MCS7860

USB 2.0 Link Controller with PHY

Features

- USB host to host communication
- High-speed data transfer rate.
- Single cable solution for data communication
- Plug & Play and hot pluggable
- No external power needed
- Windows 98 *SE*, 2000 and XP.
- Easy drag-n-drop file transfer
- Includes print function for remote files
- Low power consumption. Includes power management.
- Compliant with USB specification 2.0
- Integrated USB2.0 PHYs.

Applications

- PC-to-PC File transfer
- Link type Cable

Application notes

- AN-7860

Evaluation Board

- MCS7860-EVB

General Description

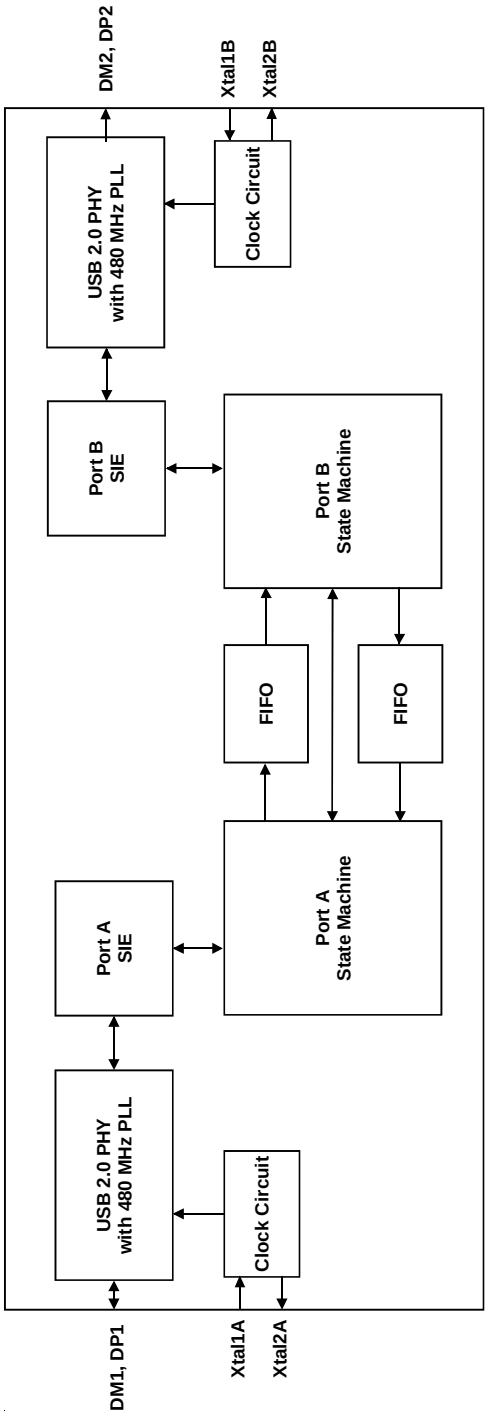
This device connects two PCs through the USB. It's based on USB2.0 and offers much higher bandwidth compared to existing PC to PC communication devices. This device comes in as a great help for home networking because LAN is a kind of overkill for a couple of PCs one might have at home. It's a simple flexible medium for sharing large files between laptops and the other stationary servers at work place. This device installs itself as a network card and connects the PCs as though they are connected through a NIC based LAN, only difference is that the medium here is USB and not the ETHERNET. This device is not just a file sharing device; rather it is a complete network solution where one can execute network applications.

Ordering Information

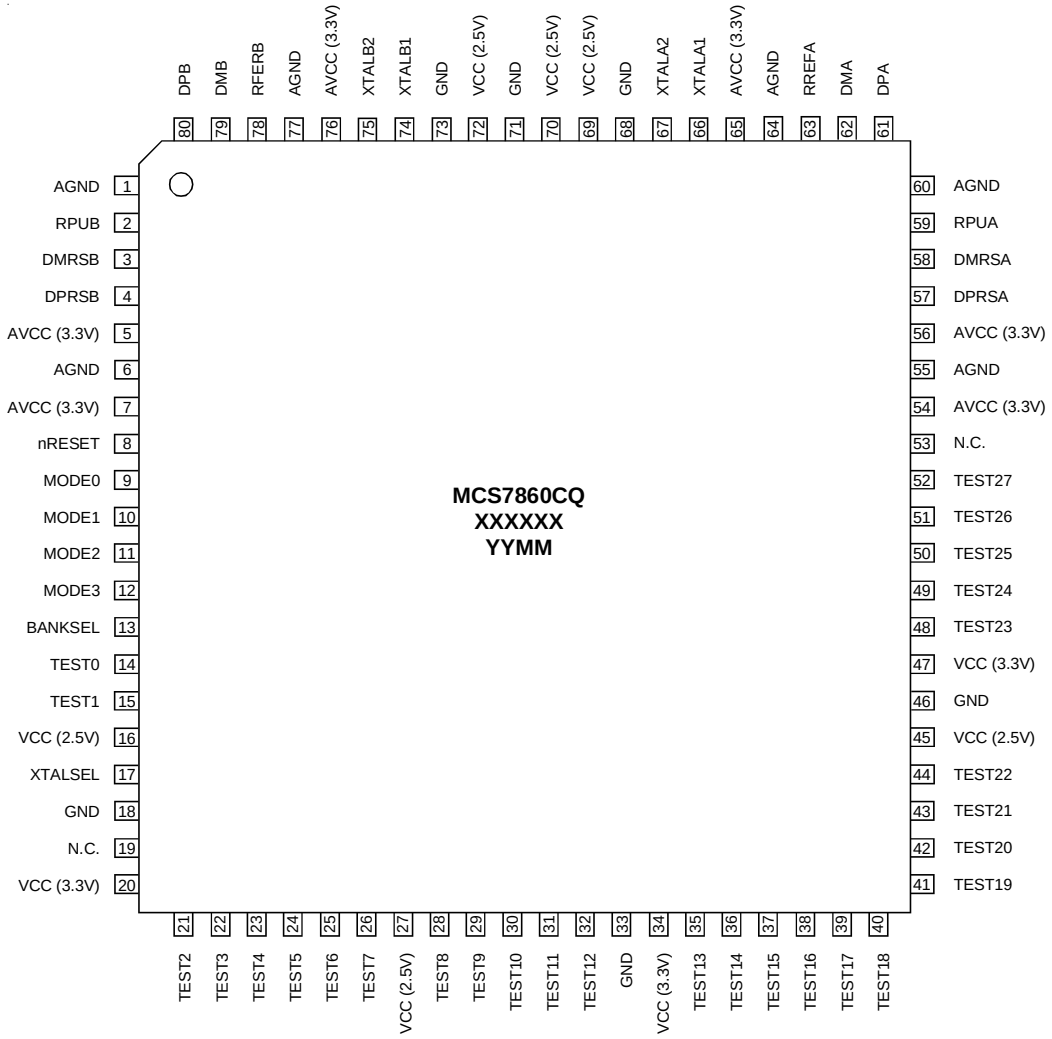
Commercial Grade		
MCS7860CQ	80-LQFP	0° C to +70° C

MosChip Semiconductor ♦ 3335 Kifer Rd, Santa Clara, CA 95051 ♦ Tel (408) 737-7141 ♦ Fax (408) 737-7708

MCS7860 Block Diagram



80-Pin QFP Package



Pin Name	80	Type	Description
DPA	61	I/O	USB2.0 data pin Data positive pin terminal.
DMA	62	I/O	USB2.0 data pin Data negative pin terminal.
RREFA	63	I	Internal voltage reference input. Requires external resistor to GND, (12.1k-ohm +/- 0.1 %).
XTAL2A	67	O	Crystal Oscillator Output (12/30 MHz). See XTALSEL pin description.
XTAL1A	66	I	Crystal Oscillator Input (12/30 MHz).
DPRSA	58	I/O	USB1.1 data in Data positive pin terminal. Connects to DPA via external series resistor (39 ohm +/- 0.1%).
DMRSA	57	I/O	USB1.1 data in Data negative pin terminal. Connects to DMA via external series resistor (39 ohm +/- 0.1%).
RPUA	59	I	Internal reference current input. Requires external resistor (1.5 k-ohm +/- 0.1%) to AVCC (Analog 3.3V Power).
nRESET	8	I	Hardware Reset to the Chip (pull-down). Active low signal. Requires external capacitor (0.1uF) to 3.3V Power Supply.
BANKSEL	13	I	Address bank select (pull-down). This is reflected in the INTIN transfers for assigning the MAC address. "0"-> Software selects "A" pair of addresses. "1"-> Software selects "B" pair of addresses.
MODE0	9	I	Mode selection bit (pull-up). One of the four mode pins used for selecting the device operation mode.
MODE1	10	I	Mode selection bit (pull-up). One of the four mode pins used for selecting the device operation mode.
MODE2	11	I	Mode selection bit (pull-up). One of the four mode pins used for selecting the device operation mode.
MODE3	12	I	Mode selection bit (pull-up). One of the four mode pins used for selecting the device operation mode.
DPRSB	4	I/O	USB1.1 data in Data positive pin terminal, connects to DPB via external series resistor (39 ohm +/- 0.1%).
DMRSB	3	I/O	USB1.1 data in Data negative pin terminal, connects to DMB via external series resistor (39 ohm +/- 0.1%).

Pin Name	80	Type	Description
RPUB	2	I	Internal reference current input. Requires external resistor (1.5 k-ohm +/- 0.1%) to AVCC (Analog 3.3V Power).
DPB	80	I/O	USB2.0 data pin Data positive pin terminal.
DMB	79	I/O	USB2.0 data pin Data negative pin terminal.
RREFB	78	I	Internal voltage reference input. Requires external resistor to GND, (12.1k-ohm +/- 0.1 %).
XTAL2B	75	O	Crystal Oscillator Output (12/30 MHz). See XTALSEL pin description.
XTAL1B	74	I	Crystal Oscillator Input (12/30 MHz).
XTALSEL	17	I	Crystal frequency select pin (pull-up). When it is connected to 3.3V Power Supply or left open 30 MHz external crystal or clock is required to function. a Low on this pin will operate with external 12 MHz crystal or clock.
AGND		Pwr	Analog Ground: 1, 6, 55, 60, 64, 77
AVCC (3.3V)		Pwr	3.3V Analog power supply. 5, 7, 54, 56, 65, 76
GND		Pwr	Digital Ground. 18, 33, 46, 68, 71, 73
VCC (2.5V)		Pwr	2.5V Digital power supply. 16, 27, 45, 69, 70, 72
VCC (3.3V)		Pwr	Digital Power Supply (3.3V). 20, 34, 47
TEST		I/O	Following pins are used for functional testing and should be left open. 14-15, 21-26, 28-32, 35-44, 48-53

DEVICE OPERATION
This device consists of three sections, two USB device controllers along with transceivers and a 'bridge section' with internal buffers.

The USB device controller supports four endpoints (control, bulk_in, bulk_out and interrupt_in end points). The 'bridge' has the following data-flow paths.

Data flow from USB A to USB B or USB B to USB A. The data flow can also be simultanious. The USB A to USB B path uses bulk_out endpoint of USB A to send data from USB A device controller to the USB B device controller through the 'bridge'.

Similarly The USB B to USB A path uses bulk_out endpoint of USB B to send data from USB B device controller to the USB A device controller through the 'bridge'.

There are two sets of 16-bit interrupt registers. They reflect current status of the device and the transaction. These bits are set/cleared through vendor specific commands. The current status is reported to the corresponding USB host through interrupt endpoint.

Control path used for the configuration of the device and for processing the vendor specific commands.

Interface
The device-controller works on UTMI based interface to the USB 2.0 PHY on the host side. The application side of the device-controller has got a generic interface consisting of data-bus, the endpoint_select and read_valid, write_valid and error signals.

Functions
Functionally the device-controller has 10 parts.

1. Mode Selector
Performs the functions of selecting speed and implements USB defined test modes. Performs the suspend/resume operations.

2. Serial Interface Engine.
SIE implements the first stage decoding of the received packets. It checks for the address and valid endpoint numbers. It also checks for the PID errors and CRC errors. It then enables appropriate endpoint to process the further stages of the transfer. It also performs the functions required for packet transmits.

3. Standard Command Processor.
It processes the standard USB commands. Decodes the commands and generates the control signals to descriptors' block device_state_machine and endpoints.

4. Device State Machine.

5. Descriptors Block.
It consists of registers set that stores descriptor-database. It also has the logic to read /write these registers.

6. Control Endpoint.
Implements the logic for flow of control commands.

7. Bulk_out Endpoint.
Implements the functions to control data flow from USB-HOST to the USB-device.

8. Bulk_in Endpoint.
Implements the functions to control data flow from USB-device to the USB-host.

9. Interrupt Endpoint.
The USB-host accesses this endpoint periodically to read the status of the device. This endpoint implements the functions to control the status flow from USB-device to the USB-host.

10. Advanced Command Processor.
It processes the vendor specific commands.

DEVICE DESCRIPTORS

Offset	Field	Size	Value	Description
0	Length	1	12	Device descriptor size
1	DescriptorType	1	01	DEVICE
2	USB	2	0110	USB Release Number for FS
			0200	USB Release Number for HS
4	DeviceClass	1	FF	Class code
5	DeviceSubclass	1	00	Subclass code
6	DeviceProtocol	1	FF	Protocol code
7	MaxPacketSize(0)	1	40	Max packet size for endpoint 0
8	Vendor ID	2	9710	Vendor ID
10	Product ID	2	7860	Product ID
12	Device	2	0100	Device Release Number
14	Manufacturer	1	00	Index of string descriptor for the manufacturer
15	Product	1	00	Index of string descriptor for the Product
16	SerialNumber	1	00	Index of string descriptor for the Serial Number
17	NumConfigurations	1	01	Number of Possible Configurations

CONFIGURATION DESCRIPTOR 0

Offset	Field	Size	Value	Description
0	Length	1	09	Descriptor size in bytes
1	DescriptorType	1	02	CONFIGURATION
2	TotalLength	2	0027	Size of all data returned for this configuration in bytes
4	NumInterfaces	1	01	Number of interfaces the configuration supports
5	ConfigurationValue	1	01	Identifier for Set_Configuration and Get_Configuration
6	Configuration	1	00	Index of string descriptor for the configuration
7	Attributes	1	A0	Self/Bus power and remote wakeup settings
8	MaxPower	1	32	Bus power required

INTERFACE DESCRIPTOR 0

Offset	Field	Size	Value	Description
0	Length	1	09	Descriptor size in bytes
1	DescriptorType	1	04	INTERFACE
2	InterfaceNumber	1	00	Number identifying this
3	AlternateSetting	1	00	Value used to select an alternate setting
4	NumEndPoints	1	03	Number of endpoints supported, except endpoint 0
5	InterfaceClass	1	FF	Class code
6	InterfaceSubclass	1	00	Subclass code
7	InterfaceProtocol	1	FF	Protocol code
8	Interface	1	00	Index of string descriptor for the interface

ENDPOINT DESCRIPTOR 1

Offset	Field	Size	Value	Description
0	Length	1	07	Descriptor size in bytes
1	DescriptorType	1	05	ENDPOINT
2	EndpointAddress	1	81	Endpoint Number and direction
3	Attributes	1	02	Transfer Type supported
4	MaxPacketSize	2	0040	Maximum packet size supported for FSpeed
			0200	Maximum packet size supported for HSpeed
5	Interval	1	FF	Polling Interval

ENDPOINT DESCRIPTOR 2

Offset	Field	Size	Value	Description
0	Length	1	07	Descriptor size in bytes
1	DescriptorType	1	05	ENDPOINT
2	EndpointAddress	1	02	Endpoint Number and direction
3	Attributes	1	02	Transfer Type supported
4	MaxPacketSize	2	0040	Maximum packet size supported for Fspeed
			0200	Maximum packet size supported for Hspeed
5	Interval	1	FF	Polling Interval

ENDPOINT DESCRIPTOR 3

Offset	Field	Size	Value	Description
0	Length	1	07	Descriptor size in bytes
17	DescriptorType	1	05	ENDPOINT
2	EndpointAddress	1	83	Endpoint Number and direction
3	Attributes	1	03	Transfer Type supported
4	MaxPacketSize	2	0002	Maximum packet size supported
5	Interval	1	1	Polling Interval

Functionality

When the peer is ready for the normal data transfers, after the enumeration process, it sets its P_En bit.

The Bridge is ready for data transfers only when both of the P_En bits are set.

The P_En and P_En_Nl are reported during the INTIN transactions. Along with this bits the information regarding the RES_Out, RES_In, Memory and data availability in the bridge, BANKSEL are reported during interrupt in transactions by the host.

All the status bits are reported through interrupt transactions. Normal data transfers are through bulk_in and bulk_out endpoints.

Data Flow

WRSM

It interfaces the USB device controller's bulk out endpoint to the bridge. It receives data from USB with a maximum packet size of 512 bytes in High speed (64 bytes in Full speed), per transaction and writes to the SRAM. It works under the control of bulk_out_endpoint of the usb_device_controller. WRSM keeps track of the write pointer, generates write addresses and other signals for write operation to the memory. It also updates the ready for write flags for each of the banks.
It generates the busy / ready signal to the bulk_out endpoint so as to enable the usb_device_controller to generate appropriate ACK/NAK/NYET handshake to the host.

If the bulk_out_endpoint reports packet error, the WRSM flushes the memory banks by moving write_pointers back to their initial value (initial value of the write_pointer is the value in this register before the start of this transfer).

Configuration Function

HS on both Ends	Reads packets from the banks considering the max_packet_size to be 512 bytes.
FS on both ends	Reads packets from the banks considering max_packet_size to be 64 bytes.
HS write and FS read	High speed device considers 512 as the max packet size. Full Speed end reads packets considering max_packet_size to be 64 bytes. The conditions than need to be considered are as follows. The WRSM writes max 512 bytes (or <512) and the FS read controller read in chunks of 64 bytes (max) until it encounters a partial packet or zero lenght packet.
FS write and HS read	FS device writes 8 packets of 64 bytes length. Switches for every 8 packets. Stops after writing the zero length packet. HS read- controller reads 1 packet of 512 bytes (<512 bytes) length. Stops after reading the zero length packet.

RDSM

It interfaces USB device controller's bulk_in endpoint to the bridge. It reads from SRAM and sends data to USB with a maximum packet size of 512 bytes in High speed (64 bytes in Full speed), per transaction. It works under the control of bulk_in_endpoint of the usb_device_controller. RDSM keeps track of the read pointer, generates read addresses and other signals for read operation to the memory. At the end of read, it asserts set_ready for write 0.

Before start of any transfer, the RDSM reads control information from the last location(Location 257) of the FIFO. It gets the information about packet length and packet type from this read.
It generates the data_ready signal to the bulk_in endpoint so as to enable the usb_device_controller to generate appropriate handshake to the host.

If the bulk_in_endpoint reports packet error, the RDSM moves the read_pointers back to their initial value (initial value of the read_pointer is the value in this register before the start of this transfer). This enables retry of packet-reads.

RequestType	Request	Value	Index	Length
0100_0000	0000_1101	Feature Selector	0000	0000

WRSM

It interfaces the USB device controller's bulk out endpoint to the bridge. It receives data from USB with a maximum packet size of 512 bytes in High speed (64 bytes in Full speed), per transaction and writes to the SRAM. It works under the control of bulk_out_endpoint of the usb_device_controller. The bridge memory segment has got two banks of memory for data flow in each direction. The WRSM keeps track of the bank to be written.

It keeps track of the write pointer, generates write addresses and other signals for write operation to the memory. It also updates the ready for write flags for each of the banks.

It generates the busy/ready signal to the bulk_out_endpoint so as to enable the usb_device_controller to generate appropriate ACK/NAK/NYET handshake to the host.

If the bulk_out_endpoint reports packet error, the WRSM flushes the memory banks by moving write_pointers back to their initial value (initial value of the write_pointer is the value in this register before the start of this transfer).

At the end of a transfer, control information is loaded into the last location (Location 257) of the FIFO.

bmRequestType	bRequest	wValue	wIndex	wLength
0100_0000	0000_1101	Feature Selector	0000	0000

Commands:

This device supports nine standard USB commands and three two vendor specific commands. They are as follows.

USB Standard Commands:

1. SET ADDRESS.
2. CLEAR FEATURE.
3. GET STATUS.
4. GET CONFIGURATION.
5. SET CONFIGURATION.
6. GET DESCRIPTOR.
7. SET FEATURE.
8. GET INTERFACE.
9. SET INTERFACE.

The format of the standard commands is as per the USB Specification.

Vendor Specific Commands:

Set_Link_Feature.

Vendor specific command to set the bits in the interrupt register.

Value field is the value of the feature to be set. wLength and wIndex should be "0". If the wValue field is non zero, the value and the feature selector specified in the wValue field are invalid and the device responds with a "STALL" handshake.

Value	Feature
1	P_En
2	T_Request
3	T_Com
4	RES_In
5	RES_Out

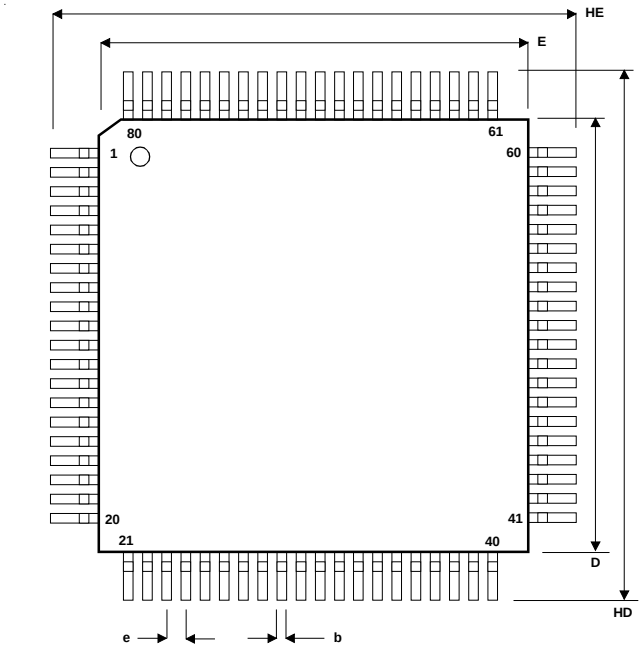
Clear_Link_Feature

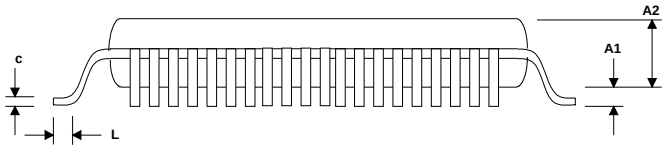
Vendor specific command to clear the interrupt register bits.

bmRequestType	bRequest	wValue	wIndex	wLength
0100_0000	0000_1110	Feature Selector	0000	0000

Value field is the value of the feature to be set. Length and wIndex should be “0”. If the Value field is non zero, value and the feature selector specified in the Value are invalid and the device responds with a “STALL” handshake.

Packaging Information





SYMBOL	MILLIMETERS		INCHES	
	MIN	MAX	MIN	MAX
A1	0.05	0.14	0.002	0.006
A2	1.35	1.45	0.053	0.057
b	0.13	0.23	0.005	0.009
c	0.09	0.20	0.004	0.008
e	0.40 TYP		0.016 TYP	
L	0.45	0.75	0.018	0.030
HD	12		0.472	
D	10		0.393	
HE	12		0.472	
E	10		0.393	