FEAL60851D-02
DATE: SEP. 3, 2001

OKI

ML60851D
Application Manual

USB device controller

Revision 2

ML60851D Application Manual

CONTENTS

1. INTRODUCTION

The ML60851D is a general-purpose device controller conforming to the Universal Serial Bus (USB) Standard Rev.1.X. 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, and interrupt transfer as the data transfer modes.

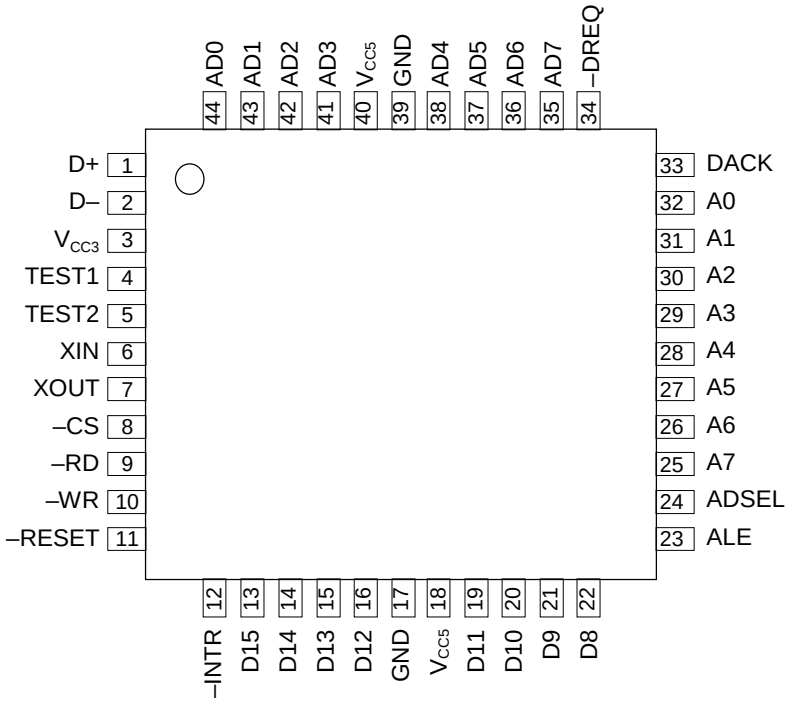
1.1. Product Features

- Conforms to USB1.1
- Supports Full-speed (12 Mbps).
- Supports three data transfer types.
  Control transfer, bulk transfer, and interrupt transfer.
- Four end points
- Built-in FIFO for data storage
- The FIFO for EP1 has a 2-layer configuration (64 Bytes x 2)
- 8 or 16 bit DMA Transfer is supported (EP1).
- Built-in USB transceiver circuit
- V_{CC3} = 3.0 to 3.6 V
  V_{CC5} = 3.0 to 5.5 V
- Built-in 48 MHz oscillator circuit
- Packages : 44-pin TQFP
  56-pin LGA

1.2. Pin Configuration and Description

1.2.1. Pin configuration

44-Pin TQFP (Top View)



56-Pin LGA (Transparent View)

NC	D8	D9	D11	GND	D12	D14	–INTR	NC	J
ALE	NC	D10	V_{CC5}	NC	D13	D15	NC	–RESET	H
A7	ADSEL						–RD	–WR	G
A5	A6						–XOUT	–CS	F
A4	NC						NC	–XIN	E
A2	A3						TEST1	TEST2	D
A0	A1						D–	V_{CC3}	C
DACK	NC	AD7	AD5	NC	V_{CC5}	AD2	NC	D+	B
NC	–DREQ	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.
A7 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	(Note 1)	Interrupt request signal output pin.
–DREQ	O	(Note 1)	DMA Request signal output pin.
DACK	I	(Note 2)	DMA Acknowledge signal input pin.
ALE	I	Positive logic	Address latch enable signal input pin.
ADSEL	I		Address input format signal input pin.
–RESET	I	Negative logic	Reset signal input pin.

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

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

1.3. Example of External Connections

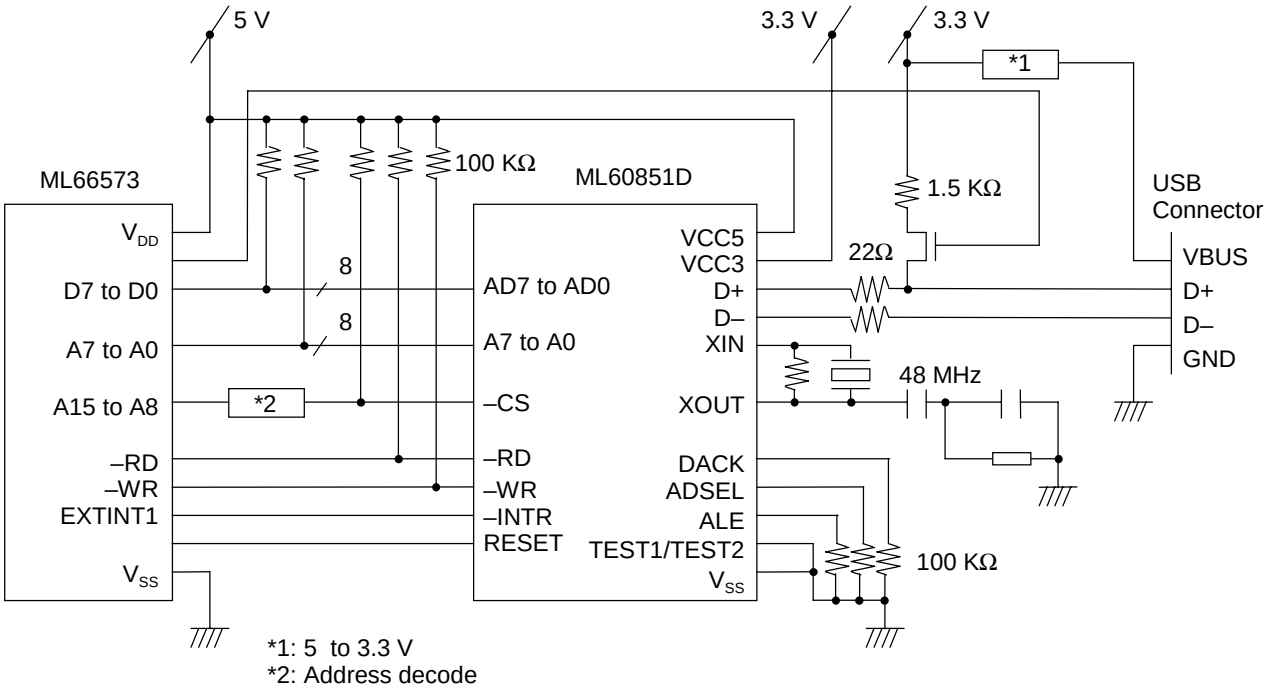


Figure 1 Example of connections between OKI's 16-bit microcontroller MSM66573 and ML60851D

2. EXAMPLES OF USB TRANSFER PROCEDURE

This chapter gives a description of a sample procedure for carrying out USB communication using the ML60851D.

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 ML60851D

Carry out the settings of the operating conditions of the ML60851D 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. Outline flow of 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. ML60851D has a built-in 8-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

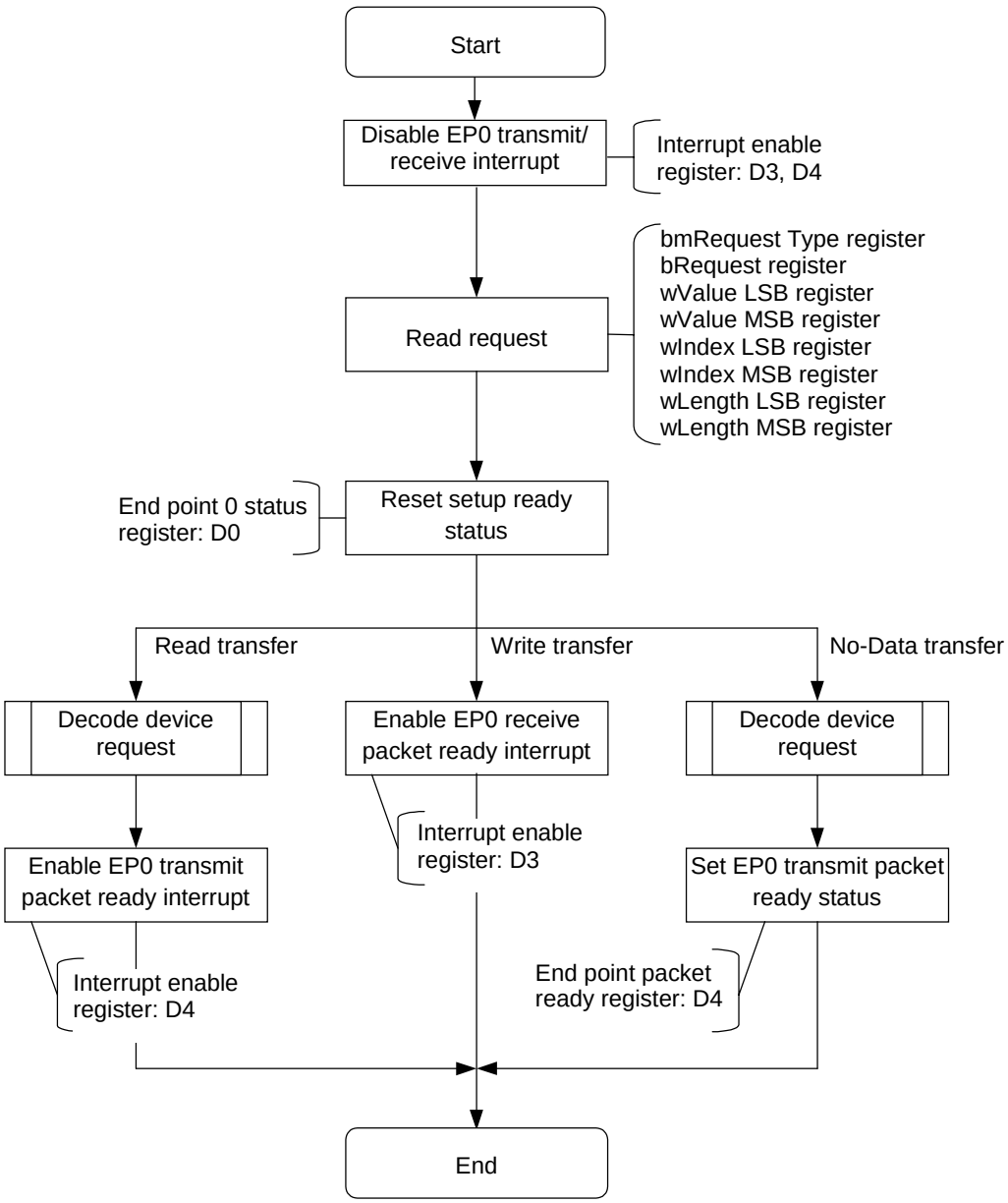
Function	Register name	Bits
Device request	bRequest Setup	D7 to D0
Device request	wValue LSB Setup	D7 to D0
Device request	wValue MSB Setup	D7 to D0
Device request	bmRequest Type Setup	D7 to D0
Device request	wIndex LSB Setup	D7 to D0
Device request	wIndex MSB Setup	D7 to D0
Device request	wLength LSB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

Table 2 Registers used in the data stage or status stage

Function	Register name	Bits
Packet ready	PKTRDY	D0/D4
Packet ready interrupt status	INTSTAT	D3/D4
Packet ready interrupt enable	INTENBL	D3/D4
End point control	EP0RXCON/EP0TXCON	D0/D2 to D6
Data sequence toggle	EP0RXTGL/EP0TXTGL	D0
Maximum packet size	EP0RXPLD/EP0TXPLD	D7 to D0
Status	EP0STAT	D4 to D2
Receive byte count	EP0RXCNT	D6 to D0
FIFO Status	FIFOSTAT1/ FIFOSTAT2	D1 to D0
Transmit/receive FIFO	EP0RXFIFO/EP0TXFIFO	D7 to D0

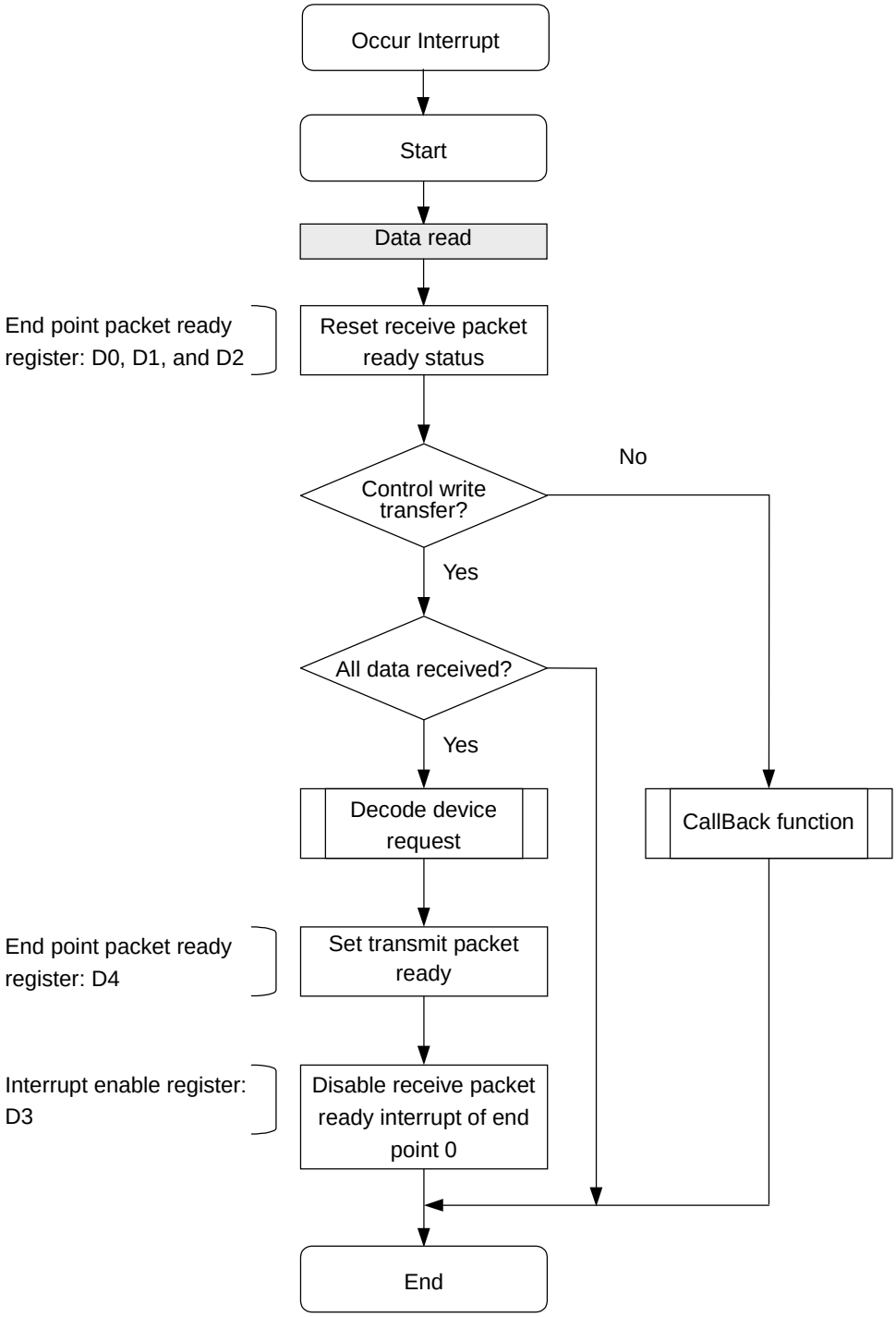
2.2.2. Set up ready interrupt procedure

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 detected satisfactorily by ML60851D). 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.



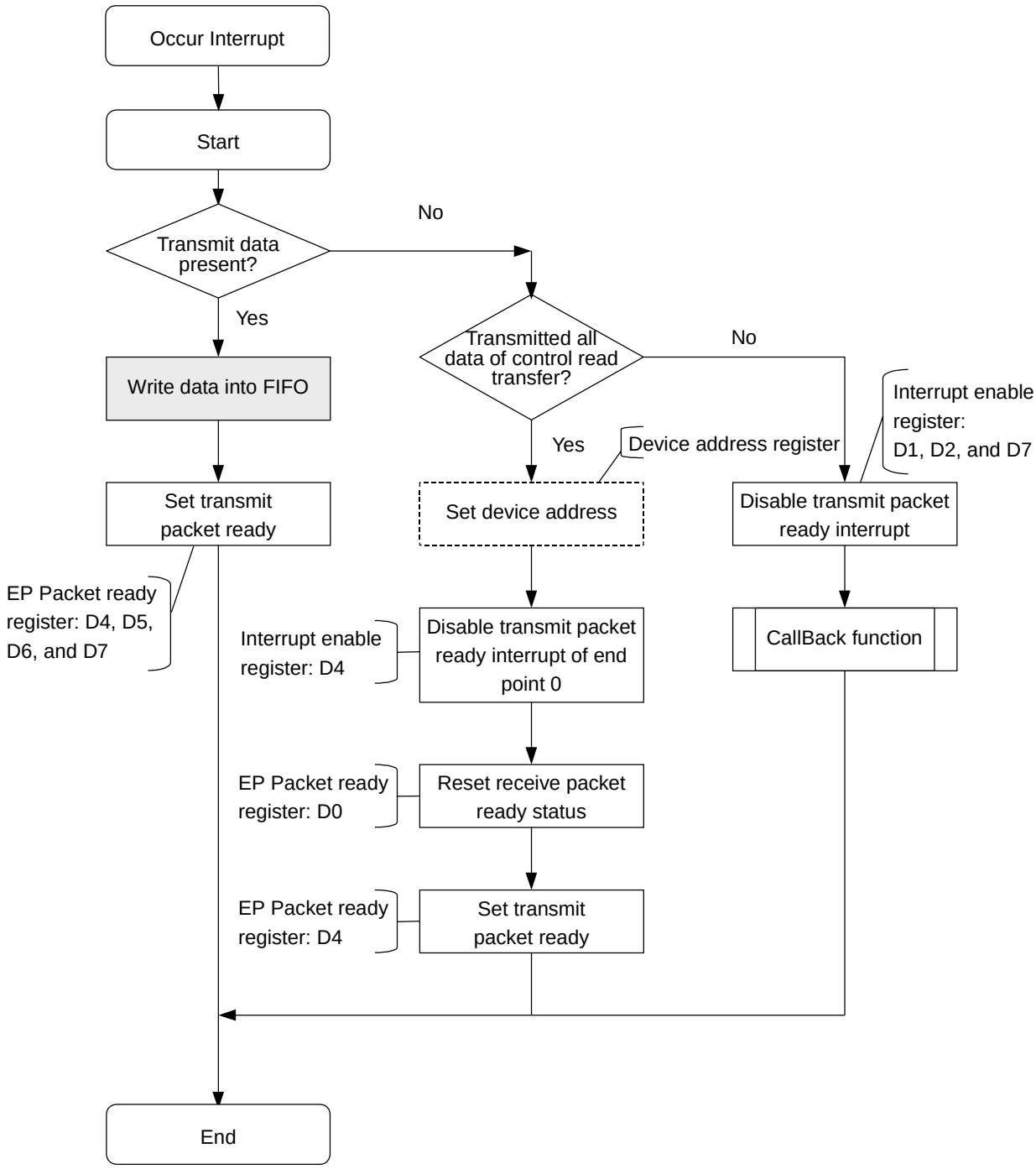
2.2.3. Reading received data in receive packet ready interrupt procedure

The following flow chart illustrates the data reception process from the point of view of the application firmware controlling the ML60851D. 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 Receive FIFO of ML60851D. Hence, ML60851D generates an interrupt cause. A typical processing procedure of the received data 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.



2.2.4. Writing transmit data in transmit packet ready interrupt procedure

The following flow chart illustrates typical procedures for transmitting data from ML60851D 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 transmissions is done through a Control Read transfer which is a message pipe (structured pipe) and hence much more tedious than other types of transfers.



2.3. Bulk Transfer

2.3.1. Outline of bulk transfer

Bulk transfer is used when transmitting or receiving data whose integrity 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 ML60851D, EP1 and EP2 are provided as the end points for bulk transfer. These end points can be allocated individually for both bulk-in and bulk-out. The FIFO size is 64 bytes for both EP1 and EP2. EP1 of the ML60851D has the DMA transfer function. In addition, the FIFO of EP1 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.

The registers used during bulk transfer are the following:

Table 3 Registers used when allocated to EP1

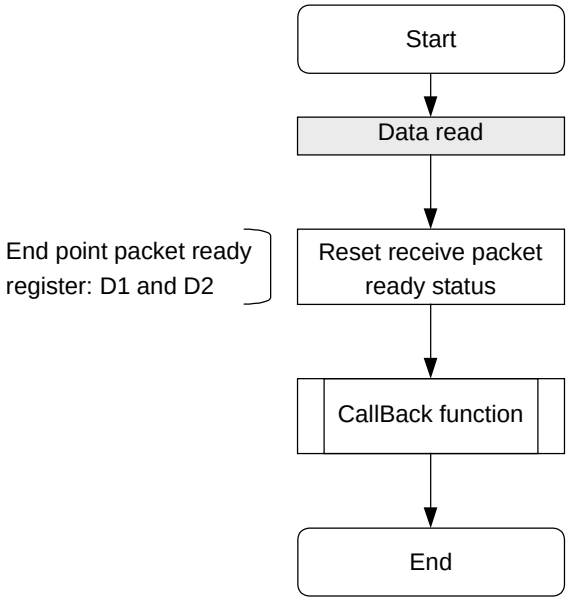
Function	Register name	Bits
Packet ready	PKTRDY	D1/D5
Packet ready interrupt status	INTSTAT	D1
Packet ready interrupt enable	INTENBL	D1
End point control	EP1CON	D7 to D0
Data sequence toggle	EP1TGL	D0
Maximum packet size	EP1PLD	D6 to D0
Receive byte count	EP1RXCNT	D6 to D0
FIFO Status	FIFOSTAT1	D3 to D2
Transmit/receive FIFO	EP1RXFIFO/EP1TXFIFO	D7 to D0

Table 4 Registers used when allocated to EP2

Function	Register name	Bits
Packet ready	PKTRDY	D2/D6
Packet ready interrupt status	INTSTAT	D2
Packet ready interrupt enable	INTENBL	D2
End point control	EP2CON	D7 to D0
Data sequence toggle	EP2TGL	D0
Maximum payload size	EP2PLD	D6 to D0
Receive byte count	EP2RXCNT	D6 to D0
FIFO Status	FIFOSTAT2	D3 to D2
Transmit/receive FIFO	EP2RXFIFO/EP2TXFIFO	D7 to D0

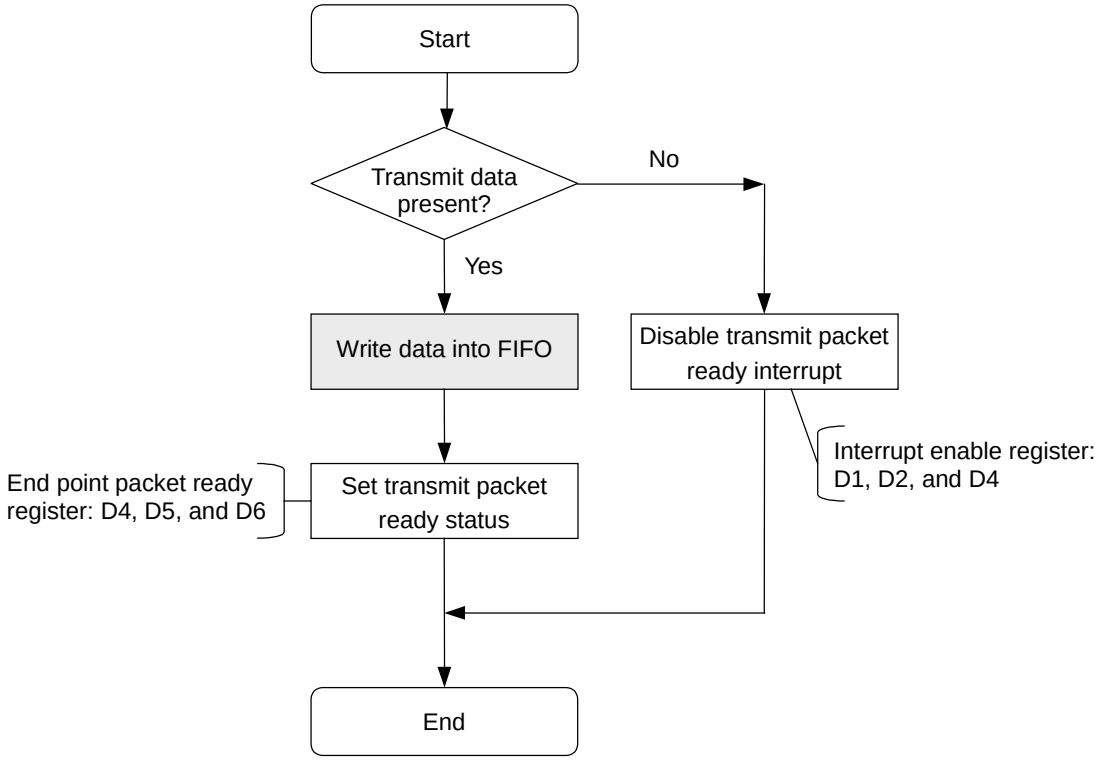
2.3.2. Packet ready interrupt procedure in Bulk-Out transfer

The following flow chart illustrates the outline of a typical Bulk-Out transfer from the point of view of the application firmware controlling the ML60851D. In ML60851D, endpoints 1 and 2 are the endpoints available for Bulk-Out transfers. 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 ML60851D's end point FIFO.



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 ML60851D. In ML60851D, end points 1, 2 and 3 can be used for bulk out transfers. 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. Although end point 3 is identified as an interrupt endpoint on this device, given the similarities between Bulk-In and interrupt transfer structures, the processing flow outlined below can be used for both kinds of transfers.



2.4. 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 ML60851D, it is possible to carry out interrupt transfers using EP3. In addition, only interrupt-in transfer (from device to host) is supported by the ML60851D. The interrupt-out transfer (from host to device) is not supported.

The registers used during interrupt transfer are the following:

Table 5 Registers used during interrupt transfer

Function	Register name	Bits
Packet ready	PKTRDY	D7
Packet ready interrupt status	INTSTAT	D7
Packet ready interrupt enable	INTENBL	D7
End point control	EP3CON	D7 to D0
Data sequence toggle	EP3TGL	D0
FIFO Status	FIFOSTAT2	D5 to D4
Transmit/receive FIFO	EP3TXFIFO	D7 to D0

3. EXTERNAL INTERFACE

The functions and the method of use of the pins for connecting a microcontroller or an external peripheral device to the ML60851D are described in this chapter.

3.1. Bus Interface

3.1.1. ADSEL Pin

The ML60851D supports Separate and Multiplex methods of accessing data and address buses. Which of these methods is used is determined by the ADSEL pin.
The Separate method is used when ADSEL is "L". In this case, address lines are connected to A7 to A0 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. The pins A7 to A0 are not used in the Multiplex method of accessing. Connect these pins A7 to A0 either to V_{DD} or GND (see Chapter 6. "HANDLING UNUSED PINS").

3.1.2. ALE Pin

This pin is used in the Multiplex method (when ADSEL is "H").
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 is invalid. In this case, tie this pin to V_{DD} or GND (see Chapter 6. "HANDLING UNUSED PINS").

3.2. DMA Interface

The ML60851D supports DMA transfer. DMA Transfer is possible only when EP1 is being used and is impossible when other end points are being used.
The pins –DREQ and DACK are used for carrying out DMA transfer.
The settings related to DMA transfer are made by the registers POLSEL, DMACON, and DMAINTVL.

3.2.1. Selection of –DREQ and DACK polarities

It is possible to select the polarities of –DREQ and DACK individually. The polarity of –DREQ is set by bit D1 of the register POLSEL and that of DACK is set by bit D2 of the same register.
When D2 is "0", DACK becomes active-High, and becomes active-Low when D2 is "1".
When D1 is "0", –DREQ becomes active-Low, and becomes active-High when D1 is "1".
After a hardware reset or a software reset, –DREQ will be active-Low and DACK will be active-High.

3.2.2. –DREQ Active conditions

3.2.2.1. During transmission

The ML60851D makes –DREQ 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 ML60851D makes –DREQ inactive.
Subsequently, when all the data written into the FIFO has been transmitted and an ACK handshake is received from the host computer, the ML60851D resets again the transmit packet ready bit to "0" and makes –DREQ active.

3.2.2.2. During reception

The ML60851D sets the receive packet ready bit and makes –DREQ 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.
The receive packet ready bit is reset (a "1" is written) and –DREQ is made inactive by the ML60851D after all the received data is read out.
When the next data packet is received normally again, the ML60851D sets again the receive packet ready bit to "1" and makes –DREQ active.

3.2.3. DMA Enable

The control of enabling and disabling DMA transfer is made using bit D0 of the register DMACON.
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 –DREQ becomes active when the conditions for generating a request are satisfied (see Section 3.2.2 "–DREQ Active conditions").

3.2.4. Address modes during DMA transfer

The ML60851D supports two address modes.
When the bit D1 of the register DMACON is "0", the ML60851D operates in the single address mode, and operates in the dual address mode when this bit is "1".
In the single address mode, the DMA controller specifies only the memory address and the I/O device is selected by DACK.
In the dual address mode, both the memory address and the I/O device address are specified by the DMA controller. DACK is not used in the dual address mode (see Chapter 6. "HANDLING UNUSED PINS" for the handling of DACK during the dual address mode).

3.2.5. Transfer modes during DMA transfer

The ML60851D supports the single transfer and demand transfer modes. The transfer mode is determined by using the bit D4 of the register DMACON.
This bit D4 is set to "0" when using the single transfer mode during DMA transfer and to "1" when using the demand transfer mode.
During DMA transfer, –DREQ is made active when it becomes possible to read or write a packet of data. In the demand transfer mode, –DREQ is made inactive by the DMA controller when all the receive or transmit data has been transferred. 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.

3.2.6. Byte count data insertion

The ML60851D 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 bit D2 of the register DMACON to "1" when wanting to insert the byte count data. The byte count data is not inserted when this bit D2 is "0".
During the 16-bit transfer mode, the upper byte of the byte count data (the leading word) will become 00h.

3.2.7. Transfer data width during DMA transfer

The ML60851D supports 8-bit DMA transfer and 16-bit DMA transfer. This transfer data width is selected using bit D3 of the register DMACON.
Reset bit D3 to "0" for 8-bit DMA transfer and set this 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.

3.2.8. Interrupting DMA transfer

In the ML60851D, it is possible to interrupt DMA transfer in the middle. Interrupting and restarting DMA transfer is controlled using the bit D7 of the register DMACON.
Set this bit D7 to "1" in order to temporarily halt DMA transfer in the middle. When D7 is set to "1", –DREQ becomes inactive and the DMA transfer is interrupted. When D7 is reset to "0", –DREQ becomes active again and the DMA transfer is restarted.
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.

3.2.9. −DREQ Signal interval during single transfer

In the ML60851D, it is possible to set the interval during the single transfer mode, that is, the interval until –DREQ is made active again after the completion of DMA transfer of the previous byte (or word). This interval setting is made using the register DMAINTVL.
Interval duration = (DREQ Enable time) + 84(1-bit time) x n (ns)
The value of the parameter "n" of the above equation is set in the register DMAINTVL.
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 arrives) and should not be modified thereafter.

3.2.10. Setting packet ready during DMA transfer

When writing data into the FIFO of the ML60851D using DMA transfer (bulk-in transfer), there is no need to set the EP1 packet ready bit from the local MCU side. When data with the maximum packet size is written in the FIFO by the DMA controller, the ML60851D automatically sets the packet ready bit. Therefore, there will be no need for the local MCU to access the ML60851D 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 EP1 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 ML60851D has an interrupt function. The ML60851D 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 bit D0 of the POLSEL register.
When D0 is "0", –INTR is active-Low.
When D0 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 ML60851D 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 ML60851D interrupt enable register.
Next, clear the enable register (all bits are made "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.

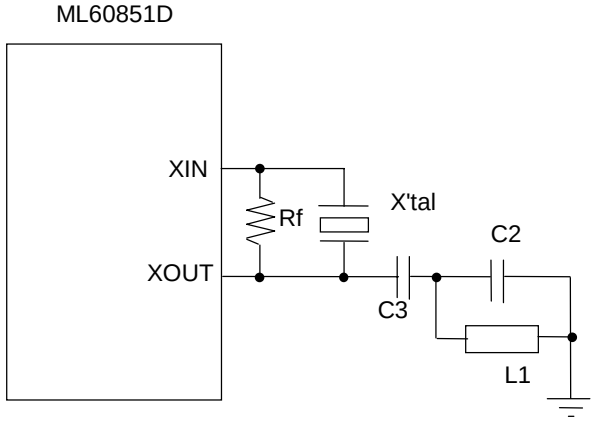
3.4. Oscillator Circuit

3.4.1. Sample configuration of oscillator circuit

The ML60851D has a built-in 48 MHz oscillator circuit and it is possible to obtain the clock signal necessary for the operation of the ML60851D by merely connecting an external crystal.
The clock oscillation accuracy must be within 48 MHz ± 0.25%.

3.4.1.1. When a crystal is used.

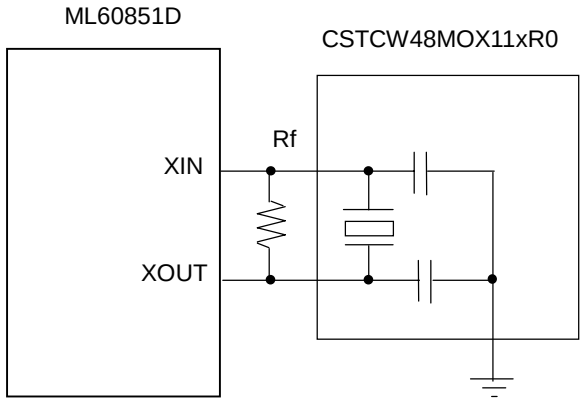
A sample circuit configuration of the oscillator circuit using a crystal is shown in the figure below.



Crystal: Model HC-49U manufactured by Kinseki
C2 = 5 pF C3 = 1000 pF Rf = 1 MΩ L1 = 2.2 µH

Figure 2 Example of the oscillator circuit using a crystal.

3.4.1.2 When a ceramic resonator is used



Ceramic resonator: Model CSTCW48MOX11xR0 (built-in capacitor type) manufactured by Murata FMG.
Rf = 1 MΩ

Figure 3 Example of the oscillator circuit using a ceramic resonator.

3.4.2. Supplying an external clock signal

When using the ML60851D, it is possible to supply externally a 48 MHz clock signal without using the built-in oscillator circuit. Please take the following precautions when using an external clock.

- 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 oscillator circuit of the ML60851D operates from the power supplied to V_{CC3} Therefore, supply a 3 V clock signal. Never input a 5 V clock signal.
- Make sure that the external clock signal accuracy is within 48 MHz ± 0.25% as when using the built-in oscillator circuit.

3.4.3. Stopping the oscillations

The ML60851D has the function of stopping the oscillations.
When the bits D7 to D4 of the system control register SYSCON are set to "1010", the oscillations of the ML60851D will be stopped and the LSI goes into the standby state.
In order to restart the oscillations, it is necessary to carry out either a hardware reset (by making –RESET active) or a software reset (by setting the bit D0 of the SYSCON register to "1").
The oscillations will not be restarted even if a value other than "1010" is written in the bits D7 to D4 of SYSCON.

3.5. USB Interface

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

The ML60851D 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 in below.

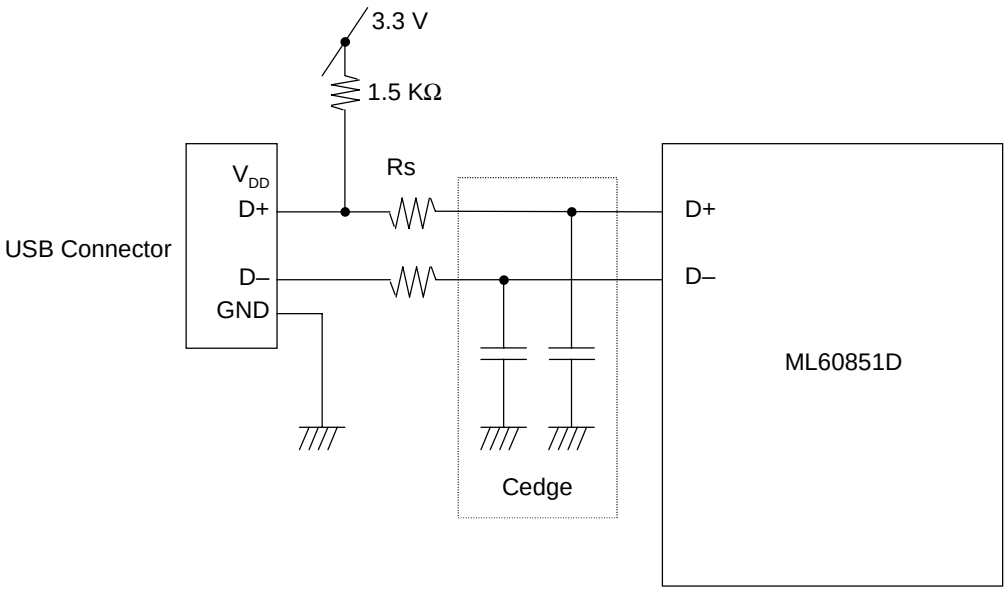


Figure 4 A sample circuit configuration of the USB bus interface.

- 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 of the D+ line should be connected on the USB connector side of the series resistor Rs.
- It is recommended that the pull-up voltage of 3.3 V for 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.
- Make the distance between the ML60851D and the USB connector as short as possible. Also, it is recommended that the connections between the ML6081C and the USB connector are made on the same printed circuit board.

4. INTERRUPTS

There are eight types of interrupt causes in the ML60851D. The interrupt causes are assigned to bits of the register INTSTAT, and the corresponding bit becomes "1" when an interrupt cause is generated provided that the corresponding interrupt has been enabled in the INTENBL register. The local MCU can identify the source of the interrupt by referring to the interrupt status register (INTSTAT).
It is possible to enable or disable each interrupt using the register INTENBL. The bits of the register INTENBL correspond to the respective bits of the register INTSTAT. 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 register INTSTAT will always be "0" when the corresponding bits of the INTENBL register is "0" (disable).

Table 6 Interrupt causes (INTSTAT register)

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

Each of the interrupt causes are described in detail below.

4.1. Setup Ready Interrupt

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 ML60851D, the ML60851D sets a "1" in bit D0 (Setup ready) of the register EP0STAT.

At this time, if bit D0 of the register INTENBL has already been set to "1", bit D0 of the register INTSTAT will be set to "1" at the same time as the bit D0 of the register EP0STAT is set to "1", and –INTR becomes active.

After reading all the setup data, writing a "1" into bit D0 of the register EP0STAT will reset (to "0") the EP0 setup ready status and hence remove the setup-ready interrupt cause. This in turn will cause bit D0 of the register INTSTAT to be reset to "0" and as a result, -INTR becomes inactive.
The setup ready interrupt is automatically 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 Interrupt

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 ML60851D sets to "1" the receive packet ready bit of the end point packet ready register (PKTRDY) corresponding to that end point.
At this time, if this interrupt has already been enabled by setting the corresponding bit of the register INTENBL to "1", the corresponding status bit of the register INTSTAT will be set to "1" at the same time as the receive packet ready bit is set to "1" (carried out automatically by ML60851D), and –INTR becomes active.
After reading all the data in the receive FIFO, when the MCU writes a "1" into the receive packet ready status bit thereby resetting the packet ready status, the corresponding bit of the register INTSTAT is also reset to "0" and –INTR becomes inactive.
The ML60851D automatically recognizes the end of a packet when EOP (end of packet) is received in the case of both full packets (packets containing data of the maximum packet size) and short packets.
The mutual correspondences between the end points (EP), the packet ready bits, and interrupts are shown below.

Table 7 Correspondence between endpoints, packet ready bits, and bits of the interrupt register (during reception).

EP	PKTRDY	INTENBL (INTSTAT)
0	D0	D3
1	D1	D1
2	D2	D2

4.3. Transmit Packet Ready Interrupt

If an endpoint has been configured for transmission, when the corresponding packet ready interrupt enable bit of the INTENBL register is set (write "1"), an interrupt cause is generated. Hence the corresponding bit of the INTSTAT register is automatically set to "1" and an -INTR is generated. The cause of the interrupt is packet ready status bit (in register PKTRDY) 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 (in PKTRDY register) of the end point. This action will remove the interrupt cause and hence deassert the interrupt. At this point, ML60851D 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 a 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 ML60851D to automatically reset to "0" the corresponding transmit packet ready bit of the PKTRDY 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 INTENBL register.

A short packet (shorter than the maximum payload size) can be transmitted by writing a shorter than maximum packet size data into the FIFO of ML60851D and setting (write "1") the transmit packet ready status bit (in PKTRDY register) of the endpoint.

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

Table 8 Correspondence between endpoints, packet ready bits, and bits of the interrupt register (during transmission).

EP	PKTRDY	INTENBL (INTSTAT)
0	D4	D4
1	D5	D1
2	D6	D2
3	D7	D7

4.4. USB Bus Reset Interrupt

A bus reset is caused when a SE0 state continues for 2.5 μs or more on the USB bus.
The USB bus reset interrupt is enabled when a "1" is set in bit D5 of the register INTENBL.
In this case, when the ML60851D detects an SE0 state for 2.5 μs or more, it sets bit D5 of the INTSTAT register to "1" and makes –INTR active.
When the MCU writes a "1" in bit D5 (clear USB bus reset status) of the device state register (DVSTAT), bit D5 of the register INTSTAT is reset to "0" and –INTR becomes inactive.

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

4.5. Suspend State Interrupt

When ML60851D detects that no bus activity is present (no start-of-frame) for more than 3 ms, it recognizes the bus is in idol state and hence, ML60851D enters the Suspend State. The suspend state interrupt is enabled if bit D6 of the register INTENBL is set to "1". At this time, if the ML60851D detects the idle state for 3 ms or more, it sets to "1" bit D6 of the register INTSTAT and makes –INTR active.
When the ML60851D detects the resume state on the USB bus or a USB bus reset, it resets to "0" bit D6 of the register INTSTAT and makes –INTR inactive.

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

5. OTHER FUNCTIONS

The other functions of the ML60851D are described in this chapter.

5.1. System Reset

The ML60851D 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 ML60851D.
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 bit D0 of the system control register (SYSCON). The hardware reset and the software reset are functionally identical.

When the oscillations have been stopped, it is possible to restart the oscillations by carrying out a hardware reset or a software reset. (See Section 3.4.3 "Stopping the oscillations".)

5.2. Suspend Function

The ML60851D goes into the suspend state when the idle state in which the USB bus does not have any activity continues for 3 ms or more.
If bit D6 of the register INTENBL has already been set to "1", when the ML60851D detects an idle state for longer than 3 ms, bit D6 of the register INTSTAT is set to "1" and –INTR becomes active. (See Chapter 4. "INTERRUPTS".)

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

5.3. Detection of Resume State

The ML60851D cannot detect the resume state (the restart signal) in the condition in which the oscillations have been stopped. Therefore, do not stop the oscillations in the suspend state.

See Section 7.1.7.5 "Resume" of the USB Standard Rev.1.1 for details of the resume function.

5.4. Remote Wakeup

The ML60851D supports the remote wakeup function.
The remote wakeup function can only be used in the suspend state.
Bit D4 of the device state register (DVCSTAT) is set to "1" in order to use the remote wakeup function.
If bit D4 of the register DVCSTAT is set to "1" when the time elapsed after the USB bus has entered the idle state is less than 5 ms, the ML60851D waits until the idle state continues for 5 ms or more and then outputs the remote wakeup signal onto the USB bus. If bit D4 of the DVCSTAT register is set to "1" when the idle condition has already lasted for more than 5 ms, the remote wakeup signal is output onto the USB bus immediately after this bit is set. 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 ML60851D automatically resets bit D4 of the register DVCSTAT to "0".
The remote wakeup function cannot be used in the condition in which the oscillations have been stopped.

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

6. HANDLING UNUSED PINS

In the ML60851D, 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 (A7 to A0, ALE)

The bus access control pins (A7 to A0, ALE) are respectively not used under the following conditions:

- During the Multiplex mode : A7 to A0
- During the Separate mode : ALE

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

- A7 to A0 : Fixed at the "H" or the "L" level.
- ALE : Fixed at the "H" or the "L" level.

6.2. DMA Transfer Control Pins (D15 to D8, DREQ, –DACK)

The DMA transfer control pins (D15 to D8, DREQ, –DACK) are respectively not used under the following conditions:

- When DMA transfer is not used : D15 to D8, DREQ, –DACK
- When 8-bit DMA transfer is used : D15 to D8
- When dual address mode is used : –DACK

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

- D15 to D8 : Fixed at the "H" or the "L" level.
- DREQ : Left open.
- –DACK : Fixed at the "L" level.

6.3. Crystal Connection Pin (XOUT)

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

- XOUT: Left open.

6.4. Test Pins (TEST1, TEST2)

The ML60851D 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 9 List of handling unused pins.

Pin name	Pin No.	I/O	Condition in which the pins are not used	Pin handling method
A7 to A0		I	When the Multiplex mode is used	"H" or "L"
ALE	23	I	When the Separate mode is used	"H" or "L"
–DRE0	34	O	When DMA is not used	Open
DACK	33	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	7	O	When an external 48 MHz clock is used	Open
TEST1/TEST2	4/5	I	During normal use	"L"

7. REVISION HISTORY

- Ver. 0.50 was prepared.
- Ver. 0.51
  1: Damper resistors were added in the D+/D- lines as shown in the Section 1.3 “Example of External Connections”.
  2: The flow of control transfer and bulk transfer was corrected.
  3: The model number of the ceramic resonator in Figure 3 was corrected to CSTCW48MOX11xR0
  4: Misprints were corrected.

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.