Intel 82371FB (PIIX) and 82371SB (PIIX3) PCI ISA IDE Xcelerator Specification Update

March 1998

Order Number 297658-004

The Intel 82371FB (PIIX) and 82371SB (PIIX3) may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are documented in this Specification Update.

CONTENTS

REVISION HISTORY

Date of Revision	Version	Description
May 1996	-001	This is the first revision of the 82371SB Specification Update
October 1997	-002	Incorporated 82371FB (PIIX) Items into this document. Added PIIX3 documentation changes 4 and 5, PIIX3 Specification Clarification #3, and PIIX Documentation Change #1. The PIIX3 Specification Clarification #2 was reclassified as Errata #13 Conversion to new template.
December	-003	Added PIIX3 Errata #14
March	-004	Added PIIX4 Errata #15

PREFACE

This document is an update to the specifications contained in:

82371FB and 82371SB PCI ISA IDE XCELERATOR (PIIX and PIIX3) Datasheet (Order Number 290550-002). The Datasheet also references the Universal Host Controller Interface (UHCI) Design Guide (Order Number 297650) for USB Host Controller implementation. This document will provide information on PIIX3 compliance to that guide.

Intel 430HX PCIset 82371SB PCI ISA IDE XCELERATOR (PIIX3) Timing Specification Datasheet Addendum (Order Number 272963).

It is intended for hardware system manufacturers and software developers of applications, operating systems, or tools. It contains Specification Changes, Errata, Specification Clarifications, and Documentation Changes, and is divided into the following two parts:

Part I:	Specification Update for 82371FB (PIIX)
Part II:	Specification Update for 82371SB (PIIX3)

Nomenclature

Specification Changes are modifications to the current published specifications. These changes will be incorporated in the next release of the specifications.

Errata are design defects or errors. Errata may cause the 82371FB (PIIX) and 82371SB (PIIX3), behavior to deviate from published specifications. Hardware and software designed to be used with any given stepping must assume that all errata documented for that stepping are present on all devices.

Specification Clarifications describe a specification in greater detail or further highlight a specification's impact to a complex design situation. These clarifications will be incorporated in the next release of the specifications.

Documentation Changes include typos, errors, or omissions from the current published specifications. These changes will be incorporated in the next release of the specifications.

Component Identification via Programming Interface

The 82371FB (PIIX) and 82371SB (PIIX3) may be identified by the following register contents:

PCI Register	PCI Offset	PIIX3 Value	PIIX Value
Vendor ID	00-01h	8086h	8086h
Device ID	02-03h	7000h (function 0) 7010h (function 1) 7020h (function 2)	122Eh (function 0) 1230h (function 1)
Revision Number	08h	00h (A-1 function 0) 00h (A-1 function 1) 00h (A-1 function 2) 01h (B-0 function 0) 00h (B-0 function 1) 01h (B-0 function 2)	02h (A-1 function 0) 02h (A-1 function 1)

Part I:

Specification Update for 82371FB (PIIX)

GENERAL INFORMATION

This section covers the 82371FB (PIIX).

Component Markings

82371FB (PIIX) COMPONENT MARKING INFORMATION

Stepping	S-Spec	Top Marking	Freq.	Notes
A-1	SZ964	SB82371FB, SZ964	33	Production
A-1	SZ997	SB82371FB, SZ997	33	Production
A-1p	SZ967	SB82371FB, SZ967	33	Limited Production
A-1	Q281	SB82371FB Q281	33	Engineering Samples
A-1	Q321	SB82371FB Q321	33	Engineering Samples
A-1p	Q293	SB82371FB Q293	33	Engineering Samples

Summary Table of Changes

The following table indicates the Specification Changes, Errata, Specification Clarifications or Documentation Changes, which apply to the listed 82371FB (PIIX) steppings. Intel intends to fix some of the errata in a future stepping of the component, and to account for the other outstanding issues through documentation or Specification Changes as noted. This table uses the following notations:

CODES USED IN SUMMARY TABLE

X:	Erratum, Specification Change or Clarification that applies to this stepping.
Doc:	Document change or update that will be implemented.
Fix:	This erratum is intended to be fixed in a future stepping of the component.
Fixed:	This erratum has been previously fixed.
NoFix	There are no plans to fix this erratum.
(No mark) or (Blank Box):	This erratum is fixed in listed stepping or specification change does not apply to listed stepping.
Shaded:	This item is either new or modified from the previous version of the document.

82371FB (PIIX)

NO.	A-1	A-1p	Plans	SPECIFICATION CHANGES	Documentation Update Status
				There are currently no known 82371FB (PIIX) Specification Changes.	
NO.	**A-1**	**A-1p**	**Plans**	**ERRATA**	**Documentation Update Status**
1	X		Doc	Type FDMA Writes to Main Memory don't work.	In Rev 2 Datasheet, Section 3.4.1
2		X	Doc	Pull-up Resistors Required on Specific Pins.	Described in Specification Update
NO.	**A-1**	**A-1p**	**Plans**	**SPECIFICATION CLARIFICATIONS**	**Documentation Update Status**
1	X	X	Doc	Reset Control Register (CF9) Operation.	In Rev 2 Datasheet, Section 2.5.4.3
2	X	X	Doc	Signal States During Reset.	In Rev 2 Datasheet, Section 1.15
3	X	X	Doc	BMIDE Status Register Bit Descriptions.	Identified in this document
NO.	**A-1**	**A-1p**	**Plans**	**DOCUMENTATION CHANGES**	**Documentation Update Status**
1	X	X	Doc	SERR# is an I/O (s/t/s) pin.	Identified in this document

82371FB (PIIX) SPECIFICATION CHANGES

There are currently no known 82371FB (PIIX) Specification Changes.

82371FB (PIIX) ERRATA

1. Type FDMA Writes to Main Memory Do Not Function Properly

PROBLEM: During 82430FX compatibility validation it was found that, under certain scenarios, the PIIX does not function properly when Type FDMA write cycles are enabled. When Type FDMA write cycles are enabled, the PIIX can incorrectly attempt to own the PCI bus on behalf of an ISA Master or DMA function. The PIIX functions properly on compatible mode DMA write cycles. Type FDMA *read* timings on DMA channels function properly as well.

IMPLICATION: A system using Type FDMA on write cycles to main memory can lock-up the system under certain scenarios.

WORKAROUND: Do not program Type FDMA write timings on DMA channels, use AT compatible mode timings. Type FDMA read timings on DMA channels can be used.

This has been incorporated in revision -002 of the datasheet, Section 3.4.1.

STATUS: This erratum will not be fixed. There are no future steppings of PIIX planned.

2. Missing Pullup Resistors on PIIX A-1p Results in Floating Pins

PROBLEM: The PIIX A-1p is missing internal pullup resistors on five pins. FERR#, TESTIN#, EXTSMI#, TC and DDAK0#. As a result these signals may be *floating* at reset time. The critical signals for a 430FX design will be the TC and DDAK0# signals. At the rising edge of POWEROK the TC and DDAK0# signals are sampled by the PIIX. These signals must be pulled high and sampled by the PIIX as a logic 1 for normal system operation.

IMPLICATION: Of the 5 signals listed above the TC and DDAK0# signals are typically the pins that must be pulled up. Failure to do so may result in PIIX DMA being disabled and Bus Master IDE functions being disabled. See the table below for possible issues on all five signals.

Signal	Normal Function	Issue	Board Status
FERR#	Floating-point error signal from processor.	PIIX defaults to disabling this as an input. At reset, should not generate IRQ13 to CPU.	Direct connect from CPU to PIIX. Most designs should have this connected.
TESTIN#	Test Mode	Low probability of issue here. See board status.	Designs should have this connected to V_{CC}.
EXTSMI#	Input to PIIX, *if used*, hooks to external green button.	PIIX defaults to disabling this as an input. At reset, should not send SMI# to CPU.	Designs should have this connected to V_{CC} if not used. If green button exists a pullup resistor should be used.
TC	Terminal count on DMA cycles. Bidirectional pin.	Pullup required here. A floating input can cause disabling of DMA.	Most designs will have this input floating.
DDAK0#	IDE Disk DMA Acknowledge.	Pullup required here. A floating input can cause disabling of BM IDE.	Most designs will have this input floating. Some drives may pullup this signal.

WORKAROUND: The workaround suggestions listed below are designed to allow operation with any PIIX silicon, without any additional changes on the motherboard except for those listed below.

For signals TC and DDAK0# add an external 20 K ohm resistor to each of these lines. Typically in close proximity to the PIIX. For non A-1p silicon, the nominal internal pullup value on the PIIX TC pin is 20 K ohms. The other four signals have a 50 K ohm pullup.

STATUS: This erratum was fixed in other A-1 steppings of the PIIX.

82371FB (PIIX) SPECIFICATION CLARIFICATIONS

1. Reset Control Register (CF9h) Bit 2 Operation on Resets

In order to perform a proper reset, e.g. soft reset after a soft reset, bit 2 should be cleared when writing the type of reset (bit 1 in this register) to be performed and then set again to initiate a transition from 0 to 1 on bit 2. Example: Initiate a soft reset via the CF9h Reset Control Register (write 00h then 04h). Then read the CF9h register. In order to perform another reset clear bit 2 and set it again. This has been incorporated in revision -002 of the datasheet, Section 2.5.4.3.

2. PIIX Signal States During Reset

The 82430FX PCIset ISA Bridge datasheet incorrectly states in Table 1 (column 1) that when RST# is asserted the AD[31:0] and C/BE[3:0]# lines are “high.” The correct statement is “low.”

Also in Table 1 (column 3), the signals BIOSCS#, KBCS#, and RTCCS# are incorrectly stated as “high.” The correct state for these signals is “Undefined.” This has been corrected in revision -002 of the datasheet, Section 1.15.

3. Bus Master IDE Status Register Bit Descriptions

The 82430FX PCIset ISA Bridge datasheet bit description in Section 2.7.2 that describes bit 2 and bit 0 of the Bus Master IDE Status register is incorrect. Second sentence for bit 2 should state: When bit 2=1 *and bit 1=0 and bit 0=0, i.e. 100b,* all read data from the IDE device has been transferred to main memory and all write data has been transferred to the IDE device.

Third sentence for bit 0 should state: The PIIX also sets this bit to 0 when bit 0 of the BMICOM Register is set to 0 *or when bit 1 of this register is set to 1*.

82371FB (PIIX) DOCUMENTATION CHANGES

The Documentation Changes listed in this section apply to the 82371FB (PIIX).

1. SERR# Signal Description Correction

The datasheet signal description in Section 1.1 incorrectly shows SERR# as an input pin. This pin is an I/O (s/t/s) pin. This will be reflected in future revisions of the datasheet.

Part II:

Specification Update for 82371SB (PIIX3)

GENERAL INFORMATION

This section covers the 82371SB (PIIX3).

Component Markings

82371SB (PIIX3) COMPONENT MARKING INFORMATION

Stepping	S-Spec	Top Marking	Freq.	Notes
B-0	U093	SB, 82371SB S U093	33	Production
A-1	U052	SB, 82371SB S U052	33	Production
B-0	Q501	SB, 82371SB Q 501	33	Engineering Samples
B-0	Q469	SB, 82371SB Q 469	33	Engineering Samples
B-0	Q470	SB, 82371SB Q 470	33	Engineering Samples
B-0	Q471	SB, 82371SB Q 471	33	Engineering Samples
A-1	Q449	SB, 82371SB Q 449	33	Engineering Samples
A-1	Q450	SB, 82371SB Q 450	33	Engineering Samples
A-1	Q451	SB, 82371SB Q 451	33	Engineering Samples
A-1	Q453	SB, 82371SB Q 453	33	Engineering Samples

Summary Table of Changes

The following table indicates the Specification Changes, Errata, Specification Clarifications, or Documentation Changes which apply to the listed 82371SB (PIIX3). Intel intends to correct some of the errata in a future stepping of the component, and to account for the other outstanding issues through documentation or specification changes as noted. Any items that are shaded are new for this revision of the document. This table uses the following notations:

CODES USED IN SUMMARY TABLE

X:	Erratum, Specification Change or Clarification that applies to this stepping.
Doc:	Document change or update that will be implemented.
Fix:	This erratum is intended to be fixed in a future stepping of the component.
Fixed:	This erratum has been previously fixed.
NoFix	There are no plans to fix this erratum.
(No mark) or (Blank Box):	This erratum is fixed in listed stepping or specification change does not apply to listed stepping.
Shaded:	This item is either new or modified from the previous version of the document.

82371SB (PIIX3)

NO.	A1	B0	Plans	430HX/VX	440FX	SPECIFICATION CHANGES	Documentation Update Status
						There are currently no 82371SB (PIIX3) Specification Changes.	
NO.	**A1**	**B0**	**Plans**	**430HX/VX**	**440FX**	**ERRATA**	**Documentation Update Status**
1	X		Fixed	X		ISA DMA with Delayed Transaction	Identified in this document
2	X		Fixed	X	X	USB Low Speed Driver Edge Rate	Identified in this document
3	X		Fixed	X	X	USB Stops Execution after an ISOC-OUT Transaction in Debug Mode	Identified in this document
4	X		Fixed	X	X	USB Halt bit is not Set after the Run/Stop bit is Cleared	Identified in this document
5	X		Fixed	X	X	USB Interrupt bit functionality in USB Status Register	Identified in this document
6	X		Fixed	X	X	USB Halt bit not set under specific schedule configuration	Identified in this document
7	X		Fixed	X	X	USB Incorrectly reads Frame Pointer	Identified in this document
8	X	X	NoFix	X	X	Multiple USB Babble Interrupts	In Rev 2 Datasheet, Section 3.6
9	X	X	NoFix	X	X	LS Length Babble can cause USB Lockup	In Rev 2 Datasheet, Section 3.6
10	X	X	NoFix	X	X	LS Length Babble causes USB Port disable	In Rev 2 Datasheet,Section 3.6

NO.	A1	B0	Plans	430HX/VX	440FX	ERRATA	Documentation Update Status
11	X	X	NoFix	X	X	USB Bit Stuff Error	In Rev 2 Datasheet, Section 3.6
12	X	X	NoFix	X	X	USB Status Update Delay	In Rev 2 Datasheet, Section 3.6
13	X	X	NoFix	X	X	TESTIN# External Pull-up	In Rev 2 Datasheet, Section1.13
14	X	X	NoFix	X	X	USB-PCI Latency	Identified in this document
15	X	X	NoFix	X	X	USB Resume from Selective Suspend	Identified in this document
NO	**A1**	**B0**	**Plans**	**430HX/VX**	**440FX**	**SPECIFICATION CLARIFICATION**	**Documentation Update Status**
1	X	X	Doc	HX only		INTR Signalling with IO-APIC in Virtual Wire Mode	In Rev 2 Datasheet, Section 3.10
2	X	X	NoFix	X	X	TESTIN# External Pull-up	Has been reclassified to Errata 13
3	X	X	Doc	X	X	BMIDE Status Register Bit descriptions.	Identified in this document
NO	**A1**	**B0**	**Plans**	**430HX/VX**	**440FX**	**DOCUMENTATION CHANGES**	**Documentation Update Status**
1	X	X	Doc	X	X	Revision ID (RID) Register Values	Identified in this document
2	X	X	Doc	X	X	Serial Bus Release Supported	Identified in this document
3	X	X	Doc	X	X	IRQ Level Mode Active High	In Rev 2 Datasheet, Section 3.8.2
4	X	X	Doc	X	X	SERR# is an I/O (s/t/s) pin	Identified in this document
5	X	X	Doc	X	X	MSTAT Register is 6B:6Ah, not 82h	Identified in this document

82371SB (PIIX3) SPECIFICATION CHANGES

There are currently no known 82371SB (PIIX3) Specification Changes.

82371SB (PIIX3) ERRATA

1. ISA DMA with Delayed Transactions

PROBLEM: In the case where the PIIX3 has a PHOLD# request active, and the CPU is polling a PIIX3 location (i.e. a DMA register), a data transfer limiting condition exists. The PIIX3 never gets access to the bus due to servicing delayed transactions on behalf of the CPU. The CPU will continue to poll until the DMA register changes (which never occurs), or until an interrupt stops the polling. When the interrupt acknowledge is run, PIIX3 may acquire the bus and complete the DMA cycle.

IMPLICATION: Data transfer under/over runs may occur to DMA devices on the ISA bus.

WORKAROUND: The delayed transaction feature should be disabled when using the PIIX3 A-1 stepping. The feature can be disabled by setting bit 0 at Register Offset 82H to 0.

STATUS: This was fixed on the PIIX3 B-0 stepping.

2. USB Low Speed Driver Edge Rate

PROBLEM: The edge rate for the low speed USB driver does not meet the USB specification.

IMPLICATION: This will cause excessive electromagnetic radiation on unshielded USB cables.

WORKAROUND: All systems should be “USB ready” (i.e., boards are laid out to support USB). The A-1 stepping of the PIIX3 can be used for USB development and validation testing only. Production systems shipping prior to PIIX3 B-0 stepping availability should not have a USB connector installed.

STATUS: This was fixed on the PIIX3 B-0 stepping.

3. USB Stops Execution after an ISOC-OUT Transaction in Debug Mode

PROBLEM: After an ISOC-OUT transaction occurs in debug mode, both the Run/Stop bit in the USB Command Register and the HCHalt bit in the USB Status Register get set to 0.

IMPLICATION: The USB Host Controller will stop responding when Run/Stop bit and HCHalt bit get set to 0.

WORKAROUND: The USB Host Controller needs to be reset via a global or HC reset.

STATUS: This was fixed on the PIIX3 B-0 stepping.

4. USB Halt bit is not Set after the Run/Stop bit is Cleared

PROBLEM: When the USB host controller is running, software can stop the controller by clearing the Run/Stop bit. If this is done during an ISOC OUT transaction, the controller stops but does not indicate that it has stopped via the HALT bit.

IMPLICATION: Software can not detect that the USB host controller is halted.

WORKAROUND: A work around can be implemented in software by setting the Run/Stop bit and then clearing it during a non-ISO transaction.

STATUS: This was fixed on the PIIX3 B-0 stepping.

5. USB Interrupt bit (USBINT) Functionality in USB Status Register

PROBLEM: If the USB Interrupt (USBINT) bit is set, the PIIX3 USB Host Controller will automatically reset it upon start of the next frame.

IMPLICATION: Software may not be able to recognize that USB interrupt occurred.

WORKAROUND: Software must recognize the interrupt condition prior to the start of the next frame in order to process transactions correctly.

STATUS: This was fixed on the PIIX3 B-0 stepping.

6. USB Halt bit (HCHalted) not set under specific schedule configuration

PROBLEM: The PIIX3 USB Host Controller is only checking the Run/Halt condition of USB upon fetch of a transfer descriptor or queue head with the “T” bit (bit 0, TD Link Pointer) set. If software writes a 0 to the Run/Stop (RS) bit in USB Command Register to halt processing, the halt condition would not be recognized until transfer descriptor fetch or fetch of a queue head with the “T” bit set. If a schedule is set up which has no transfer descriptors and recursive (i.e., points to itself) queue heads, then the host controller will never recognize the halt condition.

IMPLICATION: Software can not halt the USB host controller.

WORKAROUND: If a schedule has been set up with a recursive queue head, software should set the “T” bit within that queue head prior to writing a 0 to the RS bit in the USB Command Register to halt the host controller.

STATUS: This was fixed on the PIIX3 B-0 stepping.

7. USB Incorrectly Reads Frame Pointer

PROBLEM: The PIIX3 USB Host Controller can be interrupted between reading a queue head and reading it’s linked transfer descriptor by an internal signal used to signal an upcoming End of Frame condition. This subsequently causes the PIIX3 to read four DWORDs from the frame pointer list at the beginning of the next frame, instead of a single DWORD. This error also causes the host controller to stop fetching data from memory at a later time thus effectively halting all transfers on the USB.

IMPLICATION: An incorrect frame order can appear on the USB bus, with subsequent halt of USB transactions.

WORKAROUND: Software should set up the USB schedule such that all USB transactions in the frame will complete well before the End of Frame. This will prevent the host controller from having its transfer descriptor fetch interrupted.

STATUS: This was fixed on the PIIX3 B-0 stepping.

8. Multiple USB Babble Interrupts

PROBLEM: If a babble occurs in a frame, a hardware interrupt will be generated at the end of the frame. Additionally, the interrupt will occur in empty frames which immediately follow the frame in which the babble occurred and also in the first non-empty frame. An empty frame is defined as one with no Active transfer descriptors (TDs).

IMPLICATION: Extra hardware interrupts can be generated to the system.

WORKAROUND: There is no recommended work around for this errata. These interrupts will be treated as spurious interrupts by the HCD interrupt handler.

STATUS: There are currently no plans to fix this errata in the PIIX3. It is documented in the revision -002 of the Datasheet, Section 3.6.

9. Low Speed Length Babble can cause USB Lockup

PROBLEM: A low speed length babble occuring in a frame followed by a subsequent clearing of the Run/Stop bit will cause the host controller to lock up from which it can only be restarted by a hardware reset. A length babble occurs when a device transmits more than the maximum number of bits as specified in the transfer descriptor. A EOF babble occurs when a device transmits past the frame EOF time point. This condition does not exist if an EOF babble occurs.

IMPLICATION: USB host controller can not be restarted, disabling the system from performing any further USB transactions.

WORKAROUND: A software workaround can prevent a length babble from being recognized by the host controller for low speed transactions. The software will then be responsible for detecting the occurance of a low speed length babble. The host controller detection of low speed length babble is prevented by setting the Max Packet Length field in each low speed transfer descriptor to a value greater than the actual number of low speed bytes that can be sent in a frame (approximately 200). The software will detect the babble by comparing the Actual Length recorded in the transfer descriptor with a value representing the true Max Packet Length for the endpoint.

STATUS: There are currently no plans to fix this errata in the PIIX3. It is documented in the revision -002 of the Datasheet, Section 3.6.

10. Low Speed Length Babble will cause USB Port Disable

PROBLEM: If a low speed length babble (see description in Errata 9 above) occurs in a frame, the root hub through which it is attached will be disabled. This is a normal condition for a babble which exceeds past the EOF point, but is an incorrect condition for length babbles occurring within a frame.

IMPLICATION: The USB port which is disabled will not receive subsequent packets until port is reenabled by software.

WORKAROUND: See recommendation for Errata 9.

STATUS: There are currently no plans to fix this errata in the PIIX3. It is documented in the revision -002 of the Datasheet, Section 3.6.

11. USB Bit Stuff Error

PROBLEM: A bit stuff error occurs on a USB transaction transferring data from the USB device to the PIIX3. The bit stuff occurs, but with wrong data value (extra stuff bit was a 1 instead of 0). The data and CRC value are transferred correctly with no errors. The PIIX3 will detect the bit stuff error, discard the data, clear the Active bit, and set the Bit Stuff Error bit. The PIIX3 will incorrectly acknowledge (ACK) the transfer to the USB device, will not update the queue header, and does not set the Stall bit. This condition can occur if the USB device inserts the wrong stuff bit or if noise on the USB subsystem causes only this single bit to change.

IMPLICATION: The USB device's data toggle bit will become out of "sync" with the host schedule's data toggle bit. This will effectively result in a stalled USB device endpoint, halting data transfers to or from the USB device.

WORKAROUND: The USB host controller software (UHCI drivers) must check for a set Bit Stuff Error bit with no corresponding Stall bit set or simply check for an inactive transfer descriptor with a non-zero status field. When this condition is detected, the USB host controller software must stall the device endpoint and reinitialize the USB device.

STATUS: There are currently no plans to fix this errata in the PIIX3. It is documented in revision -002 of the Datasheet, Section 3.6.

12. USB Status Update Delay

PROBLEM: If PCI latency at the end of a USB transaction pushes the status update beyond the start of the next frame, then the Babble, NAK, and Timeout error condition detection may be delayed by one frame.

IMPLICATION: No end user visible effect on functionality or performance. An alternative method of status monitoring in software is necessary for immediate detection.

WORKAROUND: Each of the 3 conditions can be detected as described below.

1. NAK Status: If the Active bit is set then no patch is needed as the transaction will be automatically retried.
2. Babble Status: If Error Count <> 0, which precludes CRC/Timeout conditions and no other status bits are set other than the Stalled bit, precluding Bit Stuff & Data Buffer conditions, then treat this Babble condition like a Stall.
3. Timeout Status: If the Stalled bit is set and Error Count = 0 AND either there is no Data Buffer error or there is no Bitstuff error, then treat this condition as a CRC/Timeout error.

STATUS: There are currently no plans to fix this errata in the PIIX3. Software modifications are expected in the next commercial release of related driver. This errata is documented in revision -002 of the Datasheet, Section 3.6.

13. External Pull-Up Recommended on TESTIN#

ISSUE: The 82371SB (PIIX3) TESTIN# pin contains and internal pull-up, as disclosed in the datasheet. This pull-up is disabled if TESTIN# is sampled LO at reset. Designs without an external pull-up on TESTIN# may not float high enough to be sampled HI and therefore not enable the internal pull-up. In this event, the PIIX3 may go into testmode at reset.

IMPLICATION: Systems may not boot properly if they do not have an external pull-up on TESTIN#. Another implication is that placing an oscilloscope probe on TESTIN#, for example during signal analysis or debug activity, may pull this signal down enough to enable the testmode, and thus prevent proper operation of the system. This is an expected behavior.

WORKAROUND: On the PIIX3 TESTIN# pin, provide a weak external pull-up to the 5V power-supply, with a 4.7K to 20K ohm resistor.

STATUS: This errata will not be fixed. The requirement for the pull-up is currently identified in revision -002 of the PIIX/PIIX3 Datasheet.

14. USB-PCI Latency

PROBLEM: Under certain circumstances, PIIX3 will start an isochronous USB transfer when there is not enough time to successfully complete the transaction.

IMPLICATION: This failure only occurs when some PCI devices introduce large (>15usec) latencies on the PCI bus in combination with the USB transfer. In this situation, the USB port shuts down and requires the user to unplug the device, then plug it back in to get the device operational again. The rest of the system will continue to operate normally.

WORKAROUND: In all cases found to date, the software drivers of the PCI devices causing large delays can be modified to reduce the latency to less than 15usec. When the PCI delays are reduced to this level the isochronous USB transfers will operate normally.

STATUS: There are no plans to fix this erratum.

15 USB Resume from Selective Suspend

PROBLEM: A USB resume sequence signaled by a downstream device, from the PiiX3, may not be properly detected by the PiiX3 if the USB clock is running and the USB port is in a Selective Suspend mode. A combination of VCRS level and device speed (HS/LS) may allow the PiiX3 to detect a SE1 level on a USB clock edge which the PiiX3 resume detect hardware cannot recognize.

Symptoms include either HC responds to downstream J to K transition by driving K state, but does not set PORTSC[Resume_Detect], or the HC does not respond to downstream J to K transition by driving K state back onto the cable. These symptoms will manifest themselves as either the PIRQD interrupt will not assert and not interrupt or wake the system, or polling of PORTSC will never return a detect response and the K state will remain driven by the HC and locked up.

IMPLICATION: If the system is in a state where USB clocks are running, such as the normal state, and the USB port is in Selective Suspend mode, a resume attempt initiated by the USB device, such as a keyboard, may not be detected and the suspended port may not resume. This failure to resume will prevent normal operation of the affected USB device, and if in a power managed state where USB clocks are still running, the system may not be awaken. In this case, the user will have to awaken the system another way and may have to un-plug and re-install the USB device to get it to work.

WORKAROUND:

1) Ensure that USB peripheral devices do not support remote wake-up (peripheral workaround), or

2) Do not use the Selective Suspend feature of the PiiX3, use only Global Suspend (OS workaround).

STATUS: This will not be fixed in the PiiX3.

82371SB (PIIX3) SPECIFICATION CLARIFICATIONS

1. INTR Signaling with Pentium® Processor Local APIC in Virtual Wire Mode

The Pentium® processor with a Local APIC enabled in Virtual Wire (also called Through Local) mode requires a minimum deassertion time on the INTR signal. The PIIX3 asserts INTR asynchronously in a method compatible with the 8259A Programmable Interrupt Controller, which does not guarantee this minimum deassertion time.

This only affects Pentium processors which have a Local APIC and the Local APIC is in Virtual Wire mode of operation. This results in the following system impacts:

- Pentium Uni-processor system with no IO-APIC: No impact since the Pentium processor Local APIC must be placed in Bypass mode of operation. This includes 430VX systems with PIIX3 and 430HX systems with PIIX3 but no IO-APIC.
- Pentium uni-processor or dual processor with IO-APIC: Affects systems with use Local APIC in Virtual Wire mode of operation. This can include 430HX systems with PIIX3 and an IO-APIC. See recommendations below.
- Pentium Pro (uni-processor or dual processor): No impact as the Pentium Pro processor INTR signal does not require an INTR deassertion. This includes 440FX systems with PIIX3.

The system BIOS should incorporate one of the following recommendations:

1. In Dual processor systems with only a single processor installed, the Local APIC should be disabled (placed in Bypass or Masked Mode). If a second processor is later installed, the multiprocessing operating system may need to be reinstalled.
2. To overcome the minimum deassertion requirement in dual processor systems with both processors installed, set the Pentium processor register TR12 bit 14 to '1', similar to workaround 4 of erratum 9AP in the *Pentium® Processor Specification Update Part II*. This solution should be used in systems whose software uses interrupt gate or task gate interrupt handling. This solution should not be used if the system has software which uses trap gate interrupt handling. Systems which use trap gate handling should use method 3 below.
3. In dual processor systems with both processors installed, the IO-APIC can be placed into Virtual Wire model of operation via the IO-APIC. This solution can result in increased system interrupt handling latency times.

2. External Pull-Up Recommended on TESTIN#

This has been reclassified as Errata 13.

3. Bus Master IDE Status Register Bit Descriptions

The datasheet bit description in Section 2.7.2 that describes bit 2 and bit 0 of the Bus Master IDE Status register is incorrect. The second sentence for bit 2 should state: When bit 2=1, and bit 1=0, and bit0=0, ie 100b, all read data from the IDE device has been fully transferred to main memory and all write data has been transferred to the IDE device.

82371SB (PIIX3) DOCUMENTATION CHANGES

The Documentation Changes listed in this section apply to the 82371SB (PIIX3).

1. PCI Revision ID (RID) Register Values

CHANGE: The RID register (PCI offset 08h) values for functions 0, 1 and 2 are shown below:

FUNCTION	STEPPING	
	A-1	B-0
0	00h	01h
1	00h	00h
2	00h	01h

STATUS: Not updated in Data Sheet. This is the standard reference document.

2. Serial Bus Release Supported

CHANGE: The SBRNUM register (Function 2, offset 60h) values are for A-1 Stepping Pre-release 1.0 (00h) and for B-0 Stepping Release 1.0 (10h).

STATUS: Not updated in Datasheet. This is the standard reference document.

3. IRQ Level Mode Active High

CHANGE: Section 8.3.2, Edge and Level Triggered Mode, of the 82371SB PIIX3 datasheet incorrectly describes the interrupt requests as active low when in level mode. The interrupt requests are active high when in level mode.

STATUS: This has been updated in revision -002 of the Datasheet, Section 3.8.2.

4. SERR# Signal Description Correction

The datasheet signal description in Section 1.1 incorrectly shows SERR# as an input pin. This pin is an I/O (s/t/s) pin. This will be reflected in future revisions of the datasheet.

5. The MSTAT Register is Incorrectly Identified

The description for bit 6 of the SMIREQ Register in Section 2.2.22, incorrectly identifies the MSTAT Register as being at location 82h. This MSTAT register is at 6B and 6A, as correctly identified in Section 2.2.12 that describes the MSTAT register.