FEAL60852A-02
DATE: OCT. 9, 2001

ML60852A
Application Manual

USB device controller

Version 1.02

ML60852A Application Manual

CONTENTS

1. INTRODUCTION

The ML60852A is a general-purpose device controller conforming to the Universal Serial Bus (USB) Standard Rev.1.1. This LSI contains a USB serial interface engine, a USB transceiver, FIFOs, control and status registers, application interface circuits, and an oscillator circuit, and allows easy realization of a USB system.

This LSI supports control transfer, bulk transfer, interrupt transfer, and isochronous transfer as the data transfer modes and allows five or six end points to be used.

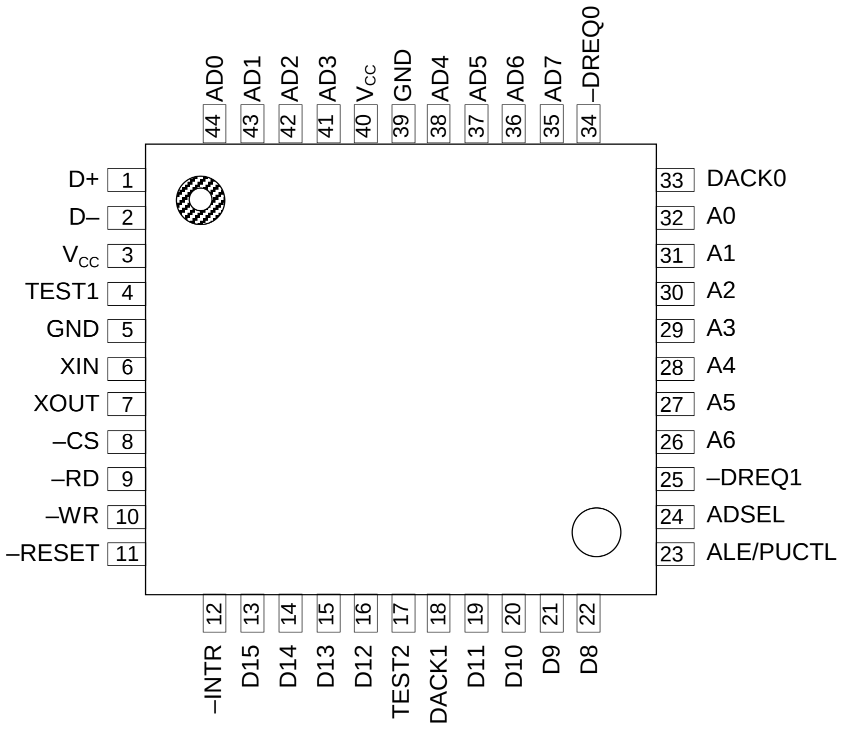
1.1. Product Features

- Conforms to USB1.1.
- Supports Full-speed (12 Mbps).
- Supports four data transfer types:
  Control transfer, bulk transfer, interrupt transfer, and isochronous transfer.
- Five or six end points
- Built-in FIFO for data storage
- The FIFO for EP1, EP2, EP4, and EP5 has a 2-layer configuration.
- 8 or 16 bit DMA Transfer is possible (EP1, EP2, EP4, EP5) with two channels.
- Supports bus-powered devices
  With the operation mode setting, the ML60852A detects the suspend condition automatically and enters the low-power mode. The LSI automatically returns to the normal operation when it detects the resume condition.
- Built-in USB transceiver circuit
- V_{CC} = 3.0 to 3.6 V
- Can be interfaced with 5 V circuits (5 V tolerant input, TTL output).
- Built-in 6 MHz/12 MHz selectable oscillator circuit
- Packages : 44-pin QFP/TQFP
  56-pin LGA

1.2. Pin Configuration and Description

1.2.1. Pin configuration

44-Pin QFP (Top View)



56-Pin LGA (Transparent View)

NC	D8	D9	D11	TEST2	D12	D14	–INTR	NC	J
ALE/ PUCTL	NC	D10	DACK1	NC	D13	D15	NC	–RESET	H
–DREQ1	ADSEL						–RD	–WR	G
A5	A6						–XOUT	–CS	F
A4	NC						NC	–XIN	E
A2	A3						TEST1	GND	D
A0	A1						D–	V_{CC}	C
DACK1	NC	AD7	AD5	NC	DACK1	AD2	NC	D+	B
NC	–DREQ0	AD6	AD4	GND	AD3	AD1	AD0	NC	A
9	8	7	6	5	4	3	2	1	

1.2.2. Pin description

USB Interface

Signal name	I/O	Polarity	Description
D+	I/O	—	Pin for connecting USB Data (+)
D–	I/O	—	Pin for connecting USB Data (–)

When D+ and D– are indeterminate, it is impossible to write data in the registers that are reset by a USB bus reset.

Crystal oscillator interface

Signal name	I/O	Polarity	Description
XIN	I	—	Pin for connecting a crystal
XOUT	O	—	Pin for connecting a crystal

Application interface

Signal name	I/O	Polarity	Description
D15 to D8	I/O	—	Upper byte of data bus (MSB)
AD7 to AD0	I/O	—	Lower byte of the data bus (LSB) and address input pin
A6 to A0	I	—	Address input pins
–CS	I	Negative logic	Chip select signal input pin
–RD	I	Negative logic	Read signal input pin
–WR	I	Negative logic	Write signal input pin
–INTR	O	(*1)	Interrupt request signal output pin
–DREQ0/–DREQ1	O	(*1)	DMA Request signal output pins
DACK0/DACK1	I	(*2)	DMA Acknowledge signal input pins
ALE/PUCTL	I	(*3)	Address latch enable signal input and pull-up resistor control pin
ADSEL	I		Address input format signal input pin
–RESET	I	Negative logic	Reset signal input pin

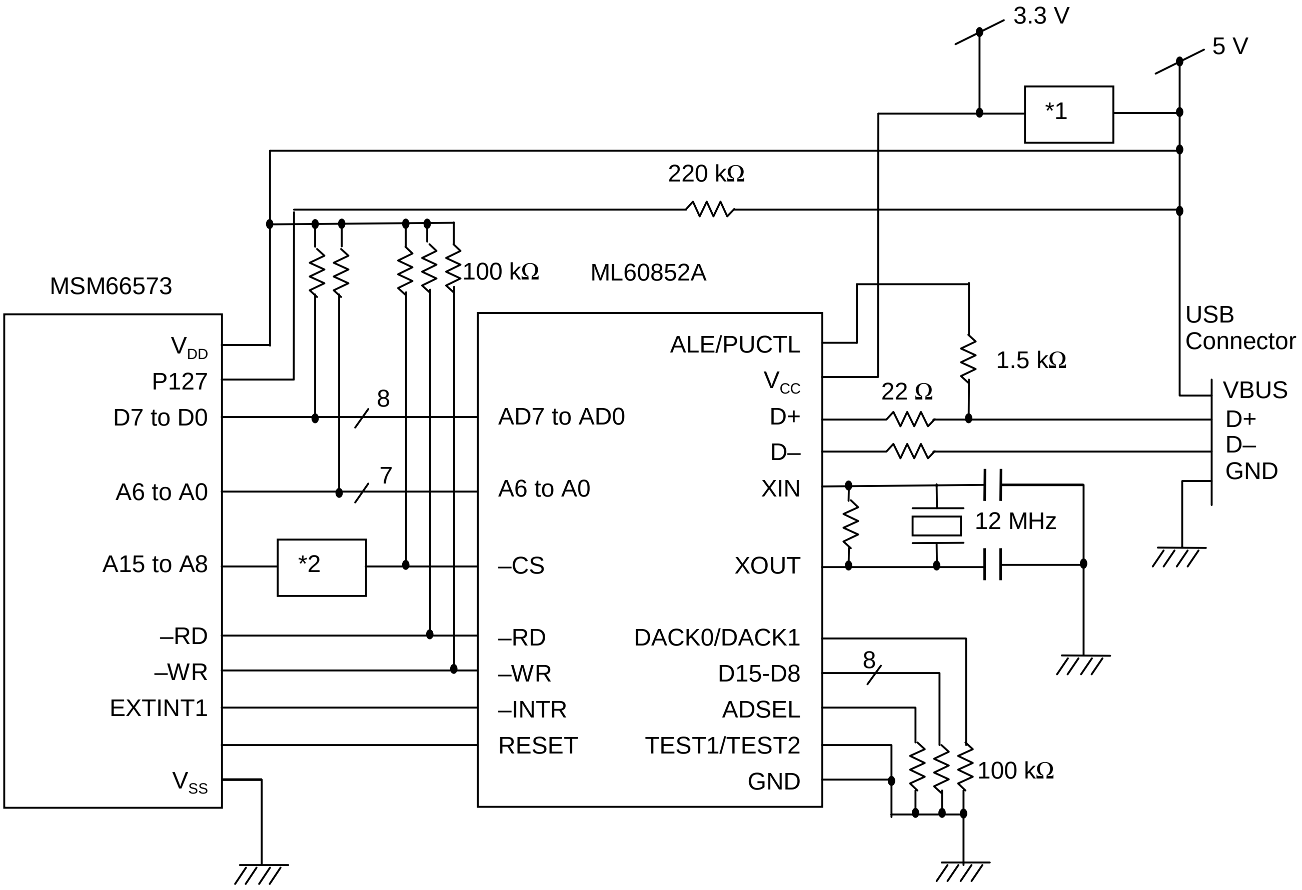
*1: Although the default value immediately after reset is negative logic, the polarity can be changed by overwriting the polarity selection register.

*2: Although the default value immediately after reset is positive logic, the polarity can be changed by overwriting the polarity selection register.

*3: The polarity becomes positive logic when this signal functions as an address latch enable (ALE) signal.

1.3. Example of External Connections

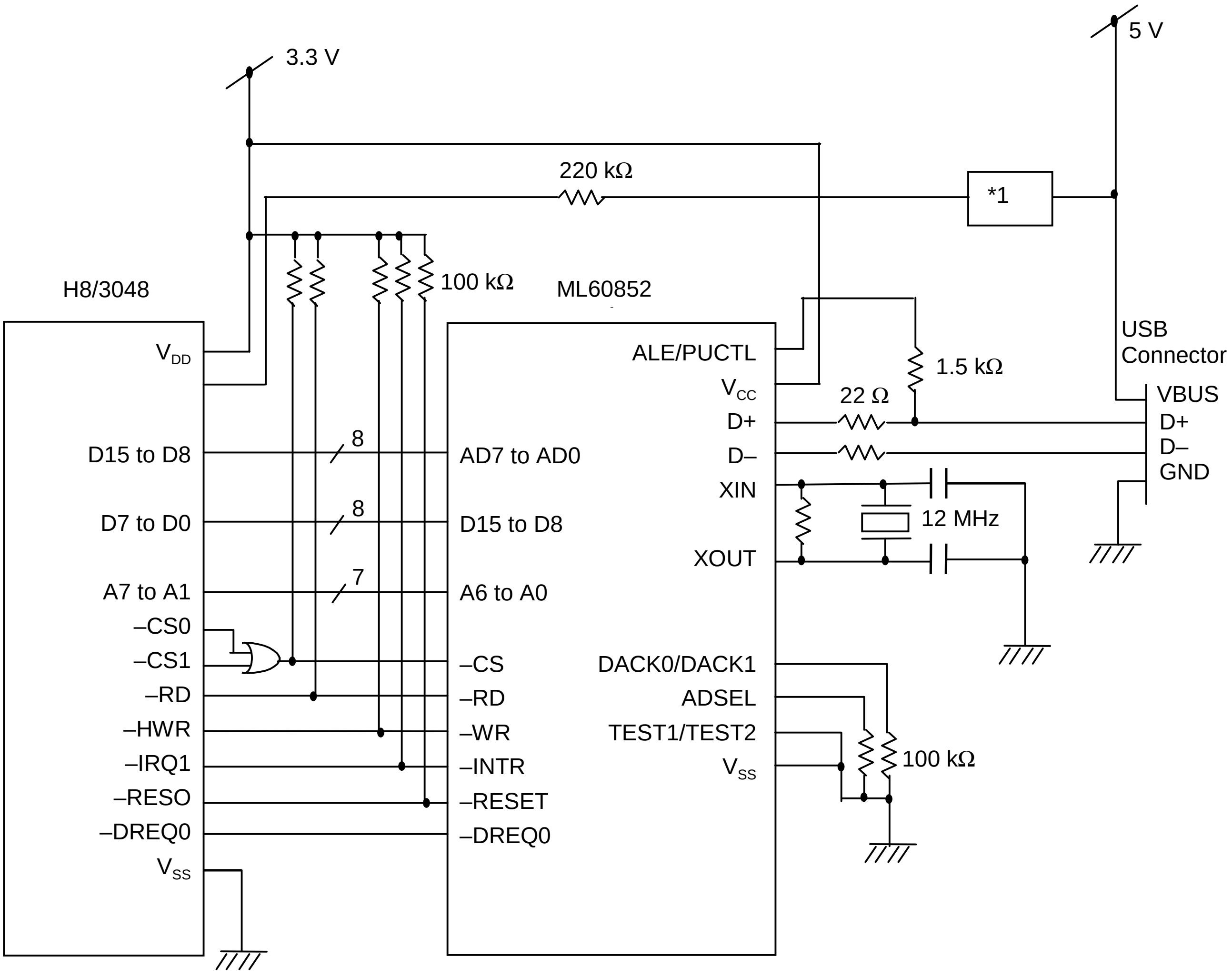
1.3.1. Example of connections between MSM66573 (OKI make) and ML60852A



Notes:
*1: 5 to 3.3 V
*2: Address decode
*3: Pull-up control is used in this connection.
*4: This circuit is bus-powered.

Figure 1 Example of connections between OKI's 16-bit microcontroller MSM66573 and ML60852A

1.3.2. Example of connections between H8/3048 (Hitachi make) and ML60852A



Notes:
*1: 5 to 3.3 V
*2: Pull-up control is used in this connection.
*3: This circuit is bus-powered.

Figure 2 Example of connections between Hitachi's 16-bit microcontroller (DMA transfer in dual address mode) H8/3048 and ML60852A

2. EXAMPLES OF USB TRANSFER PROCEDURE

This chapter gives a description of a sample procedure for carrying out USB communication using the ML60852A.

2.1. Device Initialization

The device initialization consists of setting the hardware operating conditions and settings based on the standard request from a host computer.

2.1.1. Setting the operating conditions of ML60852A

Carry out the settings of the operating conditions of the ML60852A to suit the system after referring to Chapter 3 “EXTERNAL INTERFACE”.

2.1.2. Settings based on standard request

Eleven types of standard requests have been defined in USB Standard Rev. 1.1 (See Chapter 9 of the USB Standard Rev. 1.1 for detailed definitions of the standard requests). Standard requests are issued from a host computer via a control pipe. Some of these requests require device settings to be changed. Hence, it is necessary for the USB firmware to support these requests and set device registers accordingly.

2.2. Control Transfer

2.2.1. Control transfer

A control transfer is started when a device request is issued by a host computer. EP0 is defined as the default end point by USB Specifications and it is used for control transfers. The end point used during a control transfer is EP0. The ML60852A has a built-in 32-byte transmit/receive FIFO for EP0.

Depending on the type of request, the control transfer can be a control read transfer, a control write transfer, or a control transfer without data.

- Control read transfer: Control data is transmitted to the host computer.
- Control write transfer: Control data is received from the host computer.
- Control transfer without data: Control transfer without a data stage.

The registers used during a control transfer are the following:

Table 1 Registers used in the setup stage

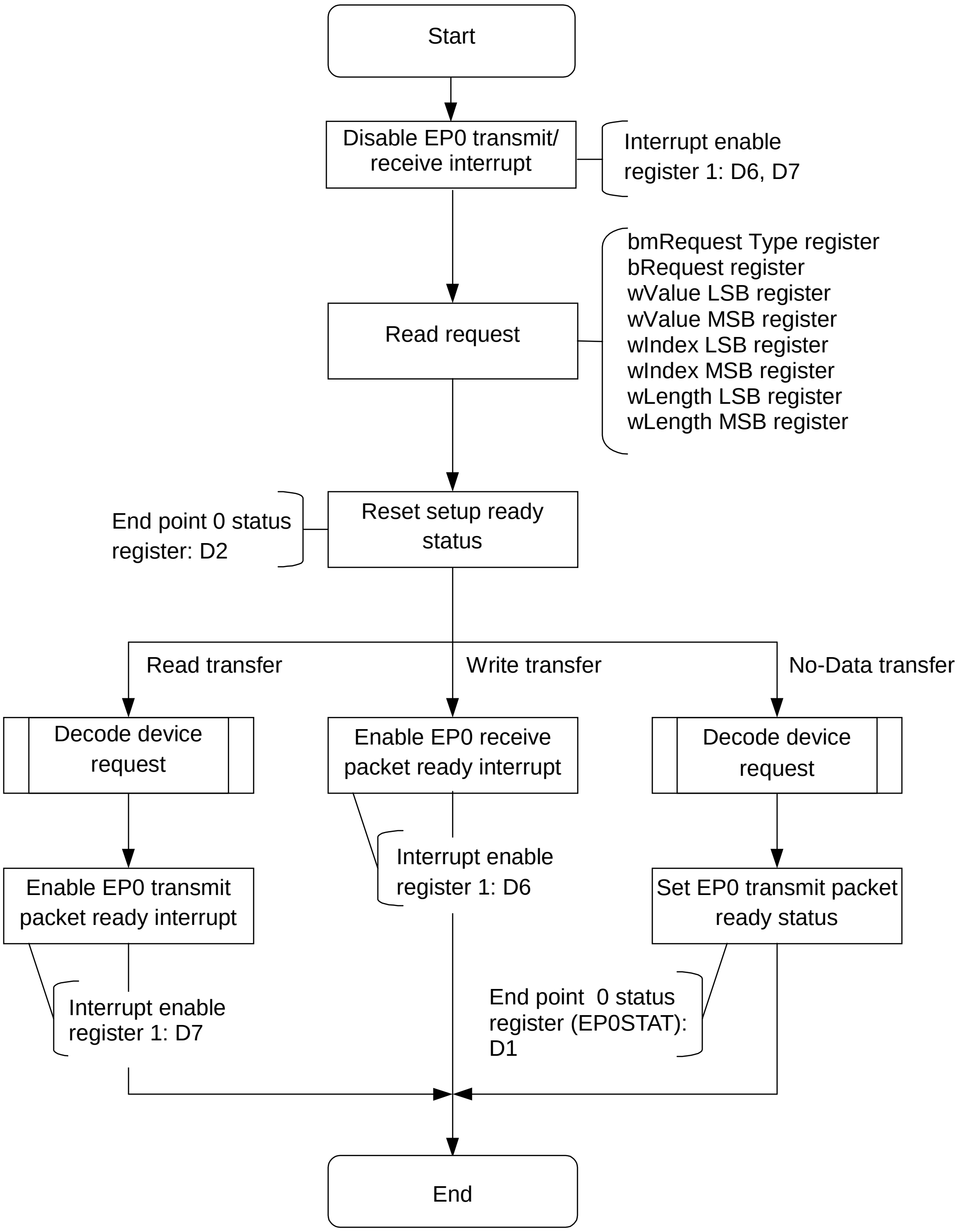
Function	Register name	Bits
Device request	bRequest Setup	D7 to D0
Device request	wValue LSB Setup	D7 to D0
Device request	wValue MSB Setup	D7 to D0
Device request	bmRequest Type Setup	D7 to D0
Device request	wIndex LSB Setup	D7 to D0
Device request	wIndex MSB Setup	D7 to D0
Device request	wLength LSB Setup	D7 to D0
Device request	wLength MSB Setup	D7 to D0
Setup ready	EP0STAT	D2
Setup ready interrupt status	INTSTAT1	D0
Setup ready interrupt enable	INTENBL1	D0

Table 2 Registers used in the data stage or status stage

Function	Register name	Bits
Packet ready	EP0STAT	D0/D1
Packet ready interrupt status	INTSTAT1	D6/D7
Packet ready interrupt enable	INTENBL1	D6/D7
End point control	EP0CONT	D0
Data sequence toggle	EP0CONT	D1/D4
Maximum packet size	EP0PLD	D7 to D0
Status	EP0STAT	D4/D5
Receive byte count	EP0RXCNT	D5 to D0
Transmit/receive FIFO	EP0RXFIFO/EP0TXFIFO	D7 to D0

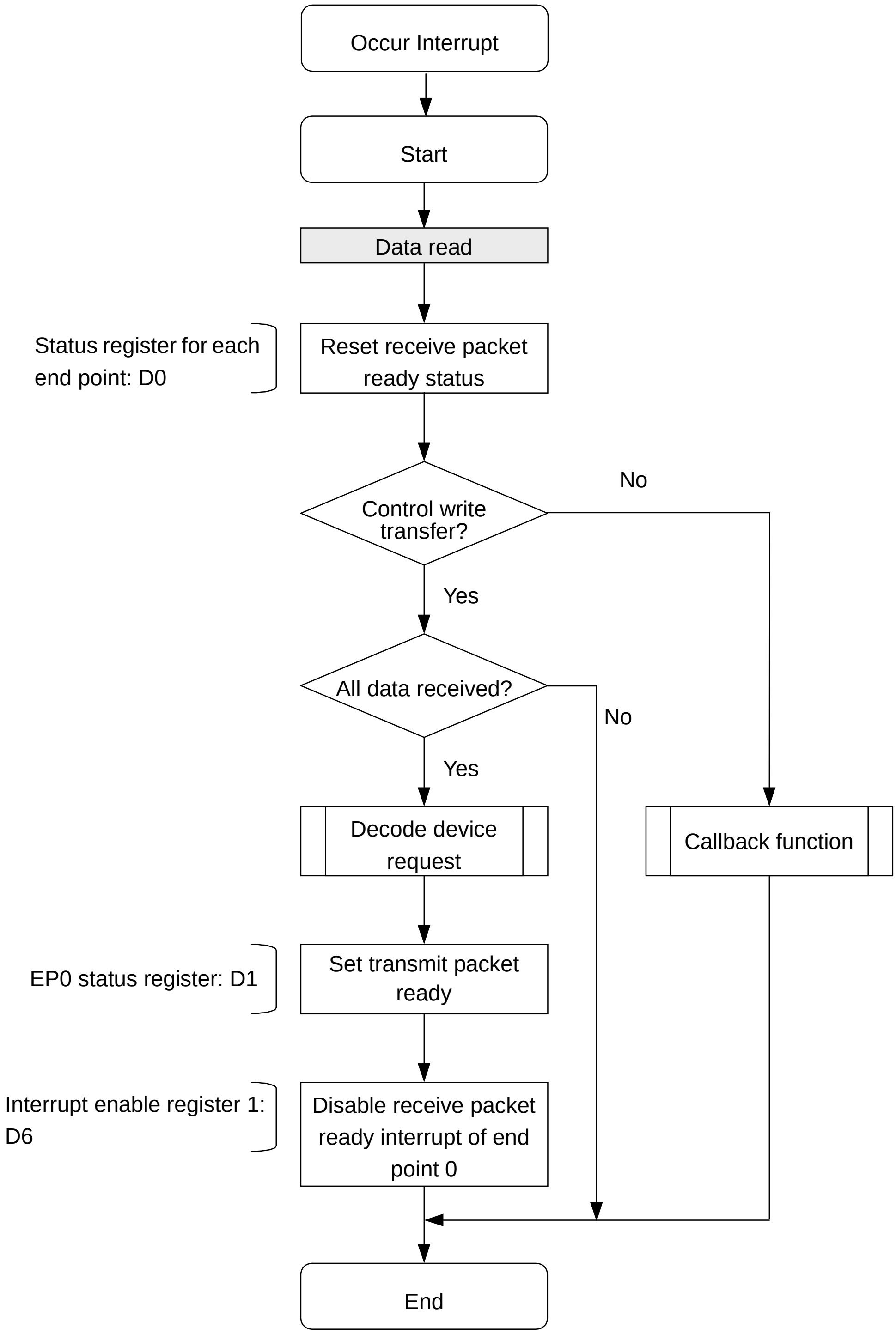
2.2.2. Setup ready interrupt procedure

The following flowchart, illustrates the outline flow of a control transfer from the point of view of the USB application firmware. The IN block denotes the entry point to an event driven firmware where a Setup-Ready interrupt has been detected by the device (i.e. a Setup packet has been sent from the host to the device and stored satisfactorily ML60852A's receive FIFO). Based on the type of request, the application firmware must determine whether the transfer is a Control Read, Control Write, or a Control transfer without a data stage. Hence, the transfer will be processed according to the Request Type.


Start
Disable EP0 transmit/ receive interrupt
Interrupt enable register 1: D6, D7
bmRequest Type register
bRequest register
wValue LSB register
wValue MSB register
wIndex LSB register
wIndex MSB register
wLength LSB register
wLength MSB register
Read request
End point 0 status register: D2
Reset setup ready status
Read transfer
Write transfer
No-Data transfer
Decode device request
Enable EP0 receive packet ready interrupt
Decode device request
Interrupt enable register 1: D6
Enable EP0 transmit packet ready interrupt
Set EP0 transmit packet ready status
Interrupt enable register 1: D7
End point 0 status register (EP0STAT): D1
End

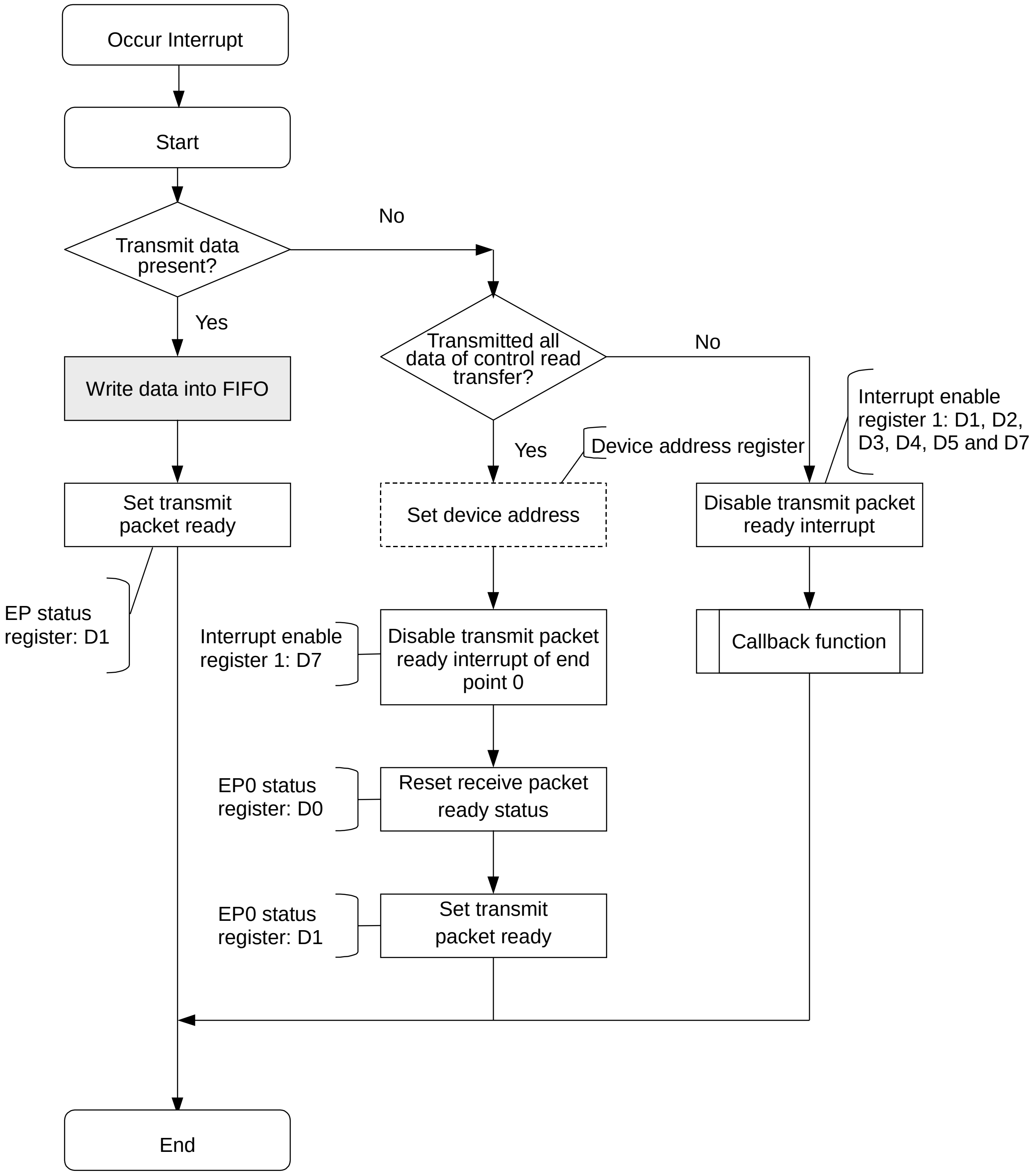
2.2.3. Reading received data in receive packet ready interrupt procedure

The following flow chart illustrates the data reception process from the point of view of the application firmware controlling the ML60852A. The IN block signifies the entry point to an event driven software, where a packet of data has been successfully received and stored in the FIFO of ML60852A. Hence, ML60852A generates an interrupt cause. A typical interrupt service procedure is outlined in this diagram. Note that the processing for the data stage of a Control Write transfer is also included below. Given that a control transfer is a message pipe (structured transfer) it can be seen that the processing of received data for this type of transfer is much more intricate than other types of transfers.


Occur Interrupt
Start
Data read
Status register for each end point: D0
Reset receive packet ready status
Control write transfer?
No
Yes
All data received?
No
Yes
Decode device request
Callback function
EP0 status register: D1
Set transmit packet ready
Interrupt enable register 1: D6
Disable receive packet ready interrupt of end point 0
End

2.2.4. Writing transmit data in transmit packet ready interrupt procedure

The following flow chart illustrates typical procedures for transmitting data from ML60852A device from the point of view of the application firmware controlling the device. The IN block below, signifies the entry point to an event driven firmware where a transmit interrupt cause has been generated. Note that most of the complexity of the transmission process lies in the end point 0 transmit procedure. This is due to the fact that end point 0 transmission is done through a Control Read transfer which is a message pipe (structured pipe) and hence much more tedious than other types of transfers.


Occur Interrupt
Start
Transmit data present?
No
Yes
Write data into FIFO
Set transmit packet ready
EP status register: D1
Transmitted all data of control read transfer?
No
Yes
Device address register
Set device address
Interrupt enable register 1: D7
Disable transmit packet ready interrupt of end point 0
EP0 status register: D0
Reset receive packet ready status
EP0 status register: D1
Set transmit packet ready
Interrupt enable register 1: D1, D2, D3, D4, D5 and D7
Disable transmit packet ready interrupt
Callback function
End

2.3. Bulk Transfer

2.3.1. Outline of bulk transfer

Bulk transfer is used when transmitting or receiving data whose quality has to be assured, such as print data, etc. Although the data quality is assured by a CRC check, since the priority order of transfer is lower than that of interrupt transfer or isochronous transfer, this mode of transfer is one in which the transfer efficiency may become lower when the load on the bus is heavy.

In the ML60852A, EP1 through EP4 (when in EP5 mode) or EP1 through EP5 (when in EP6 mode) can be used for bulk transfer. These end points can be allocated individually for both bulk-in and bulk-out. The FIFO size for bulk transfer is 64 bytes for each end point except EP3. The maximum FIFO size for EP3 is 32 bytes.

EP1, EP2, and EP4 (and also EP5 when in 6EP mode) of the ML60852A have the DMA transfer function. In addition, the FIFO of EP1, EP2, and EP4 (and also EP5 when in 6EP mode) have a 2-layer configuration, and it is possible to increase the transfer efficiency because one layer can be accessed by the local MCU when the other layer is exchanging data with the USB bus.

The registers used during bulk transfer are the following:

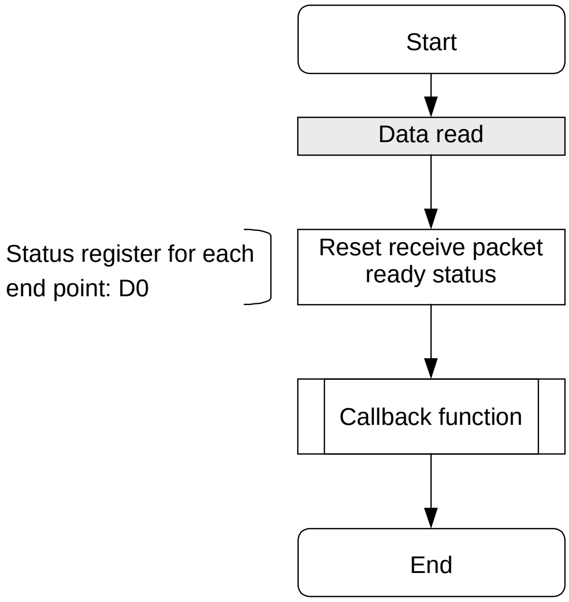
Table 3 Registers used during bulk transfer

Function	Register name	Bits
Packet ready	EP( )STAT	D0/D1
Packet ready interrupt status	INTSTAT1	D5 to D1
Packet ready interrupt enable	INTENBL1	D5 to D1
Configuration	EP( ) CONF	D0/D1/D4/D7
End point control	EP( )CONT	D0/D3
Data sequence toggle	EP( )CONT	D1
Maximum packet size (excluding EP3)	EP( )PLD	D6 to D0
Maximum packet size (EP3)	EP3PLD	D5 to D0
Receive byte count (excluding EP3)	EP( )RXCNT	D6 to D0
Receive byte count (EP3)	EP3RXCNT	D5 to D0
Transmit/receive FIFO	EP( )RXFIFO/EP( )TXFIFO	D7 to D0

Note: In the register name column, ( ) is a substitute for each number of the end points that are used for this transfer.

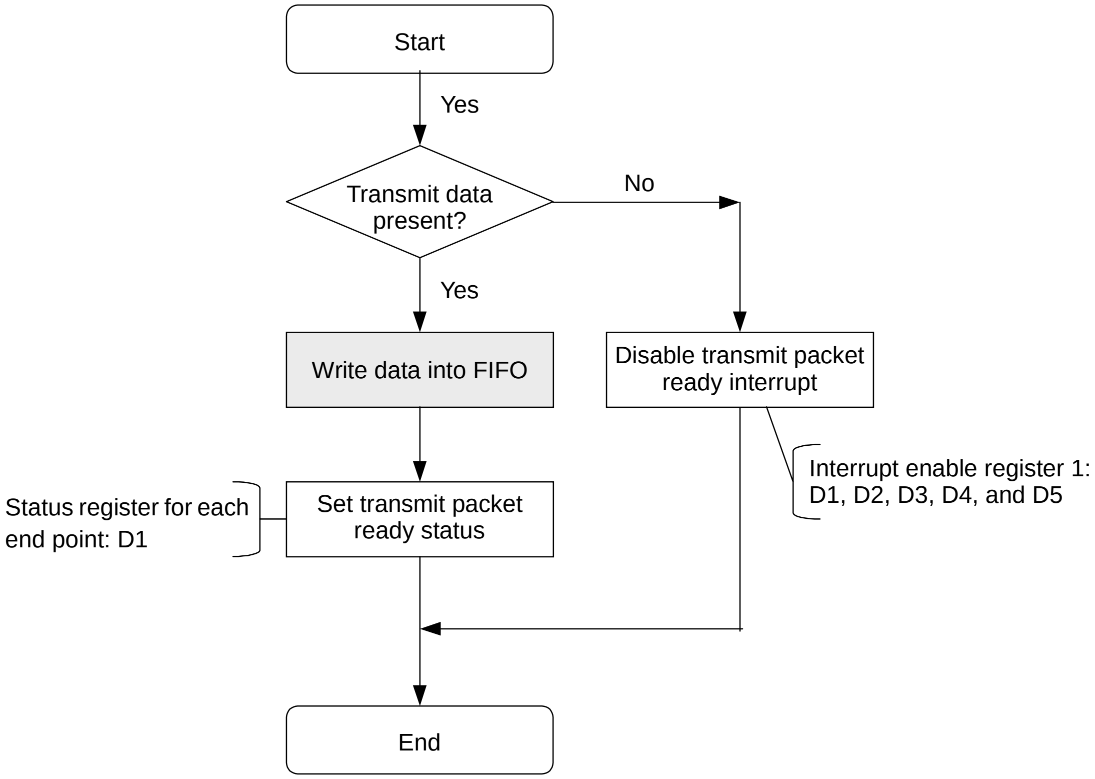
2.3.2. Packet ready interrupt procedure in Bulk-Out Transfer

The following flow chart illustrates the outline of a typical Bulk-Out transfer from the point of view of the application firmware controlling the ML60852A. The IN block shown below, denotes the entry point to an event driven application firmware where a receive interrupt cause has been generated in response to a data packet successfully received and stored in ML60852A's end point FIFO.


Start
Data read
Status register for each end point: D0
Reset receive packet ready status
Callback function
End

2.3.3. Packet ready interrupt procedure in Bulk-In Transfer

The following flow chart illustrates a typical procedure for carrying out a Bulk-In transfer from the point of view of an application firmware controller ML60852A. The IN block shown below, denotes the entry point to an event driven firmware where a Transmit Packet Ready interrupt cause has been generated and the firmware should service it.


Start
Yes
Transmit data present?
No
Yes
Write data into FIFO
Disable transmit packet ready interrupt
Interrupt enable register 1: D1, D2, D3, D4, and D5
Status register for each end point: D1
Set transmit packet ready status
End

2.4. Interrupt Transfer

2.4.1. Outline of interrupt transfer

Interrupt transfer is the method of periodically polling a data source for transmit information. This method is most suitable for transmitting moderate amounts of data within a specific amount of time. An interrupt transfer has a guaranteed maximum length between transaction attempts. In other words, the interrupt end point will be pinged (receive an IN token) within specific intervals specified in the device’s descriptor.. Also, interrupt transfers are carried out in only one direction; they are either all IN or all OUT transactions.

The structure of an interrupt transfer is identical to that of a bulk transfer. Please refer to section 2.5 “Bulk Transfer”, for the outline flow of interrupt transfer processing.

In ML60852A, EP1 through EP4 (when in EP5 mode) or EP1 through EP5 (when in EP6 mode) can be used for interrupt transfer. These end points can be allocated individually for both interrupt-in and interrupt-out. The FIFO size for interrupt transfer is 64 bytes for each end point except EP3. If EP3 is used for interrupt transfer, the FIFO size is 32 bytes.

The registers used during interrupt transfer are the following:

Table 4 Registers used during interrupt transfer

Function	Register name	Bits
Packet ready	EP( )STAT	D0/D1
Packet ready interrupt status	INTSTAT1	D5 to D1
Packet ready interrupt enable	INTENBL1	D5 to D1
Configuration	EP( ) CONF	D0/D1/D4/D7
End point control	EP( )CONT	D0/D3
Data sequence toggle	EP( )CONT	D1
Maximum packet size (excluding EP3)	EP( )PLD	D6 to D0
Maximum packet size (EP3)	EP3PLD	D5 to D0
Receive byte count (excluding EP3)	EP( )RXCNT	D6 to D0
Receive byte count	EP3RXCNT	D5 to D0
Transmit/receive FIFO	EP( )RXFIFO/EP( )TXFIFO	D7 to D0

Note: In the register name column, ( ) is a substitute for each number of the end points that are used for this transfer.

2.5. Isochronous Transfer

2.5.1. Outline of isochronous transfer

Isochronous transfer is used to transfer data such as voice data for which the real-time performance takes precedence. The priority of this transfer is high and data with a certain size can be cyclically transferred. In order to allow high reliability communication, USB standards devised isochronous transfers in which bandwidth reservation takes precedence over error control.
In isochronous transfer, however, no handshake (ACK/NAK) is provided, so occurrence of CRC errors is not reported to the sender and re-transmit is not performed. To re-transmit the packet that resulted in an error, an original protocol should be formulated.

In the ML60852A, in 5EP mode, EP4 is available and in 6EP mode, EP4 and EP5 are available for isochronous transfer. For these end points, the transfer direction can be individually specified. The FIFO size of EP4 in 5EP mode is 512 bytes. In 6EP mode, the FIFO size of EP4/EP5 is 256 bytes.
EP4 used in isochronous transfer (or EP4 and EP5 in 6EP mode) allows DMA transfer. The FIFO of EP4 (or EP4 and EP5 in 6EP mode) has a 2-layer configuration, and it is possible to increase the transfer efficiency because one layer can be accessed by the local MCU when the other layer is exchanging data with the USB bus.

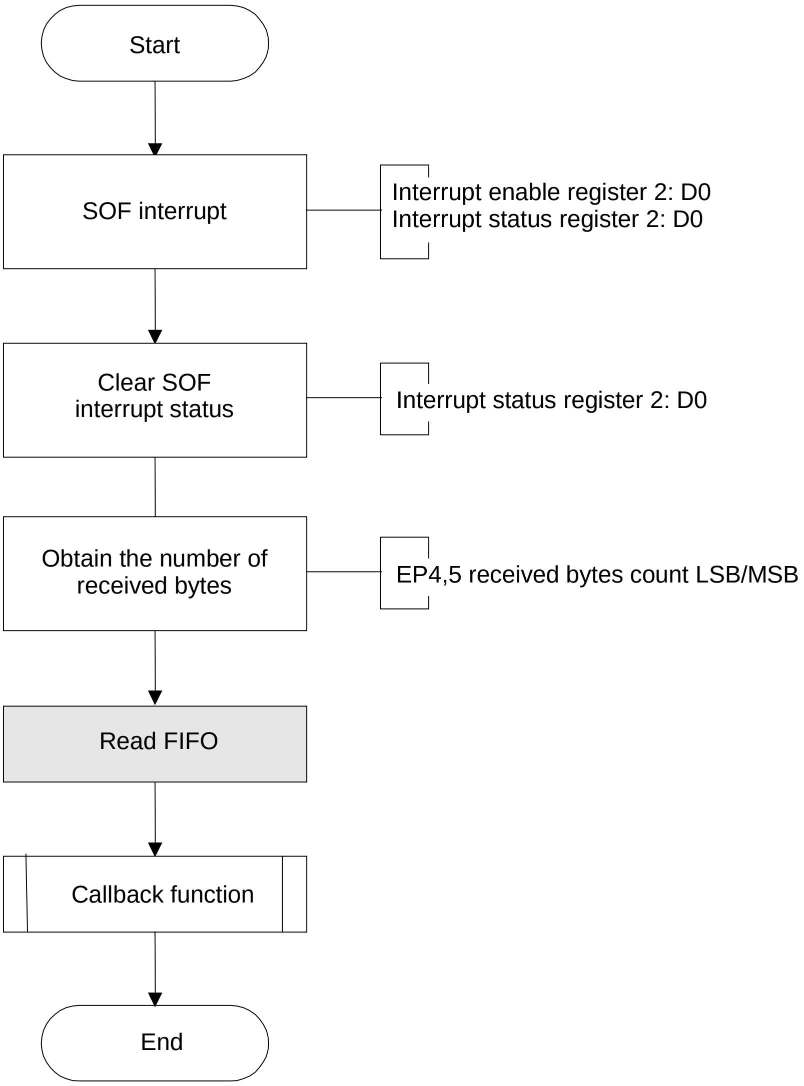
Table 5 Registers used during isochronous transfer

Function	Register name	Bits
SOF interrupt status	INTSTAT2	D0
SOF interrupt enable	INTENBL2	D0
Configuration	EP( )CONF	D0/D1/D4/D7
End point control	EP( )CONT	D0/D3
Maximum packet size (LSB)	EP( )PLDLSB	D7 to D0
Maximum packet size (MSB)	EP( )PLDMSB	D1 to D0
Receive byte count (LSB)	EP( )RXCNTLSB	D7 to D0
Receive byte count (MSB)	EP3RXCNTMSB	D1 to D0
Transmit/receive FIFO	EP( )RXFIFO/EP( )TXFIFO	D7 to D0

Note: In the register name column, ( ) is a substitute for each number of the end points that are used for this transfer.

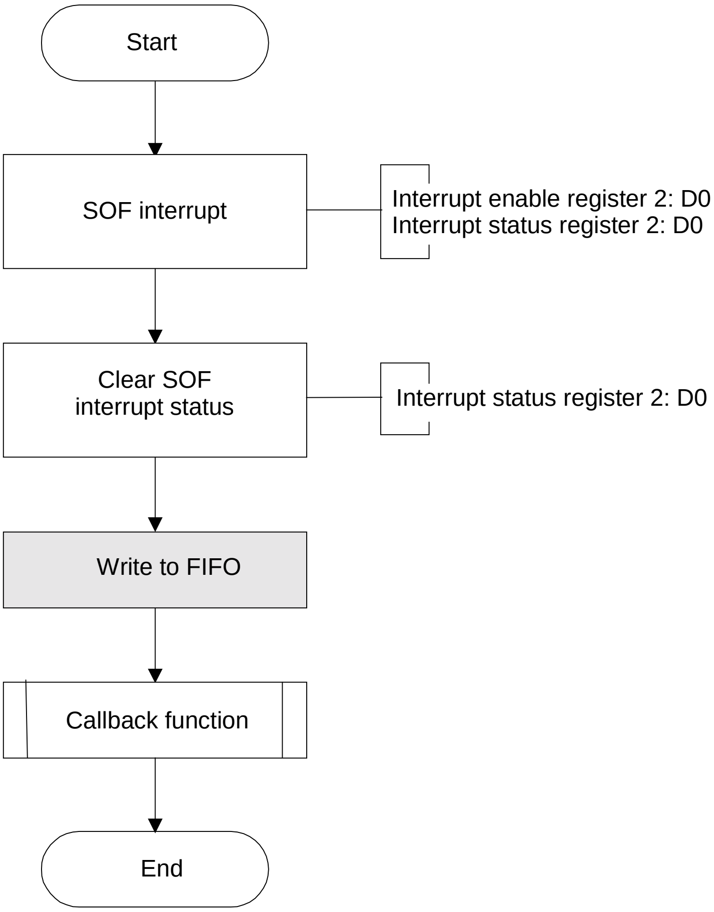
2.5.2. Outline flow of isochronous-out transfer

The following diagram is an illustration of outline flow of an isochronous-out transfer from the point of view of an application firmware controlling ML60852A. As specified by USB standards, a single packet of isochronous data may be received in each USB frame. As a result, the packet reception process is started with detection of a SOF PID on the USB bus.


Start
SOF interrupt
Interrupt enable register 2: D0
Interrupt status register 2: D0
Clear SOF interrupt status
Interrupt status register 2: D0
Obtain the number of received bytes
EP4,5 received bytes count LSB/MSB
Read FIFO
Callback function
End

2.5.3. Outline flow of isochronous-in transfer

The following diagram is an illustration of outline flow of an isochronous-in transfer from the point of view of an application firmware controlling ML60852A. As shown below, given that a single packet of isochronous data may be transmitted in each USB frame, the transmission processing procedure starts with detection of a SOF packet on the bus.


Start
SOF interrupt
Interrupt enable register 2: D0
Interrupt status register 2: D0
Clear SOF
interrupt status
Interrupt status register 2: D0
Write to FIFO
Callback function
End

2.5.4. Errors in isochronous-out transfer

If a CRC error occurs during isochronous-out transfer, the temporary received bytes count register of the ML60852A stores the number of received bytes at that point.

Although the host does not make a re-transmit request when a CRC error occurs in isochronous transfer, a function for re-transmit can be devised by using the temporary received bytes count register.

The ML60852A has temporary received bytes count register for EP4 and EP5.

3. EXTERNAL INTERFACE

The functions and the method of use of the pins for connecting a microcontroller or an external peripheral device to the ML60852A are described in this chapter.

3.1. Bus Interface

3.1.1. ADSEL pin

The ML60852A supports the Separate and Multiplex methods of accessing data and address buses. ALE pin determines which of these two methods is used.
The Separate method is used when ADSEL is “L”. In this case, address lines are connected to AD6 to AD0 and data lines are connected to AD7 to AD0.
The Multiplex method is used when ADSEL is “H”. In this case, address and data lines are connected to AD7 to AD0.

3.1.2. ALE/PUCTL pin

This pin functions differently depending on the ADSEL pin level.
When ADSEL is “H”, the address input via AD7 to AD0 and –CS are latched at the trailing edge of ALE.

When ADSEL is “L”, ALE/PUCTL functions as the pull-up control pin on the D+ line. For details of the pull-up control on the D+ line, see Section 3.5. “USB Interface”.

3.2. DMA Interface

The ML60852A supports DMA transfer on two channels, which makes it possible to carry out DMA transfer with two end points. Channel 0 and channel 1 are assigned to –DREQ and DACK respectively. The settings related to DMA transfer are made by the registers POLSEL, DMA0CON, DMA0INTVL, DMA1CON, and DMA1INTVL.

3.2.1. Selection of –DREQ and DACK polarities

It is possible to select the polarities of –DREQ and DACK individually. The polarities are set by bits D3 to D0 of the register POLSEL at address 30h.
When the corresponding bit is “0”, –DREQ0 and –DREQ1 become active-Low and DACK0 and DACK1 become active-High.
When the corresponding bit is “1”, –DREQ0 and –DREQ1 become active-High and DACK0 and DACK1 become active-Low.

After a hardware reset or a software reset, –DREQ0 and –DREQ1 will be active-Low and DACK0 and DACK1 will be active-High.

3.2.2. –DREQ Active conditions

3.2.2.1. During transmission

The ML60852A makes –DREQ0 and –DREQ1 active when the preparations have been completed for writing data in the transmit FIFO.
When writing data into the transmit FIFO is completed and the transmit packet ready bit is set to “1”, the ML60852A makes –DREQ0 and –DREQ1 inactive.
Subsequently, when all the data written into the FIFO has been transmitted and an ACK handshake is received from the host computer, the ML60852A resets again the transmit packet ready bit to “0” and makes –DREQ0 and –DREQ1 active.

3.2.2.2. During reception

The ML60852A sets the receive packet ready bit and makes –DREQ0 and –DREQ1 active when the entire data packet from the host computer has been received normally and the ACK handshake is sent back to the host computer.
When the receive packet ready bit is reset (a “1” is written) after all the received data is read out, –DREQ0 and –DREQ1 are made inactive by the ML60852A.
When the next data packet is received normally again, the ML60852A sets again the receive packet ready bit to “1” and makes –DREQ0 and –DREQ1 active.

3.2.3. DMA Enable

The control of enabling and disabling DMA transfer is made using bit D0 of the registers DMA0CON (address 10h) and DMA1CON (address 12h).
When the DMA enable bit is “0”, DMA transfer is disabled and –DREQ will not become active.
When the DMA enable bit is “1”, DMA transfer is enabled, and then –DREQ becomes active when the conditions for generating –DREQ are satisfied.

3.2.4. Selecting the EP that uses DMA transfer

EPs that can use DMA transfer in the ML60852A are EP1, EP2, EP4, and EP5.
Which EP will use DMA transfer is set with D6 and D5 of the DMA0CON and DMA1CON registers.
If D6 and D5 = “00”, EP1 is selected. If D6 and D5 = “01”, EP2 is selected. If D6 and D5 = “10”, EP4 is selected. If D6 and D5 = “11”, EP5 is selected.
If channels 0 and 1 of DMA specify the same value, –DREQ0 is equivalent to –DREQ1 and DACK0 is equivalent to DACK1.

3.2.5. Address modes during DMA transfer

The ML60852A supports two address modes.

When the bit D1 of the registers DMA0CON and DMA1CON is “0”, the ML60852A operates in the single address mode, and operates in the dual address mode when this bit is “1”.

In the dual address mode, DACK is not used (see Chapter 6 “HANDLING UNUSED PINS” for the handling of DACK during the dual address mode).

3.2.6. Transfer modes during DMA transfer

The ML60852A supports the single transfer and demand transfer modes. The transfer mode is determined by using D4 of the registers DMA0CON and DMA1CON.

Set D4 to “0” when using the single transfer mode during DMA transfer and to “1” when using the demand transfer mode.

During the demand transfer, –DREQ is made active when it becomes possible to read or write a packet of data. –DREQ is made inactive when all the receive or transmit data has been transferred by the DMA controller. Therefore, other devices cannot access the local bus during DMA transfer.

On the other hand, in the single transfer mode, since –DREQ becomes inactive every time the bytes (or words) of one transfer have been transferred completely, the other devices can access the local bus during this interval.

3.2.7. Byte count data insertion

The ML60852A can insert the number of bytes of the packet received during DMA transfer into the leading byte (or leading word) of the transferred data.

Insertion of byte count data is made when it is necessary for the local MCU to know the number of data bytes in the received packet.

Set D2 of the registers DMA0CON and DMA1CON to “1” when wanting to insert the byte count data. The byte count data is not inserted when D2 is “0”.

During the 16-bit transfer mode, the upper byte of the byte count data (the leading word) will become 00h.

3.2.8. Transfer data width during DMA transfer

The ML60852A supports 8-bit DMA transfer and 16-bit DMA transfer. This transfer data width is selected using D3 of the registers DMA0CON and DMA1CON.

Reset D3 to “0” for 8-bit DMA transfer and set the bit to “1” for 16-bit DMA transfer.

The upper byte and lower byte are assigned to the data line in the little-endian sequence. That is, the LSB corresponds to AD0 to AD7 and the MSB corresponds to D8 to D15.

The upper byte of the last word is 00h when 16-bit transfer is used and the packet size is an odd number of bytes.

During 16-bit transfer, an odd number of bytes of data cannot be transmitted.

3.2.9. Interrupting DMA transfer

In the ML60852A, it is possible to interrupt a DMA transfer in progress. Interrupting and restarting DMA transfer is controlled using D7 of the registers DMA0CON and DMA1CON.

Set D7 to “1” in order to temporarily halt a DMA transfer in progress. When restarting DMA transfer by resetting D7 to “0” again, the transfer can be restarted from the byte (or word) next to the one at which the transfer is interrupted.

The setting of bits related to DMA transfer other than D7 must be completed during initialization (at the latest, before the token packet for EP1 to EP5 arrives) and should not be modified thereafter.

3.2.10. –DREQ Signal interval during single transfer

In the ML60852A, it is possible to set the interval during the single transfer mode, that is, the interval until –DREQ is asserted again after the completion of DMA transfer of the previous byte (or word).
This interval setting is made using the registers DMA0INTVL (address 11h) and DMA1INTVL (address 13h).
One bit time is 84 ns (12 MHz, duration for 1 cycle).

Interval duration = (DREQ Enable time) + 84 × n (ns)

3.2.11. Setting packet ready during DMA transfer

When writing data into the FIFO of the ML60852A using DMA transfer (bulk-in transfer), there is no need to set the packet ready bit of each end point from the local MCU side. When data with the maximum packet size is written in the FIFO by the DMA controller, the ML60852A automatically sets the packet ready bit. Therefore, there will be no need for the local MCU to access the ML60852A in the middle of DMA transfer by setting the size of data to be transferred to the byte count register of the DMA controller at the beginning of DMA transfer.

However, when the last packet to be transferred is a short packet (or a zero-length packet), it is necessary for the local MCU to set the packet ready bit. In this case, it is possible to set the packet ready bit by triggering the DMA transfer end interrupt that is issued by the DMA controller to the local MCU.

3.3. Interrupt Interface

The ML60852A has an interrupt function. The ML60852A issues an interrupt request to the local MCU by making the –INTR pin active.
See Chapter 4 "INTERRUPTS" for the interrupt causes.

3.3.1. Selection of –INTR pin polarity

It is possible to select the polarity of the –INTR pin by using D0 of the POLSEL register at address 30h.
When D4 is "0", –INTR is active-Low.
When D4 is "1", –INTR is active-High.

Immediately after a hardware reset or a software reset, –INTR is active-Low.

3.3.2. Processing method for nested interrupts

When the MCU processes interrupts by edge detection, it is impossible to detect interrupts when there are multiple interrupt causes because the ML60852A continues to maintain –INTR in the active state and no edge is generated.

In this case, it is possible to take the following countermeasures:

Countermeasures:
At the time of completing the servicing of an interrupt, clear the external interrupt cause from the MCU, and save the contents of the ML60852A interrupt enable register.
Next, clear the enable register (make all bits "0") and write again the saved contents thereby generating an edge.

Note: This is only one sample countermeasure, and the method of interrupt control differs depending on the MCU. Therefore, please verify the operation for individual MCUs.

3.4. Oscillation Circuit

The ML60852A has a built-in oscillator circuit. By connecting a crystal or a ceramic resonator externally, the necessary oscillation clock for running the ML60852A can be obtained.

Make sure that the accuracy of the oscillation clock is within ±0.25%, taking temperature characteristics and aging into account.

3.4.1. Oscillation frequency

In the ML60852A, either 6 MHz or 12 MHz can be selected as the oscillation frequency of the crystal or ceramic resonator.
The oscillation frequency is selected by using D6 of the SYSCON register at address 2Fh.
To use the 12 MHz crystal or ceramic resonator, set D6 to “0”. To use the 6 MHz one, set D6 to “1”.

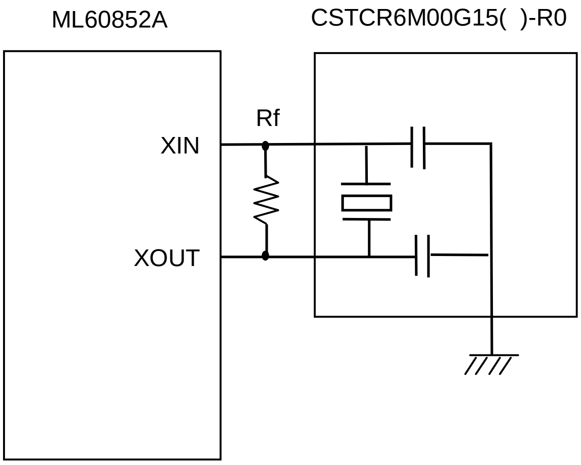
When supplying an external clock instead of using the built-in oscillator circuit, 48 MHz can also be used in addition to 6 MHz and 12 MHz.
For use of an external clock, see Section 3.4.3 “Supplying an external clock”.

3.4.2. Oscillator circuit configuration examples

When using a crystal or ceramic resonator, it is required to enable the built-in PLL. The PLL can be enabled/disabled with D5 of the SYSCON register.
If D5 is “1”, the PLL is enabled. In this case, the signal that is input to the XIN pin is multiplied by the PLL and becomes the source oscillation of the internal circuit.
If D5 is “0”, the PLL is disabled. In this case, the signal that is input to the XIN pin becomes the source oscillation of the internal circuit.
The PLL is only disabled when a 48 MHz clock is externally supplied. (See Section 3.4.3 “Supplying an external clock”)

3.4.2.1. Using a ceramic resonator

An example of configuration using a ceramic resonator is shown below.



Ceramic resonator: CSTCR6M00G15( )-R0 of Murata MFG. make (built-in capacitor type)
Rf = 1 MΩ

Figure 3 Example of oscillator circuit using a ceramic resonator (6 MHz)

3.4.3. Supplying an external clock

Instead of using the built-in oscillator circuit, an external clock can be supplied to the ML60852A.
For the external clock, 48 MHz can also be used in addition to 6 MHz and 12 MHz.
When using a 6 or 12 MHz external clock, set D5 of the SYSCON register to “1” to enable the PLL then select a frequency with D6 of the SYSCON register.
To supply a 48 MHz clock externally, set D5 of the SYSCON register to “0” to disable the internal PLL.

When using an external clock, care must be taken to the following:

- Input the clock to the XIN pin and leave the XOUT pin open. (See Chapter 6 “HANDLING UNUSED PINS”.)
- Supply a clock with a duty ratio of 40 to 60%.
- The XIN pin of the ML60852A does not support 5 V input. Therefore, supply a 3 V-base external clock. Never supply a 5 V-base clock.
- As when using the built-in oscillation circuit, make sure that the external clock accuracy is within ±0.25%, taking temperature characteristics and aging into account.

3.4.4. Stopping the oscillator circuit

The ML60852A has a function that stops the oscillator circuit to enter the low-power state.
If D1 of the SYSCON register at address 2Fh is set to “1” in advance, the ML60852A stops oscillation 2 ms after suspended state interrupt generation. If D1 of the SYSCON register is “0”, oscillation does not stop even after occurrence of the suspended state.

Furthermore, the oscillator circuit of ML60852A can be arbitrarily stopped anytime except during the suspended state. Setting D2 of the OSCTEST register (address 3Ah) to “1” will stop the ML60852A’s oscillator circuit. To resume oscillation, set D2 of the OSCTEST register to “0”. Always set the other bits of the OSCTEST register to “0”.

Note: Use the oscillation stop function of OSCTEST only when USB communication is not in progress (e.g. the cable is disconnected). To stop oscillation during the suspend state, set D1 of the SYSCON register to “1” and set all bits of the OSCTEST register to “0”. If oscillation is stopped during normal operation, system malfunction may occur.

Table 6 OSCTEST register setting for stopping/resuming oscillation

D2 of OSCTEST register	Other than D2	Action
1	0	Stop oscillation
0	0	Resume oscillation

3.5. USB Interface

3.5.1. Series resistors in the D+/D– lines

The ML60852A has the D+ and D– lines as the interface with the USB bus. It is necessary to insert series resistors between D+/D– and the USB connector. An example of the circuit near the USB connector is shown below.

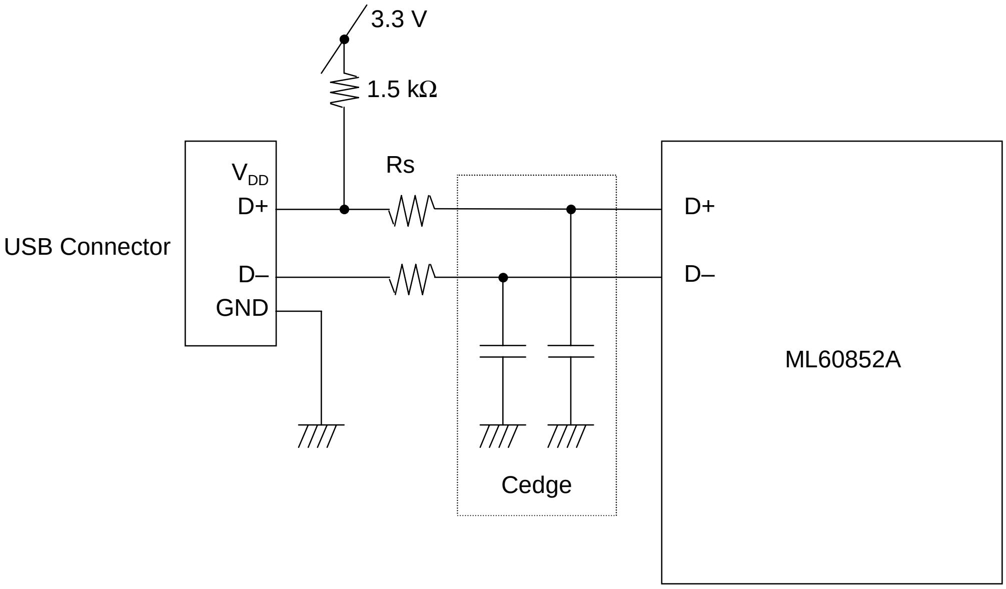


Figure 4 Sample circuit configuration of the USB bus interface

- Rs (recommended) = 22Ω (accuracy: ±5%, rated power dissipation: 0.125 W)
- The edge rate control capacitors (Cedge) are basically not needed. However, when needed, provide them as shown in the dotted line block in the above figure.
- The pull-up resistor on the D+ line should be connected on the USB connector side of the series resistor Rs.
- It is recommended that the pulled-up of 3.3 V supply on the D+ line should be either generated by regulating VBUS or should be controlled by VBUS. This is for preventing the pulling up being done in the condition in which the cable is not connected (see Section 3.5.2 “Pull-up control on the D+ line”).
- Make the distance between the ML60852A and the USB connector as short as possible. Also, it is recommended that the connections between the ML60852A and the USB connector be made on the same printed circuit board.

3.5.2. VBUS monitoring

The USB requires a system configuration that does not allow the D+ line to be pulled up when VBUS (USB power supply line) is OFF. For more information, refer to Section 7.1.5 “Device Speed Identification” in the USB Standard Rev. 1.1.
Although it is possible to configure the system mentioned above by regulating VBUS to 3.3 V thereby directly pulling up the D+ line, system stability is further improved if the local MCU monitors the VBUS state and controls pull-up.
Therefore, it is recommended that the following functions be implemented in the system:

- The D+ line will not be pulled up → if VBUS is OFF (i.e. the cable is disconnected or the host power is OFF)
- The D+ line will be pulled up → if VBUS is ON and the device is ready

One of the ports of the local MCU can be used to monitor the VBUS.

4. INTERRUPTS

There are 17 types of interrupt causes in the ML60852A. The interrupt causes are assigned to each bit of the registers INTSTAT1 at address 21h, and INTSTAT2 at address 22h, and the corresponding bit becomes “1” when an interrupt cause is generated. Although the –INTR pin is a single line, the local MCU can identify the contents of the interrupt by referring to INTSTAT1 and INTSTAT2.
It is possible to enable or disable each interrupt using the registers INTENBL1 at address 24h, INTENBL2 at address 25h, and INTENBLA at address 39h. The bits of the registers correspond to the respective bits of the registers INTSTAT1 and INTSTAT2. By setting the bit corresponding to the necessary interrupt to “1”, it is possible to enable only the required interrupt. Resetting the bit to “0” disables the corresponding interrupt.
The bits of the registers INTSTAT1 and INTSTAT2 will always be “0” when the corresponding bits of the INTENBL1, INTENBL2, and INTENBLA registers are “0” (disabled).

Table 7 Interrupt causes (INTSTAT1 register)

Bit	Interrupt cause
D0	Setup ready interrupt
D1	EP1 packet ready interrupt
D2	EP2 packet ready interrupt
D3	EP3 packet ready interrupt
D4	EP4 packet ready interrupt
D5	EP5 packet ready interrupt
D6	EP0 receive packet ready interrupt
D7	EP0 transmit packet ready interrupt

Table 8 Interrupt causes (INTSTAT2 register)

Bit	Interrupt cause
D0	SOF interrupt
D1	USB bus reset assert interrupt
D2	USB bus reset deassert interrupt
D3	Device suspend state interrupt
D4	Device awake state interrupt
D5 to D7	Reserved

Each of the interrupt causes are described in detail below.

4.1. Setup Ready

This is an interrupt that is generated during a control transfer in which the host computer issues a device request via the control pipe. When the 8 bytes of setup data are received in the setup stage of control transfer and are properly stored in the setup registers of the ML60852A, the ML60852A sets a “1” in bit D2 (Setup ready) of the register EP0STAT at address 60h.

At this time, if D0 of the register INTENBL1 has already been set to “1”, D0 of the register INTSTAT1 will be set to “1” at the same time as D2 of the register EP0STAT is set to “1”, and –INTR becomes active.

After reading all the setup data, writing a “1” into D2 of the register EP0STAT and resetting the setup ready status make D0 of the register INTSTAT1 be reset to “0” and –INTR becomes inactive.
The setup ready interrupt is also enabled after a system reset (hardware reset or software reset). (All other interrupts will be disabled after a system reset.)

4.2. Receive Packet Ready

In each transfer mode, when one packet of data is received from the host computer and stored correctly in the receive FIFO of the corresponding end point, the ML60852A automatically sets to “1” the receive packet ready bit of the end point status register (EPnSTAT) corresponding to that end point.
At this time, if this interrupt has already been enabled by setting the corresponding bit of the register INTENBL1 to “1”, the corresponding status bit of the register INTSTAT1 will be set to “1” at the same time as the receive packet ready bit is set to “1”, and –INTR becomes active.
After reading all the data in the receive FIFO, when the local MCU writes a “1” into the receive packet ready status bit thereby resetting the packet ready status, the corresponding bit of the register INTSTAT1 is also reset to “0” and –INTR becomes inactive.
The ML60852A recognizes the end of a packet when EOP (end of packet) is received, whether full packets (packets containing data of the maximum packet size) or short packets.
The mutual correspondences between the end points (EP), the packet ready bits, and interrupts are shown below.

Table 9 Correspondence between end points, packet ready bits, and bits of the interrupt register (during reception)

EP	EP status register	Status register address	Status register bit	INTENBL1 (INTSTAT1)
0	EP0STAT	60h	D0	D6
1	EP1STAT	61h	D0	D1
2	EP2STAT	62h	D0	D2
3	EP3STAT	63h	D0	D3
4	EP4STAT	64h	D0	D4
5	EP5STAT	65h	D0	D5

4.3. Transmit Packet Ready

If an endpoint has been configured for transmission, when the corresponding packet ready interrupt enable bit of the INTENBL1 register is set (write “1”), an interrupt cause is generated. Hence the corresponding bit of the INTSTAT1 register is automatically set to “1” and -INTR pin is asserted. The cause of the interrupt is transmit packet ready status of the endpoint (EPnSTAT) being low (“0”). This interrupt is an indication to the application firmware (local MCU) to write the transmit data into the transmit FIFO of the endpoint.

Once the outgoing data has been written to the transmit FIFO of the endpoint, the local MCU should set the transmit packet ready status (bit D1 of EPnSTAT register) of the end point. This action will remove the interrupt cause and hence deassert the interrupt. At this point, ML60852A is ready to transmit its FIFO contents on to the USB bus in response to an IN token from the host.

When the contents of the endpoint FIFO are transmitted onto the bus due to reception of an IN token, the host will transmit an ACK handshake packet in response to the successful transmission of the data packet. This ACK will cause the ML60852A to automatically reset to “0” the corresponding transmit packet ready bit of the EPnSTAT (bit D1) register and hence generate an interrupt cause. This interrupt indicates that the device transmit FIFO is ready to be written to again. This process can be continued for as long as there is available data to be transmitted.

When all the transmit data has been successfully transmitted, the application firmware (local MCU) should disable the corresponding packet ready interrupt bit in INTENBL1 register.

It is possible to transmit a short packet by setting the packet ready status bit to “1” after writing data shorter than the maximum packet size into the FIFO.

The mutual correspondences between the end points (EP), the packet ready bits, and interrupts are shown below.

Table 10 Correspondence between end points, packet ready bits, and bits of the interrupt register (during transmission)

EP	EP status register	Status register address	Status register bit	INTENBL1 (INTSTAT1)
0	EP0STAT	60h	D1	D7
1	EP1STAT	61h	D1	D1
2	EP2STAT	62h	D1	D2
3	EP3STAT	63h	D1	D3
4	EP4STAT	64h	D1	D4
5	EP5STAT	65h	D1	D5

4.4. SOF

If SOF (Start Of Frame) is detected on the USB bus, an SOF interrupt occurs.

If D0 of the INTENBL2 register at address 25h is set to “1”, the SOF interrupt is enabled. If the ML60852A detects SOF on the USB bus, D0 of the INTSTAT2 register at address 22h is set to “1” and –INTR becomes active.

When the local MCU writes “1” to D0 of the INTSTAT2 register, D0 of the INTSTAT2 register is reset to "0" and –INTR becomes inactive.

4.5. USB Bus Reset Assert

A bus reset is caused when the SE0 state continues for 2.5 µs or more on the USB bus.
The USB bus reset assert interrupt is enabled when a “1” is written to D1 of the register INTENBL2 at address 25h.
In this case, when the ML60852A detects an SE0 state for 2.5 µs or more, D1 of the INTSTAT2 register at address 22h is set to “1” and –INTR becomes active.
When the local MCU writes a “1” to D1 of the INSTAT2 register, D1 of the register INTSTAT2 is reset to “0” and –INTR becomes inactive.

The USB bus reset assert interrupt allows the local MCU to run the function within firmware that corresponds to the bus reset.

4.6. USB Bus Reset Deassert

If the SE0 state continues for 2.5 µs or more on the USB bus (bus reset) and then the SE0 state changes into the J state (idle state), USB bus reset completes.
If D2 of the INTENBL2 register at address 25h is set to “1”, the USB bus reset deassert interrupt is enabled. If the ML60852A detects the return to the J state (idle state) after an SE0 state that continued for 2.5 µs or more, D2 of the INTSTAT2 register at address 22h is set to “1” and –INTR becomes active.
When the local MCU writes “1” to D2 of the INTSTAT2 register, D2 of the INTSTAT2 register is reset to “0” and –INTR becomes inactive.

With this interrupt, the local MCU can recognize completion of bus reset.

See Section 7.1.7.3 “Reset Signaling” in the USB Standard Rev.1.1 for details of a bus reset.

4.7. Suspend State

When there is no bus activity (no SOF) on the USB bus for 3 ms or more (the idle state), the bus goes into the suspend state. The suspend state interrupt is enabled if D3 of the register INTENBL2 is set to “1”. At this time, if the ML60852A detects the idle state for 3 ms or more, D3 of the register INTSTAT2 at address 22h is set to “1” and –INTR becomes active.

See Section 7.1.7.4 “Suspending” or Section 9.1.1.6 “Suspended” in the USB Standard Rev.1.1 for details of the suspend function.

4.8. Awake

If the resume state (i.e. SE0 state that continues for approximately 1344 ns immediately after the K state) is detected in the suspend state, awake (i.e. resume) occurs.
If D4 of the INTENBL2 register is set to “1”, the awake interrupt is enabled. In this case, if the ML60852A detects the SE0 state for approximately 1344 ns immediately after the K state, D4 of the INTSTAT2 register at address 22h is set to “1” and –INTR becomes active.
When the local MCU writes “1” to D4 of the INTSTAT2 register, D4 of the INTSTAT2 register is reset to “0” and -INTR becomes inactive.

Although the awake interrupt occurs upon ending of the resume state, oscillation in power saving mode resumes when the resume state starts (i.e. when the K state is detected).

5. OTHER FUNCTIONS

The other functions of the ML60852A are described in this chapter.

5.1. 5EP Mode and 6EP Mode

The ML60852A is provided with two EP modes.
If D2 of the SYSCON register at address 2Fh is set to "0", the 6EP mode is entered. If D2 of the SYSCON register is set to "1", the 5EP mode is entered.
In 5EP mode, EP0 to EP4 are available. In 6EP mode, EP0 to EP5 are available.
When EP4 is used for isochronous transfer in 5EP mode, the FIFO of EP4 is switched to a 512-byte two-layer configuration.
When EP4/EP5 is used for isochronous transfer in 6EP mode, the FIFO of EP4/EP5 is switched to a 256-byte two-layer configuration.

5.2. System Reset

The ML60852A has the functions of hardware reset and software reset as methods of carrying out a system reset.
The hardware reset is initiated by making active the –RESET pin of the ML60852A.
The same purpose can also be achieved by a software reset which is that of carrying out a system reset by writing a “1” into D0 of the system control register (SYSCON) at address 2Fh. The hardware reset and the software reset are functionally identical.

5.3. Self-powered and Bus-powered

The ML60852A supports both Self-powered and Bus-powered operation.
If the ML60852A is used in the Bus-powered mode, the low-power mode should be selected during the suspend state.
If D1 of the SYSCON register at address 2Fh is set to “1”, the ML60852A stops oscillation in the suspend state and enters the low-power state.

For the low-power mode during the suspend state, see Section 5.4 “Suspend Function”.

5.4. Suspend Function

The ML60852A goes into the suspend state when the idle state in which there is no bus activity on the USB bus, continues for 3 ms or more.
If D3 of the register INTENBL2 at address 25h has already been set to “1”, when the ML60852A detects an idle state for 3 ms or longer, D3 of the register INTSTAT2 at address 22h is set to “1” and –INTR becomes active. The suspend state interrupt allows the local MCU to start processing for suppressing power consumption. (See Chapter 5 “INTERRUPTS”.)

The ML60852A has a function to suppress power consumption during suspend state. If bit D1 of register SYSCON at address 2Fh is preset to “1”, the ML60852A will stop oscillation 2 ms after a suspended state interrupt is caused. When D1 of register SYSCON is “0”, oscillation will not stop even if the suspend state is entered

See Section 7.1.7.4 “Suspending” or Section 9.1.1.6 “Suspended” in the USB Standard Rev.1.1 for details of the suspend function.

5.5. Remote Wakeup

The ML60852A supports the remote wakeup function.
The remote wakeup function can only be used in the suspend state.
To use the remote wakeup function, set D4 of the register SYSCON at address 2Fh to “1”. If D4 of the SYSCON register is set to “1” when the time elapsed after the USB bus has entered the idle state is less than 5 ms, the ML60852A waits until the idle state continues for 5 ms or more and then restarts oscillation and outputs the remote wakeup signal onto the USB bus. If D4 of the SYSCON register is set to “1” when the idle condition has already lasted for more than 5 ms, the ML60852A restarts oscillation immediately after this bit is set, and the remote wakeup signal is output onto the USB bus.
After transmitting the remote wakeup signal onto the USB bus, when the resume signal is received from the USB bus thereby releasing the suspend state, the ML60852A automatically resets D4 of the register SYSCON to “0”.

See Section 7.1.7.5 “Resume” in the USB Standard Rev.1.1 for details of the remote wakeup function.

6. HANDLING UNUSED PINS

In the ML60852A, there are some pins that are not used depending on the operating mode. The handling of unused pins is described below.

6.1. Bus Access Control Pins-1 (A6 to A0)

The bus access control pins-1 (A6 to A0) are not used under the following condition:

- During the Multiplex mode: A6 to A0

The handling of the pins in this case is as follows:

- A6 to A0: Fixed at the “H” or the “L” level.

6.2. Bus Access Control Pin-2 (ALE/PUCTL)

The bus access control pin-2 (ALE/PUCTL) is not used under the following condition:

- Pull-up control is not used during the Separate mode: ALE/PUCTL

The handling of the pin in this case is as follows:

- ALE/PUCTL: Left open.

6.3. DMA Transfer Control Pins (D15 to D8, –DREQ0, –DREQ1, DACK0, DACK1)

The DMA transfer control pins (D15 to D8, –DREQ0, –DREQ1, DACK0, DACK1) are respectively not used under the following conditions:

• When DMA transfer is not used:	D15 to D8, –DREQ0, –DREQ1, DACK0, DACK1
• When 8-bit DMA transfer is used:	D15 to D8
• When dual address mode is used:	DACK0, DACK1

The handling of the pins in these cases is as follows:

• D15 to D8:	Fixed at the “H” or the “L” level.
• –DREQ0, –DREQ1:	Left open.
• DACK0, DACK1:	Fixed at the “L” level.

6.4. Crystal Connection Pin (XOUT)

The XOUT pin is not used when an external 48 MHz clock is used. In this case:

- XOUT: Left open.

6.5. Test Pins (TEST1, TEST2)

The ML60852A has two test pins (TEST1 and TEST2). These pins are used for testing the LSI in the factory and are not used during normal use.

- TEST1, TEST2: Fixed at the "L" level.

Table 11 List of handling unused pins

Pin name	I/O	Condition in which the pins are not used	Pin handling method
A6 to A0	I	When the Multiplex mode is used	"H" or "L"
ALE/PUCTL	I	When the Separate mode is used and pull-up control is not used	Open
–DREQ0/–DREQ1	O	When DMA is not used	Open
DACK0/DACK1	I	When DMA is not used When the dual address mode is used	"L"
D15 to D8	I/O	When DMA is not used When 8-bit DMA is used	"H" or "L"
XOUT	O	When an external clock is used	Open
TEST1/TEST2	I	During normal use	"L"

7. DIFFERENCEES IN PIN ASSIGNMENT BETWEEN ML60851 AND ML60852A

The ML60852A provides pin-for-pin replacement with the ML60851. Pin assignments of these two devices are almost the same. However, several functions have newly been added on the ML60852A, so there are some changes.

Pin assignments of the ML60851 and ML60852A are compared below.

Table 12 Comparison of pin assignments between the ML60851 and ML60852A

Pin No.	ML60851	ML60852A	Function
1	D+	D+	USB bus
2	D–	D–	
3	V_{CC3}	V_{CC}	V_{CC}
4	TEST1	TEST1	Test pin/GND
5	TEST2	GND	
6	XIN	XIN	Crystal or ceramic resonator connection pin
7	XOUT	XOUT	
8	–CS	–CS	Chip select
9	–RD	–RD	Read
10	–WR	–WR	Write
11	–RESET	–RESET	Reset
12	–INTR	–INTR	Interrupt request
13	D15	D15	Data bus (upper)
14	D14	D14	
15	D13	D13	
16	D12	D12	
17	GND	TEST2	Test pin/GND
18	V_{CC5}	DACK1	V_{CC}/DMA acknowledge
19	D11	D11	Data bus (upper)
20	D10	D10	
21	D9	D9	
22	D8	D8	
23	ALE	ALE/PUCTL	Address latch/pull-up control
24	ADSEL	ADSEL	Address input format selection
25	A7	–DREQ1	Address input/DMA request
26	A6	A6	Address input
27	A5	A5	
28	A4	A4	
29	A3	A3	
30	A2	A2	
31	A1	A1	
32	A0	A0	
33	DACK	DACK0	DMA acknowledge
34	–DREQ	–DREQ0	–DMA request
35	AD7	AD7	Data bus (lower)/ address input (multiplexed)
36	AD6	AD6	
37	AD5	AD5	
38	AD4	AD4	
38	GND	GND	GND
40	V_{CC5}	V_{CC}	V_{CC}
41	AD3	AD3	Data bus (lower)/ address input (multiplexed)
42	AD2	AD2	
43	AD1	AD1	
44	AD0	AD0	

Handle the pins whose functions have been changed, as explained below.

7.1. Test Pin ⇔ GND (pin 5, pin 17)

For pin 5 and pin 17, the GND and TEST2 pins have been replaced with each other between the ML60851 and ML60852A. However, the TEST pin is fixed to “L” in normal use, so no change is required.

7.2. VCC5 ⇒ DACK1 Pin (pin 18)

For pin 18, V_{CC5} has been changed to the DACK1 pin. To use DMA channel 1 for ML60852A, connect the DACK1 pin to the DMA controller. If channel 1 is not to be used, no change is required. However, since the DACK1 pin is active-High after system reset, be sure to set the DMA enable bit of the DMA1CON register to "0".

7.3. ALE Pin ⇒ ALE/PUCTL Pin (pin 23)

For use in multiplexed mode (ADSEL = “H”), no change is required.
To use the pull-up control function in separate mode (ADSEL = “L”), connect the ALE/PUCTL pin to a 1.5 kΩ pull-up resistor.
Leave this pin open if the pull-up control function is not used.

7.4. A7 Pin ⇒ –DREQ1 Pin (pin 25)

This pin is an input pin on the ML60851 but it is an output pin on the ML60852A. Therefore, connect it to the DMA controller when DMA channel 1 is used.
Leave this pin open if channel 1 is not used.

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 contained herein may be reprinted or reproduced without our prior permission.