MAN-000670-000
Revision 1.8
March 12, 1999

USB0670/673

670: 5V PCI-USB ASIC
673: 3.3V PCI-USB ASIC

CMD Technology, Inc.
19 Morgan
Irvine, California 92618
(949) 454-0800
www.cmd.com

Revision History

Revision	Issue Date	Comments
0.1	01/05/96	Preliminary draft
0.2	01/12/96	Added Chapters 6, 7, 8
0.3	01/15/96	Corrections to all chapters
0.4	01/17/96	Corrections to all chapters. Added Chapter 9.
0.5	01/23/96	Changed identification of pin 25 to VSS/TEST1 in Chapter 2.
0.5A	3/8/96	Removed description of legacy support and inserted reference to OpenHCI Legacy Support Interface Specification v. 0.8.
0.6	3/15/96	Added low-speed source electrical characteristics in Chapter 6. Specified default values for Command Register and Device Status. Changed offsets for legacy support registers, and bit definitions for HcRevision Register. Deleted references to Em_Event bit. Corrected DC Characteristics in Chapter 8.
0.7	4/16/96	Changed the descriptions for pins 1, 54, 73, and 97 in response to the release of OpenHCI Legacy Support Interface Specification v. 1.01.
0.8	7/25/96	Added information for USB0673 chip
1.0	9/18/96	Minor corrections for first release
1.1	11/15/96	
1.2	1/8/97	Corrected typographical error in USB0673 pinout diagram (Figure 2-2). Added Appendices A and B.
1.3	2/5/97	Added Appendix C on bit stuffing.
1.4	4/24/97	Changed Figure 7-1 to remove input to pin 16. Revised Appendix A.
1.5	10/22/97	In section 8.2, changed minimum value for Input Voltage High (VIH) from 2.2 to 2.0.
1.6	2/11/98	Changed text in section 7.1.4; added sections 7.2 and 8.4 and figure 7-2.
1.7	3/23/98	Changed signal on pin 92 from VDD to UHCI on both 670 and 673 pinout diagrams. Inserted pin name UHCI for pin 92 in Table 2-2: removed pin name VDD for pin 92 in same table. Added information for signal name UHCI to Table 3-3.
1.8	1/8/99	Made numerous changes to text to better reflect performance and function of the new version of the silicon. Updated USB Specification revision level to 1.1.

Copyright

Portions of this document have been quoted from the following specifications:
Universal Serial Bus Specification, revision 1.1, published by Compaq Computer Corporation, Digital Equipment Corporation, IBM PC Company, Intel Corporation, Microsoft Corporation, NEC, Northern Telecom.
OpenHCI: Open Host Controller Interface Specification for USB, revision 1.0, published by Compaq Computer Corporation, Microsoft Corporation, and National Semiconductor Corporation.
PCI Local Bus Specification, revision 2.1, published by Intel Corporation.

Trademarks

PowerPC is a registered trademark of Motorola Corporation. Alpha is a registered trademark of Digital Equipment Corporation. MIPS is a registered trademark of NEC Corporation. UNIX is a registered trademark of UNIX System Laboratories. Compaq is a registered trademark of Compaq Computer Corporation. Pentium is a registered trademark of Intel Corporation. All product names are trademarks, registered trademarks, or servicemarks of their respective owners.

Notice

CMD Technology reserves the right to make changes, without notice, in the product described herein in order to improve design and/or performance. CMD assumes no responsibility or liability for the use of any of this product, conveys no license or title under any patent, copyright, or mask work right to this product, and makes no representations or warranties that this product is free from patent, copyright, or mask work right infringement, unless otherwise specified. Applications that are described herein are for illustrative purposes only. CMD makes no representation or warranty that such applications will be suitable for the specified purpose without further testing or modification.

CMD Technology products are not designed for use in life support devices, or devices where malfunction of a CMD semiconductor can reasonably be expected to result in a personal injury. Customers using or selling CMD semiconductors for use in such applications do so at their own risk and agree to fully indemnify CMD Technology for any damages resulting from such improper use or sale.

This page left blank intentionally.

Table of Contents

This page left blank intentionally.

1 Introduction

NOTE
The 5.0V USB0670 and the 3.3V USB0673 chips are identical in function and nearly identical in design. Therefore, the bulk of the information in this manual applies to both chips. Differences between the two chips will be made clear throughout the document with explicit references to the USB0670 or USB0673. In the absence of an explicit reference or the presence of the generic name USB067x, you may assume that the text applies to both chips.

1.1 Features

- Bridges PCI bus and the Universal Serial Bus (USB)
- USB device bandwidth of up to 12 Mb/s
- Optimized logic design for highest bus bandwidth
- Wide range of packet sizes for maximum range of device buffering options
- Full support of real time dynamic insertion and removal of devices
- Multiple transfers of data and message streams between host and devices
- Isochronous, bulk, interrupt, and control transfer types over the same wires
- 32-bit PCI local bus interface
- PCI Bus Master
- Supports up to 127 devices (concurrent operation)
- Complies with OpenHCI 1.0, USB 1.1, PCI local Bus 2.1 specifications (USB0673B) and PCI local Bus 2.2 specifications (USB0670B)
- Designed for PCI-based 486, Pentium, or P6 motherboards
- Compatible with other leading platforms, including PowerPC, Alpha, MIPS, and virtually all Unix RISC-based computing platforms.
- Supports all USB-compliant peripherals (e.g. keyboard, mouse, monitor, telephone, joystick, etc.)
- Integrates seamlessly with CMD TECHNOLOGY'S device-side ASICs and other manufacturers' USB hub-based products.
- The 5.0V USB0670 comes in a 100-pin PQFP. The 3.3V USB0673 comes in a 100-pin TQFP.
- Power requirements:
  USB0670-5.0V DC @ 0.10 amps peak
  USB0673-3.3V DC @ 0.06 amps peak
- Supports legacy keyboards and mouse devices.

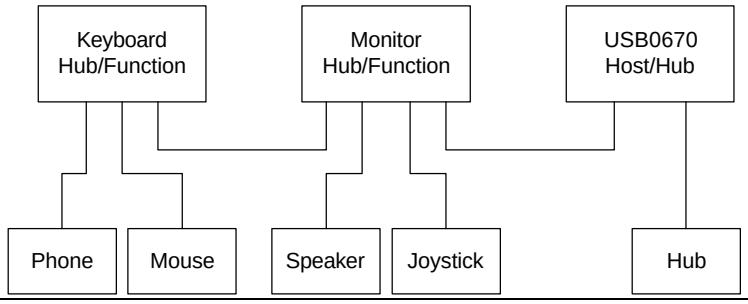


Figure 1-1: USB desktop environment example

1.2 Description

The USB067x is perfectly suited for USB device bandwidths ranging from a few kilobits per second to 12 megabits per second (12 Mb/s). Both the 5.0 volt and 3.3 volt versions support isochronous, bulk, interrupt, and control transfer types over the same set of wires.

A multiple-connections architecture supports simultaneous operation of many USB devices (up to 127 physical USB devices) while maintaining top speeds. The chip supports the transfer of multiple data and message streams between the host and devices, and provides compound device support (i.e., peripherals composed of many functions). Most importantly, the chip is designed with optimized logic, resulting in the highest possible bus utilization.

Applications such as telephony and audio are guaranteed bandwidth and low latencies. Isochronous workload may use most of the bus bandwidth, thus maximizing performance and efficiency.

The USB067x's wide range of packet sizes allows the maximum range of device buffering options. This wide range of device data rates is accomplished by accommodating packet buffer size and latencies. Flow control for buffer handling is built into the protocol architecture, as well as an error handling/fault recovery mechanism.

Devices may be inserted on and removed from the USB dynamically. The USB067x will identify device changes in real time as perceived by the user. There will be a full range of device drivers running on most operating systems to support a variety of USB devices.

Designed in full compliance with relevant industry standards, the USB067x supports all compliant device-side USB products with low-cost, off-the-shelf cables and connectors. This assures affordable connection of the widest range of USB devices.

The architecture of the USB067x may be upgraded to support multiple CMD USB controllers. CMD also provides an extensive upgrade and expansion path to our complete family of tightly coupled USB device-side ASIC solutions and USB Hub products.

The USB0670 is a 5.0V PCI bus master chip with Universal Serial Bus (USB) functionality, available in 100-pin PQFP or TQFP packages. The USB0673 3.3V PCI bus master chip offers similar functionality in a 100-pin TQFP package. Both chips comply with the OpenHCI 1.0 and USB 1.1 specifications, and integrate easily into motherboard designs as PCI devices.

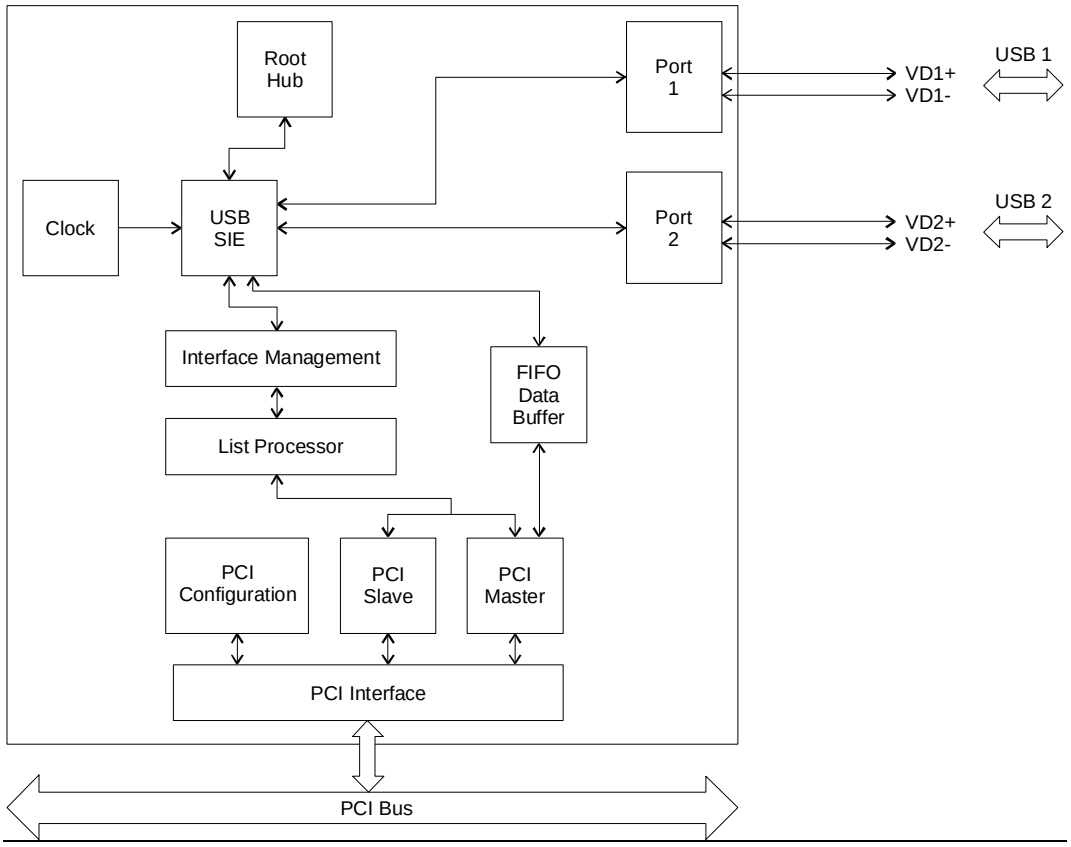


Figure 1-2: Block diagram

1.3 Applicable Documents

1 *Universal Serial Bus (USB) Specification*, revision 1.1
2 *Open HCI Specification*, revision 1.0
3 *PCI Specifications*, revision 2.1 (USB0673B) and revision 2.2 (USB0670B)
4 *Open HCI Legacy Keyboard Support Specification*, revision 1.01
5 MIL-STD-217F
6 ISO-9000

This page left blank intentionally.

2 Pin Assignments

The USB0670 comes in both 100-pin PQFP and TQFP packages. The PQFP is available only for the USB0670. Figure 2-1 shows the pinout for the PQFP. The pins are listed alphabetically in Table 2-2.

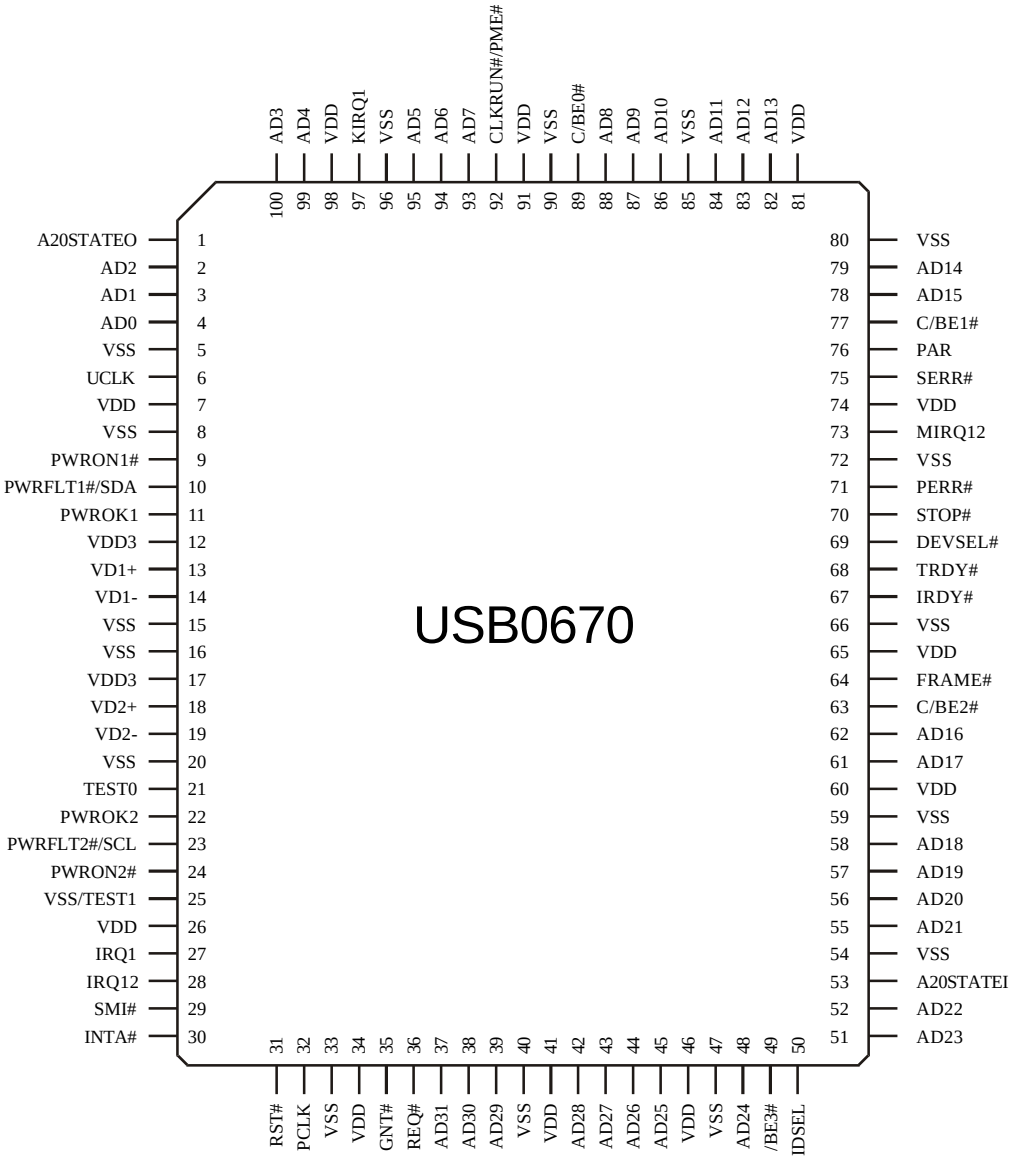


Figure 2-1: PQFP Pinout Diagram

The USB0673 is only available in the 100-pin TQFP package. The TQFP pinout is shown in Figure 2-2. The pins are listed alphabetically in Table 2-2.

NOTE
USB0670 chips using the TQFP package will be labeled “USB0670”

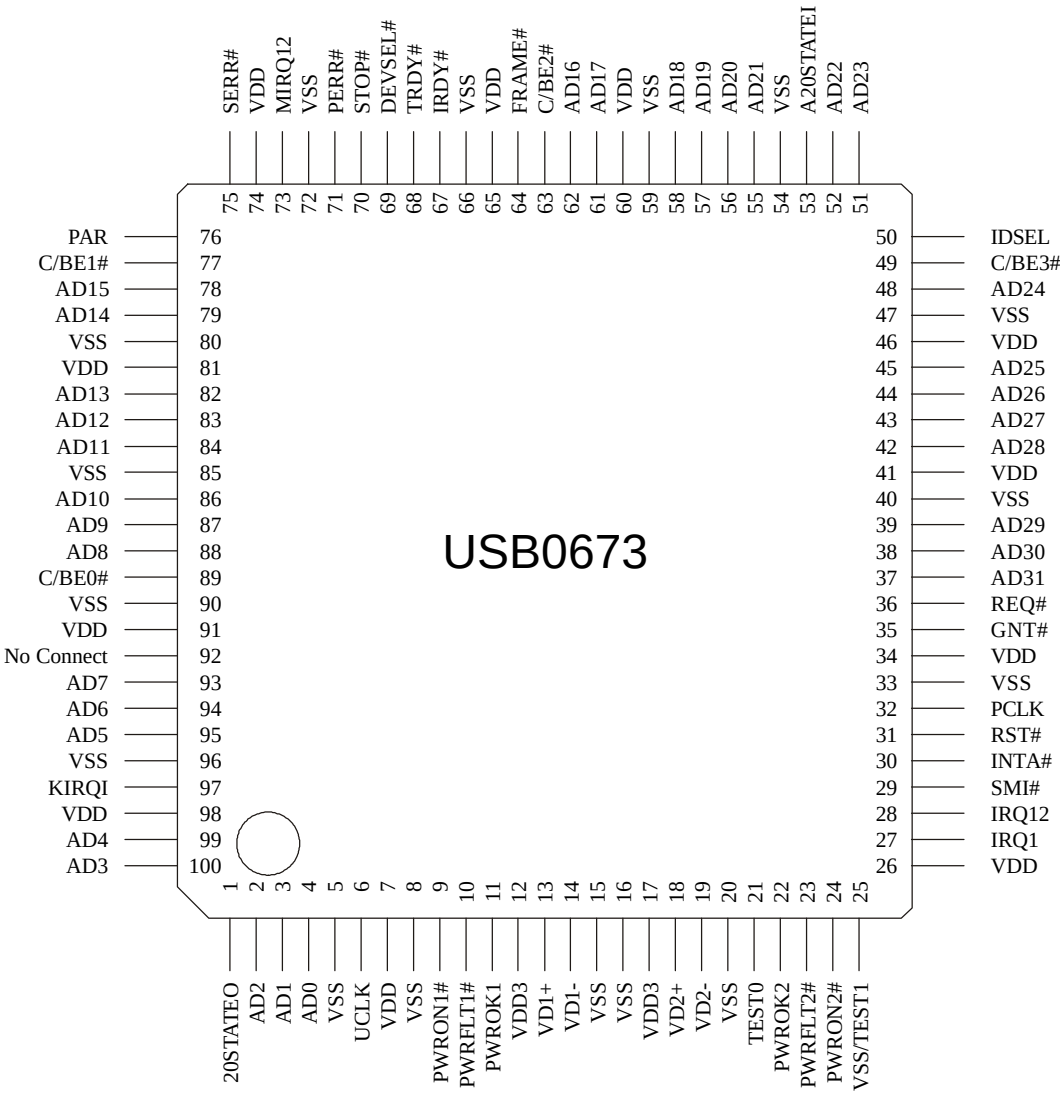


Figure 2-2: TQFP Pinout Diagram

Table 2-1 provides a legend for the signal types found in Table 2-2.

Table 2-1: Signal Types

Signal Type	Description
I	Input: a standard, input-only signal
O	Totem Pole Output: a standard active driver
OD	Open Drain Output
I/O	Input/Output: a bidirectional, tri-state pin
s/t/s	Sustained Tri-State: an active low tri-state signal that is owned and driven by one and only one agent at a time. When an agent drives an s/t/s signal low, it must drive it high for at least one clock before permitting it to float. Once an agent tri-states an s/t/s signal, a new owner must wait at least one clock before it can begin driving the signal. A pull-up sustains the inactive state until another agent drives it and is provided by the central resource.
t/s/o	Tri-State Output
V	Power supply connection (VDD, VDD3, or VSS)

Table 2-2: Alphabetical Pin Listing

Pin Name	Pin #	Type
A20_STATEI	53	I
A20_STATEO	1	OD
AD0	4	I/O
AD1	3	I/O
AD10	86	I/O
AD11	84	I/O
AD12	83	I/O
AD13	82	I/O
AD14	79	I/O
AD15	78	I/O
AD16	62	I/O
AD17	61	I/O
AD18	58	I/O
AD19	57	I/O
AD2	2	I/O
AD20	56	I/O
AD21	55	I/O
AD22	52	I/O
AD23	51	I/O
AD24	48	I/O
AD25	45	I/O
AD26	44	I/O

Pin Name	Pin #	Type
AD27	43	I/O
AD28	42	I/O
AD29	39	I/O
AD3	100	I/O
AD30	38	I/O
AD31	37	I/O
AD4	99	I/O
AD5	95	I/O
AD6	94	I/O
AD7	93	I/O
AD8	88	I/O
AD9	87	I/O
C/BE0#	89	I/O
C/BE1#	77	I/O
C/BE2#	63	I/O
C/BE3#	49	I/O
DEVSEL#	69	I/O (s/t/s)
FRAME#	64	I/O (s/t/s)
GNT#	35	I
IDSEL	50	I
INTA#	30	OD
IRDY#	67	I/O (s/t/s)
IRQ1	27	t/s/o
IRQ12	28	t/s/o
KIRQ1I	97	I
MIRQ12I	73	I
PAR	76	I/O
PCLK	32	I
PERR#	71	I/O (s/t/s)
PWRFLT1#/SDA	10	I/OD (Note 1)
PWRFLT2#/SCL	23	OD (Note 1)
PWROK1	11	I
PWROK2	22	I
PWRON1#	9	I/O
PWRON2#	24	I/O
REQ#	36	t/s
RST#	31	I
SERR#	75	OD

Pin Name	Pin #	Type
SMI#	29	OD
STOP#	70	I/O (s/t/s)
TEST0	21	I
TEST1	25	I
TRDY#	68	I/O (s/t/s)
UCLK	6	I
670: CLKRUN#/PME# 673: No connect	92 92	I/O (Note 1)
VD1-	14	I/O
VD1+	13	I/O
VD2-	19	I/O
VD2+	18	I/O
VDD	7	V
VDD	26	V
VDD	34	V
VDD	41	V
VDD	46	V
VDD	60	V
VDD	65	V
VDD	74	V
VDD	81	V
VDD	91	V
VDD	98	V
VDD3	12	V
VDD3	17	V
VSS	5	V
VSS	8	V
VSS	15	V
VSS	16	V
VSS	20	V
VSS	33	V
VSS	40	V
VSS	47	V
VSS	54	V
VSS	59	V
VSS	66	V
VSS	72	V
VSS	80	V

Pin Name	Pin #	Type
VSS	85	V
VSS	90	V
VSS	96	V

NOTES
1) The SDA, SCL, CLKRUN# and PME# signals are available only with the USB670B to meet USB 1.1 and PCI 2.2.

3 Signal Descriptions

The # suffix indicates that the signal's active, or asserted state occurs when the signal is at a low voltage level. The absence of a # suffix means that the signal is asserted when at the high voltage level.

The term "assert" is used to describe the act of making a signal active, regardless of whether it is active at a low or high voltage level. To "negate" (deassert) a signal is to make it inactive.

3.1 PCI Bus Interface

Table 3-1: PCI Bus Interface Signals

Signal Name	Type	Description
PCLK	I	PCI CLOCK: The timing for all transactions on the PCI bus is based on PCLK. All other PCI signals are sampled on the rising edge of PCLK and timing parameters refer to this edge. The USB067x supports frequencies from 25 to 33 MHz.
RST#	I	RESET: RST# causes the USB067x to enter its default state. The following signals are tri-stated from the leading edge of RST# and remain so until driven by the USB067x as either a master or slave: FRAME#, IRDY#, TRDY#, STOP#, and DEVSEL#. The assertion of RST# set all registers to their default values. RST# may be asynchronous to PCLK when asserted or negated, but negation must occur with a clean, bounce-free edge. RST# must be asserted for more than 1 µs.
AD[31:0]	I/O	PCI ADDRESS/DATA: These signals constitute a multiplexed address and data bus. During the initial clock of a transaction, AD[31:0] contain a 32-bit physical byte address. After the first clock of the transaction, these lines contain data. Bus transactions have two phases: an address phase, followed by one or more data phases. AD[7:0] comprise the least significant byte (LSB). AD[31:24] comprise the most significant byte (MSB). AD[31:0] serve as inputs during the address phase when the USB067x is the target. For the one or more data phases that follow, the USB067x may supply data on AD[31:0] in the case of a read or accept data in the case of a write. When it acts as master, the USB067x drives a valid address on AD[31:2] during the address phase, and drives write or accepts read data on AD[31:0] during the data phase. The USB067x always drives AD[1:0] low as master.

Signal Name	Type	Description
C/BE[3:0]#	I/O	BUS COMMAND AND BYTE ENABLES: The same PCI pins carry the command and byte enable signals. In the address phase, C/BE[3:0]# identify the bus command. C/BE[3:0]# serve as Byte Enables during the data phase and flag which byte lanes carry meaningful data. C/BE[0] corresponds to byte 0, C/BE[1] to byte 1, C/BE[2] to byte 2, and C/BE[3] to byte 3. When the USB067x is the initiator of a PCI bus cycle, it drives C/BE[3:0]#. When it is the target, it samples C/BE[3:0]#.
FRAME#	I/O (s/t/s)	CYCLE FRAME: The current master drives FRAME# to indicate the beginning and duration of an access. The master asserts FRAME# at the beginning of a bus transaction, sustains the assertion during the data transfers, and then negates FRAME# in the final data phase. FRAME# is an input to the USB067x when the USB067x is the target and an output when the USB067x is the initiator. FRAME# is tri-stated from the leading edge of RST# and remains tri-stated until driven by the USB067x as either a master or slave.
TRDY#	I/O (s/t/s)	TARGET READY: TRDY#, in conjunction with IRDY#, indicates whether the USB067x is able to complete the current data phase of the transaction. TRDY# and IRDY# are both asserted when a data phase is completed. During a read, the USB067x, acting as the target, asserts TRDY# to indicate that it has placed valid data on AD[31:0]. During a write, the USB067x, again acting as a target, asserts TRDY# to indicate that is prepared to accept data. TRDY# is an input to the USB067x when the USB067x is the initiator and an output when the USB067x is a target. TRDY# is tri-stated from the leading edge of RST# and remains so until driven by the USB067x as either a master or a slave.
IRDY#	I/O (s/t/s)	INITIATOR READY:IRDY#, in conjunction with TRDY#, indicates whether the USB067x is able to complete the current data phase of the transaction. The sampling of both IRDY# and TRDY# as asserted on any clock indicates that a data phase is completed. During a write, the USB067x asserts IRDY# to indicate that it has valid data on AD[31:0]. During a read, the USB067x asserts IRDY# to indicate that it is prepared to accept data. When the USB067x is a target, IRDY# is an input to the USB067x, and when the USB067x is an initiator, IRDY# is an output. IRDY# is tri-stated from the leading edge of RST# and remains so until driven by the USB067x as either a master or a slave.
STOP#	I/O (s/t/s)	STOP: When the USB067x is a target, it asserts STOP# to request that a master stop the current transaction. When the USB067x is a master, the assertion of STOP# forces the USB067x to stop the current transaction. STOP# is an output when the USB067x is a target and an input when the USB067x is an initiator. STOP# is tri-stated from the leading edge of RST# and remains so until driven by the USB067x as either a master or slave.

Signal Name	Type	Description
IDSEL	I	INITIALIZATION DEVICE SELECT: IDSEL is used as a chip select during configuration read and write transactions. The USB067x samples IDSEL during the address phase of a transaction. If IDSEL is found to be active and the bus command is a configuration read or write, the USB067x asserts DEVSEL# on the next cycle.
DEVSEL#	I/O (s/t/s)	DEVICE SELECT: The USB067x claims a PCI transaction through either positive or subtractive decoding by asserting DEVSEL#. The signal serves two purposes as an output. When the USB067x samples IDSEL active in configuration transactions, it asserts DEVSEL#. Secondly, the USB067x asserts DEVSEL# when it decodes an internal USB067x address or when it subtractively decodes a cycle. When DEVSEL# is an input, the signal indicates the target's response to a USB067x master-initiated transaction. The USB067x also samples this signal for all PCI transactions to decide to subtractively decode the cycle. DEVSEL# is tri-stated from the leading edge of RST# and remains so until driven by the USB067x as either a master or a slave.
PAR	I/O	CALCULATED PARITY SIGNAL: PAR is "even" parity and is calculated on 36 bits—AD[31:0] plus C/BE[3:0]#. "Even" parity means that the sum of the 36 bit values plus PAR is always an even number. PAR is always calculated on 36 bits, even if one or more bits of C/BE[3:0] indicate invalid data. The USB067x calculates PAR for address and data phases. PAR is valid one PCI clock after the associated address or data phase, but may or may not be valid for subsequent clocks.
PERR#	I/O (s/t/s)	PARITY ERROR: Error may be pulsed active by an agent that detects a parity error. PERR# can be used by any agent to signal data corruptions. However, on detection of a PERR# pulse, the central resource may generate a nonmaskable interrupt to the host CPU, which often implies that the system will be unable to continue operation once error processing is complete.
SERR#	OD	System Error is for reporting address parity errors, data parity errors on the Special Cycle command, or any other system error where the result will be catastrophic. For more information see section 2.2.5 of the PCI Local Bus Specification version 2.1.
INTA#	OD	Interrupt A is used to request an interrupt.

3.2 PCI Arbiter

Table 3-2: PCI Arbiter Signals

Signal Name	Type	Description
REQ#	t/s	REQUEST: The USB067x asserts this signal when it requires permission to use the PCI bus.

Signal Name	Type	Description
GNT#	I	GRANT: The USB067x samples this signal for an active low assertion, indicating that it has been granted permission to use the PCI bus.

3.3 USB Interface

Table 3-3: USB Interface

Signal Name	Type	Description
VD1+	I/O	Bidirectional positive data line of port 1
VD2+	I/O	Bidirectional positive data line of port 2
VD1-	I/O	Bidirectional negative data line of port 1
VD2-	I/O	Bidirectional negative data line of port 2
PWROK1	I	Analog input/schmitt trigger to sense VCC power on each USB port. (This can be a logic input, on/off, or a resistor divider to sense VCC greater than 4.0V.)
PWROK2	I	Analog input/schmitt trigger to sense VCC power on each USB port. (This can be a logic input, on/off, or a resistor divider to sense VCC greater than 4.0V.)
PWRON1#	I/O	Logic output to turn on the respective USB port VCC power. Boot strap low for high active and boot strap high for low active.
PWRON2#	I/O	Logic output to turn on the respective USB port VCC power. Boot strap low for high active and boot strap high for low active.
PWRFLT1# SDA	I I/OD	Logic input indicating an overcurrent fault on each of the USB ports. Serial Data signal for interfacing to external serial EPROM for retrieving Sub-Vendor ID.
PWRFLT2# SCL	I OD	Logic input indicating an overcurrent fault on each of the USB ports. Serial clock for clocking data into and out of the external serial EPROM for SDA.
UCLK	I	Clock for sampling USB data (48 Mhz).
TEST0 TEST1	I	Test input is for configuring chip level tests. These pins have internal pull down and may be tied to ground or left open. Set both signals to zero for normal mode. Other combinations are reserved.
CLKRUN# PME#	I/OD OD	Input from the system indicating the status of the system clock. Also can be used by the 670 to request the start or speed up of the system clock. Power Management Event. An open drain active low signal that requests a change in the current power management state.

3.4 USB Legacy

Table 3-4: USB Legacy Signals

Signal Name	Type	Description
A20_STATEI	I	Legacy A20gate input signal from keyboard controller. If keyboard controller is absent, this pin should be tied to ground.
A20_STATEO	O	Legacy A20gate output signal to chipset's memory controller.
IRQ1	O	SYSTEM KEYBOARD INTERRUPT: This pin should be tied to the ISA IRQ1 to support legacy keyboards.
IRQ12	O	SYSTEM MOUSE INTERRUPT: This pin should be tied to the ISA IRQ12 to support legacy mouse devices.
KIRQ1I	I	LEGACY KEYBOARD CONTROLLER INTERRUPT: IRQ1 input from keyboard controller.
MIRQ12I	I	LEGACY MOUSE CONTROLLER INTERRUPT: IRQ12 input from mouse controller.
SMI#	O	SYSTEM MANAGEMENT INTERRUPT: This pin should be tied to the SMI# input pin on the Pentium® processor.

This page left blank intentionally.

4 PCI Configuration Registers

4.1 Register Overview

Note:
This section outlines the USB067x's PCI interface, but is not intended to be a definitive discussion. Refer to version 2.2 of the *PCI Local Bus Specification* for detailed information about the PCI interface.

The USB067x's configuration registers are located in PCI configuration space and may be accessed from the PCI bus only in Byte, Word (16-bit), or Dword (32-bit) quantities. The low address contains the least significant bit (LSB). Addresses for configuration registers are offset values that appear on AD[7:2] and C/BE#[3:0].

The non-configuration registers may be accessed from the PCI bus and, in some cases, the universal serial bus (USB). All transactions must be in Byte quantities. Multi-byte transactions will result in the USB067x's asserting a target abort.

The programmer should take special care not to alter the values of reserved bits. Since these bits may not always have predictable values, the programmer should use masks on reads to extract only the non-reserved bits. On writes, the programmer should first read the values of reserved bits and then combine them unchanged with the values to be written.

As required by the PCI specification, the Mandatory Header Registers are located in the first 64 bytes of configuration space. The configuration registers specific to the USB067x reside in offsets 40h to FFh. Refer to Table 4-1.

Table 4-1: Configuration Registers

Configuration Offset	Register	Register Access	Default Value
00h-01h	Vendor Identification	R	1095h
02h-03h	Device Identification	R	0670h/0673h
04h-05h	Command	R/W	0000h
06h-07h	Status	R/W	0200h
08h	Revision Identification	R	xxh
09h-0Bh	Class Code	R	C0310h
0Dh	Latency Timer	R/W	00h
10h-13h	OHCI Base Address Register	R/W	Xxxxx000h
14h-2Bh	Reserved	--	0
2Ch-2Fh	Subsystem, Subsystem Vendor ID	R/W	06701095h
30h-33h	Reserved	--	0

Configuration Offset	Register	Register Access	Default Value
34h	Capabilities Pointer	R	0040h
3Ch	Interrupt	R/W	00h
3Dh	Interrupt Pin	R	01h
3Eh	Min_Gnt	R	02h
3Fh	Max_Lat	R	04h
40h	Capability Identifier	R	x1h
41h	Next Item Pointer	R	00h
42h-43h	Power Management	R	
44h-45h	Power Management Control	R/W (Word Access Only)	
46h-4Dh	Reserved	--	0
4Eh	USB Reset Signal Length	R/W	0Ah
4Fh	CMD Enhancement	R/W	00h

4.1.1 Vendor Identification Register

Address Offset	00h-01h
Default Value	1095h
Attribute	Read Only
Size	16 bits

The 16-bit Vendor Identification number (combined with the Device Identification number) uniquely identifies the USB067x among all PCI devices. This value cannot be modified.

4.1.2 Device Identification Number

Address Offset	02h-03h
Default Value	0670h/0673h
Attribute	Read Only
Size	16 bits

The 16-bit Device Identification number (combined with the Vendor Identification number) uniquely identifies the USB067x among all PCI devices. This value cannot be modified.

4.1.3 Command Register

Address Offset	04h-05h
Default Value	0000h

Attribute	Read/Write
Size	16 bits

Bits	Field	Reset
15:9	Refer to PCI specification for definition.	0
2	BUS_MASTER Indicates the USB067x's ability to act as a bus master.	0
1	MEM_ACCESS Indicates the USB067x's ability to respond to PCI memory cycles.	0
0	Refer to PCI specification for definition.	0

4.1.4 Device Status

Address Offset	06h-07h
Default Value	0200h
Attribute	Read/Write (writes can reset but not set)
Size	16 bits

Please refer to the PCI Specification version 2.1 for more information.

4.1.5 Revision Identification

Address Offset	08h
Default Value	xxh (dependent on part revision)
Attribute	Read Only
Size	4 bits (upper nibble reserved)

4.1.6 Class Code

Address Offset	09h-0Bh
Default Value	C0310h
Attribute	Read Only
Size	24 bits

4.1.7 Latency Timer

Address Offset	0Dh
Default Value	00h
Attribute	Read/Write
Size	8 bits

This register specifies, in units of PCI bus clocks, the value of the Latency Timer for this PCI bus master.

4.1.8 OHCI Base Address Register

Address Offset	10h-13h
Default Value	xxxxx000h
Attribute	Read Only and Read/Write
Size	32 bits

This register specifies the base address of a 4KB contiguous memory space in the PC host's main memory. This space is reserved for the operational registers defined by the OpenHCI specification. All the USB067x's operational registers are directly mapped into this memory space.

Bits	Field	Reset
31:12	BASE_ADDR: The BASE_ADDR value specifies the upper 20 bits of the 32-bit base address. This forces the host to select a 4KB memory space. The USB067x permits read/write access to these bits.	xxxxxh
11:4	These bits are read only and always contain 00h.	00h
3	PREF_MEM: This read-only bit is always set to 0. This indicates that the USB067x does not support pre-fetchable memory.	0
2:1	TYPE: These read-only bits are always set to 0. This indicates that the base register is 32 bits wide and can be placed anywhere in the 32-bit memory space of the host's main memory.	0
0	INDICAT: This read-only bit is always set to 0. This indicates that the operational registers of the USB067x are mapped into the memory space of the host's main memory.	0

4.1.9 Subsystem ID and Subsystem Vendor ID (Available only on USB0670)

Address Offset	2Ch
Default Value	0670 1095h
Attribute	Read/Write[1]
Size	32 bits

[1]To write to this register content you need to enable the Subsystem ID and Subsystem Vendor ID write enable bit located at CMD Enhancement Register (Bit 0, Offset 4Fh)

4.1.10 Capabilities Pointer (Available only on USB0670)

Address Offset	34h
Default Value	40h
Attribute	Read
Size	8 bits

4.1.11 Interrupt Line

Address Offset	3Ch
Default Value	00h
Attribute	Read/Write
Size	8 bits

The Interrupt Line register is used to communicate interrupt line routing information.

4.1.12 Interrupt Pin

Address Offset	3Dh
Default Value	01h
Attribute	Read Only
Size	8 bits

The Interrupt Pin register tells which interrupt pin the device (or device function) uses. A value of 1 corresponds to INTA#.

4.1.13 Min_Gnt

Address Offset	3Eh
Default Value	02h
Attribute	Read Only
Size	8 bits

Min_Gnt in combination with Max_Lat specifies the device's desired settings for Latency Timer values. Min_Gnt is used for specifying how long of a burst period the device needs assuming a clock rate of 33 MHz. The value specifies a period of time in units of ¼microsecond.

4.1.14 Max_Lat

Address Offset	3Fh
Default Value	04h
Attribute	Read Only
Size	8 bits

Max_Lat is used for specifying how often the device needs to gain access to the PCI bus. The value specifies a period of time in units of ¼microsecond.

4.1.15 Capability Identifier (Available only on USB0670)

Address Offset	40h
Default Value	0x1h
Attribute	Read
Size	8 bits

Identifies the linked list items as being the PCI Power Management registers.

4.1.16 Next Item Pointer (Available only on USB0670)

Address Offset	41h
Default Value	00h
Attribute	Read
Size	8 bits

The end of the capability list.

4.1.17 Power Management Capabilities (Available only on USB0670)

Address Offset	42h
Attribute	Read
Size	16 bits

Bit Offset	Default Value	Read/ Write	Description
0 - 2	0x2	Read Only	Indicates that this function complies with Revision 1.1 of the PCI Power Management Interface Specification.
3	0	Read Only	Indicates that no PCI clock is required for the function to generate PME#
4	0	Read Only	Reserved
5	0	Read Only	Indicates that no Device Specific Initialization is required.
6 – 8	0	Read Only	PME# generation from $D3_{cold}$ is not supported.
9	0x1	Read Only	Indicates that D1 is supported in this function.
10	0x1	Read Only	Indicates that D2 is supported in this function.

Bit Offset	Default Value	Read/ Write	Description
11–15	00110b	Read Only	Indicates that PME# can be asserted from D1 and D2.

4.1.18 Power Management Control/Status (Available only on USB0670)

Address Offset	44h
Attribute	Read/Write
Size	16 bits

Note: These registers can only be accessed in Word format only.

Bit	Default Value	Read/ Write	Description
0 – 1	0	Read/ Write	PowerState This 2-bit field is used both to determine the current power state and to set the function into a new power state.
2 – 7	0	Read Only	Reserved
8	0	Read/ Write	PME_En A “1” enables the function to assert PME#. When “0”, PME# assertion is disabled.
	0	Read Only	Not used in this function
15	0	Read/ Write-Clear	PME_Status This bit is set when the function would normally assert the PME# signal independent of the state of PME_En bit. Writing “1” to this bit will clear it and cause the function to stop asserting a PME# (if enabled). Writing “0” has no effect.

4.1.19 CMD Enhancements (Available only on USB0670)

Address Offset	4Eh
Attribute	Read/Write
Size	16 bits

Bit	Default Value	Definition
0 – 5	0xa	**USB Reset Signal Length** The minimum and default value is set to 10 ms (0Ah). USB 1.1 suggests USB Host controller support up to 50ms.
6 – 7	0	Reserved
8	0	**Subsystem ID and Subsystem vendor ID write enable.** Set this bit to 1 to allow to modify Subsystem ID and Subsystem vendor ID at PCI Configuration Space Header offset 0x2c
9	0	**Single Transaction Mode** Set this bit to 1 to allow only one transaction for an endpoint per frame.
10	0	**Control/Bulk OUT Performance mode** Set this bit to 1 to turn on the Control/Bulk OUT performance mode. In regular mode, the host controller starts the Control/Bulk Out transaction until all the OUT data has been loaded into the FIFO. In performance mode, Data Token transmission starts immediately after FIFO is not empty. I.e. the USB transaction and PCI transaction will be performed concurrently.
11	0	**PME#/CLKRUN#** Set this bit to 0 to configure the PME#/CLKRUN# pin to be used as PME# for ACPI power management. Set this bit to 1 to configure the PME#/CLKRUN# pin to be used as CLKRUN# for PCI mobility.

12	0	**Control list PCI bandwidth saving mode** Set this bit to 1 to turn on the Control list PCI bandwidth saving mode. If all the transactions performed of the Control list receive NAK, the Control list will not be processed until either start of frame or two-third of frame is reached*. The tradeoff of this mode is that Control-Bulk Ratio is no longer enforced.
13	0	**Bulk list PCI bandwidth saving mode** Set this bit to 1 to turn on the Bulk list PCI bandwidth saving mode. If all the transactions performed of the Bulk list receive NAK, the Bulk list will not be processed until either start of frame or two-third of frame is reached*. The tradeoff of this mode is that Control-Bulk Ratio is no longer enforced.
14 – 15	0	Reserved

*If both control and bulk list PCI bandwidth saving mode are on, the NAK transactions will only be processed again until start of frame.

This page left blank intentionally.

5 USB HCI Data Structures

Note:
This chapter outlines the USB067x's USB data structures but is not intended to be a definitive discussion of the subject. The USB067x is fully compatible with all relevant industry standards. For more information about the USB067x's data structures refer to *OpenHCI: Open Host Controller Interface Specification for USB,* version 1.0. This is available on the Internet.

5.1 Overview

The USB067x contains a set of internal operational registers that are mapped to a non-cacheable portion of the system addressable space. Refer to Table 5-1. These registers are used by the host controller driver (HCD). All unused bits are set to zero.

Table 5-1: Operational Registers

Offset	Register
00h	HcRevision
04h	HcControl
08h	HcCommandStatus
0Ch	HcInterruptStatus
10h	HcInterruptEnable
14h	HcInterruptDisable
18h	HcHCCA
1Ch	HcPeriodCurrentED
20h	HcControlHeadED
24h	HcControlCurrentED
28h	HcBulkHeadED
2Ch	HcBulkCurrentED
30h	HcDoneHead
34h	HcFmInterval
38h	HcFmRemaining
3Ch	HcFmNumber
40h	HcPeriodicStart
44h	HcLSThreshold
48h	HcRhDescriptorA
4Ch	HcRhDescriptorB
50h	HcRhStatus
54h	HcRhPortStatus[1:2]
100h	EmuControl †

Offset	Register
104h	EmuInputBuffer †
108h	EmuOutputBuffer †
10Ch	EmuStatus †

† These bits are required for legacy keyboard and mouse support and are not part of the OpenHCI specification. See section 5.7 on page 5-11.

5.2 Control and Status Partition

5.2.1 HcRevision Register

This is a read-only field that contains the revision (in binary coded decimal format) of the Host Controller Interface (HCI) specification implemented by the USB067x. For example: a value of 10h translates to version 1.0, and a value of 11h translates to version 1.1.

Bits 31:9 will always read as zero. Bit 8 is 1 to indicate that the legacy support registers are present in this host controller.

Bits	Field	Reset	HCD	HC
7:0	Revision (REV)	10h	R	R
8	Legacy (L)	1b	R	R

5.2.2 HcControl Register

This register defines the operating modes for the Host Controller. Most of the fields in this register (except HostControllerFunctionalState and RemoteWakeupConnected) are only modified by the Host Controller Driver.

Bits	Field	Reset	HCD	HC
1:0	Control Bulk Service Ratio (CBSR)	01b	R/W	R
2	Periodic List Enable (PLE)	0b	R/W	R
3	Isochronous Enable (IE)	0b	R/W	R
4	Control List Enable (CLE)	0b	R/W	R
5	Bulk List Enable (BLE)	0b	R/W	R
7:6	Host Controller Functional State (HCFS) 00b: UsbReset 01b: UsbResume 10b: UsbOperational 11b: UsbSuspend	00b	R/W	R/W
8	Interrupt Routing (IR)	0b	R/W	R
9	Remote Wakeup Connected (WC)	0b	R/W	R/W
10	Remote Wakeup Enable (RWE)	0b	R/W	R

5.2.3 HcCommandStatus Register

The HcCommandStatus register receives commands from the host controller driver and holds information regarding the current status of the USB067x. This is a “write to set” register, which means the host controller driver may write a “1” to set a bit, but it may not clear a bit by writing “0.” Writes of “0” will have no effect on the register. Only the USB067x may clear the bits in this register. The host controller driver may issue more than one distinct command to the USB067x without risking the corruption of previous commands. The host controller driver's read access to the register is unrestricted.

Bit	Field	Reset	HCD	HC
0	Host Controller Reset (HCR)	0b	R/W	R/W
1	Control List Filled (CLF)	0b	R/W	R/W
2	Bulk List Filled (BLF)	0b	R/W	R/W
3	Ownership Change Request (OCR)	0b	R/W	R/W
17:16	Scheduling Overrun Count (SOC)	0b	R	R/W

5.2.4 HcInterruptStatus Register

The USB067x uses this register to convey the status of events that cause hardware interrupts. Each time the USB067x detects an event, it sets the corresponding bit in this register, updates the content of HCCA, and generates a hardware interrupt if the interrupt is enabled in the HcInterruptMask register.

Bit	Field	Reset	HCD	HC
0	Scheduling Overrun (SO)	0b	R/W	R/W
1	Writeback Done Head (WDH)	0b	R/W	R/W
2	Start Of Frame (SF)	0b	R/W	R/W
3	Resume Detected (RD)	0b	R/W	R/W
4	Unrecoverable Error (UE)	0b	R/W	R/W
5	Frame Number Overflow (FNO)	0b	R/W	R/W
6	Root Hub Status Change (RHSC)	0b	R/W	R/W
31	Ownership Change (OC)	0b	R/W	R/W

5.2.5 HcInterruptEnable Register

Each enable bit in the HcInterruptEnable register corresponds to an associated interrupt bit in the HcInterruptStatus register. The HcInterruptEnable register is used to control which events generate a hardware interrupt. When a bit is set in the HcInterruptStatus register, the corresponding bit is set in the HcInterruptEnable register *and* the MasterInterruptEnable bit is set, then a hardware interrupt is requested on the host bus.

Bit	Field	Reset	HCD	HC
0	Scheduling Overrun (SO) 0—Ignore 1—Enable interrupt	0b	R/W	R
1	Writeback Done Head (WDH) 0—Ignore 1—Enable interrupt	0b	R/W	R
2	Start Of Frame (SF) 0—Ignore 1—Enable interrupt	0b	R/W	R
3	Resume Detected (RD) 0—Ignore 1—Enable interrupt	0b	R/W	R
4	Unrecoverable Error (UE) 0—Ignore 1—Enable interrupt	0b	R/W	R
5	Frame Number Overflow (FNO) 0—Ignore 1—Enable interrupt	0b	R/W	R
6	Root Hub Status Change (RHSC) 0—Ignore 1—Enable interrupt	0b	R/W	R
30	Ownership Change (OC) 0—Ignore 1—Enable interrupt	0b	R/W	R
31	Master Interrupt Enable (MIE) 0—Ignored by HC 1—Enables interrupt generation due to events specified in the other bits of this register	0b	R/W	R

5.3 HcInterruptDisable Register

Each disable bit in the HcInterruptDisable register corresponds to an associated interrupt bit in the HcInterruptStatus register. The HcInterruptDisable register is coupled with the HcInterruptEnable register. Thus, writing a “1” to a bit in this register clears the corresponding bit in the HcInterruptEnable register, whereas writing a “0” to a bit in this register leaves the corresponding bit in the HcInterruptEnable register unchanged. On read, the current value of the HcInterruptEnable register is returned.

Bit	Field	Reset	HCD	HC
0	Scheduling Overrun (SO) 0—Ignore 1—Disable interrupt generation	0b	R/W	R
1	Writeback Done Head (WDH) 0—Ignore 1—Disable interrupt	0b	R/W	R
2	Start Of Frame (SF) 0—Ignore 1—Disable interrupt	0b	R/W	R
3	Resume Detected (RD) 0—Ignore 1—Disable interrupt	0b	R/W	R
4	Unrecoverable Error (UE) 0—Ignore 1—Disable interrupt	0b	R/W	R
5	Frame Number Overflow (FNO) 0—Ignore 1—Disable interrupt	0b	R/W	R
6	Root Hub Status Change (RHSC) 0—Ignore 1—Disable interrupt	0b	R/W	R
30	Ownership Change (OC) 0—Ignore 1—Disable interrupt	0b	R/W	R
31	Master Interrupt Enable (MIE) 0—Ignored by HC 1—Disables interrupt generation due to events specified in the other bits of this register.	0b	R/W	R

5.4 Memory Pointer Partition

5.4.1 HcHCCA Register

This register contains the physical address of the 256-byte Host Controller Communication Area. The host controller driver gets the alignment restrictions by writing all 1s to HcHCCA and then reading back the contents of the register. When evaluating the returned value, a 0 is considered true. The Host Controller Communications Area is used to hold the control structures and the Interrupt Table used by USB067x. It also holds the host controller driver.

Bits	Field	Reset	HCD	HC
31:8	Host Controller Communications Area (HCCA) Base Address Bits 7:0 will always return 0.	0h	R/W	R

5.4.2 HcPeriodCurrentED Register

This register contains the physical address of the current Isochronous or Interrupt ED. The USB067x uses this to locate the head of periodic lists that it will process in the current frame. After the USB067x processes a periodic ED, it updates this register. The host controller driver may determine which ED is being processed by reading this register.

Bits	Field	Reset	HCD	HC
31:4	Period Current ED (PCED) Base Address Bits 3:0 will always return 0.	0h	R/W	R

5.4.3 HcControlHeadED Register

This register contains the physical address of the first ED in the control list. This is the starting point for the USB067x as it processes the control list.

Bits	Field	Reset	HCD	HC
31:4	Control Head ED (CHED) Base Address Bits 3:0 will always return 0.	0h	R/W	R

5.4.4 HcControlCurrentED Register

This register contains the physical address of the current ED in the control list.

Bits	Field	Reset	HCD	HC
31:4	Control Current ED (CCED) Base Address Bits 3:0 will always return 0.	0h	R/W	R/W

5.4.5 HcBulkHeadED Register

This register contains the physical address of the first ED in the bulk list. The USB067x uses this address as its starting point when it sets out to process the bulk list.

Bits	Field	Reset	HCD	HC
31:4	Bulk Head ED (BHED) Base Address Bits 3:0 will always return 0.	0h	R/W	R

5.4.6 HcBulkCurrentED Register

This register contains the physical address of the current endpoint of the bulk list

Bits	Field	Reset	HCD	HC
31:4	Bulk Current ED (BCED) Base Address Bits 3:0 will always return 0.	0h	R/W	R/W

5.4.7 HcDoneHead Register

This register contains the physical address of the TD that the USB067x most recently processed and appended to the done queue.

Bits	Field	Reset	HCD	HC
31:4	Done Head (DH) Base Address Bits 3:0 will always return 0.	0h	R	R/W

5.5 Frame Counter Partition

5.5.1 HcFmInterval Register

This register holds the bit time interval between consecutive frames and the largest packet size that the USB067x may send at SOF. By manipulating these values, the host controller driver may adjust the frame interval in order to synchronize with an external clock.

Bits	Field	Reset	HCD	HC
13:0	Frame Interval (FI)	2EDFh	R/W	R
30:16	FS Largest Data Packet (FSMPS)		R/W	R
31	Frame Interval Toggle (FIT)	0b	R/W	R

5.5.2 HcFmRemaining Register

This register shows the bit time remaining in the current Frame.

Bits	Field	Reset	HCD	HC
13:0	Frame Remaining (FR)	0h	R	R/W
31	Frame Remaining Toggle (FRT)	0b	R	R/W

5.5.3 HcFmNumber Register

The HcFmNumber register is a 16-bit counter. It provides a timing reference among events happening in the Host Controller and the Host Controller Driver. The Host Controller Driver may use the 16-bit value specified in this register and generate a 32-bit frame number without requiring frequent access to the register.

Bits	Field	Reset	HCD	HC
15:0	Frame Number (FN)	0h	R	R/W

5.5.4 HcPeriodicStart Register

This register determines the earliest time that the USB067x will start processing the periodic list.

Bits	Field	Reset	HCD	HC
13:0	Periodic Start (PS)	0h	R/W	R

After a hardware reset, this field is cleared. This is then set by HCD during the HC initialization. The value is calculated roughly as 10 percent off from HcFmInterval. A typical value will be 3E67h. When HcFmRemaining reaches the value specified, processing of the periodic lists will have priority over Control/Bulk processing. HC will therefore start processing the Interrupt list after completing the current Control or Bulk transaction that is in progress.

5.5.5 HcLSThreshold

The HcLSThreshold register contains an 11-bit value used by the Host Controller to determine whether to commit to the transfer of a maximum of 8-byte LS packet before EOF. Neither the Host Controller nor the Host Controller Driver are allowed to change this value.

Bits	Field	Reset	HCD	HC
11:0	LSThreshold	0700h	R/W	R

5.6 Root Hub Partition

The USB root hub is an integral part of the USB067x. The host controller driver may access the registers in the root hub partition directly through a memory map, rather than through packet transfers to an endpoint. This means that the host controller driver must differentiate the root hub from other hubs on the USB.
All the root hub registers are 32-bits wide and must be read from and written to in Dwords.
The contents of these root hub registers are not affected by a software reset, but they will be affected by a hardware reset or a transition to the UsbReset state. In the event of a transition to the UsbReset state, the USB067x will reset only the HcRhStatus and HcRhPortStatus registers.

5.6.1 HcRhDescriptorA Register

This register paired with HcRhDescriptorB describes the characteristics of the root hub.

Bits	Field	Reset	HCD	HC
7:0	NumberDownstreamPorts (NDP)	02h	R	R

Bits	Field	Reset	HCD	HC
8	PowerSwitchingMode (PSM)	0b	R/W	R
9	NoPowerSwitching (NPS)	1b	R/W	R
10	Device Type (DT)	0b	R	R
11	OverCurrentProtectionMode (OCPM)	1b	R/W	R
12	NoOverCurrentProtection (NOCP)	0b	R/W	R
24:31	PowerOnToPowerGoodTime (POTPGT)	FFh	R/W	R

5.6.2 HcRhDescriptorB Register

The HcRhDescriptorB register is the second register of two describing the characteristics of the Root Hub. These fields are written during initialization to correspond with the system implementation. Reset values are implementation-specific.

Bits	Field	Reset	HCD	HC
15:0	DeviceRemovable (DR)	0000h	R/W	R
17:16	PortPowerControlMask (PPCM)	0006h	R/W	R

5.6.3 HcRhStatus Register

The HcRhStatus register is divided into two parts. The lower word of a Dword represents the Hub Status field: the upper word represents the Hub Status Change field. Reserved bits should always be written “0.”

Bits	Field	Reset	HCD	HC
0	Local Power Status (LPS)	0b	R/W	R
1	Over Current Indicator (OCI)	0b	R	R/W
15	Device Remote Wakeup Enable (DRWE)	0b	R/W	R
16	Local Power Status Change (LPSC)	0b	R/W	R
17	Over Current Indicator Change (OCIC)	0b	R/W	R/W
31	Clear Remote Wakeup Enable (CRWE)	–	W	R

5.6.4 HcRhPortStatus[1:2] Register

The HcRhPortStatus[1:2] register is used to control and report port events on a per-port basis.

Bits	Field	Reset	HCD	HC
0	Current Connect Status (CCS)	0b	R/W	R/W
1	Port Enable Status(PES)	0b	R/W	R/W
2	Port Suspend Status (PSS)	0b	R/W	R/W
3	Port Over Current Indicator (POCI)	0b	R/W	R/W
4	Port Reset Status (PRS)	0b	R/W	R/W
8	Port Power Status (PPS)	0b	R/W	R/W
9	Low Speed Device Attached (LSDA)	Xb	R/W	R/W
16	Connect Status Change (CSC)	0b	R/W	R/W
17	Port Enable Status Change (PESC)	0b	R/W	R/W
18	Port Suspend Status Change (PSSC)	0b	R/W	R/W
19	Port Over Current Indicator Change (OCIC)	0b	R/W	R/W
20	Port Reset Status Change (PRSC)	0b	R/W	R/W

5.7 Legacy Support

The OpenHCI Legacy Support Specification v. 1.01 defines the design shown in Figure 5-1for the support of legacy devices. For more information, refer to the OpenHCI Legacy Support Specification v. 1.01.

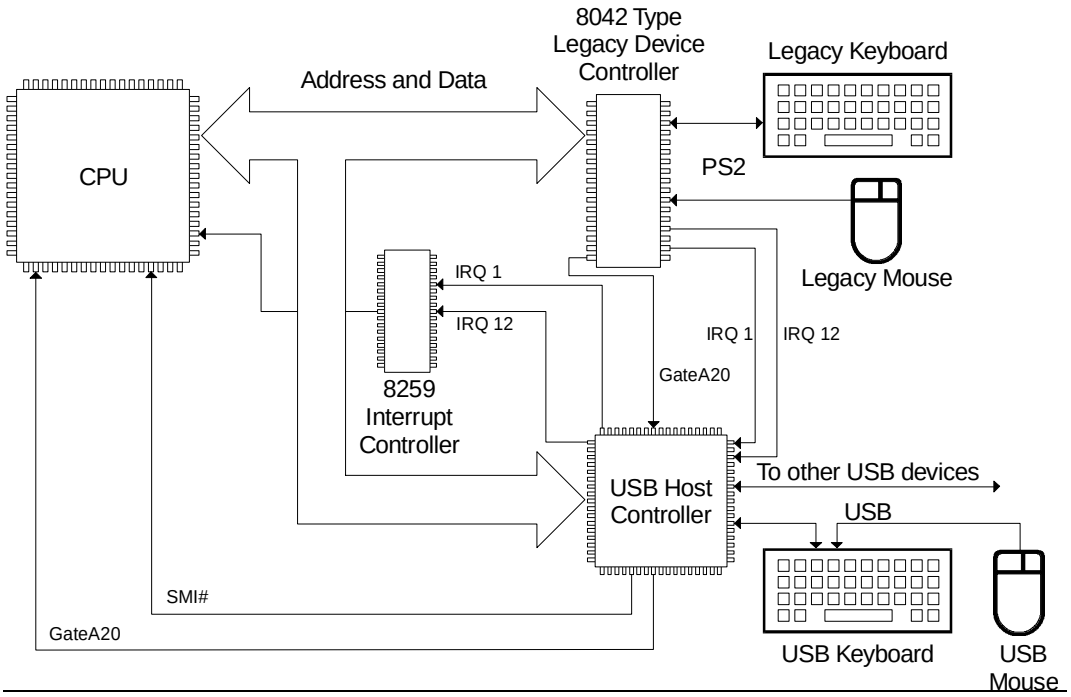


Figure 5-1: OpenHCI legacy support

5.7.1 Legacy Support Registers

Four operational registers (described in Table 5-2) are used to provide the legacy support. Each of these registers is located on a 32-bit boundary. The offset of these registers is relative to the base address of the HC operational registers with HceControl located at offset 100h.

Table 5-2: Legacy Support Registers

Offset	Register	Description
100h	HceControl	Used to enable and control the emulation hardware and report various status information.
104h	HceInput	The emulation side of the legacy Input Buffer register.
108h	HceOutput	The emulation side of the legacy Output Buffer register where keyboard and mouse data is to be written by software.
10Ch	HceStatus	The emulation side of the legacy Status register.

Three of the operation registers (HceStatus, HceInput, and HceOutput) are accessible at I/O address 60h and 64h when emulation is enabled. Reads and writes to the registers using I/O address have side effects as outlined in Table 5-3.

Table 5-3: Emulated Registers

I/O Address	Cycle Type	Register Contents Accessed/Modified	Side Effects
60h	IN	HceOutput	IN from port 60h will set OutputFull in HceStatus to 0.
60h	OUT	HceInput	OUT to port 60h will set InputFull to 1 and

I/O Address	Cycle Type	Register Contents Accessed/Modified	Side Effects
			CmdData to 0 in HceStatus.
64h	IN	HceStatus	IN from port 64h returns current value of HceStatus with no other side effect.
64h	OUT	HceInput	OUT to port 64h will set InputFull and CmdData in HceStatus to 1.

5.7.1.1 HceInput Register

Table 5-4: HceInput Register

bit	Field	R/W	Description
7:0	InputData	R/W	This register holds data that is written to I/O ports 60h and 64h.
31:8	Reserved		

I/O data that is written to ports 60h and 64h is captured in this register when emulation is enabled. This register may be read or written directly by accessing it with its memory address in the Host Controller's operational register space. When accessed directly with a memory cycle, reads and writes of this register have no side effects.

5.7.1.2 HceOutput Register

Table 5-5: HceOutput Register

bit	Field	R/W	Description
7:0	OutputData	R/W	This register holds data that is returned when an I/O read of port 60h is performed by application software.
31:8	Reserved		

The data placed in this register by the emulation software is returned when I/O port 60h is read and emulation is enabled. On a read of this location, the OutputFull bit in HceStatus is set to 0.

5.7.1.3 HceStatus Register

Table 5-6: HceStatus Register

bit	Field	R/W	Description
0	OutputFull	R/W	The host controller will set this bit to 0 on a read of I/O port 60h. If IRQEn is set and AuxOutputFull is set to 0 then an IRQ1 is generated as long as this bit is set to 1. If IRQEn is set and AuxOutputFull is set to 1 then IRQ12 will be generated as long as this bit is set to 1. While tis bit is 0 and CharacterPending in HceControl is set to 1, an emulation interrupt condition exists.

bit	Field	R/W	Description
1	InputFull	R/W	Except for the case of a Gate A20 sequence, this bit is set to 1 on an I/O write to address 60h or 64h. While this bit is set to 1 and emulation is enabled, an emulation interrupt condition exists.
2	Flag	R/W	Nominally used as a system flag by software to indicate a warm or cold boot.
3	CmdData	R/W	The host controller will set this bit to 0 on an I/O write to port 60h and on an I/O write to port 64h the host controller will set this bit to 1.
4	Inhibit Switch	R/W	This bit reflects the state of the keyboard inhibit switch and is set if the keyboard is NOT inhibited.
5	AuxOutput Full	R/W	IRQ is asserted whenever this bit is set to 1 and OutputFull is set to 1 and the IRQEn bit is set.
6	Timeout	R/W	Used to indicate a time-out.
7	Parity	R/W	Indicates parity error on keyboard/mouse data.
31:8	Reserved		

The contents of the HceStatus Register is returned on an I/O Read of port 64h when emulation is enabled. Reads and writes of port 60h and writes to port 64h can cause changes in this register.

Emulation software can directly access this register through its memory address in the host controller's operational register space. Access of this register through its memory address produces no side effects.

5.7.1.4 HceControl Register

Table 5-7: HceControl Register

bit	Field	Reset	R/W	Description
0	EmulationEnable	0b	R/W	When set to 1 the host controller will be enabled for legacy emulation. The host controller will decode accesses to I/O registers 60h and 64h and generate IRQ1 and/or IRQ12 when appropriate. Additionally, the host controller will generate an emulation interrupt at appropriate times to invoke the emulation software.
1	EmulationInterrupt	-	R	This bit is a static decode of the emulation interrupt condition.
2	CharacterPending	0b	R/W	When set, an emulation interrupt will be generated when the OutputFull bit of the HceStatus register is set to 0.

bit	Field	Reset	R/W	Description
3	IRQEn	0b	R/W	When set the host controller will generate IRQ1 or IRQ12 as long as the OutputFull bit in HceStatus is set to 1. If the AuxOutputFull bit of HceStatus is 0 then IRQ1 is generated and if it is 1, then an IRQ12 is generated.
4	ExternalIRQEn	0b	R/W	When set to 1, IRQ1 and IRQ12 from the keyboard controller will cause an emulation interrupt. The function controlled by this bit is independent of the setting of the EmulationEnable bit in this register.
5	GateA20Sequence	0b	R/W	Set by the host controller when a data value of D1h is written to I/O port 64h. Cleared by the host controller on write to I/O port 64h of any value other than D1h.
6	IRQ1Active	0b	R/W	Indicates that a positive transition on IRQ1 from the keyboard controller has occurred. Software may write a 1 to this bit to clear it (set it to 0). A software write of 0 to this bit has no effect.
7	IRQ12Active	0b	R/W	Indicates that a positive transition on IRQ12 from the keyboard controller has occurred. Software may write a 1 to this bit to clear it (set it to 0). A software write of a 0 to this bit has no effect.
8	A20State	0b	R/W	Indicates current state of Gate A20 on the keyboard controller. Used to compare against the value written to 60h when GateA20Sequence is active.
31:9	Reserved			Must read as zeros.

This page left blank intentionally.

6 Electrical and Timing

NOTE
The information in this chapter was taken from Chapter 7 of the *Universal Serial Bus Specification*, Revision 1.1, published by Compaq Computer Corporation, Digital Equipment Corporation, IBM PC Company, Intel Corporation, Microsoft Corporation, NEC, and Northern Telecom. For a complete discussion of this topic, refer to the *Universal Serial Bus Specification*.

6.1 Full Speed (12 Mbs) Driver Characteristics

A full speed USB connection is made through a shielded, twisted pair cable with a characteristic impedance (Z_0) of 90Ω ±5% and a maximum length of 5 meters. The impedance of each of the drivers must be between 29Ω and 44Ω. The data line rise and fall times must be between 4 ns and 20 ns, smoothly rising or falling (monotonic), and be well matched to minimize RFI emissions and signal skew.

6.2 Low Speed (1.5 Mbs) Driver Characteristics

A low speed USB connection is made through an unshielded, untwisted wire cable, a maximum of 3 meters in length. The rise and fall time of the signals on this cable must be greater than 75 ns to keep RFI emissions under FCC class B limits, and less than 300 ns to limit timing delays and signaling skews and distortions.

6.3 Connect and Disconnect Signaling

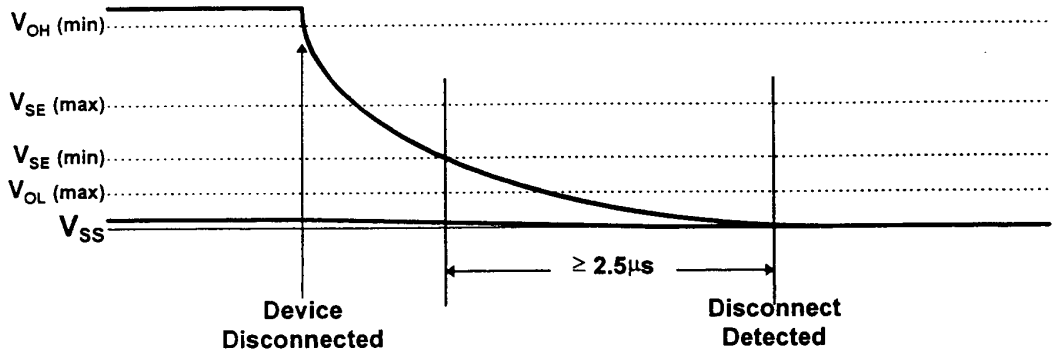


Figure 6-1: Disconnect Detection

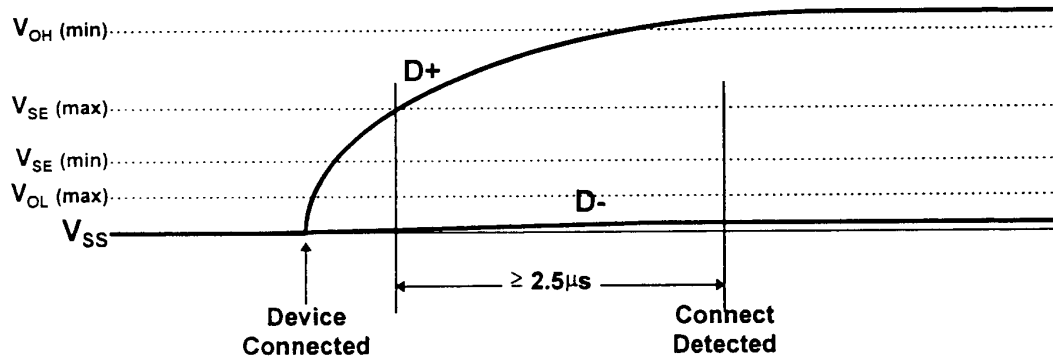


Figure 6-2: Full Speed Device Connect Detection

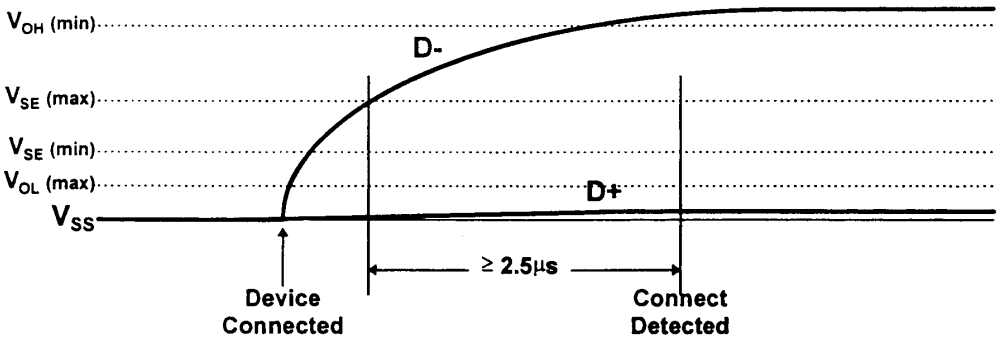


Figure 6-3: Low Speed Device Connect Detection

6.4 Data Signaling

The originating port signals the start of packet (SOP) by driving the D+ and D- lines from the idle state to the opposite logic level (“K” state). This switch in levels represents the first bit of the Sync field.

The single-ended 0 state is used to signal an end of packet (EOP). The single-ended 0 state is indicated by both D+ and D- being below 0.8V. EOP will be signaled by driving D+ and D- to the single-ended 0 state for two bit times followed by driving the lines to the idle state for one bit time.

The transition from the single-ended 0 to the idle state defines the end of the packet. The idle state is asserted for 1 bit time. Then both the D+ and D- output drivers are placed in their high-impedance state. The bus termination resistors hold the bus in the idle state.

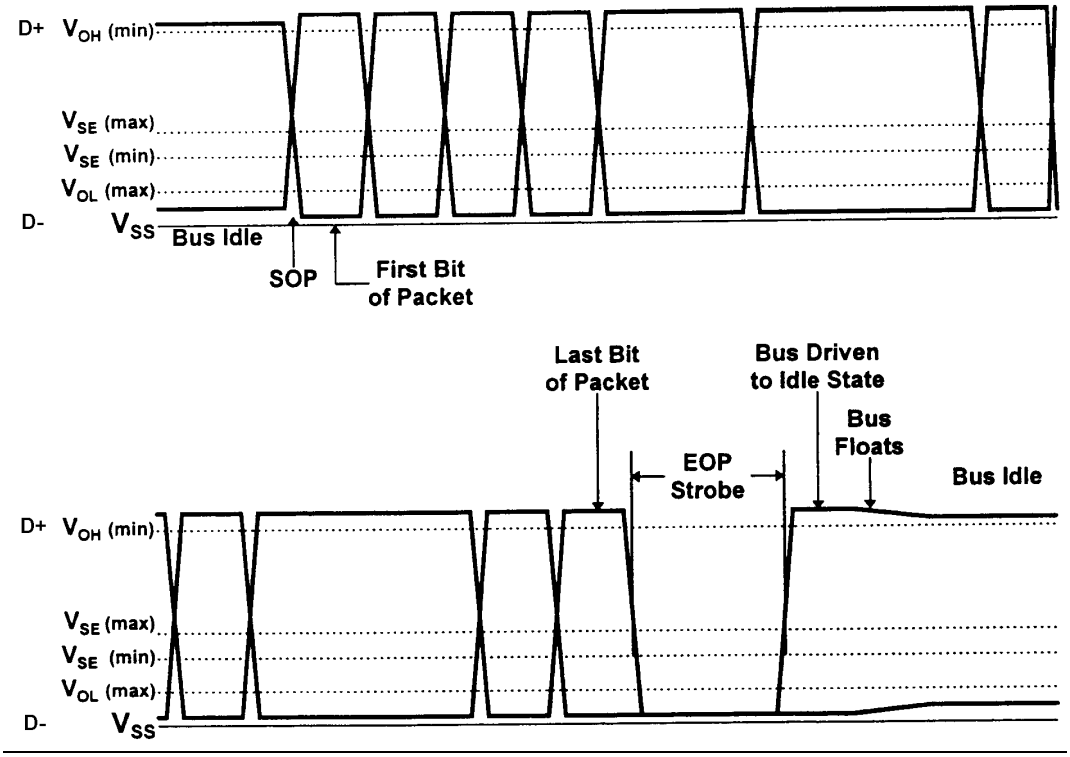


Figure 6-4: Packet Transaction Voltage Levels

6.5 EOP Width

The width of the SE0 in the EOP is about 2 * T_{PERIOD}. The EOP width is measured with the same capacitive load used for maximum rise and fall times and is measured at the same level as the differential signal crossover points of the data lines (see Figure 6-5).

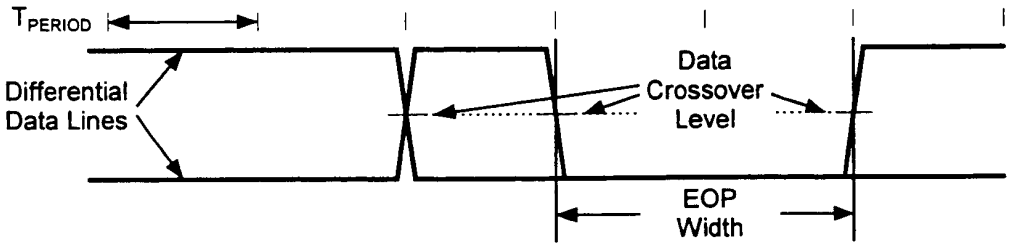


Figure 6-5: EOP Width Timing

For full speed transmissions, the EOP width from the transmitter must be between 160 ns and 175 ns. These ranges include timing variations due to differential buffer delay and rise/fall time mismatches and to noise and other random effects.

6.6 Cable Delay

Only one data transition is allowed on a USB cable at a time. A full speed signal edge has to transition, propagate to the far end of the cable, return, and settle within one full speed bit time. Therefore, the maximum allowed cable delay is 30 ns. Independent of cable velocity, the

maximum cable length is 5.0 meters for full speed devices and 3.0 meters for low speed devices. The maximum one-way data delay on a cable is measured as shown in Figure 6-6.

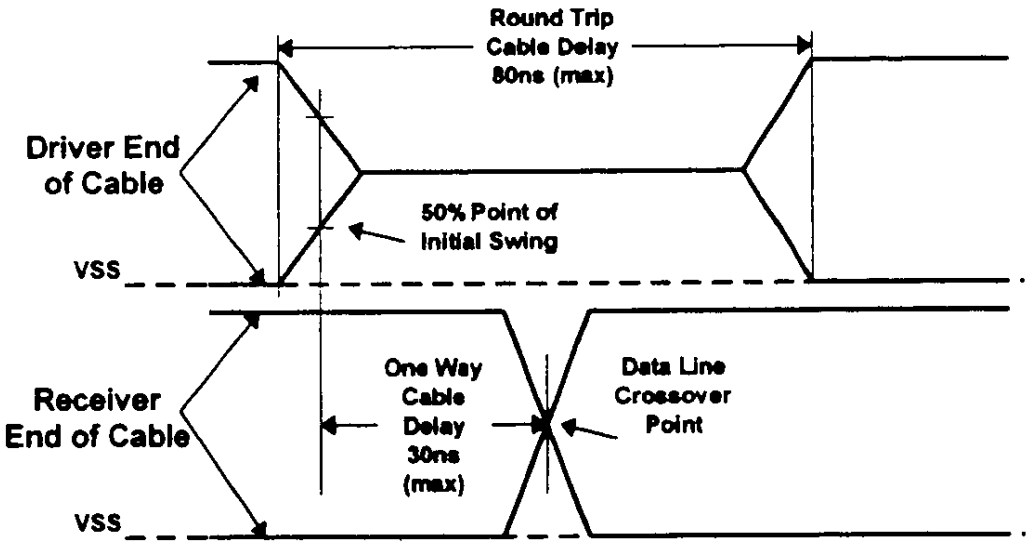


Figure 6-6: Cable Delay

6.7 Bus Timing/Electrical Characteristics

Table 6-1: Full Speed Source Electrical Characteristics

Parameter	Symbol	Conditions (Notes 1, 2, 3)	Min.	Max.	Unit
Driver Characteristics					
Transition Time:		Notes 5, 6			
Rise Time	TR	CL = 50 pF	4	20	ns
Fall Time	TF	CL = 50 pF	4	20	ns
Rise / Fall Time Matching	TRFM	(TR/TF)	90	110	%
Output Signal Crossover Voltage	VCRS		1.3	2.0	V
Driver Output Resistance	ZDRV	Steady State Drive	28	43	Ω
Data Source Timings:					

Parameter	Symbol	Conditions (Notes 1, 2, 3)	Min.	Max.	Unit
Full Speed Data Rate	TDRATE	Ave. Bit Rate (12MB/s ± 0.25%)	11.97	12.03	Mbs
Frame Interval	TFRAME	1.0 ms ± 0.05%	0.9995	1.0005	ms
Source Differential Driver Jitter		Notes 7, 8			
To Next Transition	TDJ1		-3.5	3.5	ns
For Paired Transitions	TDJ2		-4.0	4.0	ns
Source EOP Width	TEOPT	Note 8	160	175	ns
Differential to EOP Transition Skew	TDEOP	Note 8	-2	5	ns
Receiver Data Jitter Tolerance		Note 8			
To Next Transition	TJR1		-18.5	18.5	ns
For Paired Transitions	TJR2		-9	9	ns
EOP Width at Receiver		Note 8			
Must reject as EOP	TEOPR1		40		ns
Must accept as EOP	TEOPR2		82		ns

Table 6-2: Low Speed Source Electrical Characteristics

Parameter	Symbol	Conditions	Min	Max	Unit
Driver Characteristics:					
Transition Time		Note 5, 6			
Rise Time	TR	CL = 50 pF	75		ns
		CL = 350 pF		300	ns
Fall Time	TF	CL = 50 pF	75		ns
		CL = 350 pF		300	ns
Rise/Fall Time Matching	TRFM	(TR/TF)	80	120	%
Output Signal Crossover Voltage	VCRS		1.3	2.0	V
Data Source Timings:					
Low Speed Data Rate	TDRATE	Ave. Bit Rate (1.5Mb/s ±1.5%)	1.4775	1.5225	Mbs
Source Differential Driver Jitter		Note 7, 8, Figure 6-7			
Host (Downstream)					
To Next Transition	TDDJ1		-75	75	ns
For Paired Transitions	TDDJ2		-45	45	ns

Parameter	Symbol	Conditions	Min	Max	Unit
Function (Upstream)					
To Next Transition	TUDJ1		-95	95	ns
For Paired Transitions	TUDJ2		-150	150	ns
Source EOP Width	TEOPT	Note 8, Figure 6-8	1.25	1.50	❍s
Differential to EOP Transition Skew	TDEOP	Note 8, Figure 6-8	-40	100	ns
Receiver Data Jitter Tolerance		Note 8, Figure 6-9			
At Host (Upstream)					
To Next Transition	TUJR1		-152	152	ns
For Paired Transitions	TUJR2		-200	200	ns
At Function (Downstream)					
To Next Transition	TDJR1		-75	75	ns
For Paired Transitions	TDJR2		-45	45	ns
EOP Width at Receiver		Note 8, Figure 6-9			
Must reject as EOP	TEOPR1		330		ns
Must accept	TEOPR2		675		ns

Note 1: All voltages measured from the local ground potential, unless otherwise specified.

Note 2: All timings use a capacitive load (CL) to ground of 50 pF, unless otherwise specified.

Note 3: Full speed timings have a 1.5 kΩ pull-up to 2.8 V on the D+ data line.

Note 4: Low speed timings have 1.5 kΩ pull-up to 2.8 V on the D- data line.

Note 5: Measured from 10% to 90% of the data signal.

Note 6: The rising and falling edges should be smoothly transitioning (monotonic).

Note 7: Timing difference between the differential data signals.

Note 8: Measured at crossover point of differential data signals.

Note 9: The maximum load specification is the maximum effective capacitive load allowed that meets the target hub V_{BUS} droop of 330 mV.

6.8 Timing Waveforms

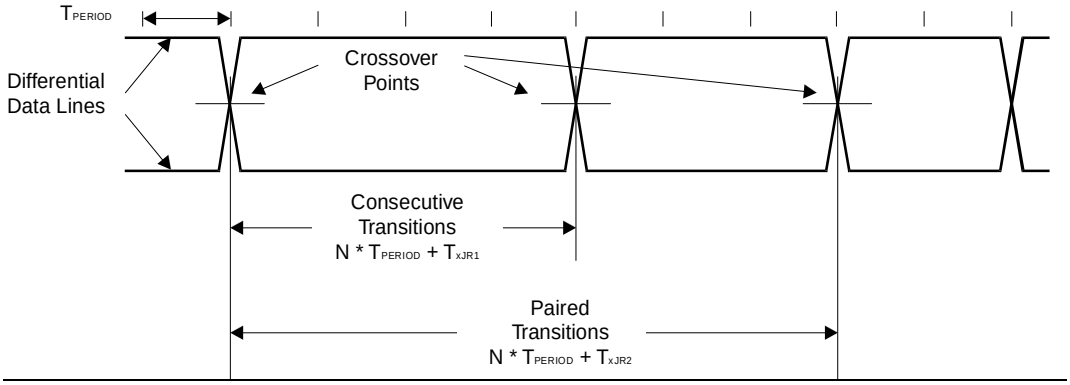


Figure 6-7: Differential Data Jitter

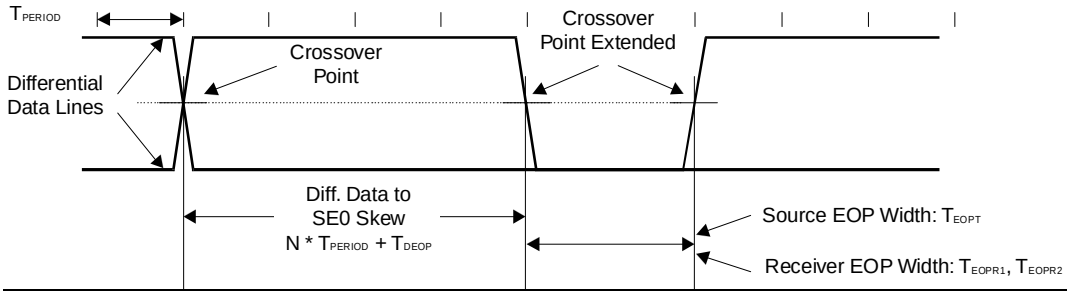


Figure 6-8: Differential to EOP Transition Skew and EOP Width

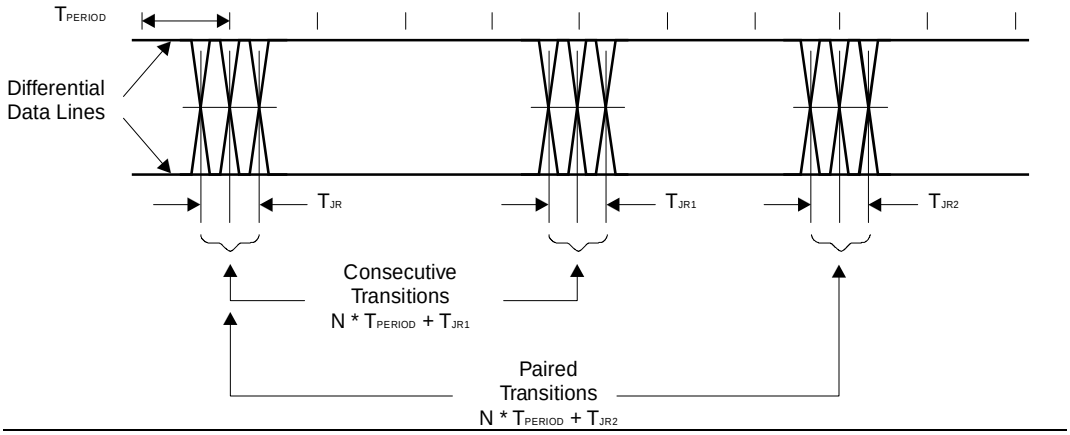


Figure 6-9: Receiver Jitter Tolerance

This page left blank intentionally.

7 Design Recommendations

This chapter offers design recommendations for the 670/673, but is not intended to exclude the use of alternative designs.

7.1 Power Connection

CMD Technology has developed three alternatives for connecting power to the 670/673 chip. The three alternatives are combined in the schematic shown in Figure 7-1.

Note
This schematic combines three alternatives. Select the one that best suits your needs, but do not use more than one alternative.

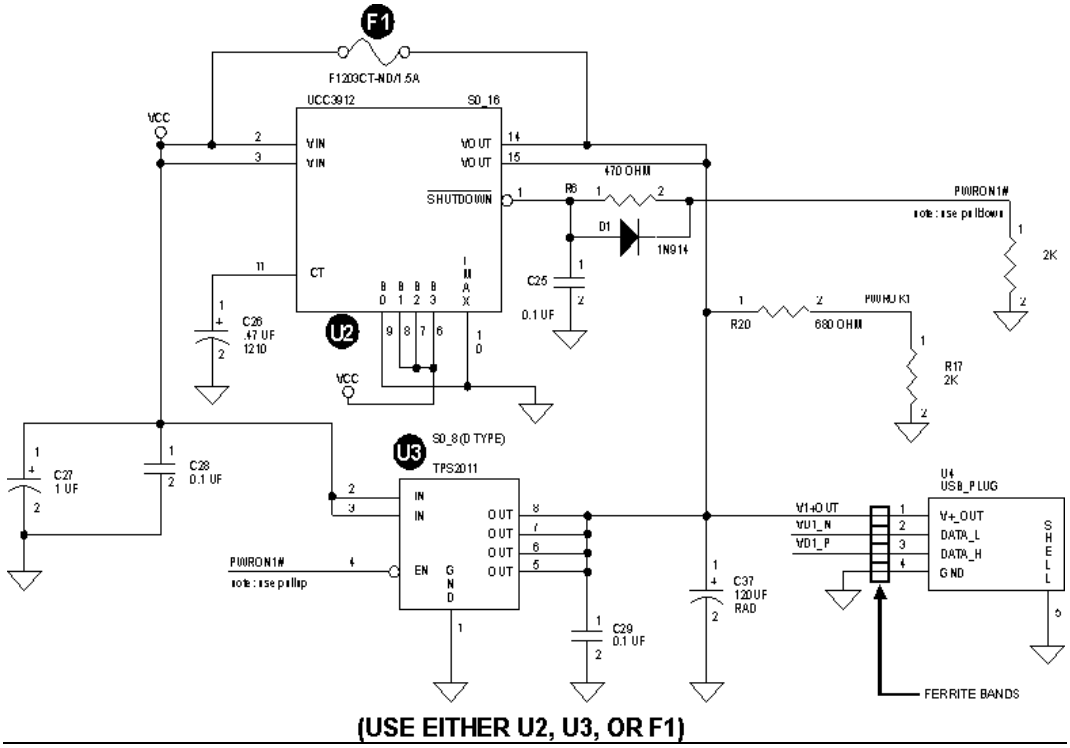


Figure 7-1: Power Connection Alternatives Schematic (Use One Only)

7.1.1 Option 1: Low Cost, Low Reliability

This option employs a fuse (F1203CT-ND/1.5A) at F1.

7.1.2 Option 2: Higher Cost, Higher Reliability

This option uses the glue logic shown at U3 with R13 and R1. Omit R12. The part number for the component is TPS2011.

7.1.3 Option 3: Highest Cost, Highest Reliability

This option uses the glue logic shown at U2 with R1 and R12. Omit R13. The part number for the component is UCC3912 from Unitrode Corporation.

7.1.4 Power Sequencing

The USB670 has two power rails: VDD at 5.0 volts, and VDD3 at 3.3 volts. It is IMPERATIVE that the VDD3 never exceeds VDD: otherwise a latchup may occur, destroying the chip.

During power-up, VDD and VDD3 may rise at the same time, provided VDD3 does not exceed VDD. In practice, limiting VDD3 to less than 0.3V more than VDD is generally acceptable if the injection current is small and of short duration.

Upon power-down, it is acceptable to have VDD collapse before VDD3, provided the current through any power supply pin is limited to less than 50 mA for no more than 5 milliseconds worst case. A Schottky diode clamp may be used to ensure this if it is not otherwise part of the system design. Undershoot upon power-down should also be limited to less than 0.3V (reversed polarity) on any power supply pin. This can also cause latchup.

These are general considerations for any multiple supply system. They are especially important where CMOS logic is used. Good engineering practice normally includes these design considerations.

7.1.5 Hub Power Switching

The USB Hub can be self-powered or bus-powered. In the event that the USB Hub is forced to switch between self-power and bus-power (such as when it loses power) the USB Hub will issue a disconnect and connect signal upstream to the USB Host. Depending on the delay between the transmission of a disconnect signal by the USB Hub and the transmission of a connect signal , the USB Host may ignore the request.

Switching power between self-power and bus-power in a USB Hub may impact devices connected downstream from it. It is therefore suggested that USB Hub designer take precautions to make sure that all downstream devices have reacted and adjusted for the power switch. If the USB Hub design does not disconnect and connect any faster than 500mS, the USB Host will be able to accommodate the change state.

7.2 Host Board Layout

The host board layout should be done as indicated in figure 7-2. The USB signal traces should be two parallel lines with ground traces outside. Said signal traces should be matched in length with series 30-ohm terminating resistors as close to the 670/673 chip as possible. Ferrite bands are needed for EMI considerations and are also required for the “hot plug” requirements of USB. The board layout must comply with the applicable USB checklist.

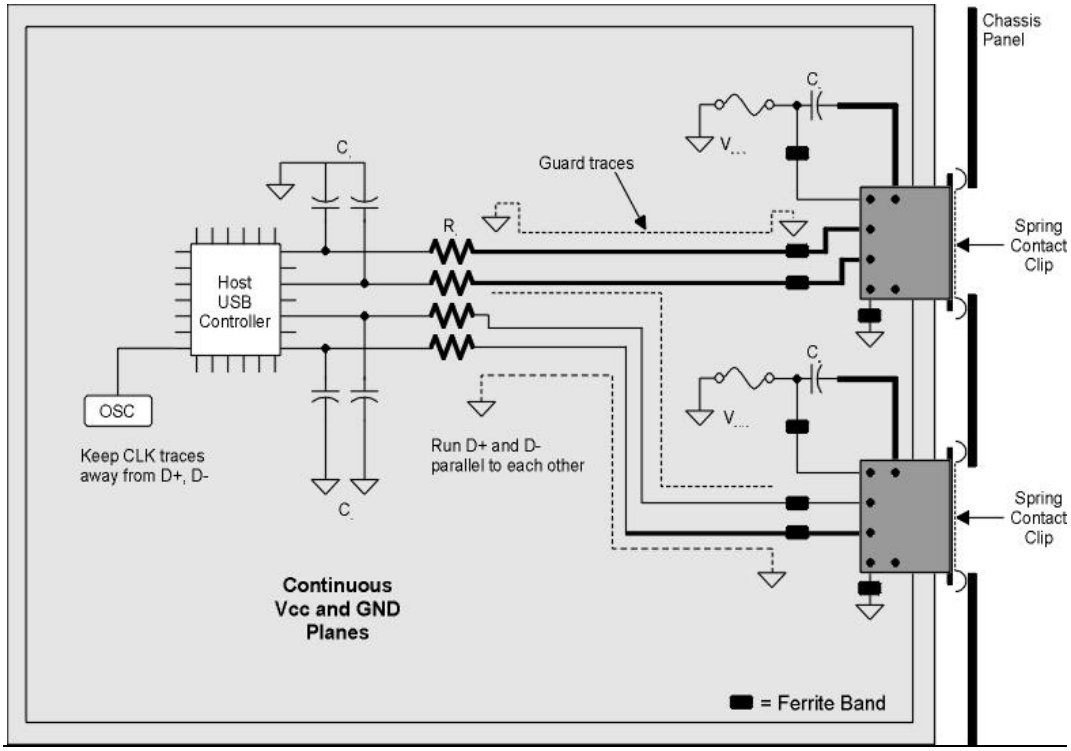


Figure 7-2: Host EMI Layout Considerations

This page left blank intentionally.

8 Power Specifications

8.1 DC Specifications

Maximum Ratings			
Symbol	**Parameter**	**Limits**	**Units**
VDD	DC Supply voltage	-0.3 to +7.0 (670) -0.3 to +4.0 (673)	Volts
Vin	DC Input Voltage	-0.3 to VDD + 0.6 (670) -0.3 to 7.3 (673)	Volts
Stg	Storage Temperature	-55 to +125	Degrees Celsius
Iin	DC Input Current	±10	❍A

Recommended Operating Conditions (VSS = 0V)				
Symbol	**Parameter**	**Min**	**Max**	**Units**
VDD	DC supply voltage	4.75 (670) 3.0 (673)	5.25 (670) 3.6 (673)	Volts
VDD3	USB DC Power	3.0	3.6	Volts
Vin	DC Input Voltage	VSS - 0.3	VDD + 0.3	Volts
Topr	Operating Temperature	0	+70	Degrees Celsius

8.2 DC Characteristics

Symbol	Parameter	Min	Max	Units	Notes
VIL	Input Voltage Low	-0.3	0.8	Volts	
VIH	Input Voltage High	2.0	5.6	Volts	Note 3
VOL	Output Voltage Low		0.55 0.40	Volts Volts	Note 1 Note 2
VOLUSB	Output Voltage USB		0.3	Volts	
VOHUSB	Output Voltage High USB	2.8	3.6	Volts	
VOH	Output Voltage High IOH = -2mA	2.4		Volts	Note 1
ICC	Standby Operating (670 Typ.) Operating (670) Operating (673 Typ.) Operating (673)		5 60 80 20 40	mA mA mA mA mA	Note 3
ILO	Output Leakage Current	-10	10	❍A	
ILI	Input High Leakage Current Vin = 2.7V	-10	10	❍A	Note 1
IIL	Input Low Leakage Current Vin = 0.5V	-10	10	❍A	Note 1
CIN	Input or I/O Capacitance		10	pF	

Notes

1. All PCI signals are PCI-compliant driver pins. See specification.
2. Non-PCI pins
3. Input pins for the USB0673 are 5V tolerant.

8.3 ASIC Power and Ground

Signal Name	Type	Description
VDD	V	USB0670: 5V power supply USB0673: 3.3V power supply
VDD3	V	USB0670: 3.3V power supply USB0673: 3.3V power supply
VSS	V	Ground

8.4 ESD Protection Characteristics

The following characteristics are for USB data pins VD1+, VD1-, VD2+ and VD2-:

Standard	Min	Units	Condition	Specification
EIAJ	200	Volts	C = 200pF; R = 0 ohm	ED4701 C-111, A
MIL	2000	Volts	C = 100pF; R = 1.5kohm	MIL-STD-883D 3015.6

This page left blank intentionally.

9 Package Dimensions

9.1 100-Pin PQFP Dimensions

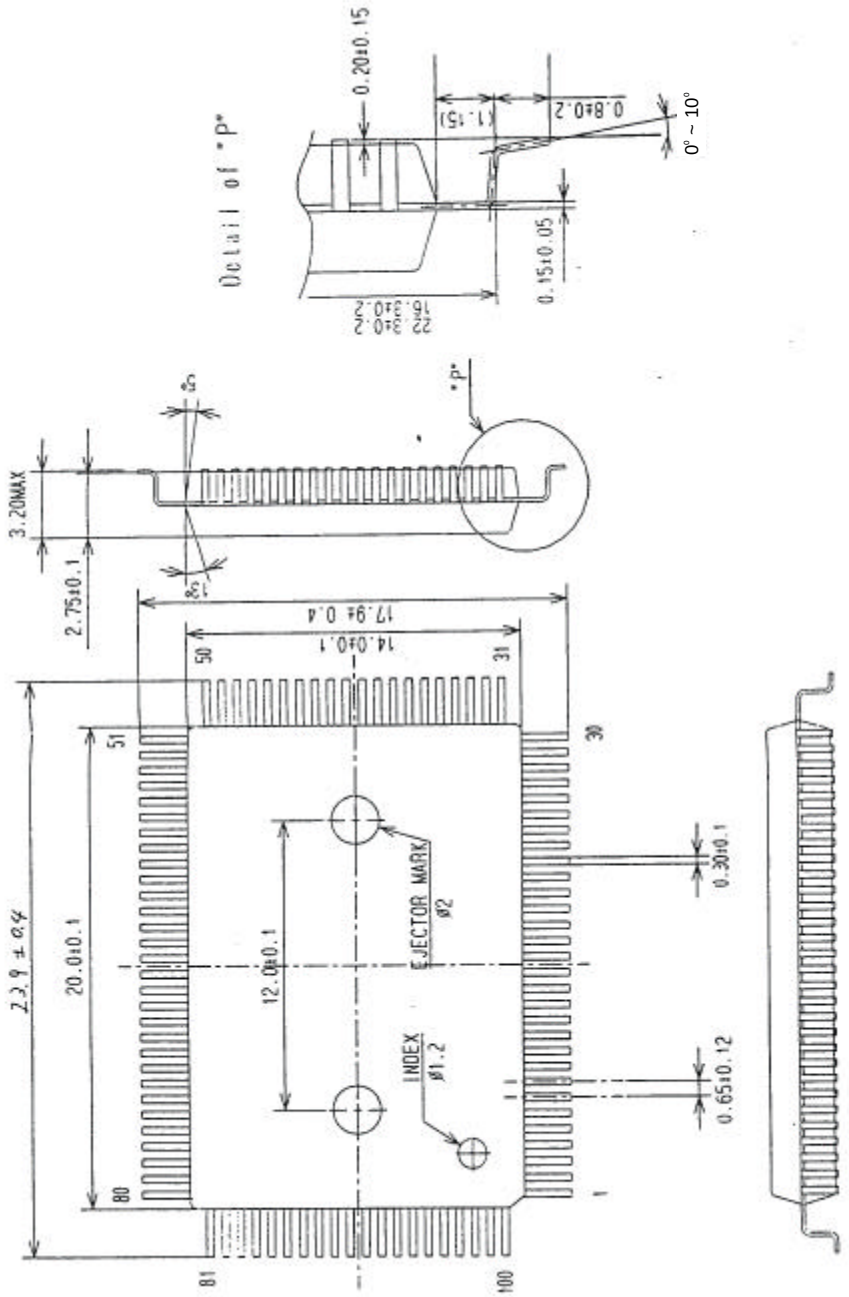


Figure 9-1: PQFP dimensions

9.2 100-Pin TQFP Dimensions

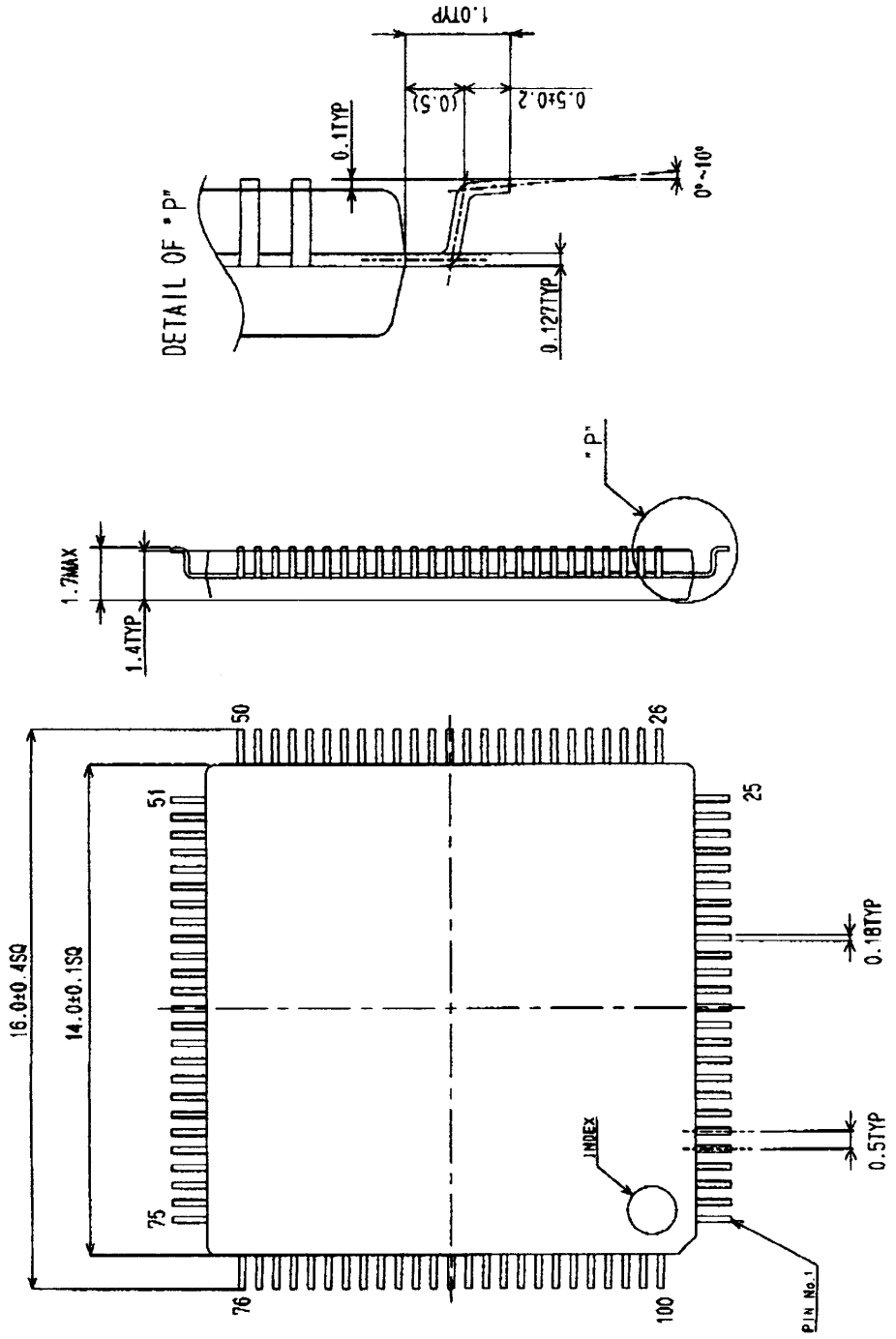
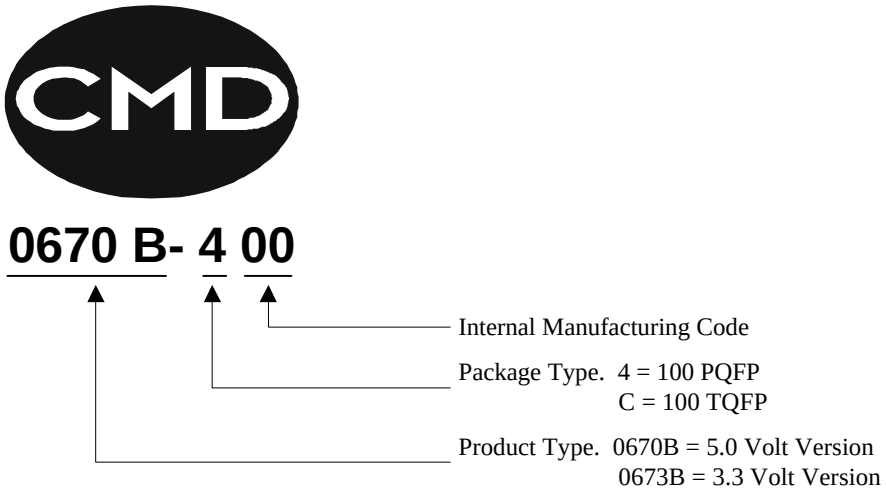


Figure 9-2: TQFP dimensions

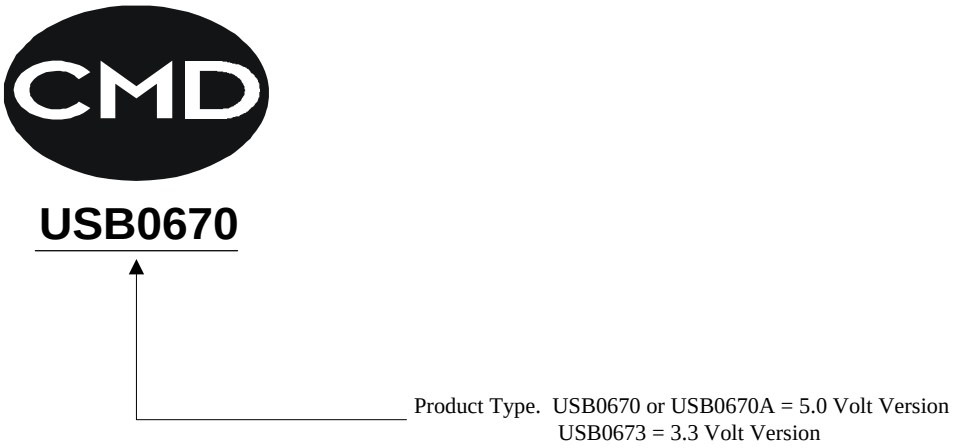
9.3 Physical Marking/Ordering Part Number

9.3.1 Physical Marking

The latest version of the PCI-to-USB Host Controller will be physically marked as follows:



Previous versions will be marked in one of two ways:



9.3.2 Ordering Part Number

Part Number	5V	3V	100 PQFP	100 TQFP
ICS-00670B-400	X		X	
ICS-00670B-C00	X			X
ICS-00673B-C00		X		X

This page intentionally left blank.

Appendix A Compliance with Microsoft's PC 97 Specification

Microsoft's PC 97 specification requires that PCI configuration headers contain non-zero Subsystem Vendor IDs. This requirement is not included in the PCI 2.1 specification.

CMD has implemented the following method to maintain compliance with the PC 97 Subsystem Vendor ID requirement:

Bit 0 of the DWORD at offset 4Ch in the PCI configuration space is designated as the read/write control bit for the Subsystem Vendor ID and Subsystem ID.

Bit 0 0 = Subsystem Vendor ID and Subsystem ID are read-only
1 = Subsystem Vendor ID and Subsystem ID are read-write

To configure the Subsystem Vendor ID and Subsystem ID, use the following steps:

1. Use a DWORD PCI configuration cycle to set bit 0 to 1.
2. Write the Subsystem Vendor ID and Subsystem ID to the DWORD at offset 2Ch.
3. Use a DWORD access to reset bit 0 to 0.

Note:
This last step is *required*. Microsoft does not allow writeable Subsystem Vendor IDs.

This page left blank intentionally.

Appendix B Bit Stuffing

The following text is taken from the Universal Serial Bus specification.

> In order to ensure adequate signal transitions, bit stuffing is employed by the transmitting device when sending a packet on the USB. A zero is inserted after every six consecutive ones in the data stream before the data is NRZI encoded to force a transition in the NRZI data stream. This gives the receiver logic a data transition at least once every seven bit times to guarantee the data and clock lock. Bit stuffing is enabled beginning with the Sync Pattern and throughout the entire transmission. The data "one" that ends the Sync Pattern is counted as the first one in a sequence. Bit stuffing on the transmitter is always enforced, without exception. If required by the bit stuffing rules, a zero bit will be inserted even if it is the last bit before the end-of-packet (EOP) signal.
>
> The receiver must decode the NRZI data, recognize the stuffed bits, and discard them. If the receiver sees seven consecutive ones anywhere in the packet, then the receiver must declare a bit stuff error and reject the packet. The time interval just before an EOP is a special case. The last data bit before the EOP can become stretched by hub switching skews. This leads to the cases illustrated in Figure B-1. The receiver must accept a packet for which there are up to six full bit times at the port with no transitions prior to the EOP. The receiver must reject a packet for which there are eight or more bits at the port with no transitions prior to the EOP. The acceptance or rejection of the packet when seven bit times are seen at the port is left as a synchronization region for the transition between differential data and single-ended zero operation and the acceptance or rejection of the packet is left to the receiver implementation.

NOTE
The PCI0670/673 will reject packets that fall within the transition region as a bit stuffing error.

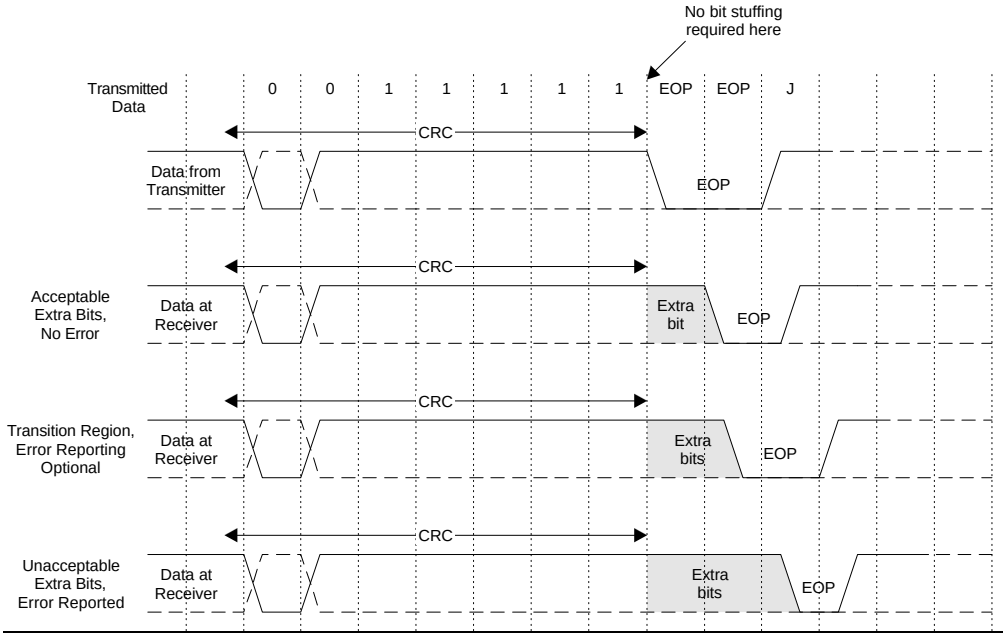


Figure B-1: Diagram of error reporting on extra bits as seen at port preceding an EOP

Appendix C Serial EPROM Support

In the USB0670 the default host configuration mode can be supported through the used of an external serial EPROM. The first 4 bytes, byte offset 0-3, are used to store the subsystem ID and subsystem vendor ID. The fifth byte, byte offset 4, is used to store the initial value of PCI configuration Space offset 0x4e. The last byte, byte offset 5, is used to store the initial value of PCI Configuration Space offset 0x4f. After Hardware reset, the host controller tries to read the Serial EPROM up to eight times. If it fails in reading the Serial EPROM after all the trials, it will set the Subsystem ID, the Subsystem vendor ID and PCI configuration Space offset 0x4e and 0x4f to their default values.

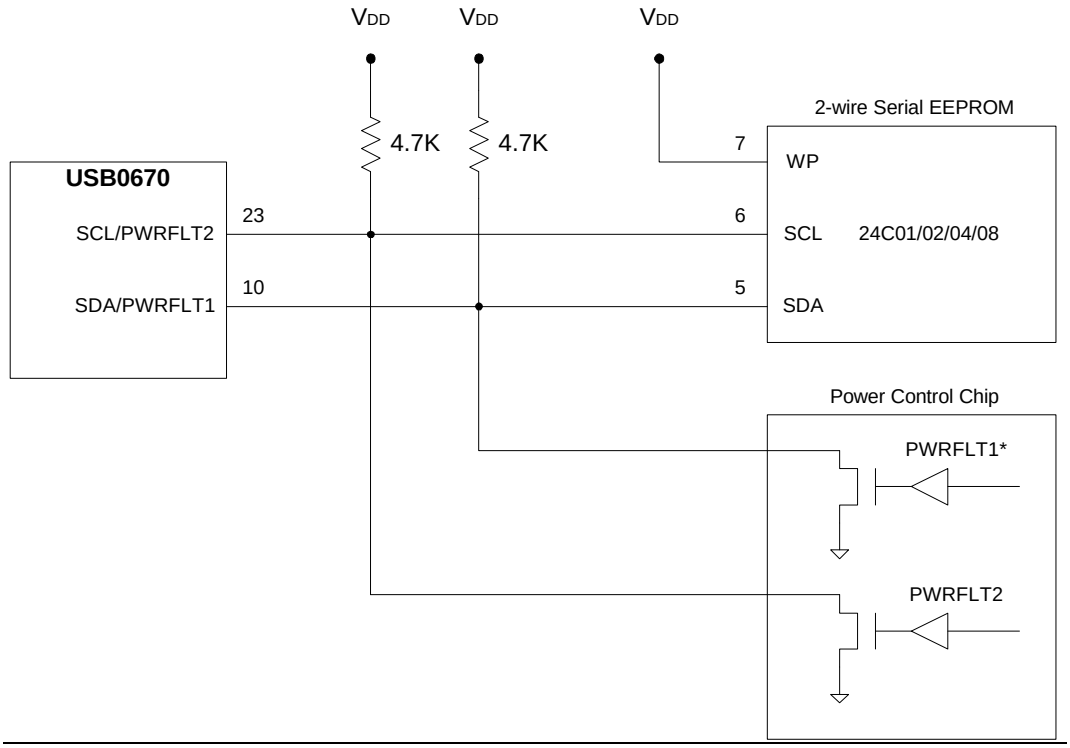


Figure C-1: Schematic diagram

* Open-drain output

NOTE
The serial EEPROM must be pre-programmed: the USB067x chip will not write data to it.