OKI

ML60851A

Application Manual

USB Device Controller ASSP

FIRST EDITION

ISSUE DATE: Feb. 1999

NOTICE

1. The information contained herein can change without notice owing to product and/or technical improvements. Before using the product, please make sure that the information being referred to is up-to-date.

2. The outline of action and examples for application circuits described herein have been chosen as an explanation for the standard action and performance of the product. When planning to use the product, please ensure that the external conditions are reflected in the actual circuit, assembly, and program designs.

3. When designing your product, please use our product below the specified maximum ratings and within the specified operating ranges including, but not limited to, operating voltage, power dissipation, and operating temperature.

4. Oki assumes no responsibility or liability whatsoever for any failure or unusual or unexpected operation resulting from misuse, neglect, improper installation, repair, alteration or accident, improper handling, or unusual physical or electrical stress including, but not limited to, exposure to parameters beyond the specified maximum ratings or operation outside the specified operating range.

5. Neither indemnity against nor license of a third party's industrial and intellectual property right, etc. is granted by us in connection with the use of the product and/or the information and drawings contained herein. No responsibility is assumed by us for any infringement of a third party's right which may result from the use thereof.

6. The products listed in this document are intended for use in general electronics equipment for commercial applications (e.g., office automation, communication equipment, measurement equipment, consumer electronics, etc.). These products are not authorized for use in any system or application that requires special or enhanced quality and reliability characteristics nor in any system or application where the failure of such system or application may result in the loss or damage of property, or death or injury to humans.
   Such applications include, but are not limited to, traffic and automotive equipment, safety devices, aerospace equipment, nuclear power control, medical equipment, and life-support systems.

7. Certain products in this document may need government approval before they can be exported to particular countries. The purchaser assumes the responsibility of determining the legality of export of these products and will take appropriate and necessary steps at their own expense for these.

8. No part of the contents cotained herein may be reprinted or reproduced without our prior permission.

9. MS-DOS is a registered trademark of Microsoft Corporation.

Printed in Japan

CONTENTS

1. OVERVIEW

The ML60851A is a Universal Serial Bus (USB) device controller and is intended mainly to be installed on the peripheral unit side. When used in such applications, the ML60851A is functionally placed at the front end of the peripheral unit and is connected to the host PC via a USB cable.

Further, usually, the ML60851A is connected to the MPU bus of the one-chip microcontroller, etc., inside the peripheral unit. In this mode of application, a firmware designed to implement operating functions suitable to the peripheral unit is also incorporated.

This application manual gives descriptions that support the system design and firmware design of the peripheral unit incorporating the ML60851A.

2. END POINT ADDRESS

◆ The ML60851A can have four end point addresses (EP).

End point address	FIFO Number	FIFO Size	Transmit/ Receive	Transfer mode	Corresponding control register
0	0	8 Bytes	Receive	C	EP0RXCON
0	4	8 Bytes	Transmit	C	EP0TXCON
1	1	64 Bytes × 2	Transmit or receive	B	EP1CON
2	2	64 Bytes	Transmit or receive	B	EP2CON
3	3	8 Bytes	Transmit	Int	EP3CON

Transfer mode:
C = Control, B = Bulk, Int = Interrupt

◆ EP0 is only for control transfer

◆ The direction of transfer can be selected for EP1 and EP2 by writing data to the corresponding control register.

◆ The FIFO of EP1 has a two layer configuration. The ML60851A stores the packet data by automatically switching between the two stages of 64-Byte FIFOs.

3. REGISTER DEFINITIONS

3.1 Register types

- Data FIFO
- Setup registers
- Interrupt registers
- DMA, control and polarity selection registers
- Common USB registers
- EP0 control and status registers
- EP1 control and status registers
- EP2 control and status registers
- EP3 control and status registers

3.2 Data FIFO

Symbol	Name	R/W
EP0RXFIFO	EP0 receive FIFO data	R
EP1RXFIFO	EP1 receive FIFO data	R
EP2RXFIFO	EP2 receive FIFO data	R
EP0TXFIFO	EP0 transmit FIFO data	W
EP1TXFIFO	EP1 transmit FIFO data	W
EP2TXFIFO	EP2 transmit FIFO data	W
EP3TXFIFO	EP3 transmit FIFO data	W

3.3 Setup registers

Symbol	Name	R/W
bmRequestType	bmRequestType	R
bRequest	bRequest	R
wValueLSB	wValueLSB	R
wValueMSB	wValueMSB	R
wIndexLSB	wIndexLSB	R
wIndexMSB	wIndexMSB	R
wLengthLSB	wLengthLSB	R
wLengthMSB	wLengthMSB	R

3.4 Interrupt registers

There are eight interrupt factors in the ML60851A. The bits of the interrupt enable register and the interrupt status register correspond to individual factors and it is possible to mask each interrupt factor.

Symbol	Name	R/W
INTENBL	Interrupt enable register	R/W
INTSTAT	Interrupt status register	R

3.5 DMA, control and polarity selection registers

These are the registers for setting the hardware operating conditions of the ML60851A such as the DMA operating mode, etc.

Symbol	Name	R/W
DMACON	DMA Control register (DMA Operation, addressing mode, data width, etc., are set in this register.)	R/W
DMAINTVL	DMA Interval register	R/W
POLSEL	Polarity selection register (The polarities of the $\overline{\text{INTR}}$, $\overline{\text{DREQ}}$, and DACK pins are set here.)	R/W

3.6 Common USB registers

Symbol	Name	R/W
DVCADR	Device address register (stores the device address)	R/W
DVCSTAT	Device state register	R/W
PKTERR	Packet error register (normally no referencing is needed)	R
FIFOSTAT1	FIFO Status register 1 (Indicates the full/empty status of FIFO0, 1)	R
FIFOSTAT2	FIFO Status register 2 (Indicates the full/empty status of FIFO0, 2, 3)	R
PKTRDY	EP Packet ready register (Packet ready setting and indication of EP0, 1, 2, 3)	R/W
CLRFIFO	Transmit FIFO clear register (The transmit FIFO is cleared by $\overline{\text{WR}}$)	W
SYSCON	System control register (Software reset and oscillation stop)	W

3.7 EP0 control and status registers

Symbol	Name	R/W
EP0RXCNT	EP0 receive byte count register (counts the number of bytes in the receive packet)	R
EP0RXCON	EP0 receive control register (configuration bit, transfer type, EP address)	R
EP0RXTGL	EP0 receive data toggle register	R
EP0RXPLD	EP0 receive payload register (maximum packet size)	R/W
EP0TXCON	EP0 transmit control register (configuration bit, transfer type, EP address)	R
EP0TXTGL	EP0 transmit data toggle register	R
EP0TXPLD	EP0 transmit payload register (invalid, general purpose register)	R/W
EP0STAT	EP0 status register (setup ready, stall bit, stage)	R/W

3.8 EP1 control and status registers

Symbol	Name	R/W
EP1RXCNT	EP1 receive byte count register (counts the number of bytes in the receive packet)	R
EP1CON	EP1 control register (configuration bit, transfer type, EP address)	R/W
EP1TGL	EP1 data toggle register	R/W
EP1PLD	EP1 payload register (maximum packet size)	R/W

3.9 EP2 control and status registers

Symbol	Name	R/W
EP2RXCNT	EP2 receive byte count register (counts the number of bytes in the receive packet)	R
DP2CON	EP2 control register (configuration bit, transfer type, EP address)	R/W
EP2TGL	EP2 data toggle register	R/W
EP2PLD	EP2 payload register (maximum packet size)	R/W

3.10 EP3 control and status registers

Symbol	Name	R/W
EP3CON	EP3 control register (configuration bit, transfer type, EP address)	R/W
EP3TGL	EP3 data toggle register	R/W
EP3PLD	EP3 payload register (invalid, general purpose register)	R/W

4. REGISTER INITIAL SETTINGS

4.1 Register address

The ML60851A has read-only registers, write-only registers, and read-write registers that can be both read out and written into. The read-write registers have different addresses during reading and writing.

4.2 Setting the hardware operating conditions

The registers for setting the following hardware operating conditions are specified by the local MCU.

- Pin polarity setting: Polarity selection register (POLSEL)

  The polarities of the $\overline{\text{INTR}}$, $\overline{\text{DREQ}}$, and DACK pins can be set.

- Setting the DMA operating condition setting:

  DMA control register (DMACON)

  DMA interval register (DMAINTVL)

  Setting of the DMA "used" or "not used" condition, the DMA transfer mode, etc., can be made.

4.3 Settings based on standard requests

Eleven types of standard requests have been defined in the USB Standard 1.0. The standard requests are issued by the host via the control pipe. Among the standard requests, the following 5 types of requests ask the local MCU to write the operating conditions to the different registers of the ML60851A. The local MCU analyzes the contents of the request at the status stage of control transfer, and if the request is one of the following requests, the MCU writes the setting conditions to the corresponding registers of the ML60851A as follows.

Request name	Related register	Description
CLEAR_FEATURE	Stall bits corresponding to each EP	That EP is not stalled.
SET_FEATURE	Stall bits corresponding to each EP	That EP is stalled.
SET_ADDRESS	Device address register	The device address is set.
SET_CONFIGURATION	Control register and payload register of each EP	The configuration of the EP is set.
SET_INTERFACE	Control register and payload register of each EP	The alternative interface setting is selected.

5. MCU INTERFACE

5.1 Normal interrupts

The MCU for control inside the peripheral unit (the local MCU) recognizes an interrupt request when the $\overline{\text{INTR}}$ signal of the ML60851A is asserted. The interrupt factor can be identified by checking the contents of the interrupt status register (INTSTAT).

The relationship between the interrupt factors and the bits of the interrupt status register are as follows.

Bit	Interrupt factor
D0	Setup ready interrupt status
D1	EP1 packet ready interrupt status
D2	EP2 packet ready interrupt status
D3	EP0 receive packet ready interrupt status
D4	EP0 transmit packet ready interrupt status
D5	USB Bus reset interrupt status
D6	Suspended state interrupt status
D7	EP3 packet ready interrupt status

(1) Setup ready

When the host computer has issued a device request using the control pipe and 8 bytes of setup control data have been written normally in the setup registers of the ML60851A, the ML60851A sets the bit D0 of EP0STAT and the bit D0 of the interrupt status register to "1" to assert the $\overline{\text{INTR}}$ pin.

This prompts the local MCU to analyze the contents of the device request from the host computer.

(2) Transmit packet ready

The communication between the ML60851A and the local MCU during data transmission in the different transfer modes is done via the transmit packet ready. The transmit packet ready bits are bits D4 to D7 of the register PKTRDY which correspond to the different end point addresses.

The local MCU should assert the transmit packet ready bit in PKTRDY. After it has written the data to be transmitted to the transmit FIFO of the corresponding EP. When the transmit packet ready has been asserted, the ML60851A transmits the data in the transmit FIFO to the host computer after converting the data into packets.

When the host receives without errors the packets transmitted by the ML60851A, the host returns the ACK packet message to the ML60851A. Upon receiving this ACK message, the ML60851A deasserts the transmit packet ready and asserts $\overline{\text{INTR}}$ again.

The last transmit packet becomes either a short packet or a null packet. When the ACK message comes back in response to the transmission of a short packet or a null packet and the transmit packet ready bit is deasserted, the transmission of all the data becomes complete.

The flow of data transmission varies depending on the transfer mode. also, in the case of EP1 transmission, there are two planes of FIFO allowing storage of data up to 128 bytes and hence highly flexible transmission is possible.

(3) Receive packet ready

The communication between the ML60851A and the local MPU during data reception in the different transfer modes is done via the receive packet ready. The receive packet ready bits are bits D0 to D2 of the register PKTRDY which correspond to the different end point addresses.

When the ML60851A has received one packet of data from the host and has stored it in the receive FIFO, it asserts the receive packet ready and asserts $\overline{\text{INTR}}$. Thereafter, the local MCU should read out the data from the receive FIFO. After one packet of receive data is read out, the local MCU should reset the receive packet ready.

When the receive packet ready bit is reset the ML60851A deasserts $\overline{\text{INTR}}$ and returns the ACK packet to the host.

When the ACK message comes back to the host computer, it transmits the next packet. The packets of data are transmitted by the host computer successively in this manner. The last receive data will be either a short packet or a null packet. The reception of all data will be completed when the local MCU reads out a short packet or a null packet from the receive FIFO and resets the receive packet ready status.

In the case of EP1, since there are two layers of FIFO allowing temporary storage of data up to 128 bytes, it is possible to carry out highly flexible data reception.

5.2 USB bus reset interrupt

The ML60851A recognizes the USB bus reset when SE0 continues for 2.5 µs or longer, and the ML60851A asserts the USB bus reset interrupt status bit of the interrupt status register (INTSTAT) and asserts $\overline{\text{INTR}}$.

These are not asserted when the USB bus reset interrupt enable bit in the interrupt enable register (INTENBL) has been deasserted.

The contents of processing during a USB bus reset is determined by the firmware.

5.3 Suspended state interrupt

When the idle state continues for 3 ms or longer, the ML60851A sets bit D6 of the interrupt status register (INTSTAT) to "1" and asserts the $\overline{\text{INTR}}$ pin. Thereafter, the local MCU can execute the processing of transition to the suspended state. The above bit D6 is reset to "0" when the state of the USB bus is resumed or reset.

5.4 Error processing and retry operation

(1) Error processing during transmission

Since the host does not return the ACK packet if the data transmitted by the ML60851A does not reach the host normally, the ML60851A does not reset the transmit packet ready. Therefore, since $\overline{\text{INTR}}$ is not asserted and the local MCU does not write the transmit data of the next packet, the ML60851A waits while retaining the current packet of data.

The ML60851A transmits the current packet of data again upon receiving the next IN token from the host computer.

(2) Error processing during reception

When the data received via the USB bus is not stored normally, the ML60851A does not assert the $\overline{\text{INTR}}$ signal to the local MCU and also sends the NAK packet to the host computer. Consequently, the local MCU does not read the data, and the host computer recognizes from the NAK packet that normal data transfer was not carried out and takes appropriate measures.

6. USB TRANSFER OPERATIONS

The operations of the USB in the different transfer modes are described below.

1) Control transfer

A device request is transferred from the host.

Depending on the type of request, this transfer is accompanied by the transmission or reception of control data as a target of the request.

(1) Control write transfer

- Setup data is transferred from the host computer to the ML60851A.
- Control data is transferred from the host computer to the ML60851A (reception).

(2) Control read transfer

- Setup data is transferred from the host computer to the ML60851A.
- Control data is transferred from the ML60851A to the host computer (transmission).

(3) Control transfer without data

- Only setup data is transferred from the host computer to the ML60851A.

2) Bulk-in transfer

- Bulk data is transferred from the ML60851A to the host computer (transmission).

3) Bulk-out transfer

- Bulk data is transferred from the host computer to the ML60851A (reception).

4) Interrupt transfer

- Interrupt data is transferred from the ML60851A to the host computer (transmission).

6.1 Control transfer

6.1.1 Outline of control transfer

◆ End point address

- The end point of control transfer is EP0.
- Both transmission and reception are possible and in either case, the end point address is 0.

◆ FIFO

- The ML60851A has 8 bytes each of FIFO for reception and for transmission individually.

◆ Setup data

- The 8 bytes of setup data transferred from the host computer in the setup stage of control transfer are received automatically and stored in the setup registers by the ML60851A. Although the ML60851A carries out the decoding related to the USB protocol, extracts the actual setup data and stores in the set of registers, it does not execute the commands corresponding to the contents of the device request.

◆ Management of control transfer operation flow

- The control transfer includes requests from the host, and operations corresponding to the request are made on the peripheral unit side. However, the progress of execution according to that operation flow is dependent on the firmware of the local MCU.

6.1.2 Types of control transfer

Control transfers are of the following three types:

- Control transfer without data,
- Control write transfer, and
- Control read transfer.

The type of control transfer to be made is determined depending on the contents of the request.

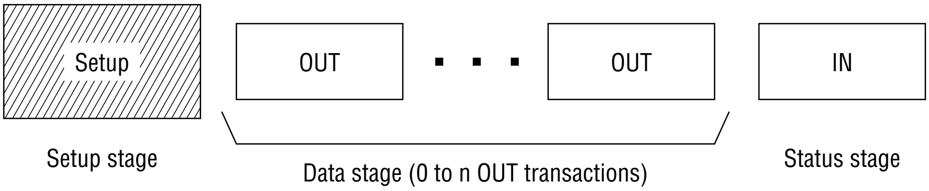
6.1.3 The three stages of control transfer

A control transfer is composed of the following three stages.

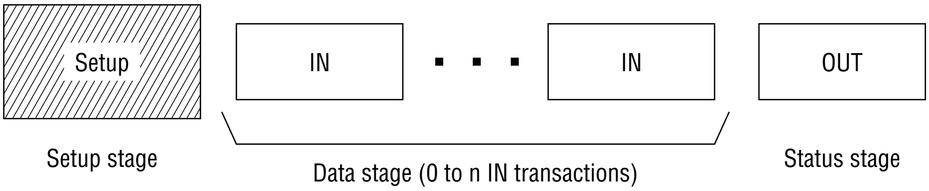
- In the setup stage, 8 bytes of setup data are transferred from the host computer.
- While the setup stage is followed by the data stage, there are also times when there is no data stage (control transfer without data). In the data stage, an optional number of OUT transactions are made in the case of a control write transfer. Likewise, an optional number of IN transactions are made in the case of a control read transfer.

  Packet data of the maximum packet size is transferred in each OUT or IN transaction excepting the last transaction. (The maximum packet size of EP0 is entered in offset 7 of the device descriptor.)
- In the status stage, the 8 bytes of data received during the setup stage described above are decoded and the corresponding command sequence is executed. The result of whether or not the command sequence was normally executed is returned to the host computer.

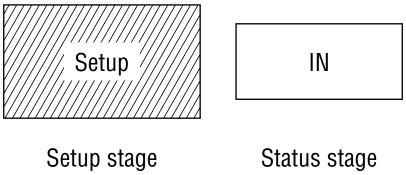
- Procedure for a control write transfer



- Procedure for a control read transfer



- Procedure for a control transfer without data



6.1.4 Report of the status result

In the status stage, the result of execution in the setup stage and data stage is reported to the host computer. The following three types of results are reported.

- The command sequence has been completed normally.
- The command sequence has not been completed.
- The function is in the busy state because command execution is in progress.

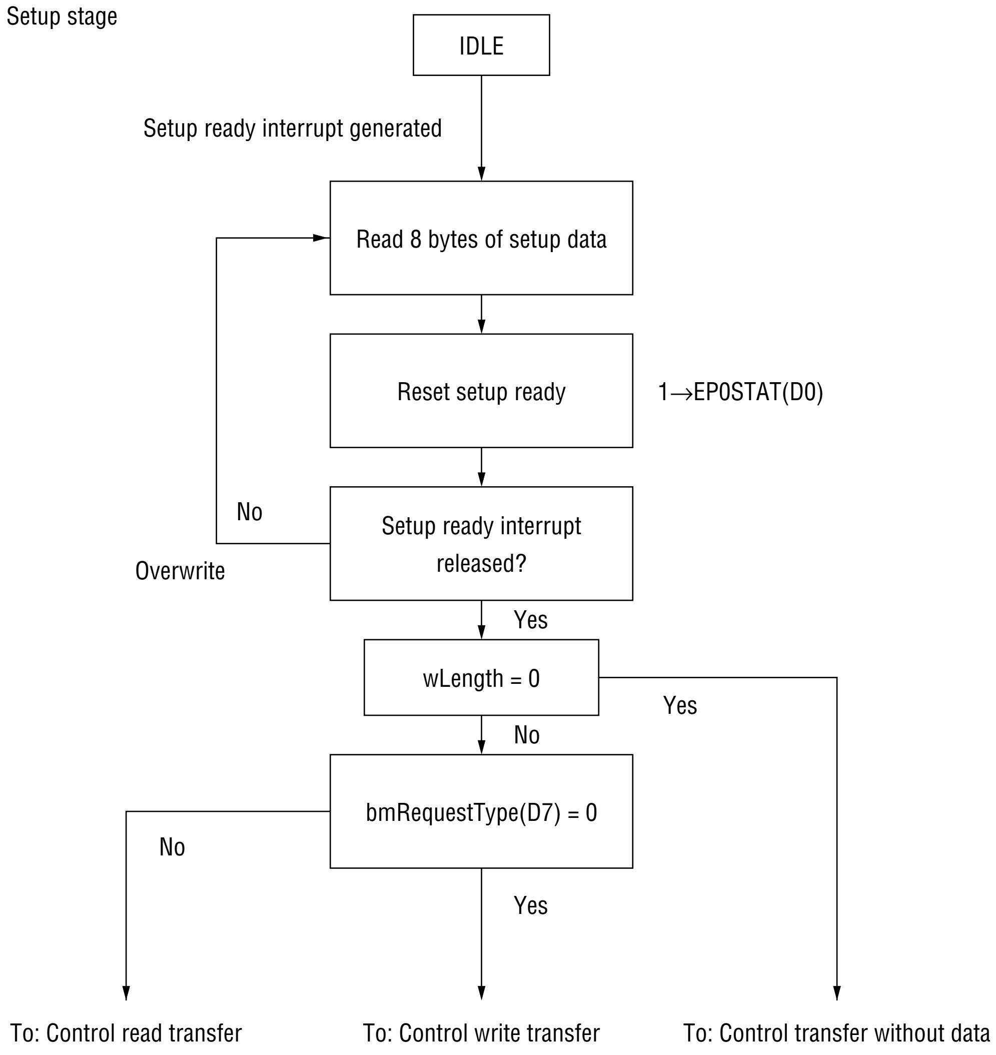
Status result report	Control write transfer (transmitted during the data phase)	Control read transfer (transmitted during the handshake phase)
The function execution has been completed normally	Data packet of length 0	ACK Handshake
An error occurred during function execution	STALL Handshake	STALL Handshake
The function is in the busy state	NAK Handshake	NAK Handshake

Control Transfer (Common)

◆ Registers used (setup stage)

Register name		Bit	R/W	Address	Stage
bmRequestType		7:0	R	D0h	Setup
bRequest		7:0	R	D1h	Setup
wValue (LSB)		7:0	R	D2h	Setup
(MSB)		7:0		D3h	
wIndex (LSB)		7:0	R	D4h	Setup
(MSB)		7:0		D5h	
wLength (LSB)		7:0	R	D6h	Setup
(MSB)		7:0		D7h	
Interrupt status	INSTAT	0	R	DCh	Setup
EP0 status	EP0STAT	0	Reset	73h	Setup
			R	F3h	

Control Transfer (Common)


Setup stage
IDLE
Setup ready interrupt generated
Read 8 bytes of setup data
Reset setup ready
1→EP0STAT(D0)
No
Setup ready interrupt released?
Overwrite
Yes
wLength = 0
Yes
No
bmRequestType(D7) = 0
No
Yes
To: Control read transfer
To: Control write transfer
To: Control transfer without data

Processing for Control Transfer (Common)

➪ Setup stage

◆ When 8 bytes of setup data are transferred from the host computer in the setup transaction, the ML60851A stores these bytes in the following registers.

- bRequest setup register (Read only)
- wValue LSB setup register (Read only)
- wValue MSB setup register (Read only)
- bmRequestType setup register (Read only)
- wIndex LSB setup register (Read only)
- wIndex MSB setup register (Read only)
- wLength LSB setup register (Read only)
- wLength MSB setup register (Read only)

◆ When the 8 bytes of setup data are transferred to and stored in the ML60851A normally, the ML60851A asserts the setup ready bit of the EP0 status register (EP0STAT) and the $\overline{\text{INTR}}$ pin to issue an interrupt request to the local MCU. This interrupt is intended to ask the local MCU to read the setup data.

◆ The local MCU responds to this interrupt request and reads out the setup data in the above registers. After reading the data, the setup ready bit of the EP0 status register (EP0STAT) should be reset (by writing a "1").

◆ Subsequently, the ML60851A normally accepts the resetting of the setup ready bit of EP0STAT and releases the setup ready interrupt.

However, if new setup data is received before reading of the 8 registers has been completed, the ML60851A does not accept the resetting to "1" of the setup ready bit of EP0STAT by the local MCU and maintains the interrupt state. (The setup ready bit also will not be reset.) In this case, the local MCU will have to read the setup data again.

◆ After the setup data has been read out and the setup ready interrupt has been released, the processing flow branches depending on the values of wLength and bit D7 of bmRequestType as follows.

◆ The processing flow branches to the status stage of control transfer without data when wLength is 0.

When wLength > 0 and bmRequestType (D7) = 0, the processing flow branches to the data stage of control write transfer.

When wLength > 0 and bmRequestType (D7) = 1, the processing flow branches to the data stage of control read transfer.

Outline of Control Write Transfer

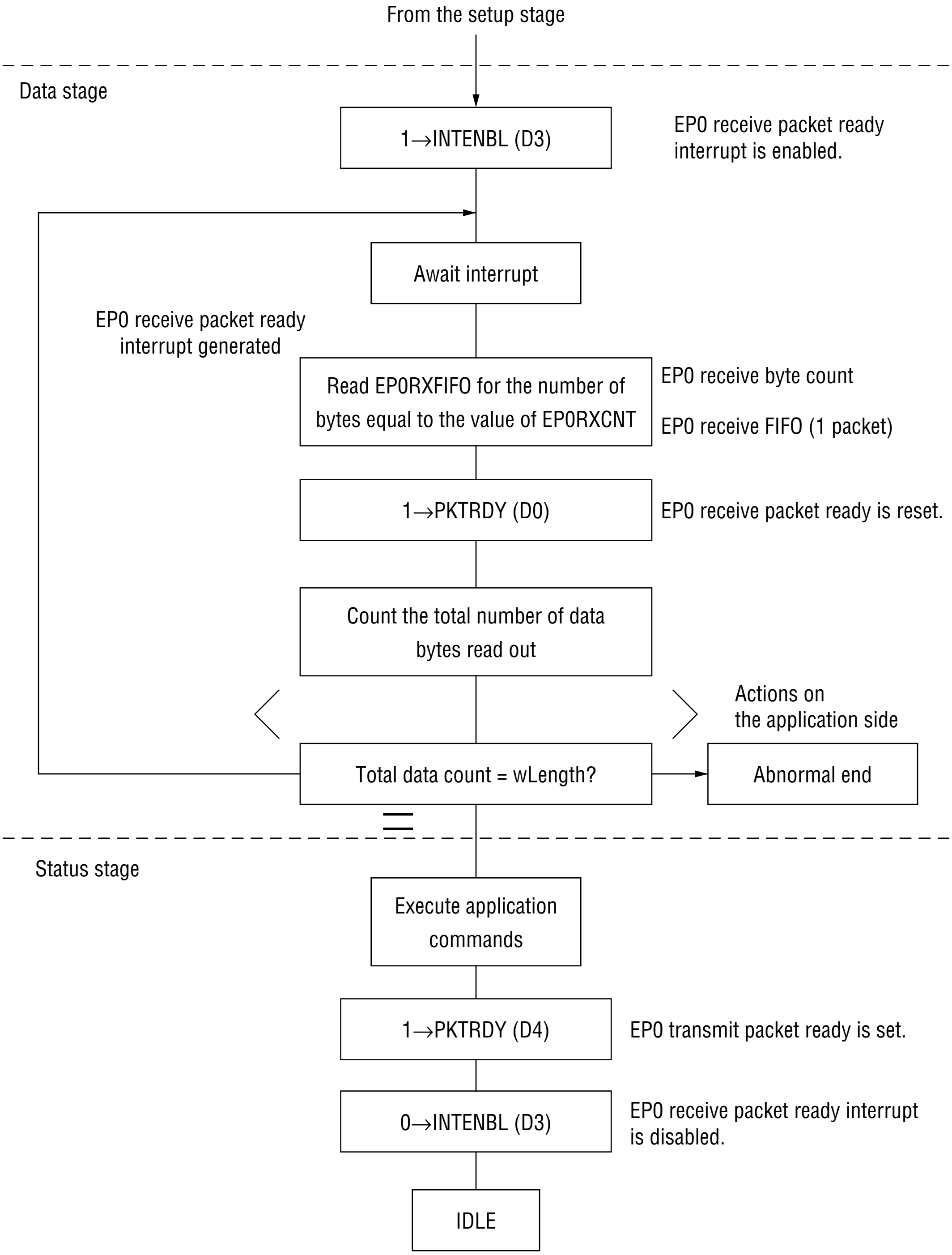
◆ Registers used (data stage, status stage)

Register name	Bit	R/W	Address	Stage
Interrupt enable (INTENBL)	D3	W	5Bh	Data
Interrupt status (INTSTAT)	D3	R	DCh	Data
Number of EP0 receive bytes (EP0RXCNT)	D6:0	R	C9h	Data
EP0 receive FIFO (EP0RXFIFO)	D7:0	R	40h	Data
EP Packet ready (PKTRDY)	D0	Reset	48h	Data
wLength (LSB) wLength (MSB)	D7:0 D7:0	R	D6h D7h	Data (Reading is completed in the setup stage)
EP Packet ready (PKTRDY)	D4	Set	48h	Status
Interrupt enable (INTENBL)	D3	W	5Bh	Status

◆ Related standard device request

- Set Descriptor

Flow of Control Write Transfer


From the setup stage
Data stage
1→INTENBL (D3)
EP0 receive packet ready interrupt is enabled.
Await interrupt
EP0 receive packet ready interrupt generated
Read EP0RXFIFO for the number of bytes equal to the value of EP0RXCNT
EP0 receive byte count
EP0 receive FIFO (1 packet)
1→PKTRDY (D0)
EP0 receive packet ready is reset.
Count the total number of data bytes read out
Actions on the application side
Total data count = wLength?
Abnormal end
Status stage
Execute application commands
1→PKTRDY (D4)
EP0 transmit packet ready is set.
0→INTENBL (D3)
EP0 receive packet ready interrupt is disabled.
IDLE

Processing for Control Write Transfer

➪ Data Stage

◆ In the data stage, first the local MCU asserts the EP0 receive packet ready interrupt enable bit and waits for generation of the EP0 receive packet ready interrupt.

◆ When a normal receive packet is stored in FIFO0, the ML60851A issues a request using the EP0 receive packet ready interrupt. Subsequently, the local MCU reads the receive FIFO data for the number of bytes received and stored (EP0RXCNT), and resets the receive packet ready bit (PKTRDY (0)). The contents of EP0RXCNT is normally the value of the maximum packet size, but only in the last packet it can be packet of any size that is less than the maximum packet size (a short packet).

(The maximum packet size in the ML60851A is 8 bytes.)

◆ When the ML60851A receives resetting of PKTRDY (0), it returns an ACK packet to the host computer and prompts for the next OUT transaction.

◆ In this manner, OUT transactions are executed successively. The data stage ends when the total number of bytes received during the data stage becomes equal to the value stored in the wLength register.

◆ During this process, an abnormal end is made if the total number of bytes received is larger than the value stored in the wLength register, and error processing will have to be made on the application side.

➪ Status Stage

◆ When the sequence of OUT transaction in the data stage are completed, the local MCU executes commands corresponding to the type of request.

◆ After command execution has been completed, the local MCU asserts the transmit packet ready bit, and the fact that application command execution was completed normally is reported to the host computer by transmitting data of zero length in response to an IN token from the host computer.

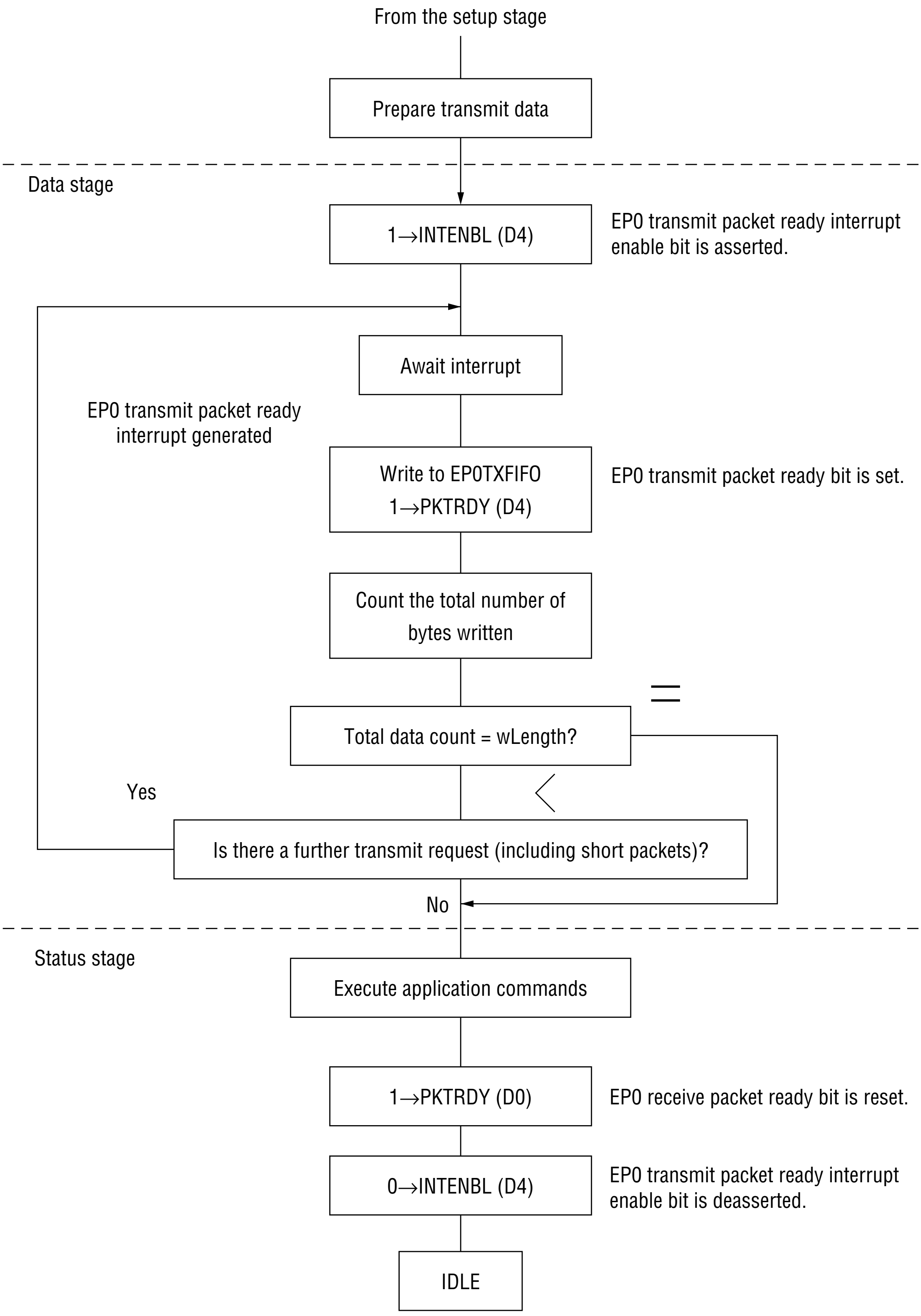
◆ Finally, the control write transfer is terminated by deasserting the EP0 receive packet ready interrupt enable bit.

Outline of Control Read Transfer

◆ Registers used (data stage, status stage)

Register name	Bit	R/W	Address	Stage
Interrupt enable (INTENBL)	D4	W	5Bh	Data
Interrupt status (INTSTAT)	D4	R	DCh	Data
EP0 transmit FIFO (EP0TXFIFO)	D7:0	W	C0h	Data
EP Packet ready (PKTRDY)	D4	Set	48h	Data
EP Packet ready (PKTRDY)	D0	Reset	48h	Status
Interrupt enable (INTENBL)	D4	W	5Bh	Status

Flow of Control Read Transfer


From the setup stage
Prepare transmit data
Data stage
1→INTENBL (D4)
EP0 transmit packet ready interrupt enable bit is asserted.
Await interrupt
EP0 transmit packet ready interrupt generated
Write to EP0TXFIFO
1→PKTRDY (D4)
EP0 transmit packet ready bit is set.
Count the total number of bytes written
=
Total data count = wLength?
<
Yes
Is there a further transmit request (including short packets)?
No
Status stage
Execute application commands
1→PKTRDY (D0)
EP0 receive packet ready bit is reset.
0→INTENBL (D4)
EP0 transmit packet ready interrupt enable bit is deasserted.
IDLE

Processing for Control Read Transfer

The ML60851A enters the following data stage when it is determined that the transfer is the control read transfer in the setup stage.

➪ Data Stage

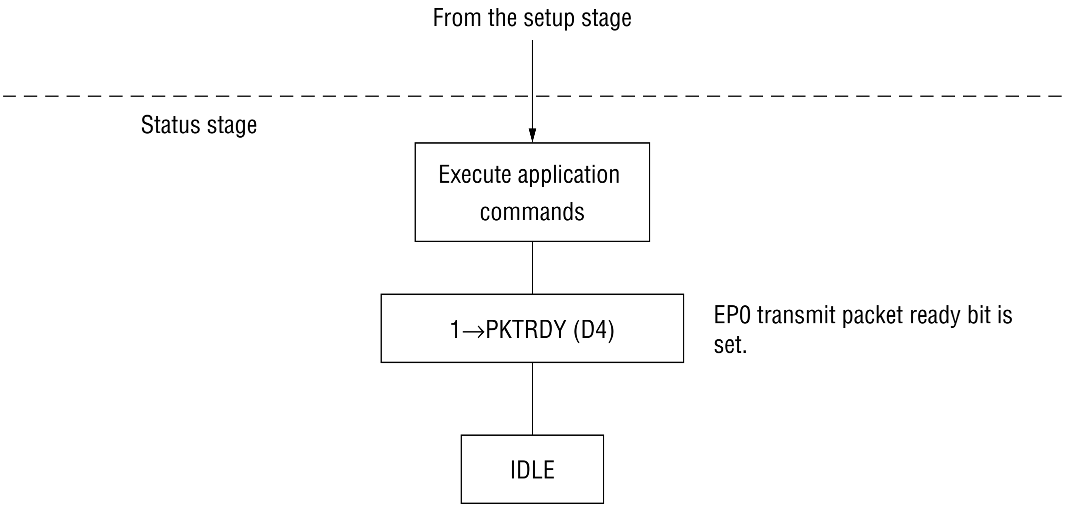
- ◆ In the data stage, first the local MCU asserts the EP0 transmit packet ready interrupt enable bit and waits for generation of the EP0 transmit packet ready interrupt.
- ◆ If the EP0 transmit FIFO is empty, the ML60851A issues an interrupt request using the EP0 transmit packet ready bit. Subsequently, the local MCU writes one packet of data to be transmitted, to the EP0 transmit FIFO and asserts the EP0 transmit packet ready bit PKTRDY (D4).
- ◆ When the EP0 packet ready bit is asserted, the ML60851A transmits the data in the transmit FIFO to the host computer upon receiving an IN transaction token from the host computer. During this period, the local MCU cannot access the EP0 transmit FIFO. When the transmission to the host is completed, the ML60851A receives the ACK handshake packet from the host computer and deasserts the EP0 transmit packet ready bit, and generates an EP0 transmit packet ready interrupt thereby starting the operation of transferring the next packet of data.
- ◆ In this manner, IN transactions are executed successively. The local MCU counts the total number of bytes written in the EP0 transmit FIFO. The data stage ends when the total number of bytes transmitted during the data stage becomes equal to the value stored in the wLength register.
- ◆ When attempting to transmit data to the host, if the total number of control bytes is smaller than the wLength value specified by the host, the last data to be transmitted is transmitted as a short packet thereby ending the data stage. When the host receives a short packet, it recognizes it as the end of control IN data.

➪ Status Stage

- ◆ When data transfer of IN transactions is completed, the local MCU decodes the 8 bytes of setup data received during the setup stage and executes appropriate commands.
- ◆ After the execution of such commands is completed, the local MCU resets the receive packet ready bit. The ML60851A asserts the EP0 receive packet ready bit upon receiving an OUT token and a zero length data from the host computer, transmits back an ACK handshake packet thereby reporting to the host computer that application command execution was completed normally.

  The ML60851A sends out an NAK handshake packet in response to the OUT token received before the receive packet ready bit is reset after command completion, thereby reporting the busy state to the host computer.
- ◆ Finally, the control read transfer is terminated by deasserting the EP0 transmit packet ready interrupt enable bit.

Control Transfer Without Data


From the setup stage
Status stage
Execute application commands
1→PKTRDY (D4)
EP0 transmit packet ready bit is set.
IDLE

Processing for Control Transfer Without Data

The ML60851A enters the following status stage when it is determined that the transfer is a control transfer without data in the setup stage.

➪ Status Stage

- ◆ The local MCU decodes the 8 bytes of setup data received during the setup stage and executes appropriate commands.
- ◆ After the execution of such commands is completed, the local MCU sets the transmit packet ready bit and transmits a zero length data packet to the host in response to an IN token from the host computer. This reports to the host computer that the application command execution has been completed normally.

6.2 Bulk-in transfer

Outline of Bulk-In Transfer

This is the mode in which bulk data is transmitted from the peripheral unit to the host computer.

➪ End point address

◆ The end point of bulk-in transfer is assigned to either EP1 or EP2. EP1 has a FIFO configuration of 2 layers of 64 bytes each, and that of EP2 is one layer of 64 bytes.

◆ EP1 and EP2 are mutually independent and also it is possible to dynamically change the direction of data transfer.

➪ Registers and bits to be referenced

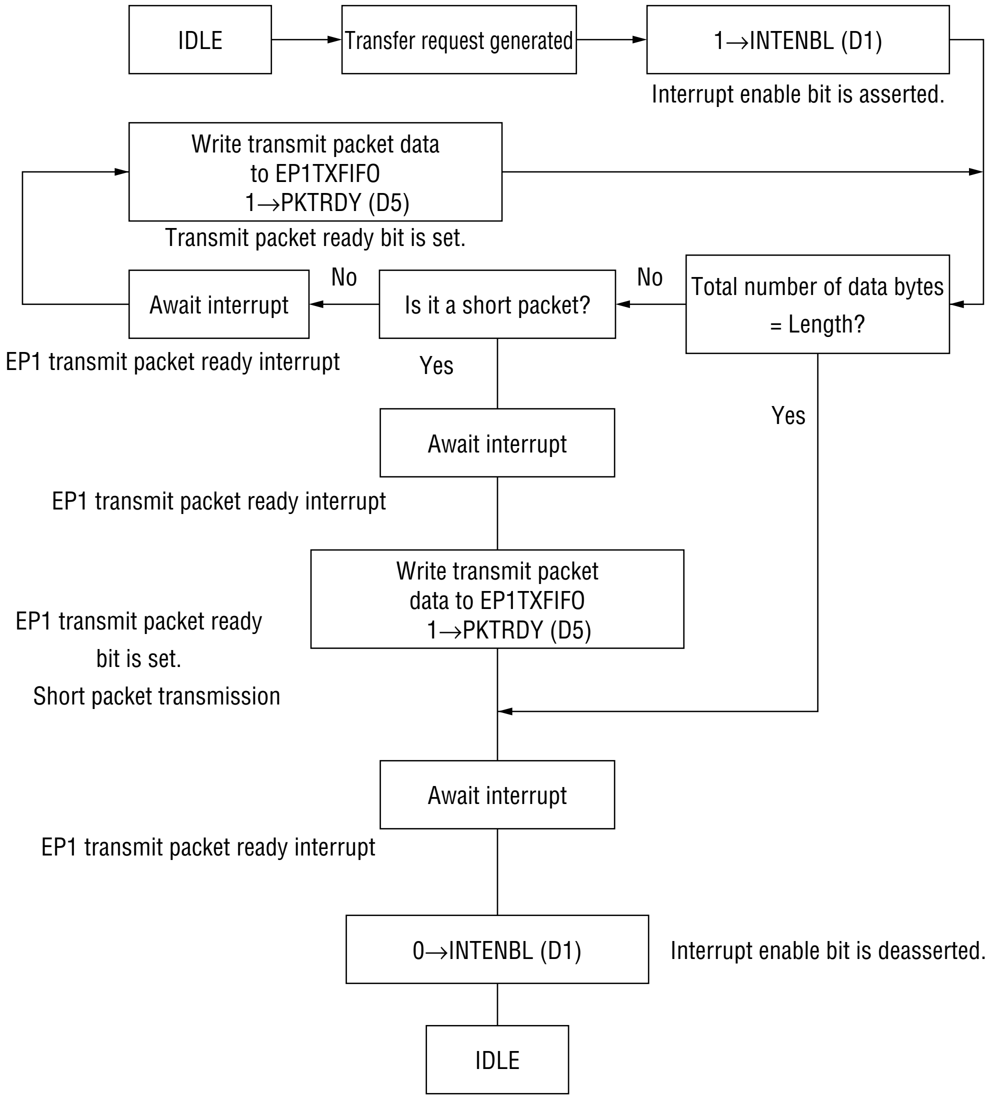
◆ When assigned to EP1

- Packet ready : Bit D5 of PKTRDY
- Interrupt enable : Bit D1 of INTENBL
- Control register : EP1CON
- Data toggle register : EP1TGL
- Payload register : EP1PLD

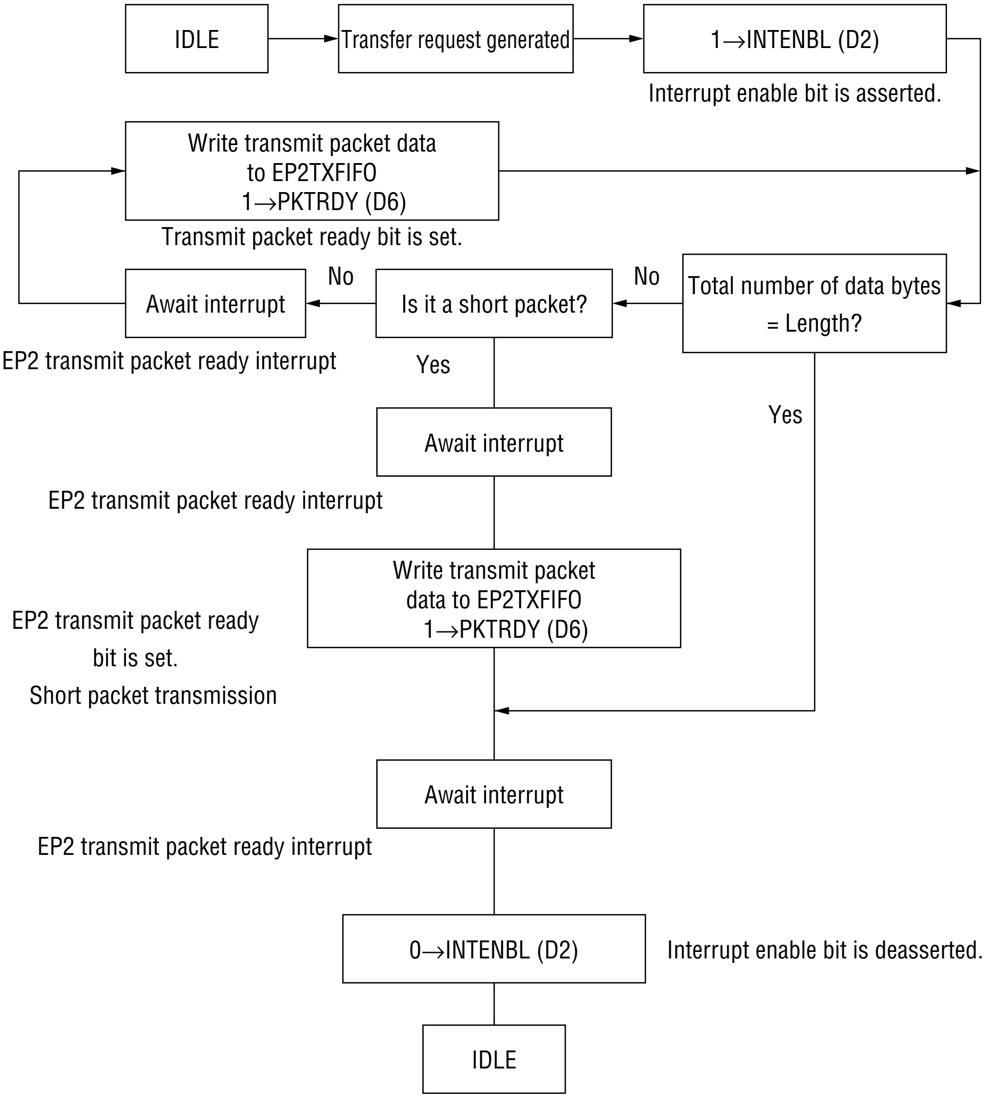
◆ When assigned to EP2

- Packet ready : Bit D6 of PKTRDY
- Interrupt enable : Bit D2 of INTENBL
- Control register : EP2CON
- Data toggle register : EP2TGL
- Payload register : EP2PLD

Bulk-In Transfer (When EP1 is used)


IDLE
Transfer request generated
1→INTENBL (D1)
Interrupt enable bit is asserted.
Write transmit packet data
to EP1TXFIFO
1→PKTRDY (D5)
Transmit packet ready bit is set.
Await interrupt
No
Is it a short packet?
No
Total number of data bytes
= Length?
EP1 transmit packet ready interrupt
Yes
Yes
Await interrupt
EP1 transmit packet ready interrupt
Write transmit packet
data to EP1TXFIFO
1→PKTRDY (D5)
EP1 transmit packet ready
bit is set.
Short packet transmission
Await interrupt
EP1 transmit packet ready interrupt
0→INTENBL (D1)
Interrupt enable bit is deasserted.
IDLE

Bulk-In Transfer (When EP2 is used)


IDLE
Transfer request generated
1→INTENBL (D2)
Interrupt enable bit is asserted.
Write transmit packet data to EP2TXFIFO 1→PKTRDY (D6)
Transmit packet ready bit is set.
Await interrupt
No
Is it a short packet?
No
Total number of data bytes = Length?
EP2 transmit packet ready interrupt
Yes
Yes
Await interrupt
EP2 transmit packet ready interrupt
Write transmit packet data to EP2TXFIFO 1→PKTRDY (D6)
EP2 transmit packet ready bit is set.
Short packet transmission
Await interrupt
EP2 transmit packet ready interrupt
0→INTENBL (D2)
Interrupt enable bit is deasserted.
IDLE

Bulk-In Transfer

- Interrupt Enable

  When a data transfer is requested by the peripheral unit, the local MCU asserts the EP1 (or EP2) packet ready interrupt enable bit and waits for an interrupt from the ML60851A.

- Interrupt Generation

  When the ML60851A becomes ready to receive transmit data from the local MCU, it deasserts the transmit packet ready bit, bit D5 (or bit D6) of the PKTRDY register to generate the EP1 (or EP2) packet ready interrupt . (The $\overline{\text{INTR}}$ pin is asserted.)

- Writing packet data to the transmit FIFO

  Upon receiving the packet ready interrupt, the local MCU writes one packet of transmit data to the transmit FIFO of the ML60851A, and sets the EP1 (or EP2) transmit packet ready bit.

- Transmission to the host

  When the transmit packet ready bit is set, the ML60851A transmits the data accumulated in the transmit FIFO to the host computer upon receiving an IN token from the host computer.

- Writing next packet of data

  The subsequent operation differs depending on whether EP1 or EP2 is being used as explained below.

  - When EP2 is used:

    During data transmission, the ML60851A does not request the packet ready interrupt. After the transmission of that packet has been completed and an ACK packet is received from the host computer, the ML60851A generates the next transmit packet ready interrupt to the local MCU. Following this, the local MCU writes the next transmit data to the transmit FIFO.
  - When EP1 is used:
    Since the FIFO of EP1 has a two layer configuration, it is possible to write transmit data of the next packet to the FIFO even during data transmission.

- Management of number of transmit bytes

  While data is transmitted successively in this manner, the firmware of the local MCU counts the total number of bytes written in the transmit FIFO. The method of terminating data transmission can be one of the following two:

  - If the total number of bytes to be transmitted (Length) is equal to an integral multiple of the maximum packet size, data transmission is terminated after transmitting data for the integer number of times.
  - If the total number of bytes to be transmitted (Length) is not equal to an integral multiple of the maximum packet size, the data transmission is terminated by finally sending a short packet.

- Interrupt Disable

  - After an ACK packet arrives from the host computer following the transmission of the last packet, the local MCU should deassert the interrupt enable bit if the ML60851A issues another request by generating a packet ready interrupt. In this case, the ML60851A goes into the idle state.

6.3 Bulk-out transfer

Outline of Bulk-Out Transfer

➪ End point address

◆ The end point of bulk-out transfer is assigned to either EP1 or EP2. EP1 has a FIFO configuration of 2 layers of 64 bytes each, and that of EP2 is one layer of 64 bytes.

◆ EP1 and EP2 are mutually independent and also it is possible to dynamically change the direction of data transfer.

➪ Registers and bits to be referenced

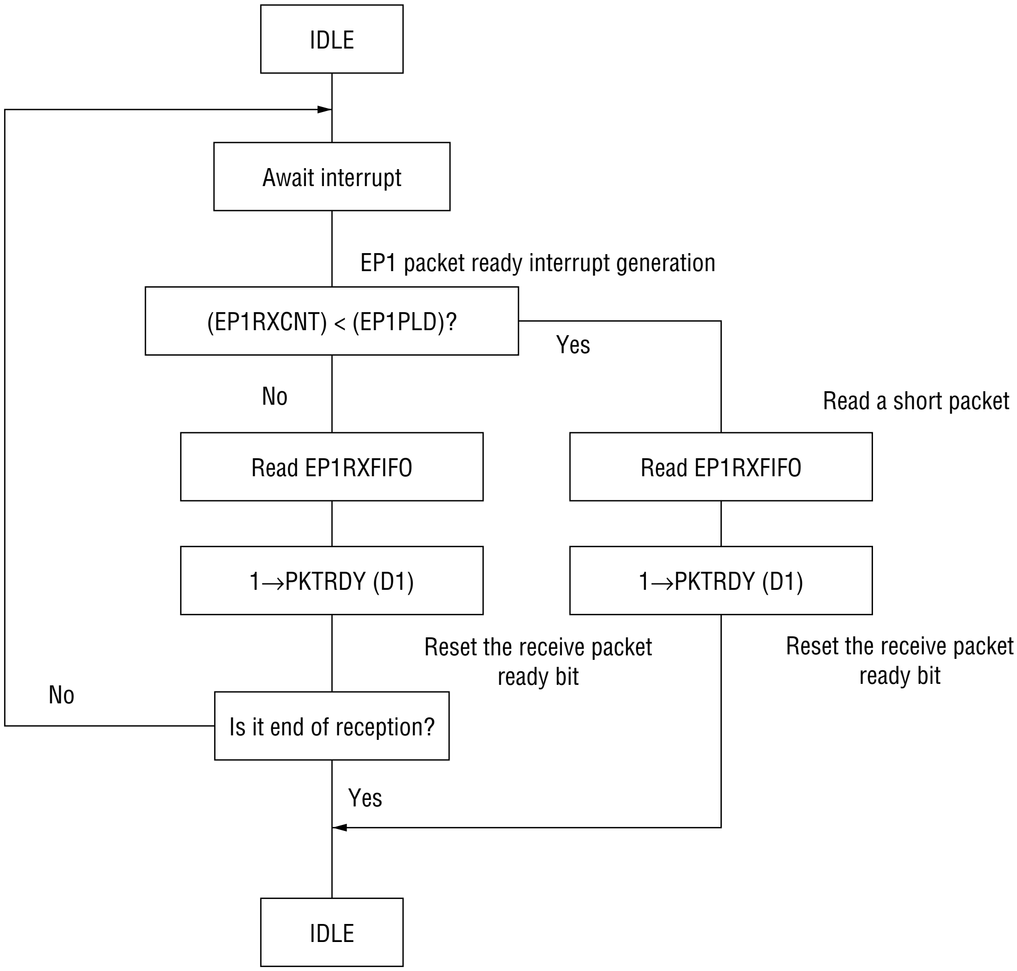
◆ When assigned to EP1

• Packet ready	: Bit D1 of PKTRDY
• Interrupt enable	: Bit D1 of INTENBL
• Interrupt status	: Bit D1 of INTSTAT
• Receive byte count	: EP1RXCNT
• Control register	: EP1CON
• Payload register	: EP1PLD

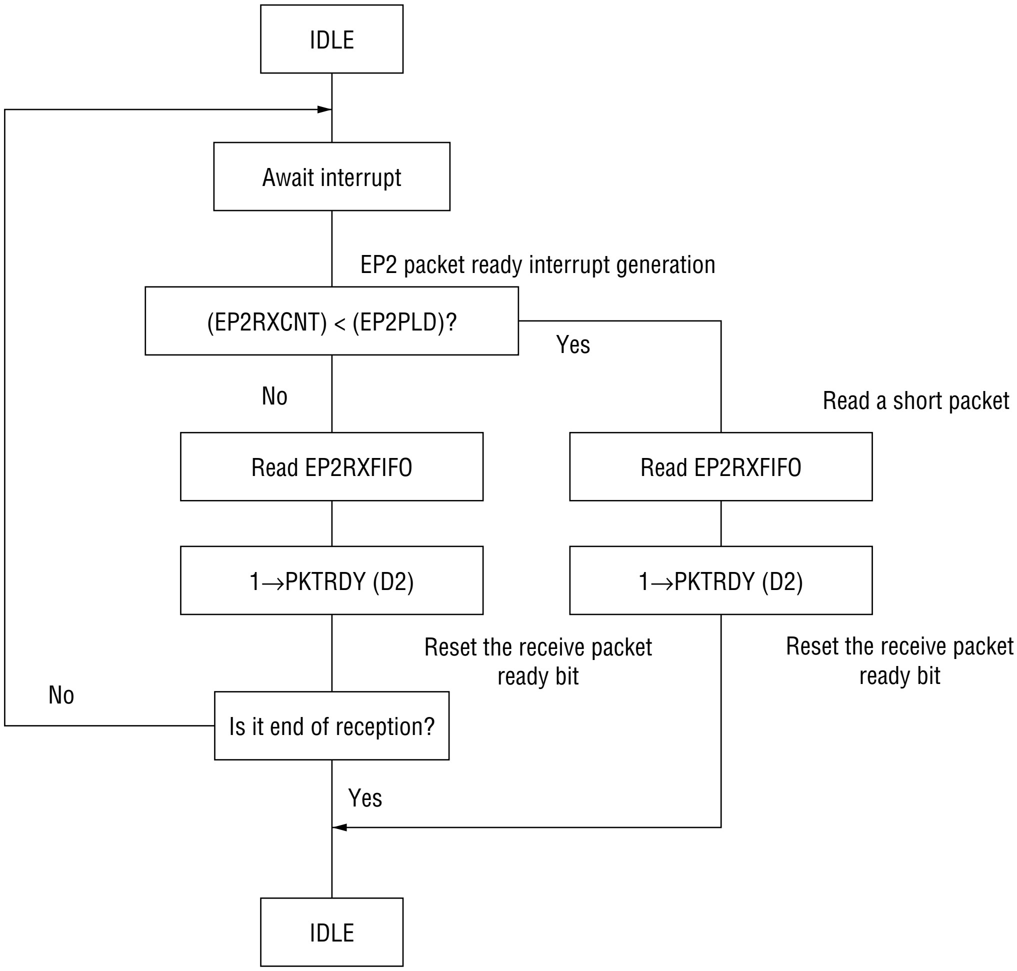
◆ When assigned to EP2

• Packet ready	: Bit D2 of PKTRDY
• Interrupt enable	: Bit D2 of INTENBL
• Interrupt status	: Bit D2 of INTSTAT
• Receive byte count	: EP2RXCNT
• Control register	: EP2CON
• Payload register	: EP2PLD

Bulk-Out Transfer (When EP1 is used)



Bulk-Out Transfer (When EP2 is used)



Bulk-Out Transfer

◆ Interrupt Enable

In the case of bulk reception, the local MCU asserts the receive packet ready interrupt enable bit at an arbitrary instant of time and prepares for data transfer from the host computer.

◆ Interrupt Generation

When the ML60851A receives one packet of receive data from the host computer and stores it in the receive FIFO, it asserts bit D1 (or D2) of the PKTRDY register and generates the EP1 (or EP2) packet ready interrupt . (The $\overline{\text{INTR}}$ pin is asserted.)

◆ Reading packet data from the receive FIFO

Upon receiving the packet ready interrupt, the local MCU reads one packet of receive data from the receive FIFO of the ML60851A, and resets the EP1 (or EP2) receive packet ready bit. When the EP1 (or EP2) receive packet ready bit is reset, the ML60851A sends an ACK handshake packet to the host computer thereby prompting for the transfer of next data packet.

◆ Reading next packet of data

The subsequent operation differs depending on whether EP1 or EP2 is being used as explained below.

- When EP2 is used:

  During data reception, the ML60851A does not issue an interrupt request using the packet ready bit. After the reception of one packet has been completed, the ML60851A generates the next receive packet ready interrupt to the local MCU. Following this, the local MCU can access the receive FIFO.

- When EP1 is used:

  Since the FIFO of EP1 has a two layer configuration, it is possible to read the previous packet of receive data in the FIFO even during data reception.

◆ Management of number of receive bytes

The data is received successively in the above manner. The method of terminating data reception can be one of the following two:

- If the total number of bytes received is not equal to an integral multiple of the maximum packet size entered in the payload register (EP1PLD or EP2PLD), the last data packet received will be a short packet. The data reception is terminated after receiving a short packet.
- If the total number of bytes received is equal to an integral multiple of the maximum packet size entered in the payload register (EP1PLD or EP2PLD), it is necessary to determine the end of IRP by some other means.

6.4 Interrupt transfer

Outline of Interrupt Transfer

This is the mode in which interrupt data is transmitted from the peripheral unit to the host computer.

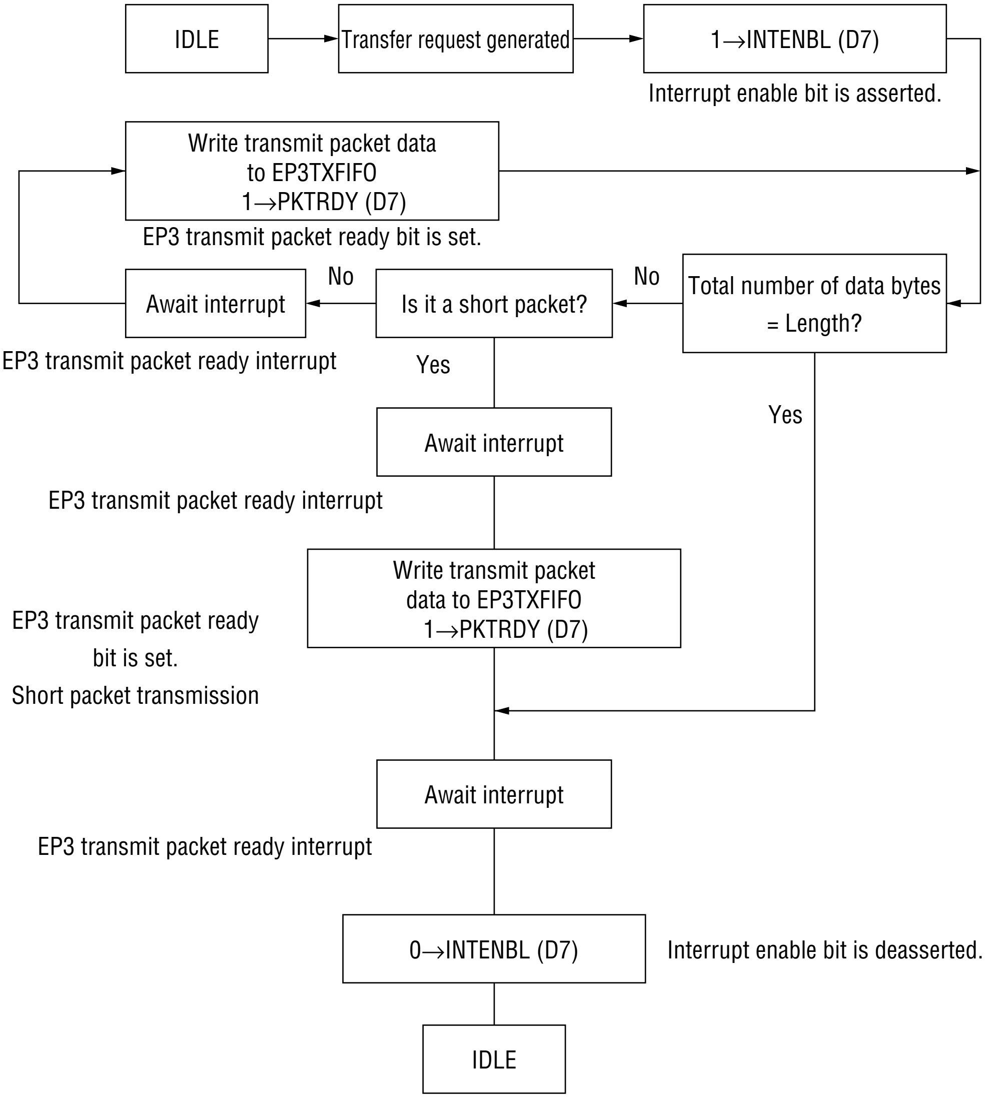
➪ End point address

◆ The end point of interrupt transfer is assigned to EP3. EP3 has a FIFO of 8 bytes.

➪ Registers and bits to be referenced

• FIFO	: EP3TXFIFO
• Packet ready	: Bit D7 of PKTRDY
• Interrupt enable	: Bit D7 of INTENBL
• Interrupt status	: Bit D7 of INTSTAT
• EP3 control register	: EP3CON
• EP3 data toggle register	: EP3TGL
• EP3 payload register	: EP3PLD

Interrupt Transfer


IDLE
Transfer request generated
1→INTENBL (D7)
Interrupt enable bit is asserted.
Write transmit packet data
to EP3TXFIFO
1→PKTRDY (D7)
EP3 transmit packet ready bit is set.
Await interrupt
No
Is it a short packet?
No
Total number of data bytes
= Length?
EP3 transmit packet ready interrupt
Yes
Yes
Await interrupt
EP3 transmit packet ready interrupt
Write transmit packet
data to EP3TXFIFO
1→PKTRDY (D7)
EP3 transmit packet ready
bit is set.
Short packet transmission
Await interrupt
EP3 transmit packet ready interrupt
0→INTENBL (D7)
Interrupt enable bit is deasserted.
IDLE

Interrupt Transfer

◆ Interrupt Enable

When a data transfer using interrupt transfer is requested by the peripheral unit, the local MCU asserts the EP3 transmit packet ready interrupt enable bit and waits for an interrupt from the ML60851A.

◆ Interrupt Generation

When the ML60851A receives an IN token specifying EP3 of the device from the host computer and becomes ready to receive transmit data from the local MCU, it deasserts the EP3 transmit packet ready bit (bit D7 of the PKTRDY register) and generates the EP3 transmit packet ready interrupt . (The $\overline{\text{INTR}}$ pin is asserted.)

◆ Writing packet data to the transmit FIFO

Upon receiving the packet ready interrupt, the local MCU writes one packet of transmit data to the transmit FIFO (EP3TXFIFO) of the ML60851A, and sets the EP3 transmit packet ready bit (bit D7 of the PKTRDY register), and deasserts the $\overline{\text{INTR}}$ pin.

◆ Transmission to the host

When the EP3 transmit packet ready bit is set, the ML60851A transmits the data accumulated in the EP3 transmit FIFO (EP3TXFIFO) to the host computer upon receiving an IN token from the host computer.

◆ Writing the next packet of data

The host computer returns an ACK handshake packet when it receives the packet normally. Subsequently, the ML60851A again deasserts the EP3 transmit packet ready bit (bit D7 or PKTRDY) and generates an EP3 transmit packet ready interrupt. (The $\overline{\text{INTR}}$ pin is also asserted.)

At this point, the local MCU writes the next packet of transmit data to EP3TXFIFO.

◆ Management of number of transmit bytes

While data is transmitted successively in this manner, the firmware of the local MCU counts the total number of bytes written in the transmit FIFO. In addition, the local MCU also maintains the maximum packet size and the total number of bytes to be transmitted. The transmission is terminated after transmitting a number of bytes equal to the total number of bytes to be transmitted. If the total number of bytes to the transmitted is equal to an integral multiple of the maximum packet size, all the data packets transmitted have sizes equal to the maximum packet size. Otherwise, the last packet transmitted will have a size less than the maximum packet size (that is, a short packet). Only the last packet transmitted can be a short packet.

◆ Interrupt Disable

After an ACK packet arrives from the host computer following the transmission of the last packet, the local MCU should deassert the interrupt enable bit if the ML60851A issues another request by generating a packet ready interrupt. In this case, the ML60851A goes into the idle state.