To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: http://www.renesas.com

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (http://www.renesas.com)

Send any inquiries to http://www.renesas.com/inquiry.

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

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

   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.

   "Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

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

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

RENESAS

User's Manual

PFESiP/V850EP1

32-bit Microcontroller Dedicated to PFESiP® EP-1

Hardware (USB Function)

Document No. A19071EJ2V0UM00 (2nd edition)
Date Published November 2009 NS

[MEMO]

NOTES FOR CMOS DEVICES

(1) VOLTAGE APPLICATION WAVEFORM AT INPUT PIN

Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (MAX) and V_{IH} (MIN) due to noise, etc., the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (MAX) and V_{IH} (MIN).

(2) HANDLING OF UNUSED INPUT PINS

Unconnected CMOS device inputs can be cause of malfunction. If an input pin is unconnected, it is possible that an internal input level may be generated due to noise, etc., causing malfunction. CMOS devices behave differently than Bipolar or NMOS devices. Input levels of CMOS devices must be fixed high or low by using pull-up or pull-down circuitry. Each unused pin should be connected to V_{DD} or GND via a resistor if there is a possibility that it will be an output pin. All handling related to unused pins must be judged separately for each device and according to related specifications governing the device.

(3) PRECAUTION AGAINST ESD

A strong electric field, when exposed to a MOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop generation of static electricity as much as possible, and quickly dissipate it when it has occurred. Environmental control must be adequate. When it is dry, a humidifier should be used. It is recommended to avoid using insulators that easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors should be grounded. The operator should be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions need to be taken for PW boards with mounted semiconductor devices.

(4) STATUS BEFORE INITIALIZATION

Power-on does not necessarily define the initial status of a MOS device. Immediately after the power source is turned ON, devices with reset functions have not yet been initialized. Hence, power-on does not guarantee output pin levels, I/O settings or contents of registers. A device is not initialized until the reset signal is received. A reset operation must be executed immediately after power-on for devices with reset functions.

(5) POWER ON/OFF SEQUENCE

In the case of a device that uses different power supplies for the internal operation and external interface, as a rule, switch on the external power supply after switching on the internal power supply. When switching the power supply off, as a rule, switch off the external power supply and then the internal power supply. Use of the reverse power on/off sequences may result in the application of an overvoltage to the internal elements of the device, causing malfunction and degradation of internal elements due to the passage of an abnormal current. The correct power on/off sequence must be judged separately for each device and according to related specifications governing the device.

(6) INPUT OF SIGNAL DURING POWER OFF STATE

Do not input signals or an I/O pull-up power supply while the device is not powered. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Input of signals during the power off state must be judged separately for each device and according to related specifications governing the device.

PFESiP is a registered trademark of NEC Electronics Corporation in Japan, Germany, and United Kingdom.

These commodities, technology or software, must be exported in accordance with the export administration regulations of the exporting country. Diversion contrary to the law of that country is prohibited.

- **The information in this document is current as of November, 2009. The information is subject to change without notice. For actual design-in, refer to the latest publications of NEC Electronics data sheets or data books, etc., for the most up-to-date specifications of NEC Electronics products. Not all products and/or types are available in every country. Please check with an NEC Electronics sales representative for availability and additional information.**
- No part of this document may be copied or reproduced in any form or by any means without the prior written consent of NEC Electronics. NEC Electronics assumes no responsibility for any errors that may appear in this document.
- NEC Electronics does not assume any liability for infringement of patents, copyrights or other intellectual property rights of third parties by or arising from the use of NEC Electronics products listed in this document or any other liability arising from the use of such products. No license, express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC Electronics or others.
- Descriptions of circuits, software and other related information in this document are provided for illustrative purposes in semiconductor product operation and application examples. The incorporation of these circuits, software and information in the design of a customer's equipment shall be done under the full responsibility of the customer. NEC Electronics assumes no responsibility for any losses incurred by customers or third parties arising from the use of these circuits, software and information.
- While NEC Electronics endeavors to enhance the quality, reliability and safety of NEC Electronics products, customers agree and acknowledge that the possibility of defects thereof cannot be eliminated entirely. To minimize risks of damage to property or injury (including death) to persons arising from defects in NEC Electronics products, customers must incorporate sufficient safety measures in their design, such as redundancy, fire-containment and anti-failure features.
- NEC Electronics products are classified into the following three quality grades: "Standard", "Special" and "Specific". The "Specific" quality grade applies only to NEC Electronics products developed based on a customer-designated "quality assurance program" for a specific application. The recommended applications of an NEC Electronics product depend on its quality grade, as indicated below. Customers must check the quality grade of each NEC Electronics product before using it in a particular application.

  "Standard": Computers, office equipment, communications equipment, test and measurement equipment, audio and visual equipment, home electronic appliances, machine tools, personal electronic equipment and industrial robots.

  "Special": Transportation equipment (automobiles, trains, ships, etc.), traffic control systems, anti-disaster systems, anti-crime systems, safety equipment and medical equipment (not specifically designed for life support).

  "Specific": Aircraft, aerospace equipment, submersible repeaters, nuclear reactor control systems, life support systems and medical equipment for life support, etc.

  The quality grade of NEC Electronics products is "Standard" unless otherwise expressly specified in NEC Electronics data sheets or data books, etc. If customers wish to use NEC Electronics products in applications not intended by NEC Electronics, they must contact an NEC Electronics sales representative in advance to determine NEC Electronics' willingness to support a given application.

  (Note 1) "NEC Electronics" as used in this statement means NEC Electronics Corporation and also includes its majority-owned subsidiaries.

  (Note 2) "NEC Electronics products" means any product developed or manufactured by or for NEC Electronics (as defined above).

(M8E0909)

Major Revisions in This Edition

Location	Contents
p.85	Modification of **3.4.1 (1) MEMC_INT**
p.96	Modification of **3.4.4 (3) UF0 EPNAK register (UF0EN)**
pp.104, 105	Modification of **3.4.4 (11) UF0 INT status 0 register (UF0IS0)**
p.122	Modification of **3.4.4 (28) UF0 DMA status 1 register (UF0DMS1)**
p.203	Modification of **3.7.6 Bulk transfer in DMA mode**
p.203	Modification of **Figure 3-26. DMA Initialization Processing**
p.204, 205	Modification of **Figure 3-27. Flow Example of Bulk Transfer Processing in DMA Mode**
p.212	Modification of **4.3 Clock and Reset**
p.212	Modification of **Table 4-2. UCLK Timings**

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

To obtain the latest documents when designing, contact an NEC sales office or a distributor.

PREFACE

Readers This manual is intended for users who understand the functions of the microcontroller function chip with an on-chip V850E2 CPU core (PFESiP/V850EP1) and wish to evaluate developing PFESiP EP-1 Series products using the chip.

Purpose This manual is intended to give users an understanding of the USB functions among the hardware functions of the PFESiP/V850EP1.

How to Read This Manual It is assumed that the readers of this manual have general knowledge in the fields of electrical engineering, logic circuits, microcontrollers, SRAM, page ROM, and SDRAM.

Conventions

Data significance:	Higher digits on the left and lower digits on the right
Active low representation:	xxxZ (Z after pin or signal name)
Note:	Footnote for item marked with **Note** in the text
Caution:	Information requiring particular attention
Remark:	Supplementary information
Numeric representation:	Binary ... xxxx or xxxxB
	Decimal ... xxxx
	Hexadecimal ... xxxxH
Prefix indicating power of 2 (address space, memory capacity):	
	K (kilo): $2^{10} = 1,024$
	M (mega): $2^{20} = 1,024^2$
	G (giga): $2^{30} = 1,024^3$
Data type:	Word ... 32 bits
	Halfword ... 16 bits
	Byte ... 8 bits

Related Documents The related documents indicated in this publication may include preliminary versions. However, preliminary versions are not marked as such. Furthermore, some of the related documents may be documents intended for specific customers, because they are prepared when each core is being developed or planned.

Documents related to PFESiP EP-1 Series

Document Name	Document No.
V850E2 Architecture User's Manual	U17135E
PFESiP EP-1 Series Design Manual	A19068E
PFESiP/V850EP1 Product Data User's Manual	A19069E
PFESiP/V850EP1 Hardware (CPU Function) User's Manual	A19070E
PFESiP/V850EP1 Hardware (USB Function) User's Manual	This manual
PFESiP/V850EP1 USB Function Sample Software Application Note	A19349E

Documents related to PFESiP EP-1 Evaluation Board

Document Name	Document No.
PFESiP EP-1 Evaluation Board Technical Information User's Manual	A19350E
PFESiP EP-1 Evaluation Board Lite Technical Information User's Manual	A19354E

Documents related to development tools

Document Name			Document No.
RX850 Pro () (Real-Time OS)	Ver. 3.21	Basics	U18165E
	Ver. 3.20	Installation	U17421E
	Ver. 3.21	Technical	U18164E
	Ver. 3.20	Task Debugger	U17422E

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1 USB FUNCTION OVERVIEW

PFESiP/V850EP1 incorporates a USB function controller and a USB host controller which comply with the Universal Serial Bus Specification.

1.1 Features

○ Complies with the Universal Serial Bus Specification.

○ USB function controller:

Supports 12 Mbps (full-speed) transfer.
Equipped with a one-channel upstream port.
Incorporates the following transfer endpoints.

Endpoint	FIFO Size (Bytes)	Transfer Type	Remark
EP0	64	Control Read/Write	
EP1	64 × 2	Bulk In	Double buffer configuration
EP2	64 × 2	Bulk Out	Double buffer configuration
EP7	8	Interrupt	

Can perform DMA transfer (two-clock transfer) of bulk in or bulk out data.

○ USB host controller: Supports 12 Mbps (full-speed) and 1.5 Mbps (low-speed) transfer.
Supports the OHCI (Open Host Controller Interface) 1.0a standard.
(The control of transitioning the USB port from disable to enable status, however, is restricted.)
Incorporates a two-channel root hub function and is equipped with two downstream ports.
Uses an 8 KB on-chip SRAM and an external SDRAM as shared memories.

○ Memory bus width: CPU interface….32 bits
External SDRAM interface of the USB host controller….16 or 32 bits

○ Clock input: f_{USB} = 48 MHz (used by the host controller or function controller)
f_{PCLK} = 25 MHz to 33 MHz (used only by the host controller)

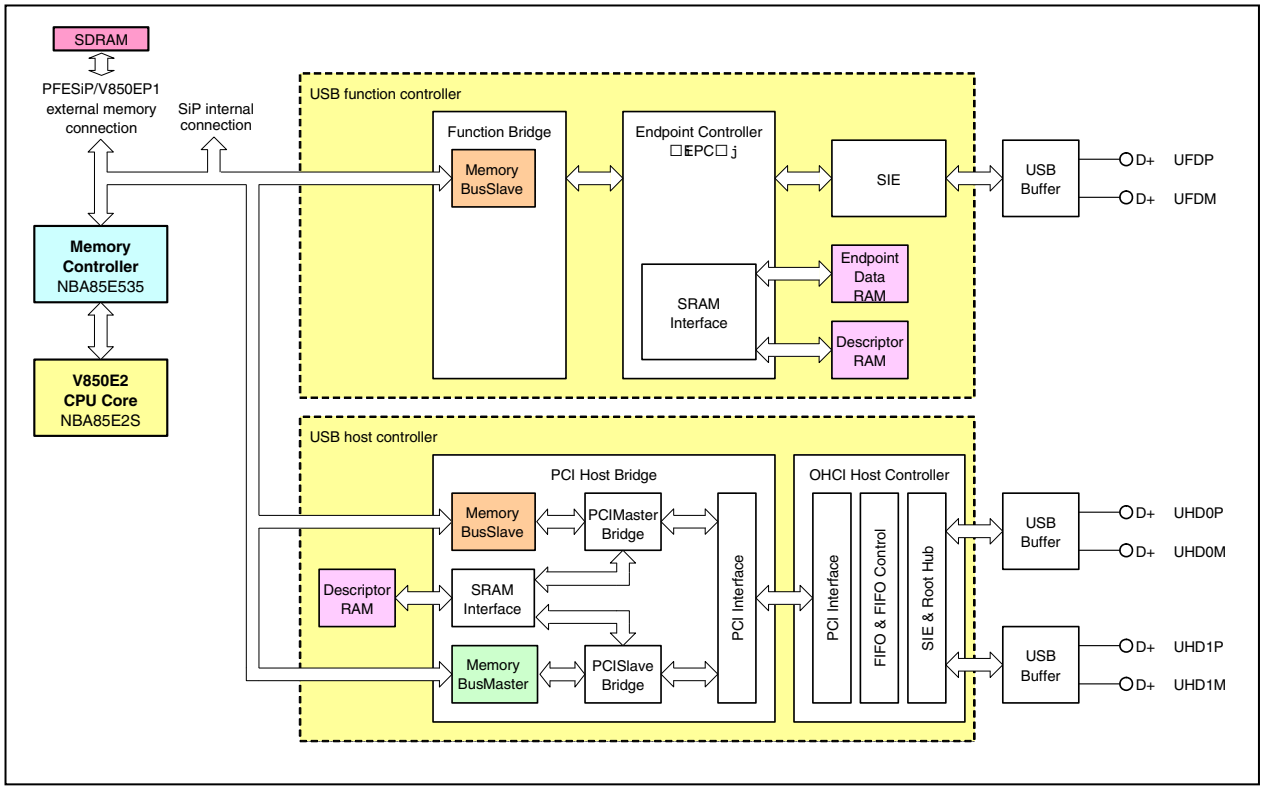
Caution When using the USB function, set the MODE0 and MODE1 pins to the appropriate modes.

MODE1	MODE0	Operation Mode
0	0	USB function invalid
0	1	Only USB function valid
1	0	Only USB host valid
1	1	USB host or USB function valid

When the USB function is disabled, the clock used by the USB function controller or USB host controller is stopped. Do not change these pin settings while PFESiP/V850EP1 is operating.

1.2 Configuration

Figure 1-1. USB Configuration



1.3 USB Pin Functions

Table 1-1. USB Pins

Pin	I/O	Function	Active
UCLK	Input	USB clock signal input	–
UHD0P	I/O	USB host channel 0 data I/O (+)	–
UHD0M	I/O	USB host channel 0 data I/O (–)	–
UHD1P	I/O	USB host channel 1 data I/O (+)	–
UHD1M	I/O	USB host channel 1 data I/O (–)	–
PPON0	Output	USB host channel 0 power supply control output	High
PPON1	Output	USB host channel 1 power supply control output	High
OCI0	Input	USB host channel 0 overcurrent detection input	Low
OCI1	Input	USB host channel 1 overcurrent detection input	Low
UFDP	I/O	USB function data I/O (+)	–
UFDM	I/O	USB function data I/O (–)	–
VBUSDET	Input	USB function insertion and removal detection input	High
PCLKIN	Input	USB bus bridge clock	–
UCLKSEL0	Input	USB clock selection input 0: Selects XT1, XT2. 1: Selects UCLK.	–
UCLKSEL1	Input	USB bus bridge clock selection input 0: Selects XT1, XT2. 1: Selects PCLKIN.	–
MODE0, MODE1	Input	PFESiP/V850EP1 USB operation mode setting	–

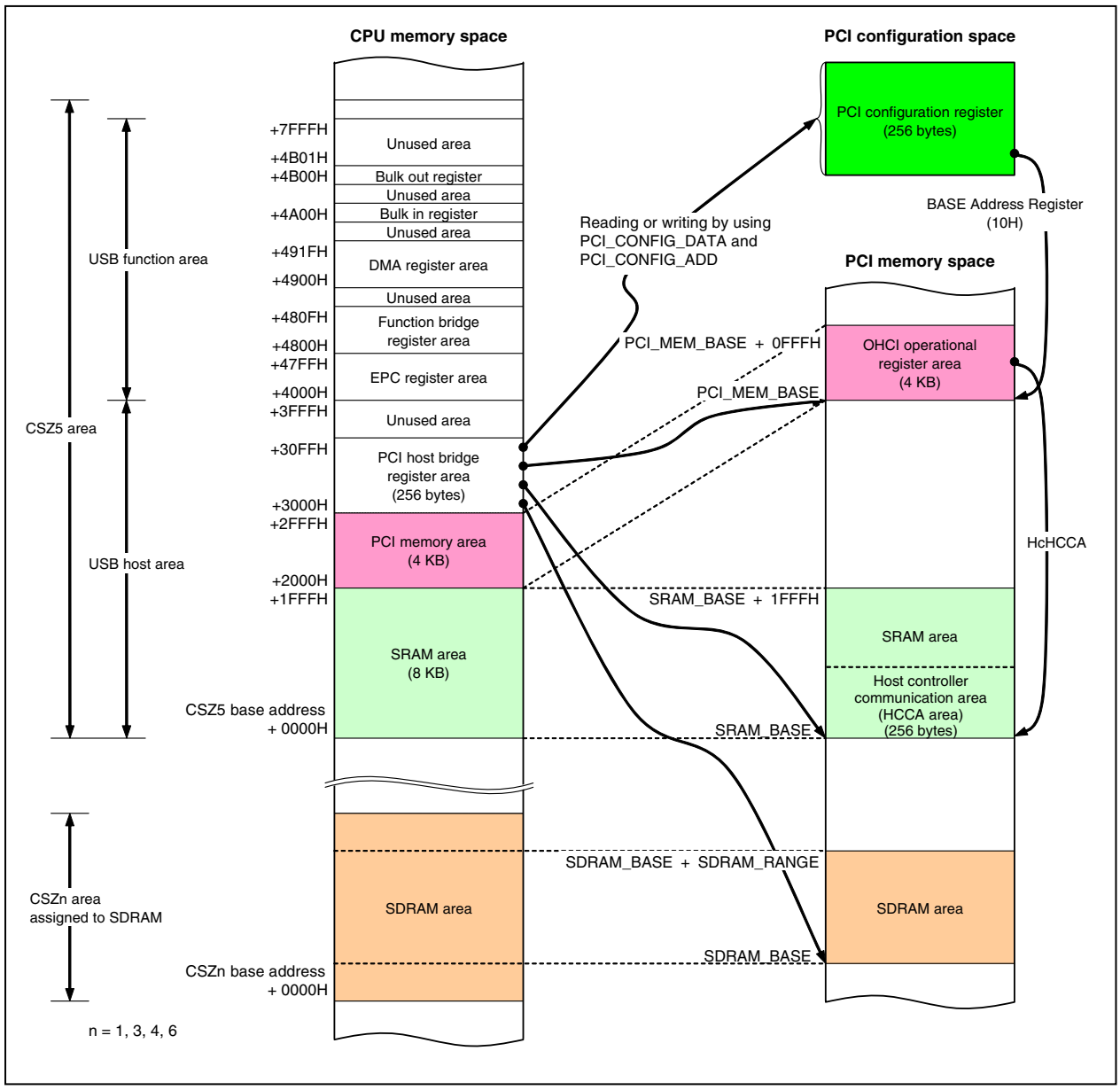
1.4 USB Memory Map

Both the USB host controller and the USB function controller are assigned to the CSZ5 space. Set the CSZ5 address range (subarea) where they are placed via the CSC1 register. Set the CSZ5 area for an SRAM and an I/O via the BCT1 register, and set the bus width as 32 bits via the LBS register.

The USB host controller has an external memory bus master function, and can use any SDRAM to which any chip select signal from among CSZ1, CSZ3, CSZ4, and CSZ6 has been assigned for use as the SDRAM interface via the UCSS register. At that time, set the bus width of the SDRAM to be shared to the same bus width (16 or 32 bits) as for SDRAM_CTL, which is used in the LBS register and USB host controller.

The OHCI host controller, which is the central function of the USB host controller, is connected via the SRAM interface of the CSZ5 space and also via the internal PCI bus bridge. The CPU and the internal PCI bus bridge share the same memory, and the CPU memory space is linked to the PCI memory space. This relationship is defined by the PCI host bridge registers and the PCI configuration register, which is accessed via the PCI host bridge registers.

Figure 1-2. USB Memory Map



1.5 USB-related Interrupts

Table 1-2. USB-related Interrupt Sources

Maskable Interrupt Source				Default Priority	Exception Code	Handler Address	Restored PC	DMA Transfer Source
Name	Control Register	Generating Source	Generating Unit					
INTUSBH0	UHIC0 (1FFFF182H)	USBH interrupt 0 (INTA, SMMI, PME)	USB host controller (OHCI HC)	57	0410H	00000410H	Next PC	○
INTUSBH1	UHIC1 (1FFFF184H)	USBH interrupt 1 (PME)	USB host controller (OHCI HC)	58	0420H	00000420H	Next PC	○
INTUSBF0	UFIC0 (1FFFF186H)	USBF interrupt 0 (endpoint event)	USB function controller (EPC, function bridge)	59	0430H	00000430H	Next PC	○
INTUSBF1	UFIC1 (1FFFF188H)	USBF interrupt 1 (endpoint1 DMA transfer completion)	USB function controller (function bridge)	60	0440H	00000440H	Next PC	○
INTUSBF2	UFIC2 (1FFFF18AH)	USBF interrupt 2 (endpoint2 DMA transfer completion)	USB function controller (function bridge)	61	0450H	00000450H	Next PC	○
INTUSBF3	UFIC3 (1FFFF18CH)	USBF interrupt 3 (Resume)	USB function controller (SIE)	62	0460H	00000460H	Next PC	○
INTUSBF4	UFIC4 (1FFFF18EH)	USBF interrupt 4 (VBUS detection)	USB function controller external pin	63	0470H	00000470H	Next PC	○

1.6 USB-related I/O Register Settings

To use the USB function, set the peripheral I/O registers as follows.

Table 1-3. Peripheral I/O Register Settings When Using USB Function

Abbreviation	Address	Setting Value	Remark
CSC1	1FFF F062H	Assign CSZ5 to any subarea (70, 71, 72, or 73).	When using CSREMAP, map CSZ5 under the default settings. The chip select signal that is used by the USB function is fixed to CSZ5, so it cannot be changed to another CSZn.
BCT1	1FFF F482H	xx8xH	Select an SRAM or I/O for CSZ5.
LBS	1FFF F48EH	xxxx11xx xxxxxxxxB	The width of CSZ5 is 32 bits.
DWC1	1FFF F486H	Any value	Set the CSZ5 data wait to any value (0 to 7).
BCC	1FFF F488H	Any value	Set the CSZ5 idle state to any value (0 to 3).
ASC	1FFF F48AH	Any value	Set the CSZ5 address setting wait to any value (0 to 3).
UCSS	1FFF F8B6H	Assign the SDRAM to any chip select signal from among CSZ1, CSZ3, CSZ4, and CSZ6.	This is used by the USB host controller. Set it to the same CSZ signal as of the SDRAM that is shared with the CPU. When using CSREMAP, it is the remapped chip select signal that is assigned via UCSS.
UDMS	1FFF F8B8H	Assign two of the four channels used for DMA interface signals as DMA interface signals for USB.	This is used by the USB function controller. Channel 0 corresponds to EP1 (Bulk in) and channel 1 corresponds to EP2 (Bulk out).

When using CSREMAP, the USB function uses the address space that has been assigned to the remapped MEM_CSZn (logically equivalent to CSZn).

CHAPTER 2 USB HOST CONTROLLER

2.1 Overview

The USB host controller uses a token based protocol to transfer data to and from external function devices via the polling method.

It complies with the OHCI (Open Host Controller Interface) 1.0a standard, and is equipped with a two-channel root hub function and two downstream ports. For data transfers with external function devices, all transfer types (control, bulk, isochronous, interrupt) can be used (however, for transfers such as isochronous transfers that involve a heavy system load, a performance evaluation is needed beforehand).

2.2 PCI Host Bridge

2.2.1 PCI host bridge function

The PCI host bridge is a bridge circuit that is connected from the CPU system to the OHCI host controller, via the PCI bus, and it includes the following functions.

- PCI master cycle control

  The following are issued in response to a PCI bus access request from the CPU (MEMC):

  PCI Configuration Register Read/Write Single Cycle

  PCI Memory Read/Write Cycle

- PCI slave cycle control

  When the SDRAM area or SRAM area is accessed from the PCI bus, a PCI memory read/write cycle (burst transfer of up to eight words) is accepted.

- PCI error processing

  An error interrupt is generated for a master abort, target abort, PERR# reception, or SERR# reception.
  (The address immediately before the error occurred is retained.)

- PFESiP/V850EP1 on-chip memory controller (NBA85E535) bus control

  When access from the CPU occurs via the memory controller bus, a hardware wait (WAIT) is used to control the bus cycle.

- SRAM control

  An 8 KB SRAM is incorporated as a shared memory. It is mainly used for assignment of descriptors.
  SRAM area access from the CPU (MEMC) and the PCI bus is arbitrated and controlled.

- SDRAM control

  An SDRAM that is externally connected to PFESiP/V850EP1 is used as a shared memory. In response to SDRAM area access from the PCI bus, bus request handshaking occurs with the memory controller, after which the SDRAM is controlled. Both 16-bit and 32-bit SDRAM data widths are supported.

2.2.2 CPU memory space

The CPU memory space is divided for use as described below.

The on-chip SRAM (8 KB) and external SDRAM that are used only by the USB function are assigned as shared memories, and the PCI memory area and the PCI host bridge register area are assigned to the CPU memory space.

Table 2-1. CPU Memory Space Division

Base Address	Offset Address	Area
Address selected via CSZ5	0000H to 1FFFH	USB-dedicated SRAM area (8 KB)
	2000H to 2FFFH	PCI memory area (4 KB)
	3000H to 30FFH	PCI host bridge register area (256 bytes)
	3100H to 3FFFH	Reserved
CSZn selected via UCSS register	0000H to any address	SDRAM area

Remark n = 1, 3, 4, 6

2.2.3 PCI host bridge registers

The PCI bridge is provided with the following PCI host bridge registers. Since they are freely mapped by CSZ5, the CSZ5 start address is set as the base address.

(Register address = base address + offset address 1 + offset address 2)

They can be accessed only in 32-bit units.

Table 2-2. PCI Host Bridge Registers

Base Address	Offset Address 1	Offset Address 2	Register	Symbol	R/W
CSZ5 start address	3000H	00H	PCI Configuration Data Register	PCI_CONFIG_DATA	R/W
		04H	PCI Configuration Address Register	PCI_CONFIG_ADD	R/W
		08H	PCI Control 1 Register	PCI_CONTROL1	R/W
		0CH	PCI Control 2 Register	PCI_CONTROL2	R/W
		10H	Reserved	–	–
		14H	PCI Memory Base Address Register	PCI_MEM_BASE	R/W
		18H	PCI Interrupt Status Register	PCI_INT_STATUS	R/W
		1CH	PCI Interrupt Control Register	PCI_INT_CTL	R/W
		20H	PCI Bus Error Address	PCI_ERR_ADD	R
		24H to 3FH	Reserved	–	–
		40H	SDRAM Area Base Address Register	SDRAM_BASE	R/W
		44H	SDRAM Area Address Range Register	SDRAM_RANGE	R/W
		48H	SDRAM Control Register	SDRAM_CTL	R/W
		4CH	Reserved	–	–
		50H	SRAM Area Base Address Register	SRAM_BASE	R/W
		54H to FFH	Reserved	–	–

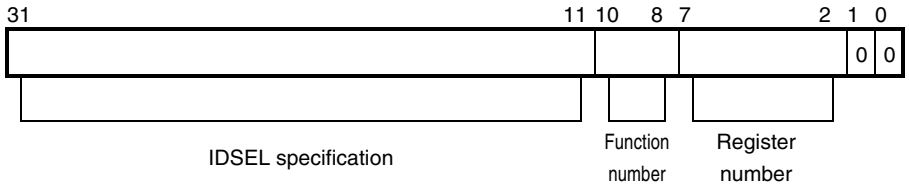
(1) PCI Configuration Data Register

	31 to 0	Offset address	After reset
PCI_CONFIG_DATA	cdata [31:0]	00H	Undefined
	R/W		

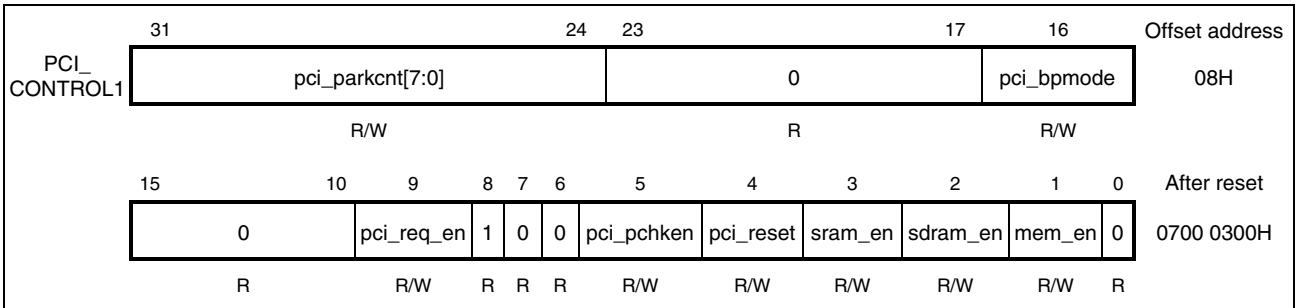
Bit position	Bit name	Function
31 to 0	cdata	The PCI configuration register can be accessed via this register. First, the PCI_CONFIG_ADD register must be set.

(2) PCI Configuration Address Register

	31 to 0	Offset address	After reset
PCI_CONFIG_ADD	cadd [31:0]	04H	0000 0000H
	R/W		

Bit position	Bit name	Function
31 to 0	cadd	These bits set the address of the PCI configuration register. How to set the PCI configuration address register  ○ Register number: Indicates the number of the PCI configuration register. ○ Function number: Indicates the function number of a PCI multi function device. ○ IDSEL specification: Selects the IDSEL signal of the PCI device to be accessed. The PCI host bridge substitutes AD [31:11] as the IDSEL signal, so configuration access to the PCI device connected to the PCI bus is set by setting only one of the bits corresponding to the AD signal that is connected to the respective IDSEL pin. With PFESiP/V850EP1, since the AD31 signal is connected to the IDSEL pin, setting (1) the cadd31 bit enables access to the PCI device (OHCI host controller). How to access the PCI configuration register The PCI configuration register can be accessed via the following procedure. ○ PCI_CONFIG_ADD register setting Set the address of the configuration cycle. ○ PCI_CONFIG_DATA register setting Read access to the address set to PCI_CONFIG_ADD is achieved by reading the PCI_CONFIG_DATA register. Also, write access can be executed by writing to PCI_CONFIG_DATA.

(3) PCI Control 1 Register



Bit position	Bit name	Function
31 to 24	pci_parkcnt[7:0]	PCI Bus Parking Timer These bits set the time for switching to bus parking. Counting starts when FRAME# = 1 & IRDY# = 1. These bits can be left set to the default value.
16	pci_bpmode	PCI Bus Parking Mode 0: Bus parking master is limited to this macro (default value). 1: Bus parking master is the master accessed last. This bit can be left set to the default value.
9	pci_req_en	PCI Request Enable 0: Request invalid 1: Request valid (default value) This bit can be left set to the default value.
5	pci_pchken	PCI Parity Check Enable 0: Invalidates parity check in PCI bus (default value). 1: Validates parity check in PCI bus. This bit can be left set to the default value.
4	pci_reset	PCI Reset 0: PCI bus is reset (default value). 1: PCI bus is released from reset. Set (1) this bit when accessing the OHCI host controller.
3	sram_en	SRAM area enable 0: Access to SRAM area from PCI bus is not supported (default value). 1: Access to SRAM area from PCI bus is supported. Set (1) this bit when starting access to the SRAM from the OHCI host controller.
2	sdram_en	SSDRAM area enable 0: Access to SDRAM area from PCI bus is not supported (default value). 1: Access to SDRAM area from PCI bus is supported. Set (1) this bit when starting access to the SDRAM from the OHCI host controller.
1	mem_en	PCI memory area enable 0: Disables access to PCI memory area from CPU (default value). 1: Enables access to PCI memory area from CPU. Set (1) this bit when accessing the OHCI register of the OHCI host controller.

(4) PCI Control2 Register

	31 9	8	7	6 4	3	2 0	Offset address	After reset
PCI_CONTROL2	0	wbuf_busy	0	wburst_size[2:0]	0	rburst_size[2:0]	0CH	0000 0077H
	R	R	R	R/W	R	R/W		

Bit position	Bit name	Function
8	wbuf_busy	PCI Write Buffer Busy This indicates the data status of the write buffer when writing to the PCI target (SDRAM). 0: Write buffer has no data to be written to SDRAM or SRAM. 1: Write buffer has data to be written to SDRAM or SRAM.
6 to 4	wburst_size[2:0]	PCI Write Burst Max Size These bits set the maximum burst length when writing to the PCI target (SDRAM). These bits can be left set to the default value.
2 to 0	rburst_size[2:0]	PCI Read Burst Max Size These bits set the maximum burst length when reading the PCI target (SDRAM). These bits can be left set to the default value.

wburst_size			Maximum burst length when writing to PCI target
2	1	0	
0	0	0	Only single transfer supported
0	0	1	2 bursts
0	1	0	3 bursts
0	1	1	4 bursts
1	0	0	5 bursts
1	0	1	6 bursts
1	1	0	7 bursts
1	1	1	8 bursts (default value)

wburst_size			Maximum burst length when reading PCI target
2	1	0	
0	0	0	Only single transfer supported
0	0	1	2 bursts
0	1	0	3 bursts
0	1	1	4 bursts
1	0	0	5 bursts
1	0	1	6 bursts
1	1	0	7 bursts
1	1	1	8 bursts (default value)

(5) PCI Memory Base Address Register

	31 12	11 0	Offset address	After reset
PCI_MEM_BASE	m_base[31:12]	0	14H	0000 2000H
	R/W	R		

Bit position	Bit name	Function
31 to 12	m_base[31:12]	PCI Memory Base Address Register These bits set the PCI memory base address for the higher address to be accessed from the PCI memory area. When memory access to the PCI bus memory space occurs via the PCI memory area, the setting of this register enables any area in the 4 GB PCI bus memory space to be accessed.

(6) PCI Interrupt Status Register

This register can be used to find the generating sources of the INTUSBH0 and INTUSBH1 signals that are output from the PCI host bridge. The interrupt sources of inta, int_smmi, and int_pme are cleared (0) at the generating sources. The interrupt sources of serr, perr, mabort, and tabort are used only when debugging, and are normally not used.

How to validate each interrupt source is explained in the description of the PCI_INT_CTL register, on the next page.

	31 12	11	10	9	8	7 4	3	2	1	0	Offset address	After reset
PCI_INT_STATUS	0	Int_ pme	Int_ smmi	0	Inta	0	serr	perr	mabor t	tabort	18H	0000 0000H
	R	R	R	R	R	R	R/W	R/W	R/W	R/W		

Bit position	Bit name	Function
11	int_pme	PCI Interrupt PME Status 0: No interrupt source 1: Generates interrupt by PME. The interrupt source is cleared (0) at the PME generating source.
10	Int_smmi	PCI Interrupt SMMI Status 0: No interrupt source 1: Generates interrupt by SMMI. The interrupt source is cleared (0) at the SMMI generating source.
8	inta	PCI Interrupt INTA Status 0: No interrupt source 1: Interrupt generated by INTA The interrupt source is cleared (0) at the INTA generating source.
3	serr	PCI Host Bridge System Error Interrupt Status 0: No interrupt source 1: Detects system error. When "1" is written, the interrupt source is cleared (0). This interrupt is used only when debugging, and is normally not used.
2	perr	PCI Host Bridge Parity Error Interrupt Status 0: No interrupt source 1: Detects parity error. When "1" is written, the interrupt source is cleared (0). This interrupt is used only when debugging, and is normally not used.
1	mabort	PCI Host Bridge Master Abort Interrupt Status 0: No interrupt source 1: Receives master abort. When "1" is written, the interrupt source is cleared (0). This interrupt is used only when debugging, and is normally not used.
0	tabort	PCI Host Bridge Target Abort Interrupt Status 0: No interrupt source 1: Receives target abort. When "1" is written, the interrupt source is cleared (0). This interrupt is used only when debugging, and is normally not used.

(7) PCI Interrupt Control Register

	31 12	11	10	9	8	7 4	3	2	1	0	Offset address	After reset
PCI_INT_CTL	0	int_pme_en	int_smmi_en	0	inta_en	0	serrint_en	perrint_en	mabortint_en	tabortint_en	1CH	0000 0000H
	R	R/W	R/W	R	R/W	R	R/W	R/W	R/W	R/W		

Bit position	Bit name	Function
11	int_pme_en	PCI Interrupt PME Enable 0: Invalidates as interrupt source (default value). 1: Validates as interrupt source.
10	int_smmi_en	PCI Interrupt SMMI Enable 0: Invalidates as interrupt source (default value). 1: Validates as interrupt source.
8	inta_en	PCI Interrupt INTA Enable 0: Invalidates as interrupt source (default value). 1: Validates as interrupt source.
3	serrint_en	PCI Host Bridge System Error Interrupt Enable 0: Invalidates as interrupt source (default value). 1: Validates as interrupt source. This interrupt is used only when debugging, and is normally not used.
2	perrint_en	PCI Host Bridge Parity Error Interrupt Enable 0: Invalidates as interrupt source (default value). 1: Validates as interrupt source. This interrupt is used only when debugging, and is normally not used.
1	mabortint_en	PCI Host Bridge Master Abort Interrupt Enable 0: Invalidates as interrupt source (default value). 1: Validates as interrupt source. This interrupt is used only when debugging, and is normally not used.
0	tabortint_en	PCI Host Bridge Target Abort Interrupt Enable 0: Invalidates as interrupt source (default value). 1: Validates as interrupt source. This interrupt is used only when debugging, and is normally not used.

(8) PCI Bus Error Address

	31 to 0	Offset address	After reset
PCI_ERR_ADD	err_adr[31:0]	20H	0000 0000H
	R		

Bit position	Bit name	Function
31 to 0	err_adr[31:0]	PCI Bus Error Address This register retains the PCI address when any of the following error sources occur. ○ SERR# input ○ Parity error occurrence ○ PCI bus master abort occurrence ○ PCI bus target abort occurrence Read access clears (0) all bits. Once a bus error has occurred and values have been set to this register, they are retained until they are read or updated by a new bus error. This function is used only when debugging, and is normally not used.

(9) SDRAM Area Base Address Register

	31 16	15 0	Offset address	After reset
SDRAM_BASE	s_base[31:16]	0	40H	0000 0000H
	R/W	R		

Bit position	Bit name	Function
31 to 16	s_base[31:16]	SDRAM Area Base Address These bits set the start address when accessing the SDRAM area. They can be set in 64 KB units. This register sets the base address of the SDRAM area within the PCI space. The settings of this register and the SDRAM_RANGE register respond to matched addresses when memory access has occurred from the PCI target.

(10) SDRAM Area Address Range Register

	31 16	15 0	Offset address	After reset
SDRAM_RANGE	s_range[31:16]	FFFFH	44H	0000 FFFFH
	R/W	R		

Bit position	Bit name	Function
31 to 16	s_range[31:16]	These bits set the range of the SDRAM area. They can be set in 64 KB units.

s_range[31:16]	SDRAM area range
000H	64 KB
:	:
00FH	1 MB
:	:
0FFH	16 MB
:	:
1FFH	32 MB

This register sets the range of the SDRAM area within the PCI space.
The SDRAM start address is set to the SDRAM_BASE register and this register is used to set the range.

(11) SDRAM Control Register

The SDRAM_CTL register sets control of access to the SDRAM area.

	31 28	27 16	Offset address
SDRAM_CTL	0	cycle_latency[11:0]	48H
	R	R/W	

15 13	12	11	10	9 8	7 6	5 4	3 2	1 0	After reset
0	bus_size	0		case_latency[1:0]	0	wait_state[1:0]	0	column_size[1:0]	0007 0230H
R	R/W	R		R/W	R	R/W	R	R/W	

Bit position	Bit name	Function
27 to 16	cycle_latency[11:0]	SDRAM Access Latency These bits set the burst data interval in BUSCLK units during access from a PCI device to the SDRAM. These bits can be left set to the default value.
12	bus_size	This bit sets the data bus width of the SDRAM. 0: 16 bits (default value) 1: 32 bits
9, 8	cas_latency[1:0]	These bits set the CAS latency of the SDRAM.
5, 4	wait_state[1:0]	These bits set a wait during SDRAM ACT → CMD, PRE → ACT, CMD → ACT operations.

cycle_latency[11:0]	SDRAM access latency
000H	No latency
:	:
007H	7 × BUSCLK (default value)
:	:
FFFH	4095 × BUSCLK

cas_latency[1:0]		SDRAM CAS latency
1	0	
0	0	Setting prohibited
0	1	CAS latency = 1
1	0	CAS latency = 2 (default value)
1	1	CAS latency = 3

wait_state[1:0]		Wait during ACT → CMD, PRE → ACT, CMD → ACT operations
1	0	
0	0	Setting prohibited
0	1	1 × BUSCLK
1	0	2 × BUSCLK
1	1	3 × BUSCLK (default value)

(2/2)

Bit position	Bit name	Function
1, 0	column_size[1:0]	These bits set the column size of the SDRAM.

column_size[1:0]		SDRAM column size
1	0	
0	0	8 bits (default value)
0	1	9 bits
1	0	10 bits
1	1	11 bits

The following shows the physical addresses where the address signals (A25 to A1) output during access to the SDRAM area are assigned.

Table 2-3. SDRAM Access Row Address Output

(a) Row address output

Address pin	A25-A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1
Row address (column_size[1:0] = 00)	a25-a18	a25	a24	a23	a22	a21	a20	a19	a18	a17	a16	a15	a14	a13	a12	a11	a10	a9
Row address (column_size[1:0] = 01)	a25-a18	a17	a25	a24	a23	a22	a21	a20	a19	a18	a17	a16	a15	a14	a13	a12	a11	a10
Row address (column_size[1:0] = 10)	a25-a18	a17	a16	a25	a24	a23	a22	a21	a20	a19	a18	a17	a16	a15	a14	a13	a12	a11
Row address (column_size[1:0] = 11)	a25-a18	a17	a16	a15	a25	a24	a23	a22	a21	a20	a19	a18	a17	a16	a15	a14	a13	a12

(b) Column address output (when the all-bank precharge command is used)

Address pin	A25-A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1
Column address (bus_size = 0)	a25-a18	a17	a16	a15	a14	a12	a11	1	a10	a9	a8	a7	a6	a5	a4	a3	a2	a1
Column address (bus_size = 1)	a25-a18	a17	a16	a15	a14	a12	1	a11	a10	a9	a8	a7	a6	a5	a4	a3	a2	a1

(c) Column address output (when a read or write command is used)

Address pin	A25-A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1
Column address (bus_size = 0)	a25-a18	a17	a16	a15	a14	a12	a11	0	a10	a9	a8	a7	a6	a5	a4	a3	a2	a1
Column address (bus_size = 1)	a25-a18	a17	a16	a15	a14	a12	0	a11	a10	a9	a8	a7	a6	a5	a4	a3	a2	a1

Remark When compared to the memory controller (NBA85E535) on the CPU side, different column addresses are assigned to address pins A12 and A13, and the column addresses corresponding to both the NBA85E535 and the USB host controller are up to 11 bits, and A12 and above are not used.

(12) SRAM Area Base Address Register

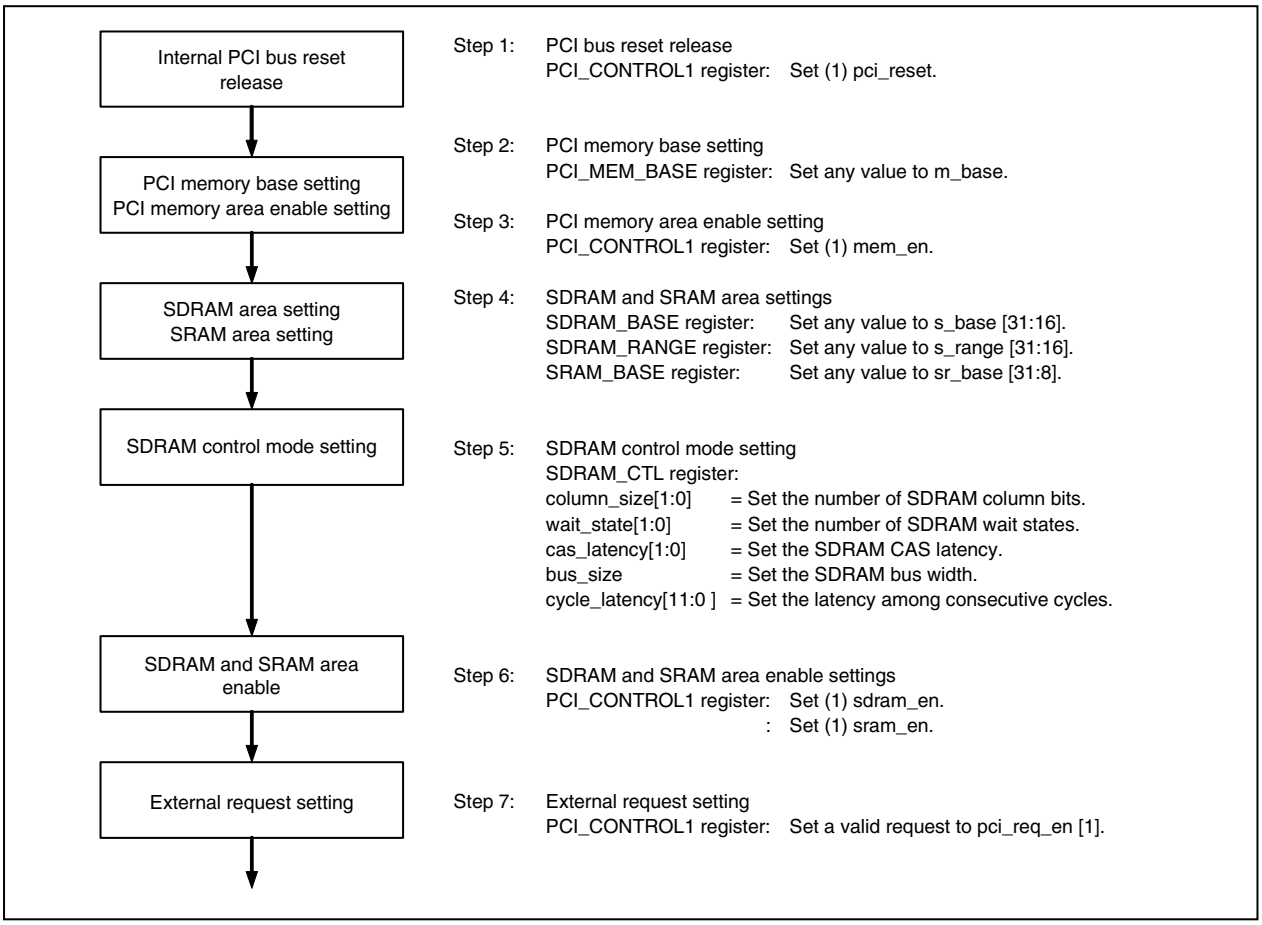
	31 13	12 0	Offset address	After reset
SRAM_BASE	sram_base[31:13]	0	50H	0000 0000H
	R/W	R		

Bit position	Bit name	Function
31 to 13	sram_base[31:13]	SRAM Area Base Address These bits set the base address when accessing the SRAM area. This register sets the base address of the SRAM area within the PCI space. The setting of this register responds to a matched address when memory access has occurred from the PCI target.

2.2.4 How to initialize PCI host bridge

The PCI host bridge macro must be initialized via the following procedure in order to accept memory access to the PCI bus as well as access from the PCI bus to the SDRAM and SRAM areas.

Figure 2-1. PCI Host Bridge Macro Initialization Procedure



The following becomes possible when the above initialization procedure has been completed:

- Access from the PCI_CONFIG_ADD/PCI_CONFIG_DATA register to the PCI configuration register of a PCI device (OHCI host controller)
- Access from the CPU to the PCI memory area (OHCI operational register) of a PCI device (OHCI host controller)
- Access from a PCI device (OHCI host controller) to the SDRAM and SRAM areas

2.3 OHCI Host Controller

2.3.1 OHCI host controller functions

The OHCI host controller is provided with the following functions.

- Complies with the OpenHCI Specification, Release 1.0a.
- Complies with the Universal Serial Bus Specification, Revision 1.1.
  Full-speed (12 Mbps) and low-speed (1.5 Mbps) USB devices are supported.
- Incorporates a two-channel root hub and supports two downstream ports.
- USB clock: 48 MHz, PCI clock: 25 to 33 MHz
- Memory space
  A 4 KB PCI memory area (OHCI operational register) is assigned.
  A 256-byte host controller communication area (HCCA) is assigned.
- Communication with the CPU
  Communicates via the operational register within the OHCI host controller and via the host controller communication area (HCCA).

There are two communication channels between the CPU and the OHCI host controller. One of these communication channels is the OHCI operational register and the target (slave) for these communications is the OHCI host controller. The BASE_Address_Register (10h) that is within the PCI configuration register is a pointer to the OHCI operational register.

Also, the OHCI operational register contains a pointer to a shared memory called the host controller communication area (HCCA), and this HCCA comprises the second communication channel. For these communications, the OHCI host controller is the master.

The descriptor information for communication is managed via the OHCI operational register and the HCCA area.

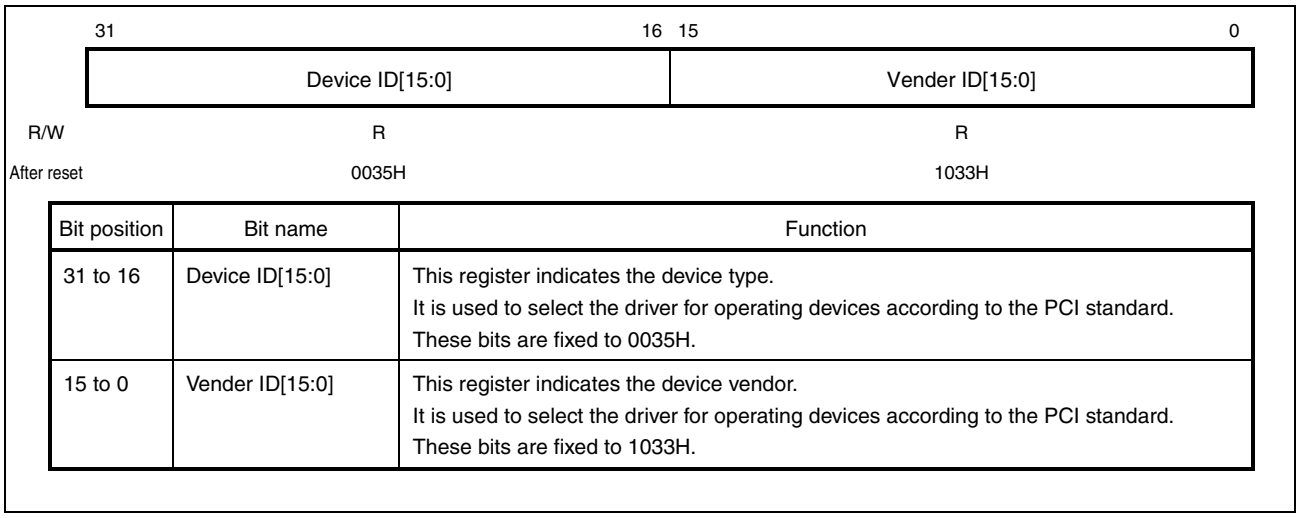
2.3.2 PCI configuration register

The PCI configuration register occupies a 256-byte register space that is incorporated in the OHCI host controller. It is accessed from the CPU system via the PCI host bridge registers (PCI_CONFIG_DATA and PCI_CONFIG_ADD).

Table 2-4. PCI Configuration Registers

Address	31 24	23 16	15 8	7 0
00H	Device ID		Vendor ID	
04H	Status		Command	
08H	Class Code			Revision ID
0CH	BIST	Header Code	Latency Timer	Cache Line Size
10H	Base Address Register			
14H	Reserved			
18H				
1CH				
20H				
24H				
28H	Reserved			
2CH	Subsystem ID		Subsystem Vendor ID	
30H	Reserved			
34H	Reserved			Cap_ptr
38H	Reserved			
3CH	Max_lat	Min_Gnt	Interrupt Pin	Interrupt Line
40H	PMC		Next_Item_Ptr	Cap_ID
44H	Data	DMCSR_BSE	PMCSR	
E0H	Reserved			

(1) Vender ID, Device ID (Offset 00H)



Bit position	Bit name	Function
31 to 16	Device ID[15:0]	This register indicates the device type. It is used to select the driver for operating devices according to the PCI standard. These bits are fixed to 0035H.
15 to 0	Vender ID[15:0]	This register indicates the device vendor. It is used to select the driver for operating devices according to the PCI standard. These bits are fixed to 1033H.

(2) Command, Status (Offset 04H)

(1/2)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Detected Parity Error	Signaled System Error	Received Master Abort	Received Target Abort	Signaled Target Abort	Devsel Timing[1:0]		Data Parity Detected	Fast Back to Back Capable			Capabilities											Fast Back to Back Enable	SERR Enable	Wait Cycle Control	Parity Error Response	VGA Pallet Snoop	Memory Write and Invalidate	Special Cycle	Bus Master	Memory Space	I/O Space
R/W	R/W	R/W	R/W	R/W	R		R/W	R			R											R	R/W	R	R/W	R	R	R	R/W	R/W	R
After reset	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position	Bit name	Function
31	Detected Parity Error	This is the parity error status bit. It is set (1) when an address or data parity error is detected. It is cleared (0) when "1" is written from the PCI bus.
30	Signaled System Error	This is the SERR status bit. It is set (1) when a system error occurs. It is cleared (0) when "1" is written from the PCI bus.
29	Received Master Abort	This is the master abort status bit of the master. It is set (1) when a master operation is terminated upon a master abort. It is cleared (0) when "1" is written from the PCI bus.
28	Received Target Abort	This is the target abort status bit of the master. It is set (1) when a master operation is terminated upon a target abort. It is cleared (0) when "1" is written from the PCI bus.
27	Signaled Target Abort	This is the target abort status bit of a slave. It is set (1) when a slave operation is terminated upon a target abort. It is cleared (0) when "1" is written from the PCI bus.
26, 25	Devsel Timing[1:0]	This field indicates the DEVSEL response speed. It is fixed to 01b due to MediumMode.
24	Data Parity Detected	This bit is set (1) when a parity error is detected during a master operation. It is cleared (0) when "1" is written from the PCI bus. It is fixed to “0” when the parity error response (command register) is disabled.
23	Fast Back to Back Capable	This bit indicates whether "Fast Back to Back" is supported. It is fixed to “0” since "Fast Back to Back" is not supported.
22, 21	–	Reserved (Be sure to write “0”.)
20	Capabilities	This bit indicates that Power Management Mode is supported. It is fixed to 01b.
19 to 10	–	Reserved (Be sure to write “0”.)
9	Fast Back to Back Enable	This is the "Fast Back to Back" enable bit. It is fixed to “0” since the host controller does not support "Fast Back to Back".
8	SERR Enable	This is the SERR enable bit. Set this bit to “1" when transmitting a system error to the SERR signal.
7	Wait Cycle Control	This is the wait cycle control enable bit. It is fixed to “0” since the host controller does not support address/data stepping.
6	Parity Error Response	This is the parity error response enable bit. Set this bit to “1” when performing a parity check error.

(2/2)

Bit position	Bit name	Function
5	VGA Pallet Snoop	This is the VGA pallet snoop enable bit. It is fixed to “0” since the host controller does not support VGA pallet snoop.
4	Memory Write and Invalidate	This is the memory write and invalidate enable bit. It is fixed to “0” since the host controller does not support memory writing and invalidating.
3	Special Cycle	This is the special cycle enable bit. It is fixed to “0” since the host controller does not support a special cycle.
2	Bus Master	This is the bus master enable bit. It is the enable signal that is used for master access to the PCI bus, and must be set to “1” when accessing the SRAM of the system bus. Set this bit to 1b when initializing the host controller.
1	Memory Space	This is the memory space access enable bit. It is the enable signal that is used for memory access according to the PCI standard, and must be set to “1” when accessing registers. Set this bit to 1b when initializing the host controller.
0	I/O Space	This is the I/O space access enable bit. It is the enable signal that is used for I/O access according to the PCI standard, but is fixed to “0” since the host controller does not use I/O access.

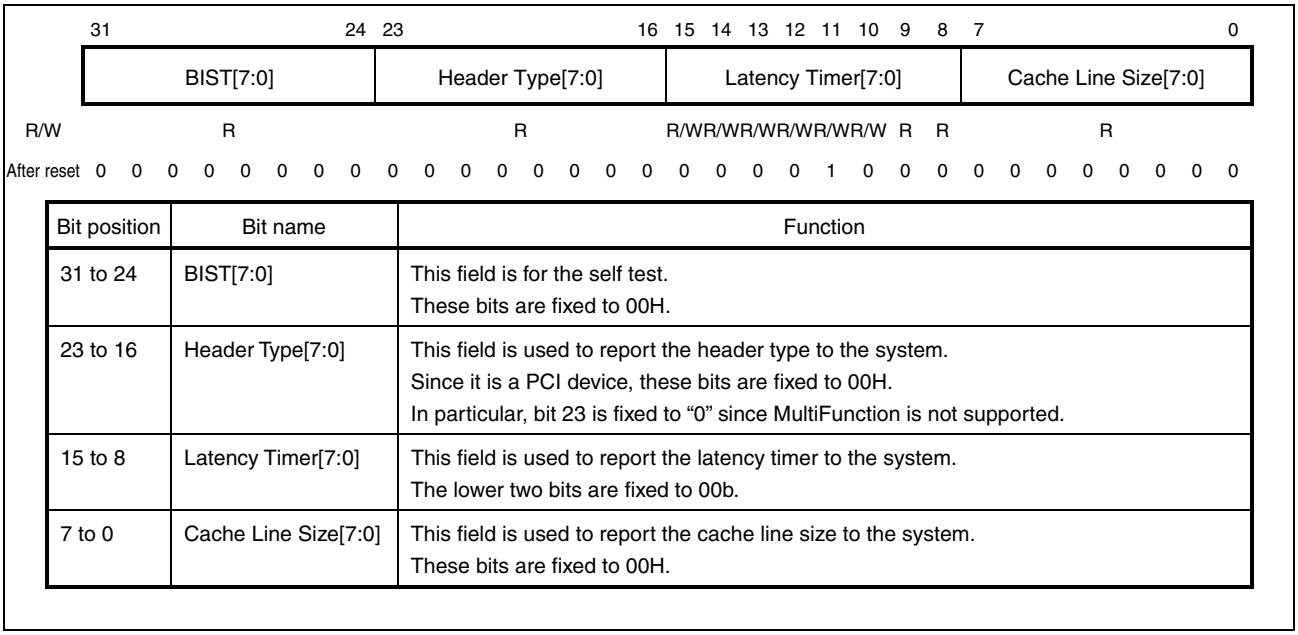
(3) Revision ID, Class Code (Offset 08H)

31 – 24	23 – 16	15 – 8	7 – 0
Class Code			Revision ID[7:0]
Base Class[7:0]	Sub Class[7:0]	Programming I/F[7:0]	
R	R	R	R
0 0 0 0 1 1 0 0	0 0 0 0 0 0 0 1	1 0 0 0 1 0 0 0	0 0 0 1 0 0 0 0

After reset

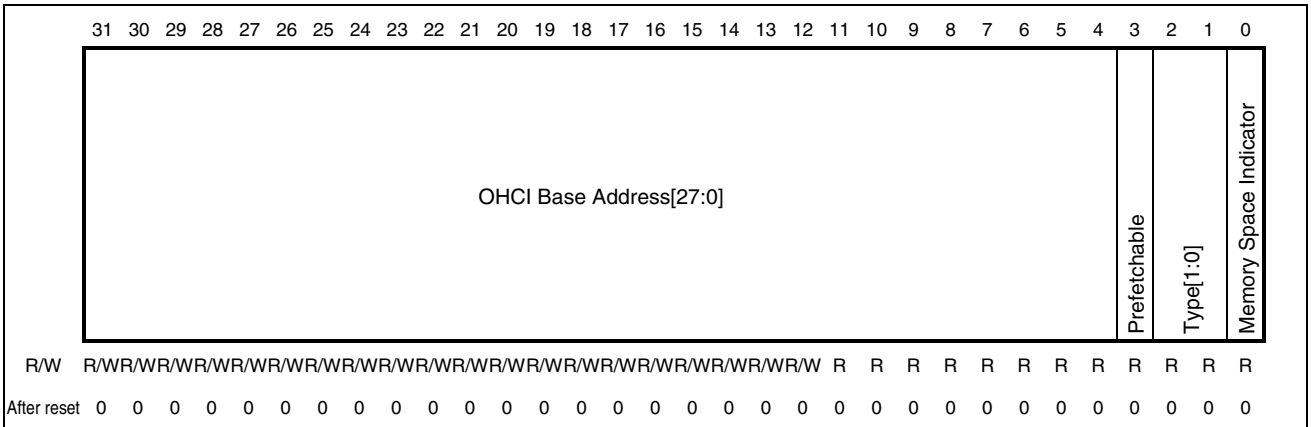
Bit position	Bit name	Function
31 to 24	Base Class[7:0]	This field indicates the base class according to the PCI standard. Since it is a controller for the serial peripheral bus, these bits are fixed to 0CH.
23 to 16	Sub Class[7:0]	This field indicates the sub class according to the PCI standard. Since it is a USB device, these bits are fixed to 03H.
15 to 8	Programming I/F[7:0]	This field indicates the programming interface according to the PCI standard. Since it is an OHCI-compliant USB, these bits are fixed to 10H.
7 to 0	Revision ID[7:0]	This field indicates the revision ID of the host controller. These bits are fixed to 42H.

(4) Cache Line Size • Latency Timer • Header Type • BIST (Offset 0CH)



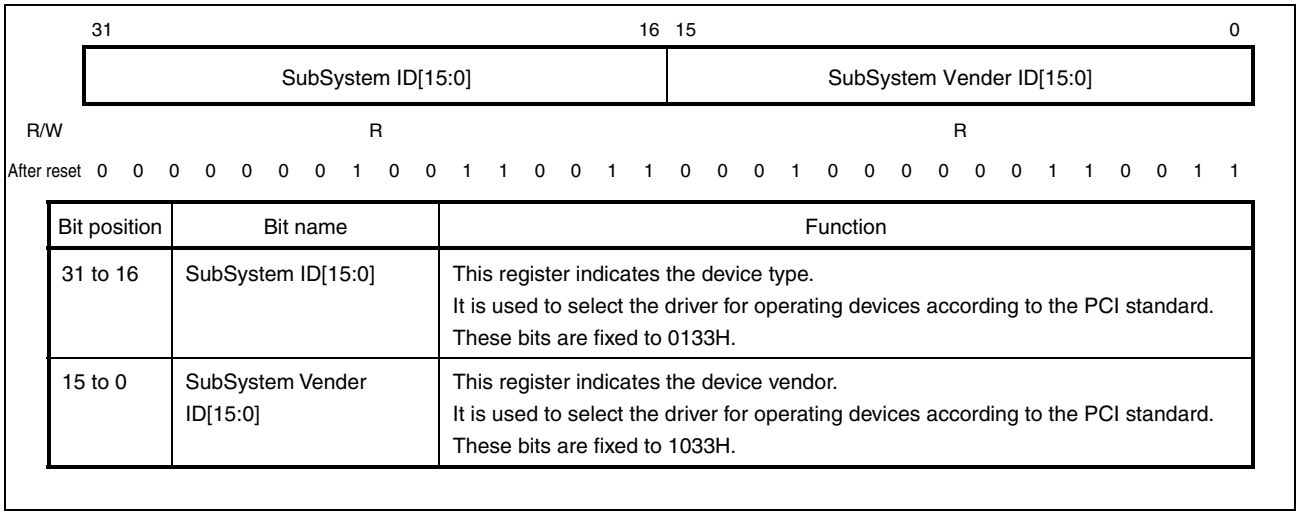
Bit position	Bit name	Function
31 to 24	BIST[7:0]	This field is for the self test. These bits are fixed to 00H.
23 to 16	Header Type[7:0]	This field is used to report the header type to the system. Since it is a PCI device, these bits are fixed to 00H. In particular, bit 23 is fixed to “0” since MultiFunction is not supported.
15 to 8	Latency Timer[7:0]	This field is used to report the latency timer to the system. The lower two bits are fixed to 00b.
7 to 0	Cache Line Size[7:0]	This field is used to report the cache line size to the system. These bits are fixed to 00H.

(5) OHCI Base Address (Offset 10H)



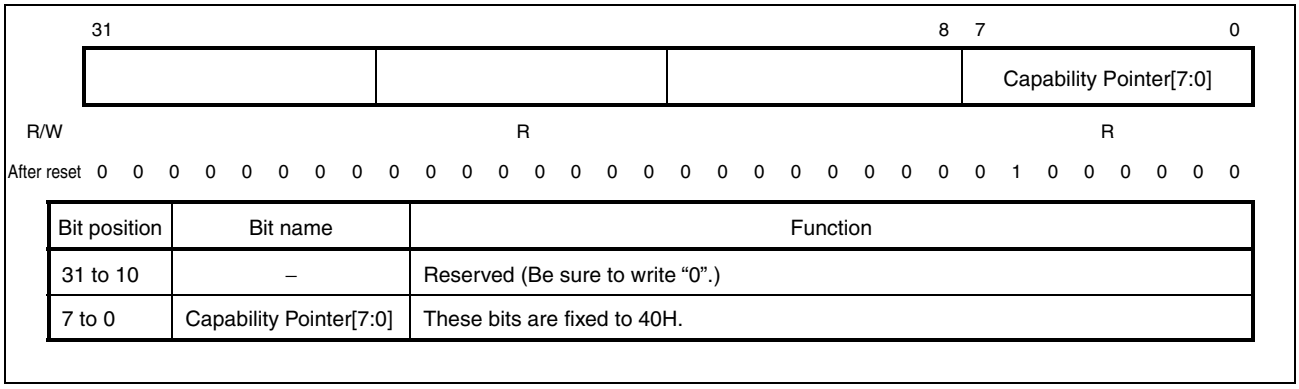
Bit position	Bit name	Function
31 to 4	OHCI Base Address[27:0]	Bits 31 to 12 indicate the addresses of the operational register. Set the base address value of the operational register, which is determined by the system during initialization. Bits 11 to 4 are fixed to 00H.
3	Prefetchable	This field indicates that the field specified by the base address is a memory space. This bit is fixed to 0b. It indicates that prefetching is disabled for this field.
2, 1	Type[1:0]	This field indicates base address type. These bits are fixed to 00b. It indicates that the OHCI base address is a “32-bit space at any position”.
0	Memory Space Indicator	This field indicates that the field specified by the base address is a memory space. This bit is fixed to 0b.

(6) SubSystem Vender ID • SubSystemID (Offset 2CH)



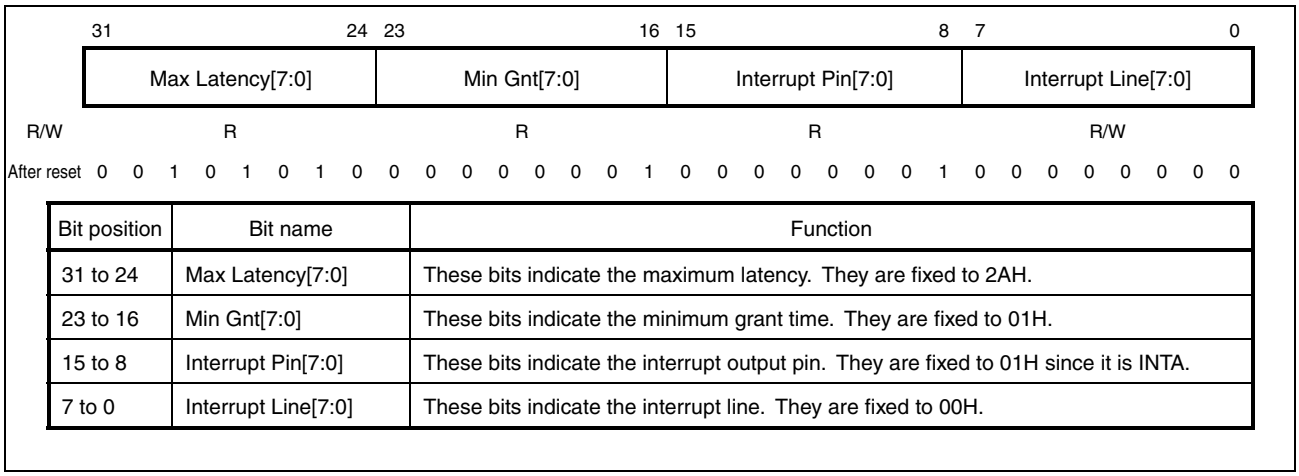
Bit position	Bit name	Function
31 to 16	SubSystem ID[15:0]	This register indicates the device type. It is used to select the driver for operating devices according to the PCI standard. These bits are fixed to 0133H.
15 to 0	SubSystem Vender ID[15:0]	This register indicates the device vendor. It is used to select the driver for operating devices according to the PCI standard. These bits are fixed to 1033H.

(7) Capability Pointer (Offset 34H)



Bit position	Bit name	Function
31 to 10	–	Reserved (Be sure to write “0”.)
7 to 0	Capability Pointer[7:0]	These bits are fixed to 40H.

(8) Interrupt Line • Interrupt Pin • Min gnt • Max Latency (Offset 3CH)



Bit position	Bit name	Function
31 to 24	Max Latency[7:0]	These bits indicate the maximum latency. They are fixed to 2AH.
23 to 16	Min Gnt[7:0]	These bits indicate the minimum grant time. They are fixed to 01H.
15 to 8	Interrupt Pin[7:0]	These bits indicate the interrupt output pin. They are fixed to 01H since it is INTA.
7 to 0	Interrupt Line[7:0]	These bits indicate the interrupt line. They are fixed to 00H.

(9) Capability Identifier • Next Item Pointer • Power Management Capabilities (Offset 40H)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Power Management Capabilities																Next Item Pointer[7:0]								Capability Identifier[7:0]							
PME Support[4:0]					D2 Support	D1 Support	AUX Current[2:0]			DSI		PME CLK	Version[2:0]																			
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
After reset	0	1	1	1	1	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit position	Bit name	Function
31	PME Support[4:0]	This bit indicates whether the D3 Cold state is supported. It is fixed to 0b since the D3 Cold state is not supported.
30 to 27		These bits indicate that PME is supported by all PCI power states (D0 to D3). They are fixed to 1111b.
26	D2 Support	This bit indicates that PCI power state D2 is supported. It is fixed to 1b.
25	D1 Support	This bit indicates that PCI power state D1 is supported. It is fixed to 1b.
24 to 22	Aux Current[2:0]	Assertion of PME interrupts from the D3Cold state is not supported. Consequently, these bits are fixed to 000b.
21	DSI	This bit indicates that special initialization is not required in order to use Power Management. It is fixed to 0b.
19	PME CLK	This bit indicates that PCLK is not required to generate PME interrupts. It is fixed to 0b.
18 to 16	Version[2:0]	This field indicates the Power Management version. These bits are fixed to 010b, according to the circuit configuration implemented in the host controller.
15 to 8	Next Item Pointer[7:0]	This field indicates that the next item does not exist. These bits are fixed to 00H.
7 to 0	Capability Identifier[7:0]	This field indicates the power management register ID. These bits are fixed to 01H.

(10) Power Management Control/Status • PMCSR Bridge Support Extensions (Offset 44H)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Data								PMCSR Bridge Support Extensions								Power Management Control/Status															
Data[7:0]								BPCC Enable	B2_B3							PME Status	Data Scale[1:0]		Data Select[3:0]				PME Enable							Power State[1:0]	
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R/W	R	R	R	R	R	R	R/W	R/W
After reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position	Bit name	Function
31 to 24	Data[7:0]	These bits are fixed to 00H. According to the PCI standard, this field is an optional field which is not supported by the host controller.
23	BPCC Enable	This is a bridge bit, and it is not supported by the host controller. It is fixed to 0b.
22	B2_B3	This is a bridge bit, and it is not supported by the host controller. It is fixed to 0b.
15	PME Status	This bit indicates the PME interrupt status. It is set (1) to 1b when the condition for PME assertion is met. It is cleared (0) to 0b when 1b is written from the PCI bus.
14, 13	Data Scale[1:0]	These bits are fixed to 00b. According to the PCI standard, this field is an optional field which is not supported by the host controller.
12 to 9	Data Select[3:0]	These bits are fixed to 0H. According to the PCI standard, this field is an optional field which is not supported by the host controller.
8	PME Enable	This bit sets how PME interrupts are used. When it is set to 1b, a PME interrupt is generated at a return from Power Management.
1, 0	Power State[1:0]	This field indicates the PCI power status. The following states result, depending on the status of [1:0]. 0 0 D0 State 0 1 D1 State 1 0 D2 State 1 1 D3 hot State

2.3.3 OHCI operational register

The OHCI operational register is incorporated in the OHCI host controller and is configured of the following. For more detailed information, see the OpenHCI Specification, Release 1.0a.

In the OpenHCI Specification, Release 1.0a, port numbers are defined as [1: Port number], and this definition is followed in the descriptions below. For example, it is explained that Port [1] corresponds to host channel 0 and Port [2] corresponds to host channel 1.

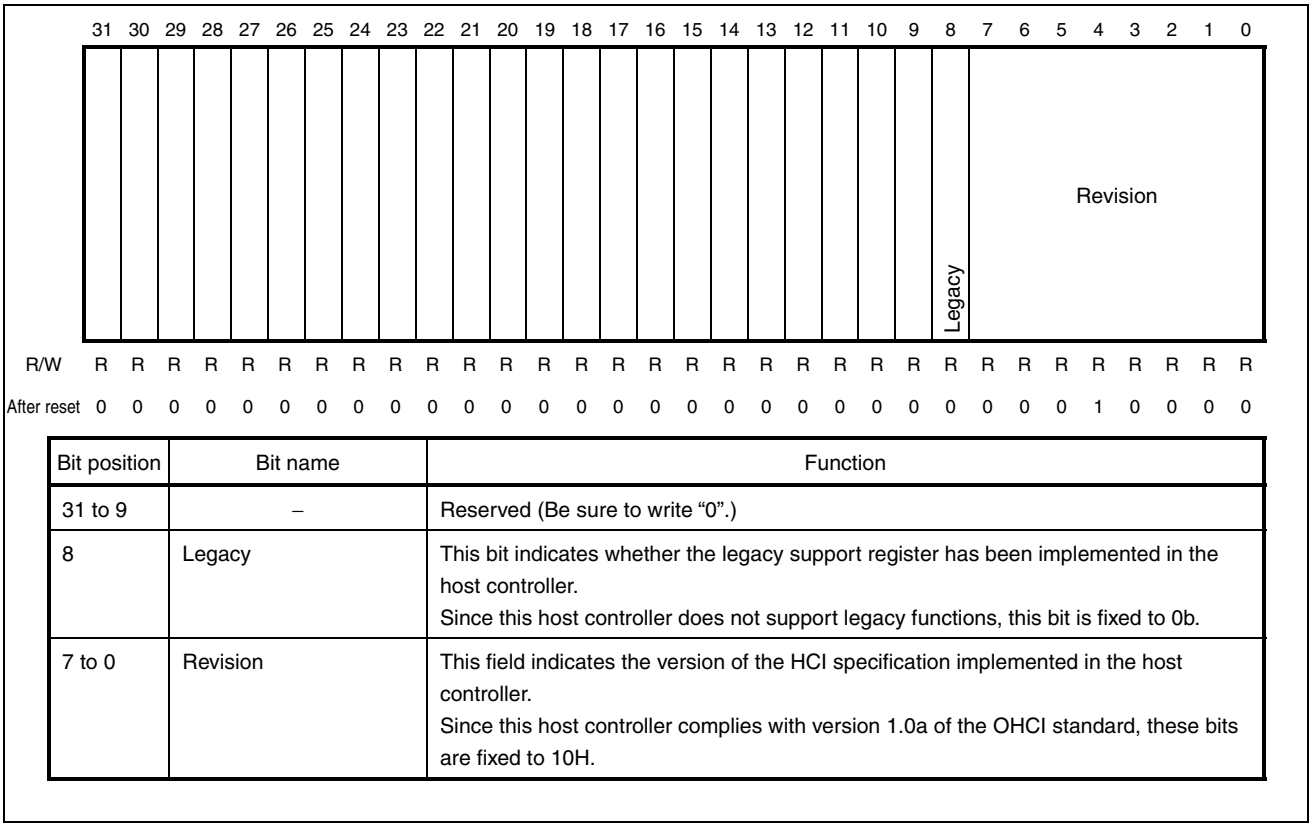
Table 2-5. OHCI Operational Registers

Address	31 24	23 16	15 8	7 0
00H	HcRevision			
04H	HcControl			
08H	HcCommandStatus			
0CH	HcInterruptStatus			
10H	HcInterruptEnable			
14H	HcInterruptDisable			
18H	HcHCCA			
1CH	HcPeriodCurrentED			
20H	HcContorlHeadED			
24H	HcControlCurrentED			
28H	HcBulkHeadED			
2CH	HcBulkCurrentED			
30H	HcDoneHead			
34H	HcFmInterval			
38H	HcFmRemaining			
3CH	HcFmNumber			
40H	HcPeriodicStart			
44H	HcLSThreshold			
48H	HcRhDescriptorA			
4CH	HcRhDescriptorB			
50H	HcRhStatus			
54H	HcRhPortStatus1			
58H	HcRhPortStatus2			
5CH to FFH	Reserved			

Terms and abbreviations

HC: HostController (indicates the OHCI host controller)
HCD: HostControllerDriver
ED: EndPointDescriptor
TD: TransferDescriptor
EOP: EndOfPacket
SOF: StartOfFrame

(1) HcRevision Register (Offset 00H)



Bit position	Bit name	Function
31 to 9	–	Reserved (Be sure to write “0”.)
8	Legacy	This bit indicates whether the legacy support register has been implemented in the host controller. Since this host controller does not support legacy functions, this bit is fixed to 0b.
7 to 0	Revision	This field indicates the version of the HCI specification implemented in the host controller. Since this host controller complies with version 1.0a of the OHCI standard, these bits are fixed to 10H.

(2) HcControl Register (Offset 04H)

(1/2)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																						RWE	RWC	IR	HCFS		BLE	CLE	IE	PLE	CBSR	
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

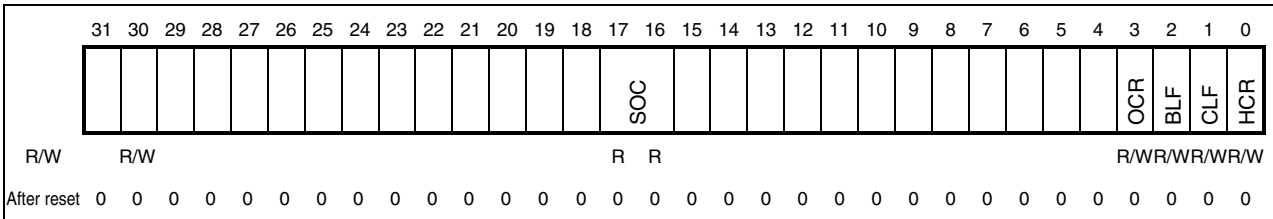
Bit position	Bit name	Function
31 to 11	–	Reserved (Be sure to write “0”.)
10	RemoteWakeUp Enable (RWE)	This bit is used to set detection of upstream resume signals. 1: Sets resume signal as RemoteWake. 0: Does not set resume signal as RemoteWake.
9	RemoteWakeUp Connect (RWC)	This bit indicates whether the host controller supports remote wakeup. When remote wakeup is supported by the system, this bit must be set (1) during initialization. 1: Supports remote wakeup. 0: Does not support remote wakeup.
8	InterruptRouting (IR)	This bit indicates the interrupt output path of the host controller. It sets how to report to the system the interrupt sources that have occurred in HcInterruptStatus. 1: Interrupt occurs via SMI. 0: Interrupt occurs via INTA.
7, 6	HostController FunctionalState (HCFS)[1:0]	This field indicates the host controller operating state. HCFS / USB status: 00 – USB Reset 01 – USB Resume 10 – USB Operational 11 – USB Suspend When transitioning to USB Operational, control of frames separated by 1 ms begins. This operating state is controlled by the host controller driver except when transitioning to USB Resume via a remote wakeup while during USB Suspend status. After a hardware reset, the USB Reset status is shown in this field, but after a software reset it transitions to USB Suspend status.
5	BulkListEnable (BLE)	This bit sets whether to perform bulk list processing. 1: Performs bulk list processing. 0: Does not perform bulk list processing. The value set to this bit becomes valid starting from the next frame. To revise the bulk list, this bit value must be 0b.

(2/2)

Bit position	Bit name	Function
4	ControlListEnable (CLE)	This bit sets whether to perform control list processing. 1: Performs control list processing. 0: Does not perform control list processing. The value set to this bit becomes valid starting from the next frame. To revise the control list, this bit value must be 0b.
3	IsochronouseEnable (IE)	This bit sets whether to perform isochronous ED processing. If an isochronous ED is discovered during list processing, check this bit to determine whether to perform isochronous ED processing. 1: Performs isochronous transfer processing. 0: Does not perform isochronous transfer processing. When this bit is enabled or disabled, isochronous processing is affected starting from the next frame.
2	PeriodicListEnable (PLE)	This bit sets whether to perform periodic list processing. 1: Performs periodic list processing. 0: Does not perform periodic list processing. When this bit is enabled or disabled, periodic list processing is started or stopped from the next frame.
1, 0	ControlBulk ServiceRatio (CBSR)[2:0]	This field is used to specify the service ratio for control transfer and bulk transfer. During processing of the periodic list, the service ratio specified via this field is maintained for the transfers being performed.

CBSR	Service ratio of bulk ED: control ED
00	1:1
01	2:1
10	3:1
11	4:1

(3) HcCommandStatus Register (Offset 08H)



Bit position	Bit name	Function
31 to 18	–	Reserved (Be sure to write “0”.)
17, 16	Scheduling OverrunCount (SOC)[2:0]	This field is used to count the number of schedule overruns. It is incremented each time a schedule overrun occurs. This incrementation continues even when SO of the InterruptStatus register has been set (1).
15 to 4	–	Reserved (Be sure to write “0”.)
3	Ownership ChangeRequest (OCR)	This bit is used to request a change of control right over the host controller.
2	BulkListFilled (BLF)	This bit is used to indicate whether TD exists in the bulk list. When adding TD to ED in the bulk list, this bit is always set to 1b by the driver (HCD). When the host controller starts processing the bulk list head, this bit is checked. When the value of this bit is 0b, bulk list processing will not be started. If it is 1b, bulk list processing starts when 0b is set. If TD is found in the bulk list, 1b is set again and bulk list processing is continued. This bit must be set before the driver reconfigures the list, sets the BLE bit of the HcCommand register, and start list processing.
1	ControlListFilled (CLF)	This bit is used to indicate whether TD exists in the control list. When adding TD to ED in the control list, this bit is always set to 1b by the driver (HCD). When the host controller starts processing the control list head, this bit is checked. When the value of this bit is 0b, control list processing will not be started. If it is 1b, control list processing starts when 0b is set. If TD is found in the control list, 1b is set again and control list processing is continued. This bit must be set before the driver reconfigures the list, sets the CLE bit of the HcCommand register, and start list processing.
0	HostController Reset (HCR)	This bit is used to start a software reset of the host controller. When this bit is set (1), the USB status transitions to USB Suspend, regardless of the functional state of the host controller. It is cleared (0) by the host controller when the reset operation is completed.

(4) HcInterruptStatus Register (Offset 0CH)

(1/2)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	OC																								RHSC	FNO	UE	RD	SF	WDH	SO
R/W	R/W																								R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position	Bit name	Function
31	–	Reserved (Be sure to write “0”.)
30	OwnershipChange (OC)	This interrupt bit indicates that an ownership change interrupt has occurred. It is set (1) when the ownership change request field of HcCommand Status has been set. An SMMI interrupt will be generated if this interrupt source is not masked. 1: OC interrupt has occurred. 0: OC interrupt has not occurred. An interrupt is cleared (0) when 1b is written to this bit.
29 to 7	–	Reserved (Be sure to write “0”.)
6	RootHubStatusChange (RHSC)	This interrupt bit indicates that the HcRhPortStatus status has changed. It is set (1) when the HcRhPortStatus has been changed by a hardware source. 1: RHSC interrupt has occurred. 0: RHSC interrupt has not occurred. An interrupt is cleared (0) when 1b is written to this bit.
5	Frame Number Overflow (FNO)	This interrupt bit indicates that the MSB of the frame number has changed. It is set (1) after the Hcca frame number has been updated in a frame in which the frame number MSB changes from 0 to 1 or from 1 to 0. 1: FNO interrupt has occurred. 0: FNO interrupt has not occurred. An interrupt is cleared (0) when 1b is written to this bit.
4	Unrecoverable Error (UE)	This interrupt bit indicates that a system error that is not related to USB has been detected on the PCI bus. 1: UE interrupt has occurred. 0: UE interrupt has not occurred. An interrupt is cleared (0) when 1b is written to this bit.
3	ResumeDetected (RD)	This interrupt bit indicates that resume has been detected. It is set (1) when it is detected that a device on the USB bus has been asserting a resume signal. It is not set (1) when USB Resume has been issued by the driver. 1: RD interrupt has occurred. 0: RD interrupt has not occurred. An interrupt is cleared (0) when 1b is written to this bit.

(2/2)

Bit position	Bit name	Function
2	StartOfFrame (SF)	This interrupt bit indicates that the Hcca frame number has been updated at the start of a frame. The host controller updates the Hcca frame number while sending SOF packets. 1: SF interrupt has occurred. 0: SF interrupt has not occurred. An interrupt is cleared (0) when 1b is written to this bit.
1	Writeback Done Head (WDH)	This interrupt bit indicates that the host controller has updated the contents of HccaDoneHead. The host controller sets (1) this bit immediately after updating HccaDoneHead, and does not update HccaDoneHead again until this bit is cleared (0). 1: WDH interrupt has occurred. 0: WDH interrupt has not occurred. An interrupt is cleared (0) when 1b is written to this bit.
0	SchedulingOverrun (SO)	This interrupt bit indicates that a USB schedule overrun has occurred in a frame. When a USB schedule overrun has occurred, this bit is set (1) after the frame number of the next frame is updated. When it is set (1), the SchedulingOverrun bit of the HcCommandStatus register is also incremented. 1: SO interrupt has occurred. 0: SO interrupt has not occurred. An interrupt is cleared (0) when 1b is written to this bit.

(5) HcInterruptEnable Register (Offset 10H)

(1/2)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	MIE	OCE																								RHSCE	FNOE	UEE	RDE	SFE	WDHE	SOE
R/W	R/W	R/W	R/WR/WR/WR/WR/WR/WR/W																													
After reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position	Bit name	Function
31	MasterInterruptEnable (MIE)	This bit is used to enable or disable the interrupt source setting set by [30:0]. 1: Enables all set interrupts. 0: Disables (ignores writing of 0b). To clear (0) this bit, write 1b to the corresponding bit of the HcInterruptDisable register.
30	OwnershipChange Enable (OCE)	This bit is used to enable or disable OC as an interrupt source. 1: Enables OC as an interrupt source. 0: Disables (ignores writing of 0b). This bit can be set (1) by writing 1b to it. To clear (0) this bit, write 1b to the corresponding bit of the HcInterruptDisable register.
29 to 7	–	Reserved (Be sure to write “0”.)
6	RootHubStatusChange Enable (RHSCE)	This bit is used to enable or disable RHSC as an interrupt source. 1: Enables RHSC as an interrupt source. 0: Disables (ignores writing of 0b). This bit can be set (1) by writing 1b to it. To clear (0) this bit, write 1b to the corresponding bit of the HcInterruptDisable register.
5	FrameNumberOverflow Enable (FNOE)	This bit is used to enable or disable FNO as an interrupt source. 1: Enables FNO as an interrupt source. 0: Disables (ignores writing of 0b). This bit can be set (1) by writing 1b to it. To clear (0) this bit, write 1b to the corresponding bit of the HcInterruptDisable register.
4	UnrecoverableError Enable (UEE)	This bit is used to enable or disable UE as an interrupt source. 1: Enables UE as an interrupt source. 0: Disables (ignores writing of 0b). This bit can be set (1) by writing 1b to it. To clear (0) this bit, write 1b to the corresponding bit of the HcInterruptDisable register.
3	ResumeDetected Enable (RDE)	This bit is used to enable or disable RD as an interrupt source. 1: Enables RD as an interrupt source. 0: Disables (ignores writing of 0b). This bit can be set (1) by writing 1b to it. To clear (0) this bit, write 1b to the corresponding bit of the HcInterruptDisable register.
2	StatoOfFrame Enable (SFE)	This bit is used to enable or disable SF as an interrupt source. 1: Enables SF as an interrupt source. 0: Disables (ignores writing of 0b). This bit can be set (1) by writing 1b to it. To clear (0) this bit, write 1b to the corresponding bit of the HcInterruptDisable register.

(2/2)

Bit position	Bit name	Function
1	WritebackDoneHead Enable (WDHE)	This bit is used to enable or disable WDH as an interrupt source. 1: Enables WDH as an interrupt source. 0: Disables (ignores writing of 0b). This bit can be set (1) by writing 1b to it. To clear (0) this bit, write 1b to the corresponding bit of the HcInterruptDisable register.
0	SchedulingOverrun Enable (SOE)	This bit is used to enable or disable SO as an interrupt source. 1: Enables SO as an interrupt source. 0: Disables (ignores writing of 0b). This bit can be set (1) by writing 1b to it. To clear (0) this bit, write 1b to the corresponding bit of the HcInterruptDisable register.

(6) HcInterruptDisable Register (Offset 14H)

(1/2)

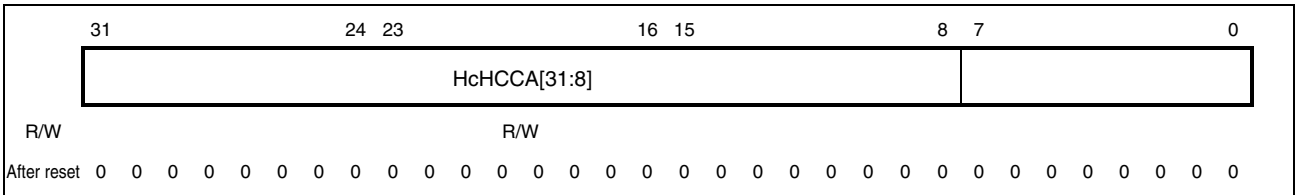
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MID	OCD																								RHSCD	FNOD	UED	RDD	SFD	WDHD	SOD
R/W	R/WR/W																								R/WR/WR/WR/WR/WR/WR/W						
After reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position	Bit name	Function
31	MasterInterruptDisable (MID)	This bit is used to disable interrupt sources set by HcInterruptEnable [30:0]. 1: Disables all set interrupts. 0: (Ignores writing of 0b.) When this HcInterruptDisable register is read, the contents of the HcInterruptEnable register are output. To set (1) this bit, write 1b to the corresponding bit of the HcInterruptEnable register.
30	OwnershipChangeDisable (OCD)	This bit is used to delete OC as an interrupt source. 1: Disables OC as an interrupt source. 0: (Ignores writing of 0b.) When this HcInterruptDisable register is read, the contents of the HcInterruptEnable register are output. This bit can be cleared (0) by writing 1b to it. To set (1) this bit, write 1b to the corresponding bit of the HcInterruptEnable register.
29 to 7	–	Reserved (Be sure to write “0”.)
6	RootHubStatusChange Disable (RHSCD)	This bit is used to delete RHSC as an interrupt source. 1: Disables RHSC as an interrupt source. 0: (Ignores writing of 0b.) When this HcInterruptDisable register is read, the contents of the HcInterruptEnable register are output. This bit can be cleared (0) by writing 1b to it. To set (1) this bit, write 1b to the corresponding bit of the HcInterruptEnable register.
5	FrameNumberOverflow Disable (FNOD)	This bit is used to delete FNO as an interrupt source. 1: Disables FNO as an interrupt source. 0: (Ignores writing of 0b.) When this HcInterruptDisable register is read, the contents of the HcInterruptEnable register are output. This bit can be cleared (0) by writing 1b to it. To set (1) this bit, write 1b to the corresponding bit of the HcInterruptEnable register.
4	UnrecoverableError Disable (UED)	This bit is used to delete UE as an interrupt source. 1: Disables UE as an interrupt source. 0: (Ignores writing of 0b.) When this HcInterruptDisable register is read, the contents of the HcInterruptEnable register are output. This bit can be cleared (0) by writing 1b to it. To set (1) this bit, write 1b to the corresponding bit of the HcInterruptEnable register.

(2/2)

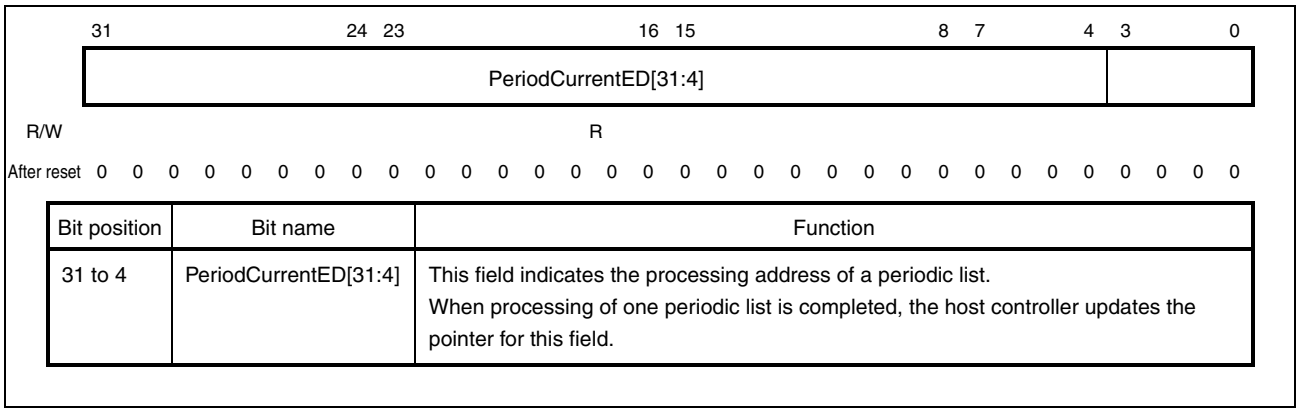
Bit position	Bit name	Function
3	ResumeDetected Disable (RDD)	This bit is used to delete RD as an interrupt source. 1: Disables RD as an interrupt source. 0: (Ignores writing of 0b.) When this HcInterruptDisable register is read, the contents of the HcInterruptEnable register are output. This bit can be cleared (0) by writing 1b to it. To set (1) this bit, write 1b to the corresponding bit of the HcInterruptEnable register.
2	StartOfFrame Disable (SFD)	This bit is used to delete SF as an interrupt source. 1: Disables SF as an interrupt source. 0: (Ignores writing of 0b.) When this HcInterruptDisable register is read, the contents of the HcInterruptEnable register are output. This bit can be cleared (0) by writing 1b to it. To set (1) this bit, write 1b to the corresponding bit of the HcInterruptEnable register.
1	WritebackDoneHead Disable (WDHD)	This bit is used to delete WDH as an interrupt source. 1: Disables WDH as an interrupt source. 0: (Ignores writing of 0b.) When this HcInterruptDisable register is read, the contents of the HcInterruptEnable register are output. This bit can be cleared (0) by writing 1b to it. To set (1) this bit, write 1b to the corresponding bit of the HcInterruptEnable register.
0	Scheduling Overrun Disable (SOD)	This bit is used to delete SO as an interrupt source. 1: Disables SO as an interrupt source. 0: (Ignores writing of 0b.) When this HcInterruptDisable register is read, the contents of the HcInterruptEnable register are output. This bit can be cleared (0) by writing 1b to it. To set (1) this bit, write 1b to the corresponding bit of the HcInterruptEnable register.

(7) HcHCCA Register (Offset 18H)



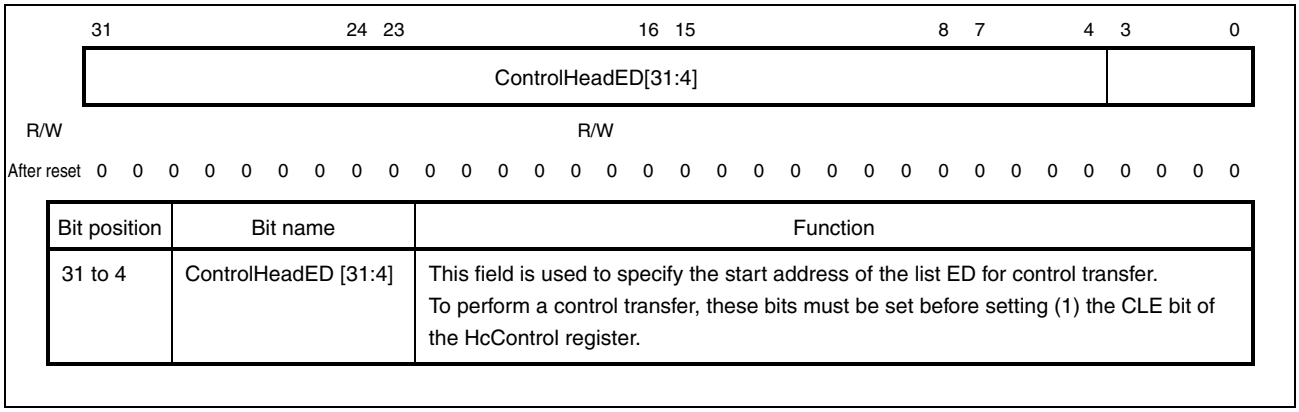
Bit position	Bit name	Function
31 to 8	HcHCCA[31:8]	This field is used to set the base address of the RAM that has been assigned as the host controller communication area. These bits must be set during initialization. The host controller requests, as the HCCA, an area of 256 bytes from the base address specified by this field.

(8) HcPeriodCurrentED Register (Offset 1CH)



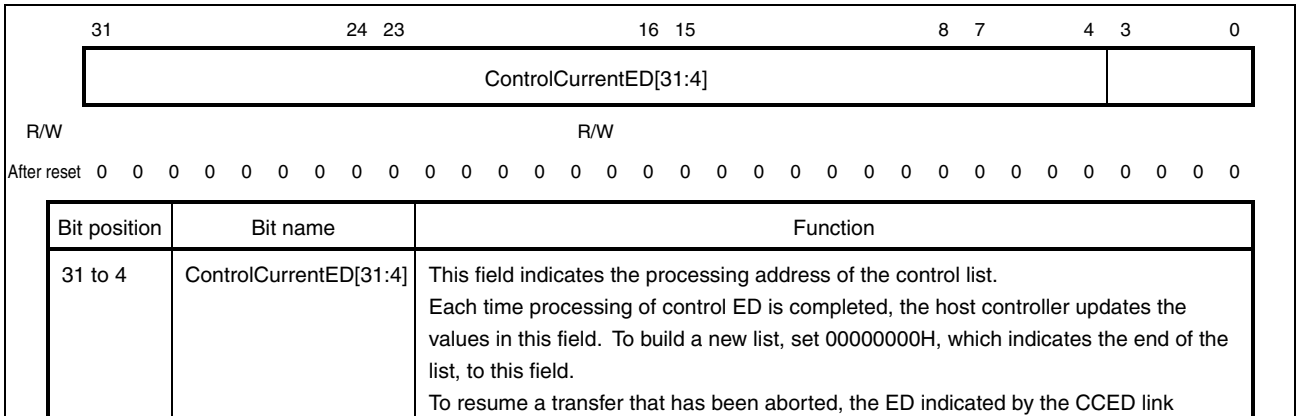
Bit position	Bit name	Function
31 to 4	PeriodCurrentED[31:4]	This field indicates the processing address of a periodic list. When processing of one periodic list is completed, the host controller updates the pointer for this field.

(9) HcControlHeadED Register (Offset 20H)



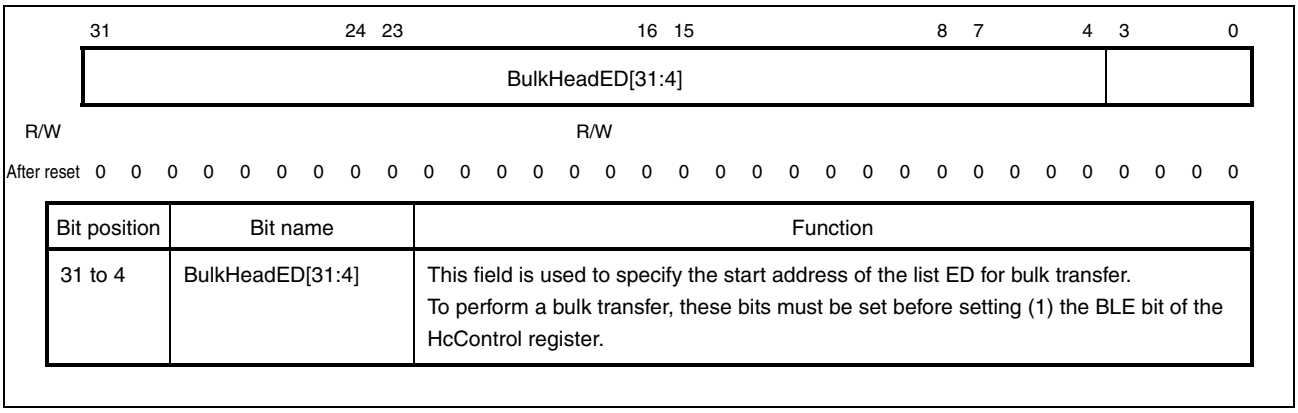
Bit position	Bit name	Function
31 to 4	ControlHeadED [31:4]	This field is used to specify the start address of the list ED for control transfer. To perform a control transfer, these bits must be set before setting (1) the CLE bit of the HcControl register.

(10) HcControlCurrentED Register (Offset 24H)



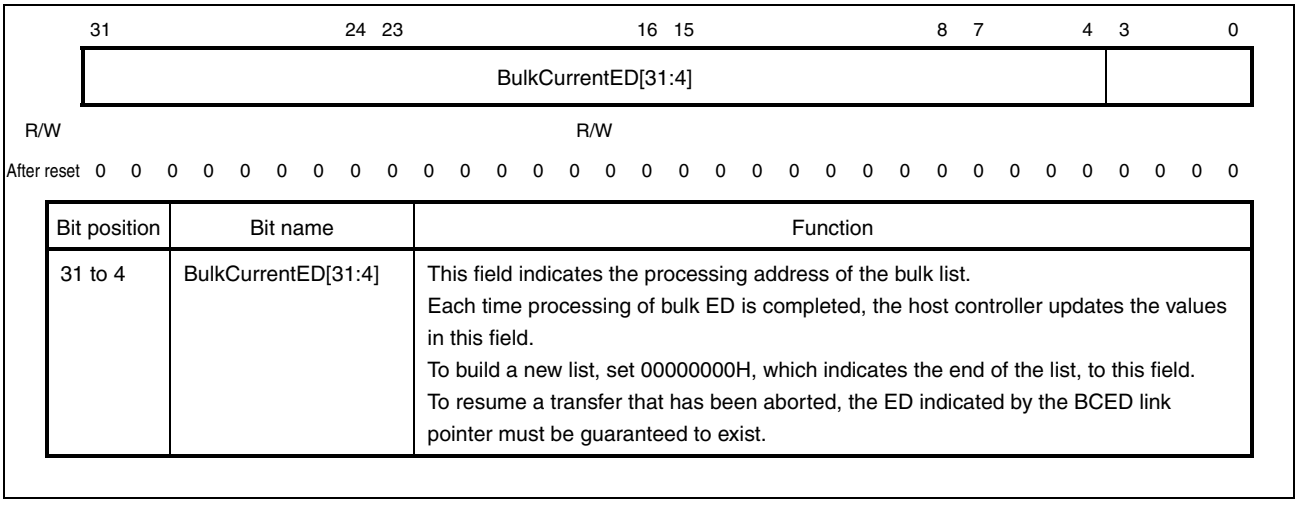
Bit position	Bit name	Function
31 to 4	ControlCurrentED[31:4]	This field indicates the processing address of the control list. Each time processing of control ED is completed, the host controller updates the values in this field. To build a new list, set 00000000H, which indicates the end of the list, to this field. To resume a transfer that has been aborted, the ED indicated by the CCED link pointer must be guaranteed to exist.

(11) HcBulkHeadED Register (Offset 28H)



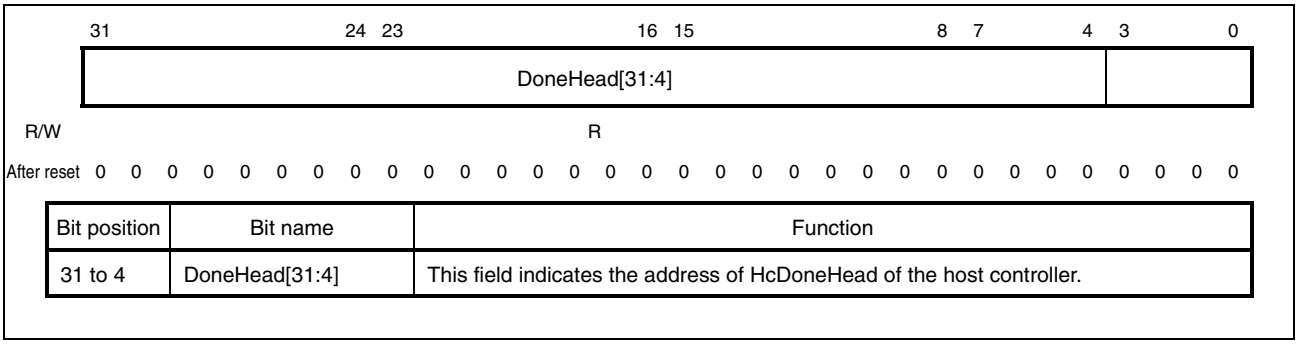
Bit position	Bit name	Function
31 to 4	BulkHeadED[31:4]	This field is used to specify the start address of the list ED for bulk transfer. To perform a bulk transfer, these bits must be set before setting (1) the BLE bit of the HcControl register.

(12) HcBulkCurrentED Register (Offset 2CH)



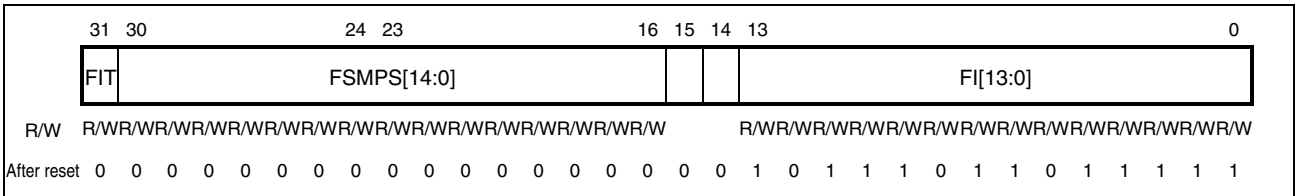
Bit position	Bit name	Function
31 to 4	BulkCurrentED[31:4]	This field indicates the processing address of the bulk list. Each time processing of bulk ED is completed, the host controller updates the values in this field. To build a new list, set 00000000H, which indicates the end of the list, to this field. To resume a transfer that has been aborted, the ED indicated by the BCED link pointer must be guaranteed to exist.

(13) HcDoneHead Register (Offset 30H)



Bit position	Bit name	Function
31 to 4	DoneHead[31:4]	This field indicates the address of HcDoneHead of the host controller.

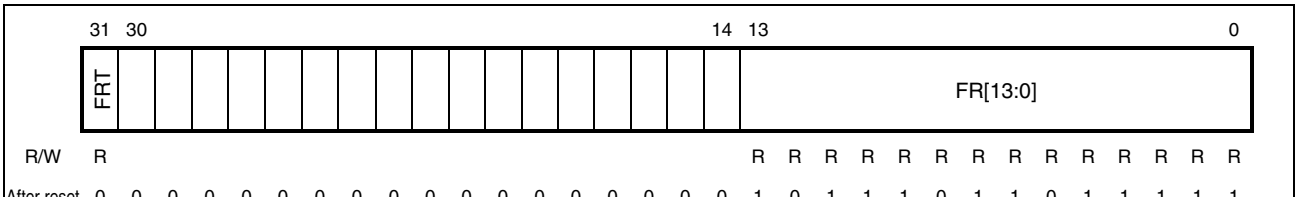
(14) HcFmInterval Register (Offset 34H)



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

Bit position	Bit name	Function
31	FrameIntervalToggle (FIT)	This bit is used to synchronize the frame setting values between HCD and HC. When the FI field is written via HCD, toggle this bit. When HC loads the FI field, the FIT value is reflected to the FRT bit of the HcFmRemaining register. HCD is able to confirm whether the newly set FI field has been reflected by comparing the FIT value that was set when the FI field was written and the FRT value that was read.
30 to 16	FSLagestDataPacket (FSMPS)[14:0]	This field is used to set the maximum amount of data that can be transmitted or received without causing a schedule overrun. The current frame position and the set value are compared to determine the extent of the frame for which a transfer can be started. Since this differs according to the system bus capacity, this value is set from the driver (HCD).
15, 14	–	Reserved (Be sure to write “0”.)
13 to 0	FrameInterval(FI) [13:0]	When at full speed, this field is used to set the interval between two consecutive StartOfFrame as a bit time value. To meet the USB standard of one frame (= 1 ms), set 2EDFH to this field value.

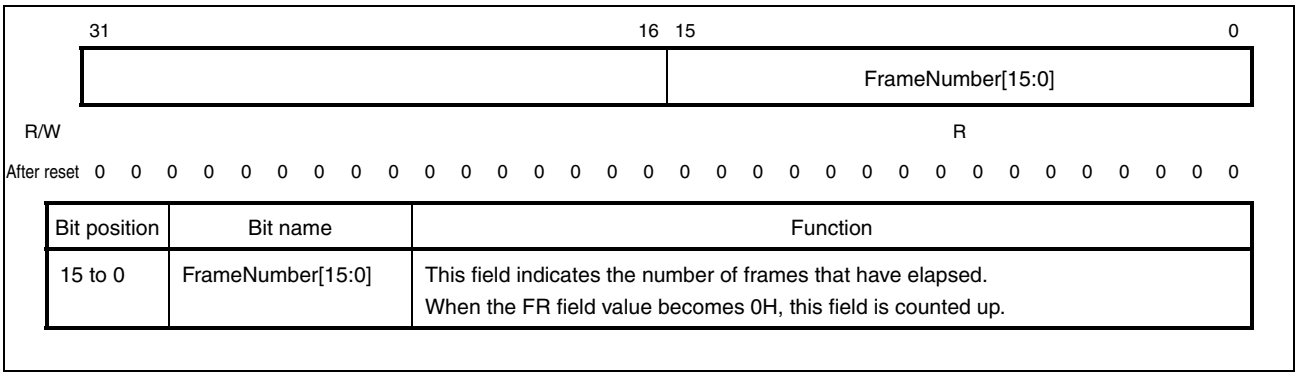
(15) HcFmRemaining Register (Offset 38H)



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

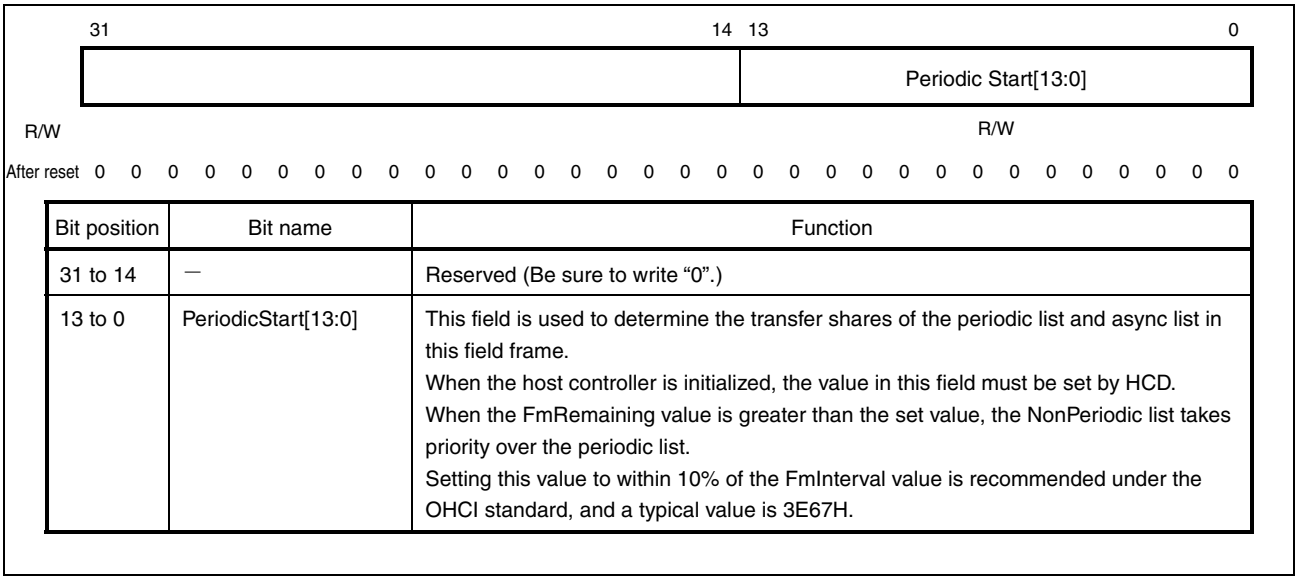
Bit position	Bit name	Function
31	FrameRemainingToggle (FRT)	This bit is used to synchronize the frame setting values between HCD and HC. When the FR field value becomes 0H and the FI field value is reloaded, HC copies the FIT bit value to this bit. HCD can confirm whether the value set to the FI field was also set to FR by comparing the FIT and FRT values.
13 to 0	FrameRemaining(FR) [13:0]	This field indicates the current frame value. The value in this field is counted down as time elapses. When the value becomes 0H, the frame value is reloaded, so the FI value is copied to this field and the field value is counted down again.

(16) HcFmNumber Register (Offset 3CH)



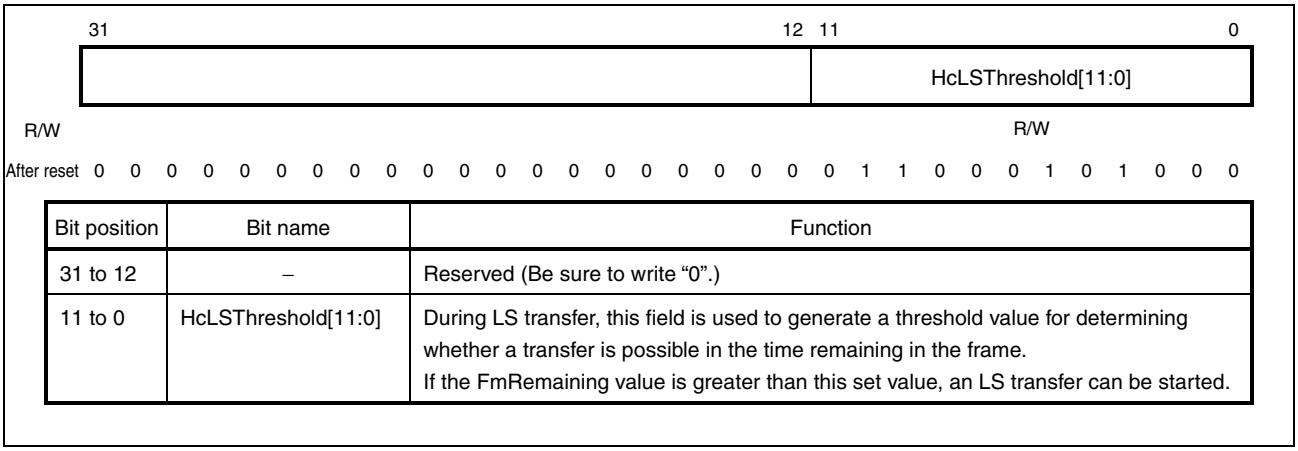
Bit position	Bit name	Function
15 to 0	FrameNumber[15:0]	This field indicates the number of frames that have elapsed. When the FR field value becomes 0H, this field is counted up.

(17) HcPeriodicStart Register (Offset 40H)



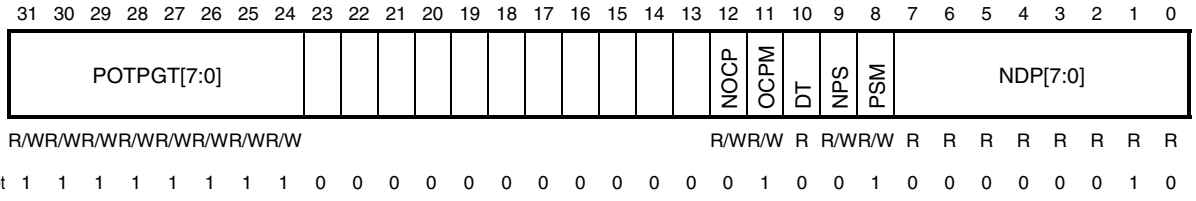
Bit position	Bit name	Function
31 to 14	—	Reserved (Be sure to write “0”.)
13 to 0	PeriodicStart[13:0]	This field is used to determine the transfer shares of the periodic list and async list in this field frame. When the host controller is initialized, the value in this field must be set by HCD. When the FmRemaining value is greater than the set value, the NonPeriodic list takes priority over the periodic list. Setting this value to within 10% of the FmInterval value is recommended under the OHCI standard, and a typical value is 3E67H.

(18) HcLSThreshold Register (Offset 44H)



Bit position	Bit name	Function
31 to 12	–	Reserved (Be sure to write “0”.)
11 to 0	HcLSThreshold[11:0]	During LS transfer, this field is used to generate a threshold value for determining whether a transfer is possible in the time remaining in the frame. If the FmRemaining value is greater than this set value, an LS transfer can be started.

(19) HcRhDescriptorA Register (Offset 48H)



Bit position	Bit name	Function
31 to 24	PowerOnToPowerGoodTime (POTPGT)[7:0]	These bits are used to specify the wait time prior to access by the host controller driver to a powered root hub. The wait time is POTPGT × 2 ms.
23 to 13	–	Reserved (Be sure to write “0”.)
12	NoOverCurrentProtection (NOCP)	This bit is used to specify whether the overcurrent protection function of the root hub is to be supported. 1: Does not support overcurrent protection function. 0: Supports overcurrent protection function.
11	OverCurrentProtectionMode (OCPM)	This bit is used to specify how to report the overcurrent status in the root hub. It must reflect the same mode as PowerSwitchingMode. 1: Reports overcurrent status to individual ports. 0: Reports overcurrent status to all ports simultaneously. This bit is valid only when NoOverCurrentProtection has been cleared (0).
10	DeviceType (DT)	This bit indicates that the root hub is not a composite device. Since the root hub is not recognized as a composite device, 0b is always read from this field.
9	NoPowerSwitching (NPS)	This bit is used to specify whether power switching is supported or whether power to ports is always on. 1: Power is always on when host controller is operating. 0: Port is power switched.
8	PowerSwitchingMode (PSM)	This bit is used to specify how to control port power switching for the root hub. 1: Power supply is controlled individually for ports. 0: Power supply is controlled simultaneously for all ports. Ports respond only to Set/ClearPortPower when the PortPowerControlMask bit has been set (1). When it is cleared (0), ports are controlled by Set/ClearGlobalPower. This bit is valid only when NoPowerSwitching has been cleared (0).
7 to 0	NumberDownstreamPort (NDP)[7:0]	This field is used to specify the number of downstream ports supported by the root hub of the host controller. These bits are fixed to 02H, because two downstream ports have been assigned to this host controller.

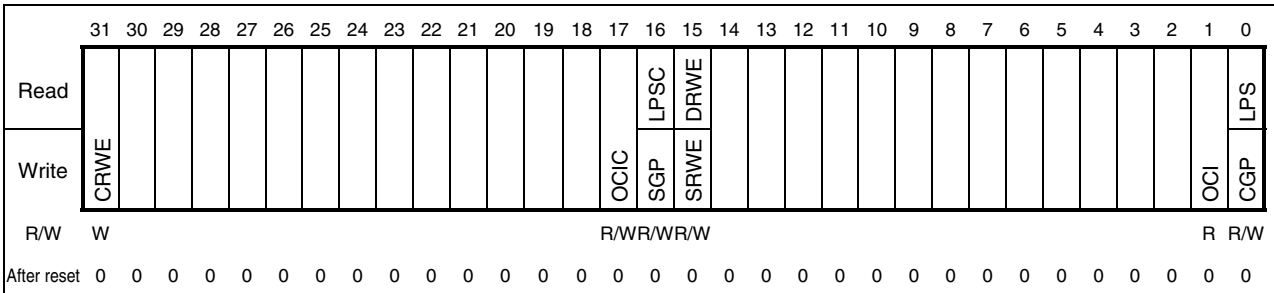
(20) HcRhDescriptorB Register (Offset 4CH)

31	16	15	0
PPCM[15:0]		DR[15:0]	

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

Bit position	Bit name	Function
31 to 16	PortPowerControlMask (PPCM)[15:0]	These bits indicate whether a port is controlled by Set/ClearGlobalPower. They are valid when PowerSwitchingMode has been set (1). • Field Bit: 0 — Description: Reserved Bit: 1 — Description: Sets device connected to port 1. Bit: 2 — Description: Sets device connected to port 2. Bit: 15 to 3 — Description: Reserved • Value 1: Port is operated only by Set/ClearPortPower. 0: Port is controlled by Set/ClearGlobalPower.
15 to 0	DeviceRemovable (DR)[15:0]	These bits indicate whether the host controller port is removable. • Field Bit: 0 — Description: Reserved Bit: 1 — Description: Sets device connected to port 1. Bit: 2 — Description: Sets device connected to port 2. Bit: 15 to 3 — Description: Reserved • Value 1: Connected device is not removable. 0: Connected device is removable.

(21) HcRhStatus Register (Offset 50H)



Bit position	R/W	Bit name	Function
31	–	ClearRemote WakeupEnable (CRWE)	This bit is used to clear (0) the device remote wakeup enable setting. When this bit is set (1), the device remote wakeup enable setting can be cleared (0). Writing 0b has no effect. 0b is always output when this field is read.
30 to 18	–	–	Reserved (Be sure to write “0”.)
17	–	OverCurrent IndicateChange (OCIC)	This bit is used to report that the OCI field has changed for bit 1. This bit is set (1) when the OCI field has changed. This bit can be cleared (0) by writing 1b to this bit when it has been set (1). 1: OverCurrent status has changed. 0: OverCurrent status has not changed.
16	R	Local Power StatusChange(LPSC)	Since LocalPowerStatus is not supported, 0b is always read from this bit.
	W	SetGlobalPower (SGP)	This bit is used to set the power on to all ports during global power mode. When this bit is set (1), power is set to on for all ports. In power mode for each port, the power is set to on only at ports for which the PortPowerControlMask bit has been cleared (0).
15	R	DeviceRemote WakeupEnable (DRWE)	This bit is used to indicate whether ConnectStatusChange is included as a RemoteWakeUp event. 1: Connect Status Change is Remote Wakeup source. 0: Connect Status Change is not Remote Wakeup source. When this bit is set (1) and a ConnectStatusChange event has occurred, the USB status transitions from USB Suspend to USB Resume and a ResumeDetect interrupt occurs.
	W	SetRemote WakeupEnable (SRWE)	This bit is used to set (1) the DRWE bit. When this bit is set (1), the DeviceRemoteWakeupEnable bit can be set (1). Writing 0b has no effect.
14 to 2	–	–	Reserved (Be sure to write “0”.)
1	–	OverCurrent Indicator (OCI)	This bit reports overcurrent status during global overcurrent detection mode. 1: Overcurrent status at port 0: Normal status at port When overcurrents are reported for individual ports, this bit is fixed to 0b.
0	R	LocalPowerStatus (LPS)	Since LocalPowerStatus is not supported, 0b is always read from this bit.
	W	ClearGlobalPower (CGP)	This bit is used to set the power off to all ports during global power mode. When this bit is set (1), power is set to off for all ports. In power mode for each port, the power is set to off only for ports for which the PortPowerControlMask bit has been cleared (0).

(22) HcRhPortStatus1/2 Register (Offset 54H / 58H)

(1/3)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read																							LSDA	PPS				PRS	POCI	PSS	PES	CSC
Write												PRSC	OCIC	PSSC	PESC	CSC							CPP	SPP				SPR	CSS	SPS	SPE	CPE
R/W	R											R/W	R/W	R/W	R/W	R/W							R/W	R/W				R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position	R/W	Bit name	Function
31 to 21	–	–	Reserved (Be sure to write “0”.)
20	–	Port Reset Status Change (PRSC)	This bit indicates that a port reset has been completed. 1: Port reset has been completed. 0: Port Reset Status has not changed. This bit is set (1) when a 10 ms hardware reset has been completed. This bit is cleared (0) when 1b is set to it by the driver.
19	–	Over Current Indicate Change (OCIC)	This bit is set (1) when overcurrent status has been detected at a port. 1: OverCurrent status has changed. 0: OverCurrent status has not changed. This bit is cleared (0) when 1b is set to it by the driver (HCD).
18	–	Port Suspend Status Change (PSSC)	This bit indicates that a RESUME sequence has been completed. 1: RESUME has been completed. 0: Port Suspend Status has not changed. This bit is set (1) when all RESUME processing by hardware has been completed. This bit is cleared (0) when 1b is set to it by the driver (HCD).
17	–	Port Enable Status Change (PESC)	This bit indicates that the PES bit has been cleared (0). 1: PES has been changed (PES cleared). 0: PES has not changed. This bit is set (1) in conjunction with when the status of a port has been transitioned from enabled to disabled due to a hardware event such as an overcurrent, disconnection, power off, a bubble, or an error, PES is cleared (0). This bit is cleared (0) when 1b is set to it by the driver (HCD).
16	–	Connect Status Change (CSC)	This bit indicates that CCS has changed. 1: Current Connect Status has changed. 0: Current Connect Status has not changed. This bit is set (1) when CCS changes (by connection or disconnection). If a port reset request, a port suspend request, or a port enable request is issued during disconnection, this bit is set (1) such that the driver re-evaluates the device connection check. This bit is cleared (0) when 1b is set to it by the driver (HCD).
15 to 10	–	–	Reserved (Be sure to write “0”.)

(2/3)

Bit position	R/W	Bit name	Function
9	R	Low Speed Device Attached (LSDA)	This bit indicates the speed of the device connected to a port. 1: Low-speed device is connected. 0: Full-speed device is connected. This status bit is valid only when CCS has been set.
	W	Clear Port Power (CPP)	This bit is used to turn off port power. Port power is turned off by writing 1b to this bit. Writing 0b has no effect.
8	R	Port Power Status (PPS)	This bit reflects the power supply status of a port. 1: Port power on 0: Port power off How to control port power differs according to the power switch time value.
	W	Set Port Power (SPP)	This bit is used to turn port power on during power control for each port. Port power is turned on by writing 1b to this bit. Writing 0b has no effect.
7 to 5	–	–	Reserved (Be sure to write “0”.)
4	R	Port Reset Status (PRS)	This bit indicates that a reset is being issued for a downstream port. 1: Port reset in progress 0: Port reset not in progress When a 10 ms port reset operation is completed, this bit is cleared (0) while PRSC is being set. This bit cannot be set when CSC has been cleared (0) and a device is not connected.
	W	Set Port Reset (SPR)	This bit is used to issue a port reset for a downstream port. When 1b is written to this bit, a 10 ms port reset operation is started. If a value is written to this bit after CCS has been cleared (0), CSC is set and that an attempt was made to reset a disconnected port is reported to the driver. Writing 0b has no effect.
3	R	Port Over Current Indicator (POCI)	This bit indicates that overcurrent status has occurred at a downstream port. 1: Overcurrent status at port 0: Normal status at port
	W	Clear Port Suspend (CPS)	This bit is used to end suspend status and start a resume sequence. When 1b is written to this bit, a resume sequence is started. Writing 0b has no effect. Resume is started only when PSS has been set.
2	R	Port Suspend Status (PSS)	This bit indicates that the port status is suspend, or during a resume sequence. 1: Port status is suspended. 0: Port status is normal transfer. The port status cannot be set if the CCS bit has been cleared (0) and the device is not connected. The port is set by writing to the SPS bit. This bit is cleared (0) when the resume operation ends, the port reset operation ends, or the status transitions to USB RESUME.
	W	Set Port Suspend (SPS)	This bit is used to transition the port status to suspend. When 1b is written to this bit, the port status transitions to suspend. Writing 0b has no effect. When 1b is written to this bit, the port status transitions to suspend. If a value is written to this bit after CCS has been cleared (0), CSC is set and that an attempt was made to suspend a disconnected port is reported to the driver.

(3/3)

Bit position	R/W	Bit name	Function
1	R	Port Enable Status (PES)	This bit indicates whether the port status is enable or disable. 1: Port status is enable. 0: Port status is disable. This bit cannot be set for an unconnected device for which the CCS bit has been cleared (0). The port status transitions to enable when the port reset operation ends. This bit is cleared (0) automatically by hardware when an overcurrent status, disconnection, power off, a bubble, or an error is detected.
	W	Set Port Enable (SPE)	This bit is used to set (1) the PES bit. Writing 0b has no effect. Use PortReset to transition the port status. Use of the SePortEnable bit to transition the port status to enable is supported under the OHCI standard but not under the USB standard, so it is not supported in this host controller.
0	R	Current Connect Status (CCS)	This bit reflects the current connection status of a downstream port. 1: Device is connected. 0: Device is not connected.
	W	Clear Port Enable (CPE)	This bit is used to clear (0) the PES bit. Writing 1b to this bit sets the port to disabled status. Writing 0b has no effect.

2.3.4 Interrupts from USB host controller

In the USB host controller, interrupts from the OHCI host controller are merged and are reported to the system as two interrupts.

Table 2-6. Interrupts from USB Host Controller

Interrupt Report Signal to System	Interrupt Report Signal Generated by OHCI Host Controller
INTUSBH0	INTA, SMMI, or PME
INTUSBH1	PME

The details of each interrupt are as follows.

(1) INTA interrupt, SMMI interrupt

(a) Interrupt reporting path

The OHCI host controller reports either an INTA interrupt or an SMMI interrupt to the system, according to the value set to the IR bit of the HcControl register, one of the OHCI operational registers. When the IR bit of the HcControl register is initialized, an INTA interrupt is selected immediately after the reset as the reporting path.

There are no differences in interrupt generating sources between INTA and SMMI, except for an OwnershipChange.

Table 2-7. INTA and SMMI Interrupt Reporting Paths

IR Bit of HcControl Register	Interrupt Report Signal
0	INTA (default value)
1	SMMI

In order to use INTA and SMMI interrupts, the inta_en and int_smmi_en bits of the PCI interrupt control register, one of the PCI host bridge registers, must be set (1).

(b) Interrupt sources

Interrupts defined by the OpenHCI standard are supported.

Interrupt sources to be reported to the system are set to the HcInterruptEnable register and are reported via the interrupt path determined by the IR bit.

The interrupts sources are shown below.

Table 2-8. INTA and SMMI Interrupt Sources

<table>
<tr><th>Interrupt Source</th><th>Description</th></tr>
<tr><td>Scheduling Overrun</td><td>Interrupt indicates that a USB scheduling overrun has occurred in the frame.</td></tr>
<tr><td>Writeback DoneHead</td><td>Interrupt indicates that the host controller has completed TD and a writeback has occurred.</td></tr>
<tr><td>Start Of Frame</td><td>Interrupt indicates that HccaFmNumber was updated at the start of the frame.</td></tr>
<tr><td>Resume Detected</td><td>Interrupt indicates that a resume signal was detected from a USB-connected device.</td></tr>
<tr><td>Unrecoverable Error</td><td>Interrupt indicates that an error not related to USB (PCI abort) was detected.</td></tr>
<tr><td>Frame Number Overflow</td><td>Interrupt indicates that bit 15 of HcFmNumber has changed from 0 to 1 or from 1 to 0.</td></tr>
<tr><td>Root Hub Status Change</td><td>Interrupt indicates that the HcRhStatus or HcRhPortStatus setting has changed.
This is divided into the following specific events.

<table><tr><td>OverCurrentIndicateChange</td><td>Indicates that overcurrent has occurred.</td></tr><tr><td>Connect Status Change</td><td>Indicates that connection or disconnection has occurred in the USB bus.</td></tr><tr><td>Port EnableStatusChange</td><td>Indicates that port has transitioned to disable status by a USB error.</td></tr><tr><td>Port Suspend Status Change</td><td>Indicates that resume sequence has ended.</td></tr><tr><td>Port Reset Status Change</td><td>Indicates that USB reset has ended.</td></tr></table></td></tr>
<tr><td>Ownership Change</td><td>Interrupt indicates that an ownership request has been issued.
This is reported only to SMMI.</td></tr>
</table>

(2) PME interrupt

PME interrupt is an interrupt signal used for power management, and is used to report to the system a change in the USB bus while there is no PCLK.

The following lists whether events and interrupts that occur in the USB bus are supported.

Table 2-9. PME Interrupt Sources

Event	Interrupt Occurrence
Over Current Indicate	Does not occur.
Connect	Occurs.
Disconnect	Occurs.
Resume（RemoteWakeUp）	Occurs.

To use a PME interrupt, the PME enable bit of the Power_Management_Control/Status register (one of the PCI configuration registers) and the int_pme_en bit of the PCI interrupt control register (one of the PCI host bridge registers) must be set (1).

CHAPTER 3 USB FUNCTION CONTROLLER

3.1 Overview

The USB function controller (USBF) uses a token based protocol to transfer data to and from external host devices via the polling method.

It complies with the Universal Serial Bus Specification and has the following features.

- ○ Supports 12 Mbps (full-speed) transfer.
- ○ Equipped with a one-channel upstream port.
- ○ Incorporates the following transfer endpoints.

Table 3-1. USB Function Controller Endpoint Configuration

Endpoint	FIFO Size (Bytes)	Transfer Type	Remark
EP0	64	Control Read/Write	
EP1	64 × 2	Bulk In	Double buffer configuration
EP2	64 × 2	Bulk Out	Double buffer configuration
EP7	8	Interrupt	

- ○ Can perform DMA transfer (two-cycle single transfer mode) of bulk in or bulk out data.

3.2 CPU Memory Space

The CPU memory space is divided for use as described below. Do not access addresses that are not described as areas and are located between divided areas.

Table 3-2. CPU Memory Space Division

Base Address	Offset Address	Area
Address selected via CSZ5	4000H to 47FFH	EPC register area
	4800H to 480FH	Bridge register area
	4900H to 491FH	DMA register area
	4A00H	Bulk in register area
	4B00H	Bulk out register area
	4810H to 48FFH, 4920H to 49FFH, 4A01H to 4AFFH, 4B01H to 7FFFH	Reserved (access prohibited)

3.3 Requests

The USB includes commands called “requests” that are used to transfer requests from the host device to a function device to make the function device respond.

Requests are received at the setup stage during a control transfer, and most requests can be processed automatically by the hardware of the USB function controller (USBF).

3.3.1 Automatic requests

(1) Decode

The following tables show the request formats and correspondence between requests and decoded values.

Table 3-3. Request Format

Offset	Field Name	
0	bmRequestType	
1	bRequest	
2	wValue	Lower side
3		Higher side
4	wIndex	Lower side
5		Higher side
6	wLength	Lower side
7		Higher side

Table 3-4. Correspondence Between Requests and Decoded Values

Request \ Offset	Decoded Value								Response			Data Stage
	bmRequestType	bRequest	wValue		wIndex		wLength		Df	Ad	Cf	
	0	1	3	2	5	4	7	6				
GET_INTERFACE	81H	0AH	00H	00H	00H	0nH	00H	01H	STALL	STALL	ACK NAK	○
GET_CONFIGURATION	80H	08H	00H	00H	00H	00H	00H	01H	ACK NAK	ACK NAK	ACK NAK	○
GET_DESCRIPTOR Device	80H	06H	01H	00H	00H	00H	XXH	XXH[Note 1]	ACK NAK	ACK NAK	ACK NAK	○
GET_DESCRIPTOR Configuration	80H	06H	02H	00H	00H	00H	XXH	XXH[Note 1]	ACK NAK	ACK NAK	ACK NAK	○
GET_STATUS Device	80H	00H	00H	00H	00H	00H	00H	02H	ACK NAK	ACK NAK	ACK NAK	○
GET_STATUS Endpoint 0	82H	00H	00H	00H	00H	00H 80H	00H	02H	ACK NAK	ACK NAK	ACK NAK	○
GET_STATUS Endpoint X	82H	00H	00H	00H	00H	$$H	00H	02H	STALL	STALL	ACK NAK	○
CLEAR_FEATURE Device[Note 2]	00H	01H	00H	01H	00H	00H	00H	00H	ACK NAK	ACK NAK	ACK NAK	×
CLEAR_FEATURE Endpoint 0[Note 2]	02H	01H	00H	00H	00H	00H 80H	00H	00H	ACK NAK	ACK NAK	ACK NAK	×
CLEAR_FEATURE Endpoint X[Note 2]	02H	01H	00H	00H	00H	$$H	00H	00H	STALL	STALL	ACK NAK	×
SET_FEATURE Device[Note 3]	00H	03H	00H	01H	00H	00H	00H	00H	ACK NAK	ACK NAK	ACK NAK	×
SET_FEATURE Endpoint 0[Note 3]	02H	03H	00H	00H	00H	00H 80H	00H	00H	ACK NAK	ACK NAK	ACK NAK	×
SET_FEATURE Endpoint X[Note 3]	02H	03H	00H	00H	00H	$$H	00H	00H	STALL	STALL	ACK NAK	×
SET_INTERFACE	01H	0BH	00H	0#H	00H	0?H	00H	00H	STALL	STALL	ACK NAK	×
SET_CONFIGURATION[Note 4]	00H	09H	00H	00H 01H	00H	00H	00H	00H	ACK NAK	ACK NAK	ACK NAK	×
SET_ADDRESS	00H	05H	XXH	XXH	00H	00H	00H	00H	ACK NAK	ACK NAK	ACK NAK	×

Notes 1. If the wLength value is less than the prepared value, the wLength value is returned; if the wLength value is greater than the prepared value, the prepared value is returned.

2. The CLEAR_FEATURE request clears (0) UF0 device status register L (UF0DSTL) and UF0 EPn status register L (UF0EnSL) (n = 0 to 2, 7) when ACK is received in the status stage.

Notes 3. The SET_FEATURE request sets (1) the UF0 device status register L (UF0DSTL) and UF0 EPn status register L (UF0EnSL) (n = 0 to 2, 7) when ACK is received in the status stage. If the E0HALT bit of the UF0E0SL register is set, a STALL response is made in the status stage or data stage of control transfer for a request other than the GET_STATUS Endpoint0 request, SET_FEATURE Endpoint0 request, and a request generated by the CPUDEC interrupt request, until the CLEAR_FEATURE Endpoint0 request is received. A STALL response to an unsupported request does not set the E0HALT bit of the UF0E0SL register to 1, and the STALL response is cleared (0) as soon as the next SETUP token has been received.

4. If the wValue is not the default value, an automatic STALL response is made.

Cautions 1. The sequence of control transfer defined by the Universal Serial Bus Specification is not satisfied under the following conditions. The operation is not guaranteed under these conditions.

- **If an IN/OUT token is suddenly received without a SETUP stage**
- **If DATA PID1 is sent in the data phase of the SETUP stage**
- **If a token of 128 addresses or more is received**
- **If the request data transmitted in the SETUP stage is of less than 8 bytes**

2. An ACK response is made even when the host transmits data other than a Null packet in the status stage.

3. If the wLength value is 00H during control transfer (read) of firmware processing, a Null packet is automatically transmitted for control transfer (without data). The firmware request does not automatically transmit a Null packet.

Remarks 1. ○: Data stage is provided
×: Data stage is not provided

2. Df: Default state, Ad: Addressed state, Cf: Configured state

3. n = 0 to 4
It is determined by the setting of the UF0 active interface number register (UF0AIFN) whether a request with Interface number 1 to 4 is correctly responded to, depending on whether the Interface number of the target is valid or not.

4. $$: Valid endpoint number including transfer direction
The valid endpoint is determined by the currently set Alternate Setting number (see **3.4.4 (36) UF0 active alternate setting register (UF0AAS)**, **(38) UF0 endpoint 1 interface mapping register (UF0E1IM) to (40) UF0 endpoint 7 interface mapping register (UF0E7IM)**).

5. ? and #: Value transmitted from host (?: Interface numbers, #: Alternate Setting)
It is determined by the UF0 active interface number register (UF0AIFN) and UF0 active alternate setting register (UF0AAS) whether an Alternate Setting request corresponding to each Interface number is correctly responded to or not, depending on whether the Interface number and Alternate Setting of the target are valid or not.

(2) Processing

The processing of an automatic request in the Default state, Addressed state, and Configured state is described below.

- Default state: State in which an operation is performed with the Default address
- Addressed state: State after an address has been allocated
- Configured state: State after SET_CONFIGURATION wValue = 1 has been correctly received

(a) CLEAR_FEATURE request

A STALL response is made in the status stage if the CLEAR_FEATURE request cannot be cleared, if FEATURE does not exist, or if the target is an interface or an endpoint that does not exist. A STALL response is also made if the wLength value is other than 0.

- Default state: The correct response is made when the CLEAR_FEATURE request has been received only if the target is a device or a request for Endpoint0; otherwise a STALL response is made in the status stage.
- Addressed state: The correct response is made when the CLEAR_FEATURE request has been received only if the target is a device or a request for Endpoint0; otherwise a STALL response is made in the status stage.
- Configured state: The correct response is made when the CLEAR_FEATURE request has been received only if the target is a device or a request for an endpoint that exists; otherwise a STALL response is made in the status stage.

When the CLEAR_FEATURE request has been correctly processed, the corresponding bit of the UF0 CLR request register (UF0CLR) is set to 1, the EnHALT bit of the UF0 EPn status register L (UF0EnSL) is cleared to 0, and an interrupt is issued (n = 0 to 2, 7). If the CLEAR_FEATURE request is received when the subject is an endpoint, the toggle bit (that controls switching between DATA0 and DATA1) of the corresponding endpoint is always re-set to DATA0.

(b) GET_CONFIGURATION request

A STALL response is made in the data stage if any of wValue, wIndex, or wLength is other than the values shown in Table 3-4.

- Default state: The value stored in the UF0 configuration register (UF0CNF) is returned when the GET_CONFIGURATION request has been received.
- Addressed state: The value stored in the UF0CNF register is returned when the GET_CONFIGURATION request has been received.
- Configured state: The value stored in the UF0CNF register is returned when the GET_CONFIGURATION request has been received.

(c) GET_DESCRIPTOR request

If the subject descriptor has a length that is a multiple of wMaxPacketSize, a Null packet is returned to indicate the end of the data stage. If the length of the descriptor at this time is less than the wLength value, the entire descriptor is returned; if the length of the descriptor is greater than the wLength value, the descriptor up to the wLength value is returned.

- Default state: The value stored in UF0 device descriptor register n (UF0DDn) and UF0 configuration/interface/endpoint descriptor register m (UF0CIEm) is returned (n = 0 to 17, m = 0 to 255) when the GET_DESCRIPTOR request has been received.
- Addressed state: The value stored in the UF0DDn register and UF0CIEm register is returned when the GET_DESCRIPTOR request has been received.
- Configured state: The value stored in the UF0DDn register and UF0CIEm register is returned when the GET_DESCRIPTOR request has been received.

A descriptor of up to 256 bytes can be stored in the UF0CIEm register. To return a descriptor of more than 256 bytes, set the CDCGDST bit of the UF0MODC register to 1 and process the GET_DESCRIPTOR request by firmware.

Store the value of the total number of bytes of the descriptor set by the UF0CIEm register – 1 in the UF0 descriptor length register (UF0DSCL). The transfer data is controlled by the value of this data + 1 and wLength.

(d) GET_INTERFACE request

If either of wValue and wLength is other than that shown in Table 3-4, or if wIndex is other than that set by the UF0 active interface number register (UF0AIFN), a STALL response is made in the data stage.

- Default state: A STALL response is made in the data stage when the GET_INTERFACE request has been received.
- Addressed state: A STALL response is made in the data stage when the GET_INTERFACE request has been received.
- Configured state: The value stored in the UF0 interface n register (UF0IFn) corresponding to the wIndex value is returned (n = 0 to 4) when the GET_INTERFACE request has been received.

(e) GET_STATUS request

A STALL response is made in the data stage if any of wValue, wIndex, or wLength is other than the values shown in Table 3-4. A STALL response is also made in the data stage if the target is an interface or an endpoint that does not exist.

- Default state: The value stored in the target status register[Note] is returned only when the GET_STATUS request has been received and when the request is for a device or Endpoint0; otherwise a STALL response is made in the data stage.
- Addressed state: The value stored in the target status register[Note] is returned only when the GET_STATUS request has been received and when the request is for a device or Endpoint0; otherwise a STALL response is made in the data stage.
- Configured state: The value stored in the target status register[Note] is returned only when the GET_STATUS request has been received and when the request is for a device or an endpoint that exists; otherwise a STALL response is made in the data stage.

Note The target status register is as follows.

- If the target is a device: UF0 device status register L (UF0DSTL)
- If the target is endpoint 0: UF0 EP0 status register L (UF0E0SL)
- If the target is endpoint n: UF0 EPn status register L (UF0EnSL) (n = 1, 2, 7)

(f) SET_ADDRESS request

A STALL response is made in the status stage if either of wIndex or wLength is other than the values shown in Table 3-4. A STALL response is also made if the specified device address is greater than 127.

- Default state: The device enters the Addressed state and changes the USB Address value to be input to SIE into a specified address value if the specified address is other than 0 when the SET_ADDRESS request has been received. If the specified address is 0, the device remains in the Default state.
- Addressed state: The device enters the Default state and returns the USB Address value to be input to SIE to the default address if the specified address is 0 when the SET_ADDRESS request has been received. If the specified address is other than 0, the device remains in the Addressed state, and changes the USB Address value to be input to SIE into a specified new address value.
- Configured state: The device remains in the Configured state and returns the USB Address value to be input to SIE to the default address if the specified address is 0 when the SET_ADDRESS request has been received. In this case, the endpoints other than endpoint 0 remain valid, and control transfer (IN), control transfer (OUT), bulk transfer and interrupt transfer for an endpoint other than endpoint 0 are also acknowledged. If the specified address is other than 0, the device remains in the Configured state and changes the USB Address value to be input to SIE into a specified new address value.

(g) SET_CONFIGURATION request

If any of wValue, wIndex, or wLength is other than the values shown in Table 3-4, a STALL response is made in the status stage.

- Default state: The CONF bit of the UF0 mode status register (UF0MODS) is set to 1 and 1 is set to the UF0 configuration register (UF0CNF) If the specified configuration value is 1 when the SET_CONFIGURATION request has been received. If the specified configuration value is 0, the CONF bit of the UF0MODS register is cleared to 0 and 0 is set to the UF0CNF register. In other words, the device skips the Addressed state and moves to the Configured state in which it responds to the Default address.
- Addressed state: The CONF bit of the UF0MODS register is set to 1 and 1 is set to the UF0CNF register and the device enters the Configured state if the specified configuration value is 1 when the SET_CONFIGURATION request has been received. If the specified configuration value is 0, the device remains in the Addressed state.
- Configured state: The CONF bit of the UF0MODS register is cleared to 0 and 0 is set to the UF0CNF register and the device returns to the Addressed state if the specified configuration value is 0 when the SET_CONFIGURATION request has been received. If the specified configuration value is 1, the device remains in the Configured state.

If the SET_CONFIGURATION request has been correctly processed, the target bit of the UF0 SET request register (UF0SET) is set to 1, and an interrupt is issued. All Halt Features are cleared after the SET_CONFIGURATION request has been completed even if the specified configuration value is the same as the current configuration value. If the SET_CONFIGURATION request has been correctly processed, the data toggle of all endpoints is always initialized to DATA0 again (it is defined that the default status, Alternate Setting 0, is set from when the SET_CONFIGURATION request is received to when the SET_INTERFACE request is received).

(h) SET_FEATURE request

A STALL response is made in the status stage if the target is an interface or an endpoint that does not exist. A STALL response is also made if the wLength value is other than 0.

- Default state: The correct response is made when the SET_FEATURE request has been received, only if the request is for a device or Endpoint0; otherwise a STALL response is made in the status stage.
- Addressed state: The correct response is made when the SET_FEATURE request has been received, only if the request is for a device or Endpoint0; otherwise a STALL response is made in the status stage.
- Configured state: The correct response is made when the SET_FEATURE request has been received, only if the request is for a device or an endpoint that exists; otherwise a STALL response is made in the status stage.

When the SET_FEATURE request has been correctly processed, the target bit of the UF0 SET request register (UF0SET) and the EnHALT bit of the UF0 EPn status register L (UF0EnSL) are set to 1, and an interrupt is issued (n = 0 to 2, 7).

(i) SET_INTERFACE request

If wLength is other than the values shown in Table 3-4, if wIndex is other than the value set to the UF0 active interface number register (UF0AIFN), or if wValue is other than the value set to the UF0 active alternate setting register (UF0AAS), a STALL response is made in the status stage.

- Default state: A STALL response is made in the status stage when the SET_INTERFACE request has been received.
- Addressed state: A STALL response is made in the status stage when the SET_INTERFACE request has been received.
- Configured state: Null packet is transmitted in the status stage when the SET_INTERFACE request has been received.

When the SET_INTERFACE request is processed normally, interrupts are issued.

All the Halt Features of the endpoint linked to the target Interface are cleared (0) after the SET_INTERFACE request has been cleared. The data toggle of all the endpoints related to the target Interface number is always initialized again to DATA0. When the currently selected Alternate Setting is to be changed by correctly processing the SET_INTERFACE request, the FIFO of the endpoint that is affected is completely cleared (0), and all the related interrupt sources are also initialized.

When the SET_INTERFACE request has been completed, the FIFO of all the endpoints linked to the target Interface are cleared (0). At the same time, Halt Feature and Data PID are initialized, and the related UF0 INT status n register (UF0ISn) is cleared to 0 (n = 0 to 4). (Only Halt Feature and Data PID are cleared (0) when the SET_CONFIGURATION request has been completed.)

If the target Endpoint is not supported by the SET_INTERFACE request during DMA transfer, the DMA request signal is immediately deasserted, and the FIFO of the Endpoint that has been linked when the SET_INTERFACE request has been completed is completely cleared (0). As a result of this clearing (0) of the FIFO, data transferred by DMA is not correctly processed.

3.3.2 Other requests

(1) Response and processing

The following table shows how other requests are responded to and processed.

Table 3-5. Response and Processing of Other Requests

Request	Response and Processing
GET_DESCRIPTOR String	Generation of CPUDEC interrupt request
GET_STATUS Interface	Automatic STALL response
CLEAR_FEATURE Interface	Automatic STALL response
SET_FEATURE Interface	Automatic STALL response
all SET_DESCRIPTOR	Generation of CPUDEC interrupt request
All other requests	Generation of CPUDEC interrupt request

3.4 Registers

The USB function controller (USBF) is equipped with the following registers. Since it is provided with a bus interface of an 8-bit width, only the lower 8 bits are used for bus access (32 bits) from the CPU. Consequently, registers are assigned at an interval of four addresses.

Since the registers are freely mapped by CSZ5, the CSZ5 start address is set as the base address.

(Register address = base address + offset address)

Table 3-6. Function Bridge Register

Base Address	Offset Address	Description	Symbol	R/W	Access Size	Bit Size	Reset
CSZ5 start address	4800H	MEMC Bus Bridge Interrupt Control	MEMC_INT	R/W	32bit	8bit	00H
	4804H	MEMC Bus Bridge Interrupt Enable	MEMC_INTEN	R/W	32bit	8bit	00H
	4808H	EPC Macro Control	EPC_CTR	R/W	32bit	8bit	00H

Table 3-7. DMA Register

Base Address	Offset Address	Description	Symbol	R/W	Access Size	Bit Size	Reset
CSZ5 start address	4900H	Endpoint 1 DMA Control1	EP1_DCR1	R/W	32bit	8bit	00H
	4904H	Endpoint 1 DMA Control2 Bit [7:0]	EP1L_DCR2	R/W	32bit	8bit	00H
	4908H	Endpoint 1 DMA Control2 Bit [15:8]	EP1M_DCR2	R/W	32bit	8bit	00H
	490CH	Endpoint 1 DMA Control2 Bit [23:16]	EP1H_DCR2	R/W	32bit	8bit	00H
	4910H	Endpoint 2 DMA Control1	EP2_DCR1	R/W	32bit	8bit	00H
	4914H	Endpoint 2 DMA Control2 Bit [7:0]	EP2L_DCR2	R/W	32bit	8bit	00H
	4918H	Endpoint 2 DMA Control2 Bit [15:8]	EP2M_DCR2	R/W	32bit	8bit	00H
	491CH	Endpoint 2 DMA Control2 Bit [23:16]	EP2H_DCR2	R/W	32bit	8bit	00H

Table 3-8. Bulk in/Bulk out Register

Base Address	Offset Address	Description	Symbol	R/W	Access Size	Bit Size	Reset
CSZ5 start address	4A00H	BulkIn1 for DMA transfer	EP1_BULK_IN	W	32bit	8bit	00H
	4B00H	BulkOut1 for DMA transfer	EP2_BULK_OUT	R	32bit	8bit	00H

Table 3-9. EPC Control Register

(1/2)

Base Address	Offset Address	Description	Symbol	R/W	Access Size	Bit Size	Reset
CSZ5 start address	4000H	EP0NAK (EP0 NAK Control)	UF0E0N	R/W	32bit	8bit	00H
	4004H	EP0NAKALL (EP0NAKALL EP0 NAK Control)	UF0E0NA	R/W	32bit	8bit	00H
	4008H	EPNAK (EP except EP0 NAK Control)	UF0EN	R/W	32bit	8bit	00H
	400CH	EPNAK Mask	UF0ENM	R/W	32bit	8bit	00H
	4010H	SNDSIE	UF0SDS	R/W	32bit	8bit	00H
	4014H	CLR Request	UF0CLR	R	32bit	8bit	00H
	4018H	SET Request	UF0SET	R	32bit	8bit	00H
	401CH	EP Status 0 (EP FIFO Status)	UF0EPS0	R	32bit	8bit	00H
	4020H	EP Status 1 (EP FIFO and USB bus Status)	UF0EPS1	R	32bit	8bit	00H
	4024H	EP Status 2 (EP Halt Status)	UF0EPS2	R	32bit	8bit	00H
	4028H to 403CH	Reserved		–	–	–	–
	4040H	INT Status 0	UF0IS0	R	32bit	8bit	00H
	4044H	INT Status 1	UF0IS1	R	32bit	8bit	00H
	4048H	INT Status 2	UF0IS2	R	32bit	8bit	00H
	404CH	INT Status 3	UF0IS3	R	32bit	8bit	00H
	4050H	INT Status 4	UF0IS4	R	32bit	8bit	00H
	4054H to 4058H	Reserved		–	–	–	–
	405CH	INT Mask 0	UF0IM0	R/W	32bit	8bit	00H
	4060H	INT Mask 1	UF0IM1	R/W	32bit	8bit	00H
	4064H	INT Mask 2	UF0IM2	R/W	32bit	8bit	00H
	4068H	INT Mask 3	UF0IM3	R/W	32bit	8bit	00H
	406CH	INT Mask 4	UF0IM4	R/W	32bit	8bit	00H
	4070H to 4074H	Reserved		–	–	–	–
	4078H	INT Clear 0	UF0IC0	W	32bit	8bit	FFH
	407CH	INT Clear 1	UF0IC1	W	32bit	8bit	FFH
	4080H	INT Clear 2	UF0IC2	W	32bit	8bit	FFH
	4084H	INT Clear 3	UF0IC3	W	32bit	8bit	FFH
	4088H	INT Clear 4	UF0IC4	W	32bit	8bit	FFH
	408CH to 4094H	Reserved		–	–	–	–
	4098H	INT & DMARQ	UF0IDR	R/W	32bit	8bit	00H
	409CH	DMA Status 0	UF0DMS0	R	32bit	8bit	00H
	40A0H	DMA Status 1	UF0DMS1	R	32bit	8bit	00H
	40A4H to 40BCH	Reserved		–	–	–	–
	40C0H	FIFO Clear 0	UF0FIC0	W	32bit	8bit	00H
	40C4H	FIFO Clear 1	UF0FIC1	W	32bit	8bit	00H
	40C8H to 40D0H	Reserved		–	–	–	–

Table 3-9. EPC Control Register

(2/2)

Base Address	Offset Address	Description	Symbol	R/W	Access Size	Bit Size	Reset
CSZ5 start address	40D4H	Data End	UF0DEND	W	32bit	8bit	00H
	40D8H	Reserved		–	–	–	–
	40DCH	GPR (Macro and USB I/F Control)	UF0GPR	W	32bit	8bit	00H
	40E0H to 40E4H	Reserved		–	–	–	–
	40E8H	Mode Control Reg (CPUDEC)	UF0MODC	R/W	32bit	8bit	00H
	40ECH	Reserved		–	–	–	–
	40F0H	Mode Status Reg (Configuration)	UF0MODS	R	32bit	8bit	00H
	40F4H to 40FCH	Reserved		–	–	–	–
	4100H	Active Interface No.	UF0AIFN	R/W	32bit	8bit	00H
	4104H	Active Alternate Setting	UF0AAS	R/W	32bit	8bit	00H
	4108H	Alternate Setting Status	UF0ASS	R	32bit	8bit	00H
	410CH	EP1 Interface Mapping	UF0E1IM	R/W	32bit	8bit	00H
	4110H	EP2 Interface Mapping	UF0E2IM	R/W	32bit	8bit	00H
	4114H to 4120H	Reserved		–	–	–	–
	4124H	EP7 Interface Mapping	UF0E7IM	R/W	32bit	8bit	00H
	4128H to 41FCH	Reserved		–	–	–	–

Table 3-10. EPC Data Hold Register

Base Address	Offset Address	Description	Symbol	R/W	Access Size	Bit Size	Reset
CSZ5 start address	4200H	EP0 Read	UF0E0R	R	32bit	8bit	Undefined
	4204H	EP0 Length	UF0E0L	R	32bit	8bit	00H
	4208H	EP0 Setup	UF0E0ST	R	32bit	8bit	00H
	420CH	EP0 Write	UF0E0W	W	32bit	8bit	Undefined
	4210H	BulkOut1 for PIO transfer	UF0BO1	R	32bit	8bit	Undefined
	4214H	BulkOut1 Length	UF0BO1L	R	32bit	8bit	00H
	4218H to 421CH	Reserved		–	–	–	–
	4220H	BulkIn1 for PIO transfer	UF0BI1	W	32bit	8bit	Undefined
	4224H	Reserved		–	–	–	–
	4228H	Interrupt 1	UF0INT1	W	32bit	8bit	Undefined
	422CH to 4284H	Reserved		–	–	–	–

Table 3-11. EPC Request Data Register

Base Address	Offset Address	Description	Symbol	R/W	Access Size	Bit Size	Reset
CSZ5 start address	4288H	Device Status	UF0DSTL	R/W	32bit	8bit	00H
	428CH to 4294H	Reserved		–	–	–	–
	4298H	EP0 Status	UF0E0SL	R/W	32bit	8bit	00H
	42A0H	EP1 Status	UF0E1SL	R/W	32bit	8bit	00H
	42A8H	EP2 Status	UF0E2SL	R/W	32bit	8bit	00H
	42ACH to 42CCH	Reserved		–	–	–	–
	42D0H	EP7 Status	UF0E7SL	R/W	32bit	8bit	00H
	42D4H to 42FCH	Reserved		–	–	–	–
	4300H	Address	UF0ADRS	R/W	32bit	8bit	00H
	4304H	Configuration	UF0CNF	R/W	32bit	8bit	00H
	4308H	Interface 0	UF0IF0	R/W	32bit	8bit	00H
	430CH	Interface 1	UF0IF1	R/W	32bit	8bit	00H
	4310H	Interface 2	UF0IF2	R/W	32bit	8bit	00H
	4314H	Interface 3	UF0IF3	R/W	32bit	8bit	00H
	4318H	Interface 4	UF0IF4	R/W	32bit	8bit	00H
	431CH to 433CH	Reserved		–	–	–	–
	4340H	Descriptor Length	UF0DSCL	R/W	32bit	8bit	00H
	4344H to 4388H	Device Descriptor	UF0DD0 to UF0DD17	R/W	32bit	8bit	Undefined
	438CH to 4788H	Configuration Descriptor	UF0CIE0 to UF0CIE255	R/W	32bit	8bit	Undefined
	478CH to	Reserved		–	–	–	–

3.4.1 Function bridge registers

(1) MEMC_INT

This register indicates the interrupt source that occurred in the function bridge.

	31 6	5	4	3 2	1	0	Offset address	After reset
MEMC_INT	Reserved	EPC_ INT1B	EPC_ INT0B	Reserved	EP2_ ENDINT	EP1_ ENDINT	4800H	0000 0000H
	R	R	R	R	RW	RW		

Bit position	Bit name	Function
5	EPC_INT1B	This bit indicates whether there are any interrupts triggered by events in the UF0IS2/3 register. It is cleared (0) by the EPC register. 0: No interrupts 1: Interrupts occurred
4	EPC_INT0B	This bit indicates whether there are any interrupts triggered by events in the UF0IS0/1 register. It is cleared (0) by the EPC register. 0: No interrupts 1: Interrupts occurred
1	EP2_ENDINT	In EP2, this bit is set (1) when a DMA transfer is completed normally or abnormally with an error. This bit is cleared (0) when “1” is written to it. 1: DMA transfer completed
0	EP1_ENDINT	In EP1, this bit is set (1) when a DMA transfer is completed normally or abnormally with an error. This bit is cleared (0) when “1” is written to it. 1: DMA transfer completed

<R> The condition and the timing when the EP2_ENDINTand EP1_ENDINT bits are set are shown below.

- DMA transfer completed normally

  The bits are set (1) after the EPC register deasserts DMAREQ, and the buffers in the bridge empties.

- DMA transfer completed abnormally with an error

  (a) The bits are set (1) after the EPC register asserts DMASTOP, and the buffers in the bridge empties when the EPC register receives a short packet.

  (b) The bits are set (1) when EPn_TCNT is decremented every transfer to become 0.

(2) MEMC_INTEN

This register sets whether to output interrupts that occurred in the function bridge.

	31 6	5	4	3 2	1	0	Offset address	After reset
MEMC_INTEN	Reserved	EPC_INT1BEN	EPC_INT0BEN	Reserved	EP2_ENDINTEN	EP1_ENDINTEN	4804H	0000 0000H
	R	RW	RW	R	RW	RW		

Bit position	Bit name	Function
5	EPC_INT1EN	This bit sets whether to generate an interrupt when the EPC_INT1B bit is set. 1: Outputs interrupt. 0: Does not output interrupt.
4	EPC_INT0EN	This bit sets whether to generate an interrupt when the EPC_INT0B bit is set. 1: Outputs interrupt. 0: Does not output interrupt.
1	EP2_ENDINTEN	This bit sets whether to generate an interrupt when the EP2_ENDINT bit is set. 1: Outputs interrupt. 0: Does not output interrupt.
0	EP1_ENDINTEN	This bit sets whether to generate an interrupt when the EP1_ENDINT bit is set. 1: Outputs interrupt. 0: Does not output interrupt.

(3) EPC_CTR

This register controls resetting of the EPC macro.

	31 1	0	Offset address	After reset
EPC_CTR	Reserved	EPC_RST	4808H	0000 0000H
	R	RW		

Bit position	Bit name	Function
0	EPC_RST	This bit generates a reset to the EPC macro. 0: Releases reset. 1: Issues reset.

3.4.2 DMA registers

(1) EP1_DCR1

This register is used to set control of DMA transfer of EP1.

	31 6	5 3	2	1	0	Offset address	After reset
EP1_DCR1	Reserved	Reserved	EP1_ STOPSTA	EP1_ REQSTA	EP1_ DMAEN	4900H	0000 0000H
	R	R	R	R	RW		

Bit position	Bit name	Function
2	EP1_STOPSTA	This bit indicates the status (DMA transfer end source) of completion of DMA transfer from EPC. 1: Ends DMA transfer by negating DQBI1MS of the UF0IDR register. 0: Ends DMA transfer by clearing EP1_TCNT value of the EP1x_DCR2 register to “0”. This bit is cleared (0) automatically the next time "1" is set to EP1_DMAEN.
1	EP1_REQSTA	This bit indicates the status of DMA requests (DQBI1MS of UF0IDR register) from EPC. 0: DMA request issued 1: No DMA requests
0	EP1_DMAEN	This bit sets control of DMA requests from EPC. 0: Masks DMA requests. 1: Enables DMA requests. This bit is cleared (0) automatically when the number of packets set via EP1_TCNT have been transferred or when DQBI1MS of the UF0IDR register is negated to end DMA transfer. **Caution Set values are not guaranteed after a forcible termination.**

(2) EP1x_DCR2

This register sets the DMA transfer size of EP1.

	31 8	7 0	Offset address	After reset
EP1L_DCR2	Reserved	EP1_TCNT[7:0]	4904H	0000 0000H
EP1M_DCR2	Reserved	EP1_TCNT[15:8]	4908H	0000 0000H
EP1H_DCR2	Reserved	EP1_TCNT[23:16]	490CH	0000 0000H
	R	RW		

Bit position	Bit name	Function
7 to 0	EP1_TCNT	The number of bytes to be transferred by DMA transfer is set by EP1. This value is decremented at each transfer, and when the EP1_TCNT value reaches "0", the DMA transfer ends. **Cautions 1. Set this bit when EP1_DMAEN = 0.** **2. This register uses a 1-origin setting. It is assumed to be "value set to the DMAC transfer size setting register DXBCn (0 origin) + 1".** **3. After a forcible termination, this is updated to the value at which bulk transfer counter BIN_TCNT within the bridge has stopped.**

(3) EP2_DCR1

This register is used to set control of DMA transfer of EP2.

	31 6	5 3	2	1	0	Offset address	After reset
EP2_DCR1	Reserved	Reserved	EP2_ STOPSTA	EP2_ REQSTA	EP2_ DMAEN	4910H	0000 0000H
	R	R	R	R	RW		

Bit position	Bit name	Function
2	EP2_STOPSTA	This bit indicates the status of completion of DMA transfer from EPC. DMA transfer end source 1: Ends DMA transfer by negating DQBO1MS of the UF0IDR register. 0: Ends DMA transfer by clearing EP2_TCNT value to “0”. This bit is cleared to 0 automatically the next time “1” is set to EP2_DMAEN.
1	EP2_REQSTA	This bit indicates the status of DMA requests (DQBO1MS of UF0IDR register) from EPC. 0: DMA request issued 1: No DMA requests
0	EP2_DMAEN	This bit sets control of DMA requests from EPC. 0: Masks DMA requests. 1: Enables DMA requests. This bit is cleared (0) automatically when the number of packets set via EP2_TCNT have been transferred or when DQBO1MS of the UF0IDR register is negated to end DMA transfer. **Caution Set values are not guaranteed after a forcible termination.**

(4) EP2x_DCR2

This register sets the DMA transfer size of EP2.

	31 8	7 0	Offset address	After reset
EP2L_DCR2	Reserved	EP2_TCNT[7:0]	4914H	0000 0000H
EP2M_DCR2	Reserved	EP2_TCNT[15:8]	4918H	0000 0000H
EP2H_DCR2	Reserved	EP2_TCNT[23:16]	491CH	0000 0000H
	R	RW		

Bit position	Bit name	Function
7 to 0	EP2_TCNT	The number of bytes to be transferred by DMA transfer is set by EP2. This value is decremented at each transfer, and when the EP2_TCNT value reaches "0", the DMA transfer ends. **Cautions 1. Set this bit when EP2_DMAEN = 0.** **2. This register uses a 1-origin setting. It is assumed to be "value set to the DMAC transfer size setting register DXBCn (0 origin) + 1".** **3. After a forcible termination, this is updated to the value at which bulk transfer counter BOUT_TCNT within the bridge has stopped.**

3.4.3 Bulk in and bulk out registers

(1) EP1_BULK_IN

This register writes bulk in transfer data in DMA mode (see UF0IDR).

	31	8	7	0	Offset address	After reset
EP1_BULK_IN	Reserved		EP1_BULK_IN		4A00H	0000 0000H
	R		RW			

Bit position	Bit name	Function
7 to 0	EP1_BULK_IN[7:0]	This register writes BULK IN transfer data for DMA transfer. When data is written to this register, data is output to the EPC macro. To use this register, set the address of this register to address setting register DXDAn of DMAC. **Caution The only supported DMAC transfer type is two-cycle transfer.**

(2) EP2_BULK_OUT

This register reads bulk out transfer data in DMA mode (see UF0IDR).

	31	8	7	0	Offset address	After reset
EP2_BULK_OUT	Reserved		EP2_BULK_OUT		4B00H	0000 0000H
	R		R			

Bit position	Bit name	Function
7 to 0	EP2_BULK_OUT[7:0]	This register reads BULK_OUT transfer data for DMA transfer. It is used to read the data input from the EPC macro. To use this register, set the address of this register to address setting register DXSAn of DMAC. **Caution The only supported DMAC transfer type is two-cycle transfer.**

3.4.4 EPC Control registers

(1) UF0 EP0NAK register (UF0E0N)

This register controls NAK of Endpoint0 (except an automatically executed request).

This register can be read or written in 8-bit units (however, bit 0 can only be read).

It takes five USB clocks to reflect the status on this register after the UF0FIC0 and UF0FIC1 registers have been set. If it is necessary to read the status correctly, therefore, separate a write signal that accesses the UF0FIC0 and UF0FIC1 registers from a read signal that accesses the UF0EPS0, UF0EPS1, UF0EPS2, UF0E0N, and UF0EN registers by at least four USB clocks.

While NAK is being transmitted to Endpoint0 Read and Endpoint2 a write access to the EP0NKR bit is ignored.

	7	6	5	4	3	2	1	0	Address	After reset
UF0E0N	0	0	0	0	0	0	EP0NKR	EP0NKW	4000H	00H
	0	0	0	0	0	0	R/W	R		

Bit position	Bit name	Function
1	EP0NKR	This bit controls NAK to the OUT token to Endpoint0 (except an automatically executed request). It is automatically set to 1 by hardware when Endpoint0 has correctly received data. It is also cleared to 0 by hardware when the data of the UF0E0R register has been read by firmware (counter value = 0). 1: Transmit NAK. 0: Do not transmit NAK (default value). Set this bit to 1 by firmware when data should not be received from the USB bus for some reason even when USBF is ready for receiving data. In this case, USBF continues transmitting NAK until this bit is cleared to 0 by firmware. This bit is also cleared to 0 as soon as the UF0E0R register has been cleared.
0	EP0NKW	This bit indicates how NAK to the IN token to Endpoint0 is controlled (except an automatically executed request). This bit is automatically cleared to 0 by hardware when the data of Endpoint0 is transmitted and the host correctly receives the transmitted data. The data of the UF0E0W register is retained until this bit is cleared. Therefore, it is not necessary to rewrite this bit even in the case of a retransmission request that is made if the host could not receive data correctly. To send a short packet, be sure to set the E0DED bit of the UF0DEND register to 1. This bit is automatically set to 1 when the FIFO is full. As soon as the E0DED bit of the UF0DEND register is set to 1, the EP0NKW bit is automatically set to 1 at the same time. 1: Do not transmit NAK. 0: Transmit NAK (default value). If control transfer enters the status stage while ACK cannot be correctly received in the data stage, this bit is cleared to 0 as soon as the UF0E0W register is cleared. This bit is also cleared to 0 when UF0E0W is cleared by firmware.

Next, the procedure of a SETUP transaction that uses IN/OUT tokens is explained below.

(a) When IN token is used (except a request automatically executed by hardware)

Firmware should be used to clear the PROT bit of the UF0IS1 register to 0 after receiving the CPUDEC interrupt and before reading data from the UF0E0ST register. Next, perform processing in accordance with the request and, if it is necessary to return data by an IN token, write data to the UF0E0W register. Confirm that the PROT bit of the UF0IS1 register is 0 after writing has been completed, and set the E0DED bit of the UF0DEND register to 1. The hardware sends out data at the first IN token after the EP0NKW bit has been set to 1. If the PROT bit of the UF0IS1 register is 1, it indicates that a SETUP transaction has occurred again before completion of control transfer. In this case, clear the PROT bit of the UF0IS1 register to 0 by clearing the PROT bit of the UF0IC1 register to 0, and then read data from the UF0E0ST register again. A request received can be read later.

(b) When OUT token is used (except a request automatically executed by hardware)

Firmware should be used to clear the PROT bit of the UF0IS1 register after receiving the CPUDEC interrupt and before reading data from the UF0E0ST register. Confirm that the PROT bit of the UF0IS1 register is 0 before reading data from the UF0E0R register. If the PROT bit is 1, it means that invalid data is retained. Clear the FIFO by firmware (the EP0NKR bit is automatically cleared to 0). If the PROT bit of the UF0IS1 register is 0, read the data of the UF0E0L register and read as many data from the UF0E0R register as set. When reading data from the UF0E0R register has been completed (when the counter of the UF0E0R register has been cleared to 0), the hardware automatically clears the EP0NKR bit to 0.

(2) UF0 EP0NAKALL register (UF0E0NA)

This register controls NAK to all the requests of Endpoint0 except a SETUP transaction. It is also valid for automatically executed requests.

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

	7	6	5	4	3	2	1	0	Address	After reset
UF0E0NA	0	0	0	0	0	0	0	EP0NKA	4004H	00H
	0	0	0	0	0	0	0	R/W		

Bit position	Bit name	Function
0	EP0NKA	This bit controls NAK to a transaction other than a SETUP transaction to Endpoint0 (including an automatically executed request). This bit is manipulated by firmware. 1: Transmit NAK. 0: Do not transmit NAK (default value). This register is used to prevent a conflict between a write access by firmware and a read access from SIE when the data used for an automatically executed request is to be changed. It postpones reflecting a write access on this bit from firmware while an access from SIE is being made. Before rewriting the request data register from firmware, confirm that this bit has been correctly set to 1. Setting this bit to 1 is reflected only in the following cases. • Immediately after USBF has been reset and a SETUP token has never been received • Immediately after reception of USB Bus Reset and a SETUP token has never been received • PID of a SETUP token has been detected • The stage has been changed to the status stage Clearing this bit to 0 is reflected immediately, except while an IN token is being received and a NAK response is being made. Setting the EP0NKA bit to 1 is reflected in the above four cases during Endpoint0 transfer, but it is reflected immediately after data has been written to the bit while Endpoint0 is transferring no data.

(3) UF0 EPNAK register (UF0EN)

This register controls NAK of endpoints other than Endpoint0.

This register can be read or written in 8-bit units (however, bits 4 and 0 can only be read).

The BKO1NK bit can be written only when the BKO1NKM bit of the UF0ENM register is 1.

The related bits are invalid if each endpoint is not supported by the setting of the UF0EnIM register (n = 1, 2, 7) and the current setting of the interface.

It takes five USB clocks to reflect the status on this register after the UF0FIC0 and UF0FIC1 registers have been set. If it is necessary to read the status correctly, therefore, separate a write signal that accesses the UF0FIC0 and UF0FIC1 registers from a read signal that accesses the UF0EPS0, UF0EPS1, UF0EPS2, UF0E0N, and UF0EN registers by at least four USB clocks.

While NAK is being transmitted to Endpoint0 Read and Endpoint2, a write access to the BKO1NK bit is ignored.

Be sure to clear bits 7 to 5, 3, and 1 to “0”. If it is set to “1”, the operation is not guaranteed.

(1/2)

	7	6	5	4	3	2	1	0	Address	After reset
UF0EN	0	0	0	IT1NK	0	BKO1NK	0	BKI1NK	4008H	00H
	0	0	0	0	0	R/W	0	R		

Bit position	Bit name	Function
4	IT1NK	This bit controls NAK to Endpoint7 (interrupt 1 transfer). It is automatically set to 1 and transmission is started when the UF0INT1 register has become full as a result of writing data to it. To send a short packet that does not make the FIFO full, set the IT1DEND bit of the UF0DEND register to 1. As soon as the IT1DEND bit has been set to 1, this bit is automatically set to 1. 1: Do not transmit NAK. 0: Transmit NAK (default value). This bit is also cleared to 0 when the UF0INT1 register has been cleared.
2	BKO1NK	This bit controls NAK to Endpoint2 (bulk 1 transfer (OUT)). 1: Transmit NAK. 0: Do not transmit NAK (default value). This bit is set to 1 only when the FIFO connected to the SIE side of the UF0BO1 register (64-byte FIFO of bank configuration) cannot receive data. It is cleared to 0 when a toggle operation is performed. The bank is changed (toggle operation) when the following conditions are satisfied. • Data correctly received is stored in the FIFO connected to the SIE side. • The value of the FIFO counter connected to the CPU side is 0 (completion of reading). Firmware should be used to read data of the UF0BO1L register when it has received the BLKO1DT interrupt request and read as many data from the UF0BO1 register as the value of that data. To not receive data from the USB bus for some reason even if USBF is ready to receive data, set this bit to 1 by firmware. In this case, USBF keeps transmitting NAK until the firmware clears this bit to 0. This bit is also cleared to 0 as soon as the UF0BO1 register has been cleared.

Cautions 1. If DMA is enabled while data is being read from the UF0BO1 register in the PIO mode, a DMA request is immediately issued.

2. If the last data of the FIFO on the CPU side is read in the DMA transfer mode, the DMA request signal becomes inactive.

(2/2)

Bit position	Bit name	Function
0	BKI1NK	This bit controls NAK to Endpoint1 (bulk 1 transfer (IN)). 1: Transmit write data. 0: Transmit NAK (default value). This bit is cleared to 0 only when the FIFO connected to the SIE side of the UF0BI1 register (64-byte FIFO of bank configuration) cannot receive data. It is set to 1 when a toggle operation is performed (the data of the UF0BI1 register is retained until transmission has been correctly completed). The bank is changed (toggle operation) when the following conditions are satisfied. • Data is correctly written to the FIFO connected to the CPU bus side (writing has been completed and the FIFO is full or the UF0DEND register is set). • The value of the FIFO counter connected to the SIE side is 0. This bit is automatically set to 1 and data transmission is started when the FIFO on the CPU side becomes full and a FIFO toggle operation is performed as a result of writing data to the FIFO. However, if the FIFO on the CPU side becomes full as a result of writing data to it by DMA while the BKI1T bit of the UF0DEND register is cleared to 0, the toggle operation is not performed because the condition of the toggle operation is not satisfied until the BKI1DED bit of the UF0DEND register is set to 1. To send a short packet that does not make the FIFO on the CPU side full, set the BKI1DED bit to 1 after completing writing data. When the BKI1DED bit is set to 1, a toggle operation is performed and at the same time, this bit is automatically set to 1. This bit is also cleared to 0 as soon as the UF0BI1 register has been cleared.

<R>

Cautions 1. If DMA is enabled while data is being written to the UF0BI1 register in the PIO mode, a DMA request is immediately issued.

2. If 64-byte data is written in the DMA transfer mode, the DMA request signal becomes inactive. If the BKI1NK bit is then set to 1, data is transmitted in synchronization with an IN token. The DMA request signal becomes active again as long as the DMA request is not masked as soon as the FIFO is toggled. If the BKI1NK bit is not set, data is not transmitted even if an IN token has been received. In this case, set the BKI1DED bit of the UF0DEND register to 1.

3. In the DMA transfer mode, if the BKI1NK bit is not set to 1, data is not transmitted even if an IN token is received. When the BKI1DED bit of the UF0DEND register is set to 1 by firmware, data is transmitted in synchronization with the IN token.

(4) UF0 EPNAK mask register (UF0ENM)

This register controls masking a write access to the UF0EN register.

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

Be sure to clear bits 7 to 3, 1, and 0 to “0”. If it is set to “1”, the operation is not guaranteed.

	7	6	5	4	3	2	1	0	Address	After reset
UF0ENM	0	0	0	0	0	BKO1NKM	0	0	400CH	00H
	0	0	0	0	0	R/W	0	0		

Bit position	Bit name	Function
2	BKO1NKM	This bit specifies whether a write access to bit 2 (BKO1NK) of the UF0EN register is masked or not. 1: Do not mask. 0: Mask (default value).

(5) UF0 SNDSIE register (UF0SDS)

This register performs manipulation such as no handshake. It can directly manipulate the pins of SIE.

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

Be sure to clear bits 7 to 4, 2, and 1 to “0”. If it is set to “1”, the operation is not guaranteed.

	7	6	5	4	3	2	1	0	Address	After reset
UF0SDS	0	0	0	0	SNDSTL	0	0	RSUMIN	4010H	00H
	0	0	0	0	R/W	0	0	R/W		

Bit position	Bit name	Function
3	SNDSTL	This bit makes Endpoint0 issue a STALL handshake. Setting this bit to 1 if a request for CPUDEC processing is not supported by the system results in a STALL handshake response. If an unsupported wValue is sent by the SET_CONFIGURATION or SET_INTERFACE request, the hardware sets this bit to 1. If a problem occurs in Endpoint0 due to overrun of an automatically executed request, this bit is also set to 1. However, the E0HALT bit of the UF0E0SL register is not set to 1. 1: Respond with STALL handshake. 0: Do not respond with STALL handshake (default value). This bit is cleared to 0 and the handshake response to the bus is other than STALL when the next SETUP token is received. To set the SNDSTL bit to 1 by firmware, do not write data to the UF0E0W register. Depending on the timing of setting this bit, the STALL response is not made in time, and it may be made to the next transfer after a NAK response has been made. Setting this bit is valid only while an firmware-executed request is under execution when this bit is set to 1. It is automatically cleared to 0 when the next SETUP token is received. **Remark** The SNDSTL bit is valid only for an firmware-executed request.
0	RSUMIN	This bit outputs the Resume signal onto the USB bus. Writing this bit is invalid unless the RMWK bit of the UF0DSTL register is set to 1. 1: Generate the Resume signal. 0: Do not generate the Resume signal (default value). While this bit is set to 1, the Resume signal continues to be generated. Clear this bit to 0 by firmware after a specific time has elapsed. Because the signal is internally sampled at the clock, the operation is guaranteed only while CLK is supplied. Care must be exercised when CLK of the system is stopped.

(6) UF0 CLR request register (UF0CLR)

This register indicates the target of the received CLEAR_FEATURE request.

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

This register is meaningful only when an interrupt request is generated. Each bit is set to 1 after completion of the status stage, and automatically cleared to 0 when this register is read.

The related bits are invalid if each endpoint is not supported by the setting of the UF0EnIM register (n = 1, 2, 7) and the current setting of the interface.

	7	6	5	4	3	2	1	0	Address	After reset
UF0CLR	0	CLREP7	0	0	CLREP2	CLREP1	CLREP0	CLRDEV	4014H	00H
	0	R	0	0	R	R	R	R		

Bit position	Bit name	Function
6, 3 to 1	CLREPn	These bits indicate that a CLEAR_FEATURE Endpoint n request is received and automatically processed. 1: Automatically processed 0: Not automatically processed (default value)
0	CLRDEV	This bit indicates that a CLEAR_FEATURE Device request is received and automatically processed. 1: Automatically processed 0: Not automatically processed (default value)

Remark n = 0 to 2, 7

(7) UF0 SET request register (UF0SET)

This register indicates the target of the automatically processed SET_XXXX (except SET_INTERFACE) request.

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

This register is meaningful only when an interrupt request is generated. Each bit is set to 1 after completion of the status stage, and automatically cleared to 0 when this register is read.

	7	6	5	4	3	2	1	0	Address	After reset
UF0SET	SETCON	0	0	0	0	SETEP	0	SETDEV	4018H	00H
	R	0	0	0	0	R	0	R		

Bit position	Bit name	Function
7	SETCON	This bit indicates that a SET_CONFIGURATION request is received and automatically processed. 1: Automatically processed 0: Not automatically processed (default value)
2	SETEP	This bit indicates that a SET_FEATURE Endpoint n request (n = 0 to 2, 7) is received and automatically processed. 1: Automatically processed 0: Not automatically processed (default value)
0	SETDEV	This bit indicates that a SET_FEATURE Device request is received and automatically processed. 1: Automatically processed 0: Not automatically processed (default value)

(8) UF0 EP status 0 register (UF0EPS0)

This register indicates the USB bus status and the presence or absence of register data.

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

The related bits are invalid if each endpoint is not supported by the setting of the UF0EnIM register (n = 1, 2, 7) and the current setting of the interface.

It takes five USB clocks to reflect the status on this register after the UF0FIC0 and UF0FIC1 registers have been set. If it is necessary to read the status correctly, therefore, separate writing to the UF0FIC0 and UF0FIC1 registers from reading from the UF0EPS0, UF0EPS1, UF0EPS2, UF0E0N, and UF0EN registers by at least four USB clocks.

(1/2)

	7	6	5	4	3	2	1	0	Address	After reset
UF0EPS0	0	IT1	0	BKOUT1	0	BKIN1	EP0W	EP0R	401CH	00H
	0	R	0	R	0	R	R	R		

Bit position	Bit name	Function
6	IT1	These bits indicate that data is in the UF0INT1 register (FIFO). By setting the IT1DED bit of the UF0DEND register to 1, the status in which data is in the UF0INT1 register can be created even if data is not written to the register (Null data transmission). As soon as the IT1DED bit of the UF0DEND register is set to 1 even when the counter of the UF0INT1 register is 0, this bit is set to 1 by hardware. It is cleared to 0 after correct transmission. 1: Data is in the register. 0: No data is in the register (default value).
4	BKOUT1	These bits indicate that data is in the UF0BO1 register (FIFO) connected to the CPU side. When the FIFO configuring the UF0BO1 register is toggled, this bit is automatically set to 1 by hardware. It is automatically cleared to 0 by hardware when reading the UF0BO1 register (FIFO) connected to the CPU side has been completed (counter value = 0). It is not set to 1 when Null data is received (toggling the FIFO does not take place either). 1: Data is in the register. 0: No data is in the register (default value).
2	BKIN1	These bits indicate that data is in the UF0BI1 register (FIFO) connected to the CPU side. By setting the BKI1DED bit of the UF0DEND register to 1, the status in which data is in the UF0BI1 register can be created even if data is not written to the register (Null data transmission). As soon as the BKI1DED bit of the UF0DEND register has been set to 1 while the counter of the UF0BI1 register is 0, this bit is set to 1 by hardware. It is cleared to 0 when a toggle operation is performed. 1: Data is in the register. 0: No data is in the register (default value).

(2/2)

Bit position	Bit name	Function
1	EP0W	This bit indicates that data is in the UF0E0W register (FIFO). By setting the E0DED bit of the UF0DEND register to 1, the status in which data is in the UF0E0W register can be created even if data is not written to the register (Null data transmission). As soon as the E0DED bit of the UF0DEND register is set to 1 even when the counter of the UF0E0W register is 0, this bit is set to 1 by hardware. It is cleared to 0 after correct transmission. 1: Data is in the register. 0: No data is in the register (default value).
0	EP0R	This bit indicates that data is in the UF0E0R register (FIFO). It is automatically cleared to 0 by hardware when reading the UF0E0R register (FIFO) has been completed (counter value = 0). It is not set to 1 if Null data is received. 1: Data is in the register. 0: No data is in the register (default value).

(9) UF0 EP status 1 register (UF0EPS1)

This register indicates the USB bus status.

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

	7	6	5	4	3	2	1	0	Address	After reset
UF0EPS1	RSUM	0	0	0	0	0	0	0	4020H	00H
	R	0	0	0	0	0	0	0		

Bit position	Bit name	Function
7	RSUM	This bit indicates that the USB bus is in the Resume or Suspend status. This bit is meaningful only when an interrupt request is generated. 1: Suspend status 0: Resume status (default value) Because sampling is internally performed with the clock, the operation is guaranteed only when CLK is supplied. Care must be exercised when CLK of the system is controlled. The INTUSBF3 signal of SIE operates even when CLK is stopped. This bit is automatically cleared to 0 when it is read.

(10) UF0 EP status 2 register (UF0EPS2)

This register indicates the USB bus status.

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

The related bits are invalid if each endpoint is not supported by the setting of the UF0EnIM register (n = 1, 2, 7) and the current setting of the interface.

	7	6	5	4	3	2	1	0	Address	After reset
UF0EPS2	0	0	HALT7	0	0	HALT2	HALT1	HALT0	4024H	00H
	0	0	R	0	0	R	R	R		

Bit position	Bit name	Function
5, 2 to 0	HALTn	These bits indicate that Endpoint n is currently stalled. They are set to 1 when a stall condition, such as occurrence of an overrun and reception of an undefined request, is satisfied. These bits are automatically set to 1 by hardware. 1: Endpoint is stalled. 0: Endpoint is not stalled (default value). The SNDSTL bit is set to 1 as soon as the HALT0 bit has been set to 1 as a result of occurrence of an overrun or reception of an undefined request. If the next SETUP token is received in this status, the SNDSTL bit is cleared to 0 and, therefore, the HALT0 bit is also cleared to 0. If Endpoint0 is stalled by the SET_FEATURE Endpoint0 request, this bit is not cleared to 0 until the CLEAR_FEATURE Endpoint0 request is received or Halt Feature is cleared by firmware. If the GET_STATUS Endpoint0, CLEAR_FEATURE Endpoint0, or SET_FEATURE Endpoint0 request is received, or if a request to be processed by firmware is received due to the CPUDEC interrupt request, the HALT0 bit is masked and cleared to 0, until the next SETUP token is received. The HALTn bit is not cleared to 0 until Endpoint n receives the CLEAR_FEATURE Endpoint request, Halt Feature is cleared by the SET_INTERFACE or SET_CONFIGURATION request to the interface to which the endpoint is linked, or Halt Feature is cleared by firmware. When the SET_INTERFACE or SET_CONFIGURATION request is correctly processed, the Halt Feature of all the target endpoints, except Endpoint0, is cleared after the request has been processed, even if the wValue is the same as the currently set value, and these bits are also cleared to 0. Halt Feature of Endpoint0 cannot be cleared if it is set because the STALL response is made in response to the SET_INTERFACE and SET_CONFIGURATION requests.

Remark n = 0 to 2, 7

(11) UF0 INT status 0 register (UF0IS0)

This register indicates the interrupt source. If the contents of this register are changed, the INTUSBF0 signal becomes active and is reported as an interrupt.

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

If an interrupt request (INTUSBF0) is generated from USBF, the firmware must read this register to identify the interrupt source.

Each bit of this register is forcibly cleared to 0 when 0 is written to the corresponding bit of the UF0IC0 register.

Caution In the USBF, multiple interrupt sources are ORed internally and are issued as a interrupt request (INTUSBF0).

If only one interrupt source is cleared (0), the INTUSBF0 interrupt request in PFESiP/V850EP1 may remain set (1) and new interrupts may not be started, since other interrupt sources will still be remaining.

In this case, after performing clear (0) processing for each interrupt request with the INTUSBF0 interrupt servicing routine, confirm the flag status for the UF0IS0, UF0IS1, UF0IS2, and UF0IS3 registers again, and if there are any interrupt sources with flags set to 1, perform flag clearing (only the applicable bits need to be cleared (do not perform a batch clearing)).

(1/2)

	7	6	5	4	3	2	1	0	Address	After reset
UF0IS0	BUSRST	RSUSPD	VBSOF	SHORT	DMAED	SETRQ	CLRRQ	EPHALT	4040H	00H
	R	R	R	R	R	R	R	R		

Bit position	Bit name	Function
7	BUSRST	This bit indicates that Bus Reset has occurred. 1: Bus Reset has occurred (interrupt request is generated). 0: Not Bus Reset status (default value)
6	RSUSPD	This bit indicates that the Resume or Suspend status has occurred. Reference bit 7 of the UF0EPS1 register by firmware. 1: Resume or Suspend status has occurred (interrupt request is generated). 0: Resume or Suspend status has not occurred (default value).
5	VBSOF	This bit indicates that the VBUS Off has occurred. 1: VBUS Off has occurred (interrupt request is generated). 0: VBUS Off has not occurred (default value).
<R> 4	SHORT	This bit indicates that data is read from the FIFO of the UF0BO1 register and that the INTUSBF2 signal is active. It is valid only when the FIFO is not full in the DMA mode. 1: Received a short packet/INTUSBF2 signal is active (interrupt request is generated). 0: Did not receive a short packet/INTUSBF2 signal is not active (default value). Please confirm whether it is end operation for which Endpoint and that the DMA transfer end by the short packet happened by UF0DMS1 register by all means. At the time of next short packet transfer, this bit is not set unless UF0DMS1 register was read. But, this bit is not cleared (0) automatically even if UF0DMS1 register is read by firmware.

(2/2)

Bit position	Bit name	Function
3	DMAED	This bit indicates that DMA has been completed 1: DMA has been completed (interrupt request is generated). 0: DMA has not been completed (default value). Please confirm whether it is end operation for which Endpoint and that the DMA transfer end happened by UF0DMS1 register by all means. At the time of the next DMA transfer, this bit is not set unless UF0DMS1 register was read. But, this bit is not cleared (0) automatically even if UF0DMS1 register is read by firmware.
2	SETRQ	This bit indicates that the SET_XXXX request to be automatically processed has been received and automatically processed (XXXX = CONFIGURATION or FEATURE). 1: SET_XXXX request to be automatically processed has been received (interrupt request is generated). 0: SET_XXXX request to be automatically processed has not been received (default value). This bit is set to 1 after completion of the status stage. Reference the UF0SET register to identify what is the target of the request. This bit is not automatically cleared to 0 even if the UF0SET register is read by firmware. The EPHALT bit is also set to 1 when the SET_FEATURE Endpoint request has been received.
1	CLRRQ	This bit indicates that the CLEAR_FEATURE request has been received and automatically processed. 1: CLEAR_FEATURE request has been received (interrupt request is generated). 0: CLEAR_FEATURE request has not been received (default value). This bit is set to 1 after completion of the status stage. Reference the UF0CLR register to identify what is the target of the request. This bit is not automatically cleared to 0 even if the UF0CLR register is read by firmware.
0	EPHALT	This bit indicates that an endpoint has stalled. 1: Endpoint has stalled (interrupt request is generated). 0: Endpoint has not stalled (default value). This bit is also set to 1 when an endpoint has stalled by setting firmware. Identify the endpoint that has stalled, by referencing the UF0EPS2 register. This bit is not automatically cleared to 0 even when the CLEAR_FEATURE Endpoint, SET_INTERFACE, or SET_CONFIGURATION request is received. It is not automatically cleared to 0, either, if the next SETUP token is received in case of overrun of Endpoint0. **Caution Even if the EnHALT bit of the UF0EnSL register is set to 1 by firmware and this interrupt request is generated, HALT0 of the UF0EPS2 register is masked and cleared to 0 between when a SET_FEATURE Endpoint0, CLEAR_FEATURE Endpoint0, or GET_STATUS Endpoint0 request, or firmware-processed request is received and when a SETUP token is received (other than the above).**

(12) UF0 INT status 1 register (UF0IS1)

This register indicates the interrupt source. If the contents of this register are changed, the INTUSBF0 signal becomes active and is reported as an interrupt.

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

If an interrupt request (INTUSBF0) is generated from USBF, the firmware must read this register to identify the interrupt source.

Each bit of this register is forcibly cleared to 0 when 0 is written to the corresponding bit of the UF0IC0 register.

Caution In the USBF, multiple interrupt sources are ORed internally and are issued as a interrupt request (INTUSBF0).

If only one interrupt source is cleared (0), the INTUSBF0 interrupt request in PFESiP/V850EP1 may remain set (1) and new interrupts may not be started, since other interrupt sources will still be remaining.

In this case, after performing clear (0) processing for each interrupt request with the INTUSBF0 interrupt servicing routine, confirm the flag status for the UF0IS0, UF0IS1, UF0IS2, and UF0IS3 registers again, and if there are any interrupt sources with flags set to 1, perform flag clearing (only the applicable bits need to be cleared (do not perform a batch clearing)).

	7	6	5	4	3	2	1	0	Address	After reset
UF0IS1	0	E0IN	E0INDT	E0ODT	SUCES	STG	PROT	CPUDEC	4044H	00H
	0	R	R	R	R	R	R	R		

Bit position	Bit name	Function
6	E0IN	This bit indicates that an IN token for Endpoint0 has been received and that the hardware has automatically transmitted NAK. 1: IN token is received and NAK is transmitted (interrupt request is generated). 0: IN token is not received (default value).
5	E0INDT	This bit indicates that data has been correctly transmitted from the UF0E0W register. 1: Transmission from UF0E0W register is completed (interrupt request is generated). 0: Transmission from UF0E0W register is not completed (default value). Data is transmitted in synchronization with the IN token next to the one that set the EP0NKW bit of the UF0E0N register to 1. This bit is automatically set to 1 by hardware when the host correctly receives that data. It is also set to 1 even if the data is a Null packet. This bit is automatically cleared to 0 by hardware when the first write access is made to the UF0E0W register.

(2/2)

Bit position	Bit name	Function
4	E0ODT	This bit indicates that data has been correctly received in the UF0E0R register. 1: Data is in UF0E0R register (interrupt request is generated). 0: Data is not in UF0E0R register (default value). This bit is automatically set to 1 by hardware when data has been correctly received. At the same time, the EP0R bit of the UF0EPS0 register is also set to 1. If a Null packet has been received, this bit is not set to 1. It is automatically cleared to 0 by hardware when the firmware reads the UF0E0R register and the value of the UF0E0L register becomes 0.
3	SUCES	This bit indicates that either a firmware-processed or hardware-processed request has been received and that the status stage has been correctly completed. 1: Control transfer has been correctly processed (interrupt request is generated). 0: Control transfer has not been processed correctly (default value). This bit is set to 1 upon completion of the status stage. It is automatically cleared to 0 by hardware when the next SETUP token is received. This bit is also set to 1 when data with Data PID of 0 (Null data) is received in the status stage of control transfer.
2	STG	This bit is set to 1 when the stage of control transfer has changed to the status stage. It is valid for both firmware-processed and hardware-processed requests. This bit is also set to 1 when the stage of control transfer (without data) has changed to the status stage. 1: Status stage (interrupt request is generated) 0: Not status stage (default value) This bit is automatically cleared to 0 by hardware when the next SETUP token is received. It is also set to 1 when the stage of control transfer has changed to the status stage while ACK cannot be correctly received in the data stage. In this case, the EP0NKW bit of the UF0E0N register is also cleared to 0 as soon as the UF0E0W register has been cleared, if the firmware is processing control transfer (read).
1	PROT	This bit indicates that a SETUP token has been received. It is valid for both firmware-processed and hardware-processed requests. 1: SETUP token is correctly received (interrupt request is generated). 0: SETUP token is not received (default value). This bit is set to 1 when data has been correctly received in the UF0E0ST register. Clear this bit to 0 by firmware when the first read access is made to the UF0E0ST register. If it is not cleared to 0 by firmware, reception of the next SETUP token cannot be correctly recognized. This bit is used to accurately recognize that a SETUP transaction has been executed again during control transfer. If the SETUP transaction is re-executed during control transfer and if a second request is executed by hardware, the CPUDEC bit is not set to 1, but the PROT bit can be used for recognition of the re-execution.
0	CPUDEC	This bit indicates that the UF0E0ST register has a request that is to be decoded by firmware. 1: Firmware processed request is in UF0E0ST register (interrupt request is generated). 0: Firmware processed request is not in UF0E0ST register (default value). This bit is automatically cleared to 0 by hardware when all the data of the UF0E0ST register is read.

(13) UF0 INT status 2 register (UF0IS2)

This register indicates the interrupt source. If the contents of this register are changed, the INTUSBF0 signal becomes active and is reported as an interrupt.

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

If an interrupt request (INTUSBF0) is generated from USBF, the firmware must read this register to identify the interrupt source.

Each bit of this register is forcibly cleared to 0 when 0 is written to the corresponding bit of the UF0IC2 register.

The related bits are invalid if each endpoint is not supported by the setting of the UF0EnIM register (n = 1, 7) and the current setting of the interface.

Caution In the USBF, multiple interrupt sources are ORed internally and are issued as a interrupt request (INTUSBF0).

If only one interrupt source is cleared (0), the INTUSBF0 interrupt request in PFESiP/V850EP1 may remain set (1) and new interrupts may not be started, since other interrupt sources will still be remaining.

In this case, after performing clear (0) processing for each interrupt request with the INTUSBF0 interrupt servicing routine, confirm the flag status for the UF0IS0, UF0IS1, UF0IS2, and UF0IS3 registers again, and if there are any interrupt sources with flags set to 1, perform flag clearing (only the applicable bits need to be cleared (do not perform a batch clearing)).

	7	6	5	4	3	2	1	0	Address	After reset
UF0IS2	0	0	BKI1IN	BKI1DT	0	0	0	IT1DT	4048H	00H
	0	0	R	R	0	0	0	R		

Bit position	Bit name	Function
5	BKI1IN	These bits indicate that an IN token has been received in the UF0BI1 register (Endpoint 1) and that NAK has been returned. 1: IN token is received and NAK is transmitted (interrupt request is generated). 0: IN token is not received (default value).
4	BKI1DT	These bits indicate that the FIFO of the UF0BI1 register (Endpoint 1) has been toggled. This means that data can be written to Endpoint 1. 1: FIFO has been toggled (interrupt request is generated). 0: FIFO has not been toggled (default value). The data written to Endpoint 1 is transmitted in synchronization with the IN token next to the one that set the BKI1NK bit of the UF0EN register to 1. When the FIFO has been toggled and then data can be written from the CPU, this bit is automatically set to 1 by hardware. It is also set to 1 when the FIFO has been toggled, even if the data is a Null packet. This bit is automatically cleared to 0 by hardware when the first write access is made to the UF0BI1 register.
0	IT1DT	These bits indicate that data has been correctly received from the UF0INT1 register (Endpoint 7). 1: Transmission is completed (interrupt request is generated). 0: Transmission is not completed (default value). Data is transmitted in synchronization with the IN token next to the one that set the IT1NK bit of the UF0EN register to 1. This bit is automatically set to 1 by hardware when the host has correctly received that data. It is automatically cleared to 0 by hardware when the first write access is made to the UF0INT1 register. This bit is also set to 1 even when the data is a Null packet.

(14) UF0 INT status 3 register (UF0IS3)

This register indicates the interrupt source. If the contents of this register are changed, the INTUSBF0 signal becomes active and is reported as an interrupt.

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

If an interrupt request (INTUSBF0) is generated from USBF, the firmware must read this register to identify the interrupt source.

Each bit of this register is forcibly cleared to 0 when 0 is written to the corresponding bit of the UF0IC3 register.

The related bits are invalid if each endpoint is not supported by the setting of the UF0EnIM register (n = 2) and the current setting of the interface.

Caution In the USBF, multiple interrupt sources are ORed internally and are issued as a interrupt request (INTUSBF0).

If only one interrupt source is cleared (0), the INTUSBF0 interrupt request in PFESiP/V850EP1 may remain set (1) and new interrupts may not be started, since other interrupt sources will still be remaining.

In this case, after performing clear (0) processing for each interrupt request with the INTUSBF0 interrupt servicing routine, confirm the flag status for the UF0IS0, UF0IS1, UF0IS2, and UF0IS3 registers again, and if there are any interrupt sources with flags set to 1, perform flag clearing (only the applicable bits need to be cleared (do not perform a batch clearing)).

	7	6	5	4	3	2	1	0	Address	After reset
UF0IS3	0	0	0	0	BKO1FL	BKO1NL	BKO1NAK	BKO1DT	404CH	00H
	0	0	0	0	R	R	R	R		

Bit position	Bit name	Function
3	BKO1FL	These bits indicate that data has been correctly received in the UF0BO1 register (Endpoint 2) and that both the FIFOs of the CPU and SIE hold the data. 1: Received data is in both the FIFOs of the UF0BO1 register (interrupt request is generated). 0: Received data is not in the FIFO on the SIE side of the UF0BO1 register (default value). If data is held in both the FIFOs of the CPU and SIE, these bits are automatically set to 1 by hardware. They are automatically cleared to 0 by hardware when the FIFO is toggled.
2	BKO1NL	These bits indicate that a Null packet (packet with a length of 0) has been received in the UF0BO1 register (Endpoint 2). 1: Null packet is received (interrupt request is generated). 0: Null packet is not received (default value). These bits are set to 1 immediately after reception of a Null packet when the FIFO is empty. They are set to 1 when the FIFO on the CPU side has been completely read if data is in that FIFO.
1	BKO1NAK	These bits indicate that an OUT token has been received to the UF0BO1 register (Endpoint 2) and that NAK has been returned. 1: OUT token is received and NAK is transmitted (interrupt request is generated). 0: OUT token is not received (default value).

(2/2)

Bit position	Bit name	Function
0	BKO1DT	These bits indicate that data has been correctly received in the UF0BO1 register (Endpoint 2). 1: Reception has been completed correctly (interrupt request is generated). 0: Reception has not been completed (default value). These bits are automatically set to 1 by hardware when data has been correctly received and the FIFO has been toggled. At the same time, BKOUT1 of the UF0EPS0 register are also set to 1. They are not set to 1 when the data is a Null packet. These bits are automatically cleared to 0 by hardware when the value of the UF0BO1L register becomes 0 as a result of reading the UF0BO1 register by firmware. These bits are automatically cleared to 0 when all the contents of the FIFO on the CPU side have been read. However, the interrupt request is not cleared if data is in the FIFO on the SIE side at this time, and the INTUSBF0 signal does not become inactive. The signal is kept active if data is successively received.

(15) UF0 INT status 4 register (UF0IS4)

This register indicates that the SET_INTERFACE request has been received and automatic processing has been performed. The interrupt report does not occur by the change in this register.

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

Each bit of this register is forcibly cleared (0) when 0 is written to the corresponding bit of the UF0IC4 register.

The related bits are invalid if each endpoint is not supported by the setting of the UF0EnIM register (n = 1, 2, 7) and the current setting of the interface.

	7	6	5	4	3	2	1	0	Address	After reset
UF0IS4	0	0	SETINT	0	0	0	0	0	4050H	00H
	0	0	R	0	0	0	0	0		

Bit position	Bit name	Function
5	SETINT	This bit indicates that the SET_INTERFACE request has been received and automatically processed. 1: The request has been automatically processed (interrupt request is generated). 0: The request has not been automatically processed (default value). The current setting of this bit can be identified by reading the UF0ASS or UF0IFn register (n = 0 to 4).

(16) UF0 INT mask 0 register (UF0IM0)

This register controls masking of the interrupt sources indicated by the UF0IS0 register.

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

Firmware can mask occurrence of an interrupt request from USBF (INTUSBF0) by writing 1 to the corresponding bit of this register.

	7	6	5	4	3	2	1	0	Address	After reset
UF0IM0	BUSRSTM	RSUSPDM	VBSOFM	SHORTM	DMAEDM	SETRQM	CLRRQM	EPHALTM	405CH	00H
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

Bit position	Bit name	Function
7	BUSRSTM	This bit masks the Bus Reset interrupt. 1: Mask 0: Do not mask (default value)
6	RSUSPDM	This bit masks the Resume/Suspend interrupt. 1: Mask 0: Do not mask (default value)
5	VBSOFM	This bit masks the VBSOFM interrupt. 1: Mask 0: Do not mask (default value)
4	SHORTM	This bit masks the Short interrupt. 1: Mask 0: Do not mask (default value)
3	DMAEDM	This bit masks the DMAED interrupt. 1: Mask 0: Do not mask (default value)
2	SETRQM	This bit masks the SETRQ interrupt. 1: Mask 0: Do not mask (default value)
1	CLRRQM	This bit masks the CLRRQ interrupt. 1: Mask 0: Do not mask (default value)
0	EPHALTM	This bit masks the EP_Halt interrupt. 1: Mask 0: Do not mask (default value)

(17) UF0 INT mask 1 register (UF0IM1)

This register controls masking of the interrupt sources indicated by the UF0IS1 register.

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

Firmware can mask occurrence of an interrupt request from USBF (INTUSBF0) by writing 1 to the corresponding bit of this register.

	7	6	5	4	3	2	1	0	Address	After reset
UF0IM1	0	E0INM	E0INDTM	E0ODTM	SUCESM	STGM	PROTM	CPUDECM	4060H	00H
	0	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

Bit position	Bit name	Function
6	E0INM	This bit masks the E0IN interrupt. 1: Mask 0: Do not mask (default value)
5	E0INDTM	This bit masks the E0INDT interrupt. 1: Mask 0: Do not mask (default value)
4	E0ODTM	This bit masks the E0ODT interrupt. 1: Mask 0: Do not mask (default value)
3	SUCESM	This bit masks the Success interrupt. 1: Mask 0: Do not mask (default value)
2	STGM	This bit masks the Stg interrupt. 1: Mask 0: Do not mask (default value)
1	PROTM	This bit masks the Protect interrupt. 1: Mask 0: Do not mask (default value)
0	CPUDECM	This bit masks the CPUDEC interrupt. 1: Mask 0: Do not mask (default value)

(18) UF0 INT mask 2 register (UF0IM2)

This register controls masking of the interrupt sources indicated by the UF0IS2 register.

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

Firmware can mask occurrence of an interrupt request from USBF (INTUSBF0) by writing 1 to the corresponding bit of this register.

The related bits are invalid if each endpoint is not supported by the setting of the UF0EnIM register (n = 1, 7) and the current setting of the interface.

Be sure to clear bits 7, 6, and 3 to 1 to “0”. If it is set to “1”, the operation is not guaranteed.

	7	6	5	4	3	2	1	0	Address	After reset
UF0IM2	0	0	BKI1INM	BKI1DTM	0	0	0	IT1DTM	4064H	00H
	0	0	R/W	R/W	0	0	0	R/W		

Bit position	Bit name	Function
5	BKI1INM	These bits mask the BKI1IN interrupt. 1: Mask 0: Do not mask (default value)
4	BKI1DTM	These bits mask the BKI1DT interrupt. 1: Mask 0: Do not mask (default value)
0	IT1DTM	These bits mask the IT1DT interrupt. 1: Mask 0: Do not mask (default value)

(19) UF0 INT mask 3 register (UF0IM3)

This register controls masking of the interrupt sources indicated by the UF0IS3 register.

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

Firmware can mask occurrence of an interrupt request from USBF (INTUSBF0) by writing 1 to the corresponding bit of this register.

The related bits are invalid if each endpoint is not supported by the setting of the UF0EnIM register (n = 2) and the current setting of the interface.

Be sure to clear bits 7 to 4 to “0”. If it is set to “1”, the operation is not guaranteed.

	7	6	5	4	3	2	1	0	Address	After reset
UF0IM3	0	0	0	0	BKO1FLM	BKO1NLM	BKO1NAKM	BKO1DTM	4068H	00H
	0	0	0	0	R/W	R/W	R/W	R/W		

Bit position	Bit name	Function
3	BKO1FLM	These bits mask the BKO1FL interrupt. 1: Mask 0: Do not mask (default value)
2	BKO1NLM	These bits mask the BKO1NL interrupt. 1: Mask 0: Do not mask (default value)
1	BKO1NAKM	These bits mask the BKO1NK interrupt. 1: Mask 0: Do not mask (default value)
0	BKO1DTM	These bits mask the BKO1DT interrupt. 1: Mask 0: Do not mask (default value)

(20) UF0 INT mask 4 register (UF0IM4)

This register controls masking the SETINT indicated by the UF0IS4 register.

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

Firmware can mask of the SETINT by writing 1 to the corresponding bit of this register.

The related bits are invalid if each endpoint is not supported by the setting of the UF0EnIM register (n = 1, 2, 7) and the current setting of the interface.

	7	6	5	4	3	2	1	0	Address	After reset
UF0IM4	0	0	SETINTM	0	0	0	0	0	406CH	00H
	0	0	R/W	0	0	0	0	0		

Bit position	Bit name	Function
5	SETINTM	This bit masks the SET_INT. 1: Mask 0: Do not mask (default value)

(21) UF0 INT clear 0 register (UF0IC0)

This register controls clearing the interrupt sources indicated by the UF0IS0 register.

This register is write-only, in 8-bit units. If this register is read, the value FFH is read.

Firmware can clear an interrupt source by writing 0 to the corresponding bit of this register. Even a bit that is automatically cleared to 0 by hardware can be cleared by firmware before it is cleared by hardware. Writing 0 to a bit of this register automatically sets the bit to 1. Writing 1 is invalid.

	7	6	5	4	3	2	1	0	Address	After reset
UF0IC0	BUSRSTC	RSUSPDC	VBSOFC	SHORTC	DMAEDC	SETRQC	CLRRQC	EPHALTC	4078H	FFH
	W	W	W	W	W	W	W	W		

Bit position	Bit name	Function
7	BUSRSTC	This bit clears the Bus Reset interrupt. 0: Clear
6	RSUSPDC	This bit clears the Resume/Suspend interrupt. 0: Clear
5	VBSOFC	This bit clears the VBSOFC interrupt. 0: Clear
4	SHORTC	This bit clears the Short interrupt. 0: Clear
3	DMAEDC	This bit clears the DMAED interrupt. 0: Clear
2	SETRQC	This bit clears the SETRQ interrupt. 0: Clear
1	CLRRQC	This bit clears the CLRRQ interrupt. 0: Clear
0	EPHALTC	This bit clears the EP_Halt interrupt. 0: Clear

(22) UF0 INT clear 1 register (UF0IC1)

This register controls clearing the interrupt sources indicated by the UF0IS1 register.

This register is write-only, in 8-bit units. If this register is read, the value FFH is read.

Firmware can clear an interrupt source by writing 0 to the corresponding bit of this register. Even a bit that is automatically cleared to 0 by hardware can be cleared by firmware before it is cleared by hardware. Writing 0 to a bit of this register automatically sets the bit to 1. Writing 1 is invalid.

	7	6	5	4	3	2	1	0	Address	After reset
UF0IC1	1	E0INC	E0INDTC	E0ODTC	SUCESC	STGC	PROTC	CPUDECC	407CH	FFH
	1	W	W	W	W	W	W	W		

Bit position	Bit name	Function
6	E0INC	This bit clears the E0IN interrupt. 0: Clear
5	E0INDTC	This bit clears the E0INDT interrupt. 0: Clear
4	E0ODTC	This bit clears the E0ODT interrupt. 0: Clear
3	SUCESC	This bit clears the Success interrupt. 0: Clear
2	STGC	This bit clears the Stg interrupt. 0: Clear
1	PROTC	This bit clears the Protect interrupt. 0: Clear
0	CPUDECC	This bit clears the CPUDEC interrupt. 0: Clear

(23) UF0 INT clear 2 register (UF0IC2)

This register controls clearing the interrupt sources indicated by the UF0IS2 register.

This register is write-only, in 8-bit units. If this register is read, the value FFH is read.

Firmware can clear an interrupt source by writing 0 to the corresponding bit of this register. Even a bit that is automatically cleared to 0 by hardware can be cleared by firmware before it is cleared by hardware. Writing 0 to a bit of this register automatically sets the bit to 1. Writing 1 is invalid.

The related bits are invalid if each endpoint is not supported by the setting of the UF0EnIM register (n = 1, 7) and the current setting of the interface.

	7	6	5	4	3	2	1	0	Address	After reset
UF0IC2	1	1	BKI1INC	BKI1DTC	1	1	1	IT1DTC	4080H	FFH
	1	1	W	W	1	1	1	W		

Bit position	Bit name	Function
5	BKI1INC	These bits clear the BKI1IN interrupt. 0: Clear
4	BKI1DTC	These bits clear the BKI1DT interrupt. 0: Clear
0	IT1DTC	These bits clear the IT1DT interrupt. 0: Clear

Remark n = 1, 2

(24) UF0 INT clear 3 register (UF0IC3)

This register controls clearing the interrupt sources indicated by the UF0IS3 register.

This register is write-only, in 8-bit units. If this register is read, the value FFH is read.

Firmware can clear an interrupt source by writing 0 to the corresponding bit of this register. Even a bit that is automatically cleared to 0 by hardware can be cleared by firmware before it is cleared by hardware. Writing 0 to a bit of this register automatically sets the bit to 1. Writing 1 is invalid.

The related bits are invalid if each endpoint is not supported by the setting of the UF0EnIM register (n = 2) and the current setting of the interface.

	7	6	5	4	3	2	1	0	Address	After reset
UF0IC3	1	1	1	1	BKO1FLC	BKO1NLC	BKO1NAKC	BKO1DTC	4084H	FFH
	1	1	1	1	W	W	W	W		

Bit position	Bit name	Function
3	BKO1FLC	These bits clear the BKO1FL interrupt. 0: Clear
2	BKO1NLC	These bits clear the BKO1NL interrupt. 0: Clear
1	BKO1NAKC	These bits clear the BKO1NK interrupt. 0: Clear
0	BKO1DTC	These bits clear the BKO1DT interrupt. 0: Clear

Remark n = 1, 2

(25) UF0 INT clear 4 register (UF0IC4)

This register controls clearing the SETINT indicated by the UF0IS4 register.

This register is write-only, in 8-bit units. If this register is read, the value FFH is read.

Firmware can clear an interrupt source by writing 0 to the corresponding bit of this register. Even a bit that is automatically cleared to 0 by hardware can be cleared by firmware before it is cleared by hardware. Writing 0 to a bit of this register automatically sets the bit to 1. Writing 1 is invalid.

The related bits are invalid if each endpoint is not supported by the setting of the UF0EnIM register (n = 1, 2, 7) and the current setting of the interface.

	7	6	5	4	3	2	1	0	Address	After reset
UF0IC4	1	1	SETINTC	1	1	1	1	1	4088H	FFH
	1	1	W	1	1	1	1	1		

Bit position	Bit name	Function
5	SETINTC	This bit clears the SETINT. 0: Clear

(26) UF0 INT & DMARQ register (UF0IDR)

This register is used to select the bulk transfer operation mode.

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

If data exists in either the UF0BO1 or UF0BO1 register, or if data can be written to the UF0BI1 or UF0BI2 register, this register selects whether it is reported to the firmware by an interrupt request, or whether starting DMA is requested. If starting DMA is requested, the DMA transfer mode can be selected according to the setting of bits 0 and 1.

The related bits are invalid if each endpoint is not supported by the setting of the UF0EnIM register (n = 1, 2) and the current setting of the interface.

Be sure to clear bits 7, 5, 3, and 2 to “0”. If they are set to “1”, the operation is not guaranteed.

Caution If the target endpoint is not supported by the SET_INTERFACE request under DMA transfer, the DMA request signal becomes inactive immediately, and the corresponding bit is automatically cleared to 0 by hardware.

(1/2)

	7	6	5	4	3	2	1	0	Address	After reset
UF0IDR	0	DQBI1MS	0	DQBO1MS	0	0	MODE1	MODE0	4098H	00H
	0	R/W	0	R/W	0	0	R/W	R/W		

Bit position	Bit name	Function
6	DQBI1MS	This bit performs mode control to determine whether the bulk in transfer operation at Endpoint 1 is performed in PIO mode or DMA mode. When in PIO mode, the INTUSBF0 signal sets BKI1DT interrupts (UF0IS2 register) as valid. When in DMA mode, it sets DMAC interface signals (UDMS register of CPU function) as valid. This bit is automatically cleared (0) by hardware when the value of the EP1x_DCR2 register becomes 0 during a DMA operation, and the operation returns to PIO mode. To continue DMA transfer, use firmware to re-set (1). 1: Sets Endpoint 1 to DMA mode. 0: Sets Endpoint 1 to PIO mode (default value).
4	DQBO1MS	This bit performs mode control to determine whether the bulk out transfer operation at Endpoint 2 is performed in PIO mode or DMA mode. When in PIO mode, the INTUSBF0 signal sets BKO1DT interrupts (UF0IS3 register) as valid. When in DMA mode, it sets DMAC interface signals (UDMS register of CPU function) as valid. This bit is automatically cleared (0) by hardware when the value of the EP2x_DCR2 register becomes 0 during a DMA operation, and the operation returns to PIO mode. To continue DMA transfer, use firmware to re-set (1). 1: Sets Endpoint 2 to DMA mode. 0: Sets Endpoint 2 to PIO mode (default value).

(2/2)

Bit position	Bit name	Function
1, 0	MODE1, MODE0	These bits select the DMA transfer mode.

MODE1	MODE2	Mode	Remark
1	1	Setting prohibited	Operation cannot be guaranteed.
1	0	Demand mode	DMA request signal becomes active as long as there is data. It becomes inactive if there is no more data.
0	X	Setting prohibited	Operation cannot be guaranteed.

Remarks 1. X: Don't care

2. PFESiP/V850EP1 supports only the demand mode.

(27) UF0 DMA status 0 register (UF0DMS0)

This register indicates the DMA status of Endpoint1 and Endpoint2.

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

The related bits are invalid if each endpoint is not supported by the setting of the UF0EnIM register (n = 1, 2) and the current setting of the interface.

	7	6	5	4	3	2	1	0	Address	After reset
UF0DMS0	0	0	0	0	DQE2	DQE1	0	0	409CH	00H
	0	0	0	0	R	R	0	0		

Bit position	Bit name	Function
3	DQE2	This bit indicates that a DMA read request is being issued from Endpoint2 to memory. 1: DMA read request from Endpoint2 is being issued. 0: DMA read request from Endpoint2 is not being issued (default value).
2	DQE1	This bit indicates that a DMA write request is being issued from memory to Endpoint1. Note that, even if data is in Endpoint1 (when the FIFO is not full and after the BKI1DED bit of the UF0DEND register has been set to 1), the DMA request signal becomes active immediately and DMA transfer is started when the DQBI1MS bit of the UF0IDR register and the EP1DMAEN bit of the EP1_DCR1 register is set to 1. 1: DMA write request for Endpoint1 is being issued. 0: DMA write request for Endpoint1 is not being issued (default value).

(28) UF0 DMA status 1 register (UF0DMS1)

This register indicates the DMA status of Endpoint1 and Endpoint2.

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

The related bits are invalid if each endpoint is not supported by the setting of the UF0EnIM register (n = 1, 2) and the current setting of the interface.

Each bit is automatically cleared to 0 when this register is read. Even when this register is read, however, bits 4 and 3 of the UF0IS0 register are not cleared to 0. If the target endpoint is no longer supported by the SET_INTERFACE request, each bit is automatically cleared to 0 by hardware.

	7	6	5	4	3	2	1	0	Address	After reset
UF0DMS1	0	0	0	DEDE2	DSPE2	DEDE1	0	0	40A0H	00H
	0	0	0	R	R	R	0	0		

Bit position	Bit name	Function
<R> 4, 2	DEDEn	These bits indicate that the EPnx_DCR2 value has become 0 and DMA has been stopped while a DMA transfer request was being issued between Endpoint n and a memory. 1: DMA end signal for Endpoint n is active. 0: DMA end signal for Endpoint n is inactive (default value). Be sure to read this register when the DMA transfer is executed. At the time of the next DMA transfer, this bit is not set unless UF0DMS1 register was read.
<R> 3	DSPEm	These bits indicate that, although a DMA read request was being issued from Endpoint m to memory, DMA has been stopped because the received data is a short packet and there is no more data to be transferred. 1: A short packet received data for Endpoint m/INTUSBF2 signal is active. 0: A short packet received data for Endpoint m/INTUSBF2 signal is inactive (default value). Be sure to read this register particularly in a short packet reception when the DMA transfer is executed. At the time of the next short packet transfer, this bit is not set unless UF0DMS1 register was read.

Remark n = 1, 2
m = 2

(29) UF0 FIFO clear 0 register (UF0FIC0)

This register clears each FIFO.

This register is write-only, in 8-bit units. If this register is read, 00H is read.

Firmware can clear the target FIFO by writing 1 to the corresponding bit of this register. The bit to which 1 has been written is automatically cleared to 0. Writing 0 to the bit is invalid.

The related bits are invalid if each endpoint is not supported by the setting of the UF0EnIM register (n = 1, 7) and the current setting of the interface.

	7	6	5	4	3	2	1	0	Address	After reset
UF0FIC0	0	0	BKI1SC	BKI1CC	0	ITR1C	EP0WC	EP0RC	40C0H	00H
	0	0	W	W	0	W	W	W		

Bit position	Bit name	Function
5	BKI1SC	These bits clear only the FIFO on the SIE side of the UF0BI1 register (reset the counter). 1: Clear Writing these bits is invalid while an IN token for Endpoint 1 is being processed with the BKI1NK bit of the UF0EN register set to 1. The BKI1NK bit is automatically cleared to 0 by clearing the FIFO. Make sure that the FIFO on the CPU side is empty when these bits are used.
4	BKI1CC	These bits clear only the FIFO on the CPU side of the UF0BI1 register (reset the counter). 1: Clear
2	ITR1C	These bits clear the UF0INT1 register (reset the counter). 1: Clear Writing these bits is invalid while an IN token for Endpoint 7 is being processed with the IT1NK bit of the UF0EN register set to 1. The IT1NK bit is automatically cleared to 0 by clearing the FIFO.
1	EP0WC	This bit clears the UF0E0W register (resets the counter). 1: Clear Writing this bit is invalid while an IN token for Endpoint0 is being processed with the EP0NKW bit of the UF0E0N register set to 1. The EP0NKW bit is automatically cleared to 0 by clearing the FIFO.
0	EP0RC	This bit clears the UF0E0R register (resets the counter). 1: Clear When the EP0NKR bit of the UF0E0N register is set to 1 (except when it has been set by firmware), the EP0NKR bit is automatically cleared to 0 by clearing the FIFO.

(30) UF0 FIFO clear 1 register (UF0FIC1)

This register clears each FIFO.

This register is write-only, in 8-bit units. If this register is read, 00H is read.

Firmware can clear the target FIFO by writing 1 to the corresponding bit of this register. The bit to which 1 has been written is automatically cleared to 0. Writing 0 to the bit is invalid.

The related bits are invalid if each endpoint is not supported by the setting of the UF0EnIM register (n = 2) and the current setting of the interface.

	7	6	5	4	3	2	1	0	Address	After reset
UF0FIC1	0	0	0	0	0	0	BKO1C	BKO1CC	40C4H	00H
	0	0	0	0	0	0	W	W		

Bit position	Bit name	Function
1	BKO1C	These bits clear the FIFOs on both the SIE and CPU sides of the UF0BO1 register (reset the counter). 1: Clear When the BKO1NK bit of the UF0EN register is set to 1 (except when it has been set by firmware), the BKO1NK bit is automatically cleared to 0 by clearing the FIFO.
0	BKO1CC	These bits clear only the FIFO on the CPU side of the UF0BO1 register (reset the counter). 1: Clear When the BKO1NK bit of the UF0EN register is set to 1 (except when it has been set by firmware), the BKO1NK bit is automatically cleared to 0 by clearing the FIFO.

(31) UF0 data end register (UF0DEND)

This register reports the end of writing to the transmission system.

This register can be accessed only in 8-bit units. Bit 6 can be read or written, and bits 3, 1, and 0 are write-only. When this register is read, 0 is read from bits other than bit 6.

Firmware can start data transfer of the target endpoint by writing 1 to the corresponding bit 3, 1, and 0 of this register. The bit to which 1 has been written is automatically cleared to 0. Writing 0 to the bit is invalid.

The related bits are invalid if each endpoint is not supported by the setting of the UF0EnIM register (n = 1, 7) and the current setting of the interface.

(1/2)

	7	6	5	4	3	2	1	0	Address	After reset
UF0DEND	0	BKI1T	0	0	IT1DEND	0	BKI1DED	E0DED	40D4H	00H
	0	R/W	0	0	W	0	W	W		

Bit position	Bit name	Function
6	BKI1T	These bits specify whether toggling the FIFO is automatically executed if the FIFO on the CPU side of the UF0BI1 register becomes full as a result of DMA. 1: Automatically execute a toggle operation of the FIFO as soon as the FIFO has become full. 0: Do not automatically execute a toggle operation of the FIFO even if the FIFO becomes full (default value).
3	IT1DEND	Set these bits to 1 to transmit the data of the UF0INT1 register. When these bits are set to 1, the IT1NK bit of the UF0EN register is set to 1 and data transfer is executed. 1: Transmit a short packet. 0: Do not transmit a short packet (default value). If the ITR1C bit of the UF0FIC0 register is set to 1 and then these bits are set to 1 (counter of UF0INT1 register = 0 and the IT1 of the UF0EPS0 register = 1), a Null packet (with a data length of 0) is transmitted. If data exists in the UF0INT1 register and if these bits are set to 1 (counter of UF0INT1 register ≠ 0 and the IT1 of the UF0EPS0 register = 1), a short packet is transmitted.
1	BKI1DED	Set these bits to 1 when writing transmit data to the UF0BI1 register has been completed. When these bits are set to 1, the FIFO is toggled as soon as possible, the BKI1NK bit of the UF0EN register is set to 1, and data is transferred. 1: Transmit a short packet. 0: Do not transmit a short packet (default value). If the BKI1CC bit of the UF0FIC0 register is set to 1 and then these bits are set to 1 (counter of UF0BI1 register = 0), a Null packet (with a data length of 0) is transmitted. If data exists in the UF0BI1 register and if these bits are set to 1 (counter of UF0BI1 register ≠ 0), and if the FIFO is not full, a short packet is transmitted. If the FIFO on the CPU side of the UF0BI1 register becomes full as a result of DMA, with the PIO or BKI1T bit set to 1, the hardware starts data transmission even if these bits are not set to 1. If the FIFO on the CPU side of the UF0BI1 register becomes full as a result of DMA, with the BKI1T bit cleared to 0, be sure to set these bits to 1 (see **(3) UF0 EPNAK register (UF0EN)**).

(2/2)

Bit position	Bit name	Function
0	E0DED	Set this bit to 1 to transmit data of the UF0E0W register. When this bit is set to 1, the EP0NKW bit of the UF0E0EN register is set to 1 and data is transferred. 1: Transmit a short packet. 0: Do not transmit a short packet (default value). If the EP0WC bit of the UF0FIC0 register is set to 1 and if this bit is set to 1 (counter of UF0E0W register = 0 and BP0W of UF0EPS0 register = 1), a Null packet (with a data length of 0) is transmitted. If data exists in the UF0E0W register and if this bit is set to 1 (counter of UF0E0W register ≠ 0 and BP0W of the UF0EPS0 register = 1), and if the FIFO is not full, a short packet is transmitted.

(32) UF0 GPR register (UF0GPR)

This register resets USBF and the USB interface.

This register is write-only, in 8-bit units. If this register is read, 00H is read. Be sure to clear bits 7 to 1 to “0”.

Firmware can reset the USBF by writing 1 to bit 0 of this register. This bit is automatically cleared to 0 after 1 has been written to it. Writing 0 to this bit is invalid.

	7	6	5	4	3	2	1	0	Address	After reset
UF0GPR	0	0	0	0	0	0	0	MRST	40DCH	00H
	0	0	0	0	0	0	0	W		

Bit position	Bit name	Function
0	MRST	Set this bit to 1 to reset USBF. 1: Reset Actually, USBF is reset two USB clocks and initialization by reset is completed five USB clocks after this bit has been set to 1 by firmware after the write signal has become inactive. Resetting USBF by the MRST bit while the system clock is operating has the same result as resetting by the RESET pin (hardware reset) (register value back to default value). However, the UF0CS and UF0BC registers are not reset by the MRST bit.

(33) UF0 mode control register (UF0MODC)

This register controls CPUDEC processing.

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

By setting this register, the setting of the UF0MODS register can be changed. The bit of this register is automatically cleared to 0 only at hardware reset and when the MRST bit of the UF0GRP register has been set to 1.

Even if the bit of this register has automatically been set to 1 by hardware, the setting by firmware takes precedence.

Be sure to clear bits 7 and 5 to 0 to “0”. If they are set to “1”, the operation is not guaranteed.

Caution This register is provided for debugging purposes. Usually, do not set this register except for verifying the operation or when a special mode is used.

	7	6	5	4	3	2	1	0	Address	After reset
UF0MODC	0	CDCGDST	0	0	0	0	0	0	40E8H	00H
	0	R/W	0	0	0	0	0	0		

Bit position	Bit name	Function
6	CDCGDST	Set this bit to 1 to switch the GET_DESCRIPTOR Configuration request to CPUDEC processing. By setting this bit to 1, the CDCGD bit of the UF0MODS register can be forcibly set to 1. 1: Forcibly change the GET_DESCRIPTOR Configuration request to CPUDEC processing (sets the CDCGD bit of the UF0MODS register to 1). 0: Automatically process the GET_DESCRIPTOR Configuration request (default value).

(34) UF0 mode status register (UF0MODS)

This register indicates the configuration status.

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

	7	6	5	4	3	2	1	0	Address	After reset
UF0MODS	0	CDCGD	0	MPACK	DFLT	CONF	0	0	40F0H	00H
	0	R	0	R	R	R	0	0		

Bit position	Bit name	Function
6	CDCGD	This bit specifies whether CPUDEC processing is performed for the GET_DESCRIPTOR Configuration request. 1: Forcibly change the GET_DESCRIPTOR Configuration request to CPUDEC processing. 0: Automatically process the GET_DESCRIPTOR Configuration request (default value).
4	MPACK	This bit indicates the transmit packet size of Endpoint0. 1: Transmit a packet of other than 8 bytes. 0: Transmit a packet of 8 bytes (default value). This bit is automatically set to 1 by hardware after the GET_DESCRIPTOR Device request has been processed (on normal completion of the status stage). It is not cleared to 0 until the USBF has been reset (it is not cleared to 0 by Bus Reset). If this bit is not set to 1, the hardware transfers only the automatically-executed request in 8-byte units. Therefore, even if data of more than 8 bytes is sent by the OUT token to be processed by firmware before completion of the GET_DESCRIPTOR Device request, the data is correctly received. This bit is ignored if the size of Endpoint0 is 8 bytes.
3	DFLT	This bit indicates the default status (DFLT bit = 1). 1: Enables response. 0: Disables response (always no response) (default value). This bit is automatically set to 1 by Bus Reset. The transaction for all the endpoints is not responded to until this bit is set to 1.
2	CONF	This bit indicates whether the SET_CONFIGURATION request has been completed. 1: SET_CONFIGURATION request has been completed. 0: SET_CONFIGURATION request has not been completed (default value). This bit is set to 1 when Configuration value = 1 is received by the SET_CONFIGURATION request. Unless this bit is set to 1, access to an endpoint other than Endpoint0 is ignored. This bit is cleared to 0 when Configuration value = 0 is received by the SET_CONFIGURATION request. It is also cleared to 0 when Bus Reset is detected.

(35) UF0 active interface number register (UF0AIFN)

This register sets the valid Interface number that correctly responds to the GET/SET_INTERFACE request. Because Interface 0 is always valid, Interfaces 1 to 4 can be selected.

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

	7	6	5	4	3	2	1	0	Address	After reset
UF0AIFN	ADDIF	0	0	0	0	0	IFNO1	IFNO0	4100H	00H
	R/W	0	0	0	0	0	R/W	R/W		

Bit position	Bit name	Function
7	ADDIF	This bit allows use of Interfaces numbered other than 0. 1: Support up to the Interface number specified by the IFNO1 and IFNO0 bits. 0: Support only Interface 0 (default value). Setting bits 1 and 0 of this register is invalid when this bit is not set to 1.
1, 0	IFNO1, IFNO0	These bits specify the range of Interface numbers to be supported.

IFNO1	IFNO0	Valid Interface No.
1	1	0, 1, 2, 3, 4
1	0	0, 1, 2, 3
0	1	0, 1, 2
0	0	0, 1

(36) UF0 active alternate setting register (UF0AAS)

This register specifies a link between the Interface number and Alternate Setting.

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

USBF of the V850E2/ME3 can set a five-series Alternate Setting (Alternate Setting 0, 1, 2, 3, and 4 can be defined) and a two-series Alternate Setting (Alternate Setting 0 and 1 can be defined) for one Interface.

	7	6	5	4	3	2	1	0	Address	After reset
UF0AAS	ALT2	IFAL21	IFAL20	ALT2EN	ALT5	IFAL51	IFAL50	ALT5EN	4104H	00H
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

Bit position	Bit name	Function
7, 3	ALTn	These bits specify whether an n-series Alternate Setting is linked with Interface 0. When these bits are set to 1, the setting of the IFALn1 and IFALn0 bits is invalid. 1: Link n-series Alternate Setting with Interface 0. 0: Do not link n-series Alternate Setting with Interface 0 (default value).
6, 5, 2, 1	IFALn1, IFALn0	These bits specify the Interface number to be linked with the n-series Alternate Setting. If the linked Interface number is outside the range specified by the UF0AIFN register, the n-series Alternate Setting is invalid (ALTnEN bit = 0). IFALn1 / IFALn0 / Interface number to be linked: 1 / 1: Links Interface 4. 1 / 0: Links Interface 3. 0 / 1: Links Interface 2. 0 / 0: Links Interface 1. Do not link a five-series Alternate Setting and a two-series Alternate Setting with the same Interface number.
4, 0	ALTnEN	These bits validate the n-series Alternate Setting. Unless these bits are set to 1, the setting of the ALTn, IFALn1, and IFALn0 bits is invalid. 1: Validate the n-series Alternate Setting. 0: Do not validate the n-series Alternate Setting (default value).

Remark n = 2, 5

For example, when the UF0AIFN register is set to 82H and the UF0AAS register is set to 15H, Interfaces 0, 1, 2, and 3 are valid. Interfaces 0 and 2 support only Alternate Setting 0. Interface 1 supports Alternate Setting 0 and 1, and Interface 3 supports Alternate Setting 0, 1, 2, 3, and 4. With this setting, requests GET_INTERFACE wIndex = 0/1/2/3, SET_INTERFACE wValue = 0 & wIndex = 0/2, SET_INTERFACE wValue = 0/1 & wIndex = 1, and SET_INTERFACE wValue = 0/1/2/3/4 & wIndex = 3 are automatically responded to, and a STALL response is made to the other GET/SET_INTERFACE requests.

(37) UF0 alternate setting status register (UF0ASS)

This register indicates the current status of the Alternate Setting.

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

The value received by the SET_INTERFACE request is reflected on the UF0IFn register (n = 0 to 4) as well as on this register.

	7	6	5	4	3	2	1	0	Address	After reset
UF0ASS	0	0	0	0	AL5ST3	AL5ST2	AL5ST1	AL2ST	4108H	00H
	R	R	R	R	R	R	R	R		

Bit position	Bit name	Function
3 to 1	AL5ST3 to AL5ST1	These bits indicate the current status of the five-series Alternate Setting.

AL5ST3	AL5ST2	AL5ST1	Selected Alternate Setting number
1	0	0	Alternate Setting 4
0	1	1	Alternate Setting 3
0	1	0	Alternate Setting 2
0	0	1	Alternate Setting 1
0	0	0	Alternate Setting 0

Bit position	Bit name	Function
0	AL2ST	This bit indicates the current status of the two-series Alternate Setting (selected Alternate Setting number). 1: Alternate Setting 1 0: Alternate Setting 0

(38) UF0 endpoint 1 interface mapping register (UF0E1IM)

This register specifies for which Interface and Alternate Setting Endpoint1 is valid.

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

The setting of this register and the Alternate Setting selected by the SET_INTERFACE request indicate whether Endpoint1 is currently valid, and the hardware determines how the GET_STATUS/CLEAR_FEATURE/SET_FEATURE Endpoint1 request and the IN transaction to Endpoint1 are responded to, and whether the related bits are valid or invalid.

	7	6	5	4	3	2	1	0	Address	After reset
UF0E1IM	E1EN2	E1EN1	E1EN0	E12AL1	E15AL4	E15AL3	E15AL2	E15AL1	410CH	00H
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

Bit position	Bit name	Function
7 to 5	E1EN2 to E1EN0	These bits set a link between the Interface of Endpoint1 and the two-/five-series Alternate Setting. The endpoint is linked with Alternate Setting 0. The endpoint linked with Alternate Setting 0 cannot be excluded from Alternate Setting 1 to 4.

E1EN2	E1EN1	E1EN0	Link status
1	1	1	Not linked with Interface
1	1	0	
1	0	1	Linked with Interface 4 and Alternate Setting 0
1	0	0	Linked with Interface 3 and Alternate Setting 0
0	1	1	Linked with Interface 2 and Alternate Setting 0
0	1	0	Linked with Interface 1 and Alternate Setting 0
0	0	1	Linked with Interface 0 and Alternate Setting 0
0	0	0	Not linked with Interface (default value)

When these bits are set to 110 or 111, they are invalid even if the E12AL1 bit is cleared to 0.

If the endpoint is linked, setting of the CONF bit of the UF0MODS register to 1 indicates that Endpoint1 is valid.

4	E12AL1	This bit validates Endpoint1 when the two-series Alternate Setting and the Alternate Setting of the linked Interface are set to 1. 1: Validate the endpoint when Alternate Setting 1 is set with CONF bit = 1. 0: Do not validate the endpoint even when Alternate Setting 1 is set with CONF bit = 1 (default value). This bit is valid when the E15AL4 to E15AL1 bits are 0000.
3 to 0	E15AL4 to E15AL1	These bits validate Endpoint1 when the five-series Alternate Setting and the Alternate Setting of the linked Interface are set to n. 1: Validate the endpoint when Alternate Setting n is set with CONF bit = 1. 0: Do not validate the endpoint even when Alternate Setting n is set with CONF bit = 1 (default value).

Remark n = 1 to 4

(39) UF0 endpoint 2 interface mapping register (UF0E2IM)

This register specifies for which Interface and Alternate Setting Endpoint2 is valid.

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

The setting of this register and the Alternate Setting selected by the SET_INTERFACE request indicate whether Endpoint2 is currently valid, and the hardware determines how the GET_STATUS/CLEAR_FEATURE/SET_FEATURE Endpoint2 request and the OUT transaction to Endpoint2 are responded to, and whether the related bits are valid or invalid.

	7	6	5	4	3	2	1	0	Address	After reset
UF0E2IM	E2EN2	E2EN1	E2EN0	E22AL1	E25AL4	E25AL3	E25AL2	E25AL1	4110H	00H
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

Bit position	Bit name	Function
7 to 5	E2EN2 to E2EN0	These bits set a link between the Interface of Endpoint2 and the two-/five-series Alternate Setting. The endpoint is linked with Alternate Setting 0. The endpoint linked with Alternate Setting 0 cannot be excluded from Alternate Setting 1 to 4.
4	E22AL1	This bit validates Endpoint2 when the two-series Alternate Setting and the Alternate Setting of the linked Interface are set to 1. 1: Validate the endpoint when Alternate Setting 1 is set with CONF bit = 1. 0: Do not validate the endpoint even when Alternate Setting 1 is set with CONF bit = 1 (default value). This bit is valid when the E25AL4 to E25AL1 bits are 0000.
3 to 0	E25AL4 to E25AL1	These bits validate Endpoint2 when the five-series Alternate Setting and the Alternate Setting of the linked Interface are set to n. 1: Validate the endpoint when Alternate Setting n is set with CONF bit = 1. 0: Do not validate the endpoint even when Alternate Setting n is set with CONF bit = 1 (default value).

E2EN2	E2EN1	E2EN0	Link status
1	1	1	Not linked with Interface
1	1	0	
1	0	1	Linked with Interface 4 and Alternate Setting 0
1	0	0	Linked with Interface 3 and Alternate Setting 0
0	1	1	Linked with Interface 2 and Alternate Setting 0
0	1	0	Linked with Interface 1 and Alternate Setting 0
0	0	1	Linked with Interface 0 and Alternate Setting 0
0	0	0	Not linked with Interface (default value)

When these bits are set to 110 or 111, they are invalid even if the E22AL1 bit is cleared to 0.

If the endpoint is linked, setting of the CONF bit of the UF0MODS register to 1 indicates that Endpoint2 is valid.

Remark n = 1 to 4

(40) UF0 endpoint 7 interface mapping register (UF0E7IM)

This register specifies for which Interface and Alternate Setting Endpoint7 is valid.

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

The setting of this register and the Alternate Setting selected by the SET_INTERFACE request indicate whether Endpoint7 is currently valid, and the hardware determines how the GET_STATUS/CLEAR_FEATURE/SET_FEATURE Endpoint7 request and the IN transaction to Endpoint7 are responded to, and whether the related bits are valid or invalid.

	7	6	5	4	3	2	1	0	Address	After reset
UF0E7IM	E7EN2	E7EN1	E7EN0	E72AL1	E75AL4	E75AL3	E75AL2	E75AL1	4124H	00H
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

Bit position	Bit name	Function
7 to 5	E7EN2 to E7EN0	These bits set a link between the Interface of Endpoint7 and the two-/five-series Alternate Setting. The endpoint is linked with Alternate Setting 0. The endpoint linked with Alternate Setting 0 cannot be excluded from Alternate Setting 1 to 4.

E7EN2	E7EN1	E7EN0	Link status
1	1	1	Not linked with Interface
1	1	0	
1	0	1	Linked with Interface 4 and Alternate Setting 0
1	0	0	Linked with Interface 3 and Alternate Setting 0
0	1	1	Linked with Interface 2 and Alternate Setting 0
0	1	0	Linked with Interface 1 and Alternate Setting 0
0	0	1	Linked with Interface 0 and Alternate Setting 0
0	0	0	Not linked with Interface (default value)

When these bits are set to 110 or 111, they are invalid even if the E72AL1 bit is cleared to 0.

If the endpoint is linked, setting of the CONF bit of the UF0MODS register to 1 indicates that Endpoint7 is valid.

4	E72AL1	This bit validates Endpoint7 when the two-series Alternate Setting and the Alternate Setting of the linked Interface are set to 1. 1: Validate the endpoint when Alternate Setting 1 is set with CONF bit = 1. 0: Do not validate the endpoint even when Alternate Setting 1 is set with CONF bit = 1 (default value). This bit is valid when the E75AL4 to E75AL1 bits are 0000.
3 to 0	E75AL4 to E75AL1	These bits validate Endpoint7 when the five-series Alternate Setting and the Alternate Setting of the linked Interface are set to n. 1: Validate the endpoint when Alternate Setting n is set with CONF bit = 1. 0: Do not validate the endpoint even when Alternate Setting n is set with CONF bit = 1 (default value).

Remark n = 1 to 4

3.4.5 EPC Data hold registers

(1) UF0 EP0 read register (UF0E0R)

The UF0E0R register is a 64-byte FIFO that stores the OUT data sent from the host in the data stage of control transfer to/from Endpoint0.

This register is read-only, in 8-bit units. A write access to this register is ignored.

The hardware automatically transfers data to the UF0E0R register when it has received the data from the host. When the data has been correctly received, the E0ODT bit of the UF0IS1 register is set to 1. The UF0E0L register holds the quantity of the received data, and an interrupt request (INTUSBF0) is issued. The UF0E0L register always updates the length of the received data while it is receiving data. If the final transfer is correct reception, the interrupt request is generated. If the reception is abnormal, the UF0E0L register is cleared to 0 and the interrupt request is not generated.

The data held by the UF0E0R register must be read by firmware up to the value of the amount of data read by the UF0E0L register. Check that all data has been read by using the EP0R bit of the UF0EPS0 register (the EP0R bit = 0 when all data has been read). If the value of the UF0E0L register is 0, the EP0NKR bit of the UF0E0N register is cleared to 0, and the UF0E0R register is ready for reception. The UF0E0R register is cleared when the next SETUP token has been received.

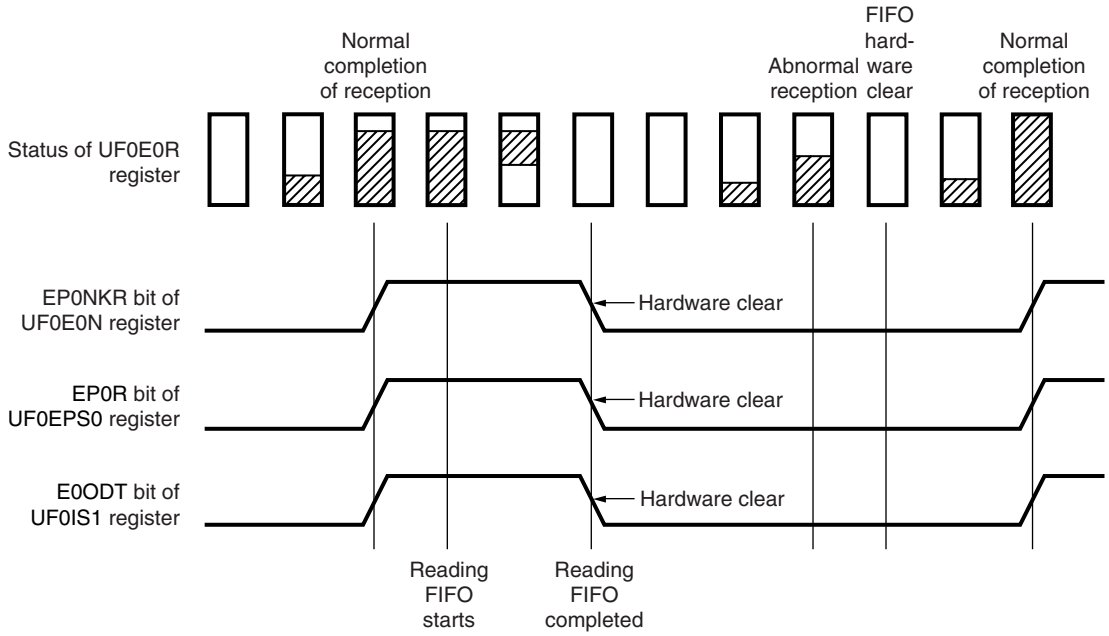
Caution Read all the data stored. Clear the FIFO to discard some data.

	7	6	5	4	3	2	1	0	Address	After reset
UF0E0R	E0R7	E0R6	E0R5	E0R4	E0R3	E0R2	E0R1	E0R0	4200H	Undefined
	R	R	R	R	R	R	R	R		

Bit position	Bit name	Function
7 to 0	E0R7 to E0R0	These bits store the OUT data sent from the host in the data stage of control transfer to/from Endpoint0.

The operation of the UF0E0R register is illustrated below.

Figure 3-1. Operation of UF0E0R Register



(2) UF0 EP0 length register (UF0E0L)

The UF0E0L register stores the data length held by the UF0E0R register.

This register is read-only, in 8-bit units. A write access to this register is ignored.

The UF0E0L register always updates the length of the received data while it is receiving data. If the final transfer is abnormal reception, the UF0E0L register is cleared to 0 and the interrupt request is not generated. The interrupt request is generated only when the reception is normal, and the firmware can read as many data from the UF0E0R register as the value read from the UF0E0L register. The value of the UF0E0L register is decremented each time the UF0E0R register has been read.

	7	6	5	4	3	2	1	0	Address	After reset
UF0E0L	E0L7	E0L6	E0L5	E0L4	E0L3	E0L2	E0L1	E0L0	4204H	00H
	R	R	R	R	R	R	R	R		

Bit position	Bit name	Function
7 to 0	E0L7 to E0L0	These bits store the data length held by the UF0E0R register.

(3) UF0 EP0 setup register (UF0E0ST)

The UF0E0ST register holds the SETUP data sent from the host.

This register is read-only, in 8-bit units. A write access to this register is ignored.

The UF0E0ST register always writes data when a SETUP transaction has been received. The hardware sets the PROT bit of the UF0IS1 register when it has correctly received the SETUP transaction. It sets the CPUDEC bit of the UF0IS1 register in the case of a firmware-processed request. Then an interrupt request (INTUSBF0) is issued.

Caution In the case of firmware-processed request, be sure to read the request in 8-byte units. If it is not read in 8-byte units, the subsequent requests cannot be correctly decoded. The read counter of the UF0E0ST register is not cleared even when Bus Reset is received. Always read this counter in 8-byte units regardless of whether Bus Reset is received or not.

Because the UF0E0ST register always enables writing, the hardware overwrites data to this register even if a SETUP transaction is received while the data of the register is being read. Even if the SETUP transaction cannot be correctly received, the CPUDEC interrupt request and Protect interrupt request are not generated, but the previous data is discarded. If a SETUP token of less than 8 bytes is received, however, the received SETUP token is discarded, and the previously received SETUP data is retained. If the SETUP token is received more than once when control transfer is executed once, be sure to check the PROT bit of the UF0IS1 register under the conditions below. If PROT bit = 1, read the UF0E0ST register again because the SETUP transaction has been received more than once.

<1> If a request is decoded by firmware and the UF0E0R register is read or the UF0E0W register is written

<2> When preparing for a STALL response for the request to which the decode result does not correspond

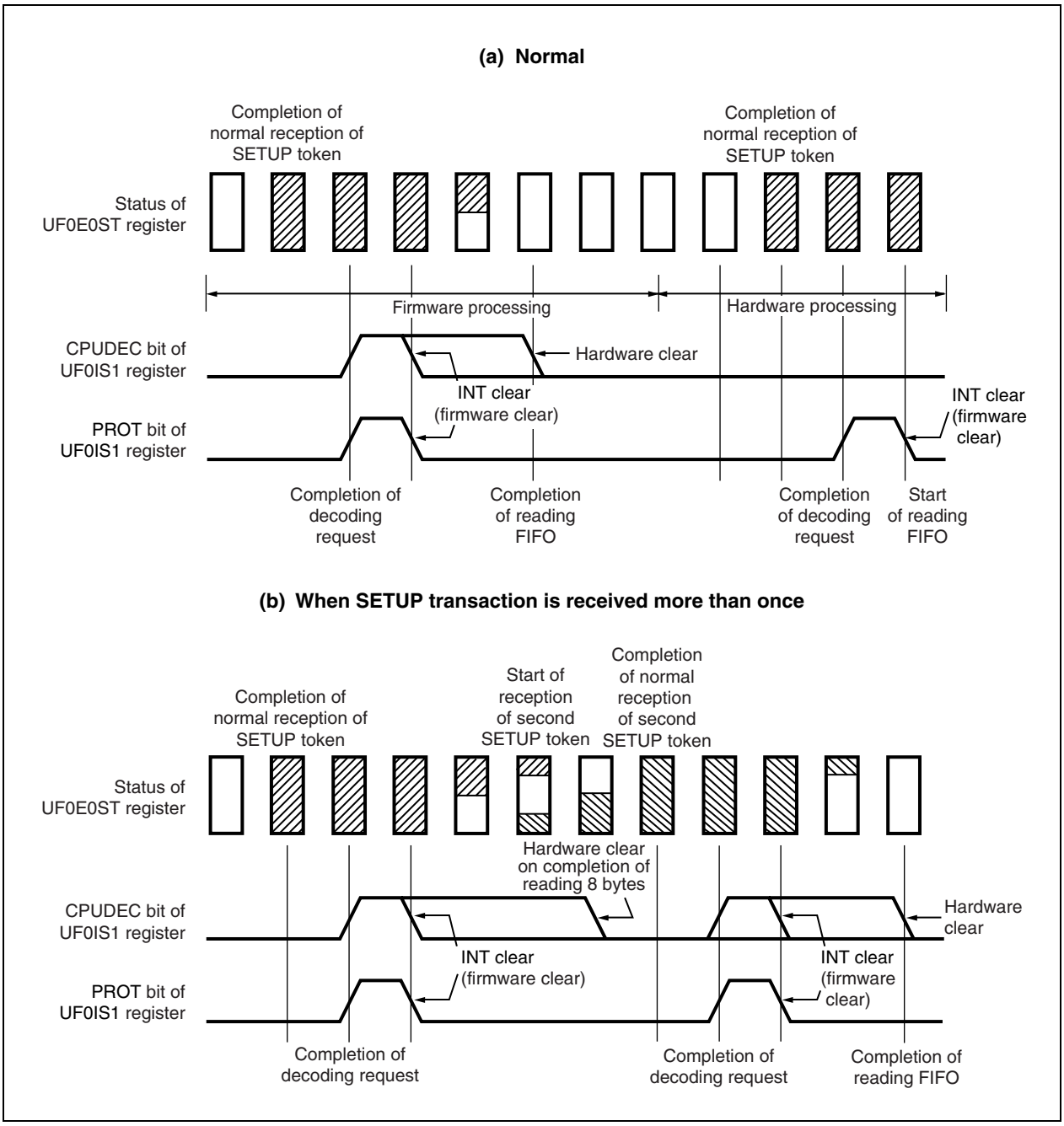
Caution Be sure to read all the stored data. The UF0E0ST register is always updated by the request in the SETUP transaction.

	7	6	5	4	3	2	1	0	Address	After reset
UF0E0ST	E0S7	E0S6	E0S5	E0S4	E0S3	E0S2	E0S1	E0S0	4208H	00H
	R	R	R	R	R	R	R	R		

Bit position	Bit name	Function
7 to 0	E0S7 to E0S0	These bits hold the SETUP data sent from the host.

The operation of the UF0E0ST register is illustrated below.

Figure 3-2. Operation of UF0E0ST Register



(4) UF0 EP0 write register (UF0E0W)

The UF0E0W register is a 64-byte FIFO that stores the IN data (passes it to SIE) sent to the host in the data stage to Endpoint0.

This register is write-only, in 8-bit units. When this register is read, 00H is read.

The hardware transmits data to the USB bus in synchronization with an IN token only when the EP0NKW bit of the UF0E0N register is set to 1 (when NAK is not transmitted). When data is transmitted and when the host correctly receives the data, the EP0NKW bit of the UF0E0N register is automatically cleared to 0 by hardware.

A short packet is transmitted when data is written to the UF0E0W register and the E0DED bit of the UF0DEND register is set to 1 (EP0W bit of the UF0EPS0 register = 1 (data exists)). A Null packet is transmitted when the UF0E0W register is cleared and the E0DED bit of the UF0DEND register is set to 1 (EP0W bit of the UF0EPS0 register = 1 (data exists)).

The UF0E0W register is cleared to 0 when the next SETUP token is received while transmission has not been completed yet. If the stage of control transfer (read) changes to the status stage while ACK has not been correctly received in the data stage, the UF0E0W register is automatically cleared to 0. At the same time, it is also cleared to 0 if the EP0NKW bit of the UF0E0N register is 1.

If the UF0E0W register is read while no data is in it, 00H is read.

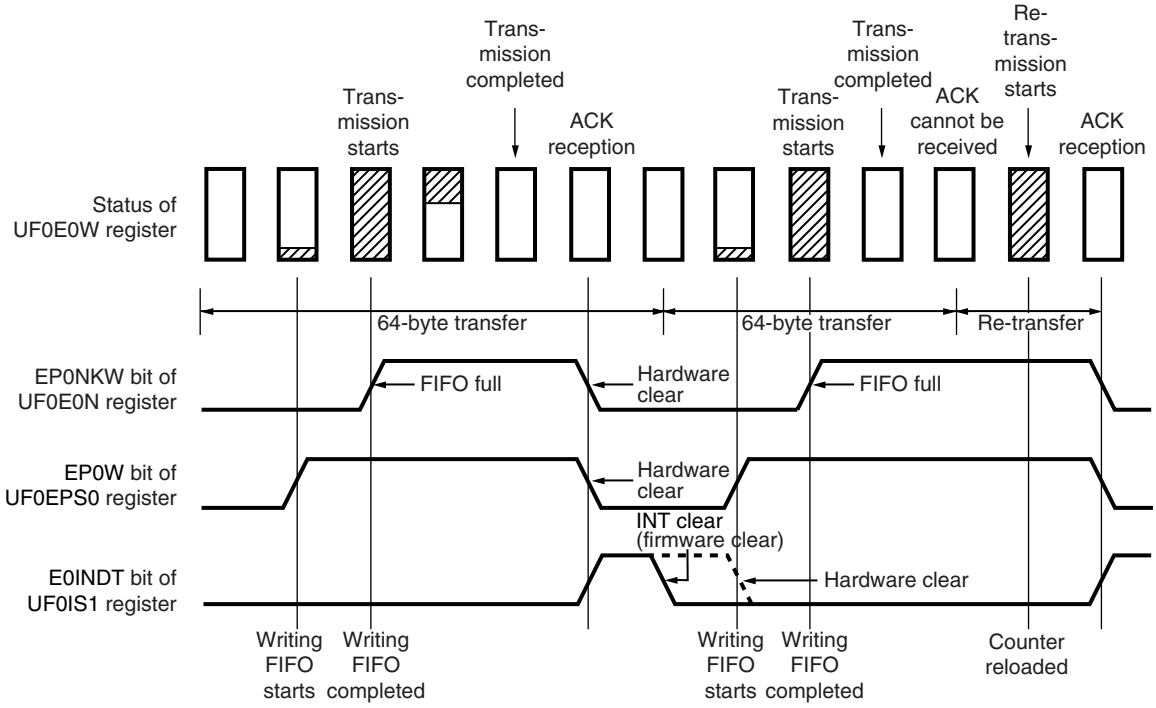
	7	6	5	4	3	2	1	0	Address	After reset
UF0E0W	E0W7	E0W6	E0W5	E0W4	E0W3	E0W2	E0W1	E0W0	420CH	Undefined
	W	W	W	W	W	W	W	W		

Bit position	Bit name	Function
7 to 0	E0W7 to E0W0	These bits store the IN data sent to the host in the data stage to Endpoint0.

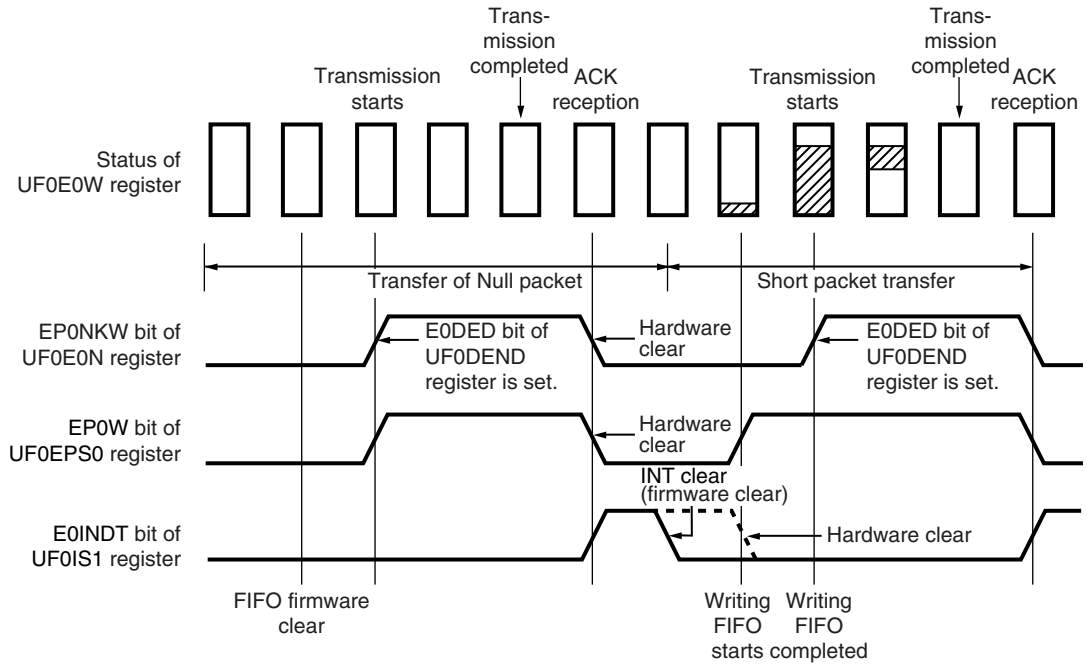
The operation of the UF0E0W register is illustrated below.

Figure 3-3. Operation of UF0E0W Register

(a) Normal



(b) When Null packet or short packet is transmitted



(5) UF0 bulk out 1 register (UF0BO1)

The UF0BO1 register is a 64-byte × 2 FIFO that stores data for Endpoint2. This register consists of two banks of 64-byte FIFOs each of which performs a toggle operation and repeatedly connects the buses on the SIE and CPU sides. The toggle operation takes place when data is in the FIFO on the SIE side and when no data is in the FIFO on the CPU side (counter value = 0).

This register is read-only, in 8-bit units. A write access to this register is ignored.

When the hardware receives data for Endpoint2 from the host, it automatically transfers the data to the UF0BO1 register. When the register correctly receives the data, a FIFO toggle operation occurs. As a result, the BKO1DT bit of the UF0IS3 register is set to 1, the quantity of the received data is held by the UF0BO1L register, and an interrupt request or DMA request is issued to the CPU. Whether the interrupt request or DMA request is issued can be selected by using the DQBO1MS bit of the UF0IDR register. In DMA mode, it is read via the EP2_BULK_OUT register.

Read the data held by the UF0BO1 register by firmware, up to the value of the amount of data read by the UF0BO1L register. When the correct received data is held by the FIFO connected to the SIE side and the value of the UF0BO1L register reaches 0, the toggle operation of the FIFO occurs, and the BKO1NK bit of the UF0EN register is automatically cleared to 0. If data greater than the value of the UF0BO1L register is read and if the FIFO toggle condition is satisfied, the toggle operation of the FIFO occurs. As a result, the next packet may be read by mistake. Note that, if the toggle condition is not satisfied, the first data is repeatedly read.

If overrun data is received while data is held by the FIFO connected to the CPU side, Endpoint2 stalls, and the FIFO on the CPU side is cleared.

When the UF0BO1 register is read while no data is in it, an undefined value is read.

Caution Be sure to read all the data stored in this register.

	7	6	5	4	3	2	1	0	Address	After reset
UF0BO1	BKO17	BKO16	BKO15	BKO14	BKO13	BKO12	BKO11	BKO10	4210H	Undefined
	R	R	R	R	R	R	R	R		

Bit position	Bit name	Function
7 to 0	BKO17 to BKO10	These bits store data for Endpoint2.

The operation of the UF0BO1 register is illustrated below.

Figure 3-4. Operation of UF0BO1 Register (1/2)

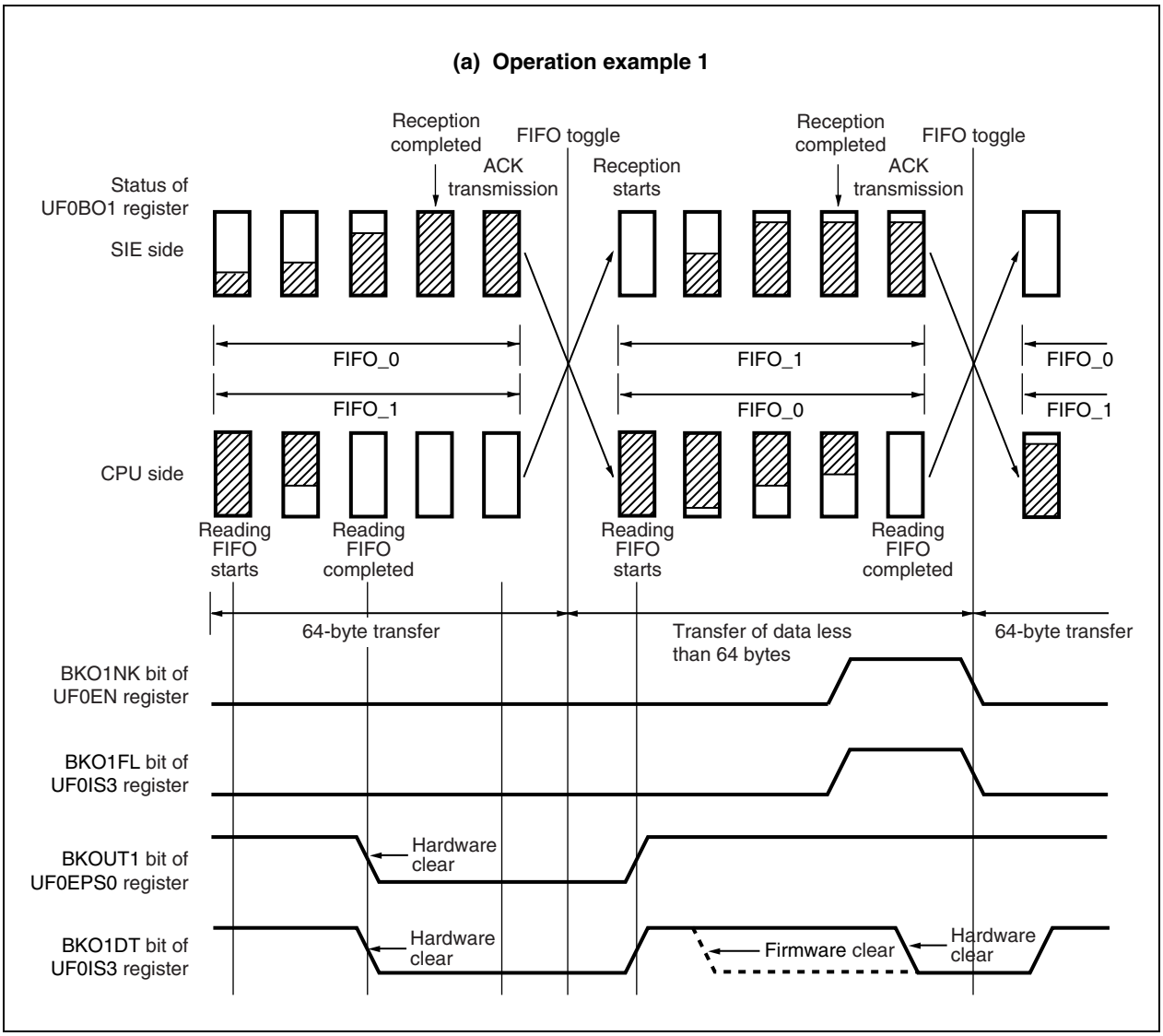
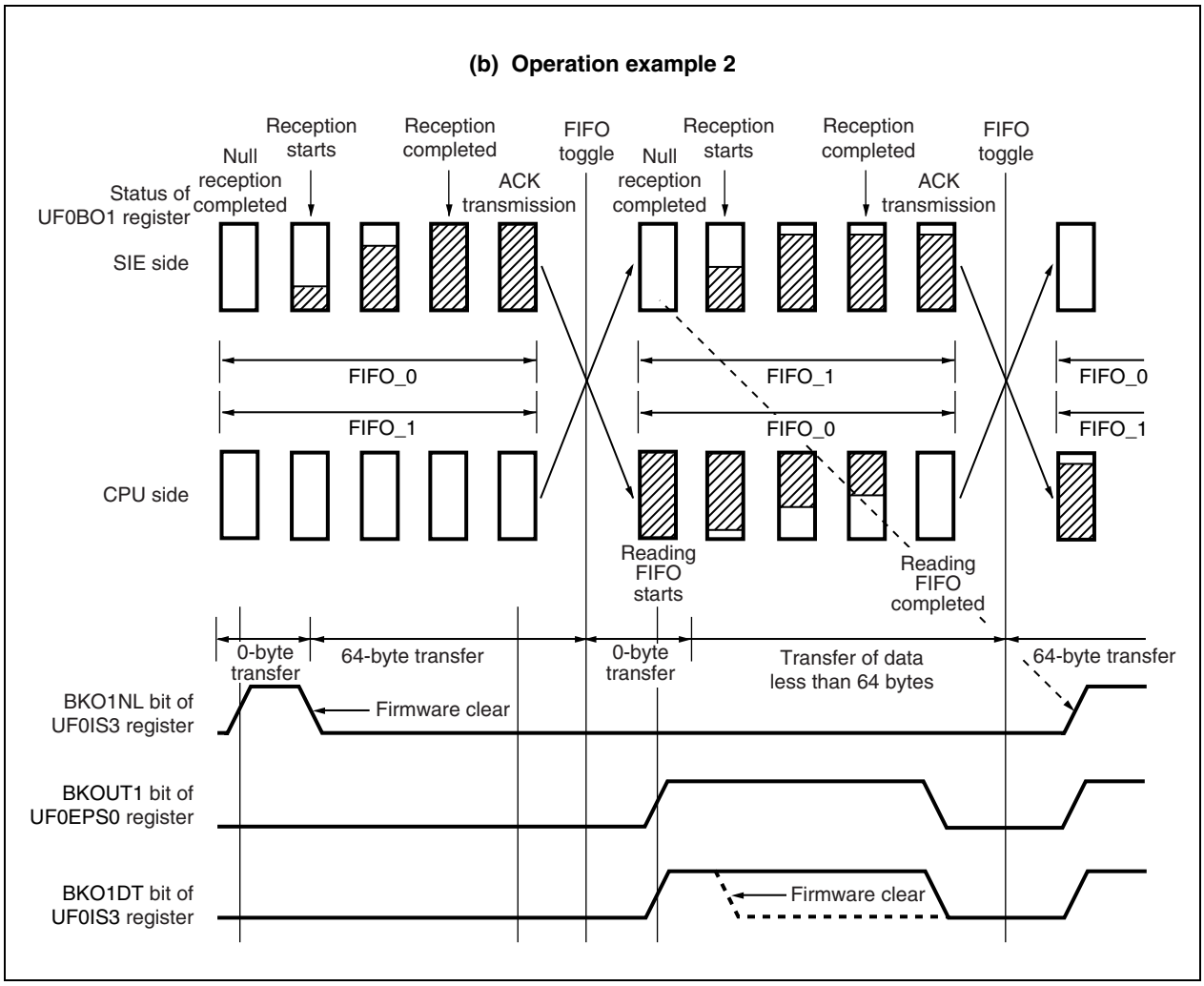


Figure 3-4. Operation of UF0BO1 Register (2/2)



(6) UF0 bulk out 1 length register (UF0BO1L)

The UF0BO1L register stores the length of the data held by the UF0BO1 register.

This register is read-only, in 8-bit units. A write access to this register is ignored.

The UF0BO1L register always updates the received data length while it is receiving data. If the final transfer is abnormal reception, the UF0BO1L register is cleared to 00H, and an interrupt request is not generated. Only if the reception is normal, the interrupt request is generated, and firmware can read as much data from the UF0BO1 register as the value read from the UF0BO1L register. The value of the UF0BO1L register is decremented each time the UF0BO1 register has been read.

	7	6	5	4	3	2	1	0	Address	After reset
UF0BO1L	BKO1L7	BKO1L6	BKO1L5	BKO1L4	BKO1L3	BKO1L2	BKO1L1	BKO1L0	4214H	00H
	R	R	R	R	R	R	R	R		

Bit position	Bit name	Function
7 to 0	BKO1L7 to BKO1L0	The data length (sum of SIE side and CPU side) retained in the UF0BO1 register is stored into these bits.

(7) UF0 bulk in 1 register (UF0BI1)

The UF0BI1 register is a 64-byte × 2 FIFO that stores data for Endpoint1. This register consists of two banks of 64-byte FIFOs each of which performs a toggle operation and repeatedly connects the buses on the SIE and CPU sides. The toggle operation takes place when no data is in the FIFO on the SIE side and when the FIFO on the CPU side is correctly written (FIFO full or BKI1DED bit of the UF0DEND register = 1).

This register is write-only, in 8-bit units. When this register is read, 00H is read.

The hardware transmits data to the USB bus in synchronization with the IN token for Endpoint1 only when the BKI1NK bit of the UF0EN register is set to 1 (when NAK is not transmitted). The address at which data is to be written or read is managed by the hardware. Therefore, firmware can transmit data to the host only by writing the data to the UF0BI1 register sequentially. A short packet is transmitted when data is written to the UF0BI1 register and the BKI1DED bit of the UF0DEND register is set to 1 (BKIN1 bit of UF0EPS0 register = 1 (data exists)). A Null packet is transmitted when the UF0BI1 register is cleared and the BKI1DED bit of the UF0DEND register is set to 1 (BKIN1 bit of the UF0EPS0 register = 1 (data exists)). After data is transmitted normally, FIFO toggle operation is started, the BKI1DT bit of the UF0IS2 register is set to 1, and an interrupt request is issued to the CPU. An interrupt request or DMA request can be selected by using the DQBI1MS bit of the UF0IDR register. In DMA mode, it is read via the EP1_BULK_IN register.

	7	6	5	4	3	2	1	0	Address	After reset
UF0BI1	BKI17	BKI16	BKI15	BKI14	BKI13	BKI12	BKI11	BKI10	4220H	Undefined
	W	W	W	W	W	W	W	W		

Bit position	Bit name	Function
7 to 0	BKI17 to BKI10	These bits store data for Endpoint1.

The operation of the UF0BI1 register is illustrated below.

Figure 3-5. Operation of UF0BI1 Register (1/3)

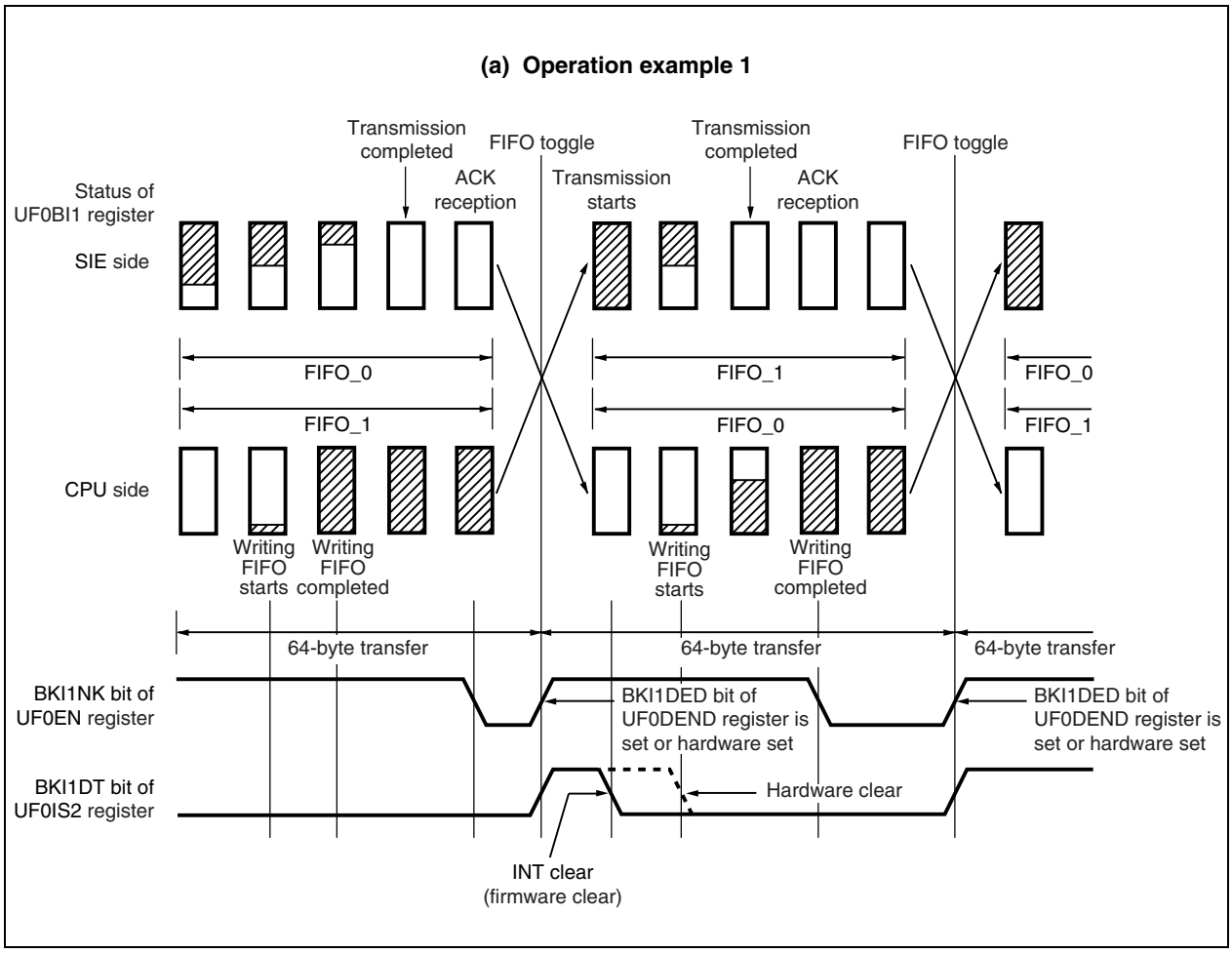


Figure 3-5. Operation of UF0BI1 Register (2/3)

(b) Operation example 2

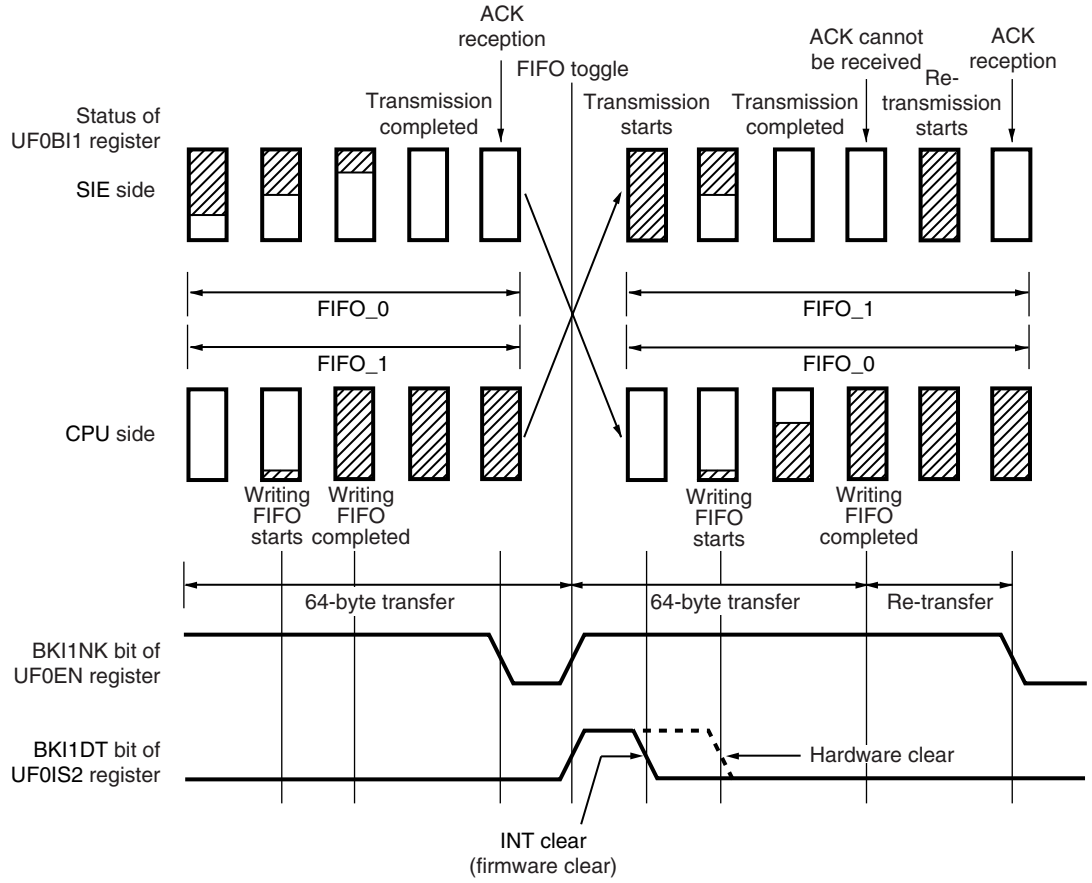
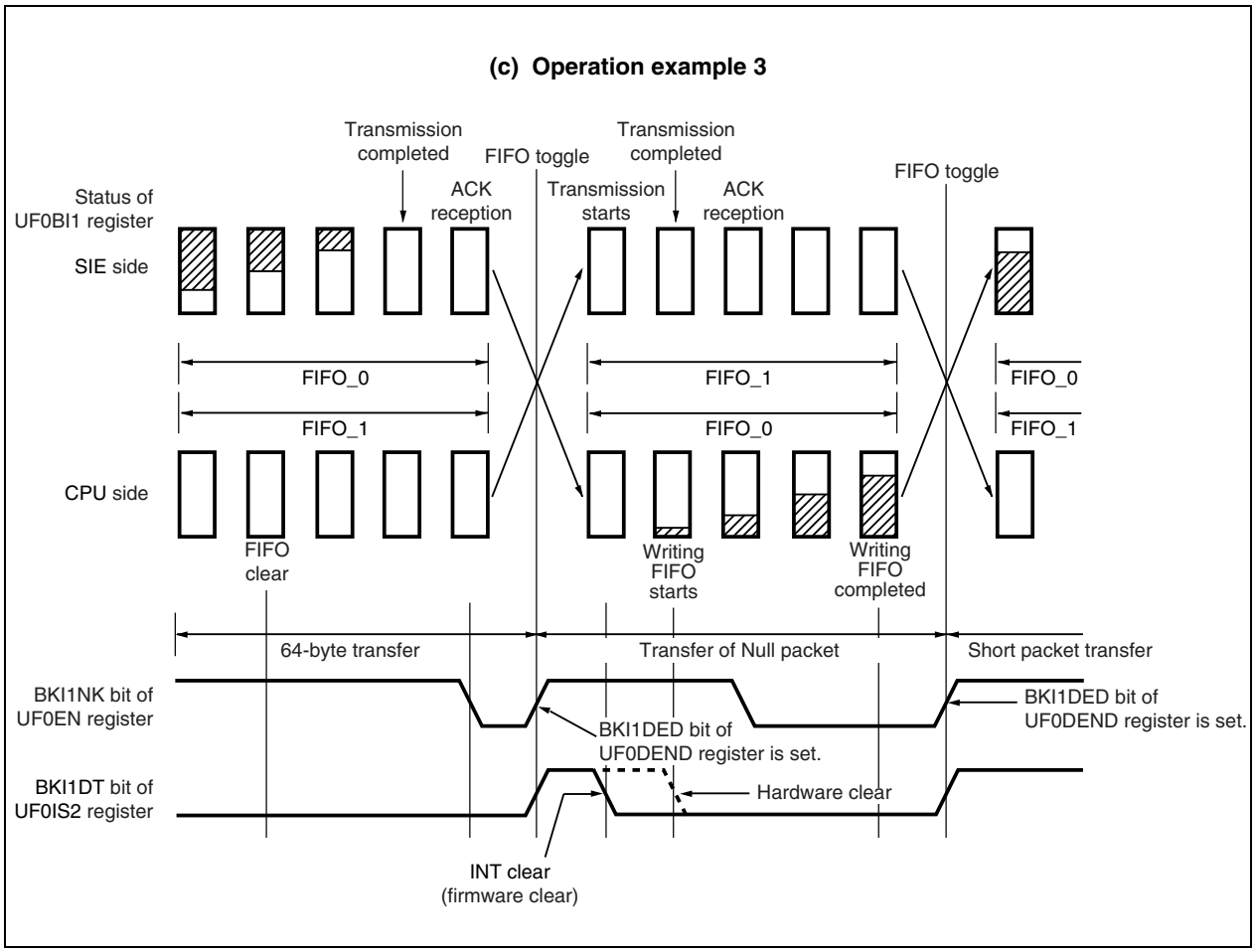


Figure 3-5. Operation of UF0BI1 Register (3/3)



(8) UF0 interrupt 1 register (UF0INT1)

The UF0INT1 register is an 8-byte FIFO that stores data for Endpoint7 (to be passed to SIE).

This register is write-only, in 8-bit units. When this register is read, 00H is read.

The hardware transmits data to the USB bus in synchronization with the IN token for Endpoint7 only when the IT1NK bit of the UF0EN register is set to 1 (when NAK is not transmitted). When the data is transmitted and the host correctly receives it, the IT1NK bit of the UF0EN register is automatically cleared to 0 by hardware. A short packet is transmitted when data is written to the UF0INT1 register and the IT1DEND bit of the UF0DEND register is set to 1 (IT1 bit of the UF0EPS0 register = 1 (data exists)). A Null packet is transmitted when the UF0INT1 register is cleared and the IT1DEND bit of the UF0DEND register is set to 1 (IT1 bit of the UF0EPS0 register = 1 (data exists)).

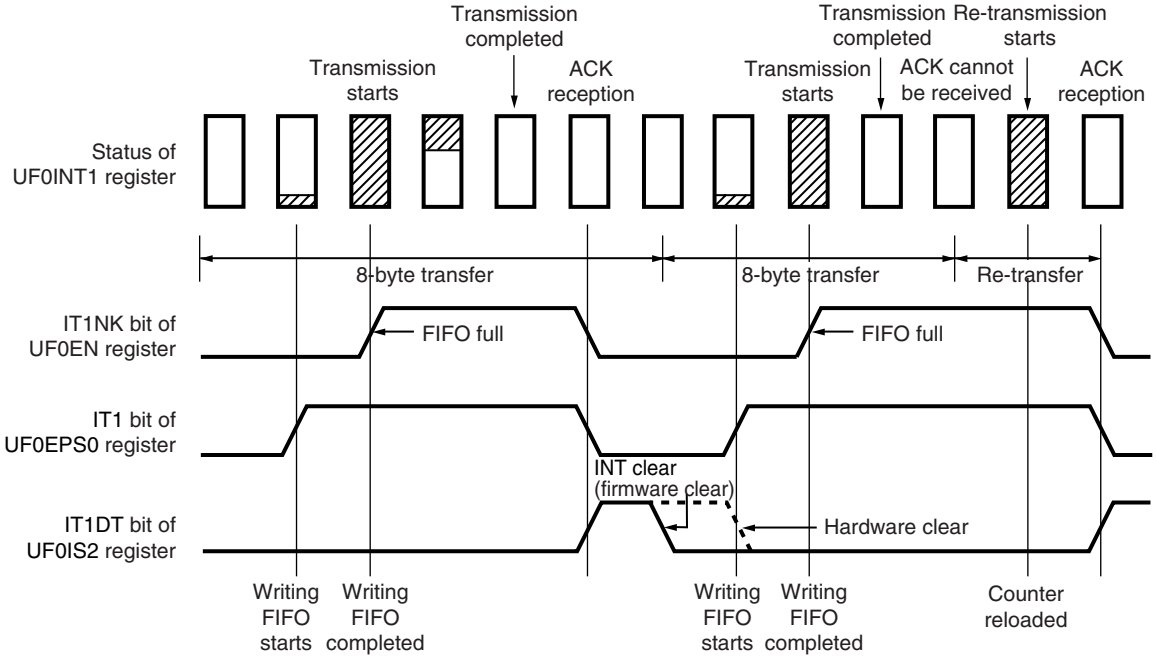
	7	6	5	4	3	2	1	0	Address	After reset
UF0INT1	IT17	IT16	IT15	IT14	IT13	IT12	IT11	IT10	4228H	Undefined
	W	W	W	W	W	W	W	W		

Bit position	Bit name	Function
7 to 0	IT17 to IT10	These bits store data for Endpoint7.

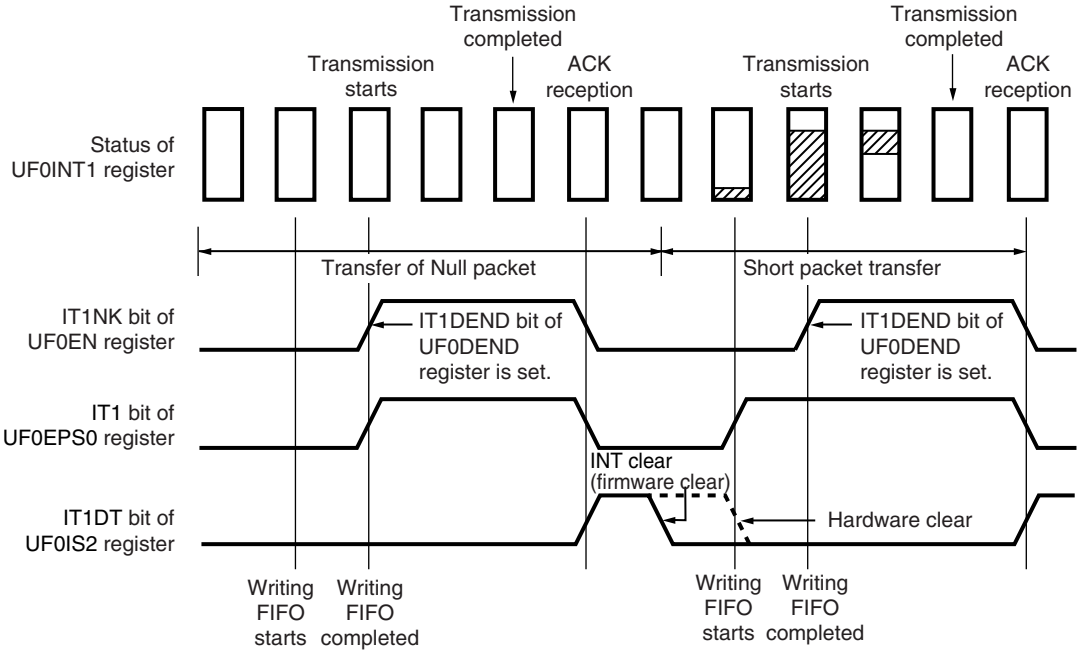
The operation of the UF0INT1 register is illustrated below.

Figure 3-6. Operation of UF0INT1 Register

(a) 8-byte transfer



(b) When Null packet or short packet is transmitted



3.4.6 EPC Request data register

(1) UF0 device status register L (UF0DSTL)

This register stores the value that is to be returned in response to the GET_STATUS Device request.

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

The hardware automatically transmits the contents of this register to the host when it has received the GET_STATUS Device request.

Caution To rewrite this register, set the EP0NKA bit of the UF0E0NA register to 1 before reading the register contents, and rewrite the register contents after confirming that the bit has been set, in order to prevent conflict between a read access by SIE and a write access.

	7	6	5	4	3	2	1	0	Address	After reset
UF0DSTL	0	0	0	0	0	0	RMWK	SFPW	4288H	00H
	0	0	0	0	0	0	R/W	R/W		

Bit position	Bit name	Function
1	RMWK	This bit specifies whether the remote wakeup function of the device is used. 1: Enabled 0: Disabled If the device supports a remote wakeup function, this bit is set to 1 by hardware when the SET_FEATURE Device request has been received, and is cleared to 0 by hardware when the CLEAR_FEATURE Device request has been received. If the device does not support a remote wakeup function, make sure that the SET_FEATURE Device request is not issued from the host.
0	SFPW	This bit indicates whether the device is self-powered or bus-powered. 1: Self-powered 0: Bus-powered

(2) UF0 EP0 status register L (UF0E0SL)

This register stores the value that is to be returned in response to the GET_STATUS Endpoint0 request.

This register can be read or written in 8-bit units. Note, however, that data can be written to this register only when the EP0NKA bit is set to 1.

A write access to this register is ignored while a USB-side access to Endpoint0 is being received.

When the E0HALT bit is set to 1 by firmware, it is not reflected until the next SETUP token is received if the control transfer immediately before is for the SET_FEATURE Endpoint0, CLEAR_FEATURE Endpoint0, GET_STATUS Endpoint0 request, or an firmware-processed request.

The hardware automatically transmits the contents of this register to the host when it has received the GET_STATUS Endpoint0 request. If Endpoint0 has stalled, the UF0E0W and UF0E0R registers are cleared, and the EP0NKW and EP0NKR bits of the UF0E0N register are cleared to 0.

Caution To rewrite this register, set the EP0NKA bit of the UF0E0NA register to 1 before reading the register contents, and rewrite the register contents after confirming that the bit has been set, in order to prevent conflict between a read access by SIE and a write access.

	7	6	5	4	3	2	1	0	Address	After reset
UF0E0SL	0	0	0	0	0	0	0	E0HALT	4298H	00H
	0	0	0	0	0	0	0	R/W		

Bit position	Bit name	Function
0	E0HALT	This bit indicates the status of Endpoint0. 1: Stalled 0: Not stalled This bit is set to 1 by hardware when the SET_FEATURE Endpoint0 request has been received, and cleared to 0 by hardware when the CLEAR_FEATURE Endpoint0 request has been received. DATA PID is initialized to DATA0.

(3) UF0 EP1 status register L (UF0E1SL)

This register stores the value that is to be returned in response to the GET_STATUS Endpoint1 request.

This register can be read or written in 8-bit units. Note, however, that data can be written to this register only when the EP0NKA bit is set to 1.

If an error occurs in Endpoint1, the E1HALT bit is set to 1. A write access to this register is ignored while a USB-side access to Endpoint1 is being received.

The hardware automatically transmits the contents of this register to the host when it has received the GET_STATUS Endpoint1 request. If Endpoint1 has stalled, the UF0BI1 register is cleared and the BKI1NK bit of the UF0EN register is cleared to 0.

Because writing this register is always masked when transfer to Endpoint1, rather than control transfer, is executed, be sure to check this register to see if data has been correctly written to it.

Caution To rewrite this register, set the EP0NKA bit of the UF0E0NA register to 1 before reading the register contents, and rewrite the register contents after confirming that the bit has been set, in order to prevent conflict between a read access by SIE and a write access.

	7	6	5	4	3	2	1	0	Address	After reset
UF0E1SL	0	0	0	0	0	0	0	E1HALT	42A0H	00H
	0	0	0	0	0	0	0	R/W		

Bit position	Bit name	Function
0	E1HALT	This bit indicates the status of Endpoint1. 1: Stalled 0: Not stalled This bit is set to 1 by hardware when the SET_FEATURE Endpoint1 request has been received. It is cleared to 0 by hardware when the CLEAR_FEATURE Endpoint1 request, SET_CONFIGURATION request, or the SET_INTERFACE request for the Interface to which Endpoint1 is linked has correctly been received. DATA PID is initialized to DATA0.

(4) UF0 EP2 status register L (UF0E2SL)

This register stores the value that is to be returned in response to the GET_STATUS Endpoint2 request.

This register can be read or written in 8-bit units. Note, however, that data can be written to this register only when the EP0NKA bit is set to 1.

If an error occurs in Endpoint2, the E2HALT bit is set to 1. A write access to this register is ignored while a USB-side access to Endpoint2 is being received.

The hardware automatically transmits the contents of this register to the host when it has received the GET_STATUS Endpoint2 request. If Endpoint2 has stalled, the UF0BO1 register is cleared and the BKO1NK bit of the UF0EN register is cleared to 0.

Because writing this register is always masked when transfer to Endpoint2, rather than control transfer, is executed, be sure to check this register to see if data has been correctly written to it.

Caution To rewrite this register, set the EP0NKA bit of the UF0E0NA register to 1 before reading the register contents, and rewrite the register contents after confirming that the bit has been set, in order to prevent conflict between a read access by SIE and a write access.

	7	6	5	4	3	2	1	0	Address	After reset
UF0E2SL	0	0	0	0	0	0	0	E2HALT	42A8H	00H
	0	0	0	0	0	0	0	R/W		

Bit position	Bit name	Function
0	E2HALT	This bit indicates the status of Endpoint2. 1: Stalled 0: Not stalled This bit is set to 1 by hardware when the SET_FEATURE Endpoint2 request has been received. It is cleared to 0 by hardware when the CLEAR_FEATURE Endpoint2 request, SET_CONFIGURATION request, or the SET_INTERFACE request for the Interface to which Endpoint2 is linked has correctly been received. DATA PID is initialized to DATA0.

(5) UF0 EP7 status register L (UF0E7SL)

This register stores the value that is to be returned in response to the GET_STATUS Endpoint7 request.

This register can be read or written in 8-bit units. Note, however, that data can be written to this register only when the EP0NKA bit is set to 1.

If an error occurs in Endpoint7, the E7HALT bit is set to 1. A write access to this register is ignored while a USB-side access to Endpoint7 is being received.

The hardware automatically transmits the contents of this register to the host when it has received the GET_STATUS Endpoint7 request. If Endpoint7 has stalled, the UF0INT1 register is cleared and the IT1NK bit of the UF0EN register is cleared to 0.

Because writing this register is always masked when transfer to Endpoint7, rather than control transfer, is executed, be sure to check this register to see if data has been correctly written to it.

Caution To rewrite this register, set the EP0NKA bit of the UF0E0NA register to 1 before reading the register contents, and rewrite the register contents after confirming that the bit has been set, in order to prevent conflict between a read access by SIE and a write access.

	7	6	5	4	3	2	1	0	Address	After reset
UF0E7SL	0	0	0	0	0	0	0	E7HALT	42D0H	00H
	0	0	0	0	0	0	0	R/W		

Bit position	Bit name	Function
0	E7HALT	This bit indicates the status of Endpoint7. 1: Stalled 0: Not stalled This bit is set to 1 by hardware when the SET_FEATURE Endpoint7 request has been received. It is cleared to 0 by hardware when the CLEAR_FEATURE Endpoint7 request, SET_CONFIGURATION request, or the SET_INTERFACE request for the Interface to which Endpoint7 is linked has correctly been received. DATA PID is initialized to DATA0.

(6) UF0 address register (UF0ADRS)

This register stores the device address.

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

The device address sent by the SET_ADDRESS request is analyzed and the resultant value is automatically written to this register. If the SET_ADDRESS request is processed by firmware, the value of this register is reflected as the device address when the SUCCESS signal is received in the status stage.

Caution Do not write to this register. Operation is not guaranteed if writing to this register is performed.

	7	6	5	4	3	2	1	0	Address	After reset
UF0ADRS	0	ADRS6	ADRS5	ADRS4	ADRS3	ADRS2	ADRS1	ADRS0	4300H	00H
	0	R	R	R	R	R	R	R		

Bit position	Bit name	Function
6 to 0	ADRS6 to ADRS0	These bits hold the device address of SIE.

(7) UF0 configuration register (UF0CNF)

This register stores the value that is to be returned in response to the GET_CONFIGURATION request.

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

When the SET_CONFIGURATION request is received, its wValue is automatically written to this register. Only 00H or 01H is taken as the wValue value. A STALL response is performed when a value other than 00H or 01H has been received.

When a change of the value of this register from 00H to other than 00H is detected, the CONF bits of the UF0MODS register are set to 1.

Caution Do not write to this register. Operation is not guaranteed if writing to this register is performed.

	7	6	5	4	3	2	1	0	Address	After reset
UF0CNF	0	0	0	0	0	0	0	CONF0	4304H	00H
	0	0	0	0	0	0	0	R		

Bit position	Bit name	Function
0	CONF0	These bits hold the data to be returned in response to the GET_CONFIGURATION request.

(8) UF0 interface 0 register (UF0IF0)

This register stores the value that is to be returned in response to the GET_INTERFACE wIndex = 0 request.

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

When the SET_INTERFACE request is received, its wValue is automatically written to this register.

If the SET_INTERFACE request is processed by firmware, wIndex and wValue are decoded, and the setting of endpoint is automatically changed. At this time, the status bit of the target endpoint and DPID are automatically cleared to 0, depending on the setting. The FIFO is not cleared automatically.

Caution Do not write to this register. Operation is not guaranteed if writing to this register is performed.

	7	6	5	4	3	2	1	0	Address	After reset
UF0IF0	0	0	0	0	0	IF02	IF01	IF00	4308H	00H
	0	0	0	0	0	R	R	R		

Bit position	Bit name	Function
2 to 0	IF02 to IF00	These bits hold the data to be returned in response to GET_INTERFACE wIndex = 0 request.

(9) UF0 interface 1 to 4 registers (UF0IF1 to UF0IF4)

These registers store the value that is to be returned in response to the GET_INTERFACE wIndex = n request (n = 1 to 4).

These registers are read-only, in 8-bit units.

When the SET_INTERFACE request is received, its wValue is automatically written to these registers.

These registers are invalidated according to the setting of the UF0AIFN and UF0AAS registers.

If the SET_INTERFACE request is processed by firmware, wIndex and wValue are decoded, and the setting of endpoint is automatically changed. At this time, the status bit of the target endpoint and DPID are automatically cleared to 0, depending on the setting. The FIFO is not cleared automatically.

Caution Do not write to this register. Operation is not guaranteed if writing to this register is performed.

	7	6	5	4	3	2	1	0	Address	After reset
UF0IF1	0	0	0	0	0	IF12	IF11	IF10	430CH	00H
	0	0	0	0	0	R	R	R		
UF0IF2	0	0	0	0	0	IF22	IF21	IF20	4310H	00H
	0	0	0	0	0	R	R	R		
UF0IF3	0	0	0	0	0	IF32	IF31	IF30	4314H	00H
	0	0	0	0	0	R	R	R		
UF0IF4	0	0	0	0	0	IF42	IF41	IF40	4318H	00H
	0	0	0	0	0	R	R	R		

Bit position	Bit name	Function
2 to 0	IFn2 to IFn0	These bits hold the data to be returned in response to GET_INTERFACE wIndex = n request.

Remark n = 1 to 4

(10) UF0 descriptor length register (UF0DSCL)

This register stores the length of the value that is to be returned in response to the GET_DESCRIPTOR Configuration request. The value of this register is the number of bytes of all the descriptors set by the UF0CIEn register minus 1 (n = 0 to 255). The total descriptor length that is to be returned in response to the GET_DESCRIPTOR Configuration request is determined according to the value of this register.

This register can be read or written in 8-bit units. However, data can be written to this register only when the EP0NKA bit is set to 1.

Processing of wLength is automatically controlled. If this register is set to 00H, it means that the descriptor to be returned is 1 byte long. If the register is set to FFH, a descriptor length of 256 bytes is returned. When a descriptor exceeding 256 bytes in length is used, set the CDCGDST bit of the UF0MODC register to 1 and process the GET_DESCRIPTOR request by firmware (at this time, the CDCGD bit of the UF0MODS register is also set to 1).

Caution To rewrite this register, set the EP0NKA bit of the UF0E0NA register to 1 before reading the register contents, and rewrite the register contents after confirming that the bit has been set, in order to prevent conflict between a read access by SIE and a write access.

	7	6	5	4	3	2	1	0	Address	After reset
UF0DSCL	DPL7	DPL6	DPL5	DPL4	DPL3	DPL2	DPL1	DPL0	4340H	00H
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

Bit position	Bit name	Function
7 to 0	DPL7 to DPL0	These bits set the value of the number of bytes of all the descriptors to be returned in response to the GET_DESCRIPTOR Configuration request minus 1.

(11) UF0 device descriptor registers 0 to 17 (UF0DD0 to UF0DD17)

These registers store the value to be returned in response to the GET_DESCRIPTOR Device request.

These registers can be read or written in 8-bit units. However, data can be written to these registers only when the EP0NKA bit is set to 1.

Cautions 1. To rewrite this register, set the EP0NKA bit of the UF0E0NA register to 1 before reading the register contents, and rewrite the register contents after confirming that the bit has been set, in order to prevent conflict between a read access by SIE and a write access.

2. Use the value defined by USB Specification Ver. 2.0 and the latest Class Specification as the set value.

	7	6	5	4	3	2	1	0	Address	After reset
UF0DDn (n = 0 to 17)									See **Table 3-12**.	Undefined
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

Table 3-12. Mapping and Data of UF0 Device Descriptor Registers

Symbol	Address	Field Name	Contents
UF0DD0	4344H	bLength	Size of this descriptor
UF0DD1	4348H	bDescriptorType	Device descriptor type
UF0DD2	434CH	bcdUSB	Value below decimal point of Rev. number of USB specification
UF0DD3	4350H		Value above decimal point of Rev. number of USB specification
UF0DD4	4354H	bDeviceClass	Class code
UF0DD5	4358H	bDeviceSubClass	Subclass code
UF0DD6	435CH	bDeviceProtocol	Protocol code
UF0DD7	4360H	bMaxPacketSize0	Maximum packet size of Endpoint0
UF0DD8	4364H	idVendor	Lower value of vendor ID
UF0DD9	4368H		Higher value of vendor ID
UF0DD10	436CH	idProduct	Lower value of product ID
UF0DD11	4370H		Higher value of product ID
UF0DD12	4374H	bcdDevice	Lower value of device release number
UF0DD13	4378H		Higher value of device release number
UF0DD14	437CH	iManufacturer	Index of string descriptor describing manufacturer
UF0DD15	4380H	iProduct	Index of string descriptor describing product
UF0DD16	4384H	lSerialNumber	Index of string descriptor describing device serial number
UF0DD17	4388H	BNumConfigurations	Number of settable configurations

(12) UF0 configuration/interface/endpoint descriptor registers 0 to 255 (UF0CIE0 to UF0CIE255)

These registers store the value to be returned in response to the GET_DESCRIPTOR Configuration request. These registers can be read or written in 8-bit units. However, data can be written to these registers only when the EP0NKA bit is set to 1.

Descriptor information of up to 256 bytes can be stored in these registers. Store each descriptor in the order of Configuration, Interface, and Endpoint (see **Table 3-13**). If there are two or more Interfaces, repeatedly store the data following the Interface descriptor.

Table 3-13. Mapping of UF0CIEn Register

Address	Descriptor Stored
438CH to 43ACH	Configuration descriptor (9 bytes)
43B0H to 43D0H	Interface descriptor (9 bytes)
43D4H to 43ECH	Endpoint1 descriptor (7 bytes)
43F0H to 4408H	Endpoint2 descriptor (7 bytes)
:	:
xxxxH	Interface descriptor (9 bytes)
xxxxH + 9	Endpoint1 descriptor (7 bytes)
xxxxH + 16	Endpoint2 descriptor (7 bytes)
:	:

The range of the valid data that can be set to these registers varies according to the setting of the UF0DSCL register. In addition to the descriptors listed in Table 3-14, descriptors peculiar to classes and vendors can also be stored.

Cautions 1. To rewrite this register, set the EP0NKA bit of the UF0E0NA register to 1 before reading the register contents, and rewrite the register contents after confirming that the bit has been set, in order to prevent conflict between a read access by SIE and a write access.

2. Use the value defined by USB Specification Ver. 2.0 and the latest Class Specification as the set value.

	7	6	5	4	3	2	1	0	Address	After reset
UF0CIEn (n = 0 to 255)									438CH to 4788H	Undefined
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W		

Table 3-14. Data of UF0CIEn Register

(a) Configuration descriptor (9 bytes)

Offset	Field Name	Contents
0	bLength	Size of this descriptor
1	bDescriptorType	Descriptor type
2	wTotalLength	Lower value of the total number of bytes of Configuration, all Interface, and all Endpoint descriptors
3		Higher value of the total number of bytes of Configuration, all Interface, and all Endpoint descriptors
4	bNumInterface	Number of Interfaces
5	bConfigurationValue	Value to select this Configuration
6	iConfiguration	Index of string descriptor describing this Configuration
7	bmAttributes	Features of this Configuration (self-powered, without remote wakeup)
8	MaxPower	Maximum power consumption of this Configuration (unit: mA)

(b) Interface descriptor (9 bytes)

Offset	Field Name	Contents
0	bLength	Size of this descriptor
1	bDescriptorType	Descriptor type
2	bInterfaceNumber	Value of this Interface
3	bAlternateSetting	Value to select alternate setting of Interface
4	bNumEndpoints	Number of usable Endpoints
5	bInterfaceClass	Class code
6	bInterfaceSubClass	Subclass code
7	bInterfaceProtocol	Protocol code
8	Interface	Index of string descriptor describing this Interface

(c) Endpoint descriptor (7 bytes)

Offset	Field Name	Contents
0	bLength	Size of this descriptor
1	bDescriptorType	Descriptor type
2	bEndpointAddress	Address/transfer direction of this Endpoint
3	bmAttributes	Transfer type
4	wMaxPaketSize	Lower value of maximum number of transfer data
5		Higher value of maximum number of transfer data
6	bInterval	Transfer interval

The following flowcharts illustrate the program execution when the host is disconnected and then reconnected, and the program execution when power is supplied.

Figure 3-7. Flowchart of Program When Host Is Disconnected and Then Reconnected

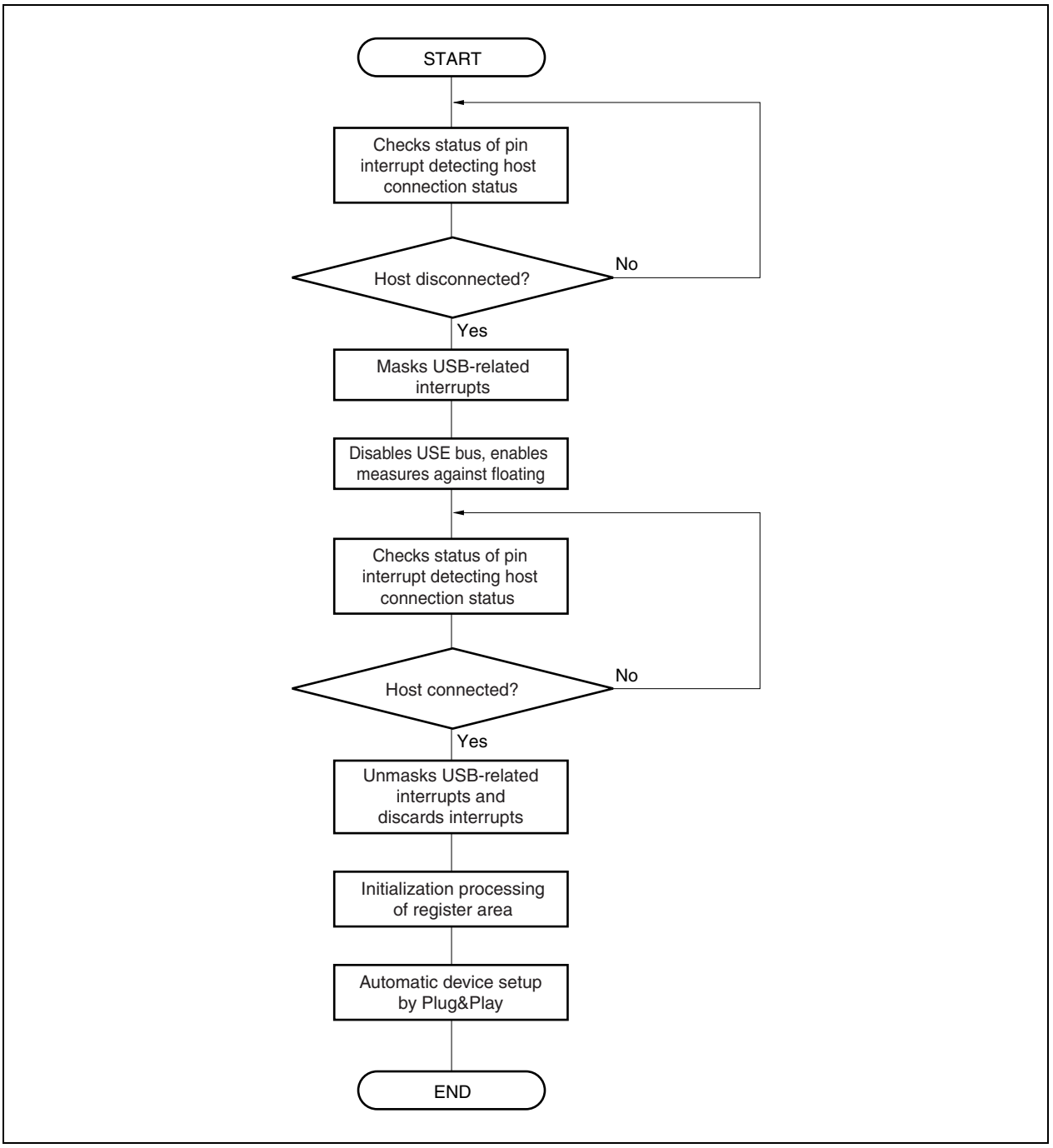
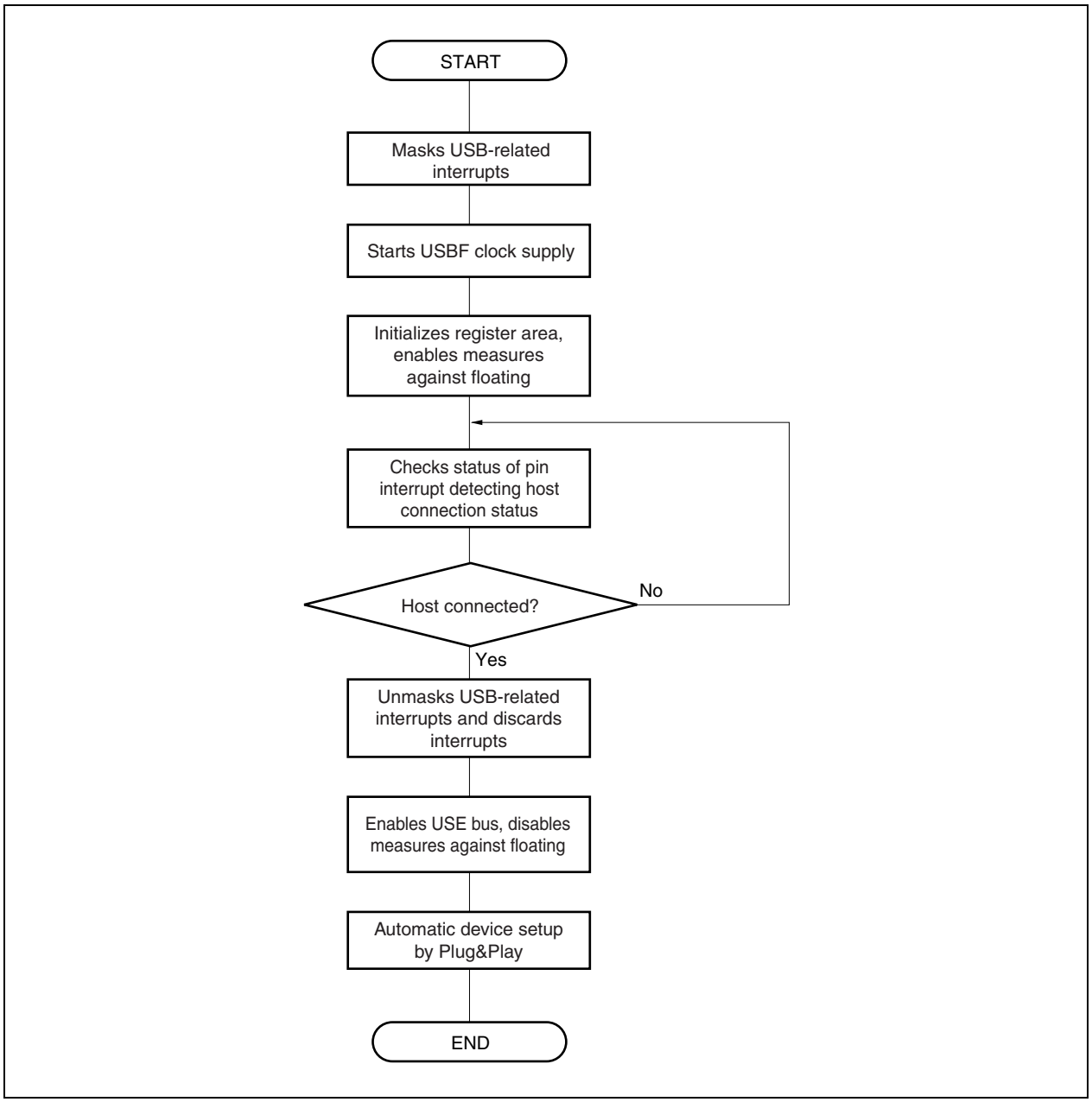


Figure 3-8. Flowchart of Program When Power Is Supplied



3.5 STALL Handshake or No Handshake

Errors of USBF are defined to be handled as follows.

Table 3-15. Errors of USB Function

Transfer Type	Transaction	Target Packet	Error Type	Function Response	Processing
Control transfer/ bulk transfer/ interrupt transfer	IN/OUT/SETUP	Token	Endpoint not supported	No response	None
			Endpoint transfer direction mismatch	No response	None
			CRC error	No response	None
			Bit stuffing error	No response	None
Control transfer/ bulk transfer	OUT/SETUP	Data	Timeout	No response	None
			PID check error	No response	None
			Unsupported PID (other than Data PID)	No response	None
			CRC error	No response	Discard received data
			Bit stuffing error	No response	Discard received data
	OUT	Data	Data PID mismatch	ACK	Discard received data
Control transfer (SETUP stage)	SETUP	Data	Overrun	No response	Discard received data
Control transfer (data stage)	OUT	Data	Overrun	No response[Note 1]	Set SNDSTL bit of UF0SDS register to 1 and discard received data
Control transfer (status stage)	OUT	Data	Overrun	ACK or no response[Note 2]	Set SNDSTL bit of UF0SDS register to 1 and discard received data
Bulk transfer	OUT	Data	Overrun	No response[Note 1]	Set EnHALT bit of UF0EnSL register (n = 0 to 2, 7) to 1
Control transfer/ bulk transfer/ interrupt transfer	IN	Handshake	PID check error	–	Hold transferred data and re-transfer data[Note 3]
			Unsupported PID (other than ACK PID)	–	Hold transferred data and re-transfer data[Note 3]
			Timeout	–	Hold transferred data and re-transfer data[Note 3]

Notes 1. A STALL response is made to re-transfer by the host.

2. An ACK response is made if the transfer data is of less than MaxPacketSize and the data received in the status stage is discarded. If MaxPacketSize is exceeded, no response is made, the SNDSTL bit of the UF0SDS register is set to 1, and the received data is discarded.

3. If an OUT transaction indicating a change from the data stage to the status stage is received during control transfer, an error is not handled and it is assumed that reception has been correctly completed.

Cautions 1. It is judged by the Alternate Setting number currently set whether the target Endpoint is valid or invalid.

2. For the response to the request included in control transfer to/from Endpoint0, see 3.3 Requests.

3.6 Register Values in Specific Status

Table 3-16. Register Values in Specific Status (1/2)

Register Name	After CPU Reset	After Bus Reset
UF0E0N register	00H	Value is held.
UF0E0NA register	00H	Value is held.
UF0EN register	00H	Value is held.
UF0ENM register	00H	Value is held.
UF0SDS register	00H	Value is held.
UF0CLR register	00H	Value is held.
UF0SET register	00H	Value is held.
UF0EPS0 register	00H	Value is held.
UF0EPS1 register	00H	Value is held.
UF0EPS2 register	00H	Value is held.
UF0IS0 register	00H	Value is held.
UF0IS1 register	00H	Value is held.
UF0IS2 register	00H	Value is held.
UF0IS3 register	00H	Value is held.
UF0IS4 register	00H	Value is held.
UF0IM0 register	00H	Value is held.
UF0IM1 register	00H	Value is held.
UF0IM2 register	00H	Value is held.
UF0IM3 register	00H	Value is held.
UF0IM4 register	00H	Value is held.
UF0IC0 register	FFH	Value is held.
UF0IC1 register	FFH	Value is held.
UF0IC2 register	FFH	Value is held.
UF0IC3 register	FFH	Value is held.
UF0IC4 register	FFH	Value is held.
UF0IDR register	00H	Value is held.
UF0DMS0 register	00H	Value is held.
UF0DMS1 register	00H	Value is held.
UF0FIC0 register	00H	Value is held.
UF0FIC1 register	00H	Value is held.
UF0DEND register	00H	Value is held.
UF0GPR register	00H	Value is held.
UF0MODC register	00H	Value is held.
UF0MODS register	00H	Bit 2 (CONF): Cleared (0), Other bits: Value is held.
UF0AIFN register	00H	Value is held.
UF0AAS register	00H	Value is held.
UF0ASS register	00H	00H
UF0E1IM register	00H	Value is held.
UF0E2IM register	00H	Value is held.

Table 3-16. Register Values in Specific Status (2/2)

Register Name	After CPU Reset	After Bus Reset
UF0E3IM register	00H	Value is held.
UF0E4IM register	00H	Value is held.
UF0E7IM register	00H	Value is held.
UF0E8IM register	00H	Value is held.
UF0E0R register	Undefined[Note 1]	Value is held.
UF0E0L register	00H	Value is held.
UF0E0ST register	00H	00H
UF0E0W register	Undefined[Note 1]	Value is held.
UF0BO1 register	Undefined[Note 1]	Value is held.
UF0BO1L register	00H	Value is held.
UF0BO2 register	Undefined[Note 1]	Value is held.
UF0BO2L register	00H	Value is held.
UF0BI1 register	Undefined[Note 1]	Value is held.
UF0BI2 register	Undefined[Note 1]	Value is held.
UF0INT1 register	Undefined	Value is held.
UF0INT2 register	Undefined	Value is held.
UF0DSTL register	00H	00H
UF0E0SL register	00H	00H
UF0E1SL register	00H	00H
UF0E2SL register	00H	00H
UF0E3SL register	00H	00H
UF0E4SL register	00H	00H
UF0E7SL register	00H	00H
UF0E8SL register	00H	00H
UF0ADRS register	00H	00H
UF0CNF register	00H	00H
UF0IF0 register	00H	00H
UF0IF1 register	00H	00H
UF0IF2 register	00H	00H
UF0IF3 register	00H	00H
UF0IF4 register	00H	00H
UF0DSCL register	00H	Value is held.
UF0DDn register (n = 0 to 17)	**Note 2**	**Note 2**
UF0CIEn register (n = 0 to 255)	**Note 2**	**Note 2**

Notes 1. This register can be cleared by the RESET signal because its write pointer, counter, and read pointer are cleared when the RESET signal becomes active, in the same manner as clearing by the UF0FICn register, as the register is controlled by FIFO.

2. This register cannot be cleared. Because data can be written to it by firmware, however, any value can be written to the register (before doing so, however, be sure to set the EP0NKA bit of the UF0E0NA register to 1).

3.7 Firmware Processing

The following firmware processing is performed.

- Setting processing on device side for the SET_CONFIGURATION, SET_INTERFACE, SET_FEATURE, and CLEAR_FEATURE requests during enumeration processing
- Analysis and processing of XXXXStandard, XXXXClass, and XXXXVendor requests not subject to automatic processing
- Reading data following bulk-transferred OUT token from receive buffer
- Writing data to be returned in response to bulk-transferred IN token
- Writing data to be returned in response to interrupt-transferred token

The following table lists the requests supported by firmware.

Table 3-17. Firmware-Supported Standard Requests

Request	Reception Side	Processing/ Frequency	Explanation
CLEAR_FEATURE	Interface	Automatic STALL response	It is considered that this request does not come to Interface because there is no function selector value, though it is reserved for bmRequestType. When this request is received, the hardware makes an automatic STALL response.
SET_FEATURE	Interface	Automatic STALL response	It is considered that this request does not come to Interface because there is no function selector value, though it is reserved for bmRequestType. When this request is received, the hardware makes an automatic STALL response.
GET_DESCRIPTOR	String	Firmware	Returns the string descriptor. When this request is received by the SETUP token, the hardware generates the CPUDEC interrupt request for firmware. Ffirmware decodes the contents of the request from the CPUDEC interrupt request, and writes the data to be returned to the host, to the UF0E0W register.
SET_DESCRIPTOR	Device	Firmware	Rewrites the device descriptor. When this request is received by the SETUP token, the hardware generates the CPUDEC interrupt request for firmware. Firmware decodes the contents of the request from the CPUDEC interrupt request, and the writes the data for the next control transfer (OUT) to the UF0DDn register (n = 0 to 17).
SET_DESCRIPTOR	Configuration	Firmware	Rewrites the configuration descriptor. When this request is received by the SETUP token, the hardware generates the CPUDEC interrupt request for firmware. Firmware decodes the contents of the request from the CPUDEC interrupt request, and the writes the data for the next control transfer (OUT) to the UF0CIEn register (n = 0 to 255).
SET_DESCRIPTOR	String	Firmware	Rewrites the string descriptor. When this request is received by the SETUP token, the hardware generates the CPUDEC interrupt request for firmware. Firmware decodes the contents of the request from the CPUDEC interrupt request, and loads the data for the next control transfer (OUT).
Other	NA	Firmware	When this request is received by the SETUP token, the hardware generates the CPUDEC interrupt request for firmware. Firmware decodes the contents of the request from the CPUDEC interrupt request, and performs the necessary processing.

3.7.1 Initialization processing

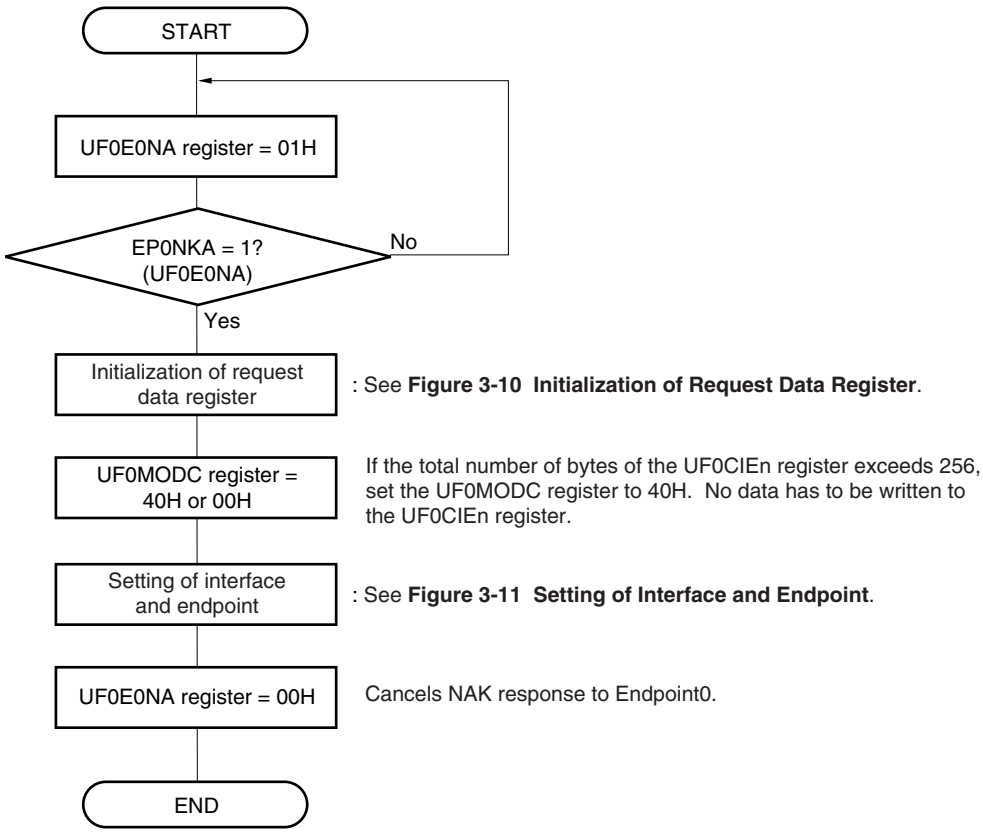
Initialization processing is executed in the following two ways.

- Initialization of request data register
- Setting of interrupt

When the request data register is initialized, data for the GET_XXXX request to which a value is to be automatically returned is written and an endpoint is allocated to an interface. In the interrupt settings, the interrupt sources that do not have to be checked can be masked by using the UF0IMn register (n = 0 to 4).

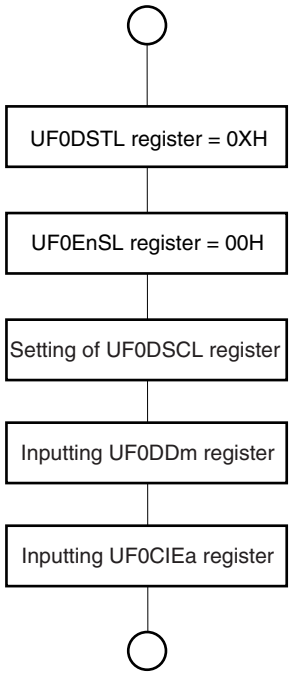
The following flowcharts illustrate the above processing.

Figure 3-9. Initializing Request Data Register



Remark n = 0 to 255

Figure 3-10. Initialization of Request Data Register



The value of 0XH depends on the power supply method.
- SFPW = 1: Self-powered
- SFPW = 0: Bus-powered

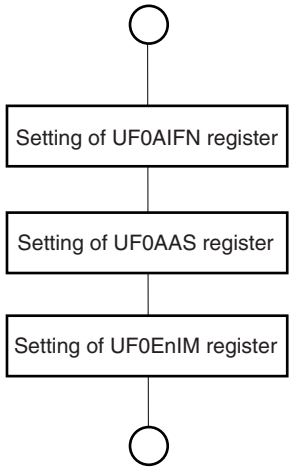
n = 0 to 2, 7. Setting is unnecessary if the target endpoint is not used.

Input the total number of bytes of the UF0CIEa register.

If the total number of bytes of the UF0CIEa register exceeds 256, set the UF0MODC register to 40H. No data has to be written to the UF0CIEa register.

Remark m = 0 to 17
a = 0 to 255

Figure 3-11. Setting of Interface and Endpoint



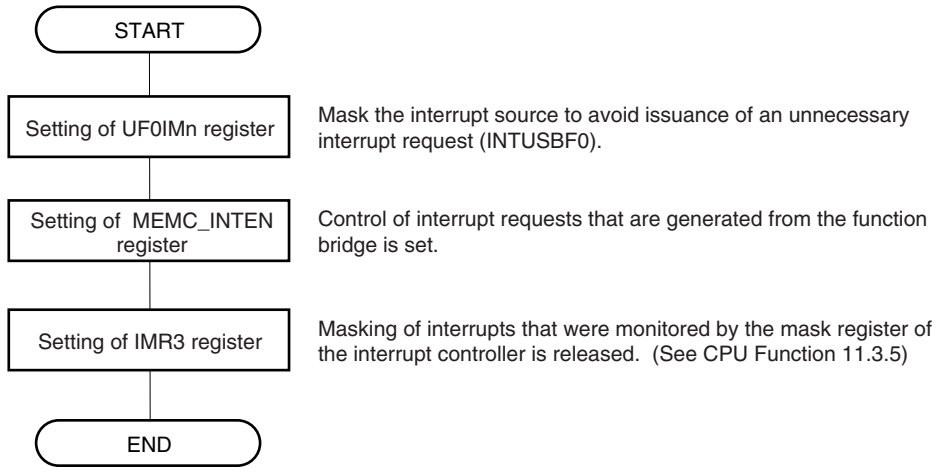
ADDIF, IFNO1, IFNO0 = 000: Interface number 0 is valid.
ADDIF, IFNO1, IFNO0 = 100: Interface numbers 0 and 1 are valid.
ADDIF, IFNO1, IFNO0 = 101: Interface numbers 0 to 2 are valid.
ADDIF, IFNO1, IFNO0 = 110: Interface numbers 0 to 3 are valid.
ADDIF, IFNO1, IFNO0 = 111: Interface numbers 0 to 4 are valid.

Set Interface number(s) and a link with the 5- or 2-series Alternate Setting.

Set a link between the target Interface of endpoint n and Alternate Setting.
Set 00H if the target endpoint is not used.

Remark n = 1, 2, 7

Figure 3-12. Setting of Interrupt

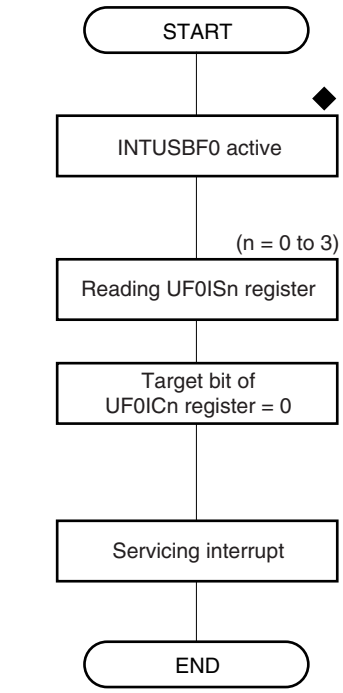


Remark n = 0 to 4

3.7.2 Interrupt servicing

The following flowchart illustrates how an interrupt is serviced.

Figure 3-13. Interrupt Servicing



Remark ♦: Processing by hardware

The following bits of the UF0ISn register are automatically cleared by hardware when a given condition is satisfied (n = 1 to 4).

- E0INDT, E0ODT, SUCES, STG, and CPUDEC bits of UF0IS1 register
- BKI2DT, BKI1DT, IT2DT, and IT1DT bits of UF0IS2 register
- BKO2FL, BKO2DT, BKO1FL, and BKO1DT bits of UF0IS3 register

Because clearing an interrupt source by the UF0ICn register is given a lower priority than setting an interrupt source by hardware, the interrupt source may not be cleared depending on the timing (n = 0 to 4).

3.7.3 USB function main processing

USB function main processing involves processing USB transactions. The types of transactions to be processed are as follows.

- Fully automatically processed request for control transfer
- Automatically processed requests for control transfer
  (SET_CONFIGURATION, SET_INTERFACE, SET_FEATURE, CLEAR_FEATURE)
- CPUDEC request for control transfer
- Processing for bulk transfer (IN)
- Processing for bulk transfer (OUT)
- Processing for interrupt transfer (IN)

Processing for endpoint n involves writing or reading for data transfer. The flowchart shown below is for PIO.

(1) Fully automatically processed request for control transfer

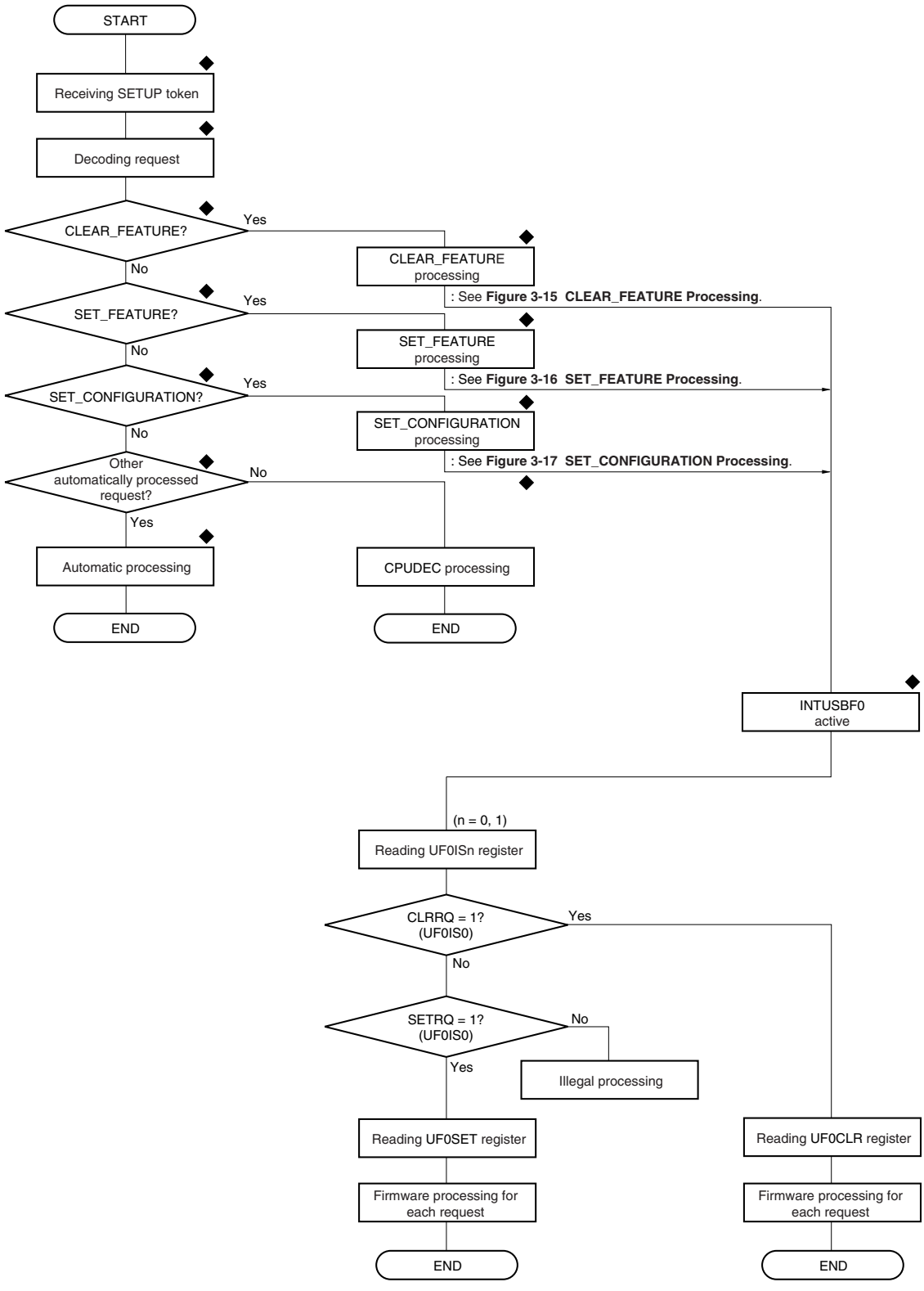
Because the fully automatically processed request for control transfer is executed by hardware, it cannot be referenced by firmware. Therefore, firmware does not have to perform any special processing for this request.

(2) Automatically processed requests for control transfer
(SET_CONFIGURATION, SET_INTERFACE, SET_FEATURE, CLEAR_FEATURE)

Processing to write a register for automatically processed requests for control transfer, such as SET_CONFIGURATION, SET_INTERFACE, SET_FEATURE, and CLEAR_FEATURE requests, is automatically executed by hardware, but an interrupt request is issued for recognition on the device side. This processing may be ignored if there is no special processing to be executed.

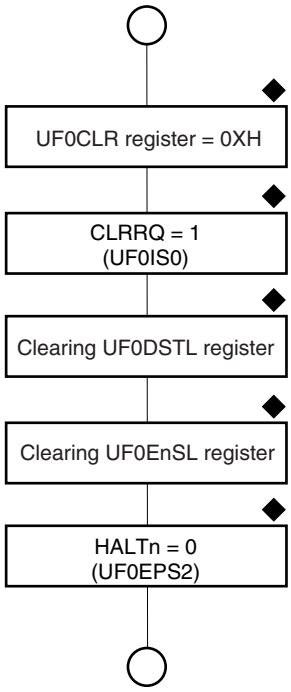
The flowcharts are shown below.

Figure 3-14. Automatically Processed Requests for Control Transfer



Remark ♦: Processing by hardware

Figure 3-15. CLEAR_FEATURE Processing

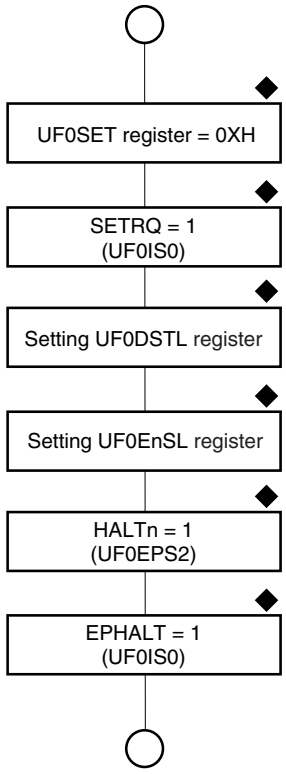


Set the corresponding bit for the value of 0XH.
The EPHALT bit of the UF0IS0 register is cleared to 0 only when all Halt Features are cleared.

Remarks 1. n = 0 to 2, 7

2. ♦: Processing by hardware

Figure 3-16. SET_FEATURE Processing

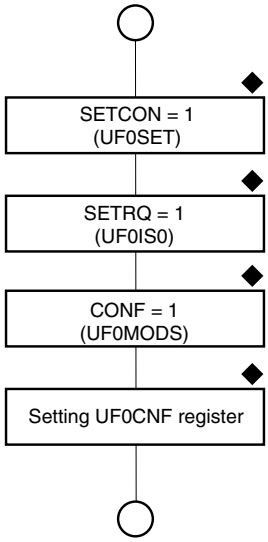


Set the corresponding bit for the value of 0XH. The EPHALT bit of the UF0IS0 register is not set to 1 by setting the UF0DSTL register.

Remarks 1. n = 0 to 2, 7

2. ♦: Processing by hardware

Figure 3-17. SET_CONFIGURATION Processing



Remark ♦: Processing by hardware

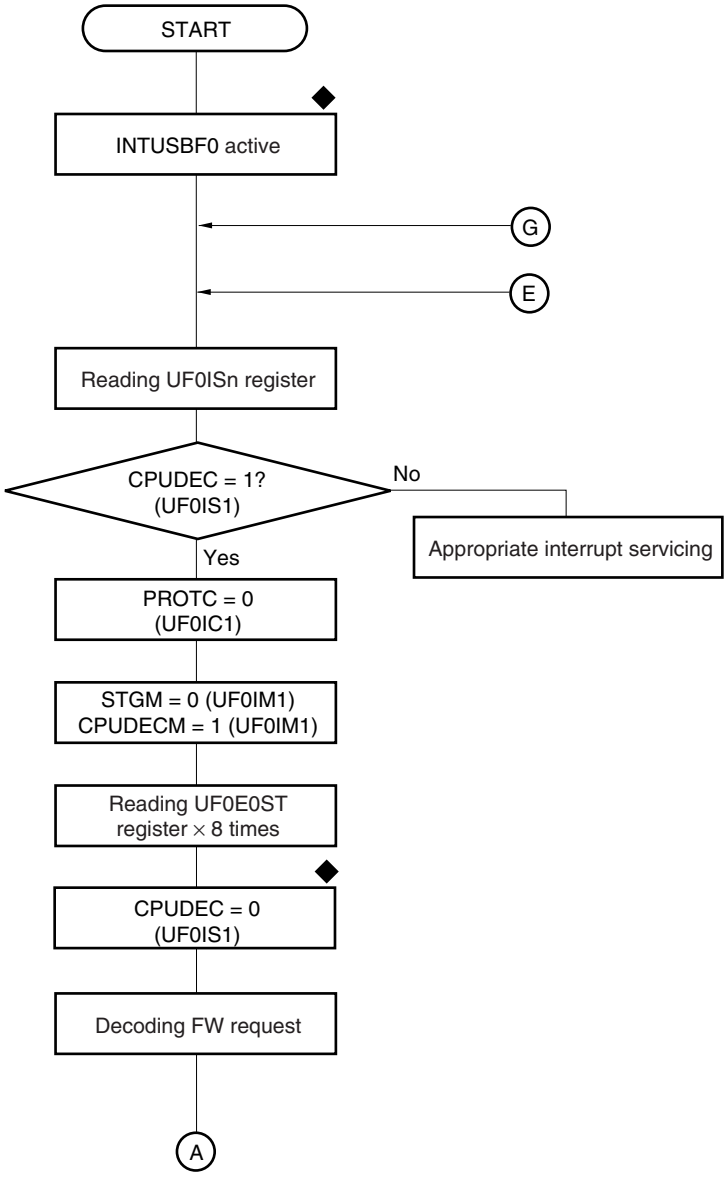
(3) CPUDEC request for control transfer

The CPUDEC request can be classified into three types of processing: control transfer (write), control transfer (read), and control transfer (without data). Control transfer (write) indicates a request that uses the OUT transaction in the data stage (e.g., SET_DESCRIPTOR), and control transfer (read) indicates a request that uses the IN transaction in the data stage (e.g., GET_DESCRIPTOR). Control transfer (without data) indicates a request that has no data stage (e.g., SET_CONFIGURATION).

The flowcharts are shown below.

Figure 3-18. CPUDEC Request for Control Transfer (1/12)

(a) Token phase (1/2)

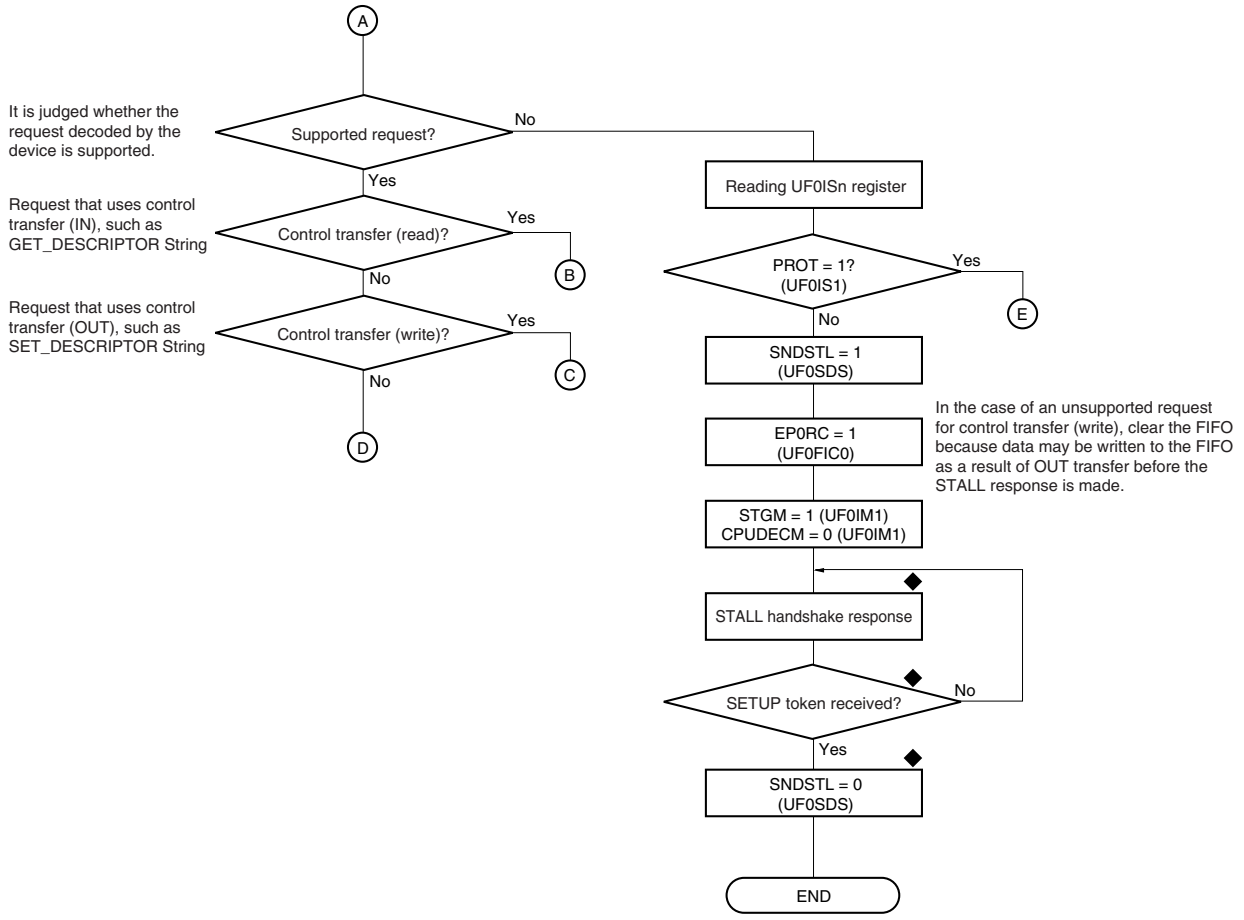


Remarks 1. n = 0, 1

2. ♦: Processing by hardware

Figure 3-18. CPUDEC Request for Control Transfer (2/12)

(a) Token phase (2/2)

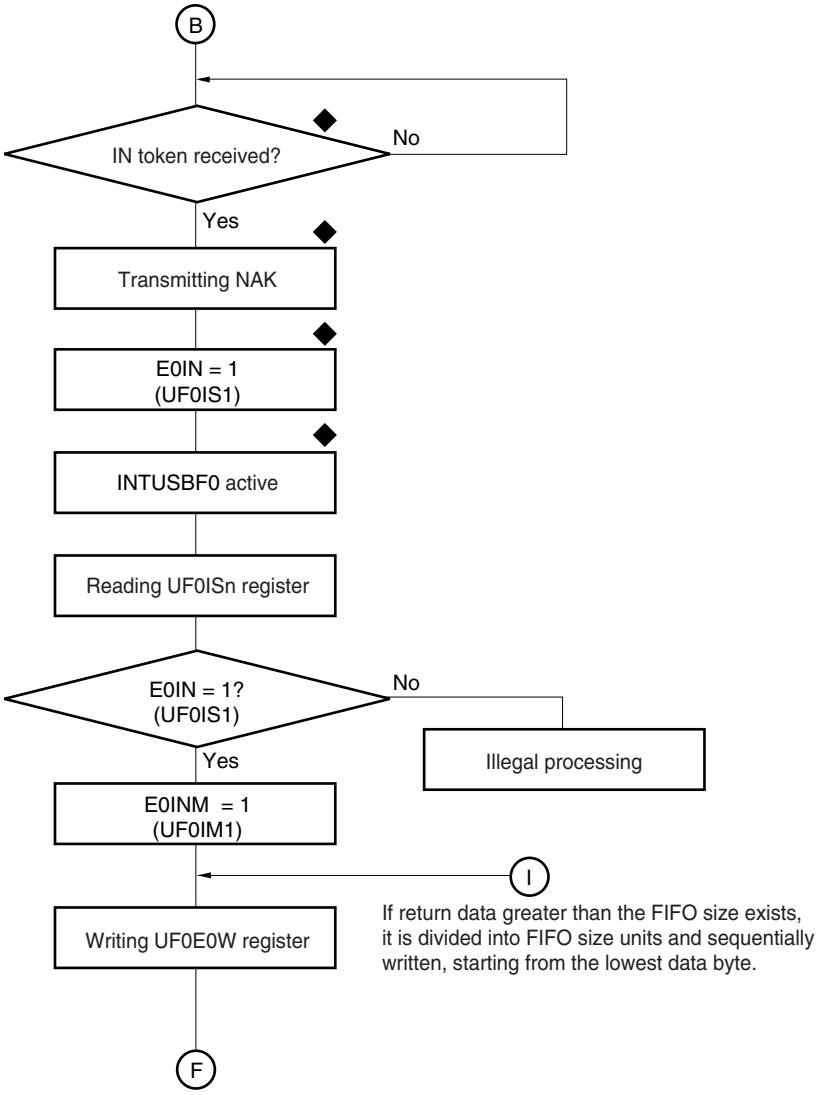


Remarks 1. n = 0, 1

2. ♦: Processing by hardware

Figure 3-18. CPUDEC Request for Control Transfer (3/12)

(b) Control transfer (read) (1/4)

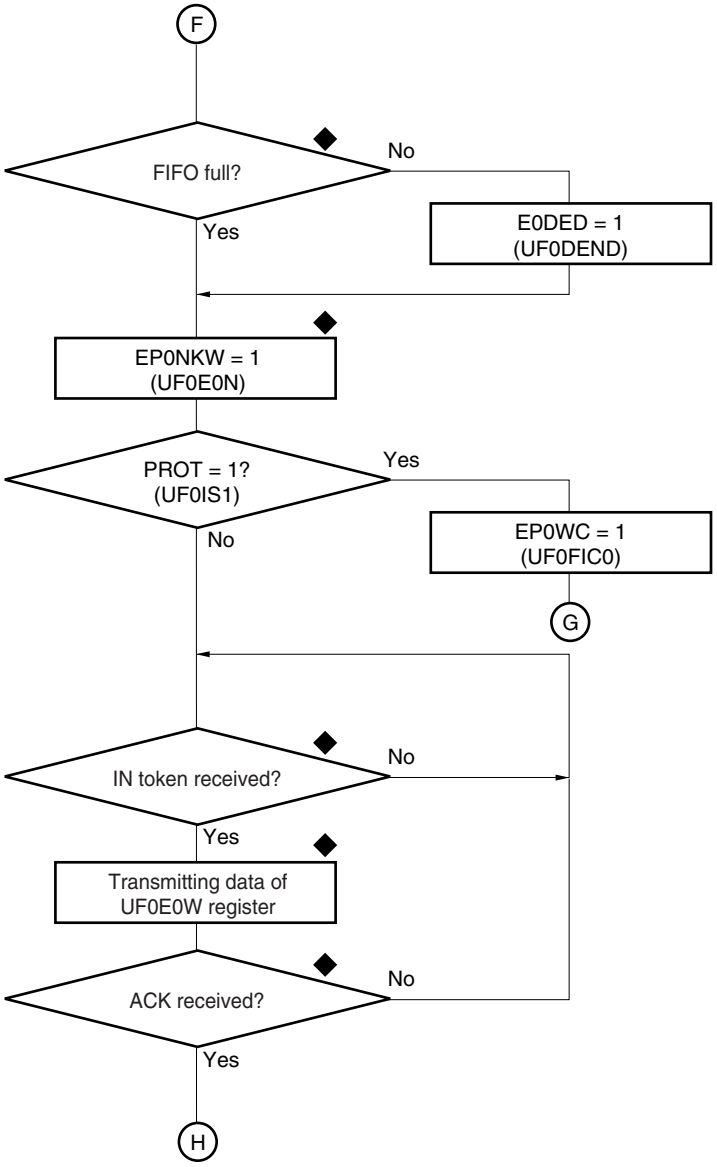


Remarks 1. n = 0, 1

2. ♦: Processing by hardware

Figure 3-18. CPUDEC Request for Control Transfer (4/12)

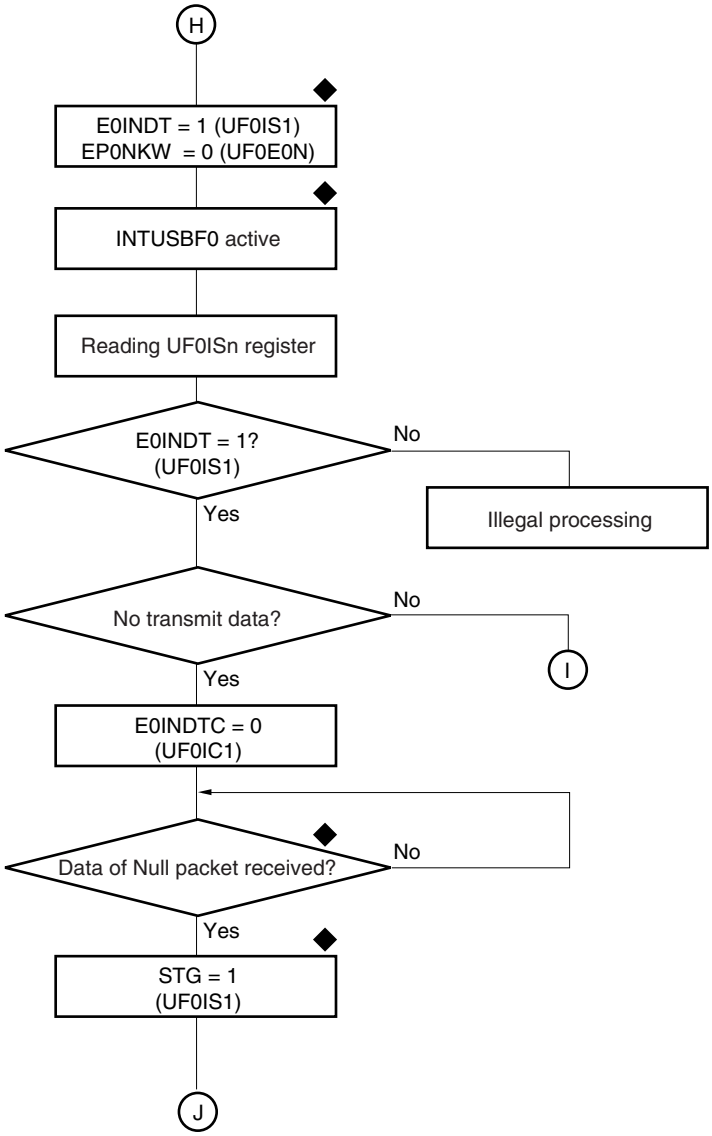
(b) Control transfer (read) (2/4)



Remark ♦: Processing by hardware

Figure 3-18. CPUDEC Request for Control Transfer (5/12)

(b) Control transfer (read) (3/4)

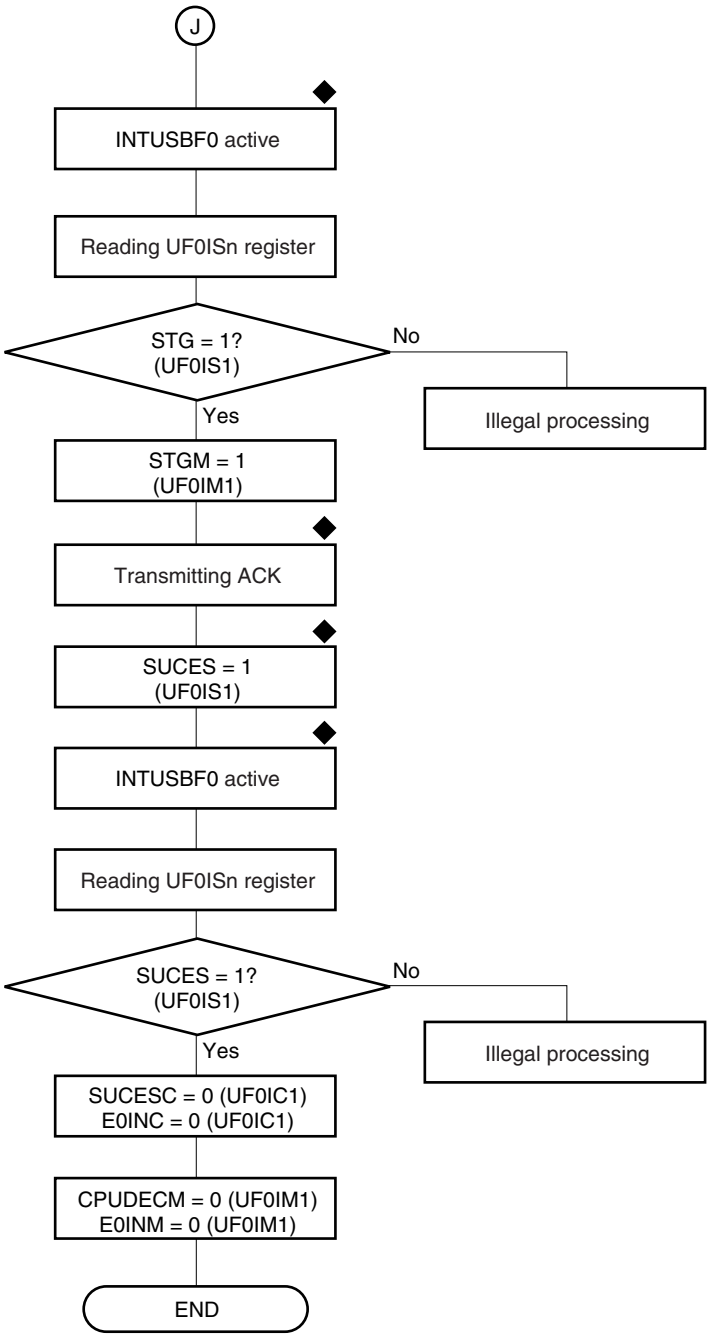


Remarks 1. n = 0, 1

2. ♦: Processing by hardware

Figure 3-18. CPUDEC Request for Control Transfer (6/12)

(b) Control transfer (read) (4/4)

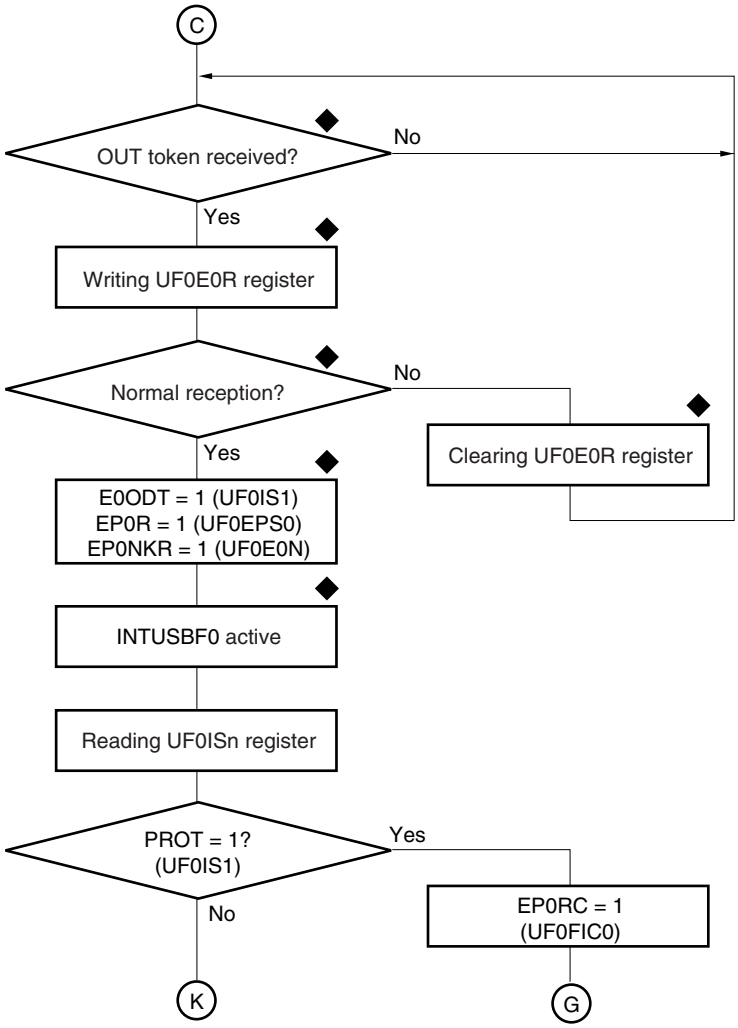


Remarks 1. n = 0, 1

2. ♦: Processing by hardware

Figure 3-18. CPUDEC Request for Control Transfer (7/12)

(c) Control transfer (write) (1/4)

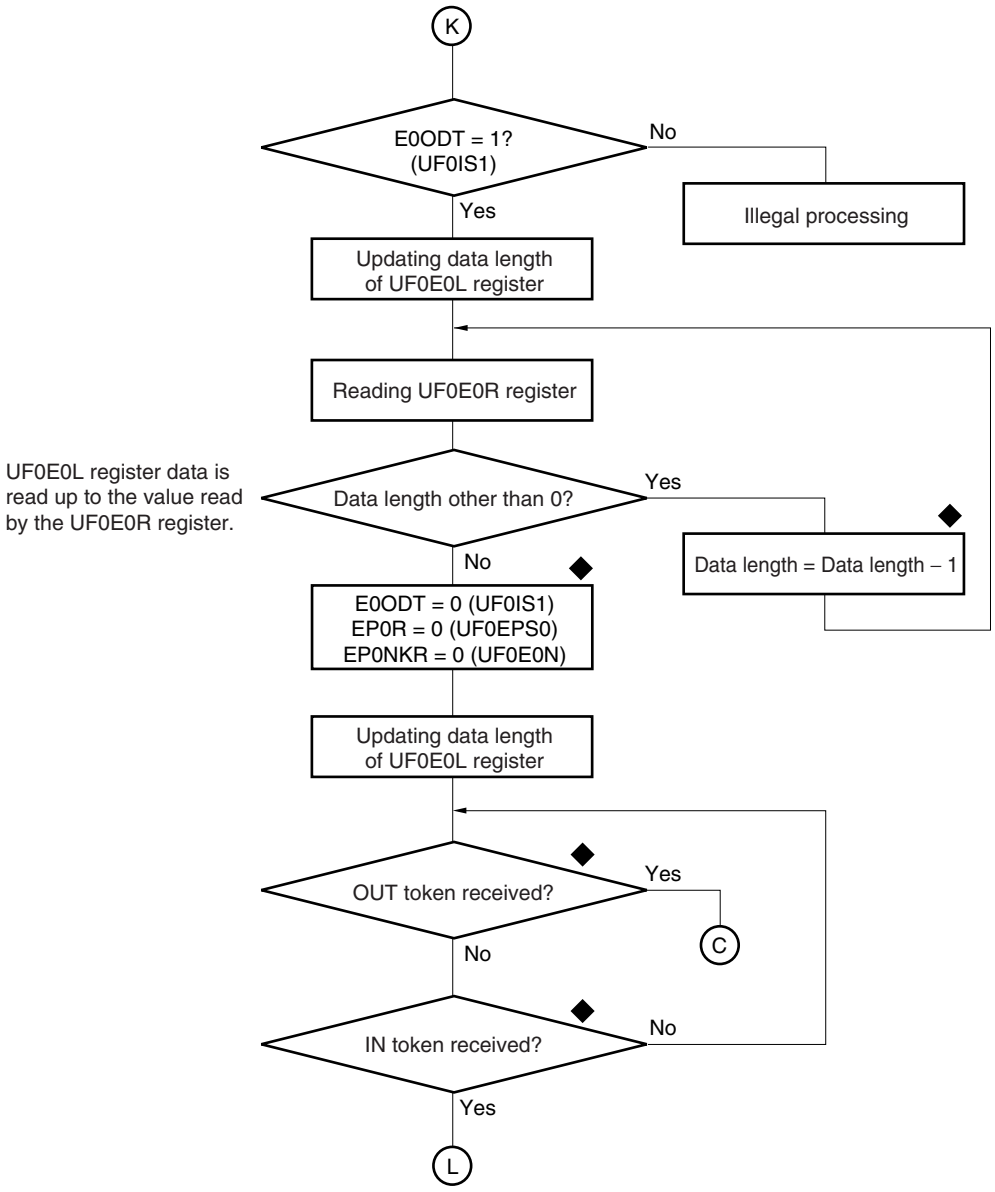


Remarks 1. n = 0, 1

2. ♦: Processing by hardware

Figure 3-18. CPUDEC Request for Control Transfer (8/12)

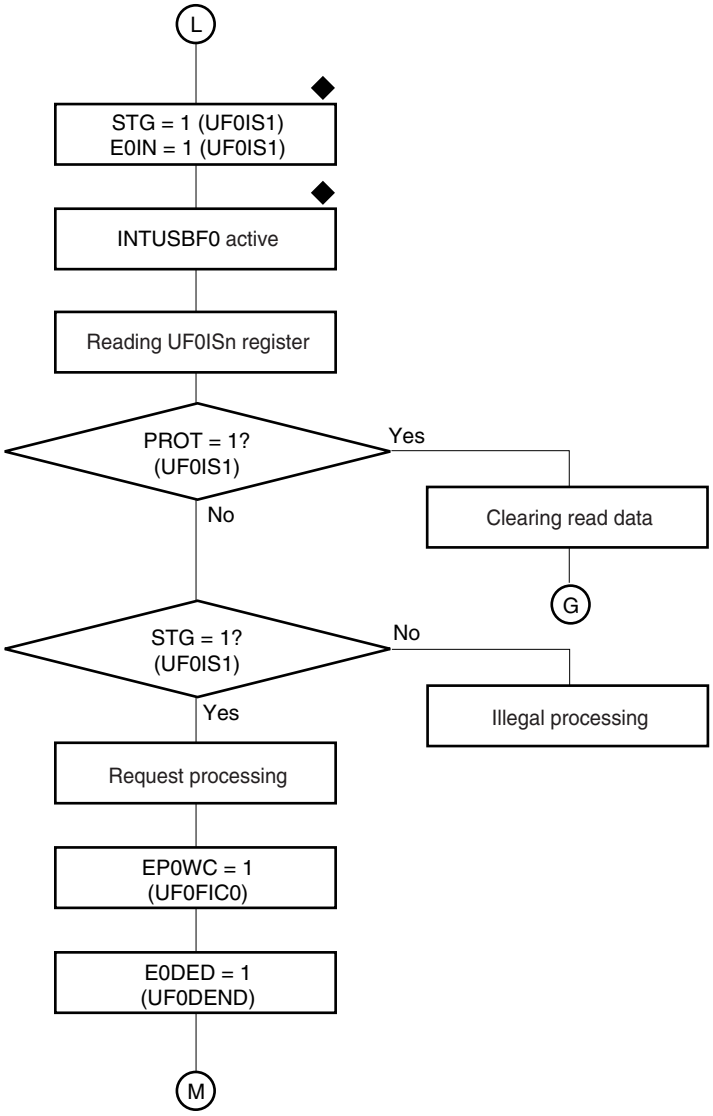
(c) Control transfer (write) (2/4)



Remark ♦: Processing by hardware

Figure 3-18. CPUDEC Request for Control Transfer (9/12)

(c) Control transfer (write) (3/4)

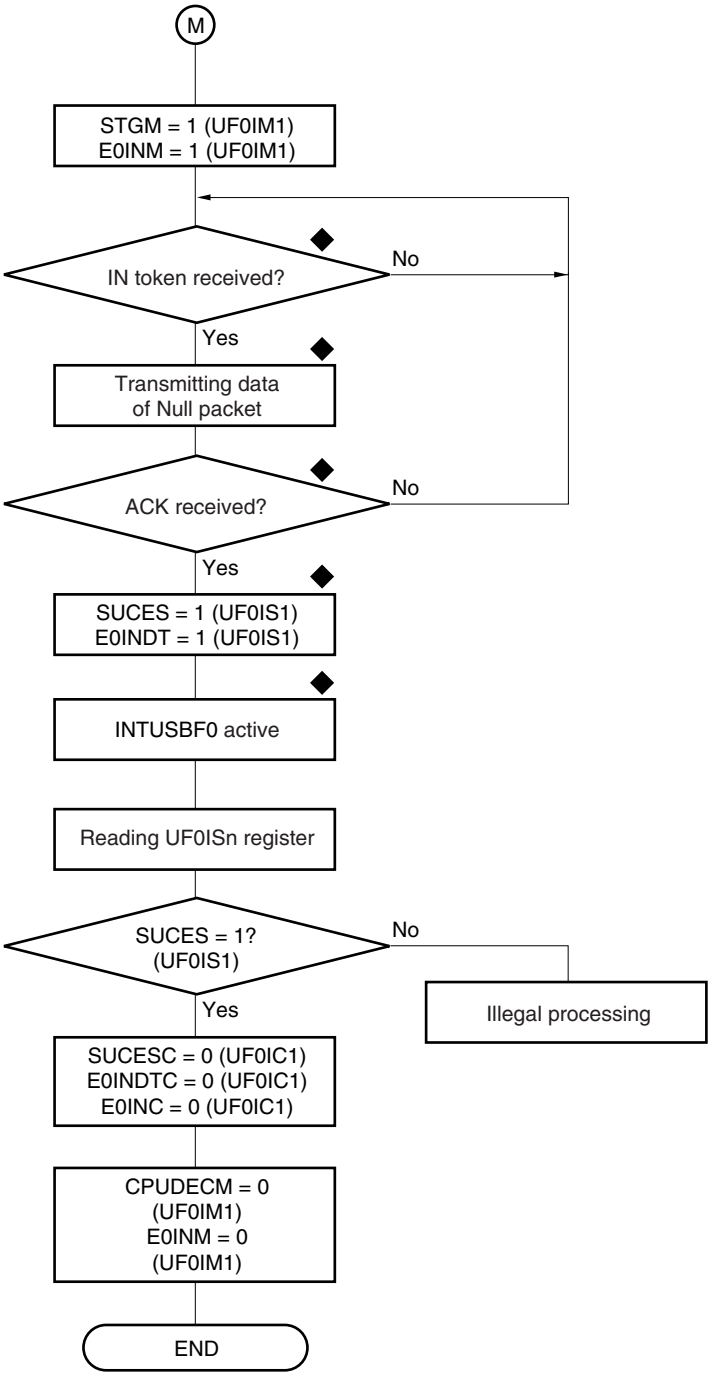


Remarks 1. n = 0, 1

2. ♦: Processing by hardware

Figure 3-18. CPUDEC Request for Control Transfer (10/12)

(c) Control transfer (write) (4/4)

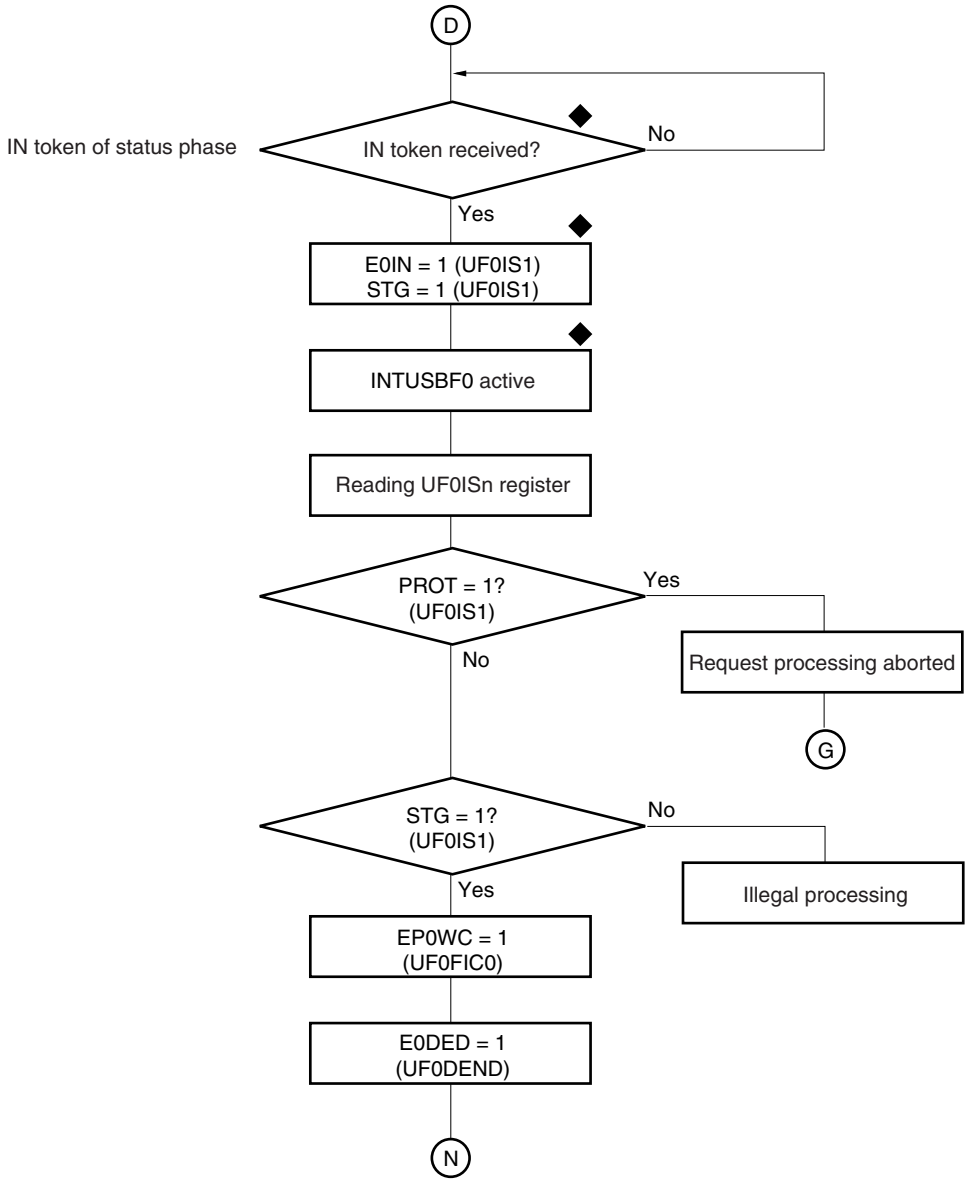


Remarks 1. n = 0, 1

2. ♦: Processing by hardware

Figure 3-18. CPUDEC Request for Control Transfer (11/12)

(d) Control transfer (without data stage) (1/2)

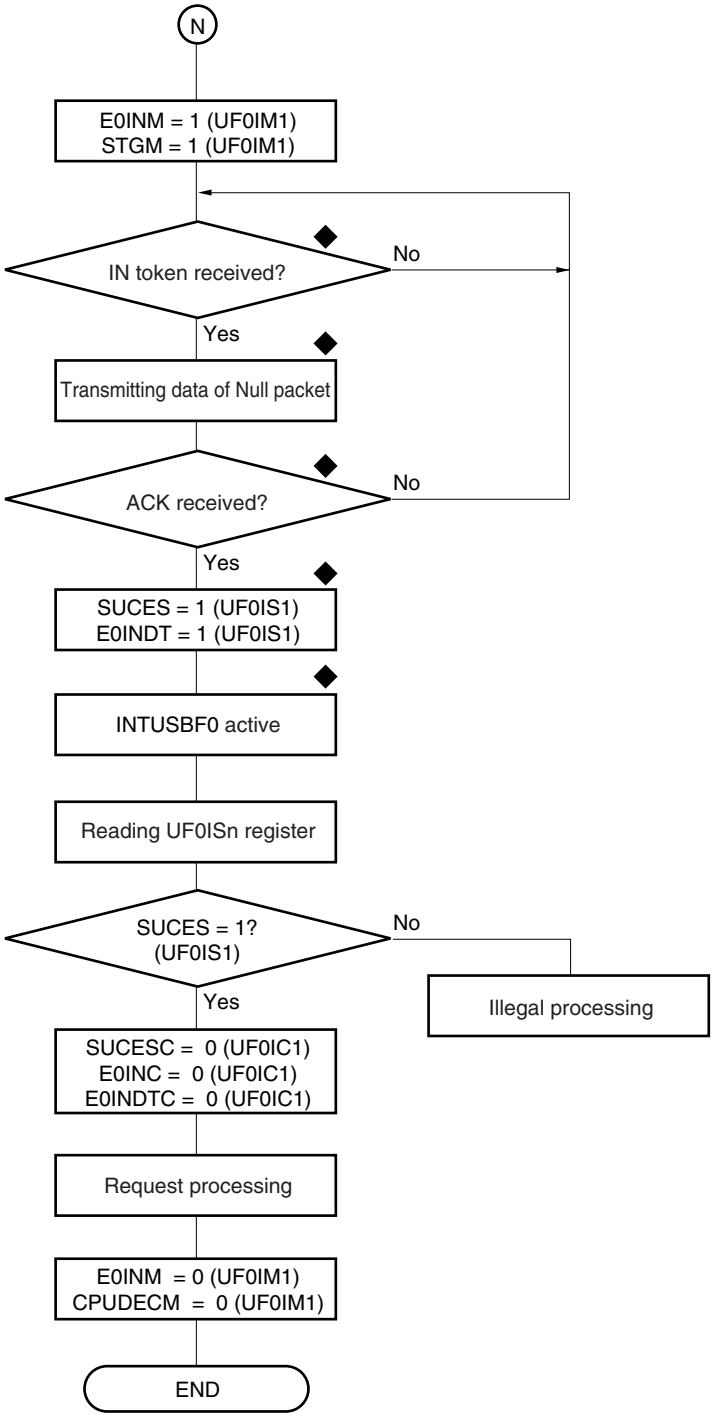


Remarks 1. n = 0, 1

2. ♦: Processing by hardware

Figure 3-18. CPUDEC Request for Control Transfer (12/12)

(d) Control transfer (without data stage) (2/2)



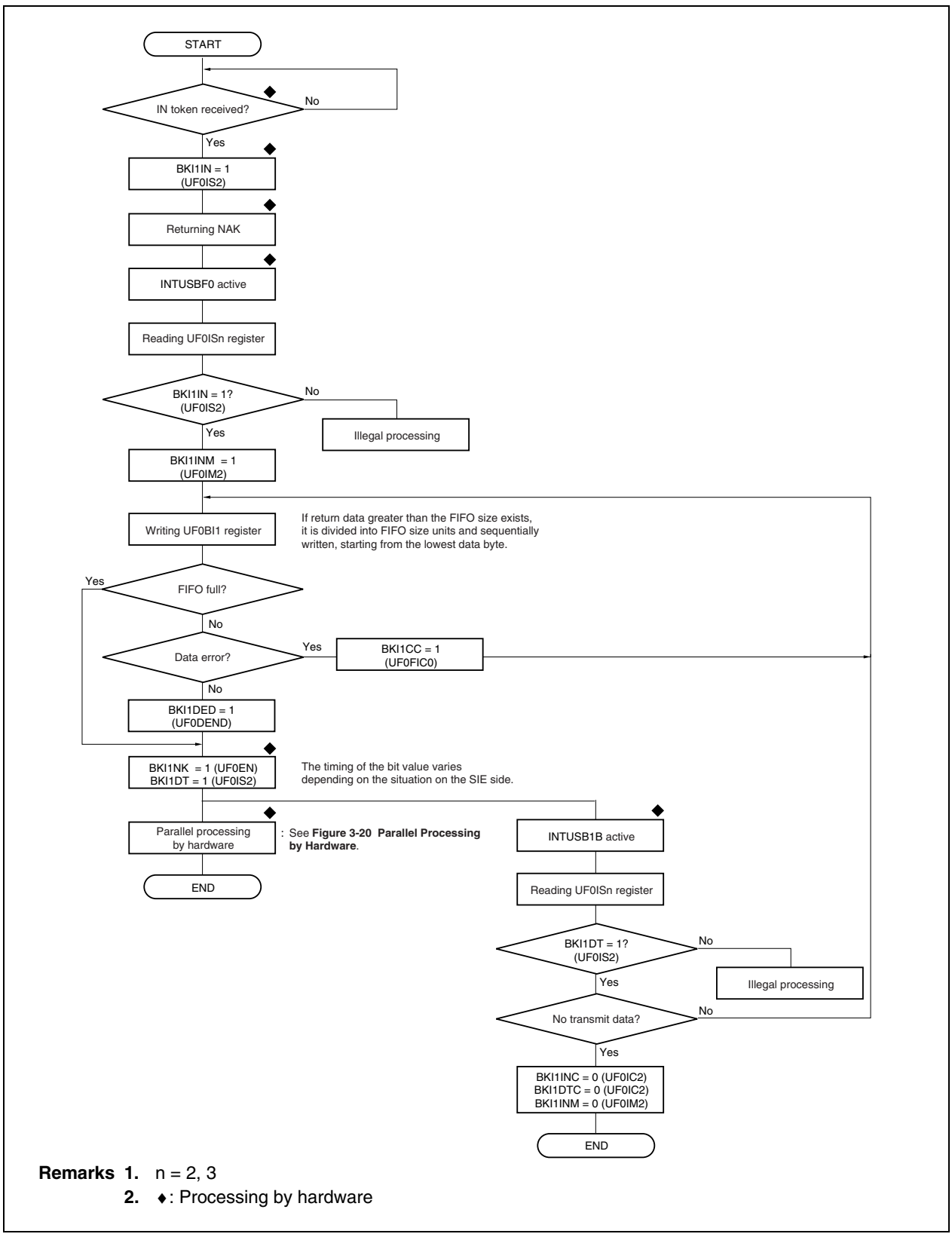
Remarks 1. n = 0, 1

2. ♦: Processing by hardware

(4) Processing for bulk transfer (IN)

Bulk transfer (IN) is allocated to Endpoint1. The flowcharts how Endpoint1 is controlled are shown below.

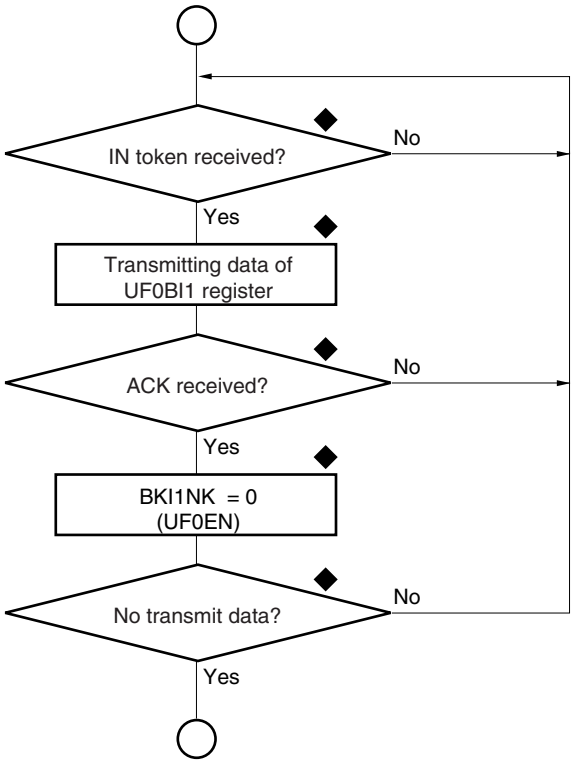
Figure 3-19. Processing for Bulk Transfer (IN) (Endpoint1)



Remarks 1. n = 2, 3

2. ♦: Processing by hardware

Figure 3-20. Parallel Processing by Hardware

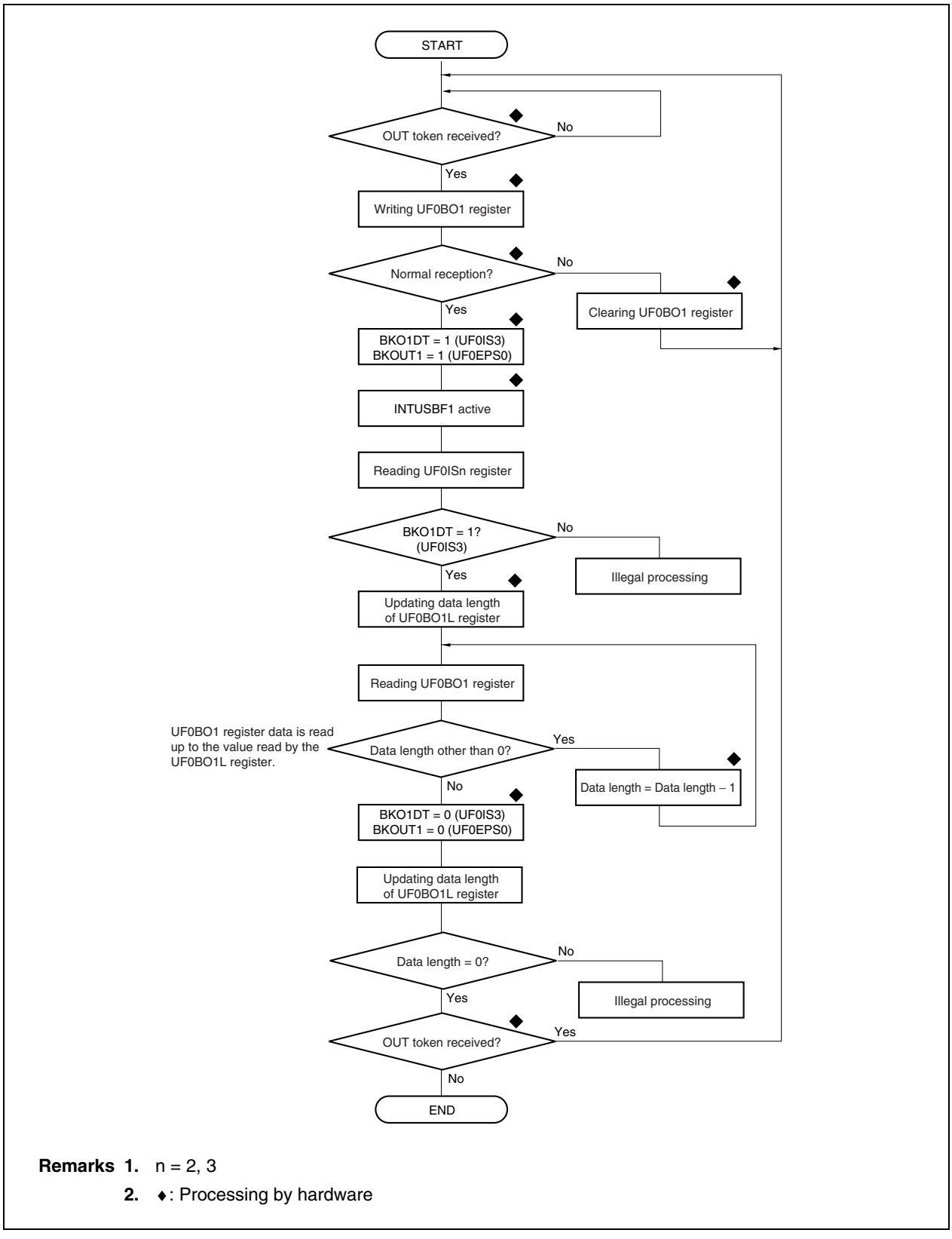


Remark ♦: Processing by hardware

(5) Processing for bulk transfer (OUT)

Bulk transfer (OUT) is allocated to Endpoint2. The flowchart how Endpoint2 is controlled is shown below.

Figure 3-21. Normal Processing for Bulk Transfer (OUT) (Endpoint2)

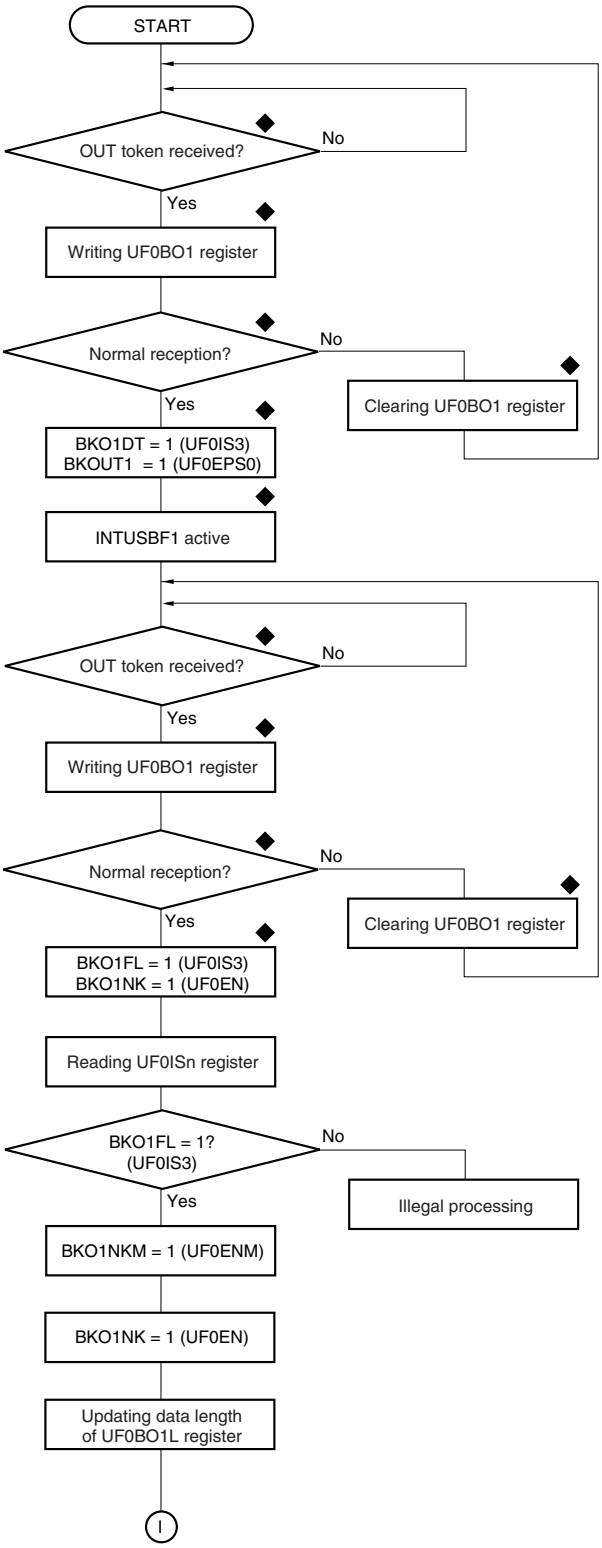


Remarks 1. n = 2, 3

2. ♦: Processing by hardware

During bulk transfer (OUT), more data may be transmitted from the host than expected by the system. Endpoint2 for bulk transfer (OUT) of the V850E2/ME3 consist of two 64-byte buffers so that NAK responses are suppressed as much as possible and data can be read from the CPU side even while the bus side is being accessed as the transfer rate of the USB bus increases. Consequently, if the host sends more data than expected by the system, up to 128 bytes of extra data may be automatically received in the worst case. In this case, change the control flow from that of the normal processing of Endpoint2 to the flow illustrated below when the quantity of data expected by the system has decreased to two packets. The flowcharts how Endpoint2 is controlled are shown below.

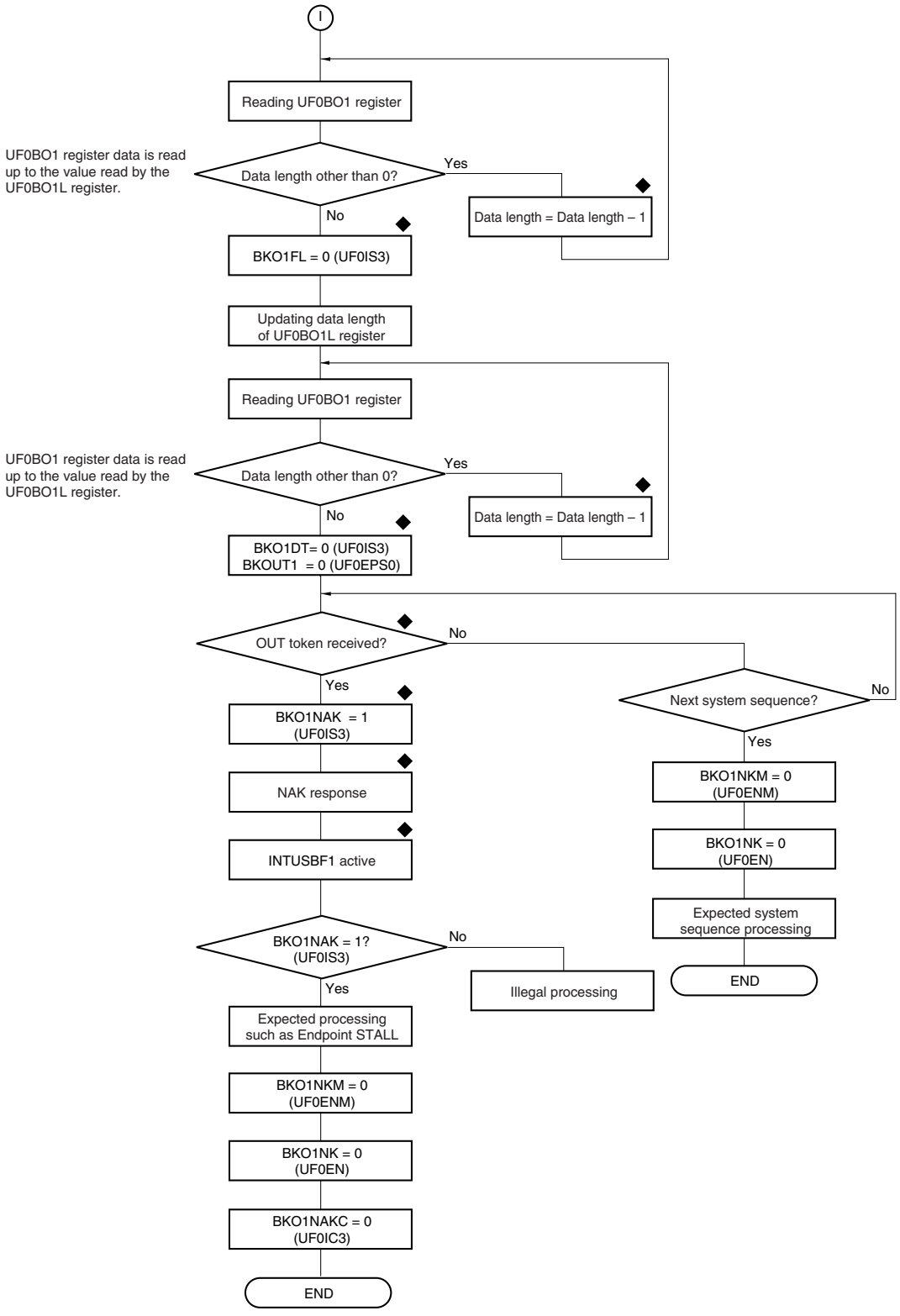
Figure 3-22. Processing If More Data Than Expected by System Is Transmitted (Endpoint2) (1/2)



Remarks 1. n = 2, 3

2. ♦: Processing by hardware

Figure 3-22. Processing If More Data Than Expected by System Is Transmitted (Endpoint2) (2/2)

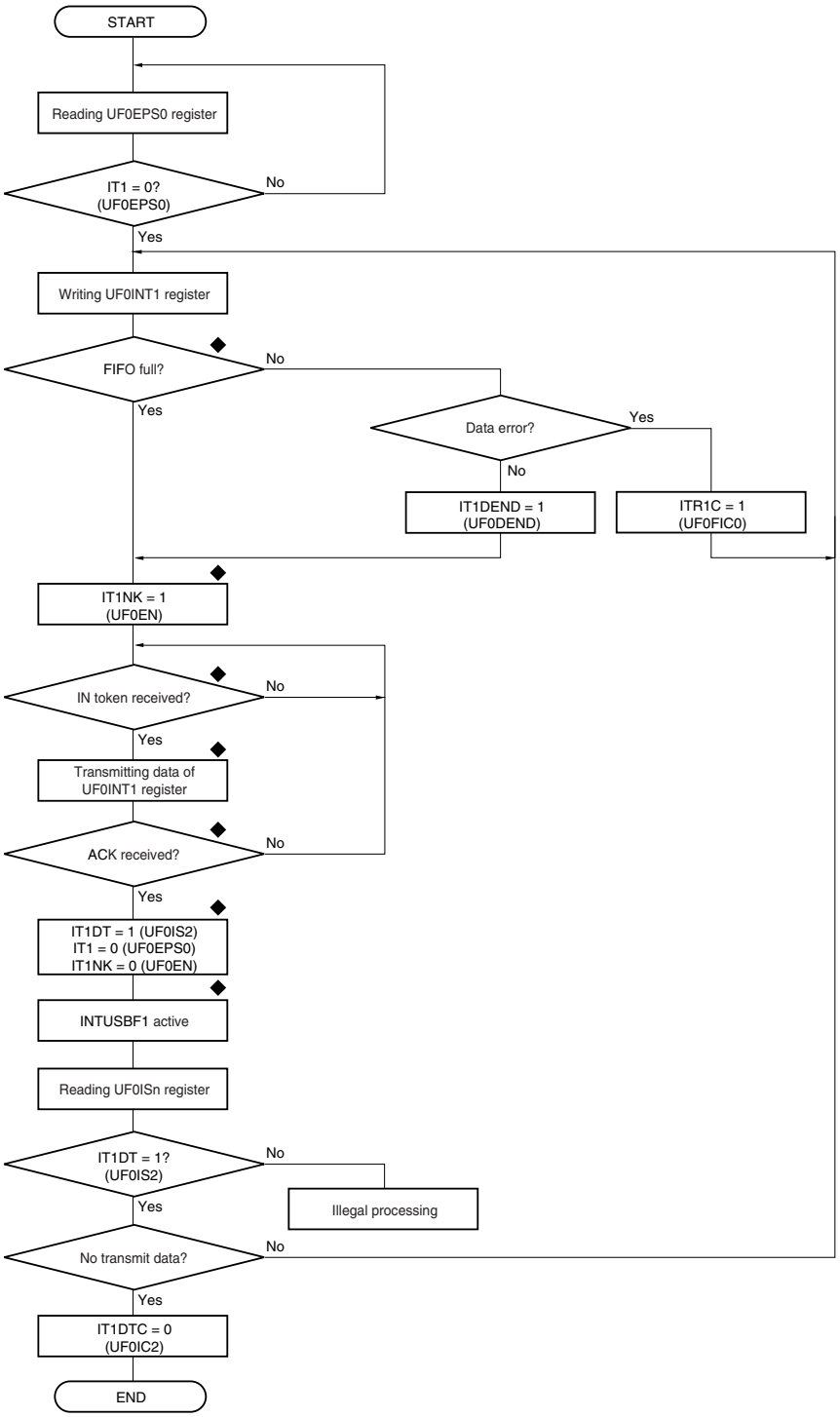


Remark ♦: Processing by hardware

(6) Processing for interrupt transfer (IN)

Interrupt transfer (IN) is allocated to Endpoint7. The flowchart how Endpoint7 is controlled is shown below.

Figure 3-23. Processing for Interrupt Transfer (IN) (Endpoint7)



Remarks 1. n = 2, 3

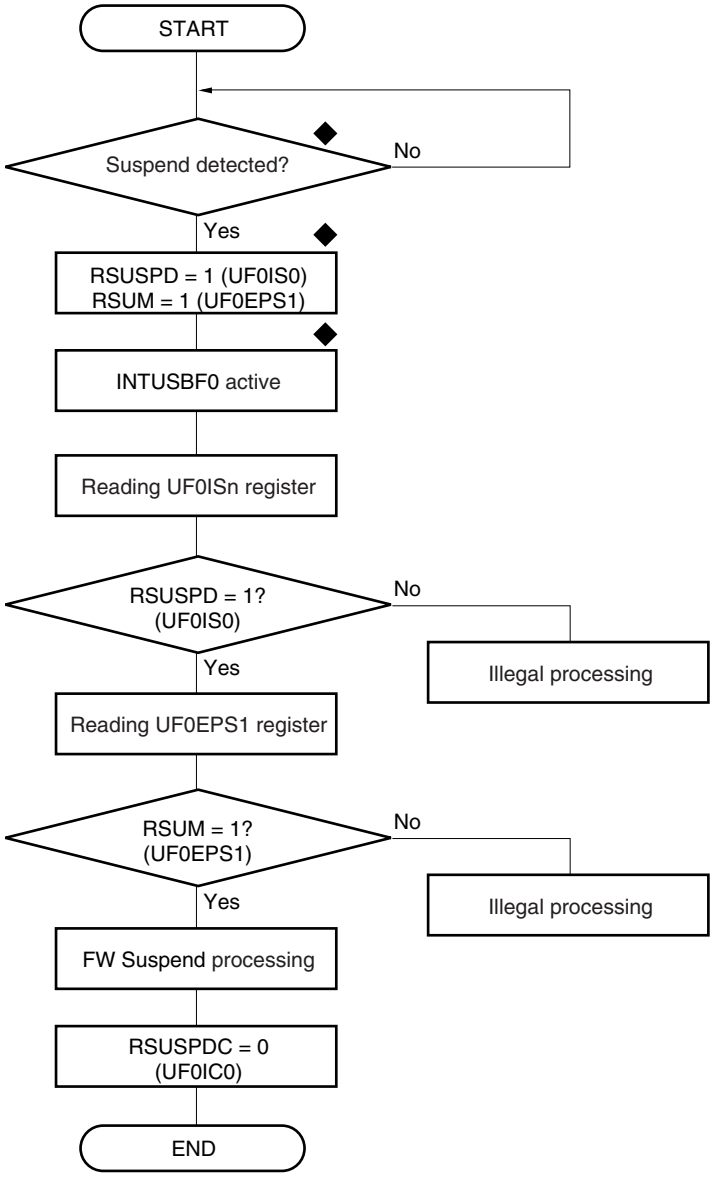
2. ♦: Processing by hardware

3.7.4 Suspend/Resume processing

How Suspend/Resume processing is performed differs depending on the configuration of the system. One example is given below.

Figure 3-24. Example of Suspend/Resume Processing (1/3)

(a) Example of Suspend processing

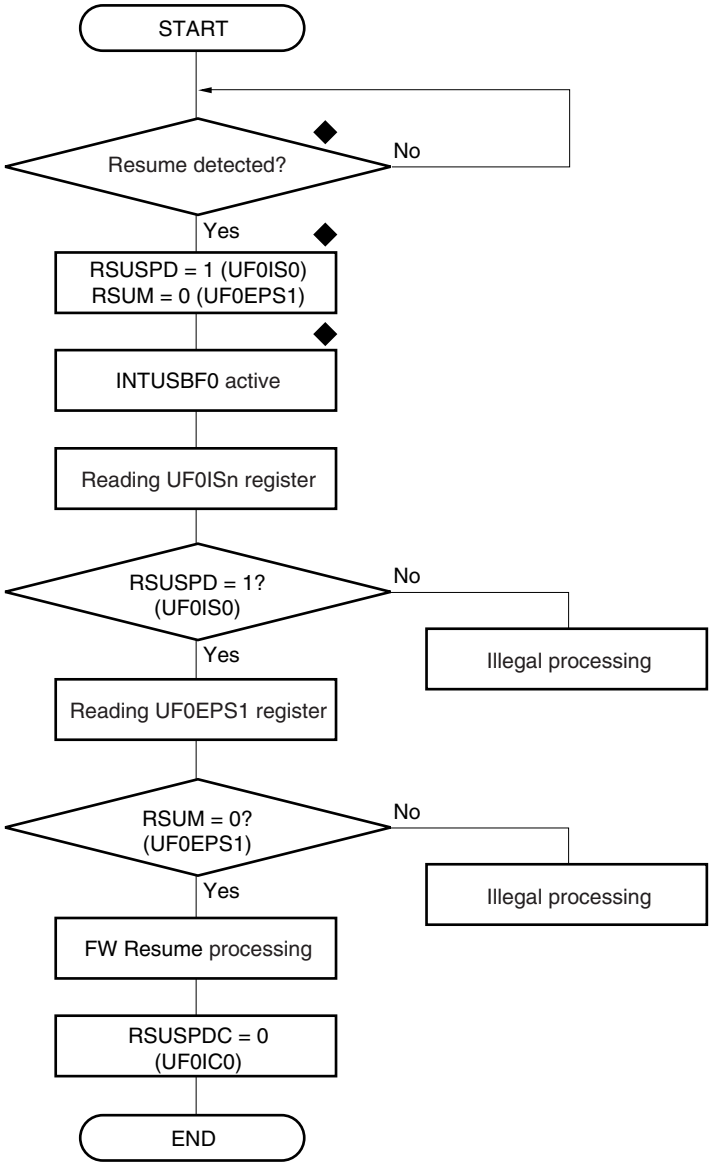


Remarks 1. n = 0, 1

2. ♦: Processing by hardware

Figure 3-24. Example of Suspend/Resume Processing (2/3)

(b) Example of Resume processing

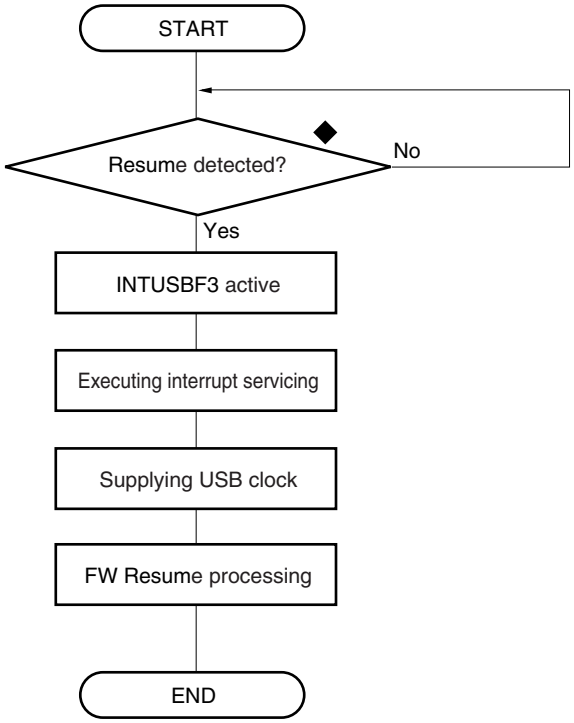


Remarks 1. n = 0, 1

2. ♦: Processing by hardware

Figure 3-24. Example of Suspend/Resume Processing (3/3)

(c) Example of Resume processing (when supply of USB clock to USBF is stopped)



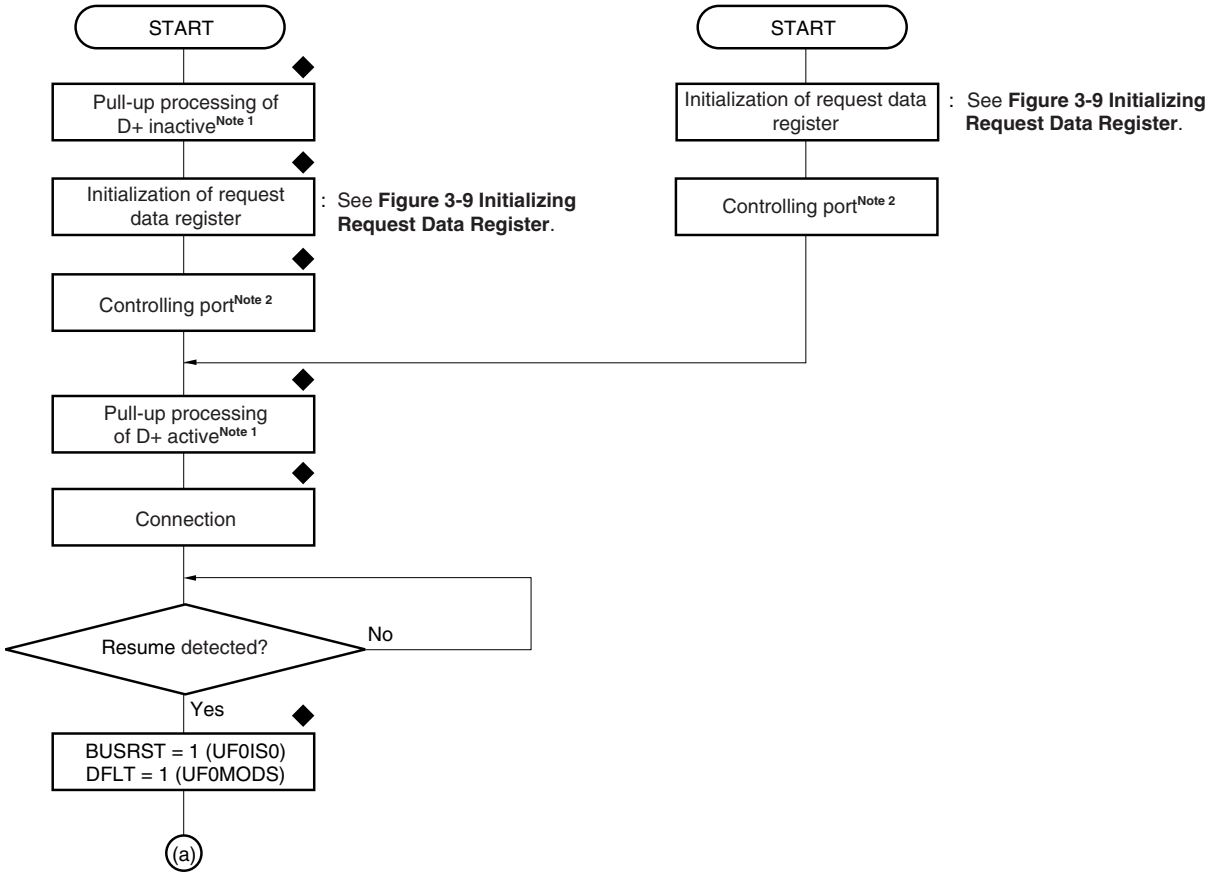
Remark ♦: Processing by hardware

3.7.5 Processing after power application

The processing to be performed after power application differs depending on the configuration of the system. One example is given below.

Figure 3-25. Example of Processing After Power Application/Power Failure (1/3)

(a) Processing after power application (1/2)



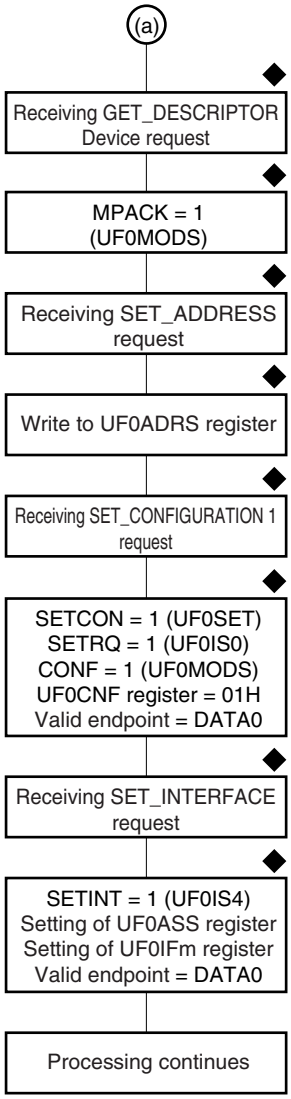
Notes 1. Use one general-purpose port pin for the signal that controls switching of the pull-up resistor of the USB bus.

2. The input mode or control mode of the general-purpose port pin allocated in **Note 1** may be selected as the default value. Note the active level of pull-up processing of D+ on power application.

Remark ♦: Processing by hardware

Figure 3-25. Example of Processing After Power Application/Power Failure (2/3)

(a) Processing after power application (2/2)

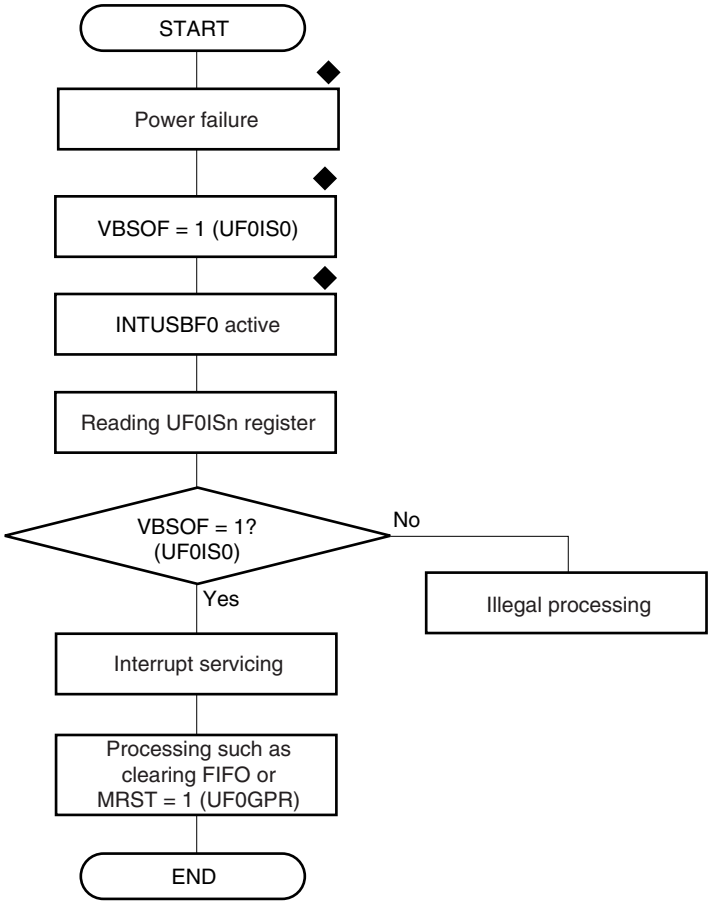


Remarks 1. m = 0 to 4

2. ♦: Processing by hardware

Figure 3-25. Example of Processing After Power Application/Power Failure (3/3)

(b) Processing on power failure



Note • Detecting disconnection of the connector in the case of self-powered mode (SFPW bit of UF0DSTL register = 1). In this case, monitor the VDD line of the USB connector, and input the result to the VBUSDET pin. Note that the noise elimination time according to the noise filter setting register is added when VBUSDET is acknowledged as interrupt INTUSBF0.

• Detecting turning off power from a HUB chip when the device is mounted on the same board as a HUB.

Remark ♦: Processing by hardware

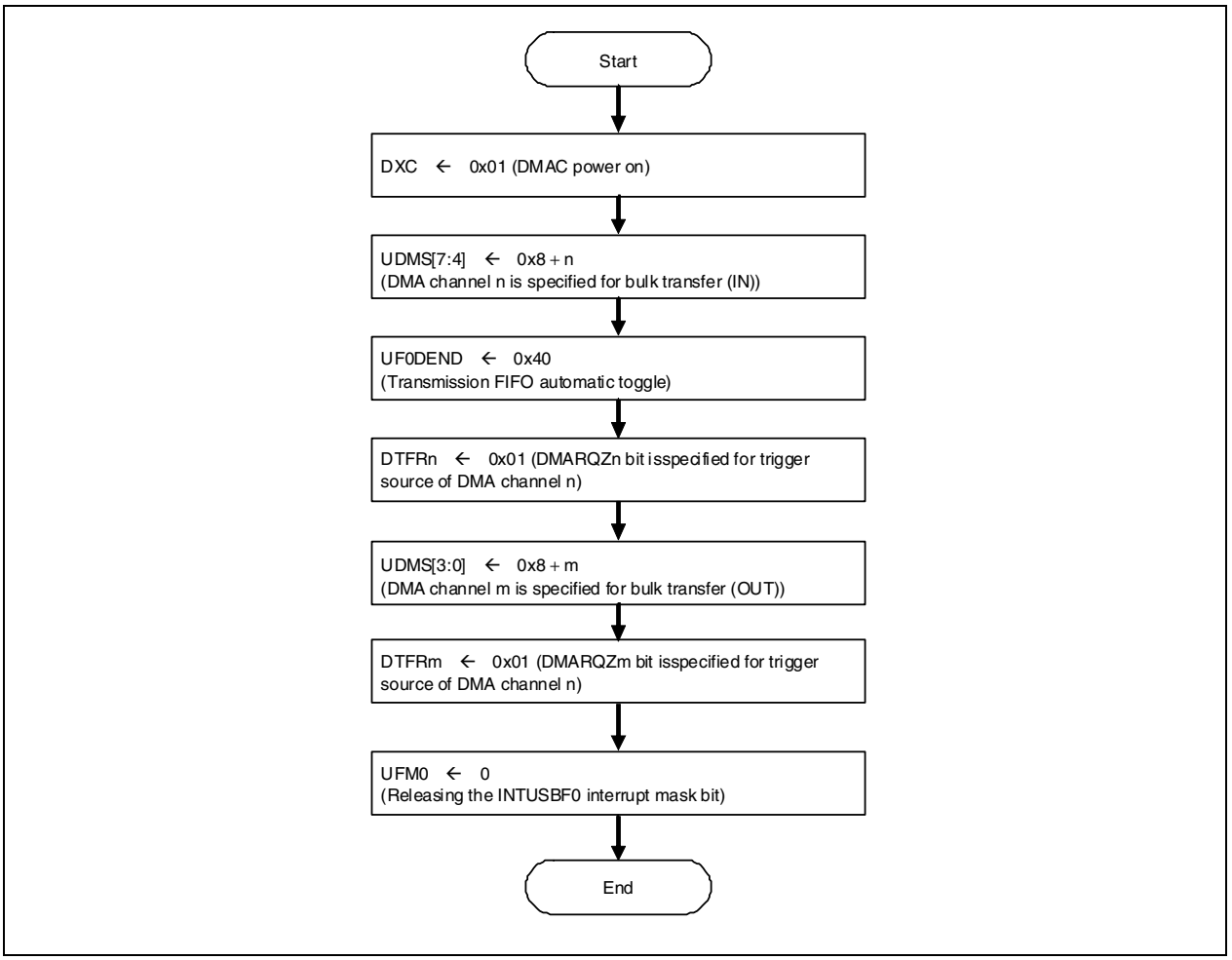
3.7.6 Bulk transfer in DMA mode

Endpoint1 and Endpoint2 are provided with an interface that is used as the transfer target I/O of the DMA controller (DMAC) incorporated in PFESiP/V850EP1 and performs bulk transfer operations. Before using the interface, the interface on the DMAC side must be assigned to it during initialization processing. An example is shown below in Figure 3-26. For details of the DMAC register, see **8.4 Control Registers** in PFESiP/V850EP1 Hardware (CPU Function) User's Manual, and for details of the UDMS register, see **10.6 DMA Interface Signals for USB Connections** in PFESiP/V850EP1 Hardware (CPU Function) User's Manual.

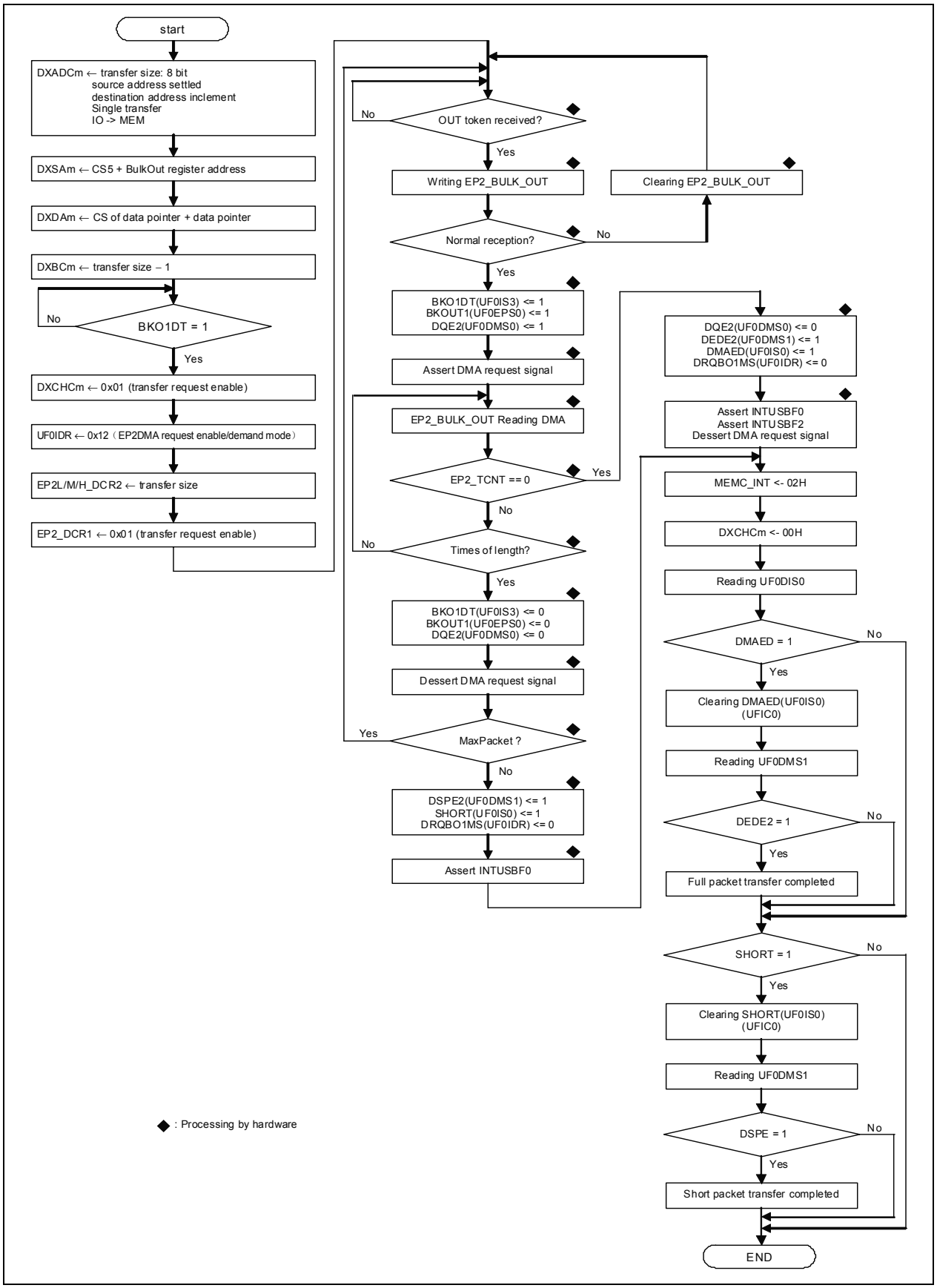
A transfer operation is started according to instructions from the host, and firmware uses interrupts to capture transfers. Figure 3-27 shows an example of the processing flow corresponding to the start instructions issued from
<R> the host using IN tokens and OUT tokens. After an interrupt is accepted and settings on the DMAC side are made, the operation is switched to DMA mode by the DQBI1MS or DQBO1MS bit of the UF0IDR register, and to start transfer using hardware is allowed by releasing the EP1_DCR1 or EP2_DCR1 mask bit. The transfer data is read or written via the EP1_BULK_IN and EP2_BULK_OUT registers, which are used only for DMA. The end of a transfer is captured using the INTUSBF1 and INTUSBF2 signal or INTUSBF0 signal. Afterward, toggle any FIFO that is not full and clear the interrupt source and end status.

The processing flow of the bulk-only method of a storage class, which is started differently, is almost identical. For details, see the application note entitled USB Function Sample Software.

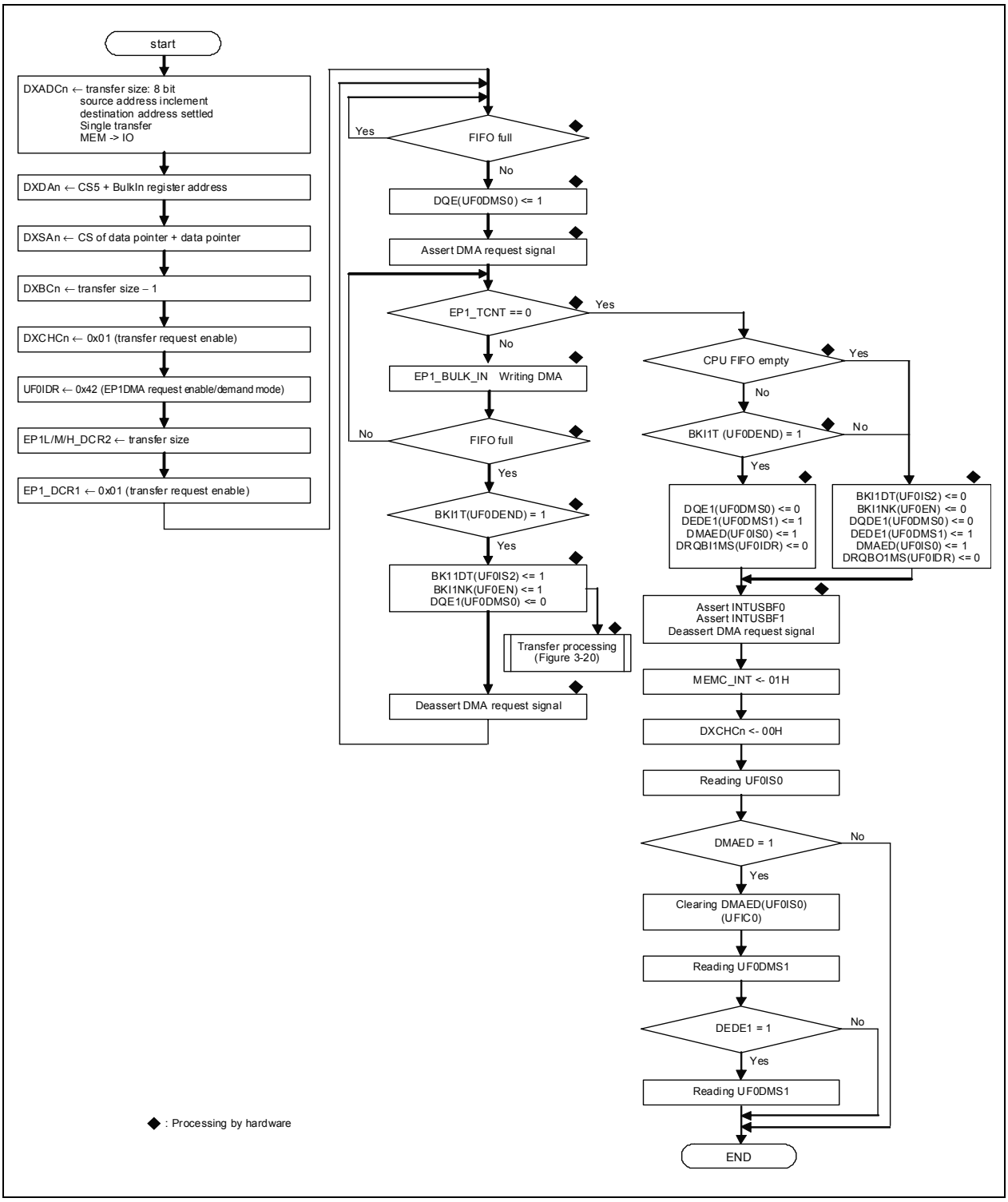
<R> **Figure 3-26. DMA Initialization Processing**



<R> **Figure 3-27. Flow Example of Bulk Transfer Processing in DMA Mode (1/2)**



<R> **Figure 3-27. Flow Example of Bulk Transfer Processing in DMA Mode (2/2)**



3.7.7 Interrupts from USB function controller

In the USB function controller, interrupts from the EPC, bridge, or an external source are merged and are reported to the system as five interrupts.

The interrupt controller uses edge detection to recognize when any of these five interrupts has occurred. Consequently, when performing interrupt servicing for INTUSBF0 to INTUSBF2, always clear all of the interrupt sources at the start of interrupt servicing. If any interrupt sources remain, new interrupts may not be started.

Table 3-18. Interrupts from USB Function Controller

Interrupt Signal	Status Register			Generating Source
	Register Name	bit	Abbreviation of Source	
INTUSBF0	UF0IS0	7	BUSRST	Bus reset occurrence
		6	RSUSPD	Status transition to Resume or Suspend
		5	VBSOF	VBUS OFF detection
		4	SHORT	Short packet issuance (during DMA transfer)
		3	DMAED	End of DMA transfer
		2	SETRQ	End of control transfer to be automatically processed
		1	CLRRQ	CLEAR_FEATURE request reception and end of automatic processing
		0	EPHALT	Endpoint stall
	UF0IS1	6	E0IN	EP0 NAK response (IN token)
		5	E0INDT	End of transmission from UF0E0W register (end of IN transaction)
		4	E0ODT	Normal data reception by UF0E0R register (end of OUT transaction)
		3	SUCES	Normal end of control transfer
		2	STG	Transition to status stage by control transfer
		1	PROT	SETUP token reception
		0	CPUDEC	SETUP token reception (reception of request other than auto response)
	UF0IS2	5	BKI1IN	EP1 NAK response (IN token)
		4	BKI1DT	Occurrence of FIFO toggle operation (data can be written to EP1)
		0	IT1DT	End of EP7 transmission
	UF0IS3	3	BKO1FL	Received data found in both FIFOs of UF0BOn register
		2	BKO1NL	Null packet reception (EP2)
		1	BKO1NAK	EP1 NAK response (OUT token)
		0	BKO1DT	End of EP2 data reception
	MEMC_INT	0	EP1_ENDINT	End of DMA transfer (EP1)
		1	EP2_ENDINT	End of DMA transfer (EP2)
INTUSBF1	MEMC_INT	0	EP1_ENDINT	End of DMA transfer (EP1)
INTUSBF2	MEMC_INT	1	EP2_ENDINT	End of DMA transfer (EP2)
INTUSBF3				Occurrence of Resume status [SIE]
INTUSBF4				VBUS ON detection [external signal]

See **3.7 Firmware Processing** for descriptions of firmware processing for each interrupt report.

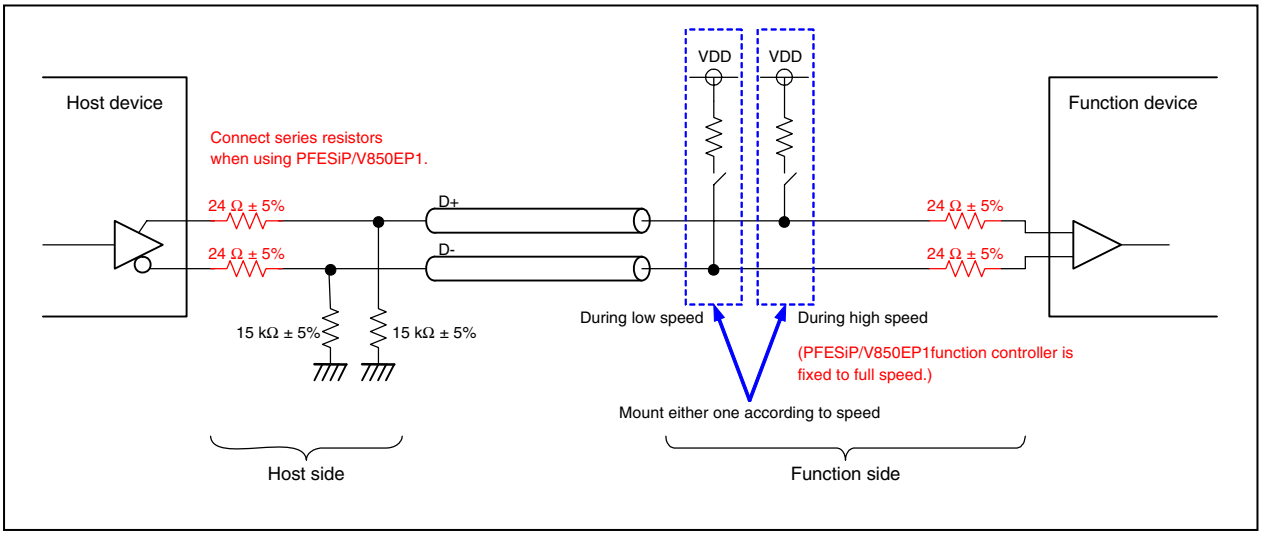
CHAPTER 4 EXTERNAL CIRCUIT CONFIGURATION

During a USB transmission, when the host controller and function controller are at opposite ends of communication, USB signals (D+ and D-) must be connected with pull-up or pull-down resistors so that the other party of communication can be recognized. With PFESiP/V850EP1, series resistors must also be connected.

Since PFESiP/V850EP1 is not incorporated with these pull-up, pull-down, or series resistors, connect them to PFESiP/V850EP1 externally.

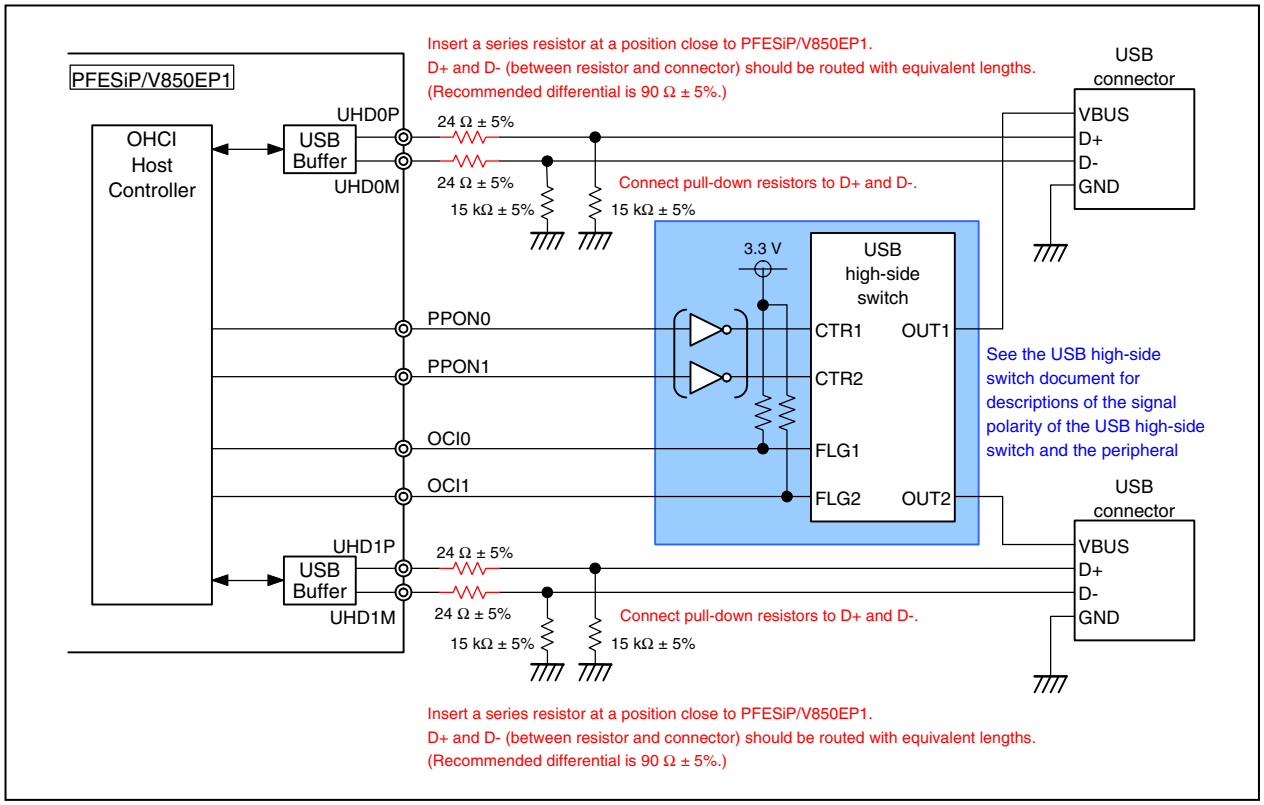
The following is an outline of the USB transmission path configuration. Details of the external configuration are described below in each item.

Figure 4-1. Configuration Outline of Pull-up, Pull-down, and Series Resistors of USB Transmission Path



4.1 USB Host Controller Connection Configuration

Figure 4-2. USB Host Controller Connection Example



4.1.1 USB signal connection

(1) Connection of series resistor to D+ and D-

Connect resistors rated at 24 Ω ±5% in series to the D+ and D- pins (UHD0P, UHD0M, UHD1P, and UHD1M) of the PFESiP/V850EP1 USB host controller. If these are not connected, the impedance rating cannot be met and the output waveform may become distorted.

Place series resistors as close as possible to PFESiP/V850EP1, and route with equivalent length so that the D+ and D- impedances from the series resistor to the USB connectors are equivalent. (Recommended differential is 90 Ω ±5%.)

(2) D+ and D- pull-down connections

Pull-down the D+ and D- pins (UHD0P, UHD0M, UHD1P, and UHD1M) to GND at 15 kΩ ±5%. In this case, the configuration is the same as when no function devices are connected.

4.1.2 USB power supply connection

(1) Overcurrent detection and power supply control

PFESiP/V850EP1 does not incorporate with an overcurrent detection circuit or a power supply controller. To enable the system to support these functions, connect an external configuration of circuits to the OCI and PPON pins.

The following shows operations related to the OCI and PPON signals that are used in external circuit control for overcurrent protection and power supply control of the USB ports

Table 4-1. OCI and PPON Signal Description

Pin	I/O	Level	Description
OCI	Input	1	Overcurrent is not detected.
		0	Overcurrent is detected.
PPON	Output	1	Power supply to VBUS is turned on.
		0	Power supply to VBUS is turned off.

Figure 4-2 shows examples of connections for overcurrent detection and power supply (VBUS) control. Although this depends on the configuration of connections to the USB connectors, this enables the VBUS of the port to be stopped to reduce power consumption when the USB bus is not being used. If the opposing USB function device is a bus-powered type, use of a high-side switch, as shown in the connection examples, is recommended.

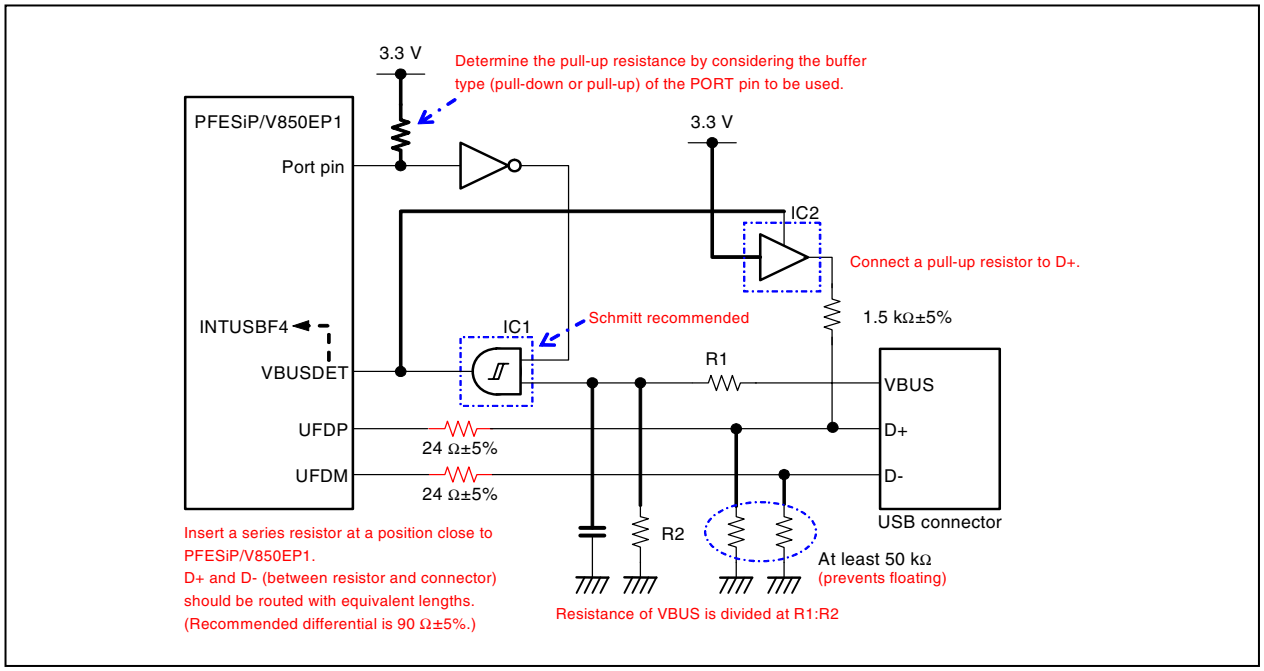
(2) Specification for VBUS control based on root hub register setting conditions

Control of the PPS bit that is used to control the VBUS changes according to the values set to the HcRhDescriptorA and HcRhDescriptorB registers (root hub control registers).

When the VBUS is controlled for each port, the NoPowerSwitching bit and the PortSwitchingMode bit must be set to “1”.

4.2 USB Function Controller Connection Configuration

Figure 4-3. USB Function Controller Connection Example



4.2.1 USB signal connection

(1) Connection of series resistor to D+ and D-

Connect resistors rated at 24 Ω ±5% in series to the D+ and D- pins (UFDP and UFDM) of the PFESiP/V850EP1 USB function controller. If these are not connected, the impedance rating cannot be met and the output waveform may become distorted.

Place series resistors as close as possible to PFESiP/V850EP1, and route with equivalent length so that the D+ and D- impedances from the series resistor to the USB connectors are equivalent. (Recommended differential is 90 Ω ±5%.)

(2) Pull-up control of D+

Since PFESiP/V850EP1 has a full-speed (FS) function controller, pull up the D+ pin (UFDP) to the 3.3 V power supply, using 1.5 kΩ ±5%.

To prohibit connection report (D+ pull-up) to the USB host/HUB (such as while higher priority processing or initialization processing is under execution), the system must control pulling up of D+ via a general-purpose port. As shown in the circuit example in Figure 4-3, control the pull-up control signal of D+ and VBUS input signal by using a general-purpose port and USB cable VBUS (AND circuit). In Figure 4-3, pulling up D+ is prohibited when the general-purpose port is high level (secure the high level of the general-purpose port pin by pulling it up because the port pin is in the input mode by default).

Use as IC2, shown in Figure 4-3, an IC to which voltage can be applied when the system power supply is off.

(3) Detecting USB cable connection/disconnection

The USB function controller requires a VBUSDET signal that recognizes connection and disconnection, because the state of the USB function controller is managed by hardware. As shown in Figure 4-3, divide the resistance of the VBUS signal and connect it to VBUSDET via IC1. (Do not connect VBUS (5 V) directly to VBUSDET). Voltage (5 V) is applied from the USB host/HUB, if the USB cable VBUS is connected to the USB host/HUB when power to the function controller is turned off. Therefore, use as IC1, shown in Figure 4-3, an IC to which voltage can be applied when the system power supply is off.

When disconnecting the USB cable in the circuit in Figure 4-3, the VBUSDET signal may become unstable while the VBUS voltage is dropping. Therefore, it is recommended to use a Schmitt buffer for IC1, shown in Figure 4-3.

(4) Preventing floating during initialization or non-use

To avoid a floating status during initialization or non-use, pull down the D+ and D- pins by using at least 50 kΩ.

4.3 Clock and Reset

UCLK is required for both the USB function controller and the USB host controller. Supply a signal of 48 MHz ±1500 ppm as UCLK. If the precision of UCLK deteriorates, the transmit data will not meet the USB standard. <R>

Table 4-2. UCLK Timings

Item	Symbol	MIN.	TYP	MAX.	Unit
UCLK frequency	φ_S	−1500 ppm	48	+1500 ppm	MHz
UCLK high-level width	t_{HWS}	8.0	–	–	ns
UCLK low-level width	t_{LWS}	8.0	–	–	ns

<R>

Only the USB host controller requires PCLKIN in order to use the PCI interface. Supply a signal of 25 to 33 MHz as PCLKIN.

Table 4-3. PCLK Timings

Item	Symbol	MIN.	TYP	MAX.	Unit
PCLK frequency	φ_S	25	–	33	MHz
PCLK high-level width	t_{HWS}	9.0	–	–	ns
PCLK low-level width	t_{LWS}	9.0	–	–	ns

The reset for the USB function controller and USB host controller is shared with the CPU system reset. Supply PCLKIN when resetting, including when resetting the USB host controller. The reset time is at least 60 ns.

CHAPTER 5 CAUTIONS

5.1 USB Port Status Transition Control

Under the OHCI standard, if a port is disabled by a bus error, there are two ways the HCD can transition the port to enabled state.

<1> Set (1) the SetPortEnable bit of the HcRhPortStatus [1:2] register
<2> When the port reset signal has been asserted and the corresponding port reset has ended by setting (1) the SetPortReset bit of the HcRhPortStatus [1:2] register

Although only <2> above is supported under the Universal Serial Bus Specification, Revision 1.1, either <1> or <2> can be selected when using this host controller. When <2> is selected, the software should be designed so that control which transitions ports to enabled states is performed via the SetPortReset bit.

5.2 Restriction on Number of Hub Stages

A restriction exists concerning the number of hub stages as they relate to inter packet delay. When using a hub that has the maximum (worst) delay value permitted by the USB standard, and with all connections made using 5-meter cables, the maximum number of hub stages is three. This maximum number of stages becomes five when using 3-meter cables or a hub with a normal delay value.

Published by: NEC Electronics Corporation (http://www.necel.com/)
Contact: http://www.necel.com/support/