

SH7262/SH7264 Group

USB Interrupt Transfer by the USB Function Controller

Summary

This application note describes the configuration to use the SH7262/SH7264 USB 2.0 host/function module as the USB function controller and transfer data to the USB host in interrupt transfer.

Target Device

SH7264 MCU (In this document, SH7262/SH7264 are described as "SH7264".)

Contents

1. Introduction.....	2
2. Applications	3
3. References.....	23

1. Introduction

1.1 Specifications

Specifies the SH7264 MCU as the USB function to transfer data to the USB host in interrupt transfer.

1.2 Modules Used

- USB 2.0 Host/Function Module (USB module)
- Direct Memory Access Controller (DMAC)
- Interrupt controller (INTC)

1.3 Applicable Conditions

MCU	SH7262/SH7264
Operating Frequency	Internal clock: 144 MHz Bus clock: 72 MHz Peripheral clock: 36 MHz
Integrated Development Environment	Renesas Technology Corp. High-performance Embedded Workshop Ver.4.07.00
C Compiler	Renesas Technology SuperH RISC engine Family C/C++ compiler package Ver.9.03 Release 00
Compiler Options	Default setting in the High-performance Embedded Workshop (-cpu=sh2afpu -fpu=single -object="\$(CONFIGDIR)\$(FILELEAF).obj" -debug -gbr=auto -chgincpath -errorpath -global_volatile=0 -opt_range=all -infinite_loop=0 -del_vacant_loop=0 -struct_alloc=1 -nologo)

1.4 Related Application Note

For more information, refer to the following application notes:

- SH7262/SH7264 Group Example of Initialization
- SH7262/SH7264 Group Implementing the USB Enumeration on the USB Function Controller
- SH7262/SH7264 Group USB Bulk Transfer by the USB Function Controller
- SH7262/SH7264 Group USB Isochronous Transfer by the USB Function Controller

2. Applications

This application uses the USB 2.0 host/function module (USB module) as the USB function to transfer data to the USB host in interrupt transfer.

2.1 Overview of USB Module

- (1) Includes the USB host controller and function controller compliant to USB high-speed
 - Includes the USB host controller and function controller
 - USB host controller and function controller can be switched by setting registers
 - Includes the USB transceiver

- (2) Reduced number of external pins and space-saving installation
 - Includes the D+ pull-up resistor (When operating as the function)
 - Includes the D+ and D- pull-down resistors (When operating as the host)
 - Includes the D+ and D- terminator (When operating at high-speed)
 - Includes the D+ and D- output resistor (When operating at full-speed)

- (3) Supports all types of USB transfer
 - Control transfer
 - Bulk transfer
 - Interrupt transfer (High-bandwidth is not supported)
 - Isochronous transfer (High-bandwidth is not supported)

- (4) Internal bus interface
 - Includes two channels of DMA interface

- (5) Pipe configuration
 - Includes 8-KB buffer memory for USB communication
 - Up to 10 pipes can be specified (including the default control pipe)
 - Programmable pipe configuration
 - Any endpoint number can be assigned to pipes 1 to 9
 - Transfer conditions for pipes are as follows:
 - Pipe 0: Control pipe (Default control pipe: DCP), 64-byte fixed single buffer
 - Pipes 1 and 2: Bulk or isochronous pipe, continuous transfer mode, programmable buffer size (Double buffering can be specified up to 2 KB)
 - Pipes 3 to 5: Bulk pipe, continuous transfer mode, programmable buffer size (Double buffering can be specified up to 2 KB)
 - Pipes 6 to 9: Interrupt pipe, 64-byte fixed single buffer

- (6) Features as the host controller
 - High-speed (480 Mbps), full-speed (12 Mbps), and low-speed (1.5 Mbps) supported
 - Communicates with multiple peripherals via a hub (tier 1)
 - Automatically responds to the reset handshake
 - Automatically schedules to transmit SOF, and packets
 - Specifies the interval on isochronous and interrupt transfers

(7) Features as the function controller

- High-speed (480 Mbps), and full-speed (12 Mbps) supported
- Automatically detects the high-speed or full-speed operation by replying to the reset handshake
- Manages stage on control transfer
- Manages the device state
- Automatically responds to the SET_ADDRESS request
- NAK response interrupt (NRDY)
- SOF Tracking and Recovery

(8) Other features

- Completes transfer by counting transactions
- Delays the BRDY interrupt event notification timing (BFRE)
- Automatically clears the buffer memory after reading data from the pipe specified by the DnFIFO (n = 0, 1) port (DCLRM)
- Specifies NAK to the response PID by the end of transfer (SHTNAK)

2.2 Interrupt Transfer

Interrupt transfer is used to transfer small quantity of data periodically, which is useful for transferring small data with bounded-latency. Typical applications which include interrupt transfer are keyboard and mouse.

Interrupt transfer includes the following features:

- Periodic transfer
- Unidirectional (Interrupt IN transfer or interrupt OUT transfer)
- Consists of three packets such as the token, data, and handshake
- Maximum packet size: 1 to 1024 bytes (for high-speed endpoints)⁽¹⁾
1 to 64 bytes (for full-speed endpoints)

As the periodic transfer is preferentially executed when scheduling (micro) frames, its latency is guaranteed. However, the number of transactions per (micro) frame is restricted to 1⁽²⁾. Interrupt transfer detects and corrects an error, since it handles the handshake in each transaction.

- Notes: 1: When using interrupt transfer with the SH7264 USB module, set its maximum packet size between 1 and 64 bytes in high-speed transfer.
2: As the High-Bandwidth endpoint supports up to three transactions, however, the SH7264 USB module does not support the High-Bandwidth.

Figure 1 shows the interrupt transfer image.

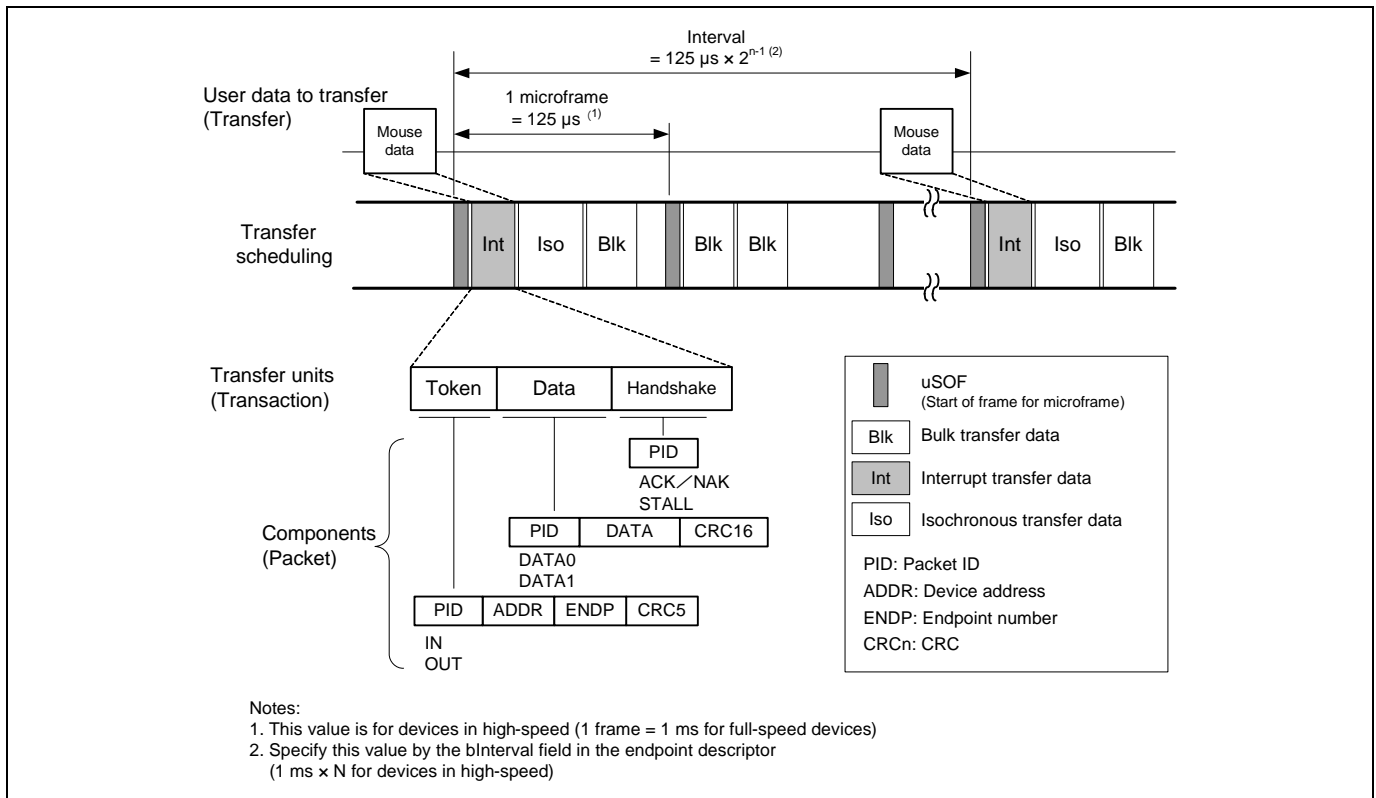


Figure 1 Interrupt Transfer Image

Figure 2 shows the packet response pattern for interrupt IN transfer. Figure 3 shows the packet response pattern for interrupt OUT transfer.

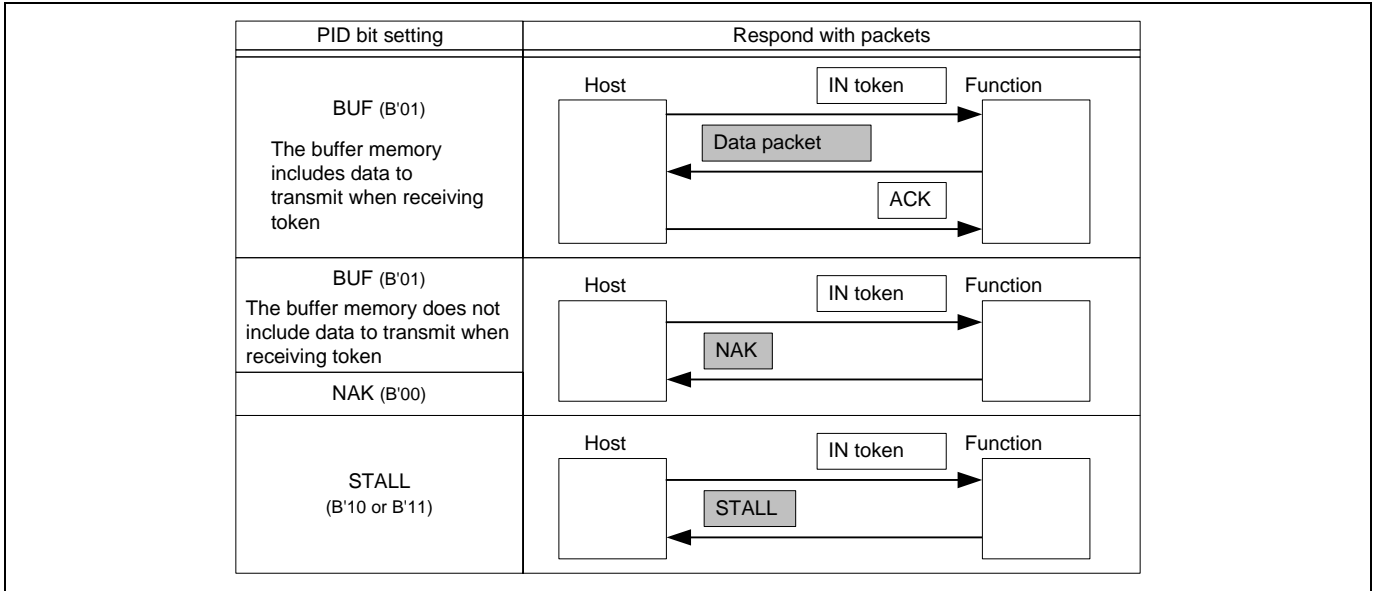


Figure 2 Packet Response Pattern for Interrupt IN Transfer

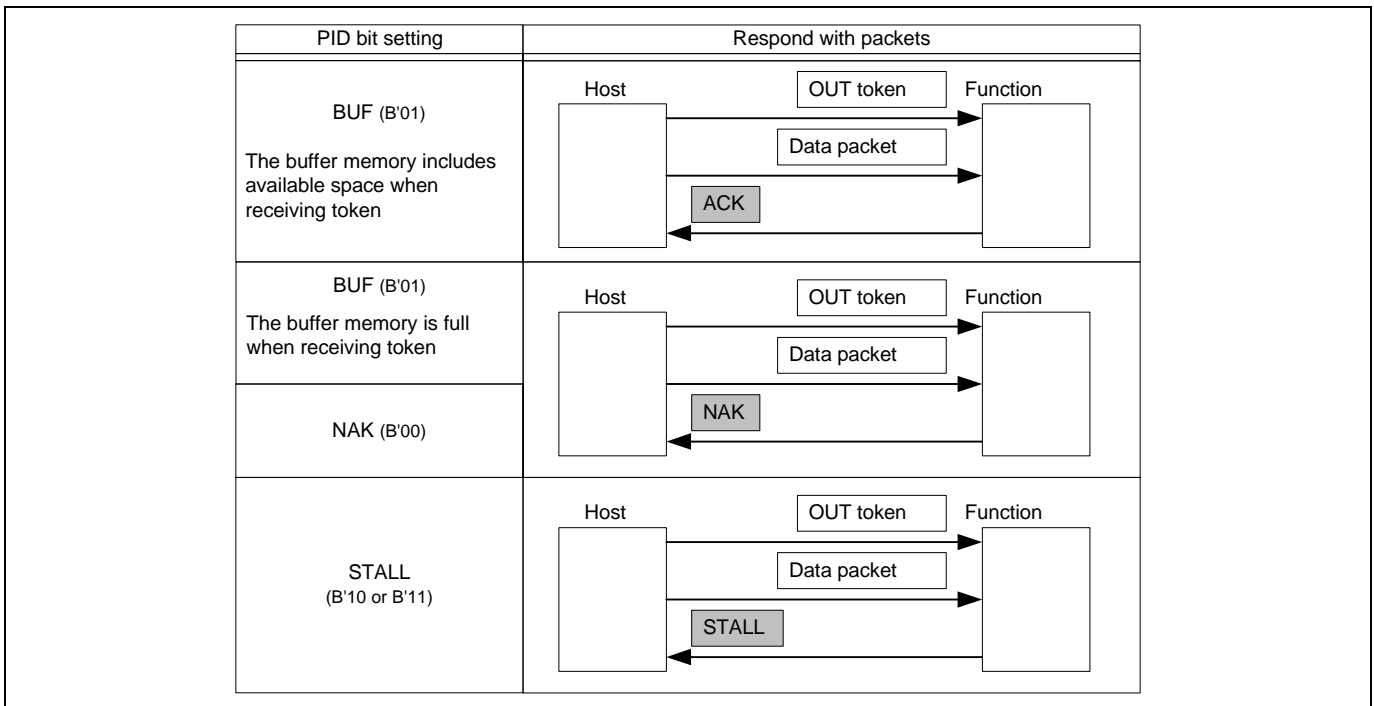


Figure 3 Packet Response Pattern for Interrupt OUT Transfer

Figure 4 shows interrupt transfer setting procedures (overview).

For more information on how to initialize the pipe, refer to 2.3 Pipes. Refer to 2.4 FIFO Port for how to access FIFO port.

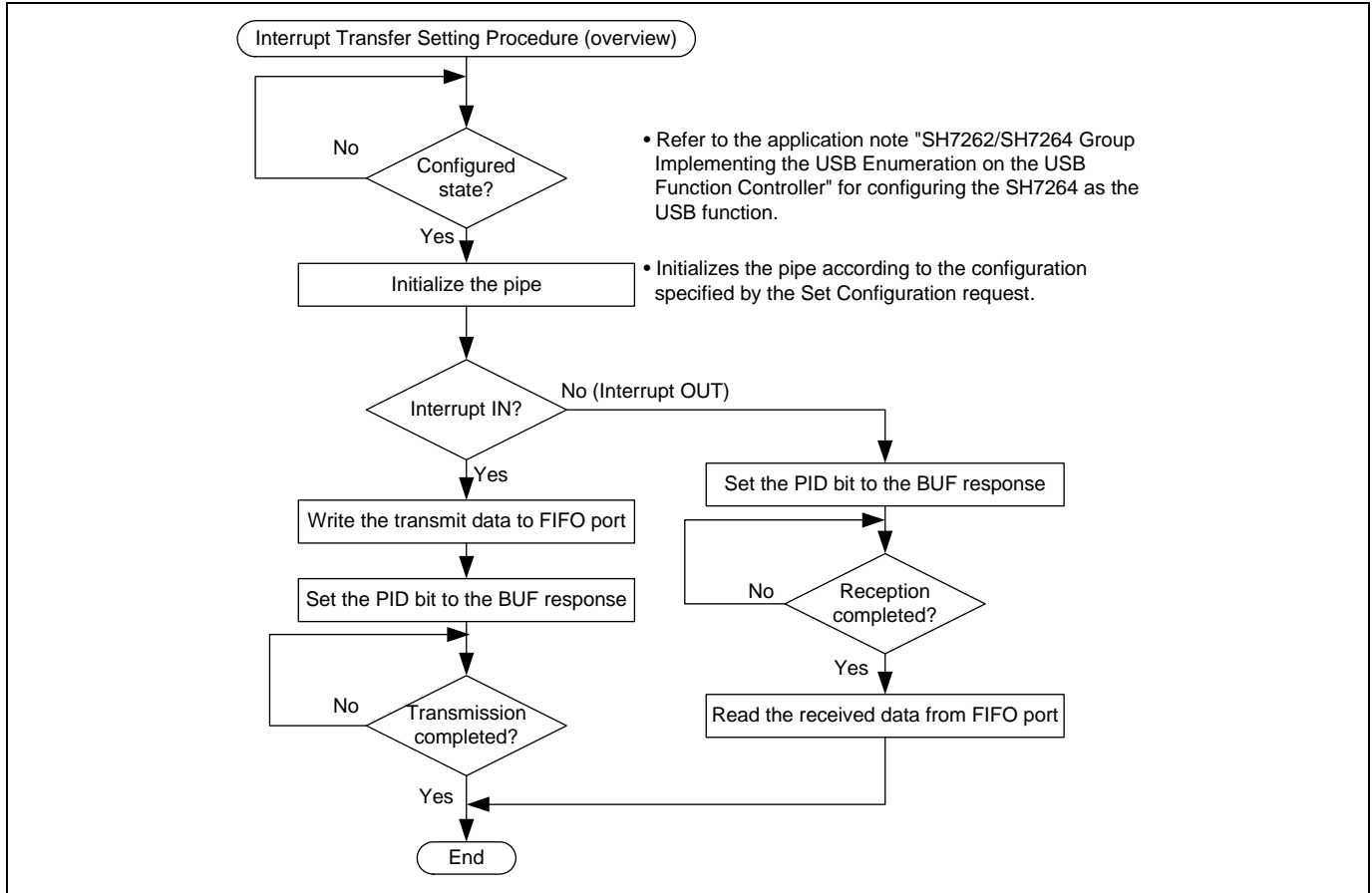


Figure 4 Interrupt Transfer Setting Procedure (Overview)

2.3 Pipes

A USB pipe is a logic communication path in the USB transfer. Specify the transfer type and direction on every pipe to execute multiple USB transfers on a device.

2.3.1 Overview

Figure 5 shows an overview of the pipe.

A USB module uses 10 pipes including the default control pipe (DCP). Pipe 0 (DCP) supports control transfer only. Refer to the "SH7262/SH7264 Group Implementing the USB Enumeration on the USB Function Controller" for details on the DCP. Pipes 1 and 2 are for isochronous transfer, pipes 3 to 5 are for bulk transfer, and pipe 6 to 9 are for interrupt transfer. Note that pipes 1 and 2 can be used for bulk transfer. Pipes 1 to 5 are allowed for using the double buffering and transaction counter to transfer large amount of data. For setting pipes 1 to 9, see 2.3.2 Pipe Configuration Procedure.

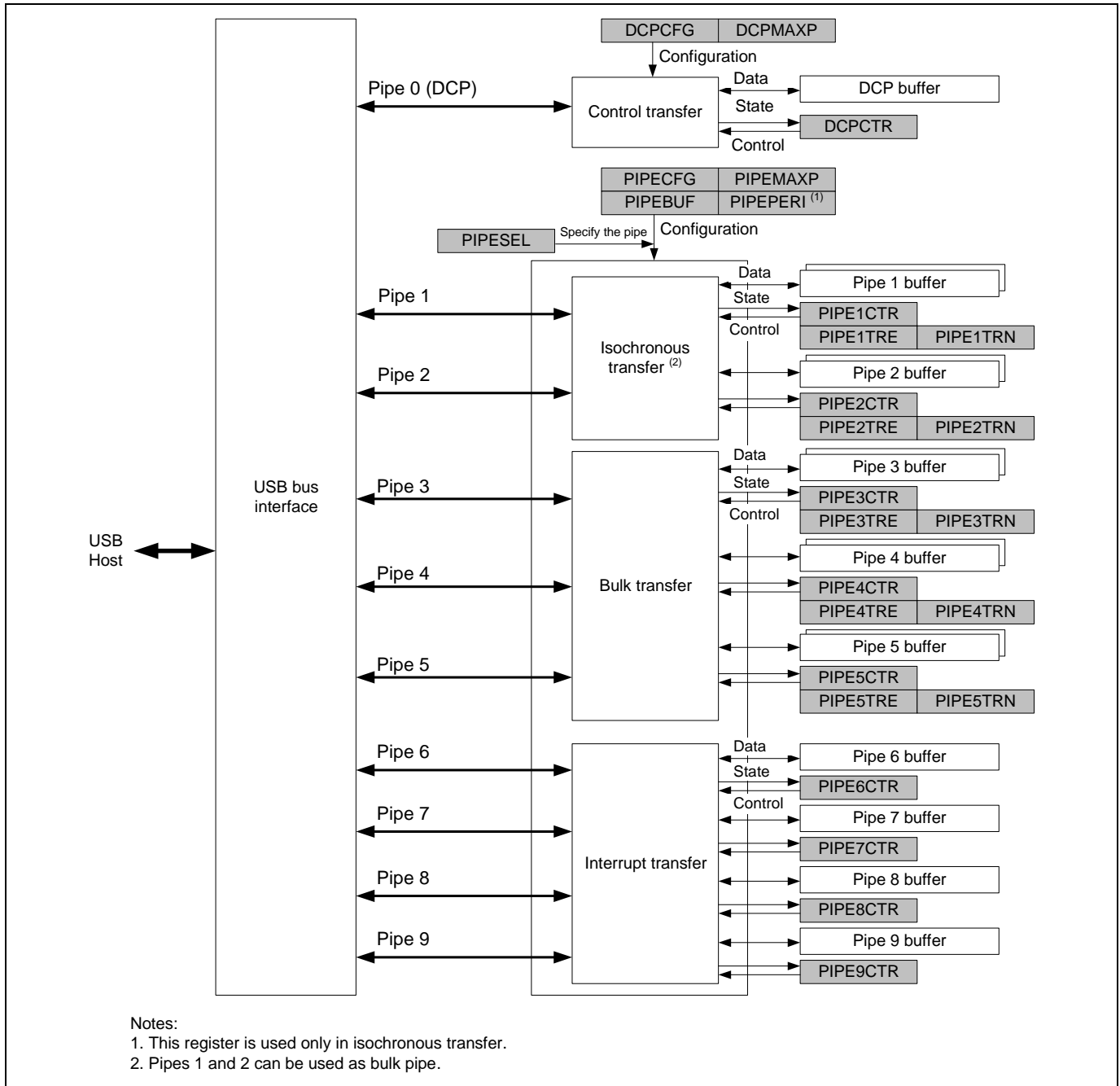


Figure 5 Pipes (Overview)

2.3.2 Pipe Configuration Procedure

Figure 6 shows the configuration procedure of pipes. For details on the pipe setting, refer to 2.3.3 Transmit Pipe Setting (Interrupt IN Transfer) and 2.3.4 Receive Pipe Setting (Interrupt OUT Transfer).

Pipes can be set dynamically. Normally, pipes are set when the target endpoint is decided. As pipes 1 to 9 use some of registers in common, set the target pipe by the Pipe window select register (PIPESEL) before configuration. Note that pipes cannot be configured when the target pipe is already allocated to the FIFO port, or when the PID bit setting is other than NAK.

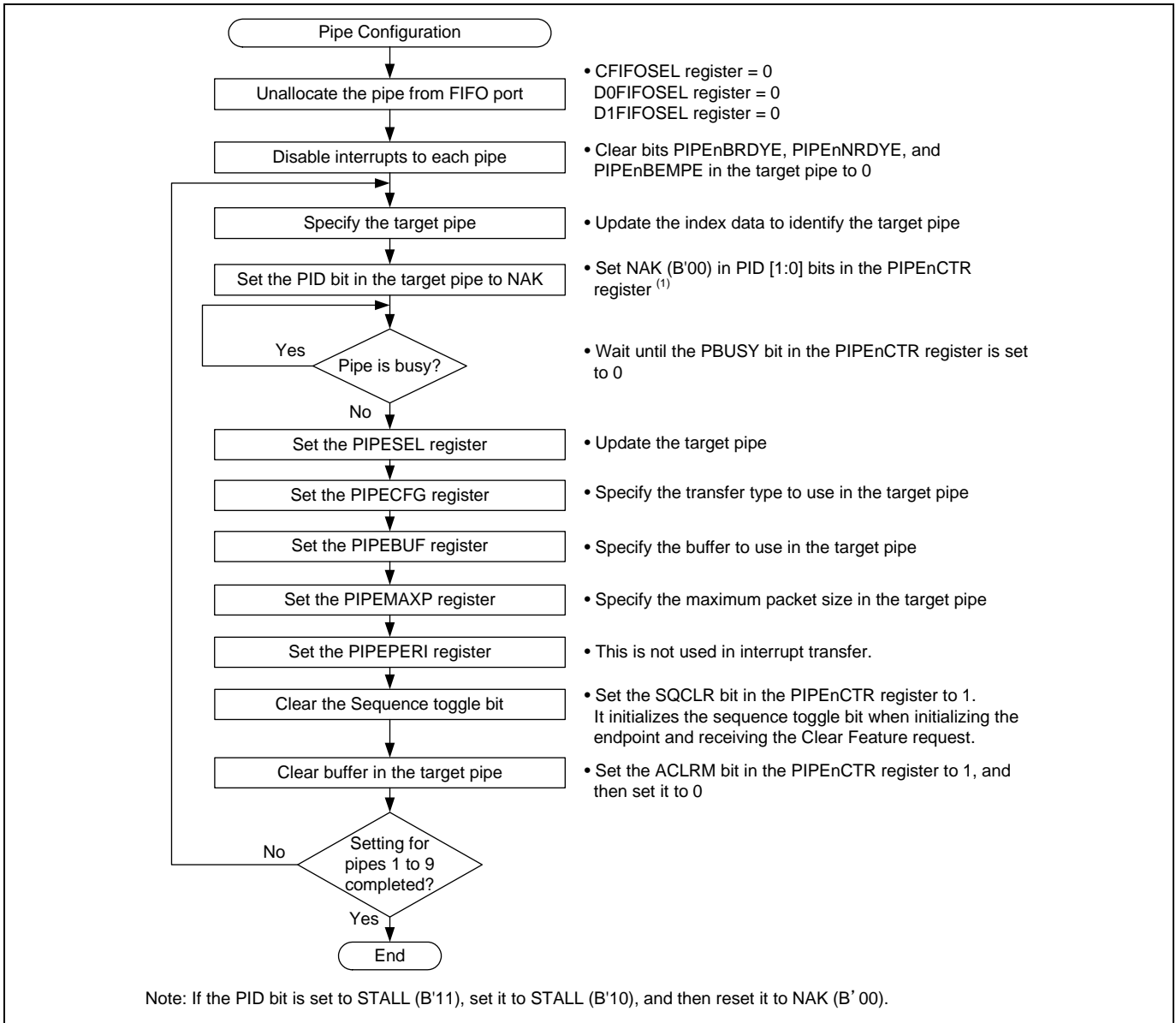


Figure 6 Pipe Configuration Procedure

2.3.3 Transmit Pipe Setting (Interrupt IN Transfer)

This section describes an example of pipe setting when transferring data in interrupt IN transfer.

Table 1 lists the setting example of interrupt IN transfer using pipe 6. Figure 7 shows the operation example using the setting listed in Table 1. As double buffering and continuous transfer mode cannot be used in this example, use BEMP interrupt to write transactions one by one to FIFO buffer memory. For details, refer to (1) to (6) described in following pages.

Table 1 Interrupt IN Transfer Setting Example Using Pipe 6

Register Name	Setting	Description	
PIPESEL register	H'0006	Set pipe 6 as the target pipe	
PIPECFG register	H'8016	TYPE [1:0] bits = 2	Set the transfer type as interrupt transfer
		BFRE bit = 0	BRDY interrupt when transmitting/receiving data (This bit is not enabled in interrupt transfer)
		DBLB bit = 0	Single buffering (This bit is not enabled in interrupt transfer)
		CNTMD bit = 0	Non-continuous transfer mode (This bit is not enabled in interrupt transfer)
		SHTNAK bit = 0	Continues the pipe when a transfer is completed (Only 0 can be specified when transmitting data)
		DIR bit = 1	Set the transfer direction as transmit
		EPNUM bit = 6	Set the endpoint number as 6
PIPEBUF register	H'0004	BUFSIZE [4:0] bits = B'00000	Set the buffer size as 64 bytes (fixed)
		BUFNMB [6:0] bits = 4	Set the first block of a buffer as 4 (fixed)
PIPEMAXP register	H'0040	Set the maximum packet size as 64 bytes	
PIPEPERI register	H'0000	IFIS bit = 0	(Only 0 can be specified in interrupt transfer)
		IITV [2:0] bits = 0	(Only 0 can be specified in interrupt transfer)
BRDYENB register	PIPE6BRDYE bit = 0	BRDY interrupt is disabled	
NRDYENB register	PIPE6NRDYE bit = 0	NRDY interrupt is disabled	
BEMPENB register	PIPE6BEMPE bit = 1	BEMP interrupt is enabled	
SOFCFG register	BRDYM bit = 0	Clearing the BRDY interrupt status automatically is disabled	

Note: The USB module executes interrupt transfer according to the cycle controlled by the USB host. Specify the interval by the `blInterval` field in the endpoint descriptor.

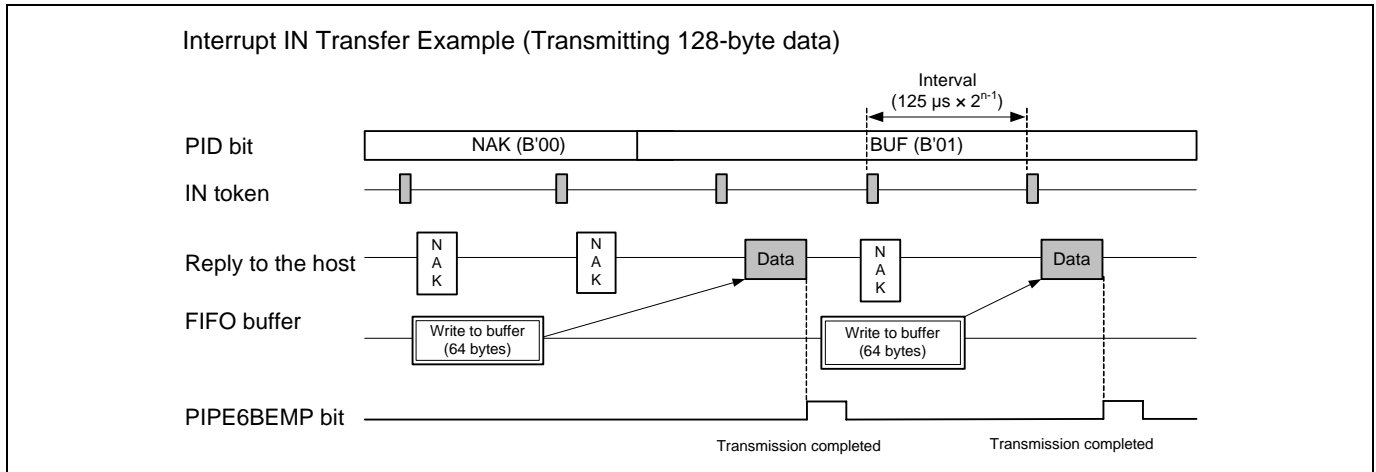


Figure 7 Interrupt IN Transfer Example

(1) Double buffering (DBLB bit)

Double buffering ensures an efficient transmission of large amounts of data. Buffer is operated either by the USB module or the CPU. When using single buffering, the USB module cannot access buffer while the CPU (or DMAC) accesses the buffer. While the CPU accesses the buffer, double buffering allows the USB module to access another buffer to execute the USB transfer efficiently.

Note: Double buffering cannot be used in interrupt transfer.

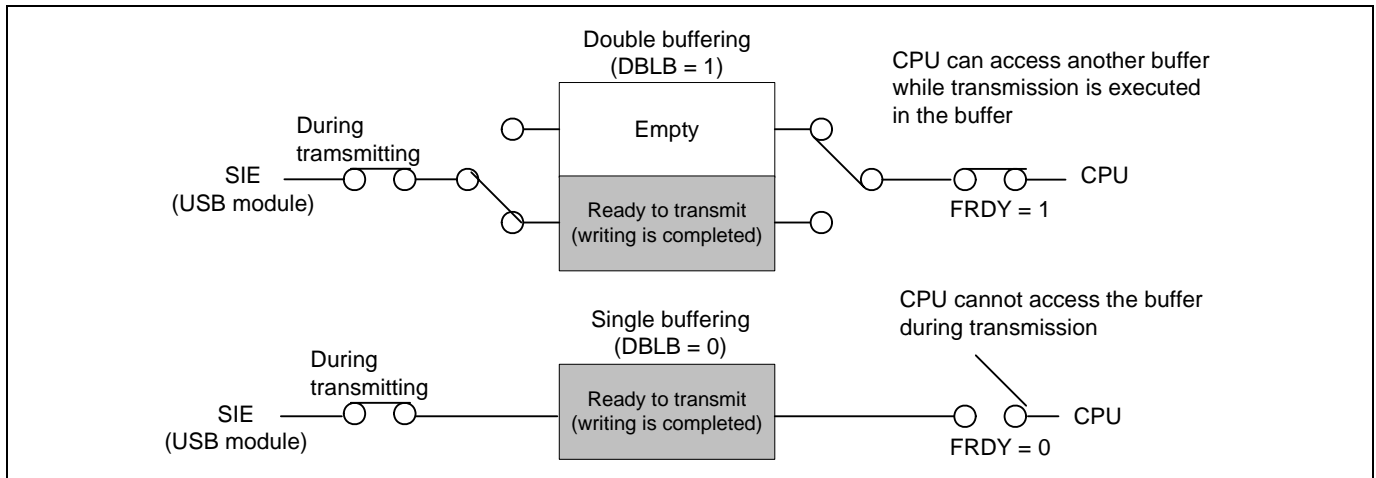


Figure 8 Double Buffering

(2) Continuous transfer mode (CNTMD bit)

Use the continuous transfer mode to transmit or receive multiple transactions continuously. In a single transfer, interrupts are generated when the size of the transmitted or received data reaches the maximum packet size. In a continuous transfer, however, data can be transferred without interrupts to CPU until the size of data reaches the buffer size allocated to each pipe. If the size of data is smaller than the buffer size, set the BVAL bit to 1 to transmit the data.

Note: Continuous transfer mode cannot be used in interrupt transfer.

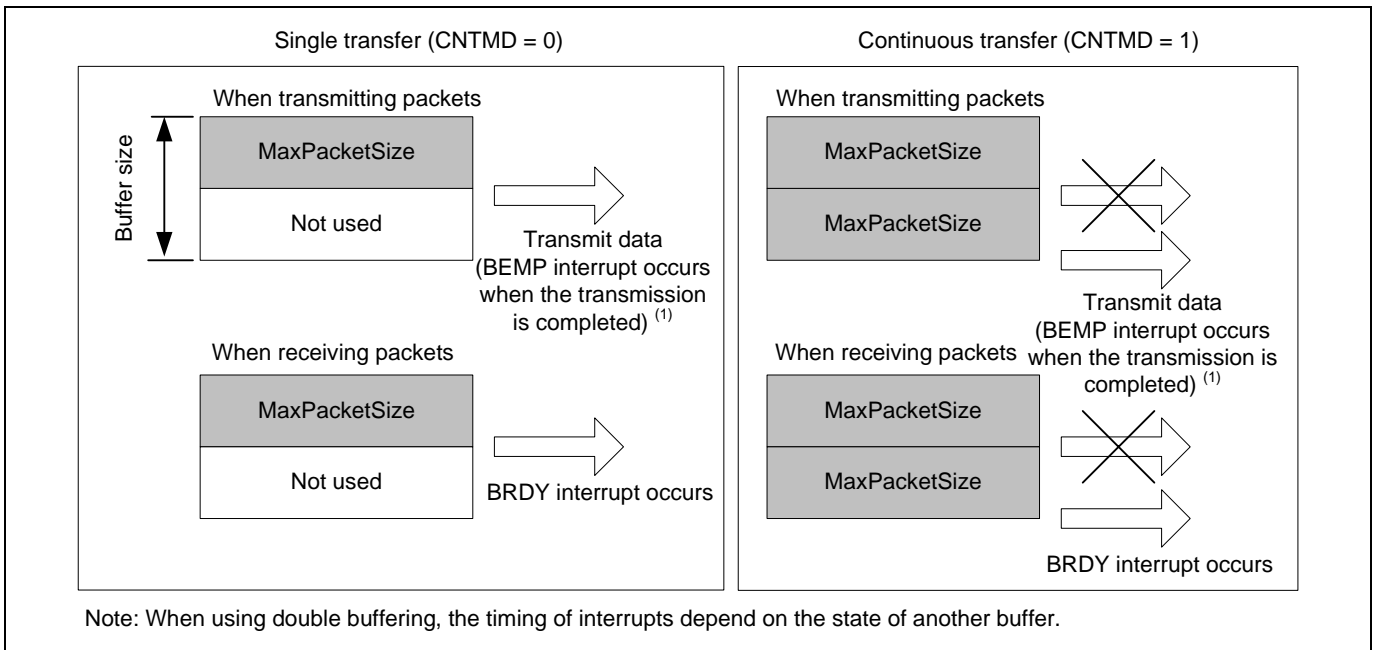


Figure 9 Continuous Transfer Mode

(3) Endpoint Number (EPNUM bit)

Specify the same value as the value of the corresponding endpoint descriptor.

(4) Maximum packet size (PIPEMAXP register)

Specify between 1 and 64 bytes, Note that the USB defines the allowable maximum packet size to be between 1 to 1024 bytes for interrupt transfer, however the SH7264 USB module supports up to 64 bytes.

(5) Buffer size and the number of the first block in the buffer (PIPEBUF register)

Figure 10 shows the setting example of the buffer size and the block number. To use the pipe, allocate the area from the USB module internal FIFO buffer memory. Specify the first block number and the number of blocks in units of 64-byte blocks as the area. Specify the first block number in the BUFNMB bit, and the value of the number of blocks to allocate -1 in the BUFSIZE bit. As pipes 6 to 9 are allocated to blocks 4 to 7, respectively, other blocks cannot be used.

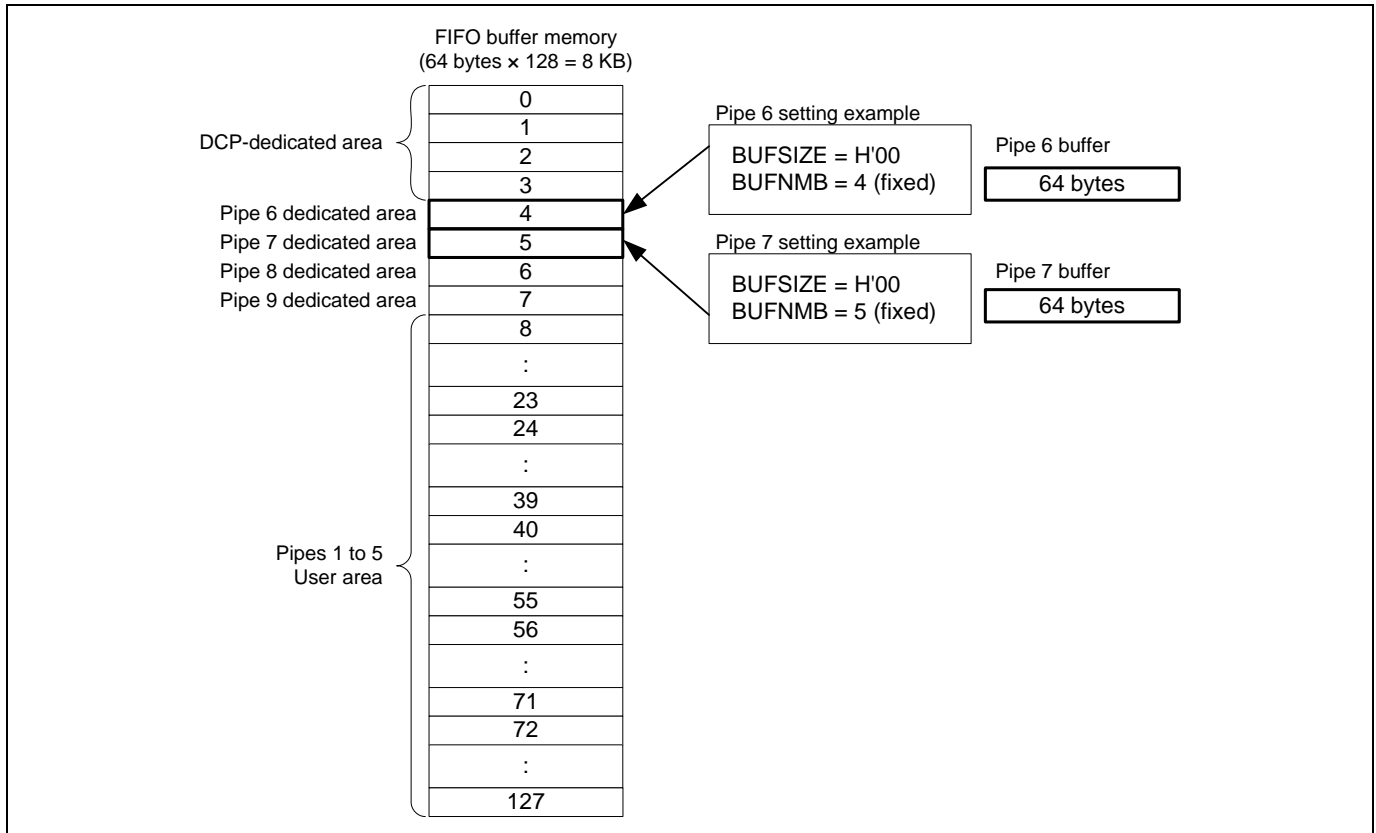


Figure 10 Buffer Size and Block Number Setting

(6) Enabling interrupts (BRDYENB register, BEMPENB register)

Figure 11 shows the interrupt timing in transmission. As interrupt IN transfer allows the single buffering only, both BRDY and BEMP interrupts occur at the same time.

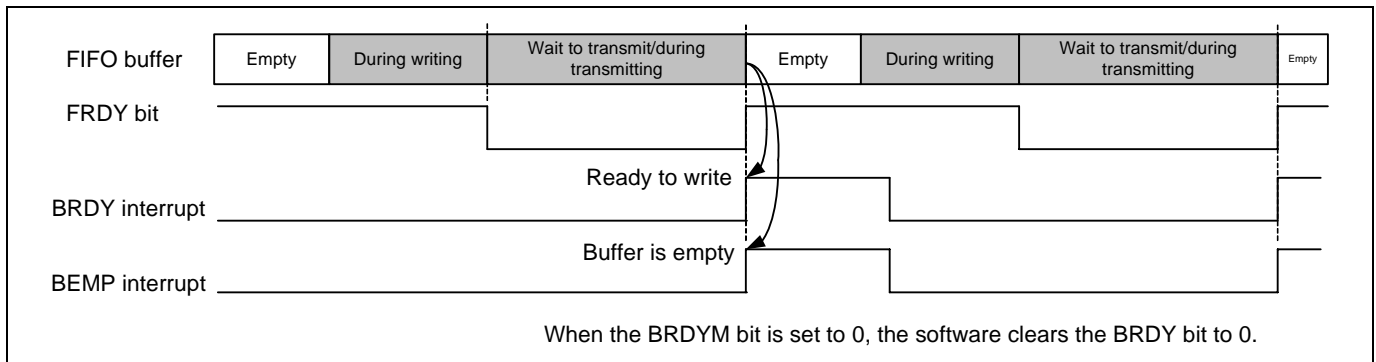


Figure 11 Interrupt Timing in Transmission

2.3.4 Receive Pipe Setting (Interrupt OUT Transfer)

This section describes an example of pipe setting when receiving data in interrupt OUT transfer.

Table 2 lists the setting example of interrupt OUT transfer using pipe 7. Figure 12 shows the operation example using the setting listed in Table 2. As double buffering and continuous transfer mode cannot be used in this example, use BRDY interrupt to read transactions one by one from FIFO buffer memory. For details, refer to (1) to (7) described in following pages.

Table 2 Interrupt OUT Transfer Setting Example Using Pipe 7

Register Name	Setting	Description	
PIPESEL register	H'0007	Set pipe 7 as the target pipe	
PIPECFG register	H'8007	TYPE [1:0] bits = 2	Set the transfer type as interrupt transfer
		BFRE bit = 0	BRDY interrupt when writing and reading data is completed (This bit is not enabled in interrupt transfer)
		DBLB bit = 0	Single buffering (This bit is not enabled in interrupt transfer)
		CNTMD bit = 0	Non-continuous transfer mode (This bit is not enabled in interrupt transfer)
		SHTNAK bit = 0	Continues the pipe when a transfer is completed (This bit is not enabled in interrupt transfer)
		DIR bit = 0	Set the transfer direction as receive
		EPNUM bit = 7	Set the endpoint number as 7
PIPEBUF register	H'0005	BUFSIZE [4:0] bits = B'00000	Specify the buffer size as 64 bytes
		BUFNMB [6:0] bits = 5	Specify the first block of a buffer as 5
PIPEMAXP register	H'0040	Specify the maximum packet size as 64 bytes	
PIPEPERI register	H'0000	IFIS bit = 0	(Only 0 can be specified in interrupt transfer)
		IITV [2:0] bits = 0	(Only 0 can be specified in interrupt transfer)
BRDYENB register	PIPE4BRDYE bit = 1	BRDY interrupt is enabled	
NRDYENB register	PIPE4NRDYE bit = 0	NRDY interrupt is disabled	
BEMPENB register	PIPE4BEMPE bit = 0	BEMP interrupt is disabled	
SOFCFG register	BRDYM bit = 0	Clearing the BRDY interrupt status automatically is disabled	

Note: The USB module executes interrupt transfer according to the cycle controlled by the USB host. Specify the interval by the `blInterval` field in the endpoint descriptor.

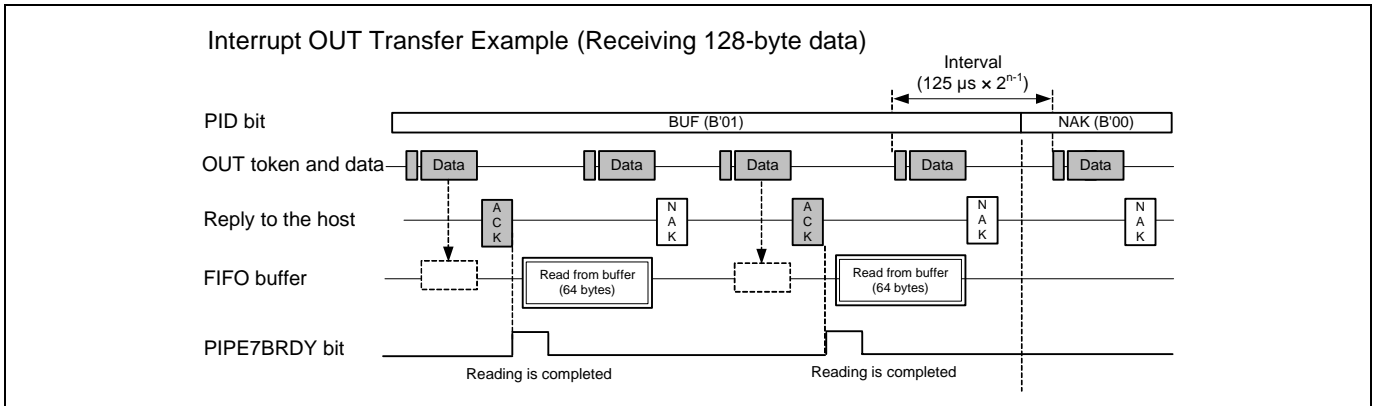


Figure 12 Interrupt OUT Transfer Example

(1) Disabling pipes when transfer is completed (SHTNAK bit)

When setting the SHTNAK bit to disable pipes when a transfer is completed, the USB module automatically changes the PID bit to NAK on receiving transfer is completed, which facilitates the transfer processing. When setting the SHTNAK of the receive pipe to 1, the USB module sets the PID bit corresponding to the target pipe to NAK. The USB module determines that the transfer is completed when the following conditions are satisfied. These functions can be used with pipes 1 to 5, receive pipes.

- When receiving the short packet data (including zero-length packet) correctly
- Using the transaction counter to receive packets of transaction counted correctly

Note: This setting (SHTNAK bit = 1) cannot be used in interrupt transfer.

(2) Enabling interrupts (BRDYENB register)

Figure 13 shows the interrupt timing in reception.

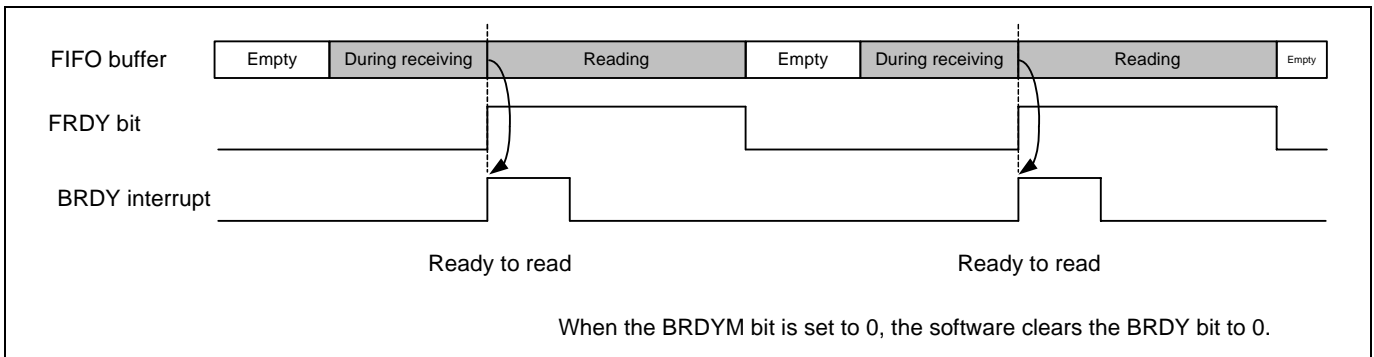


Figure 13 Interrupt Timing in Reception

- (3) Double buffering (DBLB bit)
Apply the same setting as the transmission. See 2.3.3 for details.
- (4) Continuous transfer mode (CNTMD bit)
Apply the same setting as the transmission. See 2.3.3 for details.
- (5) Maximum packet size (PIPEMAXP register)
Apply the same setting as the transmission. See 2.3.3 for details.
- (6) Endpoint number (EPNUM bit)
Apply the same setting as the transmission. See 2.3.3 for details.
- (7) Buffer size and the number of the first block in the buffer
Apply the same setting as the transmission. See 2.3.3 for details.

2.4 FIFO Port

Use FIFO port to access (read or write data) the FIFO buffer memory allocated to pipes. This section describes how to access the FIFO buffer memory.

2.4.1 FIFO Port Overview

Figure 14 shows an overview of the FIFO port. The FIFO port has three registers (C/DnFIFO port registers). Specify the pipe number in the CURPIPE bit in the C/DnFIFOSEL register to access the FIFO buffer memory allocated to the specified pipe via the C/DnFIFO port register. Specify the access bit width and endianness in the C/DnFIFOSEL register. The C/DnFIFOCTR register indicates the write end in the buffer memory, and clears buffer.

Make sure to check the setting in the FRDY bit in the C/DnFIFOCTR register before accessing the C/DnFIFO port register, since the FIFO buffer memory may be operated by the system (CPU) or by the USB module (SIE). See bits BSTS and INBUFM in the DCPCTR register and the PIPEnCTR register to check the buffer status in each pipe.

The DCP buffer can be allocated only to the CFIFO port register. The DMA transfer can be used in the D0FIFO port register and the D1FIFO port register.

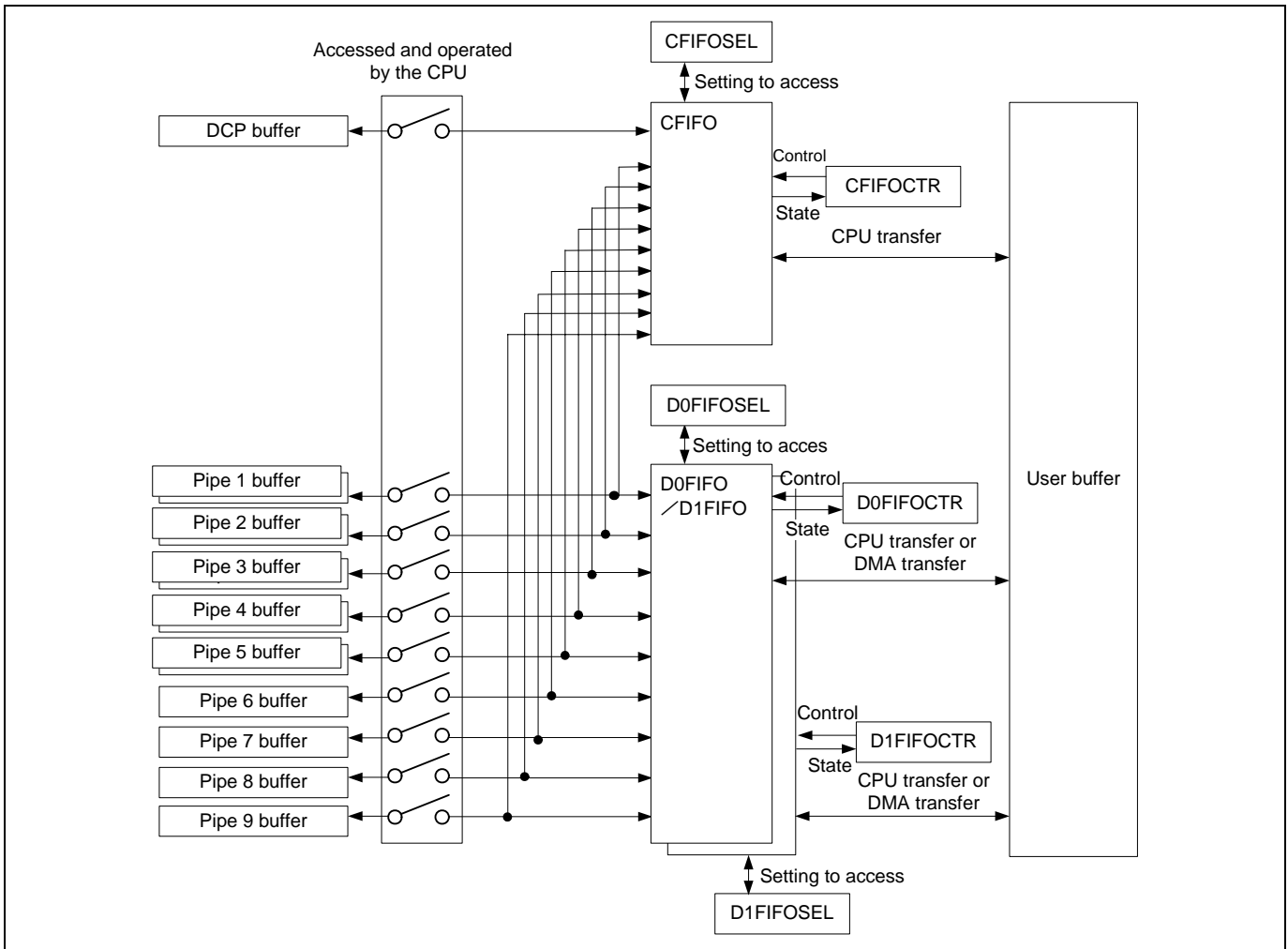


Figure 14 FIFO Port (Overview)

2.4.2 Writing Data to the FIFO Port (Interrupt IN Transfer)

The DnFIFO port register can be used to execute the DMA transfer in Interrupt IN transfer, however, this section describes the procedures on writing data to the CFIFO port register using the CPU transfer.

Figure 15, Figure 16, and Figure 17 show flow charts of writing data to FIFO port.

As the CFIFOSEL register is also used by the default control pipe, the register value may be rewritten depending on the interrupt used by control transfer. When rewriting the CFIFOSEL register by the interrupt, write back the value as appropriate.

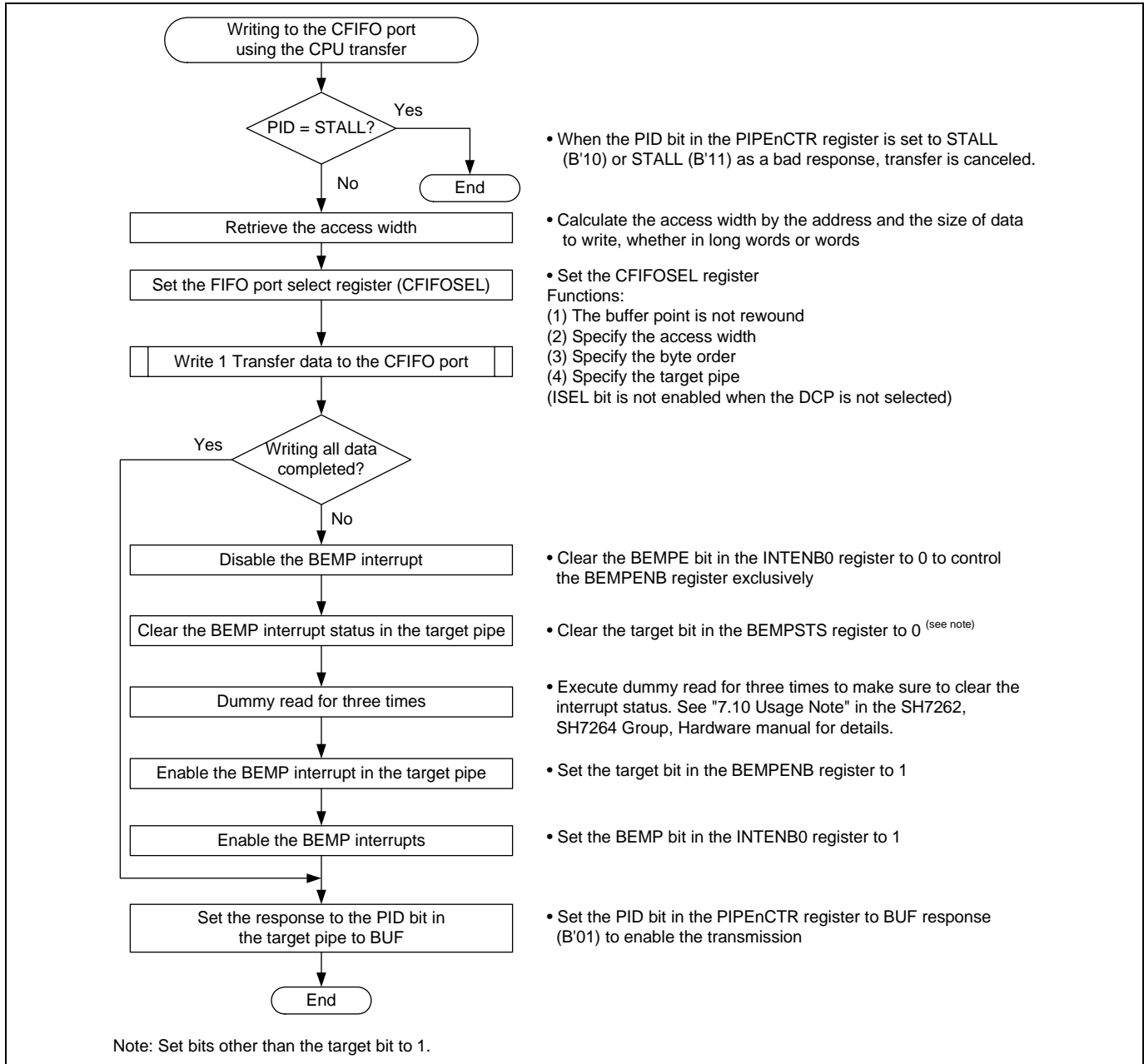


Figure 15 Writing Data to the FIFO Port

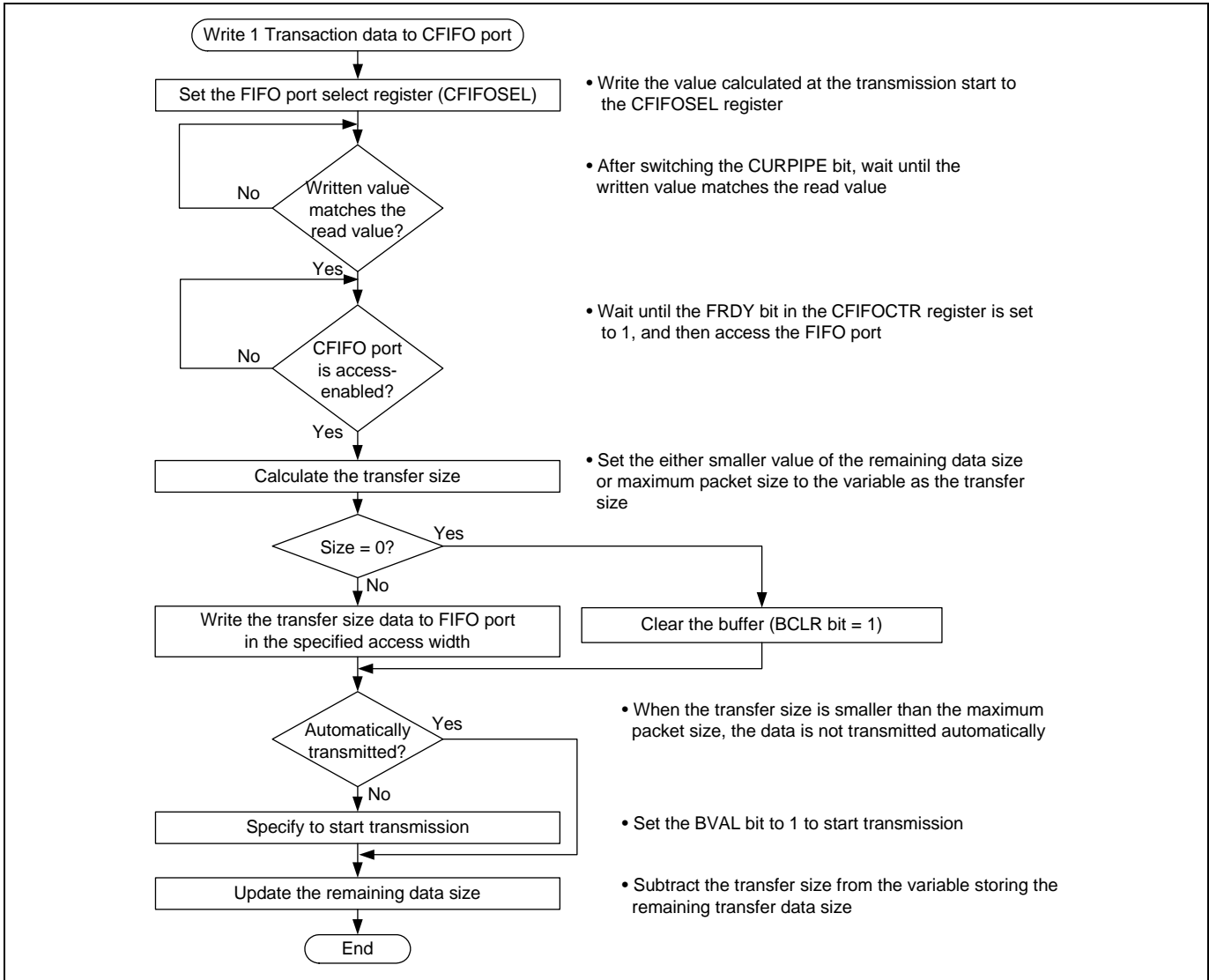


Figure 16 Writing 1 Transaction Data to CFIFO Port

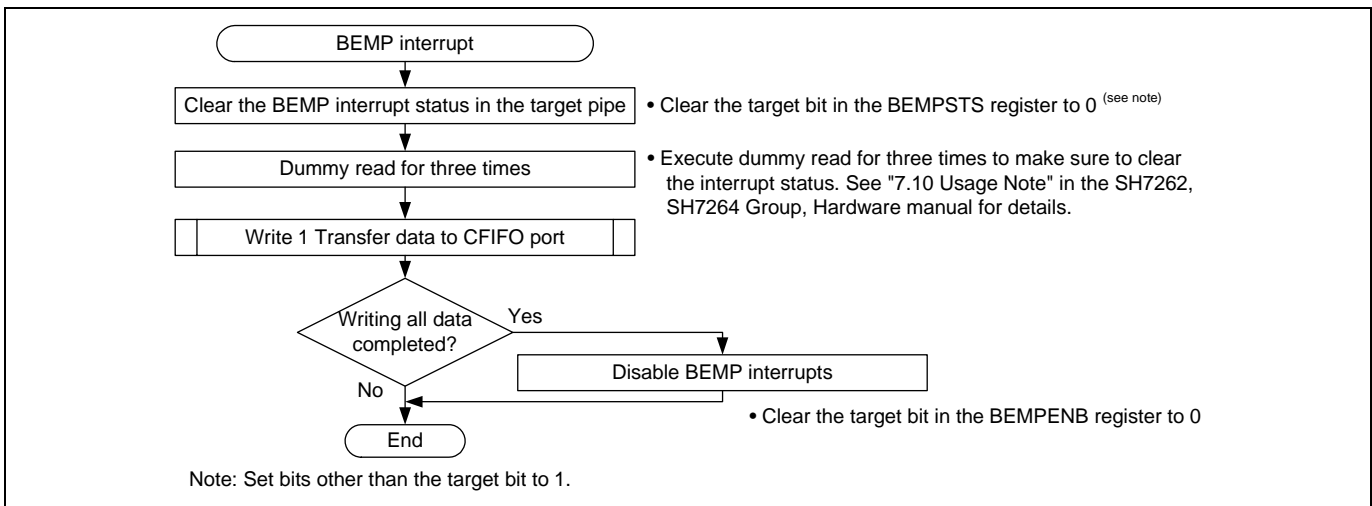


Figure 17 BEMP Interrupt Example

2.4.3 Reading Data from the FIFO Port (Interrupt OUT Transfer)

The DnFIFO port register can be used to execute the DMA transfer in Interrupt OUT transfer, however, this section describes the procedures on reading data from the CFIFO port register using the CPU transfer.

Figure 18 shows an example of reading data from FIFO port. Figure 19 shows an example of the BRDY interrupt.

As the CFIFOSEL register is also used by the default control pipe, like interrupt OUT transfer, the register value may be rewritten depending on the interrupt used by control transfer. When rewriting the CFIFOSEL register by the interrupt, write back the value as appropriate.

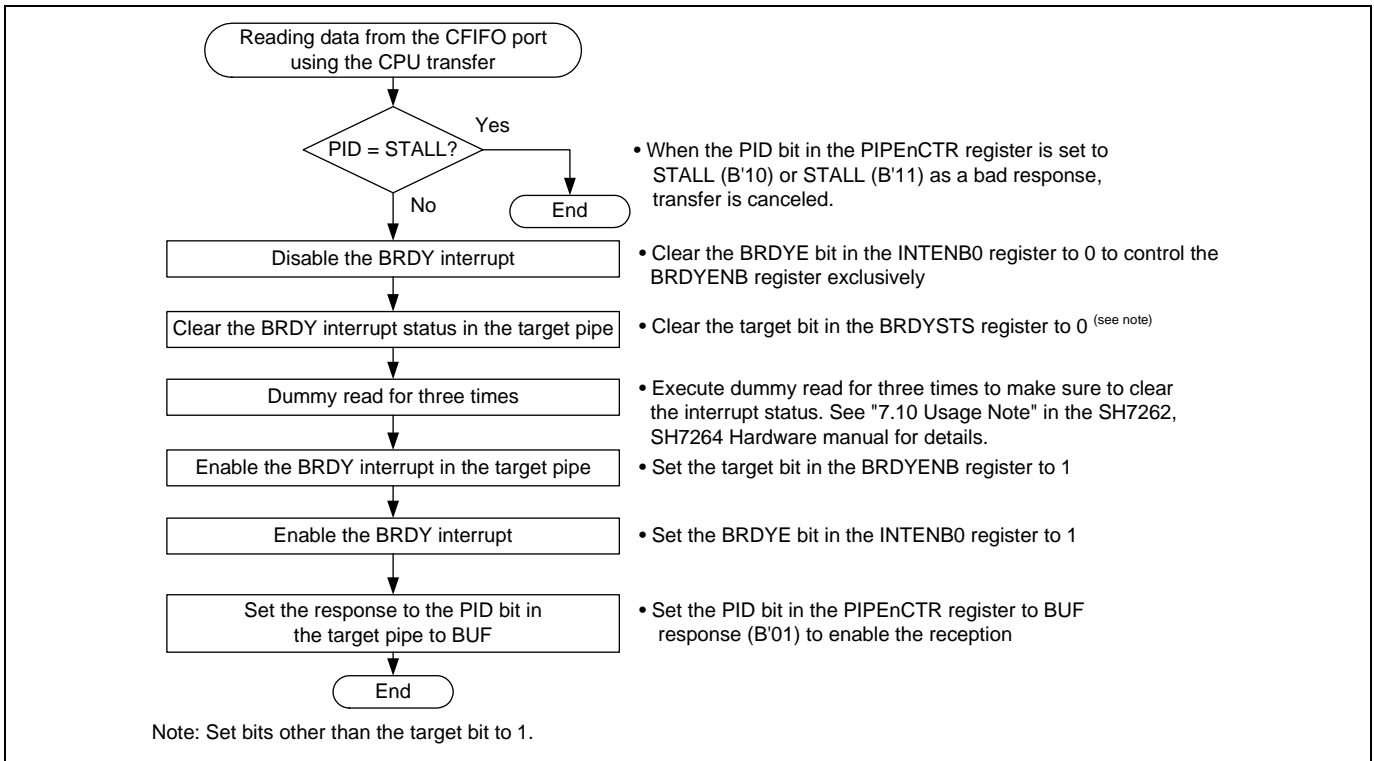
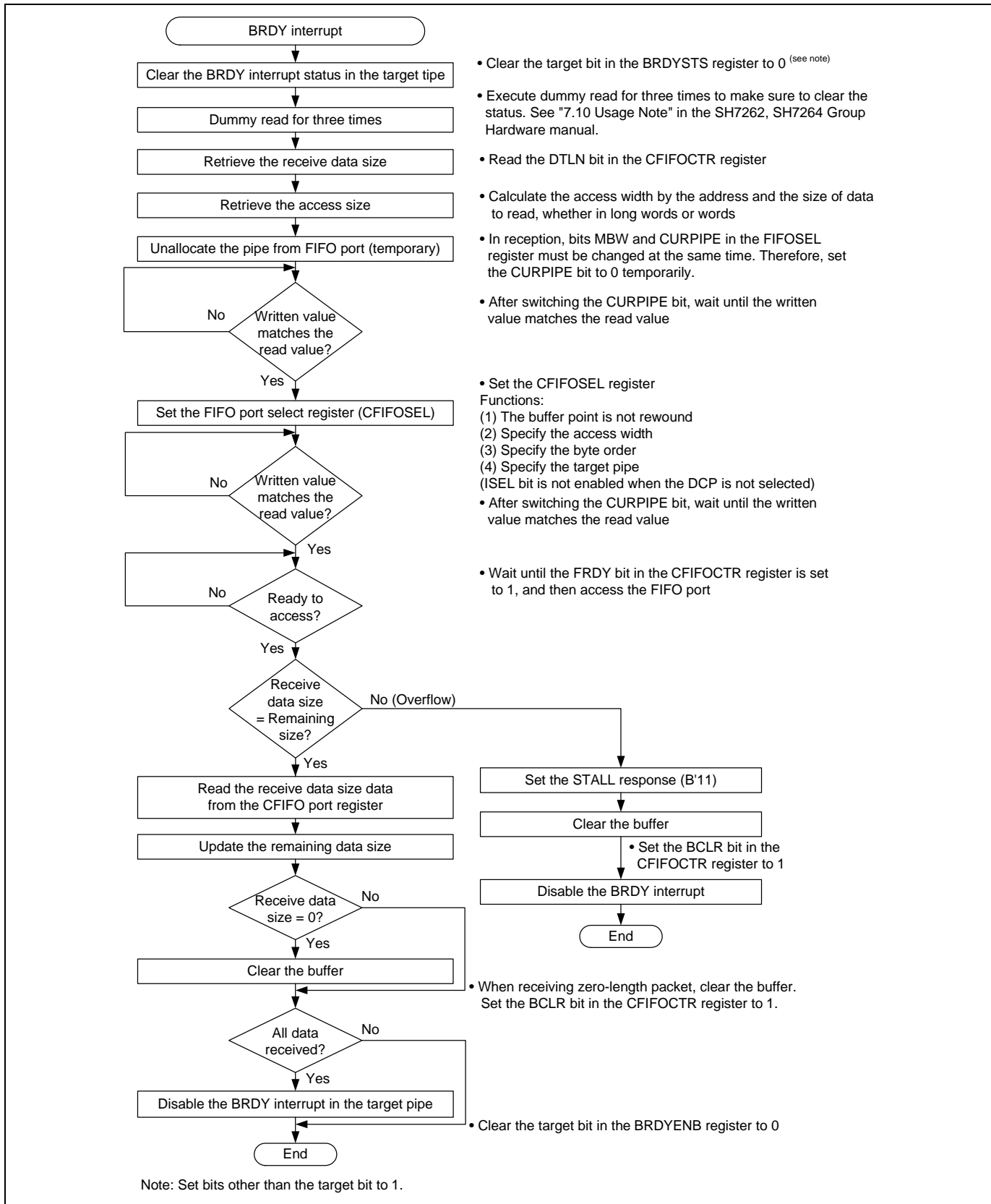


Figure 18 Reading Data from the FIFO Port



- Clear the target bit in the BRDYSTS register to 0 (see note)
- Execute dummy read for three times to make sure to clear the status. See "7.10 Usage Note" in the SH7262, SH7264 Group Hardware manual.
- Read the DTLN bit in the CFIFOCTR register
- Calculate the access width by the address and the size of data to read, whether in long words or words
- In reception, bits MBW and CURPIPE in the FIFOSEL register must be changed at the same time. Therefore, set the CURPIPE bit to 0 temporarily.
- After switching the CURPIPE bit, wait until the written value matches the read value
- Set the CFIFOSEL register Functions:
 - (1) The buffer point is not reword
 - (2) Specify the access width
 - (3) Specify the byte order
 - (4) Specify the target pipe (ISEL bit is not enabled when the DCP is not selected)
- After switching the CURPIPE bit, wait until the written value matches the read value
- Wait until the FRDY bit in the CFIFOCTR register is set to 1, and then access the FIFO port
- When receiving zero-length packet, clear the buffer. Set the BCLR bit in the CFIFOCTR register to 1.
- Clear the target bit in the BRDYENB register to 0

Note: Set bits other than the target bit to 1.

Figure 19 BRDY Interrupt

3. References

- Software Manual
SH-2A/SH-2A-FPU Software Manual Rev. 3.00
The latest version of the software manual can be downloaded from the Renesas website.
- Hardware Manual
SH7262 Group, SH7264 Group Hardware Manual Rev. 2.00
The latest version of the hardware manual can be downloaded from the Renesas website.
- USB 2.0 Specifications
Universal Serial Bus Specification Revision 2.00
(<http://www.usb.org/developers>)

Website and Support

Renesas Technology Website
<http://www.renesas.com/>

Inquiries
<http://www.renesas.com/inquiry>
csc@renesas.com

Revision History

Rev.	Date	Description	
		Page	Summary
1.00	Feb. 12, 2010	—	First edition issued

All trademarks and registered trademarks are the property of their respective owners.

Notes regarding these materials

1. This document is provided for reference purposes only so that Renesas customers may select the appropriate Renesas products for their use. Renesas neither makes warranties or representations with respect to the accuracy or completeness of the information contained in this document nor grants any license to any intellectual property rights or any other rights of Renesas or any third party with respect to the information in this document.
2. Renesas shall have no liability for damages or infringement of any intellectual property or other rights arising out of the use of any information in this document, including, but not limited to, product data, diagrams, charts, programs, algorithms, and application circuit examples.
3. You should not use the products or the technology described in this document for the purpose of military applications such as the development of weapons of mass destruction or for the purpose of any other military use. When exporting the products or technology described herein, you should follow the applicable export control laws and regulations, and procedures required by such laws and regulations.
4. All information included in this document such as product data, diagrams, charts, programs, algorithms, and application circuit examples, 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 products listed in this document, please confirm the latest product information with a Renesas sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas such as that disclosed through our website. (<http://www.renesas.com>)
5. Renesas has used reasonable care in compiling the information included in this document, but Renesas assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the information included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the information in light of the total system before deciding about the applicability of such information to the intended application. Renesas makes no representations, warranties or guaranties regarding the suitability of its products for any particular application and specifically disclaims any liability arising out of the application and use of the information in this document or Renesas products.
7. With the exception of products specified by Renesas as suitable for automobile applications, Renesas products are not designed, manufactured or tested for applications or otherwise in systems the failure or malfunction of which may cause a direct threat to human life or create a risk of human injury or which require especially high quality and reliability such as safety systems, or equipment or systems for transportation and traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea communication transmission. If you are considering the use of our products for such purposes, please contact a Renesas sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth above.
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listed below:
 - (1) artificial life support devices or systems
 - (2) surgical implantations
 - (3) healthcare intervention (e.g., excision, administration of medication, etc.)
 - (4) any other purposes that pose a direct threat to human life

Renesas shall have no liability for damages arising out of the uses set forth in the above and purchasers who elect to use Renesas products in any of the foregoing applications shall indemnify and hold harmless Renesas Technology Corp., its affiliated companies and their officers, directors, and employees against any and all damages arising out of such applications.
9. You should use the products described herein within the range specified by Renesas, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas shall have no liability for malfunctions or damages arising out of the use of Renesas products beyond such specified ranges.
10. Although Renesas endeavors to improve the quality and reliability of its products, IC products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Please be sure to implement safety measures to guard against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas 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 applicable measures. Among others, since the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
11. In case Renesas products listed in this document are detached from the products to which the Renesas products are attached or affixed, the risk of accident such as swallowing by infants and small children is very high. You should implement safety measures so that Renesas products may not be easily detached from your products. Renesas shall have no liability for damages arising out of such detachment.
12. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written approval from Renesas.
13. Please contact a Renesas sales office if you have any questions regarding the information contained in this document, Renesas semiconductor products, or if you have any other inquiries.