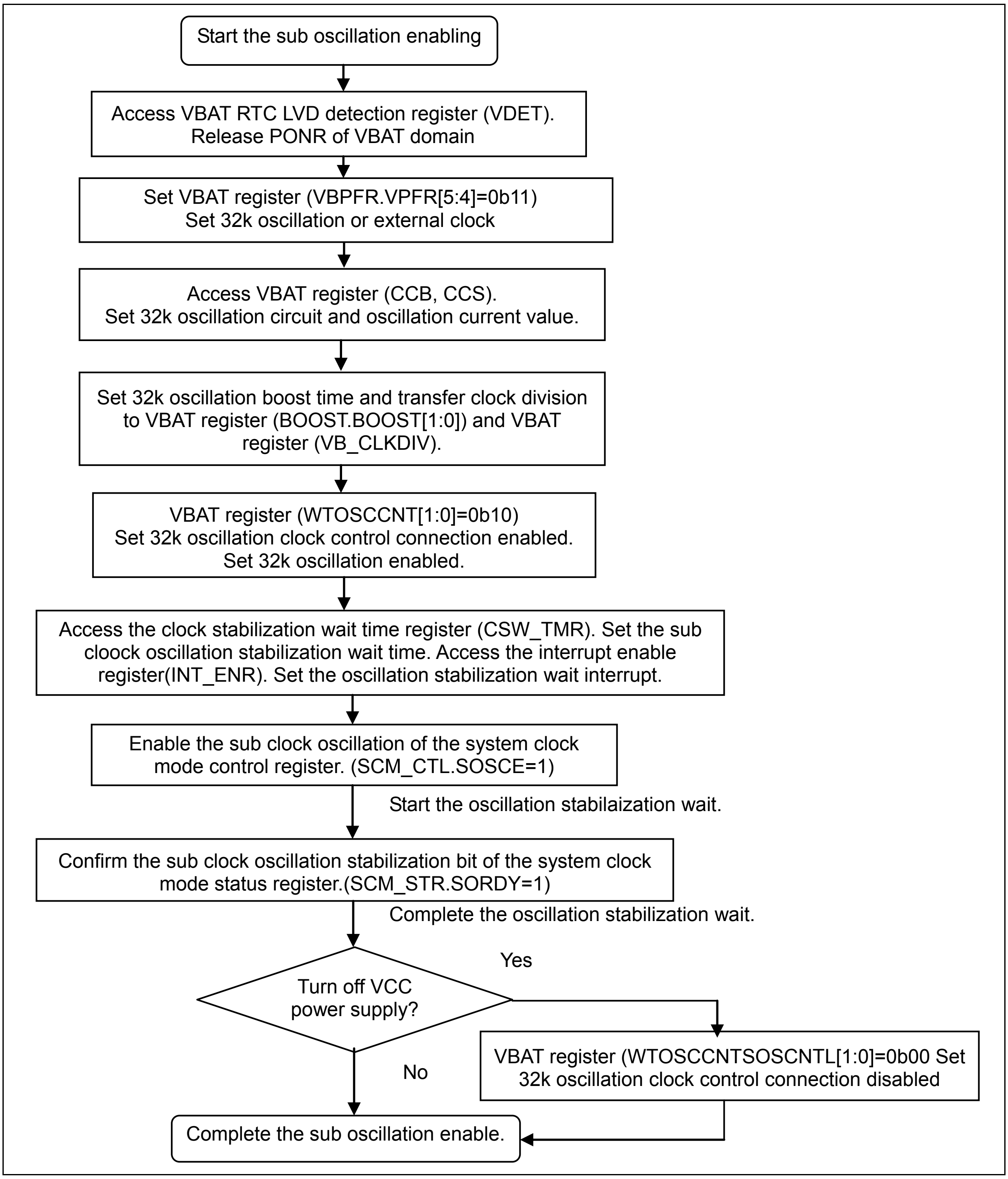

Power-on
Waiting for high-speed CR / low-speed CR oscillation stabilty
High-speed CR / low-speed CR oscillation stabilization wait completed. high-speed CR runs as a master clock.
Access each bus prescaler register.
BSC_PSR : Base Clock Prescaler
APBC0_PSR : APB0 Prescaler
APBC1_PSR : APB1 Prescaler
APBC2_PSR : APB2 Prescaler
TTC_PSR : Trace Clock Prescaler
Set bus clock frequency division.
Access the Interrupt Clear Register (INT_CLR).
Clear the oscillation satbilization wait interrupt factor.
Use the main clock?
No
Yes
Access the Clock Satbilization Wait Time Register (CSW_TMR). Set the main clock oscillation stablization wait time.
Access the Interrupt Enable Register (INT_ENR).
Set the oscillation stabilization wait interrupt.
Enable the main clock oscillation in the System Clock Mode Control Register (SCM_CTL:MOSCE=1).
Start waiting for oscillation stability.
Check main clock oscillation stable bit of the System Clock Mode Status Register (SCM_STR:MORDY=1).
Complete waiting for oscillation stability.
Use the PLL clock?
Yes
No
* Use high-speed CR clock for the PLL input clock.
Use the PLL clock?
Yes
No
* Only low-speed CR run mode is selectable.
Use the sub clock?
No
Yes
Enable the sub oscillation.
See Figure 4-2
Complete the sub oscillation enabling.
Access the PLL Clock Stabilization Wait Time Register (PSW_TMR). Set the PLL oscillation stabilization wait time and PLL input clock.
Access the Interrupt Enable Register (INT_ENR).
Set the oscillation satbilization wait interrupt.
Access the PLL control register1, 2 (PLL_CTL1, PLL_CTL2). Set the PLL multiplication ratio.
Enable the PLL oscillation in the System Clock Mode Control Register (SCM_CTL:PLLE=1).
Start waiting for oscillation stability.
Check PLL oscillation stable bit of the System Clock Mode Status Register (SCM_STR:PLRDY=).
Complete waiting for oscillation stability.
Set the master clock switch control bit og the System Clock Mode Control Register (SCM_CTL:RCS) to desired clock mode.
with SCM:STR:RCM=SCM_CTL:RCS, changing.
the mode to the selected clcock mode.

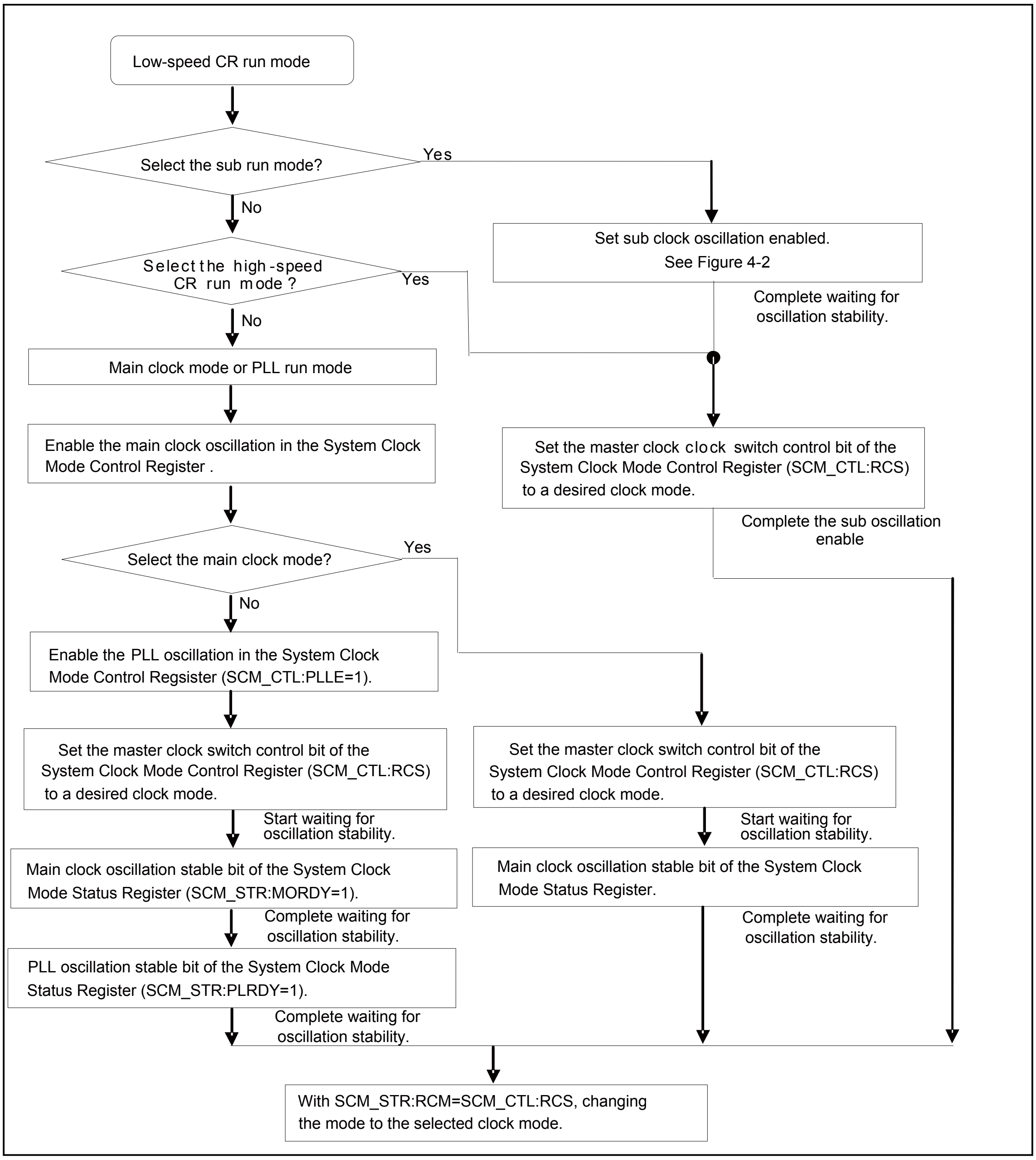
Figure 4-2 Example of sub oscillation setup procedure


Start the sub oscillation enabling
Access VBAT RTC LVD detection register (VDET). Release PONR of VBAT domain
Set VBAT register (VBPFR.VPFR[5:4]=0b11) Set 32k oscillation or external clock
Access VBAT register (CCB, CCS). Set 32k oscillation circuit and oscillation current value.
Set 32k oscillation boost time and transfer clock division to VBAT register (BOOST.BOOST[1:0]) and VBAT register (VB_CLKDIV).
VBAT register (WTOSCCNT[1:0]=0b10) Set 32k oscillation clock control connection enabled. Set 32k oscillation enabled.
Access the clock stabilization wait time register (CSW_TMR). Set the sub cloock oscillation stabilization wait time. Access the interrupt enable register(INT_ENR). Set the oscillation stabilization wait interrupt.
Enable the sub clock oscillation of the system clock mode control register. (SCM_CTL.SOSCE=1)
Start the oscillation stabilaization wait.
Confirm the sub clock oscillation stabilization bit of the system clock mode status register.(SCM_STR.SORDY=1)
Complete the oscillation stabilization wait.
Turn off VCC power supply?
Yes
No
VBAT register (WTOSCCNTSOSCNTL[1:0]=0b00 Set 32k oscillation clock control connection disabled
Complete the sub oscillation enable.

<Notes>

- Set the sub clock stabilization wait time (SCM_CTL.SOWT) longer than the VBAT RTC 32 k oscillation boost setting time [BOOST:BOOST[1:0]].
- The following combination of settings is prohibited:
  - When 32k oscillation clock control linkage of VBAT register is disabled (WTOSCCNT.SOSCNTL=0) and 32k oscillation is disabled (WTOSCCNT.SOSCEX=1), the setting combination of sub clock mode oscillation of system clock mode control register enabled (SCM_CTL.SOSCE=1) and sub CSV function of CSV control register enabled (CSV_CTL.SVSVE=1) is prohibited.
- For details of VBAT RTC, see Chapter "VBAT Domain".

Figure 4-3 Example of clock setup procedure (Low-speed CR run mode -> Desired clock run mode)


Low-speed CR run mode
Select the sub run mode?
Yes
No
Set sub clock oscillation enabled.
See Figure 4-2
Select the high-speed CR run mode ?
Yes
No
Complete waiting for oscillation stability.
Main clock mode or PLL run mode
Enable the main clock oscillation in the System Clock Mode Control Register .
Set the master clock clock switch control bit of the System Clock Mode Control Register (SCM_CTL:RCS) to a desired clock mode.
Complete the sub oscillation enable
Select the main clock mode?
Yes
No
Enable the PLL oscillation in the System Clock Mode Control Regsister (SCM_CTL:PLLE=1).
Set the master clock switch control bit of the System Clock Mode Control Register (SCM_CTL:RCS) to a desired clock mode.
Set the master clock switch control bit of the System Clock Mode Control Register (SCM_CTL:RCS) to a desired clock mode.
Start waiting for oscillation stability.
Start waiting for oscillation stability.
Main clock oscillation stable bit of the System Clock Mode Status Register (SCM_STR:MORDY=1).
Main clock oscillation stable bit of the System Clock Mode Status Register.
Complete waiting for oscillation stability.
Complete waiting for oscillation stability.
PLL oscillation stable bit of the System Clock Mode Status Register (SCM_STR:PLRDY=1).
Complete waiting for oscillation stability.
With SCM_STR:RCM=SCM_CTL:RCS, changing the mode to the selected clock mode.

<Notes>

- Figure 4-3 assumes that settings of the oscillation stabilization wait time, interrupts, PLL multiplication ratio and bus clock frequency division for each clock have been configured previously, and they are omitted from the flowchart.
- In the sub clock mode/low-speed CR clock mode, the main clock(CLKMO), high-speed CR clock (CLKHC), main PLL clock(CLKPLL) is stopped by hardware. So CLKMO/CLKHC/CLKPLL does not start oscillation only setting oscillation enable bit=1. These oscillations will start by changing the SCM_CTL:RCS bit with setting oscillation enable bit=1.
- If the main clock/sub clock oscillation stabilization wait times are short and the oscillation stabilization wait times run out before oscillators stabilize, reset may be applied by the clock supervisor function.

5. Lsit of Clock Generation Unit Registers

This section provides the register list of clock generation.

■ Clock generation unit register list

Abbreviation	Register name	Reference
SCM_CTL	System Clock Mode Control Register	5.1
SCM_STR	System Clock Mode Status Register	5.2
BSC_PSR	Base Clock Prescaler Register	5.3
APBC0_PSR	APB0 Prescaler Register	5.4
APBC1_PSR	APB1 Prescaler Register	5.5
APBC2_PSR	APB2 Prescaler Register	5.6
SWC_PSR	Software Watchdog Clock Prescaler Register	5.7
TTC_PSR	Trace Clock Prescaler Register	5.8
CSW_TMR	Clock Stabilization Wait Time Register	5.9
PSW_TMR	PLL Clock Stabilization Wait Time Setup Register	5.10
PLL_CTL1	PLL Control Register 1	5.11
PLL_CTL2	PLL Control Register 2	5.12
DBWDT_CTL	Debug Break Watchdog Timer Control Register	5.13
INT_ENR	Interrupt Enable Register	5.14
INT_STR	Interrupt Status Register	5.15
INT_CLR	Interrupt Clear Register	5.16

5.1. System Clock Mode Control Register (SCM_CTL)

The SCM_CTL selects the master clock and enables/disables the clock oscillation.

■ Register configuration

bit	7	6	5	4	3	2	1	0
Field	RCS[2:0]			PLLE	SOSCE	Reserved	MOSCE	Reserved
Attribute	R/W			R/W	R/W	-	R/W	-
Initial value	000			0	0	-	0	-

■ Register functions

[bit7:5] RCS[2:0]: Master clock switch control bits

bit7	bit6	bit5	Description
0	0	0	High-speed CR clock [Initial value]
0	0	1	Main clock
0	1	0	Main PLL clock
0	1	1	Setting is prohibited
1	0	0	Low-speed CR clock
1	0	1	Sub clock
1	1	0	Setting is prohibited
1	1	1	Setting is prohibited

[bit4] PLLE: PLL oscillation enable bit

bit	Description
0	Disables PLL oscillation [Initial value]
1	Enables PLL oscillation

[bit3] SOSCE: Sub clock oscillation enable bit

bit	Description
0	Disables sub clock oscillation [Initial value]
1	Enables sub clock oscillation

[bit2] Reserved: Reserved bit

"0" is read from this bit.
Set this bit to "0" when writing.

[bit1] MOSCE: Main clock oscillation enable bit

bit	Description
0	Disables main clock oscillation [Initial value]
1	Enables main clock oscillation

[bit0] Reserved: Reserved bit

"0" is read from this bit.
Set this bit to "0" when writing.

<Notes>

- This register is not initialized by software reset.
- When you change the clock mode, you should set the enable bit to transition for desired clock oscillation. Then, you can change the clock switch control bits (SCM_CTL:RCS[2:0]).
- When RTCE bit (PMD_CTL:RTCE) is "1", it becomes a sub clock oscillation enable state regardless of the SOSCE bit and SORDY bit values.
- Writing "1" to RTCE bit (PMD_CTL:RTCE) is enabled only when SORDY bit is "1".
- RTCE bit (PMD_CTL:RTCE) does not exist in the products that do not have RTC mode and deep standby RTC mode. See Table 1-1 in the Chapter "Low Power Consumption Mode".
- First of all, after the power supply is turned on, it is required to set the register of VBAT RTC for enabling the sub-clock oscillation. For the sub-clock oscillation enable, see Figure 4-2.

5.2. System Clock Mode Status Register (SCM_STR)

The SCM_STR indicates a clock selected for the master clock and a waiting state for clock oscillation stability.

■ Register configuration

bit	7	6	5	4	3	2	1	0
Field	RCM[2:0]			PLRDY	SORDY	Reserved	MORDY	Reserved
Attribute	R			R	R	-	R	-
Initial value	000			0	0	-	0	-

■ Register functions

[bit7:5] RCM[2:0]: Master clock selection bits

bit7	bit6	bit5	Description
0	0	0	High-speed CR clock [Initial value]
0	0	1	Main clock
0	1	0	Main PLL clock
0	1	1	Setting is prohibited
1	0	0	Low-speed CR clock
1	0	1	Sub clock
1	1	0	Setting is prohibited
1	1	1	Setting is prohibited

[bit4] PLRDY: PLL oscillation stable bit

bit	Description
0	In a stabilization wait or an oscillation stop state [Initial value]
1	In a stable state

[bit3] SORDY: Sub clock oscillation stable bit

bit	Description
0	In a stabilization wait or an oscillation stop state [Initial value]
1	In a stable state

[bit2] Reserved: Reserved bit

"0" is read from this bit.

[bit1] MORDY: Main clock oscillation stable bit

bit	Description
0	In a stabilization wait or an oscillation stop state [Initial value]
1	In a stable state

[bit0] Reserved: Reserved bit

"0" is read from this bit.

<Notes>

- This register is not initialized by software reset.
- When RTCE bit (PMD_CTL:RTCE) of RTC mode control register (PMD_CTL) is "1", it becomes a sub clock oscillation enable state regardless of the SOSCE bit and SORDY bit values.
- Writing "1" to RTCE bit (PMD_CTL:RTCE) is enabled only when SORDY bit is "1".

5.3. Base Clock Prescaler Register (BSC_PSR)

The BSC_PSR sets the frequency division ratio of the base clock.

■ Register configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved					BSR		
Attribute	-					R/W		
Initial value	-					000		

■ Register functions

[bit7:3] Reserved: Reserved bits

"0b00000" is read from these bits.
Set these bits to "0b00000" when writing.

[bit2:0] BSR: Base clock frequency division ratio setting bits

bit2	bit1	bit0	Description
0	0	0	1/1 [Initial value]
0	0	1	1/2
0	1	0	1/3
0	1	1	1/4
1	0	0	1/6
1	0	1	1/8
1	1	0	1/16
1	1	1	Setting is prohibited

<Note>

This register is not initialized by software reset.

5.4. APB0 Prescaler Register (APBC0_PSR)

The APBC0_PSR sets the APB0 bus clock frequency division.

■ Register configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved						APBC0	
Attribute	-						R/W	
Initial value	-						00	

■ Register functions

[bit7:2] Reserved: Reserved bits

"0b000000" is read from these bits.
Set these bits to "0b000000" when writing.

[bit1:0] APBC0: APB0 bus clock frequency division setting bits

bit1	bit0	Description
0	0	1/1 [Initial value]
0	1	1/2
1	0	1/4
1	1	1/8

<Note>

This register is not initialized by software reset.

5.5. APB1 Prescaler Register (APBC1_PSR)

The APBC1_PSR sets the APB1 bus clock frequency division.

■ Register configuration

bit	7	6	5	4	3	2	1	0
Field	APBC1EN	Reserved		APBC1RST	Reserved		APBC1	
Attribute	R/W	-		R/W	-		R/W	
Initial value	1	-		0	-		00	

■ Register functions

[bit7] APBC1EN: APB1 clock enable bit

bit	Description
0	Disables PCLK1 output
1	Enables PCLK1 output [Initial value]

[bit6:5] Reserved: Reserved bits

"0b00" is read from these bits.
Set these bits to "0b00" when writing.

[bit4] APBC1RST: APB1 bus reset control bit

bit	Description
0	APB1 bus reset, inactive [Initial value]
1	APB1 bus reset, active

[bit3:2] Reserved: Reserved bits

"0b00" is read from these bits.
Set these bits to "0b00" when writing.

[bit1:0] APBC1: APB1 bus clock frequency division setting bits

bit1	bit0	Description
0	0	1/1 [Initial value]
0	1	1/2
1	0	1/4
1	1	1/8

<Note>

This register is not initialized by software reset.

5.6. APB2 Prescaler Register (APBC2_PSR)

The APBC2_PSR sets the APB2 bus clock frequency division.

■ Register configuration

bit	7	6	5	4	3	2	1	0
Field	APBC2EN	Reserved		APBC2RST	Reserved		APBC2	
Attribute	R/W	-		R/W	-		R/W	
Initial value	1	-		0	-		00	

■ Register functions

[bit7] APBC2EN: APB2 clock enable bit

bit	Description
0	Disables PCLK2 output
1	Enables PCLK2 output [Initial value]

[bit6:5] Reserved: Reserved bits

"0b00" is read from these bits.
Set these bits to "0b00" when writing.

[bit4] APBC2RST: APB2 bus reset control bit

bit	Description
0	APB2 bus reset, inactive [Initial value]
1	APB2 bus reset, active

[bit3:2] Reserved: Reserved bits

"0b00" is read from these bits.
Set these bits to "0b00" when writing.

[bit1:0] APBC2: APB2 bus clock frequency division setting bits

bit1	bit0	Description
0	0	1/1 [Initial value]
0	1	1/2
1	0	1/4
1	1	1/8

<Note>

This register is not initialized by software reset.

5.7. Software Watchdog Clock Prescaler Register (SWC_PSR)

The SWC_PSR sets the frequency division and enables the output of the software watchdog clock.

■ Register configuration

bit	7	6	5	4	3	2	1	0
Field	TESTB	Reserved					SWDS	
Attribute	R/W	-					R/W	
Initial value	X	-					00	

■ Register functions

[bit7] TESTB: TEST bit

bit	Description
0	Setting is prohibited
1	Always written by "1".

Note: The read value of this bit is undefined.

[bit6:2] Reserved: Reserved bits

"0b00000" is read from these bits.
Set these bits to "0b00000" when writing.

[bit1:0] SWDS: Software watchdog clock frequency division ratio setting bits

bit1	bit0	Description
0	0	Sets 1/1 frequency of PCLK0. [Initial value]
0	1	Sets 1/2 frequency of PCLK0.
1	0	Sets 1/4 frequency of PCLK0.
1	1	Sets 1/8 frequency of PCLK0.

<Notes>

- This register is not initialized by software reset.
- Be sure to set the TESTB bit to "1" when writing a value to this register.

5.8. Trace Clock Prescaler Register (TTC_PSR)

The TTC_PSR sets the trace clock frequency division.

■ Register configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved						TTC	
Attribute	-						R/W	
Initial value	-						00	

■ Register functions

[bit7:2] Reserved: Reserved bits

"0b000000" is read from this bit.
Set these bits to "0b000000" when writing.

[bit1:0] TTC: Trace clock divide ratio setting bits

bit1	bit0	Description
0	0	1/1 [Initial value]
0	1	1/2
1	0	1/4
1	1	1/8

<Note>

This register is not initialized by software reset.

5.9. Clock Stabilization Wait Time Register (CSW_TMR)

The CSW_TMR sets the stabilization wait time of the main/sub clock.

■ Register configuration

bit	7	6	5	4	3	2	1	0
Field	SOWT				MOWT			
Attribute	R/W				R/W			
Initial value	0000				0000			

■ Register functions

[bit7:4] SOWT: Sub clock stabilization wait time setup bits

bit7	bit6	bit5	bit4	Description
0	0	0	0	2^{10}/ F_{CRL} : Approx. 10.3 ms * [Initial value]
0	0	0	1	2^{11}/ F_{CRL} : Approx. 20.5 ms *
0	0	1	0	2^{12}/ F_{CRL} : Approx. 41 ms *
0	0	1	1	2^{13}/ F_{CRL} : Approx. 82 ms *
0	1	0	0	2^{14}/ F_{CRL} : Approx. 164 ms *
0	1	0	1	2^{15}/ F_{CRL} : Approx. 327 ms *
0	1	1	0	2^{16}/ F_{CRL} : Approx. 655 ms *
0	1	1	1	2^{17}/F_{CRL} : Approx.1.31s *
1	0	0	0	2^{18}/ F_{CRL} : Approx. 2.62s *
1	0	0	1	2^{19}/ F_{CRL} : Approx. 5.24s *
1	0	1	0	2^{20}/ F_{CRL} : Approx. 10.48s *
1	0	1	1	2^{21}/ F_{CRL} : Approx. 20.96s *
Other than the above				Setting is prohibited.

*: When F_{CRL}=100 kHz

[bit3:0] MOWT: Main clock stabilization wait time setup bits

bit3	bit2	bit1	bit0	Description
0	0	0	0	2^{1} / F_{CRH} : Approx. 500 ns * [Initial value]
0	0	0	1	2^{5} / F_{CRH} : Approx. 8 µs *
0	0	1	0	2^{6} / F_{CRH} : Approx. 16 µs *
0	0	1	1	2^{7} / F_{CRH} : Approx. 32 µs *
0	1	0	0	2^{8} / F_{CRH} : Approx. 64 µs *
0	1	0	1	2^{9} / F_{CRH} : Approx. 128 µs *
0	1	1	0	2^{10}/ F_{CRH} : Approx. 256 µs *
0	1	1	1	2^{11}/ F_{CRH} : Approx. 512 µs *
1	0	0	0	2^{12}/ F_{CRH} : Approx. 1.0 ms *
1	0	0	1	2^{13}/ F_{CRH} : Approx. 2.0 ms *
1	0	1	0	2^{14}/ F_{CRH} : Approx. 4.0 ms *
1	0	1	1	2^{15}/ F_{CRH} : Approx. 8.0 ms *
1	1	0	0	2^{17}/ F_{CRH} : Approx. 33.0 ms *
1	1	0	1	2^{19}/ F_{CRH} : Approx. 131 ms *
1	1	1	0	2^{21}/ F_{CRH} : Approx. 524 ms *
1	1	1	1	2^{23}/ F_{CRH} : Approx. 2.0 s *

*: When F_{CRH}=4 MHz

<Notes>

- Set each oscillation stabilization wait time before enabling each oscillation enable bit (SOSCE, MOSCE) of the SCM_CTL register.
  If you change MOWT or SOWT bit while waiting for oscillation stability of each oscillator, each oscillation stabilization wait time is not guaranteed.
- This register is not initialized by software reset.

5.10. PLL Clock Stabilization Wait Time Setup Register (PSW_TMR)

The PSW_TMR sets the main PLL clock stabilization wait time.

■ Register configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved			PINC	Reserved	POWT		
Attribute	-			R/W	-	R/W		
Initial value	-			0	-	000		

■ Register functions

[bit7:5] Reserved: Reserved bits

"0b000" is read from these bits.
Set these bits to "0b000" when writing.

[bit4] PINC: PLL input clock select bit

bit	Description
0	Selects CLKMO (main clock oscillation) [Initial value]
1	Selects CLKHC (high-speed CR clock)

Note: Setting this bit to "1" has some restrictions.
See "1. Notes when high-speed CR is used for the master clock" in "B. List of Notes" of "APPENDIXES".

[bit3] Reserved: Reserved bit

"0" is read from this bit.
Set this bit to "0" when writing.

[bit2:0] POWT: Main PLL clock stabilization wait time setup bits

bit2	bit1	bit0	Description
0	0	0	$2^{9}/ F_{CRH}$: Approx. 128 μs * [Initial value]
0	0	1	$2^{10}/ F_{CRH}$: Approx. 256 μs *
0	1	0	$2^{11}/ F_{CRH}$: Approx. 512 μs *
0	1	1	$2^{12}/ F_{CRH}$: Approx. 1.02 ms *
1	0	0	$2^{13}/ F_{CRH}$: Approx. 2.05 ms *
1	0	1	$2^{14}/ F_{CRH}$: Approx. 4.10 ms *
1	1	0	$2^{15}/ F_{CRH}$: Approx. 8.20 ms *
1	1	1	$2^{16}/ F_{CRH}$: Approx. 16.40 ms *

*: When F_{CRH}=4 MHz

<Notes>

- Set each oscillation stabilization wait time before enabling the PLL oscillation enable bit (PLLE) of the SCM_CTL.
  If you change POWT bit while waiting for oscillation stability of the PLL oscillator, the oscillation stabilization wait time is not guaranteed.
- This register is not initialized by software reset.

5.11. PLL Control Register 1 (PLL_CTL1)

The PLL_CTL1 sets the PLL frequency division ratio.

■ Register configuration

bit	7	6	5	4	3	2	1	0
Field	PLLK				PLLM			
Attribute	R/W				R/W			
Initial value	0000				0000			

■ Register functions

[bit7:4] PLLK: PLL input clock frequency division ratio setting bits

bit 7:4	Description
0000	The frequency division is (PLLK value +1). (Frequency division : 1 to 16) Example: PLLK value (0000) +1 => 1/1 frequency [Initial value]
0001	
•	
•	
1111	

[bit3:0] PLLM: PLL VCO clock frequency division ratio setting bits

bit3:0	Description
0000	The frequency division is (PLLM value +1). (Frequency division : 1 to 16) Example: PLLM value (0000) +1 => 1/1 frequency [Initial value]
0001	
•	
•	
1111	

<Notes>

- Set each frequency division ratio before enabling the PLL oscillation enable bit (PLLE) of the SCM_CTL register.
- This register is not initialized by software reset.

5.12. PLL Control Register 2 (PLL_CTL2)

The PLL_CTL2 sets the PLL frequency division ratio.

Register configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved		PLLN					
Attribute	-		R/W					
Initial value	-		000000					

● Register functions

[bit7:6] Reserved: Reserved bits

"0b00" is read from these bits.
Set these bits to "0b00" when writing.

[bit5:0] PLLN: PLL feedback frequency division ratio setting bits

bit5:0	Description
000000	The frequency division is (PLLN value +1). (Frequency division : 1 to 64) Example: PLLN value (000000) +1 => 1/1 division [Initial value]
000001	
.	
.	
110001	
110010	
.	
111111	

<Notes>

- Set the frequency division ratio before enabling the PLL oscillation enable bit (PLLE) of the SCM_CTL register.
- This register is not initialized by software reset.

5.13. Debug Break Watchdog Timer Control Register (DBWDT_CTL)

The DBWDT_CTL sets the watchdog timer count operation for debug mode tool break.

■ Register configuration

bit	7	6	5	4	3	2	1	0
Field	DPHWBE	Reserved	DPSWBE	Reserved				
Attribute	R/W	-	R/W	-				
Initial value	0	-	0	-				

■ Register functions

[bit7] DPHWBE: HW-WDG debug mode break bit

bit	Description
0	HW-WDG stops counting at the tool break [Initial value]
1	HW-WDG continues counting at the tool break

[bit6] Reserved: Reserved bit

"0" is read from this bit.
Set this bit to "0" when writing.

[bit5] DPSWBE: SW-WDG debug mode break bit

bit	Description
0	SW-WDG stops counting at the tool break [Initial value]
1	SW-WDG continues counting at the tool break

[bit4:0] Reserved: Reserved bits

"0b00000" is read from these bits.
Set these bits to "0b00000" when writing.

<Note>

This register is not initialized by software reset.

5.14. Interrupt Enable Register (INT_ENR)

The INT_ENR enables/disables interrupts.

■ Register configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved		FCSE	Reserved		PCSE	SCSE	MCSE
Attribute	-		R/W	-		R/W	R/W	R/W
Initial value	-		0	-		0	0	0

■ Register functions

[bit7:6] Reserved: Reserved bits

"0b00" is read from these bits.
Set these bits to "0b00" when writing.

[bit5] FCSE: Anomalous frequency detection interrupt enable bit

bit	Description
0	Disables FCS interrupts
1	Enables FCS interrupts

[bit4:3] Reserved: Reserved bits

"0b00" is read from these bits.
Set these bits to "0b00" when writing.

[bit2] PCSE: PLL oscillation stabilization wait completion interrupt enable bit

bit	Description
0	Disables PLL oscillation stabilization wait completion interrupts
1	Enables PLL oscillation stabilization wait completion interrupts

[bit1] SCSE: Sub clock oscillation stabilization wait completion interrupt enable bit

bit	Description
0	Disables sub clock oscillation stabilization wait completion interrupts
1	Enables sub clock oscillation stabilization wait completion interrupts

[bit0] MCSE: Main clock oscillation stabilization wait completion interrupt enable bit

bit	Description
0	Disables main clock oscillation stabilization wait completion interrupts
1	Enables main clock oscillation stabilization wait completion interrupts

<Note>

For "Anomalous frequency detection", see Chapter "Clock supervisor".

5.15. Interrupt Status Register (INT_STR)

The INT_STR indicates the status of interrupts.

■ Register configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved		FCSI	Reserved		PCSI	SCSI	MCSI
Attribute	-		R	-		R	R	R
Initial value	-		0	-		0	0	0

■ Register functions

[bit7:6] Reserved: Reserved bits

"0b00" is read from these bits.
Set these bits to "0b00" when writing.

[bit5] FCSI: Anomalous frequency detection interrupt status bit

bit	Description
0	No FCS interrupt has been asserted.
1	An FCS interrupt has been asserted.

[bit4:3] Reserved: Reserved bits

"0b00" is read from these bits.
Set these bits to "0b00" when writing.

[bit2] PCSI: PLL oscillation stabilization wait completion interrupt status bit

bit	Description
0	No PLL oscillation stabilization wait completion interrupt has been asserted.
1	A PLL oscillation stabilization wait completion interrupt has been asserted.

[bit1] SCSI: Sub clock oscillation stabilization wait completion interrupt status bit

bit	Description
0	No sub clock oscillation stabilization wait completion interrupt has been asserted.
1	A sub clock oscillation stabilization wait completion interrupt has been asserted.

[bit0] MCSI: Main clock oscillation stabilization wait completion interrupt status bit

bit	Description
0	No main clock oscillation stabilization wait completion interrupt has been asserted.
1	A main clock oscillation stabilization wait completion interrupt has been asserted.

5.16. Interrupt Clear Register (INT_CLR)

The INT_CLR clears interrupt factors.

■ Register configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved		FCSC	Reserved		PCSC	SCSC	MCSC
Attribute	-		W	-		W	W	W
Initial value	-		0	-		0	0	0

■ Register functions

[bit7:6] Reserved: Reserved bits

"0b00" is read from these bits.
Set these bits to "0b00" when writing.

[bit5] FCSC: Anomalous frequency detection interrupt factor clear bit

bit	Description
When 0 is written	The FCS interrupt factor is not affected by the written value.
When 1 is written	Clears the FCS interrupt factor.
When read	The fixed value "0" is read.

[bit4:3] Reserved: Reserved bits

"0b00" is read from these bits.
Set these bits to "0b00" when writing.

[bit2] PCSC: PLL oscillation stabilization wait completion interrupt factor clear bit

bit	Description
When 0 is written	The PLL oscillation stabilization wait completion interrupt factor is not affected by the written value.
When 1 is written	Clears the PLL oscillation stabilization wait completion interrupt factor.
When read	The fixed value "0" is read.

[bit1] SCSC: Sub clock oscillation stabilization wait completion interrupt factor clear bit

bit	Description
When 0 is written	The sub clock oscillation stabilization wait completion interrupt factor is not affected by the written value.
When 1 is written	Clears the sub clock oscillation stabilization wait completion interrupt factor.
When read	The fixed value "0" is read.

[bit0] MCSC: Main clock oscillation stabilization wait completion interrupt factor clear bit

bit	Description
When 0 is written	The main clock oscillation stabilization wait completion interrupt factor is not affected by the written value.
When 1 is written	Clears the main clock oscillation stabilization wait completion interrupt factor.
When read	The fixed value "0" is read.

<Note>

When this register is cleared, each interrupt status bit (FCSI, PCSI, SCSI, MCSI) of the INT_STR register is also cleared.

6. Clock Generation Unit Usage Precautions

This section explains the precautions for using the clock generation unit.

- The oscillation stabilization wait time of main clock and sub clock oscillators

  Because the stabilization wait time of main clock/sub clock oscillator depends on the oscillator type (crystal, ceramic, etc.), the oscillation stabilization wait time suitable for the oscillator type must be selected.

- Changing the frequency division under stabilized PLL oscillation

  When the PLL frequency division ratio is changed after stabilization of PLL oscillation, stop the PLL oscillation once, change the frequency division ratio, and then re-enable the PLL oscillation.

- Peripherals independent of clock control by the clock generation unit

  The following peripherals run independently of clock control by the clock generation unit.
  For information about how to handle each operating clock, see the following chapter.

· USB operating clock generation unit	:	See Chapter "USB Clock Generation" in "Communication Macro Part".
· Clock supervisor	:	See Chapter "Clock supervisor".
· Watchdog timer	:	See Chapter "Watchdog Timer" in "Timer Part".
· Watch counter	:	See Chapter "Watch Counter" in "Timer Part".
· Real-time clock	:	See Chapter "REAL-TIME CLOCK" in "Timer Part".
· CAN prescaler	:	See Chapter "CAN Prescaler" in "Communication Macro Part".

- Setting the oscillation stabilization wait time

  Set the oscillation stabilization wait time of the main clock, sub clock, and PLL oscillators with relevant oscillation stabilization wait time setup registers, and then enable each oscillator.
  Do not change the oscillation stabilization wait time while waiting for oscillation to stabilize.

- Checking main clock oscillation while using the main PLL clock

  It is prohibited to stop main clock oscillation while using PLL oscillation.

- Switching clock modes

  Clock modes can be switched by changing the RCS[2:0] bits of the SCM_CTL register.
  To switch clock modes, take the following steps:

  1. Set the oscillation stabilization wait time of each oscillator.
  2. Set the oscillation enable bit of the desired clock (SCM_CTL:xxxE) to "1".
  3. Check the oscillation stable bit of the desired clock (SCM_CTL:xxxRDY) to "1".
  4. Switch SCM_CTL:RCS[2:0].
  5. Wait until SCM_STR:RCM[2:0] = SCM_CTL:RCS[2:0].

- Correlation between the clock mode switching and the oscillation stable bit

  The timings when the oscillation stable bit (SCM_STR:xxxRDY) turns to "1" vary for the following clock mode switching.

  - When switching from the high-speed CR run, main run, or PLL run to another clock mode:

    Setting SCM_CTL:xxxE to "1" can start the oscillation stabilization wait time. You can check that SCM_STR:xxxRDY is "1" after the oscillation stabilization wait time has elapsed.

  - When switching from the low-speed CR run or sub run to the high-speed CR run, main run, or PLL run:

    Setting SCM_CTL:MOSCE (or PLLE) to 1 does not start the oscillation stabilization wait time. To start the main clock (or high-speed CR or PLL) oscillation stabilization wait time, SCM_CTL:RCS [2:0] must be switched. After the oscillation stabilization wait time has elapsed, you can check that SCM_STR:xxxRDY is "1".

- If the standby mode is released by an interrupt, the device restarts in the clock mode that indicated by the RCS[2:0] bits in the SCM_CTL register.

- If any reset occurs other than software resets, the high-speed CR clock (CLKHC) is set as a master clock. In addition, high-speed CR clock mode is set as clock mode.

- If any reset other than software resets is executed, the main clock and sub clock oscillators, and PLL oscillation stop. If you want to use those oscillators again after the reset, enable them using the SCM_CTL register.

- For the correlation between each clock mode and start/stop of the oscillator, see Chapter "Low Power Consumption Mode".

- For clock stop functions for each peripheral function, see “Peripheral Clock Gating”.

- To turn off the power supply on the chip side and operate only VBAT domain, be sure to set “WTOSCCNT.SOSCNTL=”0” “ and then turn off the power supply on the chip side.

CHAPTER : Peripheral Clock Gating

This chapter explains the functions of Peripheral Clock Gating.

CODE: 9BFPCG_FM4-E1.0

1. Peripheral Clock Gating Overview

This section shows an overview of the Peripheral Clock Gating which stops the operation clocks of peripheral functions individually. By using these functions, the system can reduce the current consumption of the total system with gating the operation clocks of peripheral functions not used.

■ Overview of Peripheral Clock Gating

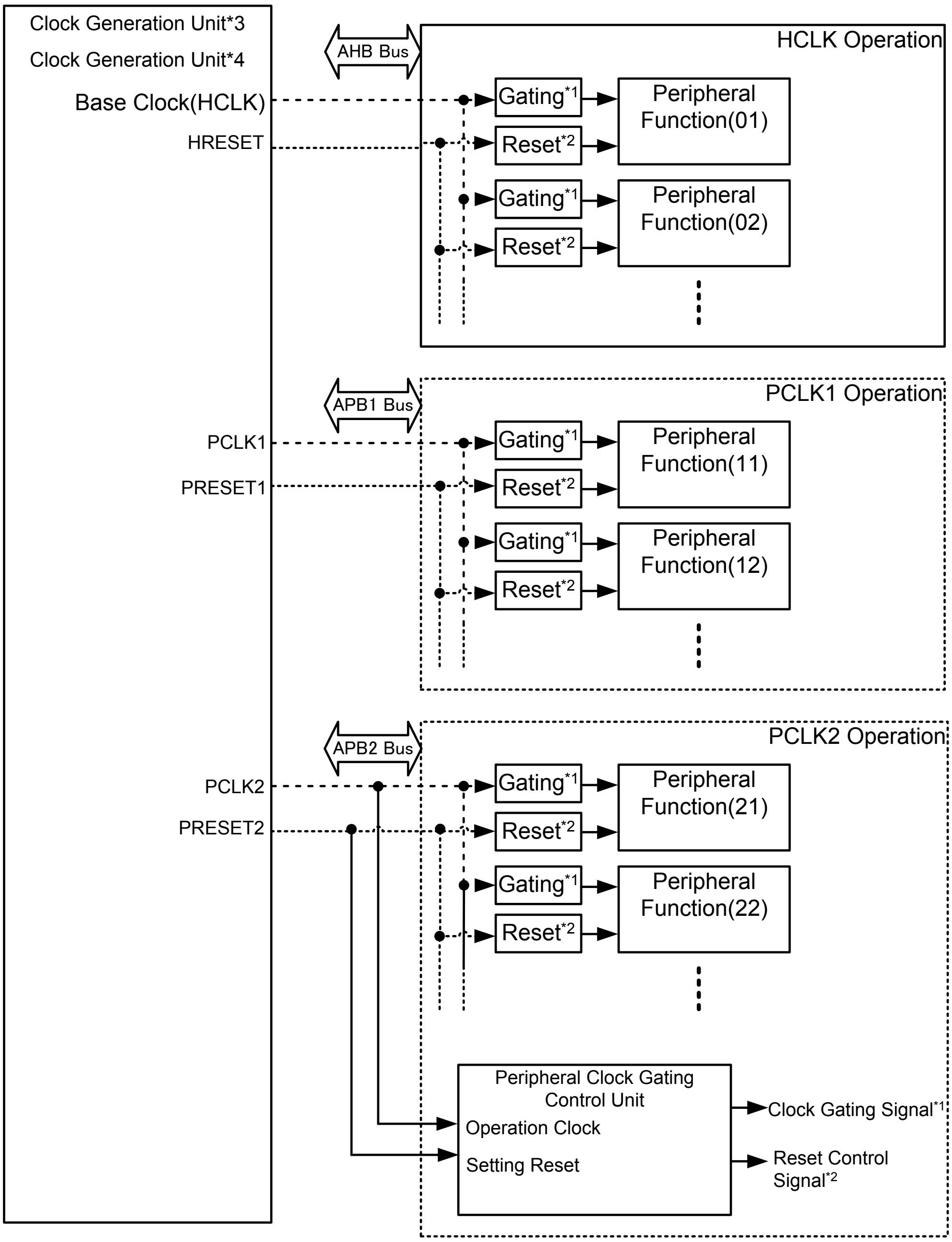
· The operation clocks of peripheral functions not used in the system operation are gated individually.
· For target clocks and units of the Peripheral Clock Gating, see "■ Gating units and their initial states of Peripheral Clock Gating".
· When a clock is gated or before a clock is supplied, the internal states of peripheral functions can be reset.

The above peripheral clock gating and reset control are implemented by the setting of a register connecting to APB2 bus.

■ Overview of Connection with Clock and Reset Generation Units

Figure 1-1 shows the connection between peripheral clock gating and clock generation unit or reset generation unit. The peripheral clock gating exists between peripheral function and clock generation unit or reset generation unit and gates clocks and controls resets in the unit of peripheral function. When the internal bus clock supply from the reset control units are stopped, the priority is given to the settings of the clock control unit and the operation clock supplies to peripheral functions are gated To use the peripheral clock gating, be sure to make the settings which enable the output of APB2 bus clock (PCLK2) in the clock generation unit to control the rest.

Figure 1-1 Clock/Reset Connection related to Peripheral Clock Gating



*1 : Clock gating (internal bus clock gating) in the unit of peripheral function

*2: Forcible reset or bus reset (see Chapter "Reset") by peripheral function unit

*3: For details, see Chapter "Clock".

*4: For details, see Chapter "Reset".

■ Gating units and their initial states of Peripheral Clock

For gating units and their initial states of Peripheral Clock Gating, see Table 1-1.

Table 1-1 Control units and their initial states of Peripheral Clock Gating

Peripheral Functions	Clock Stop Units	Initial States	Remarks
Multi-function Serial Interface	Unit of channel	Clock supply	
Base Timer	Unit of four channels	Clock supply	The clock gating can be controlled with every four channels "ch.0 to ch.3", "ch.4 to ch.7", "ch.8 to ch.11", and "ch.12 to ch.15".
Multi-function Timer	Unit of unit	Clock supply	
PPG	Unit of four channels	Clock supply	The clock gating can be controlled with every four channels "chs.0, 2, 4, and 6", "chs.8, 10, 12, and 14", "chs.16, 18, 20, and 22", and "chs.24, 26, 28, and 30".
Quad Counter	Unit of unit	Clock supply	
DMAC	Unit of unit	Clock supply	
External Bus Interface	Unit of unit	Clock supply	
CAN Controller	Unit of channel	Clock supply	
USB (Function/Host)	Unit of channel	Clock stop	
SD Card Interface	Unit of unit	Clock stop	
A/D Converter	Unit of unit	Clock supply	
I/O Port	Batch of all ports	Clock supply	For constrains at clock gating,"5. Peripheral Clock Gating Function Usage Precautions".

<Notes>

- For types and the number of mounted peripheral functions, see "Data sheet" of the product used
- The clock control of PPG shares the setting bits with the multi-function timer. For details, see "4.3 Peripheral Clock Control Register 1 (CKEN1)".
- For products equipped with Ethernet-MAC, execute the clock control by each unit with Ethernet System Control Register (ETH_CLKG).
- Execute the clock control of the DSTC unit alone with "5. DSTC Register" in"DSTC."

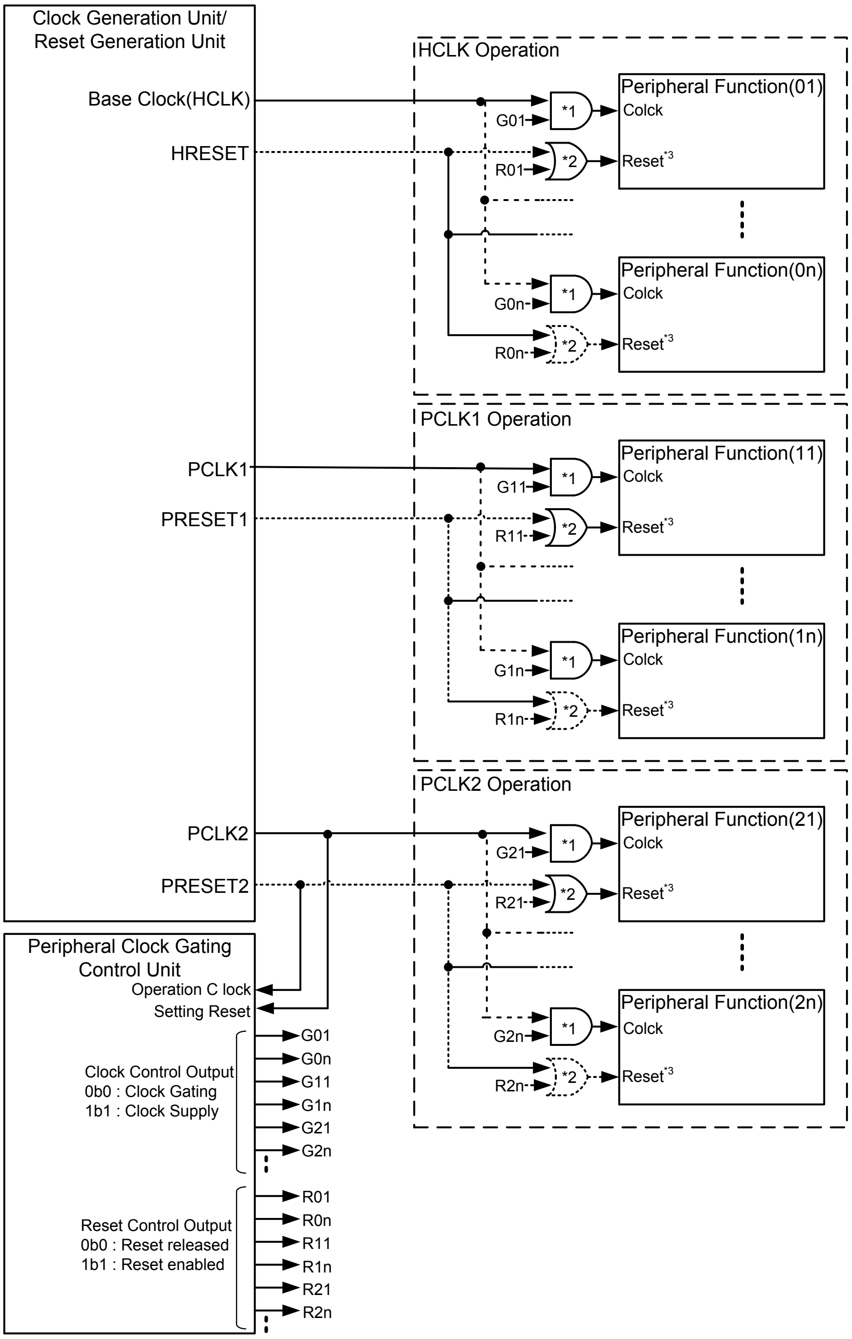
2. Peripheral Clock Gating Configuration

This section explains the configuration of the Peripheral Clock Gating.

■ Block diagram

Figure 2-1 shows the system configuration of Peripheral Clock Gating.

Figure 2-1 Block Diagram of Peripheral Clock Gating



*1 : Gates the operation clocks of peripheral functions

*2 : Resets peripheral functions

*3 : At High-level signal input: Reset enabled, At Low-level signal input: Reset released

■ Explanation on Block Diagram

● Peripheral Clock Gating Control Unit

The clock control or the reset control of each peripheral function is executed by changing the register setting value via the APB2 bus. Be sure to rewrite this register with setting APB2 clock enable bit (APBC2_PSR. APBC2EN) inAPB2 prescaler register (APBC2_PSR) of the clock control unit to the output enable and permitting PCLK2 output.

The clock of each peripheral function stops when the bit field of the target function is set to "0". When the bit field is set to "1", the clock is supplied. The initial value of a register is different by peripheral function. For details, see Table 1-1.

The reset of each peripheral function is issued when the bit field of the target function is set to"1". When the bit field is set to "0", the reset is released. The initial value of each register is always "0" to release the reset.

● Peripheral Clock Gating Logic

Internal bus clock (HCLK, PCLK1, PCLK2) is supplied or gated by each specific peripheral function according to clock gating signal from the peripheral clock gating control unit.

● Peripheral Reset Control Logic

The reset is individually controlled by each peripheral function according to the reset control signal from the peripheral clock gating control unit. The reset control unit is the same with the peripheral clock control unit. However, it does not exceptionally have the bit field of this reset control for I/O Port alone.

3. Peripheral Clock Gating Control

This section explains the control of the peripheral clock gating.

The register of the peripheral clock gating becomes an initial state by bus reset (PRESET2)*. Be sure to execute the clock control for necessary peripheral functions immediately after reset of the bus because the bus reset (PRESET2) is generated by all reset factors.

*: For the generating condition of bus reset (PRESET2), see Chapter "Reset".

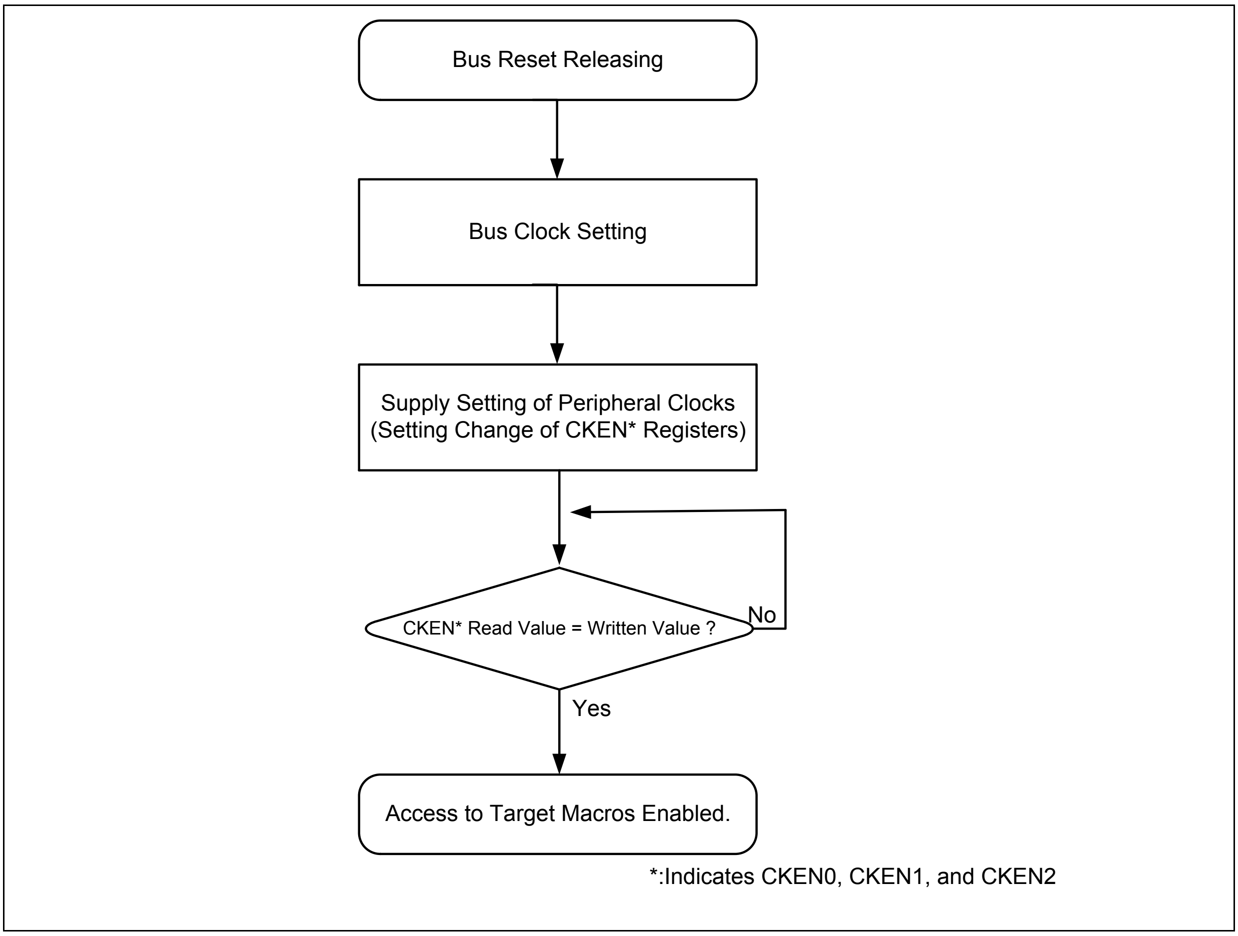
3.1. Peripheral Clock Control Procedures

This section explains the control procedures of supplying and stopping peripheral clocks.

■ Clock supply procedures

The settings of the bus clocks and the peripheral clocks are reset to the initial values immediately after the bus reset release. So, for the clocks of peripheral functions which have been stopped in the initial state, set the clock supplies conforming to the procedures in Figure 3-1.

Figure 3-1 Clock Supply Procedures


Bus Reset Releasing
Bus Clock Setting
Supply Setting of Peripheral Clocks
(Setting Change of CKEN* Registers)
CKEN* Read Value = Written Value ?
No
Yes
Access to Target Macros Enabled.
*:Indicates CKEN0, CKEN1, and CKEN2

1. Bus clock setting
   Execute the setting of each bus clock by using the register of the clock generation part.
   For the setting details, see Chapter "Clock".

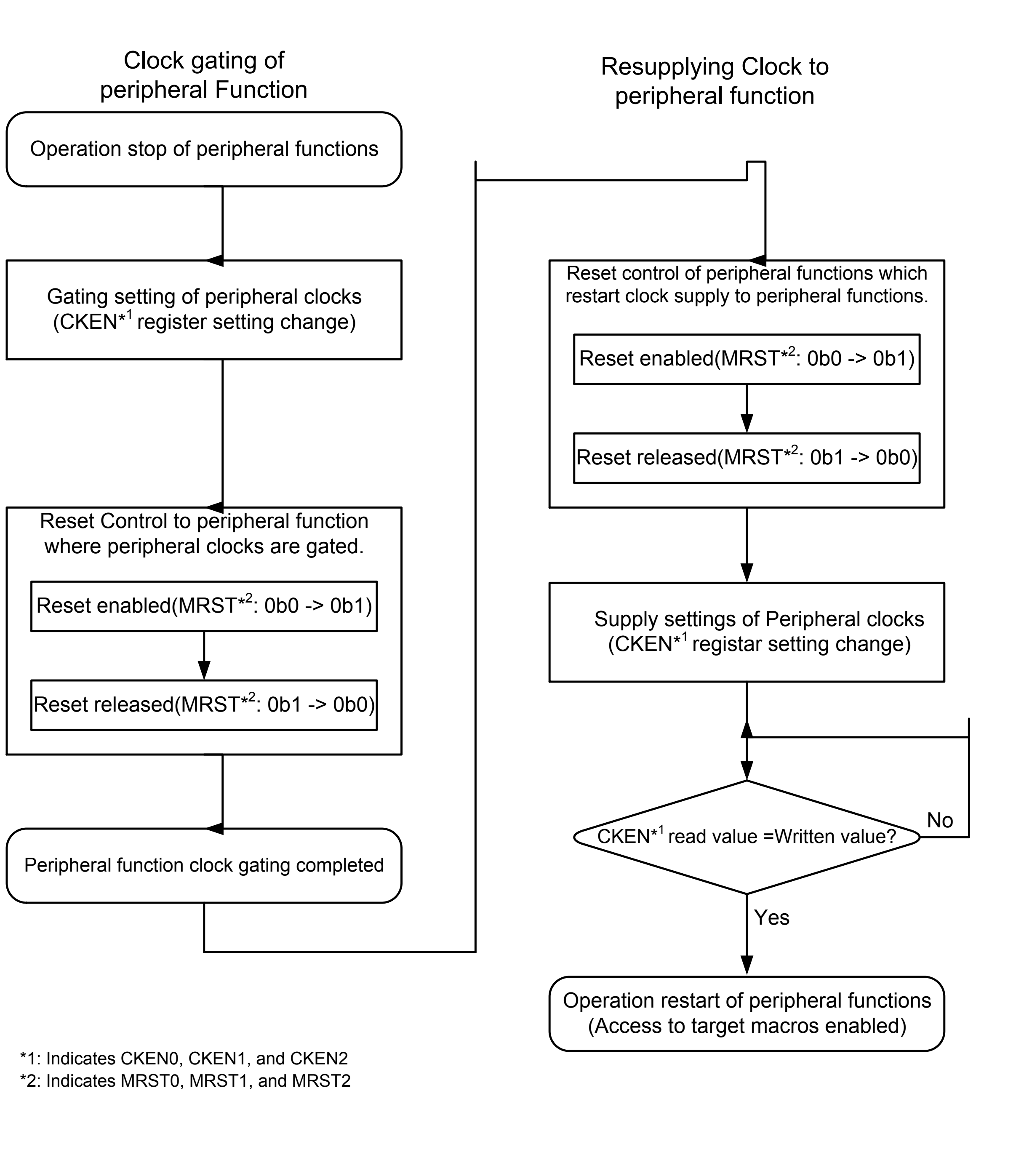
2. Supply setting of peripheral clocks
   Change the setting of the bit corresponding to the peripheral function to which the clock is to be supplied for peripheral clock control registers (CKEN0, CKEN1, and CKEN2) of the clock control in the clock gating state of the initial state.

3. Set value confirmation of peripheral clock control register
   The peripheral clock registers (CKEN0, CKEN1, and CKEN2) updates the register value to the written value at the step of starting the clock supply to the peripheral function to which the setting is changed. Be sure to start the access to the peripheral function after setting a change in the above-mentioned Item 2, reading this register, and then confirming the agreement with the written value because an access to peripheral function is invalid at clock gating.

■ Procedures of gating and resupplying clocks

Figure 3-2 explains the procedures of gating the clocks of peripheral functions and resupplying clocks to peripheral functions.

Figure 3-2 Procedures of Gating Clocks of Peripheral Functions and Resupplying Clocks to Peripheral Functions


Clock gating of peripheral Function
Operation stop of peripheral functions
Gating setting of peripheral clocks (CKEN*1 register setting change)
Reset Control to peripheral function where peripheral clocks are gated.
Reset enabled(MRST*2: 0b0 -> 0b1)
Reset released(MRST*2: 0b1 -> 0b0)
Peripheral function clock gating completed
Resupplying Clock to peripheral function
Reset control of peripheral functions which restart clock supply to peripheral functions.
Reset enabled(MRST*2: 0b0 -> 0b1)
Reset released(MRST*2: 0b1 -> 0b0)
Supply settings of Peripheral clocks (CKEN*1 registar setting change)
CKEN*1 read value =Written value?
No
Yes
Operation restart of peripheral functions (Access to target macros enabled)

*1: Indicates CKEN0, CKEN1, and CKEN2
*2: Indicates MRST0, MRST1, and MRST2

● Clock gating of peripheral functions

1. Gating setting of peripheral clocks
   For the peripheral clock control registers (CKEN0, CKEN1, and CKEN2), change the bit corresponding to the peripheral function for which the clock supply is to be stopped to "0".
   After gating the clock to the peripheral function to which the clock gating is instructed, the peripheral clock control registers (CKEN0, CKEN1, and CKEN2) updates the register value to the written value.

2. Reset control to peripheral functions whose peripheral clocks are gated
   For the peripheral functions whose clocks are gated, to reset their internal state, execute the reset control of each peripheral function according to the following procedures.

   Reset enabled :
   Write "1" to the target bits of peripheral function reset control registers (MRST0, MRST1, and MRST2).

   Reset released:
   Write "0" to the target bits of peripheral function reset control registers (MRST0, MRST1, and MRST2).

● Resupplying clocks to peripheral functions

1. Reset control of peripheral functions which restart clock supply to peripheral functions
   For the peripheral functions which gate the peripheral clocks, execute the reset control to each peripheral function by using peripheral function reset control registers (MRST0, MRST1, and MRST2) before restarting their operation. The procedures are the same as the above-mentioned procedures of reset control immediately after peripheral clocks gated.

2. Supply settings of peripheral clocks
   For the peripheral clock control registers (CKEN0, CKEN1, and CKEN2), change the settings of bit corresponding to the peripheral function for which the clock is to be resupplied.
   At this time, do not set the bit where the peripheral function is not provided and the bit whose bus clock has been gated to the values other than the initial value The reason is that the read value cannot coincide with the written value not to get out of the processing loop at the register set value confirmation in the following Item 3.

3. Confirmation of set values of peripheral clock control registers
   At the step where the clock setting change is reflected to the peripheral function whose settings are changed, the peripheral clock control registers (CKEN0, CKEN1, and CKEN2) updates the register value to the written values.
   Be sure to start the access to the peripheral function after executing the setting change of the above-mentioned Item 2 , reading the register, and then confirming the agreement with the written value because the access to the peripheral functions is invalid at clock gating.

4. Peripheral Clock Gating Function Registers

This section explains each register function of the peripheral clock gating functions.

Table 4-1 shows the list of registers of peripheral clock gating functions.

Table 4-1 Registers of the Peripheral Clock Gating Functions

Abbreviated register name	Register name	Reference
CKEN0	Peripheral Function Clock Control Register 0	4.1
MRST0	Peripheral Function Reset Control Register 0	4.2
CKEN1	Peripheral Function Clock Control Register 1	4.3
MRST1	Peripheral Function Reset Control Register 1	4.4
CKEN2	Peripheral Function Clock Control Register 2	4.5
MRST2	Peripheral Function Reset Control Register 2	4.6

4.1. Peripheral Function Clock Control Register 0 (CKEN0)

This section explains Peripheral Function Reset Clock Register 0 (CKEN0).

bit	31	30	29	28	27	26	25	24
Field	Reserved			GIOCK	Reserved	EXBCK	Reserved	DMACK
Attribute	-			R/W	-	R/W	-	R/W
Initial value	-			1	-	1	-	1

bit	23	22	21	20	19	18	17	16
Field	Reserved				ADCCK[3:0]			
Attribute	-				R/W			
Initial value	-				1111			

bit	15	14	13	12	11	10	9	8
Field	MFSCK[15:8]							
Attribute	R/W							
Initial value	0xFF							

bit	7	6	5	4	3	2	1	0
Field	MFSCK[7:0]							
Attribute	R/W							
Initial value	0xFF							

[bit31:29] Reserved: Reserved bits

Write "0" to these bits.

[bit28] GIOCK: Settings for operation clock supplying and gating to GPIO function

This bit controls the operation clock supplying and the gating to the I/O Port function. This bit controls all the operation clocks to the I/O Port functions collectively.
When this bit is set to "1", the bus clock is supplied to the I/O Port function block and the I/O Port function can be used.
When this bit is set to "0", the bus clock input to the I/O Port function block is gated. Note that the reading of the input level and the setting change of the output power level are disabled while the bus clock is gated.
For details, see "5. Peripheral Clock Gating Function Usage Precautions ".

bit	Description
0	The bus clock input to the I/O Port function block is gated.
1	The bus clock is supplied to the I/O Port function block. (Initial value) Be sure to set "1" in order to use I/O Port function.

[bit27] Reserved: Reserved bit

Write "0" to this bit.

[bit26] EXBCK: Settings for operation clock supplying and gating of external bus interface function

This bit controls the operation clock supplying and the gating to the external bus interface functions. When this bit is set to "1", the bus clock is supplied to the external bus interface function block, and the external bus interface function can be used. For products to which the external bus interface is not mounted, do not change this bit from the initial value.
When this bit is set to "0", the bus clock input to the external bus interface function is gated. While the bus clock input is gated, the external bus interface cannot be used.

bit	Description
0	The bus clock input to the external bus interface function block is gated.
1	The bus clock is supplied to the external bus interface function block. (Initial value)

[bit25] Reserved: Reserved bit

Write "0" to this bit.

[bit24] DMACK: Supplying and gating settings of DMAC operation clock

This bit controls the operation clock supplying and the gating to the DMAC function. When this bit is set to"1", the bus clock is supplied to the DMAC block and the DMAC function can be used.
When this bit is set to "0", the bus clock input to the DMAC block is gated. While the bus clock input is gated, the DMAC function cannot be used.

bit	Description
0	The bus clock input to DMAC is gated.
1	The bus clock is supplied to DMAC. (Initial value)

[bit23:20] Reserved: Reserved bits

Write "0" to these bits.

[bit19:16] ADCCK[3:0]: Settings for operation clock supplying and gating to A/D converter

These bits control the operation clock supplying and gating to the A/D converter. The following show the correspondence between each bit and the A/D converter unit:

bit16 - ADCCK0 : A/D converter unit 0
bit17 - ADCCK1 : A/D converter unit 1
bit18 - ADCCK2 : A/D converter unit 2
bit19 - ADCCK3 : A/D converter unit 3

When the relevant bit is set to "1", the bus clock is supplied to the unit of the corresponding A/D converter to enable the A/D converter function. For products to which the corresponding A/D converter is not mounted, do not change the relevant bit from the initial value.
When the relevant bit is set to "0", the bus clock input to the corresponding A/D converter is gated. While the bus clock input is gated, the relevant A/D converter cannot be used.

bit	Description
0	The bus clock input to the A/D converter unit corresponding to the relevant bit is gated.
1	The bus clock is supplied to the A/D converter unit corresponding to the relevant bit. (Initial value)

[bit15:0] MFSCK[15:0]: Settings for operation clock supply and gating to multi-function serial interface

These bits control the operation clock supply and gating to the multi-function serial interface. The correspondence between each bit and the channel is shown below:

bit0 - MFSCK0 : Multi-function serial interface channel 0
bit1 - MFSCK1 : Multi-function serial interface channel 1
bit2 - MFSCK2 : Multi-function serial interface channel 2
bit3 - MFSCK3 : Multi-function serial interface channel 3
bit4 - MFSCK4 : Multi-function serial interface channel 4
bit5 - MFSCK5 : Multi-function serial interface channel 5
bit6 - MFSCK6 : Multi-function serial interface channel 6
bit7 - MFSCK7 : Multi-function serial interface channel 7
bit8 - MFSCK8 : Multi-function serial interface channel 8
bit9 - MFSCK9 : Multi-function serial interface channel 9
bit10 - MFSCK10 : Multi-function serial interface channel 10
bit11 - MFSCK11 : Multi-function serial interface channel 11
bit12 - MFSCK12 : Multi-function serial interface channel 12
bit13 - MFSCK13 : Multi-function serial interface channel 13
bit14 - MFSCK14 : Multi-function serial interface channel 14
bit15 - MFSCK15 : Multi-function serial interface channel 15

When the relevant bit is set to "1", the bus clock is supplied to the channel of the corresponding multi-function serial interface to enable the function of the multi-function serial interface. For products to which the relevant multi-function serial interface channel is not mounted, do not change the relevant bit from the initial value.

When the relevant bit is set to "0", the bus clock input to the channel of the corresponding multi-function serial interface is gated. While the bus clock input is gated, the multi-function serial interface function of the corresponding channel cannot be used.

bit	Description
0	The bus clock input to the multi-function serial interface channel corresponding to the relevant bit is gated.
1	The bus clock is supplied to the multi-function serial interface channel corresponding to the relevant bit. (Initial value)

4.2. Peripheral Reset Control Register 0 (MRST0)

This section explains the peripheral reset control register 0 (MRST0).

bit	31	30	29	28	27	26	25	24
Field	Reserved					EXBRST	Reserved	DMARST
Attribute	-					R/W	-	R/W
Initial value	-					0	-	0

bit	23	22	21	20	19	18	17	16
Field	Reserved				ADCRST[3:0]			
Attribute	-				R/W			
Initial value	-				0000			

bit	15	14	13	12	11	10	9	8
Field	MFSRST[15:8]							
Attribute	R/W							
Initial value	0x00							

bit	7	6	5	4	3	2	1	0
Field	MFSRST[7:0]							
Attribute	R/W							
Initial value	0x00							

[bit31:27] Reserved: Reserved bits

Write "0" to these bits.

[bit26] EXBRST: Reset control for external bus interface

This bit controls the reset of the external bus interface only. When this bit is set to "1", the external bus interface is reset to disable the operation of the external bus interface. For products to which the external bus interface is not mounted, do not change this bit from the initial value. To release the reset status, be sure to set this bit to "0" again.

bit	Description
0	Releases the reset of the external bus interface. (Initial value)
1	Executes the external bus interface reset.

<Note>

The external bus interface reset control with the above register control cannot initialize the registers of the external bus interface.

[bit25] Reserved: Reserved bit

Write "0" to this bit.

[bit24] DMARST: Reset control of DMAC

This bit controls reset of the DMAC unit. If this bit is set to "1", DMAC becomes a reset state, the DMA transfer operation stops, and all the register settings are initialized. To release the reset state, be sure to set this bit to "0" again.

bit	Description
0	Releases the DMAC reset. (Initial value)
1	Issues reset signal to DMAC.

[bit23:20] Reserved: Reserved bits

Write”0” to these bits.

[bit19:16] ADCRST[3:0]: Reset control of A/D converter

These bits control the reset of each unit of the A/D converter. The correspondence between each bit and A/D converter unit is shown below:

bit16 - ADCRST0 : A/D converter unit 0
bit17 - ADCRST1 : A/D converter unit 1
bit18 - ADCRST2 : A/D converter unit 2
bit19 - ADCRST3 : A/D converter unit 3

If the relevant bit is set to "1", the corresponding A/D converter unit becomes a reset state, the analog to digital conversion operation stops, and the register settings are initialized. For products to which the relevant A/D converter unit is not mounted, do not change the relevant bit from the initial state. To release the reset state, be sure to set this bit to "0" again.

bit	Description
0	Releases the reset of the A/D converter unit corresponding to the relevant bit. (Initial value)
1	Issues the reset to the A/D converter unit corresponding to the relevant bit.

[bit15:0] MFSRST[15:0]: Control of software reset of multi-function serial interface

These bits control the reset of each channel of the multi-function serial interface. The correspondence between each bit and the channel is shown below.

bit0 - MFSRST0: Multi-function serial interface channel 0
bit1 - MFSRST1: Multi-function serial interface channel 1
bit2 - MFSRST2 : Multi-function serial interface channel 2
bit3 - MFSRST3 : Multi-function serial interface channel 3
bit4 - MFSRST4 : Multi-function serial interface channel 4
bit5 - MFSRST5 : Multi-function serial interface channel 5
bit6 - MFSRST6 : Multi-function serial interface channel 6
bit7 - MFSRST7 : Multi-function serial interface channel 7
bit8 - MFSRST8 : Multi-function serial interface channel 8
bit9 - MFSRST9 : Multi-function serial interface channel 9
bit10 - MFSRST10 : Multi-function serial interface channel 10
bit11 - MFSRST11 : Multi-function serial interface channel 11
bit12 - MFSRST12 : Multi-function serial interface channel 12
bit13 - MFSRST13 : Multi-function serial interface channel 13
bit14 - MFSRST14 : Multi-function serial interface channel 14
bit15 - MFSRST15 : Multi-function serial interface channel 15

If the relevant bit is set to "1", the channel of the corresponding multi-function serial interface becomes a reset state, its serial communications stop, and the register setting is initialized. For products to which the relevant multi-function serial interface channel is not mounted, it is prohibited to set the relevant bit to "1". To release the above-mentioned reset, be sure to set this bit to "0" again.

bit	Description
0	Releases the reset of the multi-function serial interface channel corresponding to the relevant bit. (Initial value)
1	Issues the reset the multi-function serial interface channel corresponding to the relevant bit.

4.3. Peripheral Clock Control Register 1 (CKEN1)

This section explains the peripheral clock control register 1 (CKEN1).

bit	31	30	29	28	27	26	25	24
Field	Reserved							
Attribute	-							
Initial value	-							

bit	23	22	21	20	19	18	17	16
Field	Reserved				QDUCK[3:0]			
Attribute	-				R/W			
Initial value	-				1111			

bit	15	14	13	12	11	10	9	8
Field	Reserved				MFTCK[3:0]			
Attribute	-				R/W			
Initial value	-				1111			

bit	7	6	5	4	3	2	1	0
Field	Reserved				BTMCK[3:0]			
Attribute	-				R/W			
Initial value	-				1111			

[bit31:20] Reserved: Reserved bits

Write "0" to these bits.

[bit19:16] QDUCK[3:0]: Settings for operation clock supply and gating of quad counter

These bits control the operation clock supply and gating of quad counter. The correspondence between each bit and quad counter is shown below.

bit16 - QDUCK0: Quad counter unit 0
bit17 - QDUCK1: Quad counter unit 1
bit18 - QDUCK2: Quad counter unit 2
bit19 - QDUCK3: Quad counter unit 3

When the relevant bit is set to "1", the bus clock is supplied to the unit of the corresponding quad counter to use the quad counter function. For products to which the relevant quad counter unit is not mounted, do not change the relevant bit from the initial value.
When the relevant bit is set to "0", the bus clock input to corresponding quad counter is stopped. While the bus clock input is gated, the quad counter of the relevant unit cannot be used.

bit	Description
0	Gates the bus clock input to the corresponding quad counter.
1	Supplies the bus clock to the quad counter corresponding to the relevant bit. (Initial value)

[bit15:12] Reserved: Reserved bits

Write "0" to these bits.

[bit11:8] MFTCK[3:0]: Settings for operation clock supply and gating of multi-function timer and PPG

These bits control the operation clock supply and gating to the multi-function timer and PPG. The correspondence among each bit, the multi-function timer unit, and the PPG channel is shown below.

bit8 - MFTCK0 : Multi-function timer unit 0 - PPG channels 0, 2, 4, 6
bit9 - MFTCK1 : Multi-function timer unit 1 - PPG channels 8, 10, 12, 14
bit10 - MFTCK2 : Multi-function timer unit 2 - PPG channels 16, 18, 20, 22
bit11 - MFTCK3 : Multi-function timer unit 3 - PPG channels 24, 26, 28, 30

When the relevant bit is set to "1", the bus clock is supplied to corresponding multi-function timer unit and PPG channel to use the multi-function timer and PPG function. For products to which the relevant multi-function timer unit and PPG channels is not mounted, it is prohibited to change the relevant bit from the initial value.
When the relevant bit is set to "0", the bus clock input to the corresponding multi-function timer unit and PPG channels is gated. While the bus clock is gated, the relevant multi-function timer and PPPG function cannot be used.

bit	Description
0	The bus clock input to the multi-function timer unit and the PPG channel corresponding to the relevant bit is gated.
1	The bus clock is supplied to the multi-function timer unit and the PPG channel corresponding to the relevant bit. (Initial value)

[bit7:4] Reserved: Reserved bits

Write "0" to these bits.

[bit3:0] BTMCK[3:0]: Settings operation clock supply and gating to base timer

These bits control the operation clock supply and gating to the base timer. The correspondence between each bit and the base timer channel is shown below.

bit0 - BTMCK0 : Base timer channel 0, 1, 2, 3
bit1 - BTMCK1 : Base timer channel 4, 5, 6, 7
bit2 - BTMCK2 : Base timer channel 8, 9, 10, 11
bit3 - BTMCK3 : Base timer channel 12, 13, 14, 15

When the relevant bit is set to "1", the bus clock is supplied to the corresponding base timer channel to use the base timer not change the relevant bit from the initial value.
When "0" is set to the relevant bit, the bus clock input to the corresponding base timer channel is gated. While the bus clock input is gated, the base timer function of the corresponding channel cannot be used.

bit	Description
0	The bus clock input to the base timer channel corresponding to the relevant bit is gated.
1	The bus clock is supplied to the base timer channel corresponding to the relevant bit. (Initial value)

4.4. Peripheral Function Reset Control Register 1 (MRST1)

This section explains the peripheral function reset control register 1(MRST1).

bit	31	30	29	28	27	26	25	24
Field	Reserved							
Attribute	-							
Initial value	-							

bit	23	22	21	20	19	18	17	16
Field	Reserved				QDURST[3:0]			
Attribute	-				R/W			
Initial value	-				0000			

bit	15	14	13	12	11	10	9	8
Field	Reserved				MFTRST[3:0]			
Attribute	-				R/W			
Initial value	-				0000			

bit	7	6	5	4	3	2	1	0
Field	Reserved				BTMRST[3:0]			
Attribute	-				R/W			
Initial value	-				0000			

[bit31:20] Reserved: Reserved bits

Write "0" to these bits.

[bit19:16] QDURST[3:0]: Reset control of quad counter

These bits control the reset of each unit of the quad counter. The correspondence between each bit and the quad counter unit is shown below.

bit16 - QDURST 0 : Quad counter unit 0
bit17 - QDURST 1 : Quad counter unit 1
bit18 - QDURST 2 : Quad counter unit 2
bit19 - QDURST 3 : Quad counter unit 3

If the relevant bit is set to "1", the unit of the corresponding quad counter becomes a reset state, the quad counter operation stops, and the register settings are initialized. For products to which the relevant quad counter is not mounted, do not change the relevant bit from the initial state. To release the reset state, be sure to set this bit to "0" again.

bit	Description
0	Releases the reset of the quad counter corresponding to the relevant bit. (Initial value)
1	Issues the reset to the quad counter unit corresponding to the relevant bit.

[bit15:12] Reserved: Reserved bits

Write "0" to these bits.

[bit11:8] MFTRST[3:0]: Control of multi-function timer and PPG reset control

These bits control multi-function timer reset of each unit and PPG reset of every four channels. The correspondence among each bit, quad counter unit, and the PPG channel is shown below.

bit8 - MFTRST0: Multi-function timer unit 0 - PPG channel 0, 2, 4, 6
bit9 - MFTRST1: Multi-function timer unit 1 - PPG channel 8, 10, 12, 14
bit10 - MFTRST2: Multi-function timer unit 2 - PPG channel 16, 18, 20, 22
bit11 - MFTRST3: Multi-function timer unit 3 - PPG channel 24, 26, 28, 30

If the relevant bit is set to "1", the corresponding multi-function timer unit and PPG channel become the reset states, the multi-function timer operation stops, and the register setting is initialized. For products to which the relevant multi-function timer unit and PPG channels are not mounted, do not change the relevant bit from the initial value. To release the reset state, be sure to set this bit to "0" again.

bit	Description
0	Release the resets of the multi-function timer unit and the PPG channel corresponding to the relevant bit. (Initial value)
1	Issue the resets to the multi-function timer unit and the PPG channels corresponding to the relevant bit.

[bit7:4] Reserved: Reserved bits

Write "0" to these bits.

[bit3:0] BTMRST[3:0]: Reset control of base timer

These bits control the reset for four units of the base timer. The correspondence among each bit and the base timer channels is shown below.

bit0 - BTMRST0: Base timer channels 0, 1, 2, 3
bit1 - BTMRST1: Base timer channels 4, 5, 6, 7
bit2 - BTMRST2: Base timer channels 8, 9, 10, 11
bit3 - BTMRST3: Base timer channels 12, 13, 14, 15

If the relevant bit is set to "1", the unit of the corresponding base timer channels becomes a reset state, the base timer operation stops, and the register setting is initialized. For products to which the relevant base timer channels are not mounted, do not change the relevant bit from the initial value. To release the reset state, be sure to set this bit to "0" again.

bit	Description
0	Release the reset for the base timer channel corresponding to the relevant bit. (Initial value)
1	Issue the reset to the base timer channel corresponding to the relevant bit.

4.5. Peripheral Clock Control Register 2 (CKEN2)

This section explains the peripheral clock control register 2(CKEN2).

bit	31	30	29	28	27	26	25	24
Field	Reserved							
Attribute	-							
Initial value	-							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
Attribute	-							
Initial value	-							

bit	15	14	13	12	11	10	9	8
Field	Reserved							SDCCK
Attribute	-							R/W
Initial value	-							0

bit	7	6	5	4	3	2	1	0
Field	Reserved		CANCK[1:0]		Reserved		USBCK[1:0]	
Attribute	-		R/W*		-		R/W	
Initial value	-		11*		-		00	

* : For products not mounting CAN controller, Attribute is "R" and Initial value is "00".

[bit31:9] Reserved: Reserved bits

Write "0" to these bits.

[bit8] SDCCK: Settings for operation clock supply and gating to SD card interface

This bit controls the operation clock supply and gating to the SD card interface function. When this bit is set to "1", the bus clock is supplied to the SD card interface unit to use the SD card interface function. For products to which the relevant SD card interface unit is not mounted, do not change the relevant bit from the initial value.

When this bit is set to "0", the bus clock input to the SD card interface unit is gated. While the bus clock input is gated, the functions of the SD card interface cannot be used..

bit	Description
0	Gates the bus clock input to SD card interface. (Initial value)
1	Supplies the bus clock to SD card interface.

[bit7:6] Reserved: Reserved bits

Write "0" to these bits.

[bit5:4] CANCK[1:0]: Settings for clock supply and gating to CAN controller

These bits control bus clock (base clock) supply and gating to the CAN controller. The correspondence between each bit and the CAN controller channel is shown below.

bit4 - CANCK0: CAN controller channel 0
bit5 - CANCK1: CAN controller channel 1

When the relevant bit is set to "1", the bus clock is supplied to the corresponding CAN controller channel to use the CAN controller function. For products to which the relevant CAN controller channel is not mounted, do not change the relevant bit from the initial value.
When the relevant bit is set to "0", the bus clock input to the corresponding CAN controller channel is gated. While the bus clock input is gated, the CAN controller function of the corresponding channel cannot be used.

bit	Description
0	Gates the bus clock input to the CAN controller channel corresponding to the relevant bit. (Initial value: For products not mounting CAN controller)
1	Supplies the bus clock input to the CAN controller channel corresponding to the relevant bit. (Initial value: For products mounting CAN controller)

[bit3:2] Reserved: Reserved bits

Write "0" to these bits.

[bit1:0] USBCK[1:0]: Settings for operation clock supply and gating of USB(function/host)

These bits control the operation clock supply and gating to USB (function/host). The correspondence between each bit and the USB channel is shown below.

bit0 - USBCK0: USB channel 0
bit1 - USBCK1: USB channel 1

When the relevant bit is set to "1", the bus clock is supplied to the corresponding USB channel to use the USB function. For products to which the relevant USB channel is not mounted, it is prohibited to change the relevant bit from the initial value.
When the relevant bit is set to "0", the bus clock input to the corresponding USB channel is gated. While the bus clock input is gated, the USB function of the corresponding channel cannot be used

bit	Description
0	Gates the bus clock input to the USB channel corresponding to the relevant bit. (Initial value)
1	Supplies the bus clock to the USB channel corresponding to the relevant bit.

4.6. Peripheral Function Reset Control Reset 2 (MRST2)

This section explains the peripheral function reset control register 2 (MRST2).

bit	31	30	29	28	27	26	25	24
Field	Reserved							
Attribute	-							
Initial value	-							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
Attribute	-							
Initial value	-							

bit	15	14	13	12	11	10	9	8
Field	Reserved							SDCRST
Attribute	-							R/W
Initial value	-							0

bit	7	6	5	4	3	2	1	0
Field	Reserved		CANRST[1:0]		Reserved		USBRST[1:0]	
Attribute	-		R/W		-		R/W	
Initial value	-		00		-		00	

[bit31:9] Reserved: Reserved bits

Write "0" to these bits.

[bit8] SDCRST: Reset control of SD card interface

This bit controls the reset of the SD card interface unit. If this bit is set to "1", the SD card interface becomes a reset state, the operation of the SD card interface stops, and the register settings are initialized. For products to which the SD card interface is not mounted, do not set this bit to "1". To release the above-mentioned reset state, be sure to set this bit to "0" again.

bit	Description
0	Releases the reset of SD card interface. (Initial value)
1	Issue the reset signal to SD card interface.

[bit7:6] Reserved: Reserved bits

Write "0" to these bits.

[bit5:4] CANRST[1:0]: Reset control of CAN controller

These bits control the reset of each CAN controller's channel unit. The correspondence between each bit and the CAN controller channel is shown below.

bit4 - CANRST0: CAN controller channel 0
bit5 - CANRST1 : CAN controller channel 1

If the relevant bit is set to "1", the channel of the corresponding CAN controller becomes a reset state, the CAN controller operation stops, and the register settings are initialized. For products to which the CAN controller channel is not mounted, do not change this bit from the initial value. To release the reset state, be sure to set this bit to "0" again.

bit	Description
0	Releases the reset of CAN controller channel corresponding to the relevant bit. (Initial value)
1	Issues the rest signal to CAN controller channel corresponding to the relevant bit.

[bit3:2] Reserved: Reserved bits

Write "0" to these bits.

[bit1:0] USBRST[1:0]: Reset control of USB (function/host)

These bits control the reset of each channel of USB (function/host). The correspondence between each bit and the USB channel is shown below.

bit0 - USBRST0 : USB channel 0
bit1 - USBRST1 : USB channel 1

If the relevant bit is set to "1", the channel of corresponding USB becomes a reset state, the USB operation stops, and the register settings are initialized. For products to which the relevant USB channel is not mounted, do not change this bit from the initial value. To release the reset state, be sure to set this bit to "0" again.

bit	Description
0	Releases the reset of USB channel corresponding to the relevant bit.(Initial value)
1	Issues the reset to USB channel corresponding to the relevant bit.

5. Peripheral Clock Gating Function Usage Precautions

This section explains the precautions for using peripheral clock gating functions by peripheral function.

■ Overview

● Control of a peripheral function to which a clock supply is stopped

The register access to a peripheral function to which a clock supply is stopped, both read and write, is not guaranteed. The read value is undefined, and the write operation is prohibited.

The internal state can be reset by controlling peripheral function reset control register 0 to 2 (MRST0, MRST1, and MRST2) while the peripheral clock is gated.

● Combination of peripheral clock settings

Be sure to set all the target peripheral functions to the clock supply side by the peripheral clock control registers 0 to 2 (CKEN0, CKEN1, and CKEN2) for the functions operated by combining two or more peripheral functions. For example, set a relevant unit of the A/D converter used and a relevant channel of the base timer to the clock supply side respectively by the peripheral clock control registers (CKEN0 and CKEN1) when the base timer is selected for use by the timer trigger of the A/D converter.

● Initialization conditions of peripheral clock settings

The peripheral clock gating function is initialized by the following reset. After issuing the following reset, be sure to reconfigure the peripheral clock gating function.

For details of the following resets, see Chapter "Resets".

- Power-on reset (PONR)
- Low voltage detection reset (LVDH)
- INITX pin input(INITX)
- Software watchdog reset (SWDGR)
- Hardware watchdog reset (HWDGR)
- Clock failure detection reset (CSVR)
- Anomalous frequency detection reset (FCSR)
- Software reset (SRST)
- APB2 bus reset (APBC2_PSR)
- Deep standby transition reset (DSTR)

■ Multi-Function Serial Interface

● LIN Sync field detection: LSYN

Execute the setting of the operation clock supply to the corresponding multi-function timer (input capture) separately with the setting of the peripheral clock of multi-function serial interface when the input capture (ICU) is used in the LIN bus interface mode. For the connection between the multi-function serial interface and the input capture, see "Extended Pin Function Setting Register (EPFR)" in Chapter of "I/O port".

■ Base Timer

● Clock setting unit of base timer

The peripheral clock control of the base timer is executed in the unit of four channels described in Table 5-1.

Table 5-1 Correspondence between Peripheral Clock Gating Setting and Base Timer Channels

Setting bit of Peripheral Clock Control Register (CKEN1)	Target Channels
bit 0	Base Timer ch3, ch2, ch1, ch0
bit 1	Base Timer ch7, ch6, ch5, ch4
bit 2	Base Timer ch11, ch10, ch9, ch8
bit 3	Base Timer ch15, ch14, ch13, ch12

■ Multi-function timer

● FRT Selection register

For using the following FRT selection function, set the operation clock of the multi-function timer unit on which source-side FRT is mounted to the supply side.

- OCU Connection FRT selection register (OCFS)
- ICU Connection FRT selection register (ICFS)
- ADC Start-up compare connection FRT selection register (ADCMP)

■ PPG

● Clock Control of PPG

The control of input clock to PPG synchronizes with the settings of input clock to the multi-function timer. For PPG channel numbers and unit numbers of multi-function timer, see Table 5-2.

Table 5-2 Multi-function Timer and PPG Input Clock Control

Unit number of multi-function timer	PPG channel number
Unit 0	Channel 0 to Channel 7
Unit 1	Channel 8 to Channel 15
Unit 2	Channel 16 to Channel 23
Unit 3	Channel 24 to Channel 31

■ External bus interface

● Individual reset control of external bus interface

When the external bus interface is individually reset with bit26 (EXBRST) of the peripheral function reset control register 0 (MRST0), the setting of the register in the external bus interface is not initialized though its operation stops.

■ USB (Function/Host)

● Clock control target

The gating and supplying of the clock for the USB communication cannot be controlled with USBCK bit of the peripheral clock control register 2 (CKEN2). Execute the control of the clock for the USB communication with UCEN bit of USB clock control register (UCCR) or USB/Ethernet clock control register (UCCR). For details, see "USB Clock Generation" in FM4 Family Peripheral Manual Communication Macro Part.

■ A/D Converter

● A/D Timer Trigger Selection

When the base timer is used as a startup trigger of the A/D converter, set the operation clock of the selected base timer channel to the supply side.

■ GPIO

● Restrictions when bus clock is gated

While the bus clock of GPIO is gated, some functions of I/O port cannot be used as shown in Table 5-3.
Be sure to confirm the using conditions and execute the bus clock control of GPIO.
For details on I/O port functions, see Chapter "I/O Port".

Table 5-3 Restrictions when GPIO clock is gated

Restrictions	Bus Clock Status	
	Supplied*	Gated*
GPIO Function-Input level reading (PDIR register reading)	Available	Prohibited
GPIO Function-Output Level Switching and Status Confirmation (PDOR register reading/writing)	Available	Prohibited
I/O port Mode Switching (Setting change of PFR, PCR, DDR, ADE, SPSR, EPFR, and PZR registers)	Available	Prohibited
Peripheral Function Operation (Signal Input and Output)	Available	Available
External Interrupt/NMI Control	Available	Available
Reset Input (INITX)	Available	Available
Return from Deep Standby Mode (WKUP pin input)	Available	Available

*: Available: can be used, Prohibited: cannot be used.

CHAPTER: High-Speed CR Trimming

This chapter explains the High-Speed CR Trimming Function.

CODE: 9BFCRTRIM_FM4-E01.0

1. High-Speed CR Trimming Function Overview

This section explains frequency trimming function of the high-speed CR oscillator.

The high-speed CR oscillators used for this device have fluctuation range in frequency accuracy due to process variation. The fluctuation range of frequency accuracy due to process variation and temperature change can be reduced by configuring the trimming function.

The high-speed CR trimming function consitsts of the frequency trimming setup unit and temperature trimming setup unit.

The frequency trimming setup unit has the following functions:

- It can be configured the high-speed CR frequency trimming by writing a trimming value to the High-speed CR oscillation Frequency Trimming Register (MCR_FTRM).
- By using ch.0 of Base Timer, the setting value to thefrequency tirmming register can be calculated from count value of the specified period.

The temperature trimming setup unit has the following function:

It can be configured the high-speed CR temperature compensation by writing a trimming value to the High-speed CR oscillation Temperature Trimming Register (MCR_TTRM).

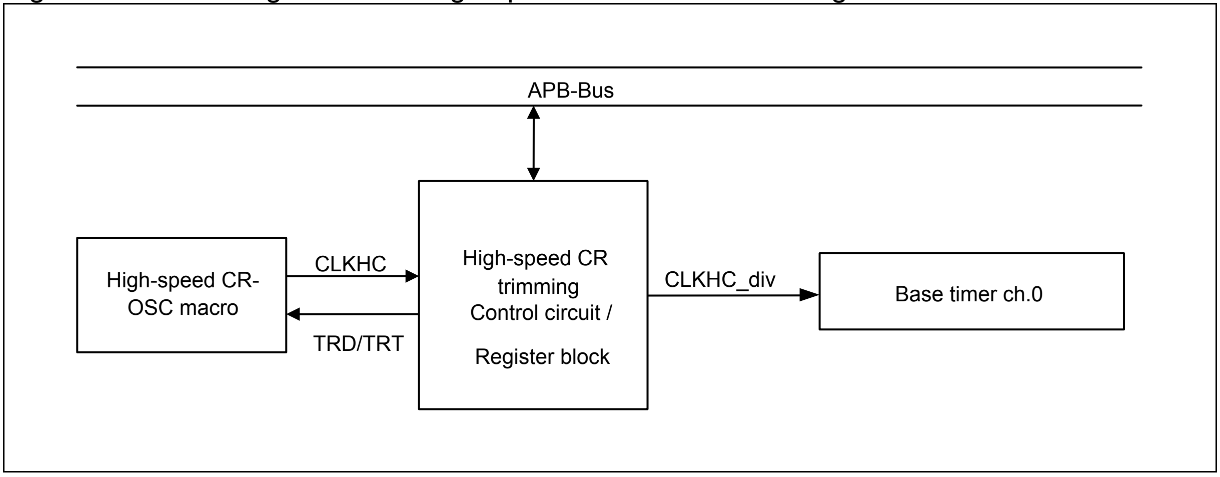
For the high-speed CR frequency accuracy, see electrical characteristics described in "Data Sheet" of the product used.

2. High-Speed CR Trimming Function Configuration and Block Diagram

This section explains the configuration and block diagram of high-speed CR oscillator frequency trimming function.

Figure 2-1 shows the block diagram of high-speed CR frequency trimming function.

Figure 2-1 Block diagram of the High-speed CR Oscillator Timing Circuit



■ Configuration

● High-speed CR OSC macro

A macro of the high-speed CR clock outputs CLKHC (high-speed CR clock).
In addition, the frequency trimming can be performed with TRD bit of high-speed CR oscillation frequency trimming register (MCR_FTRM) and TRT bit of high-speed CR oscillation temperature trimming register (MCR_TTRM).

● High-speed CR Trimming Control Circuit and register block

A control circuit and registers for trimming high-speed CR.
In addition, the high-speed CR clock (CLKHC_div) divided by the ratio set with CSR bit of high-speed CR oscillation frequency division setup register (MCR_PSR) is output to the base timer ch.0.

● Base timer

This block counts frequency before setting to calculate the frequency trimming data for high-speed CR.

<Note>

For the clock definition, see Chapter "Clock".

3. High-Speed CR Trimming Function Operation

This section explains operation conducted by frequency trimming function of the high-speed CR oscillator.

■ Operation of high-speed CR oscillation frequency trimming function

● Frequency trimming setup

The setup process writes a trimming data value to the High-speed CR oscillation Frequency Trimming Register (MCR_FTRM) to correct the misalignment of high-speed CR clock accuracy caused by process variation.

● Temperature trimming setup

The setup process writes a trimming data value to the High-speed CR oscillation Temperature Trimming Register (MCR_TTRM) to correct the misalignment of high-speed CR clock accuracy caused by temperature change.

● Register lock function

Write protect function is provided for the High-speed CR oscillation Frequency Trimming Register (MCR_FTRM) and the High-speed CR oscillation Temperature Trimming Register (MCR_TTRM), a function that protects the register from being rewritten without authorization when the system runs out of control.

● Trimming data acquisition

Data written to the High-speed CR oscillation Frequency Trimming Register (MCR_FTRM) can be acquired by one of the following three methods:

- Use the factory preset value stored in the "CR trimming" area inside the flash memory.
- Calculate by yourself the value set to the High-speed CR oscillation Frequency Trimming Register from the count value within a certain period by using base timer.
- Output high-speed CR clock to an external pin, monitor the waveform to trim the frequency and calculate by yourself the value set to the High-speed CR oscillation Frequency Trimming Register.

For data written to the High-speed CR oscillation Temperature Trimming Register (MCR_TTRM), use the factory preset value stored in the "CR trimming" area inside the flash memory.

<Notes>

- Erasing the flash memory also erases the "CR trimming" area inside the memory at the same time. If you use a value in the "CR trimming" area, therefore, save the data to other area (such as RAM) before erasing the flash memory, or only erase sectors other than in the "CR trimming" area.
- For the address of the "CR trimming" area, see "FLASH PROGRAMMING MANUAL" of the product used.

4. High-Speed CR Trimming Function Setup Procedure Example

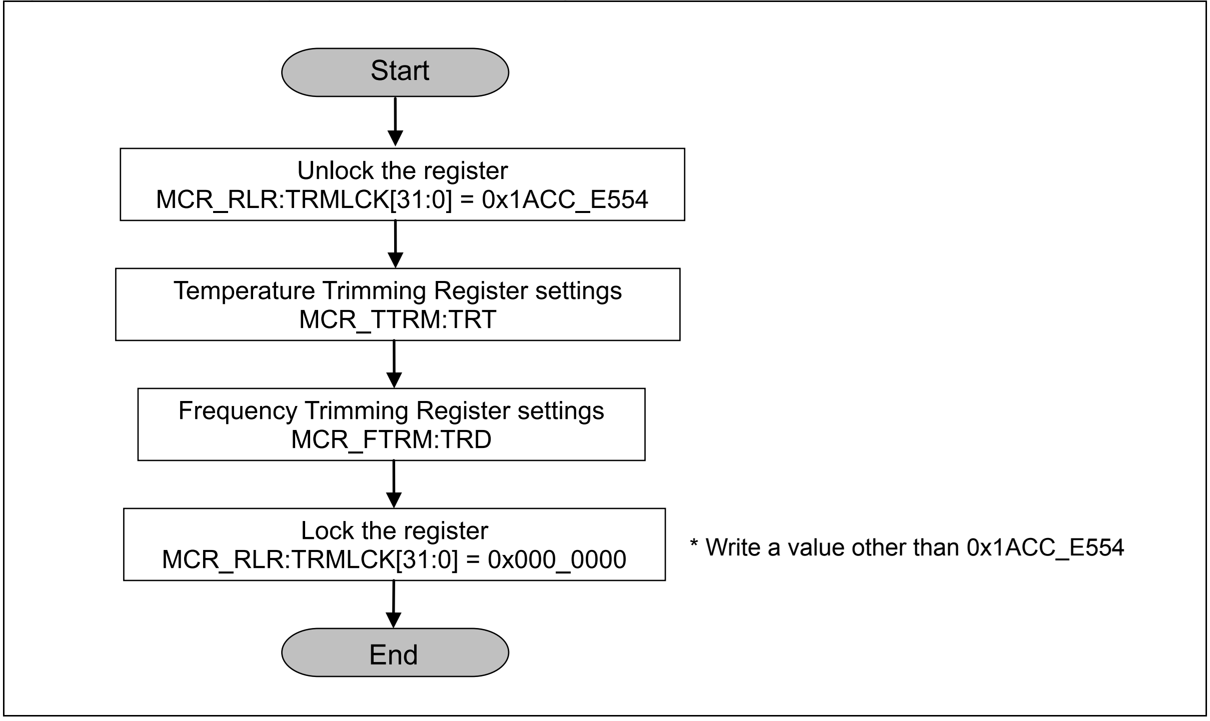
This section provides an example of setting up frequency trimming function of the high-speed CR oscillator.

■ Frequency trimming setup

Take the steps shown in Figure 4-1 to set up frequency trimming.

1. Write "0x1ACCE554" to TRMLCK[31:0] bits of High-speed CR frequency Register Write Protection register (MCR_RLR) to release the lock of Frequency Trimming Setup Register (MCR_FTRM)/ Temperature Trimming Setup Register (MCR_TTRM).
2. Set the trimming data to TRT bit of Temperature Trimming Setup Register (MCR_TTRM).
3. Set TRD bit of Frequency Trimming Setup Register (MCR_FTRM).
4. Write a value other than "0x1ACCE554" to TRMCLK[31:0] bits of High-speed CR Oscillation Register Write Protection Register (MCR_RLR) to lock the Frequency Trimming Setup Register (MCR_FTRM)/ Temperature Trimming Setup Register (MCR_TTRM).

Figure 4-1 Frequency/Temperature trimming setup


Start
Unlock the register
MCR_RLR:TRMLCK[31:0] = 0x1ACC_E554
Temperature Trimming Register settings
MCR_TTRM:TRT
Frequency Trimming Register settings
MCR_FTRM:TRD
Lock the register
MCR_RLR:TRMLCK[31:0] = 0x000_0000
* Write a value other than 0x1ACC_E554
End

■ Frequency trimming data acquisition example

When acquiring the data from the "CR trimming" area in the flash memory;

Read the "CR trimming" area in the flash memory and get the data.
Write the acquired value to TDR bit of the High-speed CR oscillation Frequency Trimming Setup Register (MCR_FTRM).

■ How to calculate the frequency trimming data

The following explains how to calculate the trimming data of high-speed CR oscillation.

1. Let Ftgt, a target oscillation frequency be 4[MHz] and Ttgt, its cycle be 250[ns](Ftgt: 4[MHz]). Let Xtrm_coarse and Xtrm_fine be the TRD[9:5] bit values and TRD[4:0] bit values of the High-speed CR Ocsilation Frequency Trimming Setup register at the time respectively.
2. Set "0b00000" to TRD[4:0] bits.
3. Let Xtrm_coarse be Xtrmmin_coarse w hen "0b00000" is set to TRD[4:0] bits. Let Tmax_coarse[sec] be the cycle at this time.
4. Let Xtrm_coarse be Xtrmmax_coarse when "0b11111" is set to TRD[9:5] bits. Let Tmin_coarse[sec] be the cycle at this time.
5. By calculating te following expression, obtain TRD[9:5] setting value, Xtrm_coarse giving the value more than target oscilation cycle, Ttgt.

$$Xtrm\_coarse = \frac{Ttgt - \frac{T\max\_coarse - T\min\_corse}{31} - T\max\_coarse}{\frac{T\min\_coarse - T\max\_coarse}{31}}$$

6. *: Round down decimals.
7. Set the obtained Xtrm_coarse to TRD[9:5] bits.
8. Confirm that the High-speed CR clock, $F_{CRH,}$ after setting TRD bits is Ftgt or less. If the F_{CRH} exceed Ftgt, subtract "1" from Xtrm_coarse and then return to Step 6.When the F_{CRH} is Ftgt or less, go to Step 8.
9. Let the value when "0b00000" is set to TRD[4:0] be Xtrimmin_fine. Let Tmax_fine[sec] be the cycle at this time.
10. Let the value when "0b11111" is set to TRD[4:0] be Xtrimmax_fine. Let Tmin_fine[sec] be the cycle at this time.
11. By calculating te following expression, obtain TRD[4:0] setting value, Xtrm_fine giving the target oscilation cycle, Ttgt.
12.
13.

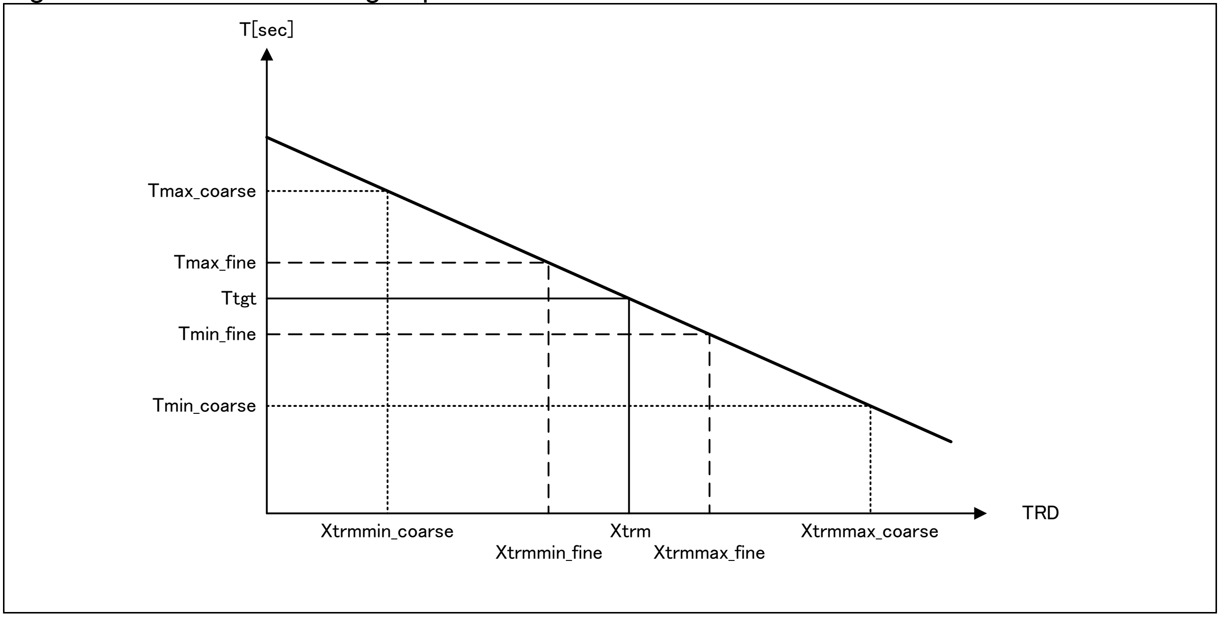
$$Xtrm\_fine = \frac{Ttgt - \frac{T\max\_fine - T\min\_fine}{31} - T\max\_fine}{\frac{T\min\_fine - T\max\_fine}{31}}$$

14. *: Round down decimals.
15. Set the obtained Xtrm_fine to TRD[4:0] bits.
16. Confirm whether the High-speed CR clock, F_{CRH}, after setting TRD bits is Ftgt or more and within the specification value of the High-speed CR clock oscilation frequency. If F_{CRH} exceeds the specification value, subtract "1" from Xtrm_fine and return to Step 9. Moreover, if F_{CRH} is less than Ftgt, add "1" to Xtrm_fine and return to Step 11. When the value is within the specification values, the calculation of trimming data is finished.

<Note>

For specifications of High-speed CR Clock Oscillation frequency, see "Data Sheet" of the product used.

Figure 4-2 Method to trim high-speed CR clock



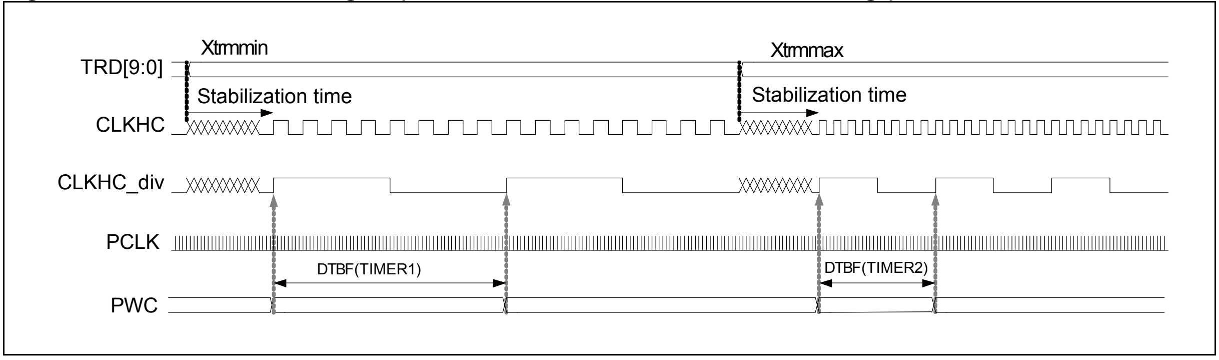
<Note>

For information about how to measure Tmin_coarse/fine and Tmax_coarse/fine, see "■ Example of trimming data acquisition using base timer".

■ Example of trimming data acquisition using base timer

Figure 4-3 shows the time chart of high-speed CR oscillation and the trimming process.

Figure 4-3 Time chart of high-speed CR oscillation and the trimming process with base timer



Run the base timer by setting the main oscillation clock (CLKMC) as the master clock (measurement reference clock).
Activate a trigger on the rising of the high-speed CR frequency division clock (CLKHC_div) when setting Xtrmmin or Xtrmmax, read the base timer value at that time, and perform the following calculations.

Tmax = (TIMER1×PCLK) / DIV

Tmin = (TIMER2×PCLK) / DIV

- · TIMER1, TIMER2 : Count value of base timer (PWC)
- · PCLK : APB1 bus clock
- · DIV : Frequency division ratio set by CSR bit of Division Setting Register(MCR_PSR)

Example: When PCLK = 40 MHz (25 ns), frequency division ratio = 1/8, and TIMER1 = 100,

Tmax = (100 × 25 ns) / 8 = 312.5 ns

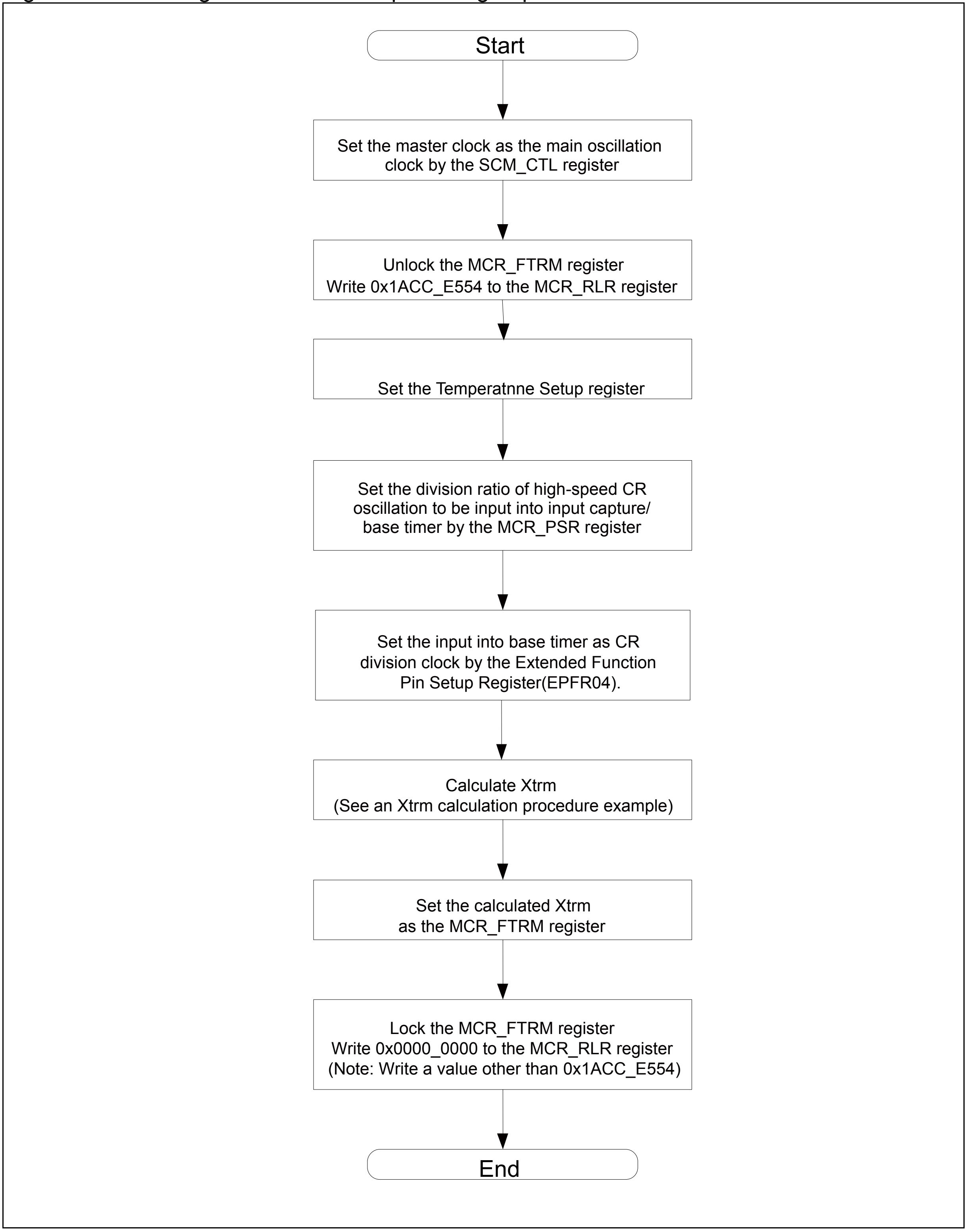
<Note>

The base timer used for trimming is ch.0.
PCLK in Figure 4-3 is an APB1 bus clock.
At this time, select the the master clock as the main oscillation for PCLK.

■ Frequency trimming procedure example

Figure 4-4 shows a trimming procedure example of high-speed CR oscillation.

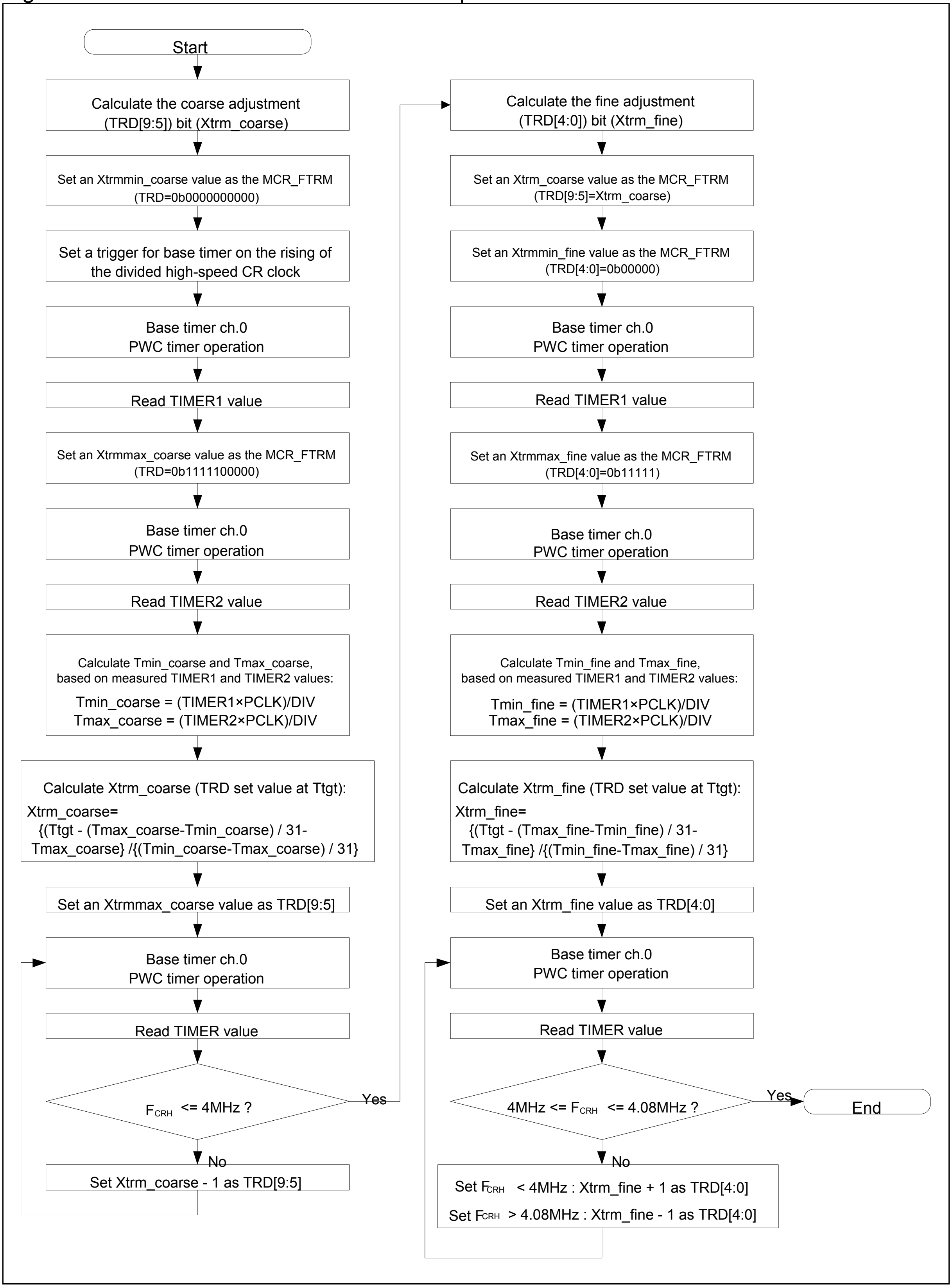
Figure 4-4 Trimming Procedure Example of High-speed CR Oscillation


Start
Set the master clock as the main oscillation clock by the SCM_CTL register
Unlock the MCR_FTRM register
Write 0x1ACC_E554 to the MCR_RLR register
Set the Temperatnne Setup register
Set the division ratio of high-speed CR oscillation to be input into input capture/ base timer by the MCR_PSR register
Set the input into base timer as CR division clock by the Extended Function Pin Setup Register(EPFR04).
Calculate Xtrm
(See an Xtrm calculation procedure example)
Set the calculated Xtrm as the MCR_FTRM register
Lock the MCR_FTRM register
Write 0x0000_0000 to the MCR_RLR register
(Note: Write a value other than 0x1ACC_E554)
End

■ Xtrm calculation procedure example

Figure 4-5 shows an Xtrm calculation procedure example. Perform frequency trimming in the two stages of coarse adjustment and fine adjustment.

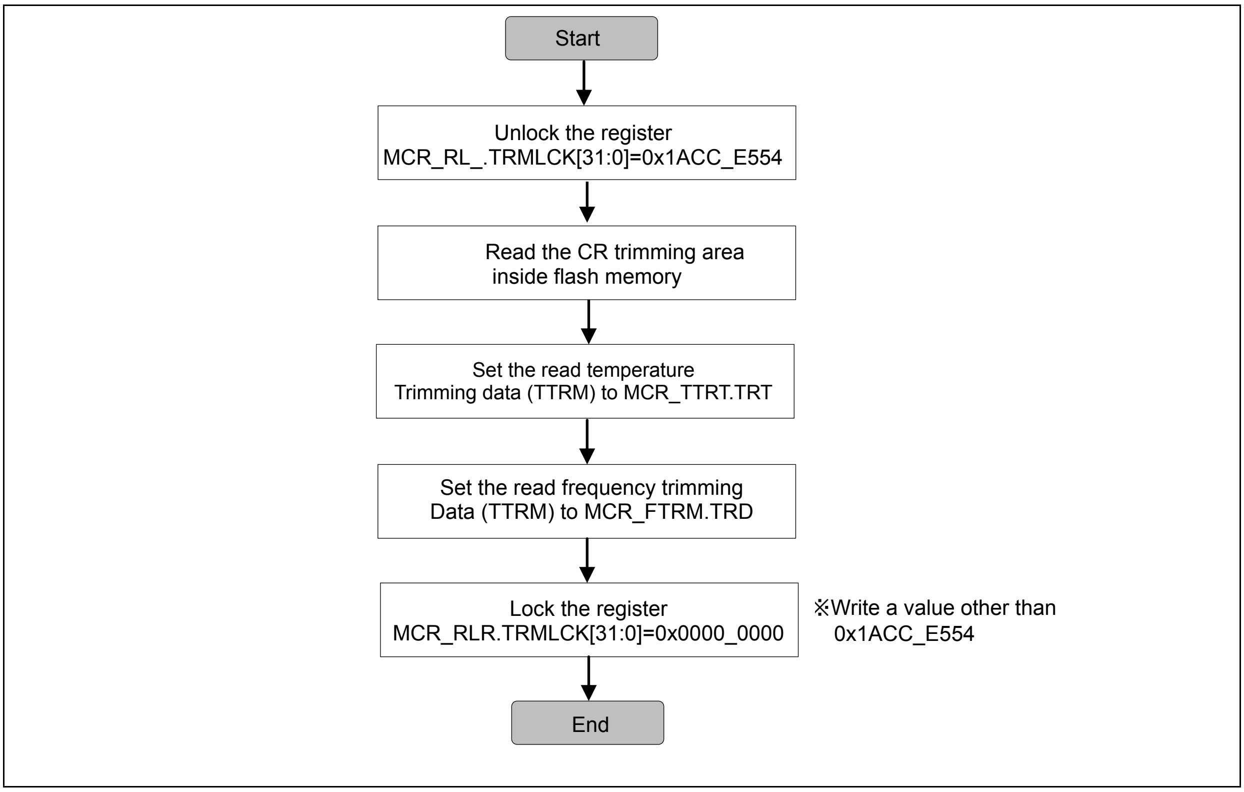
Figure 4-5 Xtrm Calculation Procedure Example


Start
Calculate the coarse adjustment (TRD[9:5]) bit (Xtrm_coarse)
Set an Xtrmmin_coarse value as the MCR_FTRM (TRD=0b0000000000)
Set a trigger for base timer on the rising of the divided high-speed CR clock
Base timer ch.0 PWC timer operation
Read TIMER1 value
Set an Xtrmmax_coarse value as the MCR_FTRM (TRD=0b1111100000)
Base timer ch.0 PWC timer operation
Read TIMER2 value
Calculate Tmin_coarse and Tmax_coarse, based on measured TIMER1 and TIMER2 values:
Tmin_coarse = (TIMER1×PCLK)/DIV
Tmax_coarse = (TIMER2×PCLK)/DIV
Calculate Xtrm_coarse (TRD set value at Ttgt):
Xtrm_coarse=
{(Ttgt - (Tmax_coarse-Tmin_coarse) / 31-
Tmax_coarse} /{(Tmin_coarse-Tmax_coarse) / 31}
Set an Xtrmmax_coarse value as TRD[9:5]
Base timer ch.0 PWC timer operation
Read TIMER value
F_{CRH} <= 4MHz ?
Yes
No
Set Xtrm_coarse - 1 as TRD[9:5]
Calculate the fine adjustment (TRD[4:0]) bit (Xtrm_fine)
Set an Xtrm_coarse value as the MCR_FTRM (TRD[9:5]=Xtrm_coarse)
Set an Xtrmmin_fine value as the MCR_FTRM (TRD[4:0]=0b00000)
Base timer ch.0 PWC timer operation
Read TIMER1 value
Set an Xtrmmax_fine value as the MCR_FTRM (TRD[4:0]=0b11111)
Base timer ch.0 PWC timer operation
Read TIMER2 value
Calculate Tmin_fine and Tmax_fine, based on measured TIMER1 and TIMER2 values:
Tmin_fine = (TIMER1×PCLK)/DIV
Tmax_fine = (TIMER2×PCLK)/DIV
Calculate Xtrm_fine (TRD set value at Ttgt):
Xtrm_fine=
{(Ttgt - (Tmax_fine-Tmin_fine) / 31-
Tmax_fine} /{(Tmin_fine-Tmax_fine) / 31}
Set an Xtrm_fine value as TRD[4:0]
Base timer ch.0 PWC timer operation
Read TIMER value
4MHz <= F_{CRH} <= 4.08MHz ?
Yes
End
No
Set F_{CRH} < 4MHz : Xtrm_fine + 1 as TRD[4:0]
Set F_{CRH} > 4.08MHz : Xtrm_fine - 1 as TRD[4:0]

■ Procedure example of using "CR trimming" area storage data inside flash memory

Figure 4-6 shows a procedure example of reading trimming data stored in the "CR trimming" area inside the flash memory and setting it in the High-speed CR oscillation Frequency Trimming Register.

Figure 4-6 Procedure Example of Using "CR Trimming" Area Storage Data


Start
Unlock the register
MCR_RL_.TRMLCK[31:0]=0x1ACC_E554
Read the CR trimming area
inside flash memory
Set the read temperature
Trimming data (TTRM) to MCR_TTRT.TRT
Set the read frequency trimming
Data (TTRM) to MCR_FTRM.TRD
Lock the register
MCR_RLR.TRMLCK[31:0]=0x0000_0000
※Write a value other than
0x1ACC_E554
End

<Note>

For the address of the CR trimming area, see "FLASH PROGRAMMING MANUAL" for the product used.

5. High-Speed CR Trimming Function Register List

The following lists and explains registers used for frequency trimming function of the high-speed CR oscillator.

Table 5-1 lists the registers.

Table 5-1 Register list

Abbreviation	Register name	Reference
MCR_PSR	High-speed CR oscillation Frequency Division Setup Register	5.1
MCR_FTRM	High-speed CR oscillation Frequency Trimming Register	5.2
MCR_TTRM	High-speed CR oscillation Temperature Trimming Register	5.3
MCR_RLR	High-speed CR oscillation Register Write-Protect Register	5.4

5.1. High-speed CR oscillation Frequency Division Setup Register (MCR_PSR)

The MCR_PSR register sets the frequency division ratio of high-speed CR oscillation.
A divided clock can be input in base timer.

■ Register configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved					CSR		
Attribute	-					R/W		
Initial value	-					001		

■ Register functions

[bit7:3] Reserved : Reserved bits

"0b00000" is read from these bits.
Set these bits to "0b00000" when writing.

[bit2:0] CSR: High-speed CR oscillation frequency division ratio setting bits

bit2	bit1	bit0	Description
0	0	0	1/4
0	0	1	1/8 [Initial value]
0	1	0	1/16
0	1	1	1/32
1	0	0	1/64
1	0	1	1/128
1	1	0	1/256
1	1	1	1/512

5.2. High-speed CR oscillation Frequency Trimming Register (MCR_FTRM)

The MCR_FTRM register sets the frequency trimming value.

■ Register configuration

bit	31	16
Field	Reserved	
Attribute	-	
Initial value	-	

bit	15	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved		TRD[9:0]									
Attribute	-		R/W									
Initial value	-		0111101111									

■ Register functions

[bit31:10] Reserved : Reserved bits

"0" is always read from these bits.
These bits have no effect when written.

[bit9:0] TRD[9:0] : Frequency trimming setup bits

bit9:5	Description
When write	These bits make fine adjustment to the high-speed CR oscillator frequency. For values to be set, see trimming data acquisition in the operation explanation of the frequency trimming function. Thess bits fluctuate in frequency steps of approximately2.3% each time ±1 setting is made.
When read	A specified value is read. As an initial value, "0b01111" is read.

bit4:0	Description
When write	Thess bits make fine adjustment to the high-speed CR oscillator frequency. For values to be set, see trimming data acquisition in the operation explanation of the frequency trimming function. Thess bits fluctuate in frequency steps of approximately 0.14% each time ±1 setting is made.
When read	A specified value is read. As an initial value, "0b01111" is read.

<Notes>

- This register is not initialized by software reset.
- For values to be set to the TRD bits, see trimming data acquisition in the operation explanation of the frequency trimming function.

5.3. High-speed CR Oscilation Temperature Trimming Setup Register (MCR_TTRM)

The MCR_TTRM register sets the temperature trimming value.

■ Regsiter Configuration

bit	31	16
Field	Reserved	
Attribute	-	
Initial value	-	

bit	15	5	4	3	2	1	0
Field	Reserved		TRT[4:0]				
Attribute	-		R/W				
Initial value	-		10000				

■ Register functions

[bit31:5] Reserved : Reserved bits

"0" is always read from these bits.
These bits have no effect when written.

[bit4:0] TRT[4:0] : Temperature trimming setup bits

bit4:0	Description
When write	Thess bits make adjustment to the high-speed CR oscillator frequency. Write the value read from Temperature Trimming bit storage area in Flash Mmemory. For Temperature Trimming bit storage area, see “FLASH PROGRAMING MANULA” of the product used.
When read	A specified value is read. As an initial value, 0b10000 is read.

<Notes>

- This register is not initialized by software reset.
- Before obtaining the frequency trimming data, be sure to set this register.

5.4. High-Speed CR Oscillation Register Write-Protect Register (MCR_RLR)

The MCR_RLR register controls the write-protect state of the frequency trimming register (MCR_FTRM)/high-speed CR oscillation temperature trimming register (MCR_TTRM).

■ Register configuration

bit	31 … 16
Field	TRMLCK[31:16]
Attribute	R/W
Initial value	0x0000

bit	15 … 0
Field	TRMLCK[15:0]
Attribute	R/W
Initial value	0x0001

■ Register functions

[bit31:0] TRMLCK[31:0] : Register write-protect bits

bit31:0	Description
When read	When 0x00000000 is read, the MCR_FTRM/MCR_TTRM register is currently unlocked. When 0x00000001 is read, the MCR_FTRM/MCR_TTRM register is currently locked.
Writing a value other than 0x1ACCE554	Locks the MCR_FTRM/MCR_TTRM register
Writing 0x1ACCE554	Unlocks the MCR_FTRM/MCR_TTRM register

<Note>

This register is not initialized by software reset.

6. High-Speed CR Trimming Function Usage Precautions

This section explains the precautions for using the high-speed CR trimming function.

- Low-speed CR oscillator
  This trimming function is only enabled for the high-speed CR oscillator.
  It cannot apply to the low-speed CR oscillator.

- Data stored in the "CR trimming" area
  The CR trimming" area stores the factory preset frequency trimming data. For the address of the "CR trimming" area, see "FLASH PROGRAMMING MANUAL" for the product used.
  When Data in flash memory is erased, the data in “CR trimming” area is also erase at the same time. To use the data in the “CR trimming” area, save the data in the “CR trimming” area to other area such as RAM before erasing the data in flash memory.
  Otherwise, erase the sectors other than those in “CR trimming”area.

- For High-seed CR oscilatior oscilation frequency accuracy
  Without setting High-speed CR oscilation temperature trimming register (MCR_TTRM) and High-speed CR oscilation temperature trimming register (MCR_FTRM), the accuracy of the High-speed CR oscillator described in “Data Sheet” cannot be guaranteed. So, be sure to set the above registers before use.

- How to use base timer
  For information about how to use base timer, see Chapters "Base Timer" in "Timer Part" and "I/O Port".

- FCS (Anomalous Frequency Detection)
  For FCS function (anomalous frequency detection), see Chapter "Clock supervisor". Do not perform CR trimming after the FCS function is enabled.

CHAPTER: Low-speed CR Prescaler

This chapter shows the functions and operation of low-speed CR Prescaler.

CODE:9BFLCPC-E01.0

1. Low-speed CR Prescaler Overview

This section shows the overview of low-speed CR prescaler.

■ Low-speed CR Prescaler

By setting the low-speed CR prescaler load register(LCR_PRSLD), the low-speed CR prescaler divides low-speed CR and generates low-speed CR clock(CLKLC).
This macro can correct the accuracy of low-speed CR. For the correcting method, see the example of correcting low-speed CR.

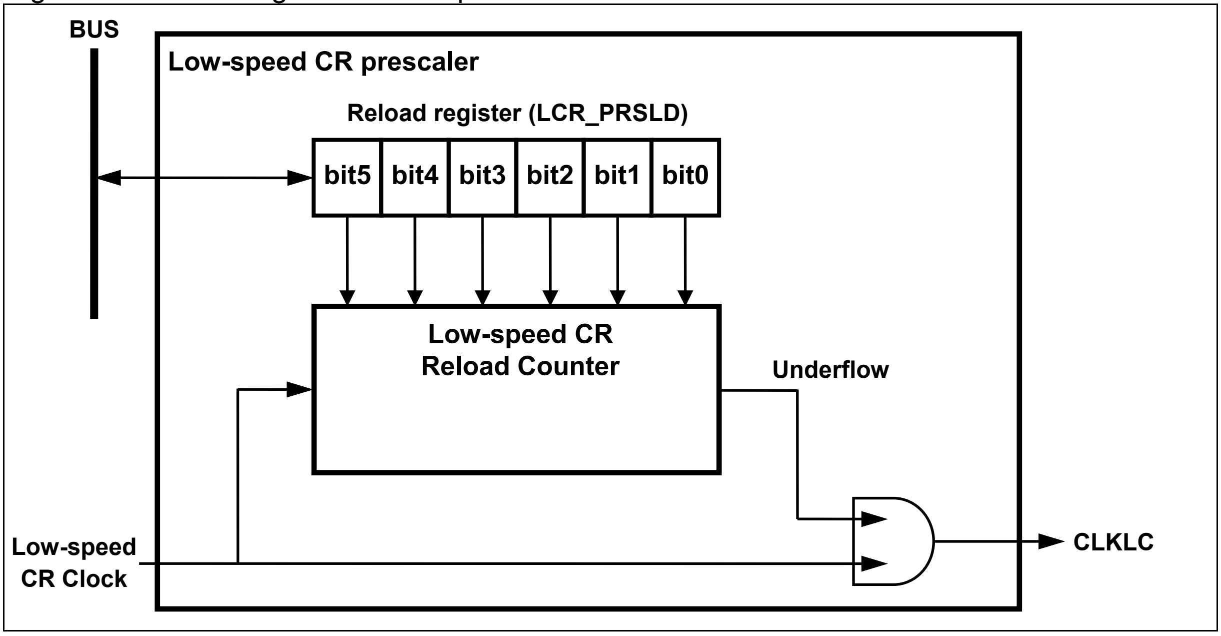
2. Low-speed CR Prescaler Configuration

This section shows the block diagram of low-speed CR prescaler.

■ Block diagram of low-speed CR prescaler

For the block diagram of low-speed CR prescaler, see Figure 2-1.

Figure 2-1 Block Diagram of Low-speed CR Prescaler



● Low-speed CR Prescaler Load Register (LCR_PRSLD)

Sets the division ratio (reload value) of Low-speed CR Prescaler.

● Low-speed CR Reload Counter

This is the down counter which generates the Low-speed CR Division Clock (CLKLC).

3. Low-speed CR Prescaler Operation and Setup Procedure Example

This section explains the operation of Low-speed CR Prescaler. This section also shows the example of setup procedures.

■ Setup procedures of Low-speed CR Prescaler

The Low-speed CR is asynchronous with the peripheral clock (PCLK).
For writing to the Low-speed CR Prescaler Reload Register, the peripheral clock is used. Therefore, if the setting change of the Low-speed CR Prescaler Load Register and the reload of the reload counter occur simultaneously, a value reloaded to the reload counter is not guaranteed.
So, execute the rewriting of the Low-speed CR Prescaler Reload Register conforming to the following procedures.

● For Switching the division clock

The initial value of the Low-speed CR Prescaler Reload Register(LCR-PRSLD) is “0”.
Thus, for changing the value from the initial value, these procedures are unnecessary.

1. Set "0" to the Low-speed CR Prescaler Reload Register (LCR_PRSLD).
2. Wait until the value of the Low-speed CR Prescaler Reload Register (LCR_PRSLD) is reloaded to the reload counter.
3. The wait time is obtained by calculating the following formula:
   Low-speed CR cycle (50 kHz: 20 µs) × "the set value before changed to "0" in Item 1."
4. Write new setup value to the Low-speed CR Prescaler Reload Register (LCR_PRSLD).

For wait time at setup change, see Table 3-1.

Table 3-1 Setup Wait Time

Reload Value before Setup	Setup Value	Wait Time
0	0	Not exists.
1	0	20 µs (20 µs × 1)
2	0	40 µs (20 µs × 2)
3	0	60 µs (20 µs × 3)
:	:	:
60	0	1200 µs (20 µs × 60)
61	0	1220 µs (20 µs × 61)
62	0	1240 µs (20 µs × 62)
63	0	1260 µs (20 µs × 63)

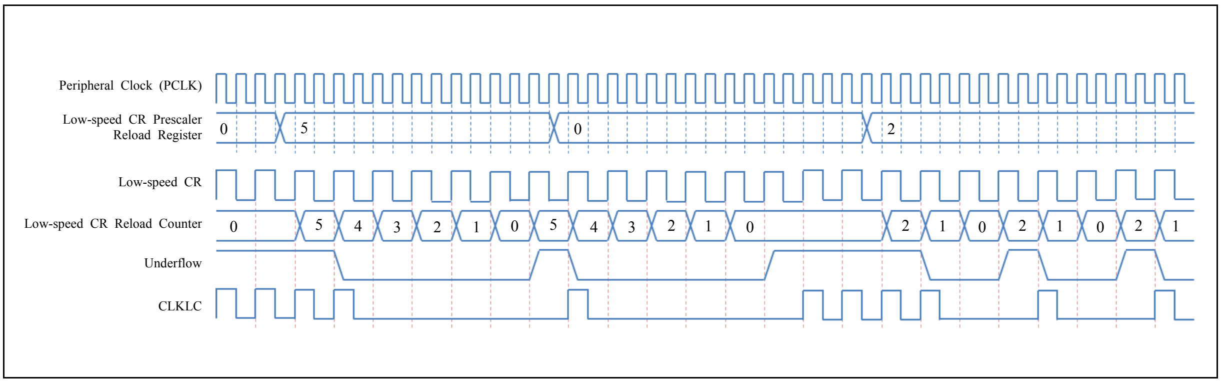
<Notes>

- The division clock cannot be stopped.
- The setting of the Low-speed CR Prescaler Reload Register (LCR_PRSLD) is executed at the underflow of the Low-speed CR Reload Counter.

■ Operation of Low-speed CR Prescaler

For the operation of the Low-speed CR Prescaler, see Figure 3-1.

Figure 3-1 Low-speed CR Prescaler Operation

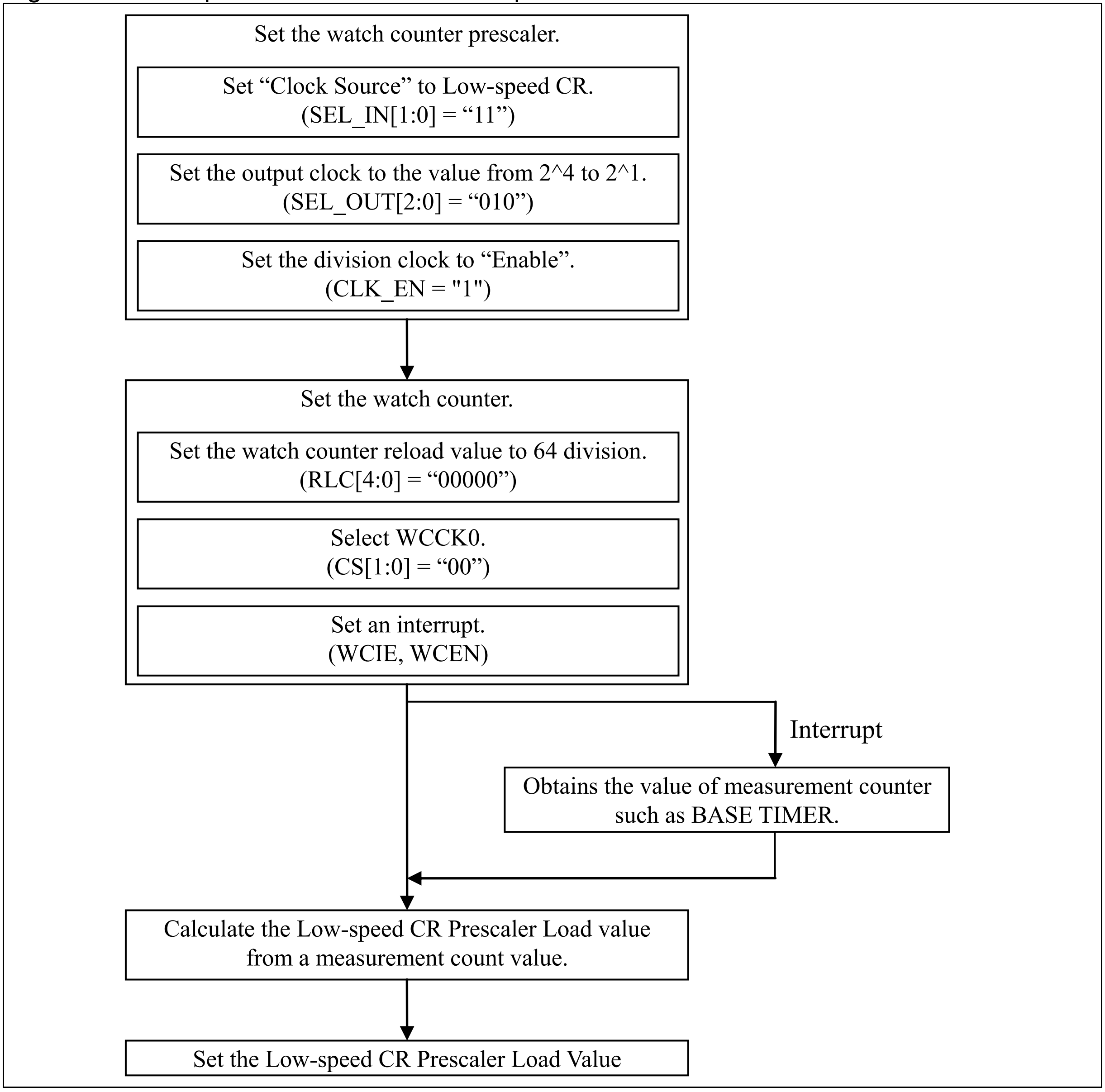


(1) Sets the Low-speed CR Prescaler Load Register (LCR_PRSLD in synchronization with the peripheral clock (PCLK)
(2) Retrieves the value of the Low-speed CR Prescaler Load Register (LCR_SLD) at the moment the Low-speed CR Reload Counter indicates "0".
(3) Outputs the Low-speed CR (CLKLC) at the moment when the Low-speed CR Reload Counter underflow occurs.

■ Low-speed CR Correction Example

For the correction example of the Low-speed C, see Figure 3-2.

Figure 3-2 Low-speed CR Correction Example


Set the watch counter prescaler.
Set “Clock Source” to Low-speed CR.
(SEL_IN[1:0] = “11”)
Set the output clock to the value from 2^4 to 2^1.
(SEL_OUT[2:0] = “010”)
Set the division clock to “Enable”.
(CLK_EN = "1")
Set the watch counter.
Set the watch counter reload value to 64 division.
(RLC[4:0] = “00000”)
Select WCCK0.
(CS[1:0] = “00”)
Set an interrupt.
(WCIE, WCEN)
Interrupt
Obtains the value of measurement counter such as BASE TIMER.
Calculate the Low-speed CR Prescaler Load value from a measurement count value.
Set the Low-speed CR Prescaler Load Value

4. Low-speed CR Prescaler Register

This section shows the list of the Low-speed CR Prescaler Register.

■ Low-speed CR Prescaler Register

Table 4-1 List of Low-speed CR Prescaler Register

Abbreviation	Register name	Reference
LCR_PRSLD	Low-speed CR Prescaler Control Register	4.1

4.1. Low-speed CR Prescaler Control Register (LCR_PRSLD)

The Low-speed CR Prescaler Control Register is used to set the division ratio of low-speed CR.

bit	7	6	5	4	3	2	1	0
Field	Reserved		LCR_PRSLD[5:0]					
Attribute	-		R/W					
Initial Value	00		000000					

[bit7:6] Reserved: Reserved bits

Always “0” is read.
They have no effect in write mode.

[bit5:0] LCR_PRSLD: Low-speed CR Prescaler Load

At writing, sets the division ratio of the Low-speed CR Prescaler (the reload value of a reload counter) .
At reading, the set value is read.

＜Note＞

This register is not initialized with software reset.

CHAPTER: Clock supervisor

This chapter explains the clock supervisor functions.

CODE: 9BFCSV_FM4-E01.0

1. Overview

This section provides an overview of the clock supervisor functions.

The clock supervisor includes the following two types of functions.

- **Clock failure detection (CSV: Clock failure detection by clock Supervisor)**

  The clock failure detection monitors the main and sub clocks. If a rising edge of the monitored clock is not detected within the specified period, this function determines that the oscillator has failed, and outputs a system reset request.

- **Anomalous frequency detection (FCS: anomalous Frequency detection by Clock Supervisor)**

  The anomalous frequency detection monitors frequency of the main clock. Within the specified period between an edge and the next edge of the divided clock of high-speed CR, this function counts up the internal counter value using the main clock. If the count value reaches out of the set window range, the function determines that the main clock frequency is anomalous, and outputs an interrupt request or a system reset request to the CPU.

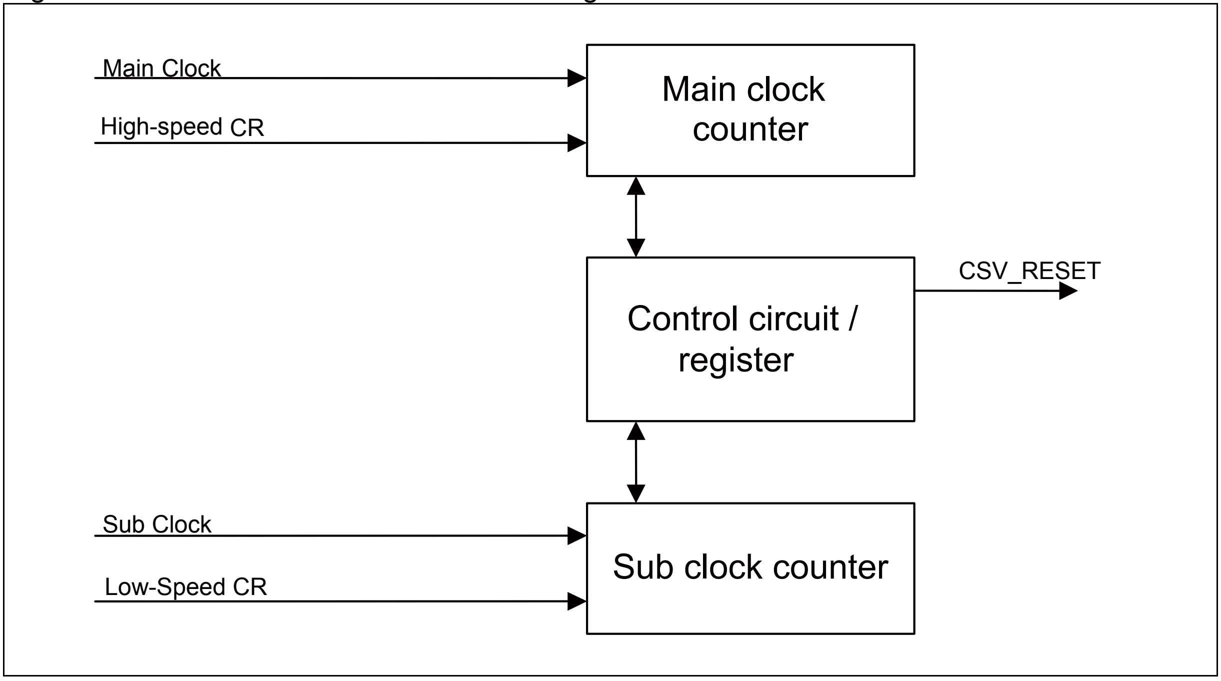
2. Configurations and Block Diagrams

This section explains the block diagram of the clock supervisor functions.

■ Clock failure detection

Figure 2-1 shows the block diagram of the clock failure detection.

Figure 2-1 Clock Failure Detection Block Diagram



The clock failure detection consists of the following three types of blocks.

● Control circuit/register

- This block includes a circuit controlling the clock failure detection,
- Also includes setup registers enabling/disabling the clock failure detection.

● Main clock counter

A counter that monitors the main clock with the high-speed CR clock.

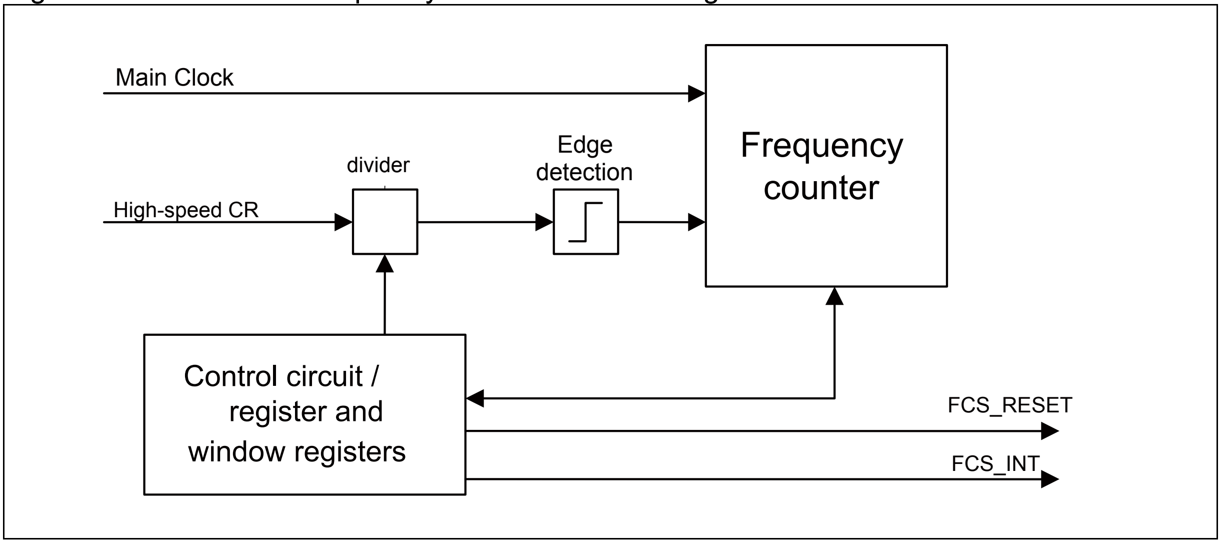
● Sub clock counter

A counter that monitors the sub clock with the low-speed CR clock.

■ Anomalous frequency detection

Figure 2-2 shows the block diagram of the anomalous frequency detection.

Figure 2-2 Anomalous Frequency Detection Block Diagram



The anomalous frequency detection consists of the following three types of blocks.

● Control circuit/register and window registers

- This block includes a circuit controlling the anomalous frequency detection.
- Also includes setup registers enabling/disabling the anomalous frequency detection.
- Also includes window registers defining the frequency range for measurements.

● Frequency counter

A counter based on the main clock.

● Divider/edge detection

- This block divides the high-speed CR.
- Also detects rising edges of the divided clock of high-speed CR.

3. Explanation of Operations

This section explains the operations of the clock supervisor functions.

■ Clock failure detection function

The clock failure detection function monitors the main and sub clocks. If a rising edge of the monitored clock is not detected within the specified period, this function determines that the oscillator has failed, and outputs a system reset request.

- This reset request is referred to as the CSV reset request.
- CSV function monitors each of the main and sub clocks independently.
- It stops monitoring when the main and sub oscillators stop oscillating.
- It stops monitoring while waiting for oscillation stabilization wait time.
- When the oscillation stabilization wait time of main and sub oscillators ends, CSV function is automatically enabled.

<Notes>

- Each of the main and sub clocks can be enabled/disabled independently using the CSV control register (CSV_CTL).
- The main clock is monitored with the high-speed CR clock, and the sub clock is monitored with the low-speed CR clock. When a rising edge is not detected within 32 clocks of high-speed CR for the main clock, or within 32 clocks of low-speed CR for the sub clock, this function determines that the oscillator has failed.

■ Anomalous frequency detection function

The anomalous frequency detection function monitors the main clock.

Within the specified period between a rising edge and the next rising edge of the divided clock of high-speed CR, this function counts up the internal counter using the main clock. If the count value reaches out of the set window range, the function determines that the main clock frequency is anomalous, and outputs an interrupt request or a system reset request to the CPU.

- This interrupt request is referred to as the FCS interrupt request, and reset request as the FCS reset request.
- The FCS function only monitors frequency of the main clock.
- It stops monitoring when the main oscillator stops oscillating.
- It stops monitoring while waiting for oscillation stabilization wait time.
- The FCS function is started with software, a user program.

<Notes>

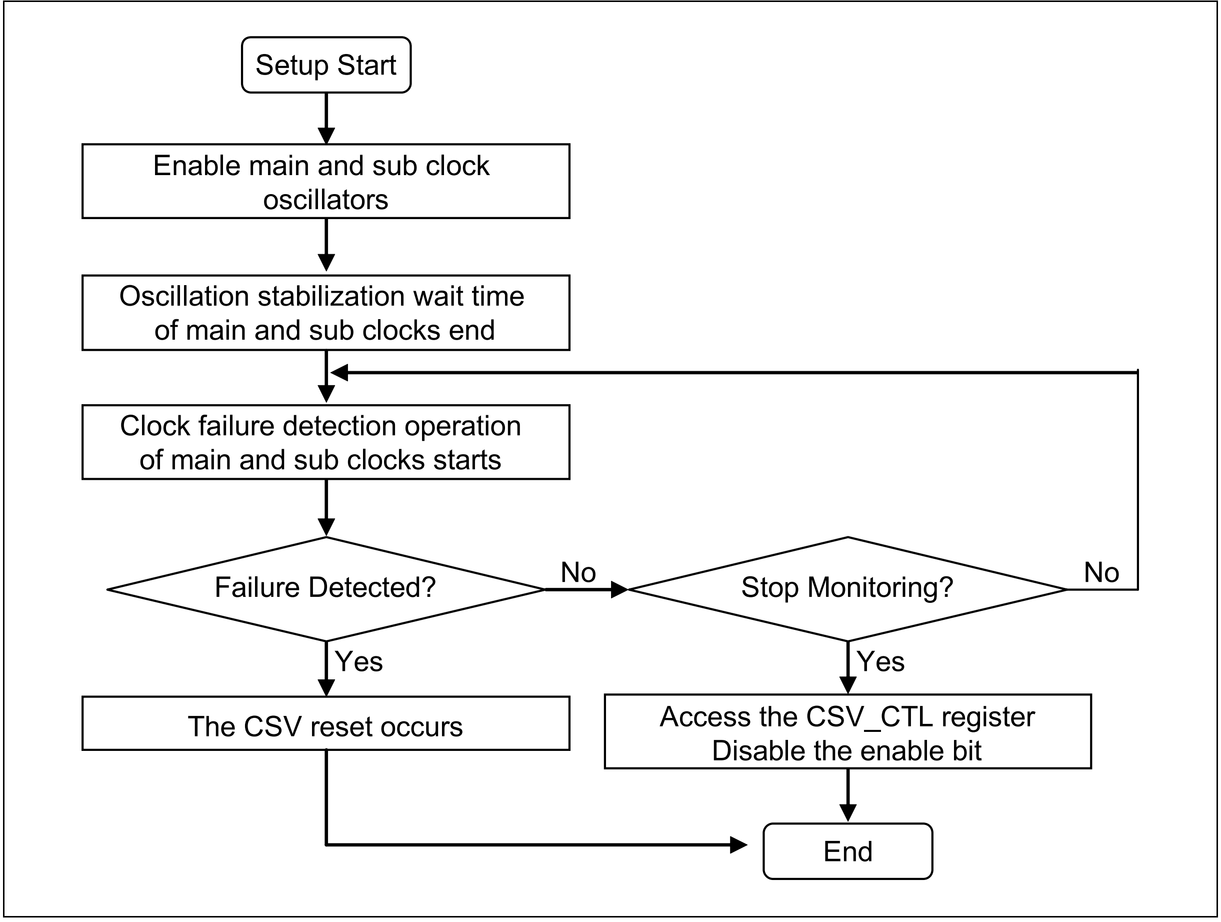
- If the FCS reset is enabled:
  An interrupt request occurs the first time a counter value deviates from the set window. If the interrupt request has not been cleared, and the counter value falls out of the specified window, a system reset request is output.
  If the FCS reset is not enabled, the reset request is masked.

- The counter value, if it goes out of the specified window, is stored in the frequency detection counter register (FCSWD_CTL).

4. Setup Procedure Examples

This section explains examples of setting up the clock supervisor functions.

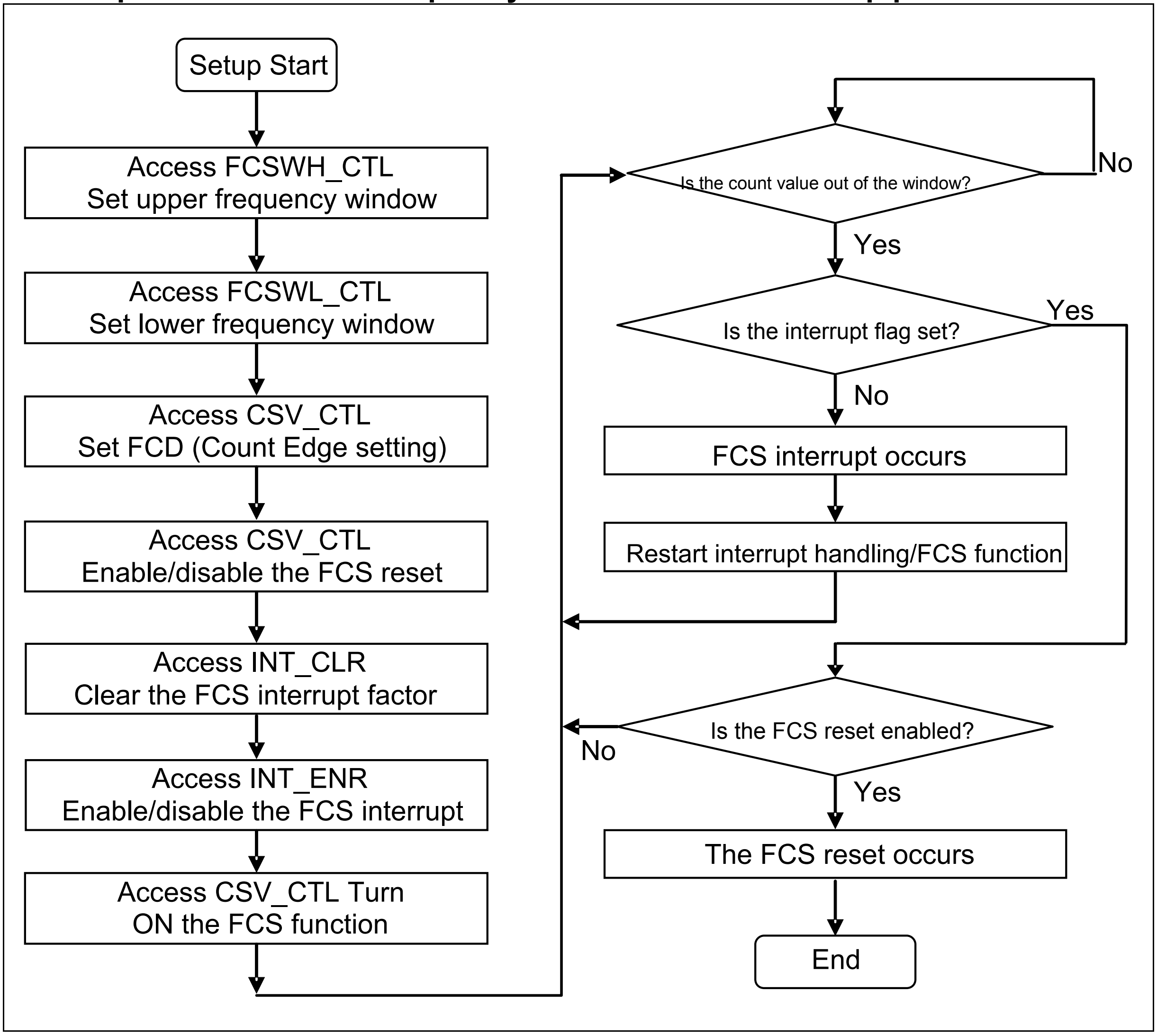
■ Example of clock failure detection function setup procedure


Setup Start
Enable main and sub clock oscillators
Oscillation stabilization wait time of main and sub clocks end
Clock failure detection operation of main and sub clocks starts
Failure Detected?
No
Stop Monitoring?
No
Yes
Yes
The CSV reset occurs
Access the CSV_CTL register Disable the enable bit
End

<Notes>

- When 32 kHz oscillation clock control linkage bit of VBAT register Sub oscillation control register (WTOSCCNT.SOSCNTL) is changed from “1” to “0”, rewrite a register value after the sub clock oscillation stabilization wait completion.
- To operate only VBAT domain with turning off the power on CHIP side, set WTOSCCNT.SOSCNTL=”0” and then turn off the power on CHIP side. Moreover, after the power on CHIP side is turned off, sub clock supervisor function does not operate.
- For details on VBAT, see Chapter "VBAT Domain".

■ Example of Anomalous frequency detection function setup procedure


Setup Start
Access FCSWH_CTL
Set upper frequency window
Access FCSWL_CTL
Set lower frequency window
Access CSV_CTL
Set FCD (Count Edge setting)
Access CSV_CTL
Enable/disable the FCS reset
Access INT_CLR
Clear the FCS interrupt factor
Access INT_ENR
Enable/disable the FCS interrupt
Access CSV_CTL Turn
ON the FCS function
Is the count value out of the window?
No
Yes
Is the interrupt flag set?
Yes
No
FCS interrupt occurs
Restart interrupt handling/FCS function
Is the FCS reset enabled?
No
Yes
The FCS reset occurs
End

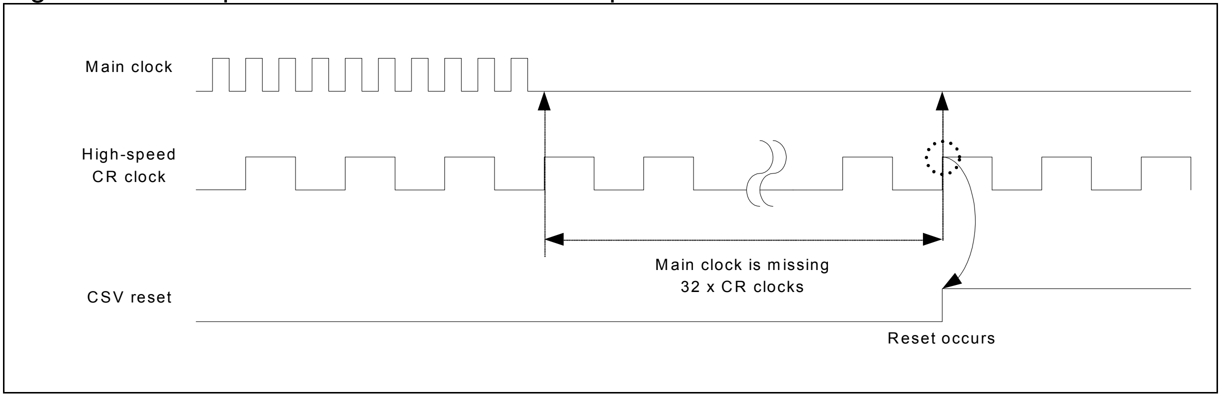
5. Operation Examples

This section explains examples of clock supervisor operations.

■ Clock failure detection

Figure 5-1 provides an example of clock failure detection operation.

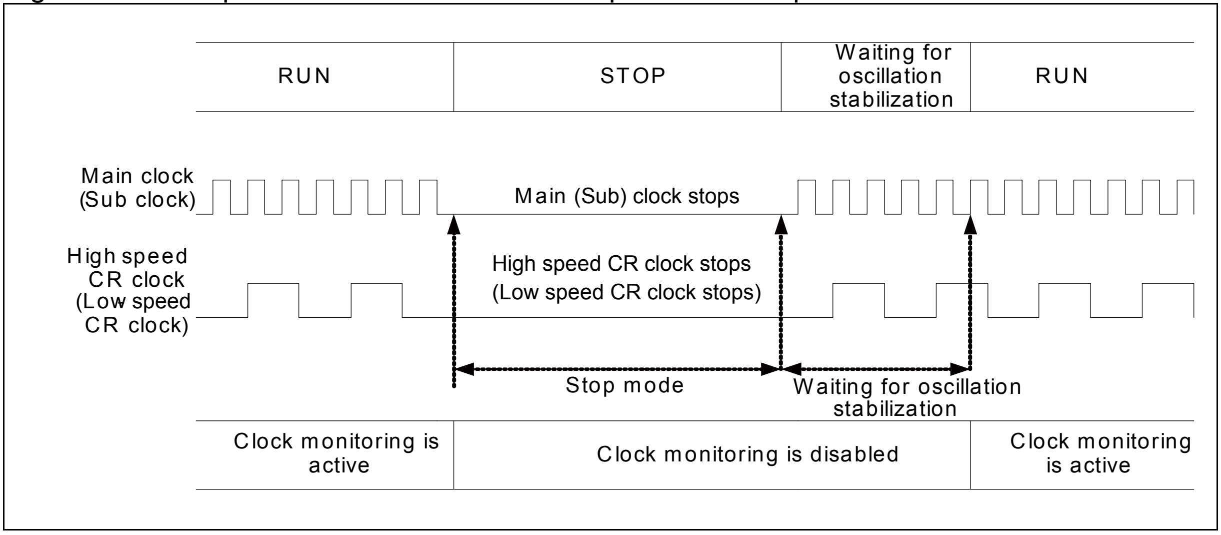
Figure 5-1 Example of clock failure detection operation



1. The main clock stops due to failure.
2. The function counts up clocks using the high-speed CR clock.
3. If the main clock keeps stopping during 32 clocks of high-speed CR, the function determines that the clock has failed and issues the CSV reset.
   Note: In case of the sub clock monitoring, the function determines that the sub clock has failed if it keeps stopping during 32 clocks of low-speed CR.

Figure 5-2 provides an example of the clock failure detection operation in stop mode.

Figure 5-2 Example of clock failure detection operation in stop mode

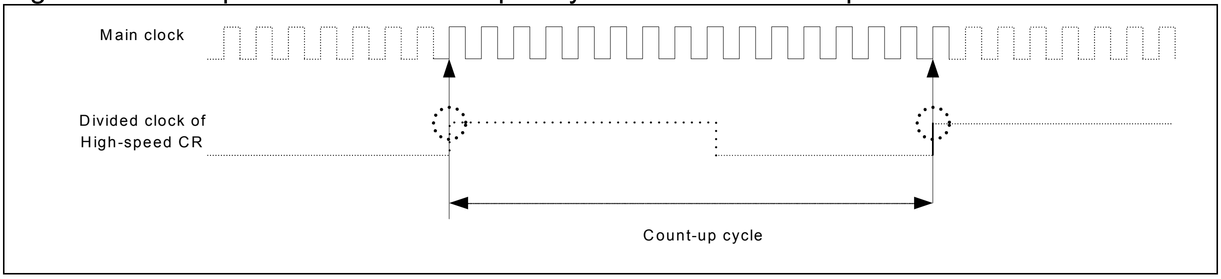

RUN
STOP
Waiting for oscillation stabilization
RUN
Main clock (Sub clock)
Main (Sub) clock stops
High speed CR clock (Low speed CR clock)
High speed CR clock stops
(Low speed CR clock stops)
Stop mode
Waiting for oscillation stabilization
Clock monitoring is active
Clock monitoring is disabled
Clock monitoring is active

1. In stop mode, the main clock and high-speed CR clock stop.
   Meanwhile, the clock monitoring function also stops.
2. Upon the release of stop mode, oscillation of main clock and high-speed CR clock restart, waiting for oscillation stabilization. Meanwhile, the clock monitoring function keeps stopping.
3. When the oscillation stabilization wait time ends, the clock monitoring restarts.

■ Anomalous frequency detection

Figure 5-3 provides an example of anomalous frequency detection function operation.

Figure 5-3 Example of anomalous frequency detection function operation



1. This function detects rising edges of the divided clock of high-speed CR.
2. After detecting edges, it counts up clocks using the main clock.
3. It keeps counting up until it detects the next rising edge of the divided clock of high-speed CR.
4. Let "α" be the count value with the main clock.
   Also let B denote the lower window value, and A the upper window value. Compare the count value α with those window values and if expression
   $B \leq \alpha \leq A$
   holds true, then the frequency is considered to be normal.
   If the count value α is out of the range, i.e., either
   $\alpha < B$, or $A < \alpha$
   is true, then the frequency is considered to be anomalous, and an interrupt occurs.
   If the interrupt flag has not been cleared after the interrupt and an anomalous frequency is detected again, then the function issues a reset depending on the setting.

■ Example of anomalous frequency detection function window setting

The anomalous frequency detection counts up between edges of the divided clock of high-speed CR. The measurement interval is also affected by the accuracy of CR. When you configure the window register value, therefore, the CR accuracy must be considered for the value.
For frequency accuracy of the CR oscillator, check the relevant "Data Sheet".

● Calculation method

The count value range of anomalous frequency detection must be added the CR accuracy, then, the window register value is set. The count range expression must be used as follows.

$$\text{Count value} = \left(\frac{1}{\text{Frequency of divided clock of CR} \;\text{x}\; \left(1 \pm \frac{\text{CR accuracy}}{100}\right)} \right) \text{x} \;\text{Frequency of main clock}$$

The count value by main clock of frequency L [Hz] can be calculated using the divide-by-Y CR oscillator clock of ±Z% accuracy with frequency K [Hz].

Count value A (positive CR frequency accuracy) = 1/[(K/Y) × (1 + Z/100)] × L
Count value B (negative CR frequency accuracy) = 1/[(K/Y) × (1 - Z/100)] × L

Those expressions lead the count value within the range A to B added internal CR accuracy.
Set the value smaller than count value A for the lower limit of the window, and larger than count value B for the upper limit.

The window setting is determined by the value allowed for frequency fluctuation of main oscillation defined by the user.

● Example calculation

The count value by main clock of frequency 4 MHz is calculated using the divide-by-1024 CR oscillator clock of ±5% accuracy with frequency 4 MHz.

Count value A (positive CR frequency accuracy)

$$\text{Count value A} = \left(\frac{1}{\frac{4\times10^{6}}{1024} \times \left(1 + \frac{5}{100}\right)} \right) \times 4\times10^{6} \approx 975$$

Count value B (negative CR frequency accuracy)

$$\text{Count value B} = \left(\frac{1}{\frac{4\times10^{6}}{1024} \times \left(1 - \frac{5}{100}\right)} \right) \times 4\times10^{6} \approx 1078$$

Those expressions yield the count value within the range 975 to 1078 including the high-speed CR error.
If the window setting value is 5%, window setting value is as follows.

Window lower limit = 975 × 0.95(-5%) = 926.25 ≈ 3.43 MHz
Window upper limit = 1078 × 1.05(+5%) = 1131.9 ≈ 4.64 MHz

Thus, you can recognize that a main clock frequency out of the 3.4 MHz to 4.6 MHz range is anomalous.
Table 5-1 provides an example of the window settings.

Table 5-1 Example of window settings

Divided clock of High-speed CR	Main clock	High-speed CR error	Count value including high-speed CR error	Lower limit of window set value	Upper limit of window set value
Divide-by-1024 clocks of CR:4 MHz	4 MHz	±5%	975 (≈ 3.61 MHz) - 1078 (≈ 4.42 MHz)	926 (≈ 3.43 MHz)	1131 (≈ 4.64 MHz)

6. Registers

This section explains the register list of the clock supervisor functions.

■ Register list

Table 6-1 shows the register list.

Table 6-1 Register list

Abbreviation	Register name	Reference
CSV_CTL	CSV control register	6.1
CSV_STR	CSV status register	6.2
FCSWH_CTL	Frequency detection window setting register (Upper)	6.3
FCSWL_CTL	Frequency detection window setting register (Lower)	6.4
FCSWD_CTL	Frequency detection counter register	6.5

6.1. CSV control register (CSV_CTL)

The CSV_CTL register configures the control of CSV function.

■ Register configuration

bit	15	14	13	12	11	10	9	8
Field	Reserved	FCD			Reserved		FCSRE	FCSDE
Attribute	-	R/W			-		R/W	R/W
Initial value	-	111			-		0	0

bit	7	6	5	4	3	2	1	0
Field	Reserved						SCSVE	MCSVE
Attribute	-						R/W	R/W
Initial value	-						1	1

■ Register functions

[bit15] Reserved: Reserved bit

"0" is read from this bit.
Set this bit to "0" when writing.

[bit14:12] FCD: FCS count cycle setting bits

bit14:12	Description
When 000 is written	Setting is prohibited
When 001 is written	
When 010 is written	
When 011 is written	
When 100 is written	
When 101 is written	1/256 frequency of high-speed CR oscillation
When 110 is written	1/512 frequency of high-speed CR oscillation
When 111 is written	1/1024 frequency of high-speed CR oscillation [Initial value]
When read	The register value is read.

[bit11:10] Reserved: Reserved bits

"0b00" is read from these bits.
Set these bits to "0b00" when writing.

[bit9] FCSRE: FCS reset output enable bit

bit	Description
When 0 is written	The FCS reset is disabled [Initial value]
When 1 is written	The FCS reset is enabled
When read	The register value is read.

[bit8] FCSDE: FCS function enable bit

bit	Description
When 0 is written	The FCS function is disabled [Initial value]
When 1 is written	The FCS function is enabled.
When read	The register value is read.

[bit7:2] Reserved: Reserved bits
"0b000000" is read from these bits.
Set these bits to "0b000000" when writing.

[bit1] SCSVE: Sub CSV function enable bit

bit	Description
When 0 is written	The sub CSV function is disabled
When 1 is written	The sub CSV function is enabled. [Initial value]
When read	The register value is read.

[bit0] MCSVE: Main CSV function enable bit

bit	Description
When 0 is written	The main CSV function is disabled
When 1 is written	The main CSV function is enabled. [Initial value]
When read	The register value is read.

<Note>

This register is not initialized by software reset.

To enable sub clock supervisor function, set sub clock oscillation enable setting of system clock mode control register (SCM_CTL.SOSCE) to “1” and wait until the sub clock oscillation stabilization bit of system clock mode control register (SCM_STR.SORDY) becomes “1” by stabilized.
The sub clock supervisor function does not operate only by enabling sub clock oscillation of VBAT RTC (WTOSCCNT.SOSCEX=0).

The following setting combination is prohibited:

- 32 kHz oscillation stop disable bit of VBAT RTC sub clock oscillation control register (WTOSCCNT.SOSCNTL) is “0”.
- 32 kHz oscillation enable bit of VBAT RTC sub clock oscillation control register (WTOSCCNT.SOSCEX) is “1”.
- Sub clock oscillation enable setting bit of system clock mode control register (SCM_CTL.SOSCE) is “1”.
- Sub CSV function enable bit of CSV control register (CSV_CTL.SCSVE) = “1”.

For details on VBAT RTC, see "VBAT Domain".

6.2. CSV status register (CSV_STR)

The CSV_STR register indicates the status of CSV function.

■ Register configuration

bit	7	6	5	4	3	2	1	0
Field	Reserved						SCMF	MCMF
Attribute	-						R	R
Initial value	-						0	0

■ Register functions

[bit7:2] Reserved: Reserved bits

"0b000000" is read from these bits.
Set these bits to "0b000000" when writing.

[bit1] SCMF : Sub clock failure detection flag

bit	Description
When written	No effect
When 0 is read	No sub clock failure has been detected. [Initial value]
When 1 is read	A sub clock failure has been detected.

[bit0] MCMF : Main clock failure detection flag

bit	Description
When written	No effect
When 0 is read	No main clock failure has been detected. [Initial value]
When 1 is read	A main clock failure has been detected.

<Note>

This register is cleared when being read.

6.3. Frequency detection window setting register (Upper) (FCSWH_CTL)

The FCSWH_CTL register configures the frequency detection window setting register (Upper).

■ Register configuration

bit	15 … 0
Field	FWH
Attribute	R/W
Initial value	0xFFFF

■ Register functions

[bit15:0] FWH: Frequency detection window setting bits (Upper)

bit15:0	Description
When written	Any value can be written to these bits.
When read	The register value is read.

<Notes>

- · Set a value larger than the value set in FCSWL_CTL (Frequency detection window setting register (Lower)).
- · This register is not initialized by software reset.

6.4. Frequency detection window setting register (Lower) (FCSWL_CTL)

The FCSWL_CTL register configures the frequency detection window setting register (Lower).

■ Register configuration

bit	15 … 0
Field	FWL
Attribute	R/W
Initial value	0x0000

■ Register functions

[bit15:0] FWL: Frequency detection window setting bits (Lower)

bit15:0	Description
When written	Any value can be written to these bits.
When read	The register value is read.

<Notes>

- Set a value smaller than the value set in FCSWH_CTL (Frequency detection window setting register (Upper)).
- This register is not initialized by software reset.

6.5. Frequency detection counter register (FCSWD_CTL)

The FCSWD_CTL register indicates the counter value of frequency detection using the main clock.

■ Register configuration

bit	15 ... 0
Field	FWD
Attribute	R
Initial value	0x0000

■ Register functions

[bit15:0] FWD: Frequency detection count data

bit15:0	Description
When written	No effect on operation
When read	The count value is read.

<Notes>

- This register retains the count value when detecting an error.
- This register is not initialized by software reset.

7. Usage Precautions

This section explains the precautions for using the clock supervisor functions.

- For details on enabling and clearing the frequency detection interrupt sources, see Chapter "Clock".

- For details on clock failure detection and anomalous frequency detection reset sources, see Chapter "Resets".

- Operation after the occurrence of a reset
  After the occurrence of a reset triggered by clock failure detection, clock mode returns to high-speed CR. Do not select the faulty clock again.

- The high-speed CR clock for use of the frequency detection
  The frequency failure detection is affected by the frequency accuracy of high-speed CR itself.
  When you configure frequency window, therefore, the accuracy of high-speed CR must be considered for the window value. Do not trim the high-speed CR clock after the anomalous frequency detection has been enabled.

- The order of the anomalous frequency detection settings before using
  Before enabling FCS (FCSDE=1), specify the count cycle (FCD), reset enable (FCSRE), and frequency window (FWH/FWL) settings.
  If you want to change any of FCD/FCSRE/FWH/FWL after FCS has been enabled, stop the FCS function before changing the setting. Do not change the setting while FCS is enabled.

- The enable settings of the anomalous frequency detection before using
  Depending on the setting of the FCSRE bit in the CSV control register (CSV_CTL), operation during anomalous frequency detection varies. Table 7-1 shows the setting list.

Table 7-1 List of the FCS function and FCSRE bit settings

	FCSRE=0	FCSRE=1
FCSDE=0	Stops FCS function	Stops FCS function
FCSDE=1	Enables FCS function Generates an interrupt upon error detection	Enables FCS function An interrupt occurs upon the first error detection A reset occurs upon the second error detection

- Interrupt settings for the frequency detection and main timer mode
  The internal bus clock stops while the clock is in main timer mode. In this mode, an interrupt does not occur even if an error is detected while FCSRE is set to "0".
  In main timer mode, therefore, do not set FCSRE bit to "0". If FCSRE bit is set to "1", a reset occurs.

- The settings for CSV OFF and external reset.
  When CSV function is set to OFF, the CSV reset is not generated even if the clock failure occurs, and moreover, the external reset (INITX) is not accepted. So, it is recommended not to turn OFF the CSV function, if you do not have special reason.

CHAPTER: Resets

This chapter explains the function and operation of the resets.

CODE: 9BFRESET_FM4-E01.0

1. Overview

This family has the following reset factors and issues a reset to initialize a device upon accepting a reset factor.

- Power-on reset
- INITX pin input
- External power supply/low-voltage detection reset
- Software watchdog reset
- Hardware watchdog reset
- Clock failure detection reset
- Anomalous frequency detection reset
- Software reset
- TRSTX pin input
- Deep standby transition reset

VBAT domain cannot be initialized with reset factors in this section.

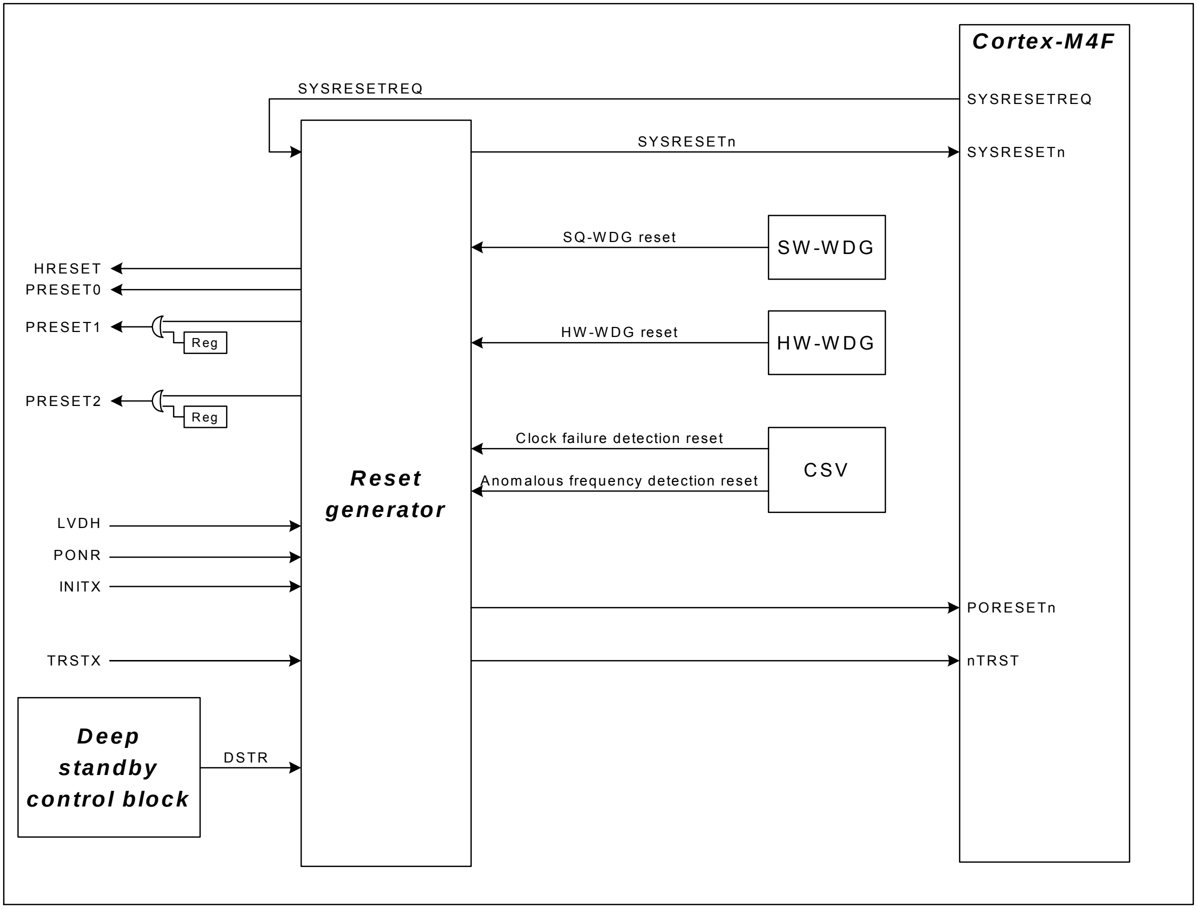
For the reset of VBAT domain, see Chapter "VBAT Domain".

2. Configuration

This section explains configuration of reset circuit.

■ Block Diagram of Resets

Figure 2-1 Block Diagram of Resets



PONR	:	Power-on reset
INITX	:	INITX pin input reset
LVDH	:	Low-voltage detection reset
TRSTX	:	TRSTX pin input reset
HRESET	:	AHB bus reset (a bus reset issued by all reset factors)
PRESET0, 1, 2	:	APB0, APB1, APB2 bus resets (bus resets issued by all reset factors)
SW-WDG reset	:	Software watchdog reset
HW-WDG reset	:	Hardware watchdog reset
PORESETn	:	Power-on reset that is input to Cortex-M4F
SYSRESETn	:	System reset that is input to Cortex-M4F
SYSRESETREQ	:	"SYSRESETREQ bit" signal of Cortex-M4F internal reset control register
nTRST	:	SWJ-DP reset
DSTR	:	Deep standby transition reset

3. Explanation of Operations

This section explains the operations of the resets of this family.

3.1. Reset Factors

This section explains reset factors.

■ Power-On Reset (PONR)

A reset that is generated at power-up.

Generated by	This signal is generated by detecting a rising edge of the power supply.
Cleared by	This signal is automatically cleared after issuing a reset.
Initialization target	Initializes all register settings and hardware.
Flag	bit0 (PONR) of reset factor register (RST_STR) = 1

■ INITX Pin Input Reset (INITX)

A reset that is externally input from a device.

Generated by	This signal is generated by inputting a low level to INITX pin.
Cleared by	This signal is cleared by inputting a high level to INITX pin.
Initialization target	Initializes all register settings and hardware except the debug circuit, deep standby control block, and RTC (some registers). Note: The following registers are not initialized. · Reset factor register (RST_STR) · bit15 to bit8 of low-voltage detection voltage setting register · Deep standby return factor registers 1 and 2 (WRFSR, WIFSR) · Deep Standby RAM Retention Register (DSRAMR) · Backup registers from 01 to 16 (BUR01 to BUR16)
Flag	bit1 (INITX) of reset factor register (RST_STR) = 1

* The content of the on-chip SRAM is retained if a reset is asynchronously input from the INITX pin.

■ Low-voltage Detection Reset, External Voltage Monitoring (LVDH)

A reset that is input from a low-voltage detection circuit when a decrease in the external voltage is detected.

Generated by	This signal is generated when an external voltage is lowered than a specified level.
Cleared by	This signal is cleared when an external voltage is more than a specified level.
Initialization target	Initializes all register settings and hardware.
Flag	bit0 (PONR) of reset factor register (RST_STR) = 1

■ Software Watchdog Reset (SWDGR)

A reset that is input from the software watchdog timer.

Generated by	This signal is generated when the software watchdog timer underflows.
Cleared by	This signal is automatically cleared after issuing a reset.
Initialization target	Initializes all register settings and hardware except the debug circuit and hardware watchdog timer (including control registers) and deep standby control block. Note: The following registers are not initialized. · Reset factor register (RST_STR) · bit15 to bit8 of low-voltage detection voltage setting register (LVD_CTL) · Deep standby return factor registers 1 and 2 (WRFSR, WIFSR) · Deep Standby RAM Retention Register (DSRAMR) · Backup registers from 01 to 16 (BUR01 to BUR16)
Flag	bit4 (SWDT) of reset factor register (RST_STR)= 1

■ Hardware Watchdog Reset (HWDGR)

A reset that is input from the hardware watchdog timer.

Generated by	This signal is generated when the hardware watchdog timer underflows.
Cleared by	This signal is automatically cleared after issuing a reset.
Initialization target	Initializes all register settings and hardware except the debug circuit, deep standby control block, and RTC (some registers). Note: The following registers are not initialized. · Reset factor register (RST_STR) · bit15 to bit8 of low-voltage detection voltage setting register (LVD_CTL) · Deep standby return factor registers 1 and 2 (WRFSR, WIFSR) · Deep Standby RAM Retention Register (DSRAMR) · Backup registers from 01 to 16 (BUR01 to BUR16)
Flag	bit5 (HWDT) of reset factor register (RST_STR) = 1

■ Clock Failure Detection Reset (CSVR)

A reset that is input when the main or sub crystal oscillator being monitored fails.

Generated by	This signal is generated when a clock failure is detected in the main or sub crystal oscillator.
Cleared by	This signal is automatically cleared after issuing a reset.
Initialization target	Initializes all register settings and hardware except the debug circuit , clock failure detection circuit (some registers), deep standby control block, and RTC (some registers). Note: The following registers are not initialized. · Reset factor register (RST_STR) · bit15 to bit8 of low-voltage detection voltage setting register (LVD_CTL) · Deep standby return factor registers 1 and 2 (WRFSR, WIFSR) · Deep Standby RAM Retention Register (DSRAMR) · Backup registers from 01 to 16 (BUR01 to BUR16)
Flag	bit6 (CSVR) of reset factor register (RST_STR) = 1 bit1 (SCMF) or bit0 (MCMF) of CSV status register (CSV_STR) = 1 Note: For details on the CSV_STR, see Chapter "Clock supervisor".

■ Anomalous Frequency Detection Reset (FCSR)

A reset that is input when an anomalous frequency is detected in the main crystal oscillator.

Generated by	This signal is generated when the frequency of the main crystal oscillator is outside of any given setting.
Cleared by	This signal is automatically cleared after issuing a reset.
Initialization target	Initializes all register settings and hardware except the debug circuit and the anomalous frequency detection (some registers) and deep standby control block. Note: The following registers are not initialized. · Reset factor register (RST_STR) · bit15 to bit8 of low-voltage detection voltage setting register (LVD_CTL) · Deep standby return factor registers 1 and 2 (WRFSR, WIFSR) · Deep Standby RAM Retention Register (DSRAMR) · Backup registers from 01 to 16 (BUR01 to BUR16)
Flag	bit7 (FCSR) of reset factor register (RST_STR) = 1

■ Software Reset (SRST)

A reset that is generated when an access to the reset control register occurs.

Generated by	This signal is generated by a write to the Cortex-M4 internal reset control register (SYSRESETREQ bit).
Cleared by	This signal is automatically cleared after issuing a reset.
Initialization target	Initializes all register settings and hardware except the following: Functions and registers that are not initialized by a software reset · Debug circuit · Deep standby control block · Some registers of RTC · Some registers related to clock control (Peripheral clock stop register can be initialized.) · Part of registers that control software and hardware watchdog timers · Part of registers in the clock failure detection circuit · Part of registers that detect an anomalous frequency · Part of registers for CR trimming · Reset factor register (RST_STR) · bit15 to bit8 of low-voltage detection voltage setting register (LVD_CTL) · RTC mode control register (PMD_CTL) · Deep standby return factor registers 1 and 2 (WRFSR, WIFSR) · Deep Standby RAM Retention Register (DSRAMR) · Backup registers from 01 to 16 (BUR01 to BUR16)
Flag	bit8 (SRST) of reset factor register (RST_STR) = 1

■ Deep standby transition reset (DSTR)

This reset occurs when transiting to deep standby mode.

Generated by	This signal is generated by transiting to deep standby mode
Cleared by	This signal is cleared by returning from deep standby mode
Initialization target	Initializes all register settings and hardware except the following: Functions and registers that are not initialized by a deep standby transition reset. · Deep standby control block · Some registers of RTC · Some registers of GPIO · Low-voltage detection circuit register · RTC mode control register (PMD_CTL) · Deep standby return factor registers 1 and 2 (WRFSR, WIFSR) · Deep standby return enable register (WIER) · WKUP pin input level register (WILVR) · Deep Standby RAM Retention Register (DSRAMR) · Backup registers from 01 to 16 (BUR01 to BUR16)
Flag	The bit of either deep standby return factor register 1 or 2 (WRFSR, WIFSR) is "1". Note: The bit that becomes "1" differs by return factors.

<Notes>

- For Cortex-M4 internal reset control register (SYSRESETREQ) that controls the software reset, see "Cortex-M4 Devices Generic User Guide ".
- The reset factor register that can determine the occurrence of each reset factor is initialized only by power-on reset.

3.2. Resetting Inside the Device

This section explains the internal reset signals of this device.

Resets that are internally connected to the device are divided into resets that are input to the Cortex-M4 core and resets that are input to peripheral circuits.

3.2.1. Resets to Cortex-M4

The device has three reset inputs to the Cortex-M4: PORESETn, SYSRESETn, and nTRST. The following provides reset factors for these three reset inputs.

■ Power-on reset PORESETn

Reset factors	· Power-on reset (PONR) · Low-voltage detection reset (LVDH) · Deep standby transition reset (DSTR)

■ System reset SYSRESETn

Reset factors	· Power-on reset (PONR) · Low-voltage detection reset (LVDH) · INITX pin input (INITX) · Software watchdog reset (SWDGR) · Hardware watchdog reset (HWDGR) · Clock Failure Detection reset (CSVR) · Anomalous frequency detection reset (FCSR) · Software reset (SRST) · Deep standby transition reset (DSTR)

■ SWJ-DP Reset nTRST

Reset factors	· Power-on reset (PONR) · Low-voltage detection reset (LVDH) · TRSTX pin input (TRSTX) · Deep standby transition reset (DSTR)

3.2.2. Resets to Peripheral Circuit

The bus resets (HRESET, PRESET0, PRESET1, and PRESET2) that are input to the peripheral circuit are basically generated by all reset factors. Resetting of PRESET1 and PRESET2 can be controlled by register settings.

The following provides reset factors for the bus resets.

■ Resets to Peripheral Circuit

● HRESET and PRESET0

Reset factors	· Power-on reset (PONR) · Low-voltage detection reset (LVDH) · INITX pin input (INITX) · Software watchdog reset (SWDGR) · Hardware watchdog reset (HWDGR) · Clock Failure Detection reset (CSVR) · Anomalous frequency detection reset (FCSR) · Software reset (SRST) · Deep standby transition reset (DSTR)

● PRESET1 and PRESET2

Reset factors	· Power-on reset (PONR) · Low-voltage detection reset (LVDH) · INITX pin input (INITX) · Software watchdog reset (SWDGR) · Hardware watchdog reset (HWDGR) · Clock Failure Detection reset (CSVR) · Anomalous frequency detection reset (FCSR) · Software reset (SRST) · APB bus resets (APBC1_PSR and APBC2_PSR) · Deep standby transition reset (DSTR)

<Notes>

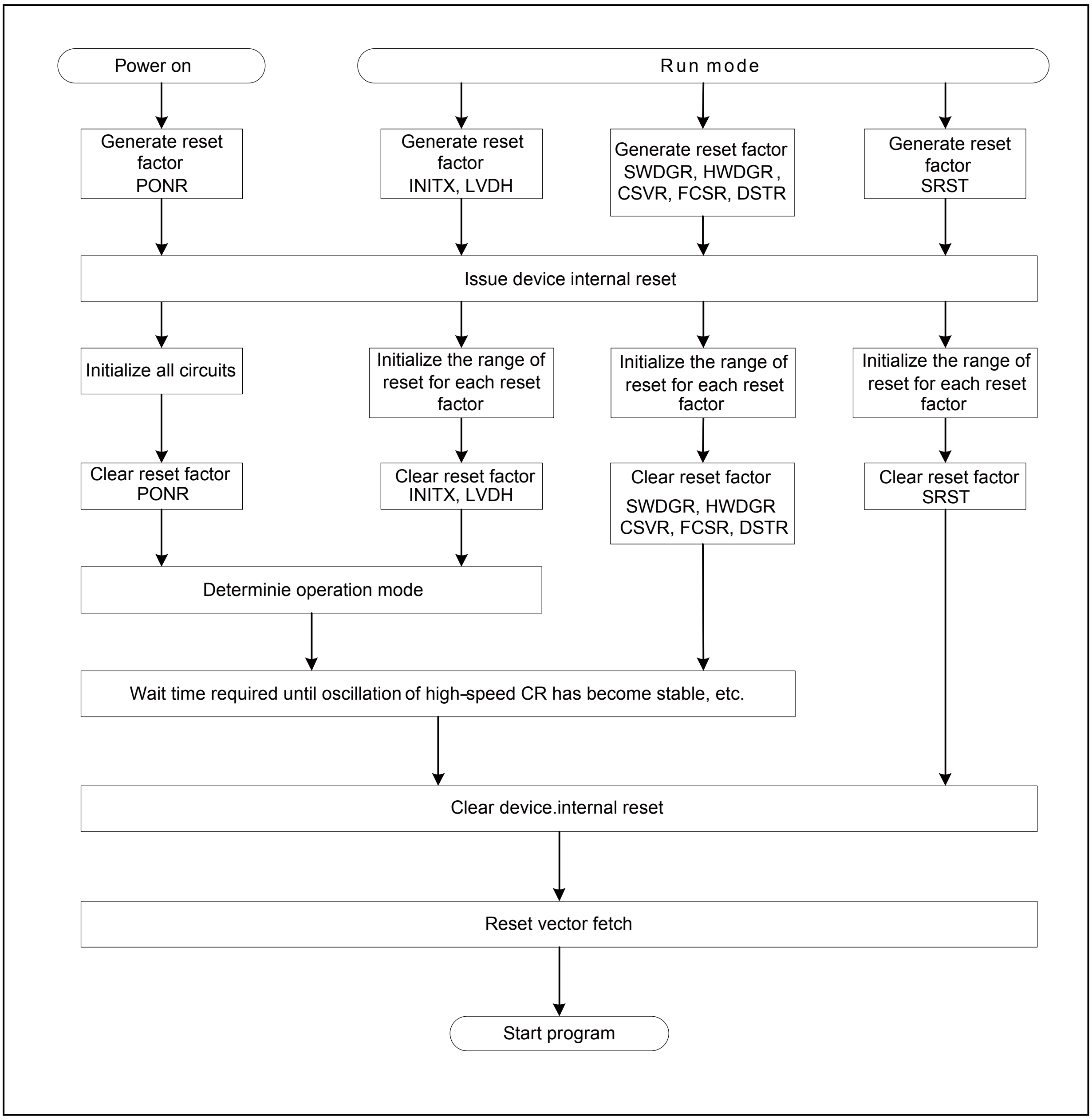
- The peripheral circuit is essentially initialized with all reset factors. Depending on the specifications of the peripheral circuit, there are registers that are initialized only with specific causes. For the initialization conditions for registers, see the initialization conditions for the registers described in the relevant chapter.
- For details on APB bus resets (APBC1_PSR and APBC2_PSR), see Chapter "Clock".

3.3. Reset Sequence

This family initiates the program and hardware operations starting with the initial state when a reset factor is cleared.
This family of operations starting with the reset and ending with the initiation of the operations is called a reset sequence.
The following explains a reset sequence.

■ State Transition Diagram for Resets

The following diagram shows a transition of reset states. The detailed operations are given in the following sections "3.4 Operations After Resets are Cleared".


Power on
Run mode
Generate reset factor PONR
Generate reset factor INITX, LVDH
Generate reset factor SWDGR, HWDGR , CSVR, FCSR, DSTR
Generate reset factor SRST
Issue device internal reset
Initialize all circuits
Initialize the range of reset for each reset factor
Initialize the range of reset for each reset factor
Initialize the range of reset for each reset factor
Clear reset factor PONR
Clear reset factor INITX, LVDH
Clear reset factor SWDGR, HWDGR CSVR, FCSR, DSTR
Clear reset factor SRST
Determinie operation mode
Wait time required until oscillation of high-speed CR has become stable, etc.
Clear device.internal reset
Reset vector fetch
Start program

1. Capturing reset factors
   Reset factors are captured and retained until a reset is issued to the device.

2. Issuing resets
   When a reset is ready to be issued, a device internal reset is issued.

3. Clearing resets
   When a reset factor is cleared, a device internal reset is extended for the amount of time required to clear the reset (for example, a wait time required until oscillation of a high-speed CR has become stable). When the extended period of time has expired, the reset is cleared.

4. Determining operation mode
   The operation mode is determined as PONR, LVDH or INITX is cleared and notified to each piece of the hardware. Any other reset factors do not cause the operation mode to change.

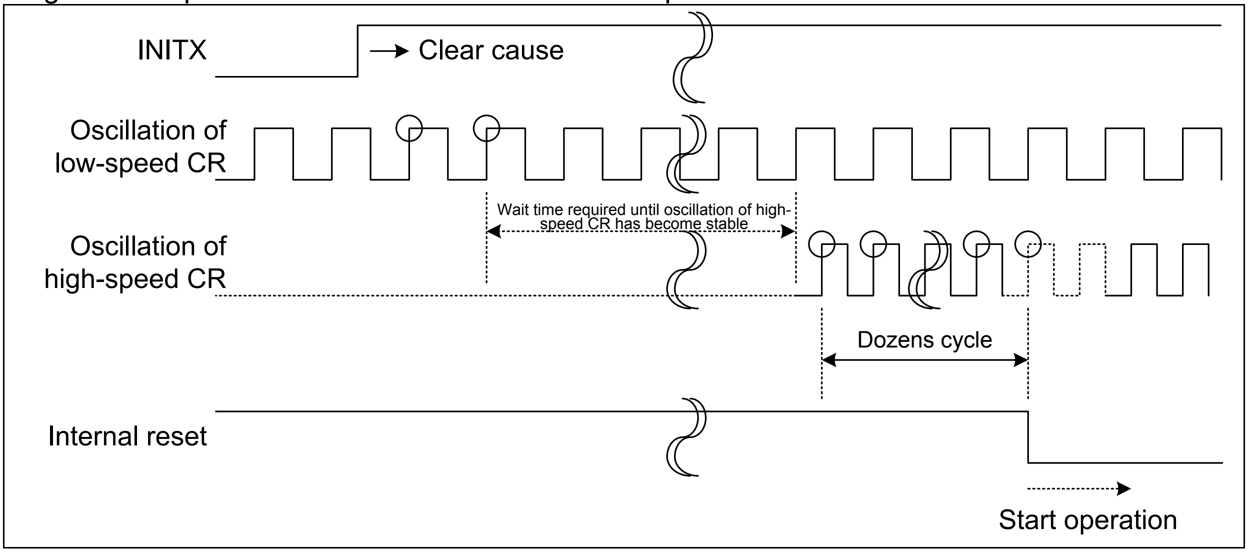
5. Reset vector fetch
   After a device internal reset is cleared, the CPU starts fetching a reset vector. The CPU fetches the obtained reset vector into the program counter and starts programmed operations.

3.4. Operations After Resets are Cleared

■ PONR, LVDH, INITX, HWDGR, SWDGR, CSVR, FCSR, DSTR

Figure 3-1 provides an example of the operation waveform after a cause of INITX pin input reset has been cleared.

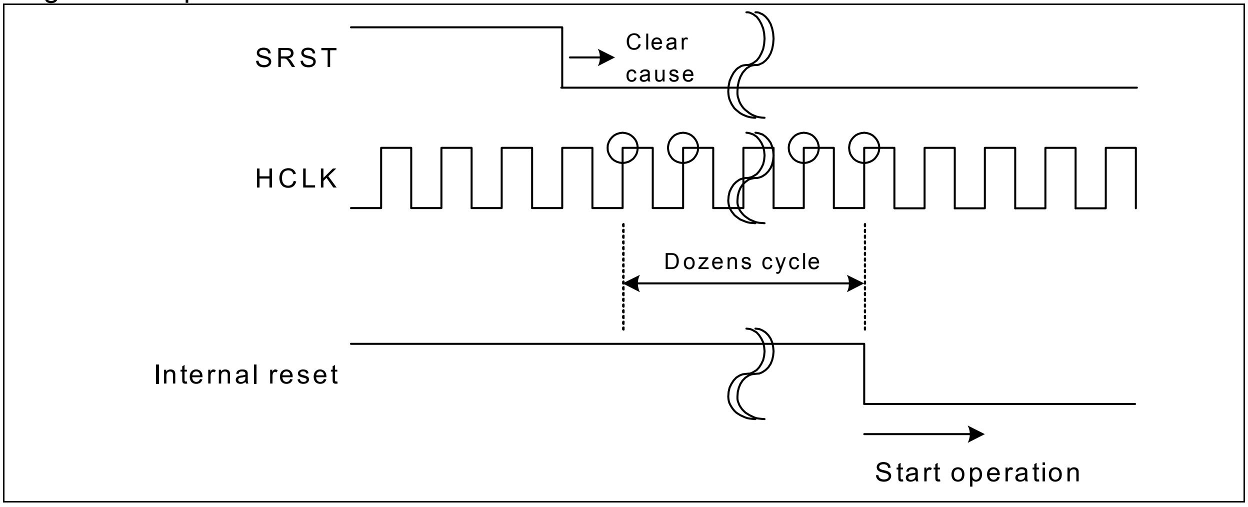
Figure 3-1 Operation Waveform After INITX Pin Input Reset has been Cleared


INITX
Clear cause
Oscillation of low-speed CR
Wait time required until oscillation of high-speed CR has become stable
Oscillation of high-speed CR
Dozens cycle
Internal reset
Start operation

■ SRST

Figure 3-2 shows an operation waveform after a software reset has been cleared.

Figure 3-2 Operation Waveform After a Software Reset has been Cleared



4. Register

This section explains the configuration and functions of the register.

■ Register list

Abbreviation	Register name	Reference
RST_STR	Reset factor register	4.1

4.1. Reset Factor Register (RST_STR: ReSeT STatus Register)

The reset factor register shows causes of resets that have just occurred and initializes values upon power-on.
Reading the register clears all bits.
It stores all reset factors that have been generated until after it has been read upon power-on.

bit	15	14	13	12	11	10	9	8
Field	Reserved							SRST
Attribute	-							R
Initial value	-							0

bit	7	6	5	4	3	2	1	0
Field	FCSR	CSVR	HWDT	SWDT	Reserved		INITX	PONR
Attribute	R	R	R	R	-		R	R
Initial value	0	0	0	0	-		0	1

Note: The initial value is the value upon power-on.

[bit15:9] Reserved: Reserved bits

The read value is undefined.
These bits have no effect when written.

[bit8] SRST: Software reset flag

Indicates a reset that is generated by writing "1" to Cortex-M4 internal reset control register (SYSRESETREQ bit).
When a software reset is generated, SRST is enabled (SRST = 1).

bit	Description
0	A software reset has not been issued.
1	A software reset has been issued.

[bit7] FCSR: Flag for anomalous frequency detection reset

Indicates a reset when an anomalous frequency is detected in the main oscillation.
When the frequency of the main oscillation is outside of a given setting, a reset is issued and FCSR is enabled (FCSR = 1).

bit	Description
0	An anomalous frequency detection reset has not been issued.
1	An anomalous frequency detection reset has been issued.

[bit6] CSVR: Clock failure detection reset flag

Indicates a reset when a failure is detected in the main or sub oscillation.
If a stop is detected, a reset is issued and CSVR is enabled (CSVR = 1).

bit	Description
0	A clock failure detection reset has not been issued.
1	A clock failure detection reset has been issued.

Note: Please refer to Chapter "Clock supervisor" for the method of judging whether the main oscillation or the sub oscillation broke down.

[bit5] HWDT: Hardware watchdog reset flag

Indicates a reset from the hardware watchdog timer.
If the timer underflows, a reset is issued and HWDT is enabled (HWDT = 1).

bit	Description
0	A hardware watchdog reset has not been issued.
1	A hardware watchdog reset has been issued.

[bit4] SWDT: Software watchdog reset flag

Indicates a reset from the software watchdog timer.
If the timer overflows, a reset is issued and SWDT is enabled (SWDT = 1).

bit	Description
0	A software watchdog reset has not been issued.
1	A software watchdog reset has been issued.

[bit3:2] Reserved: Reserved bits

The read value is undefined.
These bits have no effect when written.

[bit1] INITX: INITX pin input reset flag

Indicates a reset that is externally input.
If a reset is externally input, INITX is enabled (INITX = 1).

bit	Description
0	An INITX pin input reset has not been issued.
1	An INITX pin input reset has been issued.

[bit0] PONR: Power-on reset/low-voltage detection reset flag

Indicates a reset at power-on and when a low-voltage is detected.
If a rising edge of power supply or a low-voltage is detected, a reset is issued and PONR is enabled (PONR = 1).

bit	Description
0	A power-on reset or low-voltage detection reset has not been issued.
1	A power-on reset or low-voltage detection reset has been issued.

<Note>

This register is initialized by a power-on reset or low-voltage detection reset. It is not initialized by any other reset factors. Reading the register clears all bits.

CHAPTER: Low-voltage Detection

This chapter explains the functions and operations of the Low-voltage Detection Circuit.

CODE: 9BFLVD_A_FM4_-E01.0

1. Overview

The Low-voltage Detection Circuit monitors the power supply voltage, and generates reset and interrupt signals when the power supply voltage falls below the detection voltage.

■ Overview of Low-voltage Detection Circuit

● Operations of Low-voltage Reset Circuit

- This circuit monitors the power supply voltage (VCC) and generates a reset signal when the power supply voltage falls below the specified voltage.
- This circuit always monitors the power supply voltage.
- This circuit monitors the power supply voltage even in standby modes and deep standby modes.
- This circuit generates a reset signal when the reduction of the power supply voltage is detected in standby modes and deep standby modes.

● Operations of Low-voltage Interrupt Circuit

- This circuit monitors the power supply voltage (VCC) and generates an interrupt signal when the power supply voltage falls below the specified voltage.
- This circuit allows selection of whether to enable or disable operations. The initial value is set to disable.
- This circuit allows specification of the detection voltage.
- This circuit can monitor the power supply voltage even in standby modes and deep standby modes.
- This circuit returns from standby modes and deep standby modes when the reduction of the power supply voltage is detected in those modes

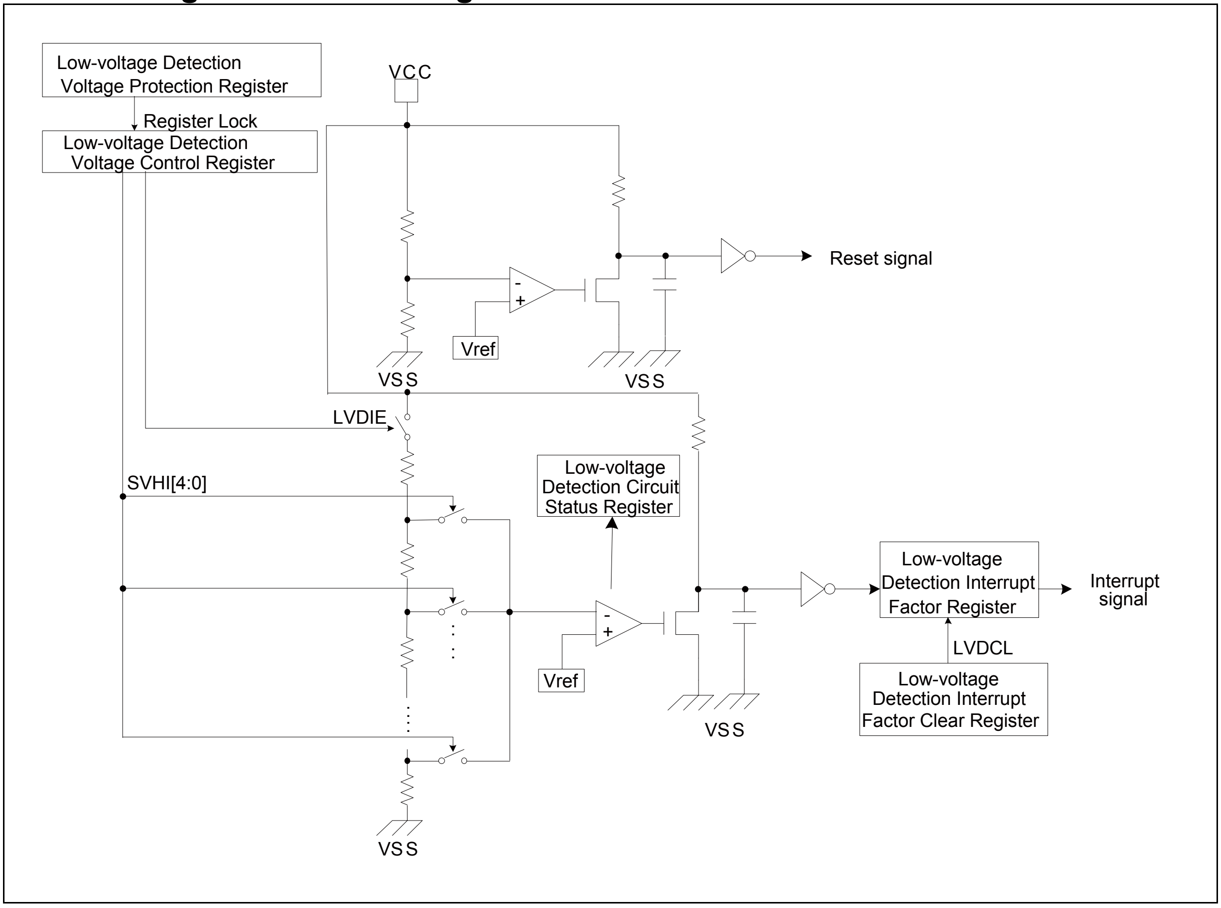
<Notes>

- If a low-voltage detection interrupt is enabled or the detection voltage is specified for a low-voltage detection interrupt, this circuit starts VCC voltage monitoring after the stabilization wait time of the Low-voltage Detection Circuit has lapsed.
  For the stabilization wait time of the Low-voltage Detection Circuit, see "Data Sheet" of the product used.
- This circuit does not conduct monitoring the power supply voltage if PCLK2 is gated by TIMER mode, RTC mode, STOP mode, Deep standby RTC mode, Deep standby STOP mode, or APB2 Prescaler Register (APBC2_PSR) while waiting for the stabilization of the Low-voltage Detection Circuit. After the status flag is read and the stabilization wait time has lapsed, change to the desired mode.
- The Low-voltage Detection Voltage Control Register (LVD_CTL) is write-protected to prevent a writing error. To release write protection mode, write 0x1ACCE553 to the Low-voltage Detection Voltage Protection Register (LVD_RLR).

2. Configuration

This section shows the block diagram of the Low-voltage Detection Circuit.

■ Block diagram of Low-voltage Detection Circuit


Low-voltage Detection Voltage Protection Register
Register Lock
Low-voltage Detection Voltage Control Register
VCC
Reset signal
Vref
VSS
VSS
LVDIE
SVHI[4:0]
Low-voltage Detection Circuit Status Register
Low-voltage Detection Interrupt Factor Register
Interrupt signal
LVDCL
Low-voltage Detection Interrupt Factor Clear Register
Vref
VSS
VSS

● Low-voltage Detection Voltage Control Register (LVD_CTL)

This register controls whether to enable monitoring the power supply voltage for a low-voltage detection interrupt and specifies the detection voltage for a low-voltage detection interrupt.

● Low-voltage Detection Voltage Protection Register (LVD_RLR)

This register write-protects the Low-voltage Detection Voltage Control Register.

● Low-voltage Detection Interrupt Factor Register (LVD_STR)

This register holds a low-voltage detection interrupt factor.

● Low-voltage Detection Interrupt Factor Clear Register (LVD_CLR)

This register clears a low-voltage detection interrupt factor.

● Low-voltage Detection Circuit Status Register (LVD_STR2)

This register checks the operation status of a low-voltage detection interrupt circuit.

■ Pins of Low-voltage Detection Circuit

The following shows the pins used in the Low-voltage Detection Circuit.

- VCC pin
  The Low-voltage Detection Circuit monitors the power supply voltage of this pin.
- VSS pin
  This pin is a GND pin used as a basis to detect the power supply voltage.

3. Explanation of Operations

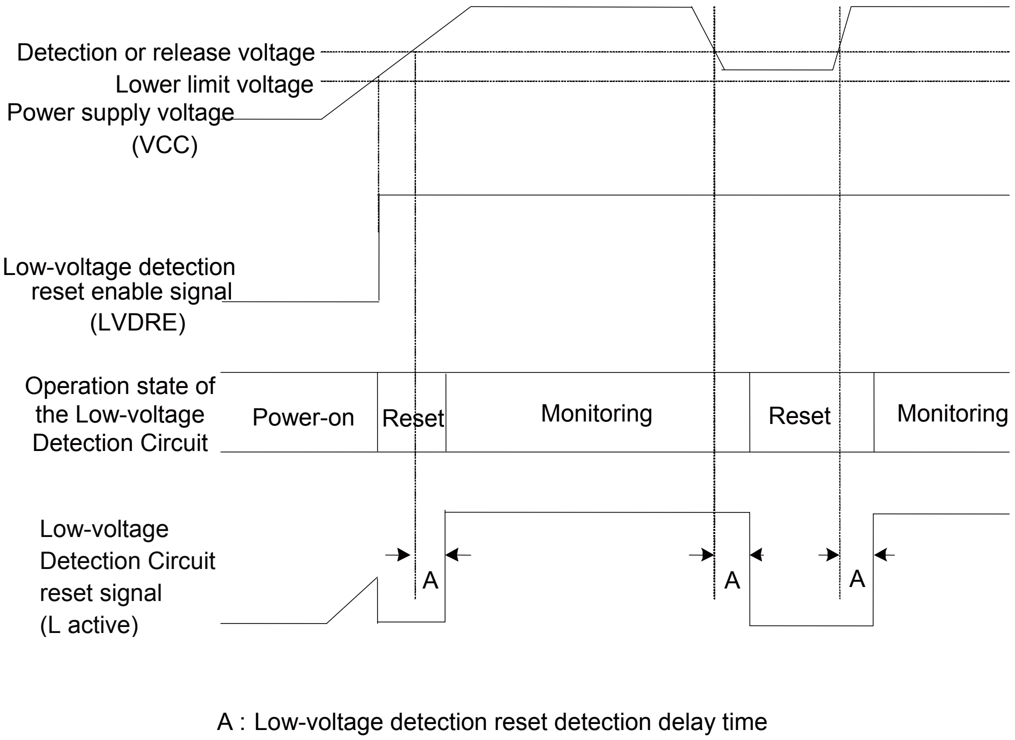
This section explains the operations of the Low-Voltage Detection Reset Circuit and the Low-voltage Detection Interrupt Circuit.

■ Operations of Low-Voltage Detection Reset Circuit

● Operations

The Low-Voltage Detection Reset Circuit always enters a monitoring state after power-on. This circuit generates a reset signal when the power supply voltage (VCC) falls below the detection voltage. A reset is released when the power supply voltage exceeds the release voltage.

This circuit is available in standby modes (SLEEP mode, TIMER mode, RTC mode, STOP mode) and deep standby modes (Deep standby RTC mode, Deep standby STOP mode).



■ Operations of Low-voltage Detection Interrupt Circuit

● Operations

The Low-voltage Detection Interrupt Circuit monitors the power supply voltage (VCC) and generates an interrupt signal when the power supply voltage falls below the specified voltage.

An interrupt request is enabled when "1" is set to the LVDIE bit of the Low-voltage Detection Voltage Control Register. The initial value is set to Not Enable. The interrupt detection voltage can be set by the SVHI bit of the Low-voltage Detection Voltage Control Register (LVD_CTL). When an interrupt request is enabled and the interrupt detection voltage is specified, the status flag LVDIRDY bit of the Low-voltage Detection Circuit Status Register (LVD_STR2) is set to "1" and this circuit starts monitoring the power supply voltage after the stabilization wait time of the Low-voltage Detection Circuit has lapsed.

This circuit is available in standby modes (SLEEP mode, TIMER mode, RTC mode, STOP mode) and deep standby modes (Deep standby RTC mode, Deep standby STOP mode). It is also applicable when the CPU returns from those modes.

● Low-voltage detection interrupt request

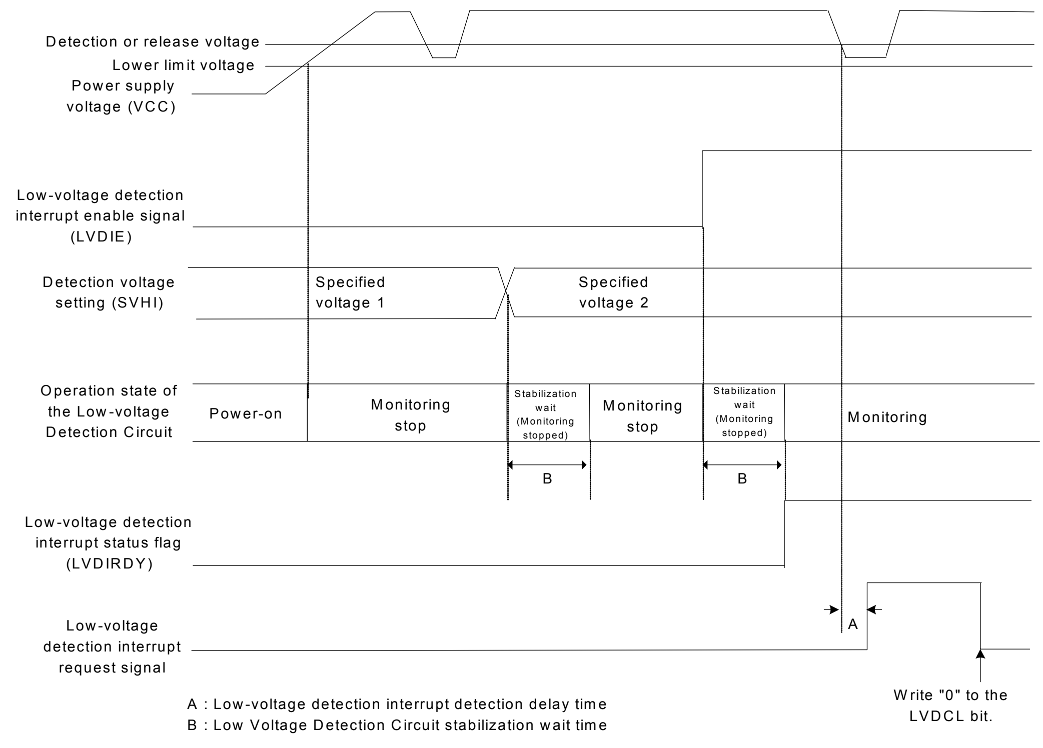
When the power supply voltage (VCC) falls below the specified voltage while a low-voltage detection interrupt is effective, "1" is set to the LVDIR bit of the Low-voltage Detection Interrupt Factor Register (LVD_STR) to generate an interrupt request signal.

An interrupt request can be checked by reading the LVDIR bit.

● Canceling a low-voltage detection interrupt request

To cancel a low-voltage detection interrupt request, write "0" to the LVDCL bit of the Low-voltage Detection Interrupt Factor Clear Register (LVD_CLR). This clears a low-voltage detection interrupt factor and cancels a low-voltage detection interrupt request.

Even when the power supply voltage is below the specified detection voltage, an interrupt request is canceled when "0" is written to the LVDCL bit.


Detection or release voltage
Lower limit voltage
Power supply voltage (VCC)
Low-voltage detection interrupt enable signal (LVDIE)
Detection voltage setting (SVHI)
Specified voltage 1
Specified voltage 2
Operation state of the Low-voltage Detection Circuit
Power-on
Monitoring stop
Stabilization wait (Monitoring stopped)
Monitoring stop
Stabilization wait (Monitoring stopped)
Monitoring
B
B
Low-voltage detection interrupt status flag (LVDIRDY)
A
Low-voltage detection interrupt request signal
Write "0" to the LVDCL bit.

A : Low-voltage detection interrupt detection delay time
B : Low Voltage Detection Circuit stabilization wait time

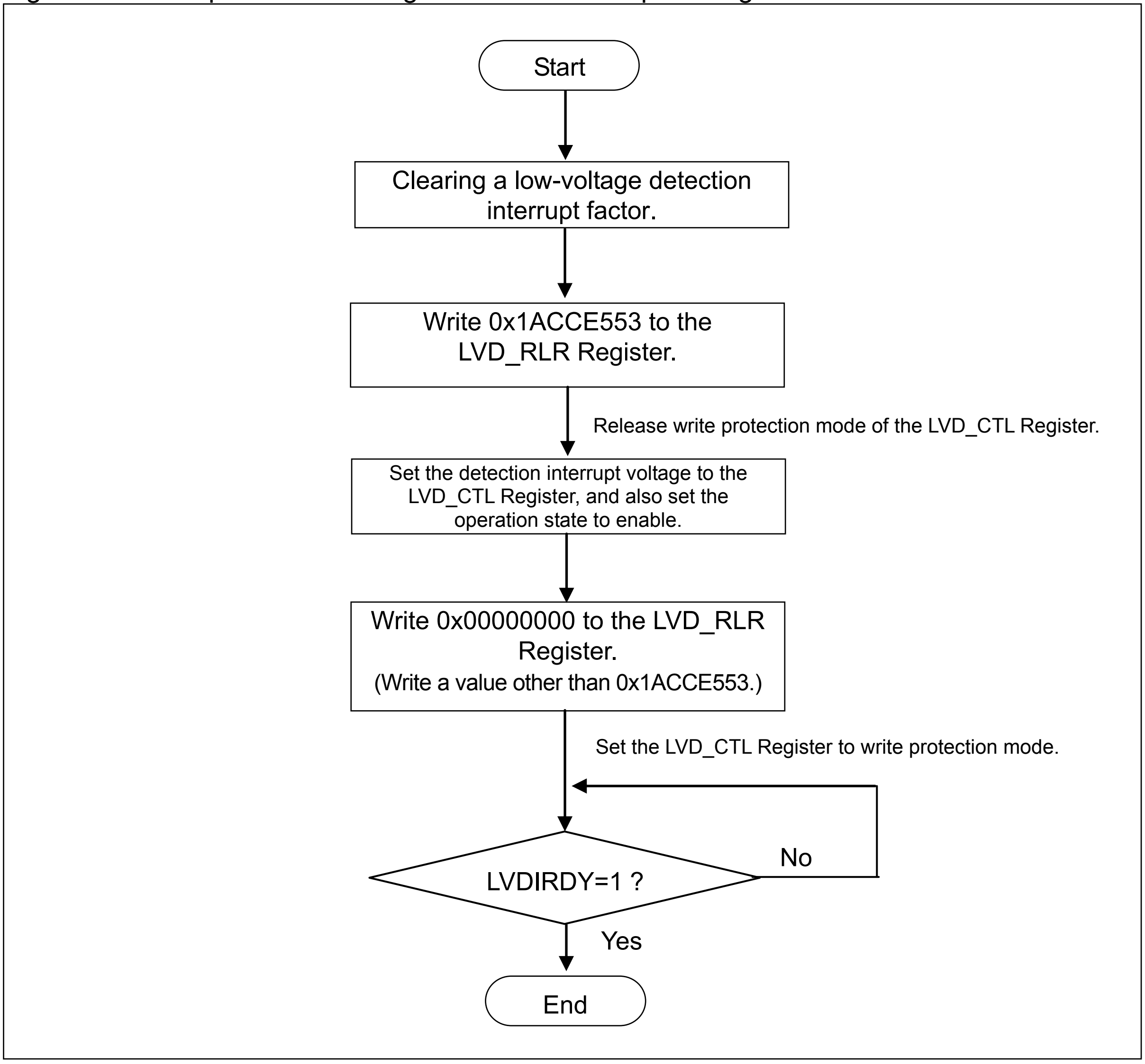
<Note>

This circuit does not conduct monitoring the power supply voltage if PCLK2 is gated by TIMER mode, RTC mode, STOP mode, Deep standby RTC mode, Deep standby STOP mode, or APB2 Prescaler Register (APBC2_PSR) while waiting for the stabilization of the Low-voltage Detection Circuit. After checking that the Low-voltage detection interrupt status flag (LVDIRDY) of Low-voltage Detection Circuit Status Register (LVD_STR2) is set to "1", change to the desired mode.

4. Setup Procedure Examples

This section explains the procedures to set up the Low-voltage Detection Circuit, giving examples.

Figure 4-1 Example of Low-voltage detection interrupt setting


Start
Clearing a low-voltage detection interrupt factor.
Write 0x1ACCE553 to the LVD_RLR Register.
Release write protection mode of the LVD_CTL Register.
Set the detection interrupt voltage to the LVD_CTL Register, and also set the operation state to enable.
Write 0x00000000 to the LVD_RLR Register.
(Write a value other than 0x1ACCE553.)
Set the LVD_CTL Register to write protection mode.
LVDIRDY=1 ?
No
Yes
End

5. Registers

This section explains the configuration and functions of the registers used in the Low-voltage Detection Circuit.

■ List of Low-voltage Detection Circuit Registers

Table 5-1 List of Low-voltage Detection Circuit Registers

Abbreviation	Register name	Reference
LVD_CTL	Low-voltage Detection Voltage Control Register	5.1
LVD_STR	Low-voltage Detection Interrupt Factor Register	5.2
LVD_CLR	Low-voltage Detection Interrupt Factor Clear Register	5.3
LVD_RLR	Low-voltage Detection Voltage Protection Register	5.4
LVD_STR2	Low-voltage Detection Circuit Status Register	5.5

5.1. Low-voltage Detection Voltage Control Register (LVD_CTL)

The Low-voltage Detection Voltage Control Register (LVD_CTL) controls whether to enable monitoring the power supply voltage for a low-voltage detection interrupt and specifies the detection voltage for a low-voltage detection interrupt.

bit	7	6	5	4	3	2	1	0
Field	LVDIE	SVHI					Reserved	
Attribute	R/W	R/W					-	
Initial value	0	00111					-	

[bit7] LVDIE: Low-voltage detection interrupt enable bit

This bit is used to enable monitoring the power supply voltage of a low-voltage detection interrupt. When not enabling monitoring the power supply voltage, the Low-voltage Detection Interrupt Circuit is stopped.

bit	Description
0	Does not enable the generation of a low-voltage detection interrupt. [Initial value]
1	Enables the generation of a low-voltage detection interrupt.

[bit6:2] SVHI: Low-voltage detection interrupt voltage setting bits

These bits are used to specify the detection voltage of a low-voltage detection interrupt.

bit6:2	Description
00111	Set the low-voltage detection interrupt voltage in the vicinity of 2.8V. [Initial value]
0001	Set the low-voltage detection interrupt voltage in the vicinity of 3.0V.
00100	Set the low-voltage detection interrupt voltage in the vicinity of 3.2V.
01100	Set the low-voltage detection interrupt voltage in the vicinity of 3.6V.
01111	Set the low-voltage detection interrupt voltage in the vicinity of 3.7V.
01110	Set the low-voltage detection interrupt voltage in the vicinity of 4.0V.
10000	Set the low-voltage detection interrupt voltage in the vicinity of 4.1V.
11000	Set the low-voltage detection interrupt voltage in the vicinity of 4.2V.
Others	Setting is prohibited.

[bit1:0] Reserved: Reserved bits

The read value is undefined.

These bits have no effect when written.

<Notes>

- The low-voltage detection interrupt enable bit (LVDIE) must be enabled after "0" was written to the LVDCL bit of the Low-voltage Detection Interrupt Clear Register (LVD_CLR) to clear the low-voltage detection interrupt factor bit (LVDIR).
- When the low-voltage detection interrupt enable bit (LVDIE) is not enabled, the Low-voltage Detection Circuit for detecting a low-voltage interrupt is stopped. Therefore, the low-voltage detection interrupt factor bit (LVDIR) is not set.
- The Low-voltage Detection Voltage Control Register (LVD_CTL) is write-protected in the initial state, which makes writing invalid unless write protection mode is released. To write the LVD_CTL Register, set 0x1ACCE553 to the Low-voltage Detection Voltage Protection Register (LVD_RLR) to release write protection mode.
- This register is not initialized by the deep standby transition reset.

5.2. Low-voltage Detection Interrupt Factor Register (LVD_STR)

The Low-voltage Detection Interrupt Factor Register (LVD_STR) holds a low-voltage detection interrupt factor.

bit	7	6	5	4	3	2	1	0
Field	LVDIR	Reserved						
Attribute	R	-						
Initial value	0	-						

[bit7] LVDIR: Low-voltage detection interrupt factor bit

bit	Description
0	A low-voltage detection interrupt request is not detected. [Initial value]
1	A low-voltage detection interrupt request has been detected.

[bit6:0] Reserved: Reserved bits

The read value is undefined.

These bits have no effect when written.

<Note>

This register is not initialized by the deep standby transition reset.

5.3. Low-voltage Detection Interrupt Factor Clear Register (LVD_CLR)

The Low-voltage Detection Interrupt Factor Clear Register (LVD_CLR) clears a low-voltage detection interrupt factor.

bit	7	6	5	4	3	2	1	0
Field	LVDCL	Reserved						
Attribute	R/W	-						
Initial value	1	-						

[bit7] LVDCL: Low-voltage detection interrupt factor clear bit

bit	Description
0	Clears the low-voltage detection interrupt bit (LVDIR) of the Low-voltage Detection Interrupt Factor Register (LVD_STR) to "0".
1	Has no effect when written. [Initial value]

The read value is always "1".

[bit6:0] Reserved: Reserved bits

The read value is undefined.

These bits have no effect when written.

<Note>

This register is not initialized by the deep standby transition reset.

5.4. Low-voltage Detection Voltage Protection Register (LVD_RLR)

The Low-voltage Detection Voltage Protection Register (LVD_RLR) write-protects the Low-voltage Detection Voltage Control Register (LVD_CTL).

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	LVDLCK[31:16]															
Attribute	R/W															
Initial value	0x0000															

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	LVDLCK[15:0]															
Attribute	R/W															
Initial value	0x0001															

[bit31:0] LVDLCK[31:0]: Low-voltage Detection Voltage Control Register protection bits

- When "0xACCE553" is written:
  Writing to the Low-voltage Detection Voltage Control Register(LVD_CTL) is enabled (releases write protection mode).

- When a value other than "0x1ACCE553" is written:
  Writing to the Low-voltage Detection Voltage Control Register(LVD_CTL) is disabled(enables write protection mode).

- When the protection of the Low-voltage Detection Voltage Control Register(LVD_CTL) is released:
  0x00000000 is read.

- When the Low-voltage Detection Voltage Control Register(LVD_CTL) is protected:
  0x00000001 is read.

<Notes>

- The Low-voltage Detection Voltage Control Register (LVD_CTL) is write-protected in the initial state. To write the LVD_CTL Register, set 0x1ACCE553 to the Low-voltage Detection Voltage Protection Register (LVD_RLR) to release write protection mode.
- To enable write protection mode of the LVD_CTL register, set a value other than 0x1ACCE553 to the Low-voltage Detection Voltage Protection Register (LVD_RLR).
- Once write protection mode is released for the LVD_CTL Register, it remains released until a value other than 0x1ACCE553 is written to the LVD_RLR Register.
- This register is not initialized by the deep standby transition reset.

5.5. Low-voltage Detection Circuit Status Register (LVD_STR2)

The Low-voltage Detection Circuit Status Register (LVD_STR2) checks the operation status of a low-voltage detection interrupt.

bit	7	6	5	4	3	2	1	0
Field	LVDIRDY	Reserved						
Attribute	R	-						
Initial value	0	-						

[bit7] LVDIRDY : Low-voltage detection interrupt status flag

bit	Description
0	Stabilization wait state or monitoring stop state [Initial value]
1	Monitoring state

This bit has no effect when written.

[bit6:0] Reserved: Reserved bits

The read value is undefined.

These bits have no effect when written.

<Note>

This register is not initialized by the deep standby transition reset.

CHAPTER: Low Power Consumption Mode

This chapter explains the functions and operations of the low power consumption mode.

CODE: 9BFLPMODE_FM4-E01.0

1. Overview of Low Power Consumption Mode

The system has two types of low power consumption mode, standby mode and deep standby mode, for reducing power consumption. The standby mode consists of SLEEP mode, TIMER mode, RTC mode and STOP mode; the deep standby mode consists of deep standby RTC mode and deep standby STOP mode.

■ Overview of CPU operation modes

The CPU operation modes consist of the following operation modes.

● Run modes

- High speed CR run mode
- Main run mode
- PLL run mode
- Low speed CR run mode
- Sub run mode

● Standby modes

- SLEEP modes
  - High speed CR sleep mode
  - Main sleep mode
  - PLL sleep mode
  - Low speed CR sleep mode
  - Sub sleep mode

- TIMER modes
  - High speed CR timer mode
  - Main timer mode
  - PLL timer mode
  - Low speed CR timer mode
  - Sub timer mode

- RTC mode
- STOP mode

● Deep standby modes

- Deep standby RTC mode
- Deep standby stop mode

■ Overview of RUN modes

A RUN mode is defined according to which clock is selected as the master clock. A base clock acquired by dividing a master clock frequency is supplied to the CPU clock, the AHB bus clock and the APB bus clock to run the CPU, buses and most peripherals.

The source clock frequency can be modified dynamically. The source clock oscillator can be stopped when the main oscillator or the sub oscillator is not used.

Below are the five RUN modes, which are defined according to the clock selected as the master clock.

● High speed CR run mode

In this mode, the high speed CR oscillator clock is used as the master clock. The main oscillator or the sub oscillator can be stopped when not used. The state of the PLL Multiplier Circuit changes according to the setting of the PLLE bit. The low speed CR oscillator is always in the active state. The system transits to this mode after a reset has been released.

● Main run mode

In this mode, the main oscillator clock is used as the master clock. The respective states of the PLL Multiplier Circuit and the sub oscillator change according to the respective settings of the PLLE bit and the SOSCE bit, respectively. The high speed CR oscillator and the low speed CR oscillator are always in the active state.

● PLL run mode

In this mode, the PLL clock acquired by multiplying a PLL multiplier to the main oscillator clock or the high speed CR oscillator clock is used as the master clock. The high speed CR oscillator and the low speed CR oscillator are always in the active state. The respective states of the main oscillator and the sub oscillator change according to the respective settings of the MOSCE bit and the SOSCE bit, respectively.

● Low speed CR run mode

In this mode, the low speed CR oscillator clock is used as the master clock. The state of the sub oscillator changes according to the setting of the SOSCE bit. The main oscillator, the high speed CR oscillator and the PLL Multiplier Circuit cannot be used.

● Sub run mode

In this mode, the sub oscillator clock is used as the master clock. The low speed CR oscillator is always in the active state. The main oscillator, the high speed CR oscillator and the PLL Multiplier Circuit cannot be used.

■ Overview of SLEEP modes

SLEEP mode is classified as a standby mode. In SLEEP mode, the clock supply to the CPU stops. Since the stop of the clock supply to the CPU causes the CPU to stop, power consumption is reduced. Resources connected to the AHB bus and the APB bus continue operating.

Below are the five SLEEP modes, which are defined according to the master clock being used at the transition to SLEEP mode.

● High speed CR sleep mode

With the high speed CR oscillator clock selected as the master clock, the system transits to high speed CR sleep mode when a request for transition to SLEEP mode is made. The respective states of the PLL Multiplier Circuit, the main oscillator and the sub oscillator change according to the respective settings of the PLLE bit, the MOSCE bit and the SOSCE bit, respectively. The low speed CR oscillator is always in the active state.

● Main sleep mode

With the main clock selected as the master clock, the system transits to main sleep mode when a request for transition to SLEEP mode is made. The respective states of the PLL Multiplier Circuit and the sub oscillator change according to the respective settings of the PLLE bit and the SOSCE bit, respectively. The high speed CR oscillator and the low speed CR oscillator are always in the active state.

● PLL sleep mode

With the PLL clock selected as the master clock, the system transits to PLL sleep mode when a request for transition to SLEEP mode is made. The high speed CR oscillator and the low speed CR oscillator are always in the active state. The respective states of the main oscillator and the sub oscillator change according to the respective settings of the MOSCE bit and the SOSCE bit, respectively.

● Low speed CR sleep mode

With the low speed CR oscillator clock selected as the master clock, the system transits to low speed CR sleep mode when a request for transition to SLEEP mode is made. The state of the sub oscillator changes according to the setting of the SOSCE bit. The main oscillator, the high speed CR oscillator and the PLL Multiplier Circuit cannot be used.

● Sub sleep mode

With the sub-clock selected as the master clock, the system transits to sub-sleep mode when a request for transition to SLEEP mode is made. The high speed CR oscillator and the low speed CR oscillator are always in the active state. The main oscillator, the high speed CR oscillator and the PLL Multiplier Circuit cannot be used.

■ Overview of TIMER modes

TIMER mode is classified as a standby mode. In TIMER mode, the base clock supply stops. Since the stop of the base clock supply causes the CPU clock, the AHB bus clock and all APB bus clocks to stop, power consumption is further reduced. In TIMER mode, all functions stop operating except for the following: all oscillators, PLL, hardware watchdog timer, watch counter, RTC, clock failure detector and Low Voltage Detection Circuit.

Below are the five TIMER modes, which are defined according to the master clock used at the transition to TIMER mode.

● High speed CR timer mode

With the high speed CR oscillator clock selected as the master clock, the system transits to high speed CR timer mode when a request for transition to TIMER mode is made. The respective states of the PLL Multiplier Circuit, the main oscillator and the sub oscillator change according to the respective settings of the PLLE bit, the MOSCE bit and the SOSCE bit, respectively. The low speed CR oscillator is always in the active state.

● Main timer mode

With the main clock selected as the master clock, the system transits to main timer mode when a request for transition to TIMER mode is made. The respective states of the PLL Multiplier Circuit and the sub oscillator change according to the respective settings of the PLLE bit and the SOSCE bit, respectively. The high speed CR oscillator and the low speed CR oscillator are always in the active state.

● PLL timer mode

With the PLL clock selected as the master clock, the system transits to PLL timer mode when a request for transition to TIMER mode is made. The high speed CR oscillator and the low speed CR oscillator are always in the active state. The respective states of the main oscillator and the sub oscillator change according to the respective settings of the MOSCE bit and the SOSCE bit, respectively.

● Low speed CR timer mode

With the low speed CR oscillator clock selected as the master clock, the system transits to low speed CR timer mode when a request for transition to TIMER mode is made. The state of the sub oscillator changes according to the setting of the SOSCE bit. The main oscillator, the high speed CR oscillator and the PLL Multiplier Circuit cannot be used.

● Sub timer mode

With the sub-clock selected as the master clock, the system transits to subtimer mode when a request for transition to TIMER mode is made. The high speed CR oscillator and the low speed CR oscillator are always in the active state. The main oscillator, the high speed CR oscillator and the PLL Multiplier Circuit cannot be used.

■ Overview of RTC mode

RTC mode is classified as a standby mode. In RTC mode, all oscillators stop except for the sub oscillator. All functions stop operating except for the watch counter, the RTC and the Low Voltage Detection Circuit.

■ Overview of STOP mode

STOP mode is classified as a standby mode. In STOP mode, all oscillators stop. All functions stop operating except for the Low Voltage Detection Circuit.

■ Overview of deep standby RTC mode

Deep standby RTC mode is classified as a deep standby mode. In deep standby RTC mode, all oscillators stop except for the sub oscillator. All functions stop operating except for the RTC and the Low Voltage Detection Circuit. The power supply for the RTC, the Low Voltage Detection Circuit, CPUs excluding GPIO, on-chip Flash memory, on-chip SRAM0/1/2* and peripherals are turned off inside the chip.

■ Overview of deep standby stop mode

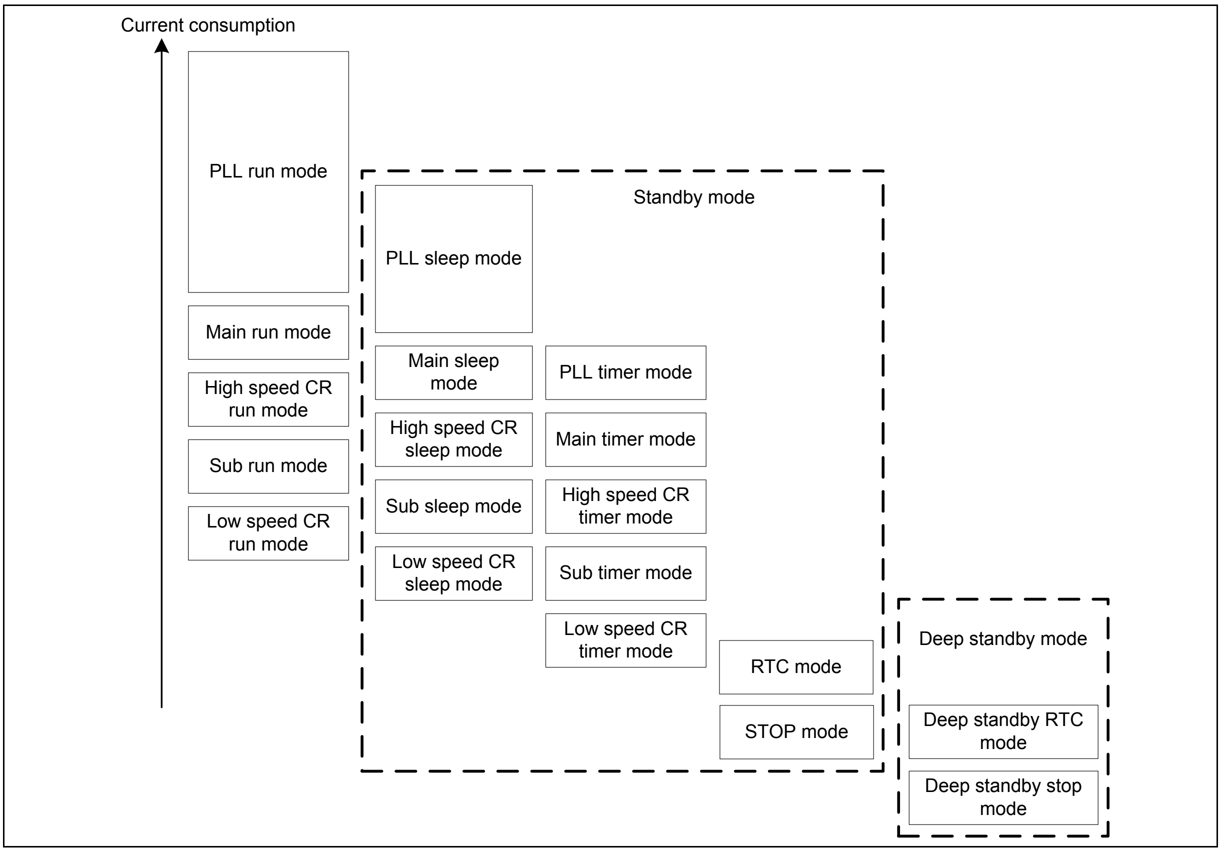
Deep standby stop mode is classified as a deep standby mode. In deep standby stop mode, all oscillators stop. All functions stop operating except for the Low Voltage Detection Circuit. The power supply for the RTC, the Low Voltage Detection Circuit, CPUs excluding GPIO, on-chip Flash memory, SRAM0/1/2* and peripherals are turned off inside the chip.

*: Data in SRAM2 can be retained.
If the setting for retaining data in SRAM2 is done, the power supply for SRAM2 is turned on.

■ Relationship between CPU operation modes and current consumption

Figure 1-1 illustrates the relationship between CPU operation modes and current consumption.

Figure 1-1 Relationship between CPU operation modes and current consumption



<Note>

Figure 1-1 shows only the relationship among CPU operation modes in terms of the size of current consumption. The actual size of current consumption varies according to the oscillators used and PLL starting conditions in a CPU operation mode, and clock configuration such as the frequency selected.

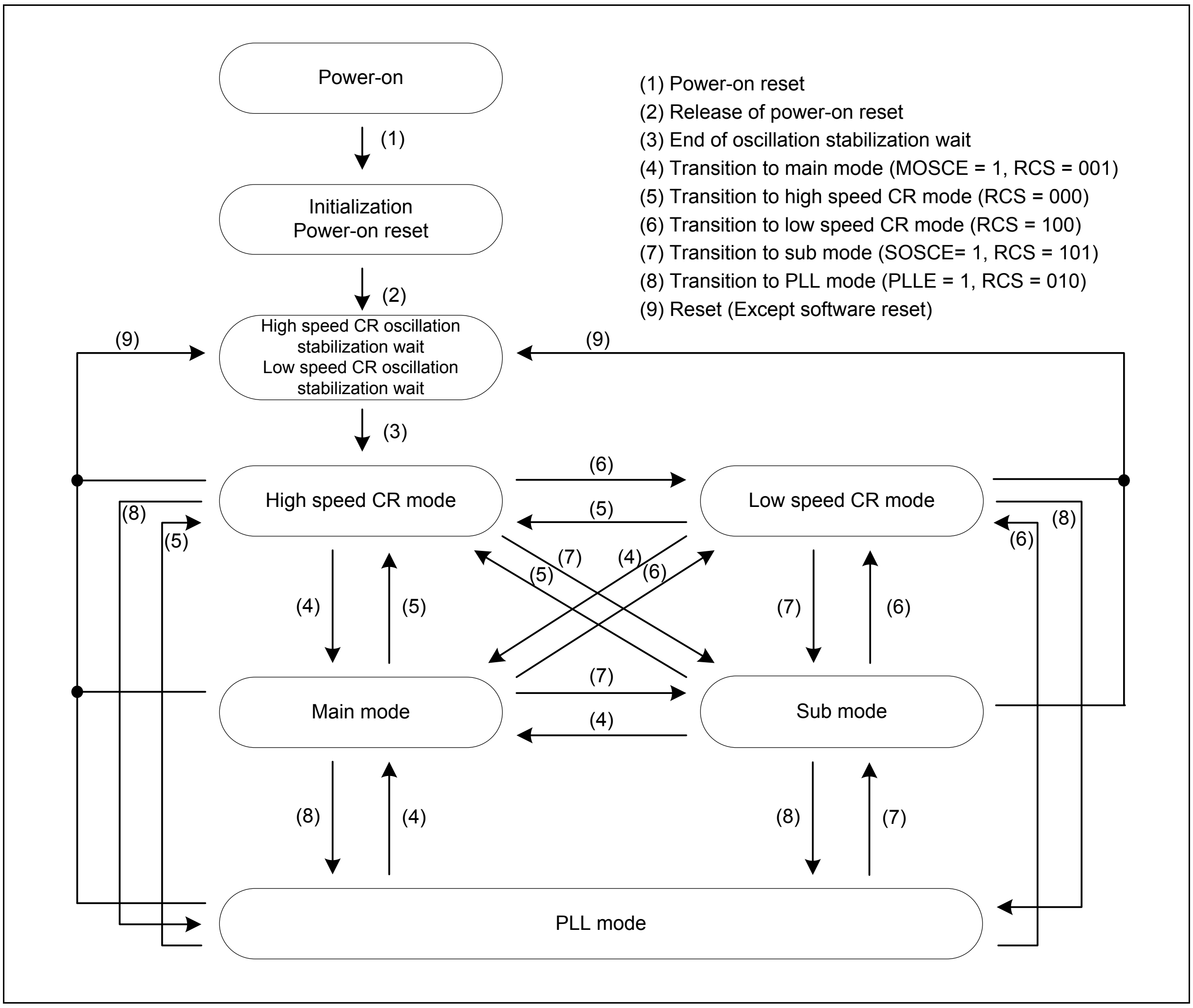
2. Configuration of CPU Operation Modes

This section explains the configuration of CPU operation modes.

■ CPU operation mode transition diagram

Figure 2-1 shows the CPU operation mode transition diagram.

Figure 2-1 CPU operation mode transition diagram


Power-on
Initialization
Power-on reset
High speed CR oscillation stabilization wait
Low speed CR oscillation stabilization wait
High speed CR mode
Low speed CR mode
Main mode
Sub mode
PLL mode
(1)
(2)
(3)
(4)
(5)
(6)
(7)
(8)
(9)
(1) Power-on reset
(2) Release of power-on reset
(3) End of oscillation stabilization wait
(4) Transition to main mode (MOSCE = 1, RCS = 001)
(5) Transition to high speed CR mode (RCS = 000)
(6) Transition to low speed CR mode (RCS = 100)
(7) Transition to sub mode (SOSCE= 1, RCS = 101)
(8) Transition to PLL mode (PLLE = 1, RCS = 010)
(9) Reset (Except software reset)

● High speed CR mode

In this mode, the high speed CR oscillator clock is used as the master clock.

● Main mode

In this mode, the main oscillator clock is used as the master clock.

● Low speed CR mode

In this mode, the low speed CR oscillator clock is used as the master clock.

● Sub mode

In this mode, the sub oscillator clock is used as the master clock.

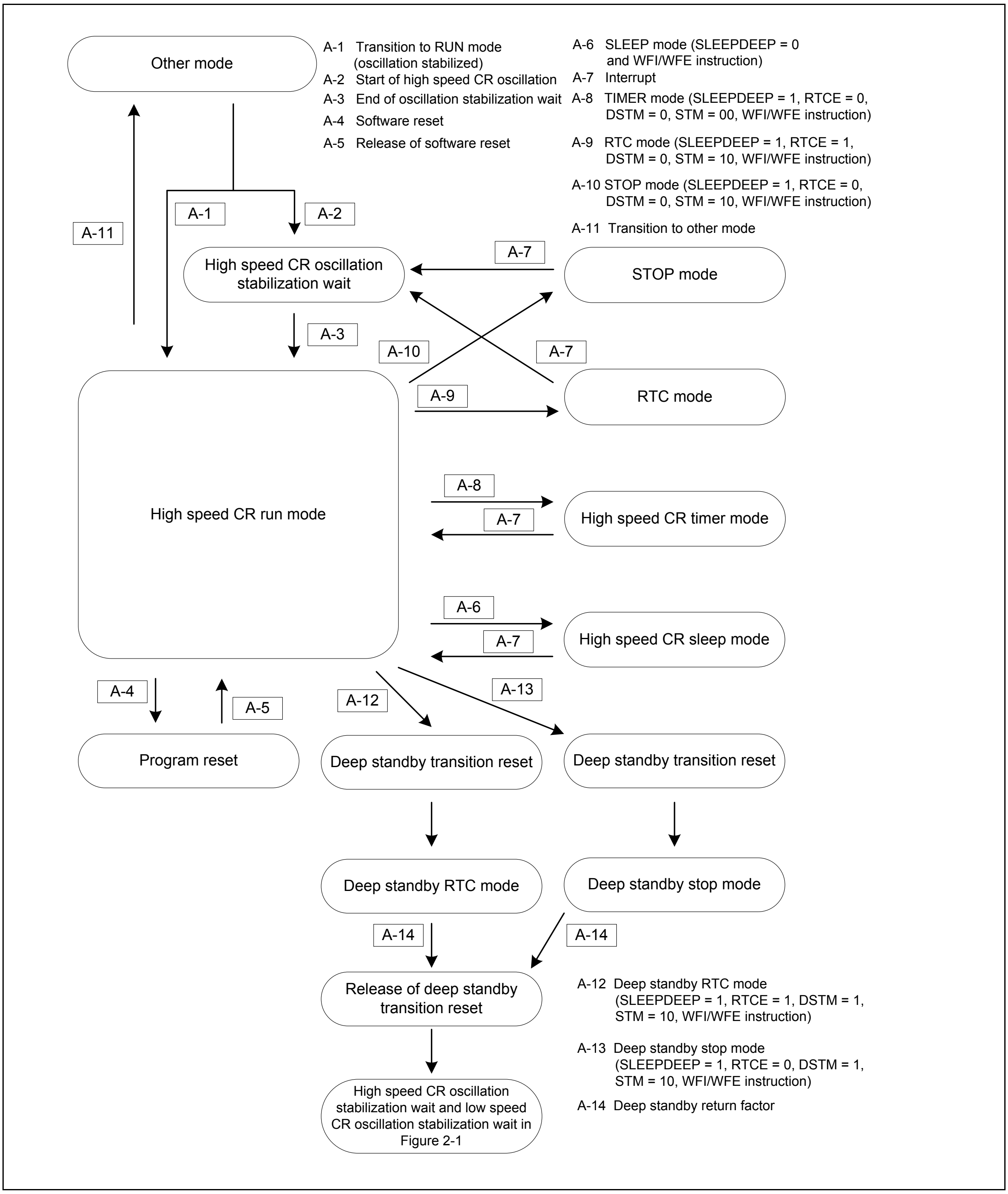
● PLL mode

In this mode, the PLL clock is used as the master clock.

■ High speed CR mode transition diagram

In high speed CR mode, the high speed CR oscillator clock is used as the master clock.

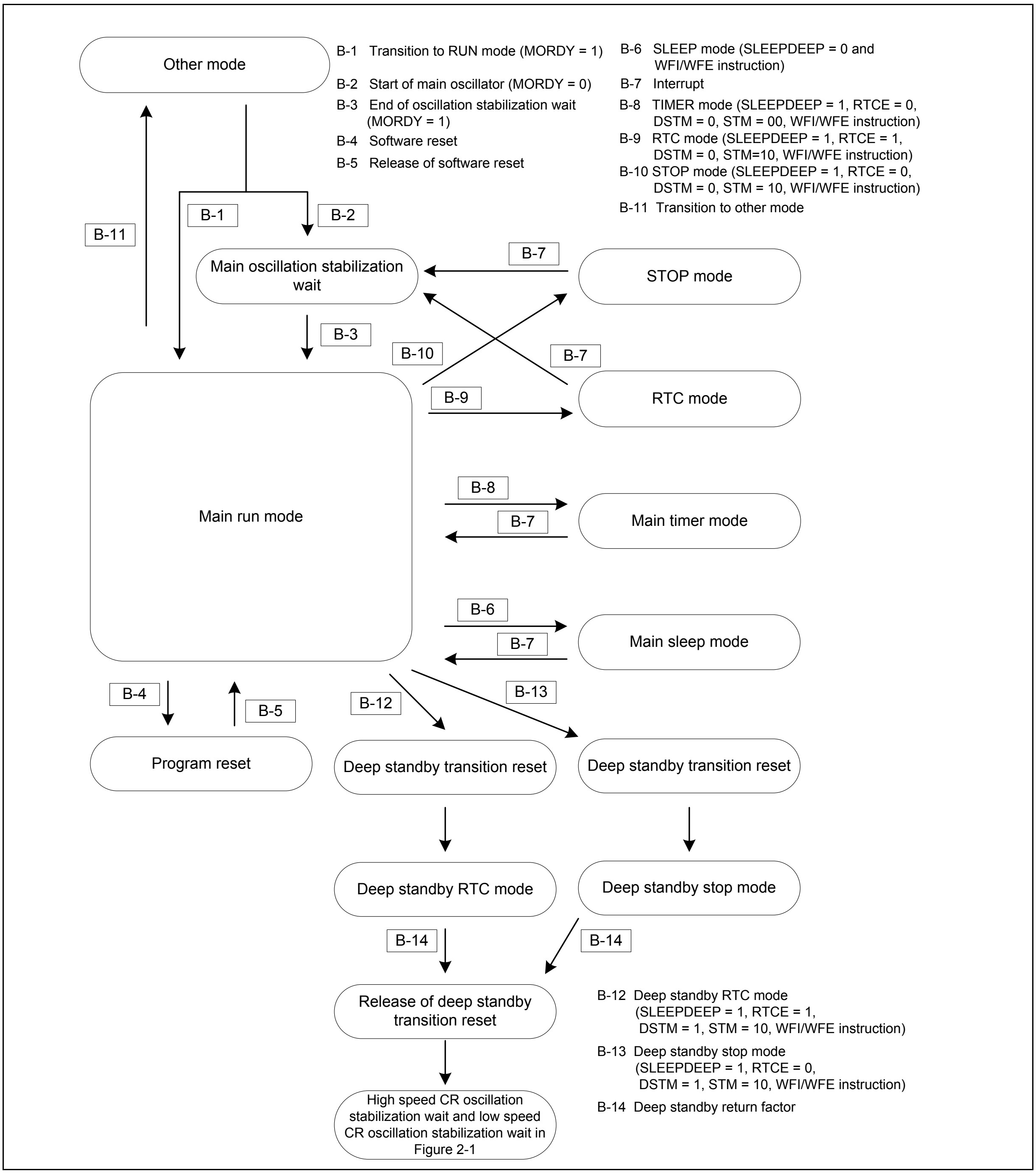
Figure 2-2 High speed CR mode transition diagram


Other mode
A-1 Transition to RUN mode (oscillation stabilized)
A-2 Start of high speed CR oscillation
A-3 End of oscillation stabilization wait
A-4 Software reset
A-5 Release of software reset
A-6 SLEEP mode (SLEEPDEEP = 0 and WFI/WFE instruction)
A-7 Interrupt
A-8 TIMER mode (SLEEPDEEP = 1, RTCE = 0, DSTM = 0, STM = 00, WFI/WFE instruction)
A-9 RTC mode (SLEEPDEEP = 1, RTCE = 1, DSTM = 0, STM = 10, WFI/WFE instruction)
A-10 STOP mode (SLEEPDEEP = 1, RTCE = 0, DSTM = 0, STM = 10, WFI/WFE instruction)
A-11 Transition to other mode
A-11
A-1
A-2
High speed CR oscillation stabilization wait
A-7
STOP mode
A-3
A-10
A-7
A-9
RTC mode
High speed CR run mode
A-8
A-7
High speed CR timer mode
A-6
A-7
High speed CR sleep mode
A-4
A-5
A-12
A-13
Program reset
Deep standby transition reset
Deep standby transition reset
Deep standby RTC mode
Deep standby stop mode
A-14
A-14
Release of deep standby transition reset
High speed CR oscillation stabilization wait and low speed CR oscillation stabilization wait in Figure 2-1
A-12 Deep standby RTC mode (SLEEPDEEP = 1, RTCE = 1, DSTM = 1, STM = 10, WFI/WFE instruction)
A-13 Deep standby stop mode (SLEEPDEEP = 1, RTCE = 0, DSTM = 1, STM = 10, WFI/WFE instruction)
A-14 Deep standby return factor

■ Main mode transition diagram

In main mode, the main oscillator clock is used as the master clock.

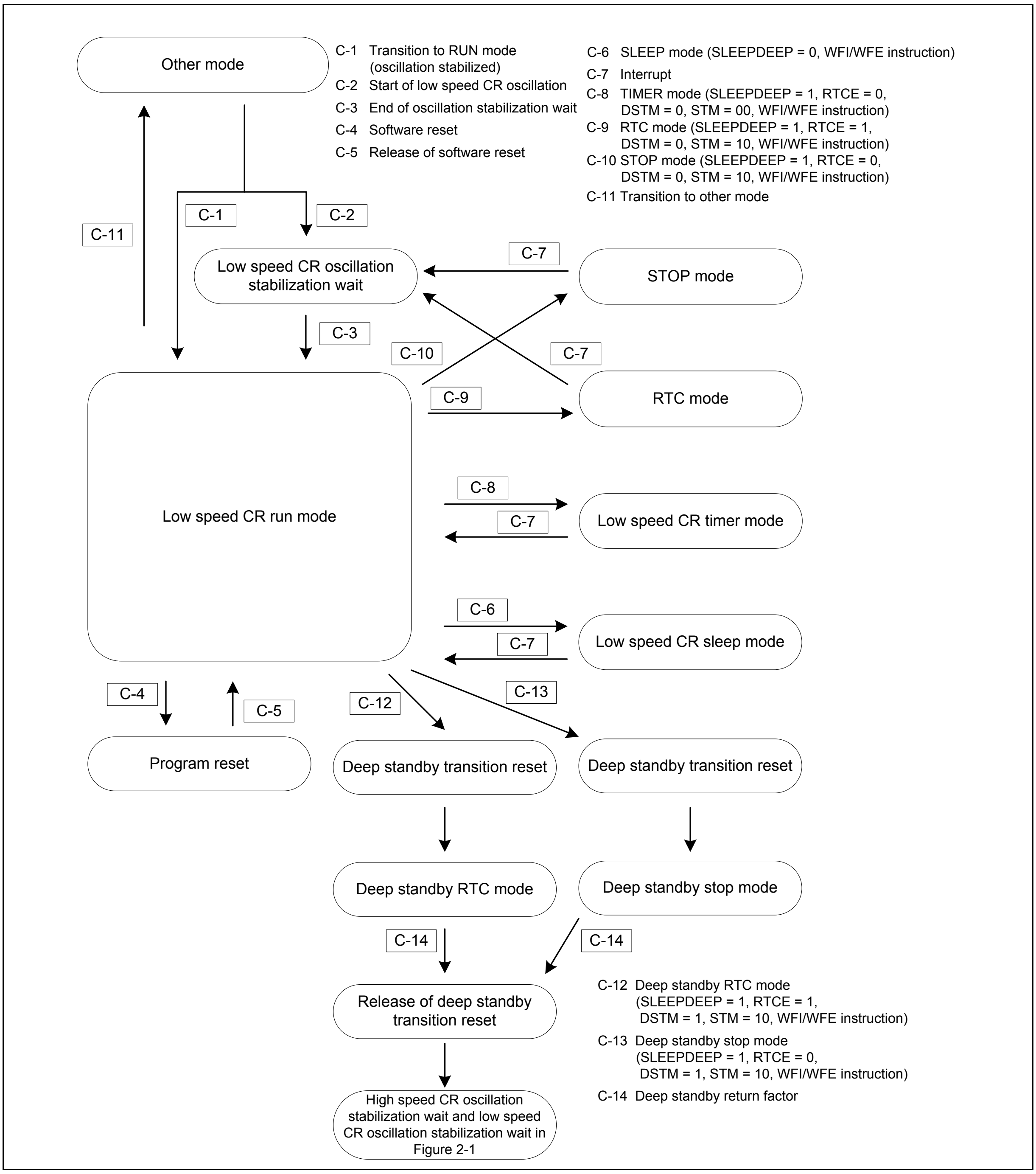
Figure 2-3 Main mode transition diagram


Other mode
B-1 Transition to RUN mode (MORDY = 1)
B-2 Start of main oscillator (MORDY = 0)
B-3 End of oscillation stabilization wait (MORDY = 1)
B-4 Software reset
B-5 Release of software reset
B-6 SLEEP mode (SLEEPDEEP = 0 and WFI/WFE instruction)
B-7 Interrupt
B-8 TIMER mode (SLEEPDEEP = 1, RTCE = 0, DSTM = 0, STM = 00, WFI/WFE instruction)
B-9 RTC mode (SLEEPDEEP = 1, RTCE = 1, DSTM = 0, STM=10, WFI/WFE instruction)
B-10 STOP mode (SLEEPDEEP = 1, RTCE = 0, DSTM = 0, STM = 10, WFI/WFE instruction)
B-11 Transition to other mode
B-11
B-1
B-2
Main oscillation stabilization wait
B-7
STOP mode
B-3
B-10
B-7
B-9
RTC mode
Main run mode
B-8
B-7
Main timer mode
B-6
B-7
Main sleep mode
B-4
B-5
B-12
B-13
Program reset
Deep standby transition reset
Deep standby transition reset
Deep standby RTC mode
Deep standby stop mode
B-14
B-14
Release of deep standby transition reset
High speed CR oscillation stabilization wait and low speed CR oscillation stabilization wait in Figure 2-1
B-12 Deep standby RTC mode (SLEEPDEEP = 1, RTCE = 1, DSTM = 1, STM = 10, WFI/WFE instruction)
B-13 Deep standby stop mode (SLEEPDEEP = 1, RTCE = 0, DSTM = 1, STM = 10, WFI/WFE instruction)
B-14 Deep standby return factor

■ Low speed CR mode transition diagram

In low speed CR mode, the low speed CR oscillator clock is used as the master clock.

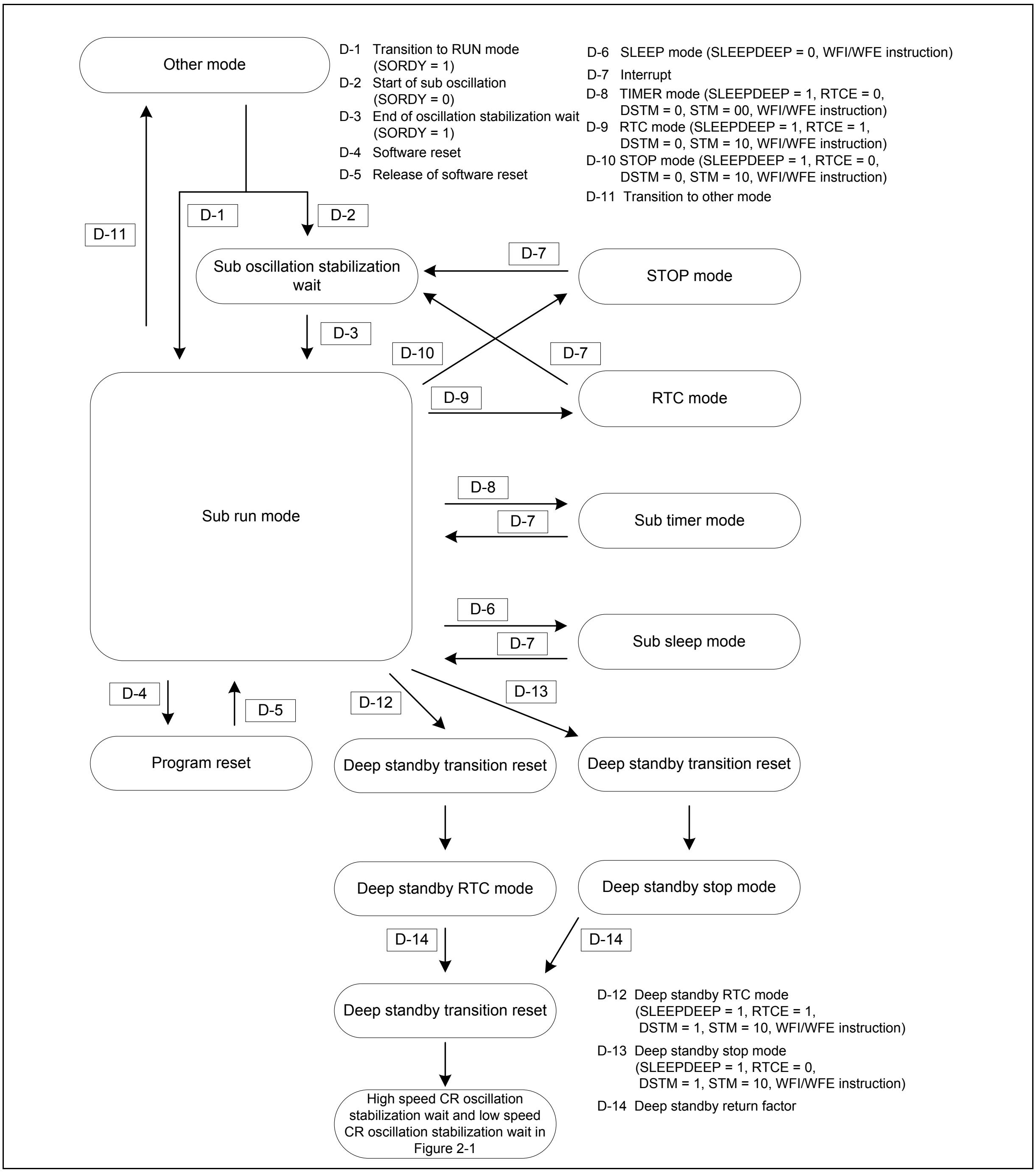
Figure 2-4 Low speed CR mode transition diagram


Other mode
C-1 Transition to RUN mode (oscillation stabilized)
C-2 Start of low speed CR oscillation
C-3 End of oscillation stabilization wait
C-4 Software reset
C-5 Release of software reset
C-6 SLEEP mode (SLEEPDEEP = 0, WFI/WFE instruction)
C-7 Interrupt
C-8 TIMER mode (SLEEPDEEP = 1, RTCE = 0, DSTM = 0, STM = 00, WFI/WFE instruction)
C-9 RTC mode (SLEEPDEEP = 1, RTCE = 1, DSTM = 0, STM = 10, WFI/WFE instruction)
C-10 STOP mode (SLEEPDEEP = 1, RTCE = 0, DSTM = 0, STM = 10, WFI/WFE instruction)
C-11 Transition to other mode
C-11
C-1
C-2
Low speed CR oscillation stabilization wait
C-7
STOP mode
C-3
C-10
C-7
RTC mode
C-9
Low speed CR run mode
C-8
C-7
Low speed CR timer mode
C-6
C-7
Low speed CR sleep mode
C-4
C-5
C-12
C-13
Program reset
Deep standby transition reset
Deep standby transition reset
Deep standby RTC mode
Deep standby stop mode
C-14
C-14
Release of deep standby transition reset
High speed CR oscillation stabilization wait and low speed CR oscillation stabilization wait in Figure 2-1
C-12 Deep standby RTC mode (SLEEPDEEP = 1, RTCE = 1, DSTM = 1, STM = 10, WFI/WFE instruction)
C-13 Deep standby stop mode (SLEEPDEEP = 1, RTCE = 0, DSTM = 1, STM = 10, WFI/WFE instruction)
C-14 Deep standby return factor

■ Sub mode transition diagram

In sub mode, the sub oscillator clock is used as the master clock.

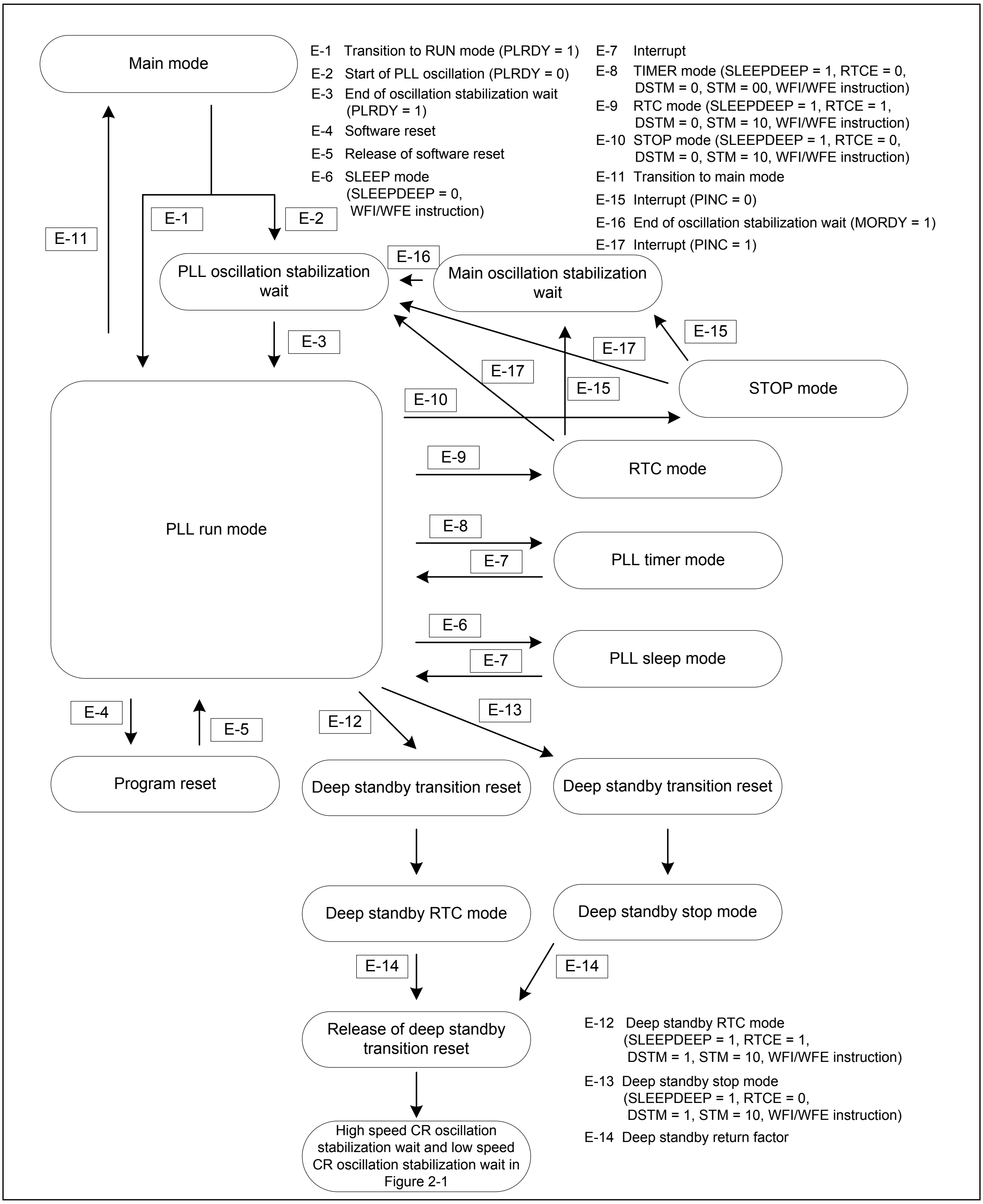
Figure 2-5 Sub mode transition diagram


Other mode
D-1 Transition to RUN mode (SORDY = 1)
D-2 Start of sub oscillation (SORDY = 0)
D-3 End of oscillation stabilization wait (SORDY = 1)
D-4 Software reset
D-5 Release of software reset
D-6 SLEEP mode (SLEEPDEEP = 0, WFI/WFE instruction)
D-7 Interrupt
D-8 TIMER mode (SLEEPDEEP = 1, RTCE = 0, DSTM = 0, STM = 00, WFI/WFE instruction)
D-9 RTC mode (SLEEPDEEP = 1, RTCE = 1, DSTM = 0, STM = 10, WFI/WFE instruction)
D-10 STOP mode (SLEEPDEEP = 1, RTCE = 0, DSTM = 0, STM = 10, WFI/WFE instruction)
D-11 Transition to other mode
D-11
D-1
D-2
Sub oscillation stabilization wait
D-7
STOP mode
D-3
D-10
D-7
RTC mode
D-9
Sub run mode
D-8
D-7
Sub timer mode
D-6
D-7
Sub sleep mode
D-4
D-5
D-12
D-13
Program reset
Deep standby transition reset
Deep standby transition reset
Deep standby RTC mode
Deep standby stop mode
D-14
D-14
Deep standby transition reset
High speed CR oscillation stabilization wait and low speed CR oscillation stabilization wait in Figure 2-1
D-12 Deep standby RTC mode (SLEEPDEEP = 1, RTCE = 1, DSTM = 1, STM = 10, WFI/WFE instruction)
D-13 Deep standby stop mode (SLEEPDEEP = 1, RTCE = 0, DSTM = 1, STM = 10, WFI/WFE instruction)
D-14 Deep standby return factor

■ PLL mode transition diagram

In PLL mode, the PLL clock is used as the master clock.

Figure 2-6 PLL mode transition diagram


Main mode
E-1 Transition to RUN mode (PLRDY = 1)
E-2 Start of PLL oscillation (PLRDY = 0)
E-3 End of oscillation stabilization wait (PLRDY = 1)
E-4 Software reset
E-5 Release of software reset
E-6 SLEEP mode (SLEEPDEEP = 0, WFI/WFE instruction)
E-7 Interrupt
E-8 TIMER mode (SLEEPDEEP = 1, RTCE = 0, DSTM = 0, STM = 00, WFI/WFE instruction)
E-9 RTC mode (SLEEPDEEP = 1, RTCE = 1, DSTM = 0, STM = 10, WFI/WFE instruction)
E-10 STOP mode (SLEEPDEEP = 1, RTCE = 0, DSTM = 0, STM = 10, WFI/WFE instruction)
E-11 Transition to main mode
E-15 Interrupt (PINC = 0)
E-16 End of oscillation stabilization wait (MORDY = 1)
E-17 Interrupt (PINC = 1)
E-11
E-1
E-2
E-16
PLL oscillation stabilization wait
Main oscillation stabilization wait
E-3
E-15
E-17
E-17
E-15
E-10
STOP mode
E-9
RTC mode
PLL run mode
E-8
E-7
PLL timer mode
E-6
E-7
PLL sleep mode
E-4
E-5
E-12
E-13
Program reset
Deep standby transition reset
Deep standby transition reset
Deep standby RTC mode
Deep standby stop mode
E-14
E-14
Release of deep standby transition reset
High speed CR oscillation stabilization wait and low speed CR oscillation stabilization wait in Figure 2-1
E-12 Deep standby RTC mode (SLEEPDEEP = 1, RTCE = 1, DSTM = 1, STM = 10, WFI/WFE instruction)
E-13 Deep standby stop mode (SLEEPDEEP = 1, RTCE = 0, DSTM = 1, STM = 10, WFI/WFE instruction)
E-14 Deep standby return factor

MOSCE	:MOSCE bit in System Clock Mode Control Register (SCM_CTL)
SOSCE	:SOSCE bit in System Clock Mode Control Register (SCM_CTL)
PLLE	:PLLE bit in System Clock Mode Control Register (SCM_CTL)
RCS	:RSC bit in System Clock Mode Control Register (SCM_CTL)
MORDY	:MORDY bit in System Clock Mode Status Register (SCM_STR)
SORDY	:SORDY bit in System Clock Mode Status Register (SCM_STR)

PLRDY :PLRDY bit in System Clock Mode Status Register (SCM_STR)
PINC :PINC bit in PLL Clock Stabilization Wait Time Setup Register (PSW_TMR)

*: For details of the SCM_CTL, SCM_STR and PSW_TMR registers, see Chapter "Clock".

<Note>

The CPU automatically secures a voltage stabilization wait time (a few hundred μs) for the built-in regulator immediately before returning from low speed CR timer mode, sub timer mode, RTC mode, stop mode, deep standby RTC mode or deep standby stop mode. After the voltage stabilization wait time has lapsed, the CPU returns to a RUN mode.

3. Operations in Standby Modes

This section explains the operations in standby modes.
There are four types of standby mode: SLEEP mode (high speed CR sleep mode, main sleep mode, PLL sleep mode, low speed CR sleep mode and sub sleep mode), TIMER mode (high speed CR timer mode, main timer mode, PLL timer mode, low speed CR timer mode and sub timer mode), RTC mode and STOP mode.

■ Clock operation states in standby mode

The table below shows the respective states of the oscillator clocks, CPU clock, AHB bus clock and APB bus clocks in SLEEP mode, TIMER mode, RTC mode and STOP mode.

Table 3-1 Clock operation states in SLEEP mode

	SLEEP mode				
	High speed CR sleep mode	Main sleep mode	PLL sleep mode	Low speed CR sleep mode	Sub sleep mode
High speed CR clock	Operating			Stopped	
Main clock	The state changes according to the setting of the MOSCE bit.	Operating	The state changes according to the setting of the MOSCE bit.	Stopped	
Main PLL clock	The state changes according to the setting of the PLLE bit.		Operating	Stopped	
Low speed CR clock	Operatingx				
Sub clock	The state changes according to the setting of the SOSCE bit.				Operating
USB PLL clock	Setting prohibited	The state changes according to the setting of the UPLLEN bit.		Stopped	
CPU clock	Stopped				
AHB bus clock	High speed CR clock	Main clock	PLL clock	Low speed CR clock	Sub clock
APB0 bus clock	High speed CR clock	Main clock	PLL clock	Low speed CR clock	Sub clock
APB1 bus clock	High speed CR clock	Main clock	PLL clock	Low speed CR clock	Sub clock
	* The APBC1EN bit enables or disables the APB1 bus clock.				
APB2 bus clock	High speed CR clock	Main clock	PLL clock	Low speed CR clock	Sub clock
	* The APBC2EN bit enables or disables the APB2 bus clock.				

Table 3-2 Clock operation states in TIMER mode

	TIMER mode				
	High speed CR timer mode	Main timer mode	PLL timer mode	Low speed CR timer mode	Sub timer mode
High speed CR clock	Operating			Stopped	
Main clock	The state changes according to the setting of the MOSCE bit.	Operating	The state changes according to the setting of the MOSCE bit.	Stopped	
Main PLL clock	The state changes according to the setting of the PLLE bit.		Operating	Stopped	
Low speed CR clock	Operating				
Sub clock	The state changes according to the setting of the SOSCE bit.				Operating
USB PLL clock	Stopped				
CPU clock	Stopped				
AHB bus clock	Stopped				
APB0 bus clock	Stopped				
APB1 bus clock	Stopped				
APB2 bus clock	Stopped				

Table 3-3 Clock operation states in RTC mode and STOP mode

	RTC mode	STOP mode
High speed CR clock	Stopped	Stopped
Main clock		
Main PLL clock		
Low speed CR clock		
Sub clock	Operating	
USB PLL clock	Stopped	
CPU clock		
AHB bus clock		
APB0 bus clock		
APB1 bus clock		
APB2 bus clock		

MOSCE	:MOSCE bit in System Clock Mode Control Register (SCM_CTL)
SOSCE	:SOSCE bit in System Clock Mode Control Register (SCM_CTL)
PLLE	:PLLE bit in System Clock Mode Control Register (SCM_CTL)
UPLLEN	:UPLLEN bit in USB-PLL Control Register 1 (UPCR1)
APBC1EN	:APBC1EN bit in APB1 Prescaler Register (APBC1_PSR)
APBC2EN	:APBC2EN bit in APB2 Prescaler Register (APBC2_PSR)

*: For details of the SCM_CTL, APBC1_PSR and APBC2_PSR registers, see Chapter "Clock".
For details of the UPCR1 register, refer to chapter "USB Clock Generation" in "Communication Macro Part”.

■ Factors for returning from standby mode

Table 3-4 shows factors for returning from SLEEP, TIMER, RTC and STOP modes.

Table 3-4 Factors for returning from standby mode

	SLEEP mode	TIMER mode	RTC mode	STOP mode
Reset return factor	· INITX pin input reset · Low voltage detection reset · Software watchdog reset · Hardware watchdog reset · Clock failure detection reset · Anomalous frequency detection reset	· INITX pin input reset · Low voltage detection reset · Hardware watchdog reset · Clock failure detection reset · Anomalous frequency detection reset (main timer mode, PLL timer mode)	· INITX pin input reset · Low voltage detection reset	· INITX pin input reset · Low voltage detection reset
Interrupt return factor	· Effective interrupt from a peripherals	· NMI interrupt · External interrupt · Hardware watchdog timer interrupt · USB wakeup interrupt · Watch counter interrupt · RTC interrupt · Low voltage detection interrupt · Anomalous frequency detection interrupt (main timer mode, PLL timer mode)	· NMI interrupt · External interrupt · USB wakeup interrupt · RTC interrupt · Low voltage detection interrupt	· NMI interrupt · External interrupt · USB wakeup interrupt · Low voltage detection interrupt

3.1. Operations in SLEEP Modes (High Speed CR Sleep Mode, Main Sleep Mode, PLL Sleep Mode, Low Speed CR Sleep Mode, Sub Sleep Mode)

SLEEP mode is classified as a standby mode. In SLEEP mode, the CPU clock stops and, in turn, power consumption is reduced.

■ Functions of SLEEP mode

● CPU and on-chip memory

In SLEEP mode, the clock being supplied to the CPU stops. The AHB bus clock continues operating. The on-chip memory continues operating and retains data.

● Peripherals

The APB0 bus clock runs even in SLEEP mode. The APB1 bus clock is enabled or disabled by the APBC1EN bit and the APB2 bus clock by the APBC2EN bit. Peripherals operate in their respective states set at the transition to SLEEP mode.

● Watch counter and RTC

The watch counter and the RTC are not affected by SLEEP mode. They continue operating according to the settings set prior to transition to SLEEP mode.

● Oscillator clocks

Table 3-1 shows the status of each oscillator clock.

● Resets and interrupts

Resets and interrupts can be used for returning from SLEEP mode.

● External bus

The external bus is still active in SLEEP mode.

● Pin state

All pin settings are held in SLEEP mode.

■ Procedure for setting SLEEP mode

Execute the following procedure to make the CPU transit to SLEEP mode.

1. Write "0" to the SLEEPDEEP bit in the Cortex-M4F System Control Register.
2. Execute the WFI instruction or the WFE instruction.
   The CPU transits to a SLEEP mode corresponding to the current clock mode indicated in the RCM[2:0] bits in the System Clock Mode Status Register (SCM_STR).

For details of the System Clock Mode Control Register (SCM_CTL), see Chapter "Clock".

■ Return from SLEEP mode

The CPU returns from SLEEP mode in one of the following situations.

● Return due to reset

If a reset (INITX pin input reset, low voltage detection reset, software watchdog reset, hardware watchdog reset, clock supervisor reset or anomalous frequency detection reset) occurs, the CPU switches to high speed CR run mode regardless of the clock mode.

● Return due to interrupt

On receiving an effective interrupt from a peripheral in SLEEP mode, the CPU returns from SLEEP mode and transits to a RUN mode corresponding to the clock mode indicated in the RCM[2:0] bits in the System Clock Mode Status Register (SCM_STR).

Table 3-5 Operation mode after return from SLEEP mode due to interrupt

	Status of master clock before transition to SLEEP mode				
	RCM = 000 (High speed CR oscillator)	RCM = 001 (Main oscillator)	RCM = 010 (PLL oscillator)	RCM = 100 (Low speed CR oscillator)	RCM = 101 (Sub oscillator)
Operation mode after return due to interrupt	High speed CR run mode	Main run mode	PLL run mode	Low speed CR run mode	Sub run mode

RCM: RCM[2:0] bits in System Clock Mode Status Register (SCM_STR)

*: For details of the SCM_CTL and SCM_STR registers, see Chapter "Clock".

● Oscillation stabilization wait at return

On returning from SLEEP mode due to a reset, the CPU waits for the stabilization of high speed CR clock oscillation and that of low speed CR clock oscillation. In the case of returning from SLEEP mode due to an interrupt, the CPU does not have to wait for the oscillation to stabilize.

3.2. Operations in TIMER Modes (High Speed CR Timer Mode, Main Timer Mode, PLL Timer Mode, Low Speed CR Timer Mode, Sub Timer Mode)

In TIMER mode, the base clock supply stops. Since the stop of the base clock supply causes the CPU clock, the AHB bus clock and all APB bus clocks to stop, power consumption is further reduced. In this mode, all functions stop operating except for the following: all oscillators, PLL, hardware watchdog timer, watch counter, RTC, clock failure detector and Low Voltage Detection Circuit.

■ Functions of TIMER mode

● CPU and on-chip memory

In TIMER mode, the CPU clock supplied to the CPU, and the AHB bus clock supplied to the on-chip memory and the DMA controller stop. However, data in the on-chip memory is retained. In addition, the debug function stops.

● Peripherals

In TIMER mode, all APB bus clocks stop. Except for the hardware watchdog timer, the watch counter, the RTC and the clock supervisor, all resources stop at their respective states they were in immediately before the CPU transits to TIMER mode.

● Watch counter and RTC

The watch counter and the RTC are not affected by TIMER mode. They continue operating according to the settings set prior to transition to TIMER mode.

● Oscillator clocks

Table 3-2 shows the status of each oscillator clock.

● Resets and interrupts

Resets and interrupts can be used for returning from TIMER mode.

● External bus

The external bus stops in TIMER mode.

● Pin state

The SPL bit in the Standby Mode Control Register (STB_CTL) can control whether an external pin stays at the state it was in immediately before the CPU transits to TIMER mode or changes to high impedance state.

■ Procedure for setting TIMER mode

Execute the following procedure to make the CPU transit to TIMER mode.

1. Write "0" to the RTCE bit in the RTC Mode Control Register (PMD_CTL).
2. Write "0x1ACC", "0" and "0b00" to the KEY bits, DTSM bit and STM bits in the Standby Mode Control Register (STB_CTL) respectively. Set the state of each pin in TIMER mode by using the SPL bit.
3. Write "1" to the SLEEPDEEP bit in the Cortex-M4F System Control Register.
4. Execute the WFI instruction or the WFE instruction.
   The CPU transits to a TIMER mode corresponding to the current clock mode indicated in the RCM[2:0] bits in the System Clock Mode Status Register (SCM_STR).

■ Return from TIMER mode

The CPU returns from TIMER mode in one of the following situations.

● Return due to reset

If a reset (INITX pin input reset, low voltage detection reset, hardware watchdog reset, clock supervisor reset or anomalous frequency detection reset (main timer mode or PLL timer mode)) occurs, the CPU switches to high speed CR run mode regardless of the clock mode.

In TIMER mode, since the software watchdog reset is not available, it cannot be used to make the CPU return from TIMER mode.

● Return due to interrupt

On receiving a request for an effective interrupt, which can be an NMI interrupt, an external interrupt, a hardware watchdog timer interrupt, a USB wakeup interrupt, a watch counter interrupt, an RTC interrupt, a low voltage detection interrupt or an anomalous frequency detection interrupt (main timer mode or PLL timer mode), in TIMER mode, the CPU returns from TIMER mode and transits to a RUN mode corresponding to the clock mode indicated in the RCM[2:0] bits in the System Clock Mode Status Register (SCM_STR).

Table 3-6 Operation mode after return from TIMER mode due to interrupt

	Status of master clock before transition to TIMER mode				
	RCM = 000 (High speed CR oscillator)	RCM = 001 (Main oscillator)	RCM = 010 (PLL oscillator)	RCM = 100 (Low speed CR oscillator)	RCM = 101 (Sub oscillator)
Operation mode after return due to interrupt	High speed CR run mode	Main run mode	PLL run mode	Low speed CR run mode	Sub run mode

● Oscillation stabilization wait at return

On returning from TIMER mode due to a reset, the CPU waits for the stabilization of high speed CR clock oscillation and that of low speed CR clock oscillation. In the case of returning from TIMER mode due to an interrupt, the CPU does not have to wait for the oscillation to stabilize.

● Built-in regulator voltage stabilization wait at return

The CPU automatically secures a voltage stabilization wait time (a few hundred μs) for the built-in regulator before returning from low speed CR timer mode or sub timer mode due to a reset or an interrupt. After the voltage stabilization wait time has lapsed, the CPU executes the return operation.

<Notes>

- If the priority of an interrupt used for making the CPU return from TIMER mode is not set to a level for making the CPU return from TIMER mode, after an interrupt has been generated, the clock returns from TIMER mode, but the CPU does not return from TIMER mode and continues stopping. Therefore, always set the priority of an interrupt used for making the CPU return from TIMER mode to a level that can make the CPU return from TIMER mode.
- Before making the CPU transit to TIMER mode, ensure that no factor for returning from TIMER mode shown in Table 3-4 has been set. If such factor has been set, clear that factor.
- If the CPU transits to TIMER mode during debugging, as the clock supply to the CPU stops, the CPU cannot return to a RUN mode by using the ICE. Use a reset or an interrupt to make the CPU return to a RUN mode.
- Before making the CPU transit to low speed CR timer mode or sub timer mode, ensure that the Flash memory automatic algorithm has terminated.

3.3. Operations in RTC Mode

In RTC mode, all oscillators stop except for the sub oscillator. All functions stop operating except for the watch counter, the RTC and the Low Voltage Detection Circuit.

■ Functions of RTC mode

● CPU and on-chip memory

In RTC mode, the CPU clock supplied to the CPU, and the AHB bus clock supplied to the on-chip memory and the DMA controller stop. However, data in the on-chip memory is retained. In addition, the debug function stops.

● Peripherals

In RTC mode, all APB bus clocks stop. Except for the watch counter, the RTC and the Low Voltage Detection Circuit, all resources stop at their respective states they were in immediately before the CPU transits to RTC mode.

● Watch counter and RTC

The counting operation of the watch counter is not affected by the transition to RTC mode and the watch counter continues its operation according to the settings set prior to the transition. But, the watch counter cannot return to from the RTC mode due to the watch counter interrupt. The RTC is not affected by RTC mode transition. The RTC continues its operation according to the settings set prior to transition to RTC mode.

● Oscillator clocks

Table 3-3 shows the status of each oscillator clock.

● Resets and interrupts

Resets and interrupts can be used for returning from RTC mode.

● External bus

The external bus stops in RTC mode.

● Pin state

The SPL bit in the Standby Mode Control Register (STB_CTL) can control whether an external pin stays at the state it was in immediately before the CPU transits to RTC mode or changes to high impedance state.

■ Procedure for setting RTC mode

Execute the following procedure to make the CPU transit to RTC mode.

1. With "1" written to the SORDY bit in the System Clock Mode Status Register (SCM_STR), write "1" to the RTCE bit in the RTC Mode Control Register (PMD_CTL).
2. Write "0x1ACC", "0" and "0b10" to the KEY bits, DTSM bit and STM bits in the Standby Mode Control Register (STB_CTL) respectively. Set the state of each pin in RTC mode by using the SPL bit.
3. Write "1" to the SLEEPDEEP bit in the Cortex-M4F System Control Register.
4. Execute the WFI instruction or the WFE instruction.

■ Return from RTC mode

The CPU returns from RTC mode in one of the following situations.

● Return due to reset

If a reset (INITX pin input reset or low voltage detection reset) occurs, the CPU switches to high speed CR run mode regardless of the clock mode.

In RTC mode, since the software watchdog reset, the hardware watchdog reset, the clock supervisor reset and the anomalous frequency detection reset are not available, these resets cannot be used to make the CPU return from RTC mode.

● Return due to interrupt

On receiving a request for an effective interrupt, which can be an NMI interrupt, an external interrupt, a USB wakeup interrupt, an RTC interrupt or a low voltage detection interrupt, in RTC mode, the CPU returns from RTC mode and transits to a RUN mode corresponding to the clock mode indicated in the RCM[2:0] bits in the System Clock Mode Status Register (SCM_STR)

Table 3-7 Operation mode after return from RTC mode due to interrupt

	Status of master clock before transition to RTC mode				
	RCM = 000 (High speed CR oscillator)	RCM = 001 (Main oscillator)	RCM = 010 (PLL oscillator)	RCM = 100 (Low speed CR oscillator)	RCM = 101 (Sub oscillator)
Operation mode after return due to interrupt	High speed CR run mode	Main run mode	PLL run mode	Low speed CR run mode	Sub run mode

● Oscillation stabilization wait at return

On returning from RTC mode due to a reset, the CPU waits for the stabilization of high speed CR clock oscillation and that of low speed CR clock oscillation. If the CPU returns from RTC mode due to an interrupt, the oscillation stabilization wait changes according to the master clock that had been used before the CPU transited to RTC mode. Table 3-8 shows the relationship between the oscillation stabilization wait and the master clock.

Table 3-8 Oscillation stabilization wait at return from RTC mode due to interrupt

		Status of master clock before transition to RTC mode				
		RCM = 000 (High speed CR oscillator)	RCM = 001 (Main oscillator)	RCM = 010 (PLL oscillator)	RCM = 100 (Low speed CR oscillator)	RCM = 101 (Sub oscillator)
Oscillation stabilization wait at return due to interrupt	High speed CR clock	Yes			No	
	Main clock	MOSCE = 0 : Yes MOSCE = 1 : No	Yes	PINC = 0 : Yes PINC = 1 : No	No	
	Main PLL clock	No	PLLE = 0 : No PLLE = 1 : Yes	Yes	No	
	Low speed CR clock	Yes	Yes	Yes	Yes	Yes
	Sub clock	No	No	No	No	No

● Built-in regulator voltage stabilization wait at return

The CPU automatically secures a voltage stabilization wait time (a few hundred μs) for the built-in regulator before returning from RTC mode. After the voltage stabilization wait time has lapsed, the CPU executes the return operation.

<Notes>

- If the priority of an interrupt used for making the CPU return from RTC mode is not set to a level for making the CPU return from RTC mode, after an interrupt has been generated, the clock returns from RTC mode, but the CPU does not return from RTC mode and continues stopping. Therefore, always set the priority of an interrupt used for making the CPU return from RTC mode to a level that can make the CPU return from RTC mode.
- Before making the CPU transit to RTC mode, ensure that no factor for returning from RTC mode shown in Table 3-4 has been set. If such factor has been set, clear that factor.
- If the CPU transits to RTC mode during debugging, as the clock supply to the CPU stops, the CPU cannot return to a RUN mode by using the ICE. Use a reset or an interrupt to make the CPU return to a RUN mode.
- Before making the CPU transit to RTC mode, ensure that the Flash memory automatic algorithm has terminated.

3.4. Operations in STOP Mode

In STOP mode, all oscillators stop. All functions stop operating except for the Low Voltage Detection Circuit.

■ Functions of STOP mode

● CPU and on-chip memory

In STOP mode, the CPU clock supplied to the CPU, and the AHB bus clock supplied to the on-chip memory and the DMA controller stop. However, data in the on-chip memory is retained. In addition, the debug function stops.

● Peripherals

In STOP mode, all APB bus clocks stop. Except for the Low Voltage Detection Circuit, all resources stop at their respective states they were in immediately before the CPU transits to STOP mode.

● Oscillator clocks

All oscillator clocks stop.

● Resets and interrupts

Resets and interrupts can be used for returning from STOP mode.

● External bus

The external bus stops in STOP mode.

● Pin state

The SPL bit in the Standby Mode Control Register (STB_CTL) can control whether an external pin stays at the state it was in immediately before the CPU transits to STOP mode or changes to high impedance state.

■ Procedure for setting STOP mode

Execute the following procedure to make the CPU transit to STOP mode.

1. Write "0" to the RTCE bit in the RTC Mode Control Register (PMD_CTL).
2. Write "0x1ACC", "0" and "0b10" to the KEY bits, DTSM bit and STM bits in the Standby Mode Control Register (STB_CTL) respectively. Set the state of each pin in STOP mode by using the SPL bit
3. Write "1" to the SLEEPDEEP bit in the Cortex-M4F System Control Register.
4. Execute the WFI instruction or the WFE instruction.

■ Return from STOP mode

The CPU returns from STOP mode in one of the following situations.

● Return due to reset

If a reset (INITX pin input reset or low voltage detection reset) occurs, the CPU switches to high speed CR run mode regardless of the clock mode.

In STOP mode, since the software watchdog reset, the hardware watchdog reset, the clock supervisor reset and the anomalous frequency detection reset are not available, these resets cannot be used to make the CPU return from STOP mode.

● Return due to interrupt

On receiving a request for an effective interrupt, which can be an NMI interrupt, an external interrupt, a USB wakeup interrupt, a watch counter interrupt or a low voltage detection interrupt, in STOP mode, the CPU returns from STOP mode and transits to a RUN mode corresponding to the clock mode indicated in the RCM[2:0] bits in the System Clock Mode Status Register (SCM_STR).

Table 3-9 Operation mode after return from STOP mode due to interrupt

	Status of master clock before transition to STOP mode				
	RCM = 000 (High speed CR oscillator)	RCM = 001 (Main oscillator)	RCM = 010 (PLL oscillator)	RCM = 100 (Low speed CR oscillator)	RCM = 101 (Sub oscillator)
Operation mode after return due to interrupt	High speed CR run mode	Main run mode	PLL run mode	Low speed CR run mode	Sub run mode

● Oscillation stabilization wait at return

On returning from STOP mode due to a reset, the CPU waits for the stabilization of high speed CR clock oscillation and that of low speed CR clock oscillation. If the CPU returns from STOP mode due to an interrupt, the oscillation stabilization wait changes according to the master clock that had been used before the CPU transited to STOP mode. Table 3-10 shows the relationship between the oscillation stabilization wait and the master clock.

Table 3-10 Oscillation stabilization wait at return from STOP mode due to interrupt

		Status of master clock before transition to STOP mode				
		RCM = 000 (High speed CR oscillator)	RCM = 001 (Main oscillator)	RCM = 010 (PLL oscillator)	RCM = 100 (Low speed CR oscillator)	RCM = 101 (Sub oscillator)
Oscillation stabilization wait at return due to interrupt	High speed CR clock	Yes			No	
	Main clock	MOSCE = 0 : Yes MOSCE = 1 : No	Yes	PINC = 0 : Yes PINC = 1 : No	No	
	Main PLL clock	No	PLLE = 0 : No PLLE = 1 : Yes	Yes	No	
	Low speed CR clock	Yes				
	Sub clock	SOSCE = 0 : No SOSCE = 1 : Yes	SOSCE = 0 : No SOSCE = 1 : Yes	SOSCE = 0 : No SOSCE = 1 : Yes	SOSCE = 0 : No SOSCE = 1 : Yes	Yes

● Built-in regulator voltage stabilization wait at return

The CPU automatically secures a voltage stabilization wait time (a few hundred μs) for the built-in regulator before returning from STOP mode. After the voltage stabilization wait time has lapsed, the CPU executes the return operation.

<Notes>

- If the priority of an interrupt used for making the CPU return from STOP mode is not set to a level for making the CPU return from STOP mode, after an interrupt has been generated, the clock returns from STOP mode, but the CPU does not return from STOP mode and continues stopping. Therefore, always set the priority of an interrupt used for making the CPU return from STOP mode to a level that can make the CPU return from STOP mode.
- Before making the CPU transit to STOP mode, ensure that no factor for returning from STOP mode shown in Table 3-4 has been set. If such factor has been set, clear that factor.
- If the CPU transits to STOP mode during debugging, as the clock supply to the CPU stops, the CPU cannot return to a RUN mode by using the ICE. Use a reset or an interrupt to make the CPU return to a RUN mode.
- Before making the CPU transit to STOP mode, ensure that the Flash memory automatic algorithm has terminated.

4. Examples of Procedure for Setting Standby Mode

This section provides examples of procedure for setting a standby mode.

Figure 4-1 Example of procedure for setting main timer mode

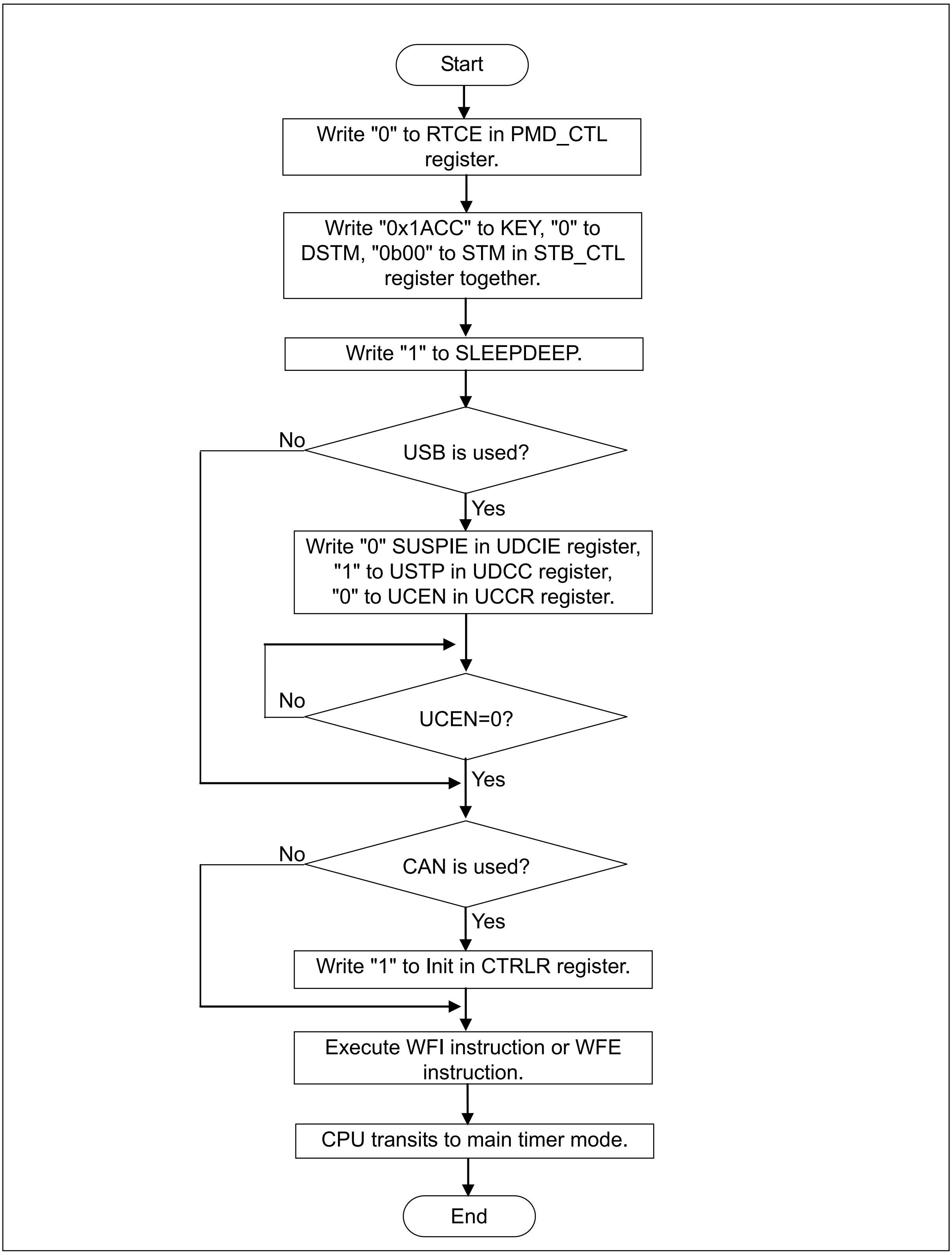
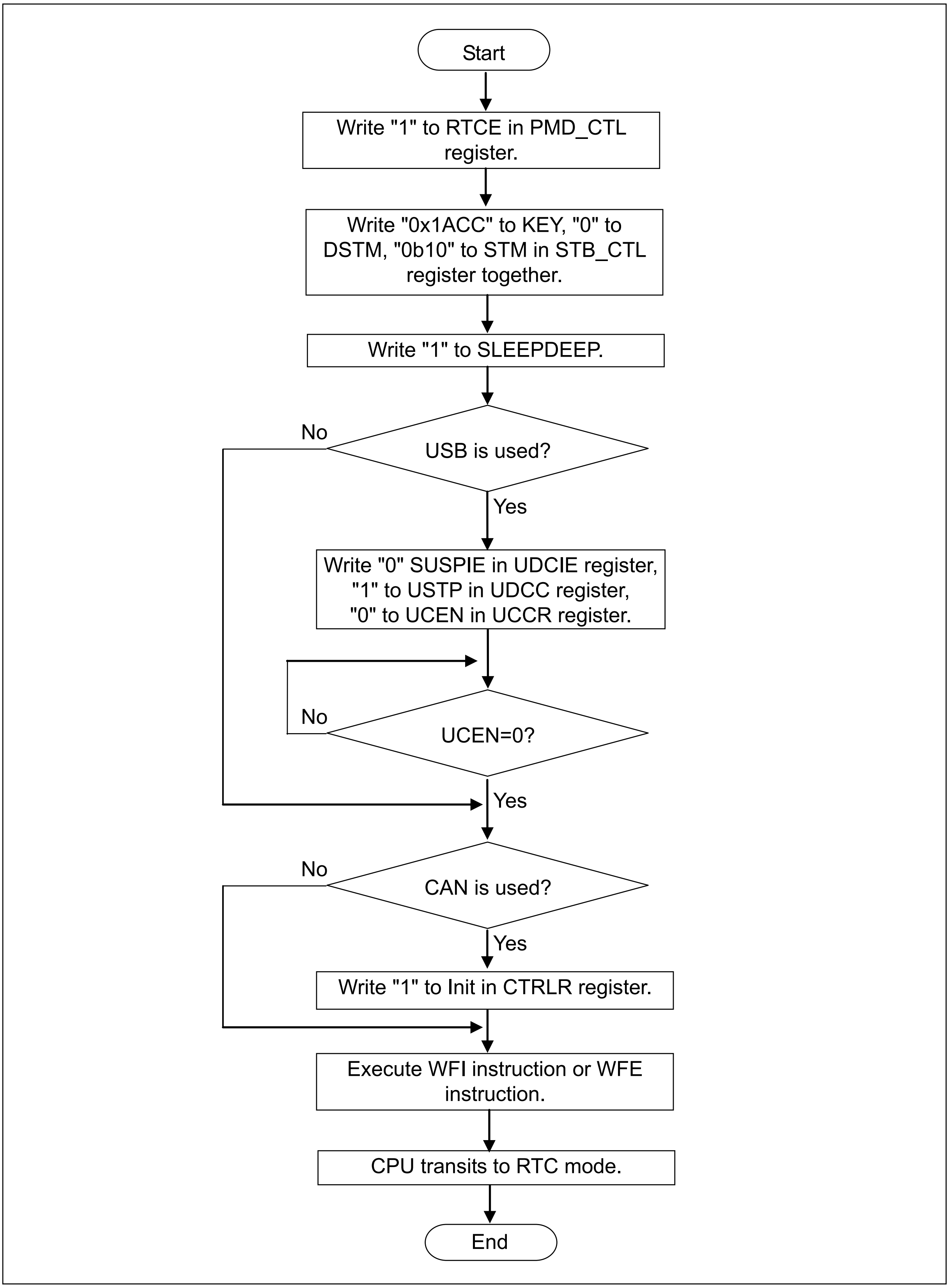

Start
Write "0" to RTCE in PMD_CTL register.
Write "0x1ACC" to KEY, "0" to DSTM, "0b00" to STM in STB_CTL register together.
Write "1" to SLEEPDEEP.
USB is used?
No
Yes
Write "0" SUSPIE in UDCIE register, "1" to USTP in UDCC register, "0" to UCEN in UCCR register.
UCEN=0?
No
Yes
CAN is used?
No
Yes
Write "1" to Init in CTRLR register.
Execute WFI instruction or WFE instruction.
CPU transits to main timer mode.
End

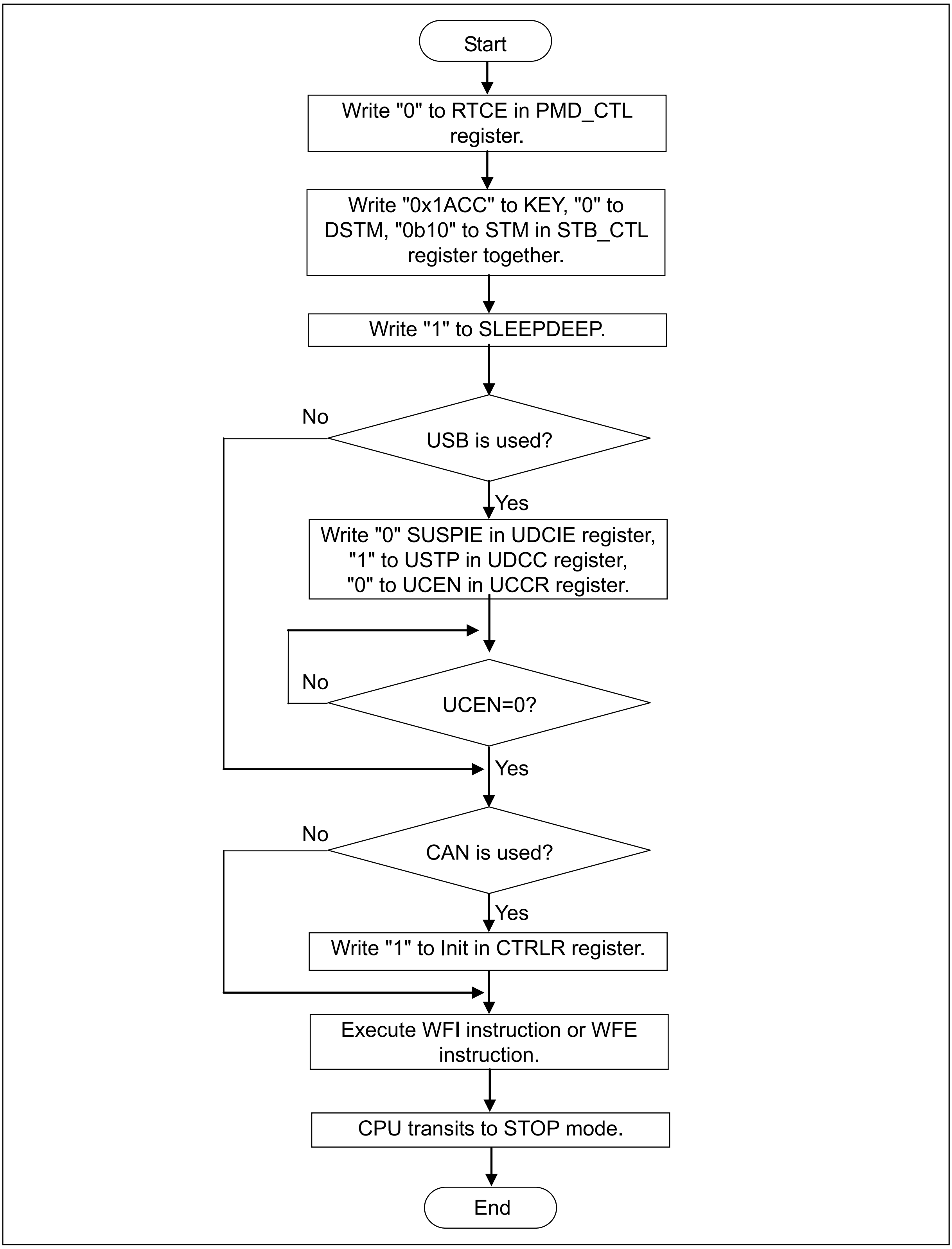
Figure 4-2 Procedure for setting RTC mode (with main clock selected as master clock)


Start
Write "1" to RTCE in PMD_CTL register.
Write "0x1ACC" to KEY, "0" to DSTM, "0b10" to STM in STB_CTL register together.
Write "1" to SLEEPDEEP.
No
USB is used?
Yes
Write "0" SUSPIE in UDCIE register, "1" to USTP in UDCC register, "0" to UCEN in UCCR register.
No
UCEN=0?
Yes
No
CAN is used?
Yes
Write "1" to Init in CTRLR register.
Execute WFI instruction or WFE instruction.
CPU transits to RTC mode.
End

<Notes>

- Before making the CPU transit to RTC mode, ensure that the Flash memory automatic algorithm has terminated.
- Writing "1" to the RTCE bit in the RTC Mode Control Register (PMD_CTL) is effective only if the SORDY bit in the System Clock Mode Status Register (SCM_STR) is "1".

Figure 4-3 Procedure for setting STOP mode (with main clock selected as master clock)


Start
Write "0" to RTCE in PMD_CTL register.
Write "0x1ACC" to KEY, "0" to DSTM, "0b10" to STM in STB_CTL register together.
Write "1" to SLEEPDEEP.
USB is used?
No
Yes
Write "0" SUSPIE in UDCIE register, "1" to USTP in UDCC register, "0" to UCEN in UCCR register.
UCEN=0?
No
Yes
CAN is used?
No
Yes
Write "1" to Init in CTRLR register.
Execute WFI instruction or WFE instruction.
CPU transits to STOP mode.
End

<Note>

Before making the CPU transit to STOP mode, ensure that the Flash memory automatic algorithm has terminated.

5. Operations in Deep Standby Modes

This section explains the operations in deep standby modes.
There are two deep standby modes: deep standby RTC mode and deep standby stop mode.

■ Clock operation states in deep standby mode

The table below shows the respective states of the oscillator clocks, CPU clock, AHB bus clock and APB bus clocks in deep standby RTC mode and deep standby stop mode.

Table 5-1 Clock operation states in deep standby mode

	Deep standby RTC mode	Deep standby stop mode
High speed CR clock	Stopped	Stopped
Main clock		
Main PLL clock		
Low speed CR clock		
Sub clock	Operating	
USB PLL clock	Stopped	
CPU clock		
AHB bus clock		
APB0 bus clock		
APB1 bus clock		
APB2 bus clock		

■ Factors for returning from deep standby mode

This section shows factors for returning from deep standby RTC mode and deep standby stop mode.

Table 5-2 Factors for returning from deep standby mode

	Deep standby RTC mode	Deep standby stop mode
Factor for returning from deep standby mode	・ INITX pin input reset ・ Low voltage detection reset	・ INITX pin input reset ・ Low voltage detection reset
	・ Low voltage detection interrupt ・ RTC interrupt	・ Low voltage detection interrupt
	・ WKUP pin input	・ WKUP pin input

<Note>

After the CPU has returned from a deep standby mode, the interrupt factor is retained. However, since NVIC is initialized by the deep standby transition reset, no interrupt processing is to be executed.

■ Internal power supply status and reset status in deep standby mode

This section shows the power supply status of each function in deep standby mode and the reset status on a deep standby transition reset.

Table 5-3 Internal power supply status and reset status in deep standby mode

	Power supply status	Reset status
CPU	Off	Initialized
On-chip Flash memory	Off	*1
SRAM0/1	Off	Data not retained
SRAM2	Off *2	*3
RTC	On	Not initialized
Low voltage detection circuit	On	Not initialized
GPIO	On	Partly initialized *4
Deep standby control block	On	Not initialized
Peripherals other the above	Off	Initialize

*1: Data in the on-chip Flash memory is retained.

*2: Data in SRAM2 can be retained.
If the setting for retaining data in on-chip SRAM is done, the power supply for SRAM2 is turned on.

*3: If the power supply is turned off, data in SRAM2 is not retained.
If the setting for retaining data in SRAM2 is done, data in SRAM2 is retained.

*4: All bits in the PFRx registers are initialized except for bit4:0 in the PFR0 register. bit4:0 in the PFR0 register, PCRx, DPDRx, PDIRx, PDORx, ADE, EPFRx, SPSR and PZRx registers are not initialized.

5.1. Operations in Deep Standby RTC Mode

In deep standby RTC mode, all oscillators stop except for the sub oscillator. All functions stop operating except for the RTC and the Low Voltage Detection Circuit. The power supply for the RTC, the Low Voltage Detection Circuit, CPUs excluding GPIO, on-chip Flash memory, SRAM0 /1/2* and peripherals are turned off inside the chip.

■ Functions of deep standby RTC mode

● CPU and on-chip memory

In deep standby RTC mode, the CPU clock supplied to the CPU, the AHB bus clock supplied to the on-chip memory and the DMA controller stop, and the power supply for the CPU, the on-chip Flash memory and SRAM0/1/2* is turned off. Data in the registers of the CPU and that in SRAM0/1/2 is not retained*. Data in the on-chip memory is retained. In addition, the debug function stops and its power supply is turned off.

*: Data in SRAM2 can be retained.
If the setting for retaining data in SRAM2 is done, the power supply for SRAM2 is turned on.

● Peripherals

In deep standby RTC mode, all APB bus clocks stop, and the power supply for all resources, except for the RTC, the Low Voltage Detection Circuit and the GPIO, is turned off.

● RTC

The RTC is not affected by deep standby RTC mode. It continues operating according to the settings set prior to transition to deep standby RTC mode.

● Oscillator clocks

Table 5-1 shows the status of each oscillator clock.

● Resets, interrupts and WKUP pin input

Resets, interrupts and WKUP pin input can be used for returning from deep standby RTC mode.

● Pin state

The SPL bit in the Standby Mode Control Register (STB_CTL) can control whether an external pin switches to a GPIO in deep standby RTC mode or changes to high impedance state.

■ Procedure for setting deep standby RTC mode

Execute the following procedure to make the CPU transit to deep standby RTC mode.

1. With "1" written to the SORDY bit in the System Clock Mode Status Register (SCM_STR), write "1" to the RTCE bit in the RTC Mode Control Register (PMD_CTL).
2. Write "0x1ACC", "1" and "0b10" to the KEY bits, DTSM bit and STM bits in the Standby Mode Control Register (STB_CTL) respectively. Set the state of each pin in deep standby RTC mode by using the SPL bit.
3. Write "1" to the SLEEPDEEP bit in the Cortex-M4F System Control Register.
4. Execute the WFI instruction or the WFE instruction.

■ Return from deep standby RTC mode

The CPU returns from deep standby RTC mode in one of the following situations.

● Return due to resets, interrupts and WKUP pin input

If a reset (INITX pin input reset or low voltage detection reset) occurs, or the CPU receives a request for an effective RTC interrupt, for an effective low voltage detection interrupt or for WKUP pin input, the CPU returns from deep standby RTC mode, and regardless of the clock mode, switches to high speed CR run mode on a deep standby transition reset.

In deep standby RTC mode, since the software watchdog reset, the hardware watchdog reset, the clock supervisor reset and the anomalous frequency detection reset are not available, these resets cannot be used to make the CPU return from deep standby RTC mode.

● Oscillation stabilization wait at return

On returning from deep standby RTC mode, regardless of the return factor, the CPU waits for the stabilization of high speed CR clock oscillation and that of low speed CR clock oscillation.

● Built-in regulator voltage stabilization wait at return

The CPU automatically secures a voltage stabilization wait time (a few hundred μs) for the built-in regulator before returning from deep standby RTC mode. After the voltage stabilization wait time has lapsed, the CPU executes the return operation.

<Notes>

- Before making the CPU transit to deep standby RTC mode, ensure that no factor for returning from deep standby RTC mode shown in Table 5-2 has been set. If such factor has been set, clear that factor.
- If the CPU transits to deep standby RTC mode during debugging, as the clock supply to the CPU stops, the CPU cannot return to a RUN mode by using the ICE. Use a reset, an interrupt or WKUP pin input to make the CPU return to a RUN mode.
- Before making the CPU transit to deep standby RTC mode, ensure that the Flash memory automatic algorithm has terminated.

5.2. Operations in Deep Standby Stop Mode

In deep standby stop mode, all oscillators stop. All functions stop operating except for the Low Voltage Detection Circuit. The power supply for the RTC, the Low Voltage Detection Circuit, CPUs excluding GPIO, on-chip Flash memory, SRAM0/1/2* and peripherals are turned off inside the chip.

■ Functions of deep standby stop mode

● CPU and on-chip memory

In deep standby stop mode, the CPU clock supplied to the CPU, the AHB bus clock supplied to the on-chip memory and the DMA controller stop, and the power supply for the CPU, the on-chip Flash memory and SRAM0/1/2* is turned off. Data in the registers of the CPU and that in SRAM0/1/2 is not retained*. Data in the on-chip memory is retained. In addition, the debug function stops and its power supply is turned off.

*: Data in SRAM2 can be retained.
If the setting for retaining data in SRAM2 is done, the power supply for SRAM2 is turned on.

● Peripherals

In deep standby stop mode, all APB bus clocks stop, and the power supply for all resources, except for the RTC, the Low Voltage Detection Circuit and the GPIO, is turned off.

● Oscillator clocks

All oscillator clocks stop.

● Resets and WKUP pin input

Resets and WKUP pin input can be used for returning from deep standby stop mode.

● Pin state

The SPL bit in the Standby Mode Control Register (STB_CTL) can control whether an external pin switches to a GPIO in deep standby stop mode or changes to high impedance state.

■ Procedure for setting deep standby stop mode

Execute the following procedure to make the CPU transit to deep standby stop mode.

1. Write "0" to the RTCE bit in the RTC Mode Control Register (PMD_CTL).
2. Write "0x1ACC", "1" and "0b10" to the KEY bits, DTSM bit and STM bits in the Standby Mode Control Register (STB_CTL) respectively. Set the state of each pin in deep standby stop mode by using the SPL bit.
3. Write "1" to the SLEEPDEEP bit in the Cortex-M4F System Control Register.
4. Execute the WFI instruction or the WFE instruction.

■ Return from deep standby stop mode

The CPU returns from deep standby stop mode in one of the following situations.

● Return due to resets, interrupts and WKUP pin input

If a reset (INITX pin input reset or low voltage detection reset) occurs, or the CPU receives a request for an effective low voltage detection interrupt or for WKUP pin input, the CPU returns from deep standby stop mode, and regardless of the clock mode, switches to high speed CR run mode on a deep standby transition reset.

In deep standby stop mode, since the software watchdog reset, the hardware watchdog reset, the clock supervisor reset and the anomalous frequency detection reset are not available, these resets cannot be used to make the CPU return from deep standby stop mode.

● Oscillation stabilization wait at return

On returning from deep standby RTC mode, regardless of the return factor, the CPU waits for the stabilization of high speed CR clock oscillation and that of low speed CR clock oscillation.

● Built-in regulator voltage stabilization wait at return

The CPU automatically secures a voltage stabilization wait time (a few hundred μs) for the built-in regulator before returning from deep standby stop mode. After the voltage stabilization wait time has lapsed, the CPU executes the return operation.

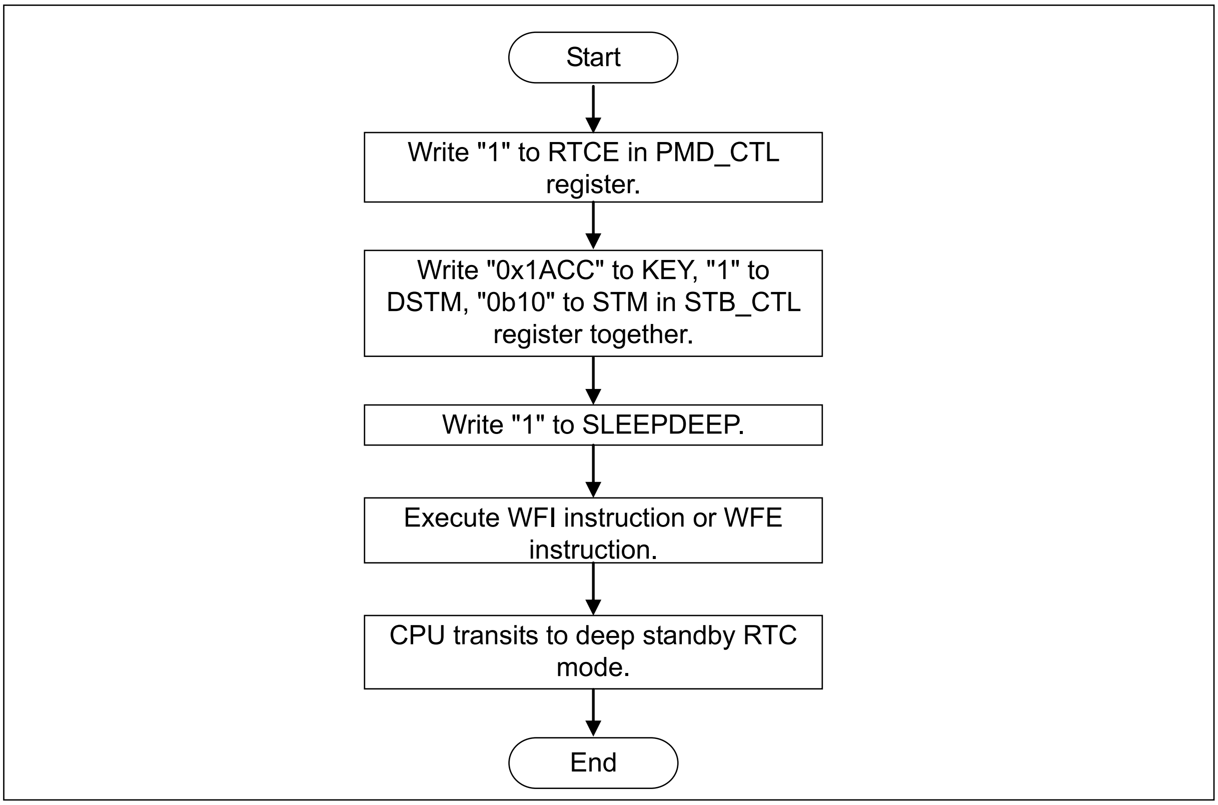
<Notes>

- Before making the CPU transit to deep standby stop mode, ensure that no factor for returning from deep standby stop mode shown in Table 5-2 has been set. If such factor has been set, clear that factor.
- If the CPU transits to deep standby stop mode during debugging, as the clock supply to the CPU stops, the CPU cannot return to a RUN mode by using the ICE. Use a reset, an interrupt or WKUP pin input to make the CPU return to a RUN mode.
- Before making the CPU transit to deep standby stop mode, ensure that the Flash memory automatic algorithm has terminated.

6. Examples of Procedure for Setting Deep Standby Mode

This section provides examples of procedure for setting a deep standby mode.

Figure 6-1 Example of procedure for setting deep standby RTC mode

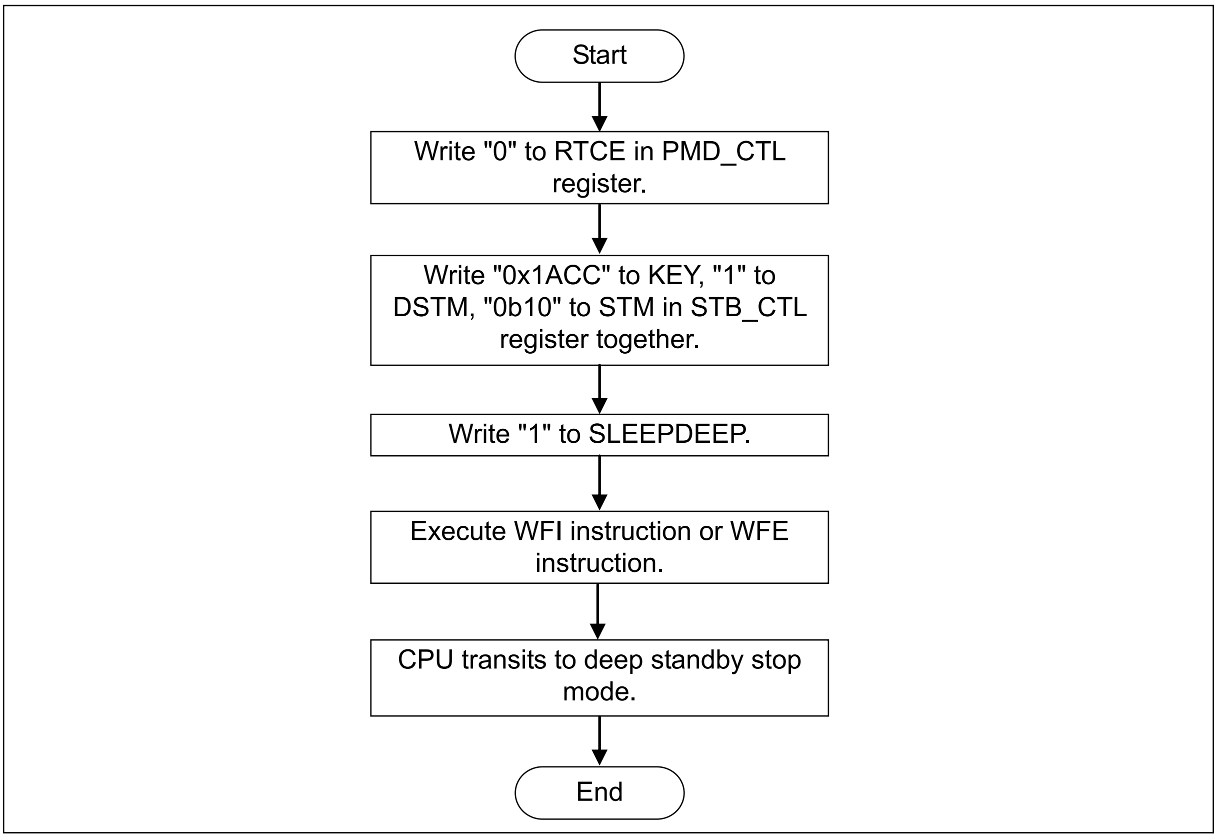


Start
Write "1" to RTCE in PMD_CTL register.
Write "0x1ACC" to KEY, "1" to DSTM, "0b10" to STM in STB_CTL register together.
Write "1" to SLEEPDEEP.
Execute WFI instruction or WFE instruction.
CPU transits to deep standby RTC mode.
End

<Notes>

- Before making the CPU transit to deep standby RTC mode, ensure that the Flash memory automatic algorithm has terminated.
- Writing "1" to the RTCE bit in the RTC Mode Control Register (PMD_CTL) is effective only if the SORDY bit in the System Clock Mode Status Register (SCM_STR) is "1".

Figure 6-2 Example of procedure for setting deep standby stop mode


Start
Write "0" to RTCE in PMD_CTL register.
Write "0x1ACC" to KEY, "1" to DSTM, "0b10" to STM in STB_CTL register together.
Write "1" to SLEEPDEEP.
Execute WFI instruction or WFE instruction.
CPU transits to deep standby stop mode.
End

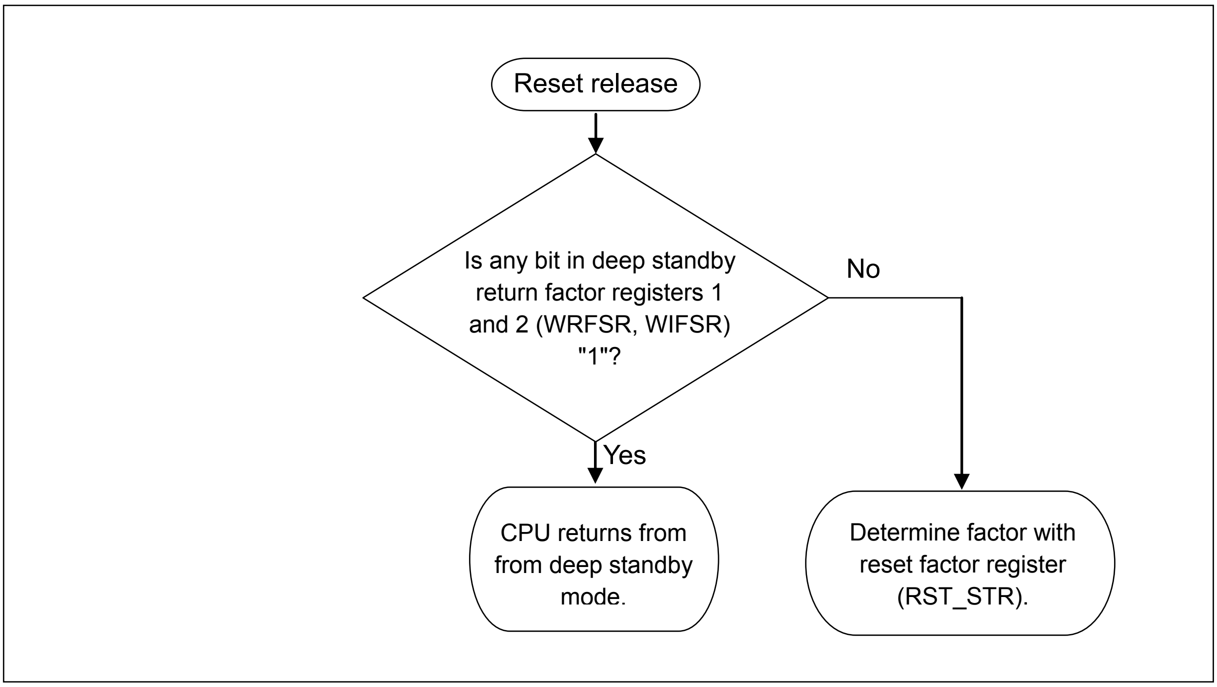
<Note>

Before making the CPU transit to deep standby stop mode, ensure that the Flash memory automatic algorithm has terminated.

7. Procedure for Determining Factor for Returning from Deep Standby Mode

Figure 7-1 shows an example of procedure for determining the factor for returning from a deep standby mode.

Figure 7-1 Procedure for determining the factor for returning from a deep standby mode


Reset release
Is any bit in deep standby return factor registers 1 and 2 (WRFSR, WIFSR) "1"?
No
Yes
CPU returns from from deep standby mode.
Determine factor with reset factor register (RST_STR).

<Note>

After the CPU has transited to a deep standby mode, the power supply for the CPU is turned off on a deep standby transition reset. Therefore, after the CPU has returned from a deep standby mode, the value of the reset factor register (RST_STR) becomes invalid.

8. List of Low Power Consumption Mode Registers

This section explains the configuration and functions of registers used in low power consumption mode.

■ List of low power consumption mode registers

Abbreviation	Register name	Reference
STB_CTL	Standby Mode Control Register	8.1

● Registers of deep standby control block

Abbreviation	Register name	Reference
RCK_CTL	Sub Clock Supply Control Register	8.2
PMD_CTL	RTC Mode Control Register	8.3
WRFSR	Deep Standby Return Factor Register 1	8.4
WIFSR	Deep Standby Return Factor Register 2	8.5
WIER	Deep Standby Return Enable Register	8.6
WILVR	WKUP Pin Input Level Register	8.7
DSRAMR	Deep Standby RAM Retention Register	8.8
BUR01 to 16	Backup Register 01 to 16	8.9

<Notes>

- For details of the System Clock Mode Control Register (SCM_CTL), see Chapter "Clock".
- The power supply for registers of the deep standby control block is not turned off in any deep standby mode.

8.1. Standby Mode Control Register (STB_CTL)

The Standby Mode Control Register controls standby modes and deep standby modes. The value of the SPL bit, DSTM bit or STM bit is effective only when it is written at the same time as "0x1ACC" is written to the KEY bits.

bit	31	16	15	8
Field	KEY		Reserved	
Attribute	R/W		-	
Initial value	0x0000		0x00	

bit	7	6	5	4	3	2	1	0
Field	Reserved			SPL	Reserved	DSTM	STM	
Attribute	-			R/W	-	R/W	R/W	
Initial value	000			0	0	0	00	

[bit31:16] KEY: Standby mode control write control bits

These bits release the write control of the SPL bit, DSTM bit and STM bit.

- When "0x1ACC" is written to these bits
  Writing to the SPL bit, DSTM bit or STM bit is effective.
- When a value other than "0x1ACC" is written to these bits
  Writing tothe SPL bit, DSTM bit or STM bit is not effective.
- These bits always read "0x0000".

[bit15:5] Reserved: Reserved bits

These bits always read "0x00".
Writing a value to these bits has no effect on operation.

[bit4] SPL: Standby pin level setting bit

This bit sets the state of a pin in TIMER mode, RTC mode, STOP mode, deep standby RTC mode, deep standby stop mode.

bit	Description
0	Holds the state of a pin in TIMER mode, RTC mode and STOP mode, and switches a pin to a GPIO in deep standby RTC mode and deep standby stop mode. [initial value]
1	Sets the state of a pin to high impedance in TIMER mode, RTC mode, STOP mode, deep standby RTC mode, deep standby stop mode.

[bit3] Reserved: Reserved bit

This bit always reads "0".
Writing a value to this bit has no effect on operation.

[bit2] DSTM: Deep standby mode select bit

This bit selects whether the CPU transits to a standby mode or a deep standby mode.

[bit1:0] STM: Standby mode select bits

These bits, together with the DSTM bit in this register and the RTCE bit in the RTC Mode Control Register (PMD_CTL), select a standby mode to which the CPU transits from one of the following modes: TIMER mode, RTC mode, STOP mode, deep standby RTC mode or deep standby stop mode.

DSTM	STM		PMD_CTL: RTCE	Description
	bit1	bit0		
0	0	0	0	TIMER mode [initial value]
0	0	0	1	Setting is prohibited.
0	0	1	0	Setting is prohibited.
0	0	1	1	Setting is prohibited.
0	1	0	0	STOP mode
0	1	0	1	RTC mode
0	1	1	0	Setting is prohibited.
0	1	1	1	Setting is prohibited.
1	0	0	0	Setting is prohibited.
1	0	0	1	Setting is prohibited.
1	0	1	0	Setting is prohibited.
1	0	1	1	Setting is prohibited.
1	1	0	0	Deep standby stop mode
1	1	0	1	Deep standby RTC mode
1	1	1	0	Setting is prohibited.
1	1	1	1	Setting is prohibited.

<Note>

The value of the SPL bit, DSTM bit or STM bit is effective only when it is written at the same time as "0x1ACC" is written to the KEY bits. If a value other than "0x1ACC" is written to the KEY bits, the values written to the SPL bit, DSTM bit and STM bit become ineffective.

8.2. Sub Clock Supply Control Register (RCK_CTL)

The Sub Clock Supply Control Register controls clock supply to the RTC.
Power consumption can be reduced by stopping the clock supply to the RTC if it is not in use.

bit	7	6	5	4	3	2	1	0
Field	Reserved							RTCCKE
Attribute	-							R/W
Initial value	0000000							1

[bit7:1] Reserved: Reserved bits

These bits always read "0b000000".
Writing a value to these bits has no effect on operation.

[bit0] RTCCKE: RTC clock control bit

This bit controls the sub clock supplied to the RTC macro.

bit	Description
0	The sub clock is not supplied to the RTC macro.
1	The sub clock is supplied to the RTC macro. [initial value]

8.3. RTC Mode Control Register (PMD_CTL)

The RTC Mode Control Register selects whether the CPU transits to either RTC mode or STOP mode, or to either deep standby RTC mode or deep standby stop mode.

bit	7	6	5	4	3	2	1	0
Field	Reserved							RTCE
Attribute	-							R/W
Initial value	0000000							0

[bit7:1] Reserved: Reserved bits

These bits always read "0b0000000".
Writing a value to these bits has no effect on operation.

[bit0] RTCE: RTC mode control bit

This bit selects whether the CPU transits to STOP mode, deep standby stop mode, or to RTC mode, deep standby RTC mode.

bit	Description
0	STOP mode or deep standby stop mode [initial value]
1	RTC mode or deep standby RTC mode

Standby mode is selected when DSTM bit is "0" and deep standby mode is selected when DSTM bit is "1".

<Notes>

- This register is not initialized by the software reset or the deep standby transition reset.
- Writing "1" to the RTCE bit is effective only if the SORDY bit in the System Clock Mode Status Register (SCM_STR) is "1".
- If the RTCE bit is "1", the sub oscillator is enabled, regardless of the setting of the SOSCE bit in the System Clock Mode Control Register (SCM_CTL) and that of the SORDY bit in the System Clock Mode Status Register (SCM_STR).

8.4. Deep Standby Return Factor Register 1 (WRFSR)

The Deep Standby Return Factor Register 1 indicates the return factors of the low voltage detection reset and the INITX pin input reset that have occurred in a deep standby mode.

bit	7	6	5	4	3	2	1	0
Field	Reserved						WLVDH	WINITX
Attribute	-						R	R
Initial value	000000						0	0

[bit7:2] Reserved: Reserved bits

These bits always read "0b000000".
Writing a value to these bits has no effect on operation.

[bit1] WLVHD: Low voltage detection reset return bit

This bit indicates the CPU has returned from a deep standby mode due to the low voltage detection reset.

bit	Description
0	The CPU has not returned from a deep standby mode due to the low voltage detection reset. [initial value]
1	The CPU has returned from a deep standby mode due to the low voltage detection reset.

[bit0] WINITX: INITX pin input reset return bit

This bit indicates the CPU has returned from a deep standby mode due to the INITX pin input reset.

bit	Description
0	The CPU has not returned from a deep standby mode due to the INITX pin input reset. [initial value]
1	The CPU has returned from a deep standby mode due to the INITX pin input reset.

<Notes>

- This register is initialized by the power-on reset and the low voltage detection reset. It is not initialized by any reset other than the two mentioned before. In addition, reading this register clears all its bits.
- This register can be set only in a deep standby mode.

8.5. Deep Standby Return Factor Register 2 (WIFSR)

The Deep Standby Return Factor Register 2 indicates the return factors of the WKUPx pin input, the low voltage detection (LVD) interrupt and the RTC interrupt that have occurred in a deep standby mode.

bit	15	14	13	12	11	10	9	8
Field	Reserved							
Attribute	-							
Initial value	0x00							

bit	7	6	5	4	3	2	1	0
Field	WUI5	WUI4	WUI3	WUI2	WUI1	WUI0	WLVDI	WRTCI
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	0	0	0	0	0	0

[bit15:8] Reserved: Reserved bits

These bits always read "0x00".
Writing a value to these bits has no effect on operation.

[bit7:2] WUI5 to WUI0: WKUPx pin input return bits

These bits indicate the CPU has returned from a deep standby mode due to the WKUPx pin input.

bit	Description
0	The CPU has not returned from a deep standby mode due to the WKUPx pin input. [initial value]
1	The CPU has returned from a deep standby mode due to the WKUPx pin input.

[bit1] WLVDI: LVD interrupt return bit

This bit indicates the CPU has returned from a deep standby mode due to the LVD interrupt.

bit	Description
0	The CPU has not returned from a deep standby mode due to the LVD interrupt. [initial value]
1	The CPU has returned from a deep standby mode due to the LVD interrupt.

[bit0] WRTCI: RTC interrupt return bit

This bit indicates the CPU has returned from a deep standby mode due to the RTC interrupt.

bit	Description
0	The CPU has not returned from a deep standby mode due to the RTC interrupt. [initial value]
1	The CPU has returned from a deep standby mode due to the RTC interrupt.

<Notes>

- This register is initialized by the power-on reset and the low voltage detection reset. It is not initialized by any reset other than the two mentioned before. In addition, reading this register clears all its bits.
- This register can be set only in a deep standby mode.

8.6. Deep Standby Return Enable Register (WIER)

The Deep Standby Return Enable Register enables the CPU to return from a deep standby mode due to the WKUPx pin input, the low voltage detection (LVD) interrupt and the RTC interrupt that have occurred in a deep standby mode.

bit	15	14	13	12	11	10	9	8
Field	Reserved							
Attribute	-							
Initial value	0x00							

bit	7	6	5	4	3	2	1	0
Field	WUI5E	WUI4E	WUI3E	WUI2E	WUI1E	Reserved	WLVDE	WRTCE
Attribute	R/W	R/W	R/W	R/W	R/W	-	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

[bit15:8] Reserved: Reserved bits

These bits always read "0x00".
Writing a value to these bits has no effect on operation.

[bit7:3] WUI5E to WUI1E: WKUPx pin input return enable bits

These bits disable or enable the CPU to return from a deep standby mode due to the WKUPx pin input.

bit	Description
0	Disables the CPU to return from a deep standby mode due to the WKUPx pin input. [initial value]
1	Enables the CPU to return from a deep standby mode due to the WKUPx pin input.

[bit2] Reserved: Reserved bit

This bit always reads "0".
Writing a value to this bit has no effect on operation.

[bit1] WLVDE: LVD interrupt return enable bit

This bit disables or enables the CPU to return from a deep standby mode due to the LVD interrupt.

bit	Description
0	Disables the CPU to return from a deep standby mode due to the LVD interrupt. [initial value]
1	Enables the CPU to return from a deep standby mode due to the LVD interrupt.

[bit0] WRTCE: RTC interrupt return enable bit

This bit disables or enables the CPU to return from a deep standby mode due to the RTC interrupt.

bit	Description
0	Disables the CPU to return from a deep standby mode due to the RTC interrupt. [initial value]
1	Enables the CPU to return from a deep standby mode due to the RTC interrupt.

<Notes>

- The CPU returning from a deep standby mode due to the WKUP0 pin input is always enabled.
- This register is not initialized by the deep standby transition reset.

8.7. WKUP Pin Input Level Register (WILVR)

The WKUP Pin Input Level Register selects the respective valid levels for the WKUP1 to WKUP5 pin inputs having occurred in a deep standby mode.

bit	7	6	5	4	3	2	1	0
Field	Reserved				WUI4LV	WUI3LV	WUI2LV	WUI1LV
Attribute	-				R/W	R/W	R/W	R/W
Initial value	0000				0	0	0	0

[bit7:4] Reserved: Reserved bits

These bits always read "0b000".
Writing a value to these bits has no effect on operation.

[bit3:0] WUI4LV to WUI1LV: WKUPx pin input level select bits

These bits select the respective valid levels for the WKUPx pin inputs.

bit	Description
0	A return request is made if the WKUPx pin input is "L" level. [initial value]
1	A return request is made if the WKUPx pin input is "H" level.

<Notes>

- "L" level is the only valid level of the WKUP0 pin input for making a return request.
  For example, with the WUI1LV bit set to "0", if "L" level is input to the WKUP1 pin, as soon as the CPU transits to a deep standby mode, it returns to a RUN mode.
- This register is not initialized by the deep standby transition reset.

8.8. Deep Standby RAM Retention Register (DSRAMR)

The Deep Standby RAM Retention Register controls the retention of data in SRAM2 in a deep standby mode.

bit	7	6	5	4	3	2	1	0
Field	Reserved						SRAMR	
Attribute	-						R/W	
Initial value	000000						00	

[bit7:2] Reserved: Reserved bits

These bits always read "0b000000".
Writing a value to these bits has no effect on operation.

[bit1:0] SRAMR: SRAM2 retention control bits

These bits control the retention of data in SRAM2 in a deep standby mode.

bit1	bit0	Description
0	0	Data in SRAM2 is not retained in a deep standby mode. [initial value]
0	1	Setting is prohibited.
1	0	Setting is prohibited.
1	1	Data in SRAM2 is retained in a deep standby mode.

<Note>

This register is initialized by the power-on reset and the low voltage detection reset. It is not initialized by any reset other than the two mentioned before.

8.9. Backup Registers 01 to 16 (BUR01 to BUR16)

The Backup Registers are general-purpose registers retaining values in a deep standby mode.

bit	31 24	23 16	15 8	7 0
Field	BUR04	BUR03	BUR02	BUR01
Attribute	R/W	R/W	R/W	R/W
Initial value	0x00	0x00	0x00	0x00

bit	31 24	23 16	15 8	7 0
Field	BUR08	BUR07	BUR06	BUR05
Attribute	R/W	R/W	R/W	R/W
Initial value	0x00	0x00	0x00	0x00

bit	31 24	23 16	15 8	7 0
Field	BUR12	BUR11	BUR10	BUR09
Attribute	R/W	R/W	R/W	R/W
Initial value	0x00	0x00	0x00	0x00

bit	31 24	23 16	15 8	7 0
Field	BUR16	BUR15	BUR14	BUR13
Attribute	R/W	R/W	R/W	R/W
Initial value	0x00	0x00	0x00	0x00

<Note>

This register is initialized by the power-on reset and the low voltage detection reset. It is not initialized by any reset other than the two mentioned before.

9. Usage Precautions

Note the following when using the low power consumption mode

For a pin shared between analog input and WKUP, if the ADE bit in the Analog Input Setting Register (ADE) is set to "1", the WKUPx pin input is blocked, even when the CPU returning from a deep standby mode due to the WKUPx pin input is enabled. To enable the CPU to return from a deep standby mode due to the WKUPx pin input, write "0" to the ADE bit in the Analog Input Setting Register (ADE) before making the CPU transit to a deep standby mode.

CHAPTER: VBAT Domain

This chapter explains the functions and operations of the VBAT power domain.

CODE: 9BFVBATPD-E01.0

1. Overview of VBAT Domain

The power consumed while the RTC is in operation can be reduced by using the VBAT power supply pin, which provides independent power supply for the RTC (calendar circuit) and the 32 kHz oscillator.

■ Configuration of power supply domain

This family consists of the following three power supply domains.

● CPU Domain

This domain consists of the following circuits.

- CPU
- On-chip Flash memory
- On-chip SRAM*
- Peripheral functions

This domain receives power supply from the VCC power supply pin. The power supply is cut off in deep standby RTC mode and deep standby stop mode.

*: SRAM0, SRAM1, and SRAM2 are provided and SRAM2 can be set to keep data.

● Always-ON Domain

This domain consists of the following circuits.

- On-chip regulator
- Power management circuit
- Port circuit
- Main oscillation circuit and I/O Port

This domain receives power supply from the VCC power supply pin.
The VCC power supply pin receives power from the system power supply (on-board regulator).

● VBAT Domain

This domain consists of the following circuits.

- RTC
- 32 kHz oscillation circuit
- Power-on circuit
- Backup registers
- Port circuit

This domain always receives power supply from the VBAT power supply pin.
The VBAT power supply pin receives power from a backup power supply (such as a cell) and the system power supply.

■ On-chip power gating

In deep standby RTC mode and deep standby stop mode, this family cuts off the power supply for the CPU Domain by using the power switch function built in the chip.

The Always-ON Domain keeps the power supply on even in deep standby RTC mode and deep standby stop mode.

For details of deep standby RTC mode and deep standby stop mode, see Chapter "Low Power Consumption Mode".

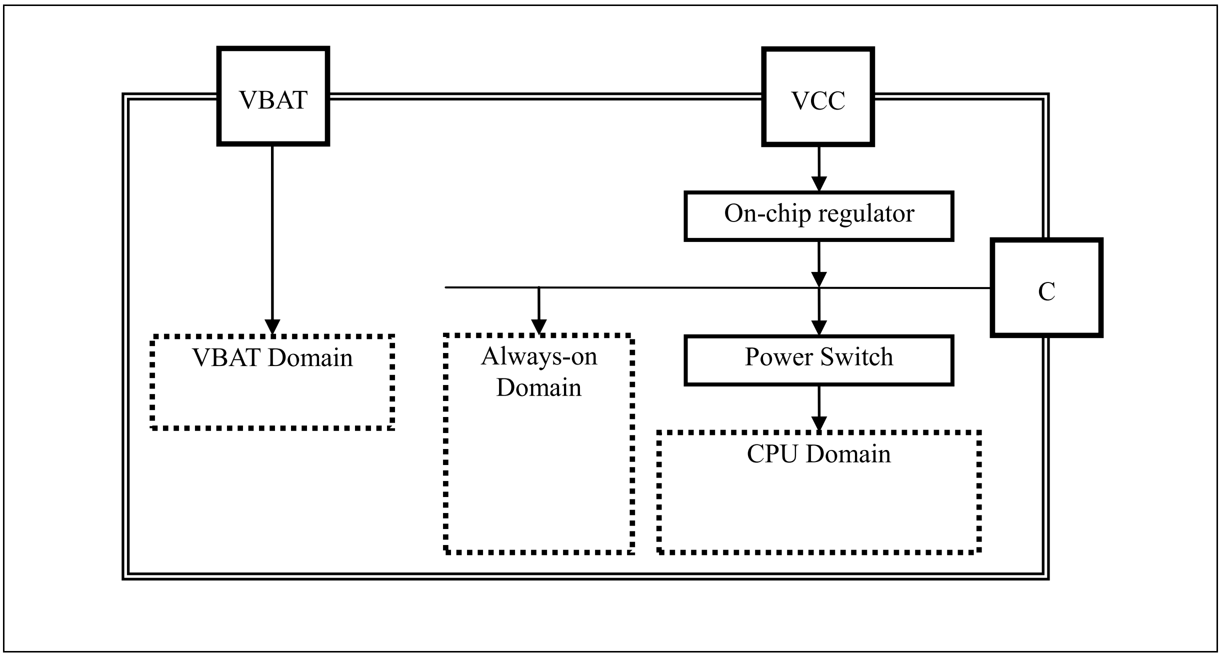
■ Off-chip power gating

If the system power supply supplying power to the VCC pin of this family is cut off, the power supply for the CPU Domain and that for the Always-on Domain are cut off.

In this situation, the power supply for the VBAT Domain can remain on if a backup power supply supplies power to the VBAT Domain.

The on-board regulator, which controls the system power supply with an alarm time set in the RTC or with a signal input from an external pin, can be turned on or off.

Figure 1-1 Power supply configuration of FM4 Family



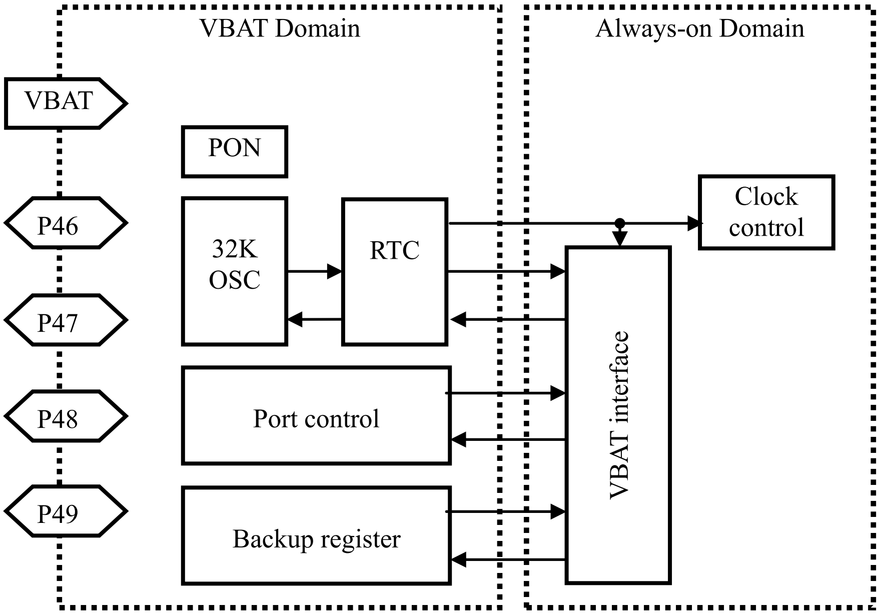
2. Configuration of VBAT Domain

This section explains the internal configuration of the VBAT Domain.

■ Internal configuration of VBAT Domain

Figure 2-1 shows the internal configuration of the VBAT Domain and the connection between the VBAT Domain and the Always-on Domain.

Figure 2-1 Internal configuration of VBAT Domain and connection between VBAT Domain and Always-on Domain



● RTC

This is a calendar circuit with the frequency compensation function.

● 32 kHz oscillation circuit (32K OSC)

This is an oscillation circuit that can be connected to a (32768 Hz) crystal oscillator for clocks.

● Power-on circuit (PON)

This detects the power-on of the VBAT Domain and generates the circuit initialization signal.

● Backup register

This 32-byte register retains data while power is being supplied to the VBAT power supply pin.

● VBAT I/O ports (P46 to 49, Port Control)

They are I/O ports driven by power supplied from the VBAT power supply pin.

The control circuit for the VBAT I/O ports is independent of the control circuit for other than I/O ports except P46 to P49.

2.1. Interfacing with Always-on Domain

This section explains the methods of interfacing the VBAT Domain withe the Always-on Domain.

■ Overview of interfacing

The VBAT Domain is driven by the 32 kHz oscillation circuit or a clock divided from PCLK.
Therefore, if an internal bus is directly connected to a register belonging to the VBAT Domain, a bus master such as the CPU is made to wait when accessing such register.
The FM4 Family has the following two mechanisms to prevent an access from being made to wait.

- A buffer is built in the Always-on Domain. An access from an internal bus is directed to that buffer.
- Data is transferred between the buffer of the Always-on Domain and the register of the VBAT Domain.

In the documents of the FM4 Family, data transfer operations between the buffer of the Always-on Domain and the register of the VBAT Domain are called as stated below.

- Recall: data transfer from the register of the VBAT Domain to the buffer of the Always-on Domain
- Save: data transfer from the buffer of the Always-on Domain to the register of the VBAT Domain

Since data written to the buffer is erased if the VCC power supply is off, save the data in the register of the VBAT Domain while the VCC power supply is on.
Immediately after the VCC power supply is turned on, the buffer is initialized by the reset signal of the Always-on Domain.
Therefore, before reading data from the buffer, execute a recall operation to restore data retained in the register while the VBAT power supply (backup power supply) was on to the buffer.
The calendar data of the RTC in the buffer is not automatically updated.
Before reading the time data from the buffer, execute the recall operation to transfer the time data saved in the register of the VBAT Domain to the buffer.

■ Types of interface circuit

There are three types of interface circuit as shown in Table 2-1.

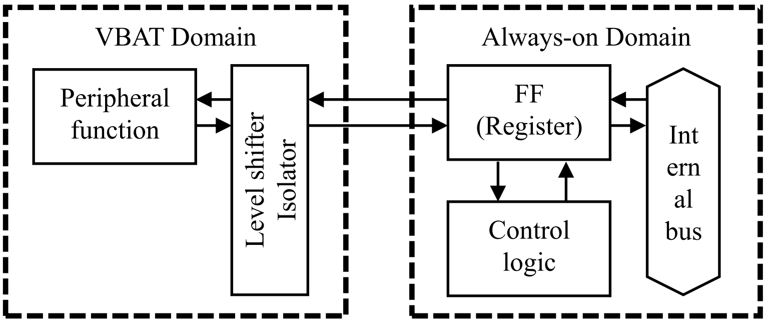
Table 2-1 Types of interface circuit

Circuit type	Always-on Domain	VBAT Domain	Transfer clock	Figure number
Type 1	FF available	FF unavailable	-	Figure 2-2
Type 2	FF available	FF available	32 kHz	Figure 2-3
Type 3	FF available	FF available	PCLK2 (divided)	Figure 2-4

A signal sent from the Always-on Domain to the VBAT Domain is clipped to the VSS by the level shifter and the isolator when the Always-on Domain is powered off.
This function enables the following operations to be executed when the Always-on Domain is powered off: continuing the operation of the calendar function, alarm function and timer function of the RTC, holding the pin states of the VBAT I/O ports, retaining data in the backup registers.

● Interface circuit type 1

Figure 2-2 Configuration of interface circuit type 1



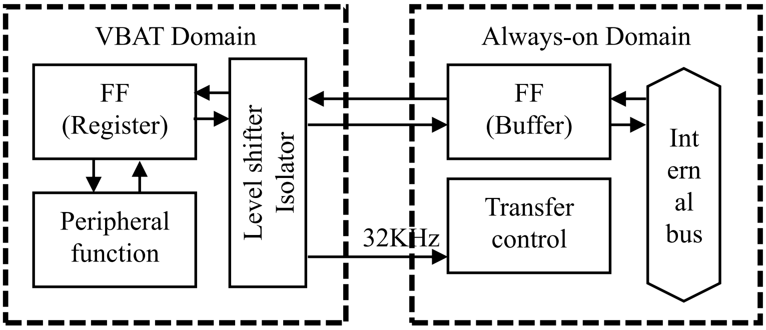
Use this circuit type if the register does not have to retain data when the VCC power supply is off.

Table 2-2 Behavior of register of interface circuit type 1

	Behavior of register
Initialization of register	Initialization of register by the reset signal of the Always-on Domain
Bus read	The state of the control circuit (Always-on Domain) and that of the peripheral circuit (VBAT Domain) can be read directly.
Bus write	The register directly affects the operations of the control circuit (Always-on Domain) and those of the peripheral circuit (VBAT Domain).

● Interface circuit type 2

Figure 2-3 Configuration of interface circuit type 2



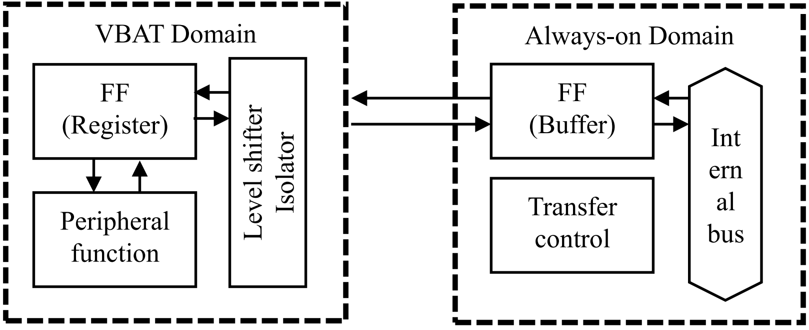
Use this circuit type if the register has to retain data even when the VCC power supply is off.

Table 2-3 Behavior of register of interface circuit type 2

	Behavior of register/buffer
Initialization of register	Initialization of register by the power-on signal of the VBAT Domain
Initialization of buffer	Initialization of register by the reset signal of the Always-on Domain
Bus read	Data in the buffer is read to the bus.
Bus write	Data is written to the buffer.
Recall operation	Data is transferred from the register to the buffer.
Save operation	Data is transferred from the buffer to the register.

● Interface circuit type 3

Figure 2-4 Configuration of interface circuit type 3



Use this circuit type if the register has to retain data even when the VCC power supply is off.

Table 2-4 Behavior of register of interface circuit type 3

	Behavior of register/buffer
Initialization of register	Initialization of register by the power-on signal of the VBAT Domain
Initialization of buffer	Initialization of register by the reset signal of the Always-on Domain
Bus read	Data in the buffer is read to the bus.
Bus write	Data is written to the buffer.
Recall operation	Data is transferred from the register to the buffer.
Save operation	Data is transferred from the buffer to the register.

The difference between Type 2 Circuit and Type 3 Circuit is the clock for the recall operation and save operation.

■ Circuit connected to interface circuit

The major circuits in the VBAT Domain are the RTC, the VBAT port and the buffer register.

The VBAT Domain executes the save operation or the recall operation on the buffer and registers of each circuit together.

(For details of the function of the WTCR20 Register in the following explanation, see "7.5 Control Register (WTCR20)" in chapter "RTC Count Block" in "Timer Part".

● RTC

The registers of the RTC consist of those listed in Table 2-5.

Table 2-5 Register List of RTC

No.	Register name	Reference	No.	Register name	Reference
1	WTSR	[RTCCAL]	2	WTMIR	[RTCCAL]
3	WTHR	[RTCCAL]	4	WTDR	[RTCCAL]
5	WTDW	[RTCCAL]	6	WTMOR	[RTCCAL]
7	WTYR	[RTCCAL]	8	ALMIR	[RTCCAL]
9	ALHR	[RTCCAL]	10	ALDR	[RTCCAL]
11	ALMOR	[RTCCAL]	12	ALYR	[RTCCAL]
13	WTTR0	[RTCCAL]	14	WTTR1	[RTCCAL]
15	WTTR2	[RTCCAL]	16	WTCR11	[RTCCAL]
17	WTCR10	[RTCCAL]	18	WTCR21	[RTCCAL]

For the function of each register, see [RTCCAL], which stands for Chapter "RTC Count Block" in "Timer Part".

The interface circuit type for registers No.1 to No.18 is type 2.

A save operation is started if "1" is written to bit1 in the WTCR20 Register. This save operation is called a CWRITE operation.

A recall operation is started if "1" is written to bit0 in the WTCR20 Register. This recall operation is called a CREAD operation.

The subclock is used as the transfer clock.

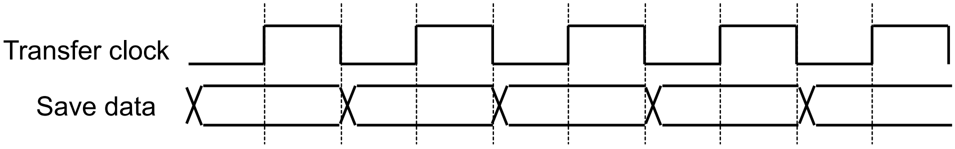
The RTC transfers 1 byte of data for one transfer clock.

In one CREAD/CWRITE operation, the RTC transfers data of registers in sequence from No.1 to No.18 shown in Table 2-5.

When the transfer in the CREAD operation ends, bit0 in the WTCR20 Register returns to "0". When the transfer in the CWRITE operation ends, bit1 in the WTCR20 Register returns to "0".

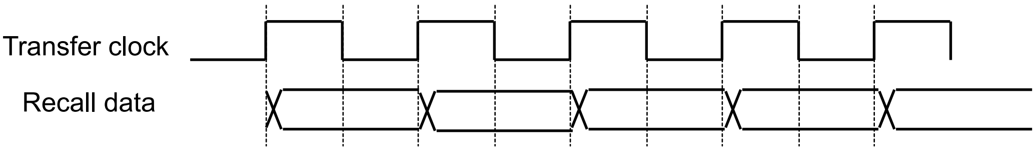
Since the registers are 18 bytes in total, the data transfer ends as 18 transfer clocks elapse.

- CWRITE operation waveform



The save data is output from the buffer at a falling edge of the transfer clock and is written to the register at a rising edge of the transfer clock.
Three transfer clocks are required for preprocessing before the start of a transfer and three transfer clocks are also required for postprocessing after the end of a transfer.

- CREAD operation waveform



The recall data is output from the register at a rising edge of the transfer clock and is written to the buffer at a falling edge of the transfer clock.
Three transfer clocks are required for preprocessing before the start of a transfer and three transfer clocks are also required for postprocessing after the end of a transfer.

● VBAT port circuit (VB_PORT)

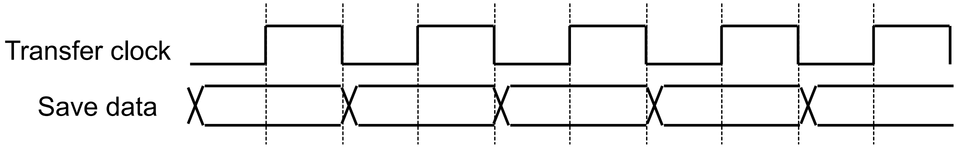
The registers of the VBAT port circuit consist of those listed in Table 2-6.

Table 2-6 Register list of VBAT port circuit

No.	Register name	Reference	No.	Register name	Reference
1	WTCAL0	[RTCCLK]	2	WTCAL1	[RTCCLK]
3	WTCALEN	[RTCCLK]	4	WTDIV	[RTCCLK]
5	WTDIVEN	[RTCCLK]	6	WTCALPRD	[RTCCLK]
7	WTCOSEL	[RTCCLK]	8	CCS	2.3 32 kHz Oscillation Circuit
9	CCB	2.3 32 kHz Oscillation Circuit	10	Reserved	
11	BOOST	2.3 32 kHz Oscillation Circuit	12	WTOSCCNT	2.3 32 kHz Oscillation Circuit
13	VBPFR	2.6 VBAT I/O Ports	14	VBPCR	2.6 VBAT I/O Ports
15	VBDDR	2.6 VBAT I/O Ports	16	VBPZR	2.6 VBAT I/O Ports
17	VBDOR	2.6 VBAT I/O Ports	18	VB_CLKDIV	2.3 32 kHz Oscillation Circuit
19	VDET	2.3 32 kHz Oscillation Circuit	20	EWKUP	2.3 32 kHz Oscillation Circuit
21	HIBRST	2.3 32 kHz Oscillation Circuit	22	VBDIR	2.6 VBAT I/O Ports

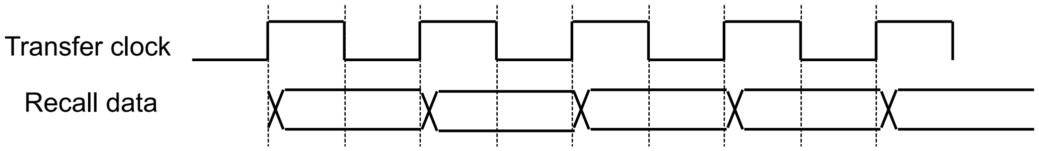
For the function of each register, see [RTCCLK], which stands for Chapter "RTC Clock Control Block" in "Timer Part", and "2.6 VBAT I/O Ports" and "2.3 32 kHz Oscillation Circuit" in this chapter.
The interface circuit type for registers No.1 to No.17 of the VBAT port circuit is type 3, and that for registers No.18 to No.22 of the VBAT port circuit is type 1.
A save operation is started if "1" is written to bit5 in the WTCR20 Register. This save operation is called a PWRITE operation.
A recall operation is started if "1" is written to bit4 in the WTCR20 Register. This recall operation is called a PREAD operation.
The transfer clock is created by dividing PCLK2 by the value of the VB_CLKDIV Register.
The RTC transfers 1 byte of data for one transfer clock.
In one PREAD/PWRITE operation, the RTC transfers data of registers in sequence from No.1 to No.18 shown in Table 2-6.
When the transfer in the PREAD operation ends, bit4 in the WTCR20 Register returns to "0". When the transfer in the PWRITE operation ends, bit5 in the WTCR20 Register returns to "0".
Since the registers of the VBAT port circuit are 17 bytes in size, the data transfer ends as 17 transfer clocks elapse.

- PWRITE operation waveform



The save data is output from the buffer at a falling edge of the transfer clock and is written to the register at a rising edge of the transfer clock.
One transfer clock is required for preprocessing before the start of a transfer.

- PREAD operation waveform

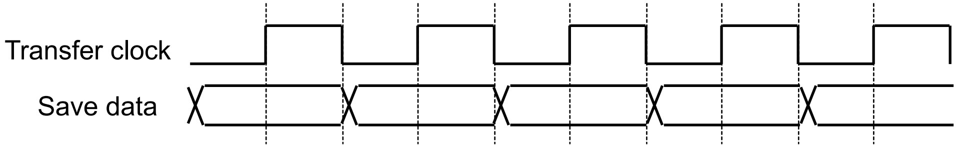


The recall data is output from the register at a rising edge of the transfer clock and is written to the buffer at a falling edge of the transfer clock.
One transfer clock is required for preprocessing before the start of a transfer and one transfer clock is also required for postprocessing after the end of a transfer.

● Backup registers

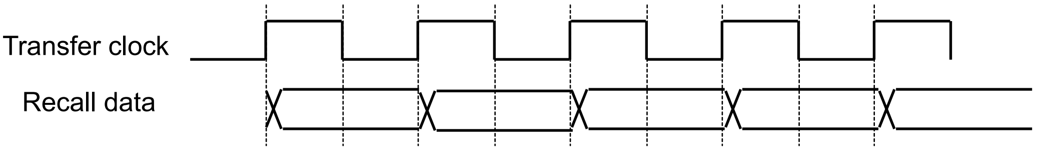
The backup registers refer to the 32-byte register area from BREG00 to BREG1F.
For the functions of the backup registers, see "2.5 Backup Registers".
The interface circuit type for the backup registers is type 3.
A save operation is started if "1" is written to bit3 in the WTCR20 Register. This save operation is called a BWRITE operation.
A recall operation is started if "1" is written to bit2 in the WTCR20 Register. This recall operation is called a BREAD operation.
The transfer clock is created by dividing PCLK2 by the value of the VB_CLKDIV Register.
The RTC transfers 1 byte of data for one transfer clock.
In one BREAD/BWRITE operation, the RTC transfers data of registers in sequence from BREG00 to BREG1F.
The RTC starts the transfer from BREG00. The transfer destination or the transfer source is shifted to the next register whenever one transfer clock elapses.
Since the backup register of the FM4 Family is 32 bytes in size, the data transfer ends as 32 transfer clocks elapse.

· BWRITE operation waveform



The save data is output from the buffer at a falling edge of the transfer clock and is written to the register at a rising edge of the transfer clock.
One transfer clock is required for preprocessing before the start of a transfer.

· BREAD operation waveform



The recall data is output from the register at a rising edge of the transfer clock and is written to the buffer at a falling edge of the transfer clock.
One transfer clock is required for preprocessing before the start of a transfer and one transfer clock is also required for postprocessing after the end of a transfer.

● Allowed transfer combination

Though it should be checked that the TRANS bit in the WTCR0 Register is "0" before the start of a recall operation or of a save operation, the transfers in a combination with the "o" mark in the following table can be executed simultaneously.

	CREAD	CWRITE	PREAD	PWRITE	BREAD	BWRITE
CREAD		×	×	×	o	o
CWRITE	×		×	×	o	o
PREAD	×	×		×	o	o
PWRITE	×	×	×		o	o
BREAD	o	o	o	o		×
BWRITE	o	o	o	o	×	

"o" indicates that the transfers in that combination can be executed simultaneously.
"×" indicates that the transfers in that combination cannot be executed simultaneously.

● Notes on description

In the peripheral manuals of the FM4 Family, a read access and a write access to a register of interface circuit "type 2" or of interface circuit "type 3" are defined as follows.
Read access: A recall operation is executed and then data in the buffer is read.
Write access: A recall operation is executed to update the entire buffer. Afterward, the part in the buffer corresponding to the data of the write access is replaced with such data, and then a save operation is executed.

● Usage Precautions

- Execute CREAD/CWRITE under the following frequency condition: PCLK2 (APB2 bus clock) ≥ 1 MHz
- Do not access the buffer during a save operation or a recall operation. It is not possible to identify whether correct data has been saved while accessing the buffer during a save operation, neither is it possible to identify whether correct data has been read while accessing the buffer during a recall operation.
- Set the VB_CLKDIV Register to a value that makes the transfer clock for PREAD, PWRITE, BREAD and BWRITE generated by dividing PCLK2 become 1 MHz or below.

2.2. RTC

The RTC of the FM4 Family is a calendar circuit with a 32 kHz frequency compensation function.

■ Overview of RTC functions

The RTC has the following functions.

- Clock function
- Alarm function
- Timer function
- Frequency compensation function

■ Configuration of RTC

For details of the functions of the RTC, see Chapter "RTC Count Block" in "Timer Part".

2.3. 32 kHz Oscillation Circuit

The 32 kHz oscillation circuit is an oscillation circuit exclusively for the crystal oscillator for the clock, and creates the subclock.

■ Overview of functions of 32 kHz oscillation circuit

The 32 kHz oscillation circuit has the following functions.

- Oscillation current switch function
- Oscillation boost function
- Clock generator cooperative operation function

● Oscillation current switch function

The amplifier circuit of the 32 kHz oscillation circuit is driven by a constant current source.
The current value of the constant current source can be controlled by the value of the CCS Register.

● Oscillation boost function

In the case of the crystal oscillator for the clock, it takes longer time for the oscillation frequency to stabilize.
The oscillation stabilization time can be shortened by increasing the current supplied to the amplifier circuit for a certain period of time after the start of oscillation.
During the period from the start of oscillation to the end of the oscillation boost time set in the BOOST Register, the current the constant current source supplies is the current value set in the CCB Register. After the above period has elapsed, the current the constant current source supplies switches to the current value set in the CCS Register.
If a current value larger than the one set in the CCS Register is set in the CCB Register, the oscillation boost function can work effectively.
If not using the oscillation boost function, set in the CCB Register a value same as the one set in the CCS Register.

● Clock generator cooperative operation function

The SOSCNTL bit in the WTOSCCNT Register enables or disables the cooperative operation between the 32 kHz oscillation circuit and the clock generator belonging to the CPU Domain.
With the cooperative operation enabled, the 32 kHz oscillation of this circuit stops when the CPU transits to stop mode or deep standby stop mode.
If the cooperative operation is disabled, the 32 kHz oscillation of this circuit does not stop regardless of the operation mode of the CPU.

<Notes>

- The 32 kHz oscillation circuit in the VBAT Domain does not have the oscillation stabilization wait function or the clock failure detection function. Enable the clock cooperative function and use the clock failure detection function of the CPU Domain.
- The appropriate amount of current flowing to the amplifier circuit varies depending on the characteristic (ESR) and load capacitance (CL) of the oscillator connected to the VBAT Domain. Select an appropriate amount of current by performing a matching evaluation between the VBAT Domain and the crystal oscillator.
- If using the RTC with a backup power supply for the VBAT Domain instead of the VCC power supply, disable the cooperative operation with the clock generator.
- After the 32 kHz oscillation of this circuit has started, do not update the CCB Register or the CCS Register.

■ Application of 32 kHz oscillation circuit

See "5 Procedure for Setting 32 kHz Clock" for details of application.

■ Registers used for 32 kHz oscillation circuit

bit	31 - 24	23 - 16	15 - 8	7 - 0	Initial value	Attribute
	Reserved	Reserved	Reserved	VB_CLKDIV	0x00000007	R/W
	Reserved	Reserved	Reserved	WTOSCCNT	0x00000001	R/W
	Reserved	Reserved	Reserved	CCS	0x00000008	R/W
	Reserved	Reserved	Reserved	CCB	0x00000010	R/W
	Reserved	Reserved	Reserved	BOOST	0x00000003	R/W
	Reserved	Reserved	Reserved	EWKUP	0x00000000	R/W
	Reserved	Reserved	Reserved	VDET	0x00000080	R/W
	Reserved	Reserved	Reserved	HIBRST	0x00000000	R/W

The interface circuit types for the above registers are type 1 and type 3.

The save operation and recall operation of the 32 kHz oscillation circuit are PWRITE and PREAD respectively.

2.4. Power-on circuit

The FM4 Family has a power-on circuit independent of the VCC power supply pin detecting the power-on of the VBAT Domain.

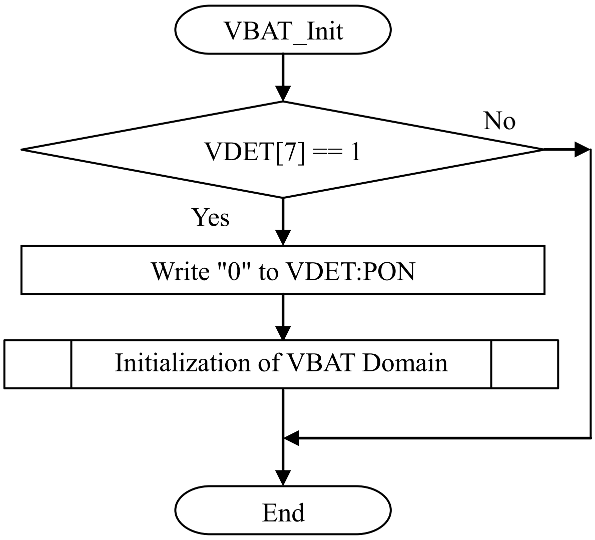
■ Overview of function of power-on circuit

The power-on circuit in the VBAT Domain has the following function.

VBAT power supply pin rising edge detection function

While the power-on circuit is outputting the power-on signal, bit7 in the VDET Register reads "1".
The power-on signal keeps being asserted until "0" is written to bit7 in the VDET Register.
The power-on signal and the value of bit7 in the VDET Register are not affected by turning on and off of the VCC power supply.
If the power-on circuit in the VBAT Domain is initialized according to the following flow, while a peripheral function is initialized at a VCC rising edge, the initialization of the VBAT Domain that is in operation is skipped and the RTC continues running.

Figure 2-5 Flow of Power-on Judgment and Initialization of VBAT Power Domain



<Notes>

- The power-on circuit in the VBAT Domain does not have the VBAT power supply pin low voltage detection function. (The FM4 Family does not have the VBAT power supply pin low voltage detection function.)
- While the power-on signal is being asserted, the values of the registers of the VBAT Domain (RTC, 32 kHz oscillation circuit, VBAT I/O port control circuit, backup registers) are fixed at their respective initial values. Before setting these circuits, write "0" to bit7 in the VDET Register to clear the power-on signal.

2.5. Backup Registers

The FM4 Family has 32-byte backup registers retaining data with the VBAT power supply.

■ Overview of function of backup registers

A backup register retains values written to it while power is being supplied to the VBAT power supply pin.
The backup register is reset by the power-on circuit immediately after the VBAT power supply has been turned on.
As the application of the VBAT power supply can be checked by reading the value of bit7 in the VDET Register, initialize the backup register with a program whenever necessary.

■ Configuration of backup register and method of accessing backup register

The interface circuit type for the backup register is type 3.
For details, see "2.1 Interfacing with Always-on Domain".
The data transfer between a backup register and a data retention register is a batch transfer of data of all areas.
Update data according to the following procedure.

1. Set the VB_CLKDIV Register to a value that makes the transfer clock become 1 MHz or below.
2. Recall (BREAD) data from the data retention register to the buffer register.
   If "1" is written to bit2 in the WTCR20 Register, the recall operation starts and bit7 in the WTCR10 Register becomes "1".
   If the recall operation ends, bit7 in the WTCR10 becomes "0".
3. Modify the content of the buffer register.
   The buffer register allows random read access and random write access.
4. Save data in the buffer register to the data retention register.
   If "1" is written to bit3 in the WTCR20 Register, the save operation starts and bit7 in the WTCR10 Register becomes "1".
   If the save (BWRITE) operation ends, bit7 becomes "0".

* If the power supply of the Always-on Domain is turned off, data of the buffer register is lost. Therefore, always transfer data retained in the buffer register to the data retention register by executing a save (BWRITE) operation.
* While transferring data to the data retention register (bit7 in the WTCR10 Register is "1"), do not access the buffer register.
* If the Always-on Domain has been reset during the data transfer or the VCC power supply is turned off, the integrity of the data of the data retention register cannot be guaranteed.

■ Details of backup registers

● List of backup registers

bit	31 - 24	23 - 16	15 - 8	7 - 0	Initial value	Attribute
	BREG03	BREG02	BREG01	BREG00	0x00000000	R/W
	BREG07	BREG06	BREG05	BREG04	0x00000000	R/W
	BREG0B	BREG0A	BREG09	BREG08	0x00000000	R/W
	BREG0F	BREG0E	BREG0D	BREG0C	0x00000000	R/W
	BREG13	BREG12	BREG11	BREG10	0x00000000	R/W
	BREG17	BREG16	BREG15	BREG14	0x00000000	R/W
	BREG1B	BREG1A	BREG19	BREG18	0x00000000	R/W
	BREG1F	BREG1E	BREG1D	BREG1C	0x00000000	R/W

The interface circuit type for the above registers is type 3.
The save operation and recall operation of the backup registers are BWRITE and BREAD respectively.
The backup registers retain data with the VBAT pin power supply.
They can be accessed by byte access, halfword access and word access.

2.6. VBAT I/O Ports

The FM4 Family has four I/O ports assigned to the VBAT Domain. These I/O ports (VBAT I/O ports) are controlled by the port control circuit (VBAT port control circuit) of the VBAT Domain, and continues operating even when the VCC power supply is turned off. The VBAT port control circuit is independent of the port control circuit explained in "CHAPTER I/O Port" in "FM4 Family PERIPHERAL MANUAL". The addresses of registers of the VBAT port control circuit are mapped to an area different the one to which the addresses of registers of the port control circuit are mapped.

■ Overview of function of VBAT I/O ports

The VBAT I/O ports keep operating as long as the VBAT power supply is turned on, even when the VCC power supply is turned off.

The VBAT I/O ports do not have the relocate function.

■ Configuration of VBAT I/O ports

The registers of the VBAT port control circuit select the I/O direction, and the function of an I/O port between GPIO port and peripheral function I/O pin.

For the configuration of the VBAT I/O ports, see Figure 2-1 in chapter "I/O Port". (For registers having the same function, substitute an actual register name for the one used in Figure 2-1.)

Table 2-7 shows a register list and explains the function of each register.

Table 2-7 Register list and register function

Register name	Function
VBPFR[5:4]	This is a register setting whether to use a VBAT I/O port as a special pin (for oscillation) or as a digital I/O pin.
VBPFR[3:0]	This is a register setting whether to use a VBAT I/O port as a GPIO port or as a peripheral function I/O pin.
VBPCR[3:0]	With a VBAT I/O port used as a digital input pin or as a digital bidirectional pin, this is a register setting whether to connect or disconnect the pull-up resistor of a VBAT I/O port.
VBDDR[3:0]	With a VBAT I/O port used as a GPIO port, this is a register setting whether to use that GPIO port as an input pin or as an output pin. (Note) If a VBAT I/O port is used as a peripheral function I/O pin, the setting of this register becomes invalid.
VBDIR[3:0]	This is a register reading the level of a VBAT I/O port. ・ If a VBAT I/O port is used as a digital input pin, this register reads the input level. ・ If a VBAT I/O port is used as a digital output pin, this register reads the output level. ・ If a VBAT I/O port is used as a special pin, this register always reads "0".
VBDOR[3:0]	With a VBAT I/O port used as a GPIO output pin, this is a register setting the output level. ・ If a bit in VBDOR[3:0] is set to "0", a GPIO output pin outputs "L" level. ・ If a bit in VBDOR[3:0] is set to "1", a GPIO output pin outputs "H" level. (Note) If a VBAT I/O port is used as a GPIO input pin or as a peripheral function I/O pin, the setting of this register becomes invalid.
VBPZR[1:0]	This is a register controlling the open drain of a VBAT I/O port. ・ If a VBAT I/O port outputs "L" level, I/O Port is set to "L" level output. (The pull-up resistor is disconnected regardless of the setting of the PCR Register.) ・ If a VBAT I/O port outputs "H" level, I/O Port is set to “Hi-Z”, and the open drain is controlled in pseudo manner. (The pull-up resistor is disconnected regardless of the setting of the PCR Register.) ・ If a VBAT I/O port is used as an input port, I/O Port is set to “Hi-Z”, and their I/O direction changes to the input direction. (The pull-up resistor is disconnected regardless of the setting of the PCR Register.)

The interface circuit type of the VBDIR Register is type 1. The interface circuit type of the other registers is type 3.

The save operation and recall operation of the 32 kHz oscillation circuit are PWRITE and PREAD respectively. (register of interface circuit type 3)

<Note>

The settings of the I/O Port Control Registers (PFR4[6:9], PCR4[6:9], DDR4[6:9], DIR4[6:9], DOR4[6:9], PZR4[6:9]) have no effect on the operations of the VBAT I/O ports.

■ Initial settings of VBAT I/O ports

Table 2-8 shows the respective initial states of the VBAT I/O ports.

Table 2-8 Initial states of VBAT I/O ports.

No.	Pin	Initially selected function
1	P46/X0A	This pin can be used as an oscillation pin. (The oscillation has stopped.) The digital input has been cut off and "0" has been input to this pin.
2	P47/X1A	This pin can be used as an oscillation pin. (The oscillation has stopped.) The digital input has been cut off and "0" has been input to this pin.
3	P48/VREGCTL	This is a digital input pin. The output is open drain.
4	P49/VWAKEUP	This is a digital input pin. The output is open drain.

The VBAT I/O ports remain in their respective states described in Table 2-8 while the VBAT power-on circuit is resetting the VBAT Domain.

■ Procedure for setting VBAT I/O ports

● In the case of using 32 kHz oscillation circuit

See "5 Procedure for Setting 32 kHz Clock" for different setting procedures.

● In the case of controlling hibernation

See "4 Hibernation Control" for the setting procedure as well as the procedure for setting I/O.

● In the case of using VBAT I/O port as GPIO port

For the setting method, refer to "CHAPTER I/O Port" in "FM4 Family PERIPHERAL MANUAL".
(For registers having the same function, substitute an actual register name for the one used in that chapter.)

■ Registers of VBAT I/O ports

● List of registers of VBAT I/O ports

bit 31 - 24	23 - 16	15 - 8	7 - 0	Initial value	Attribute
Reserved	Reserved	Reserved	VBPFR	0x0000001C	R/W
Reserved	Reserved	Reserved	VBPCR	0x00000000	R/W
Reserved	Reserved	Reserved	VBDDR	0x00000000	R/W
Reserved	Reserved	Reserved	VBDIR	0x000000XX	R/W
Reserved	Reserved	Reserved	VBDOR	0x0000000F	R/W
Reserved	Reserved	Reserved	VBPZR	0x00000003	R/W

■ Configuration of registers of VBAT I/O ports and method of accessing those registers

The interface circuit type for the VBAT I/O port registers is type 3.
For details, see "2.1 Interfacing with Always-on Domain".
The data transfer between a buffer register and a VBAT I/O port register is a batch transfer of data of all areas.
Update data according to the following procedure.

1. Set the VB_CLKDIV Register to a value that makes the transfer clock become 1 MHz or below.
2. Recall data from the VBAT I/O port retention register to the buffer register.
   If "1" is written to bit4 in the WTCR20 Register, the recall operation starts and bit7 in the WTCR10 Register becomes "1".
   If the recall operation ends, the TRANS bit becomes "0".
3. Modify the content of the buffer register.
   The buffer register allows random read access and random write access.
4. Save data in the buffer register to the VBAT I/O port register.
   If "1" is written to bit5 in the WTCR20 Register, the save operation starts and bit7 in the WTCR10 Register becomes "1".
   If the save operation ends, bit7 becomes "0".

- Modifying new data in the buffer register alone does not change the state of a VBAT I/O port pin.
  To change a VBAT I/O port register value (pin state), execute a save operation to transfer data in a buffer register corresponding to that VBAT I/O port register to that VBAT I/O port register.
- While transferring data to the data retention register (bit7 in the WTCR10 Register is "1"), do not access the buffer register.
- If the Always-on Domain has been reset during the data transfer or the VCC power supply is turned off, the integrity of the data of the data retention register cannot be guaranteed.

3. Chip Power Supply Control

This section explains details of applying and cutting off chip power supply.

■ Table of combinations of VCC power supply and VBAT power supply

Table 3-1 shows the respective states of the VCC power supply and the VBAT power supply.

Table 3-1 Combination of VCC power supply state and VBAT power supply state

	VBAT power supply on	VBAT power supply off
VCC power supply on	Normal operation	This combination is prohibited.
VCC power supply off	Only the VBAT Domain continues operating.	Stop of operation

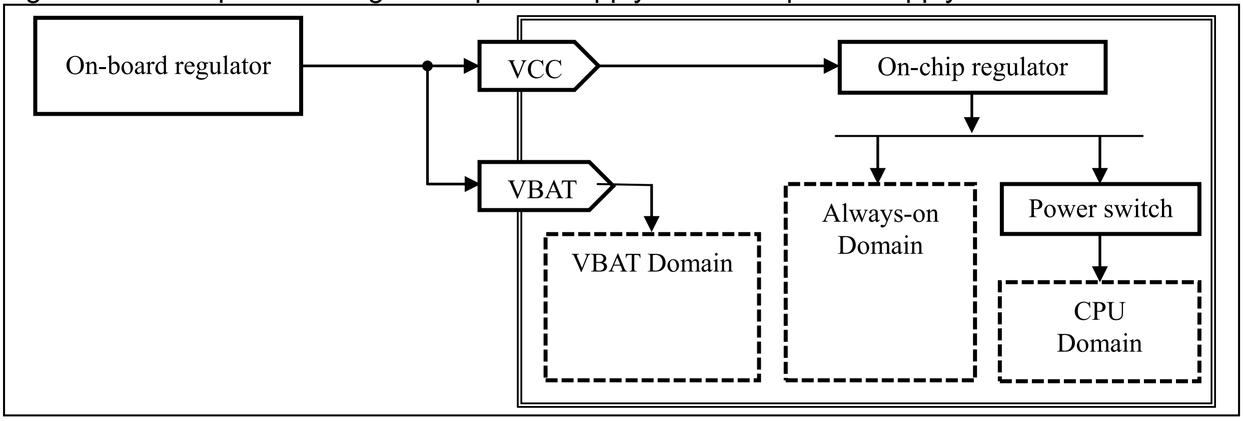
■ Driving VBAT power supply and VCC power supply with same power supply

● Transition of power supply state

If the VBAT power supply and the VCC power supply are driven by the same power supply, the chip power supply state transits between "normal operation" and "stop of operation" shown in .Table 3-1.

When driving the VBAT power supply and the VCC power supply with the same power supply, initialize the VBAT Domain whenever applying the VCC power supply.

Figure 3-1 Example of driving VBAT power supply with VCC power supply



If not using the backup power supply for the VBAT power supply, connect the VBAT power supply pin directly to the VCC power supply pin inside the chip.

If the VBAT power supply pin is directly connected to the VCC power supply pin, the prohibited combination of "VCC power supply on and VBAT power supply off" can be avoided.

■ Driving VBAT power supply with cell

● Transition of power supply state

Figure 3-2 shows how the state of power supply transits when a cell is used as the VBAT power supply. Figure 3-3 shows the respective waveforms of circuits.

bit7 in the VDET Register indicates whether the system power supply has been turned on for the first time. If the system power supply has been turned on for the first time, do the settings of the circuits in the VBAT Domain.

Figure 3-2 Transition of states with cell used as VBAT power supply

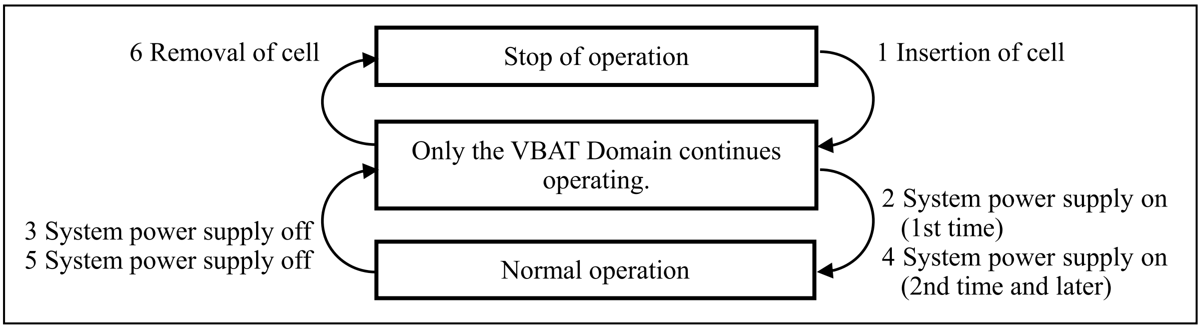
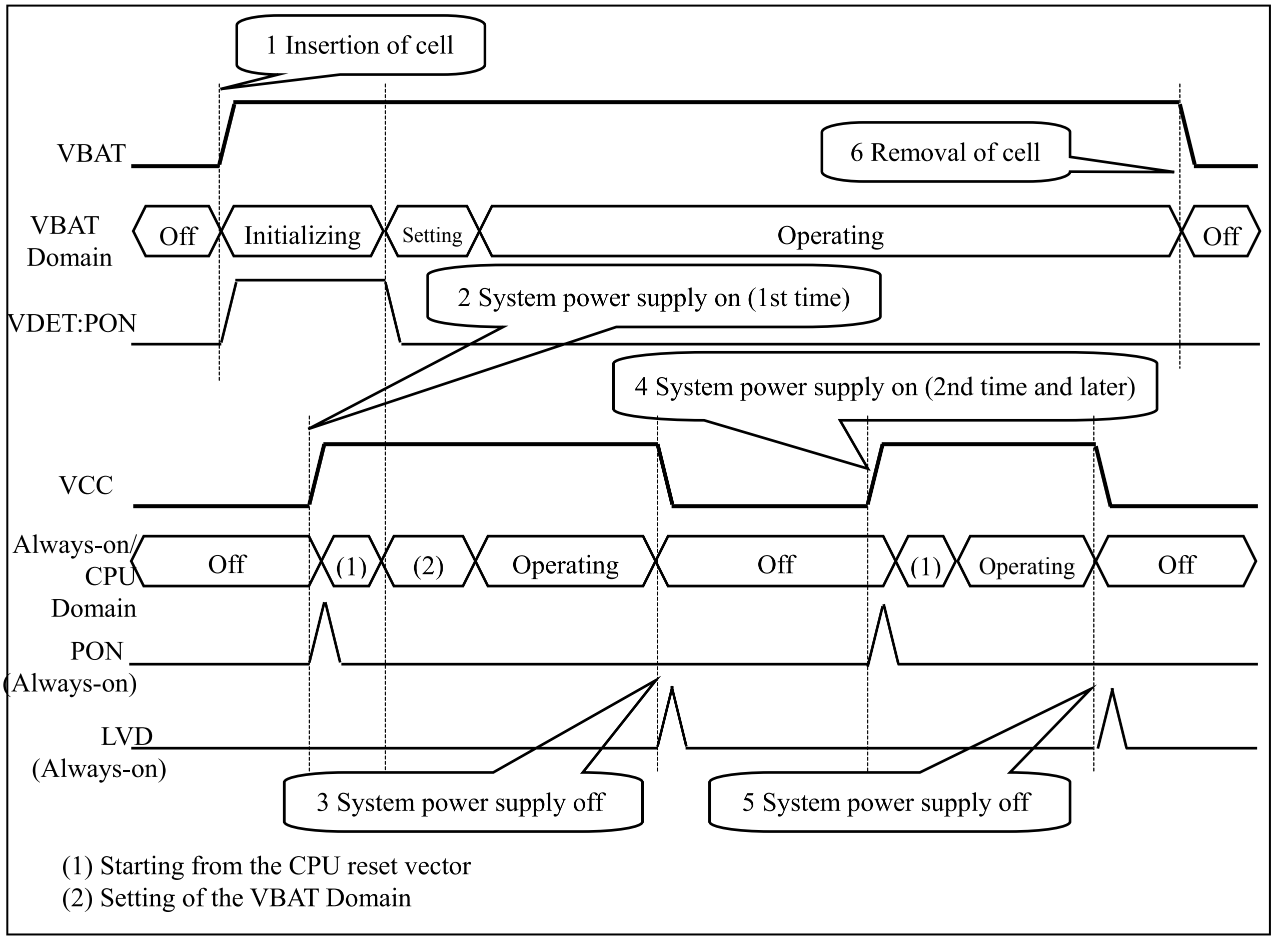

6 Removal of cell
Stop of operation
1 Insertion of cell
Only the VBAT Domain continues operating.
2 System power supply on (1st time)
4 System power supply on (2nd time and later)
3 System power supply off
5 System power supply off
Normal operation

Figure 3-3 Example of waveforms for using cell


1 Insertion of cell
VBAT
6 Removal of cell
VBAT Domain
Off
Initializing
Setting
Operating
Off
VDET:PON
2 System power supply on (1st time)
4 System power supply on (2nd time and later)
VCC
Always-on/ CPU Domain
Off
(1)
(2)
Operating
Off
(1)
Operating
Off
PON (Always-on)
LVD (Always-on)
3 System power supply off
5 System power supply off
(1) Starting from the CPU reset vector
(2) Setting of the VBAT Domain

● Examples of power supply configuration

Figure 3-4 Example of using primary cell as backup power supply

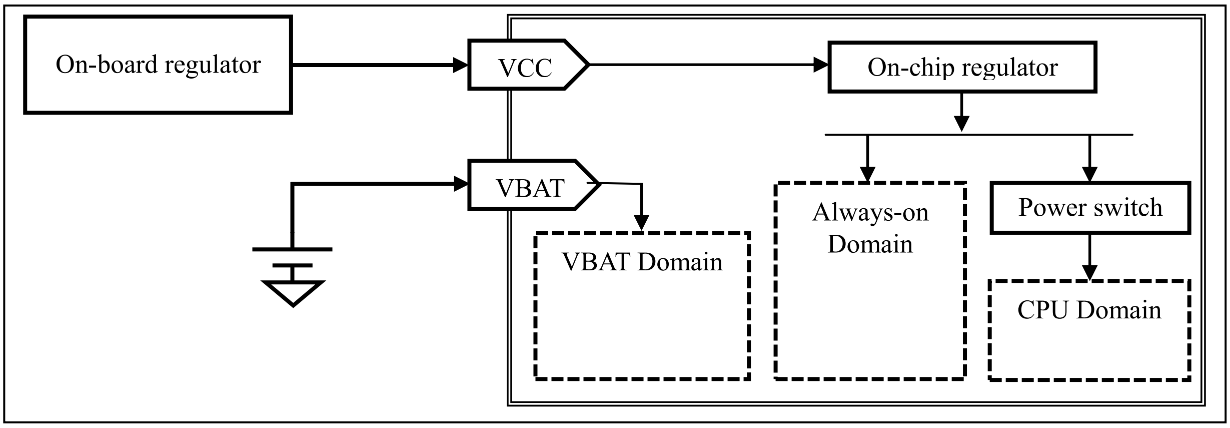
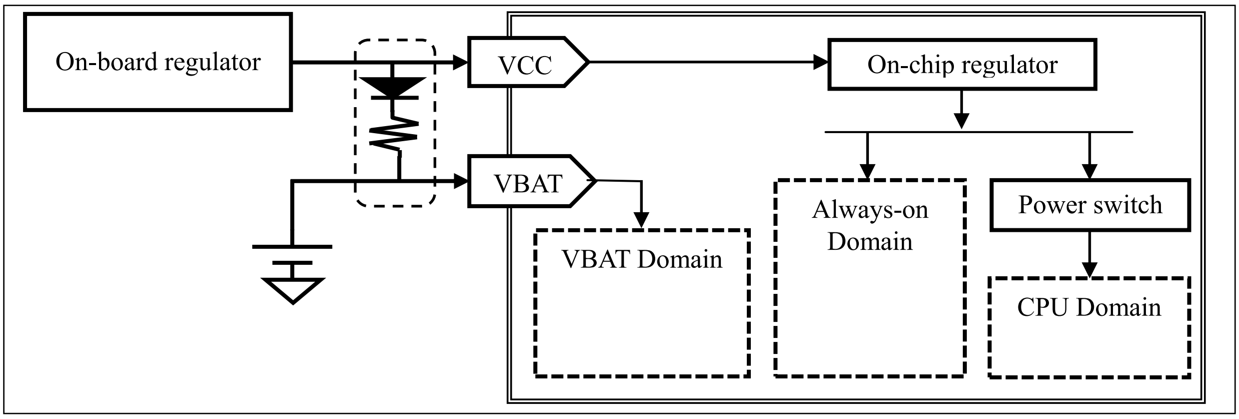


Figure 3-5 Example of using secondary cell as backup power supply



The diode and resistor inside the dotted line box trickle charges the secondary cell.
When setting the resistance, check whether the secondary cell used can be trickle charged, and the maximum current of trickle charging.
If the secondary cell used cannot be trickle charged, use it the same as a primary cell.

<Notes>

- Turn off the system power supply before inserting or removing a cell.
- If a primary cell is used as the backup power supply, it is not recommended to connect the system power supply to the backup power supply through a diode.

4. Hibernation Control

This section shows an example of circuit configuration for controlling off-chip power gating through the microcontroller and an example of the sequence of controlling off-chip gating through the microcontroller.

■ Overview of hibernation control

Hibernation control turns on or off the VCC power supply (for both Always-on Domain and CPU Domain) through the VBAT Domain by controlling the standby function of the on-board regulator through the VBAT Domain.

To execute hibernation control, supply the VBAT pin with a backup power supply other than the VCC power supply (system power supply).

Below are the two sources for making the microcontroller return from the hibernation state.

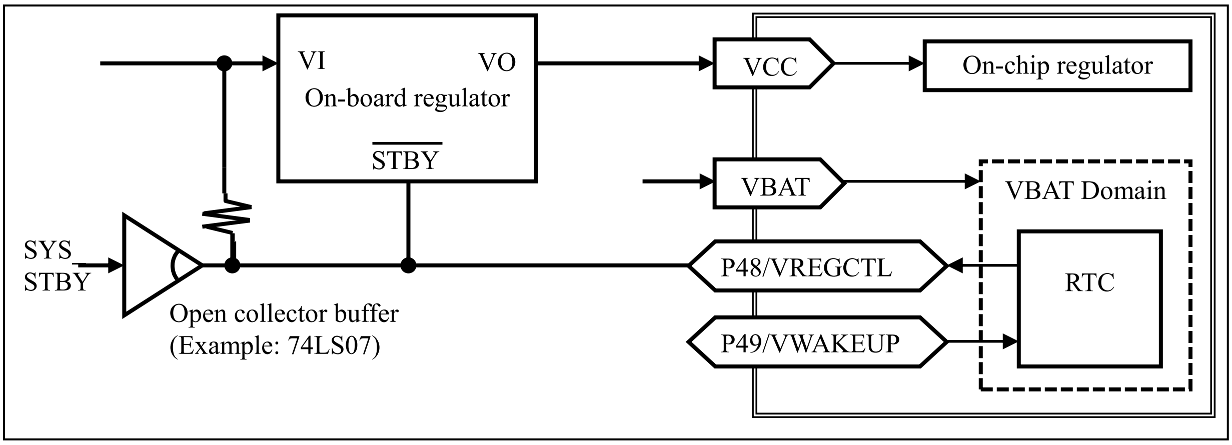
- Alarm interrupt of the RTC
- Wakeup request to the P49/VWAKEUP pin (A request is made at a rising edge.)

To use the alarm interrupt of the RTC to make the microcontroller return from the hibernation state, keep the 32 kHz oscillation operating even when the VCC power supply is off.

If the wakeup request is used to make the microcontroller return from the hibernation state, the 32 kHz oscillation can be stopped while the VCC power supply is off.

■ External connection examples of FM4 Family

Figure 4-1 Example of external connection with input voltage (VI) of on-board regulator lower than 5.5 V



Many on-board regulators enter the standby state when their "STBY" input is "L" level.

The P48/VREGCNTL pin of the VBAT I/O ports is 5 V tolerant and is a pseudo-open drain pin.

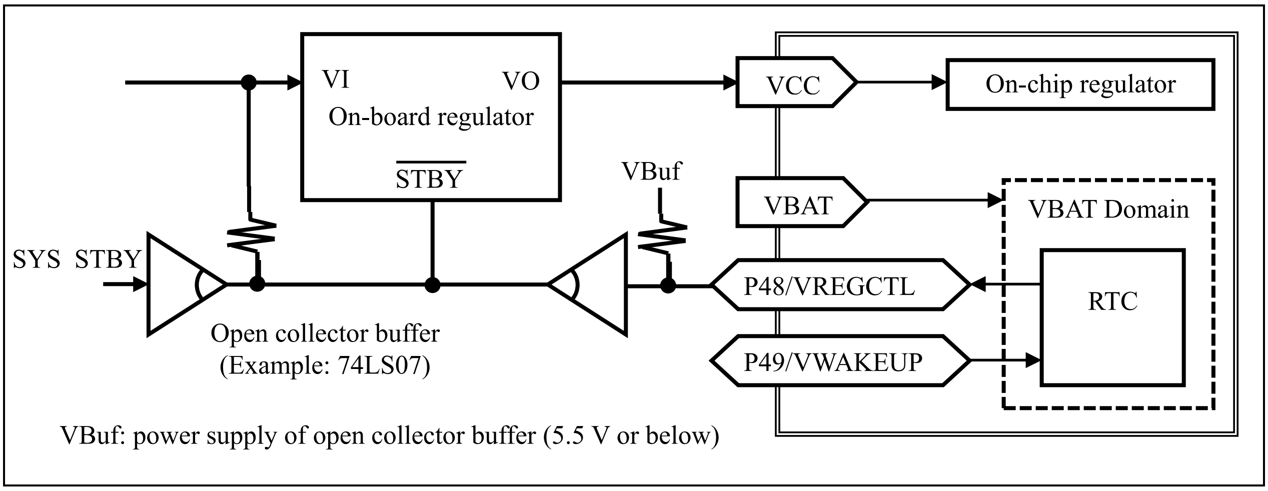
Connect the P48/VREGCNTL to the "STBY" input of the on-board regulator, and connect a pull-up resistor to the input voltage (VI) of the on-board regulator

Table 4-1 shows how the on-board regulator operates when the standby signal of the system is buffered by an open collector buffer, and the buffered standby signal and the P48/VREGCNTL are connected by a wired OR logic circuit.

Table 4-1 Operation of on-board regulator

SYS_STBY	VREGCTL = "L"	VREGCTL = "H"
"L"	Standby mode	Standby mode
"H"	Standby mode	Normal operation mode

Figure 4-2 Example of external connection with input voltage (VI) of on-board regulator higher than 5.5 V

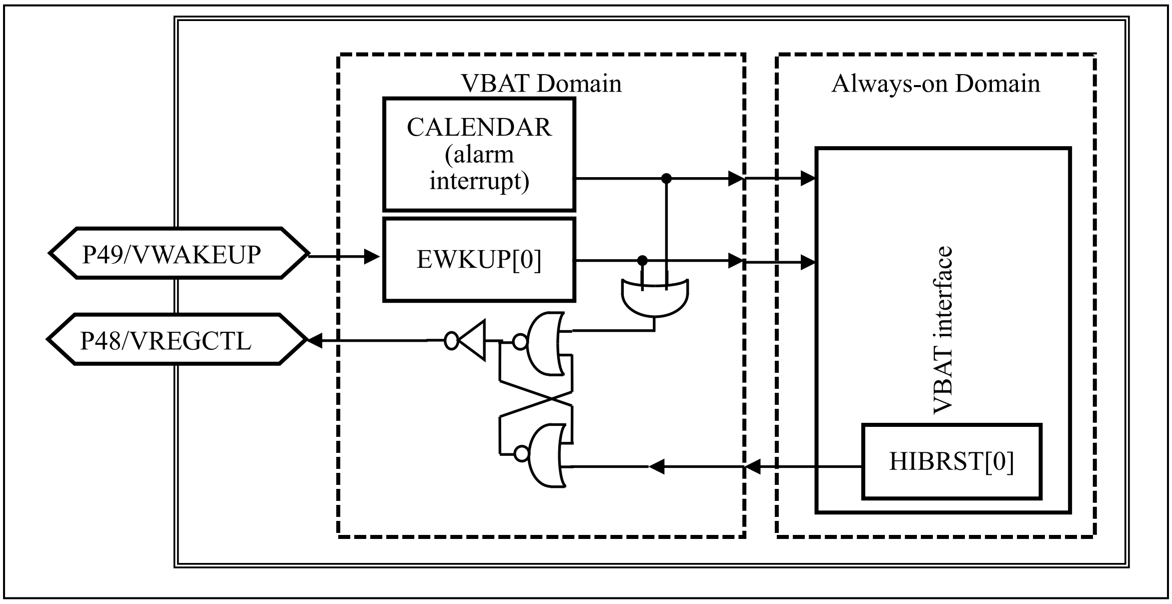


If the input voltage of the on-board regulator is higher than 5.5 V, the standby pin of the on-board regulator cannot be directly controlled by the P48/VREGCNTL pin.
Execute buffering with an open collector buffer whose voltage resistance is higher than the input voltage of the on-board regulator.

■ Block configuration of hibernation controller

The hibernation controller is part of the RTC circuit. Figure 4-3 shows the configuration of the hibernation controller.

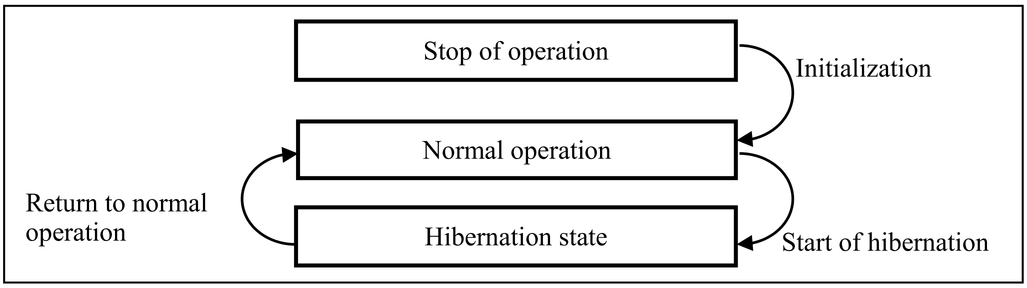
Figure 4-3 Hibernation controller



■ Example of hibernation operation flow

Figure 4-4 shows the hibernation operation flow.

Figure 4-4 Example of hibernation operation flow



● Initial settings of hibernation operation

Below are the initial settings required for the hibernation operation.

· Alarm setting and software reset of the RTC
After setting the alarm, execute the software reset.
For the method of setting the alarm, refer to "CHAPTER: RTC Count Block" in "FM4 Family PERIPHERAL MANUAL Timer Part".

· Setting of the P49/VWAKEUP pin
Write "1" to the VPFR0 bit in the VBPFR Register.
· Setting of the P48/VREGCNTL pin
Write "1" to the VPFR1 bit in the VBPFR Register.

The CPU core can transit to the hibernation state even when the alarm setting of the RTC and the setting of the P49/WAKEUP pin are not done.
If the CPU core transits to the hibernation state with both settings not done, it cannot return to the normal operation state.

● Setting of hibernation start

With both alarm interrupt of the RTC and wakeup (P49/VWAKEUP pin) cleared, if "1" is written to bit0 in the HIBRST Register, the P48/VREGCNTL pin becomes "0", the on-board regulator transits to the standby state and the VCC power supply is turned off.

● Judging return from hibernation state and operations after return from hibernation state

If an alarm interrupt of the RTC or a wakeup request occurs, the P48/VREGCNTL pin becomes "1", the on-board regulator returns from the standby state and the VCC power supply is turned on.
If the VCC power supply is turned on, the CPU core executes the normal power-on operation.
To judge whether the CPU core has returned from the hibernation state, check whether the following three conditions are met.

· The VBAT Domain has been powered on (VDET[7]).
· The alarm interrupt of the RTC has occurred (WTCR12 Register).
· A wakeup up request has been made (EWKUP[0]).

<Notes>

· The P48/VREGCNTL pin becomes "0" immediately after "1" has been written to bit0 in the HIBRST Register.
Complete all operations for turning off the VCC power supply before witting "1" to bit0 in the HIBRST Register.
· Even if the alarm of RTC is not set, execute the software reset.
· In the hibernation operation, the VCC power supply is assumed to be tuned off with the control of P48/VREGCNTL pin.
When the on-board regulator is not directly controlled with P48/VREGCNTL pin at debugging, turn off the VCC power supply once by manual operation.

5. Procedure for Setting 32 kHz Clock

This section explains recommended sequences of setting the 32 kHz oscillation circuit when using the RTC.

■ Features of 32 kHz oscillation circuit in VBAT Domain

With the 32 kHz oscillation circuit incorporated in the VBAT Domain, even when the CPU Domain and the Always-on Domain are turned off, the 32 kHz oscillation circuit can continue operating and the RTC can continue counting the time.

■ Linking with clock oscillation circuit

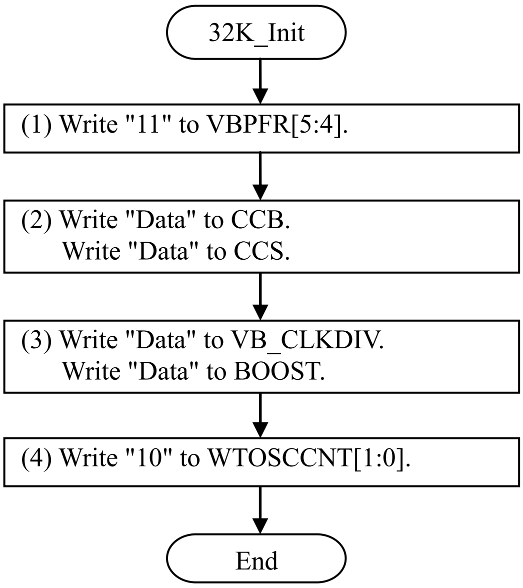
If the 32 kHz oscillation circuit in the VBAT Domain is linked with the clock control circuit is executed, the VBAT Domain becomes compatible with the FM3 Family.

In addition, with the 32 kHz oscillation circuit in the VBAT Domain linked with the clock control circuit and the VCC power supply turned off, if the CPU transits to deep standby mode or deep standby stop mode as the VCC power supply is turned on, the 32 kHz oscillation automatically stops.

If the 32 kHz clock is only used as the clock for subrun mode, power consumption of the backup power supply can be reduced by linking the 32 kHz oscillation circuit with the clock control circuit.

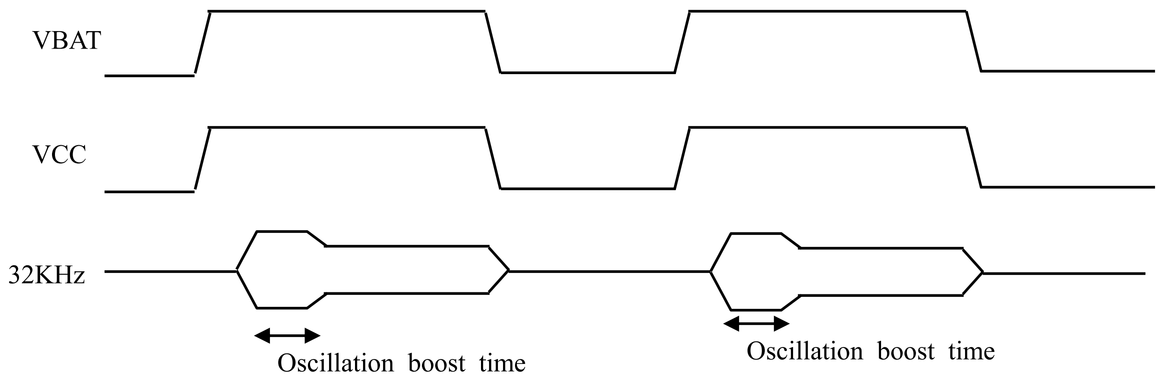
● Example of setting procedure

(1) Set VBAT I/O Port to use the 32k oscillation circuit.
(2) Set an appropriate current value for the current that is to flow to the oscillation amplifier circuit.
(3) Set the oscillation boost time.
(4) Enable the cooperative operation with the clock control circuit.
In addition, enable the oscillation.

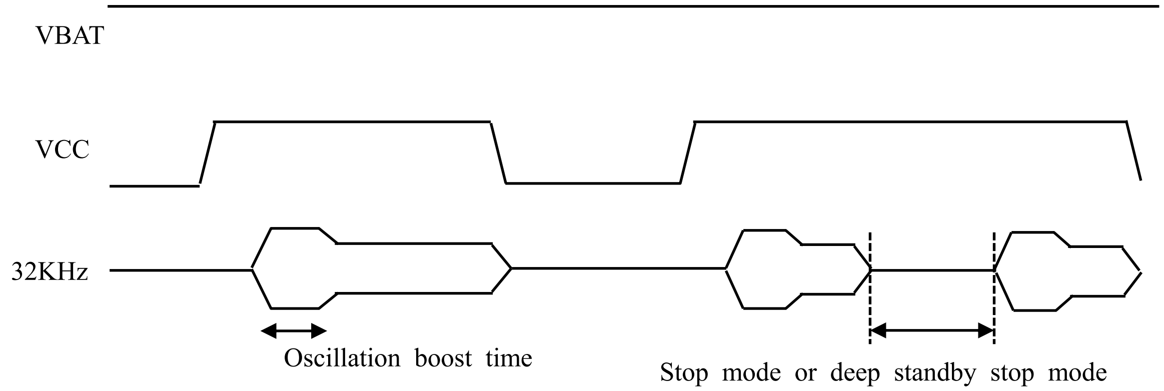

32K_Init
(1) Write "11" to VBPFR[5:4].
(2) Write "Data" to CCB.
Write "Data" to CCS.
(3) Write "Data" to VB_CLKDIV.
Write "Data" to BOOST.
(4) Write "10" to WTOSCCNT[1:0].
End

● Examples of operation

- No backup power supply is used.



- The backup power supply is used, and the 32 kHz oscillation circuit is linked with the clock control circuit.



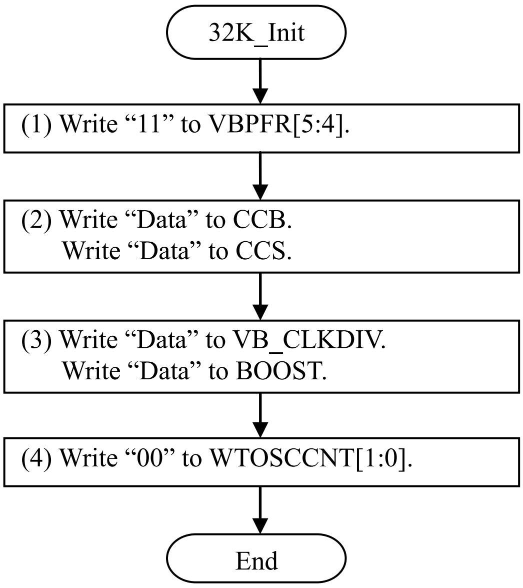
■ Not linking with clock oscillation circuit

If always using the backup power supply to keep the RTC operating, do not link the 32 kHz oscillation circuit with the clock oscillation circuit.

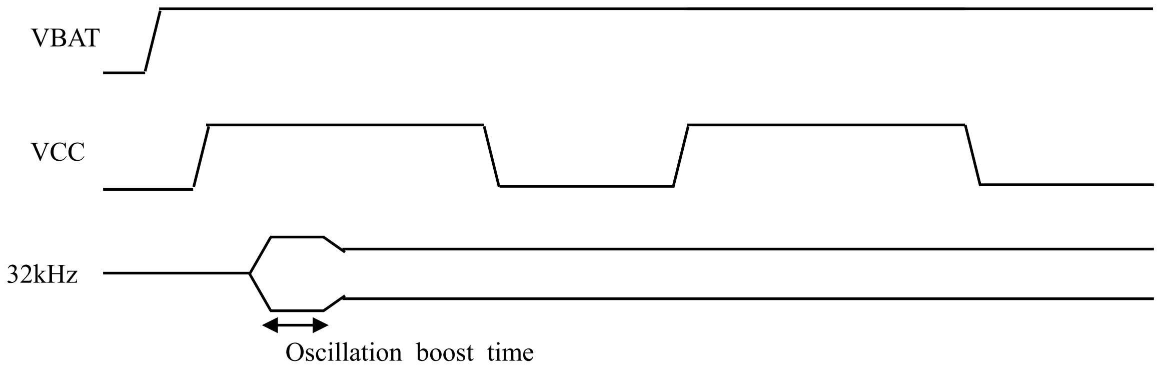
The average power consumption of the entire system can be reduced by executing the following operations: keep only the VBAT operating with the backup power supply, and use the hibernation control of the VBAT Domain or the external circuit to turn off the VCC power supply while processes by the CPU are not necessary.

● Example of setting procedure

(1) Set the Register of VBAT I/O Port to use the 32 kHz oscillation circuit.
(2) Set an appropriate current value for the current that is to flow to the oscillation amplifier circuit.
(3) Set the oscillation boost time.
(4) Disable the cooperative operation with the clock control circuit.
In addition, enable the oscillation.


32K_Init
(1) Write “11” to VBPFR[5:4].
(2) Write “Data” to CCB.
Write “Data” to CCS.
(3) Write “Data” to VB_CLKDIV.
Write “Data” to BOOST.
(4) Write “00” to WTOSCCNT[1:0].
End

● Examples of operation



■ Not linking with clock control circuit but waiting for oscillation stabilization

It is necessary to not link the 32 kHz oscillation circuit with the clock oscillation circuit when always using the backup power supply to keep the RTC operating. Nonetheless, the 32 kHz oscillation circuit and RTC in the VBAT Domain do not have the oscillation stabilization wait function.

When the 32 kHz clock is used only for the RTC, a software timer can be used to count the oscillation stabilization wait time. However, if the 32 kHz clock is also used in subrun mode, the clock oscillation stabilization wait function becomes necessary.

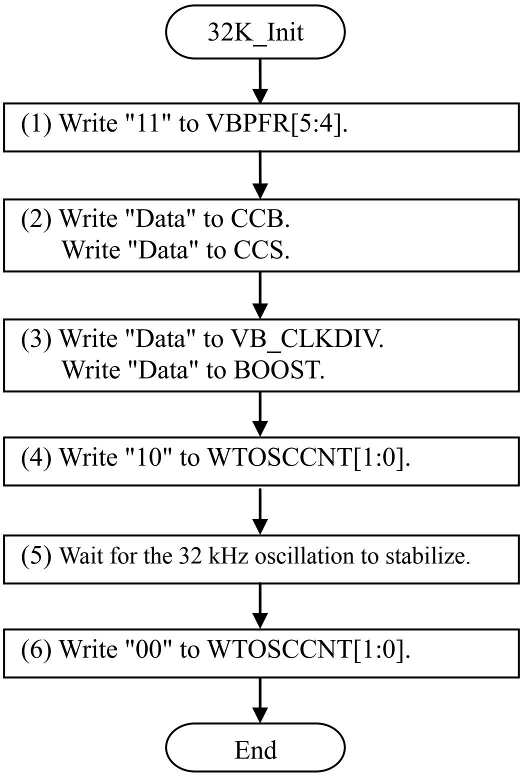
In the above situation, follow the procedure below to enable the oscillation stabilization wait function only at the start of oscillation.

● Example of setting procedure

Enable the cooperative operation with the clock control circuit and start the oscillation.

After the oscillation stabilization wait time has elapsed, disable the cooperative operation with the clock control circuit.

(1) Set the Register of VBAT I/O Port to use the 32 kHz oscillation circuit.
(2) Set an appropriate current value for the current that is to flow to the oscillation amplifier circuit.
(3) Set the oscillation boost time.
(4) Enable the cooperative operation with the clock control circuit. In addition, enable the oscillation.
(5) Wait for the stabilization of the 32 kHz oscillation.
(6) Disable the cooperative operation with the clock control circuit. (The oscillation keeps being enabled.)


32K_Init
(1) Write "11" to VBPFR[5:4].
(2) Write "Data" to CCB.
Write "Data" to CCS.
(3) Write "Data" to VB_CLKDIV.
Write "Data" to BOOST.
(4) Write "10" to WTOSCCNT[1:0].
(5) Wait for the 32 kHz oscillation to stabilize.
(6) Write "00" to WTOSCCNT[1:0].
End

6. Registers

This section explains the register list of the VBAT Domain unit.

Table 6-1 shows the registers of the VBAT Domain unit.

Table 6-1 Registers of VBAT Domain unit.

Abbreviation	Register name	Reference
VB_CLKDIV	VB_CLKDIV Register	6.1
WTOSCCNT	WTOSCCNT Register	6.2
CCS/CCB	CCS/CCB Register	6.3
BOOST	BOOST Register	6.4
EWKUP	EWKUP Register	6.5
HIBRST	HIBRST Register	6.6
VDET	VEDT Register	6.7
VBPFR	Port Function Set Register	6.8
VBPCR	Pull-up Set Register	6.9
VBDDR	Port I/O Direction Set Register	6.10
VBDIR	Port I/O Data Register	6.11
VBDOR	Port Output Data Register	6.12
VBPZR	Port Pseudo-Open Drain Set Register	6.13

6.1. VB_CLKDIV Register

VB_CLKDIV register set the frequency of transfer clock when the buck-up register and port register are transferred simultaneously.

bit	7	6	5	4	3	2	1	0
Field	DIV7	DIV6	DIV5	DIV4	DIV3	DIV2	DIV1	DIV0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	1	1	1

The interface circuit type for this register is type 3.

[bit7:0] DIV[7:0]: Transfer clock set bits for PREAD, PWRITE, BREAD, BWRITE

These bits set the transfer clock cycle used in the batch transfer of the backup register and of the port register.
Equation of computing the register value: transfer clock = PCLK / (VB_CLKDIV + 2)
(Set these bits to a value that makes the frequency of the transfer clock used in BREAD/BWRITE and PREAD/PWRITE 1 MHz or below.)
*:Do not set the DIV7 to DIV0 bits to "0x00".

6.2. WTOSCCNT Register

WTOSCCNT Register specifies the operation of 32 kHz Oscillation circuit.

bit	7	6	5	4	3	2	1	0
Field	Reserved						SOSCNTL	SOSCEX
Attribute	-						R/W	R/W
Initial value	-						0	1

The interface circuit type for this register is type 3.

[bit7:2] Reserved: Reserved bits

These bits read "0b000000".
In a write access to these bits, write "0b000000" to them.

[bit1] SOSCNTL: Cooperative operation control bit

This bit enables or disables the cooperative operation between the 32 kHz oscillation circuit and the clock control circuit.

bit		Description
Reading		A read access reads the value of this bit.
Writing	0	The 32 kHz oscillation circuit operates independently as VBAT Dmain. (Initial value)
	1	The 32 kHz oscillation circuit is linked with the clock oscillation circuit.

[bit0] SOSCEX: Oscillation enable bit

This bit enables or disables the operation of the oscillation circuit when the 32 kHz oscillation circuit operates independently as VBAT Domain.
If the 32 kHz oscillation circuit is linked with the clock oscillation circuit, this bit cannot control the operation of the oscillation circuit.

bit		Description
Reading		A read access reads the value of this bit.
Writing	0	Starts the oscillation.
	1	Stops the oscillation. (Initial value)

6.3. CCS/CCB Register

CCS Register sets the current value when the oscillation sustains.
CCB Register sets the boost current at the oscillation start.

bit	7	6	5	4	3	2	1	0
Field	CCS							
Attribute	R/W							
Initial value	00001000							

The interface circuit type for this register is type 3.

[bit7:0] CCS: Oscillation sustain current set bits

These bits set the value of current for sustaining oscillation.

bit	7	6	5	4	3	2	1	0
Field	CCB							
Attribute	R/W							
Initial value	00010000							

The interface circuit type for this register is type 3.

[bit7:0] CCB: Oscillation boost current set bits

These bits set the value of boost current at the start of oscillation.
Table 6-2 shows the relationship between the settings of the CCS/CCB Register and the current values.

Table 6-2 Relationship between settings of CCS/CCB Register and current values.

CCS/CCB	Current value (Type)	Remark
00000000	(0 nA)	Setting the CCS/CCB Register to this value is prohibited if the 32 kHz oscillation is enabled.
00000001	Undefined	Setting prohibited
0000001x	Undefined	Setting prohibited
000001xx	385 nA	
00001xxx	445 nA	(Initial value of CCS Register)
0001xxxx	510 nA	(Initial value of CCB Register)
	Undefined	Setting prohibited
01xxxxxx	Undefined	Setting prohibited
1xxxxxxx	Undefined	Setting prohibited

6.4. BOOST Register

BOOST Register sets the clock value of oscillation boost.

bit	7	6	5	4	3	2	1	0
Field	Reserved						BOOST1	BOOST0
Attribute	-						R/W	R/W
Initial value	-						1	1

The interface circuit type for this register is type 3.

[bit7:2] Reserved: Reserved bits

These bits read "0b000000".
In a write access to these bits, write "0b000000" to them.

[bit1:0] BOOST1, BOOST0: Oscillation boost time set bits

These bits set the number of clocks for oscillation boost.

Table 6-3 Settings of oscillation boost time

BOOST1	BOOST0	Oscillation boost time
1	1	500 ms [Initial value]
1	0	250 ms
0	1	125 ms
0	0	62.5 ms

6.5. EWKUP Register

EWKUP Register displays and clears the request state of the wakeup.

bit	7	6	5	4	3	2	1	0
Field	Reserved							WUP0
Attribute	-							R/W
Initial value	-							0

The interface circuit type for this register is type 3.

[bit7:1] Reserved: Reserved bits

These bits read "0b0000000".
In a write access to these bits, write "0b0000000" to them.

[bit0] WUP0: Wakeup request bit

bit		Description
Reading	0	The VBAT Domain has accepted no wakeup request.
	1	The VBAT Domain has accepted a wakeup request.
Writing	0	The VBAT Domain clears a wakeup request.
	1	Writing "1" to this bit has no effect on operation.

The wakeup request function is enabled if VPFR0 in VBPFR register is set to "1".
With the wakeup request function enabled, if the VBAT Domains detects a rising edge of the P49/VWAKEUP pin, it accepts a wakeup request and makes an RTC interrupt to the interrupt control circuit.
The wakeup request can be accepted at the 7th PCLK cycle from the wakeup request clear or later. To clear a register in VBAT Domain, 7 PCLK cycles are required. So, if the standby mode is entered before 7 PCLK cycles have elapsed, the wakeup request is not accepted because PCLK is stopped.

6.6. HIBRST Register

HIBRST Register sets the hibernation start.

bit	7	6	5	4	3	2	1	0
Field	Reserved							HIBRST
Attribute	-							R/W
Initial value	-							0

The interface circuit type for this register is type 1.

[bit7:1] Reserved: Reserved bits

These bits read "0b0000000".
In a write access to these bits, write "0b0000000" to them.

[bit0] HIBRST: Hibernation start bit

bit		Description
Reading		This bit reads "0".
Writing	0	Writing "0" to this bit has no effect on operation.
	1	Starts the hibernation.

The hibernation can be started if VPFR0 bit in Port Function Setup Register (VBPFR) is set to "1".
If VPFR0 bit in Port Function Setup Register (VBPFR) is set to "0", the access to this bit has no effect on operation.

6.7. VDET Register

VDET Register indicates the state of power-on circuit and clears the power-on signal.

bit	7	6	5	4	3	2	1	0
Field	PON	Reserved						
Attribute	R/W	-						
Initial value	1	-						

The interface circuit type for this register is type 1.

[bit7] PON: Power-on bit

This bit indicates the state of the power-on circuit and clears the power-on signal.

bit		Description
Reading	0	Indicates that the initialization signal of the power-on circuit has been cleared.
	1	Indicates that the power-on circuit has output the initialization signal.
Writing	0	Clears the power-on signal.
	1	Writing "1" to this bit has no effect on operation.

[bit6:0] Reserved: Reserved bits

These bits read "0b0000000".
In a write access to these bits, write "0b0000000" to them.

6.8. Port Function Set Register (VBPFR)

VBPFR Register selects the usage of pins.

bit	7	6	5	4	3	2	1	0
Field	Reserved		SPSR1	SPSR0	VPFR3	VPFR2	VPFR1	VPFR0
Attribute	-		R/W	R/W	R/W	R/W	R/W	R/W
Initial value	-		0	1	1	1	0	0

The interface circuit type for this register is type 3.

[bit7:6] Reserved: Reserved bits

These bits read "0b00".
In a write access to these bits, write "0b00" to them.

[bit5:4] SPSR1, SPSR0: Oscillation pin function set bits

bit5	bit4	Function
0	0	The P46 and P47 pins are used as digital (GPIO) pins.
0	1	The P46 and P47 pins are used as 32 kHz oscillation pins. [Initial value]
1	0	The P46 and P47 pins are used as digital (GPIO) pins.
1	1	The P46 pin is used as an external clock input pin. The P47 pin is used as a digital (GPIO) pin.

[bit3] VPFR3: Port function of P46/X0A pin set bit

[bit2] VPFR2: Port function of P47/X1A pin set bit

bit		Description
Reading		A read access reads the value of this bit.
Writing	0	The pin corresponding to the VPFR3 bit or to the VPFR2 bit is used as a GPIO port.
	1	The pin corresponding to the VPFR3 bit or to the VPFR2 bit is used as an I/O pin of a peripheral function. (Initial value)

[bit1] VPFR1: Port function of P49/VWAKEUP pin set bit

[bit0] VPFR0: Port function of P48/VREGCTL pin set bit

bit		Description
Reading		A read access reads the value of this bit.
Writing	0	The pin corresponding to the VPFR1/VPFR0 bit is used as a GPIO port. [Initial value]
	1	The pin corresponding to the VPFR1/VPFR0 bit is used as an I/O pin of a peripheral function.

6.9. Pull-up Set Register (VBPCR)

VBPCR Register sets the pull-up of pins.

bit	7	6	5	4	3	2	1	0
Field	Reserved				VPCR3	VPCR2	VPCR1	VPCR0
Attribute	-				R/W	R/W	R/W	R/W
Initial value	-				0	0	0	0

The interface circuit type for this register is type 3.

[bit7:4] Reserved: Reserved bits

These bits read "0b0000".
In a write access to these bits, write "0b0000" to them.

[bit3] VPCR3: P46/X0A pin pull-up set bit

[bit2] VPCR2: P47/X1A pin pull-up set bit

[bit1] VPCR1: P49/VWAKEUP pin pull-up set bit

[bit0] VPCR0: P48/VREGCTL pin pull-up set bit

bit		Description
Reading		A read access reads the value of this bit. (Initial value = 0)
Writing	0	The pull-up resistor of the pin corresponding to the VPCR3/VPCR2/VPCR1/VPCR0 bit is disconnected from the pin.
	1	If the pin corresponding to the VPCR3/VPCR2/VPCR1/VPCR0 bit is in the input state (either GPIO function or peripheral function), the pull-up resistor is connected to the pin. If the pin corresponding to the VPCR3/VPCR2/VPCR1/VPCR0 bit is in the output state, the pull-up resistor is disconnected from the pin.

6.10. Port I/O Direction Set Register (VBDDR)

VBDDR Register sets the I/O direction of pins.

bit	7	6	5	4	3	2	1	0
Field	Reserved				VDDR3	VDDR2	VDDR1	VDDR0
Attribute	-				R/W	R/W	R/W	R/W
Initial value	-				0	0	0	0

The interface circuit type for this register is type 3.

[bit7:4] Reserved: Reserved bits

These bits read "0b0000".
In a write access to these bits, write "0b0000" to them.

[bit3] VDDR3: Port direction of P46/X0A pin set bit

[bit2] VDDR2: Port direction of P47/X1A pin set bit

[bit1] VDDR1: Port direction of P49/VWAKEUP pin set bit

[bit0] VDDR0: Port direction of P48/VREGCTL pin set bit

bit		Description
Reading		A read access reads the value of this bit. [Initial value = 0]
Writing	0	The GPIO port is used as an input port. If the pin corresponding to the VDDR3/VDDR2/VDDR1/VDDR0 bit is used as an I/O pin of a peripheral function, the setting of the VDDR3/VDDR2/VDDR1/VDDR0 bit becomes invalid.
	1	The GPIO port is used as an output port. If the pin corresponding to the VDDR3/VDDR2/VDDR1/VDDR0 bit is used as an I/O pin of a peripheral function, the setting of the VDDR3/VDDR2/VDDR1/VDDR0 bit becomes invalid.

6.11. Port Input Data Register (VBDIR)

VBDIR Register indicates the input data of pins.

bit	7	6	5	4	3	2	1	0
Field	Reserved				VDIR3	VDIR2	VDIR1	VDIR0
Attribute	-				R/W	R/W	R/W	R/W
Initial value	-				x	x	x	x

The interface circuit type for this register is type 1.

[bit7:4] Reserved: Reserved bits

These bits read "0b0000".
In a write access to these bits, write "0b0000" to them.

[bit3] VDIR3: Port input data of P46/X0A pin bit

[bit2] VDIR2: Port input data of P47/X1A pin bit

[bit1] VDIR1: Port input data of P49/VWAKEUP pin bit

[bit0] VDIR0: Port input data of P48/VREGCTL pin bit

bit		Description
Reading	0	Regardless of the pin function settings (VBPFR[3:0], VBDDR and VBDOR Registers), this bit indicates that the pin is in the "L" level input state or the "L" level output state. If the P46 and P47 pins are used as special function pins according to the settings of the SPSR1 and SPSR0(bit[5:4]) in the VBPFR Register, this bit always reads "0" as the input is blocked.
	1	Regardless of the pin function settings (VBPFR[3:0], VBDDR and VBDOR Registers), this bit indicates that the pin is in the "H" level input state or the "H" level output state.
Writing		Writing a value to this bit has no effect on operation.

6.12. Port Output Data Register (VBDOR)

VBDOR Register sets the data output to pins.

bit	7	6	5	4	3	2	1	0
Field	Reserved				VDOR3	VDOR2	VDOR1	VDOR0
Attribute	-				R/W	R/W	R/W	R/W
Initial value	-				1	1	1	1

The interface circuit type for this register is type 3.

[bit7:4] Reserved: Reserved bits

These bits read "0b0000".
In a write access to these bits, write "0b0000" to them.

[bit3] VDOR3: Port output data of P46/X0A pin bit

[bit2] VDOR2: Port output data of P47/X1A pin bit

[bit1] VDOR1: Port output data of P49/VWAKEUP pin bit

[bit0] VDOR0: Port output data of P48/VREGCTL pin bit

bit		Description
Reading		A read access reads the value of this bit. (Initial value = 1)
Writing	0	Outputs "L" level to the GPIO port. If the pin is used as an input pin or as a peripheral function I/O pin, the setting of this bit becomes invalid.
	1	Outputs "H" level to the GPIO port. If the pin is used as an input pin or as a peripheral function I/O pin, the setting of this bit becomes invalid.

6.13. Port Pseudo-Open Drain Set Register (VBPZR)

VBPZR Register sets the port pseudo-open drain of a pin.

bit	7	6	5	4	3	2	1	0
Field	Reserved						VPZR1	VPZR0
Attribute	-						R/W	R/W
Initial value	-						1	1

The interface circuit type for this register is type 3.

[bit7:2] Reserved: Reserved bits

These bits read "0b000000".
In a write access to these bits, write "0b000000" to them.

[bit1] VPZR1: P49/VWAKEUP pin pseudo-open drain set bit

[bit0] VPZR0: P48/VREGCTL pin pseudo-open drain set bit

bit		Description
Reading		A read access reads the value of this bit. [Initial value = 1]
Writing	0	If digital "H" level is output from a GPIO port or a peripheral macro, the pin becomes "H" level.
	1	If digital "H" level is output from a GPIO port or a peripheral macro, the pin becomes "Hi-Z". The pull-up resistor is disconnected regardless of the setting of the PCR Register.

7. Usage Precautions

Note the following when using the backup power supply.

- Charging a primary cell or overcharging a secondary cell may cause cell leakage or fire. Check the features of the cell to be used before deciding the configuration of the circuit around the cell.
- The hibernation control function cannot be used if the on-board regulator has no standby pin. To control the hibernation, select a product that has a standby pin.

CHAPTER: Interrupts

This chapter explains details of the interrupt controller.

CODE: 9BFIRQC_B_FM4-E01.0

1. Overview

The Cortex-M4 CPU core is equipped with the Nested Vectored Interrupt Controller (NVIC) inside the core. The NVIC supports reserved system exceptions and 128 peripheral interrupts, and can set the priority order of 16 interrupts (with a built-in 4-bit register). This section explains interrupt signals from peripheral functions installed in the microcontroller and the connection between the NVIC and the interrupt signals.

■ Configuration

Figure 1-1 Connection between interrupt signals and NVIC

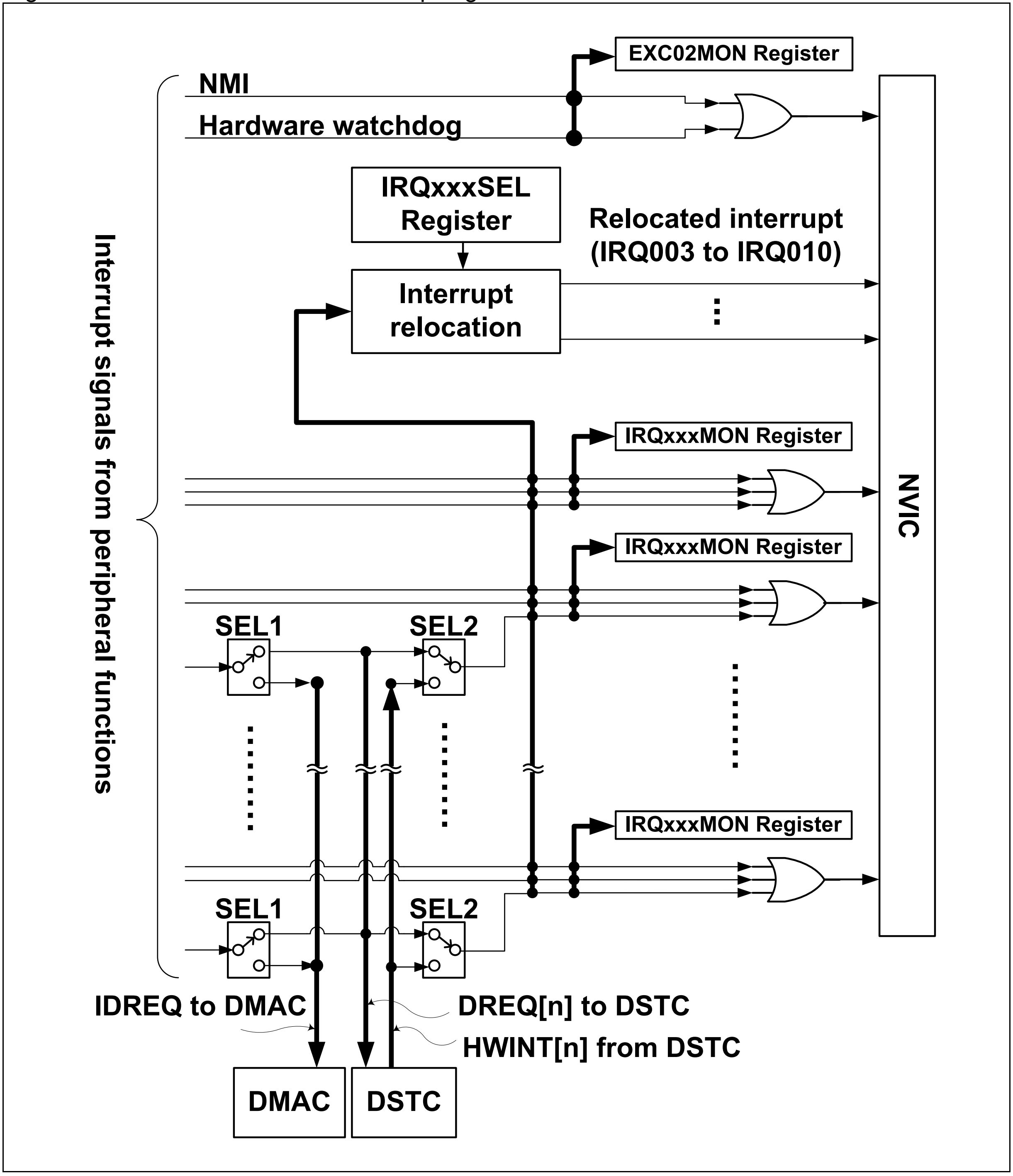


Figure 1-1 illustrates how the NVIC is connected to the interrupt signals input from peripheral functions, the DMAC and the DSTC. Details of the connection are explained below.

■ NVIC

The NVIC supports reserved system exceptions and 128 peripheral interrupts. For details of the NVIC, refer to "Cortex-M4 Technical Reference Manual". In "Cortex-M4 Technical Reference Manual", an exception other than the reserved system exceptions is defined as "external interrupt (IRQ)". In this document, the external interrupt (IRQ) is called a peripheral interrupt to differentiate the external interrupt (IRQ) from the external interrupt from a microcontroller external input pin.

The interrupt priority register of the NVIC has a 4-bit configuration and can set 16 interrupt priority levels.

The respective priorities of exception no. 4 to no. 15 can be set by using the System Handler Priority Registers (addresses: 0xE000ED18, 0xE000ED1C, 0xE000ED20) installed in the NVIC.

The respective priorities of peripheral interrupts of exception no. 16 to no. 143 can be set by using the IRQ Priority Registers (addresses: 0xE000E400 to 0xE000E42C) installed in the NVIC.

The NVIC supports non-maskable interrupt (NMI) input.

■ Interrupt aggregation and batch read registers

Interrupt signals to be input from all peripheral functions ("Interrupt signals from peripheral functions in Figure 1-1) are aggregated by the logic OR circuit in the figure. The aggregated interrupt signals are then connected to one of the 128 peripheral interrupts of the NVIC. See Table 2-1 to check to which peripheral function interrupt signal an interrupt output of the NVIC is assigned.

Since interrupt signals are aggregated by the logical OR circuit, one interrupt of the NVIC is generated by multiple sources. When an interrupt is generated, the source that caused that interrupt can be identified by reading Interrupt Batch Read Registers (IRQxxxMON Register in Figure 1-1). The Interrupt Batch Read Registers (IRQ000MON to IRQ127MON) cover all interrupt inputs of the NVIC.

The non-maskable interrupt signals (NMI) from the external interrupt and NMI controllers, and the interrupt signal (HW-Watchdog) from the hardware watchdog timer are aggregated by a logic OR circuit and then connected to the input of exception no. 2 of the NVIC. When an interrupt of exception no. 2 is generated, the source of the interrupt, which is either external interrupt and NMI controllers or hardware watchdog timer, can be identified by reading the EXC02MON Register.

The NMI pin of the microcontroller is shared with a general-purpose port. After a reset has been released, the initial function of the pin is general-purpose port, and NMI input is masked. To use the NMI function, enable the NMI function using the I/O port setting. For details, see Chapter "I/O Port". The NMI input signals are input to the NVIC via the external interrupt and NMI controllers.

■ Interrupt relocate function

When an interrupt whose source is aggregated with other sources is generated, the interrupt source is identified by the software. The identification of the interrupt source can be avoided by using the interrupt relocate function (Interrupt relocation in Figure 1-1).

Select the interrupt that needs to be relocated with the Relocate Interrupt Select Register (IRQxxxSEL in Figure 1-1). The interrupt signal selected is to be generated not at its original position but as a relocate interrupt. As the interrupt signal selected is separated from the logical OR circuit and is input to the NVIC independently of other interrupt signals, it can be processed by another interrupt vector. Since it is no longer necessary to identify an interrupt source, the NVIC can execute interrupt processing more efficiently. There are eight relocate interrupts, IRQ003 to IRQ010, available in the NVIC.

■ DMAC transfer request connection selection

Certain interrupt signals from peripheral functions can be used as DMA transfer request signals to the DMAC. The output selector circuit (SEL in Figure 1-1) determines whether such interrupt signals are connected to the NVIC or are connected to the DMAC as DMA transfer request signals (IDREQ to DMAC in Figure 1-1). Change the SEL1 setting with the DRQSEL Register. For details of the DMAC, see chapter "DMAC".

If an interrupt signal is connected to the DMAC by the SEL1, the DMA transfer of the DMAC can be started. In this situation, the signal to the NVIC is fixed at "Low" level. The bit corresponding to that interrupt signal in the IRQxxxMON Register reads "0" and no interrupt is generated to the NVIC. In addition, no transfer request notification is sent to the DSTC.

■ DSTC transfer end notification selection

Certain interrupt signals from peripheral functions can be used as DMA transfer request signals to the DSTC. The interrupt signal from a peripheral function can be identified as a DMA transfer request, depending on the setting of the DREQENB[n] Register of the DSTC. For details, see Chapter "DSTC".

The selector circuit (SEL in Figure 1-1) determines whether the interrupt signal to be input to the NVIC is connected to the DMA transfer end notification signal to be output from the DSTC (HWINT[n] from DSTC in Figure 1-1) or connected to the interrupt signal from a peripheral function. The function of SEL2 is changed according to the setting of the DREQENB[n] Register of the DSTC.

If the DMA transfer by the DSTC is used, the transfer end interrupt of from the DSTC is generated instead of the interrupt from a peripheral function. According to the circuit configuration shown in Figure 1-1, the generation of a transfer end interrupt from the DSTC can be checked by reading the corresponding to that transfer end interrupt in the IRQxxxMON Register. In addition, the interrupt relocate function can be applied to the transfer end interrupt from the DSTC.

■ DMA transfer acceptance signal connection

There are peripheral function blocks for which transfer request signals (interrupt signals) have to be cleared after the DMA transfer to those peripherals has ended. The transfer request signals for such peripheral functions are to be cleared by the DMAC or the DSTC. If the DMA transfer by the DMAC is selected in SEL1 or the DMA transfer by the DSTC is selected in SEL2, which is not illustrated in Figure 1-1, the DMA transfer acknowledge signal from the DMA or the DSTC is connected to a peripheral function.

■ DMA transfer stop signal connection

The DMA transfer stop request signal is output from the multi-function serial unit (to be called MFS later in this document). According to the selection made in SEL1 and SEL2, which is not illustrated in Figure 1-1, the MFS is connected to the DMAC or the DSTC as explained below.

If the connection between the DMAC and the MFS is selected in SEL1, the DMAC stops a transfer operation according to the transfer stop request signal. The DMAC cannot execute the transfer operation until the transfer stop request signal from the MFS is negated.

If the connection between the DSTC and the MFS is selected in SEL2, the DSTC stops a transfer operation according to the transfer stop request signal. The DSTC cannot execute the transfer operation until the transfer stop request signal from the MFS is negated. The transfer stop request signal from the MFS is aggregated with the transfer end interrupt (HWINT[n] signal) of the DSTC by logical OR, and is notified to the NVIC as an interrupt signal.

2. Lists of Interrupts

This section shows a list of sources of exceptions and interrupt sources input to the NVIC, a list of interrupts that can be transferred by the DMA transfer by the DMAC, and a list of interrupts that can be transferred by the DMA transfer by the DSTC.

■ List of exceptions and interrupts

Table 2-1 shows a list of sources of exceptions and interrupt to be input to the NVIC. Below are details of columns in the table.

Exc no.: NVIC exception number

IRQ no.: Peripheral interrupt number (number = Exc no. - 16)

Vector offset: Storage offset address of the vector that an interrupt refers to

bit: This indicates the number of a bit in a Batch Read Register (IRQxxxMON or EXC02MON) from which an interrupt source is read out. In the case of a single IRQ having multiple bit numbers. multiple sources are aggregated by logical OR, and a source can be read output from its corresponding bit. In the case of a single IRQ having only bit number "0", no multiple sources are aggregated by logical OR. "-" in this column indicates that there is no Batch Read Register for that exception or interrupt.

DMAC: This indicates the bit number in the DRQSEL Register corresponding to an interrupt signal for the DMA transfer by the DMAC. "-" in this column indicates that interrupt signal cannot be used in the DMA transfer by the DMAC. The connection in SEL1 in Figure 1-1 changes according to the setting of the bit corresponding to that interrupt signal in the DRQSEL Register.

DSTC: This indicates the bit number in the DREQENB[n] Register corresponding to an interrupt signal for the DMA transfer by the DSTC. "-" in this column indicates that interrupt signal cannot be used in the DMA transfer by the DSTC. The connection in SEL2 in Figure 1-1 changes according to the setting of the bit corresponding to that interrupt signal in the DREQENB[n] Register of the DSTC.

Exception source and interrupt source: This column contains exception sources and interrupt sources. Some interrupts have multiple sources. Such interrupt sources of a peripheral function are aggregated by logical OR. Even if only one interrupt source from a peripheral source is shown, such peripheral function may have multiple interrupt sources aggregated by logical OR. For details, refer to the respective details of peripheral functions.

Table 2-1 List of exception sources and interrupt sources

Exc. no.	IRQ no.	Vector offset	bit	DMAC	DSTC	Exception and interrupt source
0	-	0x000	-	-	-	(Stack pointer initial value)
1	-	0x004	-	-	-	Reset
2	-	0x008	0	-	-	Non-maskable interrupt (NMI)
			1	-	-	Hardware watchdog timer interrupt
3	-	0x00C	-	-	-	Hard fault
4	-	0x010	-	-	-	Mem manage fault
5	-	0x014	-	-	-	Bus fault
6	-	0x018	-	-	-	Usage fault
7 | 10	-	0x01C | 0x02B	-	-	-	Reserved
11	-	0x02C	-	-	-	SVCall (supervisor call)
12	-	0x030	-	-	-	Debug monitor
13	-	0x034	-	-	-	Reserved
14	-	0x038	-	-	-	PendSV
15	-	0x03C	-	-	-	SysTick
16	0	0x040	0	-	-	Clock supervisor anomalous frequency detection interrupt (FCS)
17	1	0x044	0	-	-	Software watchdog timer interrupt
18	2	0x048	0	-	-	Low-voltage detection (LVD) interrupt
19	3	0x04C	7-0	-	-	Relocate interrupt (selected by the IRQ003SEL Register)
20	4	0x050	7-0	-	-	Relocate interrupt (selected by the IRQ004SEL Register)
21	5	0x054	7-0	-	-	Relocate interrupt (selected by the IRQ004SEL Register)
22	6	0x058	7-0	-	-	Relocate interrupt (selected by the IRQ004SEL Register)
23	7	0x05C	7-0	-	-	Relocate interrupt (selected by the IRQ004SEL Register)
24	8	0x060	7-0	-	-	Relocate interrupt (selected by the IRQ004SEL Register)
25	9	0x064	7-0	-	-	Relocate interrupt (selected by the IRQ004SEL Register)
26	10	0x068	7-0	-	-	Relocate interrupt (selected by the IRQ004SEL Register)
27	11	0x06C	0	28	0	External pin interrupt ch.0
28	12	0x070	0	29	1	External pin interrupt ch.1
29	13	0x074	0	30	2	External pin interrupt ch.2
30	14	0x078	0	31	3	External pin interrupt ch.3
31	15	0x07C	0	-	4	External pin interrupt ch.4
32	16	0x080	0	-	5	External pin interrupt ch.5
33	17	0x084	0	-	6	External pin interrupt ch.6
34	18	0x088	0	-	7	External pin interrupt ch.7

Exc. no.	IRQ no.	Vector offset	bit	DMAC	DSTC	Exception and interrupt source
35	19	0x08C	5	-	81	QPRC ch.0 PC match & RC match interrupt
			4	-	80	QPRC ch.0 out-of-range interrupt
			3	-	79	QPRC ch.0 count inversion interrupt
			2	-	78	QPRC ch.0 overflow interrupt QPRC ch.0 underflow interrupt QPRC ch.0 zero index interrupt
			1	-	77	QPRC ch.0 PC & RC match interrupt
			0	-	76	QPRC ch.0 PC match interrupt
36	20	0x090	5	-	87	QPRC ch.1 PC match & RC match interrupt
			4	-	86	QPRC ch.1 out-of-range interrupt
			3	-	85	QPRC ch.1 count inversion interrupt
			2	-	84	QPRC ch.1 overflow interrupt QPRC ch.1 underflow interrupt QPRC ch.1 zero index interrupt
			1	-	83	QPRC ch.1 PC & RC match interrupt
			0	-	82	QPRC ch.1 PC match interrupt
37	21	0x094	3	-	34	MFT unit 0 WFG timer 54 interrupt
			2	-	33	MFT unit 0 WFG timer 32 interrupt
			1	-	32	MFT unit 0 WFG timer 10 interrupt
			0	-	-	MFT unit 0 DTIF (motor emergency stop) interrupt
38	22	0x098	3	-	53	MFT unit 1 WFG timer 54 interrupt
			2	-	52	MFT unit 1 WFG timer 32 interrupt
			1	-	51	MFT unit 1 WFG timer 10 interrupt
			0	-	-	MFT unit 1 DTIF (motor emergency stop) interrupt
39	23	0x09C	3	-	162	MFT unit 2 WFG timer 54 interrupt
			2	-	161	MFT unit 2 WFG timer 32 interrupt
			1	-	160	MFT unit 2 WFG timer 10 interrupt
			0	-	-	MFT unit 2 DTIF (motor emergency stop) interrupt
40	24	0x0A0	2	-	37	MFT unit 0 FRT ch.2 peak value detection interrupt
			1	-	36	MFT unit 0 FRT ch.1 peak value detection interrupt
			0	-	35	MFT unit 0 FRT ch.0 peak value detection interrupt
41	25	0x0A4	2	-	40	MFT unit 0 FRT ch.2 zero detection interrupt
			1	-	39	MFT unit 0 FRT ch.1 zero detection interrupt
			0	-	38	MFT unit 0 FRT ch.0 zero detection interrupt
42	26	0x0A8	3	-	44	MFT unit 0 ICU ch.3 input edge detection interrupt
			2	-	43	MFT unit 0 ICU ch.2 input edge detection interrupt
			1	-	42	MFT unit 0 ICU ch.1 input edge detection interrupt
			0	-	41	MFT unit 0 ICU ch.0 input edge detection interrupt
43	27	0x0AC	5	-	50	MFT unit 0 OCU ch.5 match detection interrupt
			4	-	49	MFT unit 0 OCU ch.4 match detection interrupt
			3	-	48	MFT unit 0 OCU ch.3 match detection interrupt
			2	-	47	MFT unit 0 OCU ch.2 match detection interrupt
			1	-	46	MFT unit 0 OCU ch.1 match detection interrupt
			0	-	45	MFT unit 0 OCU ch.0 match detection interrupt

Exc. no.	IRQ no.	Vector offset	bit	DMAC	DSTC	Exception and interrupt source
44	28	0x0B0	2	-	56	MFT unit 1 FRT ch.2 peak value detection interrupt
			1	-	55	MFT unit 1 FRT ch.1 peak value detection interrupt
			0	-	54	MFT unit 1 FRT ch.0 peak value detection interrupt
45	29	0x0B4	2	-	59	MFT unit 1 FRT ch.2 zero detection interrupt
			1	-	58	MFT unit 1 FRT ch.1 zero detection interrupt
			0	-	57	MFT unit 1 FRT ch.0 zero detection interrupt
46	30	0x0B8	3	-	63	MFT unit 1 ICU ch.3 input edge detection interrupt
			2	-	62	MFT unit 1 ICU ch.2 input edge detection interrupt
			1	-	61	MFT unit 1 ICU ch.1 input edge detection interrupt
			0	-	60	MFT unit 1 ICU ch.0 input edge detection interrupt
47	31	0x0BC	5	-	69	MFT unit 1 OCU ch.5 match detection interrupt
			4	-	68	MFT unit 1 OCU ch.4 match detection interrupt
			3	-	67	MFT unit 1 OCU ch.3 match detection interrupt
			2	-	66	MFT unit 1 OCU ch.2 match detection interrupt
			1	-	65	MFT unit 1 OCU ch.1 match detection interrupt
			0	-	64	MFT unit 1 OCU ch.0 match detection interrupt
48	32	0x0C0	2	-	165	MFT unit 2 FRT ch.2 peak value detection interrupt
			1	-	164	MFT unit 2 FRT ch.1 peak value detection interrupt
			0	-	163	MFT unit 2 FRT ch.0 peak value detection interrupt
49	33	0x0C4	2	-	168	MFT unit 2 FRT ch.2 zero detection interrupt
			1	-	167	MFT unit 2 FRT ch.1 zero detection interrupt
			0	-	166	MFT unit 2 FRT ch.0 zero detection interrupt
50	34	0x0C8	3	-	172	MFT unit 2 ICU ch.3 input edge detection interrupt
			2	-	171	MFT unit 2 ICU ch.2 input edge detection interrupt
			1	-	170	MFT unit 2 ICU ch.1 input edge detection interrupt
			0	-	169	MFT unit 2 ICU ch.0 input edge detection interrupt
51	35	0x0CC	5	-	178	MFT unit 2 OCU ch.5 match detection interrupt
			4	-	177	MFT unit 2 OCU ch.4 match detection interrupt
			3	-	176	MFT unit 2 OCU ch.3 match detection interrupt
			2	-	175	MFT unit 2 OCU ch.2 match detection interrupt
			1	-	174	MFT unit 2 OCU ch.1 match detection interrupt
			0	-	173	MFT unit 2 OCU ch.0 match detection interrupt
52	36	0x0D0	2	-	72	PPG ch.4 interrupt
			1	-	71	PPG ch.2 interrupt
			0	-	70	PPG ch.0 interrupt
53	37	0x0D4	2	-	75	PPG ch.12 interrupt
			1	-	74	PPG ch.10 interrupt
			0	-	73	PPG ch.8 interrupt

Exc. no.	IRQ no.	Vector offset	bit	DMAC	DSTC	Exception and interrupt source
54	38	0x0D8	2	-	181	PPG ch.20 interrupt
			1	-	180	PPG ch.18 interrupt
			0	-	179	PPG ch.16 interrupt
55	39	0x0DC	1	-	17	Base timer ch.0 source 1 (IRQ1) interrupt
			0	8	16	Base timer ch.0 source 0 (IRQ0) interrupt
56	40	0x0E0	1	-	19	Base timer ch.1 source 1 (IRQ1) interrupt
			0	-	18	Base timer ch.1 source 0 (IRQ0) interrupt
57	41	0x0E4	1	-	21	Base timer ch.2 source 1 (IRQ1) interrupt
			0	9	20	Base timer ch.2 source 0 (IRQ0) interrupt
58	42	0x0E8	1	-	23	Base timer ch.3 source 1 (IRQ1) interrupt
			0	-	22	Base timer ch.3 source 0 (IRQ0) interrupt
59	43	0x0EC	1	-	25	Base timer ch.4 source 1 (IRQ1) interrupt
			0	10	24	Base timer ch.4 source 0 (IRQ0) interrupt
60	44	0x0F0	1	-	27	Base timer ch.5 source 1 (IRQ1) interrupt
			0	-	26	Base timer ch.5 source 0 (IRQ0) interrupt
61	45	0x0F4	1	-	29	Base timer ch.6 source 1 (IRQ1) interrupt
			0	11	28	Base timer ch.6 source 0 (IRQ0) interrupt
62	46	0x0F8	1	-	31	Base timer ch.7 source 1 (IRQ1) interrupt
			0	-	30	Base timer ch.7 source 0 (IRQ0) interrupt
63	47	0x0FC	1	-	-	Dual timer ch.2 interrupt
			0	-	-	Dual timer ch.1 interrupt
64	48	0x100	0	-	88	Watch counter interrupt
65	49	0x104	0	-	-	External bus error output interrupt
66	50	0x108	0	-	-	Real timer counter interrupt
67	51	0x10C	0	-	8	External pin interrupt ch.8
68	52	0x110	0	-	9	External pin interrupt ch.9
69	53	0x114	0	-	10	External pin interrupt ch.10
70	54	0x118	0	-	11	External pin interrupt ch.11
71	55	0x11C	0	-	12	External pin interrupt ch.12
72	56	0x120	0	-	13	External pin interrupt ch.13
73	57	0x124	0	-	14	External pin interrupt ch.14
74	58	0x128	0	-	15	External pin interrupt ch.15
75	59	0x12C	3	-	-	PLL of USB / Ethernet oscillation stabilization wait completion interrupt
			2	-	-	Main PLL oscillation stabilization wait completion interrupt
			1	-	-	Sub clock oscillation stabilization wait completion interrupt
			0	-	-	Main clock oscillation stabilization wait completion interrupt

Exc. no.	IRQ no.	Vector offset	bit	DMAC	DSTC	Exception and interrupt source
76	60	0x130	0	12	89	MFS ch.0 reception interrupt
77	61	0x134	1	-	-	MFS ch.0 status interrupt
			0	13	90	MFS ch.0 transmission interrupt
78	62	0x138	0	14	91	MFS ch.1 reception interrupt
79	63	0x13C	1	-	-	MFS ch.1 status interrupt
			0	15	92	MFS ch.1 transmission interrupt
80	64	0x140	0	16	93	MFS ch.2 reception interrupt
81	65	0x144	1	-	-	MFS ch.2 status interrupt
			0	17	94	MFS ch.2 transmission interrupt
82	66	0x148	0	18	95	MFS ch.3 reception interrupt
83	67	0x14C	1	-	-	MFS ch.3 status interrupt
			0	19	96	MFS ch.3 transmission interrupt
84	68	0x150	0	20	97	MFS ch.4 reception interrupt
85	69	0x154	1	-	-	MFS ch.4 status interrupt
			0	21	98	MFS ch.4 transmission interrupt
86	70	0x158	0	22	99	MFS ch.5 reception interrupt
87	71	0x15C	1	-	-	MFS ch.5 status interrupt
			0	23	100	MFS ch.5 transmission interrupt
88	72	0x160	0	24	101	MFS ch.6 reception interrupt
89	73	0x164	1	-	-	MFS ch.6 status interrupt
			0	25	102	MFS ch.6 transmission interrupt
90	74	0x168	0	26	103	MFS ch.7 reception interrupt
91	75	0x16C	1	-	-	MFS ch.7 status interrupt
			0	27	104	MFS ch.7 transmission interrupt
92	76	0x170	4	-	-	A/D converter unit 0 range comparison result interrupt
			3	-	-	A/D converter unit 0 conversion result comparison interrupt
			2	-	-	A/D converter unit 0 FIFO overrun interrupt
			1	5	111	A/D converter unit 0 scan conversion interrupt
			0	-	110	A/D converter unit 0 priority conversion interrupt
93	77	0x174	4	-	-	A/D converter unit 1 range comparison result interrupt
			3	-	-	A/D converter unit 1 conversion result comparison interrupt
			2	-	-	A/D converter unit 1 FIFO overrun interrupt
			1	6	113	A/D converter unit 1 scan conversion interrupt
			0	-	112	A/D converter unit 1 priority conversion interrupt
94	78	0x178	4	4	109	USB ch.0 function endpoint 5 DRQ interrupt
			3	3	108	USB ch.0 function endpoint 4 DRQ interrupt
			2	2	107	USB ch.0 function endpoint 3 DRQ interrupt
			1	1	106	USB ch.0 function endpoint 2 DRQ interrupt
			0	0	105	USB ch.0 function endpoint 1 DRQ interrupt

Exc. no.	IRQ no.	Vector offset	bit	DMAC	DSTC	Exception and interrupt source
95	79	0x17C	5	-	-	USB ch.0 host SOFIRQ interrupt USB ch.0 host CMPIRQ interrupt
			4	-	-	USB ch.0 host DIRQ interrupt USB ch.0 host URIRQ interrupt USB ch.0 host RWKIRQ interrupt USB ch.0 host CNNIRQ interrupt
			3	-	-	USB ch.0 function SPK interrupt
			2	-	-	USB ch.0 function SUSP interrupt USB ch.0 function SOF interrupt USB ch.0 function BRST interrupt USB ch.0 function CONF interrupt USB ch.0 function WKUP interrupt
			1	-	-	USB ch.0 function endpoint 0 DRQO interrupt
			0	-	-	USB ch.0 function endpoint 0 DRQI interrupt
96	80	0x180	0	-	-	CAN ch.0 interrupt
97	81	0x184	0	-	-	CAN ch.1 interrupt
98	82	0x188	2	-	-	Ethernet ch.0 SBD interrupt
			1	-	-	Ethernet ch.0 PMT interrupt
			0	-	-	Ethernet ch.0 LPI interrupt
99	83	0x18C	0	-	-	DMAC ch.0 interrupt
100	84	0x190	0	-	-	DMAC ch.1 interrupt
101	85	0x194	0	-	-	DMAC ch.2 interrupt
102	86	0x198	0	-	-	DMAC ch.3 interrupt
103	87	0x19C	0	-	-	DMAC ch.4 interrupt
104	88	0x1A0	0	-	-	DMAC ch.5 interrupt
105	89	0x1A4	0	-	-	DMAC ch.6 interrupt
106	90	0x1A8	0	-	-	DMAC ch.7 interrupt
107	91	0x1AC	1	-	-	DSTC ERINT interrupt
			0	-	-	DSTC SWINT interrupt
108	92	0x1B0	3	-	131	External pin interrupt ch.19
			2	-	130	External pin interrupt ch.18
			1	-	129	External pin interrupt ch.17
			0	-	128	External pin interrupt ch.16
109	93	0x1B4	3	-	135	External pin interrupt ch.23
			2	-	134	External pin interrupt ch.22
			1	-	133	External pin interrupt ch.21
			0	-	132	External pin interrupt ch.20
110	94	0x1B8	3	-	139	External pin interrupt ch.27
			2	-	138	External pin interrupt ch.26
			1	-	137	External pin interrupt ch.25
			0	-	136	External pin interrupt ch.24
111	95	0x1BC	3	-	143	External pin interrupt ch.31
			2	-	142	External pin interrupt ch.30
			1	-	141	External pin interrupt ch.29
			0	-	140	External pin interrupt ch.28

Exc. no.	IRQ no.	Vector offset	bit	DMAC	DSTC	Exception and interrupt source
112	96	0x1C0	5	-	187	QPRC ch.2 PC match & RC match interrupt
			4	-	186	QPRC ch.2 out-of-range interrupt
			3	-	185	QPRC ch.2 count inversion interrupt
			2	-	184	QPRC ch.2 overflow interrupt QPRC ch.2 underflow interrupt QPRC ch.2 zero index interrupt
			1	-	183	QPRC ch.2 PC & RC match interrupt
			0	-	182	QPRC ch.2 PC match interrupt
113	97	0x1C4	5	-	193	QPRC ch.3 PC match & RC match interrupt
			4	-	192	QPRC ch.3 out-of-range interrupt
			3	-	191	QPRC ch.3 count inversion interrupt
			2	-	190	QPRC ch.3 overflow interrupt QPRC ch.3 underflow interrupt QPRC ch.3 zero index interrupt
			1	-	189	QPRC ch.3 PC & RC match interrupt
			0	-	188	QPRC ch.3 PC match interrupt
114	98	0x1C8	1	-	145	Base timer ch.8 source 1 (IRQ1) interrupt
			0	-	144	Base timer ch.8 source 0 (IRQ0) interrupt
115	99	0x1CC	1	-	147	Base timer ch.9 source 1 (IRQ1) interrupt
			0	-	146	Base timer ch.9 source 0 (IRQ0) interrupt
116	100	0x1D0	1	-	149	Base timer ch.10 source 1 (IRQ1) interrupt
			0	-	148	Base timer ch.10 source 0 (IRQ0) interrupt
117	101	0x1D4	1	-	151	Base timer ch.11 source 1 (IRQ1) interrupt
			0	-	150	Base timer ch.11 source 0 (IRQ0) interrupt
118	102	0x1D8	7	-	159	Base timer ch.15 source 1 (IRQ1) interrupt
			6	-	158	Base timer ch.15 source 0 (IRQ0) interrupt
			5	-	157	Base timer ch.14 source 1 (IRQ1) interrupt
			4	-	156	Base timer ch.14 source 0 (IRQ0) interrupt
			3	-	155	Base timer ch.13 source 1 (IRQ1) interrupt
			2	-	154	Base timer ch.13 source 0 (IRQ0) interrupt
			1	-	153	Base timer ch.12 source 1 (IRQ1) interrupt
			0	-	152	Base timer ch.12 source 0 (IRQ0) interrupt
119	103	0x1DC	0	-	194	MFS ch.8 reception interrupt
120	104	0x1E0	1	-	-	MFS ch.8 status interrupt
			0	-	195	MFS ch.8 transmission interrupt
121	105	0x1E4	0	-	196	MFS ch.9 reception interrupt
122	106	0x1E8	1	-	-	MFS ch.9 status interrupt
			0	-	197	MFS ch.9 transmission interrupt
123	107	0x1EC	0	-	198	MFS ch.10 reception interrupt
124	108	0x1F0	1	-	-	MFS ch.10 status interrupt
			0	-	199	MFS ch.10 transmission interrupt
125	109	0x1F4	0	-	200	MFS ch.11 reception interrupt
126	110	0x1F8	1	-	-	MFS ch.11 status interrupt
			0	-	201	MFS ch.11 transmission interrupt

Exc. no.	IRQ no.	Vector offset	bit	DMAC	DSTC	Exception and interrupt source
127	111	0x1FC	4	-	-	A/D converter unit 2 range comparison result interrupt
			3	-	-	A/D converter unit 2 conversion result comparison interrupt
			2	-	-	A/D converter unit 2 FIFO overrun interrupt
			1	7	115	A/D converter unit 2 scan conversion interrupt
			0	-	114	A/D converter unit 2 priority conversion interrupt
128	112	0x200	0	-	-	Reserved
129	113	0x204	4	-	206	USB ch.1 function endpoint 5 DRQ interrupt
			3	-	205	USB ch.1 function endpoint 4 DRQ interrupt
			2	-	204	USB ch.1 function endpoint 3 DRQ interrupt
			1	-	203	USB ch.1 function endpoint 2 DRQ interrupt
			0	-	202	USB ch.1 function endpoint 1 DRQ interrupt
130	114	0x208	5	-	-	USB ch.1 host SOFIRQ interrupt USB ch.1 host CMPIRQ interrupt
			4	-	-	USB ch.1 host DIRQ interrupt USB ch.1 host URIRQ interrupt USB ch.1 host RWKIRQ interrupt USB ch.1 host CNNIRQ interrupt
			3	-	-	USB ch.0 function SPK interrupt
			2	-	-	USB ch.1 function SUSP interrupt USB ch.1 function SOF interrupt USB ch.1 function BRST interrupt USB ch.1 function CONF interrupt USB ch.1 function WKUP interrupt
			1	-	-	USB ch.1 function endpoint 0 DRQO interrupt
			0	-	-	USB ch.1 function endpoint 0 DRQI interrupt
131	115	0x20C	0	-	-	Reserved
132	116	0x210	0	-	-	Reserved
133	117	0x214	0	-	-	Reserved
134	118	0x218	1	-	-	SD card interrupt
			0	-	-	Aggregation of all SD I/F interrupt sources
135	119	0x21C	0	-	-	Flash I/F
136	120	0x220	0	-	209	MFS ch.12 reception interrupt
137	121	0x224	1	-	-	MFS ch.12 status interrupt
			0	-	210	MFS ch.12 transmission interrupt
138	122	0x228	0	-	211	MFS ch.13 reception interrupt
139	123	0x22C	1	-	-	MFS ch.13 status interrupt
			0	-	212	MFS ch.13 transmission interrupt
140	124	0x230	0	-	213	MFS ch.14 reception interrupt
141	125	0x234	1	-	-	MFS ch.14 status interrupt
			0	-	214	MFS ch.14 transmission interrupt
142	126	0x238	0	-	215	MFS ch.15 reception interrupt
143	127	0x23C	1	-	-	MFS ch.15 status interrupt
			0	-	216	MFS ch.15 transmission interrupt

■ Interrupt signals input to DMAC

Table 2-2 shows interrupt signals input as transfer request signals to the DMAC. Numbers in the table correspond to the bit numbers of the DRQSEL Register (IDREQ numbers of the DMAC).

Table 2-2 List of interrupt signals input to DMAC

Number	Interrupt signal name
0	USB ch.0 function endpoint 1 DRQ interrupt
1	USB ch.0 function endpoint 2 DRQ interrupt
2	USB ch.0 function endpoint 3 DRQ interrupt
3	USB ch.0 function endpoint 4 DRQ interrupt
4	USB ch.0 function endpoint 5 DRQ interrupt
5	A/D converter unit 0 scan conversion interrupt
6	A/D converter unit 1 scan conversion interrupt
7	A/D converter unit 2 scan conversion interrupt
8	Base timer ch.0 source 0 (IRQ0) interrupt
9	Base timer ch.2 source 0 (IRQ0) interrupt
10	Base timer ch.4 source 0 (IRQ0) interrupt
11	Base timer ch.6 source 0 (IRQ0) interrupt
12	MFS ch.0 reception interrupt
13	MFS ch.0 transmission interrupt
14	MFS ch.1 reception interrupt
15	MFS ch.1 transmission interrupt
16	MFS ch.2 reception interrupt
17	MFS ch.2 transmission interrupt
18	MFS ch.3 reception interrupt
19	MFS ch.3 transmission interrupt
20	MFS ch.4 reception interrupt
21	MFS ch.4 transmission interrupt
22	MFS ch.5 reception interrupt
23	MFS ch.5 transmission interrupt
24	MFS ch.6 reception interrupt
25	MFS ch.6 transmission interrupt
26	MFS ch.7 reception interrupt
27	MFS ch.7 transmission interrupt
28	External pin interrupt ch.0
29	External pin interrupt ch.1
30	External pin interrupt ch.2
31	External pin interrupt ch.3

■ Interrupt signals input to DSTC

Table 2-3 shows interrupt signals input as transfer request signals to the DSTC. Numbers in the table correspond to the numbers of the DREQENB[n] Registers of the DSTC.

Table 2-3 List of interrupt signals input to DSTC

Number	Interrupt signal name
0	External pin interrupt ch.0
1	External pin interrupt ch.1
2	External pin interrupt ch.2
3	External pin interrupt ch.3
4	External pin interrupt ch.4
5	External pin interrupt ch.5
6	External pin interrupt ch.6
7	External pin interrupt ch.7
8	External pin interrupt ch.8
9	External pin interrupt ch.9
10	External pin interrupt ch.10
11	External pin interrupt ch.11
12	External pin interrupt ch.12
13	External pin interrupt ch.13
14	External pin interrupt ch.14
15	External pin interrupt ch.15
16	Base timer ch.0 source 0 (IRQ0) interrupt
17	Base timer ch.0 source 1 (IRQ1) interrupt
18	Base timer ch.1 source 0 (IRQ0) interrupt
19	Base timer ch.1 source 1 (IRQ1) interrupt
20	Base timer ch.2 source 0 (IRQ0) interrupt
21	Base timer ch.2 source 1 (IRQ1) interrupt
22	Base timer ch.3 source 0 (IRQ0) interrupt
23	Base timer ch.3 source 1 (IRQ1) interrupt
24	Base timer ch.4 source 0 (IRQ0) interrupt
25	Base timer ch.4 source 1 (IRQ1) interrupt
26	Base timer ch.5 source 0 (IRQ0) interrupt
27	Base timer ch.5 source 1 (IRQ1) interrupt
28	Base timer ch.6 source 0 (IRQ0) interrupt
29	Base timer ch.6 source 1 (IRQ1) interrupt
30	Base timer ch.7 source 0 (IRQ0) interrupt
31	Base timer ch.7 source 1 (IRQ1) interrupt
32	MFT unit 0 WFG timer 10 interrupt
33	MFT unit 0 WFG timer 32 interrupt
34	MFT unit 0 WFG timer 54 interrupt
35	MFT unit 0 FRT ch.0 peak value detection interrupt
36	MFT unit 0 FRT ch.1 peak value detection interrupt
37	MFT unit 0 FRT ch.2 peak value detection interrupt

Number	Interrupt signal name
38	MFT unit 0 FRT ch.0 zero detection interrupt
39	MFT unit 0 FRT ch.1 zero detection interrupt
40	MFT unit 0 FRT ch.2 zero detection interrupt
41	MFT unit 0 ICU ch.0 input edge detection interrupt
42	MFT unit 0 ICU ch.1 input edge detection interrupt
43	MFT unit 0 ICU ch.2 input edge detection interrupt
44	MFT unit 0 ICU ch.3 input edge detection interrupt
45	MFT unit 0 OCU ch.0 match detection interrupt
46	MFT unit 0 OCU ch.1 match detection interrupt
47	MFT unit 0 OCU ch.2 match detection interrupt
48	MFT unit 0 OCU ch.3 match detection interrupt
49	MFT unit 0 OCU ch.4 match detection interrupt
50	MFT unit 0 OCU ch.5 match detection interrupt
51	MFT unit 1 WFG timer 10 interrupt
52	MFT unit 1 WFG timer 32 interrupt
53	MFT unit 1 WFG timer 54 interrupt
54	MFT unit 1 FRT ch.0 peak value detection interrupt
55	MFT unit 1 FRT ch.1 peak value detection interrupt
56	MFT unit 1 FRT ch.2 peak value detection interrupt
57	MFT unit 1 FRT ch.0 zero detection interrupt
58	MFT unit 1 FRT ch.1 zero detection interrupt
59	MFT unit 1 FRT ch.2 zero detection interrupt
60	MFT unit 1 ICU ch.0 input edge detection interrupt
61	MFT unit 1 ICU ch.1 input edge detection interrupt
62	MFT unit 1 ICU ch.2 input edge detection interrupt
63	MFT unit 1 ICU ch.3 input edge detection interrupt
64	MFT unit 1 OCU ch.0 match detection interrupt
65	MFT unit 1 OCU ch.1 match detection interrupt
66	MFT unit 1 OCU ch.2 match detection interrupt
67	MFT unit 1 OCU ch.3 match detection interrupt
68	MFT unit 1 OCU ch.4 match detection interrupt
69	MFT unit 1 OCU ch.5 match detection interrupt
70	PPG ch.0 interrupt
71	PPG ch.2 interrupt
72	PPG ch.4 interrupt
73	PPG ch.8 interrupt
74	PPG ch.10 interrupt
75	PPG ch.12 interrupt
76	QPRC ch.0 PC match interrupt
77	QPRC ch.0 PC & RC match interrupt
78	QPRC ch.0 overflow / underflow / zero index interrupt

Number	Interrupt signal name
79	QPRC ch.0 count inversion interrupt
80	QPRC ch.0 out-of-range interrupt
81	QPRC ch.0 PC match & RC match interrupt
82	QPRC ch.1 PC match interrupt
83	QPRC ch.1 PC & RC match interrupt
84	QPRC ch.1 overflow / underflow / zero index interrupt
85	QPRC ch.1 count inversion interrupt
86	QPRC ch.1 out-of-range interrupt
87	QPRC ch.1 PC match & RC match interrupt
88	Watch counter interrupt
89	MFS ch.0 reception interrupt
90	MFS ch.0 transmission interrupt
91	MFS ch.1 reception interrupt
92	MFS ch.1 transmission interrupt
93	MFS ch.2 reception interrupt
94	MFS ch.2 transmission interrupt
95	MFS ch.3 reception interrupt
96	MFS ch.3 transmission interrupt
97	MFS ch.4 reception interrupt
98	MFS ch.4 transmission interrupt
99	MFS ch.5 reception interrupt
100	MFS ch.5 transmission interrupt
101	MFS ch.6 reception interrupt
102	MFS ch.6 transmission interrupt
103	MFS ch.7 reception interrupt
104	MFS ch.7 transmission interrupt
105	USB ch.0 function endpoint 1 DRQ interrupt
106	USB ch.0 function endpoint 2 DRQ interrupt
107	USB ch.0 function endpoint 3 DRQ interrupt
108	USB ch.0 function endpoint 4 DRQ interrupt
109	USB ch.0 function endpoint 5 DRQ interrupt
110	A/D converter unit 0 priority conversion interrupt
111	A/D converter unit 0 scan conversion interrupt
112	A/D converter unit 1 priority conversion interrupt
113	A/D converter unit 1 scan conversion interrupt
114	A/D converter unit 2 priority conversion interrupt
115	A/D converter unit 2 scan conversion interrupt
116 to 127	Reserved
128	External pin interrupt ch.16
129	External pin interrupt ch.17
130	External pin interrupt ch.18

Number	Interrupt signal name
131	External pin interrupt ch.19
132	External pin interrupt ch.20
133	External pin interrupt ch.21
134	External pin interrupt ch.22
135	External pin interrupt ch.23
136	External pin interrupt ch.24
137	External pin interrupt ch.25
138	External pin interrupt ch.26
139	External pin interrupt ch.27
140	External pin interrupt ch.28
141	External pin interrupt ch.29
142	External pin interrupt ch.30
143	External pin interrupt ch.31
144	Base timer ch.8 source 0 (IRQ0) interrupt
145	Base timer ch.8 source 1 (IRQ1) interrupt
146	Base timer ch.9 source 0 (IRQ0) interrupt
147	Base timer ch.9 source 1 (IRQ1) interrupt
148	Base timer ch.10 source 0 (IRQ0) interrupt
149	Base timer ch.10 source 1 (IRQ1) interrupt
150	Base timer ch.11 source 0 (IRQ0) interrupt
151	Base timer ch.11 source 1 (IRQ1) interrupt
152	Base timer ch.12 source 0 (IRQ0) interrupt
153	Base timer ch.12 source 1 (IRQ1) interrupt
154	Base timer ch.13 source 0 (IRQ0) interrupt
155	Base timer ch.13 source 1 (IRQ1) interrupt
156	Base timer ch.14 source 0 (IRQ0) interrupt
157	Base timer ch.14 source 1 (IRQ1) interrupt
158	Base timer ch.15 source 0 (IRQ0) interrupt
159	Base timer ch.15 source 1 (IRQ1) interrupt
160	MFT unit 2 WFG timer 10 interrupt
161	MFT unit 2 WFG timer 32 interrupt
162	MFT unit 2 WFG timer 54 interrupt
163	MFT unit 2 FRT ch.0 peak value detection interrupt
164	MFT unit 2 FRT ch.1 peak value detection interrupt
165	MFT unit 2 FRT ch.2 peak value detection interrupt
166	MFT unit 2 FRT ch.0 zero detection interrupt
167	MFT unit 2 FRT ch.1 zero detection interrupt
168	MFT unit 2 FRT ch.2 zero detection interrupt
169	MFT unit 2 ICU ch.0 input edge detection interrupt
170	MFT unit 2 ICU ch.1 input edge detection interrupt
171	MFT unit 2 ICU ch.2 input edge detection interrupt

Number	Interrupt signal name
172	MFT unit 2 ICU ch.3 input edge detection interrupt
173	MFT unit 2 OCU ch.0 match detection interrupt
174	MFT unit 2 OCU ch.1 match detection interrupt
175	MFT unit 2 OCU ch.2 match detection interrupt
176	MFT unit 2 OCU ch.3 match detection interrupt
177	MFT unit 2 OCU ch.4 match detection interrupt
178	MFT unit 2 OCU ch.5 match detection interrupt
179	PPG ch.16 interrupt
180	PPG ch.18 interrupt
181	PPG ch.20 interrupt
182	QPRC ch.2 PC match interrupt
183	QPRC ch.2 PC & RC match interrupt
184	QPRC ch.2 overflow / underflow / zero index interrupt
185	QPRC ch.2 count inversion interrupt
186	QPRC ch.2 out-of-range interrupt
187	QPRC ch.2 PC match & RC match interrupt
188	QPRC ch.3 PC match interrupt
189	QPRC ch.3 PC & RC match interrupt
190	QPRC ch.3 overflow / underflow / zero index interrupt
191	QPRC ch.3 count inversion interrupt
192	QPRC ch.3 out-of-range interrupt
193	QPRC ch.3 PC match & RC match interrupt
194	MFS ch.8 reception interrupt
195	MFS ch.8 transmission interrupt
196	MFS ch.9 reception interrupt
197	MFS ch.9 transmission interrupt
198	MFS ch.10 reception interrupt
199	MFS ch.10 transmission interrupt
200	MFS ch.11 reception interrupt
201	MFS ch.11 transmission interrupt
202	USB ch.1 function endpoint 1 DRQ interrupt
203	USB ch.1 function endpoint 2 DRQ interrupt
204	USB ch.1 function endpoint 3 DRQ interrupt
205	USB ch.1 function endpoint 4 DRQ interrupt
206	USB ch.1 function endpoint 5 DRQ interrupt
207 to 208	Reserved
209	MFS ch.12 reception interrupt
210	MFS ch.12 transmission interrupt
211	MFS ch.13 reception interrupt
212	MFS ch.13 transmission interrupt
213	MFS ch.14 reception interrupt

Number	Interrupt signal name
214	MFS ch.14 transmission interrupt
215	MFS ch.15 reception interrupt
216	MFS ch.15 transmission interrupt
217 to 255	Reserved

3. Registers

This section explains the respective details of registers.

■ Register list

Abbreviation	Register name	Reference
DRQSEL	DMAC DMA Request Selection Register	3.1
IRQ003SEL	Relocate Interrupt Selection Register	3.2
IRQ004SEL	Relocate Interrupt Selection Register	3.2
IRQ005SEL	Relocate Interrupt Selection Register	3.2
IRQ006SEL	Relocate Interrupt Selection Register	3.2
IRQ007SEL	Relocate Interrupt Selection Register	3.2
IRQ008SEL	Relocate Interrupt Selection Register	3.2
IRQ009SEL	Relocate Interrupt Selection Register	3.2
IRQ010SEL	Relocate Interrupt Selection Register	3.2
EXC02MON	EXC02 Batch Read Register	3.3
IRQ000MON	IRQ000 Batch Read Register	3.4
IRQ001MON	IRQ001 Batch Read Register	3.5
IRQ002MON	IRQ002 Batch Read Register	3.6
IRQ003MON	IRQ003 Batch Read Register	3.7
IRQ004MON	IRQ004 Batch Read Register	
IRQ005MON	IRQ005 Batch Read Register	
IRQ006MON	IRQ006 Batch Read Register	
IRQ007MON	IRQ007 Batch Read Register	
IRQ008MON	IRQ008 Batch Read Register	
IRQ009MON	IRQ009 Batch Read Register	
IRQ010MON	IRQ010 Batch Read Register	
IRQ011MON	IRQ011 Batch Read Register	3.8
IRQ012MON	IRQ012 Batch Read Register	
IRQ013MON	IRQ013 Batch Read Register	
IRQ014MON	IRQ014 Batch Read Register	
IRQ015MON	IRQ015 Batch Read Register	
IRQ016MON	IRQ016 Batch Read Register	
IRQ017MON	IRQ017 Batch Read Register	
IRQ018MON	IRQ018 Batch Read Register	
IRQ019MON	IRQ019 Batch Read Register	3.9

Abbreviation	Register name	Reference
IRQ020MON	IRQ020 Batch Read Register	3.9
IRQ021MON	IRQ021 Batch Read Register	3.10
IRQ022MON	IRQ022 Batch Read Register	
IRQ023MON	IRQ023 Batch Read Register	
IRQ024MON	IRQ024 Batch Read Register	3.11
IRQ025MON	IRQ025 Batch Read Register	3.12
IRQ026MON	IRQ026 Batch Read Register	3.13
IRQ027MON	IRQ027 Batch Read Register	3.14
IRQ028MON	IRQ028 Batch Read Register	3.11
IRQ029MON	IRQ029 Batch Read Register	3.12
IRQ030MON	IRQ030 Batch Read Register	3.13
IRQ031MON	IRQ031 Batch Read Register	3.14
IRQ032MON	IRQ032 Batch Read Register	3.11
IRQ033MON	IRQ033 Batch Read Register	3.12
IRQ034MON	IRQ034 Batch Read Register	3.13
IRQ035MON	IRQ035 Batch Read Register	3.14
IRQ036MON	IRQ036 Batch Read Register	3.15
IRQ037MON	IRQ037 Batch Read Register	
IRQ038MON	IRQ038 Batch Read Register	
IRQ039MON	IRQ039 Batch Read Register	3.16
IRQ040MON	IRQ040 Batch Read Register	
IRQ041MON	IRQ041 Batch Read Register	
IRQ042MON	IRQ042 Batch Read Register	
IRQ043MON	IRQ043 Batch Read Register	
IRQ044MON	IRQ044 Batch Read Register	
IRQ045MON	IRQ045 Batch Read Register	
IRQ046MON	IRQ046 Batch Read Register	
IRQ047MON	IRQ047 Batch Read Register	3.17
IRQ048MON	IRQ048 Batch Read Register	3.18
IRQ049MON	IRQ049 Batch Read Register	3.19
IRQ050MON	IRQ050 Batch Read Register	3.20
IRQ051MON	IRQ051 Batch Read Register	3.8
IRQ052MON	IRQ052 Batch Read Register	
IRQ053MON	IRQ053 Batch Read Register	
IRQ054MON	IRQ054 Batch Read Register	

Abbreviation	Register name	Reference
IRQ055MON	IRQ055 Batch Read Register	3.8
IRQ056MON	IRQ056 Batch Read Register	
IRQ057MON	IRQ057 Batch Read Register	
IRQ058MON	IRQ058 Batch Read Register	
IRQ059MON	IRQ059 Batch Read Register	3.21
IRQ060MON	IRQ060 Batch Read Register	3.22
IRQ061MON	IRQ061 Batch Read Register	3.23
IRQ062MON	IRQ062 Batch Read Register	3.22
IRQ063MON	IRQ063 Batch Read Register	3.23
IRQ064MON	IRQ064 Batch Read Register	3.22
IRQ065MON	IRQ065 Batch Read Register	3.23
IRQ066MON	IRQ066 Batch Read Register	3.22
IRQ067MON	IRQ067 Batch Read Register	3.23
IRQ068MON	IRQ068 Batch Read Register	3.22
IRQ069MON	IRQ069 Batch Read Register	3.23
IRQ070MON	IRQ070 Batch Read Register	3.22
IRQ071MON	IRQ071 Batch Read Register	3.23
IRQ072MON	IRQ072 Batch Read Register	3.22
IRQ073MON	IRQ073 Batch Read Register	3.23
IRQ074MON	IRQ074 Batch Read Register	3.22
IRQ075MON	IRQ075 Batch Read Register	3.23
IRQ076MON	IRQ076 Batch Read Register	3.24
IRQ077MON	IRQ077 Batch Read Register	3.24
IRQ078MON	IRQ078 Batch Read Register	3.25
IRQ079MON	IRQ079 Batch Read Register	3.26
IRQ080MON	IRQ080 Batch Read Register	3.27
IRQ081MON	IRQ081 Batch Read Register	3.27
IRQ082MON	IRQ082 Batch Read Register	3.28
IRQ083MON	IRQ083 Batch Read Register	3.29
IRQ084MON	IRQ084 Batch Read Register	
IRQ085MON	IRQ085 Batch Read Register	
IRQ086MON	IRQ086 Batch Read Register	
IRQ087MON	IRQ087 Batch Read Register	
IRQ088MON	IRQ088 Batch Read Register	
IRQ089MON	IRQ089 Batch Read Register	

Abbreviation	Register name	Reference
IRQ090MON	IRQ090 Batch Read Register	3.29
IRQ091MON	IRQ091 Batch Read Register	3.30
IRQ092MON	IRQ092 Batch Read Register	3.31
IRQ093MON	IRQ093 Batch Read Register	
IRQ094MON	IRQ094 Batch Read Register	
IRQ095MON	IRQ095 Batch Read Register	
IRQ096MON	IRQ096 Batch Read Register	3.9
IRQ097MON	IRQ097 Batch Read Register	
IRQ098MON	IRQ098 Batch Read Register	3.16
IRQ099MON	IRQ099 Batch Read Register	
IRQ100MON	IRQ100 Batch Read Register	
IRQ101MON	IRQ101 Batch Read Register	
IRQ102MON	IRQ102 Batch Read Register	3.32
IRQ103MON	IRQ103 Batch Read Register	3.22
IRQ104MON	IRQ104 Batch Read Register	3.23
IRQ105MON	IRQ105 Batch Read Register	3.22
IRQ106MON	IRQ106 Batch Read Register	3.23
IRQ107MON	IRQ107 Batch Read Register	3.22
IRQ108MON	IRQ108 Batch Read Register	3.23
IRQ109MON	IRQ109 Batch Read Register	3.22
IRQ110MON	IRQ110 Batch Read Register	3.23
IRQ111MON	IRQ111 Batch Read Register	3.24
IRQ112MON	IRQ112 Batch Read Register	3.35
IRQ113MON	IRQ113 Batch Read Register	3.25
IRQ114MON	IRQ114 Batch Read Register	3.26
IRQ115MON	IRQ115 Batch Read Register	3.35
IRQ116MON	IRQ116 Batch Read Register	
IRQ117MON	IRQ117 Batch Read Register	
IRQ118MON	IRQ118 Batch Read Register	3.33
IRQ119MON	IRQ119 Batch Read Register	3.34
IRQ120MON	IRQ120 Batch Read Register	3.22
IRQ121MON	IRQ121 Batch Read Register	3.23
IRQ122MON	IRQ122 Batch Read Register	3.22
IRQ123MON	IRQ123 Batch Read Register	3.23
IRQ124MON	IRQ124 Batch Read Register	3.22

Abbreviation	Register name	Reference
IRQ125MON	IRQ125 Batch Read Register	3.23
IRQ126MON	IRQ126 Batch Read Register	3.22
IRQ127MON	IRQ127 Batch Read Register	3.23
ODDPKS	USB ch.0 Odd Packet Size DMA Enable Register	3.36

3.1. DMAC DMA Request Selection Register (DRQSEL)

The DMA Request Selection Register (DRQSEL) enables using an interrupt signal from a peripheral function as a transfer request to the DMAC. Such interrupt signal can be transferred through the DMA transfer by the DMAC.

■ Register configuration

bit	31	0
Field	DRQSEL[31:0]	
Attribute	R/W	
Initial value	0x00000000	

■ Register function

[bit31:0] DRQSEL[31:0]

The connection in the SEL1 selector in Figure 1-1 changes according to the setting of a bit in the DRQSEL Register. If the setting of a bit is "1", an interrupt signal is connected to the SEL1 selector as a transfer request signal to the DMAC. If the setting of a bit is "0", an interrupt signal is connected to the SEL1 selector as an interrupt signal to the NVIC or as a transfer request signal to the DSTC.

bit	Corresponding interrupt signal name
31	External pin interrupt ch.3
30	External pin interrupt ch.2
29	External pin interrupt ch.1
28	External pin interrupt ch.0
27	MFS ch.7 transmission interrupt
26	MFS ch.7 reception interrupt
25	MFS ch.6 transmission interrupt
24	MFS ch.6 reception interrupt
23	MFS ch.5 transmission interrupt
22	MFS ch.5 reception interrupt
21	MFS ch.4 transmission interrupt
20	MFS ch.4 reception interrupt
19	MFS ch.3 transmission interrupt
18	MFS ch.3 reception interrupt
17	MFS ch.2 transmission interrupt
16	MFS ch.2 reception interrupt
15	MFS ch.1 transmission interrupt
14	MFS ch.1 reception interrupt
13	MFS ch.0 transmission interrupt
12	MFS ch.0 reception interrupt
11	Base timer ch.6 source 0 (IRQ0) interrupt
10	Base timer ch.4 source 0 (IRQ0) interrupt
9	Base timer ch.2 source 0 (IRQ0) interrupt
8	Base timer ch.0 source 0 (IRQ0) interrupt
7	A/D converter unit 2 scan conversion interrupt
6	A/D converter unit 1 scan conversion interrupt
5	A/D converter unit 0 scan conversion interrupt

bit	Corresponding interrupt signal name
4	USB ch.0 function endpoint 5 DRQ interrupt
3	USB ch.0 function endpoint 4 DRQ interrupt
2	USB ch.0 function endpoint 3 DRQ interrupt
1	USB ch.0 function endpoint 2 DRQ interrupt
0	USB ch.0 function endpoint 1 DRQ interrupt

<Notes>

- ・If an interrupt signal is selected as a transfer request to the DMAC, the read value of the bit in the interrupt request batch read register (IRQxxxMON, xxx = 000 to 127) corresponding to that interrupt signal is "0", regardless of whether the interrupt of that interrupt signal occurs.
- ・Before modifying the settings of DRQSEL, clear the interrupt request signals from the peripheral functions related to the settings to be modified.
- ・The DMA transfer of an interrupt signal not included in the settings of DRQSEL cannot be started by the hardware start by the DMAC.
- ・If an interrupt signal is connected to the DMAC according to the settings of DRQSEL, it cannot be connected to the DSTC.

3.2. Relocate Interrupt Selection Register (IRQxxxSEL)

The Relocate Interrupt Selection Register (IRQxxxSEL) is a register selecting a relocate interrupt to be input to one of the exceptions between exception no. 19 and exception no. 24 (IRQ003 to IRQ010). There are eight Relocate Interrupt Selection Registers, IRQ003SEL to IRQ010SEL. The IRQ003SEL to IRQ010SEL Registers select the interrupts to be input to IRQ003 to IRQ010 respectively.

■ Register configuration

bit	31 24	23 16	15 8	7 0
Field	Reserved	SELBIT[7:0]	Reserved	SELIRQ[7:0]
Attribute	R	R/W	R	R/W
Initial value	0x00	0x00	0x00	0x00

■ Register function

[bit31:24] Reserved: Reserved bits

A reserved bit reads "0".

[bit23:16] SELBIT[7:0]

A bit in SELBIT[7:0] specifies the bit position to which one of the interrupt sources aggregated by logical OR is moved to the relocate interrupt. The peripheral interrupt to be generated by those interrupt sources is specified in SELIRQ[7:0].

bit	Value	Description
7	0	Bit7 of the interrupt source is not moved to the relocate interrupt.
	1	Bit7 of the interrupt source is moved to bit7 of the relocate interrupt.
6	0	Bit6 of the interrupt source is not moved to the relocate interrupt.
	1	Bit6 of the interrupt source is moved to bit6 of the relocate interrupt.
5	0	Bit5 of the interrupt source is not moved to the relocate interrupt.
	1	Bit5 of the interrupt source is moved to bit5 of the relocate interrupt.
4	0	Bit4 of the interrupt source is not moved to the relocate interrupt.
	1	Bit4 of the interrupt source is moved to bit4 of the relocate interrupt.
3	0	Bit3 of the interrupt source is not moved to the relocate interrupt.
	1	Bit3 of the interrupt source is moved to bit3 of the relocate interrupt.
2	0	Bit2 of the interrupt source is not moved to the relocate interrupt.
	1	Bit2 of the interrupt source is moved to bit2 of the relocate interrupt.
1	0	Bit1 of the interrupt source is not moved to the relocate interrupt.
	1	Bit1 of the interrupt source is moved to bit1 of the relocate interrupt.
0	0	Bit0 of the interrupt source is not moved to the relocate interrupt.
	1	Bit0 of the interrupt source is moved to bit0 of the relocate interrupt.

[bit15:8] Reserved: Reserved bits

A reserved bit reads "0".

[bit7:0] SELIRQ[7:0]

The SELIRQ[7:0] bits specify the IRQ no. of a peripheral interrupt to be relocated.

Value	Description
0x0B to 0x3F	IRQ no. of a peripheral interrupt to be relocated
0x00	No peripheral interrupt to be relocated
Value other than the above	Setting prohibited

■ Examples of and notes on setting

For instance, the SELIRQ[7:0] bits and the SELBIT[7:0] bits in the IRQ003SEL Register are set to "25" and "0b00000010" respectively. According to these settings, the interrupt (MFT unit 0 FRT ch.1 zero detection interrupt) assigned to bit1 in the IRQ025MON Register is relocated to bit1 in the IRQ003MON Register. If an MFT unit 0 FRT ch.1 zero detection interrupt is generated, an IRQ003 interrupt is generated and "1" can be read out from bit1 in the IRQ003MON Register. From the original interrupt position (bit1 in the IRQ025MON Register), no interrupt is generated any more. From bit1 in the IRQ025MON Register, "0" is always read out.

On the other hand, the interrupt (MFT unit 0 FRT ch.0 zero detection interrupt) can still be received from bit0 in the IRQ025MON Register, to which the interrupt is assigned. Therefore, the two interrupt sources aggregated by logical OR to bit0 and bit1 in the IRQ025MON Register can be received from the IRQ003MON Register and the IRQ025MON Register respectively.

The same IRQ no. can be specified in the SELIRQ[7:0] bits in more than one Relocate Interrupt Selection Register from IRQ003SEL to IRQ010SEL. For instance, in addition to the settings of the IRQ003SEL Register mentioned above, if the SELIRQ[7:0] bits and the SELBIT[7:0] bits in the IRQ003SEL Register are set to "25" and "0b00000100" respectively, the interrupt (MFT unit 0 FRT ch.2 zero detection interrupt) assigned to bit2 in the IRQ025MON Register can be received from bit2 in the IRQ004MON Register. Therefore, the three interrupt sources aggregated by logical OR to bit0, bit1 and bit2 in the IRQ025MON Register can be received from the IRQ003MON Register, the IRQ004MON Register and the IRQ025MON Register respectively.

However, one interrupt source cannot be selected for different relocate interrupts. (If the same IRQ no. is specified in the SELIRQ[7:0] bits in different Relocate Interrupt Selection Registers, their respective settings of the SELBIT[7:0] bits cannot be the same.)

If more than one bit of the SELBIT[7:0] bits are set to "1", multiple interrupt sources selected are aggregated by logical OR to become a relocate interrupt.

A bus reset initializes the settings of all Relocate Interrupt Selection Registers (IRQQ003SEL to IRQ010SEL). After the relocate interrupt selection settings in the IRQ003SEL to IRQ010SEL Registers have been initialized, the IRQ003 to IRQ010 interrupts are no longer generated.

Before modifying the settings of any of the IRQ003SEL to IRQ010SEL Registers, ensure that no interrupt signal has been asserted.

3.3. EXC02 Batch Read Register (EXC02MON)

The EXC02 Batch Read Register (EXC02MON) can read out at once all interrupts (NMI and hardware watchdog interrupt) assigned to exception no. 2.

■ Register configuration

bit	31 … 8
Field	Reserved
Attribute	R
Initial value	0x000000

bit	7	6	5	4	3	2	1	0
Field	Reserved						HWINT	NMI
Attribute	R						R	R
Initial value	000000						0	0

■ Register function

[bit31:2] Reserved: Reserved bits

A reserved bit reads "0".

[bit1] HWINT

Value	Description
0	There is no interrupt request of the hardware watchdog timer.
1	An interrupt request of the hardware watchdog timer has been made.

[bit0] NMI

Value	Description
0	There is no interrupt request of the NMIX external pin.
1	An interrupt request of the NMIX external pin has been made.

3.4. IRQ000 Batch Read Register (IRQ000MON)

The IRQ000 Batch Read Register (IRQ000MON) can read out the interrupt (clock supervisor anomalous frequency detection interrupt) assigned to exception no. 16.

■ Register configuration

bit	31	8
Field	Reserved	
Attribute	R	
Initial value	0x000000	

bit	7	6	5	4	3	2	1	0
Field	Reserved							FCSINT
Attribute	R							R
Initial value	0000000							0

■ Register function

[bit31:1] Reserved: Reserved bits

A reserved bit reads "0".

[bit0] FCSINT

Value	Description
0	There is no interrupt request of the anomalous frequency detected by the CSV.
1	An interrupt request of the anomalous frequency detected by the CSV has been made.

3.5. IRQ001 Batch Read Register (IRQ001MON)

The IRQ001 Batch Read Register (IRQ001MON) can read out the interrupt (software watchdog interrupt) assigned to exception no. 17.

■ Register configuration

bit	31	8
Field	Reserved	
Attribute	R	
Initial value	0x000000	

bit	7	6	5	4	3	2	1	0
Field	Reserved							SWWDTINT
Attribute	R							R
Initial value	0000000							0

■ Register function

[bit31:1] Reserved: Reserved bits

A reserved bit reads "0".

[bit0] SWWDTINT

Value	Description
0	There is no interrupt request of the software watchdog timer.
1	An interrupt request of the software watchdog timer has been made.

3.6. IRQ002 Batch Read Register (IRQ002MON)

The IRQ002 Batch Read Register (IRQ002MON) can read out the interrupt (low-voltage detection interrupt) assigned to exception no. 18.

■ Register configuration

bit	31	8
Field	Reserved	
Attribute	R	
Initial value	0x000000	

bit	7	6	5	4	3	2	1	0
Field	Reserved							LVDINT
Attribute	R							R
Initial value	0000000							0

■ Register function

[bit31:1] Reserved: Reserved bits

A reserved bit reads "0".

[bit0] LVDINT

Value	Description
0	There is no low-voltage detection (LVD) interrupt request.
1	A low-voltage detection (LVD) interrupt request has been made.

3.7. IRQ003/004/005/006/007/008/009/010 Batch Read Register (IRQxxxMON)

The IRQ003MON to IRQMON010 Registers can read out at once the interrupts (relocate interrupts) assigned to exception no. 19 to no. 26 respectively.

■ Register configuration

bit	31	8
Field	Reserved	
Attribute	R	
Initial value	0x000000	

bit	7	6	5	4	3	2	1	0
Field	IRQBIT[7:0]							
Attribute	R							
Initial value	0X000000							

■ Register function

[bit31:8] Reserved: Reserved bits

A reserved bit reads "0".

[bit7:0] IRQBIT[7:0]

Each of the IRQBIT[7:0] bits indicates the status of the relocate interrupt selected in the corresponding bit in the IRQxxxSEL Register.

Value	Description
0	There is no interrupt request of the interrupt selected in the corresponding bit in the IRQxxxSEL Register.
1	An interrupt request of the interrupt selected in the corresponding bit in the IRQxxxSEL Register has been made.

3.8. IRQ011/012/013/014/015/016/017/018/051/052/053/054/055/056/057/058 Batch Read Register (IRQxxxMON)

The IRQ011MON to IRQ018MON Registers can read out at once the interrupts (external pin interrupt ch.0 to ch.7) assigned to exception no. 27 to no. 34 respectively.
The IRQ051MON to IRQ058MON Registers can read out at once the interrupts (external pin interrupt ch.8 to ch.15) assigned to exception no. 67 to no. 74 respectively.

■ Register configuration

bit	31 … 8
Field	Reserved
Attribute	R
Initial value	0x000000

bit	7	6	5	4	3	2	1	0
Field	Reserved							EXTINT
Attribute	R							R
Initial value	0000000							0

■ Register function

[bit31:1] Reserved: Reserved bits
A reserved bit reads "0".

[bit0] EXTINT

Value	Description
0	There is no interrupt request of the external pin interrupt channel corresponding to the IRQxxxMON Register.
1	An interrupt request of the external pin interrupt channel corresponding to the IRQxxxMON Register has been made.

See Table 2-1 for the relationship between exception no. and interrupt.

3.9. IRQ019/020/096/097 Batch Read Register (IRQxxxMON)

The IRQ019MON, IRQ020MON, IRQ096MON and IRQ097MON Registers can read out at once the interrupts (QPRC ch.0 to ch.3) assigned to exception no. 35, no. 36, no. 112 and no. 113 respectively.

■ Register configuration

bit	31 … 8
Field	Reserved
Attribute	R
Initial value	0x000000

bit	7	6	5	4	3	2	1	0
Field	Reserved		QUDINT					
Attribute	R		R					
Initial value	00		000000					

■ Register function

[bit31:6] Reserved: Reserved bits

A reserved bit reads "0".

[bit5:0] QUDINT

bit	Value	Description
5	0	There is no PC match & RC match interrupt request of a QPRC channel corresponding to the IRQxxxMON Register.
	1	A PC match & RC match interrupt request of a QPRC channel corresponding to the IRQxxxMON Register has been made.
4	0	There is no out-of-range interrupt request of a QPRC channel corresponding to the IRQxxxMON Register.
	1	An out-of-range interrupt request of a QPRC channel corresponding to the IRQxxxMON Register has been made.
3	0	There is no count inversion interrupt request of a QPRC channel corresponding to the IRQxxxMON Register.
	1	A count inversion interrupt request of a QPRC channel corresponding to the IRQxxxMON Register has been made.
2	0	There is no overflow / underflow / zero index interrupt request of a QPRC channel corresponding to the IRQxxxMON Register.
	1	An overflow / underflow / zero index interrupt request of a QPRC channel corresponding to the IRQxxxMON Register has been made.
1	0	There is no PC & RC match interrupt request of a QPRC channel corresponding to the IRQxxxMON Register.
	1	A PC & RC match interrupt request of a QPRC channel corresponding to the IRQxxxMON Register has been made.
0	0	There is no PC match interrupt request of a QPRC channel corresponding to the IRQxxxMON Register.
	1	A PC match interrupt request of a QPRC channel corresponding to the IRQxxxMON Register has been made.

See Table 2-1 for the relationship between exception no. and interrupt.

3.10. IRQ021/022/023 Batch Read Register (IRQxxxMON)

The IRQ021MON to IRQ023MON Registers can read out at once the interrupts (WFG timer interrupts and DTIF interrupts of MFT unit 0 to MFT unit 2) assigned to exception no. 37 to no. 39 respectively.

■ Register configuration

bit	31 … 8
Field	Reserved
Attribute	R
Initial value	0x000000

bit	7	6	5	4	3	2	1	0
Field	Reserved				WAVEINT			
Attribute	R				R			
Initial value	0000				0000			

■ Register function

[bit31:4] Reserved: Reserved bits

A reserved bit reads "0".

[bit3:0] WAVEINT

bit	Value	Description
3	0	There is no interrupt request of WFG timer 54 of the MFT unit corresponding to the IRQxxxMON Register.
	1	An interrupt request of WFG timer 54 of the MFT unit corresponding to the IRQxxxMON Register has been made.
2	0	There is no interrupt request of WFG timer 32 of the MFT unit corresponding to the IRQxxxMON Register.
	1	An interrupt request of WFG timer 32 of the MFT unit corresponding to the IRQxxxMON Register has been made.
1	0	There is no interrupt request of WFG timer 10 of the MFT unit corresponding to the IRQxxxMON Register.
	1	An interrupt request of WFG timer 10 of the MFT unit corresponding to the IRQxxxMON Register has been made.
0	0	There is no interrupt request of the DTIF (motor emergency stop) of the MFT unit corresponding to the IRQxxxMON Register.
	1	An interrupt request of the DTIF (motor emergency stop) of the MFT unit corresponding to the IRQxxxMON Register has been made.

See Table 2-1 for the relationship between exception no. and interrupt.

3.11. IRQ024/028/032 Batch Read Register (IRQxxxMON)

The IRQ024MON, IRQ028MON and IRQ032MON Registers can read out at once the interrupts (FRT ch.0 to ch.2 peak detection interrupts of MFT unit 0 to unit 2) assigned to exception no. 40, no. 44 and no. 48 respectively.

■ Register configuration

bit	31 … 8
Field	Reserved
Attribute	R
Initial value	0x000000

bit	7	6	5	4	3	2	1	0
Field	Reserved					FRT_PEAK_INT		
Attribute	R					R		
Initial value	00000					000		

■ Register function

[bit31:3] Reserved: Reserved bits

A reserved bit reads "0".

[bit2:0] FRT_PEAK_INT

bit	Value	Description
2	0	There is no FRT ch.2 peak value detection interrupt request of the MFT unit corresponding to the IRQxxxMON Register.
	1	An FRT ch.2 peak value detection interrupt request of the MFT unit corresponding to the IRQxxxMON Register has been made.
1	0	There is no FRT ch.1 peak value detection interrupt request of the MFT unit corresponding to the IRQxxxMON Register.
	1	An FRT ch.1 peak value detection interrupt request of the MFT unit corresponding to the IRQxxxMON Register has been made.
0	0	There is no FRT ch.0 peak value detection interrupt request of the MFT unit corresponding to the IRQxxxMON Register.
	1	An FRT ch.0 peak value detection interrupt request of the MFT unit corresponding to the IRQxxxMON Register has been made.

See Table 2-1 for the relationship between exception no. and interrupt.

3.12. IRQ025/029/033 Batch Read Register (IRQxxxMON)

The IRQ025MON, IRQ029MON and IRQ033MON Registers can read out at once the interrupts (FRT ch.0 to ch.2 zero detection interrupts of MFT unit 0 to unit 2) assigned to exception no. 41, no. 45 and no. 49 respectively.

■ Register configuration

bit	31	8
Field	Reserved	
Attribute	R	
Initial value	0x000000	

bit	7	6	5	4	3	2	1	0
Field	Reserved					FRT_ZERO_INT		
Attribute	R					R		
Initial value	00000					000		

■ Register function

[bit31:3] Reserved: Reserved bits

A reserved bit reads "0".

[bit2:0] FRT_ZERO_INT

bit	Value	Description
2	0	There is no FRT ch.2 zero detection interrupt request of the MFT unit corresponding to the IRQxxxMON Register.
	1	An FRT ch.2 zero detection interrupt request of the MFT unit corresponding to the IRQxxxMON Register has been made.
1	0	There is no FRT ch.1 zero detection interrupt request of the MFT unit corresponding to the IRQxxxMON Register.
	1	An FRT ch.1 zero detection interrupt request of the MFT unit corresponding to the IRQxxxMON Register has been made.
0	0	There is no FRT ch.0 zero detection interrupt request of the MFT unit corresponding to the IRQxxxMON Register.
	1	An FRT ch.0 zero detection interrupt request of the MFT unit corresponding to the IRQxxxMON Register has been made.

See Table 2-1 for the relationship between exception no. and interrupt.

3.13. IRQ026/030/034 Batch Read Register (IRQxxxMON)

The IRQ026MON, IRQ030MON and IRQ034MON Registers can read out at once the interrupts (ICU ch.0 to ch.2 input edge detection interrupts of MFT unit 0 to unit 2) assigned to exception no. 42, no. 46 and no. 50 respectively.

■ Register configuration

bit	31	8
Field	Reserved	
Attribute	R	
Initial value	0x000000	

bit	7	6	5	4	3	2	1	0
Field	Reserved				ICUINT			
Attribute	R				R			
Initial value	0000				0000			

■ Register function

[bit31:4] Reserved: Reserved bits

A reserved bit reads "0".

[bit3:0] ICUINT

bit	Value	Description
3	0	There is no ICU ch.3 input edge detection interrupt request of the MFT unit corresponding to the IRQxxxMON Register.
	1	An ICU ch.3 input edge detection interrupt request of the MFT unit corresponding to the IRQxxxMON Register has been made.
2	0	There is no ICU ch.2 input edge detection interrupt request of the MFT unit corresponding to the IRQxxxMON Register.
	1	An ICU ch.2 input edge detection interrupt request of the MFT unit corresponding to the IRQxxxMON Register has been made.
1	0	There is no ICU ch.1 input edge detection interrupt request of the MFT unit corresponding to the IRQxxxMON Register.
	1	An ICU ch.1 input edge detection interrupt request of the MFT unit corresponding to the IRQxxxMON Register has been made.
0	0	There is no ICU ch.0 input edge detection interrupt request of the MFT unit corresponding to the IRQxxxMON Register.
	1	An ICU ch.0 input edge detection interrupt request of the MFT unit corresponding to the IRQxxxMON Register has been made.

See Table 2-1 for the relationship between exception no. and interrupt.

3.14. IRQ027/031/035 Batch Read Register (IRQxxxMON)

The IRQ027MON, IRQ031MON and IRQ053MON Registers can read out at once the interrupts (OCU ch.0 to ch.2 match detection interrupts of MFT unit 0 to unit 2) assigned to exception no. 43, no. 47 and no. 51 respectively.

■ Register configuration

bit	31	8
Field	Reserved	
Attribute	R	
Initial value	0x000000	

bit	7	6	5	4	3	2	1	0
Field	Reserved		OCUINT					
Attribute	R		R					
Initial value	00		000000					

■ Register function

[bit31:6] Reserved: Reserved bits

A reserved bit reads "0".

[bit5:0] OCUINT

bit	Value	Description
5	0	There is no OCU ch.5 match detection interrupt request of the MFT unit corresponding to the IRQxxxMON Register.
	1	An OCU ch.5 match detection interrupt request of the MFT unit corresponding to the IRQxxxMON Register has been made.
4	0	There is no OCU ch.4 match detection interrupt request of the MFT unit corresponding to the IRQxxxMON Register.
	1	An OCU ch.4 match detection interrupt request of the MFT unit corresponding to the IRQxxxMON Register has been made.
3	0	There is no OCU ch.3 match detection interrupt request of the MFT unit corresponding to the IRQxxxMON Register.
	1	An OCU ch.3 match detection interrupt request of the MFT unit corresponding to the IRQxxxMON Register has been made.
2	0	There is no OCU ch.2 match detection interrupt request of the MFT unit corresponding to the IRQxxxMON Register.
	1	An OCU ch.2 match detection interrupt request of the MFT unit corresponding to the IRQxxxMON Register has been made.
1	0	There is no OCU ch.1 match detection interrupt request of the MFT unit corresponding to the IRQxxxMON Register.
	1	An OCU ch.1 match detection interrupt request of the MFT unit corresponding to the IRQxxxMON Register has been made.
0	0	There is no OCU ch.0 match detection interrupt request of the MFT unit corresponding to the IRQxxxMON Register.
	1	An OCU ch.0 match detection interrupt request of the MFT unit corresponding to the IRQxxxMON Register has been made.

See Table 2-1 for the relationship between exception no. and interrupt.

3.15. IRQ036/037/038 Batch Read Register (IRQxxxMON)

The IRQ036MON to IRQ038MON Registers can read out at once the interrupts (PPG ch.0 to ch.20) assigned to exception no. 52 to no. 54 respectively.

■ Register configuration

bit	31 … 8
Field	Reserved
Attribute	R
Initial value	0x000000

bit	7	6	5	4	3	2	1	0
Field	Reserved					PPGINT		
Attribute	R					R		
Initial value	00000					000		

■ Register function

[bit31:3] Reserved: Reserved bits

A reserved bit reads "0".

[bit2:0] PPGINT

bit	Value	Description
2	0	There is no interrupt request of the PPG channel corresponding to the IRQxxxMON Register.
	1	An interrupt request of the PPG channel corresponding to the IRQxxxMON Register has been made.
1	0	There is no interrupt request of the PPG channel corresponding to the IRQxxxMON Register.
	1	An interrupt request of the PPG channel corresponding to the IRQxxxMON Register has been made.
0	0	There is no interrupt request of the PPG channel corresponding to the IRQxxxMON Register.
	1	An interrupt request of the PPG channel corresponding to the IRQxxxMON Register has been made.

See Table 2-1 for the relationship between exception no. and interrupt.

3.16. IRQ039/040/041/042/043/044/045/046/098/099/100/101 Batch Read Register (IRQxxxMON)

The IRQ039MON to IRQ046MON Registers can read out at once the interrupts (interrupts of base timer ch.0 to ch.7) assigned to exception no. 55 to no. 62 respectively.
The IRQ098MON to IRQ101MON Registers can read out at once the interrupts (interrupts of base timer ch.8 to ch.11) assigned to exception no. 114 to no. 117 respectively.

■ Register configuration

bit	31 … 8
Field	Reserved
Attribute	R
Initial value	0x000000

bit	7	6	5	4	3	2	1	0
Field	Reserved						BTINT	
Attribute	R						R	
Initial value	000000						00	

■ Register function

[bit31:2] Reserved: Reserved bits

A reserved bit reads "0".

[bit1:0] BTINT

bit	Value	Description
1	0	There is no interrupt request of source 1 (IRQ1) of the base timer channel corresponding to the IRQxxxMON Register.
	1	An interrupt request of source 1 (IRQ1) of the base timer channel corresponding to the IRQxxxMON Register has been made.
0	0	There is no interrupt request of source 0 (IRQ0) of the base timer channel corresponding to the IRQxxxMON Register.
	1	An interrupt request of source 0 (IRQ0) of the base timer channel corresponding to the IRQxxxMON Register has been made.

See Table 2-1 for the relationship between exception no. and interrupt.

The respective details of source 0 (IRQ0) and source 1 (IRQ1) to be output from the base timer vary depending on the base timer function used. Table 3-1 shows the relationship between details of IRQ0 and IRQ1 and base timer functions.

Table 3-1 Relationship between interrupt sources and base timer functions

Function	Interrupt source 0 (Interrupt source output signal IRQ0)	Interrupt source 1 (Interrupt source output signal IRQ1)
16-bit PWM timer	Detection of underflow / detection of duty match	Detection of timer start trigger
16-bit PPG timer	Detection of underflow	Detection of timer start trigger
16/32-bit reload timer	Detection of underflow	Detection of timer start trigger
16/32-bit PWC timer	Overflow detection	Detection of completion of measurement

3.17. IRQ047 Batch Read Register (IRQ047MON)

The IRQ047 Batch Read Register (IRQ047MON) can read out at once the interrupts (dual timer interrupts) assigned to exception no. 63.

■ Register configuration

bit	31 … 8
Field	Reserved
Attribute	R
Initial value	0x000000

bit	7	6	5	4	3	2	1	0
Field	Reserved						TIMINT	
Attribute	R						R	
Initial value	000000						00	

■ Register function

[bit31:2] Reserved: Reserved bits

A reserved bit reads "0".

[bit1:0] TIMINT

bit	Value	Description
1	0	There is no dual timer TIMINT2 interrupt request.
	1	A dual timer TIMINT2 interrupt request has been made.
0	0	There is no dual timer TIMINT1 interrupt request.
	1	A dual timer TIMINT1 interrupt request has been made.

3.18. IRQ048 Batch Read Register (IRQ048MON)

The IRQ048 Batch Read Register (IRQ048MON) can read out the interrupt (watch counter interrupt) assigned to exception no. 64.

■ Register configuration

bit	31	8
Field	Reserved	
Attribute	R	
Initial value	0x000000	

bit	7	6	5	4	3	2	1	0
Field	Reserved							WCINT
Attribute	R							R
Initial value	0000000							0

■ Register function

[bit31:1] Reserved: Reserved bits

A reserved bit reads "0".

[bit0] WCINT

Value	Description
0	There is no interrupt request of the watch counter.
1	An interrupt request of the watch counter has been made.

3.19. IRQ049 Batch Read Register (IRQ049MON)

The IRQ049 Batch Read Register (IRQ049MON) can read out the interrupt (external bus output error interrupt) assigned to exception no. 65.

■ Register configuration

bit	31 … 8
Field	Reserved
Attribute	R
Initial value	0x000000

bit	7	6	5	4	3	2	1	0
Field	Reserved							BMEMCS
Attribute	R							R
Initial value	0000000							0

■ Register function

[bit31:1] Reserved: Reserved bits

A reserved bit reads "0".

[bit0] BMEMCS

Value	Description
0	There is no external bus output error interrupt request.
1	An external bus output error interrupt request has been made.

3.20. IRQ050 Batch Read Register (IRQ050MON)

The IRQ050 Batch Read Register (IRQ050MON) can read out the interrupt (RTC interrupt) assigned to exception no. 66.

■ Register configuration

bit	31	8
Field	Reserved	
Attribute	R	
Initial value	0x000000	

bit	7	6	5	4	3	2	1	0
Field	Reserved							RTCINT
Attribute	R							R
Initial value	0000000							0

■ Register function

[bit31:1] Reserved: Reserved bits

A reserved bit reads "0".

[bit0] RTCINT

Value	Description
0	There is no interrupt request of the RTC.
1	An interrupt request of the RTC has been made.

3.21. IRQ059 Batch Read Register (IRQ059MON)

The IRQ059 Batch Read Register (IRQ059MON) can read out at once the interrupts (main clock oscillation stabilization wait completion interrupt, sub clock oscillation stabilization wait completion interrupt, main PLL oscillation stabilization wait completion interrupt, and PLL of USB / Ethernet oscillation stabilization wait completion interrupt) assigned to exception no. 75.

■ Register configuration

bit	31 ... 8
Field	Reserved
Attribute	R
Initial value	0x000000

bit	7	6	5	4	3	2	1	0
Field	Reserved				UPLLINT	MPLLINT	SOSCINT	MOSCINT
Attribute	R				R	R	R	R
Initial value	0000				0	0	0	0

■ Register function

[bit31:4] Reserved: Reserved bits

A reserved bit reads "0".

[bit3] UPLLINT

Value	Description
0	There is no PLL of USB / Ethernet oscillation stabilization wait completion interrupt.
1	A PLL of USB / Ethernet oscillation stabilization wait completion interrupt has been made.

[bit2] MPLLINT

Value	Description
0	There is no main PLL oscillation stabilization wait completion interrupt.
1	A main PLL oscillation stabilization wait completion interrupt has been made.

[bit1] SOSCINT

Value	Description
0	There is no sub clock oscillation stabilization wait completion interrupt.
1	A sub clock oscillation stabilization wait completion interrupt has been made.

[bit0] MOSCINT

Value	Description
0	There is no main clock oscillation stabilization wait completion interrupt.
1	A main clock oscillation stabilization wait completion interrupt has been made.

3.22. IRQ060/062/064/066/068/070/072/074/103/105/107/109/ 120/122/124/126 Batch Read Register (IRQxxxMON)

The IRQ060MON, IRQ062MON, IRQ064MON and IRQ066MON Registers can read out at once the interrupts (MFS ch.0 to ch.3 reception interrupts) assigned to exception no. 76, no. 78, no.80 and no. 82 respectively.

The IRQ068MON, IRQ070MON, IRQ072MON and IRQ074MON Registers can read out at once the interrupts (MFS ch.4 to ch.7 reception interrupts) assigned to exception no. 84, no. 86, no.88 and no. 90 respectively.

The IRQ103MON, IRQ105MON, IRQ0107MON and IRQ109MON Registers can read out at once the interrupts (MFS ch.8 to ch.11 reception interrupts) assigned to exception no. 119, no. 121, no.123 and no. 125 respectively.

The IRQ120MON, IRQ122MON, IRQ124MON and IRQ126MON Registers can read out at once the interrupts (MFS ch.12 to ch.15 reception interrupts) assigned to exception no. 136, no. 138, no.140 and no. 142 respectively.

■ Register configuration

bit	31	8
Field	Reserved	
Attribute	R	
Initial value	0x000000	

bit	7	6	5	4	3	2	1	0
Field	Reserved							MFSRINT
Attribute	R							R
Initial value	0000000							0

■ Register function

[bit31:1] Reserved: Reserved bits

A reserved bit reads "0".

[bit0] MFSRINT

Value	Description
0	There is no reception interrupt request of the MFS channel corresponding to the IRQxxxMON Register.
1	A reception interrupt request of the MFS channel corresponding to the IRQxxxMON Register has been made.

See Table 2-1 for the relationship between exception no. and interrupt.

3.23. IRQ061/063/065/067/069/071/073/075/104/106/108/110/ 121/123/125/127 Batch Read Register (IRQxxxMON)

The IRQ061MON, IRQ063MON, IRQ065MON and IRQ067MON Registers can read out at once the interrupts (transmission interrupts and status interrupts of MFS ch.0 to ch.3) assigned to exception no. 77, no. 79, no. 81 and no. 83 respectively.

The IRQ069MON, IRQ071MON, IRQ073MON and IRQ075MON Registers can read out at once the interrupts (transmission interrupts and status interrupts of MFS ch.4 to ch.7) assigned to exception no. 85, no. 87, no. 89 and no. 91 respectively.

The IRQ104MON, IRQ106MON, IRQ108MON and IRQ110MON Registers can read out at once the interrupts (transmission interrupts and status interrupts of MFS ch.8 to ch.11) assigned to exception no. 120, no. 122, no. 124 and no. 126 respectively.

The IRQ121MON, IRQ123MON, IRQ125MON and IRQ127MON Registers can read out at once the interrupts (transmission interrupts and status interrupts of MFS ch.12 to ch.15) assigned to exception no. 137, no. 139, no. 141 and no. 143 respectively.

■ Register configuration

bit	31	8
Field	Reserved	
Attribute	R	
Initial value	0x000000	

bit	7	6	5	4	3	2	1	0
Field	Reserved						MFSINT	
Attribute	R						R	
Initial value	000000						00	

■ Register function

[bit31:2] Reserved: Reserved bits

A reserved bit reads "0".

[bit1:0] MFSINT

bit	Value	Description
1	0	There is no status interrupt request of the MFS channel corresponding to the IRQxxxMON Register.
	1	A status interrupt request of the MFS channel corresponding to the IRQxxxMON Register has been made.
0	0	There is no transmission interrupt request of the MFS channel corresponding to the IRQxxxMON Register.
	1	A transmission interrupt request of the MFS channel corresponding to the IRQxxxMON Register has been made.

See Table 2-1 for the relationship between exception no. and interrupt.

3.24. IRQ076/077/111 Batch Read Register (IRQxxxMON)

The IRQ076MON, IRQ077MON and IRQ111MON Registers can read out at once the interrupts (interrupts of A/D converter unit 0 to unit 2) assigned to exception no. 92, no. 93 and no. 127 respectively.

■ Register configuration

bit	31	8
Field	Reserved	
Attribute	R	
Initial value	0x000000	

bit	7	6	5	4	3	2	1	0
Field	Reserved			ADCINT				
Attribute	R			R				
Initial value	000			00000				

■ Register function

[bit31:5] Reserved: Reserved bits

A reserved bit reads "0".

[bit4:0] ADCINT

bit	Value	Description
4	0	There is no range comparison result interrupt request of the A/D converter unit corresponding to the IRQxxxMON Register.
	1	A range comparison result interrupt request of the A/D converter unit corresponding to the IRQxxxMON Register has been made.
3	0	There is no conversion result comparison interrupt request of the A/D converter unit corresponding to the IRQxxxMON Register.
	1	A conversion result comparison interrupt request of the A/D converter unit corresponding to the IRQxxxMON Register has been made.
2	0	There is no FIFO overrun interrupt request of the A/D converter unit corresponding to the IRQxxxMON Register.
	1	A FIFO overrun interrupt request of the A/D converter unit corresponding to the IRQxxxMON Register has been made.
1	0	There is no scan conversion interrupt request of the A/D converter unit corresponding to the IRQxxxMON Register.
	1	A scan conversion interrupt request of the A/D converter unit corresponding to the IRQxxxMON Register has been made.
0	0	There is no priority conversion interrupt request of the A/D converter unit corresponding to the IRQxxxMON Register.
	1	A priority conversion interrupt request of the A/D converter unit corresponding to the IRQxxxMON Register has been made.

See Table 2-1 for the relationship between exception no. and interrupt.

3.25. IRQ078/113 Batch Read Register (IRQxxxMON)

The IRQ078MON and IRQ113MON Registers can read out at once the interrupts (DRQ interrupts of endpoint 0 to endpoint 5 of USB ch.0, and DRQ interrupts of endpoint 0 to endpoint 5 of USB ch.1) assigned to exception no. 94 and no. 129 respectively.

■ Register configuration

bit	31	8
Field	Reserved	
Attribute	R	
Initial value	0x000000	

bit	7	6	5	4	3	2	1	0
Field	Reserved			USB_DRQ_INT				
Attribute	R			R				
Initial value	000			00000				

■ Register function

[bit31:5] Reserved: Reserved bits

A reserved bit reads "0".

[bit4:0] USB0INT

bit	Value	Description
4	0	There is no endpoint 5 DRQ interrupt request of the USB channel corresponding to the IRQxxxMON Register.
	1	An endpoint 5 DRQ interrupt request of the USB channel corresponding to the IRQxxxMON Register has been made.
3	0	There is no endpoint 4 DRQ interrupt request of the USB channel corresponding to the IRQxxxMON Register.
	1	An endpoint 4 DRQ interrupt request of the USB channel corresponding to the IRQxxxMON Register has been made.
2	0	There is no endpoint 3 DRQ interrupt request of the USB channel corresponding to the IRQxxxMON Register.
	1	An endpoint 3 DRQ interrupt request of the USB channel corresponding to the IRQxxxMON Register has been made.
1	0	There is no endpoint 2 DRQ interrupt request of the USB channel corresponding to the IRQxxxMON Register.
	1	An endpoint 2 DRQ interrupt request of the USB channel corresponding to the IRQxxxMON Register has been made.
0	0	There is no endpoint 1 DRQ interrupt request of the USB channel corresponding to the IRQxxxMON Register.
	1	An endpoint 1 DRQ interrupt request of the USB channel corresponding to the IRQxxxMON Register has been made.

See Table 2-1 for the relationship between exception no. and interrupt.

3.26. IRQ079/114 Batch Read Register (IRQxxxMON)

The IRQ079MON and IRQ114MON Registers can read out at once the interrupts (interrupts of USB ch.0 and ch.1) assigned to exception no. 95 and no. 130 respectively.

■ Register configuration

bit	31	8
Field	Reserved	
Attribute	R	
Initial value	0x000000	

bit	7	6	5	4	3	2	1	0
Field	Reserved		USB_INT					
Attribute	R		R					
Initial value	00		000000					

■ Register function

[bit31:6] Reserved: Reserved bits

A reserved bit reads "0".

[bit5:0] USB_INT

bit	Value	Description
5	0	There is no SOFIRQ / CMPIRQ interrupt request of the USB channel corresponding to the IRQxxxMON Register.
	1	An SOFIRQ / CMPIRQ interrupt request of the USB channel corresponding to the IRQxxxMON Register has been made.
4	0	There is no DIRQ / URPIRQ / RWKIRQ / CNNIRQ interrupt request of the USB channel corresponding to the IRQxxxMON Register.
	1	An DIRQ / URPIRQ / RWKIRQ / CNNIRQ interrupt request of the USB channel corresponding to the IRQxxxMON Register has been made.
3	0	There is no SPK interrupt request of the USB channel corresponding to the IRQxxxMON Register.
	1	An SPK interrupt request of the USB channel corresponding to the IRQxxxMON Register has been made.
2	0	There is no SUSP / SOF / BRST / CONF / WKUP interrupt request of the USB channel corresponding to the IRQxxxMON Register.
	1	An SUSP / SOF / BRST / CONF / WKUP interrupt request of the USB channel corresponding to the IRQxxxMON Register has been made.
1	0	There is no endpoint 0 DRQO interrupt request of the USB channel corresponding to the IRQxxxMON Register.
	1	An endpoint 0 DRQO interrupt request of the USB channel corresponding to the IRQxxxMON Register has been made.
0	0	There is no endpoint 0 DRQI interrupt request of the USB channel corresponding to the IRQxxxMON Register.
	1	An endpoint 0 DRQI interrupt request of the USB channel corresponding to the IRQxxxMON Register has been made.

See Table 2-1 for the relationship between exception no. and interrupt.

3.27. IRQ080/081 Batch Read Register (IRQxxxMON)

The IRQ080MON and IRQ081MON Registers can read out at once the interrupts (interrupts of CAN ch.0 and ch.1) assigned to exception no. 96 and no. 97 respectively.

■ Register configuration

bit	31 … 8
Field	Reserved
Attribute	R
Initial value	0x000000

bit	7	6	5	4	3	2	1	0
Field	Reserved							CANINT
Attribute	R							R
Initial value	0000000							0

■ Register function

[bit31:1] Reserved: Reserved bits

A reserved bit reads "0".

[bit0] CANINT

Value	Description
0	There is no interrupt request of the CAN channel corresponding to the IRQxxxMON Register.
1	An interrupt request of the CAN channel corresponding to the IRQxxxMON Register has been made.

See Table 2-1 for the relationship between exception no. and interrupt.

3.28. IRQ082 Batch Read Register (IRQ082MON)

The IRQ082 Batch Read Register (IRQ082MON) can read at once the interrupts (interrupts of Ethernet MAC) assigned to exception no. 98.

■ Register configuration

bit	31 … 8
Field	Reserved
Attribute	R
Initial value	0x000000

bit	7	6	5	4	3	2	1	0
Field	Reserved					MACLPI	MACPMT	MACSBD
Attribute	R					R	R	R
Initial value	00000					0	0	0

■ Register function

[bit31:3] Reserved: Reserved bits

A reserved bit reads "0".

[bit2] MACLPI

Value	Description
0	There is no LPI interrupt request of the Ethernet MAC.
1	An LPI interrupt request of the Ethernet MAC has been made.

[bit1] MACPMT

Value	Description
0	There is no PMT interrupt request of the Ethernet MAC.
1	A PMT interrupt request of the Ethernet MAC has been made.

[bit0] MACSBD

Value	Description
0	There is no SBD interrupt request of the Ethernet MAC.
1	An SBD interrupt request of the Ethernet MAC has been made.

3.29. IRQ083/084/085/086/087/088/089/090 Batch Read Register (IRQxxxMON)

The IRQ083MON to IRQMON090 Registers can read out at once the interrupts (interrupts of DMAC ch.0 to ch.7) assigned to exception no. 99 to no. 106 respectively.

■ Register configuration

bit	31 … 8
Field	Reserved
Attribute	R
Initial value	0x000000

bit	7	6	5	4	3	2	1	0
Field	Reserved							DMACINT
Attribute	R							R
Initial value	0000000							0

■ Register function

[bit31:1] Reserved: Reserved bits

A reserved bit reads "0".

[bit0] DMACINT

Value	Description
0	There is no interrupt request of the DMAC channel corresponding to the IRQxxxMON Register.
1	An interrupt request of the DMAC channel corresponding to the IRQxxxMON Register has been made.

See Table 2-1 for the relationship between exception no. and interrupt.

3.30. IRQ091 Batch Read Register (IRQ091MON)

The IRQ091 Batch Read Register (IRQ091MON) can read out at once the interrupts (DSTC interrupts) assigned to exception no. 107.

■ Register configuration

bit	31	8
Field	Reserved	
Attribute	R	
Initial value	0x000000	

bit	7	6	5	4	3	2	1	0
Field	Reserved						DSTCINT	
Attribute	R						R	R
Initial value	000000						0	0

■ Register function

[bit31:2] Reserved: Reserved bits

A reserved bit reads "0".

[bit1:0] DSTCINT

bit	Value	Description
1	0	There is no DSTC ERINT interrupt request.
	1	A DSTC ERINT interrupt request has been made.
0	0	There is no DSTC SWINT interrupt request.
	1	A DSTC SWINT interrupt request has been made.

3.31. IRQ092/093/094/095 Batch Read Register (IRQxxxMON)

The IRQ092MON to IRQ095MON Registers can read out at once the interrupts (external pin interrupt ch.16 to ch.31) assigned to exception no. 108 to no. 111 respectively.

■ Register configuration

bit	31	8
Field	Reserved	
Attribute	R	
Initial value	0x000000	

bit	7	6	5	4	3	2	1	0
Field	Reserved				EXTINT			
Attribute	R				R			
Initial value	0000				0000			

■ Register function

[bit31:4] Reserved: Reserved bits

A reserved bit reads "0".

[bit3:0] EXTINT

bit	Value	Description
3	0	There is no interrupt request of the external pin interrupt channel corresponding to the IRQxxxMON Register.
	1	An interrupt request of the external pin interrupt channel corresponding to the IRQxxxMON Register has been made.
2	0	There is no interrupt request of the external pin interrupt channel corresponding to the IRQxxxMON Register.
	1	An interrupt request of the external pin interrupt channel corresponding to the IRQxxxMON Register has been made.
1	0	There is no interrupt request of the external pin interrupt channel corresponding to the IRQxxxMON Register.
	1	An interrupt request of the external pin interrupt channel corresponding to the IRQxxxMON Register has been made.
0	0	There is no interrupt request of the external pin interrupt channel corresponding to the IRQxxxMON Register.
	1	An interrupt request of the external pin interrupt channel corresponding to the IRQxxxMON Register has been made.

See Table 2-1 for the relationship between exception no. and interrupt.

3.32. IRQ102 Batch Read Register (IRQ102MON)

The IRQ102 Batch Read Register (IRQ102MON) can read out at once the interrupts (interrupts of base timer ch.12 to ch.15) assigned to exception no. 118.

■ Register configuration

bit	31	8
Field	Reserved	
Attribute	R	
Initial value	0x000000	

bit	7	6	5	4	3	2	1	0
Field	BTINT							
Attribute	R							
Initial value	0x000000							

■ Register function

[bit31:8] Reserved: Reserved bits

A reserved bit reads "0".

[bit7:0] BTINT

bit	Value	Description
7	0	There is no interrupt request of source 1 (IRQ1) of base timer ch.15.
	1	An interrupt request of source 1 (IRQ1) of base timer ch.15 has been made.
6	0	There is no interrupt request of source 0 (IRQ0) of base timer ch.15.
	1	An interrupt request of source 0 (IRQ0) of base timer ch.15 has been made.
5	0	There is no interrupt request of source 1 (IRQ1) of base timer ch.14.
	1	An interrupt request of source 1 (IRQ1) of base timer ch.14 has been made.
4	0	There is no interrupt request of source 0 (IRQ0) of base timer ch.14.
	1	An interrupt request of source 0 (IRQ0) of base timer ch.14 has been made.
3	0	There is no interrupt request of source 1 (IRQ1) of base timer ch.13.
	1	An interrupt request of source 1 (IRQ1) of base timer ch.13 has been made.
2	0	There is no interrupt request of source 0 (IRQ0) of base timer ch.13.
	1	An interrupt request of source 0 (IRQ0) of base timer ch.13 has been made.
1	0	There is no interrupt request of source 1 (IRQ1) of base timer ch.12.
	1	An interrupt request of source 1 (IRQ1) of base timer ch.12 has been made.
0	0	There is no interrupt request of source 0 (IRQ0) of base timer ch.12.
	1	An interrupt request of source 0 (IRQ0) of base timer ch.12 has been made.

The respective details of interrupt factor 0 (IRQ0) and interrupt factor 1 (IRQ1) to be output from the base timer vary depending on the base timer function used. For details, see Table 3-1.

3.33. IRQ118 Batch Read Register (IRQ118MON)

The IRQ118 Batch Read Register (IRQ118MON) can read out at once the interrupts (SD I/F interrupts) assigned to exception no. 134.

■ Register configuration

bit	31	8
Field	Reserved	
Attribute	R	
Initial value	0x000000	

bit	7	6	5	4	3	2	1	0
Field	Reserved						SDINT	
Attribute	R							
Initial value	0x000000							

■ Register function

[bit31:2] Reserved: Reserved bits

A reserved bit reads "0".

[bit1:0] SDINT

bit	Value	Description
1	0	There is no SD card interrupt request.
	1	An SD card interrupt request has been made.
0	0	There is no interrupt request of aggregation of all SD I/F interrupt sources.
	1	An interrupt request of aggregation of all SD I/F interrupt sources has been made.

3.34. IRQ119 Batch Read Register (IRQ119MON)

The IRQ119 Batch Read Register (IRQ119MON) can read out the interrupt (Flash I/F interrupt) assigned to exception no. 135.

■ Register configuration

bit	31	8
Field	Reserved	
Attribute	R	
Initial value	0x000000	

bit	7	6	5	4	3	2	1	0
Field	Reserved							FLINT
Attribute	R							R
Initial value	0000000							0

■ Register function

[bit31:1] Reserved: Reserved bits

A reserved bit reads "0".

[bit0] FLINT

Value	Description
0	There is no interrupt request of the Flash I/F.
1	An interrupt request of the Flash I/F has been made.

3.35. IRQ112/115/116/117 Batch Read Register (IRQxxxMON)

The IRQ112MON Register, the IRQ115MON Register, the IRQ116MON Register and the IRQ117MON Register are reserved registers.

■ Register configuration

bit	31	0
Field	Reserved	
Attribute	R	
Initial value	0x00000000	

■ Register function

[bit31:0] Reserved: Reserved bits

A reserved bit reads "0".

3.36. USB ch.0 Odd Packet Size DMA Enable Register (ODDPKS)

If data is transferred in the IN direction in USB ch.0 automatic transfer in which the DMAC is used, only in the last data in the last packet, the effective bit width is compulsorily converted into 1 byte (8 bits) before the data is written to a USB endpoint.

■ Register configuration

bit	31 … 8
Field	Reserved
Attribute	R
Initial value	0x000000

bit	7	6	5	4	3	2	1	0
Field	Reserved			ODDPKS				
Attribute	R			R/W				
Initial value	000			00000				

■ Register function

[bit31:5] Reserved: Reserved bits

Write "0" to a reserved bit. A reserved bit reads "0".

[bit4] ODDPKS4

Value	Description
0	There is no conversion of the bit width for DMA transfer by the DMAC.
1	If the transfer destination address in the DMAC is USB.EP5DT, the bit width of the last transfer data is converted into one byte.

[bit3] ODDPKS3

Value	Description
0	There is no conversion of the bit width for DMA transfer by the DMAC.
1	If the transfer destination address in the DMAC is USB.EP4DT, the bit width of the last transfer data is converted into one byte.

[bit2] ODDPKS2

Value	Description
0	There is no conversion of the bit width for DMA transfer by the DMAC.
1	If the transfer destination address in the DMAC is USB.EP3DT, the bit width of the last transfer data is converted into one byte.

[bit1] ODDPKS1

Value	Description
0	There is no conversion of the bit width for the DMA transfer by the DMAC.
1	If the transfer destination address in the DMAC is USB.EP2DT, the bit width of the last transfer data is converted into one byte.

[bit0] ODDPKS0

Value	Description
0	There is no conversion of the bit width for the DMA transfer by the DMAC.
1	If the transfer destination address in the DMAC is USB.EP1DT, the bit width of the last transfer data is converted into one byte.

<Notes>

- This register is valid only when on USB ch.0. data is transferred in the IN direction in USB data size automatic transfer mode in which the DMAC is used.
- This register does not support the DMA transfer by the DSTC.
- When transferring a packet whose number of bytes is an even number, do not write "1" to any of the ODDPKS4, ODDPKS3, ODDPKS2, ODDPKS1 and ODDPKS0 bits.

4. Usage Precautions

Note the following when using the interrupt controller.

- The interrupt controller is notified of the interrupt request signals from peripheral functions in terms of level. When exiting the processing of an interrupt, always clear the interrupt request for that interrupt.
- The NMIX pin is shared with a general-purpose port. After a reset is released, the initial function of the pin is general-purpose port, and NMI input is masked. To use the NMI function, enable the NMI function using the port setting. For details, see Chapter "External Interrupt and NMI Controller".
- If the DMA transfer by the DSTC is used, the transfer end interrupt (HWINT[n]) from the DSTC is generated instead of the interrupt from a peripheral function. Due to the above configuration, the NVIC makes an interrupt from a peripheral function, and a transfer end interrupt from the DSTC jump to the same interrupt vector. Use the DREQENB[n] Register to select the interrupt to be processed.
- For the relationship between specific event detection registers and interrupt enable registers in a peripheral function, see the chapter on that peripheral function.

CHAPTER: External Interrupt and NMI Control Sections

This chapter explains the functions and operations of the external interrupt and NMI control sections.

CODE: 9BFEXTINT-E03.0_FW12-E1.04

1. Overview

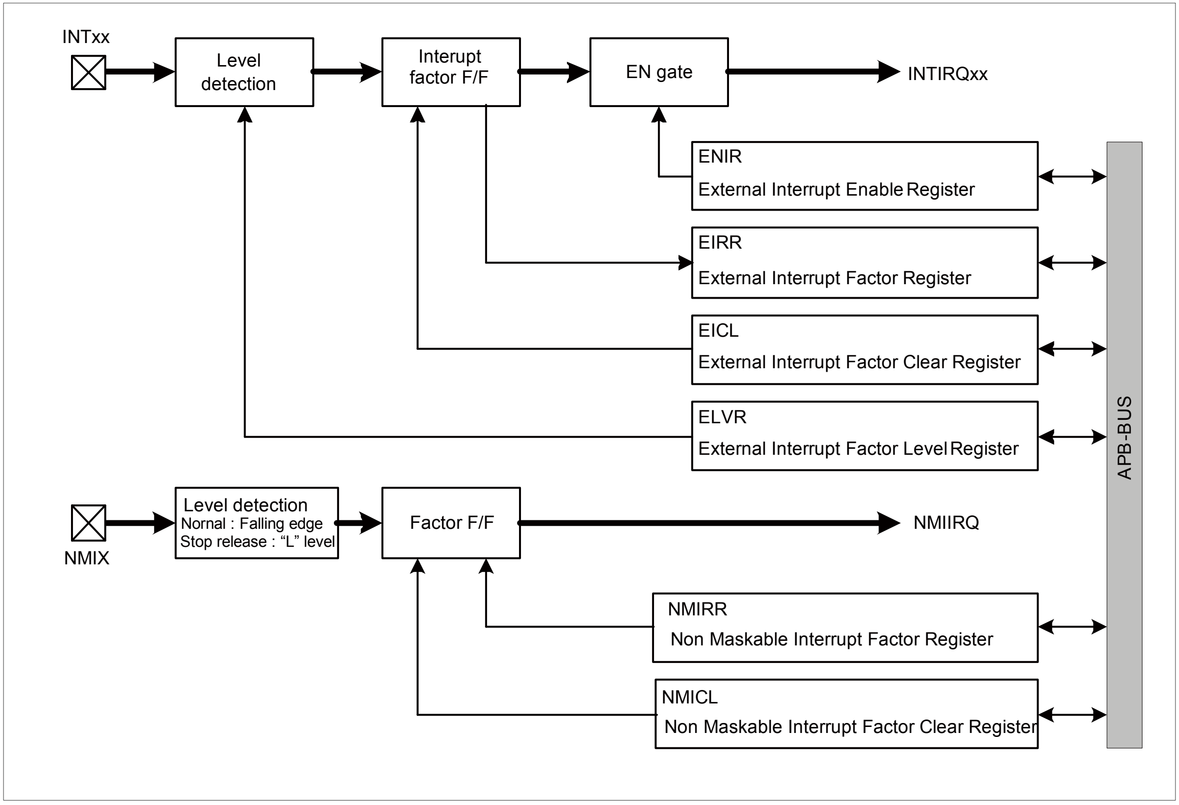
The external interrupt and NMI control sections have the following features.

- Has up to 32 external interrupt input pins and one NMI input pin mounted.
- Possible to select the "H" level, "L" level, rising edge, or falling edge to detect an external interrupt.
- Possible to use an external interrupt input or NMI input to return from standby mode.

2. Block Diagram

The following shows the block diagram of the external interrupt and NMI control sections.

Figure 2-1 Block diagram of external interrupt and NMI control sections


INTxx
Level detection
Interupt factor F/F
EN gate
INTIRQxx
ENIR
External Interrupt Enable Register
EIRR
External Interrupt Factor Register
EICL
External Interrupt Factor Clear Register
ELVR
External Interrupt Factor Level Register
APB-BUS
Level detection
Nornal : Falling edge
Stop release : "L" level
Factor F/F
NMIIRQ
NMIX
NMIRR
Non Maskable Interrupt Factor Register
NMICL
Non Maskable Interrupt Factor Clear Register

3. Operations and Setting Procedure Examples

This section explains operations and setting procedure examples.

3.1. Operations of external interrupt control section

This section shows the operations of the external interrupt control section.

■ Overview of operations in external interrupt control section

The external interrupt control section outputs an external interrupt request to the interrupt controller in the following procedure.

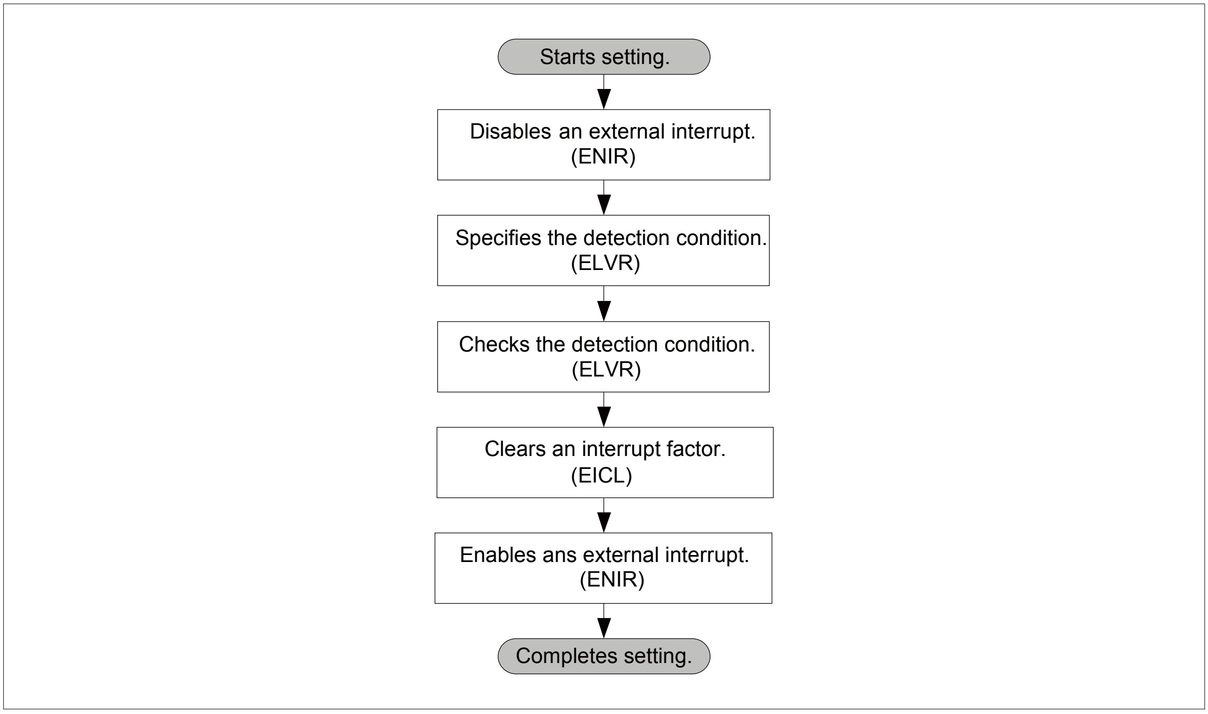
1. The signal input to pin INTxx detects the edge or level specified in the External Interrupt Level Register (ELVR). The edge or level to be detected can be selected from the following four types:
   "H" level, "L" level, rising edge, falling edge
2. The detected interrupt input is held in the interrupt factor F/F.
   It is read with the External Interrupt Factor Register (EIRR).
   The held interrupt factor is cleared with the External Interrupt Factor Clear Register (EICL).
3. If an external interrupt is enabled with the External Interrupt Enable Register (ENIR), an external interrupt request (INTIRQxx) is output to the interrupt controller.

■ Setting procedure

Execute the following steps to configure external interrupt setting.

1. Disable an external interrupt with the External Interrupt Enable Register (ENIR).
2. Specify the detection condition (effective edge or level) with the External Interrupt Factor Level Register (ELVR).
3. Read the External Interrupt Factor Level Register (ELVR).
4. Clear the external interrupt factor with the External Interrupt Factor Clear Register (EICL).
5. Enable the external interrupt with the External Interrupt Enable Register (ENIR).

Figure 3-1 External interrupt setting procedure


Starts setting.
Disables an external interrupt.
(ENIR)
Specifies the detection condition.
(ELVR)
Checks the detection condition.
(ELVR)
Clears an interrupt factor.
(EICL)
Enables ans external interrupt.
(ENIR)
Completes setting.

■ Canceling an external interrupt request

When the external interrupt detection condition is set to the "H" or "L" level, an interrupt factor is held in the External Interrupt Factor Register (EIRR) even if an external interrupt request input (INTxx) is canceled. Therefore, an external interrupt request (INTIRQxx) remains output to the interrupt controller.

Execute the following steps to cancel an external interrupt request.

1. Read the External Interrupt Factor Register (EIRR), and check the interrupt factor.
2. Write "0" to the corresponding bit in the External Interrupt Factor Clear Register (EICL) to clear it.
3. Read the External Interrupt Factor Register (EIRR), and check that the interrupt factor is cleared.

Figure 3-2 Clearing an interrupt factor

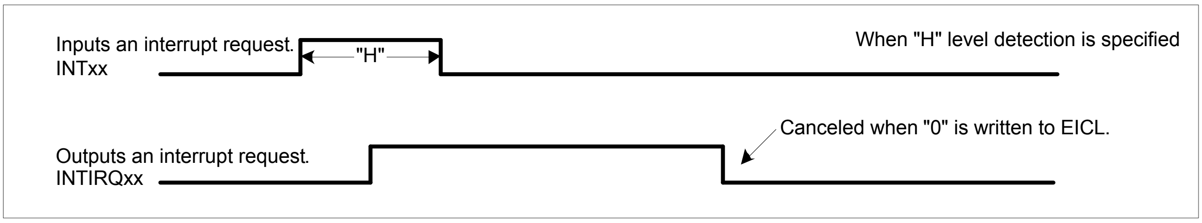
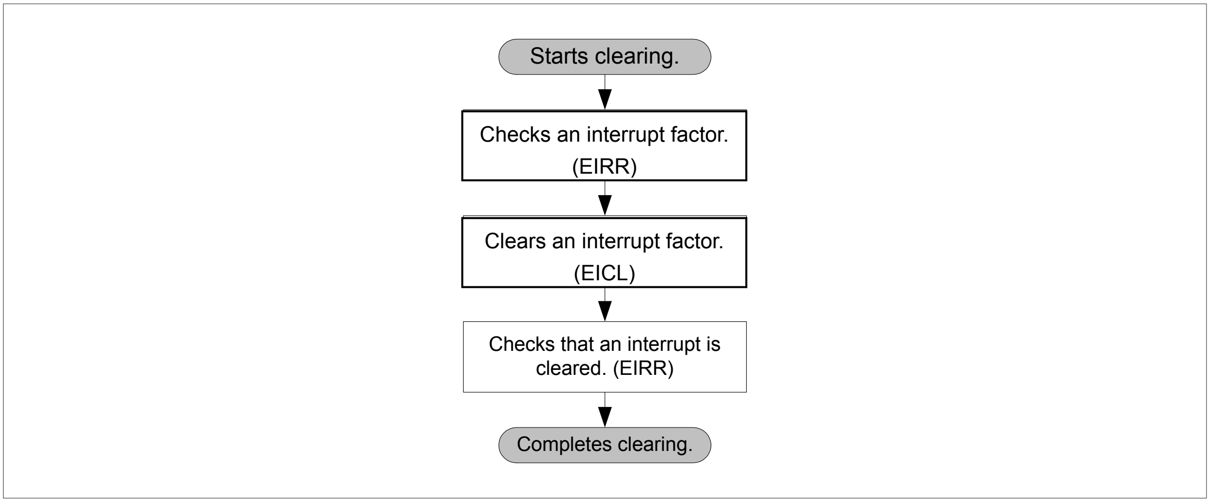

Inputs an interrupt request.
INTxx
"H"
When "H" level detection is specified
Canceled when "0" is written to EICL.
Outputs an interrupt request.
INTIRQxx

Figure 3-3 Canceling an external interrupt request


Starts clearing.
Checks an interrupt factor. (EIRR)
Clears an interrupt factor. (EICL)
Checks that an interrupt is cleared. (EIRR)
Completes clearing.

3.2. Operations of NMI control section

This section shows the operations of the NMI control section.

■ Overview of NMI control section

The NMI control section outputs an NMI interrupt request (NMIIRQ) to the CPU if the edge or level is detected from the signal input to the NMI input pin (NMIX).
The following edge or level is detected.

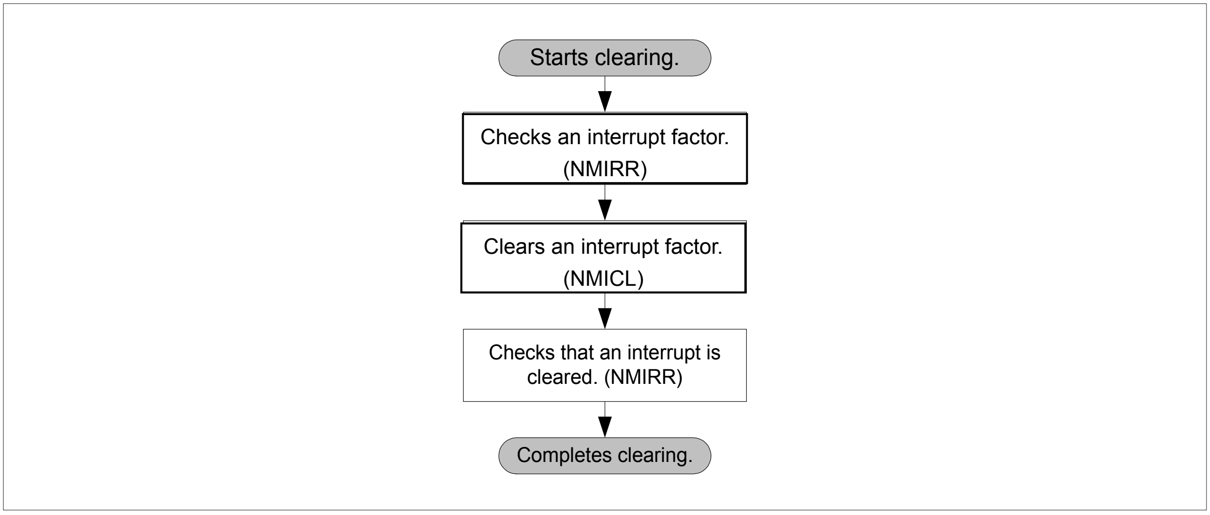
- Normal mode : Falling edge
- Stop mode : "L" level

■ Canceling an NMI request

To cancel an NMI request, clear the request register in the same way as for an external interrupt request.
Execute the following steps to cancel an NMI interrupt request.

1. Read the NMI Factor Register (NMIRR), and check the interrupt factor.
2. Write "0" to the corresponding bit in the NMI Factor Clear Register (NMICL) to clear it.
3. Read the NMI Factor Register (NMIRR), and check that the interrupt factor is cleared.

Figure 3-4 Canceling an NMI request


Starts clearing.
Checks an interrupt factor. (NMIRR)
Clears an interrupt factor. (NMICL)
Checks that an interrupt is cleared. (NMIRR)
Completes clearing.

3.3. Returning from timer or stop mode

This section shows a return from the timer or stop mode.

■ Overview

An external interrupt and NMI requests can be used to return from timer or stop mode.
In timer or stop mode, the signal first input to pin INTxx or NMIX is input asynchronously, and the device can return from timer or stop mode.

■ Setting before changing to stop mode

To use an external interrupt request, in the External Interrupt Enable Register (ENIR), specify the pin used to return from stop mode and also specify the effective detection level before changing to stop mode.

· Pin used to return from stop mode.	: Interrupt request output enable (ENIR = 1)
· Pin not used to return from stop mode.	: Interrupt request output disable (ENIR = 0)

To use an NMI request, only the "L" level is detected, and no register setting is required.

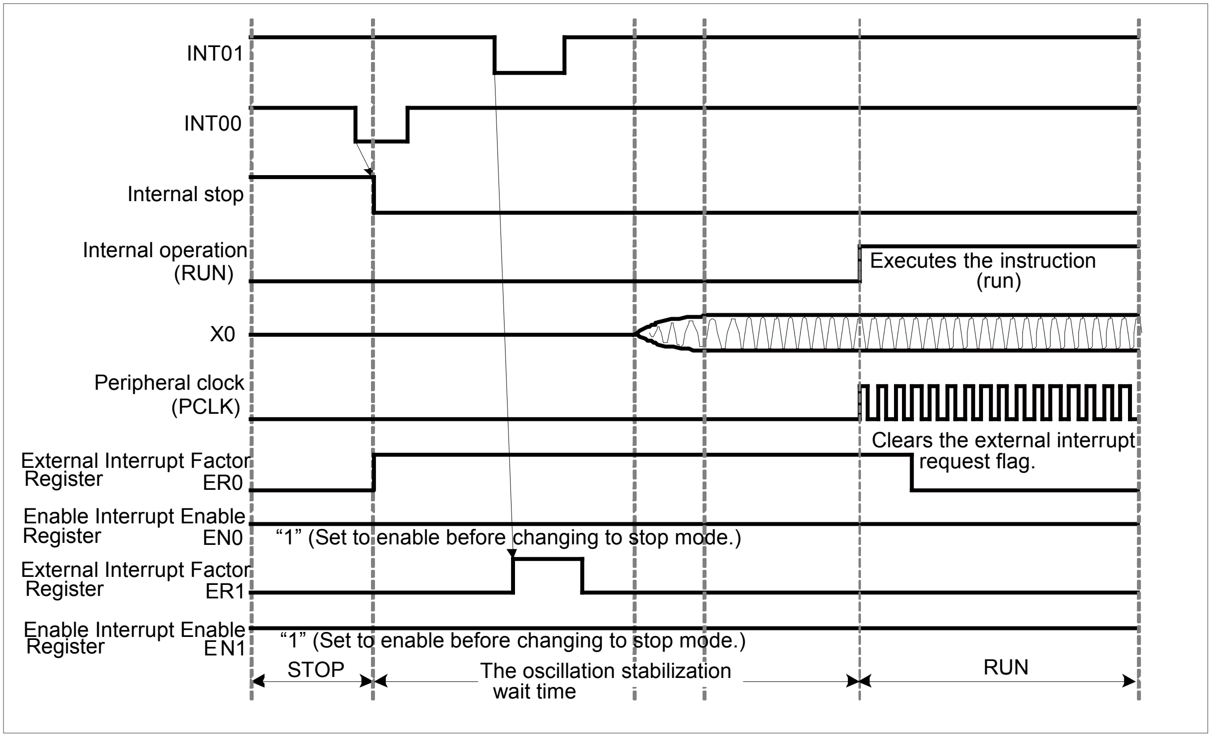
■ Returning from stop mode

For external interrupt request, if the pre-specified effective level is detected in the pin used to return from stop mode, the device returns from stop mode.
For NMI request, if the "L" level is detected in stop mode, the device returns from stop mode.

■ Notes on returning from stop mode

Any other external interrupt requests cannot be recognized until the oscillation stabilization wait time lapses after stop mode was released.
(For INT01 in Figure 3-5, any external interrupt requests cannot be recognized.)
Therefore, to input an external interrupt after stop mode was released, input an external interrupt signal after the oscillation stabilization wait time lapsed.

Figure 3-5 Returning from stop mode


INT01
INT00
Internal stop
Internal operation (RUN)
Executes the instruction (run)
X0
Peripheral clock (PCLK)
Clears the external interrupt request flag.
External Interrupt Factor Register ER0
Enable Interrupt Enable Register EN0
"1" (Set to enable before changing to stop mode.)
External Interrupt Factor Register ER1
Enable Interrupt Enable Register EN1
"1" (Set to enable before changing to stop mode.)
STOP
The oscillation stabilization wait time
RUN

4. Registers

This section provides a list of registers.

■ Register list

The following shows a list of registers in the external interrupt and NMI control sections.

Table 4-1 Registers in external interrupt and NMI control sections

Abbreviation	Register name	Reference
ENIR	External Interrupt Enable Register	4.1
EIRR	External Interrupt Factor Register	4.2
EICL	External Interrupt Factor Clear Register	4.3
ELVR	External Interrupt Factor Level Register	4.4
ELVR1	External Interrupt Factor Level Register 1	4.5
NMIRR	Non Maskable Interrupt Factor Register	4.6
NMICL	Non Maskable Interrupt Factor Clear Register	4.7

4.1. External Interrupt Enable Register (ENIR)

The ENIR register is used to control masking an external interrupt request output.

■ Register configuration

bit	31	16
Field	EN[31:16]	
Attribute	R/W	
Initial value	0x0000	

bit	15	0
Field	EN[15:0]	
Attribute	R/W	
Initial value	0x0000	

■ Register functions

[bit31:0] EN31 to EN0: External interrupt enable bits

EN31 to EN0 bits correspond to pins INT31 to INT00.
It is not possible to set the bit corresponding to a pin that is not defined in the product specifications.

bit	Description
0	Disables the output of an external interrupt request of INTx pin corresponding to the relevant bit.
1	Enables the output of an external interrupt request of INTx pin corresponding to the relevant bit.

This function enables the interrupt request output corresponding to the bit that is set to "1" in this register, and outputs a request to the interrupt controller. The pin corresponding to the bit that is set to "0" holds an interrupt factor, but outputs no request to the interrupt controller.

4.2. External Interrupt Factor Register (EIRR)

The EIRR register indicates that an external interrupt request is detected.

■ Register configuration

bit	31 … 16
Field	ER[31:16]
Attribute	R
Initial value	0xXXXX

bit	15 … 0
Field	ER[15:0]
Attribute	R
Initial value	0xXXXX

■ Register functions

[bit31:0] ER31 to ER0: External interrupt request detection bits

ER31 to ER0 bits correspond to pins INT31 to INT00.
The bit corresponding to a pin that is not defined in the product specifications is indefinite.

bit	Function
0	Detects no external interrupt request of INTx pin corresponding to the relevant bit.
1	Detects an external interrupt request of INTx pin corresponding to the relevant bit.
Writing	No effect on operation

<Notes>

- When level detection is set with ELVR and while valid level is input from INTxx pin, clearing applicable bit (write "0") with the External Interrupt Factor Clear register (EICL) will reset "1" to applicable bit in the External Interrupt Factor Register (EIRR).
- As the initial values of GPIO are set to general purpose ports, applicable bit in the External Interrupt Factor Register (EIRR) may be set to "1". After set the GPIO to external interrupt pin, clear the External Interrupt Factor Register (EIRR).

4.3. External Interrupt Factor Clear Register (EICL)

The EICL register is used to clear the held interrupt factor.

■ Register configuration

bit	31	16
Field	ECL[31:16]	
Attribute	R/W	
Initial value	0xFFFF	

bit	15	0
Field	ECL[15:0]	
Attribute	R/W	
Initial value	0xFFFF	

■ Register functions

[bit31:0] ECL31 to ECL0: External interrupt factor clear bits

ECL31 to ECL0 bits correspond to pins INT31 to INT00.

It is not possible to write "0" to the bit corresponding to a pin that is not defined in the product specifications.

bit	Function
When "0" is written	Clears an external interrupt factor of INTx pin corresponding to the relevant bit.
When "1" is written	No effect on operation
Reading	Always reads "1".

4.4. External Interrupt Factor Level Register (ELVR)

The ELVR is used to select the level or edge of the signal detected as an external interrupt request.

■ Register configuration

bit	31															16
Field	LB15	LA15	LB14	LA14	LB13	LA13	LB12	LA12	LB11	LA11	LB10	LA10	LB9	LA9	LB8	LA8
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

bit	15															0
Field	LB7	LA7	LB6	LA6	LB5	LA5	LB4	LA4	LB3	LA3	LB2	LA2	LB1	LA1	LB0	LA0
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

■ Register functions

[bit31:0] LA15 to LA0 or LB15 to LB0: External interrupt request detection level selection bits

LA15 to LA0 or LB15 to LB0 bits correspond to pins INT15 to INT00 on a 2-bit (LA and LB) basis.
It is not possible to set the bit corresponding to a pin that is not defined in the product specifications.
If the edge or level selected with this bit is detected, it is recognized as an external interrupt request.

LBx	LAx	Description
0	0	Detects the "L" level.
0	1	Detects the "H" level.
1	0	Detects the rising edge.
1	1	Detects the falling edge.

4.5. External Interrupt Factor Level Register 1 (ELVR1)

The ELVR is used to select the level or edge of the signal detected as an external interrupt request.

■ Register configuration

bit	31															16
Field	LB31	LA31	LB30	LA30	LB29	LA29	LB28	LA28	LB27	LA27	LB26	LA26	LB25	LA25	LB24	LA24
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

bit	15															0
Field	LB23	LA23	LB22	LA22	LB21	LA21	LB20	LA20	LB19	LA19	LB18	LA18	LB17	LA17	LB16	LA16
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

■ Register functions

[bit31:0] LA31 to LA16 or LB31 to LB16: External interrupt request detection level selection bits

LA31 to LA16 or LB31 to LB16 bits correspond to pins INT31 to INT16 on a 2-bit (LA and LB) basis.
It is prohibited to set the bit corresponding to a pin that is not defined in the product specifications.
If the edge or level selected with this bit is detected, it is recognized as an external interrupt request.

LBx	LAx	Description
0	0	Detects the "L" level.
0	1	Detects the "H" level.
1	0	Detects the rising edge.
1	1	Detects the falling edge.

4.6. Non Maskable Interrupt Factor Register (NMIRR)

The NMIRR Register indicates that a non maskable interrupt (NMI) request is detected.

■ Register configuration

bit	15	1	0
Field	Reserved		NR
Attribute	-		R
Initial value	-		0

■ Register functions

[bit15:1] Reserved: Reserved bits

The read value is undefined.
They have no effect in write mode.

[bit0] NR: NMI interrupt request detection bit

The NR bit corresponds to NMIX pin.

bit	Function
0	Detects no NMI interrupt request.
1	Detects an NMI interrupt request.
Writing	No effect on oparetion

4.7. Non Maskable Interrupt Factor Clear Register (NMICL)

The NMICL register is used to clear the held interrupt factor.

■ Register configuration

bit	15	1	0
Field	Reserved		NCL
Attribute	-		R/W
Initial value	-		1

■ Register functions

[bit15:1] Reserved: Reserved bits

The read value is undefined.
They have no effect in write mode.

[bit0] NCL: NMI interrupt factor clear bit

The NCL bit corresponds to NMIX pin.

bit	Function
When "0" is written	Clears an NMI interrupt factor.
When "1" is written	No effect on operation
Reading	Always reads "1".

<Notes>

- If ELVR is rewritten to change the detection condition, an invalid interrupt factor may occur.
  To avoid an invalid interrupt factor from occurring, keep the procedure shown in Figure 3-1 when changing the detection condition.
- To detect the edge or level specified in ELVR, at least 3T (T: PCLK cycle) is required as the pulse width. If a signal that does not satisfy the pulse width is input, it is not guaranteed that correct operations will be carried out.
- When level detection is specified in ELVR, the corresponding bit in the External Interrupt Factor Register (EIRR) is set to "1" again while the effective level is input from pin INTxx even if the corresponding bit is cleared (set to "0") with the External Interrupt Factor Clear Register (EICL).
- The NMI detection level setting register is not provided. In normal mode, the falling edge is detected. This register is used to return from stop mode when the "L" level is detected.
- NMI is targeted for non maskable interrupt, so an NMI Enable Interrupt Request Register is not provided.

CHAPTER: DMAC

This chapter explains DMAC.

CODE: 9BFDMAC_FM4-E01.0_MHDMAC-E01.0

1. Overview of DMAC

DMAC (Direct Memory Access Controller) is a function block that transfers data at high speed without CPU. Using DMAC improves the system performance.

■ Overview of DMAC

- DMAC has its own bus which is independent from the CPU bus; therefore, it allows for transfer operation even when the CPU bus is accessed.
- It consists of 8 channels enabled to execute 8 types of different DMA transfers independently from one another.
- It can set the address of the transfer destination, the address of the transfer source, the size of transfer data, the source of transfer request and the transfer mode, and control the start of transfer operation, the forced termination of transfer and the pause of transfer for each channel.
- It can control the batch start of transfers, the forced batch termination of transfers and the batch pause of transfers for all of the channels.
- When multiple channels are operating simultaneously, it can select the priority of such channel operations from the fixed method or the rotated method.
- It supports hardware DMA transfer using an interrupt signal from Peripherals.
- It complies with the system bus (AHB), supporting 32-bit address space (4Gbytes).

■ Overview of Functions of Each Channel

- The addresses of the transfer source and transfer destination can be incremented or fixed.
- Reload function for the addresses of the transfer source and transfer destination (i.e. function to return the values to the original settings upon completion of the transfer) is available.
- The size of data to be transferred can be selected from the following three specifications:

Transfer data width	: (Select from byte/half-word/word)
Setting the number of blocks	: (Select from 1 to 16)
Setting the number of transfers	: (Select from 1 to 65536)

  (For information about the difference between the number of blocks and the number of transfers, see "3 Functions and Operations of DMAC".)
- Whether or not to give notification of the successful completion of transfer and unsuccessful completion of transfer can be specified.
- Transfer mode can be selected from the following five types:
  Software-Block transfer
  Software-Burst transfer
  Hardware-Demand transfer
  Hardware-Block transfer
  Hardware-Burst transfer

■ Transfer Modes

Software transfer is a method used to start DMAC by direct instruction from CPU.

Hardware transfer is a method using an interrupt signal from a Peripheral as the DMAC transfer request signal to start DMAC directly when the Peripheral issues a transfer request.

Multifunction serial unit, USB unit and ADC unit directly instruct DMAC to start data transfer, when sending/receiving data or A/D conversion data needs to be transferred. External interrupt unit and Base timer unit directly instruct DMAC to start data transfer at a transfer timing. In either of the cases, data can be transferred without CPU by making such setting beforehand.

■ Abbreviations

This chapter contains the following terms: DE, DS, DH, PR, EB, PB, ST, IS, BC, TC, MS, TW, FS, FD, RC, RS, RD, EI, CI, SS, EM. All of these terms refer to each bit of DMAC control registers (DMACR, DMACSA, DMACDA, DMACA, DMACB). See "5 Registers of DMAC".

2. Configuration of DMAC

This section explains the system configuration of DMAC and the I/O signals of DMAC.

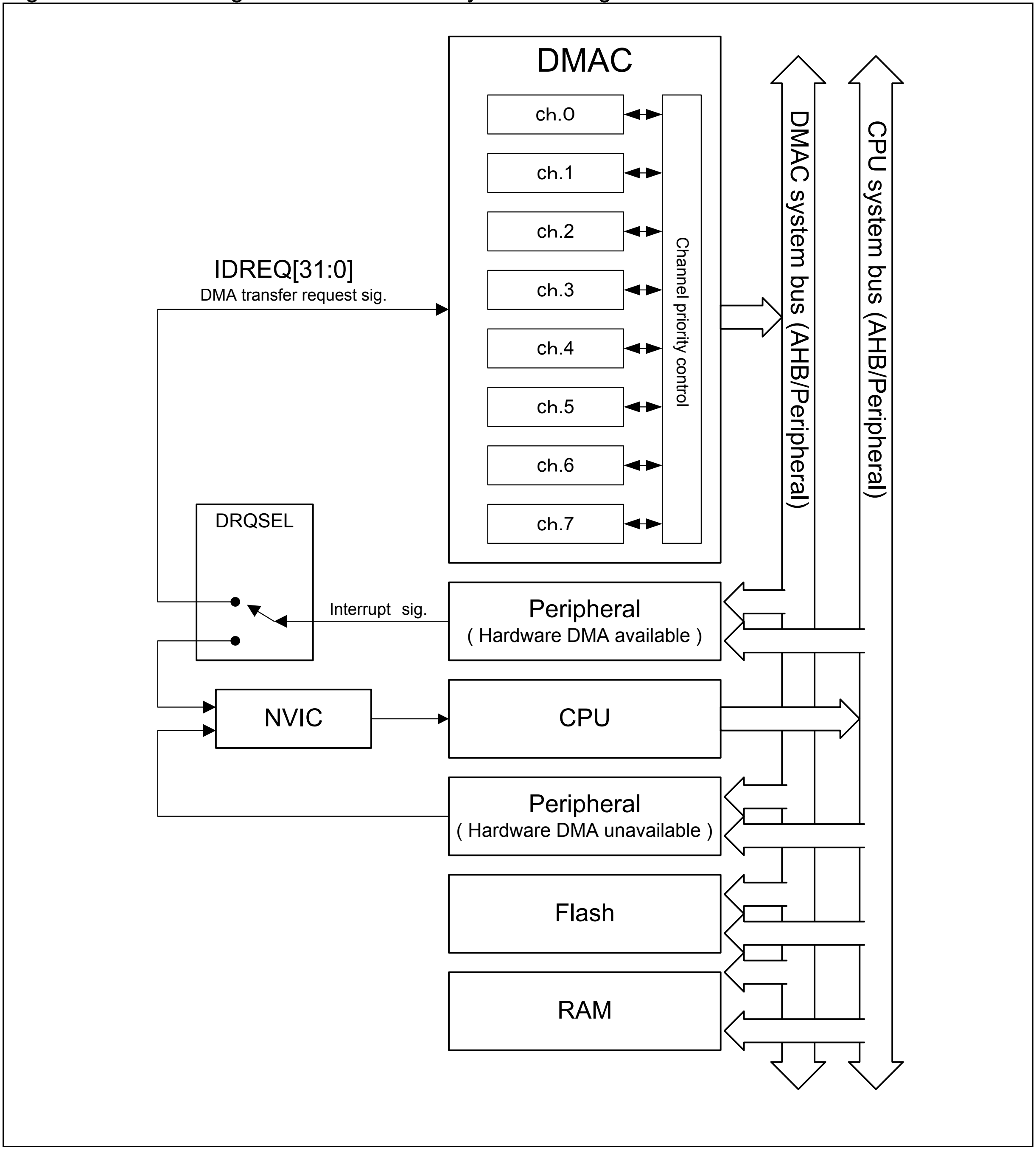
2.1. DMAC and System Configuration

This section explains DMAC and its system configuration.

■ Block Diagram

Figure 2-1 shows a diagram of DMAC and its system configuration.

Figure 2-1 Block Diagram of DMAC and System Configuration


DMAC
ch.0
ch.1
ch.2
ch.3
ch.4
ch.5
ch.6
ch.7
Channel priority control
DMAC system bus (AHB/Peripheral)
CPU system bus (AHB/Peripheral)
IDREQ[31:0]
DMA transfer request sig.
DRQSEL
Interrupt sig.
Peripheral
(Hardware DMA available)
NVIC
CPU
Peripheral
(Hardware DMA unavailable)
Flash
RAM

■ Explanation of Block Diagram

● DMAC

DMAC is in maximum 8-ch configuration. Each channel performs independent transfer. The priority controller controls the transfer operations of these channels, when there is a conflict among them.

● Connection to the system

The diagram of the system configuration in the figure has been simplified for explanation purposes. For more details, see the chapter "System Overview". DMAC is connected to CPU, Flash, RAM and Peripherals via the system bus. It has its own bus that is independent from the CPU bus, allowing for transfer operation at CPU bus access. It accesses any address area in the system by specifying the address of transfer destination and transfer source for each channel in order to transfer data between the memory and Peripheral. Since some areas cannot be accessed from DMAC, check the memory map.

● Connection of the hardware transfer request signal

The interrupt signal from the Peripheral supporting hardware transfer is selected in the interrupt controller block (indicated as DRQSEL in Figure 2-1) either to be used as the interrupt signal to CPU or the DMA transfer request signal to DMAC.

When performing DMA transfer by hardware request, connect the interrupt signal from each Peripheral as the transfer request signal to DMAC in advance by setting DRQSEL. The interrupt signal from the Peripheral that does not support hardware transfer cannot be used as the DMA transfer request signal. When the interrupt signal is used as the transfer request signal to DMAC, it cannot be used as the interrupt signal to CPU. See the chapter "Interrupts".

There are 32 DMA transfer request signals to be input to DMAC. For the correspondence between each signal and Peripheral, see Table 2-1 in the next section.

Interrupt signals from the peripheral that is not integrated cannot be selected. It should be noted that for a Peripheral with multiple channels and multiple interrupt factors, some interrupts support DMA transfer, while others don’t.

In the case of hardware transfer, each channel of DMAC selects one transfer request signal out of the above 32 transfer request signals in its operation. The IS register is used for the selection.

● Connection of the hardware transfer request clear signal

Some of the Peripherals that support hardware transfer are required to clear the transfer request signal (interrupt signal) after the completion of the transfer. Although it is not illustrated in Figure 2-1, the transfer request signal is cleared for such Peripherals via DMAC by selecting it by DRQSEL.

● Connection of the hardware transfer stop request signal

The multifunction serial unit (hereinafter abbreviated as "MFS") outputs the DMA transfer stop request signal. Although it is not illustrated in Figure 2-1, MFS’s transfer stop request signal is connected to DMAC, when MFS is selected by DRQSEL. When the transfer stop request signal is asserted, DMAC stops the transfer operation. It is configured to mask the succeeding transfer request signals.

● Interrupt signal from DMAC

Although it is not illustrated in Figure 2-1, an interrupt signal used to give notification of transfer completion is connected to NVIC. Each channel has 8 interrupt outputs.

2.2. I/O Signals of DMAC

This section explains the I/O signals of DMAC.

■ Transfer Request Signals to be Input to DMAC

Table 2-1 shows a list of the transfer request signals to be input to DMAC and the interrupt signals from the corresponding Peripherals.

Table 2-1 List of Transfer Request Signals and Interrupt Signals from Corresponding Peripherals

IDREQ No.	Interrupt Signal of Corresponding Peripheral
0	Interrupt signal from EP1 DRQ of USB ch.0
1	Interrupt signal from EP2 DRQ of USB ch.0
2	Interrupt signal from EP3 DRQ of USB ch.0
3	Interrupt signal from EP4 DRQ of USB ch.0
4	Interrupt signal from EP5 DRQ of USB ch.0
5	Scan conversion interrupt signal from A/D converter unit0
6	Scan conversion interrupt signal from A/D converter unit1
7	Scan conversion interrupt signal from A/D converter unit2
8	Interrupt signal from IRQ0 of base timer ch.0
9	Interrupt signal from IRQ0 of base timer ch.2
10	Interrupt signal from IRQ0 of base timer ch.4
11	Interrupt signal from IRQ0 of base timer ch.6
12	Receiving interrupt signal from MFS ch.0
13	Sending interrupt signal from MFS ch.0
14	Receiving interrupt signal from MFS ch.1
15	Sending interrupt signal from MFS ch.1
16	Receiving interrupt signal from MFS ch.2
17	Sending interrupt signal from MFS ch.2
18	Receiving interrupt signal from MFS ch.3
19	Sending interrupt signal from MFS ch.3
20	Receiving interrupt signal from MFS ch.4
21	Sending interrupt signal from MFS ch.4
22	Receiving interrupt signal from MFS ch.5
23	Sending interrupt signal from MFS ch.5
24	Receiving interrupt signal from MFS ch.6
25	Sending interrupt signal from MFS ch.6
26	Receiving interrupt signal from MFS ch.7
27	Sending interrupt signal from MFS ch.7
28	Interrupt signal from external interrupt unit ch.0
29	Interrupt signal from external interrupt unit ch.1
30	Interrupt signal from external interrupt unit ch.2
31	Interrupt signal from external interrupt unit ch.3

■ Interrupt Signals Output from DMAC

Table 2-2 shows a list of the interrupt signals output from DMAC.

Table 2-2 List of Interrupt Signals from DMAC

Name of Interrupt Signal	Interrupt Factor Register	Interrupt Enable Register	Interrupt Type
DIRQ0	DMACB0:SS[2:0]	DMACB0.CI	ch.0 successful transfer completion interrupt
		DMACB0.EI	ch.0 unsuccessful transfer completion interrupt
DIRQ1	DMACB1:SS[2:0]	DMACB1.CI	ch.1 successful transfer completion interrupt
		DMACB1.EI	ch.1 unsuccessful transfer completion interrupt
DIRQ2	DMACB2:SS[2:0]	DMACB2.CI	ch.2 successful transfer completion interrupt
		DMACB2.EI	ch.2 unsuccessful transfer completion interrupt
DIRQ3	DMACB3:SS[2:0]	DMACB3.CI	ch.3 successful transfer completion interrupt
		DMACB3.EI	ch.3 unsuccessful transfer completion interrupt
DIRQ4	DMACB4:SS[2:0]	DMACB4.CI	ch.4 successful transfer completion interrupt
		DMACB4.EI	ch.4 unsuccessful transfer completion interrupt
DIRQ5	DMACB5:SS[2:0]	DMACB5.CI	ch.5 successful transfer completion interrupt
		DMACB5.EI	ch.5 unsuccessful transfer completion interrupt
DIRQ6	DMACB6:SS[2:0]	DMACB6.CI	ch.6 successful transfer completion interrupt
		DMACB6.EI	ch.6 unsuccessful transfer completion interrupt
DIRQ7	DMACB7:SS[2:0]	DMACB7.CI	ch.7 successful transfer completion interrupt
		DMACB7.EI	ch.7 unsuccessful transfer completion interrupt

Reference: Interrupt Generation Factors and Clearing (For details, see "4 DMAC Control".)

Interrupt from each channel is generated by the following factors:

- Upon the successful completion of channel transfer, "101" is set to SS[2:0] of each channel. If the above value is set to SS[2:0] with CI=1 (successful transfer completion interrupt enabled), a successful transfer completion interrupt occurs.
- Upon the unsuccessful completion of channel transfer, "001", "010", "011" and "100" are set to SS[2:0] of each channel. If the above value is set to SS[2:0] with EI=1 (unsuccessful transfer completion interrupt enabled), an unsuccessful transfer completion interrupt occurs.
- The successful transfer completion interrupt and the unsuccessful transfer completion interrupt undergo logic OR; therefore, if either of the interrupts occurs, an interrupt occurs from the channel.

Interrupt from each channel can be cleared by writing "000" to SS[2:0].

3. Functions and Operations of DMAC

This section explains the operations of DMAC in each transfer mode.

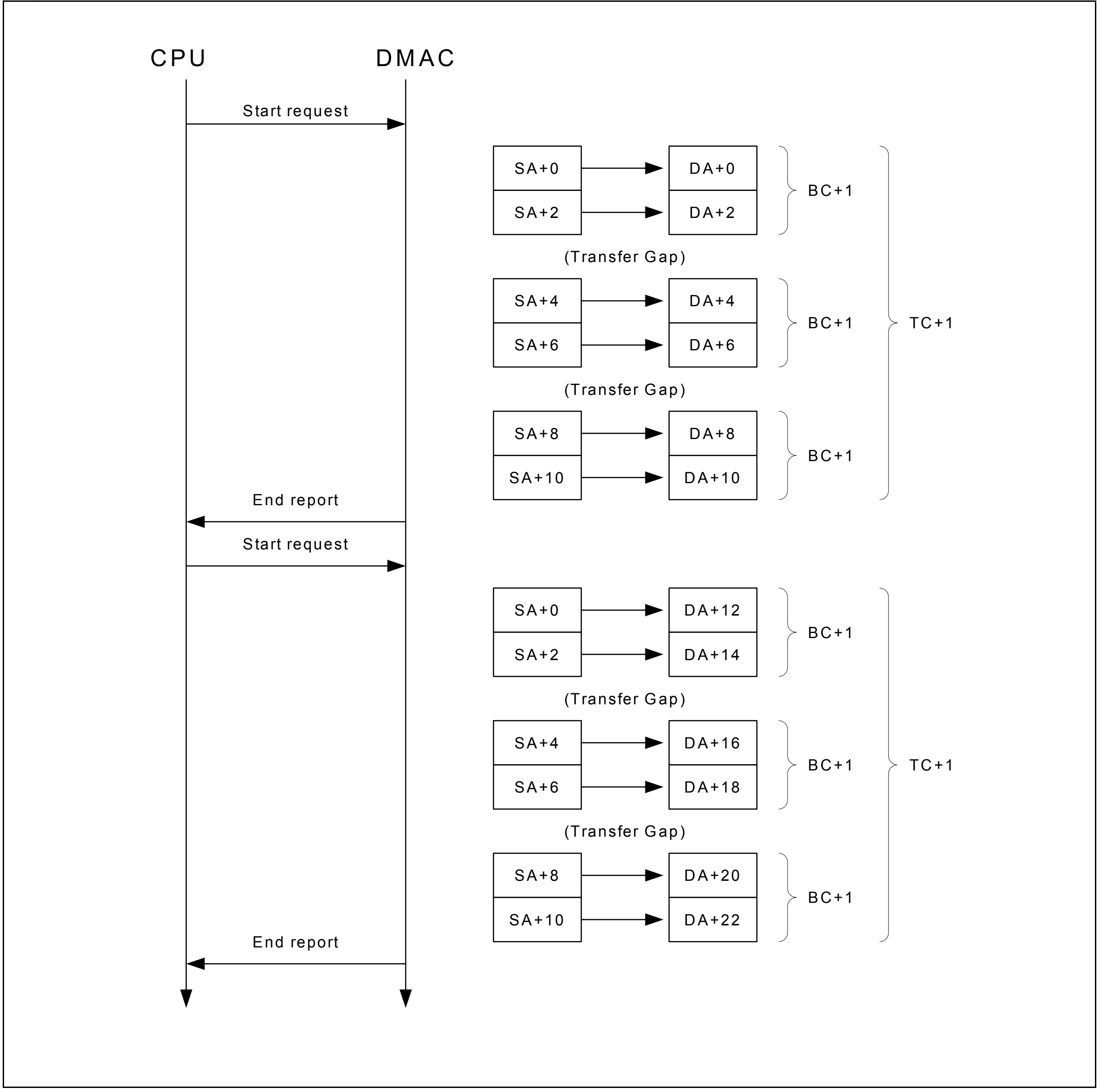
3.1. Software-Block Transfer

This section explains Software-Block transfer.

Figure 3-1 shows an example of the operation of Software-Block transfer. In this example, the following settings apply.

· Transfer mode	:	Software request Block transfer (ST=1, IS[5:0]=000000, MS=00)
· Transfer source start address	:	SA(DMACSA=SA)
· Transfer source address control	:	Increment and reload available (FS=0, RS=1)
· Transfer destination start address	:	DA(DMACDA=DA)
· Transfer destination address control	:	Increment and reload not available (FD=0, RD=0)
· Transfer data size	:	Half-word (16 bits), the number of blocks = 2, the number of transfers = 3 (TW=01, BC=1, TC=2)
· BC/TC reload	:	Reload available (RC=1)

Figure 3-1 Example of Operation of Software-Block Transfer



DMAC performs the following operation, when the transfer content is set from CPU and then the start of the transfer is instructed.

· Due to the specification of the transfer data width, each transfer is performed by half-word (16bits).
· According to the start addresses of the transfer source and transfer destination, the data width and the incremented/fixed specification, the transfer is performed in the area from the address SA to address DA, for the number of blocks (=BC+1).
· In the case of Block transfer, a Transfer Gap occurs every time transfer of one block is completed.
· DMAC performs data transfer for the number of blocks (=BC+1) by the number of transfers (=TC+1). The size of data to be transferred by each transfer request from CPU is "Data width (TW) × Number of blocks (BC+1) × Number of transfers (TC+1)".
· Once the transfer is completed, DMAC notifies CPU of the completion.
· If the start of transfer is instructed again after the completion of the transfer, the transfer is restarted from the previous transfer start address (SA+0), because the transfer source address has been set to be reloaded (RS=1). As the transfer destination address has not been specified to be reloaded (RD=0), the transfer is started from the next address (DA+12) after the previous transfer end address. Also, as the reload of BC/TC has been specified, the same values as for the previous transfer are reloaded for the number of blocks and the number of transfers for the next transfer.

Transfer Gap is a time period during which no transfer is performed, and it is inserted to prevent one of the DMAC channels from taking the possession of the system bus access right. If multiple channels have transfer requests, DMAC switches the channels that will perform the transfer operation at the timing of the Transfer Gap. The frequency of Transfer Gap generation can be controlled by adjusting the settings of BC and TC.

Moreover, the bus access right is also passed on to CPU at the Transfer Gap timing. System buses in this product are in Multi-layered configuration with a special system bus dedicated to DMA. For this reason, if there is no conflict between CPU and the destination of access, transfer can be performed at the same time as the CPU operation. Even if there is a conflict between CPU and the destination of access, the CPU operation is little affected, as long as the DMAC transfer is in a different address area group (RAM and Peripheral, or Flash memory and RAM, etc). However, if the transfer is in the same address area group (RAM and RAM, etc.), the CPU operation and/or system performance may be affected, depending on the number of blocks used; therefore, attention must be paid.

("Address area group" mentioned above refers to a group of address areas that are connected on the AHB system bus with the same bus bridge.)

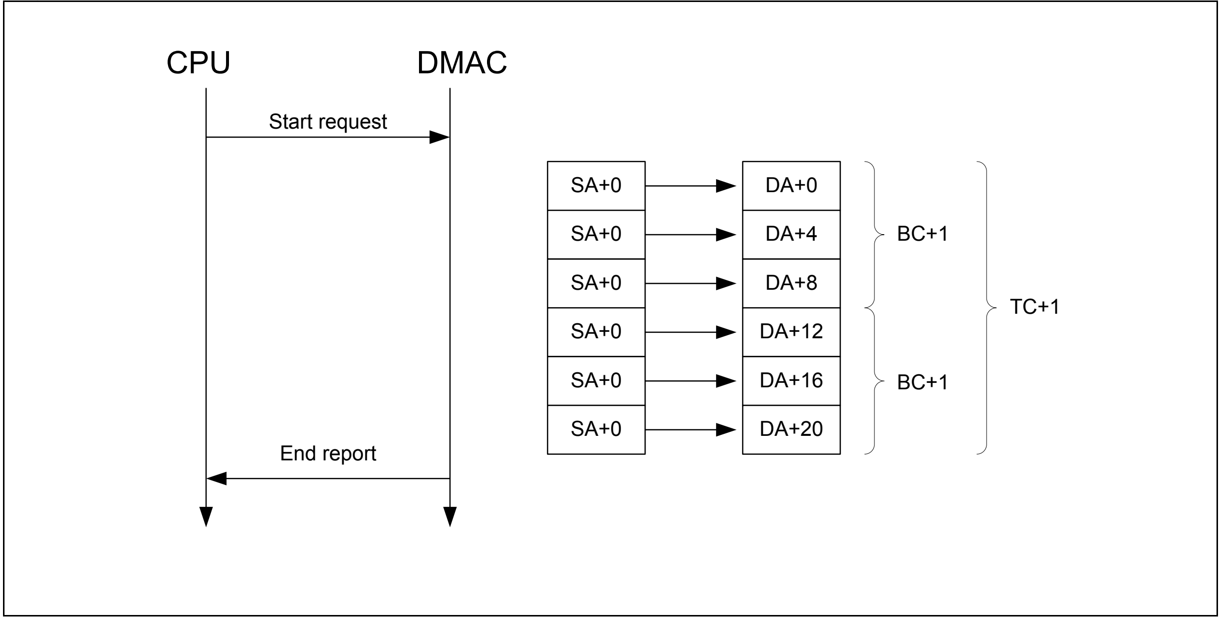
3.2. Software-Burst Transfer

This section explains Software-Burst transfer.

Figure 3-2 shows an example of the operation of Software-Burst transfer. In this example, the following settings apply.

· Transfer mode	:	Software request Burst transfer (ST=1, IS[5:0]=000000, MS=01)
· Transfer source start address	:	SA(DMACSA=SA)
· Transfer source address	:	Fixed, reload available (FS=1, RS=1)
· Transfer destination start address	:	DA(DMACDA=DA)
· Transfer destination address	:	Increment and reload not available (FD=0, RD=0)
· Transfer data size	:	Word (32 bits), the number of blocks =3, the number of transfers =2 (TW=10, BC=2, TC=1)
· Reload of the number of transfers	:	Number of transfers to be reloaded (RC=1)

Figure 3-2 Example of Operation of Software-Burst Transfer



DMAC performs the following operation, when the transfer content is set from CPU and then the start of the transfer is instructed.

- Due to the specification of the transfer data width, each transfer is performed by word (32bits).
- According to the start addresses of the transfer source and transfer destination, the data width and the incremented/fixed specification, the transfer is performed in the area from the address SA to address DA, for the number of blocks (=BC+1). As the transfer source address is specified to be fixed, it is the same as the transfer source start address (SA+0).
- In the case of Burst transfer, the transfer is executed continuously without generating Transfer Gaps.
- DMAC performs data transfer for the number of blocks (=BC+1) by the number of transfers (=TC+1). The size of data to be transferred by each transfer request from CPU is "Data width (TW) × Number of blocks (BC+1) × Number of transfers (TC+1)".
- When the transfer is completed, DMAC notifies CPU of the completion.

In the case of Burst transfer, no Transfer Gap is generated, unlike the Block transfer. As the channel to be controlled takes the possession of the system bus access right, it can be used to put the priority on that particular channel.

3.3. Hardware-Demand Transfer

This section explains Hardware-Demand transfer.

Hardware-Demand transfer is used when performing DMA transfer by the transfer request signal from the Peripherals of USB, MFS and ADC.

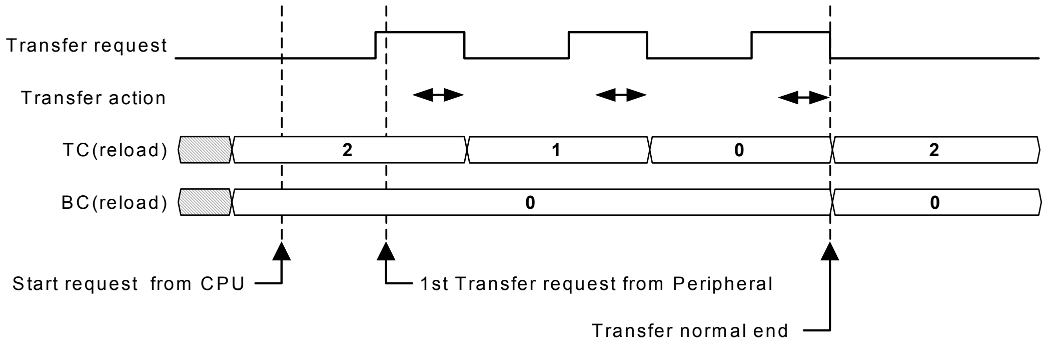
Hardware-Demand transfer is a method used to receive the transfer request signal from Peripherals on a signal level. If the transfer request signal is on High level, transfer is executed. If the transfer request signal is on Low level, no transfer is executed. Transfer is executed by setting the output of the interrupt signal from each Peripheral to High level (with interrupt request) when transfer data exists, or to Low level (without transfer request) when no transfer data exists.

In the case of Hardware-Demand transfer, always specify "1" (BC=0) as the number of blocks.

Figure 3-3 shows an example of the operation of Hardware-Demand transfer. In this example, the following settings apply. The settings of the addresses of the transfer source and transfer destination as well as the transfer data width are omitted.

·	Transfer mode	:	Hardware-Demand transfer (ST=0, IS= Peripheral at the transfer request source, MS=10)
·	Transfer data size	:	Number of blocks = 1, Number of transfers = 3 (BC=0, TC=2)

Figure 3-3 Example of Operation of Hardware-Demand Transfer



The operation of Hardware-Demand transfer is as follows:

The start of the operation is instructed by specifying the transfer content from CPU. DMAC waits for a transfer request from the Peripheral. After receiving the transfer request, it performs one transfer and then waits for the next transfer request. During the wait period, a Transfer Gap is generated. Every time a transfer request is received, it performs the same operation for the number of transfers (TC+1). The total number of transfers to be performed is (TC+1). Match the number of transfer requests from the Peripheral and the number of DMAC transfers (TC+1). Once all of the transfers are completed, DMAC notifies CPU of the completion.

3.4. Hardware-Block Transfer & Burst Transfer

This section explains Hardware-Block transfer and Burst transfer.

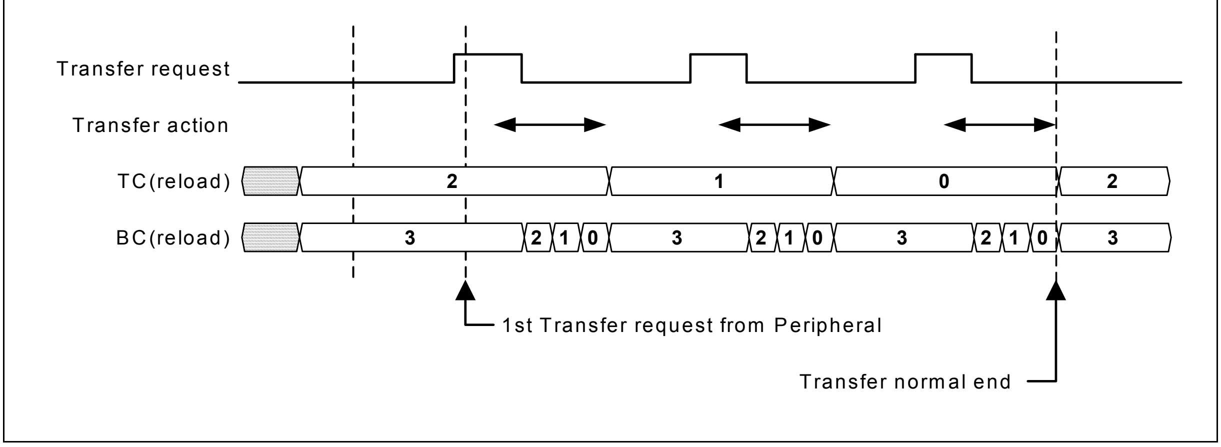
Hardware-Block transfer or Hardware-Burst transfer is used when performing DMA transfer by the transfer request signal from the Peripheral of the base timer or external interrupt.

Hardware-Block transfer and Hardware-Burst transfer are methods used to receive the transfer request signal at the rising edge of the signal. Transfer is executed, when the rising edge of the transfer request signal is detected. DMAC's transfer start timing can be specified by the output of the interrupt signal from each Peripheral.

Figure 3-4 shows an example of the operation of Hardware-Block transfer. In this example, the following settings apply. The settings of the addresses of the transfer source and transfer destination as well as the transfer data width are omitted.

- Transfer mode : Hardware-Block transfer
  (ST=0, IS= Peripheral at the transfer request source, MS=00)
- Transfer data size : Number of blocks = 4, Number of transfers = 3 (BC=3, TC=2)

Figure 3-4 Example of Operation of Hardware-Block Transfer



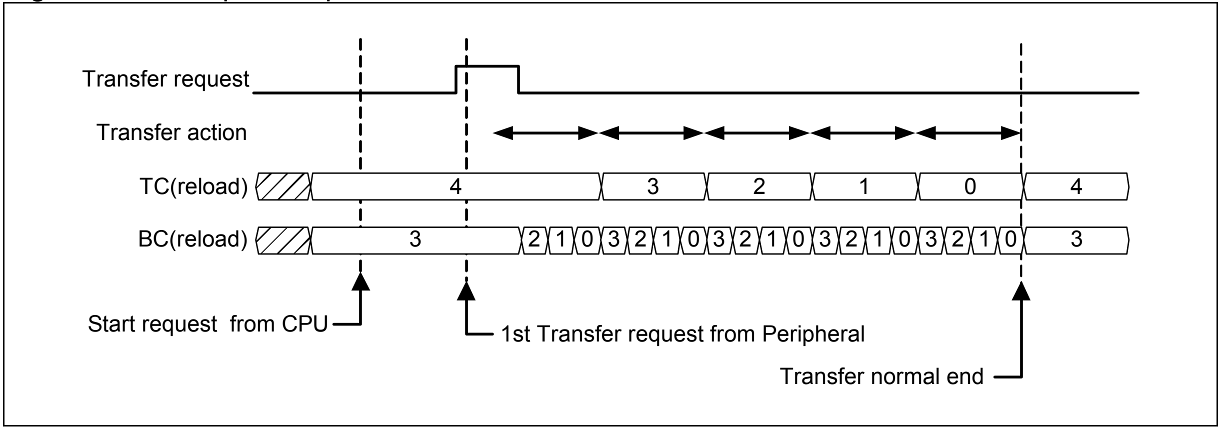
The operation of Hardware-Block transfer is as follows:

The start of the operation is instructed by specifying the transfer content from CPU. DMAC waits for a transfer request from the Peripheral. After receiving the transfer request, it performs transfers for the number of blocks (=BC+1) and then waits for the next transfer request. During the wait period, a Transfer Gap is generated. Every time a transfer request is received, it performs the same operation for the number of transfers (TC+1). The total number of transfers to be performed is (BC+1) × (TC+1). Match the number of transfer requests from the Peripheral and the number of DMAC transfers (TC+1). Once all of the transfers are completed, DMAC notifies CPU of the completion.

Figure 3-5 shows an example of the operation of Hardware-Burst transfer. In this example, the following settings apply. The settings of the addresses of the transfer source and transfer destination as well as the transfer data width are omitted.

- Transfer mode : Hardware-Burst transfer
  (ST=0, IS= Peripheral at the transfer request source, MS=01)
- Transfer data size : Number of blocks =4, Number of transfers = 5 (BC=3, TC=4)

Figure 3-5 Example of Operation of Hardware-Burst Transfer



The operation of Hardware-Burst transfer is as follows:

The start of the operation is instructed by specifying the transfer content from CPU. DMAC waits for a transfer request from the Peripheral. After receiving the first transfer request, it performs all of the transfers for the number of times calculated by (BC+1) × (TC+1). During the Hardware-Burst transfer, no Transfer Gap is generated. Once all of the transfers are completed, DMAC notifies CPU of the completion.

3.5. Channel Priority Control

This section explains the channel priority control.

■ Channel Priority Control

If multiple channels have transfer requests, DMAC switches the channel subject to the transfer among them at the timing of the Transfer Gap of each channel. At this point, the next channel to which the transfer will be performed is determined according to the priority control. The priority control can be selected from either fixed priority or rotated priority by the PR. Figure 3-6 shows an explanatory diagram. In this figure, the right axis indicates the time axis. The arrows indicate transfer timings of each channel to perform its transfer operation when all of the channels issue transfer requests simultaneously.

■ Operation in Fixed Priority Mode (PR=0)

In fixed priority mode, the channel with the smallest channel number among all the channels with a transfer request has the priority to perform transfer operation.

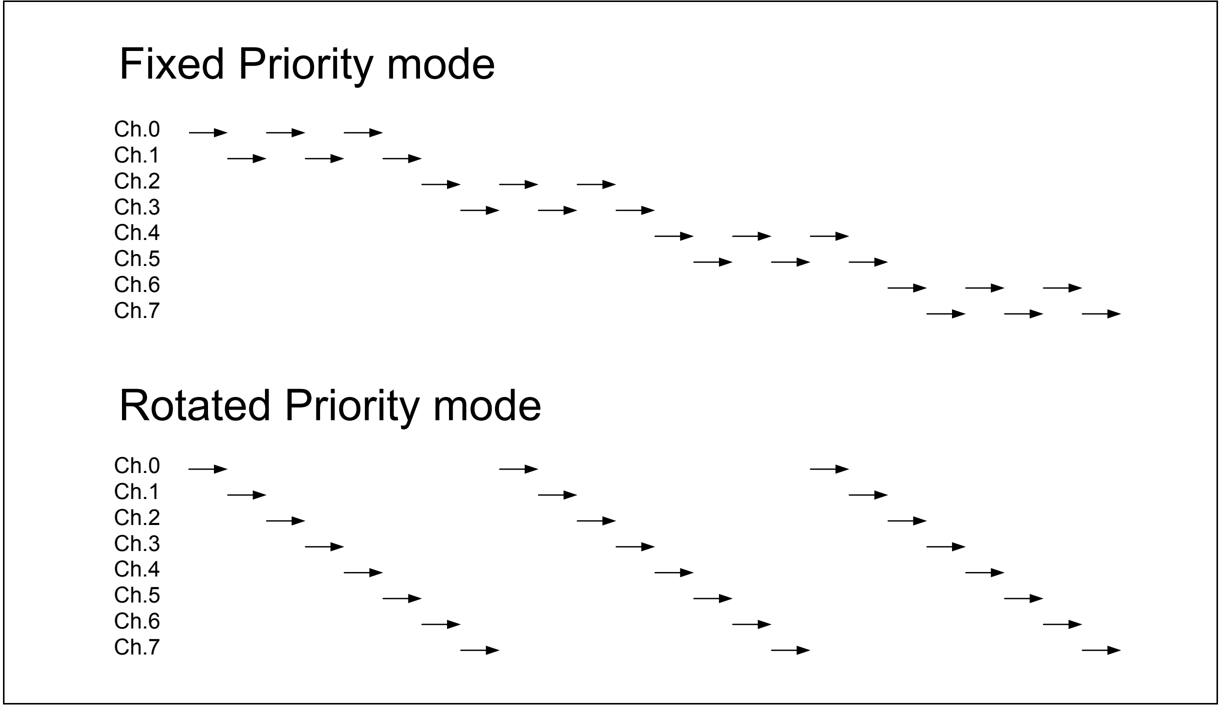
(Priority order: ch.0 > ch.1 > ch.2 > ch.3 > ch.4 > ch.5 > ch.6 > ch.7)

First, the channel with the highest priority performs its transfer (ch.0 in Figure 3-6). As the channel with the highest priority halts the transfer operation at the timing of a Transfer Gap, then, the channel with the second highest priority performs its transfer operation (ch.1 in Figure 3-6). For this reason, the channels with the highest and the second highest priority perform the transfer operations alternately. After that, when the channel with higher priority completes its transfer, the channel with lower priority starts its transfer operation (ch.3 in Figure 3-6).

■ Operation in Rotated Priority Mode (PR=1)

In rotate priority mode, all channels perform their transfer operations equally.

Figure 3-6 Explanatory Diagram of Channel Priority Control



4. DMAC Control

This section explains DMAC control methods in details.

4.1. Overview of DMAC Control

This section provides an overview of DMAC control.

The control register of each channel of DMAC has EB (individual-channel operation enable bit) and PB (individual-channel pause bit). By manipulating these bits, the start of DMA transfer operation (operation enabled), the forced termination of transfer operation (operation disabled) and the pause of transfer operation can be controlled by channel. The control register also has DE (all-channel operation enable bit) and DH (all-channel pause bit), which allow the transfer operations of all channels to be controlled at once.

Each channel is originally in the operation-prohibited state (Disable state) in which the transfer content (the address of the transfer source, the address of the transfer destination, the transfer data width, the number of transfers, the transfer mode, etc.) are specified for each channel to its configuration register. Then, the transfer operations are controlled by writing to EB, PB, DE and DH to instruct the transfer operations to be started or paused.

Once each channel completes its transfer, it sets the end code to SS (Stop Status) to give the notification of its stop state. An interrupt can be generated upon the completion of transfer. After the transfer ends, each channel clears EB and PB and returns to the operation-prohibited state (Disable state).

The following sections describe the operations of and control procedures for DMA transfer by software request and hardware DMA transfer by transfer request from Peripherals.
The following terms are used in the explanations as instructions from CPU, which refer to writing the following values to the EB, PB, DE and DH bits.

- Instruction to enable individual-channel operation (write EB=1, PB=0)
- Instruction to disable individual-channel operation (write EB=0)
- Instruction to pause individual-channel operation (write EB=1, PB=1)
- Instruction to enable all-channel operation (write DE=1, DH=0000)
- Instruction to disable all-channel operation (write DE=0)
- Instruction to pause all-channel operation (write DE=1, DH=0000)

4.2. DMAC Operation and Control Procedure for Software Transfer

This section explains DMAC operation and control procedure for software transfer.

Figure 4-1 Transitional Diagram of Software DMA Transfer State

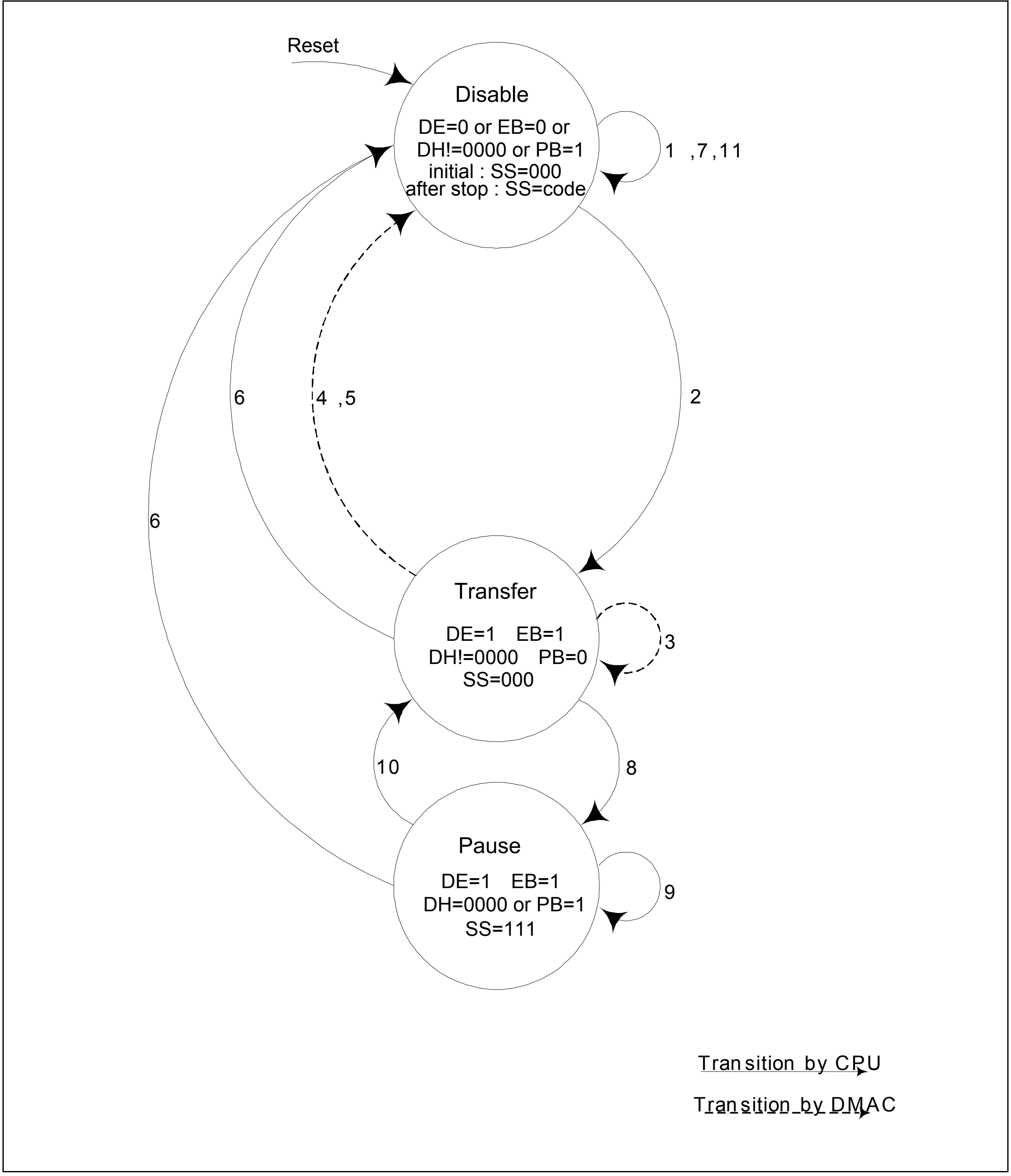


Figure 4-1 shows a transitional diagram of the states of the channel to be controlled for software transfer. The numbers next to the transitional lines in the figure correspond to the numbers which appear in the following control procedures. The solid transitional lines indicate transitions of state instructed by CPU, while the broken transitional lines indicate transitions of state due to DMAC operation.

■ Description of Each State

Disable state

In this state, the transfer of the channel to be controlled is prohibited. Channels in this state do nothing and wait for instruction from CPU. At the system reset, DE=0, EB=0, DH=0000 and PB=0 apply to this Disable state.

Transfer state

In this state, the transfer of the channel to be controlled is enabled. Channels in this state perform transfer operation as specified. Once all of the transfer operations are completed, they return to the Disable state. The state is also changed as instructed by CPU.

Pause state

In this state, the channel to be controlled has its transfer operation on pause due to an instruction to pause, issued by CPU, and is waiting for another instruction from CPU.

■ Explanation of Control Procedure

1. Disable state / Preparation for transfer

Specify via CPU the transfer content for the channel to be controlled (writing to DMACSA, DMACDA, DMACA and DMACB). For details of transfer content to be specified, see “5. Registers of DMAC”. When generating an interrupt from DMAC upon the completion of transfer, set EI and CI.

The following restrictions apply to software transfer. Specify ST=1 and IS[5:0]=000000. Demand transfer mode cannot be specified to MS. Always set "0" to EM.

Give an instruction to enable all-channel operation and set PR. Data can also be written to DMACA at the same time in Step 2.

2. Disable state => Transfer state / Start of transfer

Give an instruction to enable individual-channel operation from CPU. When DE=1, EB=1, DH=0000 and PB=0 are set, the channel to be controlled moves to Transfer state.

3. Transfer state

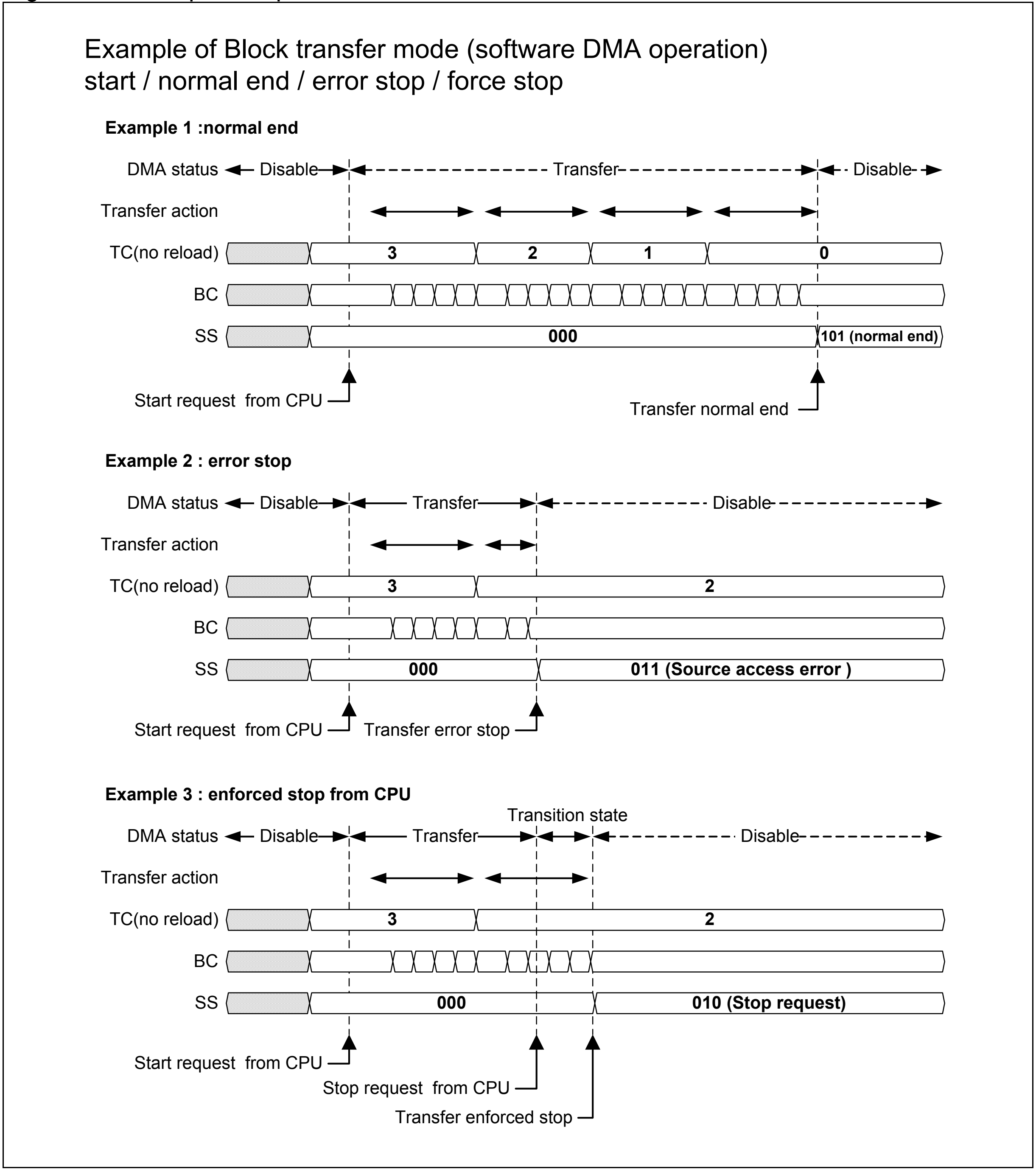
When the channel in Transfer state becomes enabled to access the system bus, it performs a transfer according to the transfer content (it may take time to start the transfer, depending on the status of other channels). In the case of Block transfer, a Transfer Gap is generated every time TC is updated. In the case of Burst transfer, no Transfer Gap is generated. During the transfer operation, BC, TC, DMACSA and DMACDA indicate the remaining number of transfers and the transfer address at that time point. The transfer status can be checked by reading from CPU.

The specified transfer content cannot be changed via CPU to the channel in Transfer state (rewriting to DMACSA, DMACDA, DMACA[29:0], DMACB[31:1]). (However, EB, PB and EM can be rewritten.)

4. Transfer state => Disable state / Successful completion of transfer

When transfers are successfully completed for the number of times calculated by (BC+1) × (TC+1), the channel in Transfer state clears EB, PB and ST and moves to Disable state. It sets SS=101 to provide the notification of the successful completion. See Example 1 in Figure 4-2. If successful transfer completion interrupt has been enabled by CI, an interrupt occurs. If reload has been specified to BC, TC, DMACSA and DMACDA, such reload is executed according to the specified transfer content.

Figure 4-2 Example of Operation of Software-Block Transfer



5. Transfer state => Disable state / Transfer error stop

The channel in Transfer state suspends the transfer process, if an address overflow, transfer source access error or transfer destination access error occurs. It clears EB, PB and ST and moves to Disable state. It sets the value that indicates the error content to SS[2:0] to give the notification of the error stop. See Example 2 in Figure 4-2. If unsuccessful transfer completion interrupt has been enabled by EI, an interrupt occurs. BC, TC, DMACSA and DMACDA to which reload has not been specified hold the values set at the time of the transfer suspension.

Normally, a transfer error occurs, when an attempt is made to access an address area that does not exist in the system bus or an address area that prohibits access from DMAC. No such error occurs in general applications.

6. Transfer state, Pause state => Disable state / Forced transfer stop

If an instruction to disable individual-channel operation or an instruction to disable all-channel operation is issued from CPU to a channel in Transfer state or Pause state, the transfer operation of that channel can be forced to stop (for the operation when an instruction to disable operation is issued to a channel in Disable state, see Step 11 in the software procedure).

If an instruction is given from CPU, the relevant channel suspends its transfer process. It clears EB, PB and ST and moves to Disable state. It sets SS[2:0]=010 and gives the notification that the transfer of that channel has been forced to stop. If unsuccessful transfer completion interrupt has been enabled by EI, an interrupt occurs. BC, TC, DMACSA and DMACDA to which reload has not been specified hold the values set at the time of the transfer suspension.

After instructed from CPU, the transfer stops at the timing when the relevant channel is not performing transfer (in Transfer Gap before the transfer starts), as shown in the Example 3 in Figure 4-2. In the case of a channel in Pause state, the transfer stops immediately. There is a time difference (Transition state) between the instruction and the stop. It may take some time, depending on the BC setting. As a new transfer cannot be set or started during this period, always make sure that the operation has stopped before setting the next transfer.

In the case of an instruction to disable all-channel operation, the timing to stop varies depending on the channel. As DS is set when all of the channels are stopped, it can confirm that all of the channels have stopped.

Even if instructed from CPU, the transfer may not be forced to stop, and instead, it may be successfully completed due to factors such as transfer mode (Burst/Block/Demand) and transfer status (the number of transfers performed, the timing of instruction to disable the operation). Also, if a transfer error occurs before the transfer stops, error stop applies to the transfer.

7. Disable state / Post-transfer process

SS is read from CPU to check the state of completion of the transfer. CPU clears SS to prepare for the next transfer. If interrupts have been enabled, the interrupt signal from DMAC is deasserted by clearing SS.

In the case of successful completion, CPU resets the transfer content, as required. If each reload has been specified, the values set before the start of the transfer are reloaded to BC, TC, DMACSA and DMACDA. If each reload has not been specified, BC and TC are initialized to "0". DMACSA and DMACDA show the address for the next transfer.

In the cases of error stop and forced stop, BC, TC, DMACSA and DMACDA must always be reset, because they may have the values set at the time of the suspension.

If the transfer is stopped due to an instruction to disable all-channel operation, DE is set to "0"; therefore, the next transfer will require an instruction to enable all-channel operation and an instruction to enable individual-channel operation.

8. Transfer state / Transfer pause

If an instruction to put individual-channel operation on pause or an instruction to put all-channel operation on pause is issued from CPU to a channel in Transfer state, the transfer operation of the relevant channel(s) can be put on pause (for the operation when an instruction to put the operation on pause is issued to a channel in Disable state, see Step 11 in the software procedure).

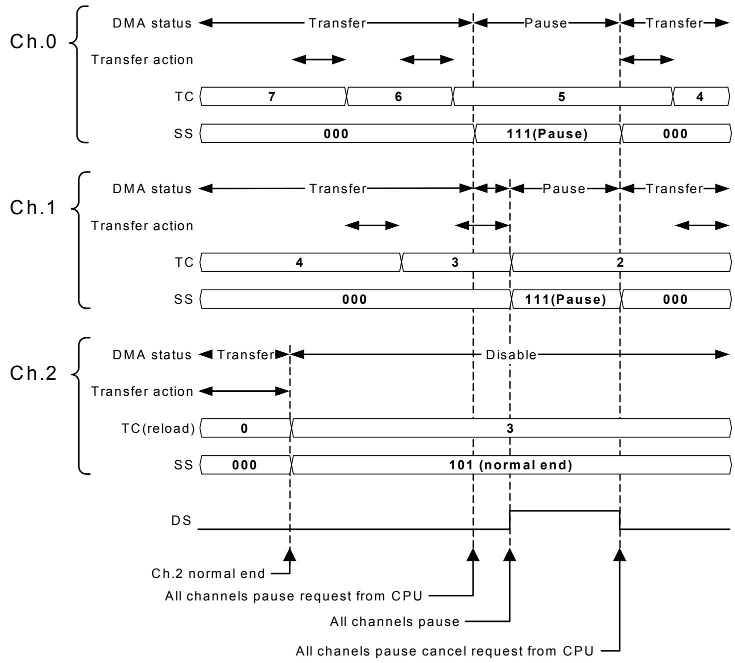
If an instruction is given from CPU, the relevant channel(s) temporarily suspends the transfer process. It sets SS=111 and gives the notification that it is in Pause state. In this case, no interrupt can be generated.

After instructed from CPU, the transfer stops at the timing when the relevant channel is not performing transfer (in Transfer Gap before the start of the transfer). There is a time difference (Transition state) between the instruction and the stop. It may take some time, depending on the BC setting. See Figure 4-3.

In the case of an instruction to put all-channel operation on pause, the timing to stop varies depending on the channel. As DS is set when all of the channels are stopped, it can confirm that all of the channels have stopped. See Figure 4-3.

Even if instructed from CPU, the transfer may not be put on pause, and instead, it may be successfully completed due to factors such as transfer mode (Burst/Block/Demand) and transfer status (the number of transfers performed, the timing of instruction to put the operation on pause). Also, if a transfer error occurs before the transfer stops, error stop applies to the transfer.

Figure 4-3 Operation when All-channel Pause is Instructed



9. **Pause state**

SS is read from CPU to confirm the pause of the transfer. The SS of a channel in Pause state is "111". While in this state, it cannot be cleared from CPU.

Even during the pause, the transfer content cannot be specified or changed (rewriting DMACSA, DMACDA, DMACA[29:0] or DMACB[31:1]). Also, when a channel in Pause state is instructed to pause, it continues to remain in the Pause state.

10. **Pause state / Cancellation of transfer pause**

If an instruction to enable individual-channel operation is issued to a channel that has been in Pause state due to an instruction to put individual-channel operation on pause, that channel returns to Transfer state. If an instruction to enable all-channel operation is issued to channels that have been in Pause state due to an instruction to put all-channel operation on pause, those channels return to Transfer state. If both of the pause instructions have been given, issue an instruction to cancel both of them.

After the instruction, SS[2:0] is cleared to "000" via DMAC.

If an instruction to enable individual-channel operation and an instruction to enable all-channel operation are issued in Pause state, they instruct the pause to be cancelled. If they are issued in Disable state, attention must be paid, as they may instruct a new transfer to be started. See Step 11 in the software procedure.

Figure 4-3 shows an example of the case where an instruction to put all-channel operation on pause. The explanation of the figure is as follows.

At the beginning, three channels, namely ch.0, ch.1 and ch.2, perform their transfer operations in Block transfer mode. ch.2 successfully completes its transfer, moves to Disable state and sets SS[2:0]=101. Then, ch.0 and ch.1 perform transfers alternately.

If an instruction to put all-channel operation on pause is issued from CPU at this point, the following operation applies. As ch.0 is subject to the Transfer Gap timing, it immediately moves to Pause state and sets SS[2:0]=111. As ch.1 is in the middle of transfer operation, it performs the transfer until the timing of the next Transfer Gap, and then moves to Pause state and sets SS[2:0]=111. As ch.2 is in Disable state, it remains in the Disable state without changing SS. DS is set, when all of the channels stop their operations.

Next, if an instruction to enable all-channel operation (instruction to cancel the pause) is issued from CPU, the following operation applies. ch.0 and ch.1 return to Transfer state and clear SS[2:0] to "000". As ch.2 is in Disable state (DE=1, EB=0), it remains in that state without starting the operation. Because the pause of all of the channels has been cancelled now, DS is reset.

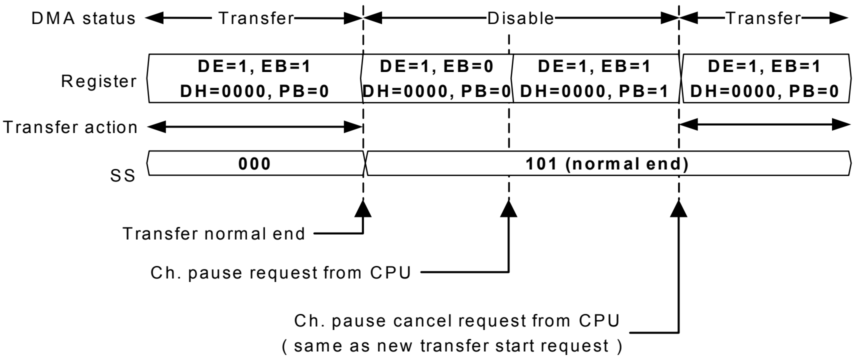
11. Operation in Disable state

A channel in Disable state remains in the Disable state, unless the conditions such as DE=1, DH=0000, EB=1, and PB=0 are established. Although in 1-2 of the software procedure, DE is set from the conditions of DE=0 and EB=0, and then, EB is set, there is no problem to set EB before DE. DE can be set last after all of the transfer settings of multiple channels subject to transfer are completed. In this case, an instruction can be issued to allow the multiple channels subject to transfer to start their transfer operations simultaneously. If such instruction for simultaneous start of transfers is issued, DMAC selects the channels to which transfers are to be started, according to the PR setting (PR can be set or changed, only when all-channel operation is disabled).

If an instruction to disable individual-channel operation, an instruction to put individual-channel operation on pause, an instruction to disable all-channel operation or an instruction to put all-channel operation on pause is issued to a channel in Disable state, only the settings of DE, DH, EB and PB are changed, but the conditions of DE=1, DH=0000, EB=1 and PB=0 are not established. Therefore, the relevant channels do nothing and do not change SS[2:0]. If an instruction to put all-channel operation on pause is issued from CPU to a channel in Disable state, as shown in the example of ch.2 operation in Figure 4-3, that channel does not change its state with SS[2:0] indicating the completion of the previous transfer.

If an instruction to put individual- or all-channel operation on pause is issued to a channel in Disable state, it may be put in Disable state with DE=1, EB=1, (DH!=0000 or PB=1). Although the bit values in this state are the same as DE, EB, DH and PB, they can be distinguished because SS[2:0] has a different value. Figure 4-4 shows such an example.

Figure 4-4 Example of Operation when Instruction to Put Individual-channel Operation on Pause is Issued in Disabled State



A certain channel is performing transfer operation. CPU issues an instruction to put individual-channel operation on pause to that channel. The instruction is issued after the transfer is completed and it moves to Disable state (DE=1, DH=0000, EB=0, PB=0). This phenomenon can occur, because the channel currently performing transfer operation changes its state outside CPU’s intention. In this case, the bit values of the relevant channel change to (DE=1, DH=0000, EB=1, PB=1) due to instruction from CPU, but SS[2:0] remains "101", the value set upon the completion. If the operation is stopped by a pause instruction, SS[2:0] will be "111"; therefore, it will be possible to distinguish between the pause state and the state in which the transfer has been completed. It should be noted that if an instruction to cancel the pause is issued without checking the state of the channel by SS[2:0], a new transfer will accidentally start, as shown in Figure 4-4.

Additional Matter 1

As ST is cleared upon the completion of a transfer, the read value of ST is "0" after the completion of the transfer. In the case of software transfer, it should be noted that "1" must always be written to ST, regardless of its read value.

Additional Matter 2

An instruction to enable individual-channel operation cannot be issued during the period after the previous instruction to enable individual-channel operation instructs the start of transfer and before the completion of the transfer is confirmed. This is because the channel to be controlled may change its state outside CPU's intention and an instruction to start a new transfer may be issued when DMAC has moved to Disable state (EB=0). Even if the SS[2:0] value confirms that the channel to be controlled is in Transfer state, the channel to be controlled may move to Disable state during the period between that point and the write operation.

Additional Matter 3

The DE and DH values can only be rewritten from CPU and these registers are never cleared from DMAC. Therefore, there is no problem to write DE=1 and DH=0000 during the transfer operation.

DH is not cleared, if an instruction to disable individual-channel operation is issued to a channel in all-channel Pause state (DE=1, DH!=0000, EB=1, PB=0). After the instruction, the relevant channel moves to Disable state (DE=1, DH!=0000, EB=0, PB=0). To start a new transfer of the relevant channel, write DE=1 and DH=0000. This indicates that the cancellation of the pause of all-channel operation is required in order to start a new transfer of the individual channel.

Additional Matter 4

The SS[2:0] value is set from DMAC upon the completion of a transfer and it is never rewritten from DMAC as long as it is in Disable state. Even if the SS[2:0] value is not cleared, the next transfer can be started. However, if it moves to Transfer state, the SS[2:0] value may be cleared from DMAC (or may not be cleared). When an interrupt from DMAC is used, it should be noted that the interrupt signal is deasserted at a timing which is not intended by CPU, if it moves to Transfer state without clearing SS[2:0].

4.3. DMAC Operation and Control Procedure for Hardware (EM=0) Transfer

This section explains DMAC operation and control procedure for hardware (EM=0) transfer.

Figure 4-5 Transitional Diagram of Hardware (EM=0) Transfer State

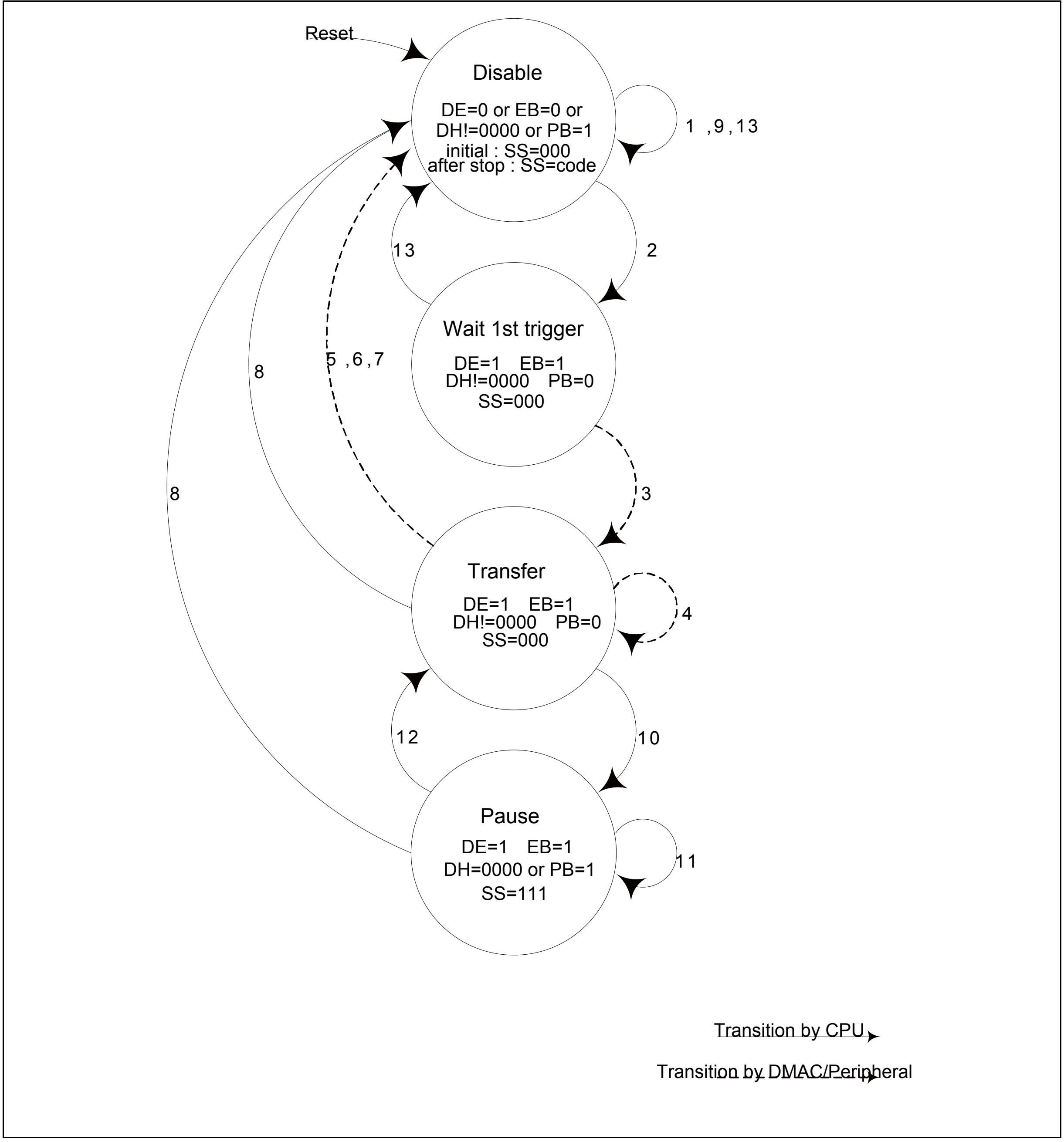


Figure 4-5 shows a transitional diagram of the states of the channel to be controlled for hardware (EM=0) transfer. The numbers next to the transitional lines in the figure correspond to the numbers which appear in the following control procedures. The solid transitional lines indicate transitions of state instructed by CPU, while the broken transitional lines indicate transitions of state due to DMAC/Peripheral operation.

Some parts of the explanation below state "See the software transfer procedure". This means that where the same control as in the software transfer procedure applies, no special mentioning is required; therefore, such redundant explanation has been omitted. In this example, the explanation assumes that EM=0 is set.

■ Description of Each State

Disable state

See Explanation of Control Procedure in "4.2 DMAC Operation and Control Procedure for Software Transfer".

Wait-1st-trigger state

In this state, the channel to be controlled is enabled to perform transfer. A channel in this state waits for the first transfer request from a Peripheral to be asserted. It also changes its state upon instruction from CPU.

Transfer state

In this state, the channel to be controlled has received the first transfer request from the Peripheral. A channel in this state performs transfer operation as specified. Once all the transfer operation is completed, it returns to Disable state. It also changes its state upon instruction from CPU.

Pause state

See Explanation of Control Procedure in "4.2 DMAC Operation and Control Procedure for Software Transfer".

■ Explanation of Control Procedure

1. Disable state / Preparation for transfer

See Step 1 in the software transfer procedure.

The following restrictions apply to hardware transfer.
Decide in advance on which Peripheral’s interrupt signal to be used as the transfer request signal to DMAC using the interrupt controller block (See "4.1 DMA Request Selection Register (DRQSEL)" in Chapter "Interrupt".). Set ST=0 and specify which Peripheral’s transfer request to be processed at the channel that will perform the transfer, by IS at the same time. Multiple channels cannot process transfer request of the same Peripheral. In the case of Demand transfer mode, set BC=0. This section explains the operation when EM=0 is set.

2. Disable state => Wait-1st-trigger state / Transfer enabled

An instruction to enable individual-channel operation is issued from CPU. When DE=1, EB=1, DH=0000 and PB=0 are set, the channel to be controlled moves to Wait-1st-trigger state.

3. Wait-1st-trigger state / Start of transfer

The channel in Wait-1st-trigger state is waiting for the transfer request signal to be asserted from the Peripheral or for an instruction from CPU. When the first transfer request signal is asserted, it moves to Transfer state.

4. Transfer state

See Step 3 in the software transfer procedure.

In the case of hardware transfer, a channel in Transfer state performs transfer operation by the transfer request signal from a Peripheral, as described in Sections “3.3 Hardware-Demand Transfer” and “3.4 Hardware-Block Transfer & Burst Transfer. In each mode, match the number of transfer requests from the Peripheral with the number of transfer requests required by DMAC. Below is the explanation for the operation when the number of transfer requests goes over or below the requirement in each operation mode.

Figure 4-6 shows a case of Demand transfer. In the case of Demand transfer, the number of transfer requests required to complete the transfer is TC+1. Unless the number of transfer requests goes over or below the requirement, CPU does not have to intervene (Example 1 in Figure 4-6).

If the number of transfer requests generated from the Peripheral exceeds the DMAC’s setting for the number of transfers, DMAC moves to Disable state after the completion of the specified number of transfers. In the Disable state, no further transfers are executed. Also, as the excessive transfer request signals are not cleared from DMAC, the asserted state continues (Example 2 in Figure 4-6).

If the number of transfer requests generated from the Peripheral is smaller than DMAC’s setting for the number of transfers, DMAC waits for the remaining number of transfer requests in Transfer state (Example 3 in Figure 4-6).

It is supposed that DMAC's transfer processing may be too slow to catch up with the generation interval of transfer requests from Peripheral. In the case of Demand transfer, the transfer request signal remains asserted; therefore, as many as TC+1 of transfers can be performed (Example 4 in Figure 4-6).

Figure 4-6 Operation of Hardware-Demand Transfer

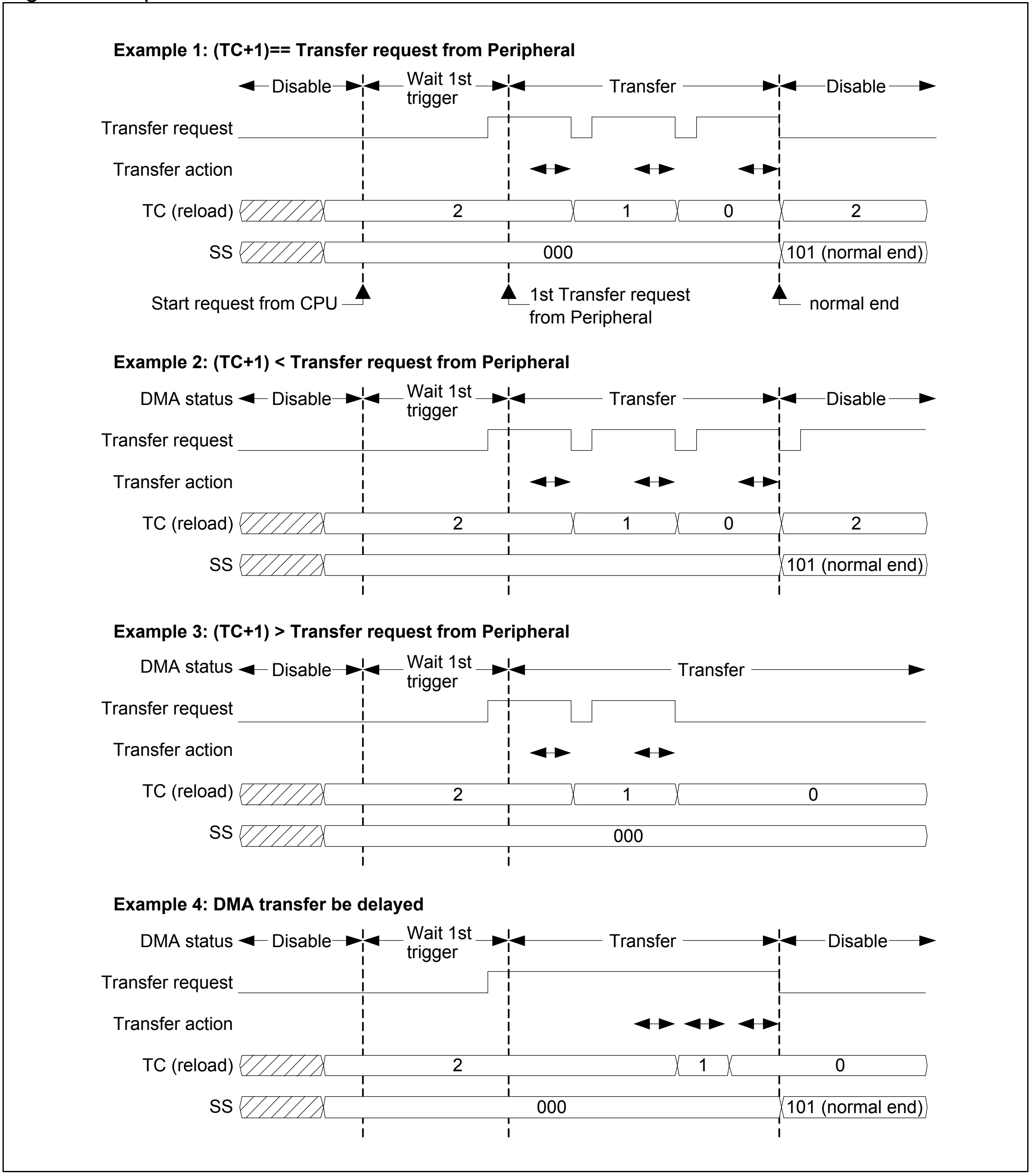
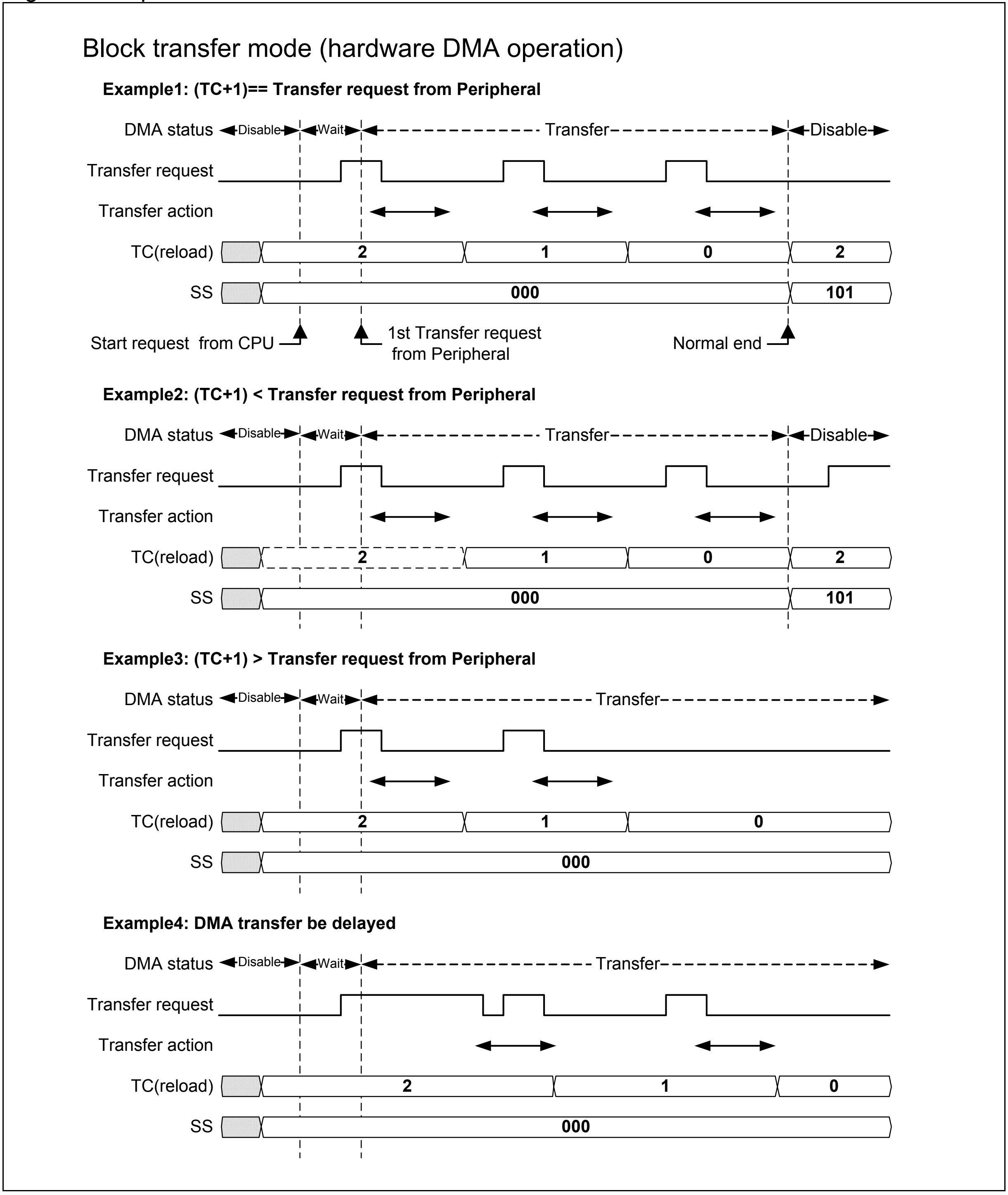


Figure 4-7 shows a case of Block transfer. In the case of Block transfer, the number of transfer requests required to complete the transfer is TC+1. Unless the number of transfer requests goes over or below the requirement, CPU does not have to intervene (Example 1 in Figure 4-7).

Figure 4-7 Operation of Hardware-Block Transfer



If the number of transfer requests generated from the Peripheral exceeds the DMAC's setting for the number of transfers, DMAC moves to Disable state after the completion of the specified number of transfers. In the Disable state, no further transfers are executed. Also, as the excessive transfer request signals are not cleared from DMAC, the asserted state continues, In this case, deassert the transfer request signal from CPU (Example 2 in Figure 4-7).

If the number of transfer requests generated from the Peripheral is smaller than DMAC's setting for the number of transfers, DMAC waits for the remaining number of transfer requests in Transfer state (Example 3 in Figure 4-7).

It is supposed that DMAC's transfer processing may be too slow to catch up with the generation interval of transfer requests from Peripheral. In the case of Block transfer, if DMAC's transfer processing is delayed from the transfer request from the Peripheral, the rising edge of the next transfer request signal during the transfer operation is ignored. Also, the transfer request signal asserted during the transfer operation is cleared from DMAC. Then, DMAC waits for the remaining transfer requests in Transfer state (Example 4 in Figure 4-7).

In the case of Burst transfer, all of the (BC+1) × (TC+1) of transfers are performed when it becomes accessible to the system bus after the first transfer request is received. The required number of transfer requests from the Peripheral is only the first one. If the number of transfer request signals generated exceeds the requirement, it is ignored in Disable state, just like Block transfer.

5. **Transfer state => Disable state / Successful completion of transfer**
   See Step 4 in the software transfer procedure.

6. **Transfer state => Disable state / Transfer error stop**
   See Step 5 in the software transfer procedure.

7. **Transfer state => Disable state / End of Peripheral stop request**
   The channel in Transfer state suspends its transfer processing, if the transfer stop request signal is asserted from the Peripheral. It clears EB, PB and ST and moves to Disable state. It sets "010" to SS[2:0] and gives the notification of the error stop. If interrupts have been enabled by EI, an unsuccessful transfer completion interrupt occurs. BC, TC, DMACSA and DMACDA to which reload has not been specified hold the values set during the suspension of the transfer. Attention must be paid to the SS[2:0] value, which is the same as the stop request from software.

8. **Transfer state, Pause state => Disable state / Forced termination of transfer**
   See Step 6 in the software transfer procedure.

9. **Disable state / Post-transfer processing**
   See Step 7 in the software transfer procedure.

   Normally, in the cases of stop request from Peripherals, forced termination from software and transfer error stop, the transfer request signal remains asserted, because the number of transfers processed is smaller than the number of transfer requests from the Peripheral. Instruct from CPU the Peripheral to deassert the transfer request signal. In the case of stop request from Peripherals, the transfer request signal is masked as long as the stop request signal is asserted. Also deassert the transfer stop request signal.

   Even if DMAC has successfully completed the specified number of transfers, the transfer request signal may remain asserted or may be reasserted, depending on Peripheral's settings. Attention must be paid to the possibility that this may affect the next transfer.

10. **Transfer state, Pause state / Transfer pause**
    See Step 8 in the software transfer procedure.

11. **Pause state**
    See Step 9 in the software transfer procedure.

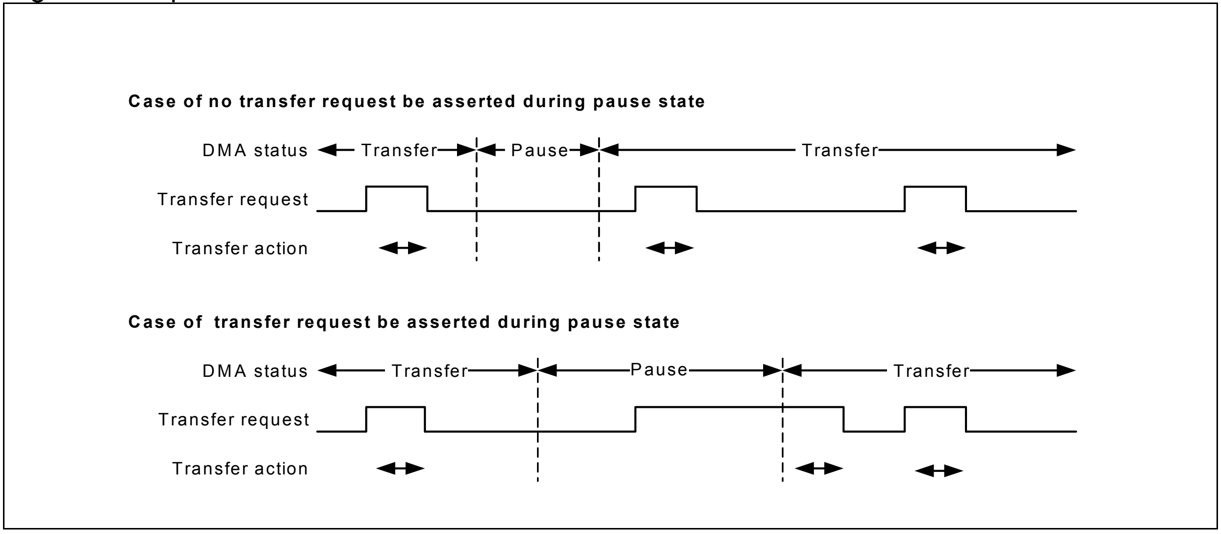
    The channel in Pause state does not execute transfer, even if the transfer request signal from the Peripheral is asserted. It does not clear the transfer request signal either.

12. **Pause state / Cancellation of transfer pause**
    See Step 10 in the software transfer procedure.

    When an instruction to cancel the pause is issued while it is in Pause state, it returns to Transfer state. If the transfer request signal was asserted in the previous Pause state, the operation to follow varies as shown below, depending on the transfer mode.

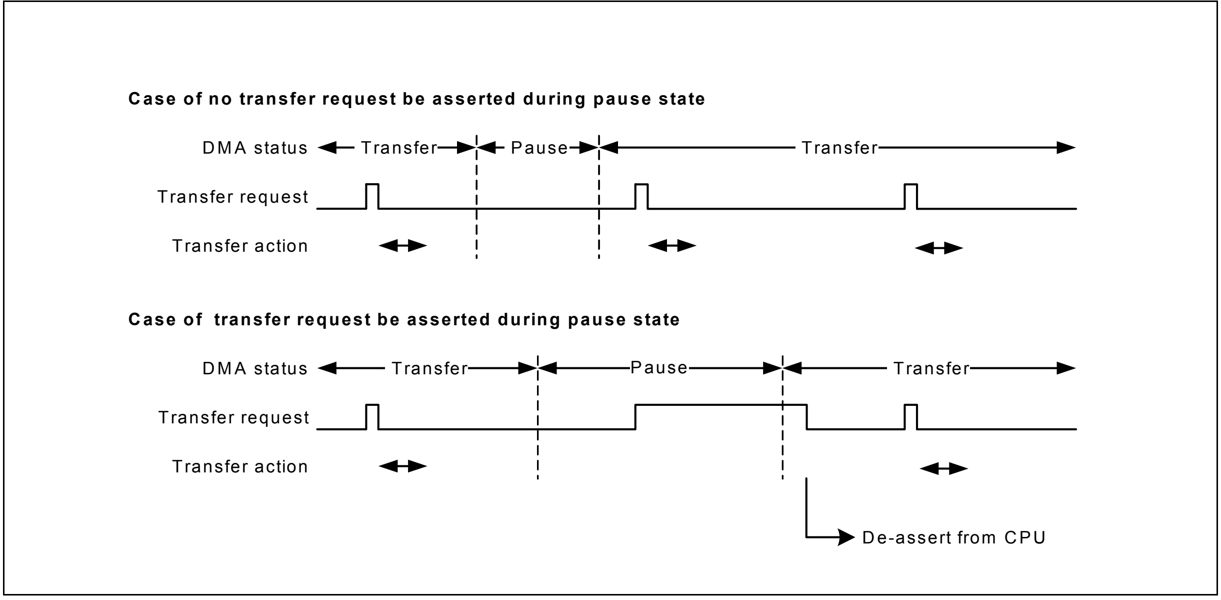
In the case of Demand transfer mode, the transfer request signal remains asserted from the Pause state. Therefore, the transfer is resumed when DMAC returns to Transfer state, and the transfer request signal is cleared as normal. See Figure 4-8.

Figure 4-8 Operation of Demand Transfer in Pause State



In the case of Block transfer mode, the transfer request signal remains asserted. Even when it returns to Transfer state, the rising edge of the transfer request signal is not detected, and the transfer is not resumed. Therefore, the transfer request is ignored during Pause state. Also, the transfer request signal is not cleared from DMAC. To resume the transfer which has been put on pause, instruct from CPU the Peripheral to deassert the transfer request signal after an instruction to cancel the pause is issued to DMAC. After that, the transfer will be resumed when the next transfer request is generated from the Peripheral. In this case, attention must be paid to the difference between the number of transfer requests output from the Peripheral and the number of transfer requests received by DMAC. See Figure 4-9.

Figure 4-9 Operation of Block Transfer in Pause State



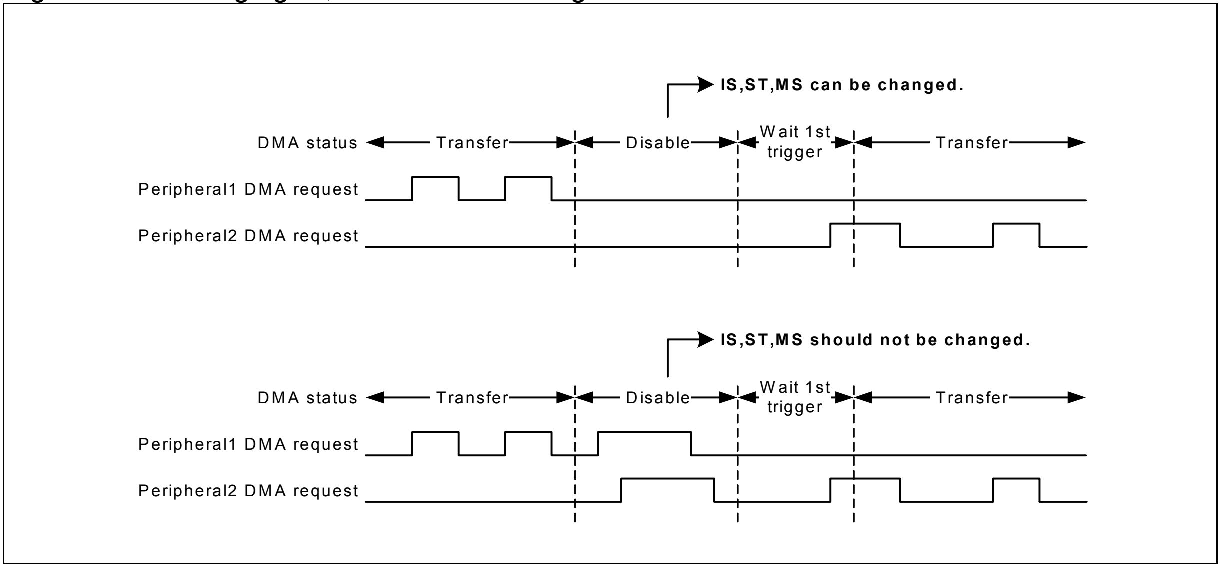
13. Operation in Disable state and Wait-1st-trigger state

See Step 11 in the software transfer procedure.

If the transfer request signal is not asserted to the channel in Disable state, the specifications of the transfer content can be changed freely (rewriting to registers DMACSA, DMACDA, DMACA[29:0], and DMACB).

If the transfer request signal is asserted or may be asserted to the channel in Disable state, the specifications of IS, ST and MS in the transfer content cannot be changed. If an attempt is made to change these settings, DMAC may perform unexpected behaviors. To change the settings of IS, ST and MS, first clear the transfer request signal to both of the Peripherals (used before and after the change) from CPU, and then always change the settings while the transfer request signal is deasserted. See Figure 4-10.

Figure 4-10 Changing IS, ST and MS Settings



The specifications of the transfer content cannot be changed to the channel in Wait-1st-trigger state from CPU

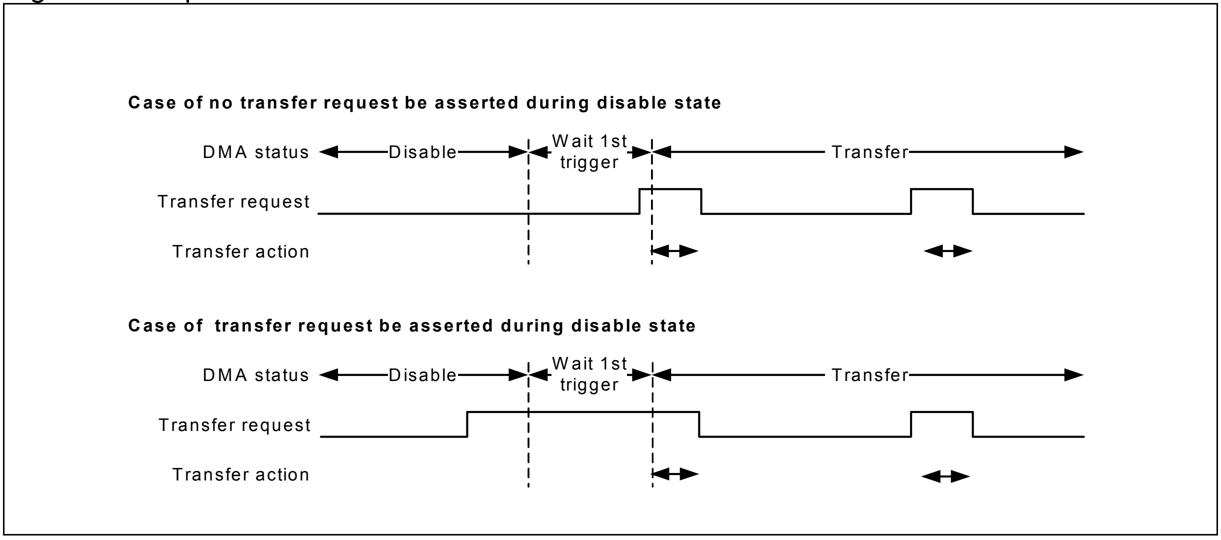
If the transfer request signal is not asserted to the channel in Wait-1st-trigger state, it moves to Disable state when CPU issues an instruction to disable individual- or all-channel operation or an instruction to put individual- or all-channel operation on pause. In this case, it is considered that the enabled transfer has been cancelled. In any case, SS does not change.

If the transfer request signal may possibly be asserted to the channel in Wait-1st-trigger state, it should be noted that DMAC has already started or completed the transfer before the attempted cancellation of the enabled transfer from CPU.

In Disable state, DMAC does not start the transfer or clear the transfer request, even if the transfer request signal is asserted. If it moves to Wait-1st-trigger state by instruction from CPU while the transfer request signal is asserted, the following operation applies (only when the settings of IS, ST and MS are not intended to be changed, as explained earlier).

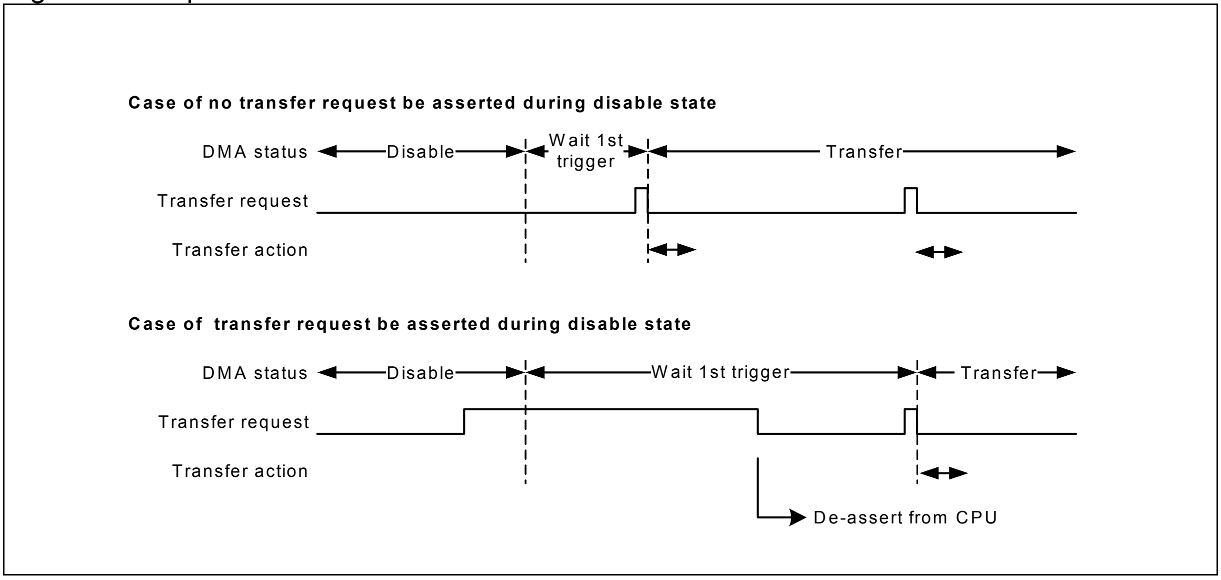
In the case of Demand transfer mode, DMAC immediately moves to Transfer state and starts the transfer, because the transfer request signal remains asserted. The transfer request signal is cleared from DMAC as normal. See Figure 4-11.

Figure 4-11 Operation of Demand Transfer in Disable State



In the case of Block transfer mode, the transfer request signal remains asserted. Even when it moves to Wait-1st-trigger state, the rising edge of the transfer request signal is not detected, and the transfer is not resumed. Therefore, the transfer request is ignored during Disable state. Also, the transfer request signal is not cleared from DMAC. To resume the transfer, instruct DMAC to move to Wait-1st-trigger state, and then instruct from CPU the Peripheral to deassert the transfer request signal. After that, it will move to Transfer state and the transfer will be resumed when the next transfer request is generated from the Peripheral. In this case, attention must be paid to the difference between the number of transfer requests output from the Peripheral and the number of transfer requests received by DMAC. See Figure 4-12.

Figure 4-12 Operation of Block Transfer in Disable State



Additional Matter 1

See Additional Matter 1 in the software transfer procedure.

In the case of hardware transfer, always write "0" to ST.

Additional Matter 2

See Additional Matter 2 in the software transfer procedure.

Additional Matter 3

See Additional Matter 3 in the software transfer procedure.

Additional Matter 4

See Additional Matter 4 in the software transfer procedure.

Additional Matter 5

If the transfer request signal (interrupt signal) from the Peripheral needs to be deasserted, the following method is available. Normally, the interrupt signal from the Peripheral is the interrupt factor flag masked (logic AND) by the interrupt enable flag. The interrupt signal can be deasserted by resetting either of the flags. When the interrupt enable flag is reset and then set, the rising edge occurs to the interrupt signal. Following this procedure can notify DMAC of the transfer request for Block transfer again. For details, check the manual for each Peripheral.

4.4. DMAC Operation and Control Procedure for Hardware (EM=1) Transfer

This section explains DMAC operation and control procedure for hardware (EM=1) transfer.

Figure 4-13 Transitional Diagram of Hardware (EM=1) Transfer State

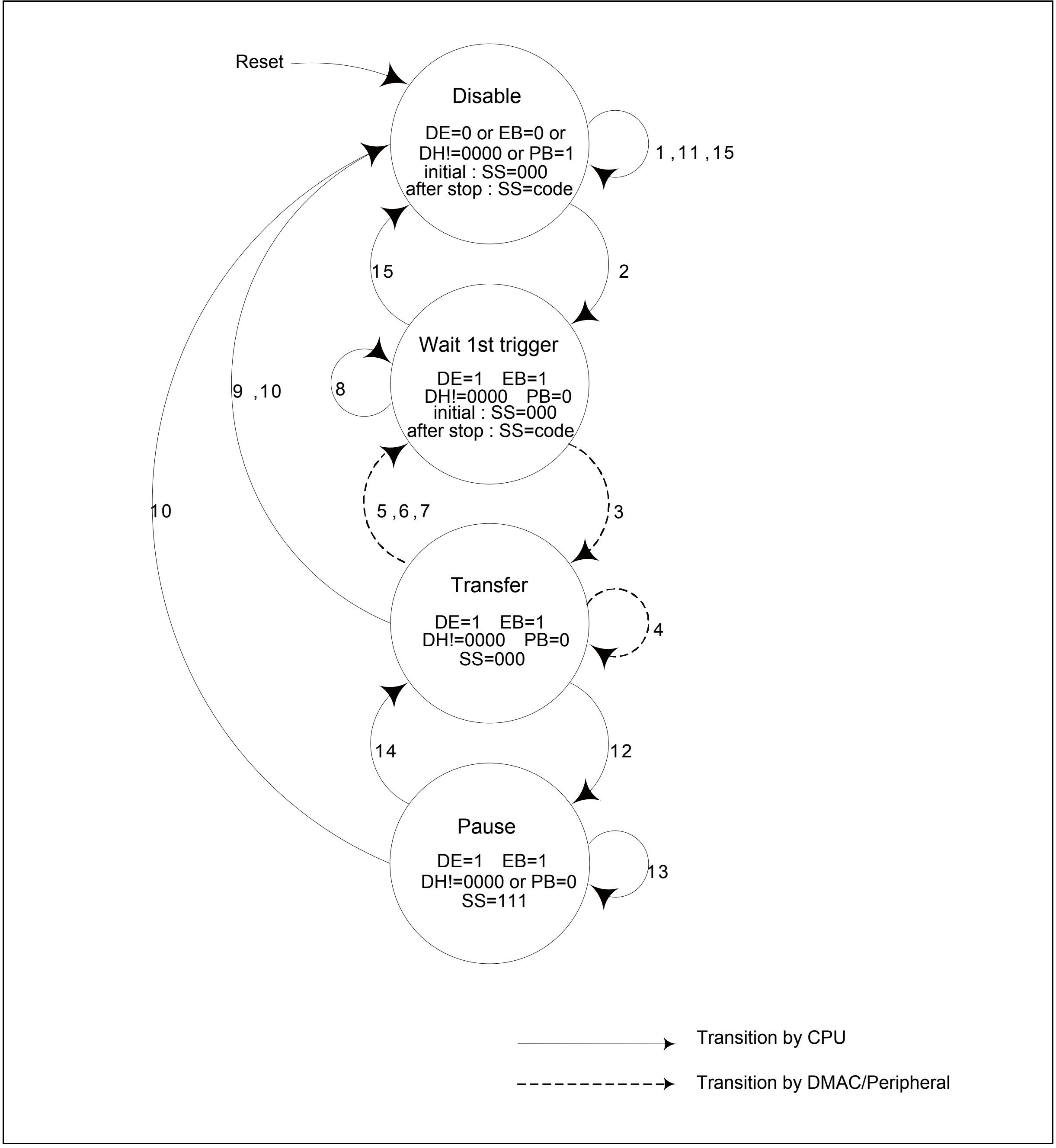


Figure 4-13 shows a transitional diagram of the states of the channel to be controlled for hardware (EM=1) transfer. The numbers next to the transitional lines in Figure 4-13 correspond to the numbers which appear in the following control procedures. The solid transitional lines indicate transitions of state instructed by CPU, while the broken transitional lines indicate transitions of state due to DMAC/Peripheral operation.

EM (Enable bit clear mask) is a bit that masks EB clear upon the completion of transfer of the channel to be controlled. EM=1 enables the same transfer process to be repeated without giving instructions from CPU.

■ Description of Each State

Disable state

See the hardware transfer (EM=0) procedure.

Wait-1st-trigger state

See the hardware transfer (EM=0) procedure.

Transfer state

In this state, the channel to be controlled has received the first transfer request from the Peripheral. A channel in this state performs transfer operation as specified. In the case of EM=1, it moves to Wait-1st-trigger state, once all the transfer operation is completed. It also changes its state upon instruction from CPU.

Pause state

See the hardware transfer (EM=0) procedure.

■ Explanation of Control Procedure

1. Disable state / Preparation for transfer

See Step 1 in the hardware transfer (EM=0) procedure.

To set EM=1, set all of the reload specifications for the transfer content (RC, RS, RD) in order to prevent data transfer in an unintended address area. Also, CI is not set, because it is meaningless to generate a successful transfer completion interrupt from DMAC. EI is set to generate an unsuccessful transfer completion interrupt from DMAC.

2. Disable state => Wait-1st-trigger state / Enabling transfer

See Step 2 in the hardware transfer (EM=0) procedure.

3. Wait-1st-trigger state / Start of transfer

See Step 3 in the hardware transfer (EM=0) procedure.

4. Transfer state

See Step 4 in the hardware transfer (EM=0) procedure.

5. Transfer state => Wait-1st-trigger state / Successful completion of transfer

When transfers are successfully completed for the number of times calculated by (BC+1) × (TC+1), the channel in Transfer state does not clear EB but does clear PB and ST and moves to Wait-1st-trigger. It sets SS[2:0]=101 to provide the notification of the successful completion. As CI is not set, no successful transfer completion interrupt is generated. Since RC, RS and RD are set, the specifications of the transfer content of BC, TC, DMACSA and DMACDA are reloaded.

6. Transfer state => Wait-1st-trigger state / Transfer error end

See Step 6 in the hardware transfer (EM=0) procedure.

In the case of EM=1, EB is not cleared even if the transfer ends due to an error. It clears PB and ST, moves to Wait-1st-trigger state and waits for the next transfer request. Therefore, it is recommended not to use DMA transfer with EM=1 in an address area where a transfer error may occur.

7. Transfer state =>Wait-1st-trigger state /End of Peripheral stop request

See Step 7 in the hardware transfer (EM=0) procedure.

In the case of EM=1, EB is not cleared even if a stop request is issued from the Peripheral. It clears PB and ST and moves to Wait-1st-trigger state. Since RC, RS and RD are set, the specifications of the transfer content of BC, TC, DMACSA and DMACDA are reloaded. As EI is set, an unsuccessful transfer completion interrupt is generated.

8. Wait-1st-trigger state / Post-transfer process

In the case of EM=1, EB is not cleared upon the completion of the transfer. (DE=1, EB=1, DH=0000, PB=0) is set and it moves to Wait-1st-trigger state. When the next transfer request is generated from the Peripheral, therefore, the next transfer starts without an instruction from CPU.

If it moves to Wait-1st-trigger state due to a stop request from the Peripheral, an unsuccessful completion interrupt occurs and that state can be confirmed. Also, the transfer request signal is masked while the stop request signal is asserted. Even if the next transfer request signal is asserted from the Peripheral, it will not be recognized and the channel to be controlled will remain in Wait-1st-trigger state, waiting for an instruction from CPU.

In the above case, SS[2:0] is read from CPU to check the state of the transfer completion. The interrupt signal is deasserted by clearing SS[2:0] from CPU. CPU clears EB and it returns to Disable state (this operation is the operation shown in Step 15 of the hardware transfer (EM=1) procedure). The transfer request signal and the stop request signal from the Peripheral are deasserted, as shown in Step 7 of the hardware transfer (EM=0) procedure

9. Transfer state => Disable state / Completion of transfer by EM=0

The operation can exit from the loop of Wait-1st-trigger state and Transfer state by writing EM=0 from CPU. At the timing when the transfer stops after the instruction, EB, ST and PB are cleared and the Transfer state changes to Disable state (DE=1, EB=0, DH=0000, PB=0) to successfully complete the transfer. In this case, no successful transfer completion interrupt is generated, as CI is not set.

10. Transfer state, Pause state => Disable state / Forced termination of transfer

See Step 8 in the hardware transfer (EM=0) procedure.

The operation can exit from the loop of Wait-1st-trigger state and Transfer state by an operation disable instruction. When an instruction to disable individual-channel operation is issued, the relevant channel moves to Disable state (DE=1, EB=0, DH=0000, PB=0) and stops the operation. When an instruction to enable all-channel operation is issued, it moves to Disable state (DE=0, EB=1, DH=0000, PB=0) and stops the operation. In the case of an instruction to disable all-channel operation, EB is not cleared either; therefore, attention must be paid.

When the operation exits from Transfer state, an unsuccessful transfer completion interrupt occurs because it is unsuccessful completion due to the forced stop. When it exits from Wait-1st-trigger state, the enabled transfer is cancelled (this operation is the operation shown in Step 15 of the hardware transfer (EM=1) procedure).

11. Disable state / Post-transfer processing

See Step 9 in the hardware transfer (EM=0) procedure.

12. Transfer state, Pause state / Transfer pause

See Step 10 in the hardware transfer (EM=0) procedure.

13. Pause state

See Step 11 in the hardware transfer (EM=0) procedure.

14. Pause state / Cancellation of transfer pause

See Step 12 in the hardware transfer (EM=0) procedure.

15. Operation in Disable state and Wait-1st-trigger state

See Step 13 in the hardware transfer (EM=0) procedure.

In the case of EM=1, the Transfer state changes directly to Wait-1st-trigger state. Therefore, the specifications of the transfer content cannot be rewritten during the repeated transfer operation (rewriting the registers DMACSA, DMACDA, DMACB[31:1] and DMACA[28:0]).

Additional Matter 1

See Additional Matter 1 in the hardware transfer (EM=0) procedure.

Additional Matter 2

See Additional Matter 2 in the hardware transfer (EM=0) procedure.

In the case of EM=1, Additional Matter 2 does not apply, because EB is not cleared during the transfer operation.

Additional Matter 3

See Additional Matter 3 in the hardware transfer (EM=0) procedure.

Additional Matter 4

See Additional Matter 4 in the hardware transfer (EM=0) procedure.

The following explains what must be noted when setting interrupts from DMAC with EM=1. As the target channel does not change from Wait-1st-trigger state due to an unsuccessful completion interrupt by a stop request from the Peripheral, the interrupt signal is not deasserted until it is cleared from CPU. Similarly, as the target channel moves to Disable state due to an unsuccessful transfer completion interrupt by a stop request from software, the interrupt signal is not deasserted until it is cleared from CPU. Other successful transfer completion interrupts and unsuccessful transfer completion interrupts may be deasserted at a timing that is not intended by CPU, if the relevant channel moves to Transfer state. Therefore, attention must be paid.

Additional Matter 5

See Additional Matter 5 in the hardware transfer (EM=0) procedure.

5. Registers of DMAC

This section explains each register function of DMAC.

5.1. List of Registers

Table 5-1 shows a list of DMAC control registers.

Table 5-1 List of DMAC Control Registers

Abbreviation	Ch. Controlled	Register name	Reference
DMACR	All	Entire DMAC configuration register	5.2
DMACA0	ch.0	Configuration A register	5.3
DMACB0		Configuration B register	5.4
DMACSA0		Transfer source address register	5.5
DMACDA0		Transfer destination address register	5.6
DMACA1	ch.1	Configuration A register	5.3
DMACB1		Configuration B register	5.4
DMACSA1		Transfer source address register	5.5
DMACDA1		Transfer destination address register	5.6
DMACA2	ch.2	Configuration A register	5.3
DMACB2		Configuration B register	5.4
DMACSA2		Transfer source address register	5.5
DMACDA2		Transfer destination address register	5.6
DMACA3	ch.3	Configuration A register	5.3
DMACB3		Configuration B register	5.4
DMACSA3		Transfer source address register	5.5
DMACDA3		Transfer destination address register	5.6
DMACA4	ch.4	Configuration A register	5.3
DMACB4		Configuration B register	5.4
DMACSA4		Transfer source address register	5.5
DMACDA4		Transfer destination address register	5.6
DMACA5	ch.5	Configuration A register	5.3
DMACB5		Configuration B register	5.4
DMACSA5		Transfer source address register	5.5
DMACDA5		Transfer destination address register	5.6
DMACA6	ch.6	Configuration A register	5.3
DMACB6		Configuration B register	5.4
DMACSA6		Transfer source address register	5.5
DMACDA6		Transfer destination address register	5.6
DMACA7	ch.7	Configuration A register	5.3
DMACB7		Configuration B register	5.4
DMACSA7		Transfer source address register	5.5
DMACDA7		Transfer destination address register	5.6

5.2. Entire DMAC Configuration Register (DMACR)

This section explains entire DMAC configuration register (DMACR).

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	DE	DS	Reserved	PR	DH[3:0]				Reserved							
Attribute	R/W	R/W	-	R/W	R/W				-							
Initial Value	0	0	-	0	0000				-							

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved															
Attribute	-															
Initial Value	-															

[bit31] DE : DMA Enable (all-channel operation enable bit)

This bit controls the enabling and disabling of transfer operations for all of the channels.
When "1" is set to this bit, the operations of all of the channels are enabled and each channel operates according to its settings.
When "0" is set to this bit, the operations of all of the channels are disabled, and no transfer is performed until "1" is set to the bit. Also, a channel in the middle of its transfer operation is forced to stop the transfer.

This bit can be used to force all of the channels that are currently performing a transfer to stop it and reset the configuration register.

bit	Function
0	Disables the operations of all of the channels. (Initial value)
1	Enables the operations of all of the channels.

[bit30] DS : DMA Stop

This bit indicates the transfer state of all of the channels.
If either of the following conditions is established during transfer operation, the bit is set to "1" by DMAC.

- When "0" is written to the DMACR:DE bit and then the transfers of all of the channels are completed.
- When a value other than "0000" is written to the DMACR:DH bit and then the transfers of all of the channels pause.

When DMACR:DE=1 and DMACR:DH=0000 are set and all of the channels become enabled to operate, this bit is set to "0" by DMAC.

Although the attribute of this bit is R/W, writing to it by CPU does not affect DMAC’s operation. If, however, the DMACR register needs to be updated without affecting the state of this bit, first read from this bit and then rewrite the same value.

bit	Function
0	Clears the disabling of all-channel operation or the setting of all-channel pause. (Initial value)
1	The transfers of all of the channels have stopped due to the disabling of all-channel operation or the setting of all-channel pause.

[bit29] Reserved: Reserved bit

"0" is read out from this bit.
When writing this bit, set it to "0".

[bit28] PR : Priority Rotation

This bit controls the order of transfer priority among channels.
When this bit is set to "0", the priority order is fixed for all of the channels.
When this bit is set to "1", the priority order is determined in a rotation method for all of the channels.

bit	Function
0	Fixes the priority order. (ch.0>ch.1>ch.2>ch.3>ch.4>ch.5>ch.6>ch.7) (Initial value)
1	Applies the rotation method to the priority order.

For selection of the transfer priority order, see Section "3.5 Channel Priority Control".

[bit27:24] DH : DMA Halt (All-channel pause bit)

This bit controls the pause/cancellation of transfer operations for all of the channels.
When this bit is set to a value other than "0000", all of the channels that are currently performing a transfer are put on pause. When it is set to "0000", the transfers are resumed.

Even if a transfer request from an external/peripheral device is asserted, the channels in Pause state ignore the transfer request. In the cases of Block transfer and Burst transfer, the relevant channel does not start a transfer, even if the pause is cleared. In order to complete a transfer when a pause is set during the transfer, an additional transfer request is required after the pause is cancelled.

This bit can be used to put a transfer on pause without resetting the configuration registers of all of the channels.

bit27:24	Function
0000	Cancels the pause of transfers for all of the channels. (Initial value)
Other than 0000	Puts the transfers of all of the channels on pause.

[bit23:0] Reserved: Reserved bits

"0" is read out from these bits.
When writing tthese bits, set them to "0".

5.3. Configuration A Register (DMACA)

This section explains configuration A register (DMACA).

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	EB	PB	ST	IS[5:0]						Reserved			BC[3:0]			
Attribute	R/W	R/W	R/W	R/W						-			R/W			
Initial Value	0	0	0	000000						-			0000			

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	TC[15:0]															
Attribute	R/W															
Initial Value	0x0000															

[bit31] EB : Enable bit (individual-channel operation enable bit)

This bit controls the enabling and disabling of the transfer operation of an individual channel.
When this bit is set to "1", the relevant channel is enabled to operate and waits for a trigger to start its transfer operation (the DMACR:DE must be set to "1").

If the EM bit (DMACB[0]) is not set to "1", DMAC clears this bit to "0" upon the completion of the transfer.

When this bit is set to "0", the relevant channel is disabled to operate and does not perform transfer operation until it is set to "1". Also, if it is in the middle of transfer operation, it is forced to stop the transfer. This bit can be used to force the relevant channel that is currently in transfer operation to stop it and reset the configuration register.

bit	Function
0	The operation of the relevant channel is disabled. (Initial value)
1	The operation of the relevant channel is enabled.

[bit30] PB : Pause bit (individual-channel pause bit)

This bit controls the pause/cancellation of the transfer operation of an individual channel.
When this bit is set to "1" and the relevant channel is currently in transfer operation, it puts the transfer on pause. When this bit is set to "0", it resumes the transfer.

This bit is cleared to "0", when the transfer operation of the channel is completed.

Even if a transfer request from an external/peripheral device is asserted, the channels in Pause state ignore the transfer request. In the cases of Block transfer and Burst transfer, the relevant channel does not start a transfer, even if the pause is cleared. In order to complete a transfer when a pause is set during the transfer, an additional transfer request is required after the pause is cancelled.

This bit can be used to put a transfer on pause without resetting the configuration register of the relevant channel.

bit	Function
0	Cancels the pause of the transfer of the relevant channel.
1	Puts the transfer of the relevant channel on pause.

[bit29] ST : Software Trigger

This bit is used to generate a software transfer request for an individual channel.

When this bit is set to "1", a trigger is generated by the software transfer request and the relevant channel starts its transfer. After the completion of the transfer, DMAC clears this bit to "0".
When this bit is set to "0" during the transfer, the transfer stops.

bit	Function
0	No software transfer request (Initial value)
1	Software transfer request available

[bit28:23] IS[5:0] : Input Select

These bits select the trigger for transfer requests.

When the transfer trigger is set to software request (ST=1), set the IS[5:0] bits to "000000".
When the transfer trigger is set to hardware request, specify which Peripheral’s interrupt signal to be used to start transfer. Any Peripheral can be selected for all of the channels.

The hardware transfer request signal to be connected to DMAC varies depending on the product used.
Check the transfer request signal to be connected in "2.2 I/O Signals of DMAC" before setting the selection.

bit28:23	Function
000000	Software (Initial value)
100000	IDREQ[0]
100001	IDREQ[1]
100010	IDREQ[2]
100011	IDREQ[3]
100100	IDREQ[4]
100101	IDREQ[5]
100110	IDREQ[6]
100111	IDREQ[7]
101000	IDREQ[8]
101001	IDREQ[9]
101010	IDREQ[10]
101011	IDREQ[11]
101100	IDREQ[12]
101101	IDREQ[13]
101110	IDREQ[14]
101111	IDREQ[15]
110000	IDREQ[16]
110001	IDREQ[17]
110010	IDREQ[18]
110011	IDREQ[19]

bit28:23	Function
110100	IDREQ[20]
110101	IDREQ[21]
110110	IDREQ[22]
110111	IDREQ[23]
111000	IDREQ[24]
111001	IDREQ[25]
111010	IDREQ[26]
111011	IDREQ[27]
111100	IDREQ[28]
111101	IDREQ[29]
111110	IDREQ[30]
111111	IDREQ[31]
Setting other than above	Setting is prohibited.

[bit22:20] Reserved: Reserved bits

"0" is read out from these bits.
When writing tthese bits, set them to "0".

[bit19:16] BC[3:0] : Block Count

These bits specify the number of blocks for Block/Burst transfer.

When the transfer mode is set to Demand transfer, set BC[3:0] to "0000".
Set the value "BC[3:0]=Number of blocks - 1". The maximum allowed number of blocks is 16.
The value of these bits can be read during a transfer. Normally, as one transfer source access or one transfer destination access is completed successfully, BC[3:0] is decreased by 1.

In the case of DMACB:RC=1, the value set when the transfer started is reloaded upon the completion of the transfer.
In the case of DMACB:RC=0, the value is set to "0" upon successful completion of the transfer, while the value remains the same value as set during the transfer suspension upon unsuccessful completion of the transfer.

bit19:16	Function
	Number of transfer blocks (Initial value : 0x0)

[bit15:0] TC[15:0] : Transfer Count

These bits specify the number of transfers for Block/Burst/Demand transfer.

Set the value "TC = Number of transfers - 1". The maximum allowed number of transfers is 65536.

The value of these bits can be read during a transfer. Normally, as the transfer of one block is completed, TC is decreased by 1.

In the case of DMACB:RC=1, the value set when the transfer started is reloaded upon the completion of the transfer.
In the case of DMACB:RC=0, the value is set to "0" upon successful completion of the transfer, while the value remains the same value as set during the transfer suspension upon unsuccessful completion of the transfer.

bit15:0	Function
	Number of transfers (Initial value : 0x0000)

5.4. Configuration B Register (DMACB)

This section explains configuration B register (DMACB).

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	Reserved		MS[1:0]		TW[1:0]		FS	FD	RC	RS	RD	EI	CI	SS[2:0]		
Attribute	R/W		R/W		R/W		R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		
Initial Value	00		00		00		0	0	0	0	0	0	0	000		

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved								Reserved							EM
Attribute	R/W															R/W
Initial Value	000000000000000															0

[bit31:30] Reserved: Reserved bits

"0" is read out from these bits.
When writing tthese bits, set them to "0".

[bit29:28] MS[1:0] : Mode Select

These bits select the transfer mode.

bit	Function
00	Block transfer mode (Initial value)
01	Burst transfer mode
10	Demand transfer mode
11	Reserved

[bit27:26] TW[1:0] : Transfer Width

These bits specify the bit width of transfer data.

bit	Function
00	Byte (8 bits) (Initial value)
01	Half-word (16 bits)
10	Word (32 bits)
11	Reserved

[bit25] FS : Fixed Source

This bit specifies whether to increment or fix the transfer source address.

bit	Function
0	Increments the transfer source address according to TW[1:0]. (Initial value)
1	Fixes the transfer source address.

[bit24] FD : Fixed Destination

This bit specifies whether to increment or fix the transfer destination address.

bit	Function
0	Increments the transfer destination address according to TW[1:0]. (Initial value)
1	Fixes the transfer destination address.

[bit23] RC : Reload Count (BC/TC reload)

This bit controls the reload function of BC[3:0] and TC[15:0].
When this bit is set to "1", the value set when the transfer started is reloaded to BC[3:0] and TC[15:0] upon completion of the transfer.

bit	Function
0	Disables the reload function of BC/TC. (Initial value)
1	Enables the reload function of BC/TC.

[bit22] RS : Reload Source

This bit controls the reload function of the transfer source address.
When this bit is set to "1", the value set when the transfer started is reloaded to DMACSA upon completion of the transfer.

bit	Function
0	Disables the reload function of the transfer source address. (Initial value)
1	Enables the reload function of the transfer source address.

[bit21] RD : Reload Destination

This bit controls the reload function of the transfer destination address (DMACDA).
When this bit is set to "1", the value set when the transfer started is reloaded to DMACDA upon completion of the transfer.

bit	Function
0	Disables the reload function of the transfer destination address. (Initial value)
1	Enables the reload function of the transfer destination address.

[bit20] EI :Error Interrupt (unsuccessful transfer completion interrupt enable)

This bit enables or disables the notification of an interrupt when a transfer has been unsuccessfully completed.
When this bit is set to "1", an interrupt is issued if SS[2:0] is in the following status upon completion of the transfer.

- Address overflow
- Stop by transfer stop request from a Peripheral, or the disabling of transfer by the EB/DE bit
- Transfer source access error
- Transfer destination access error

bit	Function
0	Disables an interrupt to be issued upon unsuccessful completion of transfer. (Initial value)
1	Enables an interrupt to be issued upon unsuccessful completion of transfer.

[bit19] CI :Completion Interrupt : (successful transfer completion interrupt enable)

This bit enables or disables the notification of an interrupt when a transfer has been successfully completed.

When this bit is set to "1", an interrupt is generated, if SS[2:0] is set to successful completion upon completion of the transfer.

bit	Function
0	Disables an interrupt to be issued upon successful completion of transfer. (Initial value)
1	Enables an interrupt to be issued upon successful completion of transfer.

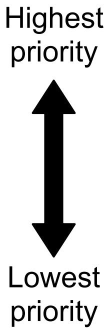
[bit18:16] SS[2:0] : Stop Status (stop status notification)

These bits represent a code that indicates the stop status or completion status of a transfer.

The following table shows the available codes.
If a successful transfer completion interrupt or unsuccessful transfer completion interrupt is issued, the interrupt signal is deasserted by writing "000" to these bits.

bit18:16	Description
000	Initial value
001	Termination by transfer error (address overflow)
010	Termination by transfer stop request (stop by transfer stop request for Peripheral or the disabling of transfer by the EB/DE bit)
011	Termination by transfer error (transfer source access error)
100	Termination by transfer error (transfer destination access error)
101	Successful transfer completion
110	Reserved
111	Transfer on pause

If various errors occur simultaneously, the termination code is indicated according to the following priority.



Reset
Clearing by writing "000"
Address overflow
Stop request
Transfer source access error
Transfer destination access error

[bit15:8] Reserved: Reserved bits

"0" is read out from these bits.
When writing tthese bits, set them to "0".

[bit7:1] Reserved: Reserved bits

"0" is read out from these bits.
When writing tthese bits, set them to "0".

[bit0] EM : Enable bit Mask (EB bit clear mask)

This bit is used to mask the clear of the EB bit (DMACA[31]) from DMAC upon completion of the transfer.

In the case of EM=0, DMAC clears the EB bit (DMACA[31]) to "0" upon completion of the transfer.
In the case of EM=1, it does not clear the EB bit upon completion of the transfer. This function allows transfers to be repeated without instruction from CPU.

This function can only be used for hardware transfer. To use the function, enable the reload function of RC, RS and RD bits.

bit	Function
0	Clears DMACA:EB upon completion of the transfer. (Initial value)
1	Dose not clear DMACA:EB upon completion of the transfer.

5.5. Transfer Source Address Register (DMACSA)

This section explains transfer source address register (DMACSA).

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	DMACSA[31:16]															
Attribute	R/W															
Initial Value	0x0000															

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	DMACSA[15:0]															
Attribute	R/W															
Initial Value	0x0000															

[bit31:0] DMACSA[31:0] : DMAC Source Address

These bits specify the transfer start address of the transfer source.
It is not possible to set unaligned address to transfer data width (TW[1:0]). The value of these bits can be read during the transfer.

In the case of DMACB:FS=1, the transfer source address is set to a fixed value and no change occurs.

In the cases of DMACB:FS=0 and DMACB:RS=0, the value is incremented according to TW[1:0].
Upon successful transfer completion, it is the next address after the transfer completion address.
Upon unsuccessful transfer completion, it is the value set during the suspension.

In the cases of DMACB:FS=0 and DMACB:RS=1, it is incremented according to TW[1:0] during the transfer. Upon completion of the transfer, the value set when the transfer started is reloaded.

bit31:0	Function
	Specifies the transfer source address from which the transfer starts. (Initial value: 0x00000000)

5.6. Transfer Destination Address Register (DMACDA)

This section explains transfer destination address register (DMACDA).

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	DMACDA[31:16]															
Attribute	R/W															
Initial Value	0x0000															

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	DMACDA[15:0]															
Attribute	R/W															
Initial Value	0x0000															

[bit31:0] DMACDA[31:0] : DMAC Destination Address

These bits specify the transfer start address of the transfer destination.
It is not possible to set unaligned address to transfer data width (TW[1:0]). The value of these bits can be read during the transfer.

In the case of DMACB:FD=1, the transfer destination address is set to a fixed value and no change occurs.

In the cases of DMACB:FD=0 and DMACB:RD=0, the value is incremented according to TW[1:0].
Upon successful transfer completion, it is the next address after the transfer completion address.
Upon unsuccessful transfer completion, it is the value set during the suspension.

In the cases of DMACB:FD=0 and DMACB:RD=1, it is incremented according to TW[1:0] during the transfer.
Upon completion of the transfer, the value set when the transfer started is reloaded.

bit31:0	Function
	Transfer destination address from which DMA transfer starts (Initial value: 0x00000000)

6. Usage Precautions

This section explains the precautions on using DMAC.

■ Precautions on register setting

When setting DMAC register, please note the following matters.

· The DMACR, DMACA, DMACB, DMACSA and DMACDA registers can be accessed by byte, half-word and word.

· The register address in DMAC cannot be set to the DMACSA or DMACDA register.

· Channel setting registers cannot be changed during DMA transfer, except the DE/DH bits of DMACR, the EB/PB bits of DMACA and the EM bit of DMACB.

■ Precautions on STOP and TIMER mode transition

When transiting to STOP mode and TIMER mode, make sure to stop the operation of all channels of the DMAC and confirm the stop of the DMAC by DS flag. If the transition is made to STOP mode and TIMER mode while DMAC is operating, an unexpected operation can be executed when returning to RUN mode.

CHAPTER: DSTC

This chapter explains details of the DSTC (Descriptor System data Transfer Controller).

CODE: 9BFDSTC-E01.0

1. Overview of DSTC

This section provides an overview of the DSTC (Descriptor System data Transfer Controller).

■ Overview

The DSTC (Descriptor System data Transfer Controller), like the DMAC, is a function block that can transfer data at high speed bypassing the CPU. Using the Descriptor (to be called DES later in this document) System Method, it directly accesses memory or a peripheral device according to the content specified in a DES created on the memory, and executes a data transfer operation.

One set of transfer control details (basic transfer settings, number of transfers, transfer source address, transfer destination address) is specified in one DES. The DSTC can multiple DES individually and can build up to 1024 transfer channels.

The data transfer operation can be started by one of the following three methods: direct start by the CPU (software start), start by an interrupt signal from a peripheral device (hardware start), and the Chain Start Function.

The chain start function executes a transfer according to the current DES, and then starts a new transfer according to the succeeding DES or according to the current DES again. It can be specified in the DES whether to use the Chain Start Function. With the chain start function, the DSTC can incorporate other types of transfer specified in multiple DES into a single start trigger (software start / hardware start) in the start DES, and execute such types of transfer together. In addition, it can also divide a transfer operation specified in a DES into several transfer operations and then execute them.

The DSTC has two reload functions for the transfer address and for the transfer count counter (the InnerRelaod Function that during a transfer makes the value return to the one at the start of the transfer, the OuterReload Function that at the end of a transfer makes the value return to the one at the start of the transfer). The two reload functions facilitate the control of repeating the same transfer operation.

The DSTC can notify the CPU of the normal end or abnormal end of a transfer operation as an interrupt.

It can control how an internal clock is stopped in a standby mode (low power consumption mode).

The DSTC has a dedicated bus, which is independent of the CPU bus, and has a configuration enabling it to execute a transfer operation when the CPU bus is being accessed.

The configuration of the dedicated bus conforms to the system bus (AHB) and supports a 32-bit address space (4 Gbyte).

■ Number of channels supported in hardware transfer of DSTC

For a product equipped with the DSTC, if the DSTC supports 256 channels, it can use all hardware transfer channels from channel 0 to channel 255. If the DSTC supports 128 channels, it cannot use channel 128 to channel 255. If the DSTC supports 64 channels, it cannot use channel 64 to channel 255.

2. DSTC Operations Overview and DSTC System Configuration

This section provides an overview of operations of the DSTC and explains the DSTC system configuration.

■ Operations overview of DSTC

● DES System

The DSTC executes a transfer operation according to the content specified in a DES built on the memory by the CPU. As shown in Table 2-1, a DES consists of seven settings, DES0 to DES7. The settings specify transfer basic settings, the number of transfers, the transfer source address and the transfer destination address in their respective areas. (For details of the bit assignment of each DES, see "5 Registers and Descriptors of DSTC".) These settings are stored in the memory area as a single group. DES0 to DES6 are all 32 bits (1 word) in size. DES4 to DES6 are optional DES. Their settings may not need to be specified depending on the content of a transfer.

Table 2-1 Types of DES and settings

Storage address	Name	Details
DESP+0x00 (fixed)	DES0	This sets the basic settings of a transfer.
DESP+0x04 (fixed)	DES1	This sets the number of transfers.
DESP+0x08 (fixed)	DES2	This sets the transfer source address (SA) at which a transfer starts.
DESP+0x0C (fixed)	DES3	This sets the transfer source address (DA) at which a transfer ends.
DESP+0x10 - (variable)	DES4	This controls the OuterReload of DES1 at the end of a transfer.
	DES5	This controls the OuterReload of DES2 at the end of a transfer.
	DES6	This controls the OuterReload of DES3 at the end of a transfer.

Figure 2-1 illustrates the configuration of the DES System Method of the DSTC. When using the DSTC, reserve adequate free memory area for storing the DES. Select a memory area that is readable and writable because the DSTC has to refer to and update the DES. Set (1. in Figure 2-1) the start address of the memory area using the DESTP (DES-Top-address) register of the DSTC. Multiple DES can be allocated to a 4096-word (16 Kbyte) area starting from the DESTP. Up to 1024 DES can be allocated in the area. The DSTC identifies a DES according to its address value (DESP: DES-pointer) relative to the DES0 area starting from the DESTP.

● Transfer start

After transfer information has been stored in a DES (2.in Figure 2-1), the DSTC transfer can be started by one of the following three start trigger methods.

・ **Software Start (to be called SW Start later in this document)**
The SW start starts the DSTC transfer directly from the CPU. The transfer is started by writing the DESP of the DES to be used to the SWTR (Software Trigger) Register (3 in Figure 2-1).

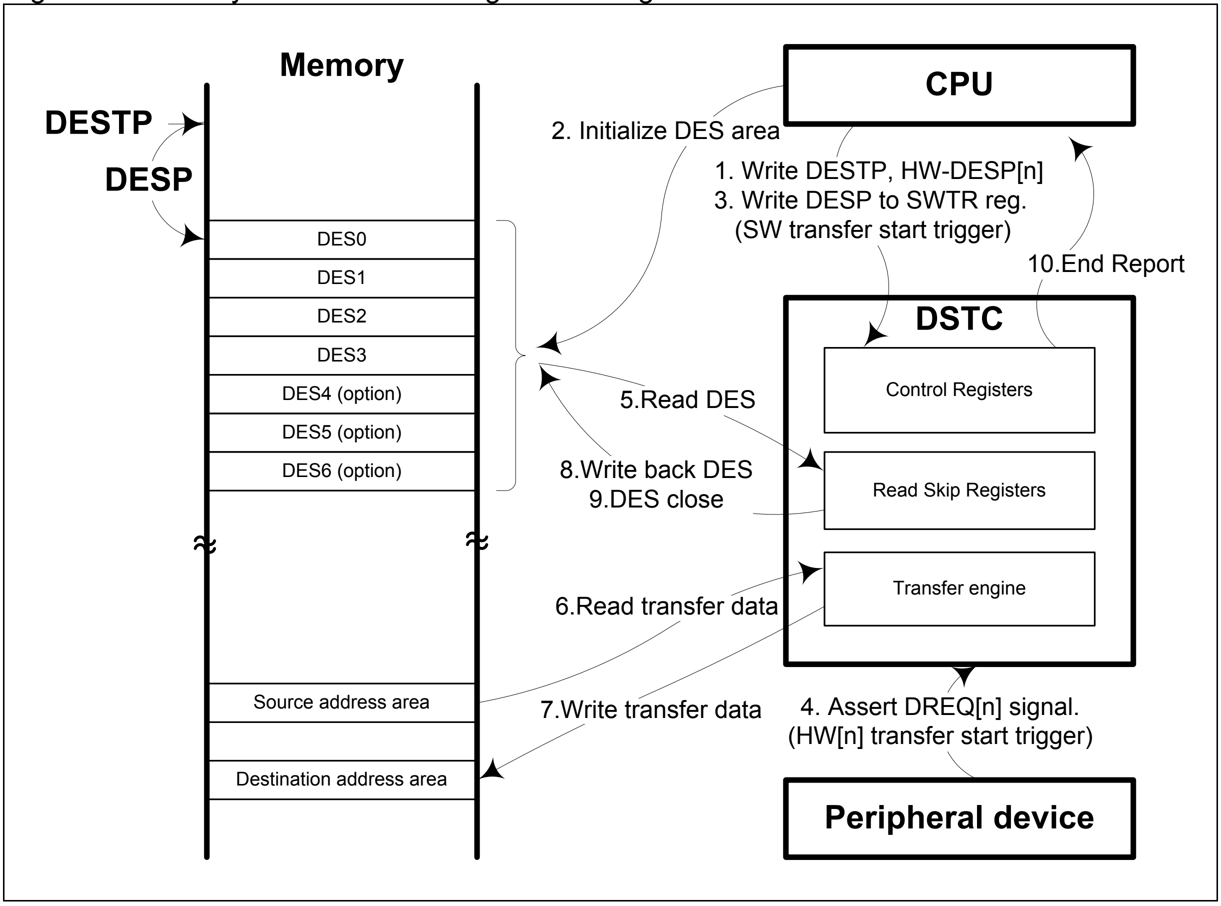
・ **Hardware Start (to be called HW Start later in this document)**
The HW start uses the interrupt signal from a peripheral as a transfer request signal to start the DSTC transfer. When an interrupt signal from a peripheral has been asserted, the HW start starts the transfer bypassing the CPU. Write the DESP of the DES to be used in the HW transfer to the HWDESP register (Hardware DESP) in the DSTC in advance (1 in Figure 2-1). If a peripheral makes a transfer request (4 in Figure 2-1), the transfer is started according to the HWDESP corresponding to an HW channel.

・ **Chain Start**

If the DES that the DSTC hast started executing as explained above has a Chain Start trigger, the DSTC starts a new transfer according to the succeeding DES or according to the same DES that the DSTC has started executing.

In the following sections, "Start Trigger" represents all the above start triggers, SW Start, HW Start and Chain Start.

Figure 2-1 DES system method configuration diagram



● Transfer operation

The DSTC refers to (5 in Figure 2-1) a DES in DETP+DESP according to a Start Trigger mentioned above. The DSTC checks (DES Open Check) details of the DES it refers to, and executes a transfer (6 and 7 in Figure 2-1) if the details have no problem. In addition, if the DES has a Chain Start Trigger, the DSTC executes a transfer data according to the Chain Start trigger.

The number of transfers to be executed at one Start Trigger varies according to details of a DES and those of the Chain Trigger. Not all transfers specified in a DES may end at one Start Trigger. If that occurs, the number of transfers ended and updated transfer addresses are written back to a DES (8 in Figure 2-1). The DSTC waits for the next Start Trigger, and continues executing the transfer after receiving the next Start Trigger.

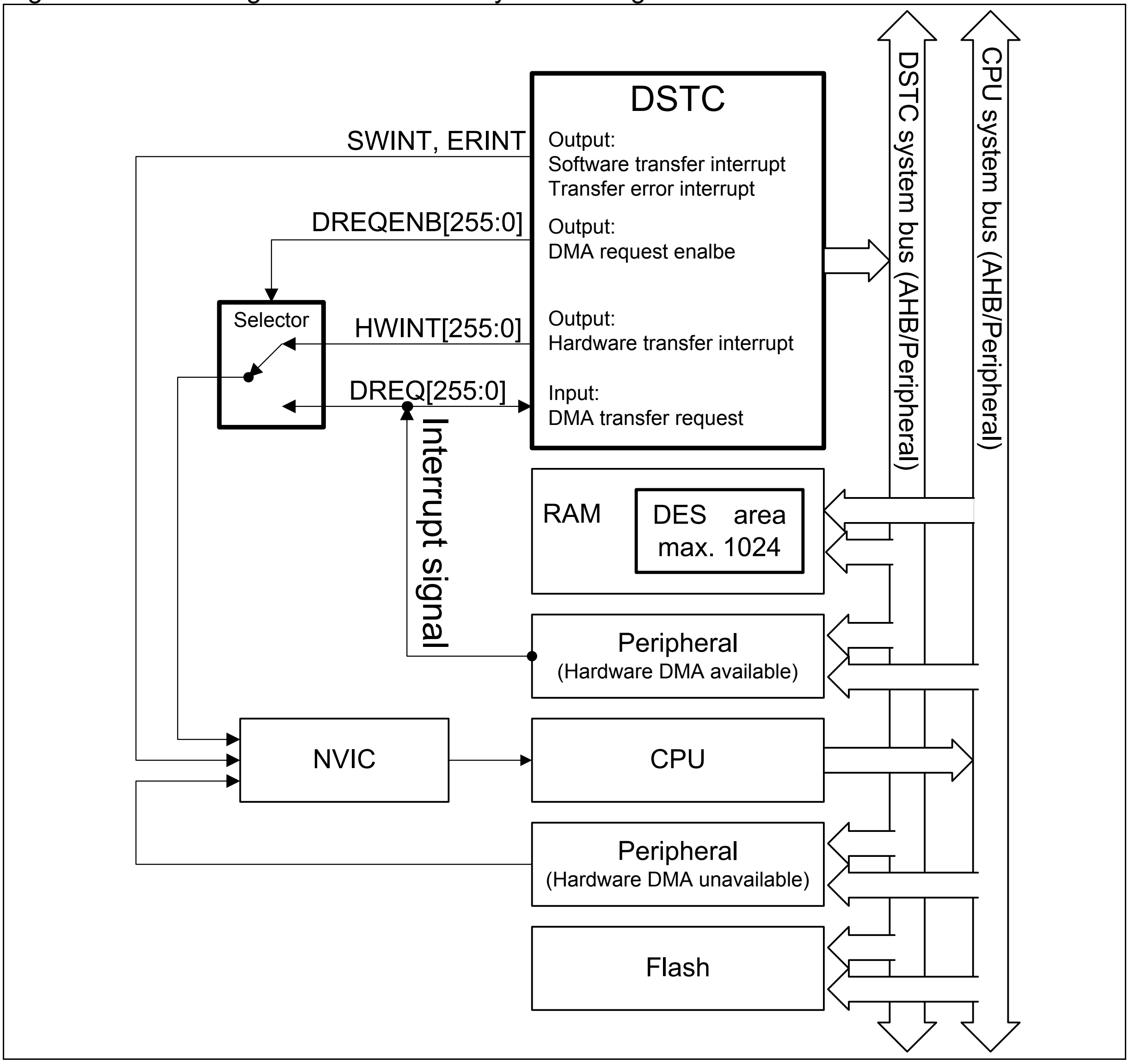
● Transfer end

The DSTC executes the DES close process (9 in Figure 2-1) after all transfers specified in a DES have ended. The DES settings can prevent the DES close process from being executed. The DSTC can notify the CPU of the following events through an interrupt: i. the DSTC is waiting for a Start Trigger; ii. a transfer has ended normally; iii. a transfer has ended abnormally. (10 in Figure 2-1)

■ DSTC and system configuration

Figure 2-2 shows the block diagram illustrating the DSTC and system configuration.

Figure 2-2 Block diagram of DSTC and system configuration.



● Connection with system

The system configuration diagram in Figure 2-2 has been simplified to facilitate explanation. For details of the system configuration, refer to chapter "System Overview" in "PERIPHERAL MANUAL". The DSTC is connected to the CPU, Flash, RAM and peripherals via the system bus. The DSTC has a dedicated bus, which is independent of the CPU bus, and has a configuration enabling it to execute a transfer operation when the CPU bus is being accessed. The DSTC accesses any address area on the system according to the specified transfer destination address and transfer source access of a channel, and executes data transfer between the memory and a peripheral. The DSTC cannot access certain areas. Refer to the memory map to check which areas the DSTC cannot access.

● Connection with DREQ and HWINT

The DSTC supports up to 256 hardware transfer request signal inputs. The interrupt signal from a peripheral supporting DSTC hardware transfer is connected to the DSTC. The DSTC can start a transfer operation with the interrupt signal from a peripheral as a DMA transfer request signal (DREQ[255:0]). The DSTC cannot start the DMA transfer of the DSTC with an interrupt signal from a peripheral not supporting DSTC hardware transfer. In the case of a peripheral having multiple channels and multiple interrupt sources, there are interrupts supporting and those not supporting the DMA transfer.

The settings of the DREQENB[255:0] determine whether hardware transfer requests from peripherals are valid. The specifications of the product equipped with the DSTC determine which bit out of 256 bits supports the interrupt signal of which peripheral. For details, see "CHAPTER: Interrupts". The following switch operation is executed in the Selector in Figure 2-2 according to the settings of the DREQENB[255:0].

If an interrupt from a peripheral is not used as a transfer request to the DSTC (DREQENB[n] = 0), an interrupt signal from that peripheral is directly connected to the NVIC. The DSTC ignores the DREQ[n] (interrupt signal) from that peripheral.

If an interrupt from a peripheral is used as a transfer request to the DSTC (DREQENB[n] = 1), the DSTC starts a transfer according to the DREQ[n]. In addition, instead of the interrupt signal (DREQ[n]) from that peripheral The HWINT[n] signal from the DSTC is connected to the NVIC (interrupt controller). The HWINT[n] signal is an interrupt signal for notifying the CPU of the event that the hardware transfer started by the DREQ[n] ended.

With the above configuration, in the process of the NVIC, an interrupt from a peripheral, and a transfer completion interrupt of the DSTC caused by an interrupt request from a peripheral jump to the same interrupt vector. Therefore, use the DREQENB[n] register to choose the interrupt to be processed.

● Connection to hardware transfer request clear signal

Among peripherals supporting the hardware transfer, there are some for which a transfer request signal (interrupt signal) has to be cleared after a transfer has ended. Though the clearing process is not mentioned in Figure 2-2, if the interrupt request signal of such peripheral is enabled by its corresponding DREQENB[n] register, the transfer request signal from the DSTC for such peripheral is to be cleared.

● Connection to hardware transfer stop request signal

The DMA transfer stop request signal from the Multi-Function Serial Unit (to be called MFS later in this document) is output. Though the clearing process is not mentioned in Figure 2-2, if the transfer stop request signal from the MFS has been asserted, the transfer request signal sent to the channel of the DSTC corresponding to the transfer stop request signal is masked, and the transfer for that transfer request signal is held. The transfer stop request signal is sent to the NVIC as an interrupt. For details, see "CHAPTER: Interrupts".

● Interrupt signal from DSTC

The transfer end interrupt for a transfer started by a software start is sent to the NVIC by the SWINT. The error interrupt generated due to the occurrence of a transfer error is sent to the NVIC by the ERINT.

3. Functions and Operations of DSTC

This section explains operations of the DSTC.

3.1. Settings of DES

This section explains setting details of the DES and operations of the DSTC.

■ Specifying transfer data size

● TW, IRM, IIN, ORM

The DSTC transfers data of the data width specified in TW in DES0 in a single transfer. There is a transfer number counter in the DSTC. The counter has a dual loop configuration consisting of an outer loop counter and an inner loop counter. ORM (outer loop remain) in DES1 indicates the remaining number of transfers of the outer loop counter, and IRM (inner loop remain) in the DES1 Register the remaining number of transfers of the inner loop counter. IIN (inner loop initial) in DES1 specifies the initial value of the inner loop counter.

At the start of a transfer, specify the transfer data width and different numbers of transfers in TW, IRM, IIN and ORM. For one DES, the DSTC transfers data of the total of TW×IIN×ORM.

Table 3-1 shows the method of specifying the transfer data width, the number of transfers and the transfer mode.

Table 3-1 Specifying transfer mode, transfer size and number of transfers

Area name	Name	Details
DES0	MODE	MODE selects a transfer mode. 0: Selects mode 0. 1: Selects mode 1.
	TW[1:0]	TW specifies the data width in a single transfer. 00: 8 bits (byte) 01: 16 bits (halfword) 10: 32 bits (word)
DES1	IIN	Specifies the initial value of the inner loop counter in the transfer number counter.
	IRM	Specifies the remain value of the inner loop counter in the transfer number counter.
	ORM	Specifies the remain value of the outer loop counter in the transfer number counter.

● MODE

If "0" is written to MODE, the DSTC executes a transfer in mode 0. In this mode, one Start Trigger makes the DSTC transfer data of the bit width specified in TW for IIN times. After having executed transfers for the number specified in IIN, the DSTC executes the Chain Start or waits for the next Start Trigger. If Start Triggers of the amount specified in ORM are sent to the DSTC, the DSTC ends the transfer of the number (IIN×ORM) specified in the DES.

If "1" is written to MODE, the DSTC executes a transfer in mode 1. In this mode, one Start Trigger makes the DSTC transfer data of the bit width specified in TW once. After having executed one transfer, the DSTC executes the Chain Start or waits for the next Start Trigger. If Start Triggers of the amount equivalent to the result of IIN×ORM are sent to the DSTC, the DSTC ends IIN×ORM times of transfer specified in the DES.

In the SW Start transfer, both mode 0 and mode 1 can be used. In the HW Start transfer, select either mode 0 or mode 1 depending on the type of peripherals that generates a Start Trigger. To make the DSTC execute a transfer at a Start Trigger from a peripheral that has to have to a handshake with the DSTC at every data transfer, use mode 1. In the case of data transfer with the MFS, ADC and USB, since the DSTC transfers data to the FIFO in a peripheral, use mode 1. In the case of a Start Trigger from a peripheral that notifies the DSTC of the transfer start timing, such as the timer, the external interrupt block, etc., both mode 0 and mode 1 can be used.

In mode 0, specifies the settings of ORM and IIN. A value of 1 to 65536 inclusive can be specified in both

ORM and IIN. The setting of IRM does not need to be specified. In the DSTC, the setting of IIN is copied to IRM.

In mode 1, specifies the settings of ORM, IIN and IRM. A value of 1 to 65536 inclusive can be specified in ORM. A value of 1 to 256 inclusive can be specified in IIN. Specify the same value in IRM and IIN.

■ Setting transfer addresses

● SA, DA, SAC[2:0], DAC[2:0]

Set the start address (SA) of the transfer source area in DES2 and the start address of the transfer destination area in DES3. Align each transfer address to a specified data width (TW). The DSTC cannot execute an unaligned transfer.

Specify the transfer address update methods during a transfer in the SAC[2:0] bits and DAC[2:0] bits in DES0. The transfer address start method for SA and that for DA can be specified separately. Table 3-2 shows the methods of specifying transfer addresses.

Table 3-2 Specifying transfer addresses

Area name	bit	Details
DES0	SAC[2:0] DAC[2:0]	Select the respective update methods of transfer addresses SA and DA during a transfer. 000: The address is increased by 1 TW at every transfer. InnerReload is unavailable. 001: The address is increased by 1 TW at every transfer. InnerReload is available. 010: The address is increased by 2 TW at every transfer. InnerReload is unavailable. 011: The address is increased by 2 TW at every transfer. InnerReload is available. 100: The address is increased by 4 TW at every transfer. InnerReload is unavailable. 101: The transfer address remains unchanged during a transfer. 110: The address is decreased by 1 TW at every transfer. InnerReload is unavailable. 111: The address is decreased by 1 TW at every transfer. InnerReload is available.
DES2	SA[31:0]	Specify the start address for the transfer source area.
DES3	DA[31:0]	Specify the start address for the transfer destination area.

● Updating transfer number counter and transfer address

エラー! 参照元が見つかりません。 is an example showing how the transfer number counter and the transfer addresses change when the following settings are used: the number of outer loop transfers is 3, the number of inner loop transfers is 4 and the transfer address is increased. The vertical axis of the figure is the time axis, indicating transfer progress.

The upper part of the figure shows the operations of the transfer number counter. The inner loop counter remain value (IRM) downcounts at every transfer. The IRM reloads the IIN value after transfers of the number specified in the initial value (IIN) of the inner loop counter have been executed. The outer loop counter remain value (ORM) downcounts at the timing when the inner loop counter reloads. The DSTC counts the number of transfers using this dual loop counter.

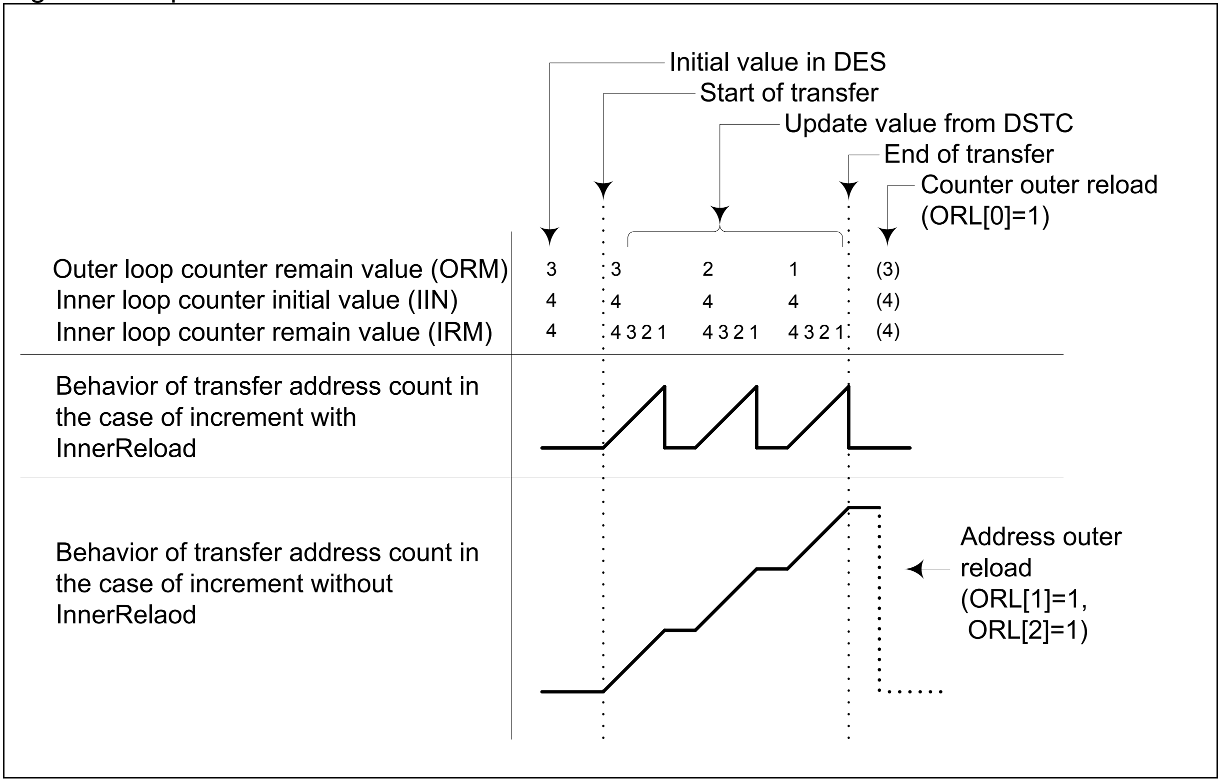
During a transfer, when the IRM reloads, it can be decided that whether the operation of returning a transfer address (SA/DA) to its initial value (InnerReload) is executed.

The middle part of the figure shows the update operation of the transfer address for increment with InnerReload. As shown in Figure 3-1, if transfer address increment and InnerReload are selected, after a transfer starts, the transfer address increases, and at the timing of IRM reload, is reset to the value at the start of the transfer.

The lower part of the figure shows the update operation of the transfer address for increment without InnerReload. The increment update of the transfer address continues for the timing of reloading the IRM.

Figure 3-1 illustrates also the operation of OuterReload. For its details, see the section on OuterReload.

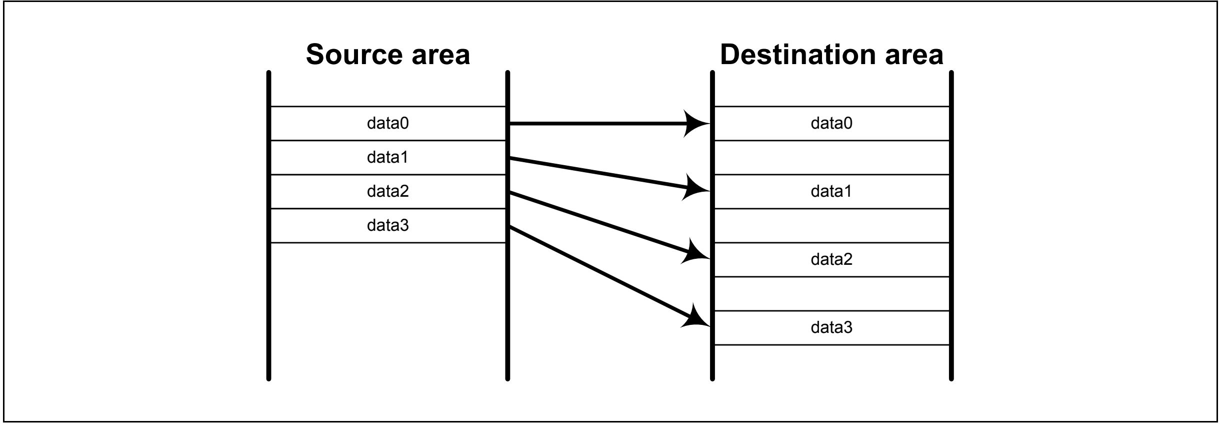
Figure 3-1 Operations of transfer number counter and transfer address


Initial value in DES
Start of transfer
Update value from DSTC
End of transfer
Counter outer reload
(ORL[0]=1)
Outer loop counter remain value (ORM)
Inner loop counter initial value (IIN)
Inner loop counter remain value (IRM)
3 3 2 1 (3)
4 4 4 4 (4)
4 4 3 2 1 4 3 2 1 4 3 2 1 (4)
Behavior of transfer address count in the case of increment with InnerReload
Behavior of transfer address count in the case of increment without InnerRelaod
Address outer reload
(ORL[1]=1,
ORL[2]=1)

● Stride transfer

If an increase of 2 TW is specified in the SAC[2:0] bits and an increase of 4 TW in the DAC[2:0] bits, a stride transfer is executed, in which at every transfer, the transfer address increases by 2 TW and by 4 TW in turn. Figure 3-2 shows an example of executing a transfer with SAC = 000 (increase by 1 TW) and DAC = 010 (increase by 2 TW). Using the stride transfer and the chain transfer together facilitates memory data rearrangement. For its details, see "0 ".

Figure 3-2 Stride transfer



■ Specifying OuterReload

● ORL[2:0]

Table 3-3 shows the method of specifying the transfer number counter and transfer address for OuterReload.

As shown in Figure 3-1, after IIN×ORM times of transfer have ended, for the next transfer, the transfer number counter (ORM/IRM/IIN) of DES1, the transfer source address (SA) of DES2 and the transfer destination address (DA) of DES3 can be reset (OuterReload) to their respective values at the start of the transfer.

Before starting a transfer, set DES4 to the same value as DES1, DES5 as DES2, and DES6 as DES3, respectively. After IIN×ORM times of transfer have ended, the values of DES4, DES5 and DES6 are copied to DES1, DES2 and DES3 respectively, and are reset to the values before the start of the transfer. In the case of using the same details in the next transfer, using OuterReload can eliminate the need of rebuilding the DES via the CPU.

Table 3-3 Specifying OuterReload

Area name	bit	Details
DES0	ORL[0]	Selects whether to execute OuterReload for the transfer number counter (DES1). 0: OuterReload for DES1 is not to be executed. DES4 area is not required. 1: OuterReload for DES1 is to be executed. DES4 area is not required.
	ORL[1]	Selects whether to execute OuterReload for the transfer source address (DES2). 0: OuterReload for DES2 is not to be executed. DES5 area is not required. 1: OuterReload for DES1 is to be executed. DES5 area is required.
	ORL[2]	Selects whether to execute OuterReload for the transfer destination address (DES3). 0: OuterReload for DES3 is not to be executed. DES6 area is not required. 1: OuterReload for DES3 is to be executed. DES6 area is required.

● DES size at using OuterReload

DES0 to DES3 are areas always required. DES4 to DES6 are areas required only when OuterReload is to be executed. The DES size and the addresses of DES4 to DES6 are defined according to the value of the ORL[2:0] bits as shown in Table 3-4. The respective relative addresses from the DESP of DES4 to DES6 vary according to the value of ORL[2:0]. An area not required is considered nonexistent.

Table 3-4 DES size and storage positions of DES4 to DES7 in OuterReload

ORL[2:0]	DES-SIZE (word)	DES4-address	DES5-address	DES6-address
000	4	No DES4	No DES5	No DES6
001	5	DESP+0x10	No DES5	No DES6
010	5	No DES4	DESP+0x10	No DES6
100	5	No DES4	No DES5	DESP+0x10
011	6	DESP+0x10	DESP+0x14	No DES6
101	6	DESP+0x10	No DES5	DESP+0x14
110	6	No DES4	DESP+0x10	DESP+0x14
111	7	DESP+0x10	DESP+0x14	DESP+0x18

● DES values after transfer end

The DES area can be saved when the OuterReload function is not used. If InnerReload is enabled for the transfer address, OuterReload does not need to be enabled for the transfer address because the values at the start of the transfer are stored in the DES. If neither OuterReload nor InnerReload is enabled, after a transfer has ended, depending on the settings, values stored in the DES may be different from those stored at the start of the transfer. In this situation, since the transfer cannot be started with the same DES values, rebuild the DES via the CPU. The following explains how the values of the DES area are updated after a transfer has ended.

The values of DES0 after the end of the transfer are the same as those at the start of the transfer, except those of the DV[1:0] bits and ST[1:0] bits. In addition, the values of DES4, DES5 and DES6 after the end of the transfer are the same as those at the start of the transfer.

The values of DES1 (transfer number counter) after the end of the transfer are updated according to the values of MODE and ORL[0] at the start of the transfer as shown in Table 3-5. "X" in Table 3-5 indicates that value has no effect on operation.

The value of IIN after the end of the transfer is the same as that at the start of the transfer. In the case of ORL[0] = 0, the value of ORM is updated to "0x0001" and the value of IRM to "0x01" when a transfer ends regardless of their values at the start of the transfer. However, if at the start of the transfer, the value of ORM is not "0x0001" or the value of IRM is not "0x01", the value of ORM or the value of IRM is different from what it was at the start of the transfer. Therefore, before restarting the transfer, rebuild the DES that makes the DSTC rewrite ORM and IRM with necessary values via the CPU. Table 3-5 summarizes conditions under which rebuilding DES1 becomes necessary.

Table 3-5 Values of DES1 at transfer end and necessity of rebuilding DES1

Values of DES0/DES1 at transfer start					Values of DES1 after transfer end			Necessity of rebuilding DES1
MODE	ORL[0]	ORM	IIN	IRM	ORM	IIN	IRM	
0	0	0x0001	X	-	0x0001	Value kept	-	Unnecessary
		Other than 0x0001	X	-				Necessary
1	0	0x0001	0x01	Same as IIN			0x01	Unnecessary
		X	Other than 0x01	Same as IIN				Necessary
		Other than 0x0001	X	Same as IIN				Necessary
0	1	X	X	-	Values of DES4 are copied.			Unnecessary
1	1			Same as IIN				Unnecessary

The values of DES2 (transfer source address) after the end of the transfer are updated according to the values of MODE, SAC[2:0] and ORL[1] of DES0 at the start of the transfer as shown in Table 3-6. "X" in Table 3-6 indicates that value has no effect on operation.

Table 3-6 Values of DES2 after transfer end

Values of DES0 at transfer start			Values of DES2 after transfer end
MODE	SAC[2:0]	ORL[1]	
0	xx0	0	Transfer source address at final outer loop start
0	xx1	0	Values at transfer start
1	xx0	0	Transfer source address of final transfer
1	xx1	0	Values at transfer start
X	X	1	Values of DES5 are copied.

The values of DES3 (transfer destination address) after the end of the transfer are updated according to the values of MODE, DAC[2:0] and ORL[2] of DES0 at the start of the transfer as shown in Table 3-7. "X" in Table 3-7 indicates that value has no effect on operation.

Table 3-7 Values of DES2 after transfer end

Values of DES0 at transfer start			Values of DES3 after transfer end
MODE	DAC[2:0]	ORL[2]	
0	xx0	0	Transfer destination address at final outer loop start
0	xx1	0	Values at transfer start
1	xx0	0	Transfer destination address of final transfer
1	xx1	0	Values at transfer start
X	X	1	Values of DES6 are copied.

■ Setting Chain Start and transfer end notification

● CHRS[5:0], CHLK

The DSTC executes transfers for the number of times specified in each DES (IIN times if MODE = 0, 1 time if MODE = 1) after receiving a Start Trigger. After executing transfers, the DSTC the next process according to the value of CHRS[5:0] in DES0. Table 3-8 shows the method of setting the Chain Start and the transfer end notification.

Table 3-8 Details of CHRS[5:0]

Area name	Name	Details
DES0	CHRS[5:4]	These bits select how the DSTC operates after the transfer number counter remain value becomes (ORM == 1) && (IRM == 1). 00: No interrupt flag is set. There is no Chain Start. The DSTC ends the transfer. 01: An interrupt flag is set. There is no Chain Start. The DSTC ends the transfer. 10: No interrupt flag is set. The DSTC executes a Chain Start on the next DES. 11: Setting prohibited (A DES open error occurs.)
	CHRS[3:2]	These bits select how the DSTC operates after the transfer number counter remain value becomes (ORM! = 1) && (IRM == 1). 00: No interrupt flag is set. There is no Chain Start. The DSTC waits for a Start Trigger. 01: An interrupt flag is set. There is no Chain Start. The DSTC waits for a Start Trigger. 10: No interrupt flag is set. The DSTC executes a Chain Start on the next DES. 11: No interrupt flag is set. The DSTC executes a Chain Start again on the current DES.
	CHRS[1:0]	If MODE is "1", these bits select how the DSTC operates after the transfer number counter remain value becomes (IRM! = 1). 00: No interrupt flag is set. There is no Chain Start. The DSTC waits for a Start Trigger. 01: An interrupt flag is set. There is no Chain Start. The DSTC waits for a Start Trigger. 10: No interrupt flag is set. The DSTC executes a Chain Start on the next DES. 11: No interrupt flag is set. The DSTC executes a Chain Start again on the current DES. If MODE is "0", the above settings are meaningless. Write "00" to CHRS[1:0] if MODE is "0". (Writing a value other than "00" to CHRS[1:0] if MODE is "0" causes a DES open error.)
	CHLK	This bit selects whether to execute the next transfer started by the Chain Start immediately after the current transfer (Chain Lock) or to enable other transfers to be executed before the next transfer started by the Chain Start. 0: After the current transfer, other transfers can be executed before the Chain Start transfer. 1: The Chain Start transfer is executed immediately after the current transfer.

If the next process is the Chain Start of the next DES, the DSTC starts transferring data according to the next DES. If the next process is executing the Chain Start again on the current DES, the DSTC starts transferring data according to the that DES again. If the next process does not involve the Chain Start, the DSTC ends the transfer (or waits for the next Start Trigger). The status of the transfer number counter determines which of CHRS[5:4], CHRS[3:2] and CHRS[1:0] the DSTC follows when executing the next process after the current DES.

In the case of not executing the Chain Start, after an interrupt flag is set, the DSTC can notify the CPU of the fact that the DSTC has ended the transfer (or is waiting for the next Start Trigger). An SW Start Trigger, and a Chain Start Trigger after that SW Start Trigger set the SWST bit to the SWTR register to "1". An HW Start Trigger, and a Chain Start Trigger after that HW Start Trigger set the HWINT[n] register to "1".

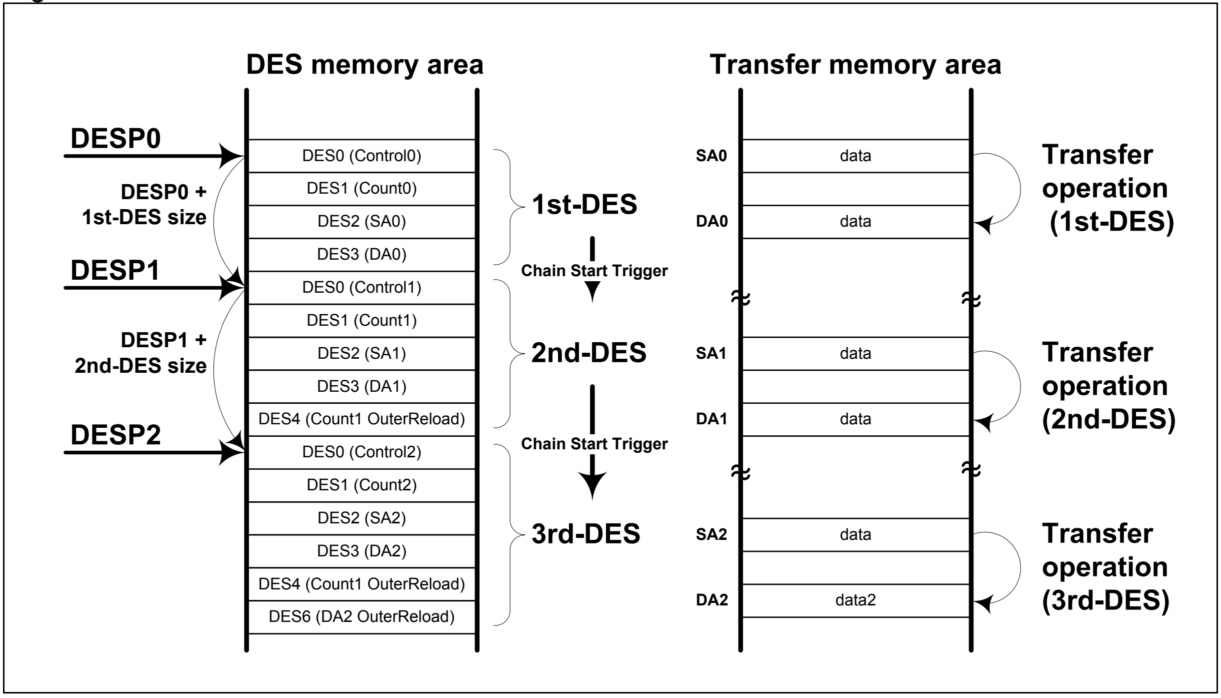
● Operations of Chain Start

Using the Chain Start enables making a Start Trigger for different transfers set in multiple DES.

Figure 3-3 illustrates how the DES reference and transfer operation are executed when the DSTC executes a Chain Start on the next DES. 1st-DES is located at the position of DESP0. The size of 1st-DES is defined by the value of ORL[2:0] of 1st-DES. If there is a Chain Start Trigger in the DES after 1st-DES, the DSTC computes DESP1 of the succeeding 2nd-DES from the DESP0 and the size of 1st-DES. The DSTC starts a transfer specified in 2nd-DES after referring to details of 2nd-DES. Therefore, the succeeding 2nd-DES to be started by the Chain Start must be located next to 1st-DES started first.

Specifying the Chain Start in 2nd-DES can start the transfer of 3rd-DES. The DSTC can keep executing the Chain Start transfer until there is no more Chain Start Trigger within the maximum number of DES that can be built.

Figure 3-3 DES reference in Chain Start transfer



● Chain Lock Function

The DSTC may execute other transfer first according to the transfer priority order if there is a transfer start request of other source at the timing of a Chain Start. If that occurs, the DSTC executes a specified Chain start transfer after executing other transfer. The Chain Lock Function ensures that in the situation described above, transfers started by a Chain Start are executed successively and that no other transfer request interrupts the transfers, regardless of the transfer priority order. The Chain Lock Function can be enabled by writing "1" to CHLK in a DES.

■ Other DES settings

● DV[1:0]

The DSTC refers to the DES area and updates it while executing a transfer operation. If the CPU updates a DES area that the DSTC is using, the DSTC may executes an illegal transfer operation, which is not set in a program. To prevent any illegal transfer operation, the DES area uses a mutually exclusive memory management system for the CPU and the DSTC. In DES0, there are the DV[1:0] bits (Descriptor Valid), indicating that the DES write update right ownership belongs to the CPU or the DSTC. Table 3-9 エラー! 参照元が見つかりません。 shows details of the DV and the related operations of the DSTC.

Table 3-9 Details of DV

Area name	Name	Details
DES0	DV[1:0]	DV specifies which of the CPU and the DSTC the ownership of the DES belongs to. DV also specifies whether a transfer is executed after the DES open process and whether the DES close process is to be executed. 00: The DES ownership belongs to the CPU. No transfer is executed after the DES open process. No DES close process is executed. (If reading this value, the DSTC notifies the system of a DES open error.) 01: The DES ownership belongs to the DSTC. A transfer is executed after the DES open process. The DES close process is executed. 10: The DES ownership belongs to the DSTC. No transfer is executed after the DES open process. The DES close process is executed. 11: The DES ownership belongs to the DSTC. A transfer is executed after the DES open process. No DES close process is executed.

That DV is "00" indicates that the CPU has the ownership of the DES area. That DV is "01", "10' or "11" indicates that the DSTC has the ownership of the DES area. After setting the initial values of a DES, the CPU sets DV to "01", "10" or "11" to notify that the ownership of that DES belongs to the DSTC. After receiving a Start Trigger, the DSTC checks the DV value of DES0 to determine subsequent operations. (The DSTC reading a DES upon a Start Trigger is called "DES open process".)

In the case of DV set to "01", the DSTC executes transfers for the number of times (ORM×IIN) specified. While the transfers are in progress, the value of DV remains "01" and the DSTC keeps the ownership of the DES. When all specified times of transfer end, the DSTV updates the value of DV to "00" and the DSTC returns the ownership of the DES to the CPU. (The process that the DSTV updates the value of DV to "00" and the DSTC returns the ownership of the DES to the CPU is called "DES close process".)

In the case of DV set to "11", the DSTC executes transfers for the number of times (ORM×IIN) specified. While the transfers are in progress, the value of DV remains "11" and the DSTC keeps the ownership of the DES. Even after all specified times of transfer have not ended, the DSTC does not execute the DES close process and keeps having the ownership of the DES.

In the case of DV set to "10", the DSTC does not execute a transfer, but executes only the DES close process.

In the case of DV set to "00", the DSTC recognizes that the DES area is being updated by the CPU. It does not execute any transfer or update the DES. The DSTC notifies the CPU of a DES open error.

The program determines the value of DV as explained below based on the way of using a DES.

If details of a transfer defined in a DES are subject to change, and the CPU has to update details of the DES at every transfer, setting DV to "01" grants the ownership of the DES to the DSTC. In this situation, after checking that the value of DV is "00" and that the ownership of the DES has been returned to the CPU, the CPU can safely update the DES. The CPU transfers the ownership of the DES to the DSTC again after completing the initialization of the DES.

If details of a transfer defined in a DES are fixed, and the CPU has to reuse details of the DES, setting DV to"11" grants the ownership of the DES to the DSTC. In this situation, since the DES close process is not executed after the transfer, the process of CPU transferring the ownership of the DES again can be omitted. After specified times (ORM×IIN) of transfer have ended, if a new Start Trigger is issued, transfers of the same details as the previous ones are started.

With DV set to "11", since the DSTC keeps having the ownership of the DES and does not return it to the CPU, the transfer becomes an infinite loop process. To escape from the infinite loop of transfer, set DV to "10" to return to CPU the ownership of the DES that has been transferred to the DSTC when DV was set to "11".

The CPU can update the DES if DV is set to "00". If DV is set to "01", "10" or "11", the DSTC cannot update details of the DES area other than DV while executing the transfer according to that DES.

If DV[1] is set to "1", the DSTC reuses the values of the DES after executing ORM×IIN times of transfer.

Therefore, specific restrictions on reloading the transfer counter and transfer address are added. If DV[1] is set to "1" and the settings of the DES make the values of DES1, DES2 and DES3 not return to their respective values, the DSTC notifies of the CPU a DES open error. (See the section on DES open error.)

● ST[1:0]

ST(Status)[1:0] of DES0 is for the DSTC to notify the CPU of the transfer end status. Table 3-10 shows data the DSTC writes to ST in a DES close process.

Table 3-10 Content of ST notification

Area name	Name	Details
DES0	ST[1:0]	After the transfer specified in a DES has ended, in a DES close process, the DSTC writes the end status value to ST. 00: The transfer has ended normally. 01: The transfer has ended abnormally because an error occurred at a transfer source access. 10: The transfer has ended abnormally because an error occurred at a transfer destination access. 11: The transfer has been ended compulsorily by a standby transition instruction, which is considered as abnormal transfer end.

If a transfer ends abnormally due to a DES access error or a DES open error, the DSTC does not execute a DES close process and does not write data to ST because the DSTC cannot access the DES area. Such error notifications are executed according to the MONERS Register of the DSTC but not ST of the DES.

● PCHK[3:0]

PCHK[3:0] (Parity Check) sets the parity (to be called "equation" below) of the DES0 area.

PCHK[3:0] != (DES0[27:24] ^ DES0[23:20] ^ DES0[19:16] ^ DES0[15:12] ^ DES0[11:8] ^ DES0[7:4])

If data in the DES area is corrupted by an event like a runaway of the CPU, the DSTC may start an unintended transfer. The parity check function is installed in DES0 to prevent the above from occurring. The CPU sets the parity of DES0 to PCHK when building the DES for the first time. The DSTC checks the consistency between the values of PCHK and those of DES0 in the DES open process. If a parity error occurs, the DSTC notifies the CPU of a DES open error and does not execute the transfer.

● ACK[1:0]

ACK[1:0] (Acknowledge) sets the value for adjusting the timing of DSTC outputting the DMA transfer request acknowledge signal to a peripheral device when the HW Start transfer is used.

If the HW Start transfer is used, set ACK to "01" for a DES to be directly started by the HW Start from a peripheral device. For other DES (the DES started by the Chain Start from the HW Start transfer, the DES used in the SW Start transfer, and the DES started by the Chain Start from the SW Start transfer), set ACK to "00".

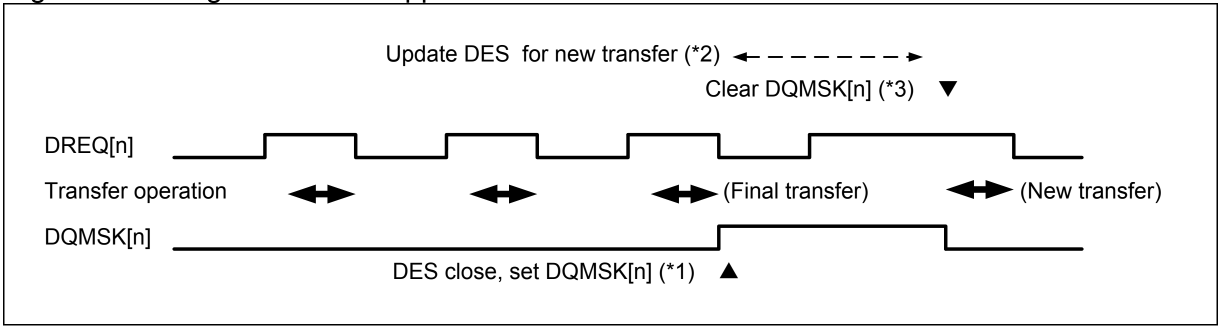
● DMSET

The DMSET (DMA request mask set) provides a function that sets the DQMSK[n] Register and masks a DMA transfer request signal from a peripheral during the period between the time at which the DSTC finishes the DES close process and the time at which the CPU finishes rebuilding the DES. For details, see "DQMSK[n] Register and DQMSKCLR[n] Register" in "3.2 Control Functions of DSTC".

If the HW Start transfer is used, the DMA transfer request signal (DREQ) from a peripheral is negated by the DMA transfer request acknowledge signal (DACK) after the transfer has ended. However, depending on peripherals, the DREQ is asserted at the following transfer request regardless of the status of the DSTC. If the DREQ is asserted during the period between the time at which the DSTC finishes the DES close process and the time at which the start of the next transfer is ready (rebuilding the DES), the DSTC notifies the CPU of a DES open error because the start of the next transfer is not ready. In this situation, setting the DMSET bit in Descriptor 0 to "1" can prevent the DSTC from notifying the CPU of a DES open error, and can suppress the start of an HW Start transfer until the completion of rebuilding a DES.

Figure 3-4 shows an operation example. For a DES to be processed by the HW Start transfer directly from a peripheral, and a DES to be started by the Chain Start from the DES mentioned before, with DMSET set to "1", if the DES close process is not executed, the channel bit corresponding to that DES in the DQMSK[n] Register is set to "1". (*1 in Figure 3-4) After that channel bit in the DQMSK register has been set to "1", the DSTC cannot recognize the DREQ, and cannot notify the CPU of the DES open error either. After the CPU has rebuilt the DES (*2 in Figure 3-4) and the next transfer is ready, the CPU clears the channel bit corresponding to that DES in the DQMSK[n] Register. (*3 in Figure 3-4) After the CPU has cleared the channel bit in the DQMSK[n] Register, the DSTC recognizes the DREQ signal. A new transfer is started according to the DES rebuilt by the CPU.

Figure 3-4 Using DMSET to suppress transfer start



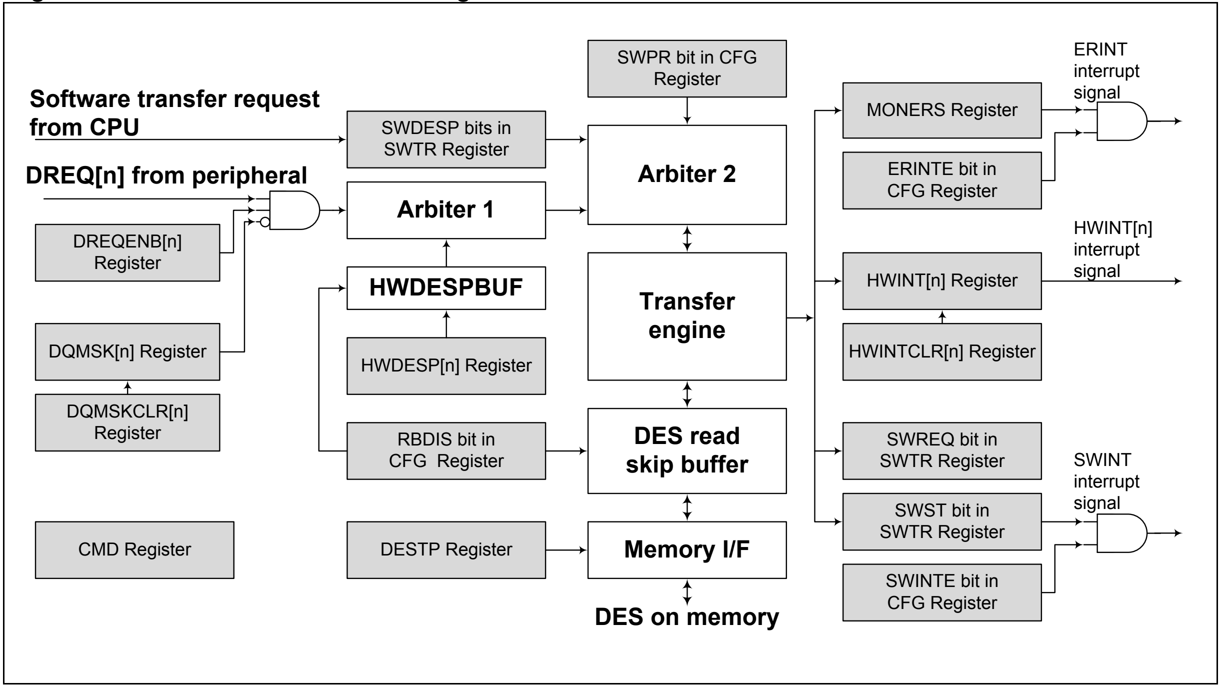
3.2. Control Functions of DSTC

This section explains the control functions of the DSTC.

■ DSTC internal block diagram

Figure 3-5 illustrates the connection between control blocks and control registers (shaded rectangles) in the DSTC that can be accessed from the CPU. The CPU starts DSTC transfer and controls end notifications via accesses to control registers. The following sections explain the operations of each block and the function overview of each register shown in the DSTC internal block diagram. For details of register functions, see "5 Registers and Descriptors of DSTC".

Figure 3-5 DSTC internal block diagram



■ DESTP Register

The DESTP (DES top address) Register is a register specifying the start address of the DES area on the memory. Specify the start address when doing the initial settings. The DSTC refers to the DES located at the address of "DESTP + DESP" and executes a transfer.

■ Control of SW Start transfer

To issue a Start Trigger of the SW Start transfer, write the DESP value of the DES to be started to SWDESP (Software DES pointer) in the SWTR (Software trigger) Register. If a Chain Start is executed during the SW Start transfer, SWDESP is updated by the DSTC to the value of DESP used in the Chain Start. The value of SWDESP is sent to Arbiter 2 in Figure 3-5 as a transfer request.

The SWREQ (Software request) bit in the SWTR Register is a read-only bit indicating whether the execution of the SW Start transfer is pending, or the SW Start transfer as well as the Chain Start transfer are being executed. A write access (Start Trigger) to the SWTR Register sets the SWREQ bit to "1". If the SW Start transfer ends normally, abnormally, or is waiting for a Start Trigger, SWREQ is reset to "0".

A SW Start transfer trigger can be issued only after the current SW Transfer has ended. If the SWREQ bit is "1", a write access to the SWTR Register is ignored.

The SWST (Software status) bit in the SWTR Register is a read-only bit for sending the SW transfer end notification to the CPU. The interrupt flag set is specified in the DES started by the SW Start, or CHRS in the DES started by the Chain Start after the DES started by the SW Start. If the SW transfer ends normally, SWDESP is set to "1". SWDESP can be cleared to "0" by sending the SWCLR command to the CMD

Register.

If the SWST bit has been set to "1", the SWINT interrupt can be enabled by writing "1" to the SWINTE bit in the CFG Register. In the case of (SWINTE==1)&(SWST==1), the SWINT interrupt signal for the NVIC is asserted.

■ Control of HW Start transfer flow

If a peripheral makes a transfer request (assertion of DREQ[n]), the DSTC starts the HW Start transfer. The DSTC controls the HW Start transfer on a transfer channel using the following registers whose number corresponds to the number of transfer channels. The CPU does the initial settings of those registers before a peripheral makes a transfer request. In addition, the CPU clears registers according to the progress of a transfer.

● DREQENB[n] Register

The DREQENB[n] (DMA request enable) Register determines whether HW channel n is used in the initial settings. Write "1" to the DREQENB[n] Register to use HW channel n. Write "0" to the DREQENB[n] Register to not use HW channel n. If the DREQENB[n] Register is "0", the interrupt signal (DREQ[n]) of a peripheral connected to the DSTC is ignored. The value of the DREQENB[n] Register cannot be modified by the DSTC.

The value of the DREQENB[n] Register determines which of the interrupt signal from a peripheral and HWINT[n] from the DSTC is selected as an interrupt signal connected to the NVIC. For its details, see "2 DSTC Operations Overview and DSTC System Configuration".

● DQMSK[n] Register and DQMSKCLR[n] Register

The DQMSK[n] (DMA request mask) Register is a read-only register. That this register is "1" indicates the HW Start transfer request (DREQ[n]) to the DSTC is being suppressed. If one of the following conditions is met, the DSTC sets DQMSK[n] to "1" and suppresses the transfer request of the HW channel corresponding to DQMSK[n].

- A transfer error has occurred at a transfer on HW channel n.
- The CPU has issued a standby transition command to the CMD Register.
- DMSET in the DES for the transfer on HW channel n is "1" and the DSTC has executed a DES close process.

After the CPU has rebuilt the DES and the HW transfer has become ready to start, the suppression of the HW Start transfer request to the DSTC can be released by the CPU. If "1" is written to the DQMSKCLR[n] (DMA request mask clear) Register, the DQMSK[n] Register is cleared to "0" and the succeeding HW transfer request (DREQ[n]) is recognized.

● HWDESP[n] Register

The HWDESP[n] (Hardware DES pointer) Register sets the DESP of the DES that the DSTC refers to and executes at a transfer request of HW channel n. Set this register before making an HW transfer request.

If an HW Start trigger is issued, the DSTC starts a transfer referring to the DES of the DESP set in the HWDESP[n] Register. The DSTC stores the DESP value of the HWDESP[n] Register in HWDESPBUF in Figure 3-5 before using it. In a Chain Start, the value stored in HWDESPBUF is updated to the DESP value set after the Chain Start. The value of the HWDESP[n] Register cannot be modified by the DSTC.

If HW Start requests of channel n are made successively, the DSTC uses the DESP value stored in HWDESPBUF, but not the DESP value of the HWDESP[n] Register. Therefore, if the values of the HWDESP[n] Register are modified via the CPU, invalidate the value stored in HWDESPBUF. The DESP value of HWDESPBUF can be invalidated by modifying the value of the RBDIS bit in the CFG Register. For its details, see "5.5 CFG Register".

● HWINT[n] Register and HWINTCLR[n] Register

The HWINT[n] (Hardware transfer interrupt) Register is a read-only register for sending the HW transfer end notification to the CPU. The interrupt flag set is specified in the DES started by the HW Start, or CHRS in the DES started by the Chain Start after the DES started by the HW Start. If the HW transfer ends normally, HWDESP is set to "1". The HWINT[n] Register can be cleared to "0" by writing "1" to the HWINTCLR[n]

Register. If the HWINT[n] Register is set to "1", the interrupt signal for the NVIC is asserted.

■ Arbitration of transfer requests

The DSTC arbitrates start triggers if multiple HW Start requests conflict with an SW Start request, and executes transfers sequentially. The arbitration of start requests are processed by two blocks, Arbiter 1 and Arbiter 2, shown in Figure 3-5. Below are details of arbitration.

● Arbiter 1

The HW transfer request is arbitrated by Arbiter 1. If there are conflicting requests, Arbiter 1 uses the rotation method explained below to select a transfer start channel. After a bus reset, the smaller the channel number, the higher the priority is in the selection priority order.

highest priority 0,1,2,3,4,5,6,7,,,,,,254,255 lowest priority

According to this priority order, for instance, if a request from channel 5 and another from channel 6 are made simultaneously, channel 5 is selected. Once a transfer channel is selected, its priority is rotated to the lowest. In the above example, as channel 5 is selected, the priority order is updated to the one below.

highest priority 6,7,8,9,10,11,,,,,,254,255, 0,1,2,3,4,5 lowest priority

According to this priority order, for instance, if a request from channel 5 and another from channel 6 are made simultaneously, channel 6 is selected. The rotation method enables multiple HW transfer requests to be processed equally.

Arbiter 1 refers to HWDESP[n] of channel n selected and notifies Arbiter 2 of the DESP of the DES used. In addition, if the Chain Start is used in the HW transfer, Arbiter 1 notifies Arbiter 2 of the updated DESP. After all Chain transfers have ended and the DSTC has started to wait for the next Start Trigger, Arbiter 1 notifies Arbiter 2 of the transfer request of the channel n selected.

● Arbiter 2

Arbiter 2 selects which of the HW transfer request selected by Arbiter 1 and the SW transfer request is to be executed. If there are conflicting transfer requests, the DSTC selects a transfer request according to the probability set in the SWPR (Software transfer priority) bits in the CFG Register and starts the transfer engine. Table 3-11 shows the settings of the SWPR bits in the CFG Register and the probability of the SW transfer acquiring the transfer right.

Table 3-11 Details of CFG:SWPR[2:0]

Area name	Name	Details
CFG	SWPR[2:0]	In the arbitration of Arbiter 2, if the SW transfer request conflicts with the HW transfer request, Arbiter 2 specifies the probability of the SW transfer acquiring the transfer right. 00: Sets the priority of the SW transfer to the highest priority. (If an SW transfer request is made while an HW transfer is in progress, the SW transfer starts after the HW transfer has ended.) 001: Sets the probability of the SW transfer acquiring the transfer right to 1/2. 010: Sets the probability of the SW transfer acquiring the transfer right to 1/3. 011: Sets the probability of the SW transfer acquiring the transfer right to 1/7. 100: Sets the probability of the SW transfer acquiring the transfer right to 1/15. (Initial value) 101: Sets the probability of the SW transfer acquiring the transfer right to 1/31. 110: Sets the probability of the SW transfer acquiring the transfer right to 1/63. 111: Sets the priority of the SW transfer to the lowest priority. (The SW transfer starts only when there is no HW transfer request.)

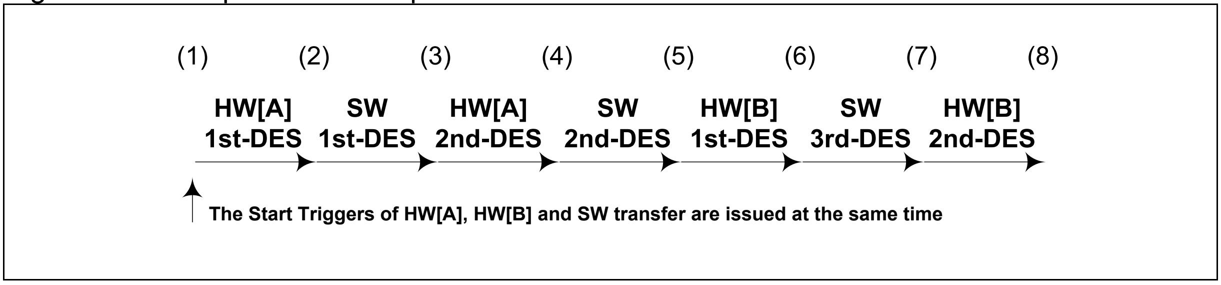
● Example of arbiter operation

Figure 3-1 shows an operation example. The horizontal axis in the figure is the time axis. The figure illustrates the order of granting the transfer right to different transfer requests during the arbitration by the DSTC. There are three transfer sources: HW channel A transfer, HW channel B transfer and SW transfer. The

HW[A] transfer and the HW[B] transfer are connected in a Chain transfer by two DES; SW transfers are connected in a Chain transfer by three DES. The SWPR bits in the CFG Register are set to "001" (probability of SW transfer: 1/2). No Chain lock is specified in any DES.

At timing (1), the respective transfer requests of HW[A] transfer, HW[B] transfer and SW transfer are made simultaneously. Arbiter 1 arbitrates the conflict between HW[A] transfer and HW[B] transfer. Arbiter 1 selects one from HW[A] transfer and HW[B] transfer according to the preceding rotation status. The following description assumes that Arbiter 2 has selected HW[A] transfer. Arbiter 2 arbitrates the conflict between HW[A] transfer and SW transfer. Arbiter 2 selects one from HW[A] transfer and SW transfer according to the preceding rotation status. The following description assumes that Arbiter 2 has selected HW[A] transfer. Arbiter 2 starts the transfer of 1st-DES of HW[A].

Figure 3-6 Example of arbiter operation



At timing (2), the transfer of HW[A] 1st-DES ends. A Chain Start request of HW[A] 2nd-DES is made. (The request of HW[B] is held until there is no more Chain Start of HW[A].) Arbiter 1 requests Arbiter 2 for 2nd-DESP of HW[A]. Arbiter 2 arbitrates the conflict between HW[A] transfer and SW transfer. As the probability for SW transfer is 1/2, and Arbiter 2 has selected HW[A] transfer at timing (1), Arbiter 2 selects SW 1st-DES transfer.

At timing (3), the transfer of SW 1st-DES ends and the Chain Start request of SW 2nd-DES is made. Arbiter 2 arbitrates the conflict between HW[A] 2nd-DES transfer and SW-2ndDES transfer. As the probability for SW transfer is 1/2, and Arbiter 2 has selected SW transfer at timing (2), Arbiter 2 selects HW[A] 2nd-DES transfer.

At timing (4), the transfer of HW[A]-2ndDES ends. Arbiter 1 makes a request for transferring HW[B] 1st-DES to Arbiter 2. Arbiter 2 arbitrates the conflict between HW[B] 1st-DES transfer and SW 2nd-DES transfer, and then selects SW 2nd-DES transfer.

At timings (5), (6) and (7), Arbiter 2 executes the same arbitration operations and selects HW[B] 2nd-DES transfer and SW 3rd-DES transfer.

As explained above, SW transfer may be executed during the Chain transfer of HW transfer, and HW transfer during the Chain transfer of SW transfer. During the Chain transfer of HW transfer, no HW transfer on any other channel is executed. If the Chain lock has been specified in the DES, regardless of the setting of the SWPR bits in the CFG Register, after the transfer of that DES has been executed, the transfers of the DES in the Chain Start are always executed successively.

In the above example, as the probability is set to 1/2 in the SWPR bits in the CFG Register, one SW transfer is executed in every two transfers. Taking account of the number of HW transfer channels of the DSTC to be used simultaneously, the number of Chains in the DES, the transfer data size in each transfer, etc., select an appropriate value for the SWPR bits in the CFG Register. The value of the SWPR bits in the CFG Register can be modified even when the DSTC is executing a transfer. After the value of the SWPR bits in the CFG Register has been modified, it is applied from the next SW Start Trigger.

■ Read skip buffer function

The transfer engine of the DSTC refers to the transfer information of the DES on the memory while executing a transfer. If all transfers do not end in one Start Trigger, the DSTC writes back to each DES the number of executed transfers of a DES and transfer addresses. If a transfer address is fixed or does not need to be updated, the DSTC skips the write-back process. At the next Start Trigger, the DSTC continues executing transfers according to updated DES information.

Since it takes time for the DSTC to refer to the DES at every Start Trigger, the DSTC has the read skip buffer function as shown in Figure 3-5. The DSTC stores in its internal read skip buffer the transfer information in the DES that the DSTC has read. If the next Start Trigger refers to the same DESP as the current Start Trigger, the DSTC does not refer to the DES on the memory, but uses the values in the read skip buffer to execute the transfer to increase the processing speed.

The read skip buffer function can be enabled and disabled by using the RBDIS bit in the CFG Register. In practice, to increase the transfer speed, enable the read skip buffer function. If "1" is written to the RBDIS bit in the CFG Register to disable the read skip buffer function, always refer to the DES directly on the memory.

In the case of stopping the transfer operation of the DSTC (infinite loop out) by modifying the value of DV in DES0 via the CPU to "10", after modifying the value of DV in DES0, invalidate the DES information stored in the read skip buffer by writing "0" to the RBDIS bit in the CFG Register. As long as the read skip buffer function remains enabled, the DSTC may skip referring to the DES and not be able to recognize any change in the value of DV by the CPU. For its details, see "5.5 CFG Register".

■ Transfer end

If the DSTC ends a transfer normally, according to details of the DES, it executes the Chain Start, sets the interrupt flag (SWTRS:SWST or HWINT[n]) and the DES close process.

If a transfer error occurs, a transfer is interrupted immediately. This interruption of a transfer is called error end. In an error end, the DSTC does not execute the Chain Start. In addition, the DSTC does not set the interrupt flag (SWTR:SWST or HWINT[n]), but records details related to the occurrence of the error in the MONERS Register instead. Details of the error determine whether the DSTC executes the DES close process.

In an HW Start transfer, if a transfer error occurs, the DSTC sets the DQMSK[n] Register corresponding to the channel on which that transfer error has occurred, and suppresses future HW transfer requests on that channel.

Since details of the DES in which an error has occurred remain the same as they were during the transfer, before starting a new transfer, rebuild the DES area with the CPU.

■ MONERS Register

If a transfer error occurs, details of that error are recorded in the MONERS Register. Table 3-12 shows details the MONERS Register displays.

Table 3-12 Details of MONERS

Area name	Name	Details
MONERS	EST[2:0]	Indicate details of an error that has occurred. 000: No error has occurred. 001: Source access error 010: Destination access error 011: A transfer has been stopped compulsorily. 100: DES access error 101: DES open error Value other than the above: Undefined
	DER	This bit indicates whether a double error has occurred. 0: Indicates that no double error has occurred. 1: Indicates that a double error has occurred.
	ESTOP	Indicates whether the DSTC is in the error stop state. 0: Indicates that the DSTC is not in the error stop state. 1: Indicates that the DSTC is in the error stop state.
	EHS	Indicates whether the DES that has caused an error has been started by the HW Start or by the SW Start. 0: An error has occurred in a transfer started by the SW Start or by the Chain Start in that SW Start. 1: An error has occurred in a transfer started by the HW Start or by the Chain Start in that HW Start.

Area name	Name	Details
MONERS	ECH	Indicates the HW channel number if the DES that has caused an error has been started by the HW Start.
	EDESP	Indicates the DESP of the DES that has caused an error.

Details of a transfer error that has occurred can be checked by referring to the MONERS Register. With the EST[2:0] bits in the MONERS Register indicating that an error has occurred, the ERINT interrupt can be enabled by writing "1" to the ERINTE bit in the CFG Register. If the ERINTE bit is set to "1", the ERINT interrupt signal for the NVIC is asserted. The values of the MONERS Register and the ERINT interrupt can be cleared by issuing an ERCLR command to the CMD Register. Details of errors that may occur are explained below.

● DES access error

If one of the following events occurs while the DSTC is referring to the DES area of DESTP+DESP, the DSTC ends a transfer in the form of error end (DES access error). The DSTC sets MONERS:EST to "100". At a DES open error, the DSTC does not execute the DES close process.

- The DES area address value calculated overflows (out of the range of 0x00000000-0xFFFFFFFF).
- The DSTC receives a bus error acknowledge from the system when accessing a DES area.

● DES open error

After the DSTC has referred to the area of DESTP+DESP, if the value of DES0 or DES1 meets one of the following conditions, the DSTC regards that as an abnormal DES specified value, and ends a transfer in the form of error end (DES open error). The DSTC sets MONERS:EST to "101". At a DES open error, the DSTC does not execute the DES close process.

- DV[1:0]==00 (No DES ownership)
- PCHK[3:0] != (DES0[27:24] ^ DES0[23:20] ^ DES0[19:16] ^ DES0[15:12] ^ DES0[11:8] ^ DES0[7:4]) (DES0 parity error)
- One of the two bits of the reserved area of DES0 is "1". (abnormal specified value)
- TW[1:0]==11 (abnormal specified value)
- CHRS[5:4]==11 (abnormal specified value)
- (CHRS[5]==0) &&(CHRS[3]==0)&&(CHRS[1]==0) &&(CHLK ==1) (abnormal Chain setting)
- (MODE==0) && (CHRS[1:0] != 00) (abnormal setting)
- (MODE==0)&&(ORM==0x0000) && (IIN≥0x2000) (Out of allowed count value range in mode 0)
- (MODE==0)&&(ORM≥0x8000) && (IIN≥0x4000) (Out of allowed count value range in mode 0)
- (MODE==0)&&(ORM≥0x4000) && (IIN≥0x8000) (Out of allowed count value range in mode 0)
- (MODE==0)&&(ORM≥0x2000) &&(IIN==0x0000) (Out of allowed count value range in mode 0)
- (MODE==1)&&(IIN!=0x00)&&(IRM==0x00) (Out of allowed count value range in mode 1)
- (MODE==1)&&(IIN!=0x00)&&(IRM>IIN) (Out of allowed count value range in mode 1)
- (MODE==0)&&(DV[1]==1)&&(ORL[0]==0)&& (ORM != 0x0001) (abnormal counter reload setting)
- (MODE==1)&&(DV[1]==1)&&(ORL[0]==0)&& (ORM != 0x0001) (abnormal counter reload setting)
- (MODE==1)&&(DV[1]==1)&&(ORL[0]==0) && (IRM != 0x01) (abnormal counter reload setting)
- (MODE==1)&&(DV[1]==1)&&(ORL[0]==0)&& (IIN != 0x01) (abnormal counter reload setting)
- (DV[1]==1)&&(SAC[0]==0)&&(ORL[1]==0) (abnormal transfer source address reload setting)
- (DV[1]==1)&&(DAC[0]==0)&&(ORL[2]==0) (abnormal transfer destination address reload setting)

● Source access error

If one of the following events occurs while the DSTC is accessing the transfer source address area, the DSTC ends a transfer in the form of error end (source access error). The DSTC sets MONERS:EST to "001". At the same time, the DSTC writes "01" to DES0:ST and executes the DES close process.

- The specified transfer source start address value (SA) is unaligned to TW.
- The transfer source address value having undergone increment calculation or decrement calculation overflows.
- The DSTC receives a bus error acknowledge from the system.

● Destination access error

If one of the following events occurs while the DSTC is accessing the transfer destination address area, the DSTC ends a transfer in the form of error end (destination access error). The DSTC sets MONERS:EST to "010". At the same time, the DSTC writes "10" to DES0:ST and executes the DES close process.

- The specified transfer destination start address value (DA) is unaligned to TW.
- The transfer destination address value having undergone increment calculation or decrement calculation overflows.
- The DSTC receives a bus error acknowledge from the system.

● Transfer compulsory stop error

If the DSTC receives a standby transition command from the CPU while executing a transfer, it ends the transfer (transfer compulsory stop error). The DSTC sets MONERS:EST to "011". At the same time, the DSTC writes "11" to DES0:ST and executes the DES close process.

● DER function and ESTOP function

If a transfer error occurs, the transfer of the DES that has caused the transfer error is interrupted and ended. After the transfer has been ended, if there is a transfer start request for another DES, the setting of the ESTE (error stop enable) bit in the CFG Register determines whether the DSTC starts the transfer requested in the transfer start request.

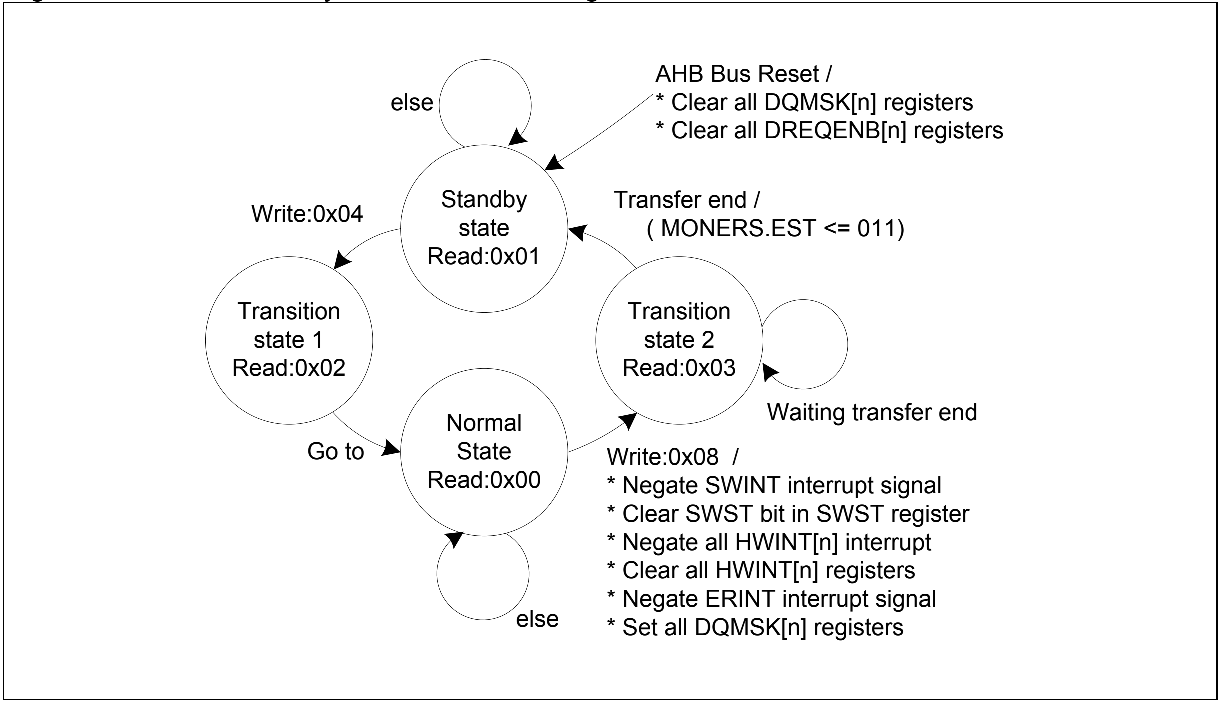
In the case of CFG:ESTE = 0, if there is a new transfer request after a transfer error has occurred, the DSTC starts the transfer for that new transfer request. The MONERS Register records error information and keeps it until the register is cleared by the CPU. While the MONERS Register is keeping error information (EST ≠ 000), if a transfer caused by another transfer request ends due to an error, the DSTC sets the DER (double error) bit to "1". The DER bit is a bit that indicates a double error has occurred. As for the second error, the DSTC notifies the CPU of only its occurrence. The MONERS Register keeps only the information of the first error, but does not keep details of the second error. Moreover, the MONERS Register does not record any error that occurs after the second error.

With CFG:ESTE set to "1", if a transfer error occurs, if a transfer error occurs, the DSTC transits to the error stop state. After transiting to the error stop state, the DSTC holds other transfer requests and no longer starts any transfer. That the ESTOP bit in the MONERS Register is set to "1" indicates that the DSTC is in the error stop state. If the CPU issues an ERCLR command to the CMD Register, the DSTC is released from the error stop state and starts transfers according to transfer requests it has been keeping.

■ Standby function

To reduce power consumption, the DSTC has a function (standby function) for stopping the internal clocks of the DSTC to make the DSTC stop operating. The state of the DSTC can be switched by the standby transition command and standby release command issued to the CMD Register. エラー! 参照元が見つかりません。 illustrates the operations executed in the issue of the standby transition command and in the state transition of the DSTC.

Figure 3-7 DSTC standby state transition diagram


else
AHB Bus Reset /
* Clear all DQMSK[n] registers
* Clear all DREQENB[n] registers
Standby
state
Read:0x01
Write:0x04
Transfer end /
(MONERS.EST <= 011)
Transition
state 1
Read:0x02
Transition
state 2
Read:0x03
Waiting transfer end
Normal
State
Read:0x00
Go to
Write:0x08 /
* Negate SWINT interrupt signal
* Clear SWST bit in SWST register
* Negate all HWINT[n] interrupt
* Clear all HWINT[n] registers
* Negate ERINT interrupt signal
* Set all DQMSK[n] registers
else

The DSTC has four states: standby state, transition state 1, normal state and transition state 2. The state of the DSTC can be checked by reading the value of the CMD Register via the CPU.

Upon a bus reset, the initial state of the DSTC is the standby state. If the CPU issues a standby release command (writing "0x04") to the CMD Register, the DSTC transits to the transition state and then to the normal state.

In the normal state, if the CPU issues a standby transition command (writing "0x08") to the CMD Register, the DSTC transits to the transition state 2 to wait for a transfer to end.

In the transition state 2, if the DSTC does not execute any transfer, it immediately transits to the standby state. But, if the DSTC executes a transfer, it transits to the standby state after that transfer has been compulsorily ended.

If a transfer is ended by issuing a standby transition command, the DSTC writes "11" (compulsory end code) to ST of the DES for that transfer to execute the DES close process. In addition, the EST bits in the MONERS Register are set to "011". If the DSTC receives both HW transfer and SW transfer, it executes the DES close processes for both transfers.

In addition, if the CPU issues a standby transition command to the DSTC, the DSTC executes the following processes at the same time.

- The DSTC negates the SWINT interrupt signal and clears SWTR:SWST.
- The DSTC clears all HWINT[n] Registers and negates all HWINT[n] interrupt signals.
- Negates the ERINT interrupt signal.
- The DSTC sets all DQMSK[n] Registers to suppress the HW transfer request.

Though the issue of a standby transition command negates the ERINT interrupt signal, the values of the MONERS Register remain unchanged. Therefore, if a transfer has been compulsorily stopped by a standby transition command, the information of that transfer can still be checked by reading the MONERS Register. In addition, the error record in the MONERS Register can be cleared only when the DSTC is in the normal state. After a standby release command has made the DSTC return to the normal state, clear the MONERS Register with the ERCLR command.

The initial values of all bits in the DQMSK[n] Register after a bus reset are "0". If a standby transition command is issued, all bits in the DQMSK[n] are set to "1". To start an HW transfer after the DSTC has returned to the normal state, clear the DQMSK[n] to be used for that HW transfer after finishing the setup of a peripheral and rebuilding the DES.

Table 3-13 shows the accessibility of each control register in each state of the DSTC. "O" indicates that register is accessible. "-" indicates that the access to that register is ignored by the DSTC and does not function. "X" indicates that the process result becomes undefined depending on the change in the state of the DSTC. It is prohibited to execute an access marked with "X".

In the standby state, transition state 1 and transition state 2, writing a value to SWTR:SWDESP cannot start a new SW Start transfer (the write access to the SWTR Register is ignored).

Table 3-13 Accessibility of each control register in each state of the DSTC

Register name	Register access	Stand-by State	Normal State	Transition State 1,2
CMD Register	CMD Register read	O	O	O
	Standby release command (write)	O	-	-
	Standby transition command (write)	-	O	-
	SWCLR / ERCLR / MKCLR Command (write)	-	O	X
HWDESP[n] Register	Read access	-	O	X
	Write access	-	O	X
Other control registers	Read access	O	O	O
	Write access	-	O	X

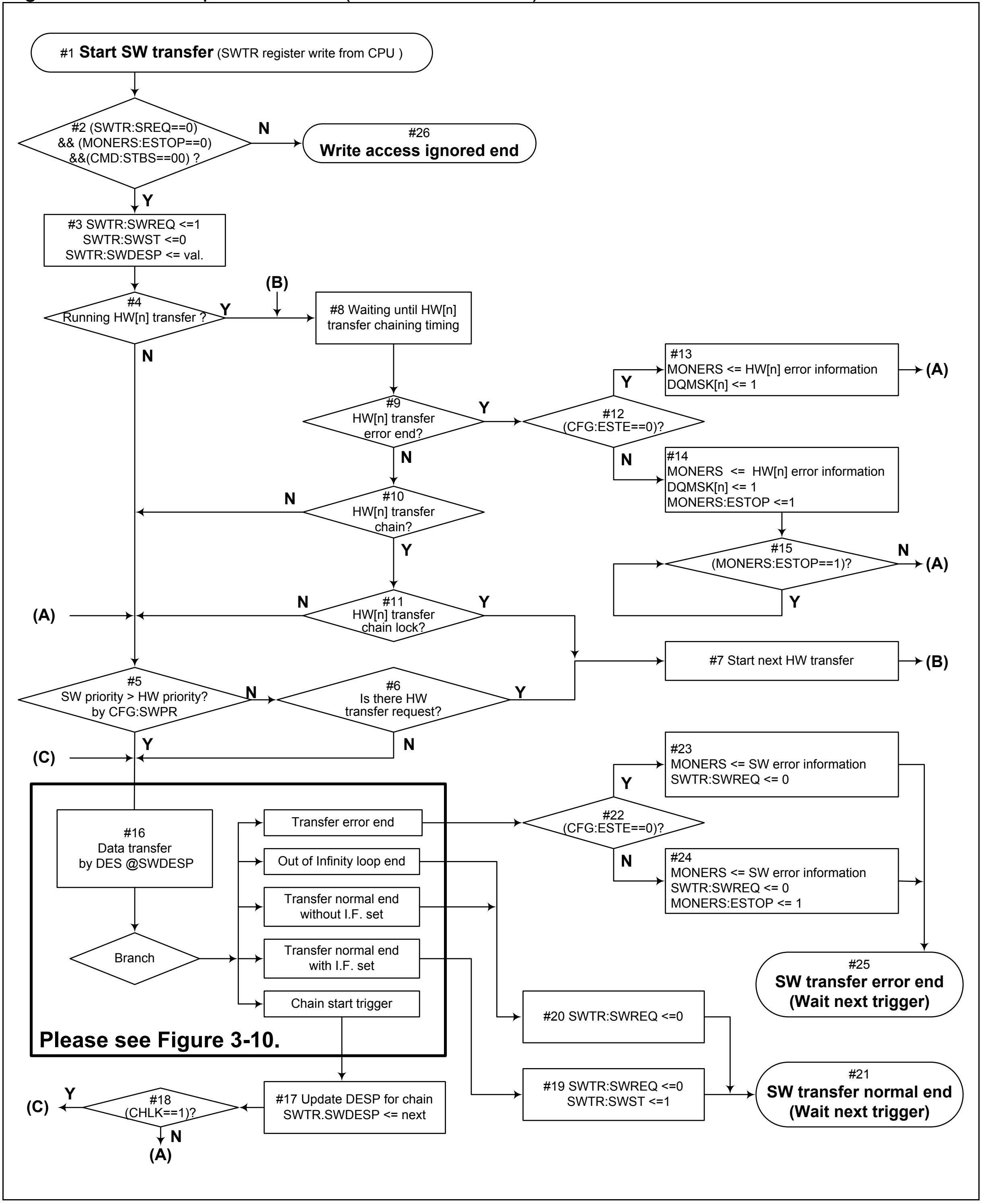
3.3. Operation Flows of DSTC

This section explains the operation of the DSTC with flow charts.

■ SW Start transfer flow

The operations the DSTC executes after receiving an SW Start Trigger from the CPU are explained below. Figure 3-8 shows a flow chart of the operations of the DSTC. Numbers in the figure correspond to those used in the explanation after the figure.

Figure 3-8 DSTC operation flow (SW Start transfer)


#1 Start SW transfer (SWTR register write from CPU)
#2 (SWTR:SREQ==0) && (MONERS:ESTOP==0) &&(CMD:STBS==00) ?
N
#26 Write access ignored end
Y
#3 SWTR:SWREQ <=1 SWTR:SWST <=0 SWTR:SWDESP <= val.
(B)
#4 Running HW[n] transfer ?
Y
#8 Waiting until HW[n] transfer chaining timing
N
#13 MONERS <= HW[n] error information DQMSK[n] <= 1
(A)
Y
#9 HW[n] transfer error end?
Y
#12 (CFG:ESTE==0)?
N
N
#14 MONERS <= HW[n] error information DQMSK[n] <= 1 MONERS:ESTOP <=1
N
#10 HW[n] transfer chain?
Y
#15 (MONERS:ESTOP==1)?
N
(A)
Y
(A)
N
#11 HW[n] transfer chain lock?
Y
#7 Start next HW transfer
(B)
#5 SW priority > HW priority? by CFG:SWPR
N
#6 Is there HW transfer request?
Y
(C)
Y
N
#23 MONERS <= SW error information SWTR:SWREQ <= 0
Y
#16 Data transfer by DES @SWDESP
Transfer error end
#22 (CFG:ESTE==0)?
Out of Infinity loop end
N
#24 MONERS <= SW error information SWTR:SWREQ <= 0 MONERS:ESTOP <= 1
Transfer normal end without I.F. set
Branch
Transfer normal end with I.F. set
#25 SW transfer error end (Wait next trigger)
Chain start trigger
#20 SWTR:SWREQ <=0
Please see Figure 3-10.
(C)
Y
#18 (CHLK==1)?
#17 Update DESP for chain SWTR.SWDESP <= next
#19 SWTR:SWREQ <=0 SWTR:SWST <=1
#21 SW transfer normal end (Wait next trigger)
N
(A)

#1 Start the SW Start transfer from a write access to the SWTR Register from the CPU.

#2 If the SWTR Register, the MONERS Register and the CMD Register are (SWTR:SWREQ==0)&&(MONERS:ESTOP==0)&&(CMD==00), the DSTC proceeds to #3. Otherwise the DSTC proceeds to #26.

#3 Write "1" to SWTR:SWREQ and clear SWTR:SWST to "0". Store a specified value in SWTR:SWDESP.

Processes explained in #4 to #15 are details of operations of Arbiter 2 and processes in another HW Start transfer.

#4 If another HW Start transfer is being executed, the DSTC proceeds to #8. Otherwise the DSTC proceeds to #5.

#5 Determine whether the SW Start transfer and the HW Start transfer has higher priority according to the setting of the SWPR bit in the CFG Register. If the SW Start transfer has higher priority, the DSTC proceeds to #16. Otherwise the DSTC proceeds to #6.

If there is another HW Start transfer request, the DSTC proceeds to #7. Otherwise the DSTC proceeds to #16.

#7 Start data transfer for that HW Start transfer.

#8 Until that HW Stat transfer has completed the Chain Start, ended the transfer and the DSTC starts waiting for the next Start Trigger, the DSTC keeps waiting for the start of the execution of the SW Start transfer started in #1.

#9 If that HW Start transfer has ended in the form of error, the DSTC proceeds to #12. Otherwise the DSTC proceeds to #10.

#10 If there is a Chain Start transfer in that HW Start transfer, the DSTC proceeds to #11. Otherwise the DSTC proceeds to #5.

#11 If the Chain Start transfer in that HW Start transfer is locked, the DSTC proceeds to #7. Otherwise the DSTC proceeds to #5.

#12 If CFG:ESTE is "0", the DSTC proceeds to #13. Otherwise the DSTC proceeds to #14.

#13 If there has been no error record (EST[2:0] = 000) in the MONERS Register up to #13, the MONERS Register records the error information of the DES of the HW Start transfer that has caused an error. If there is an error record (EST[2:0] ≠ 000), the DSTC sets the DER bit to "1". The DSTC proceeds to #5.

#14 The same process as #13 is executed. At the same time, the ESTOP bit in the MONERS Register is set to "1". The DSTC proceeds to #15.

#15 The DSTC is holding the start of the execution of the SW Start transfer started in #1 while the ESTOP bit is "1". If an ERCLR command is issued by the CPU to the CMD Register and the ESTOP bit is cleared to "0", the DSTC proceeds to #5.

#16 The flow inside the bold box shows the transfer operations of the DSTC according to a specified DES. For details of the flow inside the bold box, see section "Operation flow after specifying of DESP". In the case of the SW Start transfer, the DSTC executes the transfer according to the DES specified in SWDESP. After the transfer has been processed, the operation of the DSTC branches to one of the five operations shown in Figure 3-10.

#17 In the case of a Chain Start, the DSTC updates the value of SWTR:SWDESP.

#18 If CHLK is "1", the DSTC proceeds to #16 and successively executes the transfers started by the Chain Start. Otherwise the DSTC proceeds to #5.

#19 If the transfer ends normally and there is an interrupt flag set instruction, the DSTC executes the processes in #19. The DSTC clears SWTR:SWREQ to "0" and sets SWTR:SWST to "1".

#20 If the transfer ends normally and there is no interrupt flag set instruction, the DSTC executes the processes in #20. The DSTC clears SWTR:SWREQ to "0".

#21 The DSTC ends the transfer caused by the SW Start trigger in #1. The DSTC waits for either a new Start Trigger or a succeeding Start Trigger. The DESP of the DES whose transfer has ended is kept in SWTR:SWDESP.

#22 If that SW Start transfer has ended in the form of error and CFG:ESTE is "0", the DSTC proceeds to #23. Otherwise the DSTC proceeds to #24.

#23 If there has been no error record in the MONERS Register up to #23, the MONERS Register records the error information of the DES of the SW Start transfer that has caused an error. If there is an error record, the DSTC sets the DER bit to "1". The DSTC clears SWTR:SWREQ to "0".

#24 The same process as #23 is executed. At the same time, the ESTOP bit in the MONERS Register is set to "1", and the DSTC holds the start of the transfer of other HW Start transfer.

#25 The transfer caused by the SW Start Trigger in #1 ends in the form of error. SWTR:SWST is not set to "1" regardless of the value of CHRS. The DSTC waits for a new Start Trigger.

#26 In a write access to the SWTR Register from the CPU as explained in #1, if the condition in #2 is not fulfilled, the DSTC ignores the write access to the SWTR Register. The DSTC does not accept the SW Start transfer request.

● Additional information on controlling DSTC in SW Start transfer

If the condition in #2 is not fulfilled, meaning that the SW Start transfer instruction has been executed before #2, and that transfer has not ended (SWREQ ≠ 0) or the DSTC is not in the normal state (CMD ≠ 00) or the DSTC is in the error stop state (ESTOP ≠ 0), the DSTC ignores the new SW Start transfer request from the CPU and does not accept it.

Pay attention to this behavior of the DSTC especially when using the DSTC with CFG:ESTE set to "1". If the DSTC has stopped for an error due to another HW Start transfer, any new SW Start transfer request (write access to a register) is ignored, and SWREQ is not set to "1". Therefore, if the DSTC reads "0" from the SWREQ bit in the SWTR Register after making a write access to the SWTR Register, it cannot determine whether an SW Start request has been ignored or a transfer has ended. Moreover, if using the DSTC with CFG:ESTE set to "1", in an SW Start transfer, set DES0:CHRS to a value that when the DSTC does not execute the Chain Start, always sets SWTR:SWST to "1". With DES0:CHRS set in this way, after a write access has been made to the SWTR Register, that both SWREQ bit and SWST bit read "0" indicates that no transfer request has been accepted. If a transfer request has been accepted, since either SWREQ bit or SWST bit is "1", the DSTC can determine whether an SW Start transfer has been ignored or a transfer has ended.

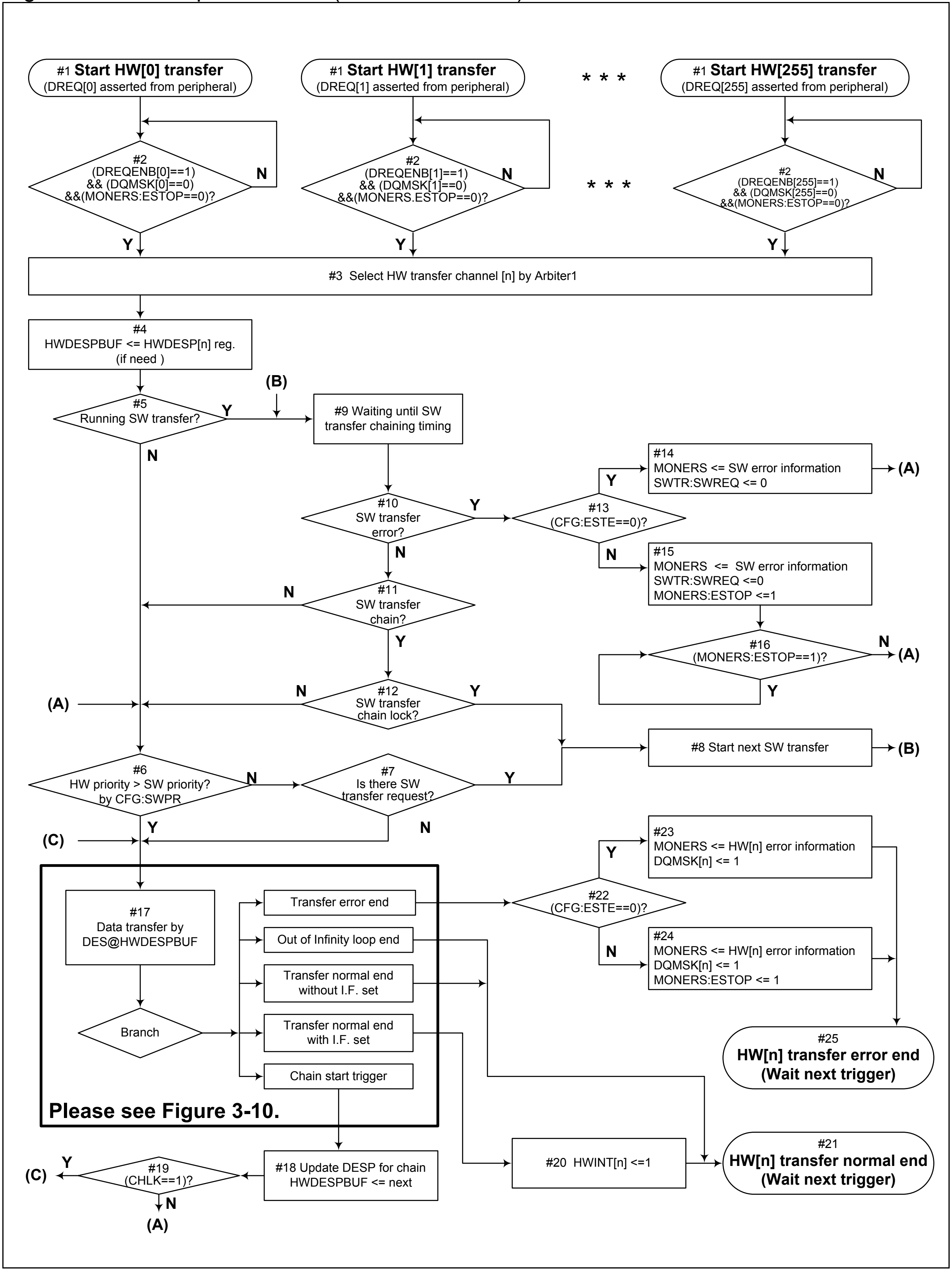
If the SWINT interrupt signal has been asserted by using the SWST bit, clear the SWST bit to "0" by issuing an SWCLR command during interrupt processing. Even if the SWST bit is not cleared to "0", a new SW Start request can be made by making a write access to the SWTR Register. However, in the process explained in #3, the SWST bit is always cleared to "0" and the SWINT interrupt signal is negated.

In #4, if another HW Start transfer is being executed or the Chain Start in that HW Start transfer has been locked, it may take time to start an SW Start transfer even if it has a high priority.

■ HW Start transfer flow

The operations the DSTC executes after receiving an HW Start Trigger from a peripheral are explained below. Figure 3-9 shows a flow chart of the operations of the DSTC. Numbers in the figure correspond to those used in the explanation after the figure.

Figure 3-9 DSTC operation flow (HW Start transfer)



#1 The DSTC starts the HW Start transfer from the assertion of the DREQ[n] signal from a peripheral.

#2 If the DREQENB[n] Register and the DQMSK[n] Register are (DREQENB[n]==1)&& (DQMSK[n]==0), the DSTC proceeds to #3. If the DQMSK[n] Register is set to "1", the DSTC ignores the DREQ[n] signal from a peripheral and holds the start of the HW Start transfer.

#3 Processes in #3 are processes to be executed by Arbiter 1. In the case of transfer requests from multiple HW channels. the DSTC selects the number of the HW channel (n) on which it executes a transfer. The DSTC keeps the transfer requests from other channels until the transfer on the HW[n] selected ends normally or ends due to an error and the DSTC starts waiting for a Start Trigger.

#4 Based on the channel number (n) selected, store the DESP value of the HWDESP[n] Register in HWDESPBUF. As the channel number is the same as the one in the previous reference, if the value of HWDESPBUF is valid, it is not necessary to refer to the HWDESP[n] Register.

Processes explained in #5 to #16 are details of operations of Arbiter 2 and processes in another SW Start transfer. The DSTC executes the same processes as #4 to #15 in "SW Start transfer flow". If there is a SW Start transfer request and that SW Start transfer has a high priority, or if the Chain Start in that SW Start transfer has been locked, the DSTC executes that SW Start transfer first. In addition, if that SW Start transfer ends in the form of error, the DSTC records the error information of the SW Start transfer in the MONERS Register. If the ESTOP bit is set to "1" due to the error end of the SW Start transfer, the DSTC holds the transfer start of HW[n].

#17 The flow inside the bold box shows the transfer operations of the DSTC according to a specified DES. For details of the flow inside the bold box, see section "Operation flow after specifying of DESP". In the case of the HW Start transfer, the DSTC executes the transfer according to the DES specified in HWDESP. After the transfer has been processed, the operation of the DSTC branches to one of the five operations shown in Figure 3-10.

#18 In the case of a Chain Start, the DSTC updates HWDESPBUF.

#19 If CHLK is "1", the DSTC proceeds to #17 and successively executes the transfers started by the Chain Start. Otherwise the DSTC proceeds to #6.

#20 If the transfer ends normally and there is an interrupt flag set instruction, the DSTC sets HWINT[n] to "1".

#21 The DSTC ends the transfer caused by the SW Start trigger in #1. The DSTC waits for either a new Start Trigger or a succeeding Start Trigger. If the DSTC keeps the HW Start Trigger for another channel, Arbiter 1 selects the channel on which a transfer is to be executed and the DSTC proceeds to #4.

#22 If that HW Start transfer has ended in the form of error and CFG:ESTE is "0", the DSTC proceeds to #23. Otherwise the DSTC proceeds to #24.

#23 If there has been no error record in the MONERS Register up to #23, the MONERS Register records the error information of the DES of the HW Start transfer being executed. If there is an error record, the DSTC sets the DER bit to "1". In addition, the DSTC sets the DQMSK[n] Register to "1" to suppress future transfer requests from HW channel n.

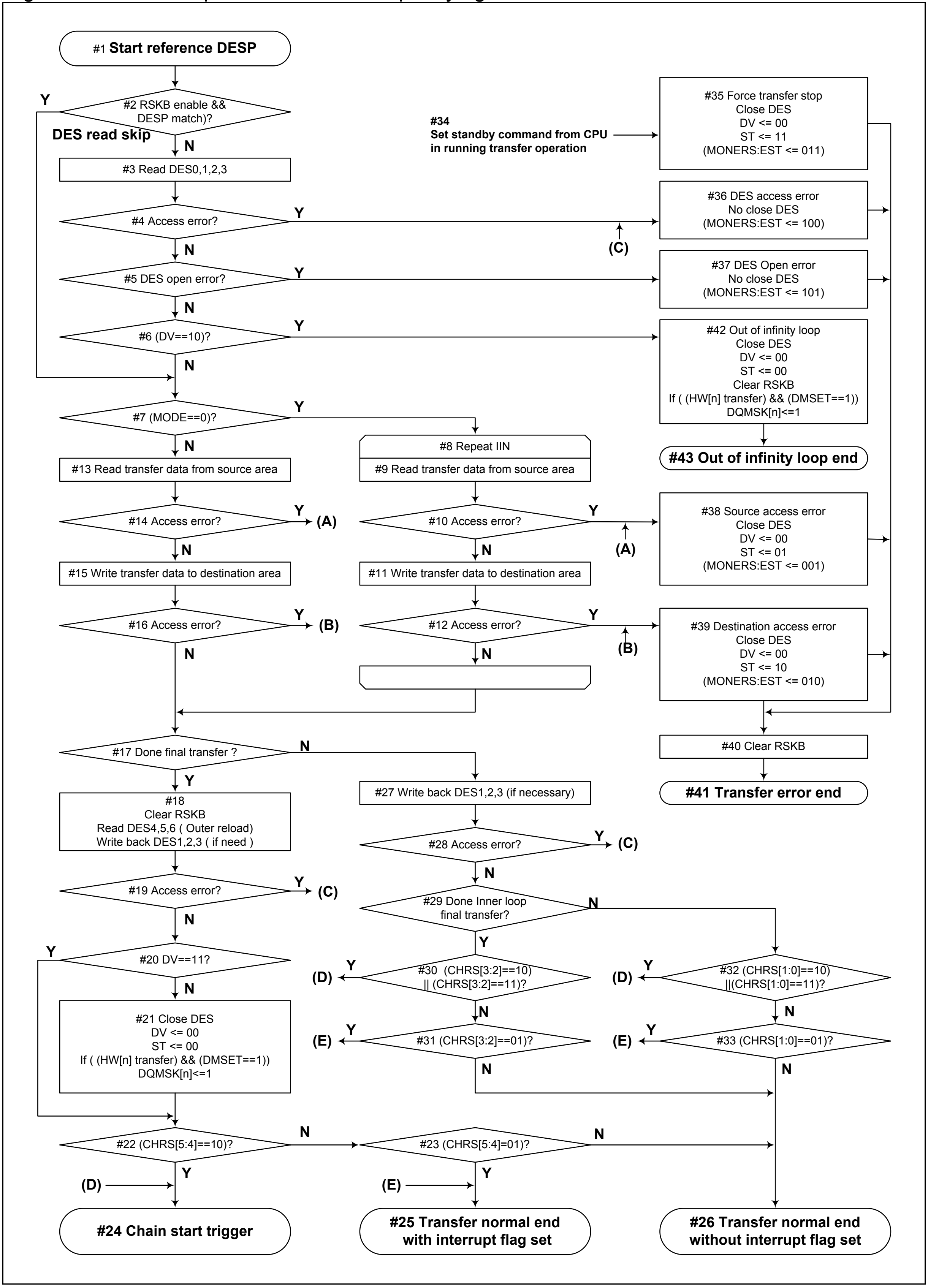
#24 The same process as #23 is executed. At the same time, the ESTOP bit in the MONERS Register is set to "1", and the DSTC holds the start of other Start transfer.

#25 The transfer caused by the HW Start Trigger in #1 ends in the form of error. The HWINT[n] Register is not set to "1" regardless of the value of CHRS. The DSTC waits for a new Start Trigger. If the DSTC keeps the HW Start Trigger for another channel, Arbiter 1 selects the channel on which a transfer is to be executed and the DSTC proceeds to #4.

■ Operation flow after specifying of DESP

The operations the DSTC executes after a DESP has been executed are explained below. Figure 3-10 shows a flow chart of the operations the DSTC executes after a DESP has been specified. Numbers in the figure correspond to those used in the explanation after the figure.

Figure 3-10 DSTC operation flow after specifying of DESP



#1 The DSTC starts its operation from referring to a DESP specified by Arbiter 2.

#2 If the read skip buffer function is enabled (CFG:RBDIS = 0) and the DESP to which the DSTC refers is the same as the one it referred to, the DSTC skips referring to the DES in the memory area and proceeds to #7. Otherwise the DSTC proceeds to #3.

#3, #4 The DSTC reads the DES in the area of DESTP+DESP specified. If an access error occurs in referring to the DES area, the DSTC proceeds to #36. Otherwise the DSTC proceeds to #5.

#5 The DSTC checks details of the DES. If a DES open error occurs, the DSTC proceeds to #37. Otherwise the DSTC proceeds to #6.

#6 If DES0:DV is "10", the DSTC proceeds to #42. Otherwise the DSTC proceeds to #7.

#7 If DES0:MODE is "0", the DSTC proceeds to #8. Otherwise the DSTC proceeds to #13.

#8 to #12 If mode 0 transfer has been specified, the DSTC executes transfers successively for the times specified in DES1:IIN. If an access error occurs in a transfer source access, the DSTC proceeds to #38. If an access error occurs in a transfer destination access, the DSTC proceeds to #39. If no access error occurs in a transfer source access or in a transfer destination access, the DSTC proceeds to #17.

#13 to #16 If mode 1 transfer has been specified, the DSTC executes one transfer. If an access error occurs in a transfer source access, the DSTC proceeds to #38. If an access error occurs in a transfer destination access, the DSTC proceeds to #39. If no access error occurs in a transfer source access or in a transfer destination access, the DSTC proceeds to #17.

#17 If IIN×ORM times of transfer have ended (both ORM and IRM of the internal transfer number counter are "1"), the DSTC proceeds to #18. Otherwise the DSTC proceeds to #27.

#18, #19 The DSTC clears the read skip buffer, reads required values from DES4 to DES6 areas according to the instruction specified in DES0:ORL, and writes the required values to DES1 to DES3 areas. If an access error occurs in updating the DES area reference, the DSTC proceeds to #36. Otherwise the DSTC proceeds to #20.

#20 If DES0:DV is "11", the DSTC proceeds to #21. If DES0:DV is "01", the DSTC proceeds to #20. (The DSTC has proceeded to #37 when DES0:DV has been "00" as stated in #5. The DSTC has proceeded to #42 when DES0:DV has been "10" as stated in #6.)

#21 The DSTC executes the DES close process. The DSTC updates DES0:DV to "00" and DES0:ST to "00". In a transfer for an HW Start trigger (or a Chain transfer from an HW Start trigger), if DES0:DMSET is "1", the DSTC sets DQMSET[n] to "1" and suppresses future HW Start transfer requests of channel n.

#22, #23 If DES0:CHRS[5:4] are "10", the DSTC proceeds to #24. If DES0:CHRS[5:4] are "01", the DSTC proceeds to #25. If DES0:CHRS[5:4] are "00", the DSTC proceeds to #26. (The DSTC has proceeded to #37 when DES0:CHRS[5:4] have been "11" as stated in #5.)

#24 The DSTC executes a Chain Start transfer. The transfer of the DESP specified in #1 ends normally. For operations to be executed afterward, see the previous section.

#25 The transfer of the DESP specified in #1 ends normally with the interrupt flag set. For operations to be executed afterward, see the previous section.

#26 The transfer of the DESP specified in #1 ends normally without the interrupt flag being set. For operations to be executed afterward, see the previous section.

#27, #28 The DSTC writes back values required for the transfer for the next Start Trigger to DES1 to DES3 areas. If an access error occurs in updating the DES area, the DSTC proceeds to #36. Otherwise the DSTC proceeds to #29.

#29 If transfers for the times of the inner loop count have ended (ORM of the internal transfer number counter is not "1" and IRM of the internal transfer number counter is "1"), the DSTC proceeds to #30. Otherwise the DSTC proceeds to #32. Nonetheless, if MODE is "0", the DSTC always proceeds to #30.

#30, #31 If DES0:CHRS[3:2] are "10" or "11", the DSTC proceeds to #24. If DES0:CHRS[3:2] are "01", the DSTC proceeds to #25. If DES0:CHRS[3:2] are "00", the DSTC proceeds to #26.

#32, #33 If DES0:CHRS[1:0] are "10" or "11", the DSTC proceeds to #24. If DES0:CHRS[1:0] are "01", the DSTC proceeds to #25. If DES0:CHRS[1:0] are "00", the DSTC proceeds to #26.

#34 If the CPU issues a standby transition command during a transfer, the DSTC interrupts the transfer and proceeds to #35.

#35 If the transfer is compulsorily stopped, the DSTC executes the DES error close process. The DSTC updates DES0:DV to "00" and DES0:ST to "11", and then proceeds to #40. If there is no error record in the MONERS Register, the DSTC sets MONERS:EST to "011".

#36 If a DES access error occurs, the DSTC does not execute the DES error close process, and proceeds to #40. If there is no error record in the MONERS Register, the DSTC sets MONERS:EST to "100".

#37 If a DES open error occurs, the DSTC does not execute the DES error close process, and proceeds to #40. If there is no error record in the MONERS Register, the DSTC sets MONERS:EST to "101".

#38 If a transfer source access error occurs, the DSTC executes the DES error close process. The DSTC updates DES0:DV to "00" and DES0:ST to "01", and then proceeds to #40. If there is no error record in the MONERS Register, the DSTC sets MONERS:EST to "001".

#39 If a transfer destination access error occurs, the DSTC executes the DES error close process. The DSTC updates DES0:DV to "00" and DES0:ST to "10", and then proceeds to #40. If there is no error record in the MONERS Register, the DSTC sets MONERS:EST to "010".

#40, #41 The DSTC clears the read skip buffer. The transfer of the DESP specified in #1 ends in the form of error. For operations to be executed afterward, see the previous section.

#42, #43 The DSTC clears the read skip buffer. The DSTC executes the DES close process. The DSTC updates DES0:DV to "00" and DES0:ST to "00". In a transfer for an HW Start trigger (or a Chain transfer from an HW Start trigger), if DES0:DMSET is "1", the DSTC sets DQMSET[n] to "1" and suppresses future HW Start transfer requests. The transfer of the DESP specified in #1 ends as an infinite loop out. For operations to be executed afterward, see the previous section.

4. Examples of DSTC Operations and Control

This section describes examples of DSTC operations and control.

4.1. Transfer Operation Example 1

This section describes transfer operation example 1. Transfer operation example 1 is an example on SW Start transfer in mode 0.

■ DES values at transfer start

Table 4-1 shows the settings of the DES in transfer operation example 1. As ORL[2:0] are set to "101", there is no DES5 area. The DES has 6-word configuration consisting of DES0 to DES4 and DES6. (The address of DES6 is DESP+0x0014.)

Table 4-1 DES values at transfer start in transfer operation example 1

Address	DES No.	Value

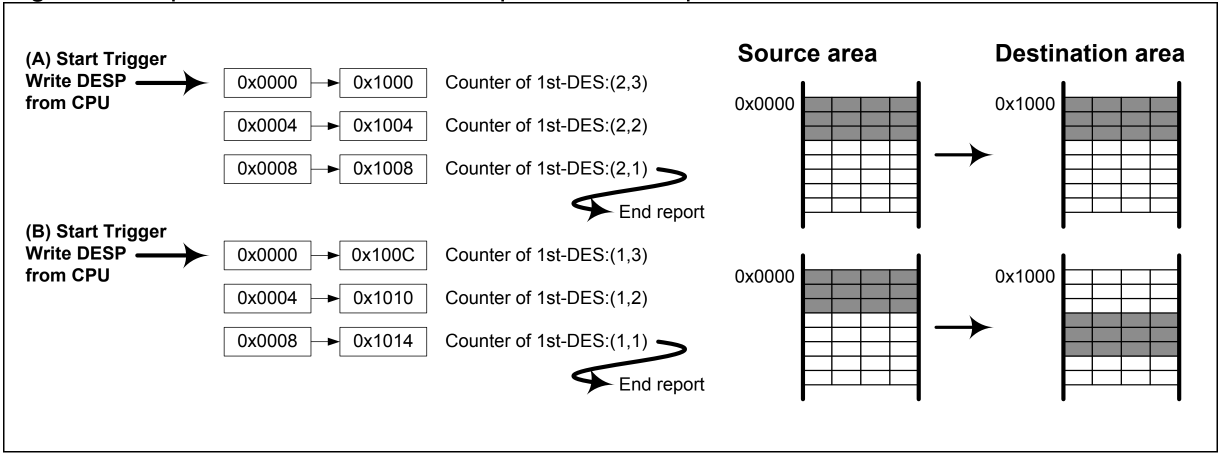
DESP+0x0000	DES0	DV = 01 : DES close process to be executed at the end of transfer
		MODE = 0, TW = 10 : Mode 0, 32-bit (word) transfer
		ORL = 101 : OuterReload of DES1 ≤ DES4, DES3 ≤ DES6
		SAC = 001 : Increment InnerReload of TW×1 available
		DAC = 000 : Increment InnerReload of TW×1 unavailable
		CHRS = 010100 : There is no Chain Start; an interrupt flag has been set.
		DMSET = 0 : Set DMSET to "0" as the transfer is an SW Start transfer.
		CHLK = 0 : No Chain lock
		ACK = 00 : Set ACK to "00" as the transfer is an SW Start transfer.

DESP+0x0004	DES1	ORM = 0x0002, IIN = 0x0003
DESP+0x0008	DES2	SA = 0x00000000
DESP+0x000C	DES3	DA = 0x00001000
DESP+0x0010	DES4	ORM = 0x0002, IIN = 0x0003 (same as DES1)
DESP+0x0014	DES6	DA = 0x0000 1000 (same as DES3)

■ Transfer operation flow

Figure 4-1 Operation flow in transfer operation example 1 shows the transfer operation flow in transfer operation example 1. The Start Triggers of (A) and (B) in the figure show write accesses of the DESP to the SWTR Register from the CPU.

Figure 4-1 Operation flow in transfer operation example 1



The DSTC starts the transfer of the DES due to the Start Trigger of (A). Values inside rectangles in Figure 4-1 are transfer source addresses and transfer destination addresses. The DSTC starts from a 32-bit transfer to the area from address 0x0000 to address 0x1000. The DSTC executes three times (IIN = 3) of 32-bit transfer successively. As for the transfer number counter for the DES, at the start of the transfer, the outer loop counter remain (ORM) is "2", and the inner loop counter remain (IRM) is "3". In Figure 4-1, the remains of the transfer number counter are expressed as (2,3). After three times of transfer, the transfer number counter reads (2,1). As ORM is not "1" and IRM is "1", the DSTC uses the value of CHRS[3:2] for determining the next process. As CHRS[3:2] are "01", the DSTC sets SWTR:SWST to "1" and waits for the next Start Trigger.

The DSTC starts the transfer of the DES again due to the Start Trigger of (B). InnerReload of address is applied to SA. DA keeps increasing. The DSTC restarts from a 32-bit transfer to the area from address 0x0000 to address 0x100C. The DSTC executes three times (IIN = 3) of transfer successively. The transfer number counter starts counting from (1,3) and reads (1,1) after three times of transfer. As ORM is "1" and IRM is "1", the DSTC executes the DES close process if DV of the DES is set to "01". The DSTC uses the value of CHRS[5:4] for determining the next process. As CHRS[5:4] are "01", the DSTC sets SWTR:SWST to "1".

■ DES values stored after transfer end

If the transfer in transfer operation example 1 ends normally, the values of DES are updated as shown in Table 4-2. Values that are different from what they were before the transfer start are in bold type in the table. According to the setting of ORL, values of DES4 and DES6 are copied to DES1 and DES3 respectively, making DES1 and DES3 have the same values as those before the start of transfer. Though OR[1] is "0", according to the settings of InnerReload, DES2 has the same value as that before the start of transfer. The DSTC updates the value of DV to "00", and returns the ownership of DES to the CPU. The DSTC updates the value of ST to "00", and notifies the CPU that the transfer has ended normally.

To execute a transfer with the updated DES mentioned above, update the value of DV via the CPU.

Table 4-2 DES values after end of transfer in transfer operation example 1

DES No.	Value
DES0	**DV = 00, ST = 00,** other values same as what they were before the start of transfer
DES1	ORM = 0x0002, IIN = 0x0003
DES2	SA = 0x00000000
DES3	DA = 0x00001000
DES4	Same as what they were before the start of transfer
DES6	Same as what they were before the start of transfer

■ Supplementary information

The Start Trigger of (B) cannot be issued until the transfer triggered by the Start Trigger of (A) ends and the DSTC enters the Start Trigger wait state. When SWTR:SWREQ is "1", the write access (SW Start Trigger) to the SWTR Register is ignored.

If a transfer triggered by the Start Trigger of (A) ends and the DSTC enters the Start Trigger wait state, an SW Start Trigger other than that of (B) can be issued for other DES. After an SW Start transfer for another DES has ended, if the DESP of that DES is written to the SWTR Register, the Start Trigger of (B) is issued, and the DSTC continues the data transfer from the location at which the transfer of (A) ended.

The Start Trigger of (B) is not always required after the transfer triggered by the Start Trigger of (A) has ended. If the Start Trigger of (B) is not issued and the DSTC does not continue transferring data, after the transfer triggered by the Start Trigger of (A) has ended (even no DES close process is executed), the CPU can modify the DES area of the transfer.

4.2. Transfer Operation Example 2

This section describes transfer operation example 2. Transfer operation example 2 is an example on HW Start transfer in mode 1.

■ DES values at transfer start

Table 4-3 shows the settings of the DES in transfer operation example 2. As ORL[2:0] are set to "000", there are no DES4 area, DES5 area or DES6 area. The DES has 4-word configuration consisting of DES0 to DES3.

Table 4-3 DES values at transfer start in transfer operation example 2

Address	DES No.	Value

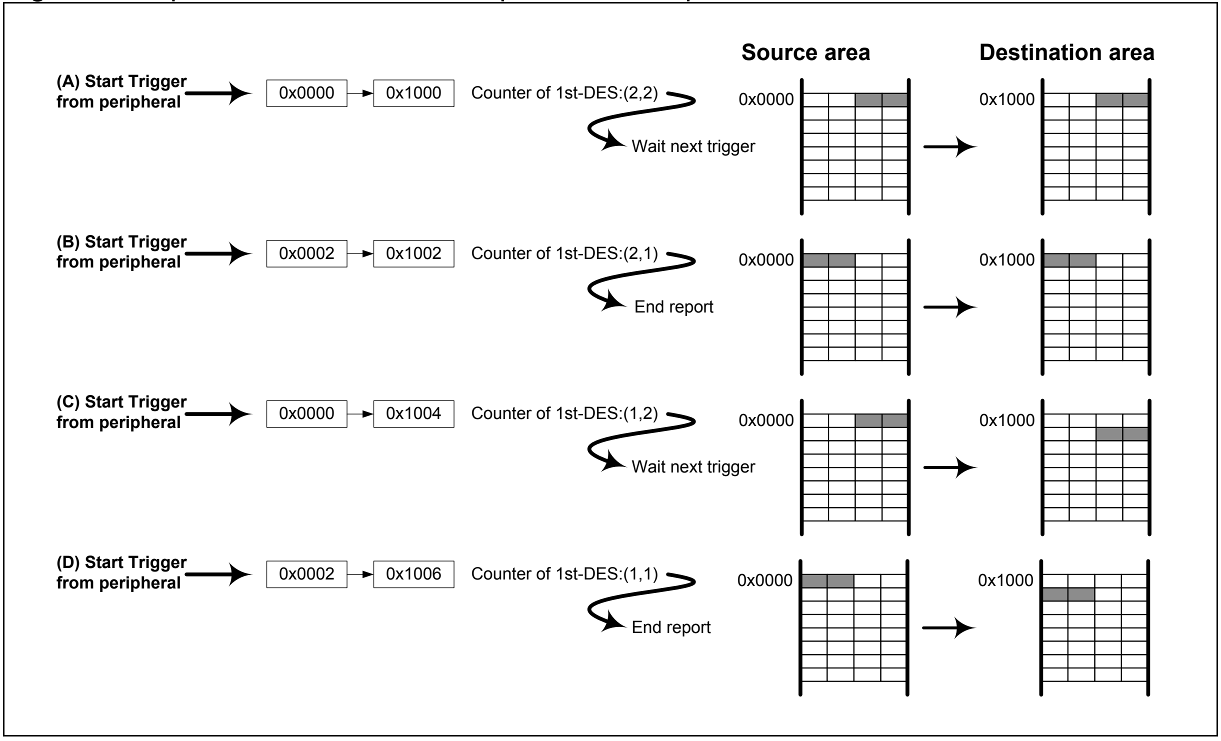
DESP+0x0000	DES0	DV = 01 : DES close process at the of transfer
		MODE = 1, TW = 01 : Mode 1, 16-bit (halfword) transfer
		ORL = 000 : No OuterReload
		SAC = 001 : Increment InnerReload of TW×1 available
		DAC = 000 : Increment InnerReload of TW×1 unavailable
		CHRS = 010100 : There is no Chain Start; an interrupt flag has been set.
		DMSET = 0 : No DQMSK[n] is set in a DES close process.
		CHLK = 0 : No Chain lock
		ACK = 01 : Set ACK to "01" as the DES is directly started by the DREQ signal.

DESP+0x0004	DES1	ORM = 0x0002, IIN = 0x02, IRM = 0x02
DESP+0x0008	DES2	SA = 0x00000000
DESP+0x000C	DES3	DA = 0x00001000

■ Transfer operation flow

Figure 4-2 shows the transfer operation flow in transfer operation example 2. The Start Trigger of (A) in Figure 4-2 corresponds to the assertion of the DREQ[n] signal from a peripheral. For DREQ[n], set the values of the DREQENB[n], DQMSK[n] and HWDESP[n] Registers before starting a transfer.

Figure 4-2 Operation flow in transfer operation example 2



The DSTC starts the transfer of the DES due to the Start Trigger of (A). The DSTC executes one 16-bit transfer to the area from address 0x0000 to address 0x1000. The transfer number counter reads (2,2). As IRM is not "1", according to the setting of CHRS[1:0] of the DES (CHRS[1:0] = 00), the DSTC does not set the HWINT[n] Register to "1". The DSTC waits for the next Start Trigger.

The DSTC starts the transfer of the DES again due to the Start Trigger of (B). The DSTC executes one 16-bit transfer to the area from address 0x0002 to address 0x1002. The transfer number counter reads (2,1). As ORM is not "1" and IRM is "1", according to the setting of CHRS[3:2] of the DES (CHRS[3:2] = 01), the DSTC sets the HWINT[n] Register to "1". The DSTC waits for the next Start Trigger.

The DSTC starts the transfer of the DES again due to the Start Trigger of (C). InnerReload of address is applied to SA. DA keeps increasing. The DSTC executes one 16-bit transfer to the area from address 0x0000 to address 0x1004. The transfer number counter reads (1,2). As IRM is not "1", according to the setting of CHRS[1:0] of the DES (CHRS[1:0] = 00), the DSTC does not set the HWINT[n] Register to "1". The DSTC waits for the next Start Trigger.

The DSTC starts the transfer of the DES again due to the Start Trigger of (D). The DSTC executes one 16-bit transfer to the area from address 0x0002 to address 0x1006. The transfer number counter reads (1,1). As ORM is "1" and IRM is "1", the DSTC executes the DES close process if DV of the DES is set to "01". According to the setting of CHRS[5:4] of the DES (CHRS[5:4] = 01), the DSTC sets the HWINT[n] Register to "1".

■ DES values stored after transfer end

If the transfer in transfer operation example 2 ends normally, the values of DES are updated as shown in Table 4-4. Values that are different from what they were before the transfer start are in bold type in the table. As ORL has been set to "000", if the values in the table are used in the next transfer, the DSTC cannot execute a transfer same as the transfer mentioned above. In this situation, rebuild the DES via the CPU.

Table 4-4 DES values after end of transfer in transfer operation example 2

DES No.	Value
DES0	**DV = 00, ST = 00,** other values same as what they were before the start of transfer
DES1	**ORM = 0x0001, IRM = 0x01**, IIN = 0x02
DES2	SA = 0x00000000
DES3	**DA = 0x00001006**

4.3. Transfer Operation Example 3

This section describes transfer operation example 3. Transfer operation example 3 is an example on using the Chain Start of the succeeding DES.

■ DES values at transfer start

In transfer operation example 3, the DSTC uses the Chain Start to re-arrange data at 0x0000 to 0x00FF and transfer data to the area between 0x0100 and 0x01FF. Four DES are used in this example. The first DES is called 1st-DES, the second DES 2nd-DES, the third DES 3rd-DES and the fourth DES 4th-DES. Table 4-5 shows the respective details of the four DES. There is no DES4 in each DES. The respective addresses of the four DES are not shown in Table 4-5. However, allocate the four DES next to each other in sequence from 1st-DES to 4th DES on the memory.

Table 4-5 DES values at transfer start in transfer operation example 3

Area	DES No.	Value

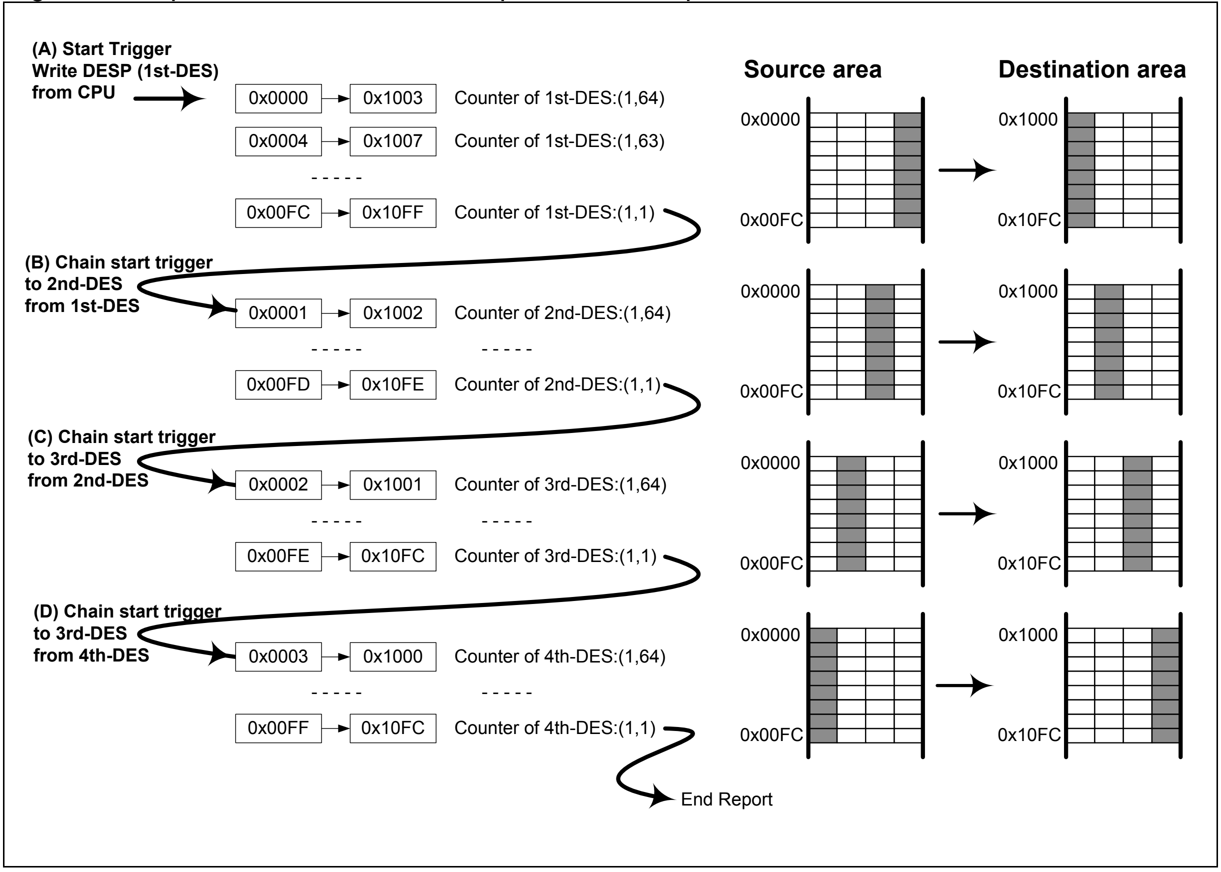
1st-DES	DES0	DV = 11 : No DES close process to be executed at the end of transfer
		MODE = 0, TW = 00 : Mode 0, 8-bit (byte) transfer
		ORL = 110 : OuterReload of DES2 <= DES5, DES3 <= DES6
		SAC = 100 : Increment InnerReload of TW×4 unavailable
		DAC = 100 : Increment InnerReload of TW×4 unavailable
		CHRS = 100000 : There is a Chain Start in the succeeding DES; no interrupt flag has been set.
		DMSET = 0 : Set DMSET to "0" as the transfer is an SW Start transfer.
		CHLK = 0 : No Chain lock
		ACK = 00 : Set ACK to "00" as the transfer is an SW Start transfer.

	DES1	ORM = 0x0001, IIN = 0x0040
	DES2	SA = 0x00000000
	DES3	DA = 0x00001003
	DES5, DES6	DES5 has the same values as DES2 of 1st-DES, and DES6 as DES3 of 1st-DES.
2nd-DES	DES0	Same as DES0 of 1st-DES
	DES1	Same as DES1 of 1st-DES
	DES2	SA = 0x00000001
	DES3	DA = 0x00001002
	DES5, DES6	DES5 has the same values as DES2 of 2nd-DES, and DES6 as DES3 of 2nd-DES.
3rd-DES	DES0	Same as DES0 of 1st-DES
	DES1	Same as DES1 of 1st-DES
	DES2	SA = 0x00000002
	DES3	DA = 0x00001001
	DES5, DES6	DES5 has the same values as DES2 of 3rd-DES, and DES6 as DES3 of 3rd-DES.
4th-DES	DES0	CHRS = 010000 : There is no Chain Start; an interrupt flag has been set. Other values are the same as those of DES0 of 1st-DES.
	DES1	Same as DES1 of 1st-DES
	DES2	SA = 0x00000003
	DES3	DA = 0x00001000
	DES5, DES6	DES5 has the same values as DES2 of 4th-DES, and DES6 as DES3 of 4th-DES.

■ Transfer operation flow

Figure 4-3 shows the transfer operation flow in transfer operation example 3. The Start Trigger of (A) in the Figure 4-3 shows write accesses to the DESP of 1st-DES to the SWTR Register from the CPU. The Start Triggers of (B), (C) and (D) in the Figure 4-3 are Chain Start Triggers.

Figure 4-3 Operation flow in transfer operation example 3



The DSTC starts the transfer of 1st-DES due to the Start Trigger of (A). The DSTC executes one 8-bit transfer to the area from address 0x0000 to address 0x1003, and another 8-bit transfer to the area from address 0x0004 to address 0x1007. The DSTC executes 64 times of transfer (IIN = 64) successively according to the order above. The transfer number counter for 1st-DES starts counting from (1,64) and reads (1,1) after 64 times of transfer. As DV of 1st-DES is "11", the DSTC does not execute the DES close process for 1st-DES. As CHRS[5:4] of 1st-DES is "10", the Chain Start Trigger for the succeeding transfer of 2nd-DES is issued.

The DSTC starts the transfer of 2nd-DES due to the Chain Start Trigger of (B). After 64 times of 8-bit transfer, the counter value becomes (1,1). As DV of 2nd-DES is "11", the DSTC does not execute the DES close process for 2nd-DES. As CHRS[5:4] of 2nd-DES is '10", the Chain Start Trigger for the succeeding transfer of 3rd-DES is issued.

The DSTC starts the transfer of 3rd-DES due to the Start Trigger of (C). After 64 times of 8-bit transfer, the counter value becomes (1,1). As DV of 3rd-DES is "11", the DSTC does not execute the DES close process for 3rd-DES. As CHRS[5:4] of 3rd-DES is "10", the Chain Start Trigger for the succeeding transfer of 4th-DES is issued.

The DSTC starts the transfer of 4th-DES due to the Start Trigger of (D). After 64 times of 8-bit transfer, the counter value becomes (1,1). As DV of 4th-DES is "11", the DSTC does not execute the DES close process for 4th-DES. As CHRS[5:4] of 4th-DES is "01", the DSTC sets SWTR:SWST to "1" and ends the transfer.

As explained in transfer operation example 3, if the Chain Start function is used, transfers defined in multiple DES can be automatically executed one after the other by just issuing a Start Trigger to the first DES.

■ DES values stored after transfer end

If the transfer in transfer operation example 3 ends normally, according to the settings of ORL and DV (ORL = 0 or 1, DV = 11), the values of DES in all DES areas are the same as what they were before the start of the transfer. Since the DSTC does not execute the DES close process, it does not update DV or ST either. In the next transfer, the transfer same as the previous transfer can be executed by just issuing the Start Trigger.

4.4. Transfer Operation Example 4

This section describes transfer operation example 4.

■ DES values at transfer start

Below are details of transfer operation example 4. This example illustrates executing the Chain Start of the current DES again with CHRS set to "11". Table 4-6 shows settings of the DES.

Table 4-6 DES values at transfer start in transfer operation example 4

DES No.	Value

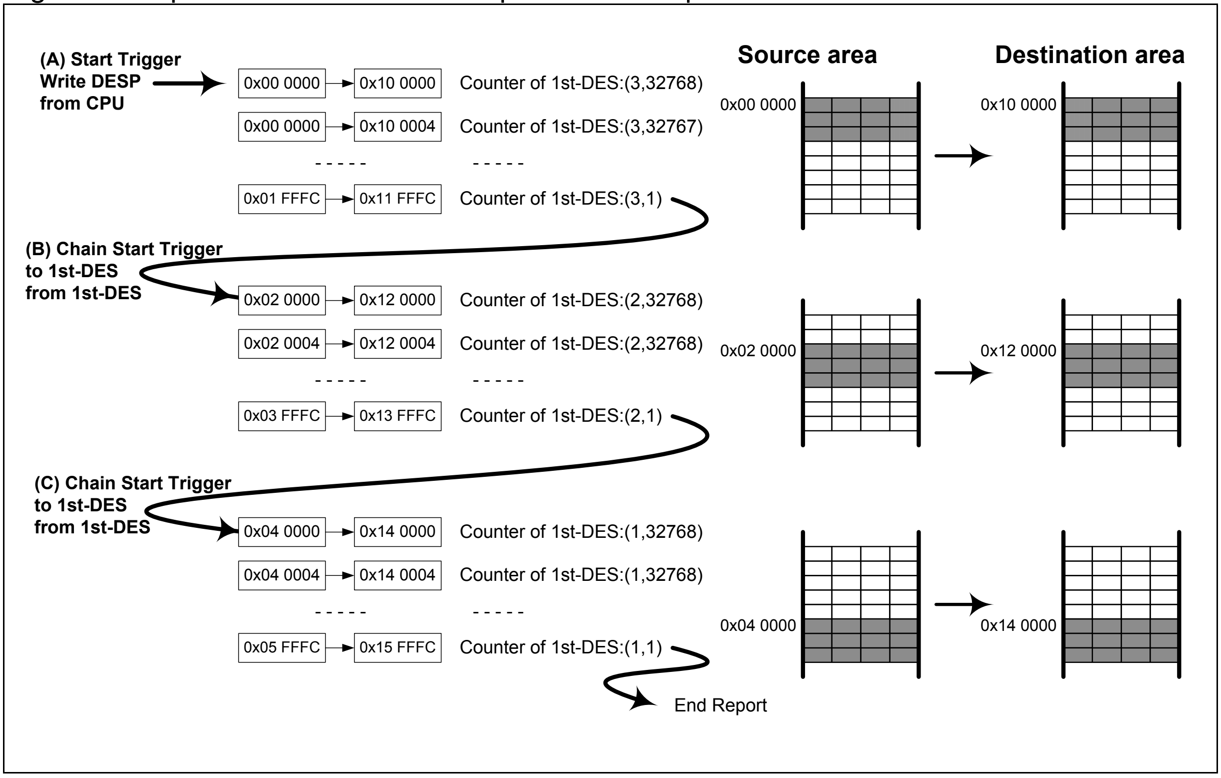
DES0	DV = 01 : DES close process to be executed at the end of transfer
	MODE = 0, TW = 10 : Mode 0, 32-bit (word) transfer
	ORL = 000 : No OuterReload
	SAC = 000 : Increment InnerReload of TW×1 unavailable
	DAC = 000 : Increment InnerReload of TW×1 unavailable
	CHRS = 011100 : There is a Chain Start in the current DES; an interrupt flag has been set.
	DMSET = 0 : Set DMSET to "0" as the transfer is an SW Start transfer.
	CHLK = 0 : No Chain lock
	ACK = 00 : Set ACK to "00" as the transfer is an SW Start transfer.

DES1	ORM = 0x0003, IIN = 0x8000
DES2	SA = 0x00000000
DES3	DA = 0x00100000

■ Transfer operation flow

Figure 4-4 shows the transfer operation flow in transfer operation example 4. The Start Trigger of (A) in the Figure 4-4 represents the write access of the DESP to the SWTR Register from the CPU. The Start Triggers of (B) and (C) in Figure 4-4 are Chain Start Triggers.

Figure 4-4 Operation flow in transfer operation example 4



The DSTC starts the transfer of the DES due to the Start Trigger of (A). The DSTC executes 32768 (IIN = 32768) times of 32-bit transfer successively with the address increasing during transfers. The transfer number counter for the DES starts counting from (3,32768) and reads (3,1) after 32768 times of transfer. As CHRS[3:2] of the DES is "11", the Chain Start Trigger for the transfer of the same DES is issued again.

The DSTC starts the transfer of the DES again due to the Chain Restart Trigger of (B). The DSTC executes 32768 times of transfer again. The transfer number counter for the DES reads (2,1) after 32768 times of transfer. As CHRS[3:2] of the DES is "11", the Chain Start Trigger for the transfer of the same DES is issued again.

The DSTC starts the transfer of the DES again due to the Chain Restart Trigger of (C). The DSTC executes 32768 times of transfer again. The transfer number counter for the DES reads (1,1) after 32768 times of transfer. As DV of the DES is "01", the DSTC executes the DES close process for the DES. As CHRS[5:4] of 4th-DES is "01", the DSTC sets SWTR:SWST to "1" and ends the transfer.

■ DES values stored after transfer end

If the transfer in transfer operation example 4 ends, the values of DES are updated as shown in Table 4-7. Values that are different from what they were before the transfer start are in bold type in the table. In DES2 and DES3, the transfer start addresses in the third outer loop have been stored.

Table 4-7 DES values after end of transfer in transfer operation example 4

DES No.	Value
DES0	**DV = 00, ST = 00,** other values same as what they were at the start of transfer
DES1	**ORM = 0x0001**, IIN = 0x8000
DES2	**SA = 0x00040000**
DES3	**DA = 0x00140000**

■ Supplementary information

As explained above, since CHRS are set to "11", the DSTC can trigger a Chain Start from the current DES to the current DES. In general, in a transfer with MODE set to "0", one Start Trigger triggers the execution of IIN times of transfer. However, if CHRS are set to "11", one Start Trigger can trigger the execution of ORM×IIN times of transfer.

In transfer operation example 4, the DSTC divides a total of 98304 times of transfer into three parts, and executes the three parts (ORM = 3) of transfers (IRM = 32768) separately. As long as the product of ORM and IRM is the same as the total number of transfers, the transfer result remains the same regardless of how many parts transfers are divided into. Once the DSTC starts a transfer, it can start processing another transfer request only when it enters the Start Trigger wait state or it meets the Chain Start time. Therefore, as explained in transfer operation example 4, when the DSTC transfers a large amount of data, the start of transfer for another HW transfer request may be delayed. To prevent such delay from occurring, adjust the value of ORM with the product of ORM and IRM the same as the total amount of transfers, so that a large amount of transfers can be divided into smaller parts. As a result, the DSTC can transfer much data without delaying other HW transfer requests.

Conversely, to prevent the DSTC from processing another HW transfer request at a Chain Start, use the Chain lock by setting CHLK to "1" so that the DSTC can execute transfers successively.

(Note) In the HW Start transfer, for a DES whose transfer is triggered by a start request from a peripheral that has to have to a handshake with the DSTC at every data transfer, CHRS cannot be set to "11".

4.5. Transfer Operation Example 5

This section describes transfer operation example 5.

■ DES values at transfer start

Below are details of transfer operation example 5. This example illustrates an operation in which relatively complicated Chain Start settings are done. Three DES are used in this example. Table 4-8 shows the respective values of 1st-DES, 2nd-DES and 3rd-DES.

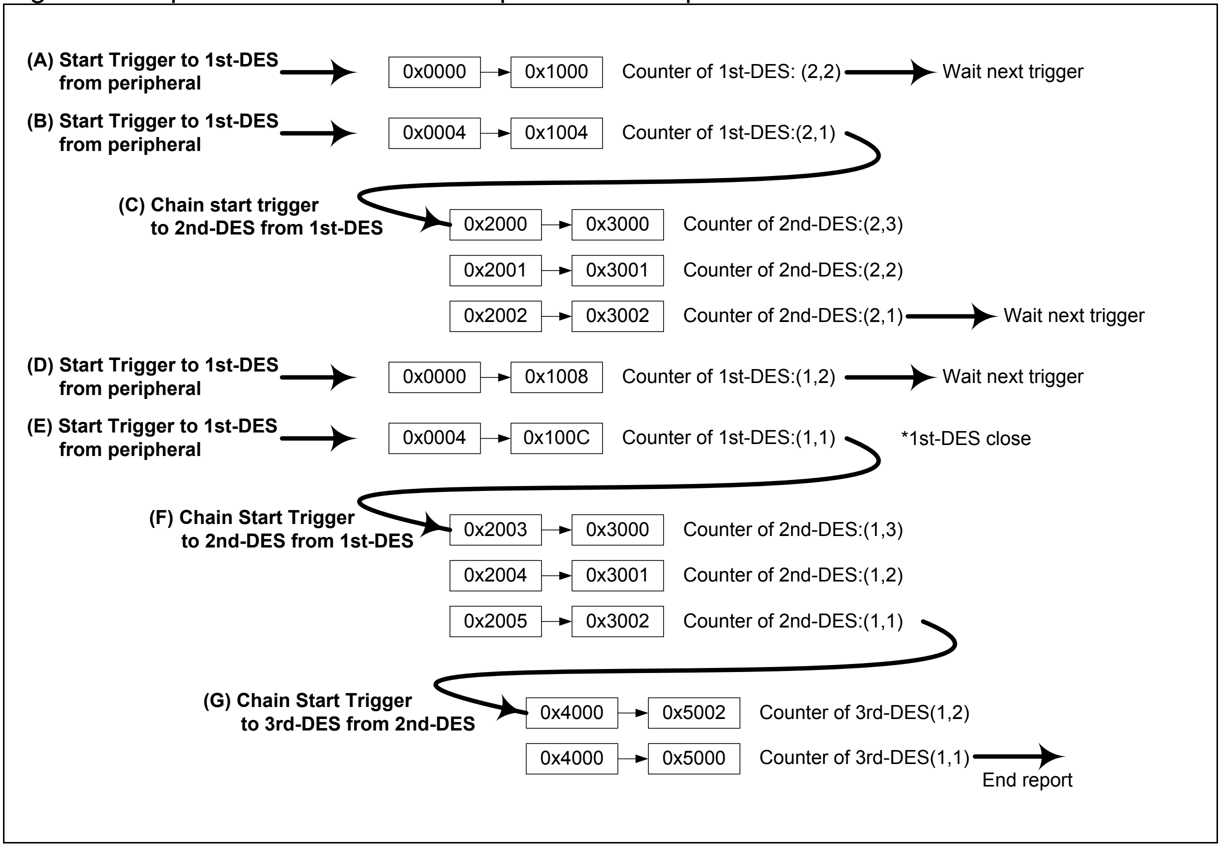
Table 4-8 DES values at transfer start in transfer operation example 5

Area	DES No.	Value
1st-DES	DES0	DV = 01 : DES close process to be executed at the end of transfer MODE = 1, TW = 10 : Mode 1, 32-bit (word) transfer ORL = 101 : OuterReload of DES1 <= DES4, DES3 <= DES6 SAC = 001 : The address is increased by 1 TW at every transfer. InnerReload is available. DAC = 000 : The address is increased by 1 TW at every transfer. InnerReload is unavailable. CHRS = 101000 : There is a Chain Start in the succeeding DES; no interrupt flag has been set. DMSET = 0 : No DQMSK[n] is set in a DES close process. CHLK = 0 : No Chain lock ACK = 01 : Set ACK to "01" as the DES is directly started by the hardware.
	DES1	ORM = 0x0002, IIN = 0x02, IRM = 0x02
	DES2	SA = 0x00000000
	DES3	DA = 0x00001000
	DES4	Same as DES1 of 1st-DES
	DES6	Same as DES3 of 1st-DES
2nd-DES	DES0	DV = 11 : No DES close process to be executed at the end of transfer MODE = 0, TW = 00 : Mode 0, 8-bit (byte) transfer ORL = 011 : OuterReload of DES1 <= DES4, DES2 <= DES5 SAC = 000 : The address is increased by 1 TW at every transfer. InnerReload is unavailable. DAC = 001 : The address is increased by 1 TW at every transfer. InnerReload is available. CHRS = 100000 : There is a Chain Start in the succeeding DES; no interrupt flag has been set. DMSET = 0 : No DQMSK[n] is set in a DES close process. CHLK = 0 : No Chain lock ACK = 00 : Set ACK to "00" as the DES is directly started by a Chain Start.
	DES1	ORM = 0x0002, IIN = 0x0003
	DES2	SA = 0x00002000
	DES3	DA = 0x00003000
	DES4	Same as DES1 of 2nd-DES
	DES5	Same as DES2 of 2nd-DES
3rd-DES	DES0	DV = 11 : No DES close process to be executed at the end of transfer MODE = 0, TW = 01 : Mode 0, 16-bit (halfword) transfer ORL = 000 : No OuterReload SAC = 101 : Fixed address DAC = 111 : The address is decreased by 1 TW at every transfer. InnerReload is available. CHRS = 010000 : There is no Chain Start; an interrupt flag has been set. DMSET = 0 : No DQMSK[n] is set in a DES close process. CHLK = 0 : No Chain lock ACK = 00 : Set ACK to "00" as the DES is directly started by a Chain Start.
	DES1	ORM = 0x0001, IIN = 0x0002
	DES2	SA = 0x00004000
	DES3	DA = 0x00005002

■ Transfer operation flow

Figure 4-5 shows the transfer operation flow in transfer operation example 5. The Start Triggers of (A), (B), (D) and (E) in Figure 4-5 indicate HW Start transfers and correspond to the assertion of the transfer request signal from a peripheral. The Start Triggers of (C), (F) and (G) in Figure 4-5 are Chain Start Triggers.

Figure 4-5 Operation flow in transfer operation example 5



The DSTC starts the transfer of 1st-DES due to the Start Trigger of (A). The DSTC executes one 32-bit transfer to the area from address 0x0000 to address 0x1000. The transfer number counter for 1st-DES is (2,2). According to the setting of CHRS[1:0] of 1st-DES (CHRS[1:0] = 00), the DSTC does not set the HWINT[n] Register to "1". The DSTC waits for the next Start Trigger.

The DSTC starts the transfer of 1st-DES again due to the Start Trigger of (B). The DSTC executes one 32-bit transfer to the area from address 0x0004 to address 0x1004. The transfer number counter for 1st-DES is (2,1). As CHRS[3:2] of 1st-DES is "10", the Chain Start Trigger for the succeeding transfer of 2nd-DES is issued.

The DSTC starts the transfer of 2nd-DES due to the Start Trigger of (C). The DSTC starts from an 8-bit transfer to the area from address 0x2000 to address 0x3000. The DSTC executes three times (IIN = 3) of 8-bit transfer successively. The transfer number counter for 2nd-DES starts counting from (2,3) and reads (2,1) after transfers. According to the setting of CHRS[3:2] of 2nd-DES (CHRS[3:2] = 00), the DSTC does not set the HWINT[n] Register to "1". The DSTC waits for the next Start Trigger.

The DSTC starts the transfer of 1st-DES again due to the Start Trigger of (D). InnerReload of address is applied to SA. DA keeps increasing. The DSTC executes one 32-bit transfer to the area from address 0x0000 to address 0x1008. The transfer number counter reads (1,2). According to the setting of CHRS[1:0] of 1st-DES (CHRS[1:0] = 00), the DSTC does not set the HWINT[n] Register to "1". The DSTC waits for the next Start Trigger.

The DSTC starts the transfer of 1st-DES again due to the Start Trigger of (E). The DSTC executes one 32-bit transfer to the area from address 0x0004 to address 0x100C. The transfer number counter for 1st-DES is (1,1). As DV of 1st-DES is "01", the DSTC executes the DES close process for 1st-DES. As CHRS[5:4] of 1st-DES is "10", the Chain Start Trigger for the succeeding transfer of 2nd-DES is issued. (Note that CHRS[5:0] in 1st-DES have been set to "101000".)

The DSTC starts the transfer of 2nd-DES again due to the Chain Start Trigger of (F). SA keeps increasing. InnerReload of address is applied to DA. The DSTC executes three times (IIN = 3) of 8-bit transfer successively. The transfer number counter for 2nd-DES starts counting from (1,3) and reads (1,1) after transfers. As DV of 2nd-DES is "11", the DSTC does not execute the DES close process for 2nd-DES. As CHRS[5:4] of 2nd-DES is '10", the Chain Start Trigger for the succeeding transfer of 3rd-DES is issued.

The DSTC starts the transfer of 3rd-DES due to the Start Trigger of (G). The DSTC executes three times (IIN = 2) of 16-bit transfer successively. The transfer number counter for 3rd-DES starts counting from (1,2) and reads (1,1) after transfers. As DV of 3rd-DES is "11", the DSTC does not execute the DES close process for 3rd-DES. According to the setting of CHRS[5:4] of 3rd-DES (CHRS[5:4] = 01), the DSTC sets the HWINT[n] Register to "1".

In transfer operation example 5, the transfer ends as the DSTC ends ORM×IIN times of transfer for 3rd-DES. The settings in this example cause the DSTC to execute the following operations. The DSTC issues a Chain Start Trigger for the transfer operation to proceed from 1st-DES to 2nd-DES in the second outer loop count and another in the fourth outer loop count. It sets the HWINT[n] Register to "1" at the end of transfer. After all transfers have ended, the DSTC notifies the CPU that all transfers ended.

■ DES values stored after transfer end

If the transfer in transfer operation example 5 ends, the values of DES are updated as shown in Table 4-9. Values that are different from what they were before the transfer start are in bold type in the table. As the CPU has to have to a handshake with the DSTC, 1st-DES is set to be closed by the DSTC after the transfer has ended. 2nd-DES and 3rd-DES are set to not be closed by the DSTC.

Table 4-9 DES values after end of transfer in transfer operation example 5

Area	DES No.	Value
1st-DES	DES0	**DV = 00, ST = 00,** other values same as what they were at the start of transfer
	DES1 to DES4, DES6	Same as what they were at the start of transfer
2nd-DES	DES0 to DES5	Same as what they were at the start of transfer
3rd-DES	DES0 to DES3	Same as what they were at the start of transfer

4.6. Examples of Controlling DSTC

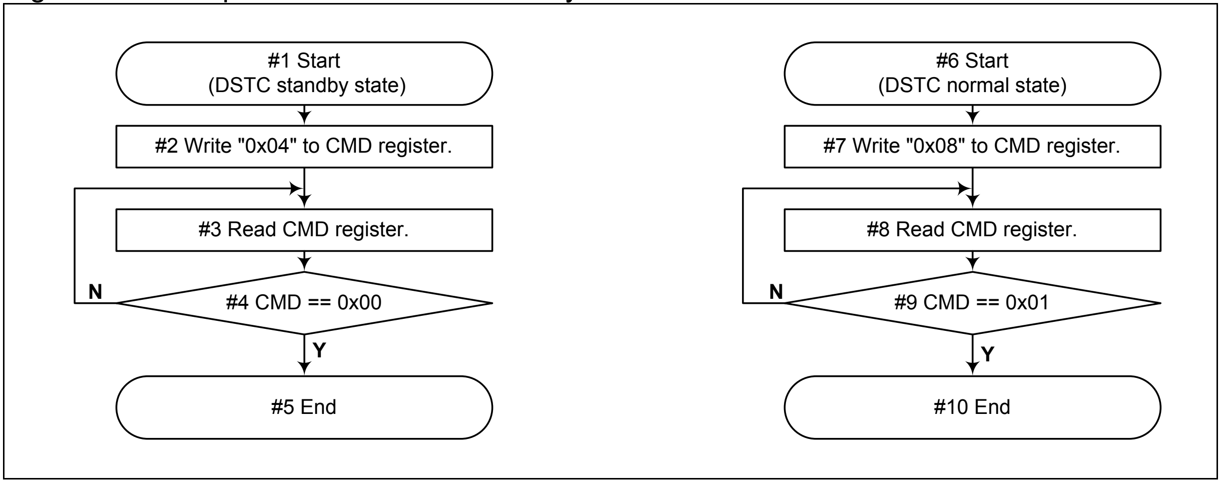
This section explains sample procedures for controlling the DSTC.

■ Sample procedures for transition to standby state and for transition to normal state

The DSTC transits to the standby state upon a bus reset. To make the DSTC execute a transfer, it is necessary to make the DSTC first transit from the standby state to the normal state. In the case of not making the DSTC execute a transfer, the power consumption of the system can be reduced by keeping the DSTC in the standby state.

Figure 4-6 shows a sample procedure for making the DSTC transit from the standby state to the normal state and another sample procedure for making the DSTC transit from the normal state to the standby state. Numbers in the figure correspond to those used in the explanation after the figure.

Figure 4-6 Examples of transition to standby state and transition to normal state



#1 to #5 show the procedure for making the DSTC transit from the standby state to the normal state. Issue a standby release command (write "0x04" to the CMD Register). If the DSTC transits to the normal state, the CMD Register reads "0x00". Check that the CMD Register reads "0x00".

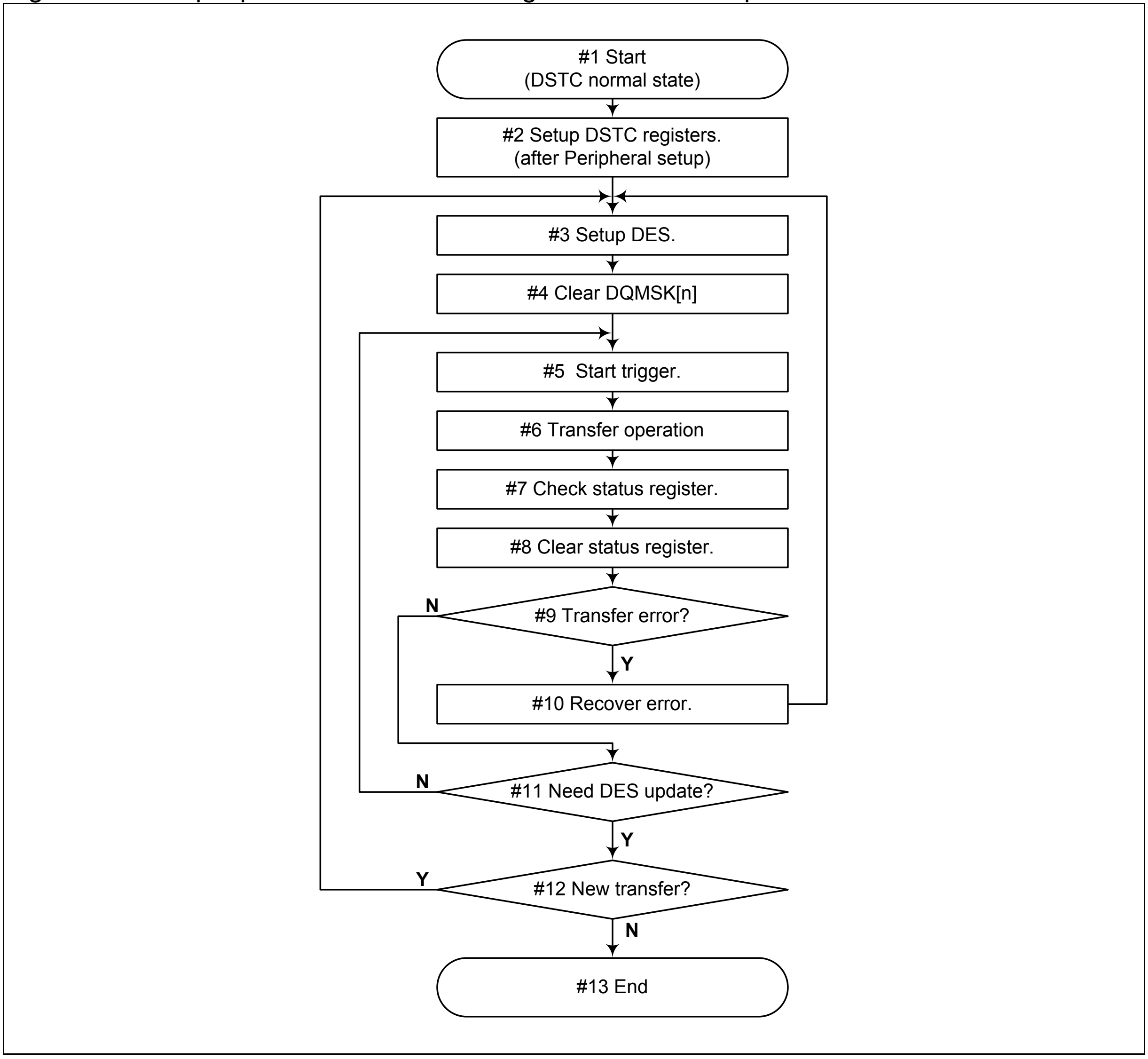
#6 to #10 show the procedure for making the DSTC transit from the normal state to the standby state. Issue a standby transition command (write "0x08" to the CMD Register). If the DSTC transits to the standby state, the CMD Register reads "0x01". Check that the CMD Register reads "0x01".

While the DSTC is executing a transfer, if a standby transition command is issued, the transfer in progress is compulsorily ended and a DES close process for that transfer is executed. After the DES close process has been completed, the DSTC transits to the standby state.

■ Sample procedure for controlling transfer operation

Figure 4-7 illustrates a sample procedure for controlling the transfer operation of the DSTC. Numbers in the Figure 4-7 correspond to those used in the explanation after the figure.

Figure 4-7 Sample procedure for controlling DSTC transfer operation



#1 This sample procedure starts from the point at which the DSTC is in the normal state. If the DSTC is not in the normal state, the following processes cannot be executed.

#2 Initialize the control registers of the DSTC. Set the DESTP Register, the CFG Register, the HWDESP[n] Register and the DREQENB[n] Register to their respective initial values. Write the initial value to the DREQENB[n] Register after completing the setup of the peripheral. The MONERS Register is cleared upon a bus reset. However, after the DSTC has been released from the standby state again, the error record of a previous compulsory end of transfer due to the transition to the standby state may be kept in the MONERS Register. Use the ERCLR Command (write "0x20" to the CMD Register) to clear the MONERS Register.

#3 Build in the CPU the DES area to be used by the DSTC.

#4 The DQMSK[n] Register has been cleared upon a bus reset. However, it may be set by an HW Start transfer error, a standby transition command or a source specified in DMSET of the DES. Write "1" to the DQMSKCLR[n] Register to clear the DQMSK[n] Register. If a transfer request signal (DREQ[n]) from a peripheral has been asserted, the clearing of the DQMSK[n] Register by the CPU becomes the transfer start trigger in #5.

#5 Issues a Start Trigger. In an SW Start transfer, write the DESP to the SWTR Register via the CPU. While an SW Start transfer is in progress, no write access can be made to the SWTR Register. In an HW Start transfer, the assertion of the DREQ[n] signal from a peripheral is the Start Trigger.

#6 The DSTC executes a transfer operation according to the DES. It updates the DES and returns the transfer status.

#7 Check the transfer status of the DSTC via the CPU. In an SW Start transfer, read the SWREQ bit and SWST bit in the SWTR Register. In an HW Start transfer, read the HWINT[n] Register. Read the MONERS Register to check whether there is a transfer error.

#8 Clear the transfer status of the DSTC via the CPU. In an SW Start transfer, issue an SWCLR command from the CPU (write "0x10" to the CMD Register) to clear the SWST bit in the SWTR Register. In an HW Start transfer, write "1" to the HWINTCLR[n] Register to clear the HWINT[n] Register.

#9, #10 After a transfer error has occurred, issue an ERCLR command (write "0x20" to the CMD Register) to clear the MONERS Register. If a transfer has been interrupted by an error, rebuild the DES. In an HW Start transfer, peripheral setup may be necessary.

#11 to #13 If it is not necessary to rebuild the DES, issue the next Start Trigger to start a transfer. If it is necessary to rebuild the DES, rebuild the DES via the CPU, then issue the next Start Trigger to start a transfer.

5. Registers and Descriptors of DSTC

This section explains the functions of registers of the DSTC and the functions of descriptors.

5.1. Lists of Control Registers and DES

This section shows the respective lists of control registers of the DSTC and DES.

Table 5-1 shows a list of the control registers of the DSTC and Table 5-2 a list of DES.

Table 5-1 List of control registers of DSTC

Address	Register name	Reference
+0x00	DESTP	5.2
+0x04	HWDESP	5.3
+0x08	CMD	5.4
+0x09	CFG	5.5
+0x0A	SWTR	5.6
+0x0C	MONERS	5.7
+0x10 to +0x2F	DREQENB	5.8
+0x30 to +0x4F	HWINT	5.9
+0x50 to +0x6F	HWINTCLR	5.10
+0x70 to +0x8F	DQMSK	5.11
+0x90 to +0xAF	DQMSKCLR	5.12

Table 5-2 List of DES

Address	Descriptor name	Reference
DESP+0x00	DES0	5.13
DESP+0x04	DES1	5.14
DESP+0x08	DES2	5.15
DESP+0x0C	DES3	5.15
DESP+0x10 -	DES4	5.15
DESP+0x10 -	DES5	5.15
DESP+0x10 -	DES6	5.15

5.2. DESTP Register

The DESTP (Descriptor top address) Register sets the start address of the DES area.

■ Register configuration

Address: +0x00

bit	31	0
Field	DESTP[31:0]	
Attribute	R/W	
Initial value	0x00000000	

■ Register function

The DSTC refers to and updates the DES located at the address of "DESTP + DESP". Set the DES area in a memory area that is readable and writable. Align the DES area to the word boundary. Always write "00" to the lower 2 bits in the DESTP Register. The DES area must be located within a 4096-word (16 KB) area starting from the DESTP. The DESTP Register cannot be set to a value larger than "0xFFFFFFF0".

Set the DESTP Register when the DSTC is in the normal state and is doing initial settings. The value of this register cannot be modified when the DSTC is executing a transfer. After the value of this register has been modified, the DSTC cannot execute any normal transfer.

bit[31:0] DESTP (Descriptor top address)

Access	Function
Writing	A write access to these bits sets the start address of the DES area.
Reading	A read access to these bits reads the value of these bits.

5.3. HWDESP[n] Register

The HWDESP[n] (Hardware DES pointer) Register sets the DESP of the DES that the DSTC refers to at a transfer request of HW channel n.

■ Register configuration

Address: +0x04

bit	31	30	29 16	15 8	7 0
Field	Reserved		HWDESP[13:0]	Reserved	CHANNEL[7:0]
Attribute	R	R	R/W	R	R/W
Initial value	0	0	XXXXXXXXXXXX00	00000000	00000000

■ Register function

Set the HWDESP[n] Register before making an HW transfer request. This register can be accessed only when the DSTC is in the normal state. Settings of an unused HW channel n are not necessary.

The number of the HWDESP[n] Registers corresponds to the number of HW channels. However, there is only one register window that can be seen from the CPU. Access this register as explained below.

- To read the value of the HWDESP[n] Register from the CPU, use an 8-bit (byte) access to write to CHANNEL[7:0] the channel number to be read first. Afterward, read the value of HWDESP[13:0] with a 16-bit (halfword) access.
- To write the value of the HWDESP[n] Register from the CPU, use an 8-bit (byte) access to write to CHANNEL[7:0] the channel number to be written first. Afterward, write a value to HWDESP[13:0] with a 16-bit (halfword) access. If the write access is a 32-bit (word) access, writing a value to CHANNEL[7:0] and writing a value to HWDESP[13:0] can be executed simultaneously.

The DSTC stores the DESP value of the HWDESP[n] Register in HWDESPBUF in Figure 3-5 before using it. If HW Start requests of channel n are made successively, the DSTC uses the DESP value stored in HWDESPBUF, but not the DESP value of the HWDESP[n] Register. Therefore, if the values of the HWDESP[n] Register are modified via the CPU, invalidate the value stored in HWDESPBUF. The DESP value of HWDESPBUF can be invalidated by modifying the value of the RBDIS bit in the CFG Register. For details of the CFG Register, see "5.5 CFG Register".

bit[7:0] CHANNEL[7:0]

Access	Function
Writing	A write access to these bits sets the channel number (n) for the HWDESP[n] to which a read access or a write access is made.
Reading	A read access to these bits reads the value of these bits.

If the DSTC with which a product equipped supports HW-128 channels, always write "0" to CHANNEL[7].

If the DSTC with which a product equipped supports HW-64 channels, always write "00" to CHANNEL[7:6].

bit[29:16] HWDESP[13:0]

Access	Function
Writing	A write access to these bits writes the channel number of HWDESP[n] specified in CHANNEL[7:0].
Reading	A read access to these bits reads the channel number of HWDESP[n] specified in CHANNEL[7:0].

Align DES to the word boundary. Always write "00" to the lower 2 bits in the HWDESP Register. HWDESP cannot be set to a value larger than "0x3FF0".

5.4. CMD Register

The CMD (Command) Register issues a command to the DSTC and reads the state of the DSTC.

■ Register configuration

Address: +0x08

bit	7	6	5	4	3	2	1	0
Field	CMD[7:0]							
Attribute	W	W	W	W	W	W	R	R
Initial value	0	0	0	0	0	0	0	1

■ Register function

A command can be issued to the DSTC by writing a value to the CMD (Command) Register. Use an 8-bit (byte) access to write a value to this register. The 16-bit write access and 32-bit write access to this register are ignored. In addition, the state (normal state, standby state, transition state 1 and transition state 2) of the DSTC can be checked by reading this register.

In each state of the DSTC, some commands can be issued and some cannot. For details, see Table 3-13.

bit[7:0] CMD[7:0]

Write value	Command	Process details
0x04	Standby release command	Instructs the DSTC to return from the standby state to the normal state.
0x08	Standby transition command	Instructs the DSTC to transit from the normal state to the standby state. Clears SWTR:SWST to "0". Negates the SWINT interrupt signal. Clears all HWINT[n] Registers. Negates the HWINT[n] interrupt signal. Negates the ERINT interrupt signal. Set all DQMSK[n] Registers to "1".
0x10	SWCLR command	Clears SWTR:SWST to "0". Negates the SWINT interrupt signal.
0x20	ERCLR command	Clears MONERS:EST to "0". Negates the ERINT interrupt signal. Clears MONERS:DER to "0". Clears MONERS:ESTOP to "0".
0x80	MKCLR command	Clears all DQMSK[n] Registers to "0". (This command is ignored in the transition state 2.)
Value other than those listed above	Writing a value other than those listed above to CMD[7:0] is prohibited. (If a value other than those listed above is written to CMD[7:0] via a write access such as an RMW access, such write access may be ignored.)	

Read value	State of DSTC
0x00	Indicates that the DSTC is in the normal state.
0x01	Indicates that the DSTC is in the standby state. (Initial value)
0x02	Indicates that the DSTC is in the transition state 1 (transiting from the standby state to the normal state).
0x03	Indicates that the DSTC is in the transition state 2 (transiting from the normal state to the standby state).
Value other than those listed above	The CMD Register never reads any value other than the above.

5.5. CFG Register

The CFG (configuration) Register sets operation functions of the DSTC.

■ Register configuration

Address: +0x09

bit	15	14	13	12	11	10	9	8
Field	Reserved	SWPR[2:0]			ESTE	RBDIS	ERINTE	SWINTE
Attribute	R	R/W			R/W	R/W	R/W	R/W
Initial value	0	100			0	0	0	0

■ Register function

The CFG (configuration) Register sets operation functions of the DSTC. Use an 8-bit (byte) access to write a value to this register. The 16-bit write access and 32-bit write access to this register are ignored. When the DSTC is in the normal state, no write access can be made to this register.

bit[8] SWINTE (Software interrupt enable)

Access	Function
Writing "0"	Disables the SWINT interrupt. (Initial value) If SWTR:SWST has been set to "1", the DSTC does not generate the SWINT interrupt.
Writing "1"	Enables the SWINT interrupt. If SWTR:SWST has been set to "1", the DSTC generates the SWINT interrupt.
Reading	A read access to this bit reads the value of this bit.

bit[9] ERINTE (Error interrupt enable)

Access	Function
Writing "0"	Disables the ERINT interrupt. (Initial value) If MONERS:EST has been set to "000", "001", "010", "100", "101", "110" or "111", the DSTC does not generate the ERINT interrupt.
Writing "1"	Enables the SWINT interrupt. If MONERS:EST has been set to "000", "001", "010", "100", "101", "110" or "111", the DSTC generates the ERINT interrupt.
Reading	A read access to this bit reads the value of this bit.

bit[10] RBDIS (Read skip buffer disable)

Access	Function
Writing "0"	Enables the read skip buffer function of the DES. (Initial value) The HWDESP[n] reference skip function of HWDESPBUF is enabled.
Writing "1"	Disables the read skip buffer function. The HWDESP[n] reference skip function of HWDESPBUF is disabled.
Reading	A read access to this bit reads the value of this bit.

If the RBDIS bit is set to "0", the read skip buffer function and HWDESPBUF function shown in Figure 3-5 are enabled. Accordingly, the DSTC skips referring to the DES and HWDESP[n] Register on the memory and, in turn, the processing speed of the DSTC increases. Nonetheless, the DSTC may not be able to recognize a change in the value of DV of DES0 by the CPU or the modification of the value of the HWDESP[n] Register. Therefore, if the value of DV of DES0 has been changed or the value of the HWDESP[n] Register has been modified, write "1" to the RBDIS bit. If the RBDIS bit is set to "1", the DSTC does not use the read skip buffer function or the HWDESPBUF function, but it operates referring to the DES value on the memory and the value of the HWDESP[n] Register. After the DSTC has executed processes with updated values of the DES and the HWDESP[n] Register, the buffer function can be enabled again by writing "0" to the RBDIS Register.

bit[11] ESTE (Error stop enable)

Access	Function
Writing "0"	The DSTC does not enter the error stop state even when a transfer error occurs. (Initial value) If there is another transfer request, the DSTC starts the transfer for that request.
Writing "1"	The DSTC enters the error stop state even when a transfer error occurs. If there is another transfer request, the DSTC holds the start of the transfer for that request.
Reading	A read access to this bit reads the value of this bit.

bit[12:14] SWPR[2:0] (Software transfer priority)

In the arbitration of Arbiter 2, if the SW transfer request conflicts with the HW transfer request, Arbiter 2 specifies the probability of the SW transfer acquiring the transfer right. The value of the SWPR bits can be modified even when the DSTC is executing a transfer. After the value of the SWPR bits in the CFG Register has been modified, it is applied from the next SW Start Trigger.

Access	Function
Writing "000"	Sets the priority of the SW transfer to the highest priority. (If an SW transfer request is made while an HW transfer is in progress, the SW transfer starts after the HW transfer has ended.)
Writing "001"	Sets the probability of the SW transfer acquiring the transfer right to 1/2.
Writing "010"	Sets the probability of the SW transfer acquiring the transfer right to 1/3.
Writing "011"	Sets the probability of the SW transfer acquiring the transfer right to 1/7.
Writing "100"	Sets the probability of the SW transfer acquiring the transfer right to 1/15. (Initial value)
Writing "101"	Sets the probability of the SW transfer acquiring the transfer right to 1/31.
Writing "110"	Sets the probability of the SW transfer acquiring the transfer right to 1/63.
Writing "111"	Sets the priority of the SW transfer to the lowest priority. (The SW transfer starts only when there is no HW transfer request.)
Reading	A read access to these bits reads the value of these bits.

bit[15] Reserved

Write "0" in a write access to this bit. In a read access to this bit, "0" is read out.

5.6. SWTR Register

The SWTR (Software trigger) Register issues the Start Trigger of the SW Start transfer.

■ Register configuration

Address: +0x0A

bit	31	30	29	16
Field	SWST	SWREQ	SWDESP[13:0]	
Attribute	R	R	R/W	
Initial value	0	0	00000000000000	

■ Register function

The SWTR (Software trigger) Register issues the Start Trigger of the SW Start transfer if a write access to this register is made. Use the 16-bit (halfword) access to write a value to this register. The 32-bit (word) write access is ignored. If the SW Start transfer instruction has been executed, and that transfer has not ended (SWREQ ≠ 0) or the DSTC is not in the normal state (CMD ≠ 00) or the DSTC is in the error stop state (ESTOP ≠ 0), the DSTC ignores the write access from the CPU and also the new SW Start transfer request.

bit[29:16] SWDESP[13:0] (Software DES pointer)

Write the value of DESP to be started. The DSTC transfers data to the DES area of DESTP+SWDEP. If a Chain Start is executed during the SW Start transfer, SWDESP is updated by the DSTC to the value of DESP used in the Chain Start. Align DES to the word boundary. Always write "00" to the lower 2 bits in SWDESP. SWDESP cannot be set to a value larger than "0x3FF0".

Access	Function
Writing	A write access to these bits specifies the DESP for transfer to be started by the SW Start.
Reading	These bits indicate the DESP for SW Start transfer that is in progress or that has been ended.

bit[30] SWREQ (Software request)

The SWREQ bit is a read-only bit indicating whether the execution of the SW Start transfer is pending, or the SW Start transfer as well as the Chain Start transfer are being executed. The value written to this bit is ignored. A write access (Start Trigger) to the SWTR Register sets the SWREQ bit to "1". If the SW Start transfer ends normally, abnormally, or is waiting for a Start Trigger, SWREQ is reset to "0".

Access	Function
Writing	The value written to this bit is meaningless.
Reading "0"	Indicates that the SW transfer started has not ended normally.
Reading "1"	Indicates that the SW transfer started has ended normally.

bit[31] SWST (Software status)

The SWST bit is a read-only bit for sending the SW transfer end notification to the CPU. The interrupt flag set is specified in the DES started by the SW Start. If the SW transfer ends normally, SWST is set to "1". SWST is cleared to "0" by the SWCLR command, the standby transition command or the write access to the SWTR Register. In the case of (CFG:SWINTE = 1)&&(SWTR:SWST == 1), the SWINT interrupt signal is asserted.

Access	Function
Writing	The value written to this bit is meaningless.
Reading "0"	Indicates that the SW transfer started has not ended normally.
Reading "1"	Indicates that the SW transfer started has ended normally.

5.7. MONERS Register

The MONERS Register shows details of a transfer error that has occurred.

■ Register configuration

Address: +0x0C

bit	31	30	29	28	27	26	25	24
Field	Reserved		EDESP[13:8]					
Attribute	R	R	R	R	R	R	R	R
Initial value	0	0	X	X	X	X	X	X

bit	23	22	21	20	19	18	17	16
Field	EDESP[7:0]							
Attribute	R	R	R	R	R	R	R	R
Initial value	X	X	X	X	X	X	0	0

bit	15	14	13	12	11	10	9	8
Field	ECH[7:0]							
Attribute	R	R	R	R	R	R	R	R
Initial value	X	X	X	X	X	X	X	X

bit	7	6	5	4	3	2	1	0
Field	Reserved	EHS	Reserved	ESTOP	DER	EST[2:0]		
Attribute	R	R	R	R	R	R		
Initial value	X	X	X	0	0	000		

■ Register function

The MONERS Register is a read-only register. The write access to this register is ignored. If a transfer error occurs, details of that error are recorded in the MONERS Register. Details of an error can be checked by referring to the MONERS Register. If MONER:EST indicates that an error has occurred, the DSTC can generate an ERINT interrupt. If the ERINTE bit is set to "1", the ERINT interrupt signal for the NVIC is asserted. The values of the MONERS Register and the ERINT interrupt can be cleared by issuing an ERCLR command to the CMD Register.

bit[2:0] EST[2:0] (Error status)

EST indicates details of an error that has occurred. In the case of EST ≠ 000, even if a new transfer error occurs, DESP keeps details of the previous transfer error. ESTOP is cleared to "000" by the ERCLR command.

Access	Function
Writing	Writing a value to DESP causes no operation to be executed.
Reading "000"	No error has occurred. (initial value)
Reading "001"	Source access error
Reading "010"	Destination access error
Reading "011"	A transfer has been stopped compulsorily.
Reading "100"	DES access error
Reading "101"	DES open error
Reading a value other than those listed above	Undefined

bit[3] DER (Double error)

The DER bit indicates whether a double error has occurred. With the EST[2:0] bits set to a value other than "000" and the DER bit set to "0", if a new transfer error occurs, the DER bit is set to "1". The DER bit is cleared to "0" by the ERCLR command.

Access	Function
Writing	Writing a value to DESP causes no operation to be executed.
Reading "0"	Indicates that no double error has occurred. (Initial value)
Reading "1"	Indicates that a double error has occurred.

bit[4] ESTOP (Error stop)

The ESTOP bit indicates that the DSTC is in the error stop state. With CFG:ESTE set to "1", if a transfer error occurs, the ESTOP bit is set to "1". In the error stop state, the transfer start of the DSTC is held. The ESTOP bit is cleared to "0" by the ERCLR command. If this bit is cleared to "0", the transfer held starts. If this bit is "1", a write access to the SWTR Register is ignored.

Access	Function
Writing	Writing a value to DESP causes no operation to be executed.
Reading "0"	Indicates that the DSTC is not in the error stop state. (Initial value)
Reading "1"	Indicates that the DSTC is in the error stop state.

bit[5] Reserved

The read value is indeterminate. The value written to this bit is ignored.

bit[6] EHS (Error hardware software)

The EHS bit indicates whether the DES that has caused an error has been started by the HW Start or by the SW Start. In the case of EST ≠ 000, even if a new transfer error occurs, DESP keeps details of the previous transfer error. In the case of EST = 000, the value of DESP is undefined.

Access	Function
Writing	Writing a value to this bit does not cause any operation to be executed.
Reading "0"	An error has occurred in a transfer started by the SW Start or by the Chain Start in that SW Start.
Reading "1"	An error has occurred in a transfer started by the HW Start or by the Chain Start in that HW Start.

bit[7] Reserved

The read value is indeterminate. The value written to this bit is ignored.

bit[15:8] ECH[7:0] (Error hardware channel)

If the EST[2:0] bits are not "000" and the EHS bit is "1", the ECH[7:0] bits indicate the number of the HW Start channel that has caused an error. In the case of EST ≠ 000, even if a new transfer error occurs, the ECH[7:0] bits keep the HW Start channel number of the previous transfer error. If the EST[2:0] bits are "000" or the EHS bit is "0", the value of the ECH[7:0] bits is indeterminate.

Access	Function
Writing	Writing a value to these bits does not cause any operation to be executed.
Reading	The ECH[7:0] bits indicate the HW channel number if the DES that has caused a transfer error was started by the HW Start.

bit[29:16] EDESP [13:0](Error DES pointer)

The EDESP[13:0] bits indicate the DESP of the DES that has caused a transfer error. In the case of EST ≠ 000, even if a new transfer error occurs, the EDESP[13:0] bits keep the DESP of the DES of the previous transfer error. In the case of EST = 000, the value of the EDESP[13:0] bits is indeterminate. The EDESP[1:0] bits always read "0".

Access	Function
Writing	Writing a value to these bits does not cause any operation to be executed.
Reading	The EDESP[13:0] bits indicate the DESP of the DES that has caused a transfer error.

bit[31:30] Reserved

The read value is "00". The value written to these bits is ignored.

5.8. DREQENB[n] Register

The DREQENB[n] Register determines whether HW channel n is used.

■ Register configuration

Address		
0x10	Field	DREQENB[31:0]
+0x14	Field	DREQENB[63:32]
+0x18	Field	DREQENB[95:64]
+0x1C	Field	DREQENB[127:96]
+0x20	Field	DREQENB[159:128]
+0x24	Field	DREQENB[191:160]
+0x28	Field	DREQENB[223:192]
+0x2C	Field	DREQENB[255:224]
Attribute (applicable to all areas)		R/W
Initial value (applicable to all areas)		0x00000000

■ Register function

The DREQENB[n] (DMA request enable) Register determines whether HW channel n is used in the initial settings. When the DSTC is in the normal state, no write access can be made to this register.

Write "1" to the DREQENB[n] Register to use HW channel n. Write "0" to the DREQENB[n] Register to not use HW channel n. If the DREQENB[n] Register is "0", the interrupt signal (DREQ[n]) of a peripheral connected to the DSTC is ignored. The value of the DREQENB[n] Register cannot be modified by the DSTC.

The value of the DREQENB[n] Register determines which of the interrupt signal from a peripheral and HWINT[n] from the DSTC is selected as an interrupt signal connected to the NVIC. For its details, see "2 DSTC Operations Overview and DSTC System Configuration".

bit[255:0] DREQENB[255:0] (DMA request enable)

Access	Function
Writing "0"	Disables the DREQ signal from the peripheral. (Initial value)
Writing "1"	Enables the DREQ signal from the peripheral.
Reading	A read access to these bits reads the value of these bits.

If the DSTC installed in a product supports HW-128 channels, the DREQENB[255:128] bits are a reserved area whose value is fixed at "0".

If the DSTC installed in a product supports HW-64 channels, the DREQENB[255:64] bits are a reserved area whose value is fixed at "0".

5.9. HWINT[n] Register

The HWINT[n] Register sends the HW transfer end notification to the CPU.

■ Register configuration

Address		
+0x30	Field	HWINT[31:0]
+0x34	Field	HWINT[63:32]
+0x38	Field	HWINT[95:64]
+0x3C	Field	HWINT[127:96]
+0x40	Field	HWINT[159:128]
+0x44	Field	HWINT[191:160]
+0x48	Field	HWINT[223:192]
+0x4C	Field	HWINT[255:224]
Attribute (applicable to all areas)		R
Initial value (applicable to all areas)		0x0000000000000000

■ Register function

The HWINT[n] (Hardware transfer interrupt) Register is a read-only register for sending the HW transfer end notification to the CPU. The write access to this register is ignored.

If the interrupt flag set is specified in the DES started by the HW Start, or CHRS in the DES started by the Chain Start after the DES started by the HW Start, the HWINT[n] Register is set to "1". The HWINT[n] Register can be cleared to "0" by writing "1" to the HWINTCLR[n] Register or issuing the standby transition command. If the HWINT[n] Register is set to "1", the interrupt signal for the NVIC is asserted.

bit[255:0] HWINT[255:0] (Hardware transfer interrupt)

Access	Function
Writing	Writing a value to DESP causes no operation to be executed.
Reading "0"	Indicates that the HW transfer started has not ended normally.
Reading "1"	Indicates that the HW transfer started has ended normally.

If the DSTC installed in a product supports HW-128 channels, the HWINT[255:128] bits are a reserved area whose value is fixed at "0".

If the DSTC installed in a product supports HW-64 channels, the HWINT[255:64] bits are a reserved area whose value is fixed at "0".

5.10. HWINTCLR[n] Register

The HWINTCLR[n] Register is a register for clearing the HWINT[n] Register.

■ Register configuration

Address		
+0x50	Field	HWINTCLR[31:0]
+0x54	Field	HWINTCLR[63:32]
+0x58	Field	HWINTCLR[95:64]
+0x5C	Field	HWINTCLR[127:96]
+0x60	Field	HWINTCLR[159:128]
+0x64	Field	HWINTCLR[191:160]
+0x68	Field	HWINTCLR[223:192]
+0x6C	Field	HWINTCLR[255:224]
Attribute (applicable to all areas)		W
Initial value (applicable to all areas)		0x0000000000000000

■ Register function

The HWINTCLR[n] Register is a write-only register for clearing the HWINT[n] Register from the CPU. When the DSTC is not in the normal state, no write access can be made to this register.

Writing "1" to this register can clear the HWINT[n] Register to "0". Writing "0" to this register is ignored. The read value is always "0".

bit[255:0] HWINTCLR[255:0] (Hardware transfer interrupt clear)

Access	Function
Writing "0"	Causes no operation to be executed.
Writing "1"	Clears the HWINT[n] Register to "0".
Reading	All bits in this register always read "0".

If the DSTC installed in a product supports HW-128 channels, the HWINTCLR[255:128] bits are a reserved area whose value is fixed at "0".

If the DSTC installed in a product supports HW-64 channels, the HWINTCLR[255:64] bits are a reserved area whose value is fixed at "0".

5.11. DQMSK[n] Register

The DQMSK[n] Register indicates whether the HW Start transfer request is being suppressed.

■ Register configuration

Address		
+0x70	Field	DQMSK[31:0]
+0x74	Field	DQMSK[63:32]
+0x78	Field	DQMSK[95:64]
+0x7C	Field	DQMSK[127:96]
+0x80	Field	DQMSK[159:128]
+0x84	Field	DQMSK[191:160]
+0x88	Field	DQMSK[223:192]
+0x8C	Field	DQMSK[255:224]
Attribute (applicable to all areas)		R
Initial value (applicable to all areas)		0x00000000

■ Register function

The DQMSK[n] Register is a read-only register. The write access to this register is ignored. That this register is "1" indicates the HW Start transfer request (DREQ[n]) to the DSTC is being suppressed. If one of the following conditions is met, the DSTC sets DQMSK[n] to "1" and suppresses transfer requests.

- A transfer error has occurred at a transfer on HW channel n.
- The CPU has issued a standby transition command to the CMD Register.
- DMSET in the DES for the transfer on HW channel n is "1" and the DSTC has executed a DES close process.

If one of the following conditions is met, the DSTC clears DQMSK[n] to "0" and releases the suppression of transfer requests.

- "1" has been written to the DQMSKCLR[n] Register.
- The CPU has issued a standby transition command to the CMD Register.

bit[255:0] DQMSK[255:0] (DMA request mask)

Access	Function
Writing	Causes no operation to be executed.
Reading "0"	Indicates that the DREQ[n] signal from the peripheral is not being suppressed.
Reading "1"	Indicates that the DREQ[n] signal from the peripheral is being suppressed.

If the DSTC installed in a product supports HW-128 channels, the DQMSK[255:128] bits are a reserved area whose value is fixed at "0".

If the DSTC installed in a product supports HW-64 channels, the DQMSK[255:64] bits are a reserved area whose value is fixed at "0".

5.12. DQMSKCLR[n] Register

The DQMSKCLR[n] Register is a register for clearing the DQMSK[n] Register.

■ Register configuration

Address		
+0x90	Field	DQMSKCLR[31:0]
+0x94	Field	DQMSKCLR[63:32]
+0x98	Field	DQMSKCLR[95:64]
+0x9C	Field	DQMSKCLR[127:96]
+0xA0	Field	DQMSKCLR[159:128]
+0xA4	Field	DQMSKCLR[191:160]
+0xA8	Field	DQMSKCLR[223:192]
+0xAC	Field	DQMSKCLR[255:224]
Attribute (applicable to all areas)		W
Initial value (applicable to all areas)		0x00000000

■ Register function

The DQMSKCLR[n] Register is a write-only register. When the DSTC is in the standby state, no write access can be made to this register.

If "1" is written to this register, the DQMSK[n] Register is cleared to "0". Clearing the DQMSK[n] Register makes a suppressed HW transfer start immediately. Therefore, complete the setup of the peripheral for that suppressed transfer and the setup of the DES before clearing the DQMSK[n] Register.

bit[255:0] DQMSKCLR[255:0] (DMA request mask clear)

Access	Function
Writing "0"	No operation is executed.
Writing "1"	Clears the DQMSK[n] Register to "0".
Reading	All bits in this register always read "0".

If the DSTC installed in a product supports HW-128 channels, the DQMSKCLR[255:128] bits are a reserved area whose value is fixed at "0".

If the DSTC installed in a product supports HW-64 channels, the DQMSKCLR[255:64] bits are a reserved area whose value is fixed at "0".

5.13. Descriptor 0 (DES0)

This section explains details of Descriptor 0 (DES0). DES0 sets the basic settings of a transfer.

■ Descriptor configuration

Address: DESP + 0x00

bit	31	30	29	28	27	26	25	24
Field	PCHK[3:0]				Reserved		ACK[1:0]	
C attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
D attribute	R	R	R	R	R	R	R	R

bit	23	22	21	20	19	18	17	16
Field	CHLK	DMSET	CHRS[5:0]					
C attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
D attribute	R	R	R	R	R	R	R	R

bit	15	14	13	12	11	10	9	8
Field	DAC[1:0]			SAC[2:0]			TW	
C attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
D attribute	R	R	R	R	R	R	R	R

bit	7	6	5	4	3	2	1	0
Field	ORL[2:0]			MODE	ST[1:0]		DV[1:0]	
C attribute	R/W	R/W	R/W	R/W	R	R	R/W	R/W
D attribute	R	R	R	R	W	W	R/W	R/W

*Symbols for DES area:
The row C attribute in the table above indicates the attribute of the R/W access from the CPU.
The row D attribute in the table above indicates the attribute of the R/W access from the DSTC.
The DES area is built on the memory. As all initial values are indeterminate, they are omitted.

■ Descriptor function

bit[1:0] DV[1:0] (Descriptor valid)

The DV[1:0] bits specify which of the CPU and the DSTC the ownership of the DES belongs to. The DV[1:0] bits also specify whether a transfer is executed after the DES open process and whether the DES close process is to be executed. In the DES close process, the DV[1:0] bits are updated to "00" by the DSTC.

If DV[1] is set to "1", specific restrictions on reloading the transfer counter and transfer address are added. If DV[1] is set to "1" and the settings of the DES make the values of DES1, DES2 and DES3 not return to their respective values, the DSTC notifies of the CPU a DES open error.

Value	Function
00	The DES ownership belongs to the CPU. No transfer is executed after the DES open process. No DES close process is executed.
01	The DES ownership belongs to the DSTC. A transfer is executed after the DES open process. The DES close process is executed.
10	The DES ownership belongs to the DSTC. No transfer is executed after the DES open process. The DES close process is executed.
11	The DES ownership belongs to the DSTC. A transfer is executed after the DES open process. No DES close process is executed.

bit[3:2] ST[1:0] (Transfer status)

After the transfer specified in a DES has ended, in a DES close process, the DSTC writes the end status value to the ST[1:0] bits. If the DSTC does not execute the DES close process, the value initially set by the CPU remains in the ST[1:0] bits. At the start of a transfer, the value in this area has no effect on the operation of the DSTC.

If a transfer ends abnormally due to a DES access error or a DES open error, the DSTC does not execute a DES close process and does not write data to the ST[1:0] bits either because the DSTC cannot access the DES area. Such error notifications are executed according to the MONERS Register of the DSTC but not ST of the DES.

Value	Function
00	The transfer has ended normally.
01	The transfer has ended abnormally because an error occurred at a transfer source access.
10	The transfer has ended abnormally because an error occurred at a transfer destination access.
11	The transfer has been ended compulsorily by a standby transition instruction, which is considered as abnormal transfer end.

bit[4] MODE (Transfer mode)

The MODE bit selects a transfer mode. In mode 0, the DSTC executes transfers for IIN times of data transfers for one Start Trigger. In mode 1, the DSTC executes transfers one transfer for one Start Trigger. To make the DSTC execute an HW Start transfer at a Start Trigger from a peripheral that has to have to a handshake with the DSTC at every data transfer, use mode 1. The DSTC does not modify the value of this area.

Value	Function
0	The transfer is to be executed in mode 0.
1	The transfer is to be executed in mode 1.

bit[7:5] ORL[2:0] (Outer reload)

The ORL[2:0] specify whether OuterReload for the transfer number counter of DES1 (ORM/IRM/IIN), the transfer source address of DES2 (SA) and the transfer destination address (DA) of DES3 is executed after ORM×IIN times of transfer have ended. The DES size is determined by the value of ORL. The respective relative addresses from the DESP of DES4 to DES6 vary according to the value of the ORL[2:0] bits. The DSTC does not modify the value of this area.

bit	Value	Function
ORL[0]	0	OuterReload for DES1 is not to be executed. DES4 area is not required.
	1	OuterReload for DES1 is to be executed. DES4 area is not required.
ORL[1]	0	OuterReload for DES2 is not to be executed. DES5 area is not required.
	1	OuterReload for DES2 is to be executed. DES5 area is required.
ORL[2]	0	OuterReload for DES3 is not to be executed. DES6 area is not required.
	1	OuterReload for DES3 is to be executed. DES6 area is required.

bit[9:8] TW[1:0] (Transfer width)

The TW[1:0] specify the data width in a single transfer. The DSTC does not modify the value of this area.

Value	Function
00	8 bits (byte)
01	16 bits (halfword)
10	32 bits (word)
11	Setting prohibited (If the DSTC reads "11" from TW, a DES open error occurs.)

bit[12:10] SAC[2:0] (Source address control)

The SAC[2:0] specify the method of updating the transfer source address during a transfer. The DSTC does not modify the value of this area.

Value	Function
000	The address is increased by 1 TW at every transfer. InnerReload is unavailable.
001	The address is increased by 1 TW at every transfer. InnerReload is available.
010	The address is increased by 2 TW at every transfer. InnerReload is unavailable.
011	The address is increased by 2 TW at every transfer. InnerReload is available.
100	The address is increased by 4 TW at every transfer. InnerReload is unavailable.
101	The transfer address remains unchanged during a transfer.
110	The address is decreased by 1 TW at every transfer. InnerReload is unavailable.
111	The address is decreased by 1 TW at every transfer. InnerReload is available.

bit[15:13] DAC[2:0] (Destination address control)

The DAC[2:0] specify the method of updating the transfer destination address during a transfer. The DSTC does not modify the value of this area.

Value	Function
000	The address is increased by 1 TW at every transfer. InnerReload is unavailable.
001	The address is increased by 1 TW at every transfer. InnerReload is available.
010	The address is increased by 2 TW at every transfer. InnerReload is unavailable.
011	The address is increased by 2 TW at every transfer. InnerReload is available.
100	The address is increased by 4 TW at every transfer. InnerReload is unavailable.
101	The transfer address remains unchanged during a transfer.
110	The address is decreased by 1 TW at every transfer. InnerReload is unavailable.
111	The address is decreased by 1 TW at every transfer. InnerReload is available.

bit[21:16] CHRS[5:0] (Chain & return status)

The CHRS[5:0] bits specify the process to be executed after specified times of transfer have been executed.

bit	Condition for selection	Value	Function
CHRS[1:0]	IRM ≠ 1 ORM: ignore	00	No interrupt flag is set. There is no Chain Start.
		01	An interrupt flag has been set. There is no Chain Start.
		10	No interrupt flag is set. There is a Chain Start in the succeeding DES.
		11	No interrupt flag is set. There is a Chain Start in the current DES.
CHRS[3:2]	IRM = 1 ORM ≠ 1	00	No interrupt flag is set. There is no Chain Start.
		01	An interrupt flag has been set. There is no Chain Start.
		10	No interrupt flag is set. There is a Chain Start in the succeeding DES.
		11	No interrupt flag is set. There is a Chain Start in the current DES.

CHRS[5:4]	IRM = 1
	ORM = 1
		00	No interrupt flag is set. There is no Chain Start.

		01	An interrupt flag has been set. There is no Chain Start.
		10	No interrupt flag is set. There is a Chain Start in the succeeding DES.
		11	11: Setting prohibited (A DES open error occurs.)

The status of the transfer number counter determines which of CHRS[5:4], CHRS[3:2] and CHRS[1:0] the DSTC follows when executing the next process after the current DES. If there is an interrupt flag set instruction, an SW Start Trigger, and a Chain Start Trigger after that SW Start Trigger set the SWST bit to the SWTR register to "1". An HW Start Trigger, and a Chain Start Trigger after that HW Start Trigger set the HWINT[n] register to "1". When MODE is "0", as CHRS[1:0] is meaningless, write "00" to it. If MODE is "0" and CHRS[1:0] is not "00", a DES open error occurs. The DSTC does not modify the value of this area.

bit[22] DMSET (DREQ mask set)

For a DES to be started by the HW Start directly from channel n of a peripheral, and a DES to be started by the Chain Start from the DES mentioned before, with the DMSET bit set to "1", if the DES close process is not executed, the bit corresponding to that DES in the DQMSK[n] Register is set to "1". The DSTC does not modify the value of this area.

Value	Function
0	The DQMSK[n] Register is not set to "1" when the DES close process for an HW Start transfer is executed.
1	The DQMSK[n] Register is set to "1" when the DES close process for an HW Start transfer is executed.

bit[23] CHLK (Chain lock)

The CHLK bit specifies whether to execute the next transfer started by the Chain Start immediately after the current transfer (Chain Lock) or to enable other transfers to be executed before the next transfer started by the Chain Start. With the CHLK bit set to "1", if any of CHRS[5], CHRS[3] and CHRS[1] is not "1" (Chain Start selected), a DES open error occurs. The DSTC does not modify the value of this area.

Value	Function
0	After the current transfer, other transfers can be executed before the Chain Start transfer.
1	The Chain Start transfer is executed immediately after the current transfer.

bit[25:24] ACK[1:0] (Acknowledge)

The ACK[1:0] bits set the value for adjusting the timing of DSTC outputting the DMA transfer request acknowledge signal to a peripheral device when the HW Start transfer is used. If the HW Start transfer is used, set the ACK[1:0] bits to "01" for a DES to be directly started by the HW Start from a peripheral device. For the DES started by the Chain Start from the HW Start transfer, the DES used in the SW Start transfer, and the DES started by the Chain Start from the SW Start transfer, set the ACK[1:0] bits to "00". The DSTC does not modify the value of this area.

Value	Function
00	The DSTC does not output the DMA transfer acknowledge signal to the peripheral connected to the DSTC.
01	The DSTC outputs the DMA transfer acknowledge signal to the peripheral connected to the DSTC.
10, 11	Reserved

bit[27:26] Reserved

Write "00" to this area. If "00" is not written to this area, the DSTC notifies the CPU of a DES open error. The DSTC does not modify the value of this area.

bit[31:28] PCHK[3:0] (Parity check)

The PCHK[3:0] bits set the parity (to be called "equation" below) of the DES0 area.

PCHK[3:0] != (DES0[27:24] ^ DES0[23:20] ^ DES0[19:16] ^ DES0[15:12] ^ DES0[11:8] ^ DES0[7:4])

The CPU calculates the parity value while building the DES. The DSTC checks the consistency between

PCHK[3:0] and the value of DES0 area. If a parity error occurs, the DSTC notifies the CPU of a DES open error. The DSTC does not modify the value of this area. The operation target of PCHK[3:0] is the area of DES0[27:4], which the DSTC does not modify. A change in the value of DES0[3:0] does not affect the value of PCHK[3:0].

5.14. Descriptor 1 (DES1)

This section explains details of Descriptor 1 (DES1). DES1 sets the number of transfers. The configuration and functions of DES1 area in mode 0 are different from those in mode 1.

■ Descriptor configuration (in mode 0)

Address: DESP + 0x04

bit	31 16	15 0
Field	ORM[15:0]	IIN[15:0]
C attribute	R/W	R/W
D attribute	R/W	R/W

■ Descriptor function (in mode 0)

bit[15:0] IIN[15:0] (Inner loop initial)

The IIN[15:0] bits specify the initial value of the inner loop counter in the transfer number counter. They can be set to a value in the range of "1" to "65536" inclusive. Setting the IIN[15:0] bits to "0x0000" is equivalent to setting them to "65536". The DSTC imports the value of IIN and uses it as the internal loop counter remain value (IRM). Therefore, in mode 0, it is not necessary to specify the IRM.

The DSTC does not modify the value of this area during a transfer. If OuterReload of DES1 is enabled (ORL[0] = 1), the DSTC copies the value of DES[15:0] to IIN.

bit[31:16] ORM[15:0] (Outer loop remain)

The ORM[15:0] bits specify the remain value of the outer loop counter in the transfer number counter. They can be set to a value in the range of "1" to "65536" inclusive. Setting the ORM[15:0] bits to "0x0000" is equivalent to setting them to "65536".

The DSTC decreases the value of ORM before writing back it to the DES. The DSTC stores "0x0001" in ORM at the end of a transfer. If OuterReload of DES1 is enabled, the DSTC copies the value of DES4[31:16] to ORM after the final transfer. If the transfer ends in the form of error, the DSTC stores the value appearing before the end of the transfer, and in turn, it is necessary to initialize the ORM[15:0] bits via the CPU.

In mode 0, set the number of transfers to a value within the following range. If "00" is not written to this area, the DSTC notifies the CPU of a DES open error.

If ORM = 65536, IIN < 0x2000
If ORM ≥ 0x8000, IIN < 0x4000
If ORM ≥ 0x4000, IIN < 0x8000
If ORM ≥ 0x2000, IIN < 65536
(If ORM is smaller than 0x2000, there is no limit on the value of IIN.)

■ Descriptor configuration (in mode 1)

Address: DESP + 0x04

bit	31 … 16	15 … 8	7 … 0
Field	ORM[15:0]	IRM[7:0]	IIN[7:0]
C attribute	R/W	R/W	R/W
D attribute	R/W	R/W	R/W

■ Descriptor function (in mode 1)

bit[7:0] IIN[7:0] (Inner loop initial)

The IIN[15:0] bits specify the initial value of the inner loop counter in the transfer number counter. They can be set to a value in the range of "1" to "256" inclusive. Setting the IIN[7:0] bits to "0x00" is equivalent to setting them to "256".

The DSTC does not modify the value of this area during a transfer. If OuterReload of DES1 is enabled, the DSTC copies the value of DES[7:0] to the IIN[7:0] bits after the final transfer.

bit[15:8] IRM[7:0] (Inner loop remain)

The IRM[7:0] bits specify the remain value of the inner loop counter in the transfer number counter. Set the IRM[7:0] bits to the same value as the IIN[7:0] bits.

The DSTC decreases the value of the IRM[7:0] bits before writing back it to the DES. The DSTC stores "0x01" in IRM at the end of a transfer. If OuterReload of DES1 is enabled, the DSTC copies the value of DES4[15:8] to the IRM[7:0] bits after the final transfer. If the transfer ends in the form of error, the DSTC stores the value appearing before the end of the transfer, and in turn, it is necessary to initialize the IRM[7:0] bits via the CPU. If the DSTC detects that the value of the IRM[7:0] bits is larger than the value of the IIN[7:0] bits, it notifies the system of a DES open error.

bit[31:16] ORM[15:0] (Outer loop remain)

The ORM[15:0] bits specify the remain value of the outer loop counter in the transfer number counter. They can be set to a value in the range of "1" to "65536" inclusive. Setting the ORM[15:0] bits to "0x0000" is equivalent to setting them to "65536".

The DSTC decreases the value of the ORM[15:0] bits before writing back it to the DES. The DSTC stores "0x0001" in the ORM[15:0] bits at the end of a transfer. If OuterReload of DES1 is enabled, the DSTC copies the value of DES4[31:16] to the IRM[7:0] bits after the final transfer. If the transfer ends in the form of error, the DSTC stores the value appearing before the end of the transfer, and in turn, it is necessary to initialize the ORM[15:0] bits via the CPU.

5.15. Descriptor 2 to Descriptor 6 (DES2 to DES6)

This section explains details of Descriptor 2 to Descriptor 6 (DES2 to DES6).

■ DES2 Descriptor configuration

Address: DESP + 0x08

bit	31	0
Field	SA[31:0]	
C attribute	R/W	
D attribute	R/W	

■ DES2 Descriptor function

bit[31:0] SA[31:0] (Source address)

The SA[31:0] bits set the transfer source address. The SA[31:0] bits cannot be set to a value unaligned to the data size specified in DES0:TW. The DSTC updates the value of DES3 when writing back values to the DES. If OuterReload of DES2 is enabled (ORL[1] = 1), the DSTC copies the value of DES5 to DES2. Pay attention to the value at the end of a transfer if InnerReload/OuterReload of the transfer source address is not enabled. If the transfer ends in the form of error, the DSTC stores the value appearing before the end of the transfer, and in turn, it is necessary to initialize the SA[31:0] bits via the CPU.

■ DES3 Descriptor configuration

Address: DESP + 0x0C

bit	31	0
Field	DA[31:0]	
C attribute	R/W	
D attribute	R/W	

■ DES3 Descriptor function

bit[31:0] DA[31:0] (Destination Address)

The DA[31:0] bits set the transfer destination address. The DA[31:0] bits cannot be set to a value unaligned to the data size specified in DES0:TW. The DSTC updates the value of DES3 when writing back values to the DES. If OuterReload of DES3 is enabled (ORL[2] = 1), the DSTC copies the value of DES6 to DES3. Pay attention to the value at the end of a transfer if InnerReload/OuterReload of the transfer destination address is not enabled. If the transfer ends in the form of error, the DSTC stores the value appearing before the end of the transfer, and in turn, it is necessary to initialize the DA[31:0] bits via the CPU.

■ DES4 Descriptor configuration

Address: DESP + 0x10 (ORL[2:0] = xx1)

bit	31	0
Field	DES4[31:0]	
C attribute	R/W	
D attribute	R	

■ DES4 Descriptor function

bit[31:0] DES4[31:0] (Descriptor 4)

DES4 sets the value to be loaded to DES1 (number of transfers) in OuterReload. Set DES4 to the same value as DES1 under the same configuration. The DSTC does not modify the value of this area.

■ DES5 Descriptor configuration

Address: DESP + 0x10 (ORL[2:0] = x10)
Address: DESP + 0x14 (ORL[2:0] = x11)

bit	31 … 0
Field	DES5[31:0]
C attribute	R/W
D attribute	R

■ DES5 Descriptor function

bit[31:0] DES5[31:0] (Descriptor 5)

DES5 sets the setting to be loaded to DES2 (transfer source start address) in OuterReload. Set DES5 to the same value as DES2. The DSTC does not modify the value of this area.

■ DES6 Descriptor configuration

Address: DESP + 0x10 (ORL[2:0] = 100)
Address: DESP + 0x14 (ORL[2:0] = 110, 101)
Address: DESP + 0x18 (ORL[2:0] = 111)

bit	31 … 0
Field	DES6[31:0]
C attribute	R/W
D attribute	R

■ DES6 Descriptor function

bit[31:0] DES6[31:0] (Descriptor 6)

DES6 sets the setting to be loaded to DES3 (transfer destination start address) in OuterReload. Set DES6 to the same value as DES3. The DSTC does not modify the value of this area.

CHAPTER: I/O Port

This chapter explains the I/O port.

CODE: 9BFGPIO_FM4-E01.0

1. Overview

This section provides an overview of the I/O port.

The I/O port of this series provides the following features.

- The I/O port of this series shares the following functions.
  - GPIO
    General-purpose I/O ports, which can read an input level and set an output level from the CPU.
  - Peripheral input/output
    Digital input/output signal ports of peripheral functions.
  - Special I/O ports
    - Analog input port
      An analog input port of an A/D converter and LCD controller.
    - Analog output port
      An analog output port of a D/A converter and LCD controller.
    - USB port
    - Oscillation port
- The followings settings can be made for each pin.
  - You can set whether the I/O port will be used as a GPIO, a digital pin of peripheral functions, or a special pin.
  - You can set whether the I/O port will be used as an input port or an output port.
  - You can enable or disable pull-up.
  - Peripheral functions are assigned to two or more I/O ports with input/output of the same function. You can set to which I/O port the function can be allocated (relocation function).
  - By setting registers, you can set the I/O port to Hi-Z status while the CPU is in standby mode.

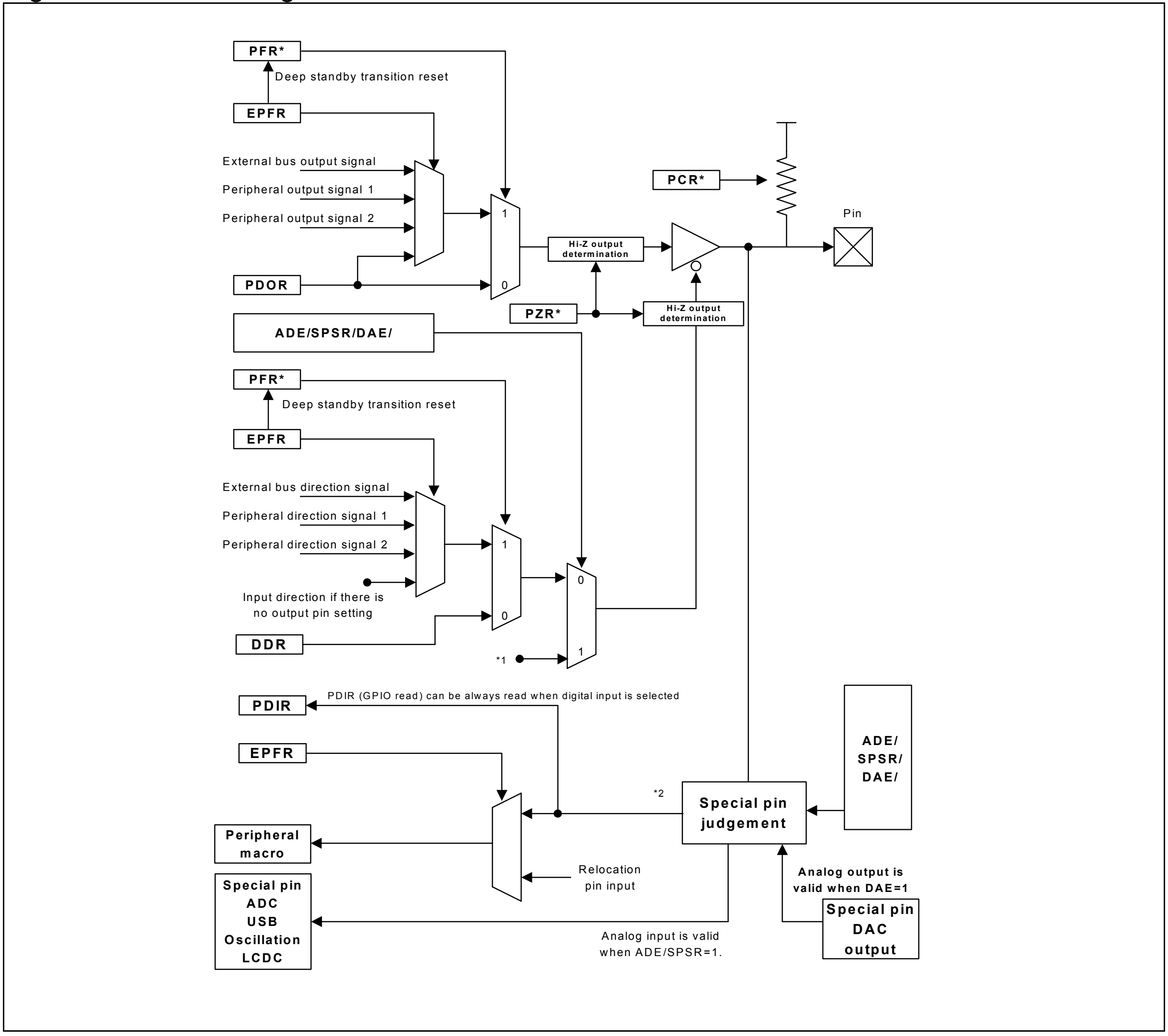
2. Configuration, Block Diagram, and Operation

This section explains the configuration, block diagram, and operation of the I/O port.

■ Configuration of the I/O Port

By setting registers of the I/O port, select Input/Output direction and select GPIO/peripheral. Figure 2-1 shows the details of the I/O port.

Figure 2-1 Block Diagram of the I/O Port


PFR*
Deep standby transition reset
EPFR
External bus output signal
Peripheral output signal 1
Peripheral output signal 2
PDOR
1
0
Hi-Z output determination
PCR*
Pin
PZR*
Hi-Z output determination
ADE/SPSR/DAE/
PFR*
Deep standby transition reset
EPFR
External bus direction signal
Peripheral direction signal 1
Peripheral direction signal 2
Input direction if there is no output pin setting
1
0
0
1
*1
DDR
PDIR
PDIR (GPIO read) can be always read when digital input is selected
EPFR
*2
ADE/ SPSR/ DAE/
Special pin judgement
Peripheral macro
Relocation pin input
Analog output is valid when DAE=1
Special pin ADC USB Oscillation LCDC
Analog input is valid when ADE/SPSR=1.
Special pin DAC output

*1: When one of the followings is set, I/O port is set to input direction.

- ADE/SPSR=1
- DAE=1

*2: When one of the followings is set, the input value is fixed to "0". Otherwise, the pin is set as the digital input pin.

- ADE/SPSR=1
- DAE=1

<Notes>

- USB pin does not have pull-up resistor.
- If it does not have a pull-up resistor, the PCR register setting is null.
- PZR register function is implemented only in some specific pins.
- Only pins described as "PZR register control is enabled" in the remarks column of "■ I/O CIRCUIT TYPE" of the Data Sheet can control this feature.
- PFR0 register is not initialized by deep standby transition reset.
- For details of DAE bit, "5.1. D/A Control Register (DACR)" in "10-bit D/A CONVERTER" in "Analog Macro Part".
- The register settings of PFR, DDR, PDIR, PDOR, PCR, and PZR registers are invalid for ports (P49 to P46) in VBAT Domain. These bits have the same configuration as I/O Ports, but they are set with VBAT Registers (VBPFR, VBDDR, VBDIR, VBDOR, and VBPZR). For details on registers, see Chapter “VBAT Domain”.

Table 2-1 describes register function.

- The PFR, DDR, PDIR, PDOR, and PCR register have 1-bit control register for each I/O port and select a function for the I/O port.
- The ADE register has 1-bit control register for each I/O port which doubles as an analog input pin and selects a function for the I/O port.
- The SPSR register selects a function for the I/O port which doubles as a USB pin or an oscillation pin.
- The EPFR register has control register for each I/O pin of peripheral functions and selects to which I/O port an I/O pin of peripheral functions will be relocated.
- PZR register sets open drain control in pseudo mode by the Hi-Zing I/O port when outputting the High level of a particular pin.
-

Table 2-1 Register Function Descriptions

Register name	Function description
ADE	A register to set whether the I/O port will be used as a special pin (an analog input pin) or a digital input/output pin.
SPSR	A register to set whether the I/O port will be used as a special pin (USB or oscillation) or a digital input/output pin.
PFR	A register to set whether the I/O port will be used as an input/output pin of GPIO function or an input/output pin of peripheral functions.
PCR	A register to set whether a pull-up resistor of the I/O port will be connected or disconnected if the I/O port is used as a digital input pin or a digital bidirectional pin.
DDR	A register to set whether the I/O port will be used as an input pin or an output pin if the I/O port is used as a GPIO function pin. Note: If a pin is selected as an I/O pin of peripheral functions, a setting value is invalid.
PDIR	A register to read the level status of the I/O port. · If the I/O port is used as a digital input pin, it reads input level. · If the I/O port is used as a digital output pin, it reads output level. · If the I/O port is used as an analog input pin, it always reads "0".
PDOR	A register to set output level if the I/O port is used as an output pin of GPIO function. · When "0" is set, it outputs Low level. · When "1" is set, it outputs High level. Note: If a pin is selected as GPIO input or input/output of peripheral functions, a setting value is invalid.
EPFR	A register to select a function for an input/output of peripheral functions and set relocation function. · Setting a peripheral output pin It sets whether to produce output for the I/O port or not. In addition, it can also set to which I/O port a pin of peripheral functions will be relocated for each pin. · Setting a peripheral input pin It can set to which I/O port a pin of peripheral functions will be relocated for each pin. · Setting a peripheral bidirectional pin It can set to which I/O port a pin of peripheral functions will be relocated for each pin
PZR	This register sets open the drain control of the I/O port. · Set the I/O port to Low output when the I/O port is outputting Low level (pull-up disconnection regardless of PCR setting value) · Set open drain control in pseudo mode by setting the I/O port on Hi-Z status when the I/O port outputs High level (pull-up disconnection regardless of PCR setting value) · Set the I/O port on Hi-Z status when the I/O port is used for input (pull-up disconnection regardless of PCR setting value) Note: This function is implemented only in some specific pins. Only pins described as "PZR register control is enabled" in the remarks column of "■ I/O CIRCUIT TYPE" of the Data Sheet can control the open drain.

Table 2-2 lists pin functions which availability depends on selected I/O port functions and register setting values.

Table 2-2 I/O Port Functions and Register Setting Values

I/O Port Function		ADE/ SPSR/ DAE	PFR	DDR	PZR	PCR	EPFR
Available main function	Available sub function						
Special pin (Analog input, Analog output, USB, Oscillation)	N/A	1	-	-	-	Disconnect	*0
GPIO function input pin	Peripheral function input pin	0	0	0	0	Valid	*1
				0	1	Disconnect	
GPIO function output pin	GPIO function input pin (FB) Peripheral function input pin (FB)			1	0	Disconnect	
				1	1	Disconnect	
Peripheral function output pin	GPIO function input pin (FB) Peripheral function input pin (FB)		1	-	0	Disconnect	*2
					1	Disconnect	
Peripheral function bidirectional pin	GPIO function input pin (FB) Peripheral function input pin (FB)				0	Valid	*3
					1	Disconnect	
Peripheral function input pin	GPIO function input pin				0	Valid	*4
					1	Disconnect	

Legends

- : Indicates that a register setting value does not affect pin functions.
Valid : Indicates that a pull-up resistor is disconnected if PCR register value is 0.
Indicates that a pull-up resistor is connected if PCR register value is 1.
Disconnect : Indicates that a pull-up resistor is disconnected regardless of PCR register value.
(FB) : Indicates that an output signal of the I/O port provides feedback and the level of the I/O port can be read from PDIR. The signal can be also used as input for peripheral functions.

*0 : If the input pin of peripheral functions is selected for the I/O port, the setting is invalid.
If the output pin of peripheral functions is selected for the I/O port, the setting is invalid.
If the bidirectional pin of peripheral functions is selected for the I/O port, the setting is invalid.

*1 : If the input pin of peripheral functions is selected for the I/O port, the setting is valid.
If the output pin of peripheral functions is selected for the I/O port, the setting is invalid.
If the bidirectional pin of peripheral functions is selected for the I/O port, the setting is invalid.

*2 : Indicates that the output pin of peripheral functions is selected for the I/O port.

*3 : Indicates that the bidirectional pin of peripheral functions is selected for the I/O port.

*4 : Indicates that neither the output pin nor the bidirectional pin of peripheral functions is selected for the I/O port.

■ Initially Selected Functions for the I/O Port

Table 2-3 describes initially selected functions for each I/O port after reset is released.

Table 2-3 Initially Selected Functions for Each I/O Port after Reset Is Released

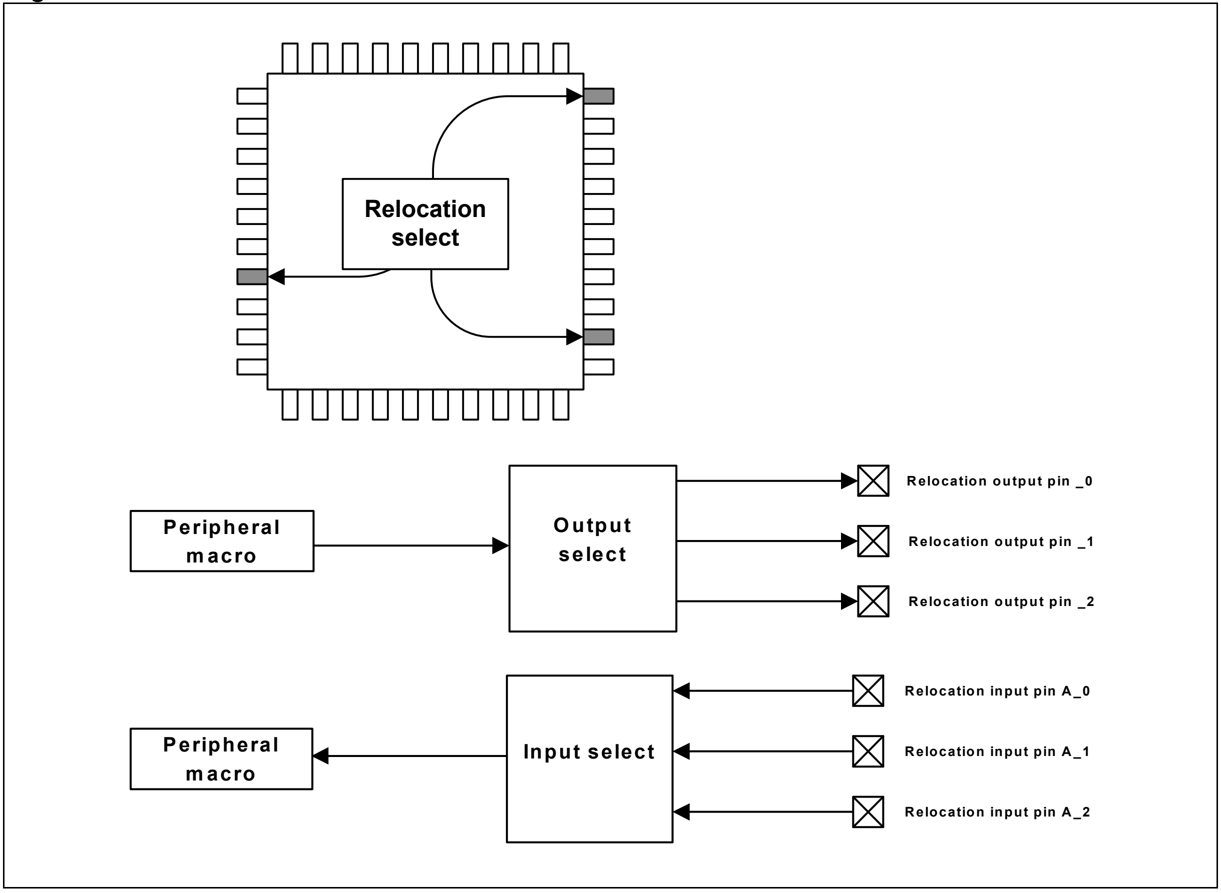
No	Pin	Initially selected function
1	TRSTX, TCK, TDI, TMS, TDO	JTAG pin is selected. Pull-up is enabled.
2	ANxx	Can be used as an analog input pin. Digital input is cut off and "0" is input.
3	X0, X1,	Can be used as an oscillation pin. Digital input is cut off and "0" is input.
4	All GPIO pins other than the above pins	Digital input. Output is Hi-Z.

Note: For the status of pins other than GPIO (MD pins, a reset pin), see "Data Sheet" of the product used.
All the output selection values of EPFR during reset are "no output".
For sub-oscillation, see Chapter “VBAT Domain”.

■ Relocation Function

- Some input/output of peripheral functions have more than one pin (relocation pin).
One of the pins can be selected by setting EPFR. Figure 2-2 shows the schematic view of relocation function.
-

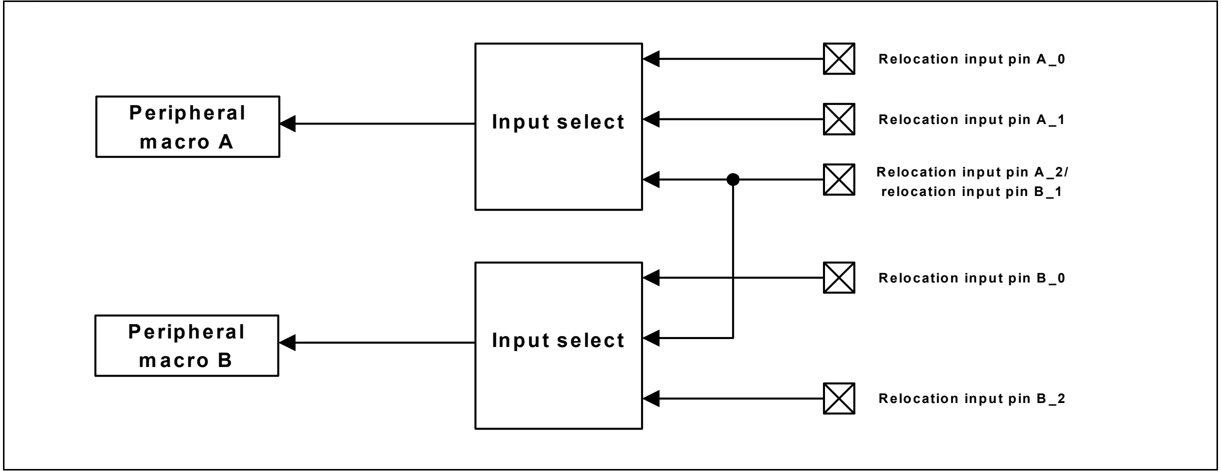
Figure 2-2 Schematic View of Relocation Function



Note: Which peripheral function is allocated to which pin depends on products.
See the pin function list of "Data Sheet" of the product used.

- Even if the input of one I/O port is connected to two or more peripheral functions, all peripheral inputs can be used by setting EPFR. For example, in Figure 2-3, by selecting input for both "Relocation input pin A_2" and "Relocation input pin B_1", simultaneous usage is possible. In this way, it is possible to use external interrupt and a multi-function serial input pin shared by one I/O port simultaneously.
-

Figure 2-3 Multiple Peripheral Inputs



- Even if an I/O pin is set as output, it can work as an input pin because input is not masked. For example, timer output can be used as external interrupt input which shared.

■ Fixed Priority of EPFR Outputs

Only one output pin function among two or more outputs is allocated to one I/O port.
By setting the EPFR register, if more than one output is set, fixed priority is applied and output pins are selected. Figure 2-4 shows output pins and fixed priority.

.

Figure 2-4 Output Pins and Fixed Priority

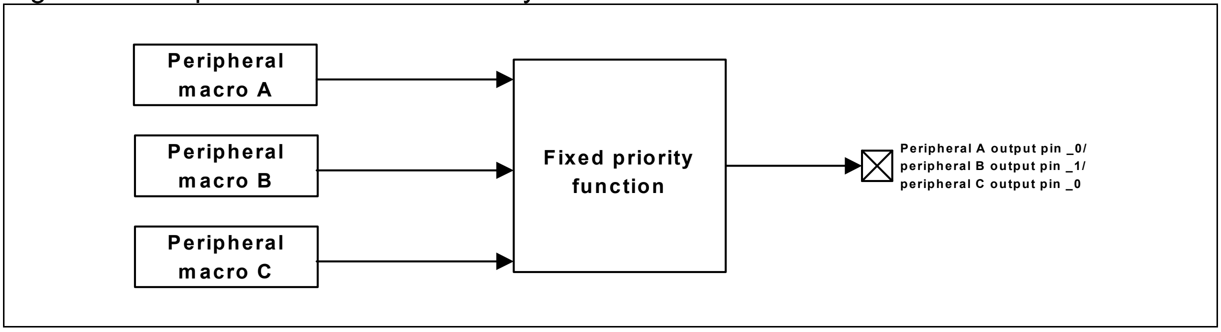

Peripheral macro A
Peripheral macro B
Peripheral macro C
Fixed priority function
Peripheral A output pin _0/
peripheral B output pin _1/
peripheral C output pin _0

Table 2-4 describes the fixed priority of EPFR.

Table 2-4 Fixed Priority of EPFR

Priority	Peripheral function	Applied pin
Higher	Special input	JTAG input, NMI input
↓	JTAG, trace	Output pin, I/O pin
↓	Ethernet-MAC	I/O pin
↓	USB (HCONX)	Output pin
↓	CAN	Output pin
↓	Multi-function serial	Output pin, I/O pin
↓	Base timer output	I/O pin
↓	Multi-function timer	Output pin
↓	External bus/SDRAM IF	Output pin, I/O pin
↓	SD card IF	Output pin, I/O pin
↓	Internal CR waveform output	Output pin
↓	RTC Output	Output pin
Lower	SUBCLK Output	Output pin

Note: The fixed priority is only applicable when "output" is set for more than one function. In case of "input", there is no fixed priority.
However, "Special input" has a higher priority than any other "output" setting. When "Special input" is set, the "output" setting allocated to the same port is invalid..

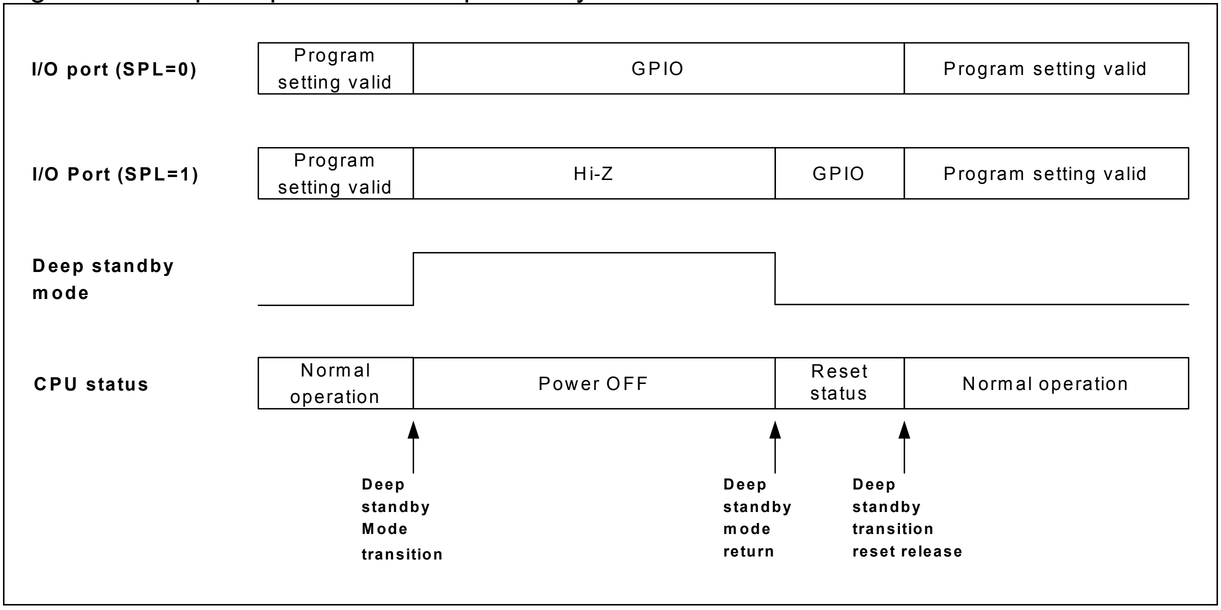
- Due to output setting on the lower part of the priority, the EPFR register always includes "no output" setting.
- If you are going to use a pin as an external input pin of peripheral functions, disable all shared output settings. If every output of a pin is not selected by the EPFR register, the pin works as an external input pin.

■ Operation in deep standby mode

GPIO function is selected in deep standby mode. Figure 2-5 shows I/O port operation in deep standby mode.

.

Figure 2-5 I/O port operation in deep standby mode



<Note>

For the state of each pin in deep standby mode, refer to the pin state table in the "Data Sheet" of the product used.

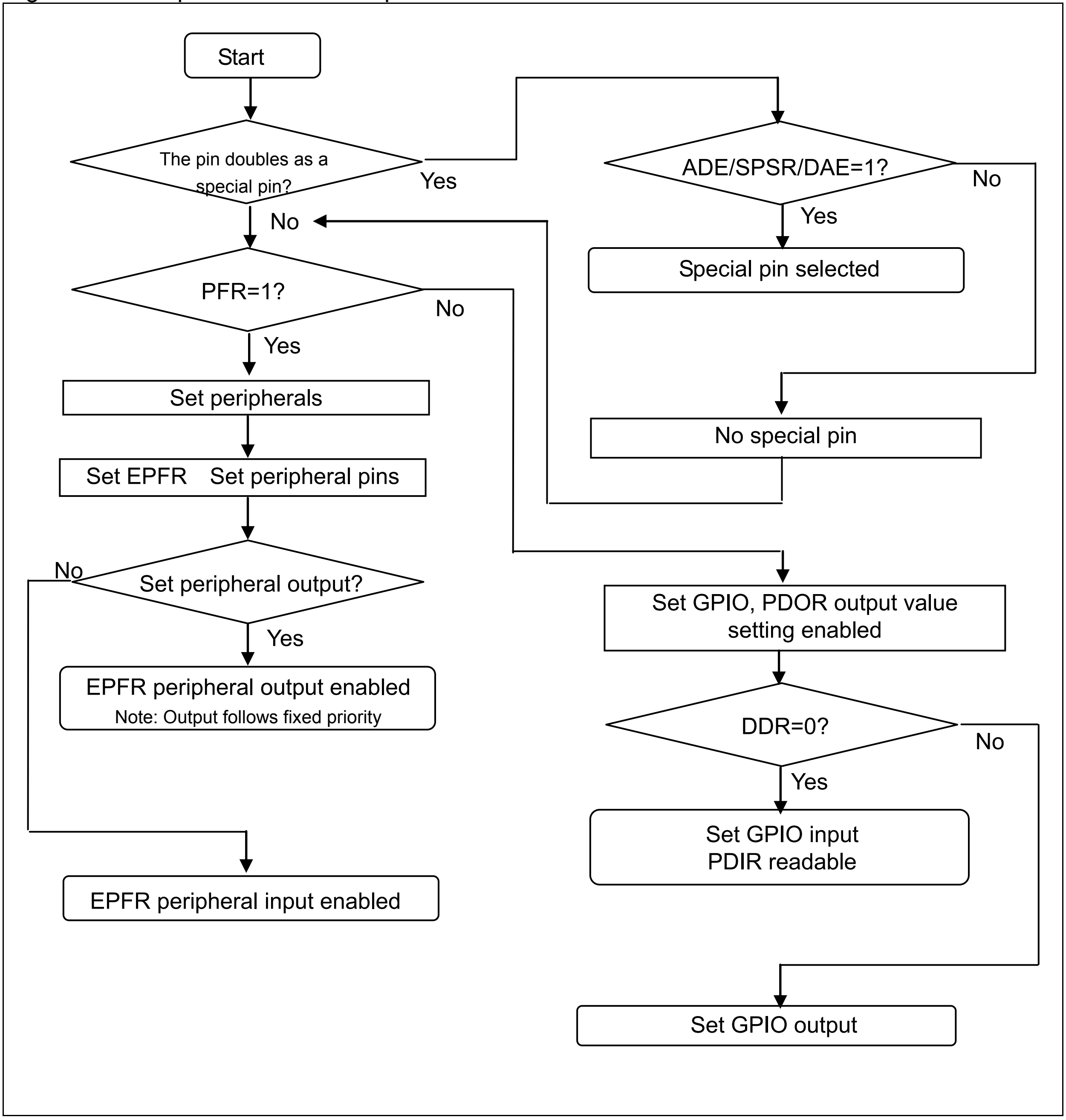
3. Setup Procedure Example

This section explains a procedure example of setting up the I/O port.

■ Setup of the I/O Port

By setting registers of the I/O port, select I/O direction and select GPIO/peripheral.
Figure 3-1 shows a setup procedure example.

Figure 3-1 Setup Procedure Example of the I/O Port


Start
The pin doubles as a special pin?
Yes
No
ADE/SPSR/DAE=1?
No
Yes
Special pin selected
PFR=1?
No
Yes
Set peripherals
No special pin
Set EPFR　Set peripheral pins
No
Set peripheral output?
Yes
EPFR peripheral output enabled
Note: Output follows fixed priority
Set GPIO, PDOR output value setting enabled
DDR=0?
No
Yes
Set GPIO input
PDIR readable
EPFR peripheral input enabled
Set GPIO output

4. Registers

This section provides the register list of the I/O port.

Table 4-1 provides the register list.

Table 4-1 Register List of the I/O Port

Abbreviation	Register name	Reference
PFR0	Port function setting register 0	4.1
PFR1	Port function setting register 1	
PFR2	Port function setting register 2	
PFR3	Port function setting register 3	
PFR4	Port function setting register 4	
PFR5	Port function setting register 5	
PFR6	Port function setting register 6	
PFR7	Port function setting register 7	
PFR8	Port function setting register 8	
PFR9	Port function setting register 9	
PFRA	Port function setting register A	
PFRB	Port function setting register B	
PFRC	Port function setting register C	
PFRD	Port function setting register D	
PFRE	Port function setting register E	
PFRF	Port function setting register F	
PCR0	Pull-up setting register 0	4.2
PCR1	Pull-up setting register 1	
PCR2	Pull-up setting register 2	
PCR3	Pull-up setting register 3	
PCR4	Pull-up setting register 4	
PCR5	Pull-up setting register 5	
PCR6	Pull-up setting register 6	
PCR7	Pull-up setting register 7	
PCR9	Pull-up setting register 9	
PCRA	Pull-up setting register A	
PCRB	Pull-up setting register B	

Abbreviation	Register name	Reference
PCRC	Pull-up setting register C	4.2
PCRD	Pull-up setting register D	
PCRE	Pull-up setting register E	
PCRF	Pull-up setting register F	
DDR0	Port input/output direction setting register 0	4.3
DDR1	Port input/output direction setting register 1	
DDR2	Port input/output direction setting register 2	
DDR3	Port input/output direction setting register 3	
DDR4	Port input/output direction setting register 4	
DDR5	Port input/output direction setting register 5	
DDR6	Port input/output direction setting register 6	
DDR7	Port input/output direction setting register 7	
DDR8	Port input/output direction setting register 8	
DDR9	Port input/output direction setting register 9	
DDRA	Port input/output direction setting register A	
DDRB	Port input/output direction setting register B	
DDRC	Port input/output direction setting register C	
DDRD	Port input/output direction setting register D	
DDRE	Port input/output direction setting register E	
DDRF	Port input/output direction setting register F	
PDIR0	Port input data register 0	4.4
PDIR1	Port input data register 1	
PDIR2	Port input data register 2	
PDIR3	Port input data register 3	
PDIR4	Port input data register 4	
PDIR5	Port input data register 5	
PDIR6	Port input data register 6	
PDIR7	Port input data register 7	
PDIR8	Port input data register 8	
PDIR9	Port input data register 9	
PDIRA	Port input data register A	
PDIRB	Port input data register B	
PDIRC	Port input data register C	

Abbreviation	Register name	Reference
PDIRD	Port input data register D	4.4
PDIRE	Port input data register E	
PDIRF	Port input data register F	
PDOR0	Port output data register 0	4.5
PDOR1	Port output data register 1	
PDOR2	Port output data register 2	
PDOR3	Port output data register 3	
PDOR4	Port output data register 4	
PDOR5	Port output data register 5	
PDOR6	Port output data register 6	
PDOR7	Port output data register 7	
PDOR8	Port output data register 8	
PDOR9	Port output data register 9	
PDORA	Port output data register A	
PDORB	Port output data register B	
PDORC	Port output data register C	
PDORD	Port output data register D	
PDORE	Port output data register E	
PDORF	Port output data register F	
ADE	Analog input setting register	4.6
SPSR	Special Port Setting Register	4.29
EPFR00	Extended pin function setting register 00	4.8
EPFR01	Extended pin function setting register 01	4.9
EPFR02	Extended pin function setting register 02	4.10
EPFR03	Extended pin function setting register 03	4.11
EPFR04	Extended pin function setting register 04	4.12
EPFR05	Extended pin function setting register 05	4.13
EPFR06	Extended pin function setting register 06	4.14
EPFR07	Extended pin function setting register 07	4.15
EPFR08	Extended pin function setting register 08	4.16
EPFR09	Extended pin function setting register 09	4.17
EPFR10	Extended pin function setting register 10	4.18
EPFR11	Extended pin function setting register 11	4.19
EPFR12	Extended pin function setting register 12	4.20
EPFR13	Extended pin function setting register 13	4.21

Abbreviation	Register name	Reference
EPFR14	Extended pin function setting register 14	4.22
EPFR15	Extended pin function setting register 15	4.23
EPFR16	Extended pin function setting register 16	4.24
EPFR17	Extended pin function setting register 17	4.25
EPFR18	Extended pin function setting register 18	4.26
EPFR19	Extended pin function setting register 19	4.27
EPFR20	Extended pin function setting register 20	4.28
PZR0	Port pseudo open drain setting register 0	4.30
PZR1	Port pseudo open drain setting register 1	
PZR2	Port pseudo open drain setting register 2	
PZR3	Port pseudo open drain setting register 3	
PZR4	Port pseudo open drain setting register 4	
PZR5	Port pseudo open drain setting register 5	
PZR6	Port pseudo open drain setting register 6	
PZR7	Port pseudo open drain setting register 7	
PZR8	Port pseudo open drain setting register 8	
PZR9	Port pseudo open drain setting register 9	
PZRA	Port pseudo open drain setting register A	
PZRB	Port pseudo open drain setting register B	
PZRC	Port pseudo open drain setting register C	
PZRD	Port pseudo open drain setting register D	
PZRE	Port pseudo open drain setting register E	
PZRF	Port pseudo open drain setting register F	

4.1. Port Function Setting Register (PFRx)

The PFRx register selects usage of a pin.

■ List of PFR Register Configuration

bit 31 16	15 0	Initial value	Attribute	Corresponding port
Reserved	PFR0	0x001F	R/W	P0F to P00
Reserved	PFR1	0x0000	R/W	P1F to P10
Reserved	PFR2	0x0000	R/W	P2F to P20
Reserved	PFR3	0x0000	R/W	P3F to P30
Reserved	PFR4	0x0000	R/W	P4F to P40
Reserved	PFR5	0x0000	R/W	P5F to P50
Reserved	PFR6	0x0000	R/W	P6F to P60
Reserved	PFR7	0x0000	R/W	P7F to P70
Reserved	PFR8	0x0000	R/W	P8F to P80
Reserved	PFR9	0x0000	R/W	P9F to P90
Reserved	PFRA	0x0000	R/W	PAF to PA0
Reserved	PFRB	0x0000	R/W	PBF to PB0
Reserved	PFRC	0x0000	R/W	PCF to PC0
Reserved	PFRD	0x0000	R/W	PDF to PD0
Reserved	PFRE	0x0000	R/W	PEF to PE0
Reserved	PFRF	0x0000	R/W	PFF to PF0

■ Detailed Register Configuration

bit	31 16	15 0
Field	Reserved	PFRx

■ Register Function

[bit31:16] Reserved: Reserved bits

"0x0000" is read out from these bits.
When writing these bits, set them to "0x0000".

[bit15:0] PFRx: Port Function Setting Register x

Selects usage of a pin.

bit		Description
Reading		Can read out the setting value of the register.
Writing	0	Uses a pin as a GPIO pin.
	1	Uses a pin as an input/output pin of peripheral functions.

<Notes>

- The "x" of PFRx is a wildcard. PFRx indicates PFR0, PFR1, PFR2, etc.
- The "x" of Px0 and PxF is a wildcard. Px0 indicates P00, P10, P20, etc. PxF indicates P0F, P1F, P2F, etc.
- Functions can be set for 16 ports from PxF to Px0.
- Each bit in the register sets each pin individually. There is a one-to-one correspondence between bit assignment and the order of pins. For example, bit15 of PFR0 sets P0F, bit14 of PFR0 sets P0E, and bit0 of PFR0 sets P00.
- As a JTAG pin is selected for P04 to P00, the initial value is "1".
- For a pin which is not available in your product, writing a value to the bit is invalid, and the read value is undefined.
  For GPIO pin sharing with oscillating pins(X0, X1), writing “PFR=1” is prohibited. To use the pin as oscillating pin, be sure to set “PFR=0”.
- The setting of P46 to P49 with this register is invalid. Use VbatIO port function control register (VBPFR) of VBAT RTC to set these pins. For details of VBAT RTC, see “VBAT Domain”.
- PFR0[4:0] register is not initialized by deep standby transition reset.

4.2. Pull-up Setting Register (PCRx)

The PCRx register sets pull-up of a pin.

■ List of PCR Register Configuration

bit 31 16	15 0	Initial value	Attribute	Corresponding port
Reserved	PCR0	0x001F	R/W	P0F to P00
Reserved	PCR1	0x0000	R/W	P1F to P10
Reserved	PCR2	0x0000	R/W	P2F to P20
Reserved	PCR3	0x0000	R/W	P3F to P30
Reserved	PCR4	0x0000	R/W	P4F to P40
Reserved	PCR5	0x0000	R/W	P5F to P50
Reserved	PCR6	0x0000	R/W	P6F to P60
Reserved	PCR7	0x0000	R/W	P7F to P70
Reserved	-	-	-	-
Reserved	PCR9	0x0000	R/W	P9F to P90
Reserved	PCRA	0x0000	R/W	PAF to PA0
Reserved	PCRB	0x0000	R/W	PBF to PB0
Reserved	PCRC	0x0000	R/W	PCF to PC0
Reserved	PCRD	0x0000	R/W	PDF to PD0
Reserved	PCRE	0x0000	R/W	PEF to PE0
Reserved	PCRF	0x0000	R/W	PFF to PF0

■ Detailed Register Configuration

bit	31 16	15 0
Field	Reserved	PCRx

■ Register Function

[bit31:16] Reserved: Register bits

"0x0000" is read out from these bits.
When writing these bits, set them to "0x0000".

[bit15:0] PCRx: Pull-up Setting Register x

Sets pull-up of a pin

bit		Description
Reading		Can read out the setting value of the register.
Writing	0	Disconnects the pull-up resistor of a pin.
	1	When a pin is in input status (for both GPIO and peripheral functions), the pull-up resistor is connected. When a pin is in output status, the pull-up resistor is disconnected.

<Notes>

- The "x" of PCRx is a wildcard. PCRx indicates PCR0, PCR1, PCR2, etc.
- The "x" of Px0 and PxF is a wildcard. Px0 indicates P00, P10, P20, etc. PxF indicates P0F, P1F, P2F, etc.
- One register allows setting 16 pull-ups from PxF to Px0.
- Each bit in the register sets each pin individually. There is a one-to-one correspondence between bit assignment and the order of pins. For example, bit15 of PCR0 sets P0F, bit14 of PCR0 sets P0E, and bit0 of PCR0 sets P00.
- As a JTAG pin is selected for P00 to P04, the initial value is "1".
- When using I^2C function, use external pull-up by setting PCRx=0.
- PCR8 is not available.
- For a pin which is not available in your product, writing a value to the bit is invalid, and the read value is undefined.
- PE0, PE1 ports do not have a pull-up resistor. Because of this, writing a value to PE register is invalid. An initial value or a write value is read in this register.
- The setting of P46 to P49 with this register is invalid. Use pull-up setting register (VBPFR) of VBAT to set these pins. For details of VBAT, see “VBAT Domain”.
- PCRx register is not initialized by deep standby transition reset.

4.3. Port input/output Direction Setting Register (DDRx)

The DDRx register sets input/output direction of a pin.

■ List of DDR Register Configuration

bit 31 ... 16	15 ... 0	Initial value	Attribute	Corresponding port
Reserved	DDR0	0x0000	R/W	P0F to P00
Reserved	DDR1	0x0000	R/W	P1F to P10
Reserved	DDR2	0x0000	R/W	P2F to P20
Reserved	DDR3	0x0000	R/W	P3F to P30
Reserved	DDR4	0x0000	R/W	P4F to P40
Reserved	DDR5	0x0000	R/W	P5F to P50
Reserved	DDR6	0x0000	R/W	P6F to P60
Reserved	DDR7	0x0000	R/W	P7F to P70
Reserved	DDR8	0x0000	R/W	P8F to P80
Reserved	DDR9	0x0000	R/W	P9F to P90
Reserved	DDRA	0x0000	R/W	PAF to PA0
Reserved	DDRB	0x0000	R/W	PBF to PB0
Reserved	DDRC	0x0000	R/W	PCF to PC0
Reserved	DDRD	0x0000	R/W	PDF to PD0
Reserved	DDRE	0x0000	R/W	PEF to PE0
Reserved	DDRF	0x0000	R/W	PFF to PF0

■ Detailed Register Configuration

bit	31 ... 16	15 ... 0
Field	Reserved	DDRx

■ Register Function

[bit31:16] Reserved: Reserved bits

"0x0000" is read out from these bits.
When writing these bits, set them to "0x0000".

[bit15:0] DDRx: Port input/output Direction Setting Register x

Sets input/output direction of a pin.

bit		Description
Reading		Can read out the setting value of the register.
Writing	0	Uses GPIO in input direction. If a pin is selected as an input/output pin of peripheral functions, this setting value is invalid.
	1	Uses GPIO in output direction. If a pin is selected as an input/output pin of peripheral functions, this setting value is invalid.

<Notes>

- The "x" of DDRx is a wildcard. DDRx indicates DDR0, DDR1, DDR2, etc.
- The "x" of Px0 and PxF is a wildcard. Px0 indicates P00, P10, P20, etc. PxF indicates P0F, P1F, P2F, etc.
- One register allows setting the input/output direction of 16 ports from PxF to Px0.
- Each bit in the register sets each pin individually. There is a one-to-one correspondence between bit assignment and the order of pins. For example, bit15 of DDR0 sets P0F, bit14 of DDR0 sets P0E, and bit0 of DDR0 sets P00.
- If the output RTO of a multifunction timer is selected, in an emergency stop due to DTTIX signal, a DDR controls pin status. For more information, see the chapter "Multifunction Timer" in "Timer Part".
- For a pin which is not available in your product, writing a value to the bit is invalid, and the read value is undefined.
- The setting of P46 to P49 with this register is invalid. Use port I/O direction setting register (VBDDR) of VBAT to set these pins. For details of VBAT, see “VBAT Domain”.
- DDRx register is not initialized by deep standby transition reset.

4.4. Port Input Data Register (PDIRx)

The PDIRx register indicates input data of a pin.

■ List of PDIR Register Configuration

bit 31 16	15 0	Initial value	Attribute	Corresponding port
Reserved	PDIR0	0x0000	R	P0F to P00
Reserved	PDIR1	0x0000	R	P1F to P10
Reserved	PDIR2	0x0000	R	P2F to P20
Reserved	PDIR3	0x0000	R	P3F to P30
Reserved	PDIR4	0x0000	R	P4F to P40
Reserved	PDIR5	0x0000	R	P5F to P50
Reserved	PDIR6	0x0000	R	P6F to P60
Reserved	PDIR7	0x0000	R	P7F to P70
Reserved	PDIR8	0x0000	R	P8F to P80
Reserved	PDIR9	0x0000	R	P9F to P90
Reserved	PDIRA	0x0000	R	PAF to PA0
Reserved	PDIRB	0x0000	R	PBF to PB0
Reserved	PDIRC	0x0000	R	PCF to PC0
Reserved	PDIRD	0x0000	R	PDF to PD0
Reserved	PDIRE	0x0000	R	PEF to PE0
Reserved	PDIRF	0x0000	R	PFF to PF0

■ Detailed Register Configuration

bit	31 16	15 0
Field	Reserved	PDIRx

■ Register Function

[bit31:16] Reserved: Reserved bits

"0x0000" is read out from these bits.
When writing these bits, set them to "0x0000".

[bit15:0] PDIRx: Port Input Data Register x

Reads out input data of a pin.

bit		Description
Reading	0	Regardless of pin function settings (PFR/EPFR/DDR/PDOR), it indicates that a pin is in the status of "L" level input or "L" level output. When a special pin is selected by ADE/SPSR, as input is cut off, 0 is always read out.
	1	Regardless of pin function settings (PFR/EPFR/DDR/PDOR), it indicates that a pin is in the status of "H" level input or "H" level output.
Writing		Writing does not affect anything.

<Notes>

- The "x" of PDIRx is a wildcard. PDIRx indicates PDIR0, PDIR1, PDIR2, etc.
- The "x" of Px0 and PxF is a wildcard. Px0 indicates P00, P10, P20, etc. PxF indicates P0F, P1F, P2F, etc.
- One register allows reading input data of 16 ports from PxF to Px0.
- Each bit in the register indicates the status of each pin individually. There is a one-to-one correspondence between bit assignment and the order of pins. For example, the 15th bit of PDIR0 indicates P0F, the 14th bit of PDIR0 indicates P0E, and the 0th bit of PDIR0 indicates P00.
- "0" is always read for a bit value of the pin which is not available in your product.
- The setting of P46 to P49 with this register is invalid. Use VbatIO port function control register (VBPFR) of VBAT RTC to set these pins. For details of VBAT RTC, see “VBAT”.
- PDIRx register is not initialized by deep standby transition reset.

4.5. Port Output Data Register x (PDORx)

The PDORx register sets output data to a pin.

■ List of PDOR Register Configuration

bit 31 - 16	15 - 0	Initial value	Attribute	Corresponding port
Reserved	PDOR0	0x0000	R/W	P0F to P00
Reserved	PDOR1	0x0000	R/W	P1F to P10
Reserved	PDOR2	0x0000	R/W	P2F to P20
Reserved	PDOR3	0x0000	R/W	P3F to P30
Reserved	PDOR4	0x0000	R/W	P4F to P40
Reserved	PDOR5	0x0000	R/W	P5F to P50
Reserved	PDOR6	0x0000	R/W	P6F to P60
Reserved	PDOR7	0x0000	R/W	P7F to P70
Reserved	PDOR8	0x0000	R/W	P8F to P80
Reserved	PDOR9	0x0000	R/W	P9F to P90
Reserved	PDORA	0x0000	R/W	PAF to PA0
Reserved	PDORB	0x0000	R/W	PBF to PB0
Reserved	PDORC	0x0000	R/W	PCF to PC0
Reserved	PDORD	0x0000	R/W	PDF to PD0
Reserved	PDORE	0x0000	R/W	PEF to PE0
Reserved	PDORF	0x0000	R/W	PFF to PF0

■ Detailed Register Configuration

bit	31 - 16	15 - 0
Field	Reserved	PDORx

■ Register Function

[bit31:16] Reserved: Reserved bits

"0x0000" is read out from these bits.
When writing these bits, set them to "0x0000".

[bit15:0] PDORx: Port Output Data Register x

Sets output data of a pin.

bit		Description
Reading		Reads out the register value.
Writing	0	Outputs "L" level to GPIO. If a pin is selected as I/O input or peripheral functions input/output, a setting value is invalid.
	1	Outputs "H" level to GPIO. If a pin is selected as I/O input or peripheral functions input/output, a setting value is invalid.

<Notes>

- The "x" of PDORx is a wildcard. PDORx indicates PDOR0, PDOR1, PDOR2, etc.
- The "x" of Px0 and PxF is a wildcard. Px0 indicates P00, P10, P20, etc. PxF indicates P0F, P1F, P2F, etc.
- One register allows setting output data of 16 ports from PxF to Px0.
- Each bit in the register sets each pin individually. There is a one-to-one correspondence between bit assignment and the order of pins. For example, bit15 of PDOR0 sets P0F, bit14 of PDOR0 sets P0E, and bit0 of PDOR0 sets P00.
- For a pin which is not available in your product, writing a value to the bit is invalid, and the read value is undefined.
- The setting of P46 to P49 with this register is invalid. Use port output data register (VBDOR) of VBAT RTC to set these pins. For details of VBAT, see “VBAT Domain”.
- PDORx register is not initialized by deep standby transition reset.
-

4.6. Analog Input Setting Register (ADE)

The ADE register sets an external pin as an analog signal input pin of ADC.

■ Register Configuration

bit	31	0
Field	ADE	
Attribute	R/W	
Initial value	0xFFFFFFFF	

■ Register Function

[bit31:0] ADE: Analog Input Setting Register

Sets as an analog signal input pin.

bit		Description
Reading		Reads out the register value.
Writing	0	Uses an external pin not as analog input but digital input/output.
	1	Uses an external pin as analog input. (An I/O cell will be in a state of input direction, input cut-off, and pull-up disconnection.)

<Notes>

- This register sets analog input pins from AN31 to AN00.
- Each bit in the register sets each pin individually. There is a one-to-one correspondence between bit assignment and the order of pins. For example, bit31 of ADE sets AN31, bit14 of ADE sets AN14, and bit0 of ADE sets AN00. The port position of ANxx differs by each product.
  For correspondence, refer to the "Data Sheet" of the product used.
- This register is not initialized by deep standby transition reset.

4.7. Extended Pin Function Setting Register (EPFRx)

The EPFRx register assigns functions to a pin if there is more than one function.

■ List of EPFRx Register Configuration

bit 31 0	Initial value	Attribute	Corresponding function
EPFR00	0x00030000	R/W	System function
EPFR01	0x00000000	R/W	Multi-function timer
EPFR02	0x00000000	R/W	
EPFR03	0x00000000	R/W	
EPFR04	0x00000000	R/W	Base timer
EPFR05	0x00000000	R/W	
EPFR06	0x00000000	R/W	External interrupt
EPFR07	0x00000000	R/W	Multi-function serial
EPFR08	0x00000000	R/W	
EPFR09	0x00000000	R/W	CAN/ADC trigger/QPRC
EPFR10	0x00000000	R/W	External bus
EPFR11	0x00000000	R/W	
EPFR12	0x00000000	R/W	Base timer
EPFR13	0x00000000	R/W	
EPFR14	0x00000000	R/W	QPRC/Ethernet-MAC
EPFR15	0x00000000	R/W	External interrupt
EPFR16	0x00000000	R/W	Multi-function serial
EPFR17	0x00000000	R/W	
EPFR18	0x00000000	R/W	QPRC/SDCARD Interface
EPFR19	0x00000000	R/W	Reserved
EPFR20	0x00000000	R/W	External bus

<Notes>

- EPFRx register is not initialized by deep standby transition reset.
-

4.8. Extended Pin Function Setting Register 00 (EPFR00)

The EPFR00 register assigns functions to a pin if there is more than one function.

■ Register Configuration

bit	31	30	29	28	27	26	25	24
Field	Reserved						TRC1E	TRC0E
Attribute	-						R/W	R/W
Initial value							0	0

bit	23	22	21	20	19	18	17	16
Field	Reserved						JTAGEN1S	JTAGEN0B
Attribute	-						R/W	R/W
Initial value							1	1

bit	15	14	13	12	11	10	9	8
Field	Reserved		USBP1E	Reserved			USBP0E	Reserved
Attribute	-		R/W	-			R/W	-
Initial value	-		0	-			0	

bit	7	6	5	4	3	2	1	0
Field	SUBOUTE		RTCCOE		Reserved	CROUTE		NMIS
Attribute	R/W		R/W		-	R/W		R/W
Initial value	00		00			00		0

■ Register Function

[bit31:26] Reserved: Reserved bits

"0b000000" is read out from these bits.
When writing these bits, set them to "0b000000".

[bit25] TRC1E: TRACED Function Select bit 1

Selects a function for TRACED2 and TRACED3 pins.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not use two pins of TRACED2 and TRACED3. [Initial value] (A shared pin is available)
	1	Uses two pins of TRACED2 and TRACED3.

[bit24] TRC0E: TRACED Function Select bit 0

Selects a function for TRACECLK, TRACED0, and TRACED1 pins.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not use three pins of TRACECLK, TRACED0, and TRACED1. [Initial value] (A shared pin is available)
	1	Uses three pins of TRACECLK, TRACED0, and TRACED1.

[bit23:18] Reserved: Reserved bits

"0b000000" is read out from these bits.
When writing these bits, set them to "0b000000".

[bit17] JTAGEN1S: JTAG Function Select bit 1

Selects a function for TRSTX and TDI pins.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not use two pins of TRSTX and TDI. (A shared pin is available.)
	1	Uses two pins of TRSTX and TDI. [Initial value]

[bit16] JTAGEN0B: JTAG Function Select bit 0

Selects a function for TCK, TMS, and TDO pins.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not use three pins of TCK, TMS, and TDO. (A shared pin is available.)
	1	Uses three pins of TCK, TMS, and TDO. [Initial value]

[bit15:14] Reserved: Reserved bits

"0b00" is read out from these bits.
When writing these bits, set them to "0b00".

[bit13] USBP1E: USB ch.1 Function Select bit 1

Selects a function for USB ch.1.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output D+ resistor control signal (HCONTX) for USB ch.1. [Initial value] (A shared pin is available.)
	1	Produces output D+ resistor control signal (HCONTX) for USB ch.1.

[bit12:10] Reserved: Reserved bits

"0b000" is read out from these bits.
When writing these bits, set them to "0b000".

[bit9] USBP0E: USB ch.0 Function Select bit 1

Selects a function for USB ch.0.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output D+ resistor control signal (HCONTX) for USB ch.0. [Initial value] (A shared pin is available.)
	1	Produces output D+ resistor control signal (HCONTX) for USB ch.0.

[bit8] Reserved: Reserved bit

"0" is read out from this bit.
When writing this bit, set it to "0".

[bit7:6] SUBOUTE: Sub clock divide output function select bit

Selects sub clock divide output.

bit		Description
Reading		Reads out the register value.
Writing	00	Sub clock divide output is not executed. [initial value]
	01	SUBOUT_0 is used as the sub clock divide output pin.
	10	SUBOUT_1 is used as the sub clock divide output pin.
	11	SUBOUT_2 is used as the sub clock divide output pin.

[bit5:4] RTCCOE: RTC clock output select bit

Selects a RTC clock output.

bit		Description
Reading		Reads out the register value.
Writing	00	RTC clock output is not executed. [initial value]
	01	RTCCOE_0 is used as the RTC clock output pin.
	10	RTCCOE_1 is used as the RTC clock output pin.
	11	RTCCOE_2 is used as the RTC clock output pin.

[bit3] Reserved: Reserved bit

"0" is read out from this bit.
When writing this bit, set it to "0".

[bit2:1] CROUTE: Internal high-speed CR Oscillation Output Function Select bit

Selects internal high-speed CR oscillation output.

bit		Description
Reading		Reads out the register value.
Writing	00	Does not produce internal high-speed CR oscillation output. [Initial value]
	01	Uses CROUT_0 at the internal high-speed CR oscillation output pin.
	10	Uses CROUT_1 at the internal high-speed CR oscillation output pin.
	11	Uses CROUT_2 at the internal high-speed CR oscillation output pin.

[bit0] NMIS: NMIX Function Select bit

Selects a function for the NMIX pin.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not use the NMIX pin. [Initial value]
	1	Uses the NMIX pin.

<Note>

This register is not initialized by deep standby transition reset.

4.9. Extended Pin Function Setting Register 01 (EPFR01)

The EPFR01 register assigns functions to a pin of the multifunction timer Unit0.

■ Register Configuration

bit	31	30	29	28	27	26	25	24
Field	IC03S			IC02S			IC01S	
Attribute	R/W			R/W			R/W	
Initial value	000			000			00	

bit	23	22	21	20	19	18	17	16
Field	IC01S	IC00S			FRCK0S		DTTI0S	
Attribute	R/W	R/W			R/W		R/W	
Initial value	0	000			00		00	

bit	15	14	13	12	11	10	9	8
Field	Reserved			DTTI0C	RTO05E		RTO04E	
Attribute	-			R/W	R/W		R/W	
Initial value	-			0	00		00	

bit	7	6	5	4	3	2	1	0
Field	RTO03E		RTO02E		RTO01E		RTO00E	
Attribute	R/W		R/W		R/W		R/W	
Initial value	00		00		00		00	

■ Register Function

[bit31:29] IC03S: IC03 Input Select bits

Selects input for IC03.

bit		Description
Reading		Reads out the register value.
Writing	000	Uses IC03_0 at the input pin of the input capture IC03. [Initial value]
	001	Same as Writing 000.
	010	Uses IC03_1 at the input pin of the input capture IC03.
	011	Uses IC03_2 at the input pin of the input capture IC03.
	100	Uses internal macro MFS ch.3 LSYN for input of the input capture IC03.
	101	Uses internal macro MFS ch.7 LSYN for input of the input capture IC03.
	110	Setting is prohibited.
	111	Uses the internal macro pin CRTRIM for input of the input capture IC03.

[bit28:26] IC02S: IC02 Input Select bits

Selects input for IC02.

bit		Description
Reading		Reads out the register value.
Writing	000	Uses IC02_0 at the input pin of the input capture IC02. [Initial value]
	001	Same as Writing 000.
	010	Uses IC02_1 at the input pin of the input capture IC02.
	011	Uses IC02_2 at the input pin of the input capture IC02.
	100	Uses internal macro MFS ch.2 LSYN for input of the input capture IC02.
	101	Uses internal macro MFS ch.6 LSYN for input of the input capture IC02.
	110	Setting is prohibited.
	111	Setting is prohibited.

[bit25:23] IC01S: IC01 Input Select bits

Selects input for IC01.

bit		Description
Reading		Reads out the register value.
Writing	000	Uses IC01_0 at the input pin of the input capture IC01. [Initial value]
	001	Same as Writing 000.
	010	Uses IC01_1 at the input pin of the input capture IC01.
	011	Uses IC01_2 at the input pin of the input capture IC01.
	100	Uses internal macro MFS ch.1 LSYN for input of the input capture IC01.
	101	Uses internal macro MFS ch.5 LSYN for input of the input capture IC01.
	110	Setting is prohibited.
	111	Setting is prohibited.

[bit22:20] IC00S: IC00 Input Select bits

Selects input for IC00.

bit		Description
Reading		Reads out the register value.
Writing	000	Uses IC00_0 at the input pin of the input capture IC00. [Initial value]
	001	Same as Writing 000.
	010	Uses IC00_1 at the input pin of the input capture IC00.
	011	Uses IC00_2 at the input pin of the input capture IC00.
	100	Uses internal macro MFS ch.0 LSYN for input of the input capture IC00.
	101	Uses internal macro MFS ch.4 LSYN for input of the input capture IC00.
	110	Setting is prohibited.
	111	Setting is prohibited.

[bit19:18] FRCK0S: FRCK0 Input Select bits

Selects input for FRCK0.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses FRCK0_0 at the input pin of the free-run timer FRCK0. [Initial value]
	01	Same as Writing 00.
	10	Uses FRCK0_1 at the input pin of the free-run timer FRCK0.
	11	Uses FRCK0_2 at the input pin of the free-run timer FRCK0.

[bit17:16] DTTI0S: DTTIX0 Input Select bits

Selects input for DTTIX0.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses DTTIX0_0 at the input pin of the waveform generator DTTIX0. [Initial value]
	01	Same as Writing 00.
	10	Uses DTTIX0_1 at the input pin of the waveform generator DTTIX0.
	11	Uses DTTIX0_2 at the input pin of the waveform generator DTTIX0.

[bit15:13] Reserved: Reserved bits

"0b000" is read out from these bits.
When writing these bits, set them to "0b000".

[bit12] DTTI0C: DTTIX0 Function Select bit

Selects a function for DTTIX0.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not switch GPIO by DTTIF0 for output of pins RTO00 to RTO05. [Initial value]
	1	Switches GPIO by DTTIF0 for output of pins RTO00 to RTO05.

[bit11:10] RTO05E: RTO05 Output Select bits

Selects output for RTO05.

bit		Description
Reading		Reads out the register value.
Writing	00	Does not produce output for the waveform generator RTO05. [Initial value]
	01	Uses RTO05_0 at the output pin of the waveform generator RTO05.
	10	Uses RTO05_1 at the output pin of the waveform generator RTO05.
	11	Setting is prohibited.

[bit9:8] RTO04E: RTO04 Output Select bits

Selects output for RTO04.

bit		Description
Reading		Reads out the register value.
Writing	00	Does not produce output for the waveform generator RTO04. [Initial value]
	01	Uses RTO04_0 at the output pin of the waveform generator RTO04.
	10	Uses RTO04_1 at the output pin of the waveform generator RTO04.
	11	Setting is prohibited.

[bit7:6] RTO03E: RTO03 Output Select bits

Selects output for RTO03.

bit		Description
Reading		Reads out the register value.
Writing	00	Does not produce output for the waveform generator RTO03. [Initial value]
	01	Uses RTO03_0 at the output pin of the waveform generator RTO03.
	10	Uses RTO03_1 at the output pin of the waveform generator RTO03.
	11	Setting is prohibited.

[bit5:4] RTO02E: RTO02 Output Select bits

Selects output for RTO02.

bit5:4		Description
Reading		Reads out the register value.
Writing	00	Does not produce output for the waveform generator RTO02. [Initial value]
	01	Uses RTO02_0 at the output pin of the waveform generator RTO02.
	10	Uses RTO02_1 at the output pin of the waveform generator RTO02.
	11	Setting is prohibited.

[bit3:2] RTO01E: RTO01 Output Select bits

Selects output for RTO01.

bit		Description
Reading		Reads out the register value.
Writing	00	Does not produce output for the waveform generator RTO01. [Initial value]
	01	Uses RTO01_0 at the output pin of the waveform generator RTO01.
	10	Uses RTO01_1 at the output pin of the waveform generator RTO01.
	11	Setting is prohibited.

[bit1:0] RTO00E: RTO00 Output Select bits

Selects output for RTO00.

bit		Description
Reading		Reads out the register value.
Writing	00	Does not produce output for the waveform generator RTO00. [Initial value]
	01	Uses RTO00_0 at the output pin of the waveform generator RTO00.
	10	Uses RTO00_1 at the output pin of the waveform generator RTO00.
	11	Setting is prohibited.

<Note>

This register is not initialized by deep standby transition reset.

4.10. Extended Pin Function Setting Register 02 (EPFR02)

The EPFR02 register assigns functions to a pin of the multifunction timer Unit1.

■ Register Configuration

bit	31	30	29	28	27	26	25	24
Field	IC13S			IC12S			IC11S	
Attribute	R/W			R/W			R/W	
Initial value	000			000			00	

bit	23	22	21	20	19	18	17	16
Field	IC11S	IC10S			FRCK1S		DTTI1S	
Attribute	R/W	R/W			R/W		R/W	
Initial value	0	000			00		00	

bit	15	14	13	12	11	10	9	8
Field	Reserved			DTTI1C	RTO15E		RTO14E	
Attribute	-			R/W	R/W		R/W	
Initial value	-			0	00		00	

bit	7	6	5	4	3	2	1	0
Field	RTO13E		RTO12E		RTO11E		RTO10E	
Attribute	R/W		R/W		R/W		R/W	
Initial value	00		00		00		00	

■ Register Function

[bit31:29] IC13S: IC13 Input Select bits

Selects input for IC13.

bit		Description
Reading		Reads out the register value.
Writing	000	Uses IC13_0 at the input pin of the input capture IC13. [Initial value]
	001	Same as Writing 000.
	010	Uses IC13_1 at the input pin of the input capture IC13.
	011	Setting is prohibited.
	100	Uses internal macro MFS ch.3 LSYN for input of the input capture IC13.
	101	Uses internal macro MFS ch.7 LSYN for input of the input capture IC13.
	110	Setting is prohibited.
	111	Setting is prohibited.

[bit28:26] IC12S: IC12 Input Select bits

Selects input for IC12.

bit		Description
Reading		Reads out the register value.
Writing	000	Uses IC12_0 at the input pin of the input capture IC12. [Initial value]
	001	Same as Writing 000.
	010	Uses IC12_1 at the input pin of the input capture IC12.
	011	Setting is prohibited.
	100	Uses internal macro MFS ch.2 LSYN for input of the input capture IC12.
	101	Uses internal macro MFS ch.6 LSYN for input of the input capture IC12.
	110	Setting is prohibited.
	111	Setting is prohibited.

[bit25:23] IC11S: IC11 Input Select bits

Selects input for IC11.

bit		Description
Reading		Reads out the register value.
Writing	000	Uses IC11_0 at the input pin of the input capture IC11. [Initial value]
	001	Same as Writing 000.
	010	Uses IC11_1 at the input pin of the input capture IC11.
	011	Setting is prohibited.
	100	Uses internal macro MFS ch.1 LSYN for input of the input capture IC11.
	101	Uses internal macro MFS ch.5 LSYN for input of the input capture IC11.
	110	Setting is prohibited.
	111	Setting is prohibited.

[bit22:20] IC10S: IC10 Input Select bits

Selects input for IC10.

bit		Description
Reading		Reads out the register value.
Writing	000	Uses IC10_0 at the input pin of the input capture IC10. [Initial value]
	001	Same as Writing 000.
	010	Uses IC10_1 at the input pin of the input capture IC10.
	011	Setting is prohibited.
	100	Uses internal macro MFS ch.0 LSYN for input of the input capture IC10.
	101	Uses internal macro MFS ch.4 LSYN for input of the input capture IC10.
	110	Setting is prohibited.
	111	Setting is prohibited.

[bit19:18] FRCK1S: FRCK1 Input Select bits

Selects input for FRCK1.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses FRCK1_0 at the input pin of the free-run timer FRCK1. [Initial value]
	01	Same as Writing 00.
	10	Uses FRCK1_1 at the input pin of the free-run timer FRCK1.
	11	Setting is prohibited.

[bit17:16] DTTI1S: DTTIX1 Input Select bits

Select input for DTTIX1.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses DTTIX1_0 at the input pin of the waveform generator DTTIX1. [Initial value]
	01	Same as Writing 00.
	10	Uses DTTIX1_1 at the input pin of the waveform generator DTTIX1.
	11	Setting is prohibited.

[bit15:13] Reserved: Reserved bits

"0b000" is read out from these bits.
When writing these bits, set them to "0b000".

[bit12] DTTI1C: DTTIX1 Function Select bit

Selects a function for DTTIX1.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not switch GPIO by DTTIF1 for output of pins RTO10 to RTO15. [Initial value]
	1	Switches GPIO by DTTIF1 for output of pins RTO10 to RTO15.

[bit11:10] RTO15E: RTO15 Output Select bits

Selects output for RTO15.

bit		Description
Reading		Reads out the register value.
Writing	00	Does not produce output for the waveform generator RTO15. [Initial value]
	01	Uses RTO15_0 at the output pin of the waveform generator RTO15.
	10	Uses RTO15_1 at the output pin of the waveform generator RTO15.
	11	Setting is prohibited.

[bit9:8] RTO14E: RTO14 Output Select bits

Selects output for RTO14.

bit		Description
Reading		Reads out the register value.
Writing	00	Does not produce output for the waveform generator RTO14. [Initial value]
	01	Uses RTO14_0 at the output pin of the waveform generator RTO14.
	10	Uses RTO14_1 at the output pin of the waveform generator RTO14.
	11	Setting is prohibited.

[bit7:6] RTO13E: RTO13 Output Select bits

Selects output for RTO13.

bit		Description
Reading		Reads out the register value.
Writing	00	Does not produce output for the waveform generator RTO13. [Initial value]
	01	Uses RTO13_0 at the output pin of the waveform generator RTO13.
	10	Uses RTO13_1 at the output pin of the waveform generator RTO13.
	11	Setting is prohibited.

[bit5:4] RTO12E: RTO12 Output Select bits

Selects output for RTO12.

bit		Description
Reading		Reads out the register value.
Writing	00	Does not produce output for the waveform generator RTO12. [Initial value]
	01	Uses RTO12_0 at the output pin of the waveform generator RTO12.
	10	Uses RTO12_1 at the output pin of the waveform generator RTO12.
	11	Setting is prohibited.

[bit3:2] RTO11E: RTO11 Output Select bits

Selects output for RTO11.

bit		Description
Reading		Reads out the register value.
Writing	00	Does not produce output for the waveform generator RTO11. [Initial value]
	01	Uses RTO11_0 at the output pin of the waveform generator RTO11.
	10	Uses RTO11_1 at the output pin of the waveform generator RTO11.
	11	Setting is prohibited.

[bit1:0] RTO10E: RTO10 Output Select bits

Selects output for RTO10.

bit		Description
Reading		Reads out the register value.
Writing	00	Does not produce output for the waveform generator RTO10. [Initial value]
	01	Uses RTO10_0 at the output pin of the waveform generator RTO10.
	10	Uses RTO10_1 at the output pin of the waveform generator RTO10.
	11	Setting is prohibited.

<Notes>

- This register is not initialized by deep standby transition reset.

4.11. Extension Function Pin Setting Register 03 (EPFR03)

EPFR03 register sets the function assignment to the multi-function timer Unit2 pin.

■ Register Configuration

bit	31	30	29	28	27	26	25	24
Field	IC23S			IC22S			IC21S	
Attribute	R/W			R/W			R/W	
Initial value	000			000			00	

bit	23	22	21	20	19	18	17	16
Field	IC21S	IC20S			FRCK2S		DTTI2S	
Attribute	R/W	R/W			R/W		R/W	
Initial value	0	000			00		00	

bit	15	14	13	12	11	10	9	8
Field	Reserved			DTTI2C	RTO25E		RTO24E	
Attribute	-			R/W	R/W		R/W	
Initial value	-			0	00		00	

bit	7	6	5	4	3	2	1	0
Field	RTO23E		RTO22E		RTO21E		RTO20E	
Attribute	R/W		R/W		R/W		R/W	
Initial value	00		00		00		00	

■ Register Function

[bit31:29] IC23S: IC23 input select bits

Selects IC23 input.

bit		Description
Reading		Reads out the register value.
Writing	000	Use IC23_0 as the input pin of input capture IC23. [initial value]
	001	Same as when writing "000"
	010	Use IC23_1 as the input pin of input capture IC23.
	011	Setting is prohibited.
	100	Use internal macro MFS ch.3 LSYN as input of input capture IC23.
	101	Use internal macro MFS ch.7 LSYN as input of input capture IC23.
	110	Setting is prohibited.
	111	Setting is prohibited.

[bit28:26] IC22S: IC22 input select bits

Selects IC22 input.

bit		Description
Reading		Reads out the register value.
Writing	000	Use IC22_0 as the input pin of input capture IC22. [initial value]
	001	Same as when writing "000"
	010	Use IC22_1 as the input pin of input capture IC22.
	011	Setting is prohibited.
	100	Use internal macro MFS ch.2 LSYN as input of input capture IC22.
	101	Use internal macro MFS ch.6 LSYN as input of input capture IC22.
	110	Setting is prohibited.
	111	Setting is prohibited.

[bit25:23] IC21S: IC21 input select bits

Selects IC21 input.

bit		Description
Reading		Reads out the register value.
Writing	000	Use IC21_0 as the input pin of input capture IC21. [initial value]
	001	Same as when writing "000"
	010	Use IC21_1 as the input pin of input capture IC21.
	011	Setting is prohibited.
	100	Use internal macro MFS ch.1 LSYN as input of input capture IC21.
	101	Use internal macro MFS ch.5 LSYN as input of input capture IC21.
	110	Setting is prohibited.
	111	Setting is prohibited.

[bit22:20] IC20S: IC20 input select bits

Selects IC20 input.

bit		Description
Reading		Reads out the register value.
Writing	000	Use IC20_0 as the input pin of input capture IC20. [initial value]
	001	Same as when writing "000"
	010	Use IC20_1 as the input pin of input capture IC20.
	011	Setting is prohibited.
	100	Use internal macro MFS ch.0 LSYN as input of input capture IC20.
	101	Use internal macro MFS ch.4 LSYN as input of input capture IC20.
	110	Setting is prohibited.
	111	Setting is prohibited.

[bit19:18] FRCK2S: FRCK2 Input Select bits

Selects input for FRCK2.

bit		Description
Reading		Reads out the register value.
Writing	00	Use FRCK2_0 as the input pin of free-run timer FRCK2. [Initial value]
	01	Same as Writing 00.
	10	Use FRCK2_1 as the input pin of free-run timer FRCK2.
	11	Setting is prohibited.

[bit17:16] DTTI2S: DTTIX2 Input Select bits

Selects input for DTTIX2.

bit		Description
Reading		Reads out the register value.
Writing	00	Use DTTIX2_0 as the input pin of waveform generator DTTIX2. [Initial value]
	01	Same as Writing 00.
	10	Use DTTIX2_1 as the input pin of waveform generator DTTIX2.
	11	Setting is prohibited.

[bit15:13] Reserved: Reserved bits

"0b000" is read from these bits.
When writing, set them to "0b000".

[bit12] DTTI2C: DTTIX2 Function Select bit

Selects the function of DTTIX2.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not switch GPIO by DTTIF2 for outputs of the pins from RTO20 to RTO25. [Initial value]
	1	Switches GPIO by DTTIF2 for outputs of the pins from RTO20 to RTO25.

[bit11:10] RTO25E: RTO25 Output Select bits

Selects the output of RTO25.

bit		Description
Reading		Reads out the register value.
Writing	00	Does not output waveform generator RTO25. [Initial value]
	01	Use RTO25_0 as the output pin of waveform generator RTO25.
	10	Use RTO25_1 as the output pin of waveform generator RTO25.
	11	Setting is prohibited.

[bit9:8] RTO24E: RTO24 Output Select bits

Selects output for RTO24.

bit		Description
Reading		Reads out the register value.
Writing	00	Does not output waveform generator RTO24. [Initial value]
	01	Use RTO24_0 as the output pin of waveform generator RTO24.
	10	Use RTO24_1 as the output pin of waveform generator RTO24.
	11	Setting is prohibited.

[bit7:6] RTO23E: RTO23 Output Select bits

Selects output for RTO23.

bit		Description
Reading		Reads out the register value.
Writing	00	Does not output waveform generator RTO23. [Initial value]
	01	Use RTO23_0 as the output pin of waveform generator RTO23.
	10	Use RTO23_1 as the output pin of waveform generator RTO23.
	11	Setting is prohibited.

[bit5:4] RTO22E: RTO22 Output Select bits

Selects output for RTO22.

bit		Description
Reading		Reads out the register value.
Writing	00	Does not output waveform generator RTO22. [Initial value]
	01	Use RTO22_0 as the output pin of waveform generator RTO22.
	10	Use RTO22_1 as the output pin of waveform generator RTO22.
	11	Setting is prohibited.

[bit3:2] RTO21E: RTO21 Output Select bits

Selects output for RTO21.

bit3:2		Description
Reading		Reads out the register value.
Writing	00	Does not output waveform generator RTO21. [Initial value]
	01	Use RTO21_0 as the output pin of waveform generator RTO21.
	10	Use RTO21_1 as the output pin of waveform generator RTO21.
	11	Setting is prohibited.

[bit1:0] RTO20E: RTO20 Output Select bits

Selects output for RTO20.

bit1:0		Description
Reading		Reads out the register value.
Writing	00	Does not output waveform generator RTO20. [Initial value]
	01	Use RTO20_0 as the output pin of waveform generator RTO20.
	10	Use RTO20_1 as the output pin of waveform generator RTO20.
	11	Setting is prohibited.

<Notes>

- This register is not initialized by deep standby transition reset.

4.12. Extended Pin Function Setting Register 04 (EPFR04)

The EPFR04 register assigns functions to pins of ch.0, ch.1, ch.2, and ch.3 of the base timer.

■ Register Configuration

bit	31	30	29	28	27	26	25	24
Field	Reserved		TIOB3S		TIOA3E		TIOA3S	
Attribute	-		R/W		R/W		R/W	
Initial value	-		00		00		00	

bit	23	22	21	20	19	18	17	16
Field	Reserved		TIOB2S		TIOA2E		Reserved	
Attribute	-		R/W		R/W		-	
Initial value	-		00		00		-	

bit	15	14	13	12	11	10	9	8
Field	Reserved		TIOB1S		TIOA1E		TIOA1S	
Attribute	-		R/W		R/W		R/W	
Initial value	-		00		00		00	

bit	7	6	5	4	3	2	1	0
Field	Reserved	TIOB0S			TIOA0E		Reserved	
Attribute	-	R/W			R/W		-	
Initial value	-	000			00		-	

■ Register Function

[bit31:30] Reserved: Reserved bits

"0b00" is read out from these bits.
When writing these bits, set them to "0b00".

[bit29:28] TIOB3S: TIOB3 Input Select bits

Selects input for TIOB3.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses TIOB3_0 at the input pin of BT ch.3 TIOB. [Initial value]
	01	Same as Writing 00.
	10	Uses TIOB3_1 at the input pin of BT ch.3 TIOB.
	11	Uses TIOB3_2 at the input pin of BT ch.3 TIOB.

[bit27:26] TIOA3E: TIOA3 Output Select bits

Selects output for TIOA3.

bit		Description
Reading		Reads out the register value.
Writing	00	Does not produce output for BT ch.3 TIOA. [Initial value]
	01	Uses TIOA3_0 at the output pin of BT ch.3 TIOA.
	10	Uses TIOA3_1 at the output pin of BT ch.3 TIOA.
	11	Uses TIOA3_2 at the output pin of BT ch.3 TIOA.

[bit25:24] TIOA3S: TIOA3 Input Select bits

Selects input for TIOA3.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses TIOA3_0 at the input pin of BT ch.3 TIOA. [Initial value]
	01	Same as Writing 00.
	10	Uses TIOA3_1 at the input pin of BT ch.3 TIOA.
	11	Uses TIOA3_2 at the input pin of BT ch.3 TIOA.

[bit23:22] Reserved: Reserved bits

"0b00" is read out from these bits.
When writing these bits, set them to "0b00".

[bit21:20] TIOB2S: TIOB2 Input Select bits

Selects input for TIOB2.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses TIOB2_0 at the input pin of BT ch.2 TIOB. [Initial value]
	01	Same as Writing 00.
	10	Uses TIOB2_1 at the input pin of BT ch.2 TIOB.
	11	Uses TIOB2_2 at the input pin of BT ch.2 TIOB.

[bit19:18] TIOA2E: TIOA2 Output Select bits

Selects output for TIOA2.

bit19:18		Description
Reading		Reads out the register value.
Writing	00	Does not produce output for BT ch.2 TIOA. [Initial value]
	01	Uses TIOA2_0 at the output pin of BT ch.2 TIOA.
	10	Uses TIOA2_1 at the output pin of BT ch.2 TIOA.
	11	Uses TIOA2_2 at the output pin of BT ch.2 TIOA.

[bit17:14] Reserved: Reserved bits

"0b0000" is read out from these bits.
When writing these bits, set them to "0b0000".

[bit13:12] TIOB1S: TIOB1 Input Select bits

Selects input for TIOB1.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses TIOB1_0 at the input pin of BT ch.1 TIOB. [Initial value]
	01	Same as Writing 00.
	10	Uses TIOB1_1 at the input pin of BT ch.1 TIOB.
	11	Uses TIOB1_2 at the input pin of BT ch.1 TIOB.

[bit11:10] TIOA1E: TIOA1 Output Select bits

Selects output for TIOA1.

bit		Description
Reading		Reads out the register value.
Writing	00	Does not produce output for BT ch.1 TIOA. [Initial value]
	01	Uses TIOA1_0 at the output pin of BT ch.1 TIOA.
	10	Uses TIOA1_1 at the output pin of BT ch.1 TIOA.
	11	Uses TIOA1_2 at the output pin of BT ch.1 TIOA.

[bit9:8] TIOA1S: TIOA1 Input Select bits

Selects input for TIOA1.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses TIOA1_0 at the input pin of BT ch.1 TIOA. [Initial value]
	01	Same as Writing 00.
	10	Uses TIOA1_1 at the input pin of BT ch.1 TIOA.
	11	Uses TIOA1_2 at the input pin of BT ch.1 TIOA.

[bit7] Reserved: Reserved bit

"0b0" is read out from this bit.
When writing this bit, set it to "0b0".

[bit6:4] TIOB0S: TIOB0 Input Select bits

Selects input for TIOB0.

bit		Description
Reading		Reads out the register value.
Writing	000	Uses TIOB0_0 at the input pin of BT ch.0 TIOB. [Initial value]
	001	Same as Writing 000.
	010	Uses TIOB0_1 at the input pin of BT ch.0 TIOB.
	011	Uses TIOB0_2 at the input pin of BT ch.0 TIOB.
	100	Setting is prohibited.
	101	Setting is prohibited.
	110	Uses SUBOUT at the input pin of BT ch.0 TIOB.
	111	Uses at the pin for measuring trimming of the high-speed CR frequency division clock.

.

[bit3:2] TIOA0E: TIOA0 Output Select bits

Selects output for TIOA0.

bit		Description
Reading		Reads out the register value.
Writing	00	Produces output for BT ch.0 TIOA. [Initial value]
	01	Uses TIOA0_0 at the output pin of BT ch.0 TIOA.
	10	Uses TIOA0_1 at the output pin of BT ch.0 TIOA.
	11	Uses TIOA0_2 at the output pin of BT ch.0 TIOA.

[bit1:0] Reserved: Reserved bits

"0b00" is read out from these bits.
When writing these bits, set them to "0b00".

<Notes>

- TIOA
  Even channels are for output only.
  Odd channels are for both input and output.
- TIOB
  Input only.
- TIOA1, TIOA3, TIOA5, TIOA7(odd number of "A") are not bidirectional pins so that choose either input pin or output pin for them.
  When output is chosen for odd TIOA channel, input setting will be ignored.

Example1: Use TIOA1 as an output pin:
When TIOA1 is output to TIOA1_0, select EPFR04:TIOA1E = 01.
When TIOA1 is output to TIOA1_1, select EPFR04:TIOA1E = 10.
When TIOA1 is output to TIOA1_2, select EPFR04:TIOA1E = 11.
Settings for EPFR04:TIOA1S will be ignored.
Select ADE=0, PFR=1 for selected pins (DDR will be ignored).
All the output of other peripheral function pins which are also used by selected pins must be OFF.

Example2: When TIOA1 is used as an input pin:
Select EPFR04:TIOA1E = 00.
When TIOA1 is input from TIOA1_0, select EPFR04:TIOA1S = 00 or 01.
When TIOA1 is input from TIOA1_1, select EPFR04:TIOA1S = 10.
When TIOA1 is input from TIOA1_2, select EPFR04:TIOA1S = 11.
Select ADE=0, PFR=1 for selected pins (DDR will be ignored).
All the output of other peripheral function pins which are also used by selected pins must be OFF.

* When a pin is set to input, the output of the pin which is also used by selected pins (GPIO, other peripheral function pins) can be input to the base timer as Feedback with a setting not described above.

This register is not initialized by deep standby transition reset.

4.13. Extended Pin Function Setting Register 05 (EPFR05)

The EPFR05 register assigns functions to pins of ch.4, ch.5, ch.6, and ch.7 of the base timer.

■ Register Configuration

bit	31	30	29	28	27	26	25	24
Field	Reserved		TIOB7S		TIOA7E		TIOA7S	
Attribute	-		R/W		R/W		R/W	
Initial value	-		00		00		00	

bit	23	22	21	20	19	18	17	16
Field	Reserved		TIOB6S		TIOA6E		Reserved	
Attribute	-		R/W		R/W		-	
Initial value	-		00		00		-	

bit	15	14	13	12	11	10	9	8
Field	Reserved		TIOB5S		TIOA5E		TIOA5S	
Attribute	-		R/W		R/W		R/W	
Initial value	-		00		00		00	

bit	7	6	5	4	3	2	1	0
Field	Reserved		TIOB4S		TIOA4E		Reserved	
Attribute	-		R/W		R/W		-	
Initial value	-		00		00		-	

■ Register Function

[bit31:30] Reserved: Reserved bits

"0b00" is read out from these bits.
When writing these bits, set them to "0b00".

[bit29:28] TIOB7S: TIOB7 Input Select bits

Selects input for TIOB7.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses TIOB7_0 at the input pin of BT ch.7 TIOB. [Initial value]
	01	Same as Writing 00.
	10	Uses TIOB7_1 at the input pin of BT ch.7 TIOB.
	11	Uses TIOB7_2 at the input pin of BT ch.7 TIOB.

[bit27:26] TIOA7E: TIOA7 Output Select bits

Selects output for TIOA7.

bit		Description
Reading		Reads out the register value.
Writing	00	Does not produce the output of the BT ch.7 TIOA. [Initial value]
	01	Uses TIOA7_0 at the output pin of BT ch.7 TIOA.
	10	Uses TIOA7_1 at the output pin of BT ch.7 TIOA.
	11	Uses TIOA7_2 at the output pin of BT ch.7 TIOA.

[bit25:24] TIOA7S: TIOA7 Input Select bits

Selects input for TIOA7.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses TIOA7_0 at the input pin of BT ch.7 TIOA. [Initial value]
	01	Same as Writing 00.
	10	Uses TIOA7_1 at the input pin of BT ch.7 TIOA.
	11	Uses TIOA7_2 at the input pin of BT ch.7 TIOA.

[bit23:22] Reserved: Reserved bits

"0b00" is read out from these bits.
When writing these bits, set them to "0b00".

[bit21:20] TIOB6S: TIOB6 Input Select bits

Selects input for TIOB6.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses TIOB6_0 at the input pin of BT ch.6 TIOB. [Initial value]
	01	Same as Writing 00.
	10	Uses TIOB6_1 at the input pin of BT ch.6 TIOB.
	11	Uses TIOB6_2 at the input pin of BT ch.6 TIOB.

[bit19:18] TIOA6E: TIOA6 Output Select bits

Selects output for TIOA6.

bit		Description
Reading		Reads out the register value.
Writing	00	Does not produce the output of the BT ch.6 TIOA. [Initial value]
	01	Uses TIOA6_0 at the output pin of BT ch.6 TIOA.
	10	Uses TIOA6_1 at the output pin of BT ch.6 TIOA.
	11	Uses TIOA6_2 at the output pin of BT ch.6 TIOA.

[bit17:14] Reserved: Reserved bits

"0b0000" is read out from these bits.
When writing these bits, set them to "0b0000".

[bit13:12] TIOB5S: TIOB5 Input Select bits

Selects input for TIOB5.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses TIOB5_0 at the input pin of BT ch.5 TIOB. [Initial value]
	01	Same as Writing 00.
	10	Uses TIOB5_1 at the input pin of BT ch.5 TIOB.
	11	Uses TIOB5_2 at the input pin of BT ch.5 TIOB.

[bit11:10] TIOA5E: TIOA5 Output Select bits

Selects output for TIOA5.

bit11:10		Description
Reading		Reads out the register value.
Writing	00	Does not produce the output of the BT ch.5 TIOA. [Initial value]
	01	Uses TIOA5_0 at the output pin of BT ch.5 TIOA.
	10	Uses TIOA5_1 at the output pin of BT ch.5 TIOA.
	11	Uses TIOA5_2 at the output pin of BT ch.5 TIOA.

[bit9:8] TIOA5S: TIOA5 Input Select bits

Selects input for TIOA5.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses TIOA5_0 at the input pin of BT ch.5 TIOA. [Initial value]
	01	Same as Writing 00.
	10	Uses TIOA5_1 at the input pin of BT ch.5 TIOA.
	11	Uses TIOA5_2 at the input pin of BT ch.5 TIOA.

[bit7:6] Reserved: Reserved bits

"0b00" is read out from these bits.
When writing these bits, set them to "0b00".

[bit5:4] TIOB4S: TIOB4 Input Select bits

Selects input for TIOB4.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses TIOB4_0 at the input pin of BT ch.4 TIOB. [Initial value]
	01	Same as Writing 00.
	10	Uses TIOB4_1 at the input pin of BT ch.4 TIOB.
	11	Uses TIOB4_2 at the input pin of BT ch.4 TIOB.

[bit3:2] TIOA4E: TIOA4 Output Select bits

Selects output for TIOA4.

bit		Description
Reading		Reads out the register value.
Writing	00	Does not produce the output of the BT ch.4 TIOA. [Initial value]
	01	Uses TIOA4_0 at the output pin of BT ch.4 TIOA.
	10	Uses TIOA4_1 at the output pin of BT ch.4 TIOA.
	11	Uses TIOA4_2 at the output pin of BT ch.4 TIOA.

[bit1:0] Reserved: Reserved bits

"0b00" is read out from these bits.
When writing these bits, set them to "0b00".

<Notes>

- TIOA
  Even channels are for output only.
  Odd channels are for both input and output.
- TIOB
  Input only.
- TIOA1, TIOA3, TIOA5, TIOA7(odd number of "A") are not bidirectional pins so that choose either input pin or output pin for them.
  When output is chosen for odd TIOA channel, input setting will be ignored.

Example1: Use TIOA1 as an output pin:
When TIOA1 is output to TIOA1_0, select EPFR04:TIOA1E = 01.
When TIOA1 is output to TIOA1_1, select EPFR04:TIOA1E = 10.
When TIOA1 is output to TIOA1_2, select EPFR04:TIOA1E = 11.
Settings for EPFR04:TIOA1S will be ignored.
Select ADE=0, PFR=1 for selected pins (DDR will be ignored).
All the output of other peripheral function pins which are also used by selected pins must be OFF.

Example2: When TIOA1 is used as an input pin:
Select EPFR04:TIOA1E = 00.
When TIOA1 is input from TIOA1_0, select EPFR04:TIOA1S = 00 or 01.
When TIOA1 is input from TIOA1_1, select EPFR04:TIOA1S = 10.
When TIOA1 is input from TIOA1_2, select EPFR04:TIOA1S = 11.
Select ADE=0, PFR=1 for selected pins (DDR will be ignored).
All the output of other peripheral function pins which are also used by selected pins must be OFF.

* When a pin is set to input, the output of the pin which is also used by selected pins (GPIO, other peripheral function pins) can be input to the base timer as Feedback with a setting not described above.

This register is not initialized by deep standby transition reset.

4.14. Extended Pin Function Setting Register 06 (EPFR06)

The EPFR06 register assigns functions to external interrupt pins.

■ Register Configuration

bit	31	30	29	28	27	26	25	24
Field	EINT15S		EINT14S		EINT13S		EINT12S	
Attribute	R/W		R/W		R/W		R/W	
Initial value	00		00		00		00	

bit	23	22	21	20	19	18	17	16
Field	EINT11S		EINT10S		EINT09S		EINT08S	
Attribute	R/W		R/W		R/W		R/W	
Initial value	00		00		00		00	

bit	15	14	13	12	11	10	9	8
Field	EINT07S		EINT06S		EINT05S		EINT04S	
Attribute	R/W		R/W		R/W		R/W	
Initial value	00		00		00		00	

bit	7	6	5	4	3	2	1	0
Field	EINT03S		EINT02S		EINT01S		EINT00S	
Attribute	R/W		R/W		R/W		R/W	
Initial value	00		00		00		00	

■ Register Function

[bit31:30] EINT15S: External Interrupt Input Select bits

Selects input for EINT15.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses INT15_0 at the input pin of EINT ch.15. [Initial value]
	01	Same as Writing 00.
	10	Uses INT15_1 at the input pin of EINT ch.15.
	11	Uses INT15_2 at the input pin of EINT ch.15.

[bit29:28] EINT14S: External Interrupt Input Select bits

Selects input for EINT14.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses INT14_0 at the input pin of EINT ch.14. [Initial value]
	01	Same as Writing 00.
	10	Uses INT14_1 at the input pin of EINT ch.14.
	11	Uses INT14_2 at the input pin of EINT ch.14.

[bit27:26] EINT13S: External Interrupt Input Select bits

Selects input for EINT13.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses INT13_0 at the input pin of EINT ch.13. [Initial value]
	01	Same as Writing 00
	10	Uses INT13_1 at the input pin of EINT ch.13.
	11	Uses INT13_2 at the input pin of EINT ch.13.

[bit25:24] EINT12S: External Interrupt Input Select bits

Selects input for EINT12.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses INT12_0 at the input pin of EINT ch.12. [Initial value]
	01	Same as Writing 00.
	10	Uses INT12_1 at the input pin of EINT ch.12.
	11	Uses INT12_2 at the input pin of EINT ch.12.

[bit23:22] EINT11S: External Interrupt Input Select bits

Selects input for EINT11.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses INT11_0 at the input pin of EINT ch.11. [Initial value]
	01	Same as Writing 00.
	10	Uses INT11_1 at the input pin of EINT ch.11.
	11	Uses INT11_2 at the input pin of EINT ch.11.

[bit21:20] EINT10S: External Interrupt Input Select bits

Selects input for EINT10.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses INT10_0 at the input pin of EINT ch.10. [Initial value]
	01	Same as Writing 00.
	10	Uses INT10_1 at the input pin of EINT ch.10.
	11	Uses INT10_2 at the input pin of EINT ch.10.

[bit19:18] EINT09S: External Interrupt Input Select bits

Selects input for EINT09.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses INT09_0 at the input pin of EINT ch.9. [Initial value]
	01	Same as Writing 00.
	10	Uses INT09_1 at the input pin of EINT ch.9.
	11	Uses INT09_2 at the input pin of EINT ch.9.

[bit17:16] EINT08S: External Interrupt Input Select bits

Selects input for EINT08.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses INT08_0 at the input pin of EINT ch.8. [Initial value]
	01	Same as Writing 00.
	10	Uses INT08_1 at the input pin of EINT ch.8.
	11	Uses INT08_2 at the input pin of EINT ch.8.

[bit15:14] EINT07S: External Interrupt Input Select bits

Selects input for EINT07.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses INT07_0 at the input pin of EINT ch.7. [Initial value]
	01	Same as Writing 00.
	10	Uses INT07_1 at the input pin of EINT ch.7.
	11	Uses INT07_2 at the input pin of EINT ch.7.

[bit13:12] EINT06S: External Interrupt Input Select bits

Selects input for EINT06.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses INT06_0 at the input pin of EINT ch.6. [Initial value]
	01	Same as Writing 00.
	10	Uses INT06_1 at the input pin of EINT ch.6.
	11	Uses INT06_2 at the input pin of EINT ch.6.

[bit11:10] EINT05S: External Interrupt Input Select bits

Selects input for EINT05.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses INT05_0 at the input pin of EINT ch.5. [Initial value]
	01	Same as Writing 00.
	10	Uses INT05_1 at the input pin of EINT ch.5.
	11	Uses INT05_2 at the input pin of EINT ch.5.

[bit9:8] EINT04S: External Interrupt Input Select bits

Selects input for EINT04.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses INT04_0 at the input pin of EINT ch.4. [Initial value]
	01	Same as Writing 00.
	10	Uses INT04_1 at the input pin of EINT ch.4.
	11	Uses INT04_2 at the input pin of EINT ch.4.

[bit7:6] EINT03S: External Interrupt Input Select bits

Selects input for EINT03.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses INT03_0 at the input pin of EINT ch.3. [Initial value]
	01	Same as Writing 00.
	10	Uses INT03_1 at the input pin of EINT ch.3.
	11	Uses INT03_2 at the input pin of EINT ch.3.

[bit5:4] EINT02S: External Interrupt Input Select bits

Selects input for EINT02.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses INT02_0 at the input pin of EINT ch.2. [Initial value]
	01	Same as Writing 00.
	10	Uses INT02_1 at the input pin of EINT ch.2.
	11	Uses INT02_2 at the input pin of EINT ch.2.

[bit3:2] EINT01S: External Interrupt Input Select bits

Selects input for EINT01.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses INT01_0 at the input pin of EINT ch.1. [Initial value]
	01	Same as Writing 00.
	10	Uses INT01_1 at the input pin of EINT ch.1.
	11	Uses INT01_2 at the input pin of EINT ch.1.

[bit1:0] EINT00S: External Interrupt Input Select bits

Selects input for EINT00.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses INT00_0 at the input pin of EINT ch.0. [Initial value]
	01	Same as Writing 00.
	10	Uses INT00_1 at the input pin of EINT ch.0.
	11	Uses INT00_2 at the input pin of EINT ch.0.

<Note>

This register is not initialized by deep standby transition reset.

4.15. Extended Pin Function Setting Register 07 (EPFR07)

The EPFR07 register assigns functions of multi-function serial ch.0 to ch.3.

■ Register Configuration

bit	31	30	29	28	27	26	25	24
Field	Reserved				SCK3B		SOT3B	
Attribute	-				R/W		R/W	
Initial value	-				00		00	

bit	23	22	21	20	19	18	17	16
Field	SIN3S		SCK2B		SOT2B		SIN2S	
Attribute	R/W		R/W		R/W		R/W	
Initial value	00		00		00		00	

bit	15	14	13	12	11	10	9	8
Field	SCK1B		SOT1B		SIN1S		SCK0B	
Attribute	R/W		R/W		R/W		R/W	
Initial value	00		00		00		00	

bit	7	6	5	4	3	2	1	0
Field	SOT0B		SIN0S		Reserved			
Attribute	R/W		R/W		-			
Initial value	00		00		-			

■ Register Function

[bit31:28] Reserved: Reserved bits

"0b0000" is read from these bits.
When writing these bits, set them to "0b0000".

[bit27:26] SCK3B: SCK3 Input/Output Select bits

Selects input/output for SCK3.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses SCK3_0 at the input pin of MFS ch.3 SCK. Does not produce output. [Initial value]
	01	Uses SCK3_0 at the input pin of MFS ch.3 SCK. Uses SCK3_0 at the output pin.
	10	Uses SCK3_1 at the input pin of MFS ch.3 SCK. Uses SCK3_1 at the output pin.
	11	Uses SCK3_2 at the input pin of MFS ch.3 SCK. Uses SCK3_2 at the output pin.

[bit25:24] SOT3B: SOT3 Input/Output Select bits

Selects input/output for SOT3.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses SOT3_0 at the input pin of MFS ch.3 SOT. Does not produce output. [Initial value]
	01	Uses SOT3_0 at the input pin of MFS ch.3 SOT. Uses SOT3_0 at the output pin.
	10	Uses SOT3_1 at the input pin of MFS ch.3 SOT. Uses SOT3_1 at the output pin.
	11	Uses SOT3_2 at the input pin of MFS ch.3 SOT. Uses SOT3_2 at the output pin.

[bit23:22] SIN3S: SIN3 Input Select bits

Selects input for SIN3.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses SIN3_0 at the input pin of MFS ch.3 SIN. [Initial value]
	01	Same as Writing 00.
	10	Uses SIN3_1 at the input pin of MFS ch.3 SIN.
	11	Uses SIN3_2 at the input pin of MFS ch.3 SIN.

[bit21:20] SCK2B: SCK2 Input/Output Select bits

Selects input/output for SCK2.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses SCK2_0 at the input pin of MFS ch.2 SCK. Does not produce output. [Initial value]
	01	Uses SCK2_0 at the input pin of MFS ch.2 SCK. Uses SCK2_0 at the output pin.
	10	Uses SCK2_1 at the input pin of MFS ch.2 SCK. Uses SCK2_1 at the output pin.
	11	Uses SCK2_2 at the input pin of MFS ch.2 SCK. Uses SCK2_2 at the output pin.

[bit19:18] SOT2B: SOT2 Input/Output Select bits

Selects input/output for SOT2.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses SOT2_0 at the input pin of MFS ch.2 SOT. Does not produce output. [Initial value]
	01	Uses SOT2_0 at the input pin of MFS ch.2 SOT. Uses SOT2_0 at the output pin.
	10	Uses SOT2_1 at the input pin of MFS ch.2 SOT. Uses SOT2_1 at the output pin.
	11	Uses SOT2_2 at the input pin of MFS ch.2 SOT. Uses SOT2_2 at the output pin.

[bit17:16] SIN2S: SIN2 Input Select bits

Selects input for SIN2.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses SIN2_0 at the input pin of MFS ch.2 SIN. [Initial value]
	01	Same as Writing 00.
	10	Uses SIN2_1 at the input pin of MFS ch.2 SIN.
	11	Uses SIN2_2 at the input pin of MFS ch.2 SIN.

[bit15:14] SCK1B: SCK1 Input/Output Select bits

Selects input/output for SCK1.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses SCK1_0 at the input pin of MFS ch.1 SCK. Does not produce output. [Initial value]
	01	Uses SCK1_0 at the input pin of MFS ch.1 SCK. Uses SCK1_0 at the output pin.
	10	Uses SCK1_1 at the input pin of MFS ch.1 SCK. Uses SCK1_1 at the output pin.
	11	Uses SCK1_2 at the input pin of MFS ch.1 SCK. Uses SCK1_2 at the output pin.

[bit13:12] SOT1B: SOT1 Input/Output Select bits

Selects input/output for SOT1.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses SOT1_0 at the input pin of MFS ch.1 SOT. Does not produce output. [Initial value]
	01	Uses SOT1_0 at the input pin of MFS ch.1 SOT. Uses SOT1_0 at the output pin.
	10	Uses SOT1_1 at the input pin of MFS ch.1 SOT. Uses SOT1_1 at the output pin.
	11	Uses SOT1_2 at the input pin of MFS ch.1 SOT. Uses SOT1_2 at the output pin.

[bit11:10] SIN1S: SIN1 Input Select bits

Selects input for SIN1.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses SIN1_0 at the input pin of MFS ch.1 SIN. [Initial value]
	01	Same as Writing 00.
	10	Uses SIN1_1 at the input pin of MFS ch.1 SIN.
	11	Uses SIN1_2 at the input pin of MFS ch.1 SIN.

[bit9:8] SCK0B: SCK0 Input/Output Select bits

Selects input/output for SCK0.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses SCK0_0 at the input pin of MFS ch.0 SCK. Does not produce output. [Initial value]
	01	Uses SCK0_0 at the input pin of MFS ch.0 SCK. Uses SCK0_0 at the output pin.
	10	Uses SCK0_1 at the input pin of MFS ch.0 SCK. Uses SCK0_1 at the output pin.
	11	Uses SCK0_2 at the input pin of MFS ch.0 SCK. Uses SCK0_2 at the output pin.

[bit7:6] SOT0B: SOT0 Input/Output Select bits

Selects input/output for SOT0.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses SOT0_0 at the input pin of MFS ch.0 SOT. Does not produce output. [Initial value]
	01	Uses SOT0_0 at the input pin of MFS ch.0 SOT. Uses SOT0_0 at the output pin.
	10	Uses SOT0_1 at the input pin of MFS ch.0 SOT. Uses SOT0_1 at the output pin.
	11	Uses SOT0_2 at the input pin of MFS ch.0 SOT. Uses SOT0_2 at the output pin.

[bit5:4] SIN0S: SIN0 Input Select bits

Selects input for SIN0.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses SIN0_0 at the input pin of MFS ch.0 SIN. [Initial value]
	01	Same as Writing 00.
	10	Uses SIN0_1 at the input pin of MFS ch.0 SIN.
	11	Uses SIN0_2 at the input pin of MFS ch.0 SIN.

[bit3:0] Reserved: Reserved bits

"0b0000" is read from these bits.

When writing these bits, set them to "0b0000".

<Note>

This register is not initialized by deep standby transition reset.

4.16. Extended Pin Function Setting Register 08 (EPFR08)

The EPFR08 register assigns functions of multi-function serial ch.4 to ch.7.

■ Register Configuration

bit	31	30	29	28	27	26	25	24
Field	CTS5S		RTS5E		SCK7B		SOT7B	
Attribute	R/W		R/W		R/W		R/W	
Initial value	00		00		00		00	

bit	23	22	21	20	19	18	17	16
Field	SIN7S		SCK6B		SOT6B		SIN6S	
Attribute	R/W		R/W		R/W		R/W	
Initial value	00		00		00		00	

bit	15	14	13	12	11	10	9	8
Field	SCK5B		SOT5B		SIN5S		SCK4B	
Attribute	R/W		R/W		R/W		R/W	
Initial value	00		00		00		00	

bit	7	6	5	4	3	2	1	0
Field	SOT4B		SIN4S		CTS4S		RTS4E	
Attribute	R/W		R/W		R/W		R/W	
Initial value	00		00		00		00	

■ Register Function

[bit31:30] CST5S: CTS5 Input/Output Select bits

Selects input/output for CTS5.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses CTS5_0 at the input pin of MFS ch.5 CTS. [Initial value]
	01	Same as Writing 00.
	10	Uses CTS5_1 at the input pin of MFS ch.5 CTS.
	11	Uses CTS5_2 at the input pin of MFS ch.5 CTS.

[bit29:28] RTS5E: RTS5 Input/Output Select bits

Selects input/output for RTS5.

bit		Description
Reading		Reads out the register value.
Writing	00	Does not produce output of MFS ch.5 RTS. [Initial value]
	01	Uses RTS5_0 at the output pin of MFS ch.5 RTS.
	10	Uses RTS5_1 at the output pin of MFS ch.5 RTS.
	11	Uses RTS5_2 at the output pin of MFS ch.5 RTS.

[bit27:26] SCK7B: SCK7 Input/Output Select bits

Selects input/output for SCK7.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses SCK7_0 at the input pin of MFS ch.7 SCK. Does not produce output. [Initial value]
	01	Uses SCK7_0 at the input pin of MFS ch.7 SCK. Uses SCK7_0 at the output pin.
	10	Uses SCK7_1 at the input pin of MFS ch.7 SCK. Uses SCK7_1 at the output pin.
	11	Uses SCK7_2 at the input pin of MFS ch.7 SCK. Uses SCK7_2 at the output pin.

[bit25:24] SOT7B: SOT7 Input/Output Select bits

Selects input/output for SOT7.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses SOT7_0 at the input pin of MFS ch.7 SOT. Does not produce output. [Initial value]
	01	Uses SOT7_0 at the input pin of MFS ch.7 SOT. Uses SOT7_0 at the output pin.
	10	Uses SOT7_1 at the input pin of MFS ch.7 SOT. Uses SOT7_1 at the output pin.
	11	Uses SOT7_2 at the input pin of MFS ch.7 SOT. Uses SOT7_2 at the output pin.

[bit23:22] SIN7S: SIN7 Input Select bits

Selects input for SIN7.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses SIN7_0 at the input pin of MFS ch.7 SIN. [Initial value]
	01	Same as Writing 00.
	10	Uses SIN7_1 at the input pin of MFS ch.7 SIN.
	11	Uses SIN7_2 at the input pin of MFS ch.7 SIN.

[bit21:20] SCK6B: SCK6 Input/Output Select bits

Selects input/output for SCK6.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses SCK6_0 at the input pin of MFS ch.6 SCK. Does not produce output. [Initial value]
	01	Uses SCK6_0 at the input pin of MFS ch.6 SCK. Uses SCK6_0 at the output pin.
	10	Uses SCK6_1 at the input pin of MFS ch.6 SCK. Uses SCK6_1 at the output pin.
	11	Uses SCK6_2 at the input pin of MFS ch.6 SCK. Uses SCK6_2 at the output pin.

[bit19:18] SOT6B: SOT6 Input/Output Select bits

Selects input/output for SOT6.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses SOT6_0 at the input pin of MFS ch.6 SOT. Does not produce output. [Initial value]
	01	Uses SOT6_0 at the input pin of MFS ch.6 SOT. Uses SOT6_0 at the output pin.
	10	Uses SOT6_1 at the input pin of MFS ch.6 SOT. Uses SOT6_1 at the output pin.
	11	Uses SOT6_2 at the input pin of MFS ch.6 SOT. Uses SOT6_2 at the output pin.

[bit17:16] SIN6S: SIN6 Input Select bits

Selects input for SIN6.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses SIN6_0 at the input pin of MFS ch.6 SIN. [Initial value]
	01	Same as Writing 00.
	10	Uses SIN6_1 at the input pin of MFS ch.6 SIN.
	11	Uses SIN6_2 at the input pin of MFS ch.6 SIN.

[bit15:14] SCK5B: SCK5 Input/Output Select bits

Selects input/output for SCK5.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses SCK5_0 at the input pin of MFS ch.5 SCK. Does not produce output. [Initial value]
	01	Uses SCK5_0 at the input pin of MFS ch.5 SCK. Uses SCK5_0 at the output pin.
	10	Uses SCK5_1 at the input pin of MFS ch.5 SCK. Uses SCK5_1 at the output pin.
	11	Uses SCK5_2 at the input pin of MFS ch.5 SCK. Uses SCK5_2 at the output pin.

[bit13:12] SOT5B: SOT5 Input/Output Select bits

Selects input/output for SOT5.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses SOT5_0 at the input pin of MFS ch.5 SOT. Does not produce output. [Initial value]
	01	Uses SOT5_0 at the input pin of MFS ch.5 SOT. Uses SOT5_0 at the output pin.
	10	Uses SOT5_1 at the input pin of MFS ch.5 SOT. Uses SOT5_1 at the output pin.
	11	Uses SOT5_2 at the input pin of MFS ch.5 SOT. Uses SOT5_2 at the output pin.

[bit11:10] SIN5S: SIN5 Input Select bits

Selects input for SIN5.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses SIN5_0 at the input pin of MFS ch.5 SIN. [Initial value]
	01	Same as Writing 00.
	10	Uses SIN5_1 at the input pin of MFS ch.5 SIN.
	11	Uses SIN5_2 at the input pin of MFS ch.5 SIN.

[bit9:8] SCK4B: SCK4 Input/Output Select bits

Selects input/output for SCK4.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses SCK4_0 at the input pin of MFS ch.4 SCK. Does not produce output. [Initial value]
	01	Uses SCK4_0 at the input pin of MFS ch.4 SCK. Uses SCK4_0 at the output pin.
	10	Uses SCK4_1 at the input pin of MFS ch.4 SCK. Uses SCK4_1 at the output pin.
	11	Uses SCK4_2 at the input pin of MFS ch.4 SCK. Uses SCK4_2 at the output pin.

[bit7:6] SOT4B: SOT4 Input/Output Select bits

Selects input/output for SOT4.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses SOT4_0 at the input pin of MFS ch.4 SOT. Does not produce output. [Initial value]
	01	Uses SOT4_0 at the input pin of MFS ch.4 SOT. Uses SOT4_0 at the output pin.
	10	Uses SOT4_1 at the input pin of MFS ch.4 SOT. Uses SOT4_1 at the output pin.
	11	Uses SOT4_2 at the input pin of MFS ch.4 SOT. Uses SOT4_2 at the output pin.

[bit5:4] SIN4S: SIN4 Input Select bits

Selects input for SIN4.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses SIN4_0 at the input pin of MFS ch.4 SIN. [Initial value]
	01	Same as Writing 00.
	10	Uses SIN4_1 at the input pin of MFS ch.4 SIN.
	11	Uses SIN4_2 at the input pin of MFS ch.4 SIN.

[bit3:2] CTS4S: CTS4 Input Select bits

Selects input for CTS4.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses CTS4_0 at the input pin of MFS ch.4 CTS. [Initial value]
	01	Same as Writing 00.
	10	Uses CTS4_1 at the input pin of MFS ch.4 CTS.
	11	Uses CTS4_2 at the input pin of MFS ch.4 CTS.

[bit1:0] RTS4E: RTS4 Output Select bits

Selects output for RTS4.

bit		Description
Reading		Reads out the register value.
Writing	00	Does not produce output for MFS ch.4 RTS. [Initial value]
	01	Uses RTS4_0 at the output pin of MFS ch.4 RTS.
	10	Uses RTS4_1 at the output pin of MFS ch.4 RTS.
	11	Uses RTS4_2 at the output pin of MFS ch.4 RTS.

<Note>

This register is not initialized by deep standby transition reset.

4.17. Extended Pin Function Setting Register 09 (EPFR09)

The EPFR09 register assigns functions to CAN, ADC trigger, and QPRC peripheral pins.

■ Register Configuration

bit	31	30	29	28	27	26	25	24
Field	CTX1E		CRX1S		CTX0E		CRX0S	
Attribute	R/W		R/W		R/W		R/W	
Initial value	00		00		00		00	

bit	23	22	21	20	19	18	17	16
Field	ADTRG2S				ADTRG1S			
Attribute	R/W				R/W			
Initial value	0000				0000			

bit	15	14	13	12	11	10	9	8
Field	ADTRG0S				QZIN1S		QBIN1S	
Attribute	R/W				R/W		R/W	
Initial value	0000				00		00	

bit	7	6	5	4	3	2	1	0
Field	QAIN1S		QZIN0S		QBIN0S		QAIN0S	
Attribute	R/W		R/W		R/W		R/W	
Initial value	00		00		00		00	

■ Register Function

[bit31:30] CTX1E: CTX1E Output Select bits

Selects output for CAN TX1.

bit		Description
Reading		Reads out the register value.
Writing	00	Does not produce output for CAN ch.1 TX. [Initial value]
	01	Sets the output pin of CAN ch.1 TX to TX1_0.
	10	Sets the output pin of CAN ch.1 TX to TX1_1.
	11	Sets the output pin of CAN ch.1 TX to TX1_2.

[bit29:28] CRX1S: CRX1S Input Select bits

Selects input for CAN RX1.

bit		Description
Reading		Reads out the register value.
Writing	00	Sets the input pin of CAN ch.1 RX to RX1_0. [Initial value]
	01	Same as Writing 00.
	10	Sets the input pin of CAN ch.1 RX to RX1_1.
	11	Sets the input pin of CAN ch.1 RX to RX1_2.

[bit27:26] CTX0E: CTX0E Output Select bits

Selects output for CAN TX0.

bit		Description
Reading		Reads out the register value.
Writing	00	Does not produce output for CAN ch.0 TX. [Initial value]
	01	Sets the output pin of CAN ch.0 TX to TX0_0.
	10	Sets the output pin of CAN ch.0 TX to TX0_1.
	11	Sets the output pin of CAN ch.0 TX to TX0_2.

[bit25:24] CRX0S: CRX0S Input Select bits

Selects input for CAN RX0.

bit		Description
Reading		Reads out the register value.
Writing	00	Sets the input pin of CAN ch.0 RX to RX0_0. [Initial value]
	01	Same as Writing 00.
	10	Sets the input pin of CAN ch.0 RX to RX0_1.
	11	Sets the input pin of CAN ch.0 RX to RX0_2.

[bit23:20] ADTRG2S: ADTRG2 Input Select bits

Selects input for ADTRG2.

bit		Description
Reading		Reads out the register value.
Writing	0000	Uses ADTG_0 at the input pin of ADC unit 2’s startup trigger. [Initial value]
	0001	Same as Writing 0000.
	0010	Uses ADTG_1 at the input pin of ADC unit 2’s startup trigger.
	0011	Uses ADTG_2 at the input pin of ADC unit 2’s startup trigger.
	0100	Uses ADTG_3 at the input pin of ADC unit 2’s startup trigger.
	0101	Uses ADTG_4 at the input pin of ADC unit 2’s startup trigger.
	0110	Uses ADTG_5 at the input pin of ADC unit 2’s startup trigger.
	0111	Uses ADTG_6 at the input pin of ADC unit 2’s startup trigger.
	1000	Uses ADTG_7 at the input pin of ADC unit 2’s startup trigger.
	1001	Uses ADTG_8 at the input pin of ADC unit 2’s startup trigger.
Writing other data		Setting is prohibited.

[bit19:16] ADTRG1S: ADTRG1 Input Select bits

Selects input for ADTRG1.

bit		Description
Reading		Reads out the register value.
Writing	0000	Uses ADTG_0 at the input pin of ADC unit 1’s startup trigger. [Initial value]
	0001	Same as Writing 0000.
	0010	Uses ADTG_1 at the input pin of ADC unit 1’s startup trigger.
	0011	Uses ADTG_2 at the input pin of ADC unit 1’s startup trigger.
	0100	Uses ADTG_3 at the input pin of ADC unit 1’s startup trigger.
	0101	Uses ADTG_4 at the input pin of ADC unit 1’s startup trigger.
	0110	Uses ADTG_5 at the input pin of ADC unit 1’s startup trigger.
	0111	Uses ADTG_6 at the input pin of ADC unit 1’s startup trigger.
	1000	Uses ADTG_7 at the input pin of ADC unit 1’s startup trigger.
	1001	Uses ADTG_8 at the input pin of ADC unit 1’s startup trigger.
Writing other data		Setting is prohibited.

[bit15:12] ADTRG0S: ADTRG0 Input Select bits

Selects input for ADTRG0.

bit		Description
Reading		Reads out the register value.
Writing	0000	Uses ADTG_0 at the input pin of ADC unit 0's startup trigger. [Initial value]
	0001	Same as Writing 0000.
	0010	Uses ADTG_1 at the input pin of ADC unit 0's startup trigger.
	0011	Uses ADTG_2 at the input pin of ADC unit 0's startup trigger.
	0100	Uses ADTG_3 at the input pin of ADC unit 0's startup trigger.
	0101	Uses ADTG_4 at the input pin of ADC unit 0's startup trigger.
	0110	Uses ADTG_5 at the input pin of ADC unit 0's startup trigger.
	0111	Uses ADTG_6 at the input pin of ADC unit 0's startup trigger.
	1000	Uses ADTG_7 at the input pin of ADC unit 0's startup trigger.
	1001	Uses ADTG_8 at the input pin of ADC unit 0's startup trigger.
Writing other data		Setting is prohibited.

[bit11:10] QZIN1S: QZIN1S Input Select bits

Selects input for QPRC ZIN1.

bit11:10		Description
Reading		Reads out the register value.
Writing	00	Uses ZIN1_0 at the input pin of QPRC ch.1's ZIN. [Initial value]
	01	Same as Writing 00.
	10	Uses ZIN1_1 at the input pin of QPRC ch.1's ZIN.
	11	Uses ZIN1_2 at the input pin of QPRC ch.1's ZIN.

[bit9:8] QBIN1S: QBIN1S Input Select bits

Selects input for QPRC BIN1.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses BIN1_0 at the input pin of QPRC ch.1's BIN. [Initial value]
	01	Same as Writing 00.
	10	Uses BIN1_1 at the input pin of QPRC ch.1's BIN.
	11	Uses BIN1_2 at the input pin of QPRC ch.1's BIN.

[bit7:6] QAIN1S: QAIN1S Input Select bits

Selects input for QPRC AIN1.

bit7:6		Description
Reading		Reads out the register value.
Writing	00	Uses AIN1_0 at the input pin of QPRC ch.1's AIN. [Initial value]
	01	Same as Writing 00.
	10	Uses AIN1_1 at the input pin of QPRC ch.1's AIN.
	11	Uses AIN1_2 at the input pin of QPRC ch.1's AIN.

[bit5:4] QZIN0S: QZIN0S Input Select bits

Selects input for QPRC ZIN0.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses ZIN0_0 at the input pin of QPRC ch.0's ZIN. [Initial value]
	01	Same as Writing 00.
	10	Uses ZIN0_1 at the input pin of QPRC ch.0's ZIN.
	11	Uses ZIN0_2 at the input pin of QPRC ch.0's ZIN.

[bit3:2] QBIN0S: QBIN0S Input Select bits

Selects input for QPRC BIN0.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses BIN0_0 at the input pin of QPRC ch.0's BIN. [Initial value]
	01	Same as Writing 00.
	10	Uses BIN0_1 at the input pin of QPRC ch.0's BIN.
	11	Uses BIN0_2 at the input pin of QPRC ch.0's BIN.

[bit1:0] QAIN0S: QAIN0S Input Select bits

Selects input for QPRC AIN0.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses AIN0_0 at the input pin of QPRC ch.0's AIN. [Initial value]
	01	Same as Writing 00.
	10	Uses AIN0_1 at the input pin of QPRC ch.0's AIN.
	11	Uses AIN0_2 at the input pin of QPRC ch.0's AIN.

<Note>

This register is not initialized by deep standby transition reset.

4.18. Extended Pin Function Setting Register 10 (EPFR10)

The EPFR10 register assigns functions to external bus peripheral pins.

■ Register Configuration

bit	31	30	29	28	27	26	25	24
Field	UEA24E	UEA23E	UEA22E	UEA21E	UEA20E	UEA19E	UEA18E	UEA17E
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	UEA16E	UEA15E	UEA14E	UEA13E	UEA12E	UEA11E	UEA10E	UEA09E
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	UEA08E	UEAOOE	UECS7E	UECS6E	UECS5E	UECS4E	UECS3E	UECS2E
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	UECS1E	UEFLSE	UEOEXE	UEDQME	UEWEXE	UECLKE	UEDTHB	UEDEFB
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

■ Register Function

[bit31] UEA24E: UEA24E Output Select bit

Selects output for external bus Address24.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output for user external bus MAD24. [Initial value]
	1	Produces output for user external bus MAD24.

[bit30] UEA23E: UEA23E Output Select bit

Selects output for external bus Address23.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output for user external bus MAD23. [Initial value]
	1	Produces output for user external bus MAD23.

[bit29] UEA22E: UEA22E Output Select bit

Selects output for external bus Address22.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output for user external bus MAD22. [Initial value]
	1	Produces output for user external bus MAD22.

[bit28] UEA21E: UEA21E Output Select bit

Selects output for external bus Address21.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output for user external bus MAD21. [Initial value]
	1	Produces output for user external bus MAD21.

[bit27] UEA20E: UEA20E Output Select bit

Selects output for external bus Address20.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output for user external bus MAD20. [Initial value]
	1	Produces output for user external bus MAD20.

[bit26] UEA19E: UEA19E Output Select bit

Selects output for external bus Address19.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output for user external bus MAD19. [Initial value]
	1	Produce output for user external bus MAD19.

[bit25] UEA18E: UEA18E Output Select bit

Selects output for external bus Address18.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output for user external bus MAD18. [Initial value]
	1	Produces output for user external bus MAD18.

[bit24] UEA17E: UEA17E Output Select bit

Selects output for external bus Address17.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output for user external bus MAD17. [Initial value]
	1	Produces output for user external bus MAD17.

[bit23] UEA16E: UEA16E Output Select bit

Selects output for external bus Address16.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output for user external bus MAD16. [Initial value]
	1	Produces output for user external bus MAD16.

[bit22] UEA15E: UEA15E Output Select bit

Selects output for external bus Address15.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output for user external bus MAD15. [Initial value]
	1	Produces output for user external bus MAD15.

[bit21] UEA14E: UEA14E Output Select bit

Selects output for external bus Address14.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output for user external bus MAD14. [Initial value]
	1	Produces output for user external bus MAD14.

[bit20] UEA13E: UEA13E Output Select bit

Selects output for external bus Address13.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output for user external bus MAD13. [Initial value]
	1	Produces output for user external bus MAD13.

[bit19] UEA12E: UEA12E Output Select bit

Selects output for external bus Address12.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output for user external bus MAD12. [Initial value]
	1	Produces output for user external bus MAD12.

[bit18] UEA11E: UEA11E Output Select bit

Selects output for external bus Address11.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output for user external bus MAD11. [Initial value]
	1	Produces output for user external bus MAD11.

[bit17] UEA10E: UEA10E Output Select bit

Selects output for external bus Address10.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output for user external bus MAD10. [Initial value]
	1	Produces output for user external bus MAD10.

[bit16] UEA09E: UEA09E Output Select bit

Selects output for external bus Address09.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output for user external bus MAD09. [Initial value]
	1	Produces output for user external bus MAD09.

[bit15] UEA08E: UEA08E Output Select bit

Selects output for external bus Address08.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output for user external bus MAD08. [Initial value]
	1	Produces output for user external bus MAD08.

[bit14] UEAOOE: UEAOOE Output Select bit

Selects output for external bus Address00.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output for user external bus MAD00. [Initial value]
	1	Produces output for user external bus MAD00.

[bit13] UECS7E: UECS7E Output Select bit

Selects output for external bus CS7.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output for user external bus MCSX7. [Initial value]
	1	Produces output for user external bus MCSX7.

[bit12] UECS6E: UECS6E Output Select bit

Selects output for external bus CS6.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output for user external bus MCSX6. [Initial value]
	1	Selects output for user external bus MCSX6.

[bit11] UECS5E: UECS5E Output Select bit

Selects output for external bus CS5.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output for user external bus MCSX5. [Initial value]
	1	Produces output for user external bus MCSX5.

[bit10] UECS4E: UECS4E Output Select bit

Selects output for external bus CS4.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output for user external bus MCSX4. [Initial value]
	1	Produces output for user external bus MCSX4.

[bit9] UECS3E: UECS3E Output Select bit

Selects output for external bus CS3.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output for user external bus MCSX3. [Initial value]
	1	Produces output for user external bus MCSX3.

[bit8] UECS2E: UECS2E Output Select bit

Selects output for external bus CS2.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output for user external bus MCSX2. [Initial value]
	1	Produces output for user external bus MCSX2.

[bit7] UECS1E: UECS1E Output Select bit

Selects output for external bus CS1.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output for user external bus MCSX1. [Initial value]
	0	Produces output for user external bus MCSX1.

[bit6] UEFLSE: UEFLSE Output Select bit

Selects output for external bus NAND-Flash control signal.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output for user external bus MNALE, MNCLE, MNWEX, or MNREX. [Initial value]
	1	Produces output for user external bus MNALE, MNCLE, MNWEX, and MNREX.

[bit5] UEOEXE: UEOEXE Output Select bit

Selects output for external bus OEX.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output for user external bus MOEX. [Initial value]
	1	Produces output for user external bus MOEX.

[bit4] UEDQME: UEDQME Output Select bit

Selects output for external bus DQM.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output for user external bus MDQM1or MDQM0. [Initial value]
	1	Produces output for user external bus MDQM1 and MDQM0.

[bit3] UEWEXE: UEWEXE Output Select bit

Selects output for external bus WEX.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output for user external bus MWEX. [Initial value]
	1	Produces output for user external bus MWEX.

[bit2] UECLKE: UECLKE Output Select bit

Selects output for external bus clock.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output for user external bus MCLKOUT. [Initial value]
	1	Produces output for user external bus MCLKOUT.

[bit1] UEDTHB: UEDTHB Input/Output Select bit

Selects input/output for external bus data.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output for user external bus WD15 to WD08. [Initial value] Input of user external bus RD15 to RD08 is connected to the pin.
	1	Produces output for user external bus WD15 to WD08. Input of user external bus RD15 to RD08 is connected to the pin.

[bit0] UEDEFB: UEDEFB Input/Output Select bit

Selects input/output for external bus signal.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output for user external bus MAD07 to MAD01. Does not produce output for user external bus MCSX0. Does not produce output for user external bus WD07 to WD00. Input of user external bus RD07 to RD00 is connected to the pin. [Initial value]
	1	Produces output for user external bus MAD07 to MAD01. Produces output for user external bus MCSX0. Produces output for user external bus WD7 to WD0. Input of user external bus RD7 to RD0 is connected to the pin.

<Notes>

- I/O selection of the external bus data[15:7] can be controlled collectively with EPFR10.bit1.
  I/O selection of the external bus data[15:7] can be controlled by each bit also with EPFR11.bit[24:17]
  EPFR10.bit1 setting has the higher priority than EPFR11.bit[24:17] setting.
  To control I/O selection by setting EPFR11.bit[24:17], it is necessary to set EPFR10.bit1=0.
- I/O selection of the external bus address[7:1], MCSX0, and external bus data[7:0] can be controlled collectively with EPFR10.bit0.
  I/O selection of the external bus address[7:1], MCSX0, and external bus data[7:0] can be controlled by each bit also with EPFR11.bit[16:1].
  EPFR10.bit0 setting has the higher priority than EPFR11.bit[16:1] setting.
  To control I/O selection by setting EPFR11.bit[16:1], it is necessary to set EPFR10.bit0=0.
- This register is not initialized by deep standby transition reset.

4.19. Extended Pin Function Setting Register 11 (EPFR11)

The EPFR11 register assigns functions to external bus peripheral pins.

■ Register Configuration

bit	31	30	29	28	27	26	25	24
Field	Reserved						UERLC	UED15B
Attribute							R/W	R/W
Initial value							0	0

bit	23	22	21	20	19	18	17	16
Field	UED14B	UED13B	UED12B	UED11B	UED10B	UED09B	UED08B	UED07B
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	15	14	13	12	11	10	9	8
Field	UED06B	UED05B	UED04B	UED03B	UED02B	UED01B	UED00B	UEA07E
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	UEA06E	UEA05E	UEA04E	UEA03E	UEA02E	UEA01E	UECS0E	UEALEE
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

■ Register Function

[bit31:26] Reserved: Reserved bits

"0b000000" is read from these bits.
When writing these bits, set them to "0b000000".

[bit25] UERLC: UERLC relocation select bit

Selects relocation of the external bus pin.

bit		Description
Reading		Reads out the register value.
Writing	0	Uses relocate number "0" for the user external bus. [Initial value]
	1	Uses relocate number "1" for the user external bus.

Note: It depends on the product which relocate number is existed for user external bus.
See "Data sheet" of each product for the relocate number.

[bit24] UED15B: UED15B Input/Output Select bit

Selects input/output for external bus data 15.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output for user external bus MADATA15. [Initial value] Input of user external bus MADATA15 is connected to the pin.
	1	Produces output for user external bus MADATA15. Input of user external bus MADATA15 is connected to the pin.

[bit23] UED14B: UED14B Output Select bit

Selects output for external bus data 14.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output for user external bus MADATA14. [Initial value] Input of user external bus MADATA 14 is connected to the pin.
	1	Produces output for user external bus MADATA 14. Input of user external bus MADATA 14 is connected to the pin.

[bit22] UED13B: UED13B Output Select bit

Selects output for external bus data 13.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output for user external bus MADATA 13. [Initial value] Input of user external bus MADATA 13 is connected to the pin.
	1	Produces output for user external bus MADATA 13. Input of user external bus MADATA 13 is connected to the pin.

[bit21] UED12B: UED12B Output Select bit

Selects output for external bus data 12.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output for user external bus MADATA12. [Initial value] Input of user external bus MADATA12 is connected to the pin.
	1	Produces output for user external bus MADATA12. Input of user external bus MADATA12 is connected to the pin.

[bit20] UED11B: UED11B Output Select bit

Selects output for external bus data 11.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output for user external bus MADATA11. [Initial value] Input of user external bus MADATA11 is connected to the pin.
	1	Produces output for user external bus MADATA11. Input of user external bus MADATA11 is connected to the pin.

[bit19] UED10B: UED10B Output Select bit

Selects output for external bus data 10.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output for user external bus MADATA10. [Initial value] Input of user external bus MADATA10 is connected to the pin.
	1	Produces output for user external bus MADATA10. Input of user external bus MADATA10 is connected to the pin.

[bit18] UED09B: UED09B Output Select bit

Selects output for external bus data 09.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output for user external bus MADATA09. [Initial value] Input of user external bus MADATA09 is connected to the pin.
	1	Produces output for user external bus MADATA09. Input of user external bus MADATA09 is connected to the pin.

[bit17] UED08B: UED08B Output Select bit

Selects output for external bus data 08.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output for user external bus MADATA08. [Initial value] Input of user external bus MADATA08 is connected to the pin.
	1	Produces output for user external bus MADATA08. Input of user external bus MADATA08 is connected to the pin.

[bit16] UED07B: UED07B Output Select bit

Selects output for external bus data 07.

bit		Description
Reading		Reads out the register value.
Writing		Does not produce output for user external bus MADATA07. [Initial value] Input of user external bus MADATA07 is connected to the pin.
		Produces output for user external bus MADATA07. Input of user external bus MADATA07 is connected to the pin.

[bit15] UED06B: UED06B Output Select bit

Selects output for external bus data 06.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output for user external bus MADATA06. [Initial value] Input of user external bus MADATA06 is connected to the pin.
	1	Produces output for user external bus MADATA06. Input of user external bus MADATA06 is connected to the pin.

[bit14] UED05B: UED05B Output Select bit

Selects output for external bus data 05.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output for user external bus MADATA05. [Initial value] Input of user external bus MADATA05 is connected to the pin.
	1	Produces output for user external bus MADATA05. Input of user external bus MADATA05 is connected to the pin.

[bit13] UED04B: UED04B Output Select bit

Selects output for external bus data 04.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output for user external bus MADATA04. [Initial value] Input of user external bus MADATA04 is connected to the pin.
	1	Produces output for user external bus MADATA04. Input of user external bus MADATA04 is connected to the pin.

[bit12] UED03B: UED03B Output Select bit

Selects output for external bus data 03.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output for user external bus MADATA03. [Initial value] Input of user external bus MADATA03 is connected to the pin.
	1	Produces output for user external bus MADATA03. Input of user external bus MADATA03 is connected to the pin.

[bit11] UED02B: UED02B Output Select bit

Selects output for external bus data 02.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output for user external bus MADATA02. [Initial value] Input of user external bus MADATA02 is connected to the pin.
	1	Produces output for user external bus MADATA02. Input of user external bus MADATA02 is connected to the pin.

[bit10] UED01B: UED01B Output Select bit

Selects output for external bus data 01.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output for user external bus MADATA01. [Initial value] Input of user external bus MADATA01 is connected to the pin.
	1	Produces output for user external bus MADATA01. Input of user external bus MADATA01 is connected to the pin.

[bit9] UED00B: UED00B Output Select bit

Selects output for external bus data 00.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output for user external bus MADATA00. [Initial value] Input of user external bus MADATA00 is connected to the pin.
	1	Produces output for user external bus MADATA00. Input of user external bus MADATA00 is connected to the pin.

[bit8] UEA07E: UEA07E Output Select bit

Selects output for external bus address07.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output for user external bus MAD07. [Initial value]
	1	Produces output for user external bus MAD07.

[bit7] UEA06E: UEA06E Output Select bit

Selects output for external bus address06.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output for user external bus MAD06. [Initial value]
	1	Produces output for user external bus MAD06.

[bit6] UEA05E: UEA05E Output Select bit

Selects output for external bus address05.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output for user external bus MAD05. [Initial value]
	1	Produces output for user external bus MAD05.

[bit5] UEA04E: UEA04E Output Select bit

Selects output for external bus address04.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output for user external bus MAD04. [Initial value]
	1	Produces output for user external bus MAD04.

[bit4] UEA03E: UEA03E Output Select bit

Selects output for external bus address03.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output for user external bus MAD03. [Initial value]
	1	Produces output for user external bus MAD03.

[bit3] UEA02E: UEA02E Output Select bit

Selects output for external bus address02.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output for user external bus MAD02. [Initial value]
	1	Produces output for user external bus MAD02.

[bit2] UEA01E: UEA01E Output Select bit

Selects output for external bus address01.

bit		Description
Reading		Reads out the register value.
Writing		Does not produce output for user external bus MAD01. [Initial value]
		Produces output for user external bus MAD01.

[bit1] UECS0E: UECS0E Output Select bit

Selects output for external bus CS0.

bit		Description
Reading		Reads out the register value.
Writing		Does not produce output for user external bus MCSX0. [Initial value]
		Produces output for user external bus MCSX0.

[bit0] UEALEE: UEALEE Output Select bit

Selects output for external bus ALE signal.

bit		Description
Reading		Reads out the register value.
Writing		Does not produce output for user external bus MALE. [Initial value]
		Produces output for user external bus MALE.

<Notes>

- I/O selection of the external bus data[15:7] can be controlled collectively with EPFR10.bit1.
  I/O selection of the external bus data[15:7] can be controlled by each bit also with EPFR11.bit[24:17]
  EPFR10.bit1 setting has the higher priority than EPFR11.bit[24:17] setting.
  To control I/O selection by setting EPFR11.bit[24:17], it is necessary to set EPFR10.bit1=0.
- I/O selection of the external bus address[7:1], MCSX0, and external bus data[7:0] can be controlled collectively with EPFR10.bit0.
  I/O selection of the external bus address[7:1], MCSX0, and external bus data[7:0] can be controlled by each bit also with EPFR11.bit[16:1].
  EPFR10.bit0 setting has the higher priority than EPFR11.bit[16:1] setting.
  To control I/O selection by setting EPFR11.bit[16:1], it is necessary to set EPFR10.bit0=0.
- This register is not initialized by deep standby transition reset.

4.20. Extended Pin Function Setting Register 12 (EPFR12)

The EPFR12 register assigns functions to pins of ch.8, ch.9, ch.10, and ch.11 of the base timer.

■ Register Configuration

bit	31	30	29	28	27	26	25	24
Field	Reserved		TIOB11S		TIOA11E		TIOA11S	
Attribute	-		R/W		R/W		R/W	
Initial value	-		00		00		00	

bit	23	22	21	20	19	18	17	16
Field	Reserved		TIOB10S		TIOA10E		Reserved	
Attribute	-		R/W		R/W		-	
Initial value	-		00		00		-	

bit	15	14	13	12	11	10	9	8
Field	Reserved		TIOB9S		TIOA9E		TIOA9S	
Attribute	-		R/W		R/W		R/W	
Initial value	-		00		00		00	

bit	7	6	5	4	3	2	1	0
Field	Reserved		TIOB8S		TIOA8E		Reserved	
Attribute	-		R/W		R/W		-	
Initial value	-		00		00		-	

■ Register Function

[bit31:30] Reserved: Reserved bits

"0b00" is read out from these bits.
When writing these bits, set them to "0b00".

[bit29:28] TIOB11S: TIOB11 Input Select bits

Selects input for TIOB11.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses TIOB11_0 at the input pin of BT ch.11 TIOB. [Initial value]
	01	Same as Writing 00.
	10	Uses TIOB11_1 at the input pin of BT ch.11 TIOB.
	11	Uses TIOB11_2 at the input pin of BT ch.11 TIOB.

[bit27:26] TIOA11E: TIOA11 Output Select bits

Selects output for TIOA11.

bit		Description
Reading		Reads out the register value.
Writing	00	Does not produce the output of the BT ch.11 TIOA. [Initial value]
	01	Uses TIOA11_0 at the output pin of BT ch.11 TIOA.
	10	Uses TIOA11_1 at the output pin of BT ch.11 TIOA.
	11	Uses TIOA11_2 at the output pin of BT ch.11 TIOA.

[bit25:24] TIOA11S: TIOA11 Input Select bits

Selects input for TIOA11.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses TIOA11_0 at the input pin of BT ch.11 TIOA. [Initial value]
	01	Same as Writing 00.
	10	Uses TIOA11_1 at the input pin of BT ch.11 TIOA.
	11	Uses TIOA11_2 at the input pin of BT ch.11 TIOA.

[bit23:22] Reserved: Reserved bits

"0b00" is read out from these bits.
When writing these bits, set them to "0b00".

[bit21:20] TIOB10S: TIOB10 Input Select bits

Selects input for TIOB10.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses TIOB10_0 at the input pin of BT ch.10 TIOB. [Initial value]
	01	Same as Writing 00.
	10	Uses TIOB10_1 at the input pin of BT ch.10 TIOB.
	11	Uses TIOB10_2 at the input pin of BT ch.10 TIOB.

[bit19:18] TIOA10E: TIOA10 Output Select bits

Selects output for TIOA10.

bit		Description
Reading		Reads out the register value.
Writing	00	Does not produce the output of the BT ch.10 TIOA. [Initial value]
	01	Uses TIOA10_0 at the output pin of BT ch.10 TIOA.
	10	Uses TIOA10_1 at the output pin of BT ch.10 TIOA.
	11	Uses TIOA10_2 at the output pin of BT ch.10 TIOA.

[bit17:14] Reserved: Reserved bits

"0b0000" is read out from these bits.
When writing these bits, set them to "0b0000".

[bit13:12] TIOB9S: TIOB9 Input Select bits

Selects input for TIOB9.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses TIOB9_0 at the input pin of BT ch.9 TIOB. [Initial value]
	01	Same as Writing 00.
	10	Uses TIOB9_1 at the input pin of BT ch.9 TIOB.
	11	Uses TIOB9_2 at the input pin of BT ch.9 TIOB.

[bit11:10] TIOA9E: TIOA9 Output Select bits

Selects output for TIOA9.

bit		Description
Reading		Reads out the register value.
Writing	00	Does not produce the output of the BT ch.9 TIOA. [Initial value]
	01	Uses TIOA9_0 at the output pin of BT ch.9 TIOA.
	10	Uses TIOA9_1 at the output pin of BT ch.9 TIOA.
	11	Uses TIOA9_2 at the output pin of BT ch.9 TIOA.

[bit9:8] TIOA9S: TIOA9 Input Select bits

Selects input for TIOA9.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses TIOA9_0 at the input pin of BT ch.9 TIOA. [Initial value]
	01	Same as Writing 00.
	10	Uses TIOA9_1 at the input pin of BT ch.9 TIOA.
	11	Uses TIOA9_2 at the input pin of BT ch.9 TIOA.

[bit7:6] Reserved: Reserved bits

"0b00" is read out from these bits.
When writing these bits, set them to "0b00".

[bit5:4] TIOB8S: TIOB8 Input Select bits

Selects input for TIOB8.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses TIOB8_0 at the input pin of BT ch.8 TIOB. [Initial value]
	01	Same as Writing 00.
	10	Uses TIOB8_1 at the input pin of BT ch.8 TIOB.
	11	Uses TIOB8_2 at the input pin of BT ch.8 TIOB.

[bit3:2] TIOA8E: TIOA8 Output Select bits

Selects output for TIOA8.

bit		Description
Reading		Reads out the register value.
Writing	00	Does not produce the output of the BT ch.8 TIOA. [Initial value]
	01	Uses TIOA8_0 at the output pin of BT ch.8 TIOA.
	10	Uses TIOA8_1 at the output pin of BT ch.8 TIOA.
	11	Uses TIOA8_2 at the output pin of BT ch.8 TIOA.

[bit1:0] Reserved: Reserved bits

"0b00" is read out from these bits.
When writing these bits, set them to "0b00".

<Notes>

- TIOA
  Even channels are for output only.
  Odd channels are for both input and output.
- TIOB
  Input only.
- TIOA9, TIOA11, TIOA13, TIOA15 (odd number of "A") are not bidirectional pins so that choose either input pin or output pin for them.
  When output is chosen for odd TIOA channel, input setting will be ignored.

  Example1: Use TIOA11 as an output pin:
  When TIOA11 is output to TIOA11_0, select EPFR12:TIOA11E = 01.
  When TIOA11 is output to TIOA11_1, select EPFR12:TIOA11E = 10.
  When TIOA11 is output to TIOA11_2, select EPFR12:TIOA11E = 11.
  Settings for EPFR12:TIOA11S will be ignored.
  Select ADE=0, PFR=1 for selected pins (DDR will be ignored).
  All the output of other peripheral function pins which are also used by selected pins must be OFF.

  Example2: When TIOA11 is used as an input pin:
  Select EPFR12:TIOA11E = 00.
  When TIOA11 is input from TIOA11_0, select EPFR12:TIOA11S = 00 or 01.
  When TIOA11 is input from TIOA11_1, select EPFR12:TIOA11S = 10.
  When TIOA11 is input from TIOA11_2, select EPFR12:TIOA11S = 11.
  Select ADE=0, PFR=1 for selected pins (DDR will be ignored).
  All the output of other peripheral function pins which are also used by selected pins must be OFF.

  * When a pin is set to input, the output of the pin which is also used by selected pins (GPIO, other peripheral function pins) can be input to the base timer as Feedback with a setting not described above.

- This register is not initialized by deep standby transition reset.

4.21. Extended Pin Function Setting Register 13 (EPFR13)

The EPFR13 register assigns functions to pins of ch.12, ch.13, ch.14, and ch.15 of the base timer.

■ Register Configuration

bit	31	30	29	28	27	26	25	24
Field	Reserved		TIOB15S		TIOA15E		TIOA15S	
Attribute	-		R/W		R/W		R/W	
Initial value	-		00		00		00	

bit	23	22	21	20	19	18	17	16
Field	Reserved		TIOB14S		TIOA14E		Reserved	
Attribute	-		R/W		R/W		-	
Initial value	-		00		00		-	

bit	15	14	13	12	11	10	9	8
Field	Reserved		TIOB13S		TIOA13E		TIOA13S	
Attribute	-		R/W		R/W		R/W	
Initial value	-		00		00		00	

bit	7	6	5	4	3	2	1	0
Field	Reserved		TIOB12S		TIOA12E		Reserved	
Attribute	-		R/W		R/W		-	
Initial value	-		00		00		-	

■ Register Function

[bit31:30] Reserved: Reserved bits

"0b00" is read out from these bits.
When writing these bits, set them to "0b00".

[bit29:28] TIOB15S: TIOB15 Input Select bits

Selects input for TIOB15.

bit29:28		Description
Reading		Reads out the register value.
Writing	00	Uses TIOB15_0 at the input pin of BT ch.15 TIOB. [Initial value]
	01	Same as Writing 00.
	10	Uses TIOB15_1 at the input pin of BT ch.15 TIOB.
	11	Uses TIOB15_2 at the input pin of BT ch.15 TIOB.

[bit27:26] TIOA15E: TIOA15 Output Select bits

Selects output for TIOA15.

bit27:26		Description
Reading		Reads out the register value.
Writing	00	Does not produce the output of the BT ch.15 TIOA. [Initial value]
	01	Uses TIOA15_0 at the output pin of BT ch.15 TIOA.
	10	Uses TIOA15_1 at the output pin of BT ch.15 TIOA.
	11	Uses TIOA15_2 at the output pin of BT ch.15 TIOA.

[bit25:24] TIOA15S: TIOA15 Input Select bits

Selects input for TIOA15.

bit25:24		Description
Reading		Reads out the register value.
Writing	00	Uses TIOA15_0 at the input pin of BT ch.15 TIOA. [Initial value]
	01	Same as Writing 00.
	10	Uses TIOA15_1 at the input pin of BT ch.15 TIOA.
	11	Uses TIOA15_2 at the input pin of BT ch.15 TIOA.

[bit23:22] Reserved: Reserved bits

"0b00" is read out from these bits.
When writing these bits, set them to "0b00".

[bit21:20] TIOB14S: TIOB14 Input Select bits

Selects input for TIOB14.

bit21:20		Description
Reading		Reads out the register value.
Writing	00	Uses TIOB14_0 at the input pin of BT ch.14 TIOB. [Initial value]
	01	Same as Writing 00.
	10	Uses TIOB14_1 at the input pin of BT ch.14 TIOB.
	11	Uses TIOB14_2 at the input pin of BT ch.14 TIOB.

[bit19:18] TIOA14E: TIOA14 Output Select bits

Selects output for TIOA14.

bit19:18		Description
Reading		Reads out the register value.
Writing	00	Does not produce the output of the BT ch.14 TIOA. [Initial value]
	01	Uses TIOA14_0 at the output pin of BT ch.14 TIOA.
	10	Uses TIOA14_1 at the output pin of BT ch.14 TIOA.
	11	Uses TIOA14_2 at the output pin of BT ch.14 TIOA.

[bit17:14] Reserved: Reserved bits

"0b0000" is read out from these bits.
When writing these bits, set them to "0b0000".

[bit13:12] TIOB13S: TIOB13 Input Select bits

Selects input for TIOB13.

bit13:12		Description
Reading		Reads out the register value.
Writing	00	Uses TIOB13_0 at the input pin of BT ch.13 TIOB. [Initial value]
	01	Same as Writing 00.
	10	Uses TIOB13_1 at the input pin of BT ch.13 TIOB.
	11	Uses TIOB13_2 at the input pin of BT ch.13 TIOB.

[bit11:10] TIOA13E: TIOA13 Output Select bits

Selects output for TIOA13.

bit		Description
Reading		Reads out the register value.
Writing	00	Does not produce the output of the BT ch.13 TIOA. [Initial value]
	01	Uses TIOA13_0 at the output pin of BT ch.13 TIOA.
	10	Uses TIOA13_1 at the output pin of BT ch.13 TIOA.
	11	Uses TIOA13_2 at the output pin of BT ch.13 TIOA.

[bit9:8] TIOA13S: TIOA13 Input Select bits

Selects input for TIOA13.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses TIOA13_0 at the input pin of BT ch.13 TIOA. [Initial value]
	01	Same as Writing 00.
	10	Uses TIOA13_1 at the input pin of BT ch.13 TIOA.
	11	Uses TIOA13_2 at the input pin of BT ch.13 TIOA.

[bit7:6] Reserved: Reserved bits

"0b00" is read out from these bits.
When writing these bits, set them to "0b00".

[bit5:4] TIOB12S: TIOB12 Input Select bits

Selects input for TIOB12.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses TIOB12_0 at the input pin of BT ch.12 TIOB. [Initial value]
	01	Same as Writing 00.
	10	Uses TIOB12_1 at the input pin of BT ch.12 TIOB.
	11	Uses TIOB12_2 at the input pin of BT ch.12 TIOB.

[bit3:2] TIOA12E: TIOA12 Output Select bits

Selects output for TIOA12.

bit		Description
Reading		Reads out the register value.
Writing	00	Does not produce the output of the BT ch.12 TIOA. [Initial value]
	01	Uses TIOA12_0 at the output pin of BT ch.12 TIOA.
	10	Uses TIOA12_1 at the output pin of BT ch.12 TIOA.
	11	Uses TIOA12_2 at the output pin of BT ch.12 TIOA.

[bit1:0] Reserved: Reserved bits

"0b00" is read out from these bits.
When writing these bits, set them to "0b00".

<Notes>

- TIOA
  Even channels are for output only.
  Odd channels are for both input and output.
- TIOB
  Input only.
- TIOA9, TIOA11, TIOA13, TIOA15 (odd number of "A") are not bidirectional pins so that choose either input pin or output pin for them.
  When output is chosen for odd TIOA channel, input setting will be ignored.

  Example1: Use TIOA11 as an output pin:
  When TIOA11 is output to TIOA11_0, select EPFR12:TIOA11E = 01.
  When TIOA11 is output to TIOA11_1, select EPFR12:TIOA11E = 10.
  When TIOA11 is output to TIOA11_2, select EPFR12:TIOA11E = 11.
  Settings for EPFR12:TIOA11S will be ignored.
  Select ADE=0, PFR=1 for selected pins (DDR will be ignored).
  All the output of other peripheral function pins which are also used by selected pins must be OFF.

  Example2: When TIOA11 is used as an input pin:
  Select EPFR12:TIOA11E = 00.
  When TIOA11 is input from TIOA11_0, select EPFR12:TIOA11S = 00 or 01.
  When TIOA11 is input from TIOA11_1, select EPFR12:TIOA11S = 10.
  When TIOA11 is input from TIOA11_2, select EPFR12:TIOA11S = 11.
  Select ADE=0, PFR=1 for selected pins (DDR will be ignored).
  All the output of other peripheral function pins which are also used by selected pins must be OFF.

  * When a pin is set to input, the output of the pin which is also used by selected pins (GPIO, other peripheral function pins) can be input to the base timer as Feedback with a setting not described above.

- This register is not initialized by deep standby transition reset.

4.22. Extended Pin Function Setting Register 14 (EPFR14)

EPFR14 register sets the function assignment to QPRC/Ethernet-MAC pins.

■ Register Configuration

bit	31	30	29	28	27	26	25	24
Field	Reserved		E_SPLC		E_PSE	E_CKE	E_MD1B	E_MD0B
Attribute	-		R/W		R/W	R/W	R/W	R/W
Initial value	-		00		0	0	0	0

bit	23	22	21	20	19	18	17	16
Field	E_MC1B	E_MC0E	E_TE1E	E_TE0E	E_TD1E	E_TD0E	Reserved	
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	-	
Initial value	0	0	0	0	0	0	-	

bit	15	14	13	12	11	10	9	8
Field	Reserved							
Attribute	-							
Initial value	-							

bit	7	6	5	4	3	2	1	0
Field	Reserved		QZIN2S		QBIN2S		QAIN2S	
Attribute	-		R/W		R/W		R/W	
Initial value	-		00		00		00	

■ Register Function

[bit31:30] Reserved: Reserved bits

"0b00" is read out from these bits.
When writing these bits, set them to "0b00".

[bit29:28] E_SPLC: Input cutoff Select bit in Standby of input Pin for Ethernet-MAC

It selects input interrupt in standby of the Ethernet-MAC input pin.

bit		Description
Reading		Reads out the register value.
Writing	00	All input pins are cut off in standby. [Initial value]
	01	Input pins used by MII ch.0 are not cut off in standby.
	10	Input pins used by RMII ch.0 are not cut off in standby.
	11	Input pins used by RMII ch.1 are not cut off in standby.

[bit27] E_PSE: PPS0_PPS1 Output Select bit for Ethernet-MAC

Selects output for PPS0_PPS1.

bit		Description
Reading		Reads out the register value.
Writing	0	E_PPS0_PPS1 is not output. [Initial value]
	1	E_PPS0_PPS1 is output. [Initial value]

[bit26] E_CKE: E_COUT Output Select bit

Selects output for E_COUT.

bit		Description
Reading		Reads out the register value.
Writing	0	E_COUT is not output. [Initial value]
	1	E_COUT is output.

[bit25] E_MD1B :E_MDO1 I/O Select bit

Selects I/O for E_MDO1.

bit		Description
Reading		Reads out the register value.
Writing	0	E_MDO1 is not output. [Initial value] Input of E_MDI1 is connected to the pin.
	1	E_MDO1 is output. Input of E_MDI1 is connected to the pin.

[bit24] E_MD0B :E_MDO0 I/O Select bit

Selects I/O for E_MDC0.

bit		Description
Reading		Reads out the register value.
Writing	0	E_MDC0 is not output. [Initial value] Input of E_MDI0 is connected to the pin.
	1	E_MDC0 is output. Input of E_MDI0 is connected to the pin.

[bit23] E_MC1B :E_MDC1 I/O Select bit

Selects I/O for E_MDC1.

bit		Description
Reading		Reads out the register value.
Writing	0	E_MDC1 is not output. [Initial value] Input of E_TCK0 is connected to the pin.
	1	E_MDC1 is output. Input of E_TCK0 is connected to the pin.

[bit22] E_MC0E :E_MDC0 Output Select bit

Selects output for E_MDC0.

bit		Description
Reading		Reads out the register value.
Writing	0	E_MDC0 is not output. [Initial value]
	1	E_MDC0 is output.

[bit21] E_TE1E: E_TXER0_TXEN1 Output Select bit

Selects output for E_TXER0_TXEN1.

bit		Description
Reading		Reads out the register value.
Writing	0	E_TXER0_TXEN1 is not output. [Initial value]
	1	E_TXER0_TXEN1 is output.

[bit20] E_TE0E: E_TXEN0 Output Select bit

Selects output for E_TXEN0.

bit		Description
Reading		Reads out the register value.
Writing	0	E_TXEN0 is not output. [Initial value]
	1	E_TXEN0 is output.

[bit19] E_TD1E: E_TX02_TX10, E_TX03_TX11 Output Select bit

Selects output for E_TX02_TX10, E_TX03_TX11.

bit		Description
Reading		Reads out the register value.
Writing	0	E_TX02_TX10 is not output. [Initial value] E_TX03_TX11 is not output.
	1	E_TX02_TX10 is output. [Initial value] E_TX03_TX11 is output.

[bit18] E_TD0E: E_TX00, E_TX01 Output Select bit

Selects output for E_TX00, E_TX01.

bit		Description
Reading		Reads out the register value.
Writing	0	E_TX00, E_TX01 is not output. [Initial value]
	1	E_TX00, E_TX00 is output.

[bit17:6] Reserved: Reserved bits

"0x000" is read out from these bits.
When writing these bits, set them to "0x000".

[bit5:4] QZIN2S: QDU-ch.2 ZIN Input Pin bits

Selects input for QDU-ch.2 as ZIN.

bit		Description
Reading		Reads out the register value.
Writing	00	ZIN2_0 is used as ZIN, the input pin of QDU ch.2. [Initial value]
	01	ZIN2_0 is used as ZIN, the input pin of QDU ch.2.
	10	ZIN2_1 is used as ZIN, the input pin of QDU ch.2.
	11	ZIN2_2 is used as ZIN, the input pin of QDU ch.2.

[bit3:2] QBIN2S: QDU-ch.2 BIN Input Pin bits

Selects input for QDU-ch.2 as BIN.

bit		Description
Reading		Reads out the register value.
Writing	00	BIN2_0 is used as BIN, the input pin of QDU ch.2. [Initial value]
	01	BIN2_0 is used as BIN, the input pin of QDU ch.2.
	10	BIN2_1 is used as BIN, the input pin of QDU ch.2.
	11	BIN2_2 is used as BIN, the input pin of QDU ch.2.

[bit1:0] QAIN2S: QDU-ch.2 AIN Input Pin bits

Selects input for QDU-ch.2 as AIN.

bit		Description
Reading		Reads out the register value.
Writing	00	AIN2_0 is used as AIN, the input pin of QDU ch.2. [Initial value]
	01	AIN2_0 is used as AIN, the input pin of QDU ch.2.
	10	AIN2_1 is used as AIN, the input pin of QDU ch.2.
	11	AIN2_2 is used as AIN, the input pin of QDU ch.2.

<Note>

- This register is not initialized by deep standby transition reset.

.

4.23. Extended Pin Function Setting Register 15 (EPFR15)

EPFR15 register sets the function assignment to external interrupt pins.

■ Register Configuration

bit	31	30	29	28	27	26	25	24
Field	EINT31S		EINT30S		EINT29S		EINT28S	
Attribute	R/W		R/W		R/W		R/W	
Initial value	00		00		00		00	

bit	23	22	21	20	19	18	17	16
Field	EINT27S		EINT26S		EINT25S		EINT24S	
Attribute	R/W		R/W		R/W		R/W	
Initial value	00		00		00		00	

bit	15	14	13	12	11	10	9	8
Field	EINT23S		EINT22S		EINT21S		EINT20S	
Attribute	R/W		R/W		R/W		R/W	
Initial value	00		00		00		00	

bit	7	6	5	4	3	2	1	0
Field	EINT19S		EINT18S		EINT17S		EINT16S	
Attribute	R/W		R/W		R/W		R/W	
Initial value	00		00		00		00	

■ Register Function

[bit31:30] EINT31S: External Interrupt Input Select bits

Selects input for EINT31.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses INT31_0 at the input pin of EINT ch.31. [Initial value]
	01	Same as Writing 00.
	10	Uses INT31_1 at the input pin of EINT ch.31.
	11	Uses INT31_2 at the input pin of EINT ch.31.

[bit29:28] EINT30S: External Interrupt Input Select bits

Selects input for EINT30.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses INT30_0 at the input pin of EINT ch.30. [Initial value]
	01	Same as Writing 00.
	10	Uses INT30_1 at the input pin of EINT ch.30.
	11	Uses INT30_2 at the input pin of EINT ch.30.

[bit27:26] EINT29S: External Interrupt Input Select bits

Selects input for EINT29.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses INT29_0 at the input pin of EINT ch.29. [Initial value]
	01	Same as Writing 00.
	10	Uses INT29_1 at the input pin of EINT ch.29.
	11	Uses INT29_2 at the input pin of EINT ch.29.

[bit25:24] EINT28S: External Interrupt Input Select bits

Selects input for EINT28.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses INT28_0 at the input pin of EINT ch.28. [Initial value]
	01	Same as Writing 00.
	10	Uses INT28_1 at the input pin of EINT ch.28.
	11	Uses INT28_2 at the input pin of EINT ch.28.

[bit23:22] EINT27S: External Interrupt Input Select bits

Selects input for EINT27.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses INT27_0 at the input pin of EINT ch.27. [Initial value]
	01	Same as Writing 00.
	10	Uses INT27_1 at the input pin of EINT ch.27.
	11	Uses INT27_2 at the input pin of EINT ch.27.

[bit21:20] EINT26S: External Interrupt Input Select bits

Selects input for EINT26.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses INT26_0 at the input pin of EINT ch.26. [Initial value]
	01	Same as Writing 00.
	10	Uses INT26_1 at the input pin of EINT ch.26.
	11	Uses INT26_2 at the input pin of EINT ch.26.

[bit19:18] EINT25S: External Interrupt Input Select bits

Selects input for EINT25.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses INT25_0 at the input pin of EINT ch.25. [Initial value]
	01	Same as Writing 00.
	10	Uses INT25_1 at the input pin of EINT ch.25.
	11	Uses INT25_2 at the input pin of EINT ch.25.

[bit17:16] EINT24S: External Interrupt Input Select bits

Selects input for EINT24.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses INT24_0 at the input pin of EINT ch.24. [Initial value]
	01	Same as Writing 00.
	10	Uses INT24_1 at the input pin of EINT ch.24.
	11	Uses INT24_2 at the input pin of EINT ch.24.

[bit15:14] EINT23S: External Interrupt Input Select bits

Selects input for EINT23.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses INT23_0 at the input pin of EINT ch.23. [Initial value]
	01	Same as Writing 00.
	10	Uses INT23_1 at the input pin of EINT ch.23.
	11	Uses INT23_2 at the input pin of EINT ch.23.

[bit13:12] EINT22S: External Interrupt Input Select bits

Selects input for EINT22.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses INT22_0 at the input pin of EINT ch.22. [Initial value]
	01	Same as Writing 00.
	10	Uses INT22_1 at the input pin of EINT ch.22.
	11	Uses INT22_2 at the input pin of EINT ch.22.

[bit11:10] EINT21S: External Interrupt Input Select bits

Selects input for EINT21.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses INT21_0 at the input pin of EINT ch.21. [Initial value]
	01	Same as Writing 00.
	10	Uses INT21_1 at the input pin of EINT ch.21.
	11	Uses INT21_2 at the input pin of EINT ch.21.

[bit9:8] EINT20S: External Interrupt Input Select bits

Selects input for EINT20.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses INT20_0 at the input pin of EINT ch.20. [Initial value]
	01	Same as Writing 00.
	10	Uses INT20_1 at the input pin of EINT ch.20.
	11	Uses INT20_2 at the input pin of EINT ch.20.

[bit7:6] EINT19S: External Interrupt Input Select bits

Selects input for EINT19.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses INT19_0 at the input pin of EINT ch.19. [Initial value]
	01	Same as Writing 00.
	10	Uses INT19_1 at the input pin of EINT ch.19.
	11	Uses INT19_2 at the input pin of EINT ch.19.

[bit5:4] EINT18S: External Interrupt Input Select bits

Selects input for EINT18.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses INT18_0 at the input pin of EINT ch.18. [Initial value]
	01	Same as Writing 00.
	10	Uses INT18_1 at the input pin of EINT ch.18.
	11	Uses INT18_2 at the input pin of EINT ch.18.

[bit3:2] EINT17S: External Interrupt Input Select bits

Selects input for EINT17.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses INT17_0 at the input pin of EINT ch.17. [Initial value]
	01	Same as Writing 00.
	10	Uses INT17_1 at the input pin of EINT ch.17.
	11	Uses INT17_2 at the input pin of EINT ch.17.

[bit1:0] EINT16S: External Interrupt Input Select bits

Selects input for EINT16.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses INT16_0 at the input pin of EINT ch.16. [Initial value]
	01	Same as Writing 00.
	10	Uses INT16_1 at the input pin of EINT ch.16.
	11	Uses INT16_2 at the input pin of EINT ch.16.

<Notes>

- This register is not initialized by deep standby transition reset.

4.24. Extended Pin Function Setting Register 16 (EPFR16)

The EPFR16 register assigns functions of multi-function serial channel 6, channel 7, channel 8, channel 9, channel 10, and channel 11.

■ Register Configuration

bit	31	30	29	28	27	26	25	24
Field	Reserved		SFMPBC	SFMPAC	SCK11B		SOT11B	
Attribute	-		R/W	R/W	R/W		R/W	
Initial value	-		0	0	00		00	

bit	23	22	21	20	19	18	17	16
Field	SIN11S		SCK10B		SOT10B		SIN10S	
Attribute	R/W		R/W		R/W		R/W	
Initial value	00		00		00		00	

bit	15	14	13	12	11	10	9	8
Field	SCK9B		SOT9B		SIN9S		SCK8B	
Attribute	R/W		R/W		R/W		R/W	
Initial value	00		00		00		00	

bit	7	6	5	4	3	2	1	0
Field	SOT8B		SIN8S		SCS7B		SCS6B	
Attribute	R/W		R/W		R/W		R/W	
Initial value	00		00		00		00	

■ Register Function

[bit31:30] Reserved: Reserved bits

"0b00" is read from these bits.
When writing these bits, set them to "0b00".

[bit29] SFMPBC : MFS ch.B I2C FastMode+ Select bit

Selects MFS ch.B I2C FastMode+.

bit		Description
Reading		Reads out the register value.
Writing	0	MFS ch.B-I2C Input/Output pin does not supports I2C FastMode+.[Initial value]
	1	MFS ch.B-I2C Input/Output pin supports I2C FastMode+.

[bit28] SFMPAC: MFS ch.A I2C FastMode+ Select bit

Selects MFS ch.A I2C FastMode+.

bit		説明
Reading		Reads out the register value.
Writing	0	MFS ch.A-I2C Input/Output pin does not supports I2C FastMode+.[Initial value]
	1	MFS ch.A-I2C Input/Output pin supports I2C FastMode+.

[bit27:26] SCK11B: SCK11 Input/Output Select bits

Selects input/output for SCK11.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses SCK11_0 at the input pin of MFS ch.11 SCK. Does not produce output. [Initial value]
	01	Uses SCK11_0 at the input pin of MFS ch.11 SCK. Uses SCK11_0 at the output pin.
	10	Uses SCK11_1 at the input pin of MFS ch.11 SCK. Uses SCK11_1 at the output pin.
	11	Uses SCK11_2 at the input pin of MFS ch.11 SCK. Uses SCK11_2 at the output pin.

[bit25:24] SOT11B: SOT11 Input/Output Select bits

Selects input/output for SOT11.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses SOT11_0 at the input pin of MFS ch.11 SOT. Does not produce output. [Initial value]
	01	Uses SOT11_0 at the input pin of MFS ch.11 SOT. Uses SOT11_0 at the output pin.
	10	Uses SOT11_1 at the input pin of MFS ch.11 SOT. Uses SOT11_1 at the output pin.
	11	Uses SOT11_2 at the input pin of MFS ch.11 SOT. Uses SOT11_2 at the output pin.

[bit23:22] SIN11S: SIN11 Input Select bits

Selects input for SIN11.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses SIN11_0 at the input pin of MFS ch.11 SIN. [Initial value]
	01	Same as Writing 00.
	10	Uses SIN11_1 at the input pin of MFS ch.11 SIN.
	11	Uses SIN11_2 at the input pin of MFS ch.11 SIN.

[bit21:20] SCK10B: SCK10 Input/Output Select bits

Selects input/output for SCK10.

bit21:20		Description
Reading		Reads out the register value.
Writing	00	Uses SCK10_0 at the input pin of MFS ch.10 SCK. Does not produce output. [Initial value]
	01	Uses SCK10_0 at the input pin of MFS ch.10 SCK. Uses SCK10_0 at the output pin.
	10	Uses SCK10_1 at the input pin of MFS ch.10 SCK. Uses SCK10_1 at the output pin.
	11	Uses SCK10_2 at the input pin of MFS ch.10 SCK. Uses SCK10_2 at the output pin.

[bit19:18] SOT10B: SOT10 Input/Output Select bits

Selects input/output for SOT10.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses SOT10_0 at the input pin of MFS ch.10 SOT. Does not produce output. [Initial value]
	01	Uses SOT10_0 at the input pin of MFS ch.10 SOT. Uses SOT10_0 at the output pin.
	10	Uses SOT10_1 at the input pin of MFS ch.10 SOT. Uses SOT10_1 at the output pin.
	11	Uses SOT10_2 at the input pin of MFS ch.10 SOT. Uses SOT10_2 at the output pin.

[bit17:16] SIN10S: SIN10 Input Select bits

Selects input for SIN10.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses SIN10_0 at the input pin of MFS ch.10 SIN. [Initial value]
	01	Same as Writing 00.
	10	Uses SIN10_1 at the input pin of MFS ch.10 SIN.
	11	Uses SIN10_2 at the input pin of MFS ch.10 SIN.

[bit15:14] SCK9B: SCK9 Input/Output Select bits

Selects input/output for SCK9.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses SCK9_0 at the input pin of MFS ch.9 SCK. Does not produce output. [Initial value]
	01	Uses SCK9_0 at the input pin of MFS ch.9 SCK. Uses SCK9_0 at the output pin.
	10	Uses SCK9_1 at the input pin of MFS ch.9 SCK. Uses SCK9_1 at the output pin.
	11	Uses SCK9_2 at the input pin of MFS ch.9 SCK. Uses SCK9_2 at the output pin.

[bit13:12] SOT9B: SOT9 Input/Output Select bits

Selects input/output for SOT9.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses SOT9_0 at the input pin of MFS ch.9 SOT. Does not produce output. [Initial value]
	01	Uses SOT9_0 at the input pin of MFS ch.9 SOT. Uses SOT9_0 at the output pin.
	10	Uses SOT9_1 at the input pin of MFS ch.9 SOT. Uses SOT9_1 at the output pin.
	11	Uses SOT9_2 at the input pin of MFS ch.9 SOT. Uses SOT9_2 at the output pin.

[bit11:10] SIN9S: SIN9 Input Select bits

Selects input for SIN9.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses SIN9_0 at the input pin of MFS ch.9 SIN. [Initial value]
	01	Same as Writing 00.
	10	Uses SIN9_1 at the input pin of MFS ch.9 SIN.
	11	Uses SIN9_2 at the input pin of MFS ch.9 SIN.

[bit9:8] SCK8B: SCK8 Input/Output Select bits

Selects input/output for SCK8.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses SCK8_0 at the input pin of MFS ch.8 SCK. Does not produce output. [Initial value]
	01	Uses SCK8_0 at the input pin of MFS ch.8 SCK. Uses SCK8_0 at the output pin.
	10	Uses SCK8_1 at the input pin of MFS ch.8 SCK. Uses SCK8_1 at the output pin.
	11	Uses SCK8_2 at the input pin of MFS ch.8 SCK. Uses SCK8_2 at the output pin.

[bit7:6] SOT8B: SOT8 Input/Output Select bits

Selects input/output for SOT8.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses SOT8_0 at the input pin of MFS ch.8 SOT. Does not produce output. [Initial value]
	01	Uses SOT8_0 at the input pin of MFS ch.8 SOT. Uses SOT8_0 at the output pin.
	10	Uses SOT8_1 at the input pin of MFS ch.8 SOT. Uses SOT8_1 at the output pin.
	11	Uses SOT8_2 at the input pin of MFS ch.8 SOT. Uses SOT8_2 at the output pin.

[bit5:4] SIN8S: SIN8 Input Select bits

Selects input for SIN4.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses SIN8_0 at the input pin of MFS ch.8 SIN. [Initial value]
	01	Same as Writing 00.
	10	Uses SIN8_1 at the input pin of MFS ch.8 SIN.
	11	Uses SIN8_2 at the input pin of MFS ch.8 SIN.

[bit3:2] SCS7B : SCS7 Input/Output Select bits

Selects input/output for SCS7.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses SCS7_0 at the input pin of MFS ch.7 SCS. Does not produce output. [Initial value]
	01	Uses SCS7_0 at the input pin of MFS ch.7 SCK. Uses SCS7_0 at the output pin.
	10	Uses SCS7_1 at the input pin of MFS ch.7 SCK. Uses SCS7_1 at the output pin.
	11	Uses SCS7_2 at the input pin of MFS ch.7 SCK. Uses SCS7_2 at the output pin.

[bit1:0] SCS6B : SCS6 Select bits

Selects input/output for SCS6.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses SCS6_0 at the input pin of MFS ch.6 SCS. Does not produce output. [Initial value]
	01	Uses SCS6_0 at the input pin of MFS ch.6 SCS Uses SCS6_0 at the output pin.
	10	Uses SCS6_1 at the input pin of MFS ch.6 SCS. Uses SCS6_1 at the output pin.
	11	Uses SCS6_2 at the input pin of MFS ch.6 SCS. Uses SCS6_2 at the output pin.

<Notes>

- For the assignment of the corresponding MFS ch.A and MFS ch.B to MFS channel supporting I2C FastMode+, see “Data Sheet” of the product used.
- This register is not initialized by deep standby transition reset.

4.25. Extended Pin Function Setting Register 17 (EPFR17)

The EPFR08 register assigns functions of multi-function serial channel 12, channel 13, channel 14, and channel 15.

■ Register Configuration

bit	31	30	29	28	27	26	25	24
Field	Reserved				SCK15B		SOT15B	
Attribute	-				R/W		R/W	
Initial value	-				00		00	

bit	23	22	21	20	19	18	17	16
Field	SIN15S		SCK14B		SOT14B		SIN14S	
Attribute	R/W		R/W		R/W		R/W	
Initial value	00		00		00		00	

bit	15	14	13	12	11	10	9	8
Field	SCK13B		SOT13B		SIN13S		SCK12B	
Attribute	R/W		R/W		R/W		R/W	
Initial value	00		00		00		00	

Bit	7	6	5	4	3	2	1	0
Field	SOT12B		SIN12S		Reserved			
Attribute	R/W		R/W		-			
Initial value	00		00		-			

■ Register Function

[bit31:28] Reserved: Reserved bits

"0b0000" is read from these bits.
When writing these bits, set them to "0b0000".

[bit27:26] SCK15B: SCK15 Input/Output Select bits

Selects input/output for SCK15.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses SCK15_0 at the input pin of MFS ch.15 SCK. Does not produce output. [Initial value]
	01	Uses SCK15_0 at the input pin of MFS ch.15 SCK. Uses SCK15_0 at the output pin.
	10	Uses SCK15_1 at the input pin of MFS ch.15 SCK. Uses SCK15_1 at the output pin.
	11	Uses SCK15_2 at the input pin of MFS ch.15 SCK. Uses SCK15_2 at the output pin.

[bit25:24] SOT15B: SOT15 Input/Output Select bits

Selects input/output for SOT15.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses SOT15_0 at the input pin of MFS ch.15 SOT. Does not produce output. [Initial value]
	01	Uses SOT15_0 at the input pin of MFS ch.15 SOT. Uses SOT15_0 at the output pin.
	10	Uses SOT15_1 at the input pin of MFS ch.15 SOT. Uses SOT15_1 at the output pin.
	11	Uses SOT15_2 at the input pin of MFS ch.15 SOT. Uses SOT15_2 at the output pin.

[bit23:22] SIN15S: SIN15 Input Select bits

Selects input for SIN15.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses SIN15_0 at the input pin of MFS ch.15 SIN. [Initial value]
	01	Same as Writing 00.
	10	Uses SIN15_1 at the input pin of MFS ch.15 SIN.
	11	Uses SIN15_2 at the input pin of MFS ch.15 SIN.

[bit21:20] SCK14B: SCK14 Input/Output Select bits

Selects input/output for SCK14.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses SCK14_0 at the input pin of MFS ch.14 SCK. Does not produce output. [Initial value]
	01	Uses SCK14_0 at the input pin of MFS ch.14 SCK. Uses SCK14_0 at the output pin.
	10	Uses SCK14_1 at the input pin of MFS ch.14 SCK. Uses SCK14_1 at the output pin.
	11	Uses SCK14_2 at the input pin of MFS ch.14 SCK. Uses SCK14_2 at the output pin.

[bit19:18] SOT14B: SOT14 Input/Output Select bits

Selects input/output for SOT14.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses SOT14_0 at the input pin of MFS ch.14 SOT. Does not produce output. [Initial value]
	01	Uses SOT14_0 at the input pin of MFS ch.14 SOT. Uses SOT14_0 at the output pin.
	10	Uses SOT14_1 at the input pin of MFS ch.14 SOT. Uses SOT14_1 at the output pin.
	11	Uses SOT14_2 at the input pin of MFS ch.14 SOT. Uses SOT14_2 at the output pin.

[bit17:16] SIN14S: SIN14 Input Select bits

Selects input for SIN14.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses SIN14_0 at the input pin of MFS ch.14 SIN. [Initial value]
	01	Same as Writing 00.
	10	Uses SIN14_1 at the input pin of MFS ch.14 SIN.
	11	Uses SIN14_2 at the input pin of MFS ch.14 SIN.

[bit15:14] SCK13B: SCK13 Input/Output Select bits

Selects input/output for SCK13.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses SCK13_0 at the input pin of MFS ch.13 SCK. Does not produce output. [Initial value]
	01	Uses SCK13_0 at the input pin of MFS ch.13 SCK. Uses SCK13_0 at the output pin.
	10	Uses SCK13_1 at the input pin of MFS ch.13 SCK. Uses SCK13_1 at the output pin.
	11	Uses SCK13_2 at the input pin of MFS ch.13 SCK. Uses SCK13_2 at the output pin.

[bit13:12] SOT13B: SOT13 Input/Output Select bits

Selects input/output for SOT13.

bit13:12		Description
Reading		Reads out the register value.
Writing	00	Uses SOT13_0 at the input pin of MFS ch.13 SOT. Does not produce output. [Initial value]
	01	Uses SOT13_0 at the input pin of MFS ch.13 SOT. Uses SOT13_0 at the output pin.
	10	Uses SOT13_1 at the input pin of MFS ch.13 SOT. Uses SOT13_1 at the output pin.
	11	Uses SOT13_2 at the input pin of MFS ch.13 SOT. Uses SOT13_2 at the output pin.

[bit11:10] SIN13S: SIN13 Input Select bits

Selects input for SIN13.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses SIN13_0 at the input pin of MFS ch.13 SIN. [Initial value]
	01	Same as Writing 00.
	10	Uses SIN13_1 at the input pin of MFS ch.13 SIN.
	11	Uses SIN13_2 at the input pin of MFS ch.13 SIN.

[bit9:8] SCK12B: SCK12 Input/Output Select bits

Selects input/output for SCK12.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses SCK12_0 at the input pin of MFS ch.12 SCK. Does not produce output. [Initial value]
	01	Uses SCK12_0 at the input pin of MFS ch.12 SCK. Uses SCK12_0 at the output pin.
	10	Uses SCK12_1 at the input pin of MFS ch.12 SCK. Uses SCK12_1 at the output pin.
	11	Uses SCK12_2 at the input pin of MFS ch.12 SCK. Uses SCK12_2 at the output pin.

[bit7:6] SOT12B: SOT12 Input/Output Select bits

Selects input/output for SOT12.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses SOT12_0 at the input pin of MFS ch.12 SOT. Does not produce output. [Initial value]
	01	Uses SOT12_0 at the input pin of MFS ch.12 SOT. Uses SOT12_0 at the output pin.
	10	Uses SOT12_1 at the input pin of MFS ch.12 SOT. Uses SOT12_1 at the output pin.
	11	Uses SOT12_2 at the input pin of MFS ch.12 SOT. Uses SOT12_2 at the output pin.

[bit5:4] SIN12S: SIN12 Input Select bits

Selects input for SIN12.

bit5:4		Description
Reading		Reads out the register value.
Writing	00	Uses SIN12_0 at the input pin of MFS ch.12 SIN. [Initial value]
	01	Same as Writing 00.
	10	Uses SIN12_1 at the input pin of MFS ch.12 SIN.
		Uses SIN12_2 at the input pin of MFS ch.12 SIN.

[bit3:0] Reserved: Reserved bits

"0b0000" is read from these bits.

When writing these bits, set them to "0b0000".

<Notes>

- This register is not initialized by deep standby transition reset.

4.26. Extended Pin Function Setting Register 18 (EPFR18)

The EPFR18 register assigns functions of QPRC and SD Card IF peripheral pins.

■ Register Configuration

bit	31	30	29	28	27	26	25	24
Field	Reserved		SDWPS		SDCDS		SDDATA3B	
Attribute	-		R/W		R/W		R/W	
Initial value	-		00		00		00	

bit	23	22	21	20	19	18	17	16
Field	SDDATA2B		SDDATA1B		SDDATA0B		SDCMDB	
Attribute	R/W		R/W		R/W		R/W	
Initial value	00		00		00		00	

bit	15	14	13	12	11	10	9	8
Field	SDCLKE		Reserved				QZIN3S	
Attribute	R/W		-				R/W	
Initial value	00		-				00	

bit	7	6	5	4	3	2	1	0
Field	QBIN3S		QAIN3S		Reserved			
Attribute	R/W		R/W		-			
Initial value	00		00		-			

■ Register Function

[bit31:30] Reserved: Reserved bits

"0x00" is read from these bits.
When writing these bits, set them to "0x00".

[bit29:28] SDWPS: S_WP input select bits

Selects input for S_WP.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses S_WP_0 at the input pin of SD CARD IF. [Initial value]
	01	Uses S_WP_0 at the input pin of SD CARD IF. (Same as witting 00.)
	10	Setting is prohibited.
	11	Setting is prohibited.

[bit27:26]SDCDS: S_CD input select bits

Selects input for S_CD.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses S_CD_0 at the input pin of SD CARD IF. [Initial value]
	01	Uses S_CD_0 at the input pin of SD CARD IF. (Same as witting 00.)
	10	Setting is prohibited.
	11	Setting is prohibited.

[bit25:24] SDDATA3B: SDDATA3 input/output select bits

Selects input/output for S_DATA3.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses S_DATA3_0 at the input pin of SD CARD IF. [Initial value] Does not produce output.
	01	Uses S_DATA3_0 at the input pin of SD CARD IF. Uses S_DATA3_0 at the output pin.
	10	Setting is prohibited.
	11	Setting is prohibited.

[bit23:22] SDDATA2B: SDDATA2 input/output select bits

Selects input/output for S_DATA2.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses S_DATA2_0 at the input pin of SD CARD IF. [Initial value] Does not produce output.
	01	Uses S_DATA2_0 at the input pin of SD CARD IF. Uses S_DATA2_0 at the output pin.
	10	Setting is prohibited.
	11	Setting is prohibited.

[bit21:20] SDDATA1B: SDDATA1 input/output select bits

Selects input/output for S_DATA1.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses S_DATA1_0 at the input pin of SD CARD IF. [Initial value] Does not produce output.
	01	Uses S_DATA1_0 at the input pin of SD CARD IF. Uses S_DATA1_0 at the output pin.
	10	Setting is prohibited.
	11	Setting is prohibited.

[bit19:18] SDDATA0B: SDDATA0 input/output select bits

Selects input/output for S_DATA0.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses S_DATA0_0 at the input pin of SD CARD IF. [Initial value] Does not produce output.
	01	Uses S_DATA0_0 at the input pin of SD CARD IF. Uses S_DATA0_0 at the output pin.
	10	Setting is prohibited.
	11	Setting is prohibited.

[bit17:16] SDCMDB: S_CMD input/output select bits

Selects input/output for S_CMD.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses S_CMD_0 at the input pin of SD CARD IF. [Initial value] Does not produce output.
	01	Uses S_CMD_0 at the input pin of SD CARD IF. Uses S_CMD_0 at the output pin.
	10	Setting is prohibited.
	11	Setting is prohibited.

[bit15:14] SDCLKE: S_CLK output select bits

Selects output for S_CLK.

bit		Description
Reading		Reads out the register value.
Writing	00	Does not produce output S_CLK_0 of SD CARD IF. [Initial value]
	01	Uses S_CLK_0 at the output pin of SD CARD IF.
	10	Setting is prohibited.
	11	Setting is prohibited.

[bit13:10] Reserved: Reserved bits

"0b0000" is read from these bits.
When writing these bits, set them to "0b0000".

[bit9:8] QZIN3S: QDU-ch3 ZIN input select bits

Selects input for QDU-ch3 ZIN.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses ZIN3_0 at the input pin of ZIN of QDU-ch.3. [Initial value]
	01	Uses ZIN3_0 at the input pin of ZIN of QDU-ch.3. (Same as writing 00.)
	10	Uses ZIN3_1 at the input pin of ZIN of QDU-ch.3.
	11	Uses ZIN3_2 at the input pin of ZIN of QDU-ch.3.

[bit7:6] QBIN3S: QDU-ch3 BIN input select bits

Selects input for QDU-ch3 BIN.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses BIN3_0 at the input pin of BIN of QDU-ch.3. [Initial value]
	01	Uses BIN3_0 at the input pin of BIN of QDU-ch.3. (Same as writing 00.)
	10	Uses BIN3_1 at the input pin of BIN of QDU-ch.3.
	11	Uses BIN3_2 at the input pin of BIN of QDU-ch.3.

[bit5:4] QAIN3S: QDU-ch3 AIN input select bits

Selects input for QDU-ch3 AIN.

bit		Description
Reading		Reads out the register value.
Writing	00	Uses AIN3_0 at the input pin of AIN of QDU-ch.3. [Initial value]
	01	Uses AIN3_0 at the input pin of AIN of QDU-ch.3.　(Same as writing 00.)
	10	Uses AIN3_1 at the input pin of AIN of QDU-ch.3.
	11	Uses AIN3_2 at the input pin of AIN of QDU-ch.3.

[bit3:0] Reserved: Reserved bits

"0b0000" is read from these bits.
When writing these bits, set them to "0b0000".

<Notes>

- This register is not initialized by deep standby transition reset.

4.27. Extended Pin Function Setting Register 19 (EPFR19)

The EPFR19 register is the reserved register.

■ Register Configuration

bit	31	30	29	28	27	26	25	24
Field	Reserved							
Attribute	-							
Initial value								

bit	23	22	21	20	19	18	17	16
Field	Reserved							
Attribute	-							
Initial value								

bit	15	14	13	12	11	10	9	8
Field	Reserved							
Attribute	-							
Initial value								

bit	7	6	5	4	3	2	1	0
Field	Reserved							
Attribute	-							
Initial value								

■ Register Function

[bit31:0] Reserved: Reserved bits

"0x00000000" is read from these bits.
When writing these bits, set them to "0x00000000".

4.28. Extended Pin Function Setting Register 20 (EPFR20)

The EPFR 20 register assigns functions of external bus peripheral pins.

■ Register Configuration

bit	31	30	29	28	27	26	25	24
Field	Reserved							UED31B
Attribute	-							R/W
Initial value	-							0

bit	23	22	21	20	19	18	17	16
Field	UED30B	UED29B	UED28B	UED27B	UED26B	UED25B	UED24B	UED23B
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8
Field	UED22B	UED21B	UED20B	UED19B	UED18B	UED17B	UED16B	UEDTHHB
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

bit	7	6	5	4	3	2	1	0
Field	UEDQM3E	UEDQM2E	UECSXE	UEDWEXE	UECASE	UERASE	UESMCKE	UESMCE
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

■ Register Function

[bit31:25] Reserved: Reserved bits

"0b00000000" is read from these bits.
When writing these bits, set them to "0b00000000".

[bit24] UED31B: UED31B input/output select bit

Selects input/output for external bus data 31.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output user external bus WD31. [initial value] Connects the input of user external bus RD31 to the pin.
	1	Produces output of user external bus MADATA31. Connects the input of user external bus RD31 to the pin.

[bit23] UED30B: UED30B input/output select bit

Selects input/output for external bus data 31.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output of user external bus MADATA30. [Initial value] Connects the input of user external bus RD30 to the pin.
	1	Produces output of user external bus MADATA30. Connects the input of user external bus RD30 to the pin.

[bit22] UED29B: UED29B input/output select bit

Selects input/output for external bus data 29.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output of user external bus WD29 [Initial value] Connects the input of user external bus RD29 to the pin.
	1	Produces output of user external bus WD29. Connects the input of user external bus RD29 to the pin.

[bit21] UED28B: UED28B input/output select bit

Selects input/output for external bus data 28.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output of user external bus WD28. [Initial value] Connects the input of user external bus RD28 to the pin.
	1	Produces output of user external bus WD28. Connects the input of user external bus RD28 to the pin.

[bit20] UED27B: UED27B input/output select bit

Selects input/output for external bus data 27.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output of user external bus WD27. [Initial value] Connects the input of user external bus RD27 to the pin.
	1	Produces output of user external bus WD27. Connects the input of user external bus RD27 to the pin.

[bit19] UED26B: UED26B input/output select bit

Selects input/output for external bus data 26.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output of user external bus WD26. [Initial value] Connects the input of user external bus RD26 to the pin.
	1	Produces output of user external bus WD26. Connects the input of user external bus RD26 to the pin.

[bit18] UED25B: UED25B input/output select bit

Selects input/output for external bus data 25.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output of user external bus WD25. [Initial value] Connects the input of user external bus RD25 to the pin.
	1	Produces output of user external bus WD25. Connects the input of user external bus RD25 to the pin.

[bit17] UED24B: UED24B input/output select bit

Selects input/output for external bus data 24.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output of user external bus WD24. [Initial value] Connects the input of user external bus RD24 to the pin.
	1	Produces output of user external bus WD24. Connects the input of user external bus RD24 to the pin.

[bit16] UED23B: UED23B input/output select bit

Selects input/output for external bus data 23.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output of user external bus WD23. [Initial value] Connects the input of user external bus RD23 to the pin.
	1	Produces output of user external bus WD23. Connects the input of user external bus RD23 to the pin.

[bit15] UED22B: UED22B input/output select bit

Selects input/output for external bus data 22.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output of user external bus WD22. [Initial value] Connects the input of user external bus RD22 to the pin.
	1	Produces output of user external bus WD22. Connects the input of user external bus RD22 to the pin.

[bit14] UED21B: UED21B input/output select bit

Selects input/output for external bus data 21.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output of user external bus WD21. [Initial value] Connects the input of user external bus RD21 to the pin.
	1	Produces output of user external bus WD21. Connects the input of user external bus RD21 to the pin.

[bit13] UED20B: UED20B input/output select bit

Selects input/output for external bus data 20.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output of user external bus WD20. [Initial value] Connects the input of user external bus RD20 to the pin.
	1	Produces output of user external bus WD20. Connects the input of user external bus RD20 to the pin.

[bit12] UED19B: UED19B input/output select bit

Selects input/output for external bus data 19.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output of user external bus WD19. [Initial value] Connects the input of user external bus RD19 to the pin.
	1	Produces output of user external bus WD19. Connects the input of user external bus RD19 to the pin.

[bit11] UED18B: UED18B input/output select bit

Selects input/output for external bus data 18.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output of user external bus WD18. [Initial value] Connects the input of user external bus RD18 to the pin.
	1	Produces output of user external bus WD18. Connects the input of user external bus RD18 to the pin.

[bit10] UED17B: UED17B input/output select bit

Selects input/output for external bus data 17.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output of user external bus WD17. [Initial value] Connects the input of user external bus RD17 to the pin.
	1	Produces output of user external bus WD17. Connects the input of user external bus RD17 to the pin.

[bit9] UED16B: UED16B input/output select bit

Selects input/output for external bus data 16.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output of user external bus WD16. [Initial value] Connects the input of user external bus RD16 to the pin.
	1	Produces output of user external bus WD16. Connects the input of user external bus RD16 to the pin.

[bit8] UEDTHHB: UEDTHHB input/output select bit

Selects input/output for external bus data31 to 16 collectively.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output of user external bus WD31 to WD16. [Initial value] Connects the input of user external bus RD31 to RD16 to the pin.
	1	Produces output of user external bus WD16. Connects the input of user external bus RD16 to the pin.

[bit7] UEDQM3E: UEDQM3E output select bit

Selects output for external bus MDQM3.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output of user external bus MDQM3. [Initial value]
	1	Produces output of user external bus MDQM3.

.

[bit6] UEDQM2E: UEDQM2E output select bit

Selects output for external bus MDQM2.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output of user external bus MDQM2. [Initial value]
	1	Produces output of user external bus MDQM2.

[bit5] UECSXE: UECSXE output select bit

Selects output for external bus SDRAM MCSX8.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output of user external bus MDWEX. [Initial value]
	1	Produces output of user external bus MDWEX.

[bit3] UECASE: UECASE output select bit

Selects output for external bus SDRAM MCASX.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output of user external bus MCASX. [Initial value]
	1	Produces output of user external bus MCASX.

[bit2] UERASE: UERASE output select bit

Selects output for external bus SDRAM MRASX.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output of user external bus MRASX. [Initial value]
	1	Produces output of user external bus MRASX.

[bit1] UESMCEE: UESMCEE output select bit

Selects output for external bus SDRAM MSDCKE.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output of user external bus MSDCKE [Initial value]
	1	Produces output of user external bus MSDCKE.

[bit0] UESMCKE: UESMCKE output select bit

Selects output for external bus SDRAM MADCLK.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not produce output of user external bus MADCLK [Initial value]
	1	Produces output of user external bus MADCLK.

<Notes>

- With EPFR20.bit[8], the input/output select of external bus data[31:16] can be controlled collectively.
  Also with EPFR20.bits[24:9], the input/output select of external bus data[31:16] can be controlled by each bit.
  The settings with EPFR20.bit[8] have the priority over those with EPFR20.bit[24:9].
  To control the input/output selection with the settings of EPFR20.bit[24:9], be sure to set "EPFR20.bit[8]=0".
- This register is not initialized by deep standby transition reset.

4.29. Special Port Setting Register (SPSR)

The SPSR register sets a pin as a signal pin of special functions.

■ Register Configuration

bit	31	30	29	28	27	26	25	24
Field	Reserved							
Attribute	-							
Initial value	-							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
Attribute	-							
Initial value	-							

bit	15	14	13	12	11	10	9	8
Field	Reserved							
Attribute	-							
Initial value	-							

bit	7	6	5	4	3	2	1	0
Field	Reserved		USB1C	USB0C	MAINXC		Reserved	
Attribute	-		R/W	R/W	R/W		-	
Initial value	-		0	0	01		-	

■ Register Function

[bit31:6] Reserved: Reserved bits

"0" is read from these bits.
When writing these bits, set them to "0".

[bit5] USB1C: USB (ch.1) Pin Setting Register

This bit sets a pin as a USB pin.

bit		Description
Reading		Reads out the register value.
Writing	0	Does not use two pins of UDM1 and UDP1 as USB pins but as digital input/output pins. [Initial value]
	1	Uses two pins of UDM1 and UDP1 as USB pins. (An I/O cell will be in a state of input direction and input cut-off.)

[bit4] USB0C: USB (ch.0) Pin Setting Register

This bit sets a pin as a USB pin.

bit4		Description
Reading		Reads out the register value.
Writing	0	Does not use two pins of UDM0 and UDP0 as USB pins but as digital input/output pins. [Initial value]
	1	Uses two pins of UDM0 and UDP0 as USB pins. (An I/O cell will be in a state of input direction and input cut-off.)

[bit3:2] MAINXC: Main Clock (Oscillation) Pin Setting Register

These bits set a pin as a main clock (oscillation) pin.

bit		Description
Reading		Reads out the register value.
Writing	00	Does not use two pins of X0 and X1 as main clock (oscillation) pins but as digital input/output pins.
	01	Uses two pins of X0 and X1 as main clock (oscillation) pins. [Initial value] (An I/O cell will be in a state of input direction, input cut-off, and pull-up disconnection.)
	10	Setting is prohibited.
	11	Uses X0A pin as an external clock input pin. Uses X1A pin as a digital input/output.

[bit1:0] Reserved: Reserved bits

"0b01" is read from these bits.

When writing these bits, set them to "0b01".

<Notes>

- Only writing "01" to the MAINXC bit does not make a main clock start oscillation.
  To start oscillation, enable oscillation by the MOSCE bit of the System Clock Mode Control Register (SCM_CTL), which is described in the chapter "Clock", after writing "01" to the MAINXC bit.
- For procedures of enabling the oscillation of sub-clock, see Chapter “Clock”..
- This register is not initialized by deep standby transition reset.

4.30. Port Pseudo Open Drain Setting Register (PZRx)

PZRx register makes I/O port Hi-Z when output is High level and sets pseudo open drain control.

■ List of PZR register configuration

bit	31 16	15 0	Initial value	Attribute
	Reserved	PZR0	0x0000	R/W
	Reserved	PZR1	0x0000	R/W
	Reserved	PZR2	0x0000	R/W
	Reserved	PZR3	0x0000	R/W
	Reserved	PZR4	0x0000	R/W
	Reserved	PZR5	0x0000	R/W
	Reserved	PZR6	0x0000	R/W
	Reserved	PZR7	0x0000	R/W
	Reserved	PZR8	0x0000	R/W
	Reserved	PZR9	0x0000	R/W
	Reserved	PZRA	0x0000	R/W
	Reserved	PZRB	0x0000	R/W
	Reserved	PZRC	0x0000	R/W
	Reserved	PZRD	0x0000	R/W
	Reserved	PZRE	0x0000	R/W
	Reserved	PZRF	0x0000	R/W

■ Details of Register Configuration

bit	31 16	15 0
Field	Reserved	PZRx

■ Register Function

[bit31:16] Reserved: Reserved bits

"0x0000" is read from these bits.
When writing these bits, set them to "0x0000".

[bit15:0] PZRx: Port Pseudo Open Drain Setting Register x

Sets the pseudo open drain of the pin.

bit15:0		Description
Reading		Reads out the register value.
Writing	0	Set the pin to High level when outputting digital High level by GPIO or peripheral macro.
	1	Set the pin to Hi-Z when outputting digital High level by GPIO or peripheral macro. Disconnect the pull-up resistor regardless of the PCR setting.

<Notes>

- The "x" description of PZRx is wildcard. It shows PZR0, PZR1, PZR2, and so on.
- The function of the PZR register is implemented only in some specific pins.
  Only pins described as "PZR register control is enabled" in remarks column of "I/O circuit type" of Data Sheet can control open drain.

- PZR register does not exist in all pins. However, even the pins that do not have PZR registers can control pseudo open drain by the setting of DDR register if they are used as GPIO.
  In such a case, after setting PFR = 0 (GPIO setting) and PDOR = 0,
  When setting L output: used as DDR = 1 (output direction).
  When setting Hi-Z output: used as DDR = 0 (input direction).

  However, in open drain by the GPIO setting, you cannot apply voltage that exceeds VCC at Hi-Z.

- The settings of P49 to P46 with this register are prohibited. The settings should be made with VBAT Port Pseud Open Drain Setting Register(VBPZR). For details on VBAT, see “VBAT Domain”.

- This register is not initialized by deep standby transition reset.

5. Usage Precautions

This section describes precautions for using the I/O port.

● ON/OFF of the Pull-up Resistance When SPL=1

SPL is a signal for turning a pin into Hi-Z state during standby mode.

- When SPL=0 Normal operations
- When SPL=1 Pin Hi-Z, input cut-off, pull-up disconnection
  However, the SPL bit cannot be used for setting external interrupts, NMIX, JTAG, or TRACE pins.

For details of the SPL bit, see Chapter "Low Power Consumption Mode".

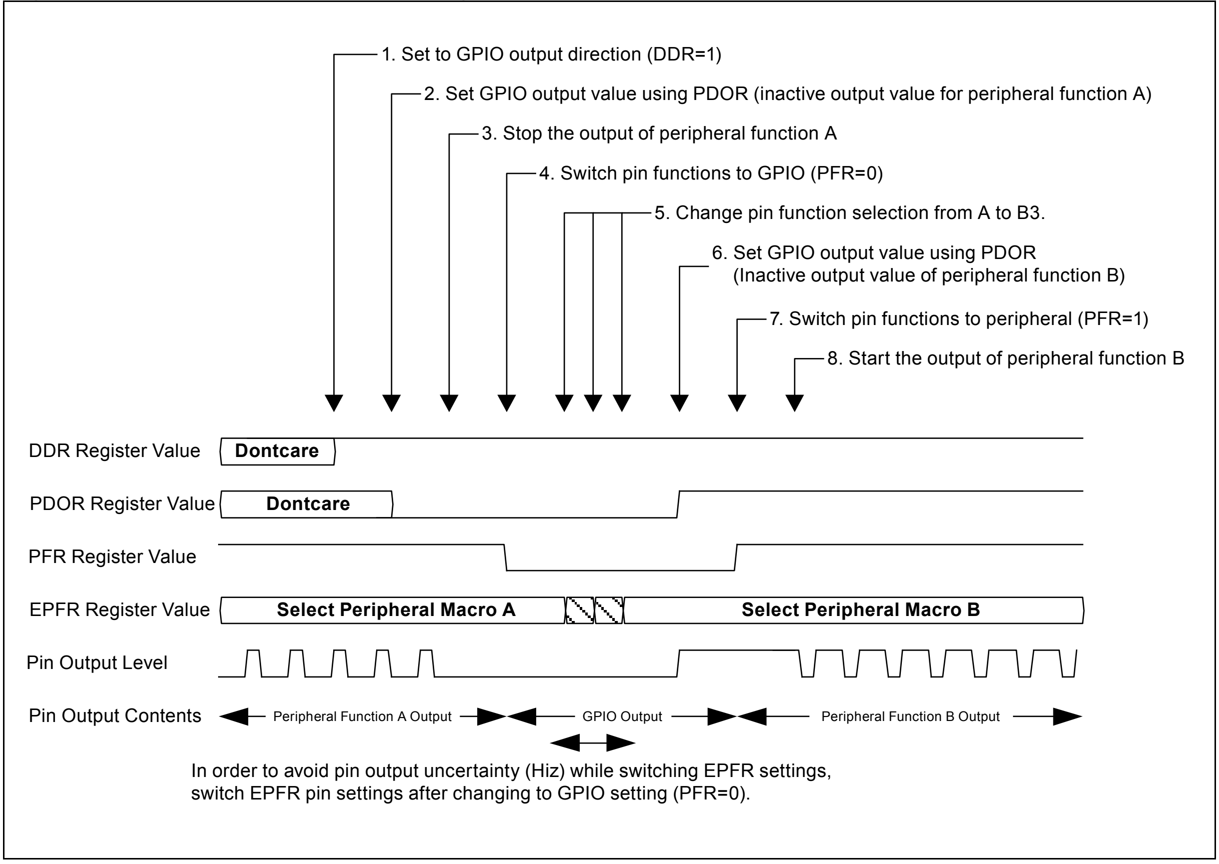
● DTTIX Input

DTTI input is an input signal for switching the dual-purpose motor control PWM output (RTO) setting output pin to its other GPIO pin setting to address a motor stop demand in an emergency.
To use this function, enable switching by EPFR.

● Procedures for Switching Pin Functions

When switching the outputs for peripheral functions using the EPFR register, to prevent pin uncertain output (Hi-Z), switch settings according to the procedures similar to the switching example shown in the following Figure 5-1.

Figure 5-1 Procedures for Switching Pin Functions


1. Set to GPIO output direction (DDR=1)
2. Set GPIO output value using PDOR (inactive output value for peripheral function A)
3. Stop the output of peripheral function A
4. Switch pin functions to GPIO (PFR=0)
5. Change pin function selection from A to B3.
6. Set GPIO output value using PDOR
(Inactive output value of peripheral function B)
7. Switch pin functions to peripheral (PFR=1)
8. Start the output of peripheral function B
DDR Register Value
Dontcare
PDOR Register Value
Dontcare
PFR Register Value
EPFR Register Value
Select Peripheral Macro A
Select Peripheral Macro B
Pin Output Level
Pin Output Contents
Peripheral Function A Output
GPIO Output
Peripheral Function B Output

In order to avoid pin output uncertainty (Hiz) while switching EPFR settings, switch EPFR pin settings after changing to GPIO setting (PFR=0).

● Reserved bit

This bit is read out as "0" except for that of ADE register. When writing, always write "0". The ADE reserved bit is read out as "1". When writing, always write "1".

● Connecting External Bus Pin and SRAM

When accessing SRAM via external bus, either perform pull-up setting for the pin or connect it to external pull-up pin.

● Multi-function Serial Pin Group

When there are some multi-function serial inputs/outputs, set each input/output to the port of the same group. "The port of the same group" means that relocate function numbers attached to the pin name are the same, just like "xxx_0" or "yyy_1".
Table 5-1 shows an example setting.

Table 5-1 Multi-function Serial Interface example setting

Serial Data Output	Serial Clock Input/Output	Serial Data Input	Effective Port
Pin SOT1_0 (Port 0)	Pin SCK1_0 (Port 0)	Pin SIN1_0 (Port 0)	Port 0
		Pin SIN1_1 (Port 1)	Setting is prohibited.
	Pin SCK1_1 (Port 1)	Pin SIN1_0 (Port 0)	
		Pin SIN1_1 (Port 1)	
Pin SOT1_1 (Port 1)	Pin SCK1_0 (Port 0)	Pin SIN1_0 (Port 0)	
		Pin SIN1_1 (Port 1)	
	Pin SCK1_1 (Port 1)	Pin SIN1 (Port 0)	
		Pin SIN1_1 (Port 1)	Port 1

● Peripheral Function Output

As output pins for peripheral functions are uniquely determined by EPFR settings, Output for peripheral functions cannot be assigned to separate pins.
(Disabled example) Assign multifunction serial output SOT1_0 and SOT1_1 to the same output.

● Pin Settings and Operation Mode

For JTAG settings, see Chapter "Debug Interface".
For state of each pin during standby mode or reset, see "Data Sheet" of the product used.

● Product Specifications and Peripheral Function Pin Assignment

Functions which are assigned to pins (GPO, peripheral I/O and special I/O) vary in different products. Please see the pin function table of "Data Sheet" to confirm the pin function of each product. Do not select a function for a pin which is not available in your product by using the EPFR register setting.

● When MDI pin is used as GPIO

To use MD1 pin, the following settings are required.

Input: By reading PDIR, the value is read.
Output: Only L output is available because I/O of MD1 pin is Nch open drain pin.
PFR=0 (Used as GPIO.)
DDR=1(Used as output)
PDOR=0 (Output data is "0".)
SPL=0 (GPIO status is retained in STOP mode.)

● External Interrupt Pin Settings in Standby Mode

When the mode is transferred to the Standby mode under the setting of SPL=1, set PFR=1 and select peripheral functions to enable the external interrupt assignment pin for returning.
If the setting of a pin used for external interrupt is remained PFR=0, unintended operation occurs.

● Effective Range of Deep Standby Transfer Reset

For the effective range of deep standby transfer reset, see Table 5-2.

Table 5-2 Effective Range of Deep Standby Transfer Reset

Registers initialized by Deep Standby Transfer Reset	Registers not initialized by Deep Standby Transfer Reset
PFRx except PFR0[4:0]	PFR0[4:0] PCRx DDRx PDIRx PDORx ADE EPFRx SPSR PZRx

CHAPTER: CRC (Cyclic Redundancy Check)

This chapter explains the CRC functions.

CODE: FS15-E02.3

1. Overview of CRC

The CRC (Cyclic Redundancy Check) is an error detection system. The CRC code is a remainder after an input data string is divided by the pre-defined generator polynomial, assuming the input data string is a high order polynomial. Ordinarily, a data string is suffixed by a CRC code when being sent, and the received data is divided by a generator polynomial as described above. If the received data is dividable, it is judged that the data is correctly received.

■ CRC functions

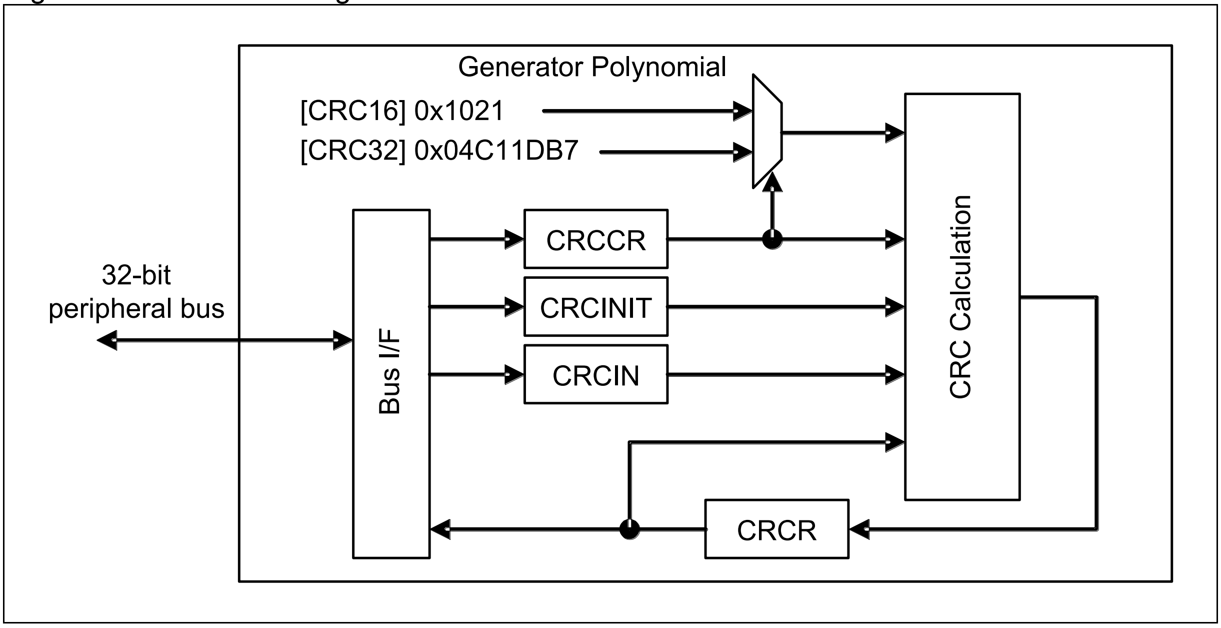
This module enables the calculation in both CCITT CRC16 and IEEE-802.3 CRC32. In this module, the generator polynomial is fixed to the numeric values for those two modes; therefore, the CRC value based on other generator polynomials cannot be calculated.

- CCITT CRC16 generator polynomial: 0x1021
- IEEE-802.3 CRC32 generator polynomial: 0x04C11DB7

■ CRC block diagram

Figure 1-1 shows the CRC block diagram.

Figure 1-1 CRC block diagram



- CRCCR (CRC Control Register)
  Used to control CRC calculation.
- CRCINIT (CRC Initial Value Register)
  Used to specify the initial values for CRC calculation.
- CRCIN (Input Data Register)
  Used to set input data for CRC calculation.
- CRCR (CRC Register)
  Used to output the CRC calculation result.
- CRC Calculation
  A circuit to perform CRC calculation.

2. CRC Operations

This section provides an overview of CRC operations.

■ CRC definition

[CCITT CRC16 Standard]

Generator polynomial	0x1021	(CRCCR:CRC32=0)
Initial value	0xFFFF	
Final XOR value	0x0000	(CRCCR:FXOR=0)
bit order	MSB First	(CRCCR:LSBFST=0)
Output bit order	MSB First	(CRCCR:CRCLSF=0)

(The input-output byte order can be specified arbitrarily.)

[IEEE-802.3 CRC32 Ethernet Standard]

Generator polynomial	0x04C11DB7	(CRCCR:CRC32=1)
Initial value	0xFFFFFFFF	
Final XOR value	0xFFFFFFFF	(CRCCR:FXOR=1)
bit order	LSB First	(CRCCR:LSBFST=1)
Output bit order	LSB First	(CRCCR:CRCLSF=1)

(The input-output byte order can be specified arbitrarily.)

■ Reset operations

When resetting, the Initial Value Register (CRCINIT) and CRC Register (CRCR) are set to 0xFFFFFFFF. Other registers are cleared to "0".

■ Initialization

Initializing with the initialization bit (INIT) of the CRCCR register loads the value of the Initial Value Register to the CRC Register (CRCR).

■ Processing byte and bit orders

The following shows how to process byte and bit orders, using examples.

Input the following one word to the CRC computing unit.

133.82.171.1 = 10000101 01010010 10101011 00000001

If the byte order is set to big endian (CRCCR:LTLEND=0), the sending sequence in bytes is configured as shown below.

```
10000101 01010010 10101011 00000001
  (1st)    (2nd)     (3rd)    (4th)
```

If the bit order is set to Little endian (CRCCR:LSBFST=1), the sending sequence in bits is configured as shown below.

```
10100001 01001010 11010101 10000000
(Head)                        (End)
```

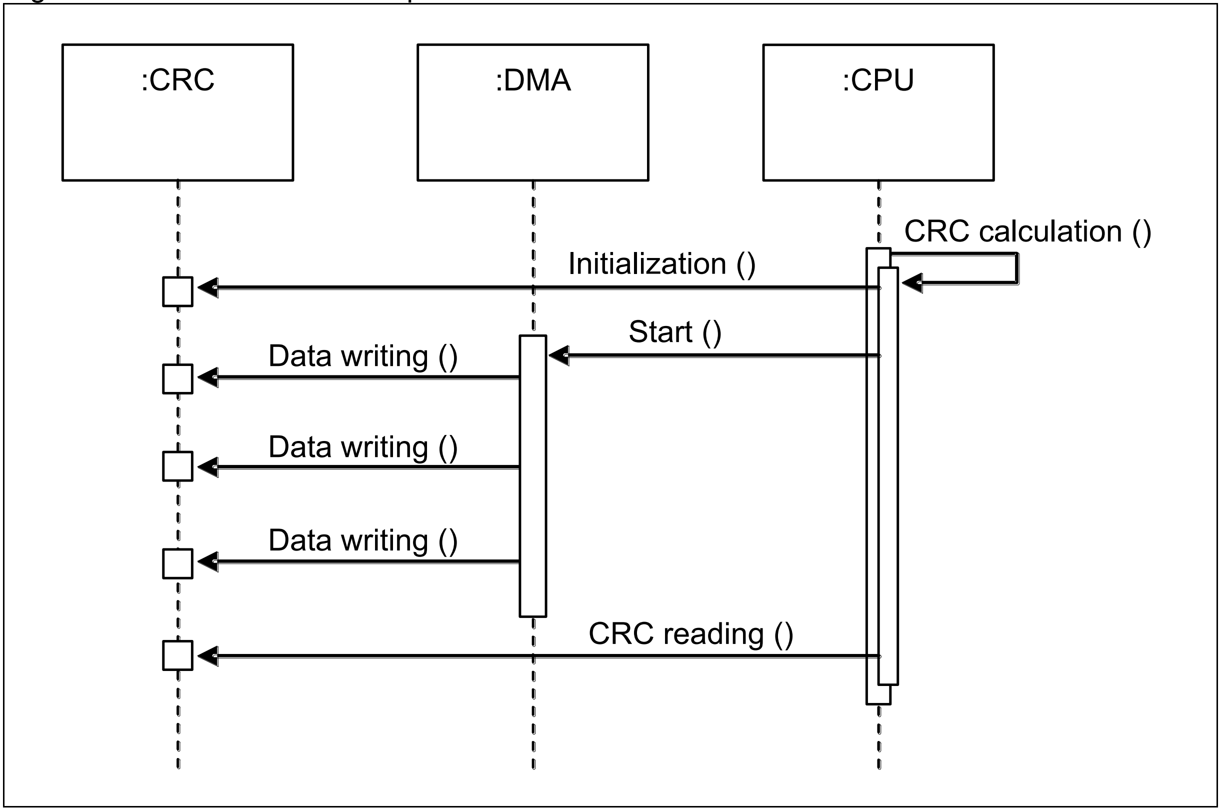
<Note>

At CRCCR:CRCLTE=1, the CRC result is rearranged in bytes with the 32-bit width in both CRC16 and CRC32.
In particular, in CRC16 mode, note that data is output to bit 31 to bit 16.

2.1. CRC calculation sequence

Figure 2-1 shows the CRC calculation sequence. In this section, it is assumed that the Initial Value Register (CRCINIT) setting, CRC16 or CRC32 mode selection (CRCCR:CRC32), and byte- or bit-order setting (CRCCR:LTLEND, CRCCR:LSBFST) have already been configured. (If the initial value can be set to ALL"H", the Initial Value Register (CRCINIT) setting can be omitted.)

Figure 2-1 CRC calculation sequence



- To perform initialization, write "1" to the initial value bit (CRCCR.INIT). The value of the Initial Value Register (CRCINT) is loaded to the CRC Register (CRCR).

- To write input data, write to the Input Data Register (CRCIN). This then starts CRC calculation. If necessary, input data can be written continuously. Furthermore, different bit widths can be used in a sequence to write input data.

- To obtain a CRC code, read the CRC Register (CRCR).

2.2. CRC use examples

Figure 2-2 to Figure 2-5 show CRC use examples.

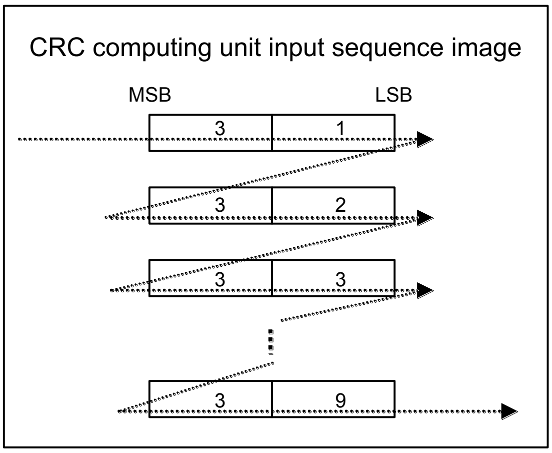
■ Use example 1 CRC16, Byte input fixed

Figure 2-2 Use example 1 (CRC16, byte input fixed, core byte order : Big endian)

```
//*************************************************************
// CRC16 (CRC ITU-T)
//   polynomial:          0x1021
//   initial value:       0xFFFF
//   CRCCR.CRC32:         0 //CRC16
//   CRCCR.LTLEND:        0 //big endian
//   CRCCR.LSBFST:        0 //MSB First
//   CRCCR.CRCLTE:        0 //CRC big endian
//   CRCCR.CRCLSF:        0 //CRC MSB First
//   CRCCR.FXOR:          0 //CRC Final XOR off
//*************************************************************

//
// Example 1-1 (Byte-base writing)
//

// Initialization
B_WRITE (CRCCR, 0x01);

// data write “123456789”
B_WRITE (CRCIN, 0x31);
B_WRITE (CRCIN, 0x32);
B_WRITE (CRCIN, 0x33);
B_WRITE (CRCIN, 0x34);
B_WRITE (CRCIN, 0x35);
B_WRITE (CRCIN, 0x36);
B_WRITE (CRCIN, 0x37);
B_WRITE (CRCIN, 0x38);
B_WRITE (CRCIN, 0x39);

// read result
H_READ (CRCR+2, data);

// check result
assert (data == 0x29B1);

//
// Example 1-2 (CRC check)
//

// Initialization
B_WRITE (CRCCR, 0x01);

// data write "123456789" + CRC
B_WRITE (CRCIN, 0x31);
B_WRITE (CRCIN, 0x32);
B_WRITE (CRCIN, 0x33);
B_WRITE (CRCIN, 0x34);
B_WRITE (CRCIN, 0x35);
B_WRITE (CRCIN, 0x36);
B_WRITE (CRCIN, 0x37);
B_WRITE (CRCIN, 0x38);
B_WRITE (CRCIN, 0x39);
B_WRITE (CRCIN, 0x29); // <-- CRC
B_WRITE (CRCIN, 0xB1); // <-- CRC

// read result
H_READ (CRCR+2, data);

// check result
assert (data == 0x0000);
```

(Assumed as follows.)

B_WRITE	-- Byte writing
H_WRITE	-- Half-word writing
W_WRITE	-- Word writing
B_READ	-- Byte reading
H_READ	-- Half-word reading
W_READ	-- Word reading
CRCCR	-- Control Register address
CRCINIT	-- Initial Value Register address
CRCIN	-- Input Data Register address
CRCR	-- Current CRC Register address



- The byte and half-word writing positions are arbitrary. In this example, data is written continuously at position +0.
- Table 2-1 shows the CPU, CRC result byte order, CRCR (CRC Register) output position, and read address in CRC16 mode.

Table 2-1 CPU, CRC result byte order, and CRCR read address

Core byte order	CRC result byte order	Output position to CRCR	CRCR H_READ address
Big endian	Big endian	bit 15 to bit 0	CRCR +2
Big endian	Little endian	bit 31 to bit 16	CRCR +0
Little endian	Big endian	bit 15 to bit 0	CRCR +0
Little endian	Little endian	bit 31 to bit 16	CRCR +2

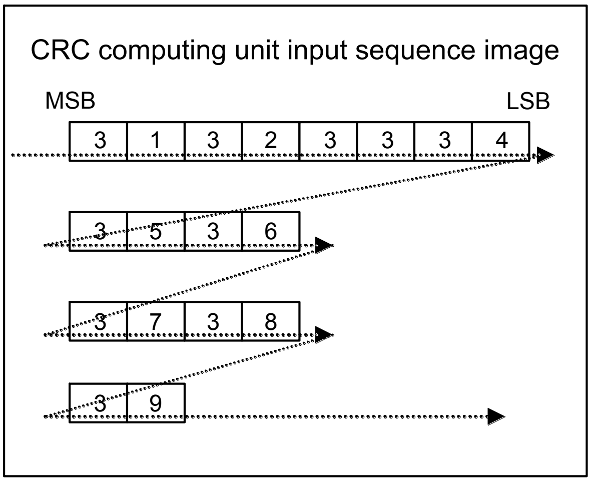
■ Use example 2　CRC16, different input bit widths mixed

Figure 2-3 Use example 2 (CRC16, different input bit widths mixed, core byte order: Big endian)

```
//*************************************************************
//  CRC16 (CRC ITU-T)
//     polynomial:            0x1021
//     initial value:         0xFFFF
//     CRCCR.CRC32:           0 //CRC16
//     CRCCR.LTLEND:          0 //big endian
//     CRCCR.LSBFST:          0 //MSB First
//     CRCCR.CRCLTE:          0 //CRC big endian
//     CRCCR.CRCLSF:          0 //CRC MSB First
//     CRCCR.FXOR:            0 //CRC Final XOR off
//*************************************************************

//
// Example 2-1 (Writing widths mixed)
//

// Initialization
B_WRITE (CRCCR, 0x01);

// data write "123456789"
W_WRITE (CRCIN, 0x31323334);
H_WRITE (CRCIN, 0x3536);
H_WRITE (CRCIN+2, 0x3738);
B_WRITE (CRCIN+3, 0x39);

// read result
H_READ (CRCR+2, data);

// check result
assert  (data == 0x29B1);

//
// Example 2-2 (CRC check)
//

// Initialization
B_WRITE (CRCCR, 0x01);

// data write "123456789" + CRC
W_WRITE (CRCIN, 0x31313334);
W_WRITE (CRCIN, 0x35363738);
H_WRITE (CRCIN, 0x3929); // <-- CRC(0x29)
B_WRITE (CRCIN, 0xB1); // <-- CRC(0xB1)

// read result
H_READ (CRCR+2, data);

// check result
assert (data == 0x0000);
```

(Assumed as follows.)

B_WRITE	-- Byte writing
H_WRITE	-- Half-word writing
W_WRITE	-- Word writing
B_READ	-- Byte reading
H_READ	-- Half-word reading
W_READ	-- Word reading
CRCCR	-- Control Register address
CRCINIT	-- Initial Value Register address
CRCIN	-- Input Data Register address
CRCR	-- Current CRC Register address



- If the byte or bit order setting is correct and the bit input sequence to the CRC computing unit is the same, the writing width can be specified arbitrarily.
  For example, if a 1-, 2-, or 3-byte fraction is finally obtained in the word-base writing mode, both byte and half-word writings may be enabled.

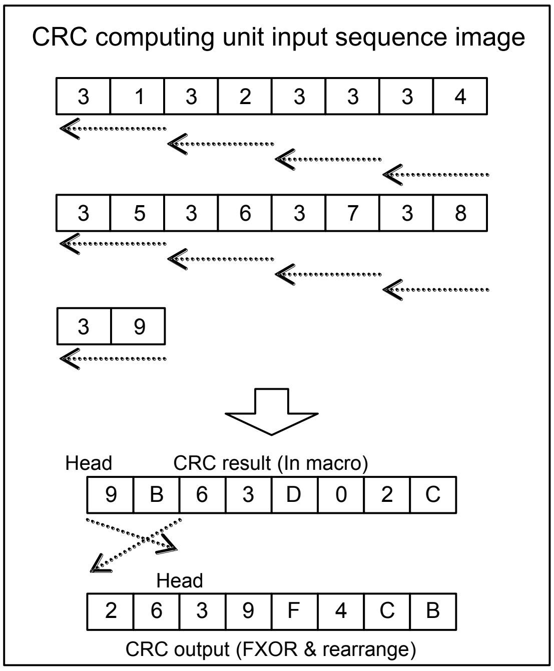
■ Use example 3 CRC32, byte order: Big endian

Figure 2-4 Use example 3 (CRC32, byte order: Big endian)

```
//*************************************************************
// CRC32 (IEEE-802.3)
// polynomial:           0x04C11DB7
// initial value:        0xFFFF_FFFF
// CRCCR.CRC32           1 // CRC32
// CRCCR.LTLEND:         0 // big endian
// CRCCR.LSBFST:         1 // LSB First
// CRCCR.CRCLTE:         0 // CRC big endian
// CRCCR.CRCLSF:         1 // CRC LSB First
// CRCCR.FXOR:           1 // CRC Final XOR on
//*************************************************************

//
// Example 3-1 (CRC32)
//

// Initialization
B_WRITE (CRCCR, 0x6B);

// data write "123456789"
W_WRITE (CRCIN, 0x31323334);
W_WRITE (CRCIN, 0x35363738);
B_WRITE (CRCIN, 0x39);

// read result
W_READ (CRCR, data);

// check CRC result
assert (data == 0x2639F4CB); // <- big endian & LSB First
```

(Assumed as follows.)

B_WRITE	-- Byte writing
H_WRITE	-- Half-word writing
W_WRITE	-- Word writing
B_READ	-- Byte reading
H_READ	-- Half-word reading
W_READ	-- Word reading
CRCCR	-- Control Register address
CRCINIT	-- Initial Value Register address
CRCIN	-- Input Data Register address
CRCR	-- Current CRC Register address



· In CRC32 (IEEE-802.3) mode, the bit order is set to LSB First. This CRC computing unit supports both the big endian and little endian as the byte order. Figure 2-4 shows an example for big endian.

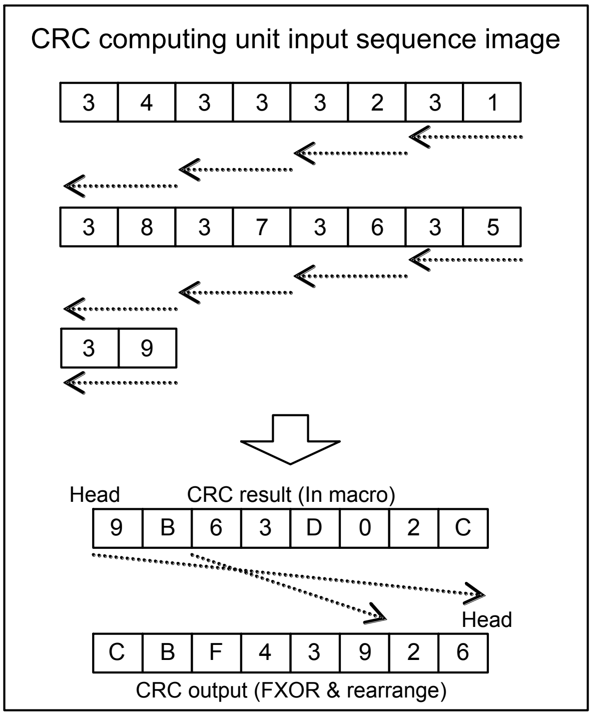
■ Use example 4 CRC32, byte order: Little endian

Figure 2-5 Use example 4 (CRC32, byte order: Little endian)

```
//************************************************************
// CRC32 (IEEE-802.3)
// polynomial:            0x04C11DB7
// initial value:         0xFFFF_FFFF
// CRCCR.CRC32            1 // CRC32
// CRCCR.LTLEND:          1 // big endian
// CRCCR.LSBFST:          1 // LSB First
// CRCCR.CRCLTE:          1 // CRC little endian
// CRCCR.CRCLSF:          1 // CRC LSB First
// CRCCR.FXOR:            1 // CRC Final XOR on
//************************************************************

//
// Example 4-1 (CRC32)
//

// Initialization
B_WRITE (CRCCR, 0x7F);

// data write "123456789"
W_WRITE (CRCIN, 0x34333231);
W_WRITE (CRCIN, 0x38373635);
B_WRITE (CRCIN, 0x39);

// read result
W_READ (CRCR, data);

// check result
assert (data == 0xCBF43926); // <- little endian & LSB First
```

(Assumed as follows.)

B_WRITE	-- Byte writing
H_WRITE	-- Half-word writing
W_WRITE	-- Word writing
B_READ	-- Byte reading
H_READ	-- Half-word reading
W_READ	-- Word reading
CRCCR	-- Control Register address
CRCINIT	-- Initial Value Register address
CRCIN	-- Input Data Register address
CRCR	-- Current CRC Register address



· In CRC32 (IEEE-802.3) mode, the bit order is set to LSB First. This CRC computing unit supports both the big endian and little endian as the byte order. Figure 2-5 shows an example for little endian.

· If bit inversion is not required for the CRC result, perform either one of the following processes to release the bit inversion for the current result.
  · Initialize with 0x3F before calculation.
  · After data was input, set the CRCCR:FXOR bit to "0" (for example, CRCCR=0x3E).

3. CRC Registers

This section provides a list of CRC registers.

■ CRC registers

Table 3-1 CRC register list

Abbreviation	Register name	Reference
CRCCR	CRC Control Register	3.1
CRCINIT	Initial Value Register	3.2
CRCIN	Input Data Register	3.3
CRCR	CRC Register	3.4

3.1. CRC Control Register (CRCCR)

The CRC Control Register (CRCCR) is used to control CRC calculation.

bit	7	6	5	4	3	2	1	0
Field	Reserved	FXOR	CRCLSF	CRCLTE	LSBFST	LTLEND	CRC32	INIT
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

[bit7] Reserved: Reserved bit

The read value is "0".
Be sure to write "0" to this bit.

[bit6] FXOR: Final XOR control bit

This bit is used to output the CRC result as the XOR value or XOR.
The OR value is set to ALL"H". This bit is inverted at FXOR=1.
This processing is performed in the latter part of the CRC Register processing. The CRC result is therefore reflected on the read value immediately after this bit was set.

bit	Description
0	None
1	Yes

[bit5] CRCLSF: CRC result bit-order setting bit

This is a bit-order setting bit for CRC result.
This bit is used to rearrange bits within each byte. Set "0" to specify MSB First and set "1" to specify LSB First.
This processing is performed in the latter part of the CRC Register processing. The CRC result is therefore reflected on the read value immediately after this bit was set.

bit	Description
0	MSB First
1	LSB First

[bit4] CRCLTE: CRC result byte-order setting bit

This is a byte-order setting bit for CRC result.

This bit is used to rearrange the byte order in each word. Set "0" to specify big endian and set "1" to specify little endian.

This processing is performed in the latter part of the CRC Register processing. The CRC result is therefore reflected on the read value immediately after this bit was set.

If this bit is set to "1" in CRC16 mode, data is output to bit31:16.

bit	Description
0	Big endian
1	Little endian

[bit3] LSBFST: bit-order setting bit

This is a bit-order setting bit.

This bit is used to specify the head bit of a byte (8 bits). Set "0" to specify MSB First and set "1" to specify LSB First.

Four types of processing orders can be specified when this bit is combined with the LTLEND bit setting.

bit	Description
0	MSB First
1	LSB First

[bit2] LTLEND: Byte-order setting bit

This is a byte-order setting bit.

This bit is used to specify the byte order with the write width. Set "0" to specify big endian and set "1" to specify little endian.

bit	Description
0	Big endian
1	Little endian

[bit1] CRC32: CRC mode selection bit

This bit is used to select the CRC16 or CRC32 mode.

bit	Description
0	CRC16
1	CRC32

[bit0] INIT: Initialization bit

This is an initialization bit. Writing "1" to this bit initializes data. This bit does not have a value, and always returns "0" at reading.

At initialization, the value of the Initial Value Register is loaded to the CRC Register.

Initialization must be performed once at the start of CRC calculation.

bit	Description	
	Write	Read
0	Invalid	Always reads "0".
1	Initialization	

3.2. Initial Value Register (CRCINIT)

The Initial Value Register (CRCINIT) is used to save the initial values for CRC calculation.

bit	31	0
Field	D[31:0]	
Attribute	R/W	
Initial value	0xFFFFFFFF	

[bit31:0] D[31:0] : Initial value bits

These bits are used to save the initial values for CRC calculation.
Write the initial values for CRC calculation to this register.

(0xFFFFFFFF at resetting)
In CRC16 mode, D15 to D0 are used while D31 to D16 are ignored.

3.3. Input Data Register (CRCIN)

The Input Data Register (CRCIN) is used to set input data for CRC calculation.

bit	31	0
Field	D[31:0]	
Attribute	R/W	
Initial value	0x00000000	

[bit31:0] D[31:0] : Input data bits

These bits are used to set input data for CRC calculation.

Write input data for CRC calculation to this register. There are three types of bit widths: 8-bit, 16-bit, and 32-bit (byte, half word, word), which can be specified together.

The byte and half-word writing positions are arbitrary. The available address positions are as follows.

Byte writing	: +0, +1, +2, +3
Half-word writing	: +0, +2

3.4. CRC Register (CRCR)

The CRC Register (CRCR) is used to output the CRC calculation result. This register must be initialized before start calculating.

bit	31	0
Field	D[31:0]	
Attribute	R	
Initial value	0xFFFFFFFF	

[bit31:0] D[31:0] : CRC bits

These bits are used to read the CRC calculation result. If "1" is written to the initialization bit (CRCCR:INIT), the value of the Initial Value Register (CRCINIT) is loaded to this register.
If input data for CRC calculation is written to the Input Data Register (CRCIN), the CRC calculation result is set to this register after one machine clock cycle has elapsed. When all input data writing has been completed, this register holds the final CRC code.
In CRC16 mode, when the byte order is set to big endian (CRCLTE=0), the result is output to D15 to D0. When the byte order is set to little endian (CRCLTE=1), the result is output to D31 to D16.

CHAPTER: External Bus Interface

This chapter explains the functions and operations of the external bus interface.

CODE: 9BFEXTBUS_FM4-E01.0_MEMCS-E1.5

1. Overview of External Bus Interface

This section explains an overview of the external bus interface.

The external bus interface allows connections with SRAM/Flash memory/SDRAM outside of the device.

■ External bus interface features

The features of the external bus interface across the products are as follows:

- Supports connections with 8-bit/16-bit wide SRAM/NOR Flash memories/NAND Flash memories. Normal SRAM accesses are used for accessing the NOR Flash memories but special pins are available for accessing the NAND Flash memories.
- Up to 8 chip select signals are available. One chip select signal is provided to SDRAM exclusively. Address and access timing parameters can be separately set for each chip select signal.
- Up to 25 bits address can be output.
- Supports NOR Flash memory page read.
- Byte lane is fixed to little endian.
- When the access width from CPU and the external bus width are different, the bus size will automatically be converted.
- Pins dedicated for NAND Flash memories allow accessing other SRAM devices which share the data line during accessing the NAND Flash memories (exclusive access control is not required).
- Separate mode and multiplex mode are supported for bus accesses. The page read of NOR flash memory, NAND flash memory, and SDRAM do not support multiplex mode.
- The access timing parameter of ALE signal is added to support the multiplex mode. In addition, more detailed parameter settings, such as CS assert timing, are possible.
- Clock output feature allows synchronous accesses with target devices.
- Supports external RDY feature.
- Supports SDRAM power down mode.

<Note>

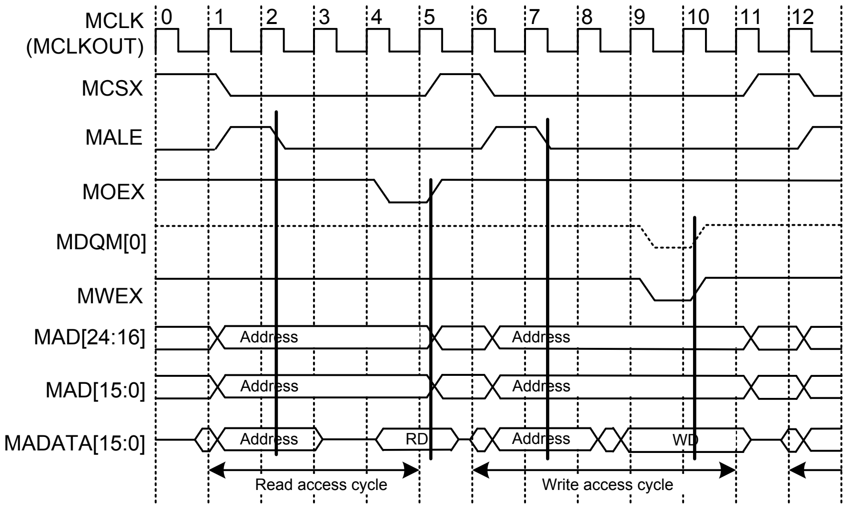
The bit width to be connected to SARAM, Flash memory, and SDRAM differs depending on products. For details, see “Data Sheet” of the product used. The further description explains 16-bit width mode.

■ Access Timing and AC Specifications

● Asynchronous Accesses

The external bus interface performs read data latching to the timing of the output enable (MOEX) signal in reading data. Make the target device to perform the write data latching to the timing of the write enable (MWEX) signal in writing data. An example of the asynchronous access is shown in Figure 1-1.

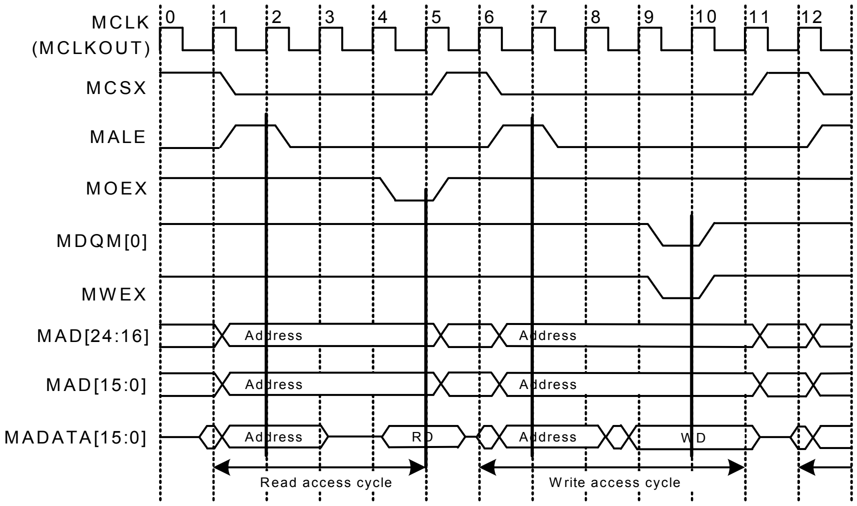
Figure 1-1 Asynchronous Access



● Synchronous Accesses

The external bus interface performs read data latching synchronized with the clock output in reading data. Make the target device to perform the write data latching synchronized with the clock output in writing data. An example of the synchronous access is shown in Figure 1-2.

Figure 1-2 Synchronous Access



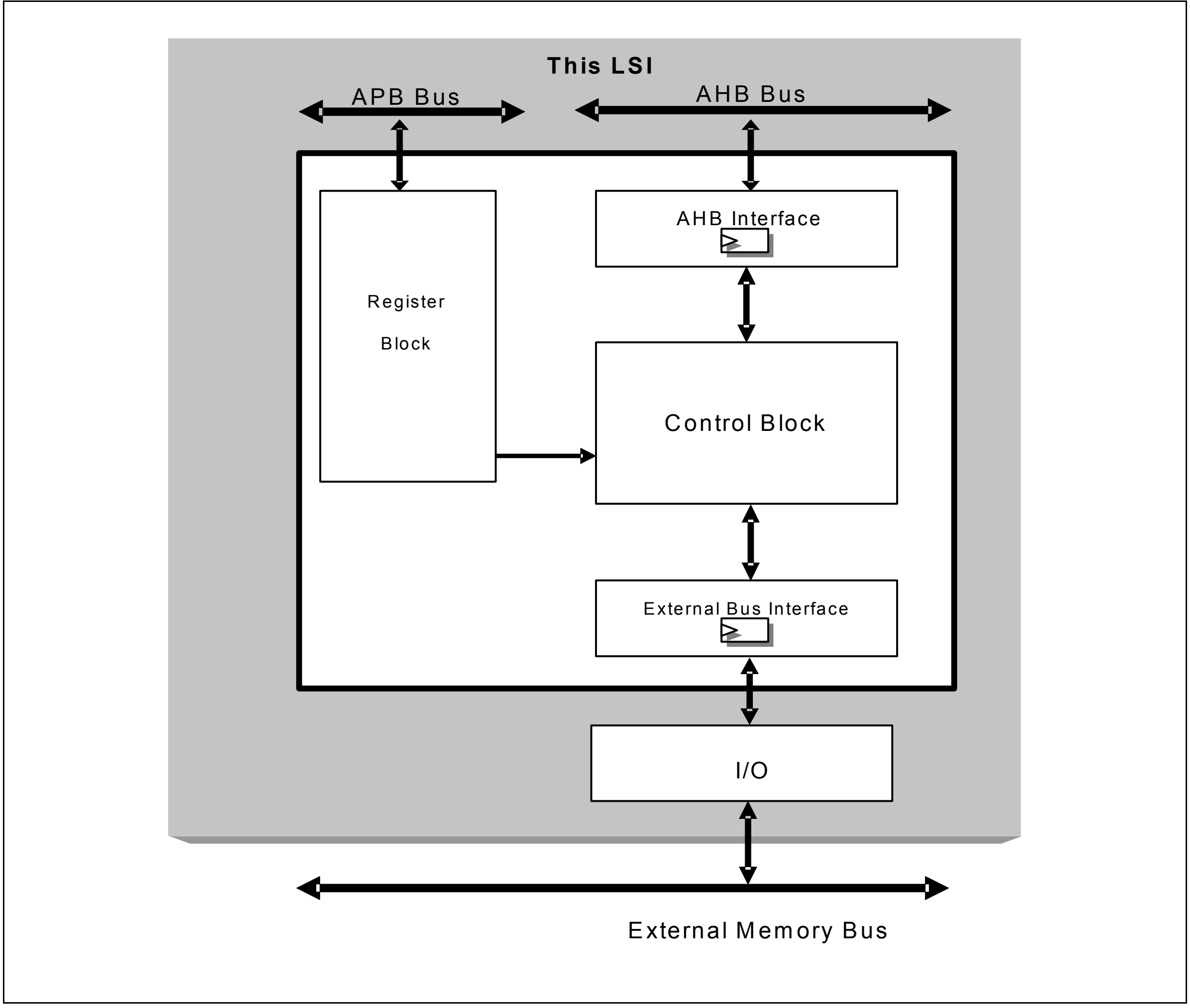
<Note>

See "Data Sheet" of the product used for details of the AC specifications.

2. Block Diagram

This section explains the block diagram of the external bus interface.

Figure 2-1 External bus interface block diagram

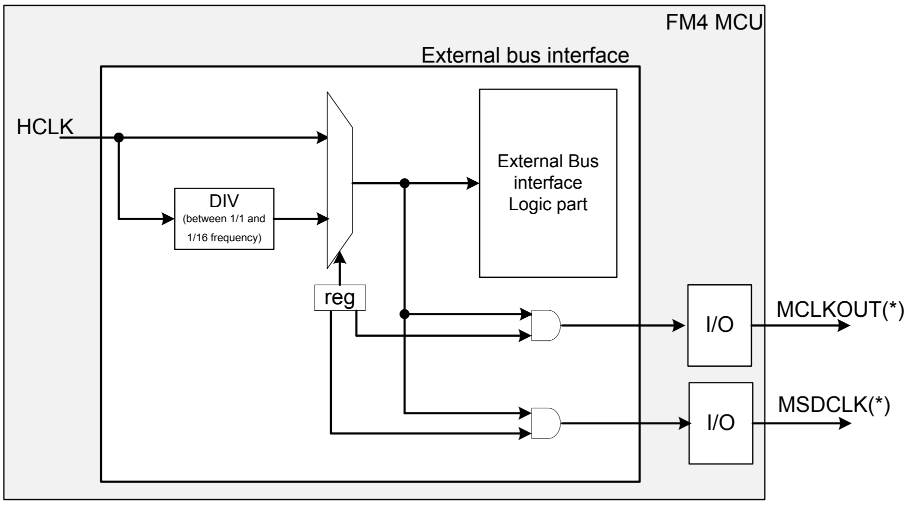


- Register Block
  Registers which set the features of this interface. These are connected with the APB bus.

- Control Block
  This block controls the operations of this interface. This block is connected with the AHB bus.

- External Bus Interface
  This interface connects the function block and the external memory bus.

Figure 2-2 External bus interface block diagram (Clock system overview diagram)


FM4 MCU
External bus interface
HCLK
DIV
(between 1/1 and 1/16 frequency)
External Bus interface Logic part
reg
I/O
MCLKOUT(*)
I/O
MSDCLK(*)

<Notes>

- HCLK indicates a master clock. For the details, see the chapter "Clock".
- To output external bus interface operation clock, a clock output setting is required in GPIO. For the details of the setting, see the chapter "I/O Port".
- To output a clock for SRAM/flash memory (MCLKOUT), make sure to set DCLKR:MCLKON = 1 and execute frequency division setting. At this time, a frequency between 1/1 and 1/16 can be set.
- To output a clock for SDRAM(MSDCLK) , make sure to set SDMODE:MSDCLKOFF=0.
- To output the external bus interface operation clock, set the clock output with GPIO. For details on the settings, see Chapter “I/O Port”.

- Pin List
  The pin list of the external bus interface is shown in Table 2-1.

Table 2-1 Pin List of External Bus Interface

Pin Name	Function
MAD[24:0]	Address output pins
MADATA[15:0]	Data input/output pins (These pins will be changed to input/output pins for address/data in multiplex mode.)
MCSX[7:0]	Chip select pins for SRAM and flash memory
MDQM[1:0]	Byte mask signal output pins
MALE	Address latch enable output pin (Multiplex mode only)
MOEX	Output enable output pin
MWEX	Write enable output pin
MRDY	RDY signal input pin
MCLKOUT	Clock output pin
MNALE	Address latch enable output pin for NAND Flash memories
MNCLE	Chip enable output pin for NAND Flash memories
MNREX	Read enable output pin for NAND Flash memories
MNWEX	Write enable output pin for NAND Flash memories
MCSX[8]	Chip select pin for SDRAM
MRASX	Low address strobe output pin for SDRAM
MCASX	Column address strobe output pin for SDRAM
MSDWEX	Write enable output pin for SDRAM
MSDCKE	Clock enable output pin for SDRAM
MSDCLK	Clock output pin for SDRAM

<Notes>

- Placement of the external bus interface pins depends on the product type. See the data sheets of products used for the details.

3. Operations

The section explains the operations of the external bus interface.

3.1. Bus Access Mode

The following explains bus access mode.

■ Access Method

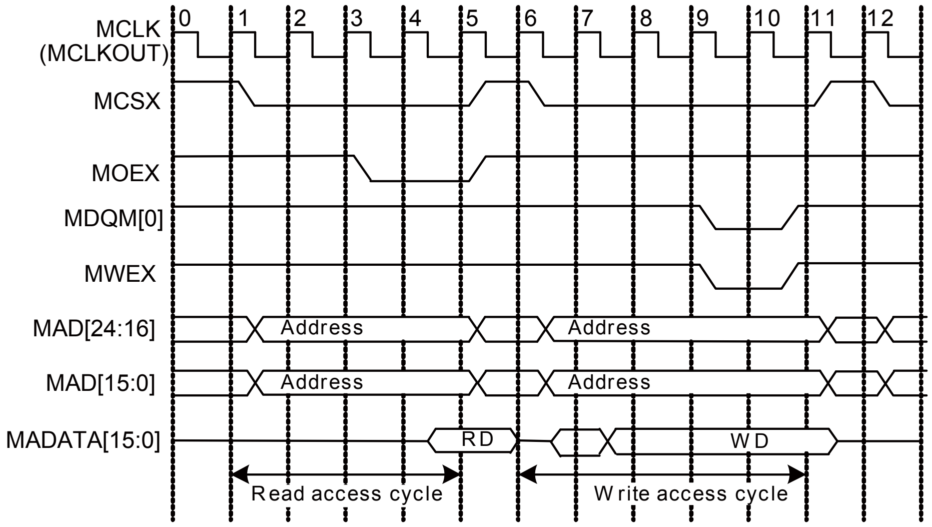
The external bus interface allows selecting separate mode or multiplex mode with the register.

● Separate Mode

This mode outputs the address to the MAD[24:0] pins and inputs/outputs the data to the MADATA[15:0] pins.

As the address pins and data pins are separated each other, connecting directly with the normal SRAM and high speed accesses can be achieved. An example of waveform in separate mode is shown in Figure 3-1.

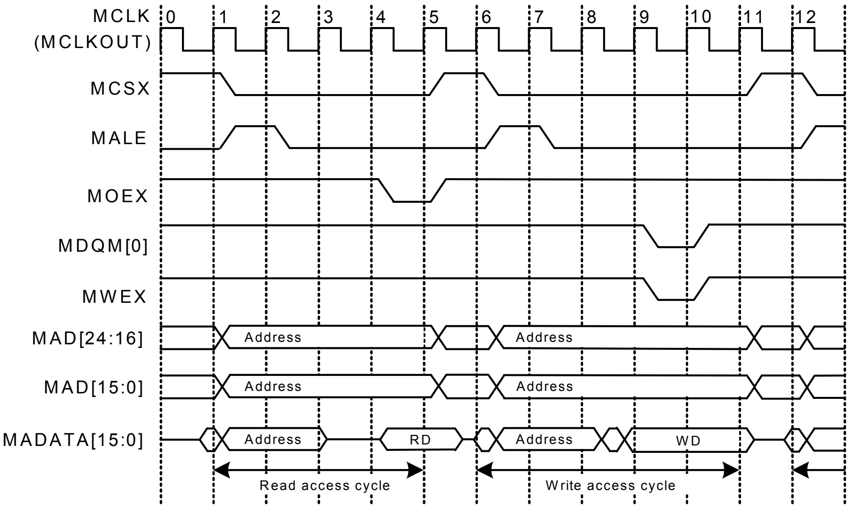
Figure 3-1 Example of Waveform in Separate Mode



● Multiplex Mode

This mode inputs/outputs the address/data to MADATA[15:0] pins in a time division manner. As the part of address pins and data pins are shared, fewer pins are used for accessing the external memories. An example of waveform in multiplex mode is shown in Figure 3-2.

Figure 3-2 Example of Waveform in Multiplex Mode



Summary of selected bus access mode and their pin functions is shown in Table 3-1.

Table 3-1 Bus Access Mode and Pin Function

Pin	16-bit Separate	16-bit Multiplex	8-bit Separate	8-bit Multiplex
MAD[24:16]	Address [24:16]	Address [24:16]	Address [24:16]	Address [24:16]
MAD[15:8]	Address [15:8]	(Address [15:8])	Address [15:8]	(Address [15:8])
MAD[7:0]	Address [7:0]	(Address [7:0])	Address [7:0]	(Address [7:0])
MADATA[15:8]	Data [15:8]	Address [15:8] Data [15:8] Multiplex	No output	(Address [15:8])
MADATA[7:0]	Data [7:0]	Address [7:0] Data [7:0] Multiplex	Data [7:0]	Address [7:0] Data [7:0] Multiplex

(Note) Address output from MAD pins in multiplex mode is available depending on the GPIO setups.

■ Bus Access Modes and Functions Setups

Bus access modes and the functions setups are shown in Table 3-2.

Table 3-2 Bus Access Modes and Functions Setups (Products other than TYPE0)

Bus Access Mode	Clock output	External RDY	Page read	NAND Flash	Clock division
Separate bus mode	○	○	○	○	○
Multiplex mode	○	○	Not allowed	Not allowed	○

Bus Access Mode	SDRAM
Separate bus mode	○
Multiplex mode	Not allowed

Do not make a setup which uses page read and NAND Flash mode at the same time.

Do not make a setup which uses page read and external RDY at the same time.

Do not make a setup which uses NAND Flash mode and external RDY at the same time.

Do not output the clock in NAND Flash mode.

<Notes>

Number and functions of external bus interface pins used depend on the product used. See the data sheets of products used for the details.

■ Bus Size Conversion and Continuous Access

If an access with an externals bus width narrower than the CPU access width is made, the access will be divided and converted into continuous accesses which continuously change the address only with holding MCSX="L". For example, when a 32-bit read access is made from the internal bus to the 8-bit bus width, the address will be changed as 0 -> 1 -> 2 -> 3 with holding MCSX="L" and the data will be output continuously from the MADATA [7:0] with the transfer timing.
The word read access waveform to the 8-bit width SRAM is shown in Figure 3-3.
The continuous word write/read access waveform to the 16-bit width SRAM is shown in Figure 3-4.

Figure 3-3 Waveform of Word Read Access to 8-bit Width SRAM

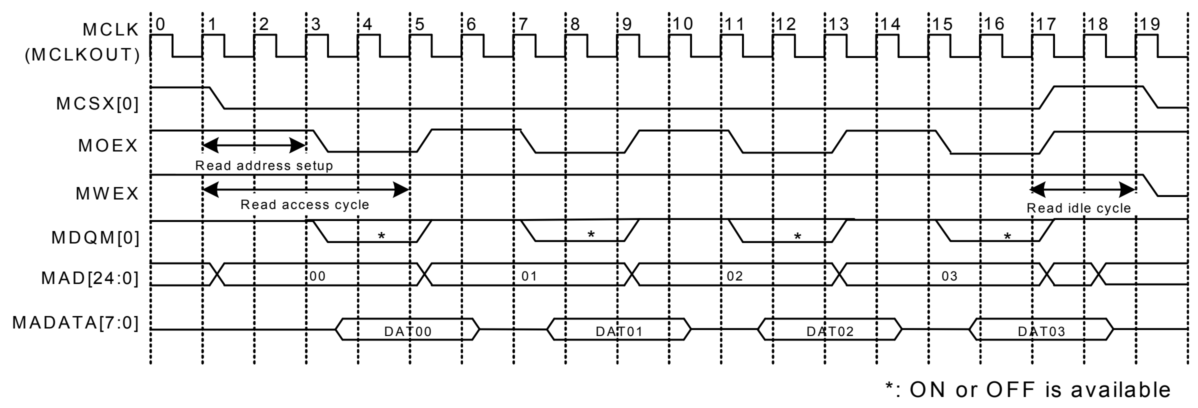
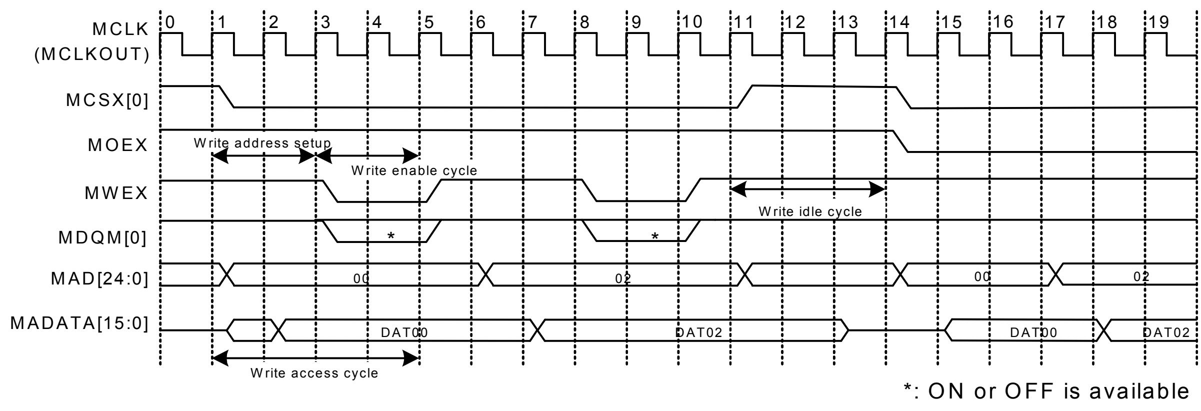


Figure 3-4 Waveform of Continuous Word Write/Read Access to 16-bit Width SRAM



<Note>

The idle cycle in continuous access will be inserted only after the access to the last address.

Table 3-3 shows the mapping between the CPU access width and the external bus width.

Table 3-3 CPU Access Width and External Bus Width Mapping

External bus width	Access from CPU		Access to external bus			
	Access type	Address	Access order	Output value from MAD[1:0]	Valid data at MADATA[15:0]	Output value from MDQM[1:0]
8-bit	Byte (8bits)	0	No division	0b00	MADATA[7:0]	0b10
		1	No division	0b01		0b10
		2	No division	0b10		0b10
		3	No division	0b11		0b10
	Half-word (16bits)	0	1/2 access	0b00	MADATA[7:0]	0b10
			2/2 access	0b01		0b10
	Half-word (16bits)	2	1/2 access	0b10		0b10
			2/2 access	0b11		0b10
	Word (32bits)	0	1/4 access	0b00	MADATA[7:0]	0b10
		1	2/4 access	0b01	MADATA[7:0]	0b10
		2	3/4 access	0b10	MADATA[7:0]	0b10
		3	4/4 access	0b11	MADATA[7:0]	0b10
16-bit	Byte (8bits)	0	No division	0b00	MADATA[7:0]	0b10
		1	No division	0b00	MADATA[15:8]	0b01
		2	No division	0b10	MADATA[7:0]	0b10
		3	No division	0b10	MADATA[15:8]	0b01
	Half-word (16bits)	0	No division	0b00	MADATA[15:0]	0b00
		2	No division	0b10	MADATA[15:0]	0b00
	Word (32bits)	0	1/2 access	0b00	MADATA[15:0]	0b00
			2/2 access	0b10	MADATA[15:0]	0b00

HADDR: AHB address input

As for a target with 8-bit width, the input/output data will be determined with the values of HADDR[1:0]. As for a target with 16-bit width, the HADDR[1] is only used for data assignment.

<Note>

General purpose ports or shared function can be selected per bit for the MAD pins and MADATA pins. See Chapter "I/O Port" for the detail of the setups.

3.2. SRAM and NOR Flash Memories Access

The following explains SRAM and NOR Flash memories access.

■ Memory Access

The target device for the SRAM and NOR Flash memories access will be determined with the MCSX [7:0]/address outputs. After that, outputting MOEX/MWEX will make a read/write to the target device.

■ Pins Used

SRAM and NOR Flash memory accesses require the pins shown in Table 3-4.

Table 3-4 External Interface Pins used for SRAM and NOR Flash Memories

Pin name	Function
MAD[24:0]	Address output pins
MADATA[15:0]	Data input/output pins (These pins will be changed to input/output pins for address/data in multiplex mode.)
MCSX[7:0]	Chip select pins
MDQM[1:0]	Byte mask signal output pins
MALE	Address latch enable output pin (Multiplex mode only)
MOEX	Output enable output pin
MWEX	Write enable output pin
MRDY	RDY signal input pin
MCLKOUT	Clock output pin

<Notes>

- Not all of the pins shown in Table 3-3 will be used depending on the setups or target devices(SRAM, or NOR Flash memory).
- Number and functions of external bus interface pins used depend on the product used. See the data sheet of the products used for the details.

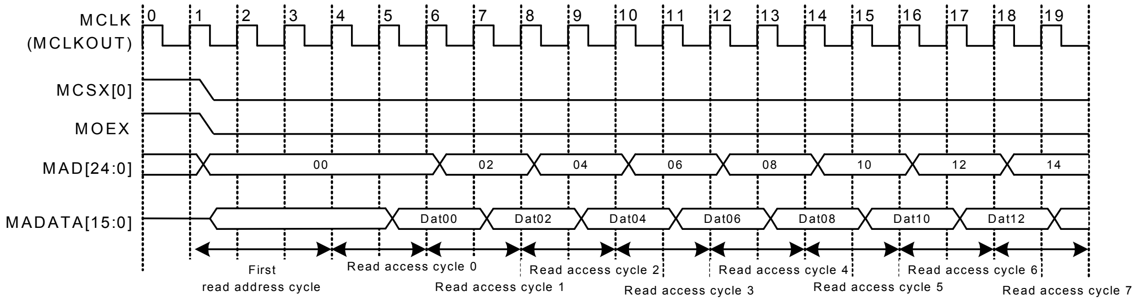
■ Page Read for 16-bit NOR Flash Memories

The page read operation is available for NOR Flash memories by setting PAGE bit which is the fifth bit of the mode register (MODE0 to MODE7) to "1".

The page read will continue read cycle while in reading operation to the boundary of the 16 bytes with MOEX=L retained. The waveform of 16-bit NOR Flash memory page read is shown in Figure 3-5.

The address is retained within the first cycle for specified cycles (First read address cycle). The accesses after the first cycle will be made with the number of cycles specified by the RACC.

Figure 3-5 Waveform of 16-bit NOR Flash Memory Page Read



- MOEX will be asserted at the same timing of MCSX.
- The first read address cycle is a FRADC cycle set with the TIM register.

<Notes>

- Page read is not available in multiplex mode.
- Page read is not available in NAND Flash memory mode.

3.3. NAND Flash memory access

The following explains NAND Flash memory access.

■ Memory Access Methods

Accesses to the NAND Flash memories will be converted as shown below, based on the base address for the area set to NAND mode.

A write access to +0x2000 is converted into the issue of an address for the NAND Flash memory (MNALE is asserted).

A write access to +0x1000 is converted into the issue of a command for the NAND Flash memory (MNCLE is asserted).

A write/read access to +0x0000 will be converted to a data access to the NAND Flash memories (MNALE and MNCLE will not be asserted).
In this case, all the access timing setups is the same as the setups used by SRAM accesses.
MNCLE will be output at the same timing of address output for accessing.
MNALE will be held asserted until a write access to +0x3000 or a write access other than an address issuance (data or command) is made after the address is issued. This is because NAND Flash memories cannot de-assert the MNALE between multiple write accesses for issuing addresses. An access to +0x3000 will de-assert the MNALE only, not perform an access. Figure 3-6 shows the process of NAND Flash memory access. (For details about the commands, see the specification of NAND Flash memory connected to this family)

■ Pins Used

NAND Flash memory accesses require the pins shown in Table 3-5.

Table 3-5 External Interface Pins for NAND Flash Memories

Pin name	Function
MADATA[15:0]	Data input/output pins (The names of the data input/output pins are "MDATA[15:0]" for the TYPE0 products.)
MCSX[7:0]	Chip select pins
MNALE	Address latch enable output pin for NAND Flash memories
MNCLE	Chip enable output pin for NAND Flash memories
MNREX	Read enable output pin for NAND Flash memories
MNWEX	Write enable output pin for NAND Flash memories

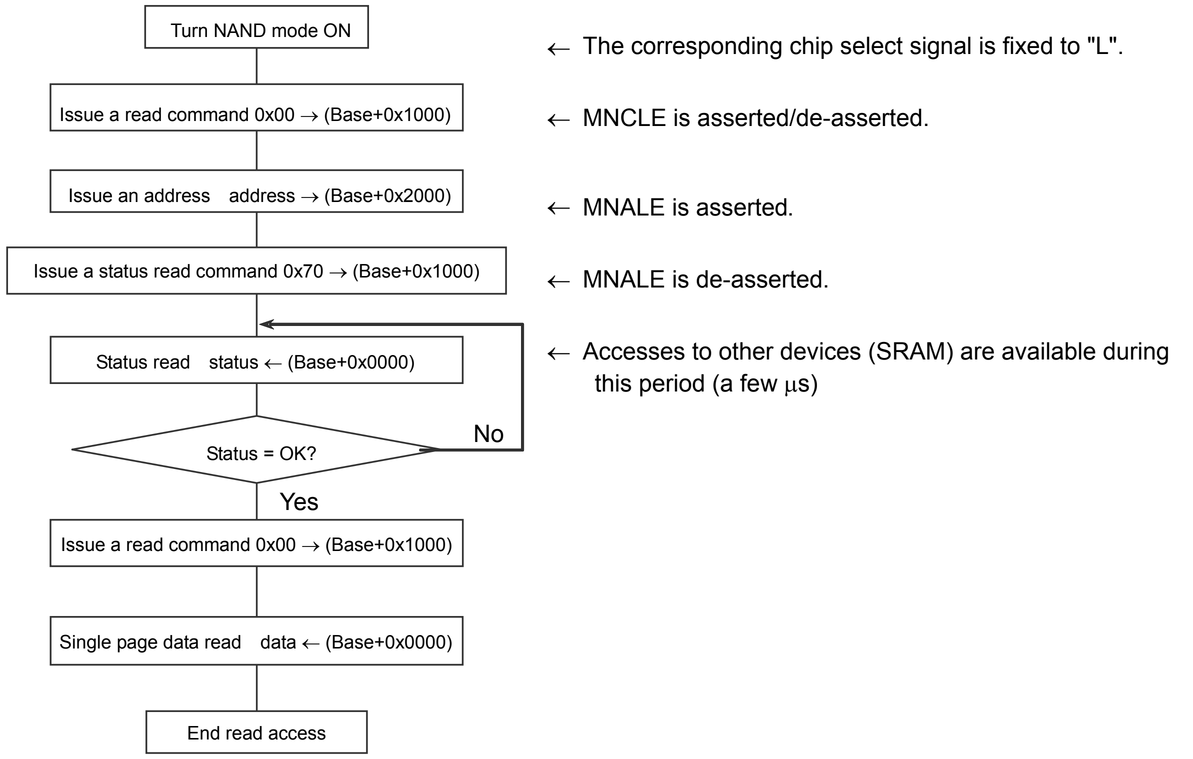
<Notes>

- Not all of the pins shown in Table 3-5 will be used depending on the setups or target devices(NAND Flash memory).
- Number and functions of external bus interface pins used depend on the product used. See the data sheets of products used for the details.
- Multiplex mode is not available for NAND Flash memory accesses.

3.3.1. Read access to NAND Flash memory

Figure 3-6 shows the flowchart of read access to NAND Flash memory.

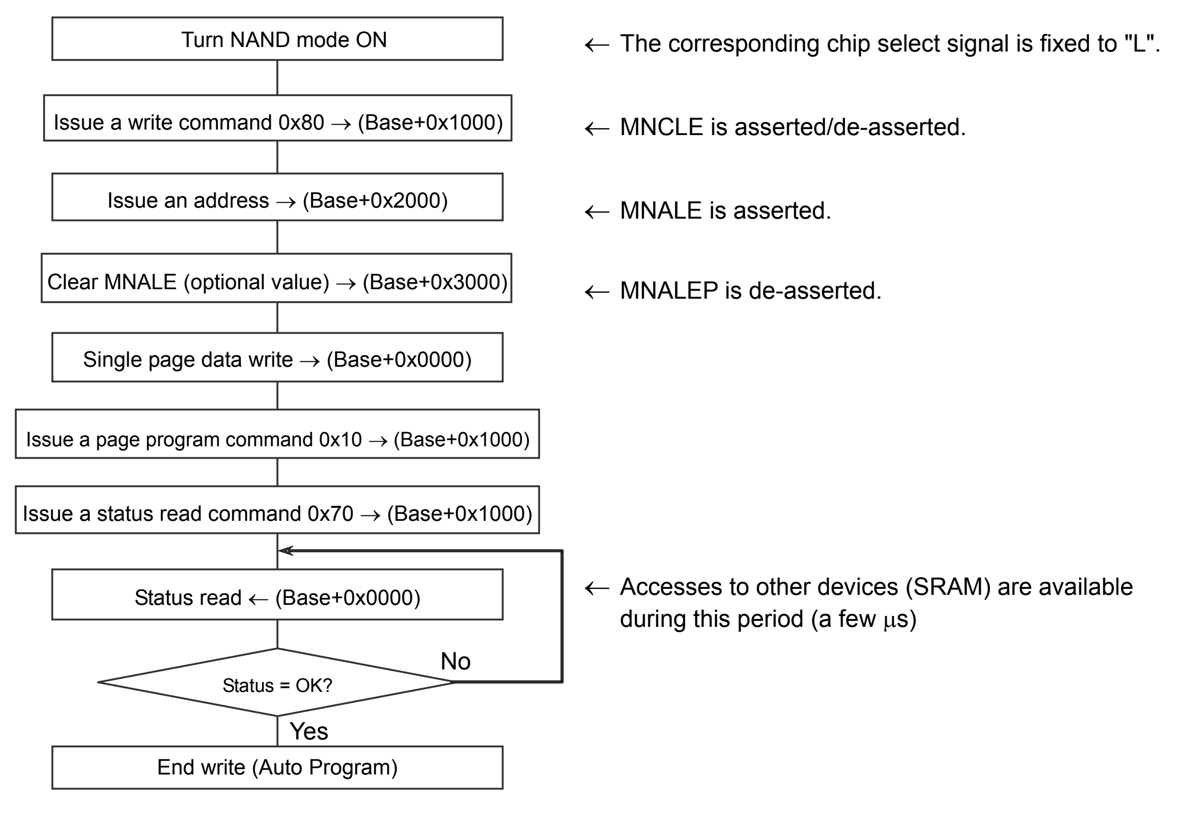
Figure 3-6 Flowchart of read access to NAND Flash memory


Turn NAND mode ON
Issue a read command 0x00 → (Base+0x1000)
Issue an address address → (Base+0x2000)
Issue a status read command 0x70 → (Base+0x1000)
Status read status ← (Base+0x0000)
Status = OK?
No
Yes
Issue a read command 0x00 → (Base+0x1000)
Single page data read data ← (Base+0x0000)
End read access
← The corresponding chip select signal is fixed to "L".
← MNCLE is asserted/de-asserted.
← MNALE is asserted.
← MNALE is de-asserted.
← Accesses to other devices (SRAM) are available during this period (a few µs)

3.3.2. Write (auto program) access

Figure 3-7 shows the flowchart of the write (auto program) access.

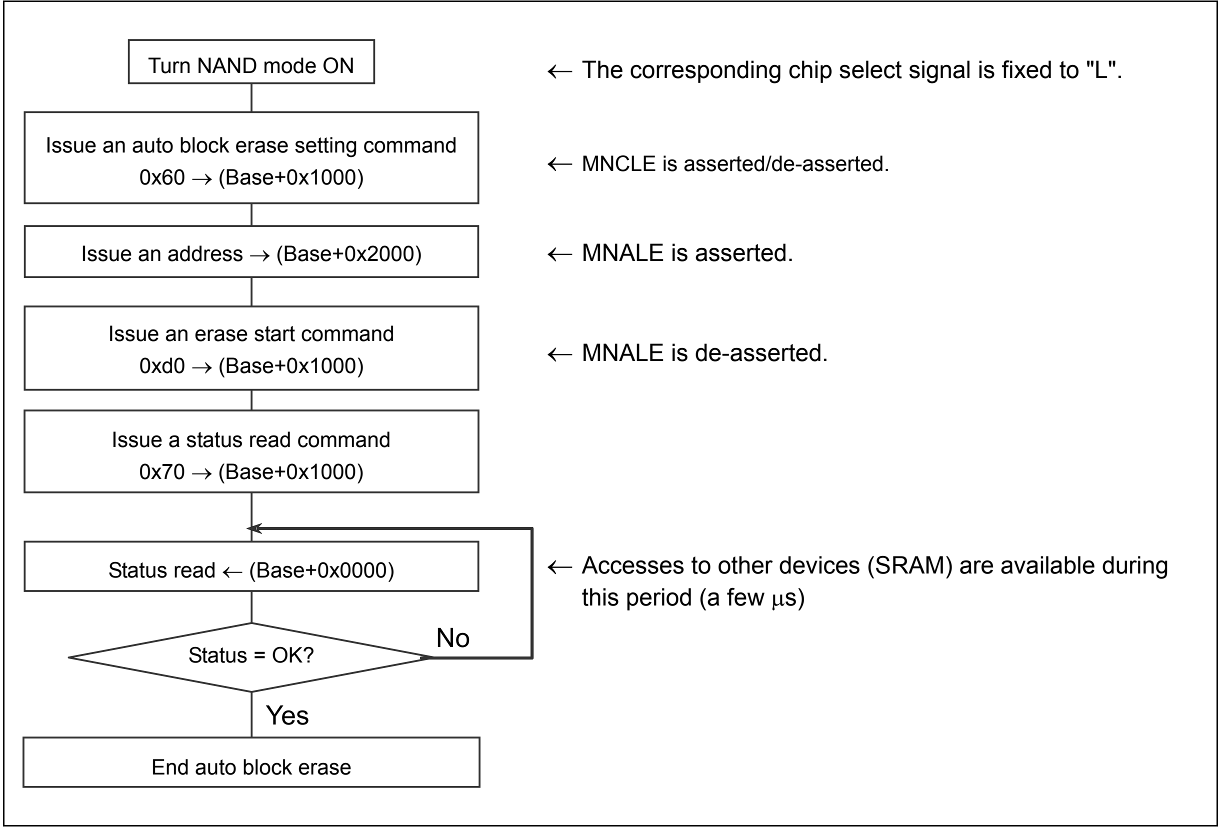
Figure 3-7 Write (auto program) access flowchart


Turn NAND mode ON
← The corresponding chip select signal is fixed to "L".
Issue a write command 0x80 → (Base+0x1000)
← MNCLE is asserted/de-asserted.
Issue an address → (Base+0x2000)
← MNALE is asserted.
Clear MNALE (optional value) → (Base+0x3000)
← MNALEP is de-asserted.
Single page data write → (Base+0x0000)
Issue a page program command 0x10 → (Base+0x1000)
Issue a status read command 0x70 → (Base+0x1000)
Status read ← (Base+0x0000)
← Accesses to other devices (SRAM) are available during this period (a few µs)
Status = OK?
No
Yes
End write (Auto Program)

3.3.3. Auto block erase access

Figure 3-8 shows the flowchart of the auto block erase access.

Figure 3-8 Auto block erase access

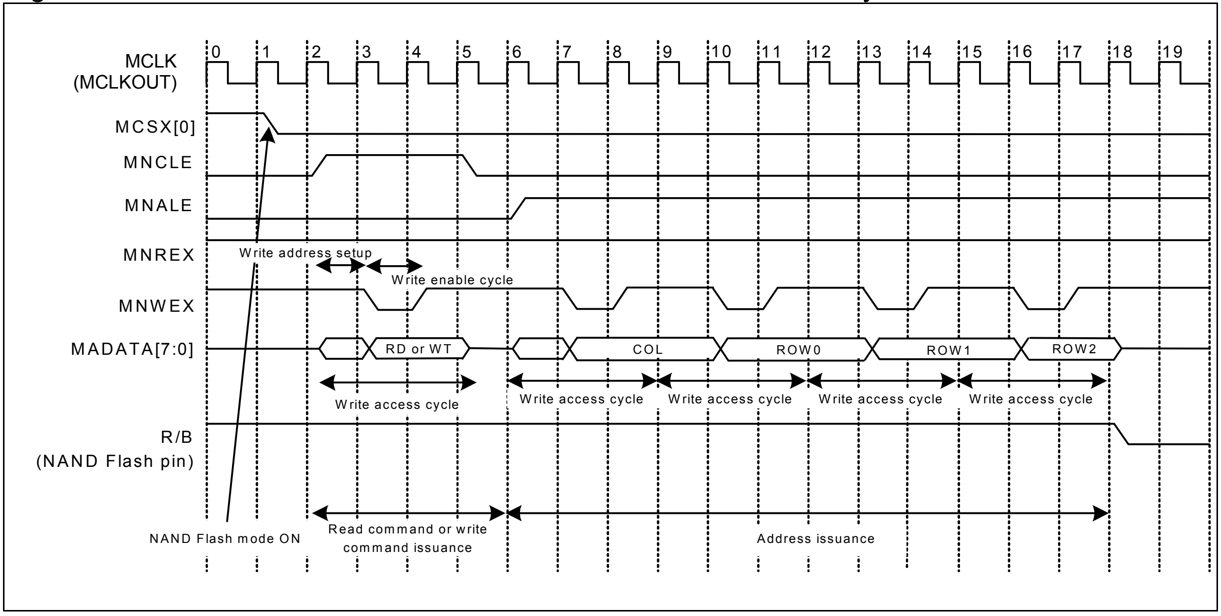

Turn NAND mode ON
← The corresponding chip select signal is fixed to "L".
Issue an auto block erase setting command
0x60 → (Base+0x1000)
← MNCLE is asserted/de-asserted.
Issue an address → (Base+0x2000)
← MNALE is asserted.
Issue an erase start command
0xd0 → (Base+0x1000)
← MNALE is de-asserted.
Issue a status read command
0x70 → (Base+0x1000)
Status read ← (Base+0x0000)
← Accesses to other devices (SRAM) are available during this period (a few µs)
Status = OK?
No
Yes
End auto block erase

As shown in the above flowchart, access to another memory device is possible even in the stage where the process of accessing NAND Flash memory has not finished. Because reading or writing data can be substituted by DMA, the processor can access the NAND Flash memory with minimum operations.

3.4. Issue of an 8-bit NAND Flash memory read/write command

Figure 3-9 shows waveforms of the issue of an 8-bit NAND Flash memory read/write command (byte access).

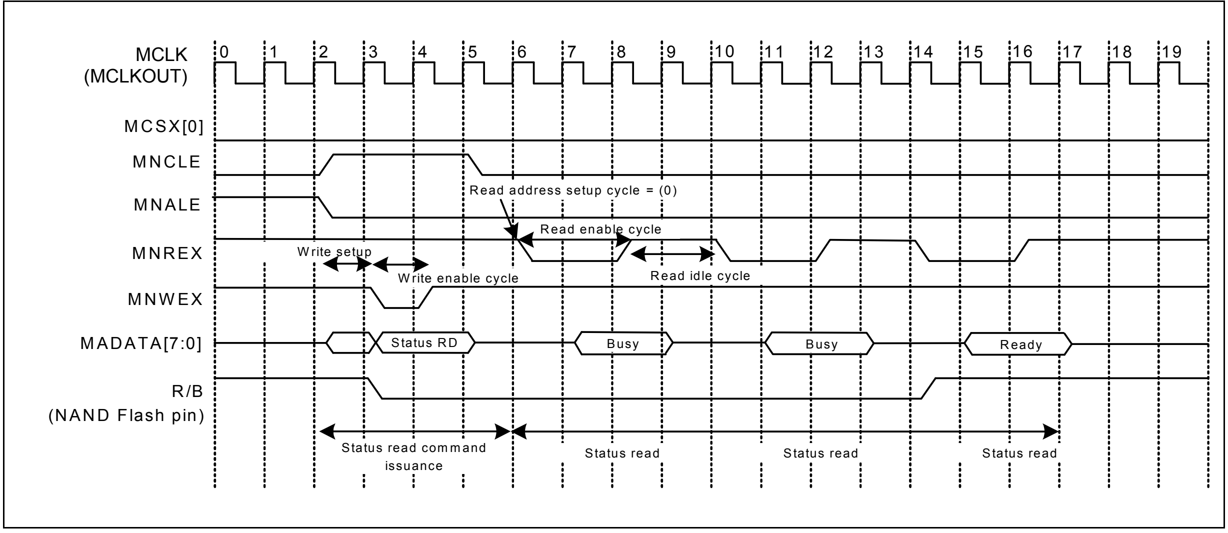
Figure 3-9 Waveforms of the issue of an 8-bit NAND Flash memory read/write command



3.5. 8-bit NAND Flash memory status read

Figure 3-10 shows waveforms of an 8-bit NAND Flash memory status read (byte access).

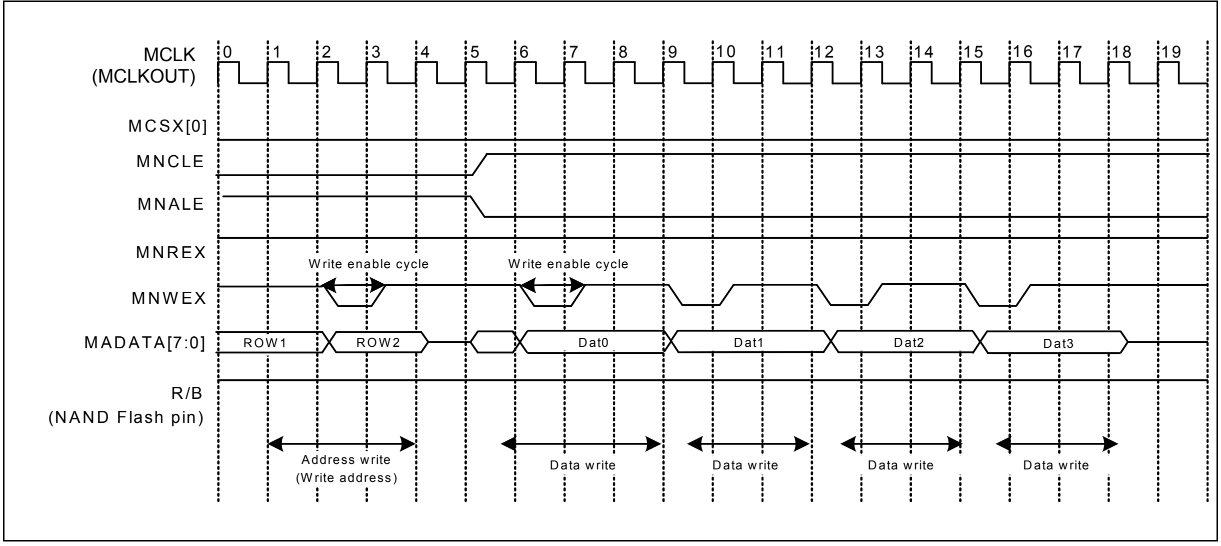
Figure 3-10 Waveforms of an 8-bit NAND Flash memory status read



3.6. 8-bit NAND Flash memory data write

Figure 3-11 shows waveforms of an 8-bit NAND Flash memory data write.

Figure 3-11 8-bit NAND Flash memory data write



3.7. Automatic Wait Setup

The following explains the automatic wait function.

The automatic wait function sets an automatic wait time per MCSX area for external accesses with the register setups. The parameters to which the automatic wait can be set are shown in Table 3-6 and Table 3-7. Figure 3-12 through Figure 3-16 show specific examples where the automatic wait is assigned.

Table 3-6 Automatic Wait Setup List

Available point	Register name	Available cycle	Remarks
Number of cycles from access start to ALE output and address output start	ATIMn:ALES	0 to 15 cycles (ALES) Cycle	Multiplex mode only
ALE output width	ATIMn:ALEW	1 to 16 cycles (ALEW+1) Cycle	Multiplex mode only
Period from access start to address output end	ATIMn:ALC	1 to 16 cycles (ALC+1) Cycle	Multiplex mode only
Number of cycles until the MOEX↓ after ALC period is ended	TIMn:RADC	0 to 15 cycles (RADC) Cycle	
CS effective period in read cycle	TIMn:RACC	1 to 16 cycles (RACC+1) Cycle	
Number of idle cycles after read	TIMn:RIDLC	1 to 16 cycles (RIDLC+1) Cycle	
Number of first read address waiting cycles at page read access	TIMn:FRADC	0 to 15 cycles	Only PAGE=1 and MOEXEUP=0
Number of cycles while MOEX is low		1 to 16 cycles (FRADC+1) Cycle	Only PAGE=0 and MOEXEUP=1
Number of cycles until the MWEX↓ after ALC period is ended	TIMn:WADC	1 to 15 cycles (WADC+1) Cycle	
Number of cycles while MWEX is low	TIMn:WWEC	1 to 15 cycles (WWEC+1) Cycle	
CS effective period in write cycle	TIMn:WACC	3 to 16 cycles (WACC+1) Cycle	
Number of idle cycles after write	TIMn:WIDLC	1 to 16 cycles (WIDLC+1) Cycle	

*: Number of cycles is counted based on MCLKOUT.

Table 3-7 Automatic Wait Setup List (SDRAM)

Available point	Register name	Available cycle
Number of CAS latency cycles	SDTIM:CL	1 to 3 cycles (CL+1) Cycle
Number of latency cycles between RASs	SDTIM:TRC	1 to 8 cycles (TRC+1) Cycle
Number of cycles in precharge period	SDTIM:TRP	1 to 4 cycles (TRP+1) Cycles
Number of latency cycles between RAS and CAS	SDTRM:TRCD	1 to 2 cycles (TRCD+1) Cycles
Number of cycles in minimum active period of Row	SDTIM:TRAS	1 to 8 cycles (TRAS+1) Cycle
Number of latency cycles of the command succeeding to refresh	SDTIM:TREFC	1 to 8 cycles (TREF+1) Cycles
Number of latency cycles from write to precharge	SDTIM:TDPL	1 to 4 cycles (TDPL+1) Cycles

*: Number of cycles is counted based on MSDCLK.

Figure 3-12 Chart Description for Automatic Wait Assignment (Separate Mode)

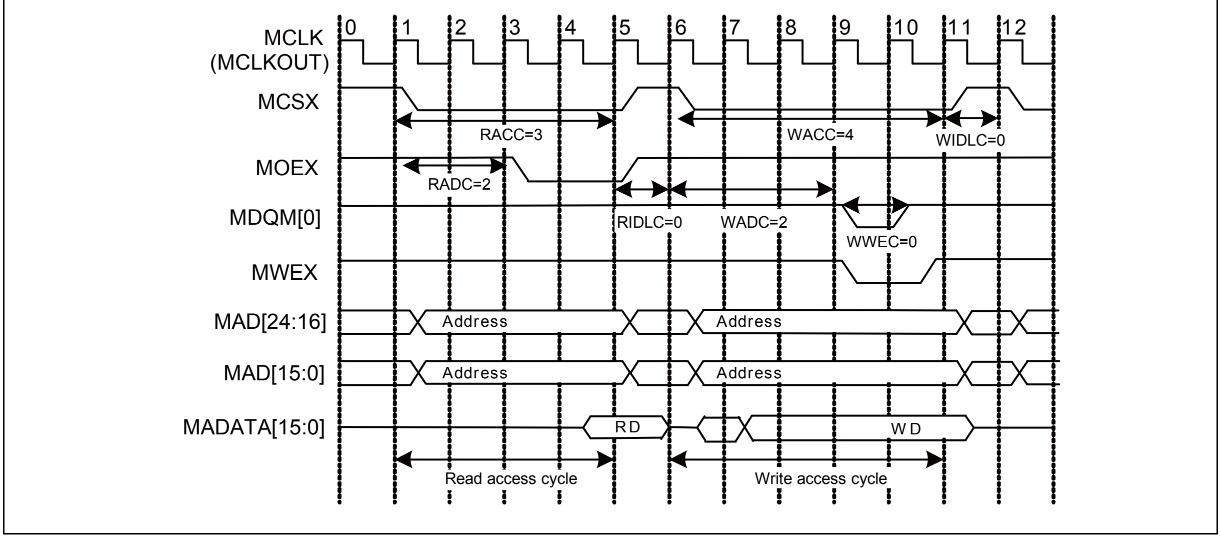


Figure 3-13 Chart Description for Automatic Wait Assignment (Multiplex Mode)

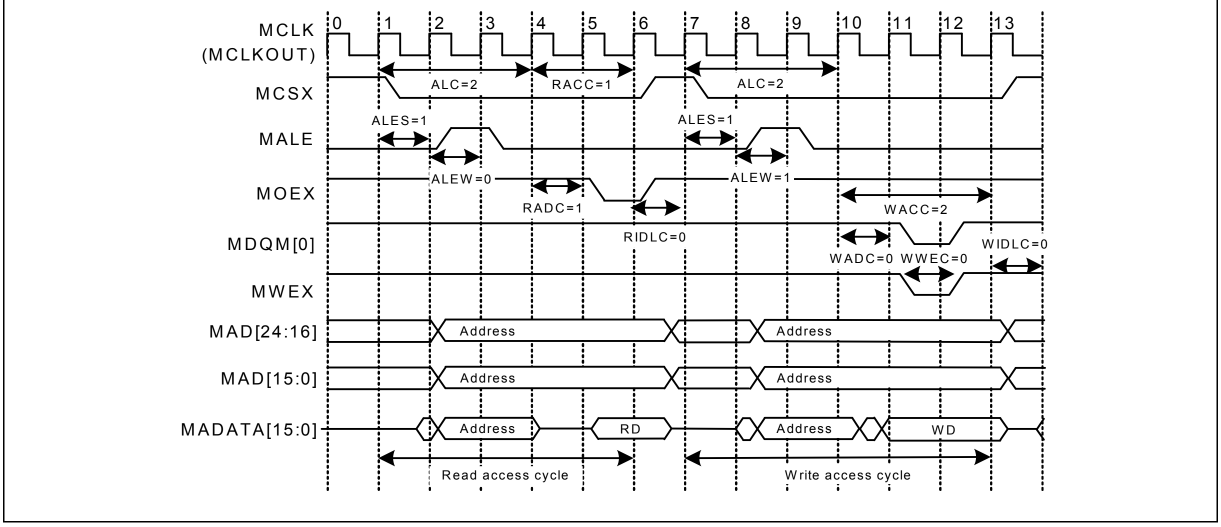


Figure 3-14 Chart Description for Automatic Wait Assignment (NAND Flash Memory Mode)

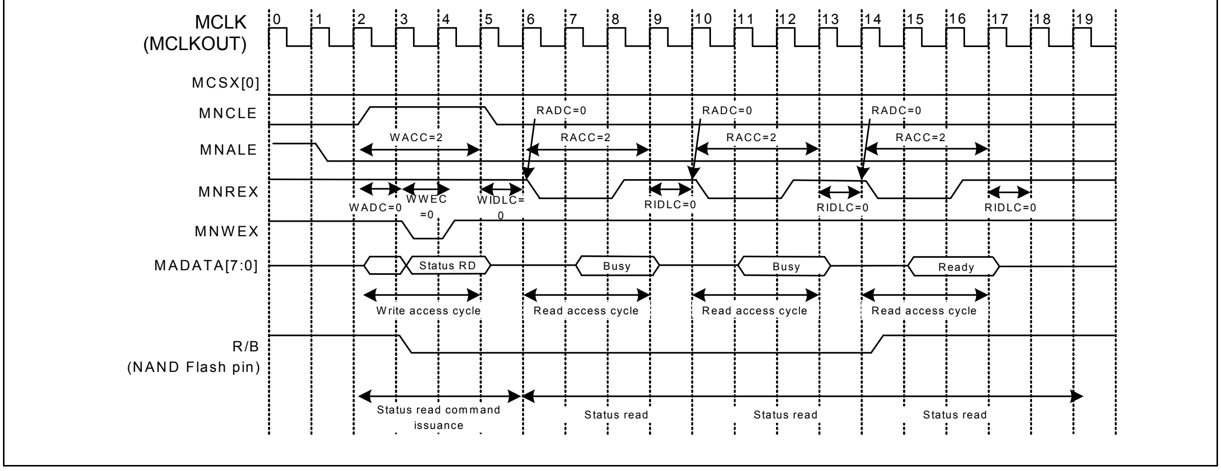


Figure 3-15 Chart Description for Automatic Wait Assignment
(NOR Flash Memory Page Read)

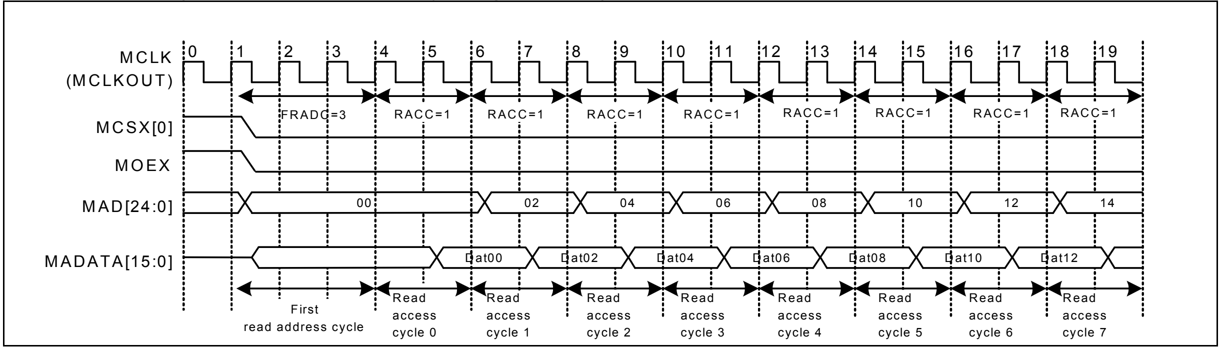


Figure 3-16 Chart Description for Automatic Wait Assignment (SRAM Continuous Read)

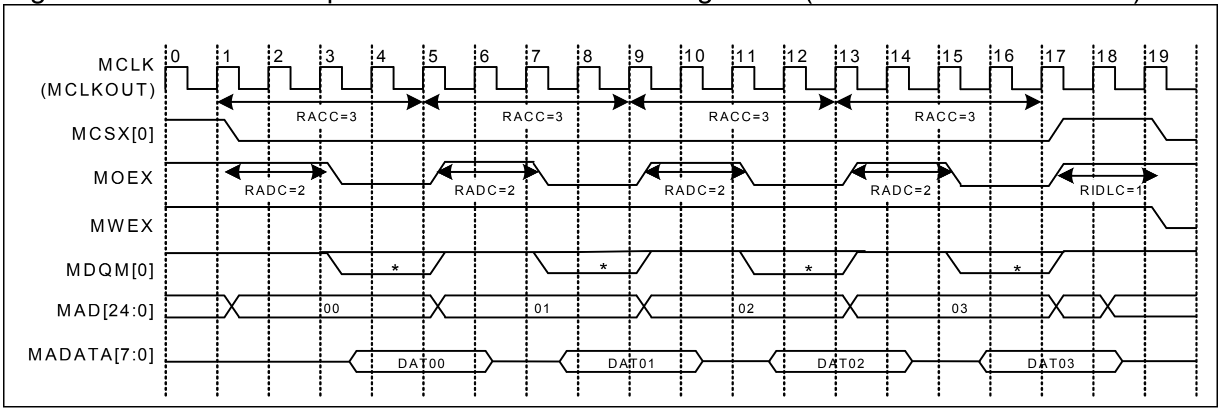
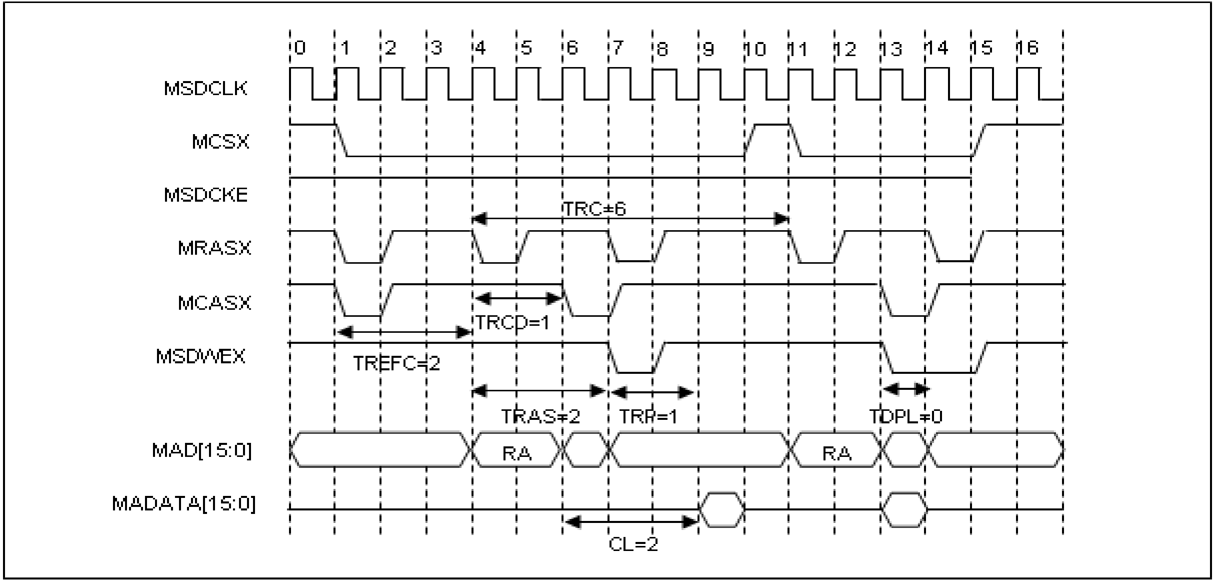


Figure 3-17 Chart Description of Automatic Wait Assignment (SDRAM)



<Notes>

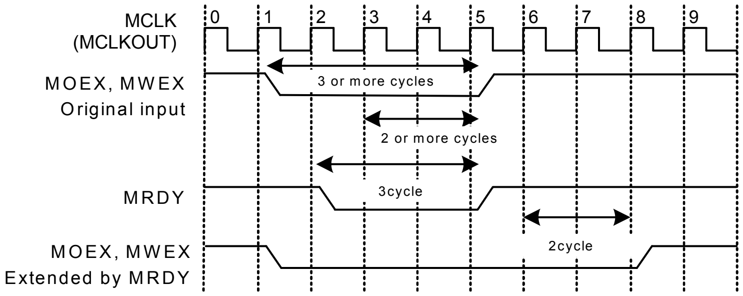
- The automatic wait setup will be enabled for all the access modes of external bus interface except for ATIMn:ALES, ALC and ALEW. The setups for the read data and write data must meet the setup and hold specifications.
- One or more cycles must be provided for address hold cycle while in asynchronous accessing.

3.8. External RDY

The following explains the external RDY function.

This function allows extended access cycle inserting wait cycles while "L" level is input to the MRDY pin. This enables accesses to the low speed external memories. The operation waveform of the RDY signal is shown in Figure 3-.

Figure 3-17 Operation of External RDY



<Notes>

- When you use the external RDY function, set the widths for MOEX and MWEX to 3 or more cycles.
- In order to enable RDY function, MRDY=L must be input 2 cycles before MOEX↑/MWEX↑. The MRDY=L less than 2 cycles will be ignored.
  See "Data Sheet" of product used for details of AC specifications.
- MOEX↑/MWEX↑ will be performed 2 cycles after MRDY=H.
- Do not set MRDY=L until MOEX↑/MWEX↑ when MRDY=H is set once.
- The external RDY function will not be available in NOR Flash memory page read and NAND Flash memory modes.
- The periods of ALC and ALE will not be extended even if MRDY=L is set.

3.9. SDRAM Access

This section explains SDRAM access.

■ Memory Access

In SDRAM access, MCSX[8] address output determines the target device. Then, by outputting MRASX/ MCASX/MSDWEX/MSDCKE , read/write operation is executed to the target device.

■ Pins Used

For SDRAM access, use the pins in Table 3-8.

Table 3-8 External Interface Pins for SDRAM Memory

Pin names	Functions
MAD[15:0]	Address output pins
MADATA[15:0]	Data input/output pins
MCSX[8]	Chip selection pins
MDQM[1:0]	Byte mask signal output pins
MRASX	Row address strobe output pin
MCASX	Column address strobe output pin
MSDWEX	Write enable output pin
MSDCKE	Clock enable output pin
MSDCLK	Clock output pin

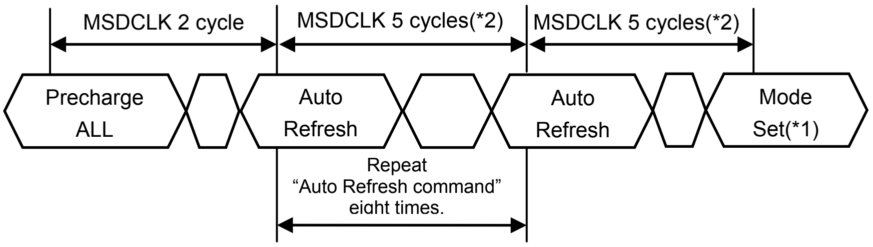
<Notes>

- Depending on the settings and the target device,(NAND Flash memory), all pins in Table 3-8 are not used.
- The external bus interface pins that appear are different by the product. For details, see “Data Sheet” of the product used.
- For SDRAM access, Multiplex mode is not available.

■ Power-on Sequence

By changing SDON bit of SDRAM mode register (SDMODE) from “0” to “1”, the power-on sequence is issued and the access to SDRAM is enabled. For the power-on sequence issued, see Figure 3-19. In Figure 3-19, “cycle “ is based on MSDCLK.

Figure 3-19 Power-on Sequence Operation



(*1) : The setting of Mode Set are as follows :
BL=1, BT=Seq, CL=SDTIM.CL Register Value, Normal Operation , Programmed Write Burst

(*2) : This is the initial cycle count of TREFC .
Before asserting SDON bit, the cycle count can be set with TREFC register.

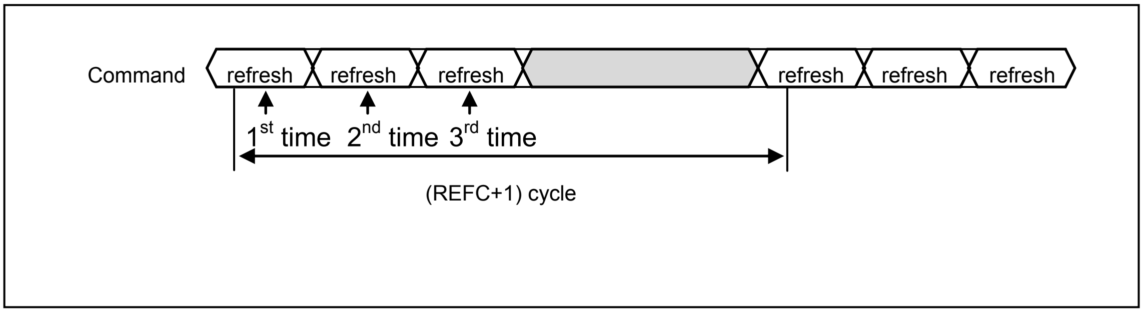
■ Refresh Operation

When ROFF bit of SDRAM mode register (SDMODE) is “0”, the refresh is executed with the setting conditions of Refresh time register (REFTIM). The Refresh time register (REFTIME) can control the interval and count of the refresh and the preceding refresh.

■ Refresh Interval

Set the interval to execute refreshes with REFC bit of the Refresh time register (REFTIM). The refresh is implemented after the set cycle count (MSDCLK) has elapsed. Moreover, set the refresh issuing count in one refresh timing with NREF bit. The following shows the operation example at the setting of NREF=2. In Figure 3-20, “cycle “ is based on MSDCLK.

Figure 3-20 Refresh Intervals and Issuing Count

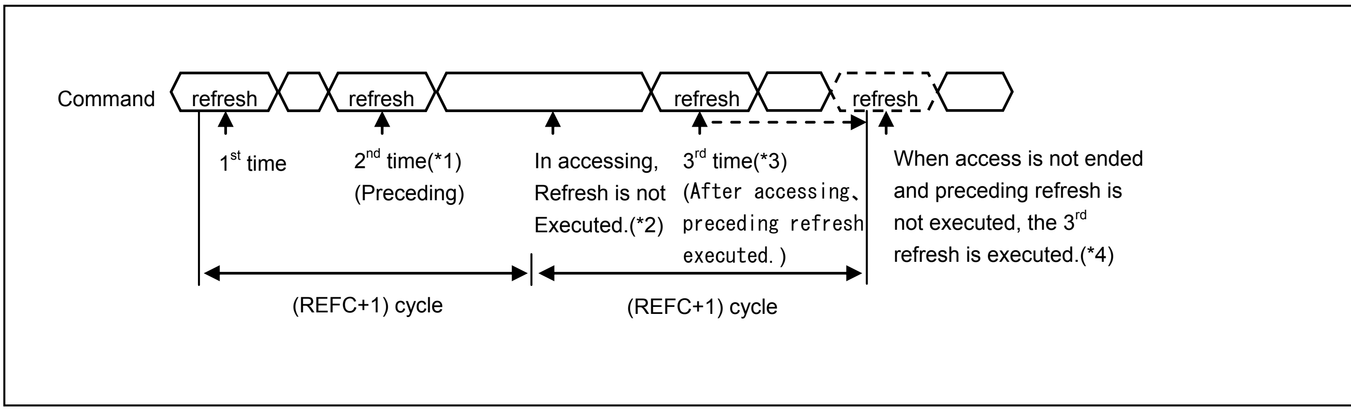


■ Preceding Refresh

Set the preceding refresh with PREF bit of the Refresh time register (REFTIM). The preceding refresh is the refresh executed earlier than the actual refresh.
By using the preceding refresh, the access to SDRAM is improved.
For the operation of the preceding refresh, see Figure 3-21. In Figure 3-21, “cycle “ is based on MSDCLK.

- Refreshing is previously executed when SDRAM is not accessed after refreshing is implemented. (*1 in Figure 3- 21)
- After the preceding refresh is executed, if the access to SDRAM is executed in the next refresh timing, the refresh is not executed. (*2 in Figure 3- 21)
- After the access to SDRAM is completed, the preceding refresh is implemented again. (*3 in Figure 3- 21)
- When the access to SDRAM is not completed, the refresh is forcibly executed in the third refresh timing. (*4 in Figure 3- 21)

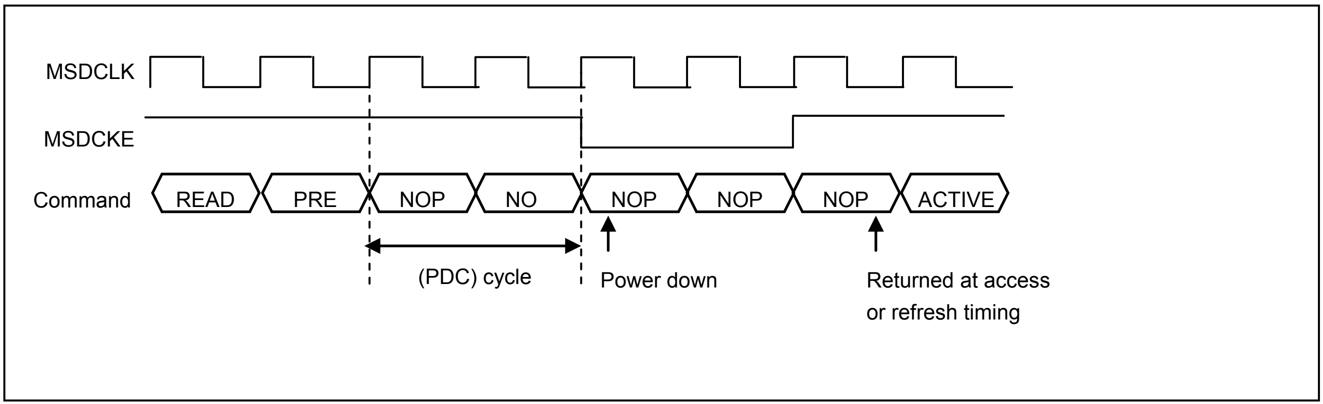
Figure 3-21 Preceding Refresh Operation



■ Power Down Mode Operation

When PDON bit of SDRAM mode register (SDMODE) is set to “1”, the mode is transferred to Power down mode if the access to SDRAM is not executed in the cycle (MADCLK) specified by Power down count register (PWRDWN). In the refresh timing, the mode is returned again and the refresh is executed. If the access is not executed in the period specified by Power down count register (PWRDWN) after refreshing, the mode is transferred to the Power down mode again. To return, one cycle is required. When PDON bit is set to “1” while SPON bit of SDRAM mode register (SDMODE) is “0”, MSDCKE is set to”L”.

Figure 3-22 Power Down Mode Operation



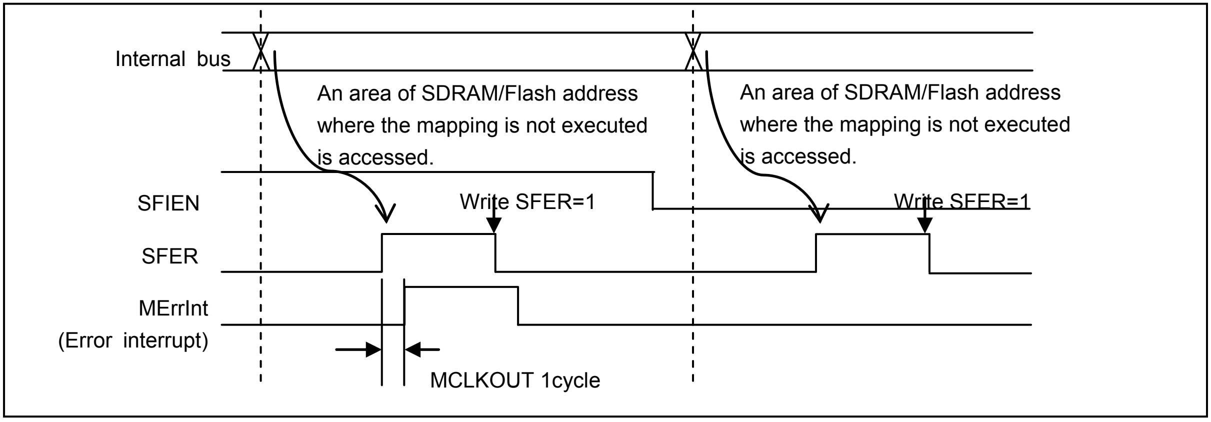
3.10. Interrupt Function

This section explains the error interrupt.

■ SRAM/Flash memory error

When the access is executed by SRAM/Flash memory error interrupt enable (MEMCERR.SFION=1) to SRAM/Flash memory address area where the mapping is not implemented with the area register, an error interrupt occurs. By writing"1" to SRAM/Flash memory error (MEMCERR.SFER), the interrupt is cleared.

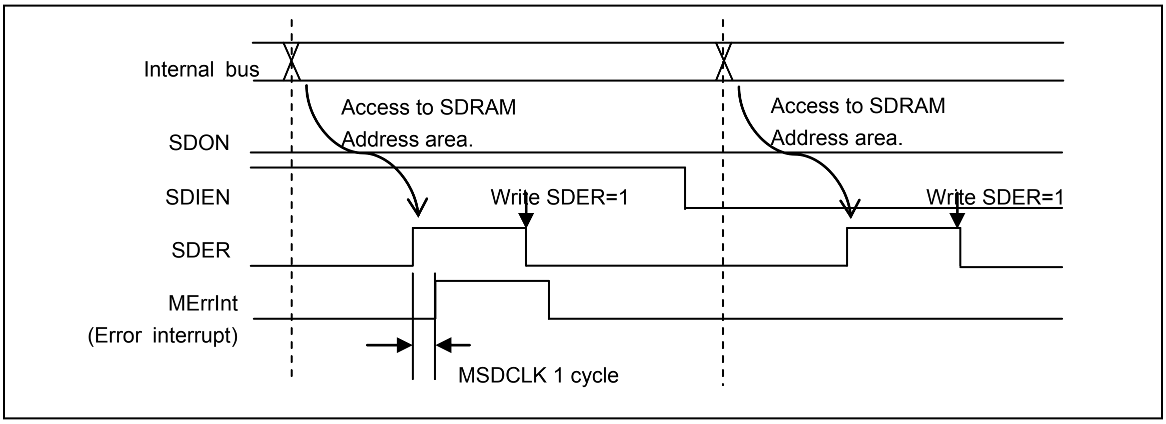
Figure 3-23 SRAM/Flash Memory Error Interrupt


Internal bus
An area of SDRAM/Flash address where the mapping is not executed is accessed.
An area of SDRAM/Flash address where the mapping is not executed is accessed.
SFIEN
Write SFER=1
Write SFER=1
SFER
MErrInt
(Error interrupt)
MCLKOUT 1cycle

■ SDRAM Error

When the access is executed by SDRAM interrupt enable (MEMCERR.SDION=1) to SDRAM address area, if SDON of SDRAM mode register (SDMODE) is "0", an error interrupt occurs. By writing"1" to SDRAM error (MEMCERR.SDER), the interrupt is cleared

Figure 3-24 SDRAM Error Interrupt



3.11. Access Mode

This section explains the access mode.

For the read/write access to the external bus interface address area, the following functions area provided to improve the access efficiency:

1. At read access: Preceding access

2. At write access: Continuous access

The above functions are available only when the frequency division ratio of Division clock register (DCLKR) is two or more. At the frequency division ratio of "1", the normal access is implemented. Moreover, the above functions are available at the initial state. So, the functions can be stopped with common setting register (AMODE: WAEN).

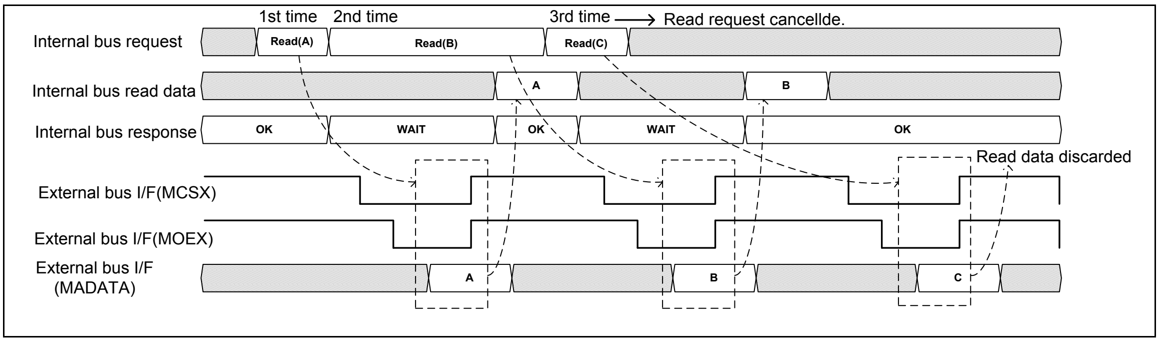
■ Read Access

To improve the efficiency of the read access of the external bus interface, the preceding read of the next read access is provided.

At the setting where the preceding access is available, the preceding access of which transfer is not cancelled may be executed to the external bus interface. In such case, the read value is disposed of and it does not affect the CPU operation.
For the operation example of preceding access, see Figure 3-25. The read access to the external memory is executed before accepting the second read request from the internal bus and after the accepting the request, the read data is returned to the CPU. At the third time, the preceding access is executed in the same way, the read data is disposed of because the read request is cancelled at accepting the access request.

However, when the above mentioned preceding access is unavailable due to the specifications of the external device, set the preceding function ineffective with register settings (AMODE:WAEN).

Figure 3-25 Operation Example of Preceding Read Access Occurred



■ Write Access

To reduce the write access interval, the mode for implementing the continuous write request is provided.

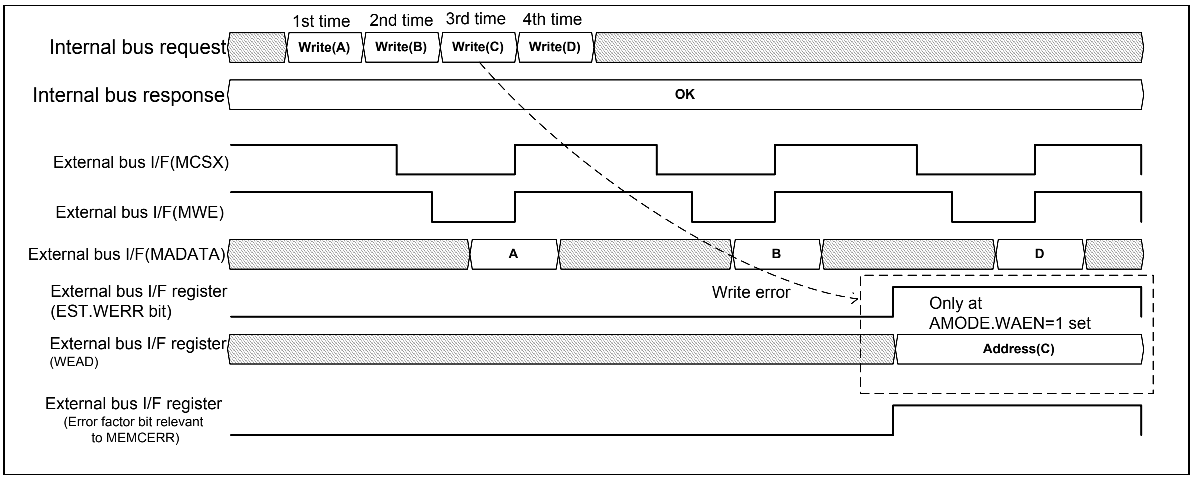
For the example of continuous write operation, see Figure 3-26. In this access, an error occurs in the 3rd write request. But, the 4th write access is accepted without waiting for an error response. When an error response occurs, the address of an error response factor is saved in the factor register (WEAD). The error response is not notified to CPU.

To stop the CPU immediately after an error response from the external bus interface is received, make the setting to disable the continuous access function with the register setting (AMODE:WAEN). However, an interrupt due to error response can be detected in the case where the continuous access function is enabled.

*: Condition of error response occurring

- When an access to an area where the mapping is not executed with area register is implemented during accessing to SRAM/Flash memory address area.
- When an access to SRAM address area is implemented at SDMODE.SDON=1 during accessing to ADRAM area.

Figure 3-26 Operation Example of Error Response Occurred in Continuous Write



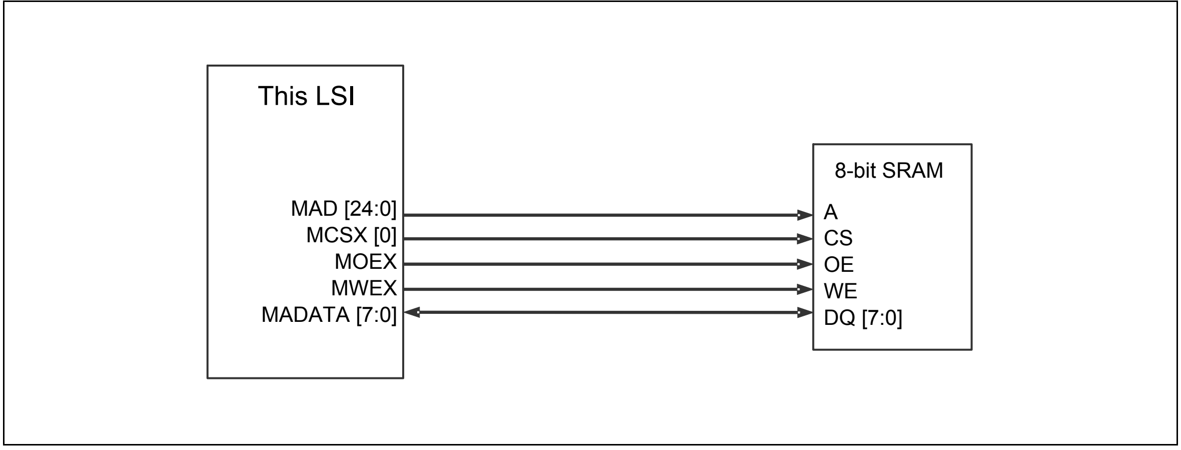
4. Connection Examples

This section provides an example of connections with external devices.

■ 8-bit SRAM

Figure 4-1 shows an example of connecting an 8-bit SRAM.

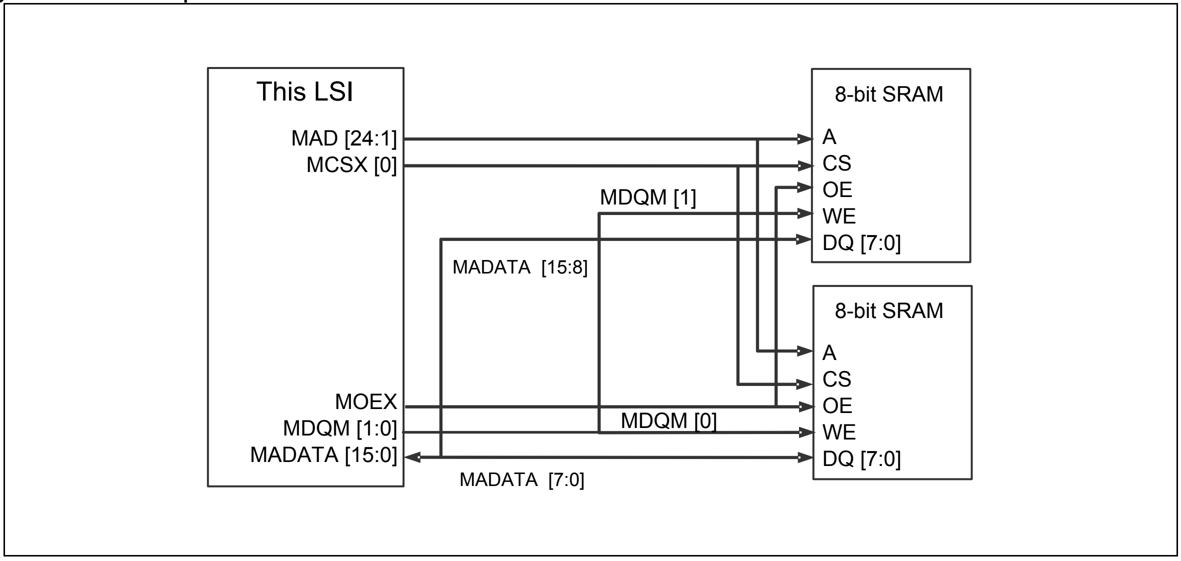
Figure 4-1 Example of 8-bit SRAM connection



■ 8-bit SRAM × 2

Figure 4-2 shows an example of connecting two 8-bit SRAMs.

Figure 4-2 Example of 8-bit SRAMx2 connection

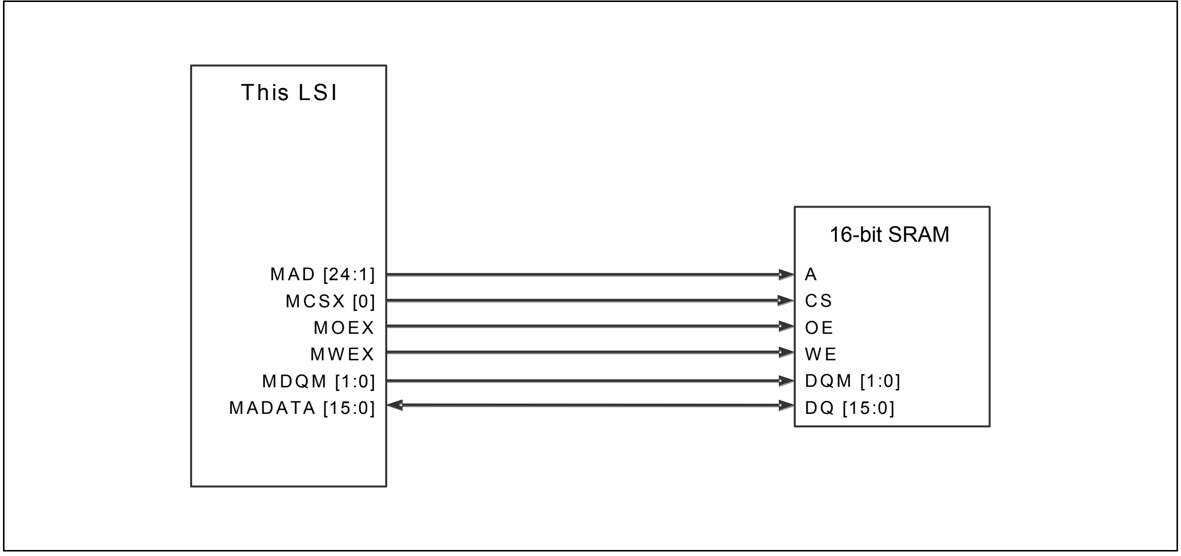


The MDQM signal can be used as a write enable for the devices without input mask feature.

■ 16-bit SRAM

Figure 4-3 shows an example of connecting a 16-bit SRAM.

Figure 4-3 Example of 16-bit SRAM connection

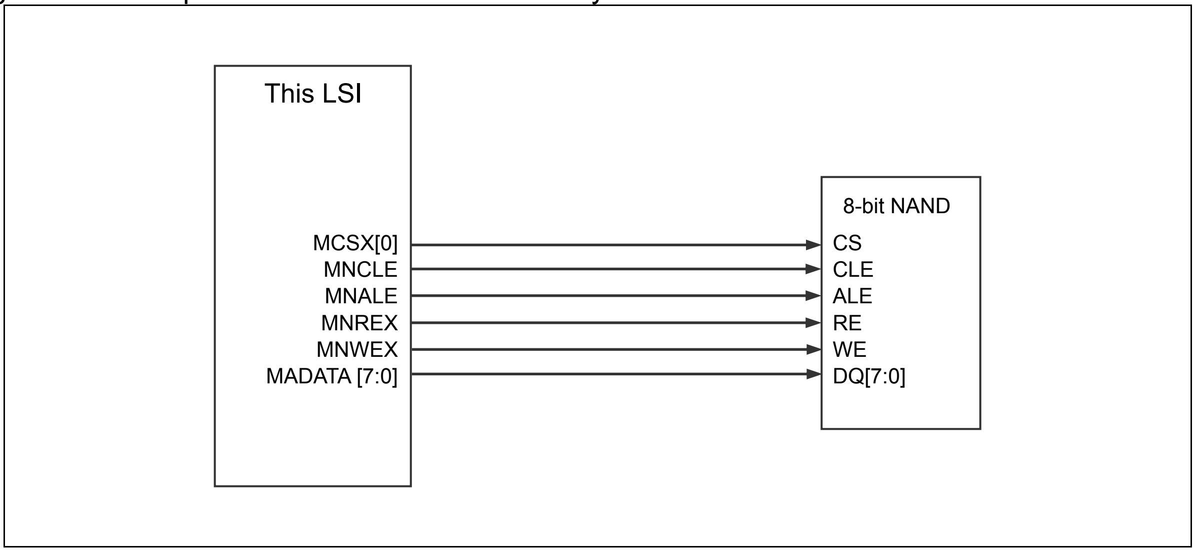


In the case where a target device employs a byte mask signal, using the MDQM control will read the data only required for the device. This resulted in reducing the power consumption while accessing.

■ 8-bit NAND

Figure 4-4 shows an example of connecting an 8-bit NAND Flash memory.

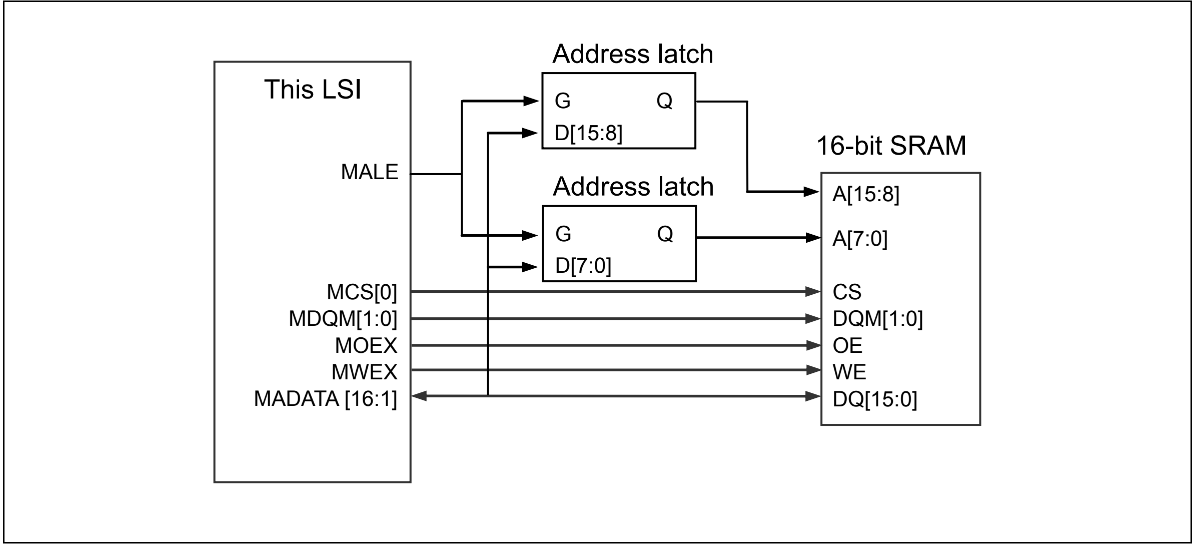
Figure 4-4 Example of 8-bit NAND Flash memory connection



■ 16-bit SRAM Multiplex Mode

Figure 4-5 shows an example of 16-bit SRAM connection for multiplex mode.

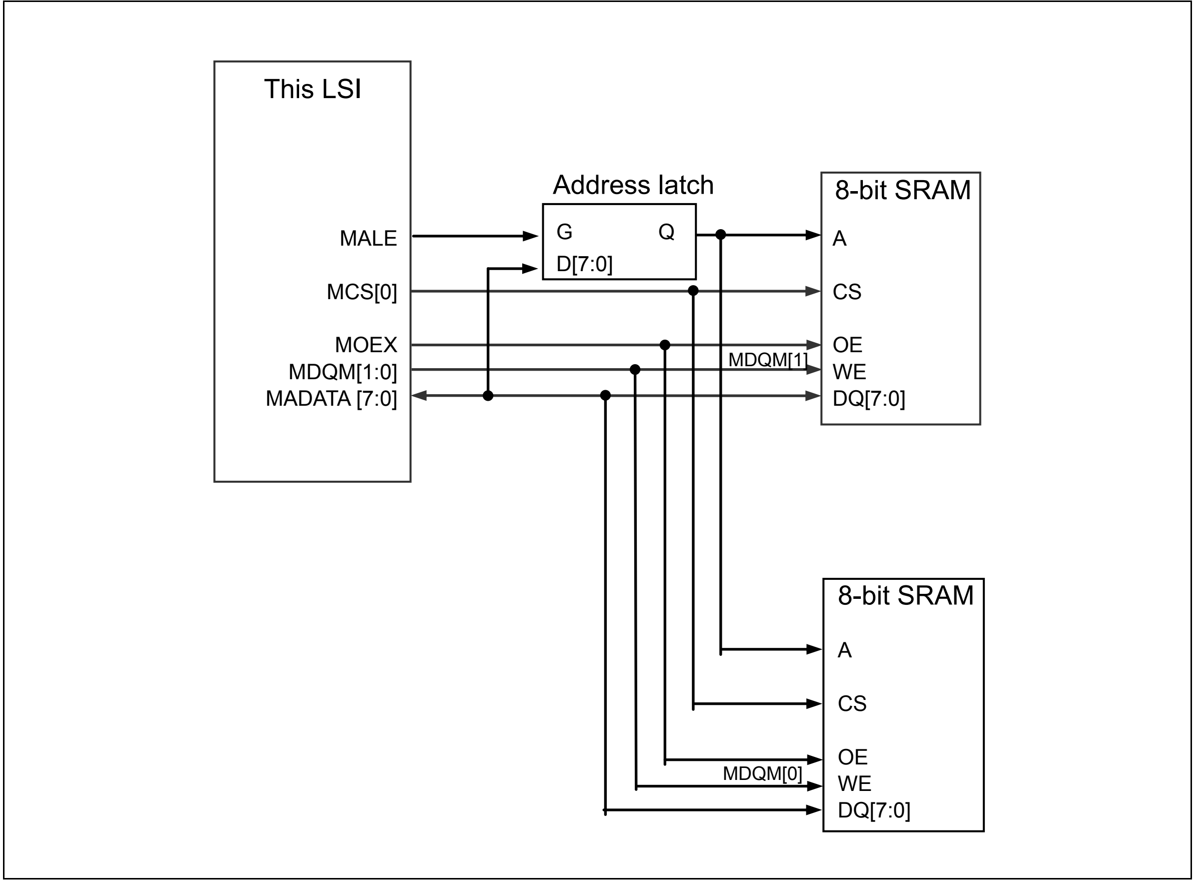
Figure 4-5 Example of 16-bit SRAM Multiplex Connection



■ 8-bit SRAM Multiplex Mode

Figure 4-6 shows an example of 8-bit SRAM connection for multiplex mode.

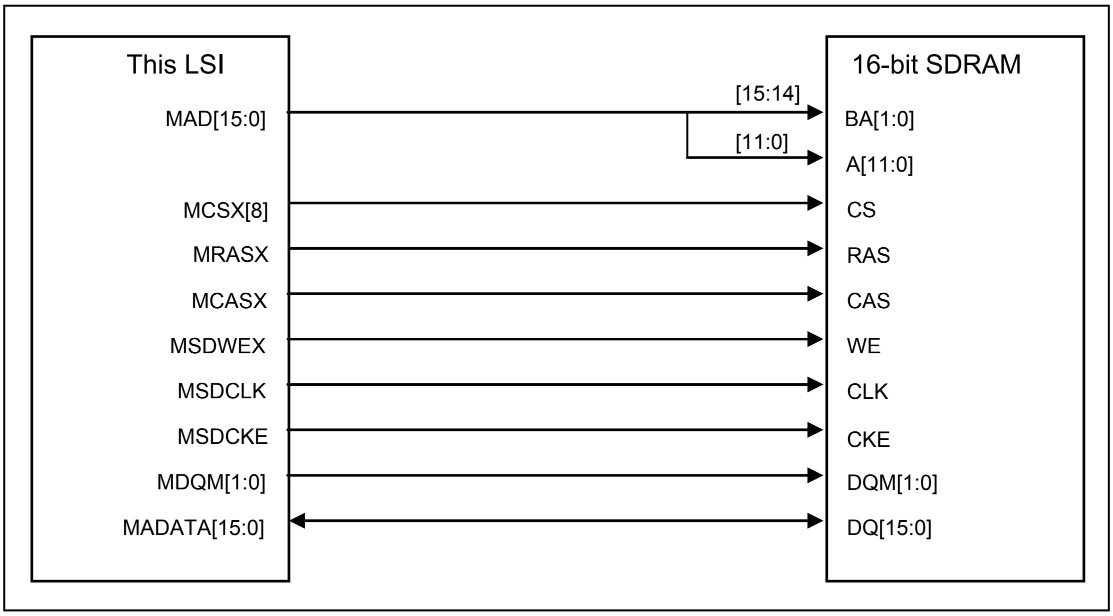
Figure 4-6 Example of 8-bit SRAM Multiplex Connection



■ 16-bit SDRAM

Figure 4-7 shows an example of connecting a 16-bit SDRAM.

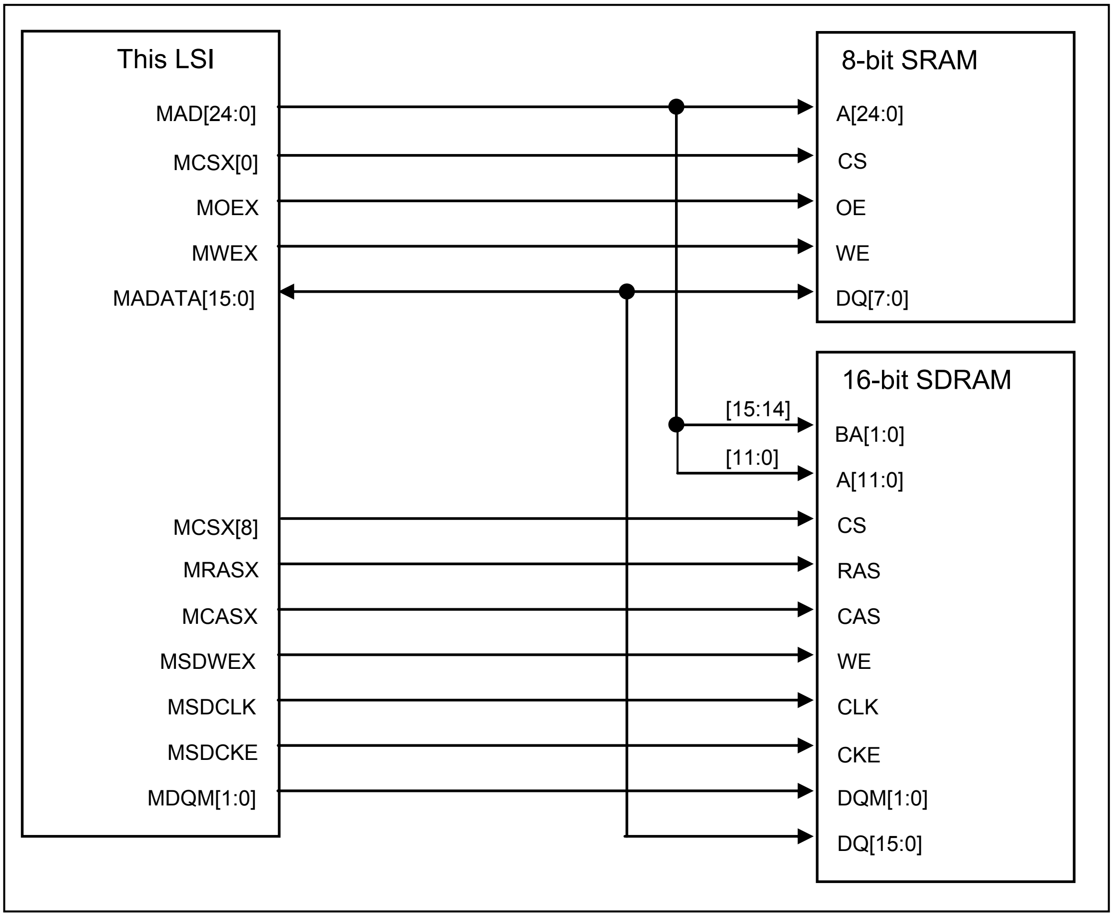
Figure 4-7 Example of 16-bit SDRAM Connection



■ 8-bit SRAM, 16-bit SDRAM

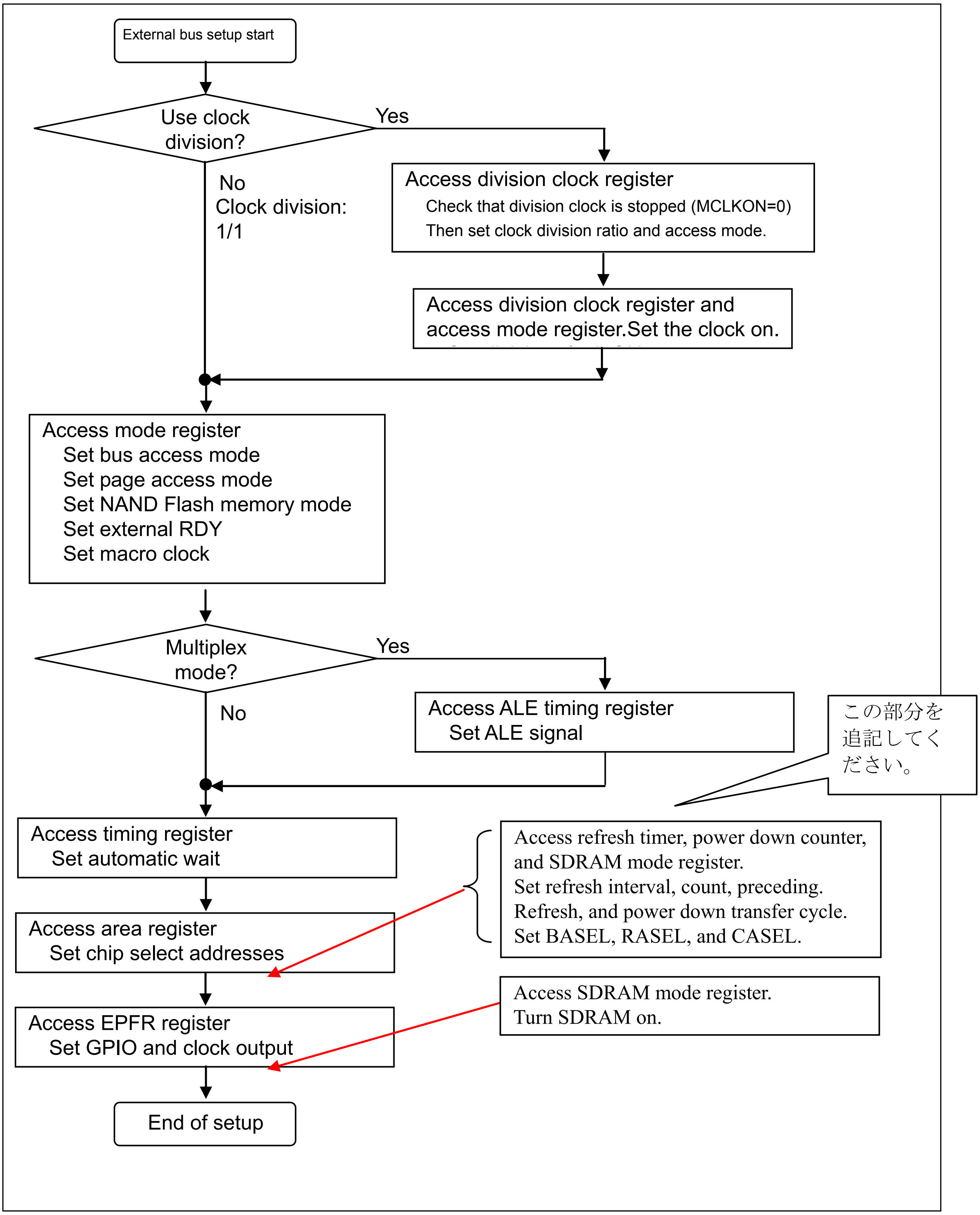
Figure 4-8 shows an example of connecting 8-bit SRAM and an 16-bit SDRAM.

Figure 4-8 Example of Connecting 8-bit SRAM and 16-bit SDRAM



5. Setup Procedure Example

The following explains an example of the external bus interface setup procedure.


External bus setup start
Use clock division?
Yes
No
Clock division: 1/1
Access division clock register
Check that division clock is stopped (MCLKON=0)
Then set clock division ratio and access mode.
Access division clock register and access mode register.Set the clock on.
Access mode register
Set bus access mode
Set page access mode
Set NAND Flash memory mode
Set external RDY
Set macro clock
Multiplex mode?
Yes
No
Access ALE timing register
Set ALE signal
この部分を追記してください。
Access timing register
Set automatic wait
Access refresh timer, power down counter, and SDRAM mode register.
Set refresh interval, count, preceding.
Refresh, and power down transfer cycle.
Set BASEL, RASEL, and CASEL.
Access area register
Set chip select addresses
Access SDRAM mode register.
Turn SDRAM on.
Access EPFR register
Set GPIO and clock output
End of setup

<Notes>

- Make sure to set up the division clock while the division clock is stopped.
- Some of setup combinations in the mode register cannot be used at the same time.

6. Registers

This section explains the configuration and functions of registers used for the external bus interface.

The following explains the registers used for the external bus interface. The bit width of every register is 32. Each register can be accessed by the APB interface with 32-bit width (word). Write "0" to reserved areas.

These registers can be rewritten while in external accessing. Actual setup value reflection will be made after current access is complete. Reading register before this reflection will read the previous setup value.

Table 6-1 lists the registers.

Table 6-1 Register list

Abbreviation	Register name	Reference
MODE0 to MODE7	Mode Register 0 to Mode Register 7	6.1
TIM0 to TIM7	Timing Register 0 to Timing Register 7	6.2
AREA0 to AREA7	Area Register 0 to Area Register 7	6.3
ATIM0 to ATIM7	ALE Timing Register 0 to 7	6.4
SDMODE	SDRAM Mode Register	6.5
REFTIM	Refresh Time Register	6.6
PWRDWN	Power Down Count Register	6.7
SDTIM	SDRAM Timing Register	6.8
SDCMD	SDRAM Command Register	6.9
MEMCERR	Memory Controller Error Register	6.10
DCLKR	Division Clock Register	6.11
EST	Error Status Register	6.12
WEAD	Write Error Address Register	6.13
ESCLR	Error Status Clear Register	6.14
AMODE	Access Mode Register	6.15

6.1. Mode 0 Register to Mode 7 Register (MODE0 to MODE7)

The MODE0 to MODE7.registers set the operation mode of SRAM/Flash memory access.

bit	31	30	29	28	27	26	25	24
Field	Reserved							
Attribute	-							
Initial value	-							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
Attribute	-							
Initial value	-							

bit	15	14	13	12	11	10	9	8
Field	Reserved		MOEXEUP	MPXCSOF	MPXDOFF	Reserved	ALEINV	MPXMODE
Attribute	-		R/W	R/W	R/W	-	R/W	R/W
Initial value	-		0	0	0	-	0	0

bit	7	6	5	4	3	2	1	0
Field	SHRTDOUT	RDY	PAGE	NAND	WEOFF	RBMON	WDTH	
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial value	0	0	0	0	0	0	00*	

*: The initial value of these bits is "01" only for MODE4 register.

[bit31:14] Reserved: Reserved bits

The read value is undefined.
Set this bit to "0" when writing.

[bit13] MOEXEUP: MOEX EUP

This bit is used to select how to set the MOEX width.

bit	Description
0	MOEX width is set with RACC-RADC (Initial value).
1	MOEX width is set with FRADC.*

* This function cannot be used combined with the page read function.
(Note) See "■Reference Information" for specific operations.

[bit12] MPXCSOF: MPX CS OF

This bit is used to select a CS assertion from the start of accessing to the end of address output (ALC cycle period) in multiplex mode.

bit	Description
0	Asserts MCSX in ALC cycle period (Initial value).
1	Does not assert MCSX in ALC cycle period.

(Note) See "■Reference Information" for specific operations.

[bit11] MPXDOFF: MPX Address Data OFF

This bit is used to select whether or not the address is output to the data lines in multiplex mode.
(The address is used from the MAD outputs but set this bit if you use the ALE signal.)

bit	Description
0	Outputs the address to the data lines (Initial value).
1	Does not output the address to the data lines (It becomes Hi-Z during ALC period).

(Note) See "■Reference Information" for specific operations.

[bit10] Reserved: Reserved bit

The read value is undefined.
Set this bit to "0" when writing.

[bit9] ALEINV: ALE Inverter

This bit is used to set up the polarity of the ALE signal.

bit	Description
0	ALE signal becomes positive polarity (Initial value).
1	ALE signal becomes negative polarity.

(Note) To connect multiple devices and use the ALE signal, it is recommended to use the same polarity for their ALE signals.

[bit8] MPXMODE: MPX MODE

This bit is used to select operation bus mode.

bit	Description
0	Selects separate mode (Initial value).
1	Selects multiplex mode.

[bit7] SHRTDOUT: Short Data Out

This bit is used to select to which idle cycle the write data output is extended.

bit	Description
0	Extends the write data output to the last idle cycle (Initial value).
1	Stops the write data output at the first idle cycle.

(Note) See "■Reference Information" for specific operations.

[bit6] RDY: RDY Mode On

This bit is used to control the external RDY function.

bit	Description
0	External RDY mode OFF (Initial value)
1	External RDY mode ON

[bit5] PAGE (PAGE access mode): NOR Flash memory page access mode

This bit controls the mode of NOR Flash memory page access.

In NOR Flash memory page access mode, the first read access cycle (FRADC) setting can generate the first address cycle. Subsequently, the read access cycle (RACC) setting can continue the access until it reaches the 16-byte boundary.

To select NOR Flash memory page access mode, set the RBMON bit to "0" and the read access cycle (RADC) to "0".

bit	Description
0	NOR Flash memory page access mode is turned OFF (Initial value)
1	NOR Flash memory page access mode is turned ON

[bit4] NAND: NAND Flash memory mode

This bit controls the mode used to connect with a NAND Flash memory.

To enable the access to a NAND Flash memory, set this bit to "1".

In NAND Flash memory mode, the corresponding MCSX is fixed to LOW and, subsequently, the pin dedicated to the NAND Flash memory is used during the access. If this bit is set to "0" while the NAND Flash memory is unused, then MCSX is fixed to HIGH, enabling the NAND Flash memory to maintain a low power consumption state.

bit	Description
0	NAND Flash memory mode is turned OFF (Initial value)
1	NAND Flash memory mode is turned ON

[bit3] WEOFF (WEX OFF): Write Enable OFF

This bit can disable the write enable signal (MWEX) operation.

When the byte mask signal (MDQM) is used as a device write enable signal, disabling unnecessary MWEX operation can reduce power consumption. When this bit is set to disable, MWEX is fixed to HIGH.

bit	Description
0	Enable [Initial value]
1	Disable

[bit2] RBMON: Read Byte Mask ON

This bit can enable the byte mask signal (MDQM) for read access.

The setting controls the output of unnecessary data from a device for which the byte mask signal is enabled. This is helpful to reduce power consumption.

bit	Description
0	Disable [Initial value]
1	Enable

[bit1:0] WDTH: Data Width

These bits specify the data bit width of a device to be connected.

bit1	bit0	Description
0	0	8 bits [Initial value]
0	1	16 bits
1	0	Setting is prohibited.
1	1	Setting is prohibited.

(Note) The initial value of these bits becomes "01" only for MODE4 register.

<Notes>

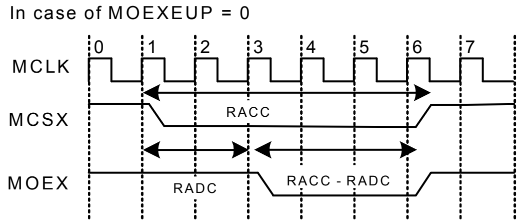
- If a disabled value is written to the WDTH bit, the operation of the external bus interface is not guaranteed.
- The NAND Flash memory mode may not be used by some of products as the NAND Flash control pin is not output. See the data sheets of products used for the details.
- Always write "0" to write to the reserved bit. The read value from the reserved bit is undefined.

■ Reference Information

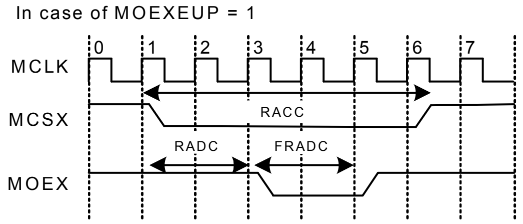
Specific effects and operations are shown as follows by setting the bits of this register.

● MOEXEUP bit: MOEX Width Setup

How to set up the MOEX width depends on the MOEXEUP bit setup.



MOEX width is determined with (RACC-RADC).



MOEX width is determined with FRADC.

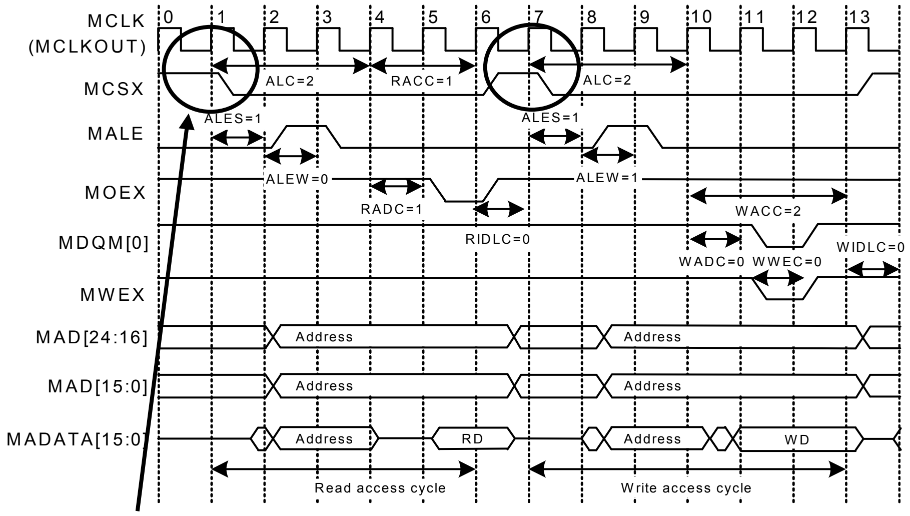
The following is the summary of MOEX width setups with page read setups and MOEXEUP setups.

	MOEXEUP=0	MOEXEUP=1
Page read: OFF	MOEX width => RACC-RADC (Note) Setup with RACC > RADC is required.	MOEX width => FRADC (Note) Setup with RACC ≥ RADC + FRADC is required.
Page read: ON	MOEX width => Simultaneous assertion with MCSX (Note) Setup with RADC=0 is required.	Not allowed

● MPXCSOF bit: CS Assert Timing Setup

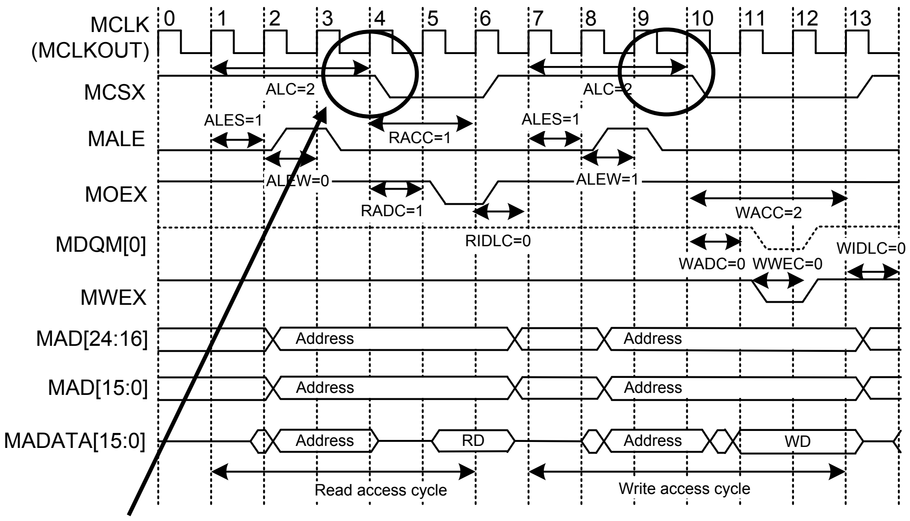
This bit is used to select MCSX assertion in multiplex mode from the start of accessing to the end of address output complete (ALC period).

In case of MPXCSOF = 0



MCSX is asserted at the beginning of ALC period.

In case of MPXCSOF = 1



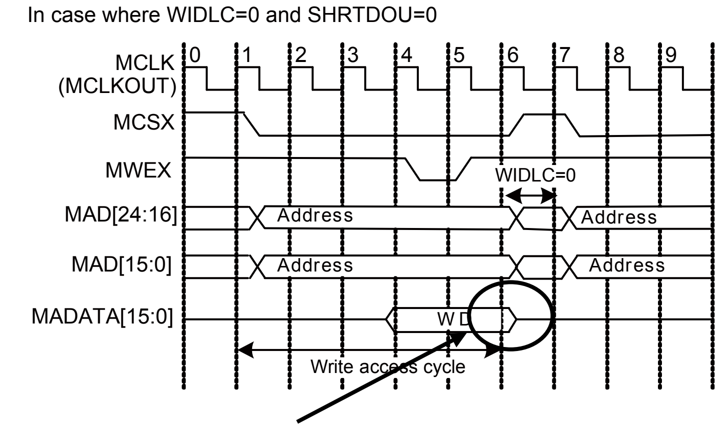
MCSX is asserted after the ALC period is completed.

Setting MPXCSOF=1 means that the MCSX assertion becomes the "address latch".
Therefore, it is enabled if address change is detected by change of MCSX.

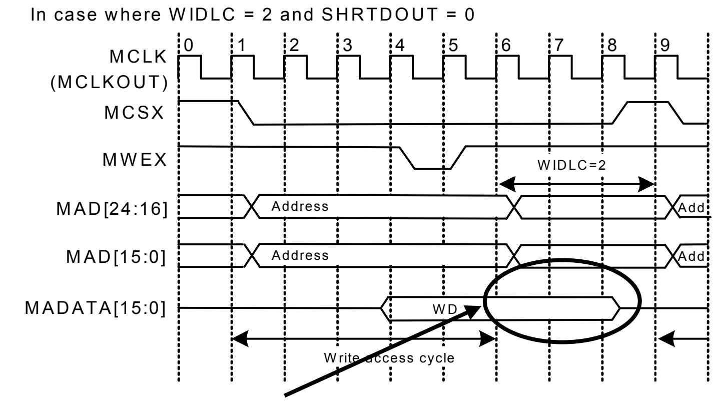
● SHRTDOUT bit: Write Data Retention Time in Idle Cycle

The WIDLC setup will extend the idle cycle.

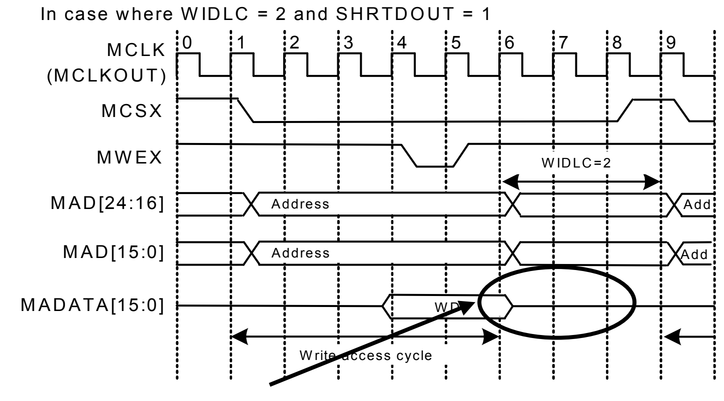
The write data at that time will also be extended same as the specified period to become Hi-Z in the last cycle.



The idle cycle is one cycle. The idle cycle will stop the data output to become Hi-Z.



Extended idle cycle makes the data Hi-Z at the last cycle.

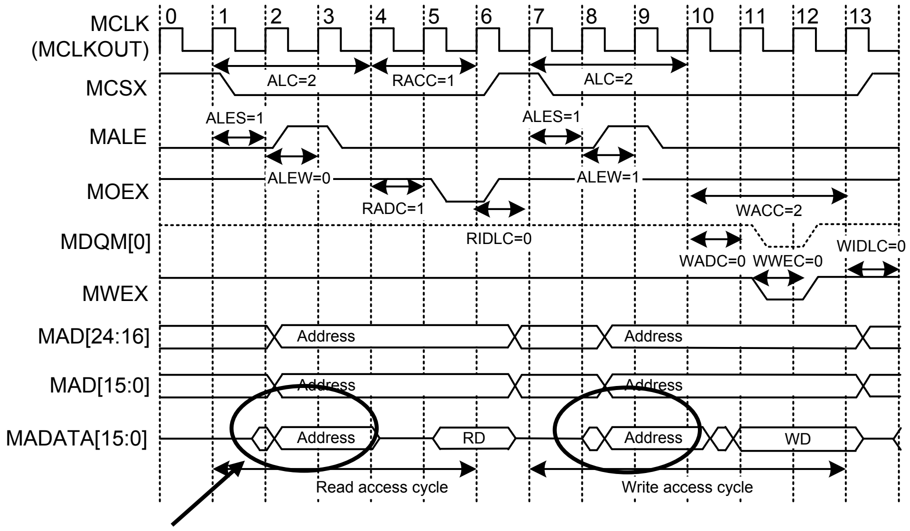


Setting SHRTDOUT=1 extended the idle cycle but the data will become Hi-Z at the beginning of the idle cycle.

● MPXDOFF bit: Address Output Availability Setup for Data Line

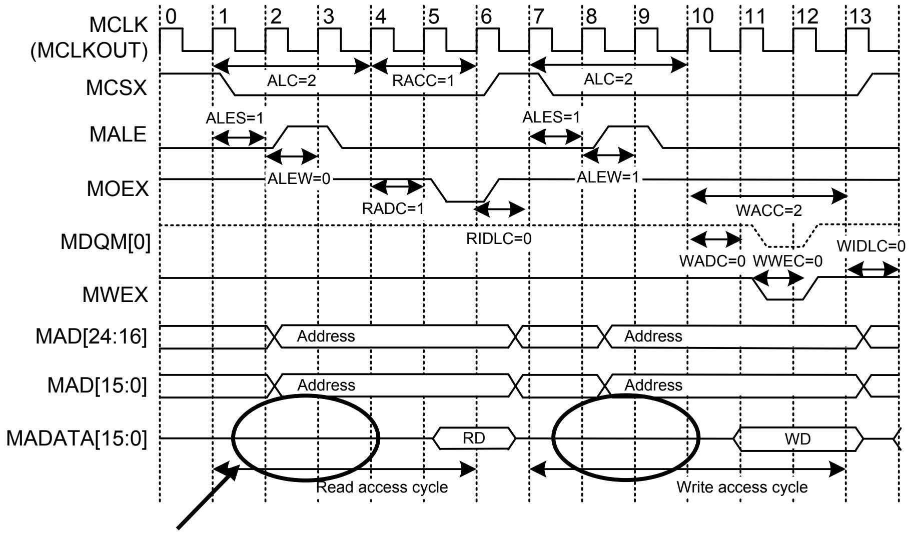
This bit is used to select the address output availability for the data line in multiplex mode.

In case of MPXDOFF = 0



Address is output to the MADATA line during ALC period.

In case of MPXDOFF = 1



Address is not output to the MADATA line during ALC period.

In the multiplex mode , to use ALE signal only or to use the ALE signal but use the MAD for the address pins, it can be used by setting MPXDOFF=1.

6.2. Timing Register 0 to Timing Register 7 (TIM0 to TIM7)

The TIM0 to TIM7 registers set the auto wait time at SRAM/Flash memory access.

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	WIDLC				WWEC				WADC				WACC			
Attribute	R/W				R/W				R/W				R/W			
Initial value	0000				0101				0101				1111			

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	RIDLC				FRADC				RADC				RACC			
Attribute	R/W				R/W				R/W				R/W			
Initial value	1111				0000				0000				1111			

[bit31:28] WIDLC: Write Idle Cycle

These bits set the number of idle cycles after write access.
Write idle cycle will be used during (WIDLC+1) cycle.

bit31	bit30	bit29	bit28	Description
0	0	0	0	1 cycle [Initial value]
⋮				⋮
1	1	1	1	16 cycles

[bit27:24] WWEC: Write Enable Cycle

These bits set the number of assert cycles of write enable.
Write enable will be asserted during (WWEC+1) cycle.

The setting of these bits affects the byte mask signal (MDQM).

bit27	bit26	bit25	bit24	Description
0	0	0	0	1 cycle
⋮				⋮
0	1	0	1	6 cycles [Initial value]
⋮				⋮
1	1	1	0	15 cycles
1	1	1	1	Setting is prohibited.

[bit23:20] WADC: Write Address Setup cycle

These bits set the number of setup cycles of write address.
Write address setup will be used during (WADC+1) cycle.

The address is output during the cycle set by these bits, but a write enable signal is not asserted until the set cycle starts.

bit23	bit22	bit21	bit20	Description
0	0	0	0	1 cycle
⋮				⋮
0	1	0	1	6 cycles [Initial value]
⋮				⋮
1	1	1	0	15 cycles
1	1	1	1	Setting is prohibited.

[bit19:16] WACC: Write Access Cycle

These bits set the number of cycles required for write access.
Write access cycle will be used during (WACC+1) cycle.

The address remains unchanged during the cycle set by these bits.

The number of cycles set by these bits must be equal to or more than the sum of the address setup cycle (WADC) and the write enable cycle (WWEC).

bit19	bit18	bit17	bit16	Description
0	0	0	0	Setting is prohibited.
0	0	0	1	Setting is prohibited.
0	0	1	0	3 cycles
⋮				⋮
1	1	1	1	16 cycles [Initial value]

[bit15:12] RIDLC: Read Idle Cycle

These bits set the number of idle cycles after read access.
Read access cycle will be used during (RIDLC+1) cycle.

They are used to avoid data collision caused by a write access occurring immediately after a read access.

bit15	bit14	bit13	bit12	Description
0	0	0	0	1 cycle
⋮				⋮
1	1	1	1	16 cycles [Initial value]

[bit11:8] FRADC: First Read Address Cycle

The functions of these bits are changed by the settings of MODE:PAGE (Page read access setting) and MOEXEUP (MOEX width setting method selection).

- When MODE:PAGE=0 (Page read access: OFF) and MOEXEUP=0
  These bits do not affect the settings of Page read access and MPEX width.

- When MODE:PAGE=0 (Page read access OFF) and MOEXEUP=1
  These bits set the MOEX width. In this case, a setup with RACC ≥ RADC+FRDC is required.

bit11	bit10	bit9	bit8	Description
0	0	0	0	1 cycle [Initial value]
⋮				⋮
1	1	1	1	16 cycles

- When MODE:PAGE=1 (Page read access: ON) and MOEXEUP=0
  These bits set the waiting time at initial access at the page read access of Flash memory. Before setting values other than "0" to these bits, SET "0" to RADC (Read Access Setup Cycle).

bit11	bit10	bit9	bit8	Description
0	0	0	0	0 cycle [Initial value]
⋮				⋮
1	1	1	1	15 cycles

- When MODE:PAGE=1(Page read access: ON) and MOEXEUP=1
  This setting is prohibited.

[bit7:4] RADC: Read Address Setup cycle

These bits set the number of setup cycles of read address.
Read address setup cycle will be used during (RADC) cycle.

Within the read address setup cycle, MCSX and address are asserted but MOEX is not asserted. If "0" is set to any of these bits, MOEX and MCSX are always asserted.

The set value must be less than the number of read access cycles. (RADC < RACC).

When using NOR Flash memory page access mode, set these bits to "0b0000".

If the access size is more than the target width, or if a device such as NAND Flash memory needs to switch HIGH and LOW of read enable (MOEX or MNREX), set these bits to "0b0001" or a higher value.

bit7	bit6	bit5	bit4	Description
0	0	0	0	0 cycles [Initial value]
⋮				⋮
1	1	1	1	15 cycles

[bit3:0] RACC: Read Access Cycle

These bits set the number of cycles required for read access.
Read access cycle will be used during (RACC+1) cycle.

The address remains unchanged during the cycle specified by these bits, and the data is captured at the last cycle.

bit3	bit2	bit1	bit0	Description
0	0	0	0	1 cycle
⋮				⋮
1	1	1	1	16 cycles [Initial value]

<Notes>

· If you write a disabled value to a WWEC, WADC or WACC bit, the operation of the external bus interface is not guaranteed.

· In NAND Flash memory read mode, the MNWEX and MNREX timings are set by the timing registers as is the case with MWEX and MOEX.

6.3. Area Register 0 to Area Register 7 (AREA0 to AREA7)

The AREA0 to AREA7 registers set the address area by CS0 to CS7.

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	Reserved									MASK						
Attribute	-									R/W						
Initial value	-									0001111 (16MB width)						

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved								ADDR							
Attribute	-								R/W							
Initial value	-								(from MCSX[0]) 00000000, 00010000, 00100000, 00110000, 01000000, 01010000, 01100000, 01110000 *1							

[bit31:23]Reserved: Reserved bits

[bit22:16] MASK: address mask

These bits set the value to mask [26:20] of the internal address (27:20) set in ADDR.

If "1" is set as a mask value according to the specified mask value, the external bus interface masks each of the internal bus and ADDR, and compares the masking results. If the results are matched, the external bus interface accesses the MCSX signal.

A bit set to "1" for masking is lost during masking process. The bit is disabled even if it is set in ADDR.

The example shown in Table 6-2 indicates the relationship between the mask setup and the address area size.

Table 6-2 MASK Setup Value and Address Area per CS

MASK setup value	Address area per CS
111_1111	128MB
011_1111	64MB
001_1111	32MB
000_1111	16MB
000_0111	8MB
000_0011	4MB
000_0001	2MB
000_0000	1MB

[bit31:23]Reserved: Reserved bits

[bit7:0] ADDR: Address

These bits specify the address to set the corresponding MCSX area.

The address is in the fixed 256 MB area assigned to the SRAM/Flash memory interface.

The address specified by bit7:0 corresponds to the internal address [27:20].

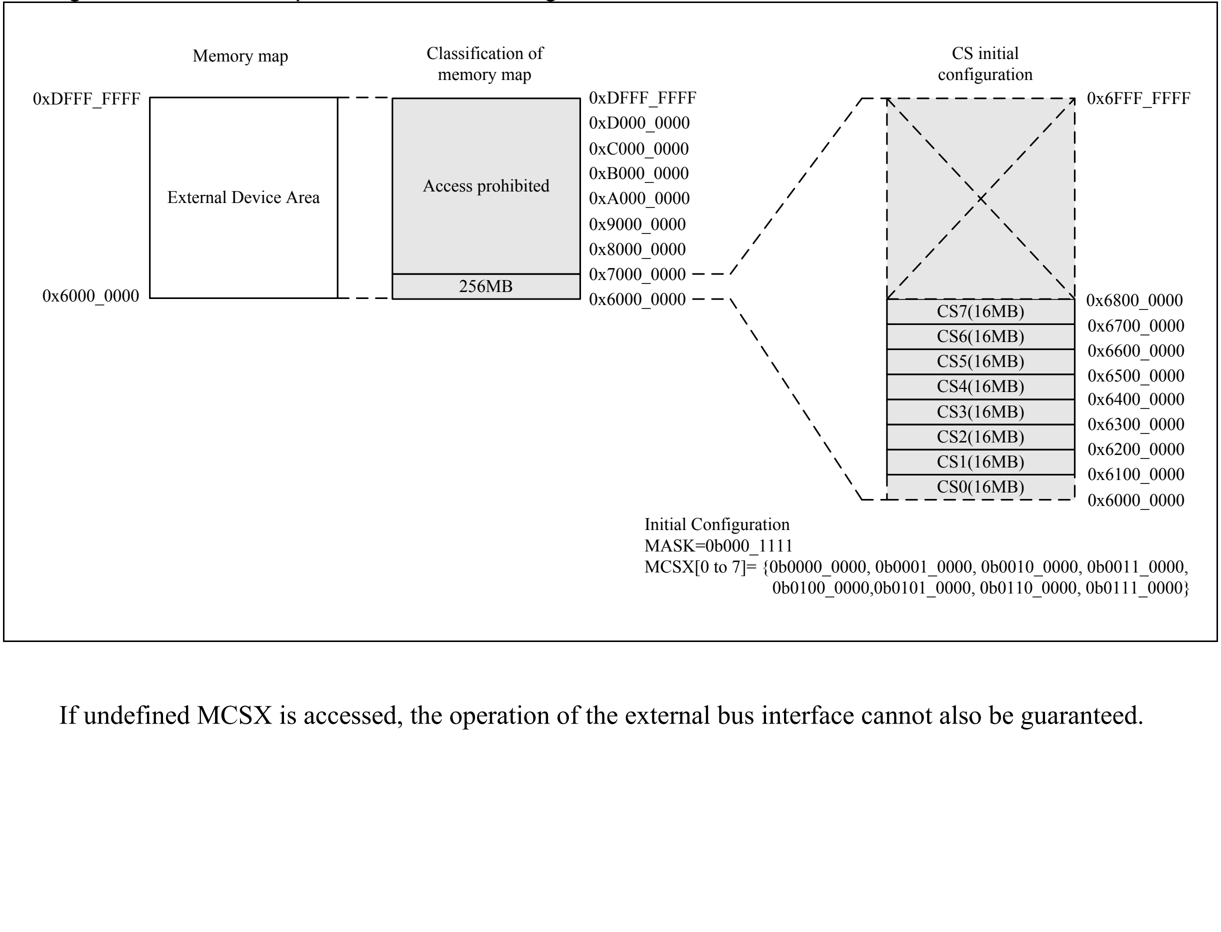
<Notes>

Address Space

- The external bus interface employs 256MB of address space.
- The address area for each chip select can be set freely to one MCSX with [27:20] up to 128MB and down to 1MB.
- The address areas of each chip select must not be overlapped.
- As the address output outside consists of 25 bits, the maximum size of address output outside becomes 32MB.

The address space on the memory map and initial state of the address space from each chip select are shown in Figure 6-1.

Figure 6-1 Address Space and Initial Configuration for External Bus Interface



Initial Configuration
MASK=0b000_1111
MCSX[0 to 7]= {0b0000_0000, 0b0001_0000, 0b0010_0000, 0b0011_0000,
0b0100_0000,0b0101_0000, 0b0110_0000, 0b0111_0000}

If undefined MCSX is accessed, the operation of the external bus interface cannot also be guaranteed.

Setup Example

ADDR = 0b0001_0000 ([27:20] of the first MCSX address. 0x6100_0000 in this setup.)
MASK = 0b000_0011 (Mask bits [26:20] for MCSX. Address area for this setup: 4MB.)

Select an area size with the mask setup values.
In the example, setup range 0x6100_0000 to 0x613F_FFFF (4MB) will be selected.

ADDR&(!MASK) = 0b0001_0000

· Device to be selected

When the internal bus address (address for external I/F) AD = 0x6101_1000:
0x6101_1000 -> 0b0110_$\boxed{0001_0000}$_0001_0001_0000_0000_0000

AD[27:20] => 0b0001_0000

Masking comparison
ADDR & (!MASK) = 0b0001_0000
AD [27:20] & (!MASK) = 0b0001_0000 } ….. Matched. A device will be selected.

· Device not to be selected

When the internal bus address (address for external I/F) AD = 0x60C0_0000:
0x60C0_0000 -> 0b0110_$\boxed{0000_1100}$_0000_0000_0000_0000_0000

AD[27:20] => 0b0000_1100

Masking comparison
ADDR & (!MASK) = 0b0001_0000
AD [27:20] & (!MASK) = 0b0000_1100 } …..Unmatched. No device will be selected.

6.4. ALE Timing Register 0 to 7 (ATIM0 to ATIM7)

The ATIM0 to ATIM7 registers set the automatic wait time of MALE..

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	Reserved															
Attribute	-															
Initial value	-															

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved				ALEW				ALES				ALC			
Attribute	-				R/W				R/W				R/W			
Initial value	-				0100				0101				1111			

[bit31:12]Reserved: Reserved bits

[bit11:8] ALEW: Address Latch Enable Width

These bits are used to set the assertion period for MALE.
MALE signal will be asserted during (ALEW+1) cycle.

bit11	bit10	bit9	bit8	Description
0	0	0	0	1 cycle
⋮				⋮
0	1	0	0	5 cycles [Initial value]
⋮				⋮
1	1	1	1	16 cycles

[bit7:4] ALES: Address Latch Enable Setup cycle

These bits are used to set the setup cycle for ALE assertion.
ALE will not be asserted from the access start during (ALES) cycle.

bit7	bit6	bit5	bit4	Description
0	0	0	0	0 cycle
⋮				⋮
0	1	0	1	5 cycles [Initial value]
⋮				⋮
1	1	1	1	15 cycles

[bit3:0] ALC : Address Latch Cycle

These bits are used to set the address latch cycle.
Address will be output from CS assert and data line during (ALC+1) cycle.

bit3	bit2	bit1	bit0	Description
0	0	0	0	1 cycle
⋮				⋮
1	1	1	1	16 cycles [Initial value]

<Notes>

- Setups to the ATIM register is available only in multiplex mode.

6.5. SDRAM Mode Register (SDMODE)

This section explains the configuration of SDMODE.

bit	31	30	29	28	27	26	25	24
Field	Reserved							
Attribute	-							
Initial	-							

bit	23	22	21	20	19	18	17	16
Field	Reserved							MSDCLK
Attribute	-							R/W
Initial	-							0

bit	15	14	13	12	11	10	9	8
Field	BASEL				RASEL			
Attribute	R/W				R/W			
Initial	0001				0011			

bit	7	6	5	4	3	2	1	0
Field			CASEL			ROFF	PDON	SDON
Attribute	-		R/W		-	R/W	R/W	R/W
Initial	-		00		-	0	0	0

[bit16]MSDCLKOFF: MSDCLK OFF

This bit sets the output of the clock for SDRAM (MSDCLK).
This bit stops the lock for SDRAM (MSDCLK). This is independent from the state of SDON. Therefore, when this bit set to “1” at SDON=1, the clock supply to SDRAM is stopped and the access is disabled.

bit	Description
0	Outputs the clock for SDRAM (MSDCLK) [Initial value]
1	Dose not output the clock for SDRAM (MSDCLK).

[bit15:12]BASEL: Bank Address Select

These bits select the address bit on the internal bus output as address bank.

bit	Description
0000	MAD[15:14] = Internal address[20:19]
0001	MAD[15:14] = Internal address [21:20] [Initial value]
0010	MAD[15:14] = Internal address [22:21]
0011	MAD[15:14] = Internal address [23:22]
0100	MAD[15:14] = Internal address [24:23]
0101	MAD[15:14] = Internal address [25:24]
0110	MAD[15:14] = Internal address [26:25]
0111 to 1111	Setting is prohibited.

[bit11:8]RASEL: Row Address Select

These bits select the address bit on the internal bus output as row address.

bit	Description
0000	MAD[13:0] = Internal address [19:6]
0001	MAD[13:0] = Internal address [20:7]
0010	MAD[13:0] = Internal address [21:8]
0011	MAD[13:0] = Internal address [22:9] [Initial value]
0100	MAD[13:0] = Internal address [23:10]
0101	MAD[13:0] = Internal address [24:11]
0110	MAD[13:0] = Internal address [25:12]
0111 to 1111	Setting is prohibited

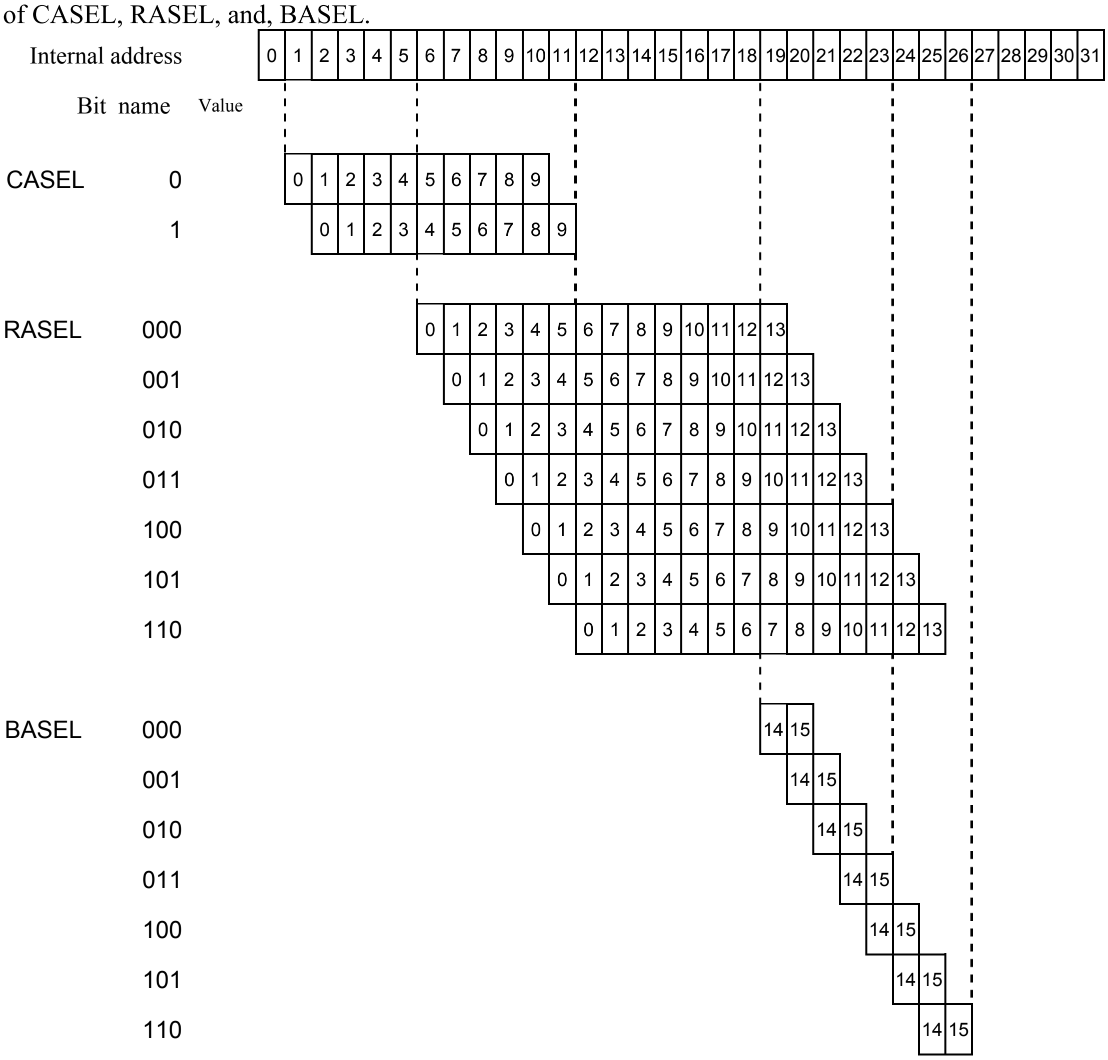
[bit7:6]Reserved: Reserved bits

[bit5:4]CASEL: Column Address Select

These bits select the address bit on the internal bus output as column address. And, this setting is used also as the setting of SDRAM bus width.

bit	Description
00	MAD[9:0] = Internal address [10:1]、16-bit width[Initial value]
01	MAD[9:0] = Internal address [11:2]、32-bit width[
10 to 11	Setting is prohibited

The following shows the table indicating the relationship between internal address bit and MAD bit at the each setting of CASEL, RASEL, and, BASEL.



Internal address		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Bit name	Value																																
CASEL	0		0	1	2	3	4	5	6	7	8	9																					
	1			0	1	2	3	4	5	6	7	8	9																				
RASEL	000							0	1	2	3	4	5	6	7	8	9	10	11	12	13												
	001								0	1	2	3	4	5	6	7	8	9	10	11	12	13											
	010									0	1	2	3	4	5	6	7	8	9	10	11	12	13										
	011										0	1	2	3	4	5	6	7	8	9	10	11	12	13									
	100											0	1	2	3	4	5	6	7	8	9	10	11	12	13								
	101												0	1	2	3	4	5	6	7	8	9	10	11	12	13							
	110													0	1	2	3	4	5	6	7	8	9	10	11	12	13						
BASEL	000																				14	15											
	001																					14	15										
	010																						14	15									
	011																							14	15								
	100																								14	15							
	101																									14	15						
	110																										14	15					

[bit2]ROFF: Refresh OFF

This bit sets the refresh.

Use this bit to stop the refresh operation temporarily for the access to SDRAM command register etc. The refresh counter itself is not stopped, but the refresh operation is not executed at the refresh timing. When this bit is released, the refresh is implemented immediately if the refresh timing is passed. However, in this time, only one refresh operation is executed even if two refresh timing is passed.

bit	Description
0	Refresh ON [Initial value]
1	Refresh OFF

[bit1]PDON: Power Down ON

This bit set the power down mode.
For the detailed operations, see “3.9 Power Down Mode”.

bit	Description
0	Power Down Mode OFF [Initial value]
1	Power Down Mode ON

[bit0] SDON: SDRAM ON

This bit enables the access to SDRAM.
In this operation, the power-on sequence is issued to SDRAM and the mode register setting etc. is automatically made to enable the access to SDRAM.

When this bit is changed to “0” during the operation, the access to SDRAM is stopped thoroughly after completing the access to SDRAM (if in refresh operation, after the refresh operation completion (after (TREFC+1） × (NREF+1) cycle). When PDON=1, MSDCKE is held at “Low”. During “OFF”, the data is not saved because the refresh operation is not executed. For the access to SDRAM address area, the error response is returned.

When SDRAM command register (SDCMD) is written in “OFF” state, the external bus interface does not issue the power-on sequence and automatically sets this bit “ON”(=1) on the assumption that a user starts SDRAM with a program. (in this case, set ROFF=1 (Refresh OFF).)

For the issued power-on sequence, see Figure 3-19.

bit	Description
0	Disables the access to SDRAM.[Initial value]
1	Enables the access to SDRAM.

<Notes>

- The division ratio of SDRAM clock (MSDCLK) is set with MDIV bit of the division clock register DCLKR).
- With beginning the clock supply by changing MSDCLKOFF from “1” to “0”, do not change SDON from “0” to “1” at the same time due to a delay in clock output.
- When the refresh is aborted temporarily wit ROFF, to hold the data, note that the temporary abort time does not exceed the refresh timing or execute the refresh explicitly by accessing SDRAM command register (SDCMD)
- Change SDON to “1” after completing the setting of REFTIM, PWRDWN, and SDTIM registers except SDMODE.SDON.

Address Area

- External bus interface has 1792 MB SDRAM address area.
- The maximum size of address output to the outside by the setting of BASEL, RASEL, and CASEL is 128 MB.
- For the address area on the memory map, see Figure 6-1.

6.6. Refresh Timer Register (REFTIM)

The REFTIM register sets the refresh timing to SDRAM.

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	Reserved							PREF	NREF							
	-							R/W	R/W							
	-							0	00000000							

bit	15	14	13	12	11	10	9	8	7	6
Field	REFC									
	R/W									
	0000000000110011									

[bit31:25] Reserved: Reserved bits

[bit24]PREF: Pre-Refresh

This bit set the preceding refresh.
For the detailed operations, see “3.9 Refresh Operation”.

bit	Description
0	Disables the preceding refresh. [Initial Value]
1	Enables the preceding refresh.

[bit23:16]NREF: Number of Refresh

These bits set the issuing count of refresh at one-time refresh.
During the refresh operation, the access is disabled, but the refresh operation can be executed intensively in the specified period. However, according to the refresh count specified with these bits, set the REFC value by calculating the following formula.

REFC = (Count value of one-time refresh operation) × (NREF+1)

bit	Description
00000000	One time [Initial Value]
⋮	⋮
11111111	256 times

[bit15:0]REFC: Refresh Count

These bits set the interval of refresh operations.

The refresh operation starts when the specified cycle count (MSDCLK) elapses. When the writing is executed during accessing, the executing access is aborted and the first refresh operation is implemented to reflect the setting immediately.

bit	Description
0x0000 to 0x0009	Setting is prohibited.
	11 cycles
⋮	⋮
0X0033	52 cycles [Initial Value]
⋮	⋮
0xFFFF	65536 cycles

<Note>

- The minimum refresh interval of REFC should not be shorter than cycle count (TREFC+1) required for one-time refresh operation. Note that a hang-up would occur if the interval is set to be shorter than the required cycle count.

6.7. Power Down Count Register (PWRDWN)

The PWRDWN register set the count value required for SDRAM to transfer to power down mode..

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	Reserved															
Attribute	-															
Initial	-															

bit	15	14	13	12	11											
Field	PDC															
Attribute	R/W															
Initial	0000000000000000															

[bit31:15] Reserved: Reserved bits

[bit15:0] PDC: Power Down Count

These bits set the count value required for SDRAM to transfer to power down mode.
When the access to SDRAM is implemented in the cycle (MSDCLK) specified with these bits, the mode is transferred to the power down mode.

bit	Description
0x0000	0 cycles [Initial Value]
⋮	⋮
0x8000	32768 cycles
⋮	⋮
0xFFFF	65535 cycles

6.8. SDRAM Timing Register (SDTIM)

The SDTIM register set the automatic wait time at SDRAM access.

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	Reserved						TDPL		TREFC				TRAS			
Attribute	-						R/W		R/W				R/W			
Initial Value	-						00		0100				0010			

bit	15	14	13	12	11	10										
Field	TRCD				TRP				TRC				Reserved		CL	
Attribute	R/W				R/W				R/W				-		R/W	
Initial Value	0001				0001				0100				-		01	

[bit31:26] Reserved: Reserved bits

[bit25:24] TDPL: Data-in to Precharge Lead Time

These bits set the latency from write to precharge.

bit	Description
00	1 cycle [Initial Value]
⋮	⋮
11	4 cycles

[bit23:20] TREFC: Refresh Cycle time

These bits set the latency for a command following the refresh operation.

bit	Description
0000	1 cycle
⋮	⋮
0100	5 cycles [Initial Value]
⋮	⋮
0111	8 cycles
1000	Setting is prohibited.
⋮	⋮
1111	Setting is prohibited.

[bit19:16] TRAS: RAS active time

These bits set the minimum active time of ROW.

bit	Description
0000	1 cycle
⋮	⋮
0010	3 cycles [Initial Value]
⋮	⋮
0111	8 cycles
1000	Setting is prohibited.
⋮	⋮
1111	Setting is prohibited.

[bit15:12] TRCD: RAS-CAS Delay

These bits set the latency from RAS to CAS.

bit	Description
0000	1 cycle
0001	2 cycles [Initial Value]
0010	Setting is prohibited.
⋮	⋮
1111	Setting is prohibited.

[bit11:8] TRP: RAS Precharge time

These bits set the precharge time.

bit	Description
0000	1 cycle
0001	2 cycles [Initial Value]
⋮	⋮
0011	4 cycles
0100	Setting is prohibited.
⋮	⋮
1111	Setting is prohibited.

[bit7:4] TRC: RAS Cycle time

These bits set the latency between RASs.

bit	Description
0000	1 cycle
⋮	⋮
0100	5 cycles [Initial Value]
⋮	⋮
0111	8 cycles
1000	Setting is prohibited.
⋮	⋮
1111	Setting is prohibited.

[bit1:0] CL: CAS Latency

These bits set CAS latency.

bit	Description
00	1 cycle
01	2 cycles [Initial Value]
10	3 cycles
11	Setting is prohibited.

6.9. SDRAM Command Register (SDCMD)

The SDCMD register outputs the set value to the external pin for controlling SDRAM.

bit	31	30	29	28	27	26	25	24
Field	PEND	Reserved						
Attribute	R	-						
Initial Value	0	-						

bit	23	22	21	20	19	18	17	
Field	Reserved			SDCKE	SDCS	SDRAS	SDCAS	SDWE
Attribute	-			R/W	R/W	R/W	R/W	R/W
Initial Value	-			0	0	0	0	

bit	15	14	13	12	11	10	9	
Field	SDAD[15:8]							
Attribute	R/W							
Initial Value	0x00							

bit	7	6	5	4	3	2	1	
Field	SDAD[7:0]							
Attribute	R/W							
Initial Value	0x00							

On detecting the writing, this register outputs the value saved in the register for one cycle. For this period, n MADATA[15:0] holds the prior values. For the external bus interface, the power-on sequence of SDRAM is automatically executed, so usually this register is not used. Only for the settings in operation (extended mode register setting etc.), this register is used.

When setting SDCK=”0”, the mode is entered into the power down mode immediately. In this case, the state of PDON bit of SDRAM mode register has no relation with this operation. The mode is returned by an access to SDRAM. Refresh, or, writing to this register. To stop the refresh operation, set ROFF=1 (Refresh OFF).

[bit31]PEND: Pend

Immediately after detecting the writing to this register, this bit is asserted because the operation cannot be executed instantly due to an access to a different device. Before writing this register, check this bit and confirm that the bit is deasserted. When this bit is written during assertion, the operation is not guaranteed.

bit	Description
0	Writing to SDCMD register is enabled.
1	Writing to SDCMD register is disabled.

[bit30:21] Reserved: Reserved bits

[bit20]SDCKE: SDRAM CKE

On detecting the writing to this register, outputs the value set in this bit to MADCKE.

[bit19] SDCS: SDRAM Chip Select

On detecting the writing to this register, outputs the value set in this bit to MCSX[8].

[bit18] SDRAS: SDRAM RAS

On detecting the writing to this register, outputs the value set in this bit to MRASX.

[bit17] SDCAS: SDRAM CAS

On detecting the writing to this register, outputs the value set in this bit to MCASX.

[bit16] SDWE: SDRAM Write Enable

On detecting the writing to this register, outputs the value set in this bit to MSDWEX.

[bit15:0] SDAD: SDRAM ADress

On detecting the writing to this register, outputs the value set in this bit to MAD[15:0].

6.10. Memory Controller Register (MEMCERR)

The MEMCERR register enables SDRAM/Flash memory/SDRAM error interrupt.

bit	31	30	29	28	27	26	25	24
Field	Reserved							
Attribute	-							
Initial Value	-							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
Attribute	-							
Initial Value	-							

bit	15	14	13	12	11	10	9	8
Field	Reserved							
Attribute	-							
Initial Value	-							

bit	7	6	5	4	3	2	1	0
Field	Reserved				SDION	SFION	SDER	SFER
Attribute	-				R/W	R/W	R/W	R/W
Initial Value	-				0	0	0	0

[bit31:4] Reserved: Reserved bits

[bit3] SDION: SDRAM error Interrupt ON

Enables an interrupt for SDRAM error.

bit	Description
0	Disables SDRAM error interrupt.
1	Enables SDRAM error interrupt.

[bit2] SFION: SRAM/Flash error Interrupt ON

Enables an interrupt for SRAM /Flash memory error.

bit	Description
0	Disables SRAM/Flash memory error interrupt.
1	Enables SRAM/ Flash memory error interrupt.

[bit1] SDER: SDRAM Error

This bit is used to indicate that the access to SDRAM area is executed in the condition where SDON=1 is not set in SDRAM mode register (SDMODE). At this time, the external bus interface returns an error response to the internal bus and sets this register at the same time. This bit is cleared by writing “1”.

bit	Description
0	No SDRAM error exists.
1	SDRAM error exists.

[bit0] SFER: SRAM/Flash Error

This bit is used to indicate that the access is executed to an area not mapped in SRAM/Flash memory area access. At this time, the controller returns an error response to the internal bus and sets this register at the same time. This bit is cleared by writing “1”.

bit	Description
0	No SRAM/Flash memory error exists.
1	SRAM/ Flash memory error exists.

6.11. Division Clock Register (DCLKR)

The following shows the configuration of DCLKR.

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	Reserved															
Attribute	-															
Initial value	-															

bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field	Reserved								Reserved			MCLKON	MDIV			
Attribute	-								-			R/W	R/W			
Initial value	-								-			0	1111			

[bit4] MCLKON: MCLKOUT ON

This bit is used to set the output of clock for SRAM/Flash memory (MCLKOUT) enable.

bit	Description
0	Does not output the clock for SRAM/Flash memory (MCLKOUT). [Initial value]
1	Outputs the clock for SRAM/Flash memory (MCLKOUT).

[bit3:0] MDIV: MCLK /MSDCLK Division Ratio Setup

These bits are used to set the division ratio (1/1 to 1/16) of the division clock.
The division clock will be divided into (MDIV+1) division.
The division ratio specified with these bits is reflected to both SRAM/Flash memory clock (MCLKOUT) and SDRAM clock (MADCLK).

bit3	bit2	bit1	bit0	Description
0	0	0	0	1 division
0	0	0	1	2 division
⋮				⋮
1	1	1	1	16 division [Initial value]

<Notes>

- When you want to output the division clock, clock output setup is required with the use of GPIO.
  See a separate chapter "I/O Port" for the details of the setups.
  To output the MCLKOUT set MCLKON=1. To output MSDCLK, set SDRAM mode register (SDMODE) to MSDCLKOFF=0.
- When MCLKOUT is set to output by dividing the frequency by one, set MCLKON=1, and MDIV=0000.
  When MSDCLK is set to output by dividing the frequency by one, set SDRAM mode register to MSDCLK=1 and MDIV=0000.
  In this case, check "External bus clock output Characteristics" in Data sheet.
- To change the division ratio, make sure to set MCLKON=0 and SDRAM mode register (SDMODE) to MSDCLKOFF=1 before change the MDIV.
- Any change to this register during external bus accessing is prohibited.

6.12. Error Status Register (EST)

The following shows the configuration of EST.

bit	31	30	29	28	27	26	25	24
Field	Reserved							
Attribute	-							
Initial Value	-							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
Attribute	-							
Initial Value	-							

bit	15	14	13	12	11	10	9	8
Field	Reserved							
Attribute	-							
Initial Value	-							

bit	7	6	5	4	3	2	1	0
Field	Reserved							WERR
Attribute	-							R
Initial Value	-							0

[bit31:1] Reserved: Reserved bits

[bit0] WERR

This bit is used to indicate the reception of error response in a write access when the continuous write access of Access Mode Register (AMODE) is enabled (WAEN=1). On the reception of error response, this bit is changed from “0” to “1”. When an error response is received in the subsequent write access, the bit is rewritten to a new state.

To clear this register to be initial state, write “0” to WERRCLR bit of Error Status Clear Register (ESCLR).

When the continuous write access of Access Mode Register (AMODE) is enabled (WAEN=0) and the frequency division ratio of Division Clock Register (DCLKR) is one (MDIV=0), the state of this bit is not held in write access.

bit	Description
0	No error response exists.
1	Error response exists.

6.13. Write Error Address Register (WEAD)

The WEAD register enables/disables the preceding read and continuous write access.

bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Field	ADDR[31:16]															
Attribute	R															
Initial	0x0000															

bit	15	14	13													
Field	ADDR[15:0]															
Attribute	R															
Initial	0x0000															

[bit31:0] ADDR

These bits are used to hold the AHB address at the reception of an error response on write access when the continuous write access of Access Mode Register (AMODE) is enabled (WAEN=1). When an error response is received in the subsequent write access, the bit is rewritten to a new state.

To clear this register to be initial state, write “0” to WERRCLR bit of Error Status Clear Register (ESCLR).

When the continuous write access of Access Mode Register (AMODE) is enabled (WAEN=0) and the frequency division ratio of Division Clock Register (DCLKR) is one (MDIV=0), the state of this bit is not held in write access.

6.14. Error Status Clear Register (ESCLR)

The ESCLR register initializes the error status register (EST) and write error address register (WEAD).

bit	31	30	29	28	27	26	25	24
Field	Reserved							
Attribute	-							
Initial	-							

bit	23							
Field	Reserved							
Attribute	-							
Initial	-							

bit	15	14						
Field	Reserved							
Attribute	-							
Initial	-							

bit	7	6						
Field	Reserved							WERRCLR
Attribute	-							
Initial	-							1

[bit31:0] Reserved: Reserved bits

[bit0] WERRCLR: Write Error Clear

By writing “ 0” to this bit, the Error Status Register (EST) and Write Error Address Register (WEAD) are cleared to the initial state. bit0 is not changed to “0” by writing this bit. “0” is always read.

6.15. Access Mode Register (AMODE)

The AMODE register enables/disables the preceding read and continuous write access..

bit	31	30	29	28	27	26	25	24
Field	Reserved							
Attribute	-							
Initial	-							

bit	23
Field	Reserved
Attribute	-
Initial	-

bit	15	14
Field	Reserved	
Attribute	-	
Initial	-	

bit	7	6	
Field	Reserved		WERRCLR
Attribute	-		R/W
Initial	-		1

[bit31:1]Reserved: Reserved bits

[bit0] WAEN

This bit is used to enable/disable the preceding read and continuous write.

bit	Description
0	Disables the preceding read and continuous write request.
1	Enables the preceding read and continuous write request.

<Note>

- The function of this register is enabled only when the frequency division ratio of MDIV bit of Division Clock Register (DCLKR) is set to be two or more.

7. Usage Precautions

This section explains the usage precautions for the external bus interface.

- **AC Specifications**

  See the data sheets for the AC specifications in each operating mode.

- **External Bus Pin and GPIO Setup**

  For uisng the GPIO as an external bus pin, it is required to set the GPIO to the external bus pin setup with the EPFR register. See Chapter "I/O Port" for the details.

- **Error Responses**

  When an access is made to an external bus area of 256MB SRAM/Flash memory address area, an area which has not been mapped with the area register, or SDRAM address area with the setting of SDRAM mode register of SDON=0, the external bus interface outputs an error response (by setting HRESP[1:0] to "01"). When this error occurs during a burst transfer, the operation of the external bus interface is not guaranteed.

- **Target Device and Functions Setups**

  For the setups for each function per target device, see Table 7-1.

Table 7-1 Target Device and Functions Setups

	Clock output	External RDY	Page mode	NAND Flash	Clock division
SRAM	Available	Available	Not allowed	Not allowed	Available
NOR Flash memory	Available	Not allowed	Available	Not allowed	Available
NAND Flash memory	Not allowed	Not allowed	Not allowed	Available	Available
SDRAM	Available	Not allowed	Not allowed	Not allowed	Available

Do not make a setup using page read and external RDY at the same time.

- **Rewrite Timing of Register Value**

  When the setup value of registers such as the timing register is rewritten from the CPU while accessing the external bus from the DMAC, written values will not be reflected until the access is completed (after the idle cycle).

CHAPTER: SD Card Interface

This chapter explains details of the SD card interface.

"2. Registers" is created based on "SD Specifications Part A2 SD Host Controller Simplified Specification Version 3.00" (to be called "SD Specifications Part A2" later in this document). Only registers having details different from those explained in "SD Specifications Part A2" are explained in "2. Registers". For details of registers not explained in "2. Registers", refer to "SD Specifications Part A2".

CODE: 9BFSDCARD-E01.0

1. Overview of SD Card Interface

This section provides an overview of the SD card interface.

■ SD card specifications

An SD card compliant with the following specifications can be used with the SD card interface.

- Part 1 Physical Layer Specification Version 3.01
- Part E1 SDIO Specification Version 3.00
- Part A2 SD Host Controller Simplified Specification Version 3.00

<Note>

The SD card interface supports only two Bus Speed modes: Default Speed mode and High Speed mode.

■ Features

- SD bus (SPI bus not supported)
- 1-bit data bus, 4-bit data bus
- Double buffer for transfer (buffer size: 2 KB)
- Data write protect detection function
- Card detection function
- Multiple read/write transfer
- Transfer data length: 1 byte to 2048 bytes
- Read Wait Option function
- Suspend/Resume function
- Wakeup function
- Shared bus function
- Default Speed mode and High Speed mode (other Bus Speed modes not supported)
- Auto CMD23 supported
- SDMA supported
- ADMA1 supported
- MMC 4.41 supported (sequential commands not supported)
- MMC boot operation supported

2. Registers

This section explains details of registers of the SD card interface.

■ List of registers of SD card interface

Table 2-1 List of registers of SD card interface

Register name	Reference
SDMA System Address / Argument 2 Register	2.1
Block Size Register	2.2
Block Size Register	2.3
Argument 1 Register	2.4
Transfer Mode Register	2.5
Command Register	2.6
Response Register	2.7
Buffer Data Port Register	2.8
Present State Register	2.9
Host Control 1 Register	2.10
Power Control Register	2.11
Block Gap Control Register	2.12
Wakeup Control Register	2.13
Clock Control Register	2.14
Timeout Control Register	2.15
Software Reset Register	2.16
Normal Interrupt Status Register	2.17
Error Interrupt Status Register	2.18
Normal Interrupt Status Enable Register	2.19
Error Interrupt Status Enable Register	2.20
Normal Interrupt Signal Enable Register	2.21
Error Interrupt Signal Enable Register	2.22
Auto CMD Error Status Register	2.23
Host Control 2 Register	2.24
Capabilities Register	2.25
Maximum Current Capabilities Register	2.26
Force Event Register for Auto CMD Error Status	2.27
Force Event Register for Error Interrupt Status	2.28

Register name	Reference
ADMA Error Status Register	2.29
ADMA System Address Register	2.30
Preset Value Registers	2.31
Shared Bus Control Register	2.32
Slot Interrupt Status Register	2.33
Host Controller Version Register	2.34
AHB Config Register	2.35
Power Switching Register	2.36
Tuning Setting Register	2.37
Tuning Status Register	0
Power Switching Interrupt Status Register	2.39
Power Switching Interrupt Status Enable Register	2.40
Power Switching Interrupt Signal Enable Register	2.41
MMC/eSD Control Register	2.42
MMC Wait IRQ Control Register	2.43
MMC Wait IRQ Control Register	2.44
MMC Response Check Bit Register	2.45
Card Detect Setting Register	2.46

The addresses 0x000 to 0x0FF is the Standard Register defined in "SD Specifications Part A2".

<Notes>

- The write access to a reserved area is prohibited.
- The attributes of the registers of the SD card interface use the register attributes defined in "SD Specifications Part A2". For the respective meanings of the register attributes, refer to "SD Specifications Part A2".

2.1. SDMA System Address / Argument 2 Register

This register contains the physical system memory address used for DMA transfers or the second argument for the Auto CMD23.

D31	D00
SDMA System Address / Argument 2	

For details of this register, refer to "SD Specifications Part A2".

2.2. Block Size Register

This register is used to configure the number of bytes in a data block.

D15	D14 D12	D11 D00
Rsvd	Host SDMA Buffer Boundary (*Note 1)	Transfer Block Size

*Note 1:

The write access to Host SDMA Buffer Boundary is enabled only when the SDMA Support bit in the Capabilities Register is "1".

For other details of this register, refer to "SD Specifications Part A2".

2.3. Block Count Register

This register is used to configure the number of data blocks.

D15	D00
Blocks Count For Current Transfer (*Note 1)	

*Note 1:

This register is divided into two registers, one for writing and another for reading.

The register for writing keeps the current value until the next access is made to this register, or a hardware reset is generated or a software reset is generated.

In addition, the read value of this register is the value of the register for reading. The content in the register for writing is written to the register for reading at the start of data transfer. The block count in the register for reading decreases whenever the transfer of 1 block of data ends.

Since the register for reading is controlled by SD_BCLK_I, if it is not synchronized with A_HCLK_I or SD_BCLK_I, a value of this register read during a transfer (The Read/Write Transfer Active bit in the Present State Register is "1".) may not be correct.

For other details of this register, refer to "SD Specifications Part A2".

2.4. Argument 1 Register

This register contains the SD Command Argument.

D31	D00
Command Argument	

For details of this register, refer to "SD Specifications Part A2".

2.5. Transfer Mode Register

This register is used to control the operation of data transfers.

D15 D06	D05	D04	D03 D02	D01	D00
Rsvd	Multi / Single Block Select	Data Transfer Direction Select	Auto Command Enable	Block Count Enable	DMA Enable

For details of this register, refer to "SD Specifications Part A2".

2.6. Command Register

The Host Driver shall check the Command Inhibit (DAT) bit and Command Inhibit (CMD) bit in the Present State Register before writing to this register. Writing to the upper byte of this register triggers SD command generation. The Host Driver has the responsibility to write this register because the Host Controller does not protect for writing when the Command Inhibit (CMD) bit is set to "1".

D15 D14	D13 D08	D07 D06	D05	D04	D03	D02	D01 D00
Rsvd	Command Index	Command Type (*Note 1)	Data Present Select	Command Index Check Enable	Command CRC Check Enable	Rsvd	Response Type Select

*Note 1:

If an Asynchronous Abort is executed while a data transfer for which Auto CMD12 Enable is selected in the Auto CMD Enable bit in the Transfer Mode Register is in progress, the operation of this macro is explained below.

- If the macro recognizes the trigger of the Asynchronous Abort before issuing the Auto CMD12, it issues an Abort Command using the Asynchronous Abort. After having received the response to the Abort Command, the macro sets the Command Complete bit in the Normal Interrupt Status Register.
- The macro ignores the trigger of the Asynchronous Abort while issuing Auto CMD12. As the Abort Command is an Auto Abort Command, the macro does not set the Command Complete bit in the Normal Interrupt Status Register to "1" even after having received the response to the Abort Command. In this situation, the completion of issuing the Abort Command is confirmed if the Transfer Complete bit in the Normal Interrupt Status Register is set to "1". Therefore, continue a data transfer until one of the conditions for setting the Transfer Complete bit is met.

Moreover, if one of the situations mentioned above occurs, execute the software reset in the Asynchronous Abort sequence after having issued the Abort Command.

For other details of this register, refer to "SD Specifications Part A2".

2.7. Response Register

This register is used to store responses from SD cards.

D31	D00
Command Response 0 - 31	

D31	D00
Command Response 32 - 63	

D31	D00
Command Response 64 - 95	

D31	D00
Command Response 96 - 127	

For details of this register, refer to "SD Specifications Part A2".

2.8. Buffer Data Port Register

This is a 32-bit data port register for accessing the internal buffer.

D31	D00
Buffer Data	

For details of this register, refer to "SD Specifications Part A2".

2.9. Present State Register

The Host Driver can get status of the Host Controller from this 32-bit read-only register.

D31 D25				D24	D23 D20	D19	D18	D17	D16
Rsvd				CMD Line Signal Level	DAT[3:0] Line Signal Level	Write Protect Switch Pin Level	Card Detect Pin Level	Card State Stable	Card Inserted
D15 D12	D11	D10	D09	D08	D07 D04	D03	D02	D01	D00
Rsvd	Buffer Read Enable	Buffer Write Enable	Read Transfer Active	Write Transfer Active	Rsvd	Re-Tuning Request	DAT Line Active	Command Inhibit (DAT)	Command Inhibit (CMD)

For details of this register, refer to "SD Specifications Part A2".

2.10. Host Control 1 Register

This register is used to control the Host Controller.

D07	D06	D05	D04 D03	D02	D01	D00
Card Detect Signal Selection	Card Detect Test Level	Extended Data Transfer Width (*Note 2)	DMA Select (*Note 1)	High Speed Enable	Data Transfer Width	LED Control

*Note 1:

Only when the ADMA2 Support bit in the Capabilities Register is "1" can "0b10" be written to the DMA Select bits.

*Note 2:

If Slot Type bits in the Capabilities Register are "0b10", the Extended Data Transfer Width bit reflects the value of the Bus Width Present bit in the Embedded Control Register.

For other details of this register, refer to "SD Specifications Part A2".

2.11. Power Control Register

This register is used to control the SD Bus voltage.

D07 D04	D03 D01	D00
Rsvd	SD Bus Voltage Select	SD Bus Power (*Note 1)

*Note 1:

This product enters the power on state it one of the following conditions is met.

1. "0b111" (setting 3.3 V) is written to the SD Bus Voltage Select bits.
2. "0b110" (setting 3.0 V) is written to the SD Bus Voltage Select bits.

For other details of this register, refer to "SD Specifications Part A2".

2.12. Block Gap Control Register

This register is used to control the block gap.

D07 D04	D03	D02	D01	D00
Rsvd	Interrupt At Block Gap	Read Wait Control	Continue Request	Stop At Block Gap Request

For details of this register, refer to "SD Specifications Part A2".

2.13. Wakeup Control Register

This register is mandatory for the Host Controller, but wakeup functionality depends on the Host Controller system hardware and software. The Host Driver shall maintain voltage on the SD Bus, by setting the SD Bus Power bit to "1" in the Power Control Register, when a wakeup event via a Card Interrupt is desired.

D07 D03	D02	D01	D00
Rsvd	Wakeup Event Enable On SD Card Removal	Wakeup Event Enable On SD Card Insertion	Wakeup Event Enable On SD Card Interrupt

For details of this register, refer to "SD Specifications Part A2".

2.14. Clock Control Register

At the initialization of the Host Controller, the Host Driver shall set the SDCLK Frequency Select bits according to the setting of the Capabilities Register.

D15 D08	D07 D06	D05	D04 D03	D02	D01	D00
SDCLK Frequency Select	Upper Bits of SDCLK Frequency Select	Clock Generator Select	Rsvd	SD Clock Enable	Internal Clock Stable	Internal Clock Enable

<table>
<tr><th>Bit</th><th>Attribute</th><th>Description</th></tr>
<tr><td>02</td><td>RW</td><td><b>SD Clock Enable</b>
This bit controls the output of the SD Clock.
After "0" has been written to this bit, a period lasting for a total of 3 bus clock cycles and 2 SD Clock cycles elapses before the SD clock stops.
When enabling the SD clock by changing the setting of this bit from "0" to "1", take account of the period mentioned above.
"1": Enable
"0": Disable
<table>
<tr><th>Set/Reset</th><th>Condition</th></tr>
<tr><td>Set</td><td>1) "1" Write</td></tr>
<tr><td>Reset ("0")</td><td>1) System reset
2) Software Reset For All
3) "0" Write</td></tr>
</table></td></tr>
<tr><td>01</td><td>ROC</td><td><b>Internal Clock Stable</b>
The Host Driver cannot write a value to the SD Clock Enable bit until this bit is set to "1".
"1": Ready
"0": Non Ready
<table>
<tr><th>Set/Reset</th><th>Condition</th></tr>
<tr><td>Set</td><td>1) The Internal Clock Enable bit is "1" and the Internal Clock becomes stable.</td></tr>
<tr><td>Reset ("0")</td><td>1) System reset
2) Software Reset For All
3) Other than Set condition 1), Reset condition 1) and Reset condition 2)</td></tr>
</table></td></tr>
<tr><td>00</td><td>RW</td><td><b>Internal Clock Enable</b>
To control the SD card interface, set this bit to "1" before starting the internal clock.
"1": Oscillate
"0": Stop
<table>
<tr><th>Set/Reset</th><th>Condition</th></tr>
<tr><td>Set</td><td>1) "1" Write</td></tr>
<tr><td>Reset ("0")</td><td>1) System reset
2) Software Reset For All
3) "0" Write</td></tr>
</table></td></tr>
</table>

For details of other bits in this register, refer to "SD Specifications Part A2".

2.15. Timeout Control Register

At the initialization of the Host Controller, the Host Driver shall set the Data Timeout Counter Value bits according to the setting of the Capabilities Register.

D07 D04	D03 D00
Rsvd	Data Timeout Counter Value

For details of this register, refer to "SD Specifications Part A2".

2.16. Software Reset Register

A reset pulse is generated when "1" is written to one of the three bits, D02 to D00, in this register.

D07 D03	D02	D01	D00
Rsvd	Software Reset For DAT Line (*Note 1)	Software Reset For CMD Line (*Note 1)	Software Reset For All (*Note 1)

*Note 1:

This bit can be set to "1" only when the Internal Clock Enable bit in the Clock Control Register is "1".

For other details of this register, refer to "SD Specifications Part A2".

2.17. Normal Interrupt Status Register

The Normal Interrupt Status Enable Register affects the read access to this register, but the Normal Interrupt Signal Enable Register does not. An interrupt is generated when its interrupt signal is enabled by its corresponding bit in the Normal Interrupt Signal Enable Register and its corresponding interrupt status bit in this register is set to "1".

D15	D14	D13	D12	D11	D10	D09	D08	D07	D06	D05	D04	D03	D02	D01	D00
Error Interrupt	Rsvd		Re-Tuning Event	INT_C	INT_B	INT_A	Card Interrupt (*Note 2)	Card Removal (*Note 1)	Card Insertion (*Note 1)	Buffer Read Ready	Buffer Write Ready	DMA Interrupt	Block Gap Event	Transfer Complete	Command Complete

*Note 1:

The Card Removal bit is set to "1" after the Card Removal Status Enable bit in the Normal Interrupt Status Enable Register has been set to "1" and the Card Inserted bit in the Present State Register has changed its value. The Card Insertion bit is set to "1" after the Card Insertion Status Enable bit in the Normal Interrupt Status Enable Register has been set to "1" and the Card Inserted bit in the Present State Register has changed its value.

For example, with the Card Inserted bit in the Present State Register set to "1", even if the Card Insertion Status Enable bit in the Normal Interrupt Status Enable Register is set to "1", the Card Insertion bit does not become "1". Therefore, always check the setting of the Card Inserted bit in the Present State Register when deciding whether to execute the interrupt wait process.

*Note 2:

The Card Interrupt bit can be cleared to "0" by setting the Card Interrupt Status Enable bit in the Normal Interrupt Status Enable Register to "0".

When using the SD card interface, always set the Card Interrupt Status Enable bit in the Normal Interrupt Status Enable Register to "0" to clear the Card Interrupt bit before receiving a succeeding interrupt from the SD card.

For other details of this register, refer to "SD Specifications Part A2".

2.18. Error Interrupt Status Register

Signals defined in this register can be enabled by the Error Interrupt Status Enable Register, but not by the Error Interrupt Signal Enable Register. An interrupt is generated when its interrupt signal is enabled by its corresponding bit in the Error Interrupt Signal Enable Register and its corresponding bit in the Error Interrupt Status Enable Register is set to "1".

D15	D14	D13	D12	D11	D10	D09	D08	D07	D06	D05	D04	D03	D02	D01	D00
Rsvd	AHB Master Error	Auto CMD19 Error	Boot Acknowledge Error	Rsvd	Tuning Error	ADMA Error	Auto CMD12 Error	Current Limit Error	Data End Bit Error	Data CRC Error	Data Timeout Error	Command Index Error	Command End Bit Error	Command CRC Error	Command Timeout Error

<table>
<tr><th>Bit</th><th>Attribute</th><th>Description</th></tr>
<tr><td>15</td><td>Rsvd</td><td>There is no Vendor Specific Error Status bit.</td></tr>
<tr><td>14</td><td>ROC</td><td><b>AHB Master Error</b>
This bit reads "1" when the AHB Memory I/F has received an error acknowledge to an access output by this SD card interface.
After the AHB Memory I/F has received an error response, the SD card interface can be initialized only by the system reset.
<table>
<tr><th>Set/Reset</th><th>Condition</th></tr>
<tr><td>Set</td><td>1) This bit is set to "1" if the AHB Master Error Status Enable bit in the Error Interrupt Status Enable Register is "1" and the AHB Memory I/F receives an error acknowledge.
2) If the AHB Master Error Status Enable bit in the Error Interrupt Status Enable Register is "1" and "1" is written to the Force Event for AHB Master Error bit in the Force Event Register for Error Interrupt Status, this bit is set to "1".</td></tr>
<tr><td>Reset ("0")</td><td>1) System reset
2) The AHB Master Error Status Enable bit in the Error Interrupt Status Enable Register is "0".</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Bit</th><th>Attribute</th><th>Description</th></tr>
<tr><td>13</td><td>RW1C</td><td><b>Auto CMD19 Error</b>
In Re-Tuning Mode 3, this bit indicates that in the execution of Auto CMD19 automatically issued by this SD card interface, a timeout or a bus conflict error has occurred, or re-tuning has failed.
<table>
<tr><th>Set/Reset</th><th>Condition</th></tr>
<tr><td>Set</td><td>1) With the Auto CMD19 Error Status Enable bit in the Error Interrupt Status Enable Register set to "1" and Auto CMD19 automatically issued by the SD card interface, if the SD card interface detects a timeout or a bus conflict while receiving a tuning pattern or re-tuning fails, this bit is set to "1".
2) If the Auto CMD19 Error Status Enable bit in the Error Interrupt Status Enable Register is "1" and "1" is written to the Force Event for Auto CMD19 bit in the Force Event Register for Error Interrupt Status, this bit is set to "1".</td></tr>
<tr><td>Reset ("0")</td><td>1) System reset
2) Software Reset For All
3) "1" Write</td></tr>
</table></td></tr>
<tr><td>12</td><td>RW1C</td><td><b>Boot Acknowledge Error</b>
This bit indicates whether the boot acknowledge data that the SD card interface has received has an error.
<table>
<tr><th>Set/Reset</th><th>Condition</th></tr>
<tr><td>Set</td><td>1) With the Boot Ack Enable for MMC bit in the MMC/eSD Control Register set to "1" and the Boot Acknowledge Error Status Enable bit in the Error Interrupt Status Enable Register set to "1", if the SD card interface receives the boot acknowledge data that is not "010" on the DAT0 line, this bit is set to "1".
2) If the Boot Acknowledge Error Status Enable bit in the Error Interrupt Status Enable Register is "1" and "1" is written to the Force Event for Boot Acknowledge bit in the Force Event Register for Error Interrupt Status, this bit is set to "1".</td></tr>
<tr><td>Reset ("0")</td><td>1) System reset
2) Software Reset For All
3) "1" Write</td></tr>
</table></td></tr>
</table>

For other details of this register, refer to "SD Specifications Part A2".

2.19. Normal Interrupt Status Enable Register

Setting a status enable bit in this register to "1" enables the interrupt status corresponding to that bit.

D15	D14	D13	D12	D11	D10	D09	D08	D07	D06	D05	D04	D03	D02	D01	D00
Fixed to 0	Rsvd		Re-Tuning Event Status Enable	INT_C Status Enable	INT_B Status Enable	INT_A Status Enable	Card Interrupt Status Enable	Card Removal Status Enable	Card Insertion Status Enable	Buffer Read Ready Status Enable	Buffer Write Ready Status Enable	DMA Interrupt Status Enable	Block Gap Event Status Enable	Transfer Complete Status Enable	Command Complete Status Enable

For details of this register, refer to "SD Specifications Part A2".

2.20. Error Interrupt Status Enable Register

Setting a status enable bit in this register to "1" enables the interrupt status corresponding to that bit.

D15	D14	D13	D12	D11	D10	D09	D08	D07	D06	D05	D04	D03	D02	D01	D00
Rsvd	AHB Master Error Status Enable	Auto CMD19 Error Status Enable	Boot Acknowledge Error Status Enable	Rsvd	Tuning Error Status Enable	ADMA Error Status Enable	Auto CMD Error Status Enable	Current limit Error Status Enable	Data End Bit Error Status Enable	Data CRC Error Status Enable	Data Timeout Error Status Enable	Command Index Error Status Enable	Command End Bit Error Status Enable	Command CRC Error Status Enable	Command Timeout Error Status Enable

<table>
<tr><th>Bit</th><th>Attribute</th><th>Description</th></tr>
<tr><td>15</td><td>Rsvd</td><td>There is no Vendor Specific Error Status Enable bit.</td></tr>
<tr><td>14</td><td>RW</td><td><b>AHB Master Error Status Enable</b>
This bit enables the detection of the AHB master error by the AHB Master Error bit in the Error Interrupt Status Register.
"1": Enabled
"0": Masked
<table>
<tr><th>Set/Reset</th><th>Condition</th></tr>
<tr><td>Set</td><td>1) "1" Write</td></tr>
<tr><td>Reset ("0")</td><td>1) System reset
2) Software Reset For All
3) "0" Write</td></tr>
</table></td></tr>
<tr><td>13</td><td>RW</td><td><b>Auto CMD19 Error Status Enable</b>
This bit enables the detection of the Auto CMD19 error by the Auto CMD19 Error bit in the Error Interrupt Status Register.
"1": Enabled
"0": Masked
<table>
<tr><th>Set/Reset</th><th>Condition</th></tr>
<tr><td>Set</td><td>1) "1" Write</td></tr>
<tr><td>Reset ("0")</td><td>1) System reset
2) Software Reset For All
3) "0" Write</td></tr>
</table></td></tr>
<tr><td>12</td><td>RW</td><td><b>Boot Acknowledge Error Status Enable</b>
This bit enables the detection of the boot acknowledge error by the Boot Acknowledge Error bit in the Error Interrupt Status Register.
"1": Enabled
"0": Masked
<table>
<tr><th>Set/Reset</th><th>Condition</th></tr>
<tr><td>Set</td><td>1) "1" Write</td></tr>
<tr><td>Reset ("0")</td><td>1) System reset
2) Software Reset For All
3) "0" Write</td></tr>
</table></td></tr>
</table>

For other details of this register, refer to "SD Specifications Part A2".

2.21. Normal Interrupt Signal Enable Register

This register is used to select which interrupt status is indicate to the Host System as the interrupt.

D15	D14	D13	D12	D11	D10	D09	D08	D07	D06	D05	D04	D03	D02	D01	D00
Fixed to 0	Rsvd		Re-Tuning Event Signal Enable	INT_C Signal Enable	INT_B Signal Enable	INT_A Signal Enable	Card Interrupt Signal Enable	Card Removal Signal Enable	Card Insertion Signal Enable	Buffer Read Ready Signal Enable	Buffer Write Ready Signal Enable	DMA Interrupt Signal Enable	Block Gap Event Signal Enable	Transfer Complete Signal Enable	Command Complete Signal Enable

For details of this register, refer to "SD Specifications Part A2".

2.22. Error Interrupt Signal Enable Register

This register is used to select which interrupt status is notified to the Host System as the interrupt.

D15	D14	D13	D12	D11	D10	D09	D08	D07	D06	D05	D04	D03	D02	D01	D00
Rsvd	AHB Master Error Signal Enable	Auto CMD19 Error Signal Enable	Boot Acknowledge Error Signal Enable	Rsvd	Tuning Error Signal Enable	ADMA Error Signal Enable	Auto CMD Error Signal Enable	Current limit Error Signal Enable	Data End Bit Error Signal Enable	Data CRC Error Signal Enable	Data Timeout Error Signal Enable	Command Index Error Signal Enable	Command End Bit Error Signal Enable	Command CRC Error Signal Enable	Command Timeout Error Signal Enable

<table>
<tr><th>Bit</th><th>Attribute</th><th>Description</th></tr>
<tr><td>15</td><td>Rsvd</td><td>There is no Vendor Specific Error Signal Enable bit.</td></tr>
<tr><td>14</td><td>RW</td><td><b>AHB Master Error Signal Enable</b>
This bit enables the output of the interrupt for the AHB master error detected by the AHB Master Error bit in the Error Interrupt Status Register.
"1": Enabled
"0": Masked
<table>
<tr><th>Set/Reset</th><th>Condition</th></tr>
<tr><td>Set</td><td>1) "1" Write</td></tr>
<tr><td>Reset ("0")</td><td>1) System reset
2) Software Reset For All
3) "0" Write</td></tr>
</table></td></tr>
<tr><td>13</td><td>RW</td><td><b>Auto CMD19 Error Signal Enable</b>
This bit enables the output of the interrupt for the Auto CMD19 error detected by the Auto CMD19 Error bit in the Error Interrupt Status Register.
"1": Enabled
"0": Masked
<table>
<tr><th>Set/Reset</th><th>Condition</th></tr>
<tr><td>Set</td><td>1) "1" Write</td></tr>
<tr><td>Reset ("0")</td><td>1) System reset
2) Software Reset For All
3) "0" Write</td></tr>
</table></td></tr>
<tr><td>12</td><td>RW</td><td><b>Boot Acknowledge Error Signal Enable</b>
This bit enables the output of the interrupt for the boot acknowledge error detected by the Boot Acknowledge Error bit in the Error Interrupt Status Register.
"1": Enabled
"0": Masked
<table>
<tr><th>Set/Reset</th><th>Condition</th></tr>
<tr><td>Set</td><td>1) "1" Write</td></tr>
<tr><td>Reset ("0")</td><td>1) System reset
2) Software Reset For All
3) "0" Write</td></tr>
</table></td></tr>
</table>

For other details of this register, refer to "SD Specifications Part A2".

2.23. Auto CMD Error Status Register

This register is used to indicate the CMD12 response error of Auto CMD12 and the CMD23 response error of Auto CMD23.

D15 D08	D07	D06 D05	D04	D03	D02	D01	D00
Rsvd	Command Not Issued by Auto CMD12 Error	Rsvd	Auto CMD Index Error	Auto CMD End Bit Error	Auto CMD CRC Error	Auto CMD Timeout Error	Auto CMD12 not executed

For details of this register, refer to "SD Specifications Part A2".

2.24. Host Control 2 Register

This register is used to control the Host Controller.

D15	D14	D13 D08	D07	D06	D05 D04	D03	D02 D00
Preset Value Enable	Asynchronous Interrupt Enable (*Note 1)	Rsvd	Sampling Clock Select	Execute Tuning (*Note 2)	Driver Strength Select	1.8V Signaling Enable	UHS Mode Select

*Note 1:

Regardless of the setting of this bit, the SD card interface handles all interrupts of interrupt periods except the interrupt period at the data block gap as asynchronous interrupts. Therefore, the interrupt reception control logic does not use the setting of this bit.

*Note 2:

Before setting the Execute Tuning bit in Divided Clock Mode, set the SDCLK Frequency Select bits in the Clock Control Register to "0x00".

For other details of this register, refer to "SD Specifications Part A2".

2.25. Capabilities Register

This register provides the Host Driver with information specific to the Host Controller implementation.

D63 D56	D55 D48
Rsvd	Clock Multiplier

D47 D46	D45	D44	D43 D40	D39	D38	D37	D36	D35	D34	D33	D32
Re-Tuning Modes	Use Tuning for SDR50	Rsvd	Timer Count for Re-Tuning	Rsvd	Driver Type D Support	Driver Type C Support	Driver Type A Support	Rsvd	DDR50 Support	SDR104 Support	SDR50 Support

D31 D30	D29	D28	D27	D26	D25	D24	D23	D22	D21	D20	D19	D18	D17 D16
Slot Type	Asynchronous Interrupt Support	64-bit System Bus Support	Rsvd	Voltage Support 1.8V	Voltage Support 3.0V	Voltage Support 3.3V	Suspend/Resume Support	SDMA Support	High Speed Support	Rsvd	ADMA2 Support	8-bit Support for Embedded Device	Max Block Length

D15 D08	D07	D06	D05 D00
Base Clock Frequency For SD Clock	Timeout Clock Unit	Rsvd	Timeout Clock Frequency

<table>
<tr><th>Bit</th><th>Attribute</th><th>Description</th></tr>
<tr><td>63-56</td><td>Rsvd</td><td>Reserved</td></tr>
<tr><td>55-48</td><td>HwInit</td><td><b>Clock Multiplier</b>
This field indicates the multiplier to be used by the clock generator in Programmable Clock Mode.
*: This Family does not use this field.
<table>
<tr><th>Bit value</th><th>Clock multiplier</th></tr>
<tr><td>0x00</td><td>Clock Multiplier is NOT Supported</td></tr>
<tr><td>0x01</td><td>Clock Multiplier M = 2</td></tr>
<tr><td>0x02</td><td>Clock Multiplier M = 3</td></tr>
<tr><td>...</td><td></td></tr>
<tr><td>0xFF</td><td>Clock Multiplier M = 26</td></tr>
</table></td></tr>
<tr><td>47-46</td><td>HwInit</td><td><b>Re-Tuning Modes</b>
This field selects a Re-Tuning Mode.
*: This Family does not use this field.
<table>
<tr><th>Bit value</th><th>Clock multiplier</th></tr>
<tr><td>0b00</td><td>Re-Tuning Mode 1</td></tr>
<tr><td>0b01</td><td>Re-Tuning Mode 2</td></tr>
<tr><td>0b10</td><td>Re-Tuning Mode 3</td></tr>
<tr><td>0b11</td><td>Reserved</td></tr>
</table></td></tr>
<tr><td>45</td><td>HwInit</td><td><b>Use Tuning for SDR50</b>
This bit is set to "1" if the Host requires tuning to operate SDR50.
The value of the CR_TUNSDR50_I pin is reflected in this bit.
"1": SDR50 (tuning required)
"0": SDR50 (tuning not required)
*: This Family does not use this field.</td></tr>
<tr><td>44</td><td>Rsvd</td><td>Reserved</td></tr>
<tr><td>43-40</td><td>HwInit</td><td><b>Timer Count for Re-Tuning</b>
This field indicates the value of the Re-Tuning Timer.
*: This Family does not use this field.
<table>
<tr><th>Bit value</th><th>Value of Re-Tuning Timer</th></tr>
<tr><td>0x0</td><td>Re-Tuning Timer disabled</td></tr>
<tr><td>0x1</td><td>1 second</td></tr>
<tr><td>0x2</td><td>2 seconds</td></tr>
<tr><td>0x3</td><td>4 seconds</td></tr>
<tr><td>0x4</td><td>8 seconds</td></tr>
<tr><td>...</td><td></td></tr>
<tr><td>N</td><td>$2^{(n-1)}$ seconds</td></tr>
<tr><td>...</td><td></td></tr>
<tr><td>0xB</td><td>1024 seconds</td></tr>
<tr><td>0xC to 0xE</td><td>Reserved</td></tr>
<tr><td>0xF</td><td>Get information from other source</td></tr>
</table></td></tr>
<tr><td>39</td><td>Rsvd</td><td>Reserved</td></tr>
<tr><td>38</td><td>HwInit</td><td><b>Driver Type D Support</b>
This bit indicates whether Driver Type D is supported in UHS-I.
"1": Driver Type D is supported.
"0": Driver Type D is not supported.</td></tr>
<tr><td>37</td><td>HwInit</td><td><b>Driver Type C Support</b>
This bit indicates whether Driver Type C is supported in UHS-I.
"1": Driver Type C is supported.
"0": Driver Type C is not supported.</td></tr>
</table>

Bit	Attribute	Description
36	HwInit	**Driver Type A Support** This bit indicates whether Driver Type A is supported in UHS-I. "1": Driver Type A is supported. "0": Driver Type A is not supported.
35	Rsvd	Reserved
34	HwInit	**DDR50 Support** "1": DDR50 is supported. "0": DDR50 is not supported.
33	HwInit	**SDR104 Support** "1": SDR104 is supported. "0": SDR104 is not supported.
32	HwInit	**SDR50 Support** "1": SDR50 is supported. "0": SDR50 is not supported.
31-30	HwInit	**Slot Type** This field indicates the type of slot used. "0b00": Removable Card Slot "0b01": Embedded Slot for One Device "0b10": Shared Bus Slot "0b11": Reserved If the Shared Bus Slot is used (0b10), the SD card interface supports the Shared Bus Control Register.
29	HwInit	**Asynchronous Interrupt Support** The value of this bit is fixed at "1". The asynchronous interrupt is supported.
28	HwInit	**64-bit System Bus Support** The value of this bit is fixed at "0". The 64-bit system bus support is not supported.
27	Rsvd	Reserved
26	HwInit	**Voltage Support 1.8V** "1": 1.8 V power supply is supported. "0": 1.8 V power supply is not supported.
25	HwInit	**Voltage Support 3.0V** "1": 3.0 V power supply is supported. "0": 3.0 V power supply is not supported.
24	HwInit	**Voltage Support 3.3V** "1": 3.3 V power supply is supported. "0": 3.3 V power supply is not supported.
23	HwInit	**Suspend/Resume Support** The value of this bit is fixed at "1". The Suspend/Resume function is supported.
22	HwInit	**SDMA Support** "1": SDMA is supported. "0": SDMA is not supported.
21	HwInit	**High Speed Support** The value of this bit is fixed at "1". The High Speed mode is supported.
20	Rsvd	Reserved
19	HwInit	**ADMA2 Support** "1": ADMA2 is supported. "0": ADMA2 is not supported.
18	HwInit	**8-bit Support for Embedded Device** "1": 8-bit bus width is supported. "0": 8-bit bus width is not supported.

<table>
<tr><th>Bit</th><th>Attribute</th><th>Description</th></tr>
<tr><td>17-16</td><td>HwInit</td><td><b>Max Block Length</b>
This field indicates the maximum block size of data that can be transferred to the buffer of the Host Controller.
"0b00": 512 bytes
"0b01": 1024 bytes
"0b10": 2048 bytes
"0b11": Reserved</td></tr>
<tr><td>15-08</td><td>R HwInit</td><td><b>Base Clock Frequency For SD Clock</b>
This field indicates the base clock frequency.
The base clock frequency can be selected from 1 MHz to 255 MHz
<table>
<tr><th>Bit value</th><th>Value of Re-Tuning Timer</th></tr>
<tr><td>0xFF</td><td>255 MHz</td></tr>
<tr><td>...</td><td>...</td></tr>
<tr><td>0x02</td><td>2 MHz</td></tr>
<tr><td>0x01</td><td>1 MHz</td></tr>
<tr><td>0x00</td><td>Get information with another method</td></tr>
</table></td></tr>
<tr><td>07</td><td>HwInit</td><td><b>Timeout Clock Unit</b>
This bit indicates the unit of clock frequency for detecting the Data Timeout Error.
"0": kHz
"1": MHz</td></tr>
<tr><td>06</td><td>Rsvd</td><td>Reserved</td></tr>
<tr><td>05-00</td><td>HwInit</td><td><b>Timeout Clock Frequency</b>
This bit indicates the clock frequency for detecting the Data Timeout Error.
Not "0": 1 kHz to 63 kHz or 1 MHz to 63 MHz
"000000": Get information with another method</td></tr>
</table>

2.26. Maximum Current Capabilities Register

This register indicates the maximum current capability for each voltage.

<table>
<tr><th colspan="4">D63 D32</th></tr>
<tr><td colspan="4">Rsvd</td></tr>
<tr><th>D31 D24</th><th>D23 D16</th><th>D15 D08</th><th>D07 D00</th></tr>
<tr><td>Rsvd</td><td>Maximum Current for 1.8V</td><td>Maximum Current for 3.0V</td><td>Maximum Current for 3.3V</td></tr>
</table>

<table>
<tr><th>Bit</th><th>Attribute</th><th>Description</th></tr>
<tr><td>63-32</td><td>Rsvd</td><td>Reserved</td></tr>
<tr><td>31-24</td><td>Rsvd</td><td>Reserved</td></tr>
<tr><td>23-16</td><td>HwInit</td><td><b>Maximum Current for 1.8 V</b>
<table>
<tr><th>Bit value</th><th>Current Value</th></tr>
<tr><td>0</td><td>Get information via another method</td></tr>
<tr><td>1</td><td>4 mA</td></tr>
<tr><td>2</td><td>8 mA</td></tr>
<tr><td>3</td><td>12 mA</td></tr>
<tr><td>...</td><td>...</td></tr>
<tr><td>255</td><td>1020 mA</td></tr>
</table></td></tr>
<tr><td>15-08</td><td>HwInit</td><td><b>Maximum Current for 3.0 V</b>
<table>
<tr><th>Bit value</th><th>Current value</th></tr>
<tr><td>0</td><td>Get information via another method</td></tr>
<tr><td>1</td><td>4 mA</td></tr>
<tr><td>2</td><td>8 mA</td></tr>
<tr><td>3</td><td>12 mA</td></tr>
<tr><td>...</td><td>...</td></tr>
<tr><td>255</td><td>1020 mA</td></tr>
</table></td></tr>
<tr><td>07-00</td><td>HwInit</td><td><b>Maximum Current for 3.3 V</b>
<table>
<tr><th>Bit value</th><th>Current value</th></tr>
<tr><td>0</td><td>Get information via another method</td></tr>
<tr><td>1</td><td>4 mA</td></tr>
<tr><td>2</td><td>8 mA</td></tr>
<tr><td>3</td><td>12 mA</td></tr>
<tr><td>...</td><td>...</td></tr>
<tr><td>255</td><td>1020 mA</td></tr>
</table></td></tr>
</table>

2.27. Force Event Register for Auto CMD Error Status

The Force Event Register for Auto CMD Error Status is not a physically implemented register, but is an address to which the settings of the Auto CMD Error Status Register can be written.

D15 D08	D07	D06 D05	D04	D03	D02	D01	D00
Rsvd	Force Event for Command Not Issued by Auto CMD12 Error	Rsvd	Force Event for Auto CMD Index Error	Force Event for Auto CMD End Bit Error	Force Event for Auto CMD CRC Error	Force Event for Auto CMD Timeout Error	Force Event for Auto CMD12 not executed

For details of this register, refer to "SD Specifications Part A2".

2.28. Force Event Register for Error Interrupt Status

The Force Event Register for Error Interrupt Status is not a physically implemented register, but is an address to which the settings of the Error Interrupt Status Register can be written.

D15	D14	D13	D12	D11	D10	D09	D08	D07	D06	D05	D04	D03	D02	D01	D00
Rsvd	Force Event for AHB Master Error	Force Event for Auto CMD19 Error	Force Event for Boot Acknowledge Error	Rsvd	Force Event for Tuning Error	Force Event for ADMA Error	Force Event for Auto CMD12 Error	Force Event for Current limit Error	Force Event for Data End Bit Error	Force Event for Data CRC Error	Force Event for Data Timeout Error	Force Event for Command Index Error	Force Event for Command End Bit Error	Force Event for Command CRC Error	Force Event for Command Timeout Error

<table>
<tr><th>Bit</th><th>Attribute</th><th>Description</th></tr>
<tr><td>15</td><td>Rsvd</td><td>There is no Force Event for Vendor Specific Error Status bit.</td></tr>
<tr><td>14</td><td>WO</td><td><b>Force Event for AHB Master Error</b>
This bit selects whether the AHB Master Error bit in the Error Interrupt Status Register is forced to become "1".
"1": An interrupt is generated.
"0": No interrupt is generated.
<table>
<tr><th>Set/Reset</th><th>Condition</th></tr>
<tr><td>Set</td><td>1) "1" Write</td></tr>
<tr><td>Reset ("0")</td><td>1) System reset
2) Software Reset For All
3) "0" Write</td></tr>
</table></td></tr>
<tr><td>13</td><td>WO</td><td><b>Force Event for Auto CMD19 Error</b>
This bit selects whether the Auto CMD19 Error bit in the Error Interrupt Status Register is forced to become "1".
"1": An interrupt is generated.
"0": No interrupt is generated.
<table>
<tr><th>Set/Reset</th><th>Condition</th></tr>
<tr><td>Set</td><td>1) "1" Write</td></tr>
<tr><td>Reset ("0")</td><td>1) System reset
2) Software Reset For All
3) "0" Write</td></tr>
</table></td></tr>
<tr><td>12</td><td>WO</td><td><b>Force Event for Boot Acknowledge Error</b>
This bit selects whether the Boot Acknowledge Error bit in the Error Interrupt Status Register is forced to become "1".
"1": An interrupt is generated.
"0": No interrupt is generated.
<table>
<tr><th>Set/Reset</th><th>Condition</th></tr>
<tr><td>Set</td><td>1) "1" Write</td></tr>
<tr><td>Reset ("0")</td><td>1) System reset
2) Software Reset For All
3) "0" Write</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Bit</th><th>Attribute</th><th>Description</th></tr>
<tr><td>10</td><td>WO</td><td><b>Force Event for Tuning Error</b>
This bit selects whether the Tuning Error bit in the Error Interrupt Status Register is forced to become "1".
"1": An interrupt is generated.
"0": No interrupt is generated.
<table>
<tr><th>Set/Reset</th><th>Condition</th></tr>
<tr><td>Set</td><td>1) "1" Write</td></tr>
<tr><td>Reset ("0")</td><td>1) System reset
2) Software Reset For All
3) "0" Write</td></tr>
</table>
</td></tr>
</table>

For other details of this register, refer to "SD Specifications Part A2".

2.29. ADMA Error Status Register

If an ADMA Error Interrupt occurs, the ADMA Error States field in this register holds the ADMA state and the ADMA System Address Register holds the address around the error descriptor.

D07 D03	D02	D01 D00
Rsvd	ADMA Length Mismatch Error	ADMA Error States

For details of this register, refer to "SD Specifications Part A2".

2.30. ADMA System Address Register

This register contains the physical Descriptor address used for ADMA data transfer.

D63 … D00
ADMA System Address

For details of this register, refer to "SD Specifications Part A2".

2.31. Preset Value Registers

The Preset Value Registers consist of the following registers for different speed modes as shown below.

Offset	Preset Value Register	Signal Voltage
0x060	Preset Value for Initialization	3.3 V or 1.8 V
0x062	Preset Value for Default Speed	3.3 V
0x064	Preset Value for High Speed	3.3 V
0x066	Preset Value for SDR12	1.8 V
0x068	Preset Value for SDR25	1.8 V
0x6A	Preset Value for SDR50	1.8 V
0x06C	Preset Value for SDR104	1.8 V
0x06E	Preset Value for DDR50	1.8 V

Each of the 8 Preset Value Registers has the following bit configuration.

D15 D14	D13 D11	D10	D09 D00
Driver Strength Select Value	Rsvd	Clock Generator Select Value	SDCLK Frequency Select Value

For details of this register, refer to "SD Specifications Part A2".

2.32. Shared Bus Control Register

The devices on a shared bus are not intended to be controlled by the Standard Host Driver.

D31	D30 D24	D23	D22 D20	D19	D18 D16	D15	D14 D08	D07 D06	D05 D04	D03	D02 D00
Rsvd	Back-End Power Control	Rsvd	Interrupt Pin Select	Rsvd	Clock Pin Select	Rsvd	Bus Width Preset	Rsvd	Number of Interrupt Input Pins	Rsvd	Number of Clock Pins

<table>
<tr><th>Bit</th><th>Attribute</th><th>Description</th></tr>
<tr><td>31</td><td>Rsvd</td><td>Reserved</td></tr>
<tr><td>30-24</td><td>RW</td><td><b>Back-End Power Control</b>
Each bit in this field controls the back-end power supply of an embedded device.

Settings:
"0": The back-end power is off.
"1": The back-end power is supplied.
<table>
<tr><th>Set/Reset</th><th>Condition</th></tr>
<tr><td>Set</td><td>1) "1" Write</td></tr>
<tr><td>Reset ("0")</td><td>1) System reset
2) Software Reset For All
3) "0" Write</td></tr>
</table></td></tr>
<tr><td>23</td><td>Rsvd</td><td>Reserved</td></tr>
<tr><td>22-20</td><td>RW</td><td><b>Interrupt Pin Select</b>
This field enables interrupt pin inputs.

Settings:
"0b000": An interrupt is detected by the interrupt cycle.
"0bxx1": INT_A is enabled.
"0bx1x": INT_B is enabled.
"0b1xx": INT_C is enabled.
<table>
<tr><th>Set/Reset</th><th>Condition</th></tr>
<tr><td>Set</td><td>1) "1" Write</td></tr>
<tr><td>Reset ("0")</td><td>1) System reset
2) Software Reset For All
3) "0" Write</td></tr>
</table></td></tr>
<tr><td>19</td><td>Rsvd</td><td>Reserved</td></tr>
</table>

<table>
<tr><th>Bit</th><th>Attribute</th><th>Description</th></tr>
<tr><td>18-16</td><td>RW</td><td><b>Clock Pin Select</b>
This field selects a clock pin output.

Settings:
"0b000": Clock pins are disabled.
"0b001": CLK[1](SD_CLK) is selected.
"0b010": CLK[2](SD_CLK) is selected.
...: ...
"0b111": CLK[7](SD_CLK) is selected.
<table>
<tr><th>Set/Reset</th><th>Condition</th></tr>
<tr><td>Set</td><td>1) "1" Write</td></tr>
<tr><td>Reset ("0")</td><td>1) System reset
2) Software Reset For All
3) "0" Write</td></tr>
</table></td></tr>
<tr><td>15</td><td>Rsvd</td><td>Reserved</td></tr>
<tr><td>14-08</td><td>HwInit</td><td><b>Bus Width Preset</b>
Each bit in this field indicates the bus width of an embedded device.

Settings::
"0": 4-bit bus width mode
"1": 8-bit bus width mode</td></tr>
<tr><td>07-06</td><td>Rsvd</td><td>Reserved</td></tr>
<tr><td>05-04</td><td>HwInit</td><td><b>Number of Interrupt Input Pins</b>
This field indicates the number of interrupt input pins supported.

Settings:
"0b00": No interrupt input pin is supported.
"0b01": INTA is supported.
"0b10": INTA and INTB are supported.
"0b11": INTA, INTB and INTC are supported.</td></tr>
<tr><td>03</td><td>Rsvd</td><td>Reserved</td></tr>
<tr><td>02-00</td><td>HwInit</td><td><b>Number of Clock Pins</b>
This field indicates the number of clock pins supported.

Settings:
"0b000": The shared bus is not supported.
"0b001": 1 SDCLK pin is supported.
"0b010": 2 SDCLK pins are supported.
...: ...
"0b111": 7 SDCLK pins are supported.</td></tr>
</table>

2.33. Slot Interrupt Status Register

This register indicates the logical OR of the interrupt signal and wakeup signal for each slot.

D15 D08	D07 D00
Rsvd	Interrupt Signal For Each Slot

*: Since this Family does not support multiple slots, it does not have this register.

2.34. Host Controller Version Register

This register indicates the vendor version number and the specification version number.

D15 D08	D07 D00
Vendor Version Number	Specification Version Number

Bit	Attribute	Description
15-08	HwInit	**Vendor Version Number** This field indicates the vendor version number. The value of this field is fixed at "0x01".
07-00	HwInit	**Specification Version Number** This field indicates the specification version number. The value of this field is fixed at "0x02" (compliant with SD Host Specification Version 3.00).

2.35. AHB Config Register

This register controls the configuration of the AHB bus.

D31	D16
Rsvd	

D15	D07	D06	D05	D04	D03	D02	D01	D00
Rsvd		Big/Little Endian Select	BSLOCK SEL	BSLOCK	SINEN	INCRSEL		

<table>
<tr><th>Bit</th><th>Attribute</th><th>Description</th></tr>
<tr><td>31-07</td><td>Rsvd</td><td>Reserved</td></tr>
<tr><td>06</td><td>RW</td><td><b>Big/Little Endian Select</b>
This bit selects the endianness from big endian and little endian.
"0": Selects little endian.
"1": Selects big endian.
*: For this Family, set this bit to "0".
<table>
<tr><th>Set/Reset</th><th>Condition</th></tr>
<tr><td>Set</td><td>1) "1" Write
However, no value can be written to this bit during the DMA transfer.</td></tr>
<tr><td>Reset ("0")</td><td>1) System reset
2) "0" Write
However, no value can be written to this bit during the DMA transfer.</td></tr>
</table></td></tr>
<tr><td>05</td><td>RW</td><td><b>BSLOCKSEL</b>
This bit selects a method of controlling the bus lock.
"0": The bus lock is used while 1 packet (512 bytes) is being transferred.
"1": The bus lock is used during the DMA transfer.
<table>
<tr><th>Set/Reset</th><th>Condition</th></tr>
<tr><td>Set</td><td>1) "1" Write
However, no value can be written to this bit during the DMA transfer.</td></tr>
<tr><td>Reset ("0")</td><td>1) XRESET_I
2) "0" Write
However, no value can be written to this bit during the DMA transfer.</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Bit</th><th>Attribute</th><th>Description</th></tr>
<tr><td>04</td><td>RW</td><td><b>BSLOCK</b>
This bit is for using the bus lock function.
"0": Disables the bus lock function.
"1": Enables the bus lock function.
If this bit is set to "1", the operation of the bus lock selected in the BSLOCKSEL bit in this register is to be executed.
<table>
<tr><th>Set/Reset</th><th>Condition</th></tr>
<tr><td>Set</td><td>1) "1" Write
However, no value can be written to this bit during the DMA transfer.</td></tr>
<tr><td>Reset ("0")</td><td>1) XRESET_I
2) "0" Write
However, no value can be written to this bit during the DMA transfer.</td></tr>
</table></td></tr>
<tr><td>03</td><td>RW</td><td><b>SINEN</b>
This bit selects the burst type for the byte transfer (out transfer only) up to the word boundary and for the word transfer up to the fixed burst boundary selected by INCRSEL[2:0].
"0": INCR is used.
"1": SINGLE is used.
<table>
<tr><th>Set/Reset</th><th>Condition</th></tr>
<tr><td>Set</td><td>1) "1" Write
However, no value can be written to this bit during the DMA transfer.</td></tr>
<tr><td>Reset ("0")</td><td>1) XRESET_I
2) "0" Write
However, no value can be written to this bit during the DMA transfer.</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Bit</th><th>Attribute</th><th>Description</th></tr>
<tr><td>02-00</td><td>RW</td><td><b>INCRSEL</b>
This field selects the AHB fixed length burst type (INCR4/8/16) to be used in the DMA transfer.
"0b000": INCR4, INCR8 and INCR16 are not used.
"0b001": INCR4 is used.
"0b010": INCR8 is used.
"0b011": INCR8 and INCR8 are used.
"0b100": INCR16 is used.
"0b101": INCR4 and INCR16 are used.
"0b110": INCR8 and INCR16 are used.
"0b111": INCR4, INCR8 and INCR8 are used.

In an AHB fixed length burst, if the address (AM_HADDR_O) is on one of the following boundaries, depending on the transfer size, the DMA controller automatically chooses a burst type according to the following priority order from top to bottom: INCR16, INCR8, INCR4, and executes a burst transfer.

INCR16 AM_HADDR_O[5:0] = 0b000000
INCR8 AM_HADDR_O[4:0] = 0b00000
INCR4 AM_HADDR_O[3:0] = 0b0000
<table>
<tr><th>Set/Reset</th><th>Condition</th></tr>
<tr><td>Set</td><td>1) "1" Write
However, no value can be written to this bit during the DMA transfer.</td></tr>
<tr><td>Reset ("0")</td><td>1) XRESET_I
2) "0" Write
However, no value can be written to this bit during the DMA transfer.</td></tr>
</table>
</td></tr>
</table>

Example of setting SINEN and INCSEL[2:0] are given below.

Example 1)

If SINEN is "0" and INCRSEL[2:0] are "0b000", the DMA controller executes all DMA transfers using INCR (undefined length burst).

Example 2)

If SINEN is "1" and INCRSEL[2:0] are "0b000", the DMA controller executes all DMA transfers using SINGLE (single transfer).

Example 3)

If SINEN is "0" and INCRSEL[2:0] are "0b111", if the address in the AHB fixed length burst is on one of the boundaries, the DMA controller uses one of INCR4, INCR8 and INCR16.

If the AHB fixed length burst boundaries of INCR4, INCR8 and INCR16 are the same, depending on the transfer size, the DMA controller automatically chooses a burst type according to the following priority order from top to bottom: INCR16, INCR8, INCR4, and executes a DMA transfer.

If the AHB fixed length burst boundary does not fall on any of the boundaries shown above, the DMA controller uses INCR (undefined length burst) to execute a DMA transfer.

Example 4)

If SINEN is "1" and INCRSEL[2:0] are "0b111", if the address in the AHB fixed length burst is on one of the boundaries, the DMA controller uses one of INCR4, INCR8 and INCR16.

If the AHB fixed length burst boundaries of INCR4, INCR8 and INCR16 are the same, depending on the transfer size, the DMA controller automatically chooses a burst type according to the following priority order from top to bottom: INCR16, INCR8, INCR4, and executes a DMA transfer.

If the AHB fixed length burst boundary does not fall on any of the boundaries shown above, the DMA controller uses SINGLE (single transfer) to execute a DMA transfer.

2.36. Power Switching Register

This register controls power switching of the SD card interface.

<table>
<tr><th colspan="2">D31 … D16</th></tr>
<tr><td colspan="2">Rsvd</td></tr>
</table>

<table>
<tr><th>D15 … D02</th><th>D01</th><th>D00</th></tr>
<tr><td>Rsvd</td><td>I/O Register Selecting</td><td>Auto Power Switching Enable</td></tr>
</table>

<table>
<tr><th>Bit</th><th>Attribute</th><th>Description</th></tr>
<tr><td>31-02</td><td>Rsvd</td><td>Reserved</td></tr>
<tr><td>01</td><td>RW</td><td><b>I/O Register Selecting</b>
In the case of controlling power switching by software, this bit selects the register used in controlling power switching by software.
<u>*: This Family does not use this bit.</u>

"0": Selects I/O Control 2.
"1": Selects I/O Control 1.
<table>
<tr><th>Set/Reset</th><th>Condition</th></tr>
<tr><td>Set</td><td>1) "1" Write</td></tr>
<tr><td>Reset ("0")</td><td>1) System reset
2) Software Reset For All
3) "0" Write</td></tr>
</table></td></tr>
<tr><td>00</td><td>RW</td><td><b>Auto Power Switching Enable</b>
This bit selects whether voltage switching is controlled by hardware or software.
"0": Voltage switching is controlled by software.
"1": Voltage switching is controlled by hardware.
<u>*: This Family does not use this bit.</u>
<table>
<tr><th>Set/Reset</th><th>Condition</th></tr>
<tr><td>Set</td><td>1) "1" Write</td></tr>
<tr><td>Reset ("0")</td><td>1) System reset
2) Software Reset For All
3) "0" Write</td></tr>
</table></td></tr>
</table>

2.37. Tuning Setting Register

This register controls the settings for tuning.

D31 D28	D27 D24	D23 D17	D16
Rsvd	Data Timeout Counter Value For Auto Re-Tuning	Rsvd	CMD Conflict Check Disable

D15 D13	D12 D11	D10	D09	D08	D07 D00
Rsvd	Re-Tuning Range Select	Re-Tuning Tap Select	Tuning Error Border Select	Tuning Phase Select Enable	Tuning Point Select

*: This Family does not support tuning or re-tuning.

<table>
<tr><th>Bit</th><th>Attribute</th><th>Description</th></tr>
<tr><td>31-28</td><td>Rsvd</td><td>Reserved</td></tr>
<tr><td>27-24</td><td>RW</td><td><b>Data Timeout Counter Value For Auto Re-Tuning</b>
In Re-Tuning Mode 3, this field sets the data timeout counter value used in automatic re-tuning (re-tuning by CMD19) triggered by a request trigger from SD_RT_REQ_I. Details of this field are the same as those of Data Timeout Counter Value in the Timeout Control Register.
<table>
<tr><th>Setting</th><th>TMCLK (SD_TOCLK_I)</th></tr>
<tr><td>0b1111</td><td>Reserved</td></tr>
<tr><td>0b1110</td><td>SD_CLK×2^{27}</td></tr>
<tr><td>0b1101</td><td>SD_CLK×2^{26}</td></tr>
<tr><td colspan="2">...</td></tr>
<tr><td>0b0001</td><td>SD_CLK×2^{14}</td></tr>
<tr><td>0b0000</td><td>SD_CLK×2^{13}</td></tr>
</table>
<table>
<tr><th>Set/Reset</th><th>Condition</th></tr>
<tr><td>Set</td><td>1) "1" Write</td></tr>
<tr><td>Reset ("0")</td><td>1) System reset
2) Software Reset For All
3) "0" Write</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Bit</th><th>Attribute</th><th>Description</th></tr>
<tr><td>16</td><td>RW</td><td><b>CMD Conflict Check Disable</b>
This bit enables or disables the CMD bus conflict check.
*: For this Family, set this bit to "1".

"0": Enables the CMD bus conflict check.
"1": Disables the CMD bus conflict check.
<table>
<tr><th>Set/Reset</th><th>Condition</th></tr>
<tr><td>Set</td><td>1) "1" Write</td></tr>
<tr><td>Reset ("0")</td><td>1) System reset
2) Software Reset For All
3) "0" Write</td></tr>
</table></td></tr>
<tr><td>15-13</td><td>Rsvd</td><td>Reserved</td></tr>
<tr><td>12-11</td><td>RW</td><td><b>Re-Tuning Range Select</b>
This field sets the phase check range for setting the Re-Tuning Request bit in Re-Tuning Mode 2 and Re-Tuning Mode 3.
Select an appropriate range according to the value of the 8 Phase Tuning/Re-Tuning Result bits in the Tuning Status Register.
(For instance, if the value of the 8 Phase Tuning/Re-Tuning Result bits is "0b11100000", select "0b00".)
If an inappropriate range is set, the normal operation of the SD card interface is not guaranteed.

"0b00": The SD card interface checks the range of ±1.
"0b01": The SD card interface checks the range of ±2.
"0b10": The SD card interface checks the range of ±3.
"0b11": Reserved
<table>
<tr><th>Set/Reset</th><th>Condition</th></tr>
<tr><td>Set</td><td>1) "1" Write</td></tr>
<tr><td>Reset ("0")</td><td>1) System reset
2) Software Reset For All
3) Writing "0" to the Sampling Clock Select bit in the Host Control 2 Register
4) "0" Write</td></tr>
</table></td></tr>
<tr><td>10</td><td>RW</td><td><b>Re-Tuning Tap Select</b>
This bit selects the number of phases to be processed in re-tuning.
"0": 8 phases is processed in re-tuning.
"1": 3 phases (current phase and two adjacent phases) is processed in re-tuning.
This bit can be set to "1" only when 3 or more bits of the 8 Phase Tuning/Re-Tuning Result bits in the Tuning Status Register are "1".

Set this bit according to the setting of the Re-Tuning Mode bit as shown below.
Re-Tuning Mode 1: "0" or "1"
Re-Tuning Mode 2: "0"
Re-Tuning Mode 3: "0"
<table>
<tr><th>Set/Reset</th><th>Condition</th></tr>
<tr><td>Set</td><td>1) "1" Write</td></tr>
<tr><td>Reset ("0")</td><td>1) System reset
2) Software Reset For All
3) Writing "0" to the Sampling Clock Select bit in the Host Control 2 Register
4) "0" Write</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Bit</th><th>Attribute</th><th>Description</th></tr>
<tr><td>09</td><td>RW</td><td><b>Tuning Border Error Select</b>
This bit sets the border for tuning fail in tuning/re-tuning (8 phases).
"0": If the SD card interface has properly received from the SD card 3 phases or more of test pattern, it considers that tuning is successful.
"1": If the SD card interface has properly received from the SD card 1 phase or more of test pattern, it considers that tuning is successful.
If the Re-Tuning Tap Select bit is set to "1", the setting of this bit becomes invalid.
Regardless of the setting of this bit, if the SD card interface has properly received from the SD card 2 phases or more of test pattern, it considers that tuning is successful.

Set this bit according to the setting of the Re-Tuning Mode bit as shown below.
Re-Tuning Mode 1: "0" or "1"
Re-Tuning Mode 2: "0"
Re-Tuning Mode 3: "0"
<table>
<tr><th>Set/Reset</th><th>Condition</th></tr>
<tr><td>Set</td><td>"1" Write</td></tr>
<tr><td>Reset ("0")</td><td>1) System reset
2) Software Reset For All
3) Writing "0" to the Sampling Clock Select bit in the Host Control 2 Register
4) "0" Write</td></tr>
</table></td></tr>
<tr><td>08</td><td>RW</td><td><b>Tuning Phase Select Enable</b>
This bit reselects the phase for tuning/re-tuning through software.
"0": Disables reselecting the phase through software.
"1": Enables reselecting the phase through software.

In the case of determining the phase through software, after tuning/re-tuning has been completed, when there is no communication, set this bit to "1" and select the desired phase in the Tuning Point Select bit simultaneously.
<table>
<tr><th>Set/Reset</th><th>Condition</th></tr>
<tr><td>Set</td><td>1) "1" Write</td></tr>
<tr><td>Reset ("0")</td><td>1) System reset
2) Software Reset For All
3) Writing "0" to the Sampling Clock Select bit in the Host Control 2 Register
4) Automatic clearing to "0" after writing "1"</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Bit</th><th>Attribute</th><th>Description</th></tr>
<tr><td>07-00</td><td>RW</td><td><b>Tuning Point Select</b>
These bits select the phase to be used in determining the phase for tuning/re-tuning through software (Tuning Phase Select Mode bit = 1).

"0b00000001": Selects the zeroth phase.
"0b00000010": Selects the first phase.
"0b00000100": Selects the second phase.
"0b00001000": Selects the third phase.
"0b00010000": Selects the fourth phase.
"0b00100000": Selects the fifth phase.
"0b01000000": Selects the sixth phase.
"0b10000000": Selects the seventh phase.

It is prohibited to set these bits to any value not listed above.

<table><tr><th>Set/Reset</th><th>Condition</th></tr><tr><td>Set</td><td>1) "1" Write</td></tr><tr><td>Reset ("0")</td><td>1) System reset
2) Software Reset For All
3) Writing "0" to the Sampling Clock Select bit in the Host Control 2 Register
4) "0" Write</td></tr></table></td></tr>
</table>

2.38. Tuning Status Register

This register indicates the reference phase selected and re-tuning results.

D31 D24	D23 D16
Rsvd	Present Tuning point

D15 D11	D10 D08	D07 D00
Rsvd	3 Phase Re-Tuning Result	8 Phase Tuning/Re-Tuning Result

*: This Family does not support tuning or re-tuning.

<table>
<tr><th>Bit</th><th>Attribute</th><th>Description</th></tr>
<tr><td>31-24</td><td>Rsvd</td><td>Reserved</td></tr>
<tr><td>23-16</td><td>ROC</td><td><b>Present Tuning point</b>
These bits indicate the reference phase currently selected.
They are updated after tuning/re-tuning has been completed.
If the phase is changed through software, these bits are updated after the new phase has been set in the Tuning Point Select bits in the Tuning Setting Register.

"0b00000001": The zeroth phase has been selected.
"0b00000010": The first phase has been selected.
"0b00000100": The second phase has been selected.
"0b00001000": The third phase has been selected.
"0b00010000": The fourth phase has been selected.
"0b00100000": The fifth phase has been selected.
"0b01000000": The sixth phase has been selected.
"0b10000000": The seventh phase has been selected.
<table>
<tr><th>Set/Reset</th><th>Condition</th></tr>
<tr><td>Set</td><td>---</td></tr>
<tr><td>Reset ("0")</td><td>1) System reset
2) Software Reset For All
3) Writing "0" to the Sampling Clock Select bit</td></tr>
</table></td></tr>
<tr><td>15-11</td><td>Rsvd</td><td>Reserved</td></tr>
</table>

<table>
<tr><th>Bit</th><th>Attribute</th><th>Description</th></tr>
<tr><td>10-08</td><td>ROC</td><td><b>3 Phase Re-Tuning Result</b>
These bits indicate the result of re-tuning for 3 phases.
Bit 10 indicates the result of re-tuning in the phase after the current phase.
Bit 09 indicates the result in the phase currently selected.
Bit 08 indicates the result of re-tuning in the phase before the current phase.

"0": Indicates that the SD card interface could not properly receive a test pattern from the SD card.
"1": Indicates that the SD card interface has properly received a test pattern from the SD card.

These bits are valid when the Re-Tuning Tap Select bit in the Tuning Setting Register is "1".
<table>
<tr><th>Set/Reset</th><th>Condition</th></tr>
<tr><td>Set</td><td>---</td></tr>
<tr><td>Reset ("0")</td><td>1) System reset
2) Software Reset For All
3) Writing "0" to the Sampling Clock Select bit</td></tr>
</table></td></tr>
<tr><td>07-00</td><td>ROC</td><td><b>8 Phase Tuning/Re-Tuning Result</b>
These bits indicate the results of 8-phase tuning/re-tuning.
Bit 07 indicates the result in the seventh phase.
Bit 06 indicates the result in the sixth phase.
Bit 05 indicates the result in the fifth phase.
Bit 04 indicates the result in the fourth phase.
Bit 03 indicates the result in the third phase.
Bit 02 indicates the result in the second phase.
Bit 01 indicates the result in the first phase.
Bit 00 indicates the result in the zeroth phase.

"0": Indicates that the SD card interface could not properly receive a test pattern from the SD card.
"1": Indicates that the SD card interface has properly received a test pattern from the SD card.
<table>
<tr><th>Set/Reset</th><th>Condition</th></tr>
<tr><td>Set</td><td>---</td></tr>
<tr><td>Reset ("0")</td><td>1) System reset
2) Software Reset For All
3) Writing "0" to the Sampling Clock Select bit</td></tr>
</table></td></tr>
</table>

2.39. Power Switching Interrupt Status Register

This register indicates the respective statuses of 1ms wait interrupt and 5ms wait interrupt.

D31	D16
Rsvd	

D15	D02	D01	D00
Rsvd		1ms Wait Interrupt	5ms Wait Interrupt

*: This Family does not support power switching.

<table>
<tr><th>Bit</th><th>Attribute</th><th>Description</th></tr>
<tr><td>31-02</td><td>Rsvd</td><td>Reserved</td></tr>
<tr><td>01</td><td>RW1C</td><td><b>1ms Wait Interrupt</b>
This bit (*1) is used when power switching is controlled through hardware (Auto Power Switching Enable bit in the Power Switching Register = 1).
*1: If the operating frequency is 400 kHz, the 1ms wait interrupt is generated after a 1 ms wait has ended.
(If the operating frequency is 100 kHz, the 1ms wait interrupt is generated after a 4 ms wait has ended.)
<table>
<tr><th>Set/Reset</th><th>Condition</th></tr>
<tr><td>Set</td><td>1) End of 1 ms wait</td></tr>
<tr><td>Reset ("0")</td><td>1) System reset
2) Software Reset For All
3) "1" Write</td></tr>
</table></td></tr>
<tr><td>00</td><td>RW1C</td><td><b>5ms Wait Interrupt</b>
This bit (*2) is used when power switching is controlled through hardware (Auto Power Switching Enable bit in the Power Switching Register = 1).
*2: If the operating frequency is 400 kHz, the 1ms wait interrupt is generated after a 5 ms wait has ended.
(If the operating frequency is 100 kHz, the 1ms wait interrupt is generated after a 20 ms wait has ended.)
<table>
<tr><th>Set/Reset</th><th>Condition</th></tr>
<tr><td>Set</td><td>1) End of 5 ms wait</td></tr>
<tr><td>Reset ("0")</td><td>1) System reset
2) Software Reset For All
3) "1" Write</td></tr>
</table></td></tr>
</table>

2.40. Power Switching Interrupt Status Enable Register

This register enables the generation of interrupts caused by the1ms Wait Interrupt bit and the 5ms Wait Interrupt bit in the Power Switching Interrupt Status Register.

D31	D16
Rsvd	

D15	D02	D01	D00
Rsvd		1ms Wait Interrupt Status Enable	5ms Wait Interrupt Status Enable

*: This Family does not support power switching.

<table>
<tr><th>Bit</th><th>Attribute</th><th>Description</th></tr>
<tr><td>31-02</td><td>Rsvd</td><td>Reserved</td></tr>
<tr><td>01</td><td>RW</td><td><b>1ms Wait Interrupt Status Enable</b>
This bit enables the 1ms Wait Interrupt bit in the Power Switching Interrupt Status Register.

"0": Disables the generation of the interrupt caused by the 1ms Wait Interrupt bit.
"1": Enables the generation of the interrupt caused by the 1ms Wait Interrupt bit.
<table>
<tr><th>Set/Reset</th><th>Condition</th></tr>
<tr><td>Set</td><td>1) "1" Write</td></tr>
<tr><td>Reset ("0")</td><td>1) System reset
2) Software Reset For All
3) "0" Write</td></tr>
</table></td></tr>
<tr><td>00</td><td>RW</td><td><b>5ms Wait Interrupt Status Enable</b>
This bit enables the 5ms Wait Interrupt bit in the Power Switching Interrupt Status Register.

"0": Disables the generation of the interrupt caused by the 5ms Wait Interrupt bit.
"1": Enables the generation of the interrupt caused by the 5ms Wait Interrupt bit.
<table>
<tr><th>Set/Reset</th><th>Condition</th></tr>
<tr><td>Set</td><td>1) "1" Write</td></tr>
<tr><td>Reset ("0")</td><td>1) System reset
2) Software Reset For All
3) "0" Write</td></tr>
</table></td></tr>
</table>

2.41. Power Switching Interrupt Signal Enable Register

This register enables the generation of interrupts caused by the1ms Wait Interrupt bit and the 5ms Wait Interrupt bit in the Power Switching Interrupt Status Register.

D31 D16
Rsvd

D15 D02	D01	D00
Rsvd	1ms Wait Interrupt Signal Enable	5ms Wait Interrupt Signal Enable

*: This Family does not support power switching.

<table>
<tr><th>Bit</th><th>Attribute</th><th>Description</th></tr>
<tr><td>31-02</td><td>Rsvd</td><td>Reserved</td></tr>
<tr><td>01</td><td>RW</td><td><b>1ms Wait Interrupt Signal Enable</b>
This bit enables the generation of the interrupt caused by the 1ms Wait Interrupt bit in the Power Switching Interrupt Status Register.

"0": Disables the generation of the interrupt caused by the 1ms Wait Interrupt bit.
"1": Enables the generation of the interrupt caused by the 1ms Wait Interrupt bit.
<table>
<tr><th>Set/Reset</th><th>Condition</th></tr>
<tr><td>Set</td><td>1) "1" Write</td></tr>
<tr><td>Reset ("0")</td><td>1) System reset
2) Software Reset For All
3) "0" Write</td></tr>
</table></td></tr>
<tr><td>00</td><td>RW</td><td><b>5ms Wait Interrupt Signal Enable</b>
This bit enables the generation of the interrupt caused by the 5ms Wait Interrupt bit in the Power Switching Interrupt Status Register.

"0": Disables the generation of the interrupt caused by the 5ms Wait Interrupt bit
"1": Enables the generation of the interrupt caused by the 5ms Wait Interrupt bit
<table>
<tr><th>Set/Reset</th><th>Condition</th></tr>
<tr><td>Set</td><td>1) "1" Write</td></tr>
<tr><td>Reset ("0")</td><td>1) System reset
2) Software Reset For All
3) "0" Write</td></tr>
</table></td></tr>
</table>

2.42. MMC/eSD Control Register

This register controls the operations of the MMC/eSD.

D31 D19	D18	D17	D16
Rsvd	Boot Mode Enable for MMC	Boot Auto Abort Enable for MMC	Boot Ack Enable for MMC

D15 D10	D09	D08	D07 D04	D03	D02	D01	D00
Rsvd	CMD / DAT Delay Select	MMC DDR Select	Rsvd	VCCQ Control for MMC	VCC Control for MMC	RST_n Control for MMC	Lock-Reset Control for eSD

<table>
<tr><th>Bit</th><th>Attribute</th><th>Description</th></tr>
<tr><td>31-19</td><td>Rsvd</td><td>Reserved</td></tr>
<tr><td>18</td><td>RW</td><td><b>Boot Mode Enable for MMC</b>
This bit enables the boot mode of the MMC.
The execution of the boot operation in boot mode is triggered by this bit being "1" and a write access to the Command Register.
For any other communication, including the boot operation in alternative boot mode, execute such communication operation by setting this bit to "0" and making a write access to the Command Register.
In addition, while a boot operation in boot mode is being executed, an Asynchronous Abort can be executed on the boot operation in boot mode. If the Asynchronous Abort has been enabled, always executes a software reset after executing an Asynchronous Abort.

"0": Disables the boot operation in boot mode.
"1": Enables the boot operation in boot mode.
<table>
<tr><th>Set/Reset</th><th>Condition</th></tr>
<tr><td>Set</td><td>1) "1" Write</td></tr>
<tr><td>Reset ("0")</td><td>1) System reset
2) Software Reset For All
3) "0" Write</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>Bit</th><th>Attribute</th><th>Description</th></tr>
<tr><td>17</td><td>RW</td><td><b>Boot Auto Abort Enable for MMC</b>
This bit controls whether the SD card interface automatically executes the boot abort of the boot operation of the MMC.
If this bit is "1", the SD card interface automatically executes boot abort of the boot operation and alternative boot operation.
With the Block Count Enable bit set to "1", the SD card interface aborts the boot operation after completing the transfer of data of the block count specified. After having detected an end attribute in an ADMA transfer, the SD card interface aborts the boot operation.

"0": The SD card does not automatically abort the boot operation.
"1": The SD card automatically aborts the boot operation.
<table>
<tr><th>Set/Reset</th><th>Condition</th></tr>
<tr><td>Set</td><td>1) "1" Write</td></tr>
<tr><td>Reset ("0")</td><td>1) System reset
2) Software Reset For All
3) "0" Write</td></tr>
</table></td></tr>
<tr><td>16</td><td>RW</td><td><b>Boot Ack Enable for MMC</b>
This bit enables or disables the boot acknowledge function of the MMC.

"0": Disables the boot acknowledge function.
"1": Enables the boot acknowledge function.
<table>
<tr><th>Set/Reset</th><th>Condition</th></tr>
<tr><td>Set</td><td>1) "1" Write</td></tr>
<tr><td>Reset ("0")</td><td>1) System reset
2) Software Reset For All
3) "0" Write</td></tr>
</table></td></tr>
<tr><td>15-10</td><td>Rsvd</td><td>Reserved</td></tr>
<tr><td>09</td><td>RW</td><td><b>CMD / DAT Delay Select</b>
This bit determines whether to delay CMD / DAT.
<u>*: For this Family, set this bit to "1".</u>

"0": CMD / DAT is not delayed by SD_CLK.
"1": CMD / DAT is delayed by SD_CLK.
<table>
<tr><th>Set/Reset</th><th>Condition</th></tr>
<tr><td>Set</td><td>1) "1" Write</td></tr>
<tr><td>Reset ("0")</td><td>1) System reset
2) Software Reset For All
3) "0" Write</td></tr>
</table></td></tr>
<tr><td>08</td><td>RW</td><td><b>MMC DDR Select</b>
This bit determines whether the SD card interface executes the DDR transfer on the MMC.
To execute a data transfer other than the DDR transfer on the MMC, set this bit to "0".
Before setting this bit to "1", set the SD Clock Enable bit in the Clock Control Register and stop SD_CLK.

"0": Selects the SDR transfer.
"1": Selects the DDR transfer.
<table>
<tr><th>Set/Reset</th><th>Condition</th></tr>
<tr><td>Set</td><td>1) "1" Write</td></tr>
<tr><td>Reset ("0")</td><td>1) XRESET_I
2) Software Reset For All
3) "0" Write</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Bit</th><th>Attribute</th><th>Description</th></tr>
<tr><td>07-04</td><td>Rsvd</td><td>Reserved</td></tr>
<tr><td>03</td><td>RW</td><td><b>VCCQ Control for MMC</b>
This bit sets the value of VCCQ for the MMC.
The value of this bit becomes the value of VCCQ.

"0": Selects "0" as the value of VCCQ.
"1": Selects "1" as the value of VCCQ.
<table>
<tr><th>Set/Reset</th><th>Condition</th></tr>
<tr><td>Set</td><td>1) "1" Write</td></tr>
<tr><td>Reset ("0")</td><td>1) System reset
2) Software Reset For All
3) "0" Write</td></tr>
</table></td></tr>
<tr><td>02</td><td>RW</td><td><b>VCC Control for MMC</b>
This bit sets the value of VCC for the MMC.
The value of this bit becomes the value of VCC.

"0": Selects "0" as the value of VCC.
"1": Selects "1" as the value of VCC.
<table>
<tr><th>Set/Reset</th><th>Condition</th></tr>
<tr><td>Set</td><td>1) "1" Write</td></tr>
<tr><td>Reset ("0")</td><td>1) System reset
2) Software Reset For All
3) "0" Write</td></tr>
</table></td></tr>
<tr><td>01</td><td>RW</td><td><b>RST_n Control for MMC</b>
This bit sets the value of RSTN for the MMC.
The value of this bit becomes the value of RSTN.

"0": Selects "0" as the value of RSTN.
"1": Selects "1" as the value of RSTN.
<table>
<tr><th>Set/Reset</th><th>Condition</th></tr>
<tr><td>Set</td><td>1) "1" Write</td></tr>
<tr><td>Reset ("0")</td><td>1) System reset
2) Software Reset For All
3) "0" Write</td></tr>
</table></td></tr>
<tr><td>00</td><td>RW</td><td><b>Lock-Reset Control for eSD</b>
This bit enables or disables the Lock-Reset function of eSD.
The value of this bit becomes the value of LOCKRST.

"0": Selects "0" as the value of LOCKRST.
"1": Enables the Lock-Reset function.
<table>
<tr><th>Set/Reset</th><th>Condition</th></tr>
<tr><td>Set</td><td>1) "1" Write</td></tr>
<tr><td>Reset ("0")</td><td>1) System reset
2) Software Reset For All
3) "0" Write</td></tr>
</table></td></tr>
</table>

2.43. MMC Wait IRQ Control Register

This register controls the operations of the MMC/eSD.

D31	D16
Wait IRQ Cancel Response	

D15	D02	D01	D00
Rsvd		Wait IRQ State	Wait IRQ Enable

<table>
<tr><th>Bit</th><th>Attribute</th><th>Description</th></tr>
<tr><td>31-16</td><td>RW</td><td><b>Wait IRQ Cancel Response</b>
These bits define the content of R5 that cancels the Wait IRQ.
They define [15:0] in R5 (48 bits).
<table>
<tr><th>Set/Reset</th><th>Condition</th></tr>
<tr><td>Set</td><td>1) "1" Write</td></tr>
<tr><td>Reset ("0")</td><td>1) System reset
2) Software Reset For All
3) "0" Write</td></tr>
</table></td></tr>
<tr><td>15-02</td><td>Rsvd</td><td>Reserved</td></tr>
<tr><td>01</td><td>RO</td><td><b>Wait IRQ State</b>
This bit indicates whether CMD40 has been issued.
If this bit is "0", do not enable the Wait IRQ.

"0": Indicates that no CMD40 has been issued or the issuance of the command has been completed.
"1": Indicates that though CMD40 has been issued, the SD card interface is waiting for the response of CMD40.
<table>
<tr><th>Set/Reset</th><th>Condition</th></tr>
<tr><td>Set</td><td>1) Though CMD40 has been issued, the SD card interface is waiting for the response of CMD40.</td></tr>
<tr><td>Reset ("0")</td><td>1) System reset
2) Software Reset For All
3) No CMD40 has been issued, or the command has been completed.</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Bit</th><th>Attribute</th><th>Description</th></tr>
<tr><td>00</td><td>RW</td><td><b>Wait IRQ Enable</b>
This bit enables or disables the Wait IRQ.
With this bit set to "1", if a CMD is issued, the SD card interface generates the Wait IRQ.

"0": Disables the Wait IRQ.
"1": Enables the Wait IRQ.
<table>
<tr><th>Set/Reset</th><th>Condition</th></tr>
<tr><td>Set</td><td>1) "1" Write</td></tr>
<tr><td>Reset ("0")</td><td>1) System reset
2) Software Reset For All
3) "0" Write</td></tr>
</table>
</td></tr>
</table>

2.44. MMC Wait IRQ Control Register

This register defines the content of R5 that cancels the Wait IRQ.

D31	D16
Wait IRQ Cancel Response	
D15	**D0**
Wait IRQ Cancel Response	

<table>
<tr><th>Bit</th><th>Attribute</th><th>Description</th></tr>
<tr><td>31-00</td><td>RW</td><td><b>Wait IRQ Cancel Response</b>
These bits define the content of R5 that cancels the Wait IRQ.
They define [47:16] in R5 (48 bits).
If "12Fh" is written to these bits, the SD card interface issues R5.
Before writing "0x12F" to these bits, write "0x128" to [31:16].
<table>
<tr><th>Set/Reset</th><th>Condition</th></tr>
<tr><td>Set</td><td>1) "1" Write</td></tr>
<tr><td>Reset ("0")</td><td>1) System reset
2) Software Reset For All
3) "0" Write</td></tr>
</table>
</td></tr>
</table>

2.45. MMC Response Check Bit Register

This register keeps the respective check bits for response R2 and response R3.

<table>
<tr><th colspan="3">D31 D16</th></tr>
<tr><td colspan="3">Rsvd</td></tr>
<tr><th>D15 D13</th><th>D12 D07</th><th>D06 D00</th></tr>
<tr><td>Rsvd</td><td>Check Bit 2</td><td>Check Bit 1</td></tr>
</table>

<table>
<tr><th>Bit</th><th>Attribute</th><th>Description</th></tr>
<tr><td>31-13</td><td>Rsvd</td><td>Reserved</td></tr>
<tr><td>12-07</td><td>RO</td><td><b>Check Bit 2</b>
These bits keep the check bits for response R2 or response R3 when the MMC is used.

The check bits kept in these bits are [133:128] of response R2 or [45:40] of response R3.

In both cases of response R2 and response R3, after the issuance of a command has been completed, check these bits through software to verify the validity of the response.
<table>
<tr><th>Set/Reset</th><th>Condition</th></tr>
<tr><td>Set</td><td>1) The response value is "1".</td></tr>
<tr><td>Reset ("0")</td><td>1) System reset
2) Software Reset For All
3) The response value is "0".</td></tr>
</table></td></tr>
<tr><td>06-00</td><td>RO</td><td><b>Check Bit 1</b>
These bits keep the check bits for response R3 when the MMC is used.

The check bits kept in these bits are [7:1] of response R3.

In the case of response R3, after the issuance of a command has been completed, check these bits through software to verify the validity of the response.
<table>
<tr><th>Set/Reset</th><th>Condition</th></tr>
<tr><td>Set</td><td>(1) The response value is "1".</td></tr>
<tr><td>Reset ("0")</td><td>(1) System reset
(2) Software Reset For All
(3) The response value is "0".</td></tr>
</table></td></tr>
</table>

2.46. Card Detect Setting Register

This register sets the debounce period in card detection.

<table>
<tr><th>D31</th><th>D16</th></tr>
<tr><td colspan="2">Rsvd</td></tr>
</table>

D15 D12	D11 D08	D07 D00
Rsvd	Card Detect Debounce Timer Counter Value	Rsvd

<table>
<tr><th>Bit</th><th>Attribute</th><th>Description</th></tr>
<tr><td>31-12</td><td>Rsvd</td><td>Reserved</td></tr>
<tr><td>11-08</td><td>HwInit</td><td><b>Card Detect Debounce Timer Counter Value</b>
These bits set the debounce period in card detection. The debounce period is measured according to the division of SDCLK. Set the divide ratio of SD_CLK using these bits.
<table>
<tr><th>Setting</th><th>SD_SDCLK</th></tr>
<tr><td>0b1111</td><td>Reserved</td></tr>
<tr><td>0b1110</td><td>SDCLK×2^{27}</td></tr>
<tr><td>0b1101</td><td>SDCLK×2^{26}</td></tr>
<tr><td colspan="2">...</td></tr>
<tr><td>0b0001</td><td>SDCLK×2^{14}</td></tr>
<tr><td>0b0000</td><td>SDCLK×2^{25}</td></tr>
</table></td></tr>
<tr><td>07-00</td><td>Rsvd</td><td>Reserved</td></tr>
</table>

3. MMC Boot Operation

This section explains details of the MMC boot operation.

3.1. Example of Controlling Boot Mode (Using ADMA)

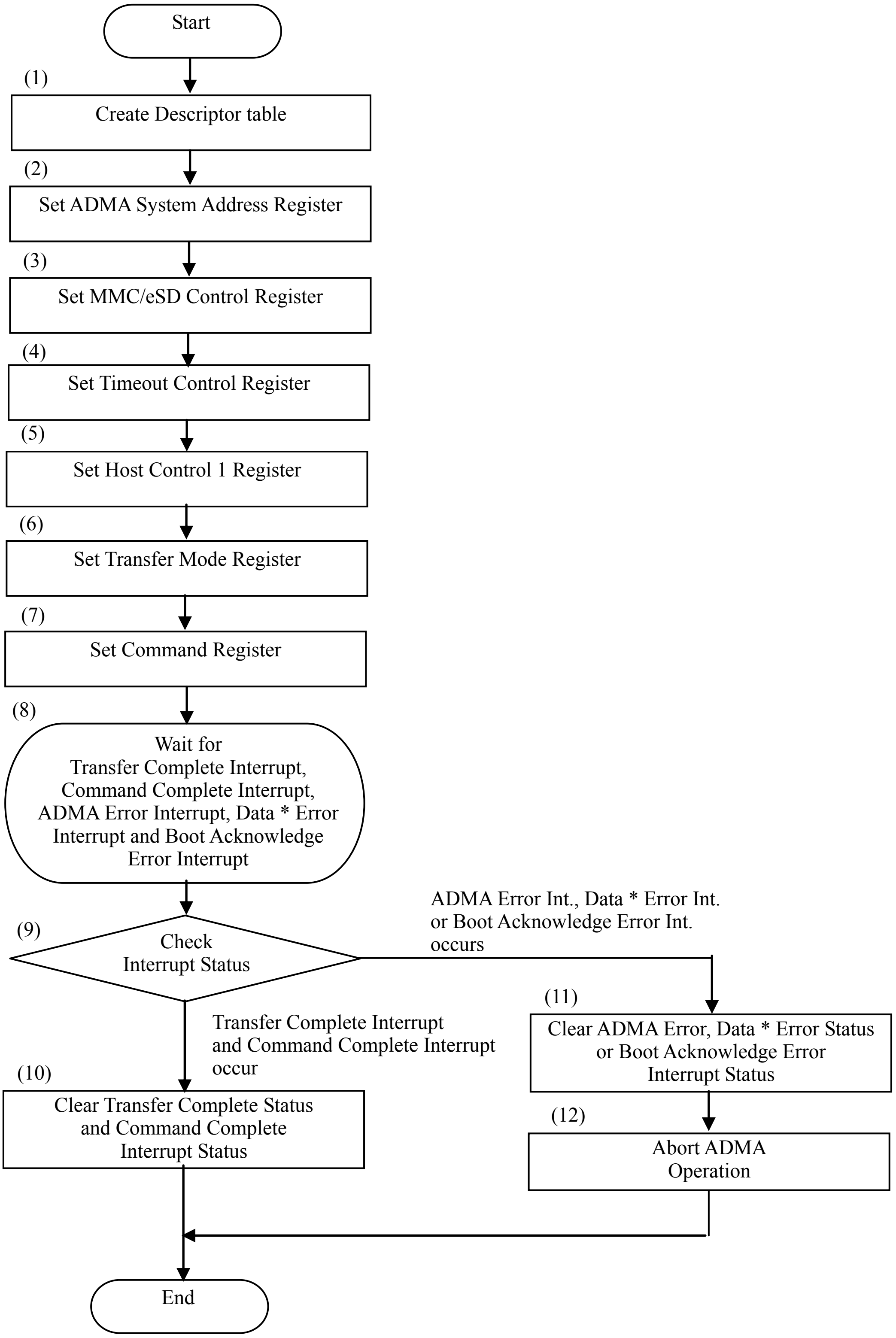

Start
(1)
Create Descriptor table
(2)
Set ADMA System Address Register
(3)
Set MMC/eSD Control Register
(4)
Set Timeout Control Register
(5)
Set Host Control 1 Register
(6)
Set Transfer Mode Register
(7)
Set Command Register
(8)
Wait for Transfer Complete Interrupt, Command Complete Interrupt, ADMA Error Interrupt, Data * Error Interrupt and Boot Acknowledge Error Interrupt
(9)
Check Interrupt Status
ADMA Error Int., Data * Error Int. or Boot Acknowledge Error Int. occurs
Transfer Complete Interrupt and Command Complete Interrupt occur
(10)
Clear Transfer Complete Status and Command Complete Interrupt Status
(11)
Clear ADMA Error, Data * Error Status or Boot Acknowledge Error Interrupt Status
(12)
Abort ADMA Operation
End

Figure 3-1 Boot Mode Operation

(1) Prepare the ADMA descriptor table in the system memory.

(2) Set the descriptor address of the ADMA in the ADMA System Address Register.

(3) Set the Boot Mode Enable for MMC bit in the MMC/eSD Control Register to "1".
In addition, to enable boot acknowledge reception, set the Boot Ack Enable for MMC bit to "1"; to complete the boot operation automatically according to the end attribute in the ADMA descriptor table, set the boot auto abort enable for MMC bit to "1".
Moreover, if a boot operation is executed with the Boot Auto Abort Enable for MMC bit set to "0", the data transfer becomes an infinite block read transfer. Setting the Boot Mode Enable for MMC bit to "0" can end the transfer.

(4) Set the Data Timeout Counter Value bits to the biggest value among tBA, tBD and NAC of timing specifications.

(5) Set the DMA Select bits to "0b10" (32-bit address ADMA2 is selected).
Set the Extended Data Transfer Width and the Data Transfer Width bit according to the type of communication.
In addition, this macro does not use the High Speed Enable bit in transfer control. (The High Speed Enable bit in this macro is meaningless.)

(6) Set the Multi / Single Block Select bit to "1" (Multiple Block).
Set the Data Transfer Direction Select bit to "1" (Read).
Set the Auto Command Enable bits to "0b00" (Auto Command Disabled).
Set the Block Count Enable bit to "0" (Disable).
Set the DMA Enable bit to "1" (DMA Data transfer).

(7) Set the Data Present Select bit to "1" (Data Present). Set all the remaining bits in the Command Register to "0".
A write access to the upper bits in the Command Register causes the boot operation to start.

(8) Wait for the interrupt to determine whether the boot operation has been completed.

(9) If the ADMA Error Interrupt, the Data Error Interrupt or the Boot Acknowledge Error Interrupt is generated, proceed to (11).
If no error interrupt is generated, but both Command Complete Interrupt and Transfer Complete Interrupt are generated, proceed to (10).
In addition, the generation of the Command Complete Interrupt and that of Transfer Complete Interrupt have no relation. (The sequence of generation of these two interrupts varies depending on the conditions of the system.)

(10) Clear the Command Complete bit and the Transfer Complete Interrupt Status bit to "0".

(11) Clear the error interrupt status bit corresponding to the interrupt generated to "0".

(12) Set the Boot Mode Enable bit in the MMC/eSD Control Register to "0" to abort the boot operation.
After aborting the boot operation, execute a software reset.

3.2. Example of Controlling Alternative Boot Mode (Using ADMA)

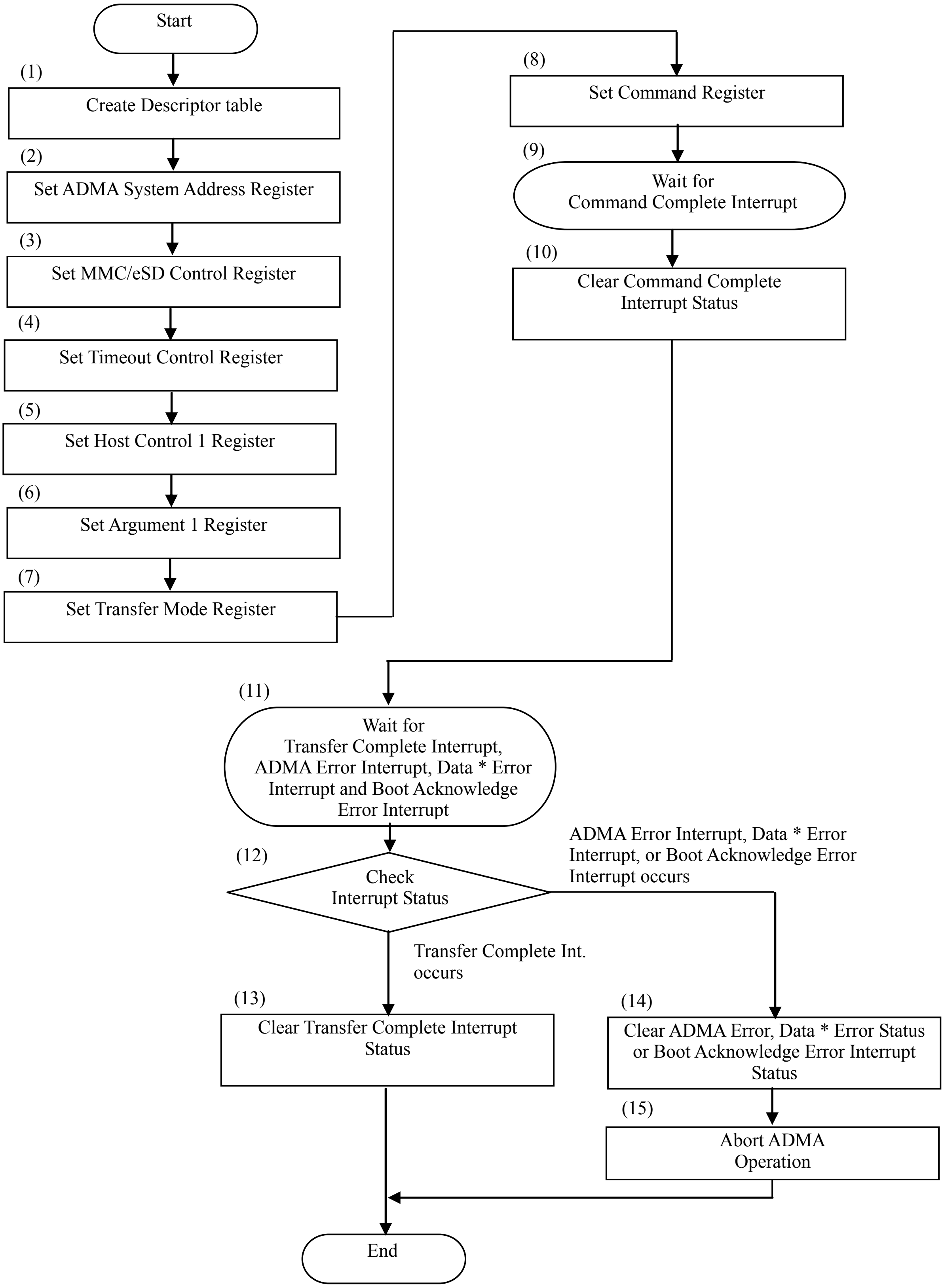

Start
(1)
Create Descriptor table
(2)
Set ADMA System Address Register
(3)
Set MMC/eSD Control Register
(4)
Set Timeout Control Register
(5)
Set Host Control 1 Register
(6)
Set Argument 1 Register
(7)
Set Transfer Mode Register
(8)
Set Command Register
(9)
Wait for Command Complete Interrupt
(10)
Clear Command Complete Interrupt Status
(11)
Wait for Transfer Complete Interrupt, ADMA Error Interrupt, Data * Error Interrupt and Boot Acknowledge Error Interrupt
(12)
Check Interrupt Status
ADMA Error Interrupt, Data * Error Interrupt, or Boot Acknowledge Error Interrupt occurs
Transfer Complete Int. occurs
(13)
Clear Transfer Complete Interrupt Status
(14)
Clear ADMA Error, Data * Error Status or Boot Acknowledge Error Interrupt Status
(15)
Abort ADMA Operation
End

Figure 3-2 Alternative Boot Mode Operation

(1) Prepare the ADMA descriptor table in the system memory.

(2) Set the descriptor address of the ADMA in the ADMA System Address Register.

(3) Set the Boot Mode Enable for MMC bit in the MMC/eSD Control Register to "0".
In addition, to enable boot acknowledge reception, set the Boot Ack Enable for MMC bit to "1"; to complete the boot operation automatically according to the end attribute in the ADMA descriptor table, set the boot auto abort enable for MMC bit to "1".
Moreover, if a boot operation is executed with the Boot Auto Abort Enable for MMC bit set to "0", the data transfer becomes an infinite block read transfer. To end the transfer, send CMD0 (Reset) as described below.
Procedure for sending CMD0 (Reset)

- Check that the Command Inhibit (CMD) bit in the Present State Register is "0".
- Set the Argument 1 Register to "0".
- Set the following bits in the Command Register as follows: the Response Type Select bits to "0b00" (No Response), the Data Select bit to "0" (No Data Present), the Command Type to "0b11" (Abort) and the Command Index bit to "0".

A write access to the upper bits in the Command Register causes CMD0 (Reset) set above to be sent.

(4) Set the Data Timeout Counter Value bits to the biggest value among tBA, tBD and NAC of timing specifications.

(5) Set the DMA Select bits to "0b10" (32-bit address ADMA2 is selected).
Set the Extended Data Transfer Width and the Data Transfer Width bit according to the type of communication.
In addition, this macro does not use the High Speed Enable bit in transfer control. (The High Speed Enable bit in this macro is meaningless.)

(6) Set to "0xFFFFFFFA".

(7) Set the Multi / Single Block Select bit to "1" (Multiple Block).
Set the Data Transfer Direction Select bit to "1" (Read).
Set the Auto Command Enable bits to "0b00" (Auto Command Disabled).
Set the Block Count Enable bit to "0" (Disable).
Set the DMA Enable bit to "1" (DMA Data transfer).

(8) Set the Response Type Select bits to "0b00" (No Response).
Set the Data Present Select bit to "1" (Data Present).
Set the Command Type bits to "0b00" (Normal).
Set the Command Index bits to "0". A write access to the upper bits in the Command Register causes the boot operation to start.

(9) Wait for the Command Complete Interrupt.

(10) Clear the Command Complete Interrupt Status bit to "0".

(11) Wait for the interrupt to determine whether the boot operation has been completed.
If the ADMA Error Interrupt, the Data Error Interrupt or the Boot Acknowledge Error Interrupt is generated, proceed to (13).
If no error interrupt is generated, but the Transfer Complete Interrupt is generated, proceed to (12).

(12) Clear the Transfer Complete Interrupt Status bit to "0".

(13) Clear the error interrupt status bit corresponding to the interrupt generated to "0".

(14) Send CMD0 (Reset) and abort the boot operation. After aborting the boot operation, execute a software reset.
See (3) for the procedure for sending CMD0 (Reset).

4. MMC Wait IRQ

This section explains details of the MMC Wait IRQ.

4.1. Example of Controlling Wait IRQ

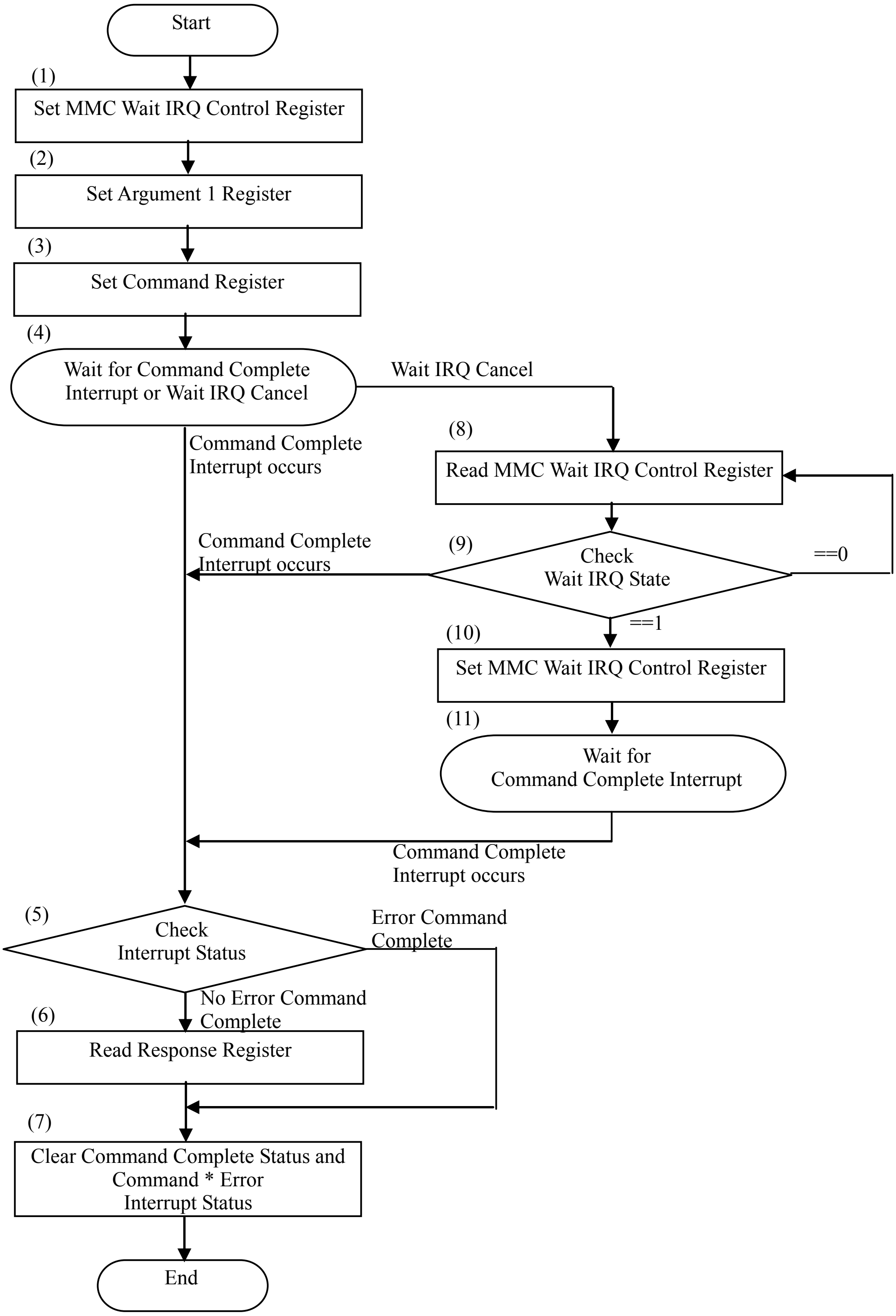

Start
(1)
Set MMC Wait IRQ Control Register
(2)
Set Argument 1 Register
(3)
Set Command Register
(4)
Wait for Command Complete Interrupt or Wait IRQ Cancel
Wait IRQ Cancel
Command Complete Interrupt occurs
(8)
Read MMC Wait IRQ Control Register
Command Complete Interrupt occurs
(9)
Check Wait IRQ State
==0
==1
(10)
Set MMC Wait IRQ Control Register
(11)
Wait for Command Complete Interrupt
Command Complete Interrupt occurs
(5)
Check Interrupt Status
Error Command Complete
No Error Command Complete
(6)
Read Response Register
(7)
Clear Command Complete Status and Command * Error Interrupt Status
End

Figure 4-1 MMC Wait Control

(1) Set the Wait IRQ Enable bit in the MMC Wait IRQ Control Register to "1" (Enable).

(2) Set the argument of CMD40 (Wait IRQ).

(3) Set the Response Type Select bits to "0b10" (Response Length 48).
Set the Command CRC Check Enable bit to "1" (Enable).
Set the Command Index Check Enable bit to "1" (Enable).
Set the Data Present Select bit to "0" (No Data Present).
Set the Command Type bits to "0b00" (Normal).
Set the Command Index bits to "40".
A write access to the upper bits in the Command Register causes the sending of CMD0 (Reset) to start.

(4) Wait for the Command Complete Interrupt, or for the cancellation of CMD40 by the Host Driver.

(5) Check whether the Command Complete Interrupt, the Command Index Error Interrupt, the Command End Bit Error Interrupt, the Command CRC Error Interrupt or the Command Timeout Error Interrupt has been generated.
If any of the Command Index Error, the Command End Bit Error Interrupt, the Command CRC Error Interrupt and the Command Timeout Error Interrupt has been generated, proceed to (7).
If none of the Command Index Error, the Command End Bit Error Interrupt, the Command CRC Error Interrupt and the Command Timeout Error Interrupt has been generated, but the Command Complete has been generated, proceed to (6).

(6) Check the response content of CMD40.

(7) Clear the interrupt status bit corresponding to the interrupt generated to "0".

(8) Start the cancellation of CMD40 by the Host Driver. Read the MMC Wait IRQ Control Register.

(9) Check the value of the Wait IRQ State bit in the MMC Wait IRQ Control Register.
The response of CMD40 can be sent from this macro only when the Wait IRQ State bit is "1".
If the Wait IRQ State bit is "1", proceed to (10) to send the response of CMD40.
The Wait IRQ State bit reading "0" means that either the sending of the CMD40 command has not been completed, or the response of CMD40 has been received. If the sending of the CMD40 command has not been completed (The Command Complete Interrupt has not been generated and the Wait IRQ State bit is "0", return to (8).
While checking the value of the Wait IRQ State bit, if the Command Complete Interrupt is generated due to the reception of the response of CMD40, abort the cancellation of the Wait IRQ and proceed to (5).

(10) Set the content of the response of CMD40 in the Wait IRQ Cancel Response bits in the MMC Wait IRQ Control Register.
Upon a write access to the Wait IRQ Cancel Response bits at the address 0x12F, output the content set in the Wait IRQ Cancel Response bits to the CMD line. In addition, the SD card interface receives output data as the response of CMD40.

CHAPTER: Debug Interface

This chapter explains the function and operation of the debug interface.

1. Overview

This family contains a Serial Wire JTAG Debug Port (SWJ-DP).

Connecting an ICE to the SWJ-DP allows system debugging.

This series also contains an Embedded Trace Macro Cell (ETM) for tracing instructions and a Trace Port Interface Unit (TPIU) that controls trace data.

This section describes the functions of the pins to be used for debugging.
For details on the SWJ-DP, ETM, TPIU and system debug, see "Cortex-M3 Technical Reference Manual".

■ Features

Five pins are assigned to the SWJ-DP.
These five pins are initially dedicated to the JTAG. It is possible to change their functions to the serial wire debug mode.

It is possible to output instruction trace by selecting it from 4-bit trace data (TRACED0 to TRACED3) and asynchronous trace data (SWO).

2. Pin Description

This section explains pins.

2.1. Pins for Debug Purposes

Five pins (TRSTX, TCK, TMS, TDI, and TDO) are assigned to the JTAG and two pins (SWCLK and SWDIO) are assigned to the serial wire. In addition, a Serial Wire Viewer signal (SWO) that outputs trace data is assigned.
TMS is shared with SWDIO, TCK is shared with SWCLK, and TDO is shared with SWO.

The following provides a list of pin functions in each debug mode.

Table 2-1 SWJ-DP pin functions in debug mode

Pin	JTAG	Serial Wire/Trace
TCK/SWCLK	TCK (JTAG Clock signal)	SWCLK (Serial Wire Clock signal)
TMS/SWDIO	TMS (JTAG State Mode signal)	SWDIO (Serial Wire Data Input/Output signal)
TDI	TDI (JTAG Data Input signal)	-
TDO/SWO	TDO (JTAG Data Output signal)	SWO (Serial Wire Viewer Output signal)
TRSTX	TRSTX (active-LOW JTAG Reset signal)	-

2.2. ETM Pins

Four trace outputs (TRACED0, TRACED1, TRACED2, and TRACED3) and one clock (TRACECLK) are assigned for ETM.

Table 2-2 shows a list of pin functions in each debug mode.

Table 2-2 Trace pin functions in debug mode

Pin	Trace
TRACED0	Synchronous Trace Data Output signal
TRACED1	Synchronous Trace Data Output signal
TRACED2	Synchronous Trace Data Output signal
TRACED3	Synchronous Trace Data Output signal
TRACECLK	Trace Clock signal

2.3. Functions Initially Assigned to Pins

The 10 pins - five JTAG pins and five ETM trace pins - are used also as GPIO.
Five JTAG pins (TRSTX, TCK, TMS, TDI, and TDO) are initially dedicated to debug function, whereas five ETM pins (TRACED0, TRACED1, TRACED2, TRACED3, and TRACECLK) are not initially dedicated to that.
When using this series, please configure these ETM pins to provide the debug function.

Note: For details on how to set the debug function, see Chapter "I/O Port"

Table 2-3 shows initial states after resets are cleared and the functions that can be changed by setting PFRs (Port function registers).

Note: For details on the PFRs, see chapter "I/O Port".

Table 2-3 Functions initially assigned to pins for debugging purposes and change of function

	Pin name	Initially assigned pin function	Change of function by setting the PFR
JTAG pins	TCK/SWCLK	TCK	GPIO
	TMS/SWDIO	TMS	GPIO
	TDI	TDI	GPIO
	TDO/SWO	TDO	GPIO
	TRSTX	TRSTX	GPIO
ETM pins	TRACED0	GPIO	TRACED0
	TRACED1	GPIO	TRACED1
	TRACED2	GPIO	TRACED2
	TRACED3	GPIO	TRACED3
	TRACECLK	GPIO	TRACECLK

2.4. Internal Pull-Ups of JTAG Pins

As specified by the IEEE Standard, this family provides the JTAG pins that have internal pull-ups.
The user can control pull-ups by setting the appropriate registers in the GPIO.

Table 2-4 Enabled or disabled state of internal pull-ups of JTAG pins

Pin name	Pull-up with JTAG pins enabled *1
TCK/SWCLK	Enabled
TMS/SWDIO	Enabled
TDI	Enabled
TDO/SWO	Enabled *2
TRSTX	Enabled

*1 : Pull-up is enabled on reset.
*2 : Pull-up is disabled on output.

CHAPTER: Flash Memory

For the flash memory, refer to the “FLASH PROGRAMMING MANUAL” of the product to be used.

CODE: 9xFLASHTOP-E01.1

CHAPTER: Unique ID Register

Functions and operations of Unique ID Register are explained as follows.

CODE: 9BFUNIQID-J01.0

1. Overview

Overview of this function is explained as follows.

41 bits of preset device unique values have been set to the Unique ID Register.

These values are different from each other in all of the devices which allow using these bits for various purposes such as security enhancement and product serial number.

This register is a read-only register which cannot be written by the user. Also, these values will not be changed due to reset or power on/off.

2. Registers

Configuration and functions of registers are explained as follows.

■ Registers list

Abbreviated Name	Register Name	Reference
UIDR0	Unique ID Register 0	2.1
UIDR1	Unique ID Register 1	2.2

2.1. Unique ID Register 0 (UIDR0: Unique ID Register 0)

Unique ID Register 0 is explained as follows.

bit	31	30	29	28	27	26	25	24
Field	UID[27:20]							
Attribute	R							

bit	23	22	21	20	19	18	17	16
Field	UID[19:12]							
Attribute	R							

bit	15	14	13	12	11	10	9	8
Field	UID[11:4]							
Attribute	R							

bit	7	6	5	4	3	2	1	0
Field	UID[3:0]				Reserved			
Attribute	R				-			

[bit31:4] UID[27:0] : Unique ID 27 through 0

Bit 27 through 0 of the unique ID.

[bit3:0] Reserved : Reserved bits

Reserved bits. Read values have no meaning.

2.2. Unique ID Register 1 (UIDR1: Unique ID Register 1)

Unique ID Register 1 is explained as follows.

bit	31	30	29	28	27	26	25	24
Field	Reserved							
Attribute	-							

bit	23	22	21	20	19	18	17	16
Field	Reserved							
Attribute	-							

bit	15	14	13	12	11	10	9	8
Field	Reserved			UID[40:36]				
Attribute	-			R				

bit	7	6	5	4	3	2	1	0
Field	UID[35:28]							
Attribute	R							

[bit31:13] Reserved : Reserved bits

Reserved bits. Read values have no meaning.

[bit12:0] UID[40:28] : Unique ID 40 through 28

Bit 40 through 28 of unique ID.

APPENDIXES

This chapter shows the register map, list of notes, limitations and product type list.

CODE: 9BFAPPENDIXES-E03.0

A. Register Map

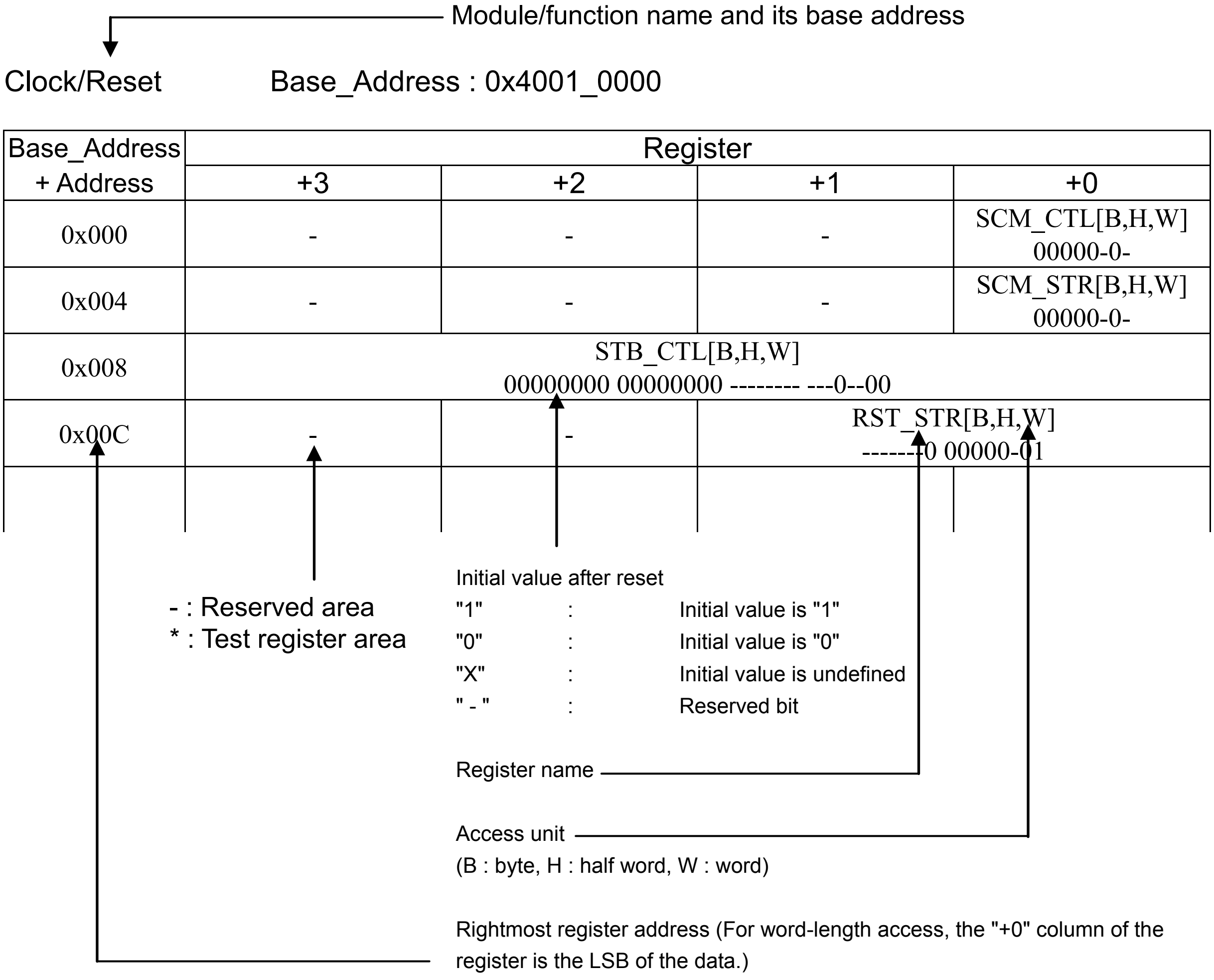
This chapter shows the register map.

CODE: 9BFREGMAP_FM4-J01.0

1. Register Map

Register map is shown on the table every module/function.

[How to read the each table]



Module/function name and its base address

Clock/Reset Base_Address : 0x4001_0000

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000	-	-	-	SCM_CTL[B,H,W] 00000-0-
0x004	-	-	-	SCM_STR[B,H,W] 00000-0-
0x008	STB_CTL[B,H,W] 00000000 00000000 -------- ---0--00			
0x00C	-	-	RST_STR[B,H,W] -------0 00000-01	

- : Reserved area
* : Test register area

Initial value after reset

"1"	:	Initial value is "1"
"0"	:	Initial value is "0"
"X"	:	Initial value is undefined
" - "	:	Reserved bit

Register name

Access unit
(B : byte, H : half word, W : word)

Rightmost register address (For word-length access, the "+0" column of the register is the LSB of the data.)

Notes:

- The register table is represented in the little-endian.
- When performing a data access, the addresses should be as below according to the access size.
  - Word access : Address should be multiples of 4 (least significant 2 bits should be "0x00")
  - Half word access : Address should be multiples of 2 (least significant bit should be "0x0")
  - Byte access : -
- Do not access the test register area.
- Do not access the area that is not written in the register table.

FLASH_IF Base_Address : 0x4000_0000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	FASZR[B,H,W]			
0x004	FRWTR[B,H,W]			
0x008	FSTR[B,H,W]			
0x00C	*			
0x010	FSYNDN[B,H,W]			
0x014	FBFCR[B,H,W]			
0x018 - 0x01C	-	-	-	-
0x020	FICR[B,H,W]			
0x024	FISR[B,H,W]			
0x028	FICLR[B,H,W]			
0x02C - 0x0FC	-	-	-	-
0x100	CRTRMM[B,H,W]			
0x104 - 0x1FC	-	-	-	-

Note:

For details of Flash I/F registers, see "FLASH PROGRAMMING MANUAL" of the product used.

Unique ID Base_Address : 0x4000_0200

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	UIDR0[W] XXXXXXXX XXXXXXXX XXXXXXXX XXXX----			
0x004	UIDR1[W] -------- -------- ---XXXXX XXXXXXXX			
0x008 - 0xDFC	-	-	-	-

ECC Capture Address Base_Address : 0x4000_0300

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	FERRAD[W] -------- -XXXXXXX XXXXXXXX XXXXXXXX			
0x004 - 0xFFC	-	-	-	-

Clock/Reset Base_Address : 0x4001_0000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	-	SCM_CTL[W] 00000-0-
0x004	-	-	-	SCM_STR[W] 00000-0-
0x008	STB_CTL[W] 00000000 00000000 -------- ---0-000			
0x00C	-	-	RST_STR[W] -------0 0000--01	
0x010	-	-	-	BSC_PSR[W] -----000
0x014	-	-	-	APBC0_PSR[W] ------00
0x018	-	-	-	APBC1_PSR[W] 1--0--00
0x01C	-	-	-	APBC2_PSR[W] 1--0--00
0x020	-	-	-	SWC_PSR[W] X-----00
0x024 – 0x027	-	-	-	-
0x028	-	-	-	TTC_PSR[W] ------00
0x02C – 0x02F	-	-	-	-
0x030	-	-	-	CSW_TMR[W] 00000000
0x034	-	-	-	PSW_TMR[W] ---0-000
0x038	-	-	-	PLL_CTL1[W] 00000000
0x03C	-	-	-	PLL_CTL2[W] --000000
0x040	-	-	CSV_CTL[W] -111--00 ------11	
0x044	-	-	-	CSV_STR[W] ------00
0x048	-	-	FCSWH_CTL[W] 11111111 11111111	
0x04C	-	-	FCSWL_CTL[W] 00000000 00000000	
0x050	-	-	FCSWD_CTL[W] 00000000 00000000	
0x054	-	-	-	DBWDT_CTL[W] 0-0-----
0x058	-	-	-	*
0x05C - 0x05F	-	-	-	-
0x060	-	-	-	INT_ENR[W] --0--000
0x064	-	-	-	INT_STR[W] --0–000
0x068	-	-	-	INT_CLR[W] --0--000
0x06C – 0xFFC	-	-	-	-

HW WDT Base_Address : 0x4001_1000

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000	WDG_LDR[W] 00000000 00000000 11111111 11111111			
0x004	WDG_VLR[W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x008	-	-	-	WDG_CTL[W] ------11
0x00C	-	-	-	WDG_ICL[W] XXXXXXXX
0x010	-	-	-	WDG_RIS[W] -------0
0x014	*			
0x018 – 0xBFC	-	-	-	-
0xC00	WDG_LCK[W] 00000000 00000000 00000000 00000001			
0xC04 – 0xFFC	-	-	-	-

SW WDT Base_Address : 0x4001_2000

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000	WdogLoad[W] 11111111 11111111 11111111 11111111			
0x004	WdogValue[W] 11111111 11111111 11111111 11111111			
0x008	-	-	-	WdogControl[W] ---00000
0x00C	WdogIntClr[W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x010	-	-	-	WdogRIS[W] -------0
0x014	*			
0x018	-	-	-	WdogSPMC[W] -------0
0x01C – 0xBFC	-	-	-	-
0xC00	WdogLock[W] 00000000 00000000 00000000 00000000			
0xC04 - 0xDFC	-	-	-	-
0xF00 - 0xF04	*			
0xF08 - 0xFDF	-	-	-	-
0xFE0 - 0xFFC	*			

Dual_Timer Base_Address : 0x4001_5000

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000	Timer1Load[W] 00000000 00000000 00000000 00000000			
0x004	Timer1Value[W] 11111111 11111111 11111111 11111111			
0x008	Timer1Control[W] -------- -------- -------- 00100000			
0x00C	Timer1IntClr[W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x010	Timer1RIS[W] -------- -------- -------- -------0			
0x014	Timer1MIS[W] -------- -------- -------- -------0			
0x018	Timer1BGLoad[W] 00000000 00000000 00000000 00000000			
0x020	Timer2Load[W] 00000000 00000000 00000000 00000000			
0x024	Timer2Value[W] 11111111 11111111 11111111 11111111			
0x028	Timer2Control[W] -------- -------- -------- 00100000			
0x02C	Timer2IntClr[W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x030	Timer2RIS[W] -------- -------- -------- -------0			
0x034	Timer2MIS[W] -------- -------- -------- -------0			
0x038	Timer2BGLoad[W] 00000000 00000000 00000000 00000000			
0x040 - 0xFFC	-	-	-	-

MFT unit0 Base_Address : 0x4002_0000
MFT unit1 Base_Address : 0x4002_1000
MFT unit2 Base_Address : 0x4002_2000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x100	OCCP0[H,W] 00000000 00000000		-	-
0x104	OCCP1[H,W] 00000000 00000000		-	-
0x108	OCCP2[H,W] 00000000 00000000		-	-
0x10C	OCCP3[H,W] 00000000 00000000		-	-
0x110	OCCP4[H,W] 00000000 00000000		-	-
0x114	OCCP5[H,W] 00000000 00000000		-	-
0x118	-	OCSD10[B,H,W] 00000000	OCSB10[B,H,W] 00000000	OCSA10[B,H,W] 00000000
0x11C	-	OCSD32[B,H, W] 00000000	OCSB32[B,H, W] 00000000	OCSA32[B,H,W] 00000000
0x120	-	OCSD54[B,H,W] 00000000	OCSB54[B,HW] 00000000	OCSA54[B,H,W] 00000000
0x124	-	-	OCSC[B,H,W] --000000	-
0x128	-	-	OCSE0[B,H,W] 00000000 00000000	
0x12C	OCSE1[B,H,W] 00000000 00000000 00000000 00000000			
0x130	-	-	OCSE2[B,H,W] 00000000 00000000	
0x134	OCSE3[B,H,W] 00000000 00000000 00000000 00000000			
0x138	-	-	OCSE4[B,H,W] 00000000 00000000	
0x13C	OCSE5[B,H,W] 00000000 00000000 00000000 00000000			
0x140	TCCP0[H,W] 11111111 11111111		-	-
0x144	TCDT0[H,W] 00000000 00000000		-	-
0x148	TCSC0[H,W] 00000000 00000000		TCSA0[B,H,W] 00000000 01000000	
0x14C	TCCP1[H,W] 11111111 11111111		-	-
0x150	TCDT1[H,W] 00000000 00000000		-	-
0x154	TCSC1[H,W] 00000000 00000000		TCSA1[B,H,W] 00000000 01000000	
0x158	TCCP2[H,W] 11111111 11111111		-	-
0x15C	TCDT2[H,W] 00000000 00000000		-	-
0x160	TCSC2[H,W] 00000000 00000000		TCSA2[B,H,W] 00000000 01000000	

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x164	TCAL[W] 00000000 00000000 11111111 11111111 *1			
	-	-	-	- *2
	*1 MFT unit0 *2 MFT unit1,unit2			
0x168	-	OCFS54[B,H,W] 00000000	OCFS32[B,H,W] 00000000	OCFS10[B,H,W] 00000000
0x16C	-	-	ICFS32[B,H,W] 00000000	ICFS10[B,H,W] 00000000
0x170	-	ACFS54[B,H,W] 00000000	ACFS32[B,H,W] 00000000	ACFS10[B,H,W] 00000000
0x174	ICCP0[H,W] 00000000 00000000		-	-
0x178	ICCP1[H,W] 00000000 00000000		-	-
0x17C	ICCP2[H,W] 00000000 00000000		-	-
0x180	ICCP3[H,W] 00000000 00000000		-	-
0x184	-	-	ICSB10[B,H,W] ------00	ICSA10[B,H,W] 00000000
0x188			ICSB32[B,H,W] ------00	ICSA32[B,H,W] 00000000
0x18C	WFTF10[H,W] 00000000 00000000		-	-
0x190	WFTB10[H,W] 00000000 00000000		WFTA10[H,W] 00000000 00000000	
0x194	WFTF32[H,W] 00000000 00000000		-	-
0x198	WFTB32[H,W] 00000000 00000000		WFTA32[H,W] 00000000 00000000	
0x19C	WFTF54[H,W] 00000000 00000000		-	-
0x1A0	WFTB54[H,W] 00000000 00000000		WFTA54[H,W] 00000000 00000000	
0x1A4	-	-	WFSA10[B,H,W] --000000 000000	
0x1A8	-	-	WFSA32[B,H,W] --000000 000000	
0x1AC	-	-	WFSA54[B,H,W] --000000 000000	
0x1B0	-	-	WFIR[H,W] 00000000 00000000	
0x1B4	-	-	NZCL[H,W] 00000000 00000000	
0x1B8	ACMP0[H,W] 00000000 00000000		-	-
0x1BC	ACMP1[H,W] 00000000 00000000		-	-
0x1C0	ACMP2[H,W] 00000000 00000000		-	-
0x1C4	ACMP3[H,W] 00000000 00000000		-	-

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x1C8	ACMP4[H,W] 00000000 00000000		-	-
0x1CC	ACMP5[H,W] 00000000 00000000		-	-
0x1D0	-	-	ACSA[B,H,W] 00000000 00000000	
0x1D4	-	-	ACSD0[B,H,W] 00000000	ACSC0[B,H,W] 00000000
0x1D8	-	-	ACSD1[B,H,W] 00000000	ACSC1[B,H,W] 00000000
0x1DC	-	-	ACSD2[B,H,W] 00000000	ACSC2[B,H,W] 00000000
0x1E0	-	-	ACSD3[B,H,W] 00000000	ACSC3[B,H,W] 00000000
0x1E4	-	-	ACSD4[B,H,W] 00000000	ACSC4[B,H,W] 00000000
0x1E8	-	-	ACSD5[B,H,W] 00000000	ACSC5[B,H,W] 00000000
0x1EC-0xFFC	-	-	-	-

PPG Base_Address : 0x4002_4000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	TTCR0 [B,H,W] 11110000	-
0x004	-	-	-	*
0x008	-	-	COMP0 [B,H,W] 00000000	-
0x00C	-	-	-	COMP2 [B,H,W] 00000000
0x010	-	-	COMP4 [B,H,W] 00000000	-
0x014	-	-	-	COMP6 [B,H,W] 00000000
0x018 - 0x01C	-	-	-	-
0x020	-	-	TTCR1 [B,H,W] 11110000	-
0x024	-	-	-	*
0x028	-	-	COMP1 [B,H,W] 00000000	-
0x02C	-	-	-	COMP3 [B,H,W] 00000000
0x030	-	-	COMP5 [B,H,W] 00000000	-
0x034	-	-	-	COMP7 [B,H,W] 00000000
0x038 - 0x03C	-	-	-	-
0x040	-	-	TTCR2 [B,H,W] 11110000	-
0x044	-	-	-	*
0x048	-	-	COMP8 [B,H,W] 00000000	-
0x04C	-	-	-	COMP10 [B,H,W] 00000000
0x050	-	-	COMP12 [B,H,W] 00000000	-
0x054	-	-	-	COMP14 [B,H,W] 00000000
0x058 - 0x0FC	-	-	-	-
0x100	-	-	TRG0 [B,H,W] 00000000 00000000	
0x104	-	-	REVC0 [B,H,W] 00000000 00000000	
0x108 - 0x13C	-	-	-	-
0x140	-	-	TRG1 [B,H,W] -------- 00000000	
0x144	-	-	REVC1 [B,H,W] -------- 00000000	
0x148 - 0x1FC	-	-	-	-

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x200	-	-	PPGC0 [B,H,W] 00000000	PPGC1 [B,H,W] 00000000
0x204	-	-	PPGC2 [B,H,W] 00000000	PPGC3 [B,H,W] 00000000
0x208	-	-	PRLH0 [B,H,W] XXXXXXXX	PRLL0 [B,H,W] XXXXXXXX
0x20C	-	-	PRLH1 [B,H,W] XXXXXXXX	PRLL1 [B,H,W] XXXXXXXX
0x210	-	-	PRLH2 [B,H,W] XXXXXXXX	PRLL2 [B,H,W] XXXXXXXX
0x214	-	-	PRLH3 [B,H,W] XXXXXXXX	PRLL3 [B,H,W] XXXXXXXX
0x218	-	-	-	GATEC0 [B,H,W] --00---00
0x21C - 0x23C	-	-	-	-
0x240	-	-	PPGC4 [B,H,W] 00000000	PPGC5 [B,H,W] 00000000
0x244	-	-	PPGC6 [B,H,W] 00000000	PPGC7 [B,H,W] 00000000
0x248	-	-	PRLH4 [B,H,W] XXXXXXXX	PRLL4 [B.H.W] XXXXXXXX
0x24C	-	-	PRLH5 [B,H,W] XXXXXXXX	PRLL5 [B,H,W] XXXXXXXX
0x250	-	-	PRLH6 [B,H,W] XXXXXXXX	PRLL6 [B,H,W] XXXXXXXX
0x254	-	-	PRLH7 [B,H,W] XXXXXXXX	PRLL7 [B,H,W] XXXXXXXX
0x258	-	-	-	GATEC4 [B,H,W] ------00
0x25C - 0x27C	-	-	-	-
0x280	-	-	PPGC8 [B,H,W] 00000000	PPGC9 [B,H,W] 00000000
0x284	-	-	PPGC10 [B,H,W] 00000000	PPGC11 [B,H,W] 00000000
0x288	-	-	PRLH8 [B,H,W] XXXXXXXX	PRLL8 [B,H,W] XXXXXXXX
0x28C	-	-	PRLH9 [B,H,W] XXXXXXXX	PRLL9 [B,H,W] XXXXXXXX
0x290	-	-	PRLH10 [B,H,W] XXXXXXXX	PRLL10 [B,H,W] XXXXXXXX
0x294	-	-	PRLH11 [B,H,W] XXXXXXXX	PRLL11 [B,H,W] XXXXXXXX
0x298	-	-	-	GATEC8 [B,H,W] --00--00
0x29C - 0x2BC	-	-	-	-
0x2C0	-	-	PPGC12 [B,H,W] 00000000	PPGC13 [B,H,W] 00000000
0x2C4	-	-	PPGC14 [B,H,W] 00000000	PPGC15 [B,H,W] 00000000
0x2C8	-	-	PRLH12 [B,H,W] XXXXXXXX	PRLL12 [B,H,W] XXXXXXXX
0x2CC	-	-	PRLH13 [B,H,W] XXXXXXXX	PRLL13 [B,H,W] XXXXXXXX

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x2D0	-	-	PRLH14 [B,H,W] XXXXXXXX	PRLL14 [B,H,W] XXXXXXXX
0x2D4	-	-	PRLH15 [B,H,W] XXXXXXXX	PRLL15 [B,H,W] XXXXXXXX
0x2D8	-	-	-	GATEC12 [B,H,W] ------00
0x2DC - 0x2FC	-	-	-	-
0x300	-	-	PPGC16 [B,H,W] 00000000	PPGC17 [B,H,W] 00000000
0x304	-	-	PPGC18 [B,H,W] 00000000	PPGC19 [B,H,W] 00000000
0x308	-	-	PRLH16 [B,H,W] XXXXXXXX	PRLL16 [B,H,W] XXXXXXXX
0x30C	-	-	PRLH17 [B,H,W] XXXXXXXX	PRLL17 [B,H,W] XXXXXXXX
0x310	-	-	PRLH18 [B,H,W] XXXXXXXX	PRLL18 [B,H,W] XXXXXXXX
0x314	-	-	PRLH19 [B,H,W] XXXXXXXX	PRLL19 [B,H,W] XXXXXXXX
0x318	-	-	-	GATEC16 [B,H,W] --00---00
0x31C - 0x33C	-	-	-	-
0x340	-	-	PPGC20 [B,H,W] 00000000	PPGC21 [B,H,W] 00000000
0x344	-	-	PPGC22 [B,H,W] 00000000	PPGC23 [B,H,W] 00000000
0x348	-	-	PRLH20 [B,H,W] XXXXXXXX	PRLL20 [B.H.W] XXXXXXXX
0x34C	-	-	PRLH21 [B,H,W] XXXXXXXX	PRLL21 [B,H,W] XXXXXXXX
0x350	-	-	PRLH22 [B,H,W] XXXXXXXX	PRLL22 [B,H,W] XXXXXXXX
0x354	-	-	PRLH23 [B,H,W] XXXXXXXX	PRLL23 [B,H,W] XXXXXXXX
0x358	-	-	-	GATEC20 [B,H,W] ------00
0x35C - 0x37C	-	-	-	-
0x380	-	-	-	-
0x384 - 0xFFC	-	-	-	-

Base Timer ch.0	Base Address : 0x4002_5000
Base Timer ch.1	Base Address : 0x4002_5040
Base Timer ch.2	Base Address : 0x4002_5080
Base Timer ch.3	Base Address : 0x4002_50C0
Base Timer ch.4	Base Address : 0x4002_5200
Base Timer ch.5	Base Address : 0x4002_5240
Base Timer ch.6	Base Address : 0x4002_5280
Base Timer ch.7	Base Address : 0x4002_52C0
Base Timer ch.8	Base Address : 0x4002_5400
Base Timer ch.9	Base Address : 0x4002_5440
Base Timer ch.10	Base Address : 0x4002_5480
Base Timer ch.11	Base Address : 0x4002_54C0
Base Timer ch.12	Base Address : 0x4002_5600
Base Timer ch.13	Base Address : 0x4002_5640
Base Timer ch.14	Base Address : 0x4002_5680
Base Timer ch.15	Base Address : 0x4002_56C0

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	PCSR/PRLL [H,W] XXXXXXXX XXXXXXXX	
0x004	-	-	PDUT/PRLH/DTBF [H,W] XXXXXXXX XXXXXXXX	
0x008	-	-	TMR [H,W] 00000000 00000000	
0x00C	-	-	TMCR [B,H,W] -0000000 00000000	
0x010	-	-	TMCR2 [B,H,W] 0------0	STC [B,H,W] 0000-000
0x014 - 0x03C	-	-	-	-

IO Selector for ch.0-ch.3 (Base Timer) Base Address : 0x4002_5100

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	BTSEL0123 [B,H,W] 00000000	-
0x004 - 0x0FC	-	-	-	-

IO Selector for ch.4-ch.7(Base Timer) Base Address : 0x4002_5300

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	BTSEL4567 [B,H,W] 00000000	-
0x004 - 0x0FC	-	-	-	-

IO Selector for ch.8-ch.11(Base Timer) Base Address : 0x4002_5500

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	BTSEL89AB [B,H,W] 00000000	-
0x004 - 0x0FC	-	-	-	-

IO Selector for ch.12-ch.15(Base Timer) Base Address : 0x4002_5700

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	BTSELCDEF [B,H,W] 00000000	-
0x004 - 0x0FC	-	-	-	-

Software-based Simulation Startup(Base Timer) Base Address : 0x4002_5F00

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000 - 0x0FB	-	-	-	-
0x0FC	-	-	BTSSSR [B,H,W] XXXXXXXX XXXXXXXX	

QPRC ch.0	Base Address : 0x4002_6000
QPRC ch.1	Base Address : 0x4002_6040
QPRC ch.2	Base Address : 0x4002_6080
QPRC ch.3	Base Address : 0x4002_60C0

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x0000	-	-	QPCR [H,W] 00000000 00000000	
0x0004	-	-	QRCR [H,W] 00000000 00000000	
0x0008	-	-	QPCCR [H,W] 00000000 00000000	
0x000C	-	-	QPRCR [H,W] 00000000 00000000	
0x0010	-	-	QMPR [H,W] 11111111 11111111	
0x0014	-	-	QICRH [B,H,W] --000000	QICRL [B,H,W] 00000000
0x0018	-	-	QCRH [B,H,W] 00000000	QCRL [B,H,W] 00000000
0x001C	-	-	QECR [B,H,W] -------- -----000	
0x0020 - 0x003B	-	-	-	-
0x003C	QPCRR[B,H,W] 00000000 00000000		QRCRR[B,H,W] 00000000 00000000	

QPRC ch.0 NF	Base Address : 0x4002_6100
QPRC ch.1 NF	Base Address : 0x4002_6110
QPRC ch.2 NF	Base Address : 0x4002_6120
QPRC ch.3 NF	Base Address : 0x4002_6130

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x0000	-	-	-	NFCTLA[B,H,W] --00-000
0x0004	-	-	-	NFCTLB[B,H,W] --00-000
0x0008	-	-	-	NFCTLZ[B,H,W] --00-000
0x000C	-	-	-	-

12bit A/DC unit0 Base_Address : 0x4002_7000
12bit A/DC unit1 Base_Address : 0x4002_7100
12bit A/DC unit2 Base_Address : 0x4002_7200

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	ADCR[B,H,W] 000-0000	ADSR[B,H,W] 00---000
0x004	-	-	-	*
0x008	-	-	SCCR[B,H,W] 1000-000	SFNS[B,H,W] ----0000
0x00C	SCFD[B,H,W] XXXXXXXX XXXX---- ---X--XX ---XXXXX			
0x010	-	-	SCIS3[B,H,W] 00000000	SCIS2[B,H,W] 00000000
0x014	-	-	SCIS1[B,H,W] 00000000	SCIS0[B,H,W] 00000000
0x018	-	-	PCCR[B,H,W] 10000000	PFNS[B,H,W] --XX--00
0x01C	PCFD[B,H,W] XXXXXXXX XXXX---- ---X-XXX ---XXXXX			
0x020	-	-	-	PCIS[B,H,W] 00000000
0x024	CMPD[B,H,W] 00000000 00------		-	CMPCR[B,H,W] 00000000
0x028	-	-	ADSS3[B,H,W] 00000000	ADSS2[B,H,W] 00000000
0x02C	-	-	ADSS1[B,H,W] 00000000	ADSS0[B,H,W] 00000000
0x030	-	-	ADST0[B,H,W] 00010000	ADST1[B,H,W] 00010000
0x034	-	-	-	ADCT[B,H,W] 00000111
0x038	-	-	SCTSL[B,H,W] ----0000	PRTSL[B,H,W] ----0000
0x03C	-	-	ADCEN[B,H,W] 11111111 ------00	
0x040	*			
0x044	-	-	-	WCMRCOT[B,H,W] 00000000
0x048	-	-	-	WCMRCIF[B,H,W] 00000000
0x04C	-	-	WCMPSR[B,H,W] 00000000	WCMPCR[B,H,W] 00100000
0x050	WCMPDH[B,H,W] 00000000 00000000		WCMPDL[B,H,W] 00000000 00000000	
0x040 - 0x0FC	-	-	-	-

CR Trim Base_Address : 0x4002_E000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	-	MCR_PSR[B,H,W] -----001
0x004	-	-	MCR_FTRM[B,H,W] ------01 11101111	
0x008	-	-	-	MCR_TTRM[B,H,W] ---10000
0x00C	MCR_RLR[W] 00000000 00000000 00000000 00000001			
0x010 - 0x0FC	-	-	-	-

EXTI Base_Address : 0x4003_0000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	ENIR[B,H,W] 00000000 00000000 00000000 00000000			
0x004	EIRR[B,H,W] XXXXXXXX XXXXXXXX XXXXXXXX XXXXXXXX			
0x008	EICL[B,H,W] 11111111 11111111 11111111 11111111			
0x00C	ELVR[R/W] 00000000 00000000 00000000 00000000			
0x010	ELVR1[R/W] 00000000 00000000 00000000 00000000			
0x014	-	-	-	NMIRR[B,H,W] -------0
0x018	-	-	-	NMICL[B,H,W] -------1
0x01C	-	-	-	-
0x020 - 0x0FC	-	-	-	-

INT-Req. READ Base_Address : 0x4003_1000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	DRQSEL[B,H,W] 00000000 00000000 00000000 00000000			
0x004 – 0x00C	-			
0x010	-	-	-	ODDPKS[B] ---00000
0x014	-	-	-	-
0x018	-	*	-	*
0x01C – 0x10C	-	-	-	-
0x110	IRQ003SEL[B,H,W] -------- 00000000 -------- 00000000			
0x114	IRQ004SEL[B,H,W] -------- 00000000 -------- 00000000			
0x118	IRQ005SEL[B,H,W] -------- 00000000 -------- 00000000			
0x11C	IRQ006SEL[B,H,W] -------- 00000000 -------- 00000000			
0x120	IRQ007SEL[B,H,W] -------- 00000000 -------- 00000000			
0x124	IRQ008SEL[B,H,W] -------- 00000000 -------- 00000000			
0x128	IRQ009SEL[B,H,W] -------- 00000000 -------- 00000000			
0x12C	IRQ010SEL[B,H,W] -------- 00000000 -------- 00000000			
0x130 – 0x1FC	-	-	-	-
0x200	EXC02MON[B,H,W] -------- -------- -------- ------00			
0x204	IRQ000MON[B,H,W] -------- -------- -------- -------0			
0x208	IRQ001MON[B,H,W] -------- -------- -------- -------0			
0x20C	IRQ002MON[B,H,W] -------- -------- -------- -------0			
0x210	IRQ003MON[B,H,W] -------- -------- -------- 00000000			
0x214	IRQ004MON[B,H,W] -------- -------- -------- 00000000			
0x218	IRQ005MON[B,H,W] -------- -------- -------- 00000000			
0x21C	IRQ006MON[B,H,W] -------- -------- -------- 00000000			
0x220	IRQ007MON[B,H,W] -------- -------- -------- 00000000			
0x224	IRQ008MON[B,H,W] -------- -------- -------- 00000000			
0x228	IRQ009MON[B,H,W] -------- -------- -------- 00000000			

<table>
<tr><th rowspan="2">Base_Address
+ Address</th><th colspan="4">Register</th></tr>
<tr><th>+3</th><th>+2</th><th>+1</th><th>+0</th></tr>
<tr><td>0x22C</td><td colspan="4">IRQ010MON[B,H,W]
-------- -------- -------- 00000000</td></tr>
<tr><td>0x230</td><td colspan="4">IRQ011MON[B,H,W]
-------- -------- -------- -------0</td></tr>
<tr><td>0x234</td><td colspan="4">IRQ012MON[B,H,W]
-------- -------- -------- -------0</td></tr>
<tr><td>0x238</td><td colspan="4">IRQ013MON[B,H,W]
-------- -------- -------- -------0</td></tr>
<tr><td>0x23C</td><td colspan="4">IRQ014MON[B,H,W]
-------- -------- -------- -------0</td></tr>
<tr><td>0x240</td><td colspan="4">IRQ015MON[B,H,W]
-------- -------- -------- -------0</td></tr>
<tr><td>0x244</td><td colspan="4">IRQ016MON[B,H,W]
-------- -------- -------- -------0</td></tr>
<tr><td>0x248</td><td colspan="4">IRQ017MON[B,H,W]
-------- -------- -------- -------0</td></tr>
<tr><td>0x24C</td><td colspan="4">IRQ018MON[B,H,W]
-------- -------- -------- -------0</td></tr>
<tr><td>0x250</td><td colspan="4">IRQ019MON[B,H,W]
-------- -------- -------- --000000</td></tr>
<tr><td>0x254</td><td colspan="4">IRQ020MON[B,H,W]
-------- -------- -------- --000000</td></tr>
<tr><td>0x258</td><td colspan="4">IRQ021MON[B,H,W]
-------- -------- -------- ----0000</td></tr>
<tr><td>0x25C</td><td colspan="4">IRQ022MON[B,H,W]
-------- -------- -------- ----0000</td></tr>
<tr><td>0x260</td><td colspan="4">IRQ023MON[B,H,W]
-------- -------- -------- ----0000</td></tr>
<tr><td>0x264</td><td colspan="4">IRQ024MON[B,H,W]
-------- -------- -------- -----000</td></tr>
<tr><td>0x268</td><td colspan="4">IRQ025MON[B,H,W]
-------- -------- -------- -----000</td></tr>
<tr><td>0x26C</td><td colspan="4">IRQ026MON[B,H,W]
-------- -------- -------- ----0000</td></tr>
<tr><td>0x270</td><td colspan="4">IRQ027MON[B,H,W]
-------- -------- -------- --000000</td></tr>
<tr><td>0x274</td><td colspan="4">IRQ028MON[B,H,W]
-------- -------- -------- -----000</td></tr>
<tr><td>0x278</td><td colspan="4">IRQ029MON[B,H,W]
-------- -------- -------- -----000</td></tr>
<tr><td>0x27C</td><td colspan="4">IRQ030MON[B,H,W]
-------- -------- -------- ----0000</td></tr>
<tr><td>0x280</td><td colspan="4">IRQ031MON[B,H,W]
-------- -------- -------- --000000</td></tr>
<tr><td>0x284</td><td colspan="4">IRQ032MON[B,H,W]
-------- -------- -------- -----000</td></tr>
<tr><td>0x288</td><td colspan="4">IRQ033MON[B,H,W]
-------- -------- -------- -----000</td></tr>
<tr><td>0x28C</td><td colspan="4">IRQ034MON[B,H,W]
-------- -------- -------- ---00000</td></tr>
<tr><td>0x290</td><td colspan="4">IRQ035MON[B,H,W]
-------- -------- -------- --000000</td></tr>
<tr><td>0x294</td><td colspan="4">IRQ036MON[B,H,W]
-------- -------- -------- -----000</td></tr>
</table>

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x298	IRQ037MON[B,H,W] -------- -------- -------- -----000			
0x29C	IRQ038MON[B,H,W] -------- -------- -------- -----000			
0x2A0	IRQ039MON[B,H,W] -------- -------- -------- ------00			
0x2A4	IRQ040MON[B,H,W] -------- -------- -------- ------00			
0x2A8	IRQ041MON[B,H,W] -------- -------- -------- ------00			
0x2AC	IRQ042MON[B,H,W] -------- -------- -------- ------00			
0x2B0	IRQ043MON[B,H,W] -------- -------- -------- ------00			
0x2B4	IRQ044MON[B,H,W] -------- -------- -------- ------00			
0x2B8	IRQ045MON[B,H,W] -------- -------- -------- ------00			
0x2BC	IRQ046MON[B,H,W] -------- -------- -------- ------00			
0x2C0	IRQ047MON[B,H,W] -------- -------- -------- ------00			
0x2C4	IRQ048MON[B,H,W] -------- -------- -------- -------0			
0x2C8	IRQ049MON[B,H,W] -------- -------- -------- -------0			
0x2CC	IRQ050MON[B,H,W] -------- -------- -------- -------0			
0x2D0	IRQ051MON[B,H,W] -------- -------- -------- -------0			
0x2D4	IRQ052MON[B,H,W] -------- -------- -------- -------0			
0x2D8	IRQ053MON[B,H,W] -------- -------- -------- -------0			
0x2DC	IRQ054MON[B,H,W] -------- -------- -------- -------0			
0x2E0	IRQ055MON[B,H,W] -------- -------- -------- -------0			
0x2E4	IRQ056MON[B,H,W] -------- -------- -------- -------0			
0x2E8	IRQ057MON[B,H,W] -------- -------- -------- -------0			
0x2EC	IRQ058MON[B,H,W] -------- -------- -------- -------0			
0x2F0	IRQ059MON[B,H,W] -------- -------- -------- ----0000			
0x2F4	IRQ060MON[B,H,W] -------- -------- -------- -------0			
0x2F8	IRQ061MON[B,H,W] -------- -------- -------- ------00			
0x2FC	IRQ062MON[B,H,W] -------- -------- -------- -------0			

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x300	IRQ063MON[B,H,W] -------- -------- -------- ------00			
0x304	IRQ064MON[B,H,W] -------- -------- -------- -------0			
0x308	IRQ065MON[B,H,W] -------- -------- -------- ------00			
0x30C	IRQ066MON[B,H,W] -------- -------- -------- -------0			
0x310	IRQ067MON[B,H,W] -------- -------- -------- ------00			
0x314	IRQ068MON[B,H,W] -------- -------- -------- -------0			
0x318	IRQ069MON[B,H,W] -------- -------- -------- ------00			
0x31C	IRQ070MON[B,H,W] -------- -------- -------- -------0			
0x320	IRQ071MON[B,H,W] -------- -------- -------- ------00			
0x324	IRQ072MON[B,H,W] -------- -------- -------- -------0			
0x328	IRQ073MON[B,H,W] -------- -------- -------- ------00			
0x32C	IRQ074MON[B,H,W] -------- -------- -------- -------0			
0x330	IRQ075MON[B,H,W] -------- -------- -------- ------00			
0x334	IRQ076MON[B,H,W] -------- -------- -------- ---00000			
0x338	IRQ077MON[B,H,W] -------- -------- -------- ---00000			
0x33C	IRQ078MON[B,H,W] -------- -------- -------- ---00000			
0x340	IRQ079MON[B,H,W] -------- -------- -------- --000000			
0x344	IRQ080MON[B,H,W] -------- -------- -------- -------0			
0x348	IRQ081MON[B,H,W] -------- -------- -------- -------0			
0x34C	IRQ082MON[B,H,W] -------- -------- -------- -----000			
0x350	IRQ083MON[B,H,W] -------- -------- -------- -------0			
0x354	IRQ084MON[B,H,W] -------- -------- -------- -------0			
0x358	IRQ085MON[B,H,W] -------- -------- -------- -------0			
0x35C	IRQ086MON[B,H,W] -------- -------- -------- -------0			
0x360	IRQ087MON[B,H,W] -------- -------- -------- -------0			
0x364	IRQ088MON[B,H,W] -------- -------- -------- -------0			
0x368	IRQ089MON[B,H,W] -------- -------- -------- -------0			

<table>
<tr><th rowspan="2">Base_Address
+ Address</th><th colspan="4">Register</th></tr>
<tr><th>+3</th><th>+2</th><th>+1</th><th>+0</th></tr>
<tr><td>0x36C</td><td colspan="4">IRQ090MON[B,H,W]
-------- -------- -------- -------0</td></tr>
<tr><td>0x370</td><td colspan="4">IRQ091MON[B,H,W]
-------- -------- -------- ------00</td></tr>
<tr><td>0x374</td><td colspan="4">IRQ092MON[B,H,W]
-------- -------- -------- ----0000</td></tr>
<tr><td>0x378</td><td colspan="4">IRQ093MON[B,H,W]
-------- -------- -------- ----0000</td></tr>
<tr><td>0x37C</td><td colspan="4">IRQ094MON[B,H,W]
-------- -------- -------- ----0000</td></tr>
<tr><td>0x380</td><td colspan="4">IRQ095MON[B,H,W]
-------- -------- -------- ----0000</td></tr>
<tr><td>0x384</td><td colspan="4">IRQ096MON[B,H,W]
-------- -------- -------- --000000</td></tr>
<tr><td>0x388</td><td colspan="4">IRQ097MON[B,H,W]
-------- -------- -------- --000000</td></tr>
<tr><td>0x38C</td><td colspan="4">IRQ098MON[B,H,W]
-------- -------- -------- ------00</td></tr>
<tr><td>0x390</td><td colspan="4">IRQ099MON[B,H,W]
-------- -------- -------- ------00</td></tr>
<tr><td>0x394</td><td colspan="4">IRQ100MON[B,H,W]
-------- -------- -------- ------00</td></tr>
<tr><td>0x398</td><td colspan="4">IRQ101MON[B,H,W]
-------- -------- -------- ------00</td></tr>
<tr><td>0x39C</td><td colspan="4">IRQ102MON[B,H,W]
-------- -------- -------- ------00</td></tr>
<tr><td>0x3A0</td><td colspan="4">IRQ103MON[B,H,W]
-------- -------- -------- -------0</td></tr>
<tr><td>0x3A4</td><td colspan="4">IRQ104MON[B,H,W]
-------- -------- -------- ------00</td></tr>
<tr><td>0x3A8</td><td colspan="4">IRQ105MON[B,H,W]
-------- -------- -------- -------0</td></tr>
<tr><td>0x3AC</td><td colspan="4">IRQ106MON[B,H,W]
-------- -------- -------- ------00</td></tr>
<tr><td>0x3B0</td><td colspan="4">IRQ107MON[B,H,W]
-------- -------- -------- -------0</td></tr>
<tr><td>0x3B4</td><td colspan="4">IRQ108MON[B,H,W]
-------- -------- -------- ------00</td></tr>
<tr><td>0x3B8</td><td colspan="4">IRQ109MON[B,H,W]
-------- -------- -------- -------0</td></tr>
<tr><td>0x3BC</td><td colspan="4">IRQ110MON[B,H,W]
-------- -------- -------- ------00</td></tr>
<tr><td>0x3C0</td><td colspan="4">IRQ111MON[B,H,W]
-------- -------- -------- ---00000</td></tr>
<tr><td>0x3C4</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>0x3C8</td><td colspan="4">IRQ113MON[B,H,W]
-------- -------- -------- ---00000</td></tr>
<tr><td>0x3CC</td><td colspan="4">IRQ114MON[B,H,W]
-------- -------- -------- --000000</td></tr>
<tr><td>0x3D0 – 0x3D8</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>0x3DC</td><td colspan="4">IRQ118MON[B,H,W]
-------- -------- -------- ------00</td></tr>
<tr><td>0x3E0</td><td colspan="4">IRQ119MON[B,H,W]
-------- -------- -------- -------0</td></tr>
</table>

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x3E4	IRQ120MON[B,H,W] -------- -------- -------- -------0			
0x3E8	IRQ121MON[B,H,W] -------- -------- -------- ------00			
0x3EC	IRQ122MON[B,H,W] -------- -------- -------- -------0			
0x3F0	IRQ123MON[B,H,W] -------- -------- -------- ------00			
0x3F4	IRQ124MON[B,H,W] -------- -------- -------- -------0			
0x3F8	IRQ125MON[B,H,W] -------- -------- -------- ------00			
0x3FC	IRQ126MON[B,H,W] -------- -------- -------- -------0			
0x400	IRQ127MON[B,H,W] -------- -------- -------- ------00			
0x404 – 0xFFC	-	-	-	-

12bit D/AC unit0 Base_Address : 0x4003_3000
12bit D/AC unit1 Base_Address : 0x4003_3008

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000	-	-	-	DACR[B,H,W] --00--00
0x004	-	-	DADR[H,W] ----XXXX XXXXXXXX	
0x010 – 0xFFC	-	-	-	-

GPIO Base_Address : 0x4006_F000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	PFR0[B,H,W] ---- ---- ---- ---- 0000 0000 0001 1111			
0x004	PFR1[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x008	PFR2[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x00C	PFR3[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x010	PFR4[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x014	PFR5[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x018	PFR6[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x01C	PFR7[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x020	PFR8[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x024	PFR9[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x028	PFRA[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x02C	PFRB[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x030	PFRC[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x034	PFRD[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x038	PFRE[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x03C	PFRF[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x040 - 0x0FC	-	-	-	-

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x100	PCR0[B,H,W] ---- ---- ---- ---- 0000 0000 0001 1111			
0x104	PCR1[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x108	PCR2[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x10C	PCR3[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x110	PCR4[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x114	PCR5[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x118	PCR6[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x11C	PCR7[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x120	-			
0x124	PCR9[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x128	PCRA[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x12C	PCRB[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x130	PCRC[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x134	PCRD[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x138	PCRE[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x13C	PCRF[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x140 – 0x1FC	-			

<table>
<tr><th>Base_Address</th><th colspan="4">Register</th></tr>
<tr><th>+ Address</th><th>+3</th><th>+2</th><th>+1</th><th>+0</th></tr>
<tr><td>0x200</td><td colspan="4">DDR0[B,H,W]
---- ---- ---- ---- 0000 0000 0000 0000</td></tr>
<tr><td>0x204</td><td colspan="4">DDR1[B,H,W]
---- ---- ---- ---- 0000 0000 0000 0000</td></tr>
<tr><td>0x208</td><td colspan="4">DDR2[B,H,W]
---- ---- ---- ---- 0000 0000 0000 0000</td></tr>
<tr><td>0x20C</td><td colspan="4">DDR3[B,H,W]
---- ---- ---- ---- 0000 0000 0000 0000</td></tr>
<tr><td>0x210</td><td colspan="4">DDR4[B,H,W]
---- ---- ---- ---- 0000 0000 0000 0000</td></tr>
<tr><td>0x214</td><td colspan="4">DDR5[B,H,W]
---- ---- ---- ---- 0000 0000 0000 0000</td></tr>
<tr><td>0x218</td><td colspan="4">DDR6[B,H,W]
---- ---- ---- ---- 0000 0000 0000 0000</td></tr>
<tr><td>0x21C</td><td colspan="4">DDR7[B,H,W]
---- ---- ---- ---- 0000 0000 0000 0000</td></tr>
<tr><td>0x220</td><td colspan="4">DDR8[B,H,W]
---- ---- ---- ---- 0000 0000 0000 0000</td></tr>
<tr><td>0x224</td><td colspan="4">DDR9[B,H,W]
---- ---- ---- ---- 0000 0000 0000 0000</td></tr>
<tr><td>0x228</td><td colspan="4">DDRA[B,H,W]
---- ---- ---- ---- 0000 0000 0000 0000</td></tr>
<tr><td>0x22C</td><td colspan="4">DDRB[B,H,W]
---- ---- ---- ---- 0000 0000 0000 0000</td></tr>
<tr><td>0x230</td><td colspan="4">DDRC[B,H,W]
---- ---- ---- ---- 0000 0000 0000 0000</td></tr>
<tr><td>0x234</td><td colspan="4">DDRD[B,H,W]
---- ---- ---- ---- 0000 0000 0000 0000</td></tr>
<tr><td>0x238</td><td colspan="4">DDRE[B,H,W]
---- ---- ---- ---- 0000 0000 0000 0000</td></tr>
<tr><td>0x23C</td><td colspan="4">DDRF[B,H,W]
---- ---- ---- ---- 0000 0000 0000 0000</td></tr>
<tr><td>0x240 - 0x2FC</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
</table>

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x300	PDIR0[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x304	PDIR1[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x308	PDIR2[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x30C	PDIR3[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x310	PDIR4[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x314	PDIR5[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x318	PDIR6[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x31C	PDIR7[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x320	PDIR8[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x324	PDIR9[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x328	PDIRA[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x32C	PDIRB[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x330	PDIRC[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x334	PDIRD[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x338	PDIRE[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x33C	PDIRF[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x340 - 0x3FC	-	-	-	-

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x400	PDOR0[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x404	PDOR1[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x408	PDOR2[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x40C	PDOR3[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x410	PDOR4[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x414	PDOR5[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x418	PDOR6[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x41C	PDOR7[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x420	PDOR8[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x424	PDOR9[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x428	PDORA[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x42C	PDORB[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x430	PDORC[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x434	PDORD[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x438	PDORE[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x43C	PDORF[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x440 - 0x4FC	-	-	-	-
0x500	ADE[B,H,W] 1111 1111 1111 1111 1111 1111 1111 1111			
0x504 - 0x57C	-	-	-	-
0x580	SPSR[B,H,W] ---- ---- ---- ---- ---- ---- --00 01--			
0x584 - 0x5FC	-	-	-	-

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x600	EPFR00[B,H,W] ---- --00 ---- --11 --0- --0- 0000 --00			
0x604	EPFR01[B,H,W] 0000 0000 0000 0000 ---0 0000 0000 0000			
0x608	EPFR02[B,H,W] 0000 0000 0000 0000 ---0 0000 0000 0000			
0x60C	EPFR03[B,H,W] 0000 0000 0000 0000 ---0 0000 0000 0000			
0x610	EPFR04[B,H,W] --00 0000 --00 00-- --00 0000 -000 00--			
0x614	EPFR05[B,H,W] --00 0000 --00 00-- --00 0000 --00 00--			
0x618	EPFR06[B,H,W] 0000 0000 0000 0000 0000 0000 0000 0000			
0x61C	EPFR07[B,H,W] 0000 0000 0000 0000 0000 0000 0000 ----			
0x620	EPFR08[B,H,W] 0000 0000 0000 0000 0000 0000 0000 0000			
0x624	EPFR09[B,H,W] 0000 0000 0000 0000 0000 0000 0000 0000			
0x628	EPFR10[B,H,W] 0000 0000 0000 0000 0000 0000 0000 0000			
0x62C	EPFR11[B,H,W] ---- --00 0000 0000 0000 0000 0000 0000			
0x630	EPFR12[B,H,W] --00 0000 --00 00-- --00 0000 --00 00--			
0x634	EPFR13[B,H,W] --00 0000 --00 00-- --00 0000 --00 00--			
0x638	EPFR14[B,H,W] --00 0000 0000 00-- ---- ---- --00 0000			
0x63C	EPFR15[B,H,W] 0000 0000 0000 0000 0000 0000 0000 0000			
0x640	EPFR16[B,H,W] --00 0000 0000 0000 0000 0000 0000 0000			
0x644	EPFR17[B,H,W] ---- 0000 0000 0000 0000 0000 0000 ----			
0x648	EPFR18[B,H,W] --00 0000 0000 0000 00-- --00 0000 ----			
0x64C	EPFR19[B,H,W] ---- ---- ---- ---- ---- ---- ---- ----			
0x650	EPFR20[B,H,W] ---- ---0 0000 0000 0000 0000 0000 0000			
0x654 – 0x6FC	-	-	-	-

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x700	PZR0[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x704	PZR1[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x708	PZR2[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x70C	PZR3[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x710	PZR4[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x714	PZR5[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x718	PZR6[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x71C	PZR7[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x720	PZR8[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x724	PZR9[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x728	PZRA[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x72C	PZRB[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x730	PZRC[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x734	PZRD[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x738	PZRE[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x73C	PZRF[B,H,W] ---- ---- ---- ---- 0000 0000 0000 0000			
0x740 - 0xEFC	-	-	-	-
0xF00 – 0xF04	*			
0xF08 – 0xFDC	-	-	-	-
0xFE0	*			
0xFE4 - 0xFFC	-	-	-	-

LVD Base_Address : 0x4003_5000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	-	LVD_CTL[B,H,W] 000111--
0x004	-	-	-	LVD_STR[B,H,W] 0-------
0x008	-	-	-	LVD_CLR[B,H,W] 1-------
0x00C	LVD_RLR[W] 00000000 00000000 00000000 00000001			
0x010	-	-	-	LVD_STR2 [B,H,W] 0------
0x014 - 0x0FC	-	-	-	-

DS_Mode Base_Address : 0x4003_5100

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	-	*
0x004	-	-	-	RCK_CTL[B,H,W] ------01
0x008 - 0x6FC	-	-	-	-
0x700	-	-	-	PMD_CTL[B,H,W] -------0
0x704	-	-	-	WRFSR[B,H,W] ------00
0x708	-	-	WIFSR[B,H,W] ------00 00000000	
0x70C	-	-	WIER[B,H,W] ------00 00000-00	
0x710	-	-	-	WILVR[B,H,W] ---00000
0x714	-	-	-	DSRAMR[B,H,W] ------00
0x718 - 0x7FC	-	-	-	-
0x800	BUR04[B,H,W] 00000000	BUR03[B,H,W] 00000000	BUR02[B,H,W] 00000000	BUR01[B,H,W] 00000000
0x804	BUR08[B,H,W] 00000000	BUR07[B,H,W] 00000000	BUR06[B,H,W] 00000000	BUR05[B,H,W] 00000000
0x808	BUR012[B,H,W] 00000000	BUR11[B,H,W] 00000000	BUR10[B,H,W] 00000000	BUR09[B,H,W] 00000000
0x80C	BUR16[B,H,W] 00000000	BUR15[B,H,W] 00000000	BUR14[B,H,W] 00000000	BUR13[B,H,W] 00000000
0x810 - 0xEFC	-	-	-	-

USB Clock Base_Address : 0x4003_6000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	-	UCCR[B,H,W] -0000000
0x004	-	-	-	UPCR1[B,H,W] ------00
0x008	-	-	-	UPCR2[B,H,W] -----000
0x00C	-	-	-	UPCR3[B,H,W] ---00000
0x010	-	-	-	UPCR4[B,H,W] -0111011
0x014	-	-	-	UP_STR[B,H,W] -------0
0x018	-	-	-	UPINT_ENR[B,H,W] -------0
0x01C	-	-	-	UPINT_CLR[B,H,W] -------0
0x020	-	-	-	UPINT_STR[B,H,W] -------0
0x024	-	-	-	UPCR5[B,H,W] ----0100
0x028	-	-	-	UPCR6[B,H,W] ----0010
0x02C	-	-	-	UPCR7[B,H,W] -------0
0x030	-	-	-	USBEN0[B,H,W] -------0
0x034	-	-	-	USBEN1[B,H,W] -------0
0x038 - 0x0FC	-	-	-	-

CAN_Prescaler Base_Address : 0x4003_7000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	-	CANPRE[B,H,W] ----1011
0x004 - 0xFFC	-	-	-	-

MFS

MFS ch.0	Base_Address : 0x4003_8000
MFS ch.1	Base_Address : 0x4003_8100
MFS ch.2	Base_Address : 0x4003_8200
MFS ch.3	Base_Address : 0x4003_8300
MFS ch.4	Base_Address : 0x4003_8400
MFS ch.5	Base_Address : 0x4003_8500
MFS ch.6	Base_Address : 0x4003_8600
MFS ch.7	Base_Address : 0x4003_8700
MFS ch.8	Base_Address : 0x4003_8800
MFS ch.9	Base_Address : 0x4003_8900
MFS ch.10	Base_Address : 0x4003_8A00
MFS ch.11	Base_Address : 0x4003_8B00
MFS ch.12	Base_Address : 0x4003_8C00
MFS ch.13	Base_Address : 0x4003_8D00
MFS ch.14	Base_Address : 0x4003_8E00
MFS ch.15	Base_Address : 0x4003_8F00

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	SCR / IBCR[B,H,W] 0--0000	SMR[B,H,W] 000-00-0
0x004	-	-	SSR[B,H,W] 0-00011	ESCR / IBSR[B,H,W] 0000000
0x008	-	-	RDR/TDR[H,W] 00000000 00000000	
0x00C	-	-	BGR1[B,H,W] 00000000	BGR0[B,H,W] 00000000
0x010	-	-	ISMK[B,H,W] --------	ISBA[B,H,W] --------
0x014	-	-	FCR1[B,H,W] ---00100	FCR0[B,H,W] -0000000
0x018	-	-	FBYTE2[B,H,W] 00000000	FBYTE1[B,H,W] 00000000
0x01C	-	-	SCSTR1/ EIBCR[B,H,W] 00000000	SCSTR0/ NFCR[B,H,W] 00000000
0x020	-	-	SCSTR3[B,H,W] 00000000	SCSTR2[B,H,W] 00000000
0x024	-	-	SACSR1[B,H,W] 00000000	SACSR0[B,H,W] 00000000
0x028	-	-	STMR1[B,H,W] 00000000	STMR0[B,H,W] 00000000

Base_Address + Address	Register			
	+3	+3	+3	+3
0x02C	-	-	STMCR1[B,H,W] 00000000	STMCR0[B,H,W] 00000000
0x030	-	-	SCSCR1[B,H,W] 00000000	SCSCR0[B,H,W] 00100000
0x034	-	-	SCSFR1[B,H,W] 10000000	SCSFR0[B,H,W] 10000000
0x038	-	-	-	SCSFR2[B,H,W] 10000000
0x03C	-	-	TBYTE1[B,H,W] 00000000	TBYTE0[B,H,W] 00000000
0x040	-	-	TBYTE3[B,H,W] 00000000	TBYTE2[B,H,W] 00000000
0x0144 - 0x1FC	-	-	-	-

CRC Base_Address : 0x4003_9000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	-	CRCCR[B,H,W] -0000000
0x004	CRCINIT[B,H,W] 11111111 11111111 11111111 11111111			
0x008	CRCIN[B,H,W] 00000000 00000000 00000000 00000000			
0x00C	CRCR[B,H,W] 11111111 11111111 11111111 11111111			

Watch Counter Base_Address : 0x4003_A000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	WCCR[B,H,W] 00--0000	WCRL[B,H,W] --000000	WCRD[B,H,W] --000000
0x004 - 0x00C	-	-	-	-
0x010	-	-	CLK_SEL[B,H,W] -----000 -------0	
0x014	-	-	-	CLK_EN[B,H,W] ------00
0x018 - 0xFFC	-	-	-	-

RTC Base_Address : 0x4003_B000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x100	-	-	-	WTCR10[B,H,W] 00000000
0x104	-	-	-	WTCR11[B,H,W] ---00000
0x108	-	-	-	WTCR12[B,H,W] 00000000
0x10C	-	-	-	WTCR13[B,H,W] 00000000
0x110	-	-	-	WTCR20[B,H,W] --000000
0x114	-	-	-	WTCR21[B,H,W] -----000
0x118	-	-	-	*
0x11C	-	-	-	WTSR[B,H,W] -0000000
0x120	-	-	-	WTMIR[B,H,W] -0000000
0x124	-	-	-	WTHR[B,H,W] --000000
0x128	-	-	-	WTDR[B,H,W] --000000
0x12C	-	-	-	WTDW[B,H,W] -----000
0x130	-	-	-	WTMOR[B,H,W] ---00000
0x134	-	-	-	WTYR[B,H,W] 00000000
0x138	-	-	-	ALMIR[B,H,W] -0000000
0x13C	-	-	-	ALHR[B,H,W] --000000
0x140	-	-	-	ALDR[B,H,W] --000000
0x144	-	-	-	ALMOR[B,H,W] ---00000
0x148	-	-	-	ALYR[B,H,W] 00000000
0x14C	-	-	-	WTTR0[B,H,W] 00000000
0x150	-	-	-	WTTR1[B,H,W] 00000000
0x154	-	-	-	WTTR2[B,H,W] ------00
0x158	-	-	-	WTCAL0[B,H,W] 00000000
0x15C	-	-	-	WTCAL1[B,H,W] ------00
0x160	-	-	-	WTCALEN[B,H,W] -------0
0x164	-	-	-	WTDIV[B,H,W] ----0000
0x168	-	-	-	WTDIVEN[B,H,W] ------00

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x16C	-	-	-	WTCALPRD[B,H,W] --010011
0x170	-	-	-	WTCOSEL[B,H,W] -------0
0x174	-	-	-	VB_DIVCLK[B,H,W] 00000111
0x178	-	-	-	WTOSCCNT[B,H,W] ------01
0x17C	-	-	-	CCS[B,H,W] 00000000
0x180	-	-	-	CCB[B,H,W] 00000000
0x184	-	-	-	TRIM[B,H,W] 00000000
0x188	-	-	-	BOOST[B,H,W] ------11
0x18C	-	-	-	EWKUP[B,H,W] -------0
0x190	-	-	-	VDET[B,H,W] 00------
0x194	-	-	-	FDET[B,H,W] 0-------
0x198	-	-	-	HIBRST[B,H,W] -------0
0x19C	-	-	-	VBPFR[B,H,W] --011100
0x1A0	-	-	-	VBPCR[B,H,W] ----0000
0x1A4	-	-	-	VBDDR[B,H,W] ----0000
0x1A8	-	-	-	VBDIR[B,H,W] ----0000
0x1AC	-	-	-	VBDOR[B,H,W] ----1111
0x0B0	-	-	-	VBPZR[B,H,W] ------11
0x1B4-1FF	-	-	-	-

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x200	BREG03[B,H,W] 00000000	BREG02[B,H,W] 00000000	BREG01[B,H,W] 00000000	BREG00[B,H,W] 00000000
0x204	BREG07[B,H,W] 00000000	BREG06[B,H,W] 00000000	BREG05[B,H,W] 00000000	BREG04[B,H,W] 00000000
0x208	BREG0B[B,H,W] 00000000	BREG0A[B,H,W] 00000000	BREG09[B,H,W] 00000000	BREG08[B,H,W] 00000000
0x20C	BREG0F[B,H,W] 00000000	BREG0E[B,H,W] 00000000	BREG0D[B,H,W] 00000000	BREG0C[B,H,W] 00000000
0x210	BREG13[B,H,W] 00000000	BREG12[B,H,W] 00000000	BREG11[B,H,W] 00000000	BREG10[B,H,W] 00000000
0x214	BREG17[B,H,W] 00000000	BREG16[B,H,W] 00000000	BREG15[B,H,W] 00000000	BREG14[B,H,W] 00000000
0x218	BREG1B[B,H,W] 00000000	BREG1A[B,H,W] 00000000	BREG19[B,H,W] 00000000	BREG18[B,H,W] 00000000
0x21C	BREG1F[B,H,W] 00000000	BREG1E[B,H,W] 00000000	BREG1D[B,H,W] 00000000	BREG1C[B,H,W] 00000000
0x220	BREG23[B,H,W] 00000000	BREG22[B,H,W] 00000000	BREG21[B,H,W] 00000000	BREG20[B,H,W] 00000000
0x224	BREG27[B,H,W] 00000000	BREG26[B,H,W] 00000000	BREG25[B,H,W] 00000000	BREG24[B,H,W] 00000000
0x228	BREG2B[B,H,W] 00000000	BREG2A[B,H,W] 00000000	BREG29[B,H,W] 00000000	BREG28[B,H,W] 00000000
0x22C	BREG2F[B,H,W] 00000000	BREG2E[B,H,W] 00000000	BREG2D[B,H,W] 00000000	BREG2C[B,H,W] 00000000
0x230	BREG33[B,H,W] 00000000	BREG32[B,H,W] 00000000	BREG31[B,H,W] 00000000	BREG30[B,H,W] 00000000
0x234	BREG37[B,H,W] 00000000	BREG36[B,H,W] 00000000	BREG35[B,H,W] 00000000	BREG34[B,H,W] 00000000
0x238	BREG3B[B,H,W] 00000000	BREG3A[B,H,W] 00000000	BREG39[B,H,W] 00000000	BREG38[B,H,W] 00000000
0x23C	BREG3F[B,H,W] 00000000	BREG3E[B,H,W] 00000000	BREG3D[B,H,W] 00000000	BREG3C[B,H,W] 00000000
0x240	BREG43[B,H,W] 00000000	BREG42[B,H,W] 00000000	BREG41[B,H,W] 00000000	BREG40[B,H,W] 00000000
0x244	BREG47[B,H,W] 00000000	BREG46[B,H,W] 00000000	BREG45[B,H,W] 00000000	BREG44[B,H,W] 00000000
0x248	BREG4B[B,H,W] 00000000	BREG4A[B,H,W] 00000000	BREG49[B,H,W] 00000000	BREG48[B,H,W] 00000000
0x24C	BREG4F[B,H,W] 00000000	BREG4E[B,H,W] 00000000	BREG4D[B,H,W] 00000000	BREG4C[B,H,W] 00000000
0x250	BREG53[B,H,W] 00000000	BREG52[B,H,W] 00000000	BREG51[B,H,W] 00000000	BREG50[B,H,W] 00000000
0x254	BREG57[B,H,W] 00000000	BREG56[B,H,W] 00000000	BREG55[B,H,W] 00000000	BREG54[B,H,W] 00000000
0x258	BREG5B[B,H,W] 00000000	BREG5A[B,H,W] 00000000	BREG59[B,H,W] 00000000	BREG58[B,H,W] 00000000
0x25C	BREG5F[B,H,W] 00000000	BREG5E[B,H,W] 00000000	BREG5D[B,H,W] 00000000	BREG5C[B,H,W] 00000000
0x260	BREG63[B,H,W] 00000000	BREG62[B,H,W] 00000000	BREG61[B,H,W] 00000000	BREG60[B,H,W] 00000000
0x264	BREG67[B,H,W] 00000000	BREG66[B,H,W] 00000000	BREG65[B,H,W] 00000000	BREG64[B,H,W] 00000000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x268	BREG6B[B,H,W] 00000000	BREG6A[B,H,W] 00000000	BREG69[B,H,W] 00000000	BREG68[B,H,W] 00000000
0x26C	BREG6F[B,H,W] 00000000	BREG6E[B,H,W] 00000000	BREG6D[B,H,W] 00000000	BREG6C[B,H,W] 00000000
0x270	BREG73[B,H,W] 00000000	BREG72[B,H,W] 00000000	BREG71[B,H,W] 00000000	BREG70[B,H,W] 00000000
0x274	BREG77[B,H,W] 00000000	BREG76[B,H,W] 00000000	BREG75[B,H,W] 00000000	BREG74[B,H,W] 00000000
0x278	BREG7B[B,H,W] 00000000	BREG7A[B,H,W] 00000000	BREG79[B,H,W] 00000000	BREG78[B,H,W] 00000000
0x27C	BREG7F[B,H,W] 00000000	BREG7E[B,H,W] 00000000	BREG7D[B,H,W] 00000000	BREG7C[B,H,W] 00000000
0x280-0xFFC	-	-	-	-

Low-speed CR Prescaler Base_Address : 0x4003_C000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x000	-	-	-	LCR_PRSLD[B,H,W], --000000
0x004 – 0x0FC	-	-	-	-

Peripheral Clock Gating Base_Address : 0x4003_C100

<table>
<tr><th rowspan="2">Base_Address
+ Address</th><th colspan="4">Register</th></tr>
<tr><th>+3</th><th>+2</th><th>+1</th><th>+0</th></tr>
<tr><td>0x000</td><td colspan="4">CKEN0[B,H,W]
---1-1-1 ----1111 11111111 11111111</td></tr>
<tr><td>0x004</td><td colspan="4">MRST0[B,H,W]
-----0-0 ----0000 00000000 00000000</td></tr>
<tr><td>0x008 – 0x00F</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>0x010</td><td colspan="4">CKEN1[B,H,W]
-------- ----1111 ----1111 ----1111</td></tr>
<tr><td>0x014</td><td colspan="4">MRST1[B,H,W]
-------- ----0000 ----0000 ----0000</td></tr>
<tr><td>0x018 – 0x01F</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
<tr><td>0x020</td><td colspan="4">CKEN2[B,H,W]
-------- -------- -------0 --**--00
Products with CAN : *="1"
Products without CAN : *="0"</td></tr>
<tr><td>0x024</td><td colspan="4">MRST2[B,H,W]
-------- -------- -------0 --00--00</td></tr>
<tr><td>0x028 – 0x67C</td><td>-</td><td>-</td><td>-</td><td>-</td></tr>
</table>

Main PLL Control Base_Address : 0x4003_C800

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000	SSCTL1[B,H,W] -------- -------- -------- --00---0			
0x004	SSCTL2[B,H,W] -------- -------- ------00 00000000			
0x008 – 0x0FC	-	-	-	-

EXT-Bus I/F Base_Address : 0x4003_F000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x0000	MODE0[W] -------- -------- --000-00 00000000			
0x0004	MODE1[W] -------- -------- --000-00 00000000			
0x0008	MODE2[W] -------- -------- --000-00 00000000			
0x000C	MODE3[W] -------- -------- --000-00 00000000			
0x0010	MODE4[W] -------- -------- --000-00 00000001			
0x0014	MODE5[W] -------- -------- --000-00 00000000			
0x0018	MODE6[W] -------- -------- --000-00 00000000			
0x001C	MODE7[W] -------- -------- --000-00 00000000			
0x0020	TIM0[W] 00000101 01011111 11110000 00001111			
0x0024	TIM1[W] 00000101 01011111 11110000 00001111			
0x0028	TIM2[W] 00000101 01011111 11110000 00001111			
0x002C	TIM3[W] 00000101 01011111 11110000 00001111			
0x0030	TIM4[W] 00000101 01011111 11110000 00001111			
0x0034	TIM5[W] 00000101 01011111 11110000 00001111			
0x0038	TIM6[W] 00000101 01011111 11110000 00001111			
0x003C	TIM7[W] 00000101 01011111 11110000 00001111			

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x0040	AREA0[W] -------- -0001111 -------- 00000000			
0x0044	AREA1[W] -------- -0001111 -------- 00010000			
0x0048	AREA2[W] -------- -0001111 -------- 00100000			
0x004C	AREA3[W] -------- -0001111 -------- 00110000			
0x0050	AREA4[W] -------- -0001111 -------- 01000000			
0x0054	AREA5[W] -------- -0001111 -------- 01010000			
0x0058	AREA6[W] -------- -0001111 -------- 01100000			
0x005C	AREA7[W] -------- -0001111 -------- 01110000			
0x0060	ATIM0[W] -------- -------- ----0100 01011111			
0x0064	ATIM1[W] -------- -------- ----0100 01011111			
0x0068	ATIM2[W] -------- -------- ----0100 01011111			
0x006C	ATIM3[W] -------- -------- ----0100 01011111			
0x0070	ATIM4[W] -------- -------- ----0100 01011111			
0x0074	ATIM5[W] -------- -------- ----0100 01011111			
0x0078	ATIM6[W] -------- -------- ----0100 01011111			
0x007C	ATIM7[W] -------- -------- ----0100 01011111			
0x0080 - 0x00FC	-	-	-	-
0x0100	SDMODE -------- -------0 00010011 --00-000			
0x0104	REFTIM -------0 00000000 0000000000110011			
0x0108	PWRDWN -------- -------- 00000000 00000000			
0x010C	SDTIM ------00 01000010 00010001 0100--01			
0x0110	SDCMD 0------- ---00000 00000000 00000000			
0x0114 - 0x01FC	-	-	-	-

Base_Address	Register			
+ Address	+3	+ 2	+1	+ 0
0x0200	MEMCERR -------- -------- -------- ----0000			
0x0204 – 0x02FC	-	-	-	-
0x0300	DCLKR[W] -------- -------- -------- ---01111			
0x0304	EST -------- -------- -------- -------0			
0x0308	WEAD 00000000 00000000 000000000 00000000			
0x030C	ESCLR -------- -------- -------- -------1			
0x0310	AMODE -------- -------- -------- -------1			
0x031C - 0x0EFC	-	-	-	-
0x0F00 – 0x0F14	*	*	*	*
0x0F18 – 0x0FFC	-	-	-	-

USB ch.0 Base_Address : 0x4004_0000
USB ch.1 Base_Address : 0x4005_0000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x2100	-	-	HCNT1[B,H,W] -----001	HCNT0[B,H,W] 00000000
0x2104	-	-	HERR[B,H,W] 00000011	HIRQ[B,H,W] 0-000000
0x2108	-	-	HFCOMP[B,H,W] 00000000	HSTATE[B,H,W] --010010
0x210C	-	-	HRTIMER(1/0)[B,H,W] 00000000 00000000	
0x2110	-	-	HADR[B,H,W] -0000000	HRTIMER(2)[B,H,W] ------00
0x2114	-	-	HEOF(1/0)[B,H,W] --000000 00000000	
0x2118	-	-	HFRAME(1/0)[B,H,W] -----000 00000000	
0x211C	-	-	-	HTOKEN[B,H,W] 00000000
0x2120	-	-	UDCC[B,H,W] -------- 10100-00	
0x2124	-	-	EP0C[H,W] ------0- -1000000	
0x2128	-	-	EP1C[H,W] 01100001 00000000	
0x212C	-	-	EP2C[H,W] 0110000- -1000000	
0x2130	-	-	EP3C[H,W] 0110000- -1000000	
0x2134	-	-	EP4C[H,W] 0110000- -1000000	
0x2138	-	-	EP5C[H,W] 0110000- -1000000	
0x213C	-	-	TMSP[H,W] -----000 00000000	
0x2140	-	-	UDCIE[B,H,W] --000000	UDCS[B,H,W] --000000
0x2144	-	-	EP0IS[H,W] 10---1-- --------	
0x2148	-	-	EP0OS[H,W] 100--00- -XXXXXXX	
0x214C	-	-	EP1S[H,W] 100-000X XXXXXXXX	
0x2150	-	-	EP2S[H,W] 100-000- -XXXXXXX	
0x2154	-	-	EP3S[H,W] 100-000- -XXXXXXX	
0x2158	-	-	EP4S[H,W] 100-000- -XXXXXXX	
0x215C	-	-	EP5S[H,W] 100-000- -XXXXXXX	

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x2160	-	-	EP0DTH[B,H,W] XXXXXXXX	EP0DTL[B,H,W] XXXXXXXX
0x2164	-	-	EP1DTH[B,H,W] XXXXXXXX	EP1DTL[B,H,W] XXXXXXXX
0x2168	-	-	EP2DTH[B,H,W] XXXXXXXX	EP2DTL[B,H,W] XXXXXXXX
0x216C	-	-	EP3DTH[B,H,W] XXXXXXXX	EP3DTL[B,H,W] XXXXXXXX
0x2170	-	-	EP4DTH[B,H,W] XXXXXXXX	EP4DTL[B,H,W] XXXXXXXX
0x2174	-	-	EP5DTH[B,H,W] XXXXXXXX	EP5DTL[B,H,W] XXXXXXXX
0x2178 - 0x217C	-	-	-	-

DMAC Base_Address : 0x4006_0000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x0000	DMACR[B,H,W] 00-00000 -------- -------- --------			
0x0010	DMACA0[B,H,W] 00000000 0---0000 00000000 00000000			
0x0014	DMACB0[B,H,W] --000000 00000000 00000000 -------0			
0x0018	DMACSA0[B,H,W] 00000000 00000000 00000000 00000000			
0x001C	DMACDA0[B,H,W] 00000000 00000000 00000000 00000000			
0x0020	DMACA1[B,H,W] 00000000 0---0000 00000000 00000000			
0x0024	DMACB1[B,H,W] --000000 00000000 00000000 -------0			
0x0028	DMACSA1[B,H,W] 00000000 00000000 00000000 00000000			
0x002C	DMACDA1[B,H,W] 00000000 00000000 00000000 00000000			
0x0030	DMACA2[B,H,W] 00000000 0---0000 00000000 00000000			
0x0034	DMACB2[B,H,W] --000000 00000000 00000000 -------0			
0x0038	DMACSA2[B,H,W] 00000000 00000000 00000000 00000000			
0x003C	DMACDA2[B,H,W] 00000000 00000000 00000000 00000000			
0x0040	DMACA3[B,H,W] 00000000 0---0000 00000000 00000000			
0x0044	DMACB3[B,H,W] --000000 00000000 00000000 -------0			
0x0048	DMACSA3[B,H,W] 00000000 00000000 00000000 00000000			
0x004C	DMACDA3[B,H,W] 00000000 00000000 00000000 00000000			
0x0050	DMACA4[B,H,W] 00000000 0---0000 00000000 00000000			
0x0054	DMACB4[B,H,W] --000000 00000000 00000000 -------0			
0x0058	DMACSA4[B,H,W] 00000000 00000000 00000000 00000000			
0x005C	DMACDA4[B,H,W] 00000000 00000000 00000000 00000000			
0x0060	DMACA5[B,H,W] 00000000 0---0000 00000000 00000000			
0x0064	DMACB5[B,H,W] --000000 00000000 00000000 -------0			
0x0068	DMACSA5[B,H,W] 00000000 00000000 00000000 00000000			
0x006C	DMACDA5[B,H,W] 00000000 00000000 00000000 00000000			

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x0070	DMACA6[B,H,W] 00000000 0---0000 00000000 00000000			
0x0074	DMACB6[B,H,W] --000000 00000000 00000000 -------0			
0x0078	DMACSA6[B,H,W] 00000000 00000000 00000000 00000000			
0x007C	DMACDA6[B,H,W] 00000000 00000000 00000000 00000000			
0x0080	DMACA7[B,H,W] 00000000 0---0000 00000000 00000000			
0x0084	DMACB7[B,H,W] --000000 00000000 00000000 -------0			
0x0088	DMACSA7[B,H,W] 00000000 00000000 00000000 00000000			
0x008C	DMACDA7[B,H,W] 00000000 00000000 00000000 00000000			
0x0090 - 0x00FC	-	-	-	-

DSTC Base_Address : 0x4006_1000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x0000	DESTP[B,H,W] 00000000 00000000 00000000 00000000			
0x0004	HWDESP[B,H,W] 00XXXXXX XXXXXX00 00000000 00000000			
0x0008	SWTR[H] 00000000 00000000		CFG[B] 01000000	CMD[B] 00000001
0x000C	MONERS[B,H,W] 00XXXXXX XXXXXX00 XXXXXXXX XXX00000			
0x0010	DREQENB[31:0] [B,H,W] 00000000 00000000 00000000 00000000			
0x0014	DREQENB[63:32] [B,H,W] 00000000 00000000 00000000 00000000			
0x0018	DREQENB[95:64] [B,H,W] 00000000 00000000 00000000 00000000			
0x001C	DREQENB[127:96] [B,H,W] 00000000 00000000 00000000 00000000			
0x0020	DREQENB[159:128] [B,H,W] 00000000 00000000 00000000 00000000			
0x0024	DREQENB[191:160] [B,H,W] 00000000 00000000 00000000 00000000			
0x0028	DREQENB[223:192] [B,H,W] 00000000 00000000 00000000 00000000			
0x002C	DREQENB[255:224] [B,H,W] 00000000 00000000 00000000 00000000			
0x0030	HWINT[31:0] [B,H,W] 00000000 00000000 00000000 00000000			
0x0034	HWINT[63:32] [B,H,W] 00000000 00000000 00000000 00000000			
0x0038	HWINT[95:64] [B,H,W] 00000000 00000000 00000000 00000000			
0x003C	HWINT[127:96] [B,H,W] 00000000 00000000 00000000 00000000			
0x0040	HWINT[159:128] [B,H,W] 00000000 00000000 00000000 00000000			
0x0044	HWINT[191:160] [B,H,W] 00000000 00000000 00000000 00000000			
0x0048	HWINT[223:192] [B,H,W] 00000000 00000000 00000000 00000000			
0x004C	HWINT[255:224] [B,H,W] 00000000 00000000 00000000 00000000			
0x0050	HWINTCLR[31:0] [B,H,W] 00000000 00000000 00000000 00000000			
0x0054	HWINTCLR[63:32] [B,H,W] 00000000 00000000 00000000 00000000			
0x0058	HWINTCLR[95:64] [B,H,W] 00000000 00000000 00000000 00000000			
0x005C	HWINTCLR[127:96] [B,H,W] 00000000 00000000 00000000 00000000			

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x060	HWINTCLR[159:128] [B,H,W] 00000000 00000000 00000000 00000000			
0x064	HWINTCLR[191:160] [B,H,W] 00000000 00000000 00000000 00000000			
0x068	HWINTCLR[223:192] [B,H,W] 00000000 00000000 00000000 00000000			
0x06C	HWINTCLR[255:224] [B,H,W] 00000000 00000000 00000000 00000000			
0x070	DQMSK[31:0] [B,H,W] 00000000 00000000 00000000 00000000			
0x074	DQMSK[63:32] [B,H,W] 00000000 00000000 00000000 00000000			
0x078	DQMSK[95:64] [B,H,W] 00000000 00000000 00000000 00000000			
0x07C	DQMSK[127:96] [B,H,W] 00000000 00000000 00000000 00000000			
0x080	DQMSK[159:128] [B,H,W] 00000000 00000000 00000000 00000000			
0x084	DQMSK[191:160] [B,H,W] 00000000 00000000 00000000 00000000			
0x088	DQMSK[223:192] [B,H,W] 00000000 00000000 00000000 00000000			
0x08C	DQMSK[255:224] [B,H,W] 00000000 00000000 00000000 00000000			
0x090	DQMSKCLR[31:0] [B,H,W] 00000000 00000000 00000000 00000000			
0x094	DQMSKCLR[63:32] [B,H,W] 00000000 00000000 00000000 00000000			
0x098	DQMSKCLR[95:64] [B,H,W] 00000000 00000000 00000000 00000000			
0x09C	DQMSKCLR[127:96] [B,H,W] 00000000 00000000 00000000 00000000			
0x0A0	DQMSKCLR[159:128] [B,H,W] 00000000 00000000 00000000 00000000			
0x0A4	DQMSKCLR[191:160] [B,H,W] 00000000 00000000 00000000 00000000			
0x0A8	DQMSKCLR[223:192] [B,H,W] 00000000 00000000 00000000 00000000			
0x0AC	DQMSKCLR[255:224] [B,H,W] 00000000 00000000 00000000 00000000			
0x00B0 - 0x0FFC	-	-	-	-

CAN ch.0 Base_Address : 0x4006_2000
CAN ch.1 Base_Address : 0x4006_3000

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x0000	STATR[B,H,W] -------- 00000000		CTRLR[B,H,W] -------- 000-0001	
0x0004	BTR[B,H,W] -0100011 00000001		ERRCNT[B,H,W] 00000000 00000000	
0x0008	TESTR[B,H,W] -------- X00000--		INTR[B,H,W] 00000000 00000000	
0x000C	-	-	BRPER[B,H,W] -------- ----0000	
0x0010	IF1CMSK[B,H,W] -------- 00000000		IF1CREQ[B,H,W] 0------- 00000001	
0x0014	IF1MSK2[B,H,W] 11-11111 11111111		IF1MSK1[B,H,W] 11111111 11111111	
0x0018	IF1ARB2[B,H,W] 00000000 00000000		IF1ARB1[B,H,W] 00000000 00000000	
0x001C	-	-	IF1MCTR[B,H,W] 00000000 0---0000	
0x0020	IF1DTA2[B,H,W] 00000000 00000000		IF1DTA1[B,H,W] 00000000 00000000	
0x0024	IF1DTB2[B,H,W] 00000000 00000000		IF1DTB1[B,H,W] 00000000 00000000	
0x0028 - 0x002F	-	-	-	-
0x0030	IF1DTA1[B,H,W] 00000000 00000000		IF1DTA2[B,H,W] 00000000 00000000	
0x0034	IF1DTB1[B,H,W] 00000000 00000000		IF1DTB2[B,H,W] 00000000 00000000	
0x0038 - 0x003C	-	-	-	-
0x0040	IF2CMSK[B,H,W] -------- 00000000		IF2CREQ[B,H,W] 0------- 00000001	
0x0044	IF2MSK2[B,H,W] 11-11111 11111111		IF2MSK1[B,H,W] 11111111 11111111	
0x0048	IF2ARB2[B,H,W] 00000000 00000000		IF2ARB1[B,H,W] 00000000 00000000	
0x004C	-	-	IF2MCTR[B,H,W] 00000000 0---0000	
0x0050	IF2DTA2[B,H,W] 00000000 00000000		IF2DTA1[B,H,W] 00000000 00000000	
0x0054	IF2DTB2[B,H,W] 00000000 00000000		IF2DTB1[B,H,W] 00000000 00000000	
0x0058 - 0x005C	-	-	-	-
0x0060	IF2DTA1[B,H,W] 00000000 00000000		IF2DTA2[B,H,W] 00000000 00000000	
0x0064	IF2DTB1[B,H,W] 00000000 00000000		IF2DTB2[B,H,W] 00000000 00000000	
0x0068 - 0x007C	-	-	-	-
0x0080	TREQR2[B,H,W] 00000000 00000000		TREQR1[B,H,W] 00000000 00000000	

Base_Address	Register			
+ Address	+3	+2	+1	+0
0x0084 - 0x008F	-	-	-	-
0x0090	NEWDT2[B,H,W] 00000000 00000000		NEWDT1[B,H,W] 00000000 00000000	
0x0094 - 0x009F	-	-	-	-
0x00A0	INTPND2[B,H,W] 00000000 00000000		INTPND1[B,H,W] 00000000 00000000	
0x00A4 - 0x00AF	-	-	-	-
0x00B0	MSGVAL2[B,H,W] 00000000 00000000		MSGVAL1[B,H,W] 00000000 00000000	
0x00B4 - 0x0FFC	-	-	-	-

SD-Card Base_Address : 0x4006_E000

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000 – 0xFFC	XXXXXXXX	XXXXXXXX	XXXXXXXX	XXXXXXXX

WorkFlash_IF Base_Address : 0x200E_0000

Base_Address + Address	Register			
	+3	+2	+1	+0
0x000	WFASZR[B,H,W]			
0x004	WFRWTR[B,H,W]			
0x008	WFSTR[B,H,W]			
0x00C - 0xFFF	-	-	-	-

Note:

For the register details of Workflash IF block, refer to the "FLASH PROGRAMMING MANUAL" of the product used.

B List of Notes

This section explains notes for each function.

1. Notes when high-speed CR is used for the master clock

CODE: 9BPRECAUTION-E01.3

1. Notes when high-speed CR is used for the master clock

This section explains notes when the high-speed CR is used for the master clock.

The frequency of the high-speed CR varies depending on the temperature and/or the power supply voltage. The following table shows notes on each function macro when the high-speed CR is used for the master clock.

Furthermore, pay attention to notes when the high-speed CR is used as an input clock of the PLL and the master clock is selected for PLL.

● Notes on Each Macro

Macro	Function/mode	Notes
Internal Bus Clock	HCLK/FCLK/PCLK0/ PCLK1/PCLK2/ TPIUCLK	When the frequency of the high-speed CR is the maximum value, the setting of the internal operating clock frequency shall not exceed the upper limit specified in the "data sheet" for the product that you are using.
Timer	Multi-function Timer Base Timer Watch Timer Dual Timer Watch Dog Timer Quadrature	The frequency variation of the high-speed CR should be considered for the timer count value of each macro.
A/D Converter	Sampling Time Compare Time	Considering the frequency variation of the high-speed CR, the sampling time and the compare time of the A/D converter shall satisfy the specification specified in the "data sheet" for the product that you are using.
USB	-	As the frequency accuracy does not meet the required specification, these macros cannot be used when the high-speed CR is used for the master clock.
Ethernet-MAC		
CAN		
Multi Function Serial Interface	UART	Even if the frequency of the high-speed CR is the minimum or the maximum value, the baud rate error should be considered. The baud rate error shall not exceed the limit.
	CSIO	The frequency variation of the high-speed CR should be considered for the communication of each macro.
	I2C	
	LIN	As the required frequency accuracy cannot be met, this function cannot be used as master. As a slave, the specified baud rate has more error at the maximum/minimum frequency of high-speed clock. So, if the error limit of the baud rate is exceeded, this function cannot be used.
Debug Interface	Serial Wire	As the frequency variation of the high-speed CR, the SWV(Serial Wire View) may not be used.
External Bus Interface	Clock Output	When the external bus clock output is used, the frequency variation of the high-speed CR should be considered for devices to be connected.
SD card Interface	-	The frequency variation of the high-speed CR should be considered for devices to be connected.

MN709-00001-1v0-E

FUJITSU SEMICONDUCTOR • CONTROLLER MANUAL

32-BIT MICROCONTROLLER
FM4 Family
PERIPHERAL MANUAL

July 2013 Rev. 1.0

Published **FUJITSU SEMICONDUCTOR LIMITED**
Edited Sales Promotion Dept.